



GRADO EN ESTADÍSTICA

— TRABAJO FIN DE GRADO —

*Diseño y desarrollo de una
aplicación software didáctica
para el aprendizaje en el
ámbito del Grado en Estadística*

Autor: Samuel Pérez Tinoco

Tutor: Luis Valencia Cabrera

Sevilla, junio de 2023

Índice general

Prólogo	III
Resumen	V
Abstract	VI
Índice de Figuras	VII
Índice de Tablas	IX
1. Introducción	1
1.1. Motivación	1
1.2. Enfoque	2
1.3. Estructura de la memoria	2
2. Preliminares	5
2.1. Técnicas estadísticas y matemáticas	5
2.1.1. Primer curso	5
2.1.2. Segundo curso	16
2.1.3. Tercer curso	32
2.1.4. Cuarto curso	38
2.2. Herramientas software	39
3. Aplicación desarrollada	43
3.1. Esqueleto de la aplicación	44
3.2. Esquema general	47
3.3. Descripción curso a curso	50
3.3.1. Primer curso	51
3.3.2. Segundo curso	62
3.3.3. Tercer curso	66
3.3.4. Cuarto curso	68
4. Conclusiones	71
4.1. Aportaciones	71
4.2. Reflexiones	72
4.3. Trabajo futuro	72
A. Apéndice: Tablas	75
A.1. Ajuste hiperparámetros regresión grid aleatorio	75
A.2. Ajuste hiperparámetros clasificación	78
B. Apéndice: Código R de la aplicación	81
B.1. AplicacionEstadística.Rmd	81

B.1.1.	IMPORT	102
	B.1.1.1. app.R	102
B.1.2.	CP	102
	B.1.2.1. index.R	102
	B.1.2.2. comb.R	102
	B.1.2.3. mod.R	108
	B.1.2.4. vect.R	125
B.1.3.	ME	129
	B.1.3.1. app.R	129
B.1.4.	AIE	129
	B.1.4.1. app.R	130

Bibliografía **152**

Prólogo

Esta memoria de Trabajo Fin de Grado tratará de transmitir la labor teórico-práctica que he llevado a cabo a lo largo de todo el curso, repasando mi viaje por todo el grado y desarrollando una aplicación de utilidad para estudiantes que, como yo, estén emprendiendo o emprendan este camino.

En general, el diseño y desarrollo de una aplicación software didáctica para el aprendizaje en el ámbito del Grado en Estadística, es un proceso complejo que requiere un enfoque cuidadoso y metódico, que puede tener un impacto significativo en la educación y en la experiencia de aprendizaje de los estudiantes. De esta forma, el enfoque esencial de la aplicación es que ayude a promover y potenciar el aprendizaje y la comprensión de los conceptos fundamentales de la estadística.

Agradecer a mi familia y tutor de este trabajo por estar en los buenos y no tan buenos momentos elaborando la aplicación y la memoria.

Resumen

En este proyecto se plantea como objetivo fundamental el de proporcionar una herramienta útil para estudiantes e interesados en el aprendizaje de los aspectos fundamentales relacionados con el grado en estadística.

Para la creación de una aplicación hay que tener bien definidas cual va a ser su funcionalidad y la necesidad de la aplicación además de contar con una buena base de conocimientos que permitan llevarla a cabo. Durante el presente curso he realizado un viaje completo a lo largo de los cursos que he ido pasando a través de toda la carrera, analizando aspectos de interés dentro de cada uno, y planteando el diseño de diversas herramientas que pudieran recoger una muestra significativa de funcionalidades del grado.

Esta memoria contiene los aspectos funcionales y las cuestiones que llevan a crearla así como las herramientas necesarias, una breve descripción del contenido de la aplicación, la programación de distinguidos objetos de las que está compuesta la aplicación, unas conclusiones personales, reflexiones y un trabajo futuro.

Abstract

In this project, the main objective is to provide an useful tool for students, and those interested in learning the fundamental aspects related to the statistics degree.

In order to create an app, it is necessary to know really well the definition of the app's function and its need, as well as having good knowledge of the base that will allow it to be carried out. During this course, I have gone through a complete journey and taken all the courses of the degree, analyzing aswell interesting aspects within each one and designing various tools that could encompass a significant set of different functionalities of the degree.

This report includes the functional aspects and the issues that led to its creation, the necessary tools, a brief description of the application's content and the programming of distinguished objects which make it up, personal conclusions, reflexions and future work.

Índice de figuras

3.1. Accediendo a plantilla flexdashboard 1ª parte.	44
3.2. Accediendo a plantilla flexdashboard 2ª parte.	45
3.3. Plantilla flexdashboard.	45
3.4. Plantilla flexdashboard interactiva.	46
3.5. Esqueleto de la aplicación.	47
3.6. Barra de navegación.	48
3.7. Pestaña Inicio. Opciones de menú incluidas.	48
3.8. Galería de imágenes vista usuario.	48
3.9. Código R de galería de imágenes.	49
3.10. Pestaña 1 ^{er} curso. Opciones de menú incluidas.	50
3.11. Pestaña 2 ^o curso. Opciones de menú incluidas.	50
3.12. Pestaña 3 ^{er} . Opciones de menú incluidas. curso.	50
3.13. Pestaña 4 ^o . Opciones de menú incluidas. curso.	50
3.14. Ventana tras clic del segundo botón del ratón en la aplicación.	51
3.15. Vista previa del diseño de los n ^o combinatorios.	52
3.16. Código R para incluir latex en shiny.	53
3.17. Código R de botón que oculta contenido, iniciándose en oculto.	54
3.18. Vista previa del diseño de los n ^o combinatorios cambiando el parámetro.	55
3.19. Código R para obtener salida dinámica.	55
3.20. Código R para cambiar la clase de un textInput.	56
3.21. Vista previa del diseño de los modelos aleatorios.	57
3.22. Vista previa del diseño de función de probabilidad.	58
3.23. Código R para obtener graficos interactivos y manipular archivos externos.	59
3.24. Vista previa del diseño de vectores aleatorios.	60
3.25. Vista previa del diseño de vectores aleatorios importando archivo.	61
3.26. Código R para obtener captura de pantalla.	61
3.27. Vista previa del diseño de muestreo estadístico. Parte 1.	62
3.28. Vista previa del diseño de la asignatura de muestreo estadístico. Parte 2.	63
3.29. Vista previa del diseño de muestreo estadístico. Parte 3.	64
3.30. Código R de botón para descargar de datos.	65
3.31. Código R de aviso de notificación.	65
3.32. Código R para obtener color de fondo.	66
3.33. Vista previa del diseño de ampliación de inferencia estadística. Parte 1.	67
3.34. Vista previa del diseño de ampliación de inferencia estadística. Parte 2	68
3.35. Vista previa del diseño de inteligencia artificial y estadística.	70

NOTA: Todas las imágenes son de elaboración propia

Índice de tablas

A.1. Métricas rsq con grid aleatorio. Modelo regresión	76
A.2. Métricas rmse con grid aleatorio. Modelo regresión	76
A.3. Métricas rsq con grid regular. Modelo regresión	76
A.4. Métricas rmse con grid regular. Modelo regresión	77
A.5. Métricas con grid aleatorio. Modelo clasificación	79
A.6. Métricas con grid regular. Modelo de clasificación	79

Capítulo 1

Introducción

Este primer capítulo establecerá el propósito del trabajo, continuará con la perspectiva que se ha seguido para ir conduciendo el mismo, y terminará con la estructura de la memoria.

1.1. Motivación

El diseño y desarrollo de una aplicación software didáctica para el aprendizaje en el ámbito del grado en estadística surge de la necesidad por parte del alumnado del grado de contar con el apoyo externo, de forma que sabiendo comprender el problema planteado, brinde una solución rápida, eficiente y correcta. A lo largo de mi formación en el grado, he venido observando que en casi todas las asignaturas hay boletines de problemas en las que en ocasiones traen alguna solución. No obstante, por lo general, el llegar a ellas requiere aprender algún tipo de programación o recurrir a aplicaciones de pago (como SPSS), además de quedar bastantes problemas sin resolver. No se pretende reemplazar en ningún caso el aprendizaje requerido en el alumno en las distintas asignaturas, sino el dotarle de una herramienta con la que pueda contrastar que ha realizado bien los ejercicios y problemas planteados y poder reforzar y complementar su aprendizaje con la guía proporcionada por el software desarrollado en el presente trabajo. Esta aplicación web viene como herramienta para el aprendizaje de forma gratuita, ayudando a llegar a las soluciones que aportan esos boletines y comprobando de hecho que esas soluciones son correctas. Además proporcionará al alumno la satisfacción personal de ser capaz de resolver el problema correctamente, a dar una solución a los problemas que están sin resolver, evitando la necesidad de programar cada ejercicio para obtener los resultados, lo que es interesante para su aprendizaje pero no como requisito indispensable y para todo problema de toda asignatura, lo que resultaría prácticamente inviable por restricciones temporales. Los boletines de ejercicios prácticos que se ven en las asignaturas son posibles casos que se pueden presentar en las empresas, y por tanto puede resultar también de interés a aquellas que tengan el mismo tipo de problema que los que se realizan en las asignaturas del grado en estadística.

Adicionalmente hay que tener en cuenta que el volumen de tareas que se puede llegar a realizar en una sola asignatura puede llegar a ser incontable, tanto por parte del alumno como por parte del profesorado. Si a esto hay que sumarle que las matrículas de los alumnos no suelen componerse de una sola asignatura, sino de varias, y que un profesor

imparte más de una asignatura, aumentando con ello sus responsabilidades, además de los compromisos laborales, sociales y/o familiares que se tienen, puede que haya momentos en que uno se encuentre agotado y abrumado mentalmente. Es por ello también que esta aplicación viene aportando soluciones rápidas, eficientes y válidas, dotándola de conceptos que se ven en las asignaturas, facilitando así el estudio y la comprensión de las mismas

Otro de los motivos que me han llevado también a la creación de esta aplicación es a terminar de afianzar bien aquellos conceptos que se me quedaron en el aire cuando estaba cursando las asignaturas, como por ejemplo la interpretación gráfica de las funciones y distribuciones de probabilidad, que no es más que la visualización del comportamiento probable de los datos.

Afrontar este trabajo me ha servido por tanto como una buena recapitulación sobre todo el recorrido realizado a lo largo de estos años y ser consciente del crecimiento experimentado, al revisar con mucha mayor madurez aspectos que cuando me hallaba debajo de un árbol en el centro del bosque de la carrera me era imposible ver con perspectiva.

1.2. Enfoque

A lo largo de este trabajo, se ha mantenido en todo momento un planteamiento que tenga presente los siguientes objetivos:

- El primero es el repaso de todo el grado, que me ha servido para refrescar y asentar todos los contenidos, madurarlos y emplearlos de forma efectiva al desarrollar la aplicación. Como por ejemplo el concepto de la interpretación gráfica de las funciones y distribuciones de probabilidad anteriormente comentada.
- El segundo es el trasfondo del diseño de la aplicación que la he desarrollado poniéndome en el lugar de estudiantes y profesores los cuales pueden sentirse abrumados ante la cantidad de tareas que deben realizar en múltiples asignaturas. Por tanto he buscado ofrecer una solución que simplifique los conceptos importantes de las asignaturas, de manera que ayude a facilitar el aprendizaje y la comprensión de los mismos así como hacerla intuitiva, moderna, accesible para todos los usuarios, al alcance de los clics del ratón y didáctica.
- El tercero es la realización de esta memoria, que tiene el firme propósito de ilustrar el uso de una muestra de las funcionalidades con las que se ha desarrollado la aplicación que pueden ser de interés para otras aplicaciones. Es importante destacar la palabra muestra anteriormente mencionada en este párrafo ya que ya que la cantidad de funcionalidades proporcionadas en la aplicación daría una extensión excesiva e introduciría elementos tecnológicos repetitivos (aunque con funcionalidades diferentes), teniendo quizá menos interés para el lector/a.

1.3. Estructura de la memoria

La memoria parte de una introducción que es clave para entender porqué, para qué y para quién se ha creado, es decir, se desarrolla la necesidad de la aplicación. Es la parte más humana del proyecto, tratando de transmitir el espíritu detrás del proyecto planteado, acercando nuestra visión al usuario o lector, de modo que se la acerquemos lo más posible, como si la hubiera creado él mismo o ella misma.

A continuación, con el capítulo 2 se introducen aspectos preliminares para entrar en materia sentando algunas de las bases de técnicas estadísticas y matemáticas que se han aplicado a lo largo del trabajo, así como las principales herramientas y paquetes empleados para desarrollar la aplicación proporcionada.

El capítulo 3 presenta el núcleo del trabajo realizado a lo largo de todo el curso académico como parte de este TFG, describiendo la aplicación desarrollada y comentar aspectos que se han programado que pueden resultar de interés, mostrando así códigos de pequeñas aplicaciones que componen a la aplicación estadística en lugar de redactar todo el contenido de la aplicación, pues se desarrollaría una memoria con alta carga de almacenamiento debido a su extensión, además de la carga mental que supone.

El capítulo 4 está compuesto por la conclusiones personales y el trabajo futuro que deja la aplicación, planteando la posibilidad de continuar este arduo proyecto.

Por último, se encuentra el apéndice que contiene todos los archivos de códigos con los que se ha desarrollado la aplicación.

NOTA se han numerado aquellas ecuaciones referenciadas dentro del texto, para facilitar su seguimiento

Capítulo 2

Preliminares

Para a la creación de una aplicación, ya sea estadística o no, el primer paso es definir claramente su funcionalidad y las necesidades de crearla. Al ser una aplicación estadística, en términos de funcionalidad debemos decidir los tipos de análisis y técnicas estadísticas que se utilizarán y cómo se presentarán los resultados al usuario final. En cuanto a la necesidad que viene a cubrir, es la que se ha descrito en la introducción.

En resumen, la creación de una aplicación estadística implica definir claramente la funcionalidad y necesidades así como utilizar las herramientas y tecnologías adecuadas para implementarla y desarrollar un prototipo inicial.

2.1. Técnicas estadísticas y matemáticas

Con respecto a técnicas estadísticas, hay una amplia variedad de herramientas disponibles para la investigación y análisis de datos. Estas técnicas incluyen la estadística descriptiva, la inferencia estadística, análisis de varianza, análisis de regresión, análisis de series de tiempo, y muchos más.

En cuanto a la matemática, las herramientas matemáticas incluyen el cálculo, la teoría de conjuntos, la optimización, la teoría de la probabilidad, y otras disciplinas que son fundamentales para la estadística. Estas técnicas se utilizan para modelar y simular situaciones en las cuales no se dispone de información empírica, permitiendo a los investigadores explorar y comprender diferentes escenarios y predecir resultados futuros.

Entre la amalgama de técnicas que se podrían cubrir para satisfacer la necesidad comentada y proporcionar una batería de funcionalidades útiles para el público objetivo del proyecto, se fueron eligiendo algunos temas clave para el alumnado en cada curso del grado.

2.1.1. Primer curso

En particular, de primer curso se decidió cubrir en nuestra aplicación la asignatura de cálculo de probabilidades. Dentro de todas sus funcionalidades, vamos a describir con especial atención en esta memoria algunos fundamentos de ciertas técnicas estadísticas y matemáticas que se han expuesto en la aplicación.

Todos los conceptos que se aportan y se analizan de la asignatura del cálculo de probabilidad viene recogidos en [7], [8], [15] y [2].

En primer lugar, expondré los principios básicos de la combinatoria, así como las estructuras combinatorias más elementales.

El objetivo de la combinatoria es contar el número de elementos que tiene un conjunto finito. El cardinal de un conjunto finito $\Omega = \{\omega_1, \omega_2, \dots, \omega_N\}$ es el número de elementos que contiene y se denota $\text{card}(\Omega)$ o también $\#(\Omega)$. Por convenio, el conjunto vacío tiene cardinal 0, i.e., $\text{card}(\emptyset) = 0$. Definimos a continuación algunos principios básicos de este conjunto:

- **Principio de la adición.** Si un conjunto finito ω puede descomponerse en k partes disjuntas entre sí, A_1, A_2, \dots, A_k y tales que su unión sea todo Ω , entonces

$$\text{card}(\Omega) = \text{card}(A_1) + \dots + \text{card}(A_k)$$

- **Principio de la multiplicación.** Si los elementos de un conjunto finito Ω pueden contarse en k etapas, de tal manera que en la primera etapa haya n_1 posibilidades, en la segunda n_2, \dots , y en la k -ésima etapa n_k posibilidades, entonces:

$$\text{card}(\Omega) = n_1 n_2 \dots n_k$$

A partir de un conjunto finito $\Omega = \{\omega_1, \omega_2, \dots, \omega_N\}$ se pueden formar otros conjuntos o estructuras más complejas. Es interesante contar cuántos elementos tienen estas estructuras.

- **Secuencias** Una secuencia de longitud n es un elemento del producto cartesiano Ω^n . Las secuencias son por lo tanto de la forma

$$(a_1, \dots, a_n), \text{ con } a_i \in \Omega, i = 1, \dots, n.$$

Nótese que si dos secuencias contienen los mismos elementos pero difieren en su orden, entonces ambas secuencias son distintas. En otras palabras, el orden en que aparecen los elementos de una secuencia es importante.

Aplicando el principio de la multiplicación, el número total de secuencias de longitud n que pueden formarse a partir de un conjunto Ω con N elementos es:

$$\text{card}(\Omega^n) = N^n$$

- **Permutaciones** Una secuencia de longitud n de un conjunto finito Ω con N elementos que no contenga elementos repetidos se denomina n -permutación o permutación de longitud n . Obviamente $N \geq n$. Las n -permutaciones son pues de la forma

$$(a_1, a_2, \dots, a_n) \in \Omega^n, \text{ tales que } a_i \neq a_j, \text{ para } i \neq j.$$

Nótese que las n -permutaciones son elementos especiales (que no contienen elementos repetidos) del producto cartesiano Ω^n y por lo tanto dos n -permutaciones que contengan los mismos elementos pero en diferente orden son distintas, es decir, el orden en que aparecen los elementos en una n -permutación es importante.

Las permutaciones son las secuencias que no tienen elementos repetidos.

El número de n -permutaciones de un conjunto finito con N elementos es

$$P_{N,n} = N(N-1) \dots (N-n+1)$$

- **Subconjuntos.** Un subconjunto con n elementos de $\Omega = \{\omega_1, \dots, \omega_N\}$ es una colección de n elementos distintos seleccionados de Ω . En los subconjuntos el orden en que aparecen los elementos no es relevante, por lo tanto dos subconjuntos son iguales si y solo si contienen los mismos elementos, aunque posiblemente estén dispuestos en distinto orden. Los subconjuntos se denotan.

$$\{a_1, \dots, a_n\}, a_i \in \Omega, i = 1, \dots, n$$

- El número de subconjuntos con n -elementos de un conjunto finito Ω con N elementos es

$$\binom{N}{n} = \frac{N(N-1)\dots(N-n+1)}{n!} = \frac{N!}{(N-n)!n!}$$

- El número total de subconjuntos (propios e impropios) de un conjunto finito Ω con N elementos es 2^N

En segundo lugar, redactaré sobre los modelos de variables aleatorias discretas, los cuales permiten observar el comportamiento de los datos, es decir, la representación de las funciones de probabilidad y de distribución. Adicionalmente se ha estudiado el comportamiento de vectores aleatorios.

Para llegar a comprender qué es un modelo de variable aleatoria discreta, así como un vector aleatorio, se detallarán una serie de conceptos de antemano.

Definición 2.1.1 Una variable aleatoria (v.a.), X , es una aplicación que transforma los elementos del espacio muestral, Ω , en números reales, es decir,

$$\begin{aligned} X : \Omega &\longrightarrow \mathbb{R} \\ \omega \in \Omega &\longmapsto X(\omega) \in \mathbb{R}. \end{aligned}$$

Definición 2.1.2 Una variable aleatoria $X : \Omega \longrightarrow \mathbb{R}$ es discreta si $Im(X)$ es un conjunto numerable.

En el caso en que una variable aleatoria sea discreta, los posibles valores que toma la variable se pueden numerar. Sea pues,

$$Im(X) = \{x_1, x_2, \dots\}.$$

Definimos los subconjuntos de Ω

$$A_i = \{\omega \in \Omega : X(\omega) = x_i\}, i \geq 1.$$

Nótese que A_i , es el subconjunto de elementos del espacio muestral, Ω , que se transforman en $x_i \in \mathbb{R}$. Estos subconjuntos (sucesos) forman una partición de Ω , es decir, $A_i \cap A_j = \emptyset$, $i \neq j$ y $\bigcup_{i \geq 1} A_i = \Omega$. La partición $\{A_i\}_{i \geq 1}$ también se denomina sistema completo de sucesos asociado a la variable aleatoria X .

Los sucesos $A_i \subset \Omega$ aparecen con frecuencia en el desarrollo de la teoría, por lo que se adopta una notación más simple para ellos:

$$A_i = \{X = x_i\}, i \geq 1$$

Considerando ahora las probabilidades de los sucesos A_i :

$$p_i = \mathbb{P}(A_i) = \mathbb{P}(X = x_i), \quad i \geq 1$$

Debido a que $\{A_i\}_{i \geq 1}$ es una partición, se tiene:

- a) $p_i \geq 0, \quad i \geq 1$
- b) $\sum_{i \geq 1} p_i = 1$

Definición 2.1.3 El conjunto de todas las probabilidades $\{p_i\}_{i \geq 1}$ con $p_i = \mathbb{P}(X = x_i), \quad i \geq 1$, se denomina función (de masa) de probabilidad de la variable aleatoria discreta X .

Pueden construirse nuevas variables aleatorias mediante operaciones básicas. Concretamente:

- Si X es una variable aleatoria y g es una función, entonces $g(X)$ es una variable aleatoria. Por ejemplo, si X es una variable aleatoria, entonces también lo son X^n, e^X, \dots
- Si X e Y son variables aleatorias, también lo son: $X + Y, X - Y, XY, X/Y$ (siempre que $Y \neq 0$).

De forma general, se supondrá que se trabaja en un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$ y supondremos que tenemos definida una v.a. X discreta, que toma los valores en el conjunto $\{x_1, x_2, \dots\}$ y con función de masa de probabilidad $p_i = \mathbb{P}(X = x_i), \quad i = 1, 2, \dots$

Definición 2.1.4 La función de distribución (fdD) de una variable aleatoria X es la función

$$F_X : \mathbb{R} \longrightarrow [0, 1]$$

$$x \in \mathbb{R} \longmapsto F_X(x) = \mathbb{P}(X \leq x).$$

En particular la función de distribución de una variable aleatoria discreta es:

$$F(x) = \mathbb{P}(X \leq x) = \sum_{i: x_i \leq x} \mathbb{P}(X = x_i)$$

A continuación también aportamos algunos teoremas esenciales para sentar las bases de las funciones de distribución que se han incluido en la aplicación.

Teorema 2.1.1 La función de distribución asociada a una variable aleatoria X, F_X , verifica:

- a) F es una función no decreciente, es decir, $x < y \Rightarrow F(x) \leq F(y)$.
- b) $\lim_{x \rightarrow +\infty} F(x) = 1$ y $\lim_{x \rightarrow -\infty} F(x) = 0$.
- c) F es continua por la derecha en cada punta $a \in \mathbb{R}$, es decir,

$$\lim_{x \rightarrow a^+} F(x) = F(a).$$

Teorema 2.1.2 Mas propiedades de la función de distribución son:

- a) $\mathbb{P}(x < X \leq y) = F(y) - F(x)$.
- b) $\mathbb{P}(X > x) = 1 - F(x)$.
- c) F_X es continua en un punto x sii $\mathbb{P}(X = x) = 0$.
- d) Si F_X es discontinua en un punto x , entonces el salto de la fdD en x es precisamente $P(X = x)$.

Además se expondrá también las definiciones y teoremas que permiten sentar las bases sobre la esperanza, la varianza, y otros elementos que tienen su implicación en la aplicación.

Definición 2.1.5 La esperanza matemática de la v.a. X es:

$$\mathbb{E}(X) = \sum_i x_i \mathbb{P}(X = x_i)$$

siempre que la serie de convergente, es decir, que se pueda sumar y que sea finita.

La esperanza matemática también se denomina valor esperado o media. La interpretación física de la esperanza matemática es que ésta es el centro de gravedad de la variable aleatoria. De forma más general, si X es una v.a. discreta y $g : \mathbb{R} \rightarrow \mathbb{R}$ es una función se tiene,

$$\mathbb{E}[g(X)] = \sum_i g(x_i) \mathbb{P}(X = x_i).$$

Teorema 2.1.3 Algunas propiedades de la esperanza matemática:

- a) Si $c \in \mathbb{R}$, entonces $\mathbb{E}(c) = c$.
- b) Si $c \in \mathbb{R}$, entonces $\mathbb{E}(X + c) = \mathbb{E}(X) + c$.
- c) Si $b \in \mathbb{R}$, entonces $\mathbb{E}(bX) = b\mathbb{E}(X)$.
- d) Si X e Y son variables aleatorias, entonces $\mathbb{E}(X + Y) = \mathbb{E}(X) + \mathbb{E}(Y)$.

Definición 2.1.6 Se define, en caso de que exista,

- el momento de orden r de la v.a. X como

$$m_r = \mathbb{E}(X^r) = \sum_i x_i^r \mathbb{P}(X = x_i),$$

- y el momento central de orden r de la v.a. X como

$$\mu_r = \mathbb{E}(X - \mu)^r = \sum_i (x_i - \mu)^r \mathbb{P}(X = x_i),$$

donde $\mu = \mathbb{E}(X)$.

Definición 2.1.7 Se define la varianza de una variable aleatoria, X , como

$$Var(X) = \mathbb{E}(X - \mu)^2 = \sum_i (x_i - \mu_X)^2 \mathbb{P}(X = x_i),$$

donde $\mu = \mathbb{E}(X)$.

Definición 2.1.8 La desviación típica de la variable aleatoria X , es

$$\sigma_X = +\sqrt{Var(X)}.$$

Teorema 2.1.4 Propiedades de la varianza

- a) $Var(X) \geq 0$
- b) $Var(X) = 0$ si y sólo si X es una v.a. degenerada, es decir, se trata de una variable aleatoria que toma un único valor con probabilidad 1.
- c) (Teorema de koning) $Var(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$
- d) Sean a y b números reales, entonces $Var(aX + b) = a^2 Var(X)$
- e) Considérese la función $f(a) = \mathbb{E}(X - a)^2$, $a \in \mathbb{R}$

$$\min_{a \in \mathbb{R}} \mathbb{E}(X - a)^2 = \mathbb{E}(X - \mu_X)^2 = \sigma^2.$$

En la práctica hay situaciones experimentales que aparecen frecuentemente y se ajustan razonablemente bien a alguno de los modelos que se exponen a continuación.

- **Variable aleatoria de bernouilli:** este modelo se utiliza en el caso de experimentos con dos posibles resultados, denominados E (éxito) y F (fracaso), con probabilidades respectivas p y $q = 1 - p$. El espacio muestral es $\Omega = \{E, F\}$. La variable aleatoria de Bernouilli se define como:

$$X(E) = 1, \quad X(F) = 0.$$

La función de probabilidad asociada es:

$$p = P(X = 1), \quad q = P(X = 0).$$

Una variable aleatoria X de Bernouilli definida en un experimento donde la probabilidad de éxito es p se denota $X \sim Be(p)$.

- **Variable aleatoria binomial:** sea un experimento de tipo Bernouilli, es decir con dos posibles resultados E (éxito) y F (fracaso). Supongamos que este experimento es repetido n veces bajo las siguientes condiciones:
 - a) Las probabilidades de éxito y de fracaso son respectivamente p y $q = 1 - p$, y permanecen constantes en los n ensayos.
 - b) Los ensayos o repeticiones del experimento son independientes entre sí, lo cual quiere decir que la realización de un ensayo no afecta a las probabilidades de los futuros ensayos.

Entonces, la variable aleatoria

$$X = \text{número de éxitos en los } n \text{ ensayos,}$$

es una variable aleatoria binomial con parámetros n (número de repeticiones) y $p \in (0, 1)$ (probabilidad de éxito en cada ensayo). Se denota $X \sim Bi(n, p)$.

Si asociamos al éxito el valor 1 y al fracaso el valor 0, un elemento genérico del espacio muestral sería de la forma $\omega = (\omega_1, \dots, \omega_n)$, donde ω_i es 1 o 0 según el ensayo i -ésimo resulte ser un éxito o un fracaso, respectivamente. Nótese que la variable aleatoria X es

$$X(\omega) = \sum_{i=1}^n \omega_i.$$

Nótese que $Im(X) = \{0, 1, \dots, n\}$ es un conjunto numerable. Por lo tanto, X es una variable aleatoria discreta. La probabilidad asociada a cada suceso elemental, según los supuestos a) y b) del experimento binomial, sería:

$$\mathbb{P}(\omega_1, \dots, \omega_n) = p^{\sum_{i=1}^n \omega_i} q^{n - \sum_{i=1}^n \omega_i}.$$

El sistema completo de sucesos asociado a la variable aleatoria está formado por los sucesos:

$$A_k = \{X = k\} = \{\omega \in \Omega : \sum_{i=1}^n \omega_i = k\}, k = 0, \dots, n.$$

Obviamente A_k es el suceso correspondiente a obtener k éxitos en los n ensayos, y su cardinal es $card(A_k) = \binom{n}{k}$. La función de masa de probabilidad de la variable aleatoria X es:

$$\mathbb{P}(X = k) = \mathbb{P}(A_k) = \binom{n}{k} p^k q^{n-k}, k = 0, \dots, n.$$

- **Variable aleatoria geométrica:** supongamos un experimento de tipo Bernoulli con dos posibles resultados: E (éxito) y F (fracaso). Este experimento se repite hasta que por primera vez se obtiene un éxito. Se supone además:
 - a) Las probabilidades de éxito y de fracaso son respectivamente p y $q = 1 - p$ y permanecen constantes.
 - b) Los ensayos o repeticiones del experimento son independientes entre sí, lo cual quiere decir que la realización de un ensayo no afecta a las probabilidades de los sucesivos ensayos.

Entonces, la variable aleatoria

$$X = \text{número de fracasos hasta obtener el primer éxito,} \quad (2.1)$$

es una variable aleatoria geométrica con un sólo parámetros $p \in (0, 1)$ (probabilidad de éxito en cada ensayo). Se denota $X \sim Ge(p)$. Como en los casos anteriores notaremos $q = 1 - p$.

Nuevamente, como en el caso anterior, el conjunto de posibles valores que toma la variable aleatoria X es $Im(X) = \{0, 1, \dots\}$, que es un conjunto numerable (en este caso infinito). Por lo tanto X es una variable aleatoria discreta.

El espacio muestral asociado a un experimento geométrico es:

$$\Omega = \{E, (F, E), (F, F, E), (F, F, F, E), \dots\}.$$

La probabilidad asociada a cada suceso elemental de este espacio, de acuerdo con los supuestos a) y b), sería:

$$\mathbb{P}(\underbrace{(F, \dots, F, E)}_{k \text{ fracasos}}) = q^k p, \quad k = 0, 1.$$

Los sucesos fundamentales asociados a la variable aleatoria geométrica son:

$$A_k = \{X = k\} = \{\underbrace{(F, \dots, F, E)}_{k \text{ fracasos}}\}, \quad k = 0, 1, \dots$$

y la función de masa de probabilidad es:

$$\mathbb{P}(X = k) = \mathbb{P}(A_k) = q^k p, \quad k = 0, 1, \dots$$

Nota: Algunos autores definen la variable aleatoria geométrica de una forma diferente, como:

$$X^* = \text{número de ensayos hasta obtener el primer éxito.} \quad (2.2)$$

La variable definida en 2.2 está relacionada con la definida en 2.1 mediante la relación

$$X^* = X + 1.$$

Notaremos $X^* \sim Ge^*(p)$ a la variable que cuenta el número de ensayos hasta el primer éxito.

- **Variable aleatoria de Poisson:** la variable aleatoria de Poisson aparece con frecuencia en las aplicaciones y en la naturaleza. Fenómenos como el número de accidentes en un periodo determinado, número de llamadas a una centralita telefónica en un intervalo de tiempo, el número de partículas emitidas por una substancia radiactiva por segundo, son ejemplos de variables aleatorias que se ajustan razonablemente bien a la distribución de Poisson.

Dado $\lambda > 0$, se dice que una variable aleatoria, X , que toma valores enteros no negativos sigue una distribución de Poisson si su función de probabilidad es

$$\mathbb{P}(X = k) = e^{-\lambda} \frac{\lambda^k}{k!}, \quad k = 0, 1, \dots$$

y se denota $X \sim Po(\lambda)$. La distribución de Poisson es un caso límite de la distribución binomial: Si $X_n \sim Bi(n, p_n)$ es tal que $\lim_{n \rightarrow \infty} np_n = \lambda > 0$ entonces

$$\mathbb{P}(X_n = k) = \binom{n}{k} p_n^k (1 - p_n)^{n-k} \longrightarrow e^{-\lambda} \frac{\lambda^k}{k!}, \quad \text{si } n \longrightarrow \infty \quad (2.3)$$

La propiedad 2.3 es útil para obtener aproximaciones numéricas a probabilidades binomiales mediante probabilidades de Poisson.

- **Variable aleatoria uniforme discreta:** supongamos que un experimento tiene un espacio muestral con N elementos, $\Omega = \{\omega_1, \dots, \omega_N\}$ y que el experimento consiste

en extraer al azar, siendo todos los sucesos elementales son igualmente probables, un elemento del conjunto Ω . Se tiene:

$$\mathbb{P}(\{\omega_k\}) = \frac{1}{N}, \quad k = 1, \dots, N.$$

Definimos la variable aleatoria

$$X(\omega_k) = k, \quad k = 1, \dots, N.$$

Esta variable aleatoria toma los valores $Im(X) = \{1, 2, \dots, N\}$, que es un conjunto finito, por lo que se trata de una variable aleatoria discreta. El sistema completo de sucesos es:

$$A_k = \{X = k\} = \{\omega_k\}, \quad k = 1, \dots, N,$$

la función de masa de probabilidad es

$$\mathbb{P}(X = k) = \frac{1}{N}, \quad k = 1, \dots, N$$

y se denota $X \sim UD(1, \dots, N)$.

- Variable aleatoria hipergeométrica:** supongamos que los N elementos de un conjunto, que denominaremos población, son de dos tipos: Tipo A y Tipo B. Si hay N_1 elementos de Tipo A, es obvio que entonces hay $N - N_1$ de Tipo B. El experimento consiste en seleccionar al azar un subconjunto (muestra) de n elementos de la población, de forma que los $\binom{N}{n}$ posibles subconjuntos tengan la misma probabilidad de ser seleccionados (probabilidad uniforme). Se considera la variable aleatoria

$X =$ número de elementos de Tipo A en la muestra seleccionada.

La variable aleatoria así definida es una variable aleatoria hipergeométrica, y se denota

$$X \sim \mathcal{H}(n; N_1, N).$$

Esta variable toma un número finito de valores, por lo que es una variable aleatoria discreta. La función de masa de probabilidad de la variable X es:

$$\mathbb{P}(X = k) = \frac{\binom{N_1}{k} \binom{N - N_1}{n - k}}{\binom{N}{n}}.$$

para $\max\{0, n - N + N_1\} \leq k \leq \min\{n, N_1\}$. Otra forma habitual en la que aparece la distribución hipergeométrica es en el muestreo sin reemplazamiento en una población finita de N elementos. En dicha población se supone que hay N_1 elementos de tipo A y $N - N_1$ de tipo B. Se extraen al azar sucesivamente n elementos de la población de tal forma que tras cada extracción el elemento seleccionado no es devuelto a la población. La variable aleatoria $X =$ número de elementos de Tipo A seleccionados en las n extracciones sigue una distribución $\mathcal{H}(n; N_1, N)$.

En un mismo experimento aleatorio es usual observar más de una variable. Esa situación conduce al concepto de vector aleatorio.

Definición 2.1.9 Un vector aleatorio es un vector cuyas componentes son variables aleatorias, es decir, $\mathbf{X} = (X_1, \dots, X_n)$ es un vector aleatorio n -dimensional si y solo si cada $X_i, i = 1, \dots, n$ es una variable aleatoria.

Definición 2.1.10 Un vector aleatorio $\mathbf{X} = (X_1, \dots, X_n)$ es de tipo discreto si y solo si todas las componentes, $X_i, i = 1, \dots, n$, son variables aleatorias discretas.

Por simplificar, consideraremos el caso de vectores aleatorios bidimensionales, es decir, $n = 2$. En tal caso, el vector aleatorio se suele denotar (X, Y) , donde X e Y son variables aleatorias.

Definición 2.1.11 Un vector aleatorio bidimensional (X, Y) es de tipo discreto si tanto X como Y son variables aleatorias discretas.

Una definición equivalente sería que el conjunto imagen de (X, Y) es un conjunto numerable. Si $Im(X) = \{x_1, x_2, \dots\}$ e $Im(Y) = \{y_1, y_2, \dots\}$ entonces $Im(X, Y)$ es también numerable.

Definición 2.1.12 Sea (X, Y) un vector aleatorio bidimensional discreto tal que $Im(X) = \{x_1, x_2, \dots\}$ e $Im(Y) = \{y_1, y_2, \dots\}$. La función de probabilidad conjunta del vector aleatorio (X, Y) es el conjunto de valores

$$\mathbb{P}(X = x_i, Y = y_j) = p_{ij}.$$

Para cualquier vector aleatorio se verifica:

- a) $p_{ij} \geq 0, \forall i, j,$
- b) $\sum_i \sum_j p_{ij} = 1.$

Si (X, Y) es un vector aleatorio bidimensional, y $g : \mathbb{R}^2 \rightarrow \mathbb{R}$, entonces $Z = g(X, Y)$ es una variable aleatoria (unidimensional).

En el caso discreto, la función de probabilidad de Z puede obtenerse a partir de la función de probabilidad conjunta del vector aleatorio (X, Y) mediante la relación.

$$\mathbb{P}(Z = z) = \sum_{(i,j):g(x_i,y_j)=z} \mathbb{P}(X = x_i, Y = y_j).$$

Definición 2.1.13 Si (X, Y) es un vector aleatorio bidimensional discreto, tanto X como Y son variables aleatorias discretas. La función de probabilidad de la variable aleatoria X se denomina función de probabilidad marginal de X y está relacionada con la función de probabilidad conjunta mediante la relación

$$\mathbb{P}(X = x_i) = \sum_j \mathbb{P}(X = x_i, Y = y_j), \quad i = 1, 2, \dots$$

De forma similar, la función de probabilidad marginal de la variable aleatoria Y se obtiene mediante

$$\mathbb{P}(Y = y_j) = \sum_i \mathbb{P}(X = x_i, Y = y_j), \quad i = 1, 2, \dots$$

Definición 2.1.14 Sea (X, Y) un vector aleatorio bidimensional discreto. Entonces X e Y son independientes si

$$\mathbb{P}(X = x_i, Y = y_j) = \mathbb{P}(X = x_i) \mathbb{P}(Y = y_j), \forall i, j = 1, 2, \dots$$

Definición 2.1.15 El valor esperado de un vector aleatorio (X, Y) es el vector cuyas componentes son los respectivos valores esperados marginales, es decir $(\mathbb{E}(X), \mathbb{E}(Y))$.

Tal como se ha visto anteriormente, para una función $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ y un vector aleatorio (X, Y) , se tiene que $Z = g(X, Y)$ es una variable aleatoria. El valor esperado de Z es

$$\mathbb{E}(Z) = \mathbb{E}(g(X, Y)) = \sum_i \sum_j g(x_i, y_j) p_{ij},$$

donde $p_{ij} = \mathbb{P}(X = x_i, Y = y_j)$ es la función de probabilidad conjunta del vector aleatorio.

Teorema 2.1.5 Si (X, Y) es un vector aleatorio y $a, b \in \mathbb{R}$, entonces

$$\mathbb{E}(aX + bY) = a\mathbb{E}(X) + b\mathbb{E}(Y).$$

Definición 2.1.16

- Los momentos ordinarios de un vector aleatorio bidimensional son (caso de que existan) los números

$$m_{r,s} = \mathbb{E}(X^r Y^s) = \sum_i \sum_j x_i^r y_j^s p_{ij}.$$

En particular

$$\begin{aligned} m_{1,0} &= \mathbb{E}(X), & m_{2,0} &= \mathbb{E}(X^2), \\ m_{0,1} &= \mathbb{E}(Y), & m_{0,2} &= \mathbb{E}(Y^2). \end{aligned}$$

- Sea $\mu_X = \mathbb{E}(X)$ y $\mu_Y = \mathbb{E}(Y)$. Los momentos centrales del vector aleatorio (X, Y) son los números (caso de que existan):

$$\mu_{r,s} = \mathbb{E}((X - \mu_X)^r (Y - \mu_Y)^s) = \sum_i \sum_j (x_i - \mu_X)^r (y_j - \mu_Y)^s p_{ij}.$$

Definición 2.1.17 Dada una v.a. bidimensional (X, Y) , se define la covarianza entre X e Y , como

$$Cov(X, Y) = \mathbb{E}((X - \mu_X)(Y - \mu_Y)).$$

Nótese que la covarianza es el momento centrado $\mu_{1,1}$ del vector aleatorio (X, Y) . Una fórmula alternativa para la covarianza es

$$Cov(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y).$$

Definición 2.1.18 Se dirá que X e Y son incorreladas si $Cov(X, Y) = 0$.

Teorema 2.1.6 Propiedades de la covarianza:

- a) $Cov(X, X) = Var(X)$.
- b) X e Y son independientes, entonces X e Y son incorreladas. El recíproco de este resultado no es cierto en general, es decir, es posible encontrar variables incorreladas pero no independientes.
- c) $Var(X \pm Y) = Var(X) + Var(Y) \pm 2Cov(X, Y)$.
- d) X e Y incorreladas (o independientes)

$$Var(X + Y) = Var(X) + Var(Y).$$

- e) Si $a, b, c, d \in \mathbb{R}$, entonces

$$Cov(a + bX, c + dY) = bdCov(X, Y).$$

Definición 2.1.19 Se define el coeficiente de correlación lineal entre las variables (X, Y) por

$$\rho = \frac{Cov(X, Y)}{\sqrt{Var(X) Var(Y)}}.$$

Nótese que la varianza es el momento central de orden 2, es decir, $\mu_2 = Var(X)$.

2.1.2. Segundo curso

De segundo curso, como en el caso del primero, se tomaron y aplicaron en el trabajo un buen número de técnicas trabajadas en la asignatura de muestreo estadístico. A la hora de seleccionar algunas funcionalidades de interés para el alumnado, se optó por trabajar en la obtención de muestras con los distintos tipos de muestreo como el aleatorio o aleatorio simple, el sistemático uniforme de paso k , el diseño de bernouilli, el diseño de poisson, el diseño PPS, el diseño de midzuno y el muestreo con reemplazamiento. Además, se ha trabajado con técnicas inferenciales como la estimación para la media poblacional, para el total poblacional, para la proporción, para el parámetro razón, estimadores de razón de medias, de razón, de diferencia y de regresión.

En resumen, existen pues dos fases o procesos bien diferenciados:

- Por una parte la fase o proceso de selección de la muestra. Es la fase denominada muestreo. Usualmente emplea métodos o algoritmos basados en el azar.
- Por otra parte, la fase consistente en procesar la información de la muestra para extenderla a toda la población. Esta fase se basa en la manipulación de los datos muestrales, y se denomina genéricamente estimación o inferencia.

Todos los conceptos que se aportan y se analizan de la asignatura de muestreo estadístico vienen recogidos en [5] y [11].

A continuación definimos algunos conceptos e ideas de interés en esta sección:

- **Población** finita, es un conjunto del tipo,

$$\mathcal{U} = \{1, 2, \dots, N\}.$$

- Sus elementos son denominados **unidades poblacionales**.
- El número N se denomina **tamaño de la población** o **tamaño poblacional**.
- Una característica cualquiera bajo estudio será representada matemáticamente por una variable, Y , que llamaremos **variable de estudio**. Los valores que toma dicha variable sobre las unidades poblacionales son desconocidos de antemano. Para denotarlos emplearemos la siguiente notación:

$$Y = (y_1, y_2, \dots, y_N)$$

siendo y_i el valor de la variable de estudio para la unidad poblacional i .

- **Censo** Investigación exhaustiva de todos y cada uno de los individuos de la población.

Usualmente, al ser muy grandes las poblaciones, los censos son muy costosos tanto en recursos económicos y humanos, como en tiempo. Por ello, el estudio de una población se realiza a través de encuestas o sondeos sobre un reducido número de elementos, seleccionados con algún procedimiento. Esta selección más pequeña o reducida que nos va a servir para investigar la característica en estudio se denomina **muestra**, y se denotaría por m .

El número de elementos que hay en la muestra se denomina tamaño de la muestra o tamaño muestral, y se denota por $n(m)$. Así,

$$m = \{i_1, i_2, \dots, i_{n(m)}\}$$

Definición 2.1.20 Dada la población, \mathcal{U} , un diseño muestral sobre \mathcal{U} es un par, $d = (M, Pr)$, donde:

- M es un conjunto de muestras o subconjuntos de \mathcal{U} . Se denomina espacio muestral.
- Pr es una distribución de probabilidad sobre M , es decir,

$$0 < Pr(m) \leq 1, \forall m \in M \quad \text{y} \quad \sum_{m \in M} Pr(m) = 1$$

Así M es el conjunto de posibles muestras a seleccionar y, para cada $m \in M$, $Pr(m)$ sería la probabilidad de que m sea la muestra seleccionada.

Definición 2.1.21 Un diseño muestral $d = (M, Pr)$ se dice que es de **tamaño fijo** si

$$n(m) = n, \quad \forall m \in M$$

Definición 2.1.22 Dada una población U y un diseño muestral $d = (M, Pr)$, se denomina probabilidad de inclusión del elemento i , $i = 1, \dots, N$, y se denota π_i , a la probabilidad de que dicho elemento forme parte de la muestra seleccionada. Las cantidades π_i , $i = 1, \dots, N$ se denominan **probabilidades de inclusión de primer orden**. Análogamente, dados dos elementos $i, j \in \mathcal{U}$, se denomina probabilidad de inclusión de i y j , y se denota π_{ij} a la probabilidad de que ambos elementos formen parte de la muestra seleccionada. Las cantidades $\pi_{i,j}$, $i, j = 1, \dots, N$ se denominan

probabilidades de inclusión de segundo orden.

- Probabilidades de inclusión de primer orden:

$$\pi_i = \sum_{m/i \in m} Pr(m) = Pr(i \in m), i = 1, \dots, N$$

π_i : Probabilidad de que el elemento i de la población sea seleccionado para formar parte de la muestra.

- Probabilidades de inclusión de segundo orden:

$$\pi_{ij} = \sum_{m/i, j \in m} Pr(m) = Pr(i, j \in m), i, j = 1, \dots, N$$

π_{ij} : Probabilidad de que los elementos i y j estén simultáneamente en la muestra seleccionada. ($\pi_{ii} = \pi_i$)

Definición 2.1.23 Diremos que un diseño es **probabilístico** cuando verifica:

$$\pi_i > 0, \forall i \in U$$

Definición 2.1.24 Diremos que un diseño es **cuantificable** cuando verifica:

$$\pi_{ij} > 0, \forall i \neq j \in U$$

A continuación se incluye la definición de algunos diseños muestrales básicos, que fueron revisados en detalle para dotar a la aplicación realizada de la funcionalidad correspondiente, como parte de los contenidos de la asignatura de muestreo estadístico

- **Diseño muestral aleatorio simple**, $MAS(N, n)$: partiendo de una población $U = \{1, \dots, N\}$ y sea n un tamaño muestral fijo, tal que $n \leq N$. Este diseño muestral tiene como:
 - espacio muestral, M : todos los posibles subconjuntos de n elementos de la población $U = \{1, \dots, N\}$ y
 - distribución de probabilidad sobre dichas muestras $Pr(m)$ constante.

Ya que el número de muestras posibles es $\binom{N}{n}$, el espacio muestral se puede representar por

$$M = \{m_1, m_2, \dots, m_{\binom{N}{n}}\}$$

siendo

$$Pr(m) = \frac{1}{\binom{N}{n}}, \forall m \in M.$$

Lema 1

En el $MAS(N, n)$ se verifica que

$$\pi_i = \frac{n}{N} = f, \forall i \quad \text{y} \quad \pi_{ij} = \frac{n(n-1)}{N(N-1)}, \forall i \neq j.$$

Luego es un diseño probabilístico y cuantificable.

NOTA También puede encontrarse con la notación m.a.s.

- **Muestreo sistemático uniforme de paso k , $SIS(N, k)$:** fijado un número k , $0 < k \leq N$, se genera un número aleatorio entero, $\gamma \sim \mathcal{U}[1 \dots k]$, entre 1 y k , que será el primer elemento de la muestra. A continuación se va incrementando este valor en k unidades y cada valor obtenido es un nuevo elemento de la muestra. El proceso continua hasta que no hay más elementos.

En un diseño sistemático uniforme de paso k el espacio muestral consta únicamente de k muestras:

$$m_h = \{h, h + k, h + 2k, \dots\}, h = 1, \dots, k$$

y todas ellas tienen la misma probabilidad de ser seleccionadas:

$$Pr(m_h) = \frac{1}{k}, h = 1, \dots, k.$$

Lema 2

- **Probabilidades de inclusión de primer orden:** Cada elemento de la población pertenece únicamente a una muestra:

$$\pi_i = \frac{1}{k}, \forall i$$

- **Probabilidades de inclusión de segundo orden:**

- Si i y j forman parte de la misma muestra sistemática

$$\pi_{ij} = \frac{1}{k}$$

- Si i y j no pertenecen a la misma muestra

$$\pi_{ij} = 0$$

Por tanto el muestreo sistemático no es cuantificable.

El diseño es o no de tamaño fijo, dependiendo de la relación entre N y k :

- Si $N = kn$ ($n \in \mathbb{N}$), todas las muestras son de tamaño n .
- Si $N = kn + r$ ($n \in \mathbb{N}$, $0 < r < k$), el diseño no es de tamaño fijo:
 - m_1, \dots, m_r son de tamaño $n + 1$,
 - m_{r+1}, \dots, m_k son de tamaño n .

Nota:

- Una de las ventajas de este diseño es su fácil implementación.
- Se necesita un listado ordenado de los elementos de la población.
- Puede generar muestras más representativas que el *MAS*.
- Presenta algunos inconvenientes por no ser cuantificable.

- **Diseño Bernouilli, ($MB(N, p)$):** recorrer la población $\{1, \dots, N\}$ elemento a elemento, incluyendo cada uno de ellos en la muestra, de forma independiente, con probabilidad p , $0 \leq p \leq 1$.

- **Diseño Poisson ($MP(p_1, p_N)$):** recorrer la población $\{1, \dots, N\}$ elemento a elemento, incluyendo cada uno de ellos en la muestra, de forma independiente, con probabilidad p_i , $0 \leq p_i \leq 1$, $i = 1, \dots, N$.

- $\pi_i = p_i$, $\pi_{ij} = p_i p_j$.

- **Diseño PPS:** Estos muestreos se realizan empleando una variable de tamaño, de tal forma que los elementos “más grandes” tengan más probabilidad de ser seleccionados.

Definición 2.1.25 Sea $U = \{1, \dots, N\}$, población finita y sea $X = \{x_1, \dots, x_N\}$ una variable de información auxiliar (conocida) con $x_i > 0, \forall i$. Se dirá que un diseño muestral es PROPORCIONAL (PPS) respecto de X si:

$$\exists \alpha > 0 / \pi_i = \alpha x_i, \forall i$$

NOTA

- PPS: viene del inglés (π -proportional - to - size.)
- X : Variable tamaño

Proposición 2.1.7 En un diseño PPS de tamaño muestral constante, n , se verifica que

$$\pi = \frac{n x_i}{t_X}, \forall i$$

En este caso se usará la notación $PPS(N, n; X)$

NOTA Nótese que se ha de verificar que:

$$\pi_i = \frac{n x_i}{t_X} \leq 1, \forall i$$

Si esta condición no se verifica, se podría actuar de alguna de las siguientes formas:

1. Asignar $\pi_k = 1$ a los elementos para los que $\frac{n x_k}{t_X} > 1$ y seleccionar una muestra con probabilidades proporcionales a X sobre el resto de la población.
2. Reducir el tamaño muestral.

Cuando se habla de un diseño PPS:

- No queda determinado el procedimiento utilizado para la selección de la muestra.
- Hace referencia a una propiedad que verifican las probabilidades de inclusión de primer orden.

- **Muestreo con reemplazamiento.** Dada una población finita, $U = \{1, \dots, N\}$, y dada una distribución de probabilidad

$$\{p_1, p_2, \dots, p_N \mid p_i > 0 \forall i, \sum_i^N p_i = 1\},$$

un diseño muestral con reemplazamiento de tamaño n es aquel en el que la muestra está formada por los elementos resultantes de realizar n extracciones independientes de la población, siendo p_i la probabilidad de seleccionar al elemento i -ésimo en cada ocasión.

Utilizaremos la notación m_r para indicar una muestra con reemplazamiento (donde pueden aparecer elementos repetidos, y por tanto, se ha de tener en cuenta el número de veces que aparece cada elemento).

Con respecto a técnicas inferenciales para la obtención de estimadores que pretenden acercarse al verdadero valor del parámetro. Si la variable de estudio es de naturaleza

cuantitativa, tomando pues valores de tipo numérico, los parámetros más usuales son los que se aportan a continuación.

▪ **Total poblacional:**

$$t_y = \sum_{i=1}^N y_i = \sum_{i \in U} y_i$$

▪ **Media poblacional:**

$$\bar{y}_U = \frac{1}{N} \sum_{i=1}^N y_i = \frac{1}{N} \sum_{i \in U} y_i$$

▪ **Cuasivarianza poblacional:**

$$S_{y_U}^2 = \frac{1}{N-1} \sum_{i \in U} (y_i - \bar{y}_U)^2 = \frac{1}{N-1} \left[\sum_{i \in U} y_i^2 - \frac{1}{N} \left(\sum_{i \in U} y_i \right)^2 \right]$$

▪ **Varianza poblacional:**

$$\sigma_{y_U}^2 = \frac{1}{N} \sum_{i \in U} (y_i - \bar{y}_U)^2 = \frac{1}{N-1} \left[\sum_{i \in U} y_i^2 - \frac{1}{N} \left(\sum_{i \in U} y_i \right)^2 \right] = \bar{y}_U^2 - \bar{y}_U^2 = \frac{N-1}{N} S_{y_U}^2$$

Otros parámetros menos usuales, pero también empleados para analizar las poblaciones, son la mediana poblacional, los percentiles poblacionales, el índice de concentración de Gini poblacional o el coeficiente de variación poblacional.

Si la variable de estudio es de naturaleza cualitativa, el parámetro más usual es la proporción asociada a cada uno de los valores o modalidades.

$$P = \frac{\text{número de elementos de la modalidad}}{N}$$

En dicho caso, definiendo:

$$y_i = \begin{cases} 1 & \text{si elto. } i \text{ presenta modalidad bajo estudio} \\ 0 & \text{en caso contrario} \end{cases}$$

Se tiene que:

$$\begin{aligned} \bar{y}_U &= P \\ \sigma_{y_U}^2 &= P - P^2 = P(1 - P) \\ S_{y_U}^2 &= \frac{N}{N-1} P(1 - P) \end{aligned}$$

Ya hemos visto la fase de muestreo, es decir, cómo construir una muestra de algunos diseños muestrales como el aleatorio simple, el de Bernoulli, Poisson, etc. Ahora estudiaremos la fase de estimación, es decir, cómo emplear los datos de esta muestra para obtener información de la población completa.

Definición 2.1.26 Sea $\mathcal{U} = \{1, \dots, N\}$ una población finita. Sea $(M, Pr(\cdot))$ un diseño sobre \mathcal{U} . Un estimador $\hat{\theta}$ es una función:

$$\begin{aligned} \hat{\theta} &: M \longrightarrow \mathbb{R} \\ m &\longrightarrow \hat{\theta}(m) = \hat{\theta}(y_i : i \in m) \end{aligned}$$

▪ **Media muestral**

$$\bar{y}_m = \frac{1}{n(m)} \sum_{i \in m} y_i$$

▪ **Varianza muestral**

$$\sigma_{ym}^2 = \frac{1}{n(m)} \sum_{i \in m} (y_i - \bar{y}_m)^2 = \overline{y^2}_m - \bar{y}_m^2$$

▪ **Cuasivarianza muestral**

$$S_{ym}^2 = \frac{1}{n(m) - 1} \sum_{i \in m} (y_i - \bar{y}_m)^2 = \frac{n(m)}{n(m) - 1} \sigma_{ym}^2$$

NOTA Los estimadores $\hat{\theta}$ son variables aleatorias (discretas). Por tanto tiene sentido hablar de su esperanza, varianza y otras características de las variables aleatorias. En particular:

$$\begin{aligned} E(\hat{\theta}) &= \sum_{m \in M} \hat{\theta}(m) Pr(m) \\ V(\hat{\theta}) &= \sum_{m \in M} \left(\hat{\theta}(m) - E(\hat{\theta}) \right)^2 Pr(m) \\ &\quad \sum_{m \in M} \hat{\theta}^2(m) Pr(m) - \left(E(\hat{\theta}) \right)^2 \end{aligned}$$

A continuación se darán los estimadores en el diseño $MAS(N, n)$, así como conceptos previos para su obtención. Para los demás diseños, se hará de forma análoga

▪ **Estimación en el $MAS(N, n)$**

Proposición 2.1.8 En el $MAS(N, n)$, se verifica que

$$\begin{aligned} \pi_i &= \frac{n}{N} = f, \text{ (fracción de muestreo) } \forall i \in \mathcal{U} \\ \pi_{ij} &= \frac{n(n-1)}{N(N-1)}, \forall i \neq j \in \mathcal{U} \\ \Delta_{ij} &= \frac{-f(1-f)}{N-1}, \forall i \neq j \in \mathcal{U} \end{aligned}$$

Proposición 2.1.9 En el $MAS(N, n)$, se verifica que

1. \bar{y}_m es un estimador insesgado de \bar{y}_U
2. Su varianza: $V(\bar{y}_m) = \frac{1-f}{n} S_{yU}^2$
3. S_{ym}^2 es insesgado de S_{yU}^2
4. Un estimador insesgado de $V(\bar{y}_m)$ viene dado por: $\hat{V}(\bar{y}_m) = \frac{1-f}{n} S_{ym}^2$

Si en lugar de la media poblacional, lo que se quiere estimar es el total poblacional, $t_y = \sum_{i \in \mathcal{U}} y_i$, basta tener en cuenta que $t_y = N\bar{y}_U$, para obtener los siguientes resultados,

- $\hat{t}_y = N\bar{y}_m$
- $V(\bar{y}_m) = N^2 \frac{1-f}{n} S_{yU}^2$
- $\hat{V}(\bar{y}_m) = N^2 \frac{1-f}{n} S_{ym}^2$

Si el parámetro a estimar es una proporción,

$$\theta = P = \text{proporción de elementos que presentan cierta cualidad}$$

Se tiene que: $\bar{y}_U = P$, y por tanto:

- $\hat{P} = \hat{y}_U = \bar{y}_m = p$ (proporción muestral)
- $V(\hat{P}) = \frac{1-f}{n} S_{yU}^2 = \frac{1-f}{n} \frac{N}{N-1} P(1-P)$
- $\hat{V}(P) = \frac{1-f}{n} S_{ym}^2 = \frac{1-f}{n-1} p(1-p)$

Ya hemos visto cómo estimar un parámetro y como estimar la varianza de la estimación. Ahora veremos cómo combinar estos resultados para dar una interpretación de los mismos que resulte útil a efectos prácticos.

Definición 2.1.27 Sea la población $\mathcal{U} = \{1, \dots, N\}$ la característica bajo estudio $Y = \{y_1, \dots, y_N\}$ y el parámetro de interés $\theta = \theta(Y)$. Sean $T_1 = T_1(m)$ y $T_2 = T_2(m)$ dos estadísticos muestrales con $T_1(m) \leq T_2(m)$, $\forall m$. Se dirá que $[T_1, T_2]$ es un intervalo de confianza para θ , con nivel de confianza $1 - \alpha$ ($0 < \alpha < 1$) si,

$$P([T_1, T_2] \ni \theta) \geq 1 - \alpha$$

y se denotará $[T_1, T_2] \equiv IC(\theta; 1 - \alpha)$

Proposición 2.1.10 Sea $\hat{\theta}$ un estimador de θ de forma que

$$\hat{\theta} \xrightarrow{\mathcal{L}} N(\theta; V(\hat{\theta}))$$

y sea $\hat{V}(\hat{\theta})$ un estimador de $V(\hat{\theta})$ de forma que

$$\hat{V}(\hat{\theta}) \xrightarrow{P} V(\hat{\theta})$$

Entonces,

$$\frac{\hat{\theta} - \theta}{\sqrt{\hat{V}(\hat{\theta})}} \xrightarrow{\mathcal{L}} Z \sim N(0, 1)$$

Proposición 2.1.11 Bajo las condiciones anteriores (de forma aproximada)

$$IC(\theta; 1 - \alpha) = \left(\hat{\theta} - Z_{1-\alpha/2} \sqrt{\hat{V}(\hat{\theta})}, \hat{\theta} + Z_{1-\alpha/2} \sqrt{\hat{V}(\hat{\theta})} \right)$$

El **radio** del intervalo de confianza, es decir, la cantidad $Z_{1-\alpha/2} \sqrt{\hat{V}(\cdot)}$ se suele denominar **error de muestreo**.

■ Intervalos de confianza en el MAS

- $IC(\bar{y}_U; 1 - \alpha) = \left(\bar{y}_m - Z_{1-\alpha/2} \sqrt{\frac{1-f}{n} S_{ym}^2}, \bar{y}_m + Z_{1-\alpha/2} \sqrt{\frac{1-f}{n} S_{ym}^2} \right)$
- $IC(P; 1 - \alpha) = \left(p - Z_{1-\alpha/2} \sqrt{\frac{1-f}{n-1} p(1-p)}, p + Z_{1-\alpha/2} \sqrt{\frac{1-f}{n} p(1-p)} \right)$

$$\bullet IC(t_y; 1-\alpha) = \left(N\bar{y}_m - Z_{1-\alpha/2} \sqrt{N^2 \frac{1-f}{n} S_{ym}^2}, N\bar{y}_m + Z_{1-\alpha/2} \sqrt{N^2 \frac{1-f}{n} S_{ym}^2} \right)$$

NOTA

- Si $V(\hat{\theta})$ fuese conocido, entonces, $IC(\theta; 1-\alpha) = \left(\hat{\theta} \mp Z_{1-\alpha/2} \sqrt{V(\hat{\theta})} \right)$
- En el caso particular del *MAS* y $\theta = P$,

$$V(\hat{P}) = \frac{1-f}{n} \frac{N}{N-1} P(1-P) \leq \frac{1-f}{n} \frac{N}{(N-1)} \frac{1}{4}$$

y por tanto,

$$IC(P; 1-\alpha) = \left(\hat{P} \mp Z_{1-\alpha/2} \sqrt{\frac{1-f}{n} \frac{N}{N-1} \frac{1}{4}} \right)$$

La elección del tamaño muestral es una de las cuestiones más relevantes del muestreo en poblaciones finitas. Se trata de decidir cuál va a ser el tamaño de la muestra, o sea n , que vamos a extraer de la población. En principio nos guiaremos por criterios de precisión, es decir, buscamos una exactitud o precisión en la estimación, predeterminada de antemano.

Dependiendo del tipo de parámetro a estimar (adimensional o dimensional), se plantea el problema:

Parámetro adimensional Fijados un error (absoluto) máximo admisible (δ) y un nivel $(1-\alpha)$, determinar n de forma que

$$Pr[|\hat{P} - P| \leq \delta] = 1 - \alpha$$

Parámetro dimensional Fijados un error relativo máximo admisible (δ) y un nivel $(1-\alpha)$, determinar n de forma que

$$Pr\left[\left|\frac{\hat{\theta}}{\theta} - 1\right| \leq \delta\right] = 1 - \alpha$$

- **Determinación del tamaño muestral en el *MAS* para P :** el problema que se plantea es determinar el tamaño muestral necesario para que

$$Pr[|\hat{P} - P| \leq \delta] = 1 - \alpha$$

es equivalente a decir que $(\hat{P} \mp \delta)$ es un $IC(P, 1-\alpha)$.

A partir de los resultados ya obtenidos en el *MAS*,

\hat{P} = proporción muestral

$$V(\hat{P}) = \frac{1-f}{n} \frac{N}{N-1} P(1-P) \equiv \frac{1-f}{n} P(1-P) \leq \frac{1-f}{n} \frac{1}{4}$$

Así pues, considerando que

$$IC(P, 1-\alpha) = \hat{P} \mp Z_{1-\alpha/2} \sqrt{\frac{1-f}{n} \frac{1}{4}}$$

se deduce que

$$n = \frac{n_0}{1 + \frac{n_0}{N}} \quad \text{con} \quad n_0 = \frac{Z_{1-\alpha/2}^2}{4\delta^2}$$

Observaciones

1. n_0 está determinado por los valores fijados para δ y $(1 - \alpha)$.
2. Para $(1 - \alpha) = 0.955$ se obtiene que $n_0 = \frac{1}{\delta^2}$.
3. Se verifica que $n \leq n_0, \forall N$.

- **Determinación del tamaño muestral en el MAS para \bar{y}_U :** ahora el problema que se plantea es determinar el tamaño muestral necesario para que

$$Pr \left[\left| \frac{\hat{\theta}}{\theta} - 1 \right| \leq \delta \right] = 1 - \alpha$$

lo cual es equivalente a decir que $(\hat{\theta} \mp \delta|\theta|)$ es un $IC(\theta, 1 - \alpha)$.
 A partir del resultado obtenido en el MAS,

$$IC(\bar{y}_U; 1 - \alpha) = \left(\bar{y}_m - Z_{1-\alpha/2} \sqrt{\frac{1-f}{n} S_{ym}^2}, \bar{y}_m + Z_{1-\alpha/2} \sqrt{\frac{1-f}{n} S_{ym}^2} \right)$$

Se deduce que

$$n = \frac{n_0}{1 + \frac{n_0}{N}} \quad \text{con} \quad n_0 = \frac{Z_{1-\alpha/2}^2 CV_{yU}^2}{\delta^2} \quad \text{con} \quad CV_{yU} = \frac{S_{yU}}{\bar{y}_U}$$

OBSERVACIONES

1. n_0 depende de parámetros desconocidos, por lo que hay varias formas de resolver este problema. Una de ellas consiste en emplear, si existe, información obtenida en otros estudios, extrapolando algunos resultados. Otra posibilidad es obtener una muestra preliminar o muestra piloto, m_1 , de tamaño n_1 , y una vez calculado $CV_{y_{m_1}}$ con dicha muestra, determinar n_0 . Posteriormente se vuelve a realizar un muestreo de $n - n_1$ elementos para obtener la información deseada

Otros estimadores de interés son lo que se presentan a continuación:

Definición 2.1.28 Dado un diseño muestral probabilístico, se define el **estimador de Horvitz-Thompson** o π -estimador de t_y como:

$$\hat{t}_y = \sum_{i \in m} \frac{y_i}{\pi_i}$$

Proposición 2.1.12 El estimador de Horvitz-Thompson es insesgado, su varianza se puede expresar como:

$$V(\hat{t}_y) = \sum_{i \in U} \sum_{j \in U} \Delta_{ij} \frac{y_i y_j}{\pi_i \pi_j}$$

Si el diseño es cuantificable, es decir $\pi_{ij} > 0 \forall i, j \in U$, entonces un estimador insesgado de la varianza es:

$$\hat{V}(\hat{t}_y) = \sum_{i \in m} \sum_{j \in m} \frac{\Delta_{ij} y_i y_j}{\pi_{ij} \pi_i \pi_j}$$

Proposición 2.1.13 Si el diseño muestral es de tamaño fijo, se obtiene una expresión alternativa de la varianza del π -estimador

$$\hat{V}(\hat{t}_y) = -\frac{1}{2} \sum_{i \in U} \sum_{j \in U} \Delta_{ij} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2$$

(Esta expresión es conocida con el nombre de Yates-Grundy-Sen). Se Obtiene, por tanto, si el diseño es cuantificable, el siguiente estimador insesgado de la varianza:

$$\hat{V}(\hat{t}_y) = -\frac{1}{2} \sum_{i \in m} \sum_{j \in m} \frac{\Delta_{ij}}{\pi_{ij}} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2$$

NOTA esta expresión de la varianza justifica que, si existe un variable auxiliar X relacionada con la variable Y , un diseño muestral PPS reduciría la varianza del estimador.

Definición 2.1.29 Considerando la media poblacional como un cociente o razón entre el total t_y y N , se define el estimador de Hajek como el cociente de los respectivos π -estimadores,

$$\hat{y}_{U,HJ} = \frac{\hat{t}_{y\pi}}{\hat{N}_\pi} = \frac{\sum_{i \in m} \frac{y_i}{\pi_i}}{\sum_{i \in m} \frac{1}{\pi_i}}$$

Definición 2.1.30 Dado un diseño muestral con reemplazamiento, se define el **estimador de Hansen-Hurwitz** de t_y como:

$$\hat{t}_{y,HH} = \sum_{i=1}^N \frac{y_i}{n p_i} f_i = \sum_{i \in m_r} \frac{y_i}{n p_i}$$

Proposición 2.1.14 El estimador de Hansen-Hurwitz es insesgado, su varianza se puede expresar como:

$$V(\hat{t}_{y,HH}) = \frac{1}{n} \left(\sum_{i \in U} \frac{y_i^2}{p_i} - t_y^2 \right)$$

siendo un estimador insesgado de dicha varianza

$$\hat{V}(\hat{t}_{y,HH}) = \frac{1}{n(n-1)} \left(\sum_{i \in m_r} \frac{y_i}{p_i} - \hat{t}_{y,HH}^2 \right)^2$$

Haciendo el cambio de variable $z_i = \frac{y_i}{p_i}$, se obtiene:

$$\begin{aligned} \hat{t}_{y,HH} &= \bar{z}_{m_r} \\ \hat{V}(\hat{t}_{y,HH}) &= \frac{1}{n(n-1)} \sum_{i \in m_r} (z_i - \bar{z}_{m_r})^2 = \frac{1}{n} S_{z,m_r}^2 \end{aligned}$$

NOTA El estimador de Hansen-Hurwitz de \bar{y}_U se define como:

$$\bar{y}_{U,HH} = \frac{1}{N} \sum_{i=1}^N \frac{y_i}{n p_i} f_i = \frac{1}{N} \sum_{i \in m_r} \frac{y_i}{n p_i}$$

Proposición 2.1.15 En un diseño muestral con reemplazamiento de tamaño n , sea

$$f_i = \text{num. de veces que } i \text{ aparece en } m_r, \quad i = 1, \dots, N$$

Entonces:

1. (f_1, \dots, f_N) sigue una distribución multinomial, $M(n; p_1, \dots, p_N)$.
2. f_i es una variable aleatoria discreta con distribución binomial, $Bi(n, p_i)$.
3. $E[f_i] = np_i$, $V(f_i) = np_i(1 - p_i)$.
4. $Cov(f_i, f_j) = -np_i p_j$, $\forall i \neq j$.

Dado cualquier diseño muestral, y un estimador θ , especialmente en aquellos en los que no sea posible obtener un estimador insesgado de la varianza bajo el diseño realizado, se puede conseguir una estimación conservadora de dicha varianza tomando el estimador para un muestreo con reemplazamiento.

Si $\hat{\theta} = \bar{z}_m$ entonces un estimador de $V(\hat{\theta})$ podría ser:

$$\hat{V}(\hat{\theta}) = \hat{V}_{CR}(\bar{z}_m) = \frac{1}{n} S_{z,m}^2$$

El estimador de la varianza resultante no es insesgado, pero es mucho más fácil de usar, tiene una expresión mucho más simple y no depende de las $p_{i,j}$

Habitualmente en un estudio por muestreo hay más de una variable objetivo. Por ejemplo:

- Ingresos de la unidad familiar: y^1
- Miembros de la unidad familiar: y^2
- Gastos en alimentación de la unidad familiar: y^3
- Gastos escolares de la unidad familiar: y^4
- ...

Hasta ahora me he centrado en el estudio de cada variable por separado. Pero existen otros parámetros de interés: ingresos por individuo, gastos totales, proporción de gastos respecto a ingresos, proporción de gastos en alimentación respecto a ingresos, etc.

En general, se puede decir que el parámetro de interés se puede expresar como:

$$\theta = f(t_{y^1}, t_{y^2}, \dots, t_{y^p}) \quad \text{con } f \text{ conocida}$$

Un estimador de dicho parámetro sería de la forma:

$$\hat{\theta} = f(\hat{t}_{y^1}, \hat{t}_{y^2}, \dots, \hat{t}_{y^p})$$

Analizaremos el estimador obtenido según sea una función lineal o no.

- $\theta \equiv$ **FUNCION LINEAL**: Si $\theta = \sum_{k=1}^p t_{y^k}$, entonces

$$\hat{\theta} = \sum_{k=1}^p \hat{t}_{y^k} = \hat{t}_{z,\pi} \quad z = \sum_{k=1}^p y^k$$

El estimador resultante es el estimador HT y se pueden aplicar las expresiones de varianza y estimador de la varianza.

- $\theta \equiv$ **FUNCIÓN NO LINEAL**: Utilizaremos el método de aproximación lineal para obtener un estimador de la varianza.

Sea $f(x, y)$ una función de 2 variables. Sea (x_0, y_0) un punto fijo de forma que existen las derivadas de f de cualquier orden en $(x_0; y_0)$. Por el desarrollo de Taylor,

$$f(x, y) = f(x_0, y_0) + f'_x(x_0, y_0) \cdot (x - x_0) + f'_y(x_0, y_0) \cdot (y - y_0) + \frac{1}{2} f''_{xx}(x_0, y_0) \cdot (x - x_0)^2 + \frac{1}{2} f''_{yy}(x_0, y_0) \cdot (y - y_0)^2 + f''_{xy}(x_0, y_0) \cdot (x - x_0)(y - y_0) + \dots$$

Utilizaremos la aproximación lineal,

$$f(x, y) \approx f(x_0, y_0) + f'_x(x_0, y_0) \cdot (x - x_0) + f'_y(x_0, y_0) \cdot (y - y_0)$$

Consideremos en particular la función $f(x, y) = \frac{y}{x}$, cuya aproximación lineal sería:

$$f(x, y) = \frac{y}{x} \approx \frac{y_0}{x_0} + \left(-\frac{y_0}{x_0^2} \right) (x - x_0) + \frac{1}{x_0} (y - y_0) = \frac{y_0}{x_0} + \frac{y - \frac{y_0}{x_0} \cdot x}{x_0}$$

Teniendo en cuenta lo anterior, un parámetro de interés y de aplicación de función no lineal, así como sus estimaciones, es el que se aporta a continuación.

Definición 2.1.31 Se define el parámetro razón como el cociente de dos totales poblacionales:

$$R = \frac{t_{yU}}{t_{xU}}$$

Proposición 2.1.16 Dado el estimador del parámetro razón,

$$\hat{R}_\pi = \frac{\hat{t}_{yU}}{\hat{t}_{xU}}$$

aplicando la linealización se obtiene:

$$\hat{R}_\pi \approx R + \frac{\hat{t}_{y,\pi} - R \cdot \hat{t}_{x,\pi}}{t_{xU}} = R + \sum_{i \in m} \frac{z_i}{\pi_i}$$

con $z_i = \frac{y_i - R x_i}{t_{xU}}$. Por tanto,

$$V(\hat{R}_\pi) \approx V\left(R + \sum_{i \in m} \frac{z_i}{\pi_i}\right) = V\left(\sum_{i \in m} \frac{z_i}{\pi_i}\right)$$

La estimación de esta varianza se puede obtener por alguno de los siguientes procedimientos:

- Aplicando expresiones del estimador de HT

$$\hat{V}(\hat{R}_\pi) = \sum_{i \in m} \sum_{j \in m} \frac{z_i}{\pi_i} \frac{z_j}{\pi_j} \frac{\Delta_{ij}}{\pi_{ij}} \quad \text{con} \quad z_i = \frac{y_i - \hat{R}_\pi x_i}{\hat{t}_{x,\pi}}$$

b) Considerando el estimador del muestreo con reemplazamiento,

$$\hat{V}_{CR}(\hat{R}_\pi) = \frac{1}{n} S_{z^*,m}^2, \quad z_i^* = \frac{n z_i}{\pi_i} = \frac{n y_i - \hat{R}_\pi x_i}{\hat{t}_{x,\pi}}$$

NOTA \hat{R}_π y $\hat{V}(\hat{R}_\pi)$ no son estimadores insesgados de R_π y $V(\hat{R}_\pi)$

OBSERVACIÓN aplicando resultados del apartado anterior

$$\hat{V}_{CR}(\hat{y}_{HJ}) = \frac{1}{n} S_{z^*,m}^2, \quad z_i^* = \frac{n z_i}{\pi_i} = \frac{n y_i - \hat{y}_{HJ}}{\hat{N}_\pi}$$

En ocasiones se dispone de información adicional conocida para todos los elementos de \mathcal{U} ,

$$X = \{x_1, \dots, x_N\}$$

Este tipo de variable de información auxiliar ha sido utilizada en algún diseño muestral, como el *PIPS*. Nos planteamos en este tema el uso de esta variable X en la fase de estimación, construyendo otro tipo de estimadores alternativos al de Horvitz-Thompson.

En general, supondremos que entre las variables Y y X existe cierta relación expresada de la forma

$$y_i \approx f(x_i)$$

siendo f una función de la que supondremos conocida de qué tipo es, pero no su expresión exacta. Por ejemplo:

- a) $y_i \approx x_i + d$ con d desconocido.
- b) $y_i \approx \alpha + \beta x_i$ con α, β desconocidos.
- c) $y_i \approx \beta x_i$ con β desconocido.

Este tipo de relaciones se suele expresar como un modelo:

$$y_i = f(x_i) + \varepsilon_i \quad \text{con} \quad E(\varepsilon_i) = 0, \quad V(\varepsilon_i) = \sigma_i^2 \quad (2.4)$$

Por ejemplo:

- a) $y_i = x_i + d + \varepsilon_i$ con $E(\varepsilon_i) = 0, \quad V(\varepsilon_i) = \sigma^2$.
- b) $y_i \approx \alpha + \beta x_i + \varepsilon_i$ con $E(\varepsilon_i) = 0, \quad V(\varepsilon_i) = \sigma^2$.
- c) $y_i \approx \beta x_i + \varepsilon_i$ con $E(\varepsilon_i) = 0, \quad V(\varepsilon_i) = \sigma_i^2$.

Supuesto alguna relación del tipo 2.4, se deben en primer lugar, determinar los parámetros del modelo $\{d, (\alpha, \beta), \beta\}$. El método usual es el de mínimos cuadrados ponderados, es decir, determinar los parámetros que minimicen la siguiente función:

$$\text{mín} \sum_{i \in m} \frac{(y_i - f(x_i))^2}{\pi_i V(\varepsilon_i)}$$

Una vez resuelto el problema, se pueden definir los valores ajustados y los residuos asociados al problema

Valores ajustados: $\hat{y}_i = \hat{f}(x_i)$

Residuos: $e_i = y_i - \hat{y}_i$

Definición 2.1.32 Dado un modelo $y_i = f(x_i) + \varepsilon_i$, se define el estimador general de regresión de t_y como:

$$\hat{t}_{y,GREG} = \sum_{i \in U} \hat{y}_i + \sum_{i \in m} \frac{(y_i - \hat{y}_i)}{\pi_i} = \sum_{i \in U} \hat{y}_i + \sum_{i \in m} \frac{e_i}{\pi_i}$$

Proposición 2.1.17

$$\begin{aligned} V(\hat{t}_{y,GREG}) &\approx \sum_{i \in U} \sum_{j \in U} \frac{e_i}{\pi_i} \frac{e_j}{\pi_j} \Delta_{ij} \\ \hat{V}(\hat{t}_{y,GREG}) &= \sum_{i \in m} \sum_{j \in m} \frac{e_i}{\pi_i} \frac{e_j}{\pi_j} \frac{\Delta_{ij}}{\pi_{ij}} \\ \hat{V}_{CR}(\hat{t}_{y,GREG}) &= \frac{1}{n} S_{z,m}^2 \quad \text{con} \quad z_i = \frac{n e_i}{\pi_i} \end{aligned}$$

Casos particulares del estimador general de regresión son los que se presentan a continuación:

- **Estimador diferencia:** Corresponde al caso particular:

$$y_i = x_i + d + \varepsilon_i \quad \text{con} \quad E(\varepsilon_i) = 0, \quad V(\varepsilon_i) = \sigma^2$$

El Estimador Mínimo Cuadrático (EMC) de d viene dado por:

$$\min_d \sum_{i \in m} \frac{(y_i - x_i - d)^2}{\pi_i \sigma^2} \equiv \sum_{i \in m} \frac{(y_i - x_i - d)^2}{\pi_i}$$

cuya solución es:

$$\hat{d} = \hat{y}_{HJ} - \hat{x}_{HJ}$$

Valores ajustados: $\hat{y}_i = x_i + \hat{d}$

Residuos: $e_i = y_i - (x_i + \hat{d})$

- Estimador diferencia para t_y ,

$$\begin{aligned} \hat{t}_{y,dif} &= \sum_{i \in U} \hat{y}_i + \sum_{i \in m} \frac{e_i}{\pi_i} = \sum_{i \in U} \hat{y}_i + \sum_{i \in m} \frac{y_i - x_i + \hat{d}}{\pi_i} = \\ &= \sum_{i \in U} \hat{y}_i = \sum_{i \in U} (x_i + \hat{d}) = N \hat{y}_{HJ} + (t_{x_U} - N \hat{x}_{HJ}) \end{aligned}$$

- Estimador de la varianza:

$$\begin{aligned} \hat{V}(\hat{t}_{y,dif}) &= \sum_{i \in m} \sum_{j \in m} \frac{e_i}{\pi_i} \frac{e_j}{\pi_j} \frac{\Delta_{ij}}{\pi_{ij}} \quad \text{con} \quad e_i = y_i - x_i - \hat{d} \\ \hat{V}_{CR}(\hat{t}_{y,dif}) &= \frac{1}{n} S_{z,m}^2 \quad \text{con} \quad z_i = \frac{n e_i}{\pi_i} \end{aligned}$$

- Estimador diferencia para \bar{y}_U ,

$$\hat{y}_{dif} = \hat{y}_{HJ} + (\bar{x}_U - N \hat{x}_{HJ})$$

- **Estimador de razón:** Corresponde al modelo:

$$y_i = \beta x_i + \varepsilon_i \quad \text{con} \quad E(\varepsilon_i) = 0, \quad V(\varepsilon_i) = \sigma^2 x_i$$

El Estimador Mínimo Cuadrático (EMC) de β viene dado por:

$$\min_{\beta} \sum_{i \in m} \frac{(y_i - \beta x_i)^2}{\pi_i \sigma^2 x_i} \equiv \sum_{i \in m} \frac{(y_i - \beta x_i)^2}{\pi_i x_i}$$

cuya solución es:

$$\hat{\beta} = \frac{\hat{t}_{y,\pi}}{\hat{t}_{x,\pi}}$$

Valores ajustados: $\hat{y}_i = \hat{\beta} x_i$

Residuos: $e_i = y_i - \hat{\beta} x_i$

- Estimador razón para t_y ,

$$\hat{t}_{y,razon} = \sum_{i \in U} \hat{y}_i + \sum_{i \in m} \frac{e_i}{\pi_i} = \sum_{i \in U} \hat{y}_i = \hat{\beta} t_{xU} = \hat{R}_{\pi} t_{xU} = \hat{t}_{y,\pi} \cdot \frac{t_{xU}}{\hat{t}_{x,\pi}}$$

- Estimador de la varianza:

$$\begin{aligned} \hat{V}(\hat{t}_{y,razon}) &= \sum_{i \in m} \sum_{j \in m} \frac{e_i e_j}{\pi_i \pi_j} \frac{\Delta_{ij}}{\pi_{ij}} \quad \text{con} \quad e_i = y_i - \hat{\beta} x_i \\ \hat{V}_{CR}(\hat{t}_{y,razon}) &= \frac{1}{n} S_{z,m}^2 \quad \text{con} \quad z_i = \frac{n e_i}{\pi_i} \end{aligned}$$

- Estimador razón para \bar{y}_U ,

$$\hat{y}_{razon} = \hat{R}_{\pi} \bar{x}_U$$

- **Estimador de regresión:** Corresponde al modelo:

$$y_i = \alpha + \beta x_i + \varepsilon_i \quad \text{con} \quad E(\varepsilon_i) = 0, \quad V(\varepsilon_i) = \sigma^2$$

El Estimador Mínimo Cuadrático (EMC) de (α, β) viene dado por:

$$\min_{\alpha, \beta} \sum_{i \in m} \frac{(y_i - \alpha - \beta x_i)^2}{\pi_i \sigma^2} \equiv \sum_{i \in m} \frac{(y_i - \alpha - \beta x_i)^2}{\pi_i}$$

cuya solución es:

$$\begin{aligned} \hat{\alpha} &= \hat{y}_{HJ} - \hat{\beta} \cdot \hat{x}_{HJ} \\ \hat{\beta} &= \frac{\hat{x}\hat{y}_{HJ} - \hat{x}_{HJ}\hat{y}_{HJ}}{\hat{x}^2_{HJ} - \hat{x}_{HJ}^2} \end{aligned}$$

Valores ajustados: $\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i$

Residuos: $e_i = y_i - (\hat{\alpha} + \hat{\beta} x_i)$

- Estimador regresión para t_y ,

$$\hat{t}_{y,reg} = \sum_{i \in U} \hat{y}_i + \sum_{i \in m} \frac{e_i}{\pi_i} = \sum_{i \in U} \hat{y}_i = \sum_{i \in U} (\hat{\alpha} + \hat{\beta} x_i) = \hat{\alpha} + \hat{\beta} t_{xU}$$

o Estimador de la varianza:

$$\begin{aligned}\hat{V}(\hat{t}_{y,reg}) &= \sum_{i \in m} \sum_{j \in m} \frac{e_i e_j \Delta_{ij}}{\pi_i \pi_j \pi_{ij}} \quad \text{con} \quad e_i = y_i - \hat{\alpha} - \hat{\beta} x_i \\ \hat{V}_{CR}(\hat{t}_{y,razon}) &= \frac{1}{n} S_{z,m}^2 \quad \text{con} \quad z_i = \frac{n e_i}{\pi_i}\end{aligned}$$

• Estimador regresión para \bar{y}_U ,

$$\hat{y}_{reg} = \frac{\hat{\alpha}}{N} + \hat{\beta} \bar{x}_U$$

2.1.3. Tercer curso

En tercero se ven asignatura como análisis multivariante, modelos lineales, ampliación de inferencia estadística, etc enfocadas en cierta forma en hacer inferencias sobre los datos y probar hipótesis, como no es posible abarcar todas las asignaturas ni todos los contenidos de una asignatura, se ha optado por seleccionar algunas técnicas de la materia de ampliación de inferencia estadística principalmente los contrastes no paramétricos para una muestra, dos muestras independientes, dos muestras relacionadas y k muestras. A diferencia de las técnicas paramétricas, en las cuales la distribución de la variable que representa el experimento se supone conocida salvo parámetro, los métodos no paramétricos tratan de solucionar problemas en los cuales esta información distribucional, en general, se desconoce.

Todos los conceptos que se aportan y se analizan de la asignatura de ampliación de inferencia estadística viene recogidos en [6] y [4].

Con los métodos no paramétricos se pretenden dar alternativas a algunos problemas paramétricos (estudio de una población, comparación de dos o más poblaciones, etc.) o bien resolver nuevos problemas (aleatoriedad, bondad de ajuste, etc.)

Si bien los métodos no paramétricos son menos potentes que los paramétricos, las conclusiones obtenidas con ellos son más generales que las obtenidas con los procedimientos paramétricos,

A continuación, se recogen algunos estadísticos de gran utilidad en los procedimientos no paramétricos

Definición 2.1.33 Dada una m.a.s. X_1, X_2, \dots, X_n procedente de una variable aleatoria X , se define el estadístico ordenado de rango i , $X_{(i)}$, $i = 1, 2, \dots, n$, como la variable aleatoria que toma el i -ésimo valor más pequeño de la muestra.

Si la variable X es continua, los estadísticos de orden tienen la propiedad de que su distribución no depende de la distribución de X , es decir son estadísticos de distribución libre.

Definición 2.1.34 Dada una m.a.s. X_1, X_2, \dots, X_n procedente de una variable aleatoria X , se definen los estadísticos de rango-orden como cualquier conjunto de constantes que indican el orden de las observaciones muestrales. En particular:

$$R_i = R(X_{(i)}) = i \quad i = 1, 2, \dots, n$$

A continuación, se recogen las técnicas más usuales según el estudio del contraste que se plantee.

- **Contrastes de una muestra:** a continuación se describirán algunos de los principales tests que se han incorporado en una de las herramientas que se proporcionan al usuario de la aplicación (se proveen mucho más en la misma)
- **Test Chi-cuadrado:** para poder aplicar este test de bondad de ajuste, es necesario que los datos vengan agrupados de alguna forma con el objetivo de poder construir una tabla de frecuencias. Por este motivo, si la variable bajo estudio es cualitativa o cuantitativa discreta no presenta ningún problema su aplicación; en contra, si la variable fuese continua habría que recurrir a la construcción de intervalos de clase, con la correspondiente pérdida de información muestral.

El estadístico que se propone se basa en la comparación de las frecuencias absolutas observadas en cada una de las categorías con sus correspondientes valores esperados supuesta cierta H_0 .

Sean $A_1, \dots, A_i, \dots, A_k$ las categorías exhaustivas y mutuamente excluyentes donde toma valores la variable X . Sea n_i , $i = 1, \dots, k$ las frecuencias muestrales observadas en cada una de las categorías, y sea $p_i = P(X \in A_i / H_0)$, $i = 1, \dots, k$, las probabilidades teóricas de cada una de las categorías. El estadístico que se propone es:

$$\chi^2 = \sum_{i=1}^k \left[(n_i - np_i)^2 / np_i \right]$$

rechazándose la hipótesis nula si el valor del estadístico es superior al punto crítico.

La distribución exacta del estadístico, bajo H_0 , es desconocida, por lo que hay que recurrir a su distribución asintótica. Bajo H_0 ,

$$\chi^2 \xrightarrow[n \rightarrow \infty]{} \chi_{k-1}^2$$

por lo que la región crítica viene dada por: $\chi^2 \geq \chi_{k-1, 1-\alpha}^2$.

Consideraciones:

- En el caso particular $k = 2$, a este estadístico se le aplica la corrección de Yates.
- Si la hipótesis nula dependiese de algún parámetro desconocido, previamente, habría que estimar dicho parámetro a partir de los resultados muestrales. En este caso la región crítica vendría dada por $\chi^2 \geq \chi_{k-1-s, 1-\alpha}^2$ siendo s el número de parámetros desconocidos y estimados.
- **Test de Kolmogorov-Smirnov:** para el estudio de bondad de ajuste, este test se basa en el teorema Central de la Estadística o teorema de Glivenko-Cantelli, según el cual, la función de distribución empírica de la muestra, $S_n(x)$, converge a $F_X(x)$ con probabilidad 1.

El estadístico propuesto es:

$$D_n = \sup_x |F_0(x) - S_n(x)| = \max\{D_n^+, D_n^-\}$$

$$\text{siendo } D_n^+ = \max_{i=1, \dots, n} \left[\frac{i}{n} - F_0(x_{(i)}) \right] \text{ y } D_n^- = \max_{i=1, \dots, n} \left[F_0(x_{(i)}) - \frac{i-1}{n} \right]$$

Si la variable X es continua, los estadísticos D_n , D_n^+ y D_n^- son de distribución libre.

La región crítica viene dada por: $D_n \geq D_{n,\alpha}$

Los puntos críticos $D_{n,\alpha}$ han sido tabulados por Kolmogorov-Smirnov bajo la hipótesis de continuidad.

Consideraciones:

- Desde el punto de vista teórico, este test permite contrastes de tipo unilateral.
 - Si la variable es discreta, se pueden utilizar los mismos puntos críticos, pero en esta situación el test se hace conservativo.
 - Si la hipótesis nula dependiese de algún parámetro desconocido, previamente habría que estimar dicho parámetro a partir de los datos muestrales. En este caso el test se hace conservativo, excepto para la hipótesis de distribución normal o exponencial, para estas dos distribuciones hay que aplicar la corrección de Lilliefors.
- **Test de Shapiro-Wilk:** el objetivo que se persigue es estudiar si la variable continua objeto de interés sigue una distribución normal.

El estadístico propuesto es:

$$W = \frac{b^2}{(n-1)S_c^2}$$

siendo S_c^2 la cuasivarianza muestral y $b = \sum_{i=1}^k a_{n-i+1} [X_{(n-i+1)} - X_{(i)}]$ donde a_{n-i+1} son los coeficientes de Shapiro-Wilk, $i = 1, \dots, k$, siendo

$$k = \begin{cases} n/2 & \text{si } n \text{ es par} \\ (n+1)/2 & \text{si } n \text{ es impar} \end{cases}$$

La región crítica viene dada por: $W \leq W_{n,\alpha}$.

Los valores críticos $W_{n,\alpha}$ han sido tabulados por Shapiro-Wilk, para tamaños muestrales no superiores a 50.

Consideraciones:

- Otro test específico para el estudio de normalidad es el test D'Agostino.

- **Test binomial:** este test puede encuadrarse, indistintamente, en las técnicas paramétricas, en los tests de bondad de ajuste o bien como test previo a los contrastes de localización.

Dada una m.a.s. X_1, X_2, \dots, X_n procedente de una variable aleatoria X , la cual solamente puede tomar dos valores, éxito y fracaso (experimento Bernouilli), el objetivo que se persigue es analizar la hipótesis $H_0 : p = p_0$, siendo p la probabilidad de éxito en dicho experimento.

Constraste que se plantea es:

$$\begin{aligned} H_0 &: p = p_0 \\ H_1 &: p \neq p_0 \end{aligned}$$

El Estadístico propuesto es: $T = \sum_{i=1}^n X_i =$ número de éxitos en la muestra el cual, bajo H_0 , sigue una distribución $B(n, p_0)$.

La región crítica viene dada por: $\alpha_0 \leq \alpha/2$ ó $\alpha_1 \leq \alpha/2$ con

$$\alpha_0 = P(T \leq T_{exp}/H_0) \quad \text{y} \quad \alpha_1 = P(T \geq T_{exp}/H_0)$$

siendo T_{exp} el valor que toma el estadístico T en la realización muestral.

Consideraciones:

- Este test permite analizar contrastes unilaterales. Así, para

$$H_1 : p \leq p_0 \text{ la región crítica viene dada por } \alpha_0 \leq \alpha$$

y para

$$H_1 : p \geq p_0 \text{ la región crítica viene dada por } \alpha_1 \leq \alpha.$$

- **Contrastes para la comparación de dos poblaciones:** en este apartado se analizan algunos procedimientos no paramétricos para la comparación de dos variables, poblaciones, tratamientos, etc.

Para la aplicación de las diferentes técnicas se ha de tener presente el tipo de variables que se desean comparar y si las muestras son relacionadas o independientes.

- **Comparación con muestras relacionadas:** ejemplos clásicos de comparación de dos tratamientos, poblaciones, variables, etc., con muestras relacionadas son los obtenidos a través de experiencias del tipo: antes/después, izquierda/derecha, pares homogéneos, etc. Así, para cada individuo analizado, se dispone dos variables o medidas relacionadas, que se pueden representar por (X_i, Y_i) , $i = 1, \dots, n$, donde la primera componente representa el valor de la primera variable (antes, izquierda, etc.) del i -th individuo de la muestra y la segunda componente el valor de la segunda variable (después, derecha, etc.) del mismo individuo.

- **Variables cuantitativas:** si las variables, X e Y , que se desean comparar son de tipo cuantitativo, basta considerar la variable $D = X - Y$ y, a continuación, aplicar las técnicas para el estudio de una sola población.
- **Variables Bernouilli:** si las variables o experimentos son de tipo Bernouilli, el método que se aplica es el test de McNemar. En esta situación, el objetivo es comparar p_X con p_Y , siendo p_X y p_Y , respectivamente, las probabilidades de que ocurra un éxito en cada experimento (antes/después, izquierda/derecha, etc.).

La hipótesis $H_0 : p_X = p_Y$ es equivalente a $H_0 : P(X = 1, Y = 0) = P(X = 0, Y = 1)$. Por lo tanto, basta considerar un nuevo experimento Bernouilli, en el cual, son posibles los siguientes resultados:

$$E' = \{\text{Pasar de éxito a fracaso}\} \quad \text{y}$$

$$F' = \{\text{Pasar de fracaso a éxito}\}$$

y representando por $p = P(E')$, la hipótesis a analizar es equivalente a:

$$H_0 : p = 1/2$$

la cual puede ser analizada por el test Binomial.

Desde el punto de vista práctico, los resultados pueden recogerse en una tabla de frecuencias del tipo:

X/Y	éxito	fracaso
éxito	n_{11}	n_{12}
fracaso	n_{21}	n_{22}

siendo

$$n_{12} = \text{número de individuos que han pasado de éxito a fracaso}$$

$$n_{21} = \text{número de individuos que han pasado de fracaso a éxito}$$

Por tanto, el estadístico a utilizar será $T_{exp} = n_{12}$, siendo el tamaño de la muestra, en el test de McNemar, $n_{12} + n_{21} =$ número de individuos que presentan cambios en la muestra, de modo que el resto de los individuos no intervienen en los resultados.

Consideraciones:

- ◇ Dado que el test Binomial permite estudiar contrastes unilaterales, el test de McNemar también sirve para hipótesis del tipo $H_0 : p_x \leq p_y$ o $H_0 : p_x \geq p_y$.

- **Comparación con muestras independientes:** dadas dos m.a.s. independientes una procedente de una variable aleatoria X , con función de distribución F_X desconocida, y otra procedente de una variable aleatoria Y , con función de distribución F_Y desconocida, el objetivo que se plantea es la comparación de ambas variables (tratamientos, poblaciones, etc.). El contraste que se desea analizar es:

$$H_0 : F_X(t) = F_Y(t) \quad \forall t$$

$$H_1 : \exists t / F_X(t) \neq F_Y(t)$$

A continuación, se recogen una de las técnicas para dicho contraste, si bien a lo largo del trabajo se han analizado e incluido en la aplicación realizada algunas más. La aplicación de las mismas dependerá del tipo de variables que se desean comparar.

- **Test de Kolmogorov-Smirnov:** para el estudio del contraste anterior (comparación de poblaciones o, más formalmente, homogeneidad de poblaciones) el test de Kolmogorov-Smirnov, al igual que en el estudio de bondad de ajuste, se basa en el teorema Central, por lo que, el estadístico propuesto es:

$$D_{n,m} = \max_t |S_n(t) - T_m(t)|$$

siendo $S_n(t)$ y $T_m(t)$, respectivamente, las funciones de distribución empírica de cada una de las muestras.

Si las variables son continuas, el estadístico $D_{n,m}$ es de distribución libre.

La región crítica viene dada por: $D_{n,m} \geq D_{n,m,\alpha}$ siendo $D_{n,m,\alpha}$ los correspondientes puntos críticos, tabulados en las tablas de Kolmogorov-Smirnov para dos poblaciones.

Si al aplicar el test la hipótesis de homogeneidad es rechazada, es decir, las poblaciones son diferentes, este test nos permitir averiguar el porqué.

- **Contrastes para la comparación de más de dos poblaciones:** la extensión natural de los objetivos planteados en la sección anterior es la comparación de k ($k > 2$) tratamientos, poblaciones, etc.

Si se representan por X_1, \dots, X_k las k variables aleatorias que se desean comparar, con funciones de distribución, respectivamente, $F_i(x)$ $i = 1, \dots, k$, la hipótesis a analizar es:

$$H_0 : F_1(x) = \dots = F_k(x) \quad \forall x$$

Al igual que en el estudio de dos poblaciones, las técnicas a aplicar dependerán del tipo de variable bajo estudio (cuantitativa o cualitativa) y del tipo de experimento (muestras relacionadas o independientes).

Alguna de las técnicas no paramétricas para este estudio son generalizaciones naturales de las técnicas vistas en la sección anterior, como por ejemplo, el test chi-cuadrado para el estudio de homogeneidad de k poblaciones multinomiales con s categorías cada una, cuando las muestras son independientes.

A continuación se recogen una de las técnicas para el estudio del contraste anterior cuando las variables a analizar son continuas.

- **Test de Kruskal-Wallis:** la técnica no paramétrica más usual, para el contraste anterior, cuando se trabaja con muestras independientes.

El procedimiento a seguir por esta técnica es:

- Se ordenan conjuntamente (de menor a mayor) las observaciones muestrales.
- Se asignan los rangos (de forma natural) a dichas observaciones.
- Se calcula para cada muestra:

R_i = suma de los rangos que le ha correspondido a la i -ésima muestra.

Los estadísticos R_i , $i = 1, \dots, k$ cumplen la siguiente relación:

$$\sum_{i=1}^k R_i = N(N+1)/2$$

siendo $N = \sum_{i=1}^k n_i$ = número total de observaciones muestrales, donde por n_i , se representa el tamaño de la i -ésima muestra, $i = 1, \dots, k$

- Si H_0 es cierta, la suma total de rangos deberá estar repartida proporcionalmente entre las k muestras, en función de sus tamaños, por lo tanto el estadístico propuesto es:

$$\sum_{i=1}^k \left(R_i - n_i \frac{N+1}{2} \right)^2$$

por lo que la hipótesis nula, será rechazada para valores grandes del estadístico.

- Para el cálculo de los puntos críticos, previamente a dicho estadístico hay que aplicarle una modificación (tipificación), obteniéndose el siguiente estadístico:

$$H = \frac{12}{N(N+1) \sum_{i=1}^k n_i} \frac{R_i^2}{n_i} - 3(N+1)$$

- La región crítica viene dada por $H \geq H_{n_1, \dots, n_k, 1-\alpha}$.

Consideraciones:

- Desde el punto de vista práctico, en caso de empate entre observaciones de diferentes muestras, a cada una de ellas se le asigna rango medio.
- Asintóticamente, el estadístico H se distribuye, bajo H_0 , según una χ_{k-1}^2 . En esta situación, la región crítica vendrá dada por $H \geq \chi_{k-1, 1-\alpha}^2$
- Si la hipótesis nula es rechazada, se debería de recurrir a alguna técnica de comparaciones múltiples, como por ejemplo, el test de Kruskal-Nemenyi, para poder obtener más información sobre los diferentes tratamientos.

2.1.4. Cuarto curso

De cuarto curso se ha programado una pequeña aplicación representado una pequeña parte de lo que es posible hacer con la asignatura de inteligencia artificial y estadística. Esta aplicación ha seguido los pasos típicos de la ciencia de datos, donde se enumeraran con una breve descripción tanto genérica como de la parte correspondiente al trabajo de inteligencia artificial y estadística

1. Definición del problema: se debe analizar la situación y definir claramente el problema que se quiere abordar. En el trabajo la definición consistió en comprobar si a partir de un modelo estadístico es posible clasificar o predecir bien los niveles de obesidad, o medir bien el índice de masa corporal a partir de una serie de respuestas. Nótese que aunque en realidad el índice de masa corporal ya tiene una fórmula

$$IMC = \frac{\text{peso}}{\text{altura} \times \text{altura}},$$

quitando sentido a crear un modelo para desentrañar este cálculo del IMC a partir de ejemplos, el plantear este sencillo estudio permite al usuario seguir de forma asequible el aprendizaje del proceso y las técnicas en sí, y ver su capacidad para obtener un modelo que prediga correctamente la variable objetivo.

2. Recopilación de datos. Asegurarse de que tiene todos los datos relevantes que se necesitan para resolver el problema. Los datos se tomaron de la siguiente fuente: <https://archive.ics.uci.edu/>
3. Análisis exploratorio de datos. Ayuda a comprender mejor los datos, detectar patrones y tendencias para identificar relaciones entre diferentes variables. Los datos fueron depurados, se eliminaron observaciones con varias respuestas faltantes, se agruparon observaciones con poco volumen para hacerla más representativas, etc.
4. Modelización. Es la creación de un modelo de predicción que se usará para pronosticar o clasificar sobre una base de datos nueva. Para construir el modelo se deben elegir técnicas matemáticas desde la estadística inferencial, junto con otras de IA. Se hizo una división de los datos en conjunto de prueba y entrenamiento con un “`initial_split()`”, una ingeniería de características para eliminar variables con varianza nula, clasificando los datos categóricos faltantes “NA” como “desconocidos” y reemplazando observaciones numéricas con la mediana. Se especificaron los modelos de regresión y clasificación con “`decision_tree()`” del paquete “`rpart`” haciendo tuning de hiperparámetros e integrándose en un workflow. Se comenzó con un grid aleatorio y luego se utilizó un `grid_regular` con ayuda de `filter`. se analizaron las métricas con la validación cruzada y con el tuning y se seleccionaron a los mejores candidatos.
5. Evaluación. Una vez que se ha entrenado el modelo, se deben evaluar su capacidad para predecir resultados con un nivel de precisión. Posteriormente se contrastó el rendimiento sobre el conjunto de prueba.
6. Despliegue: Si el modelo satisface los requisitos de calidad, entonces puede ser desplegado como una aplicación en producción para resolver el problema empresarial para el que fue creado. Este paso no se llevó a cabo porque se trata de plantear un estudio sencillo sobre técnicas de IA.

2.2. Herramientas software

Una vez que se ha definido la funcionalidad, se puede comenzar el proceso de diseño y desarrollo. Para tal fin se pueden utilizar una variedad de lenguajes de programación, bibliotecas y frameworks, dependiendo de los requisitos y preferencias. Algunas opciones comunes incluyen Python, con las bibliotecas de análisis de datos Pandas, NumPy, SciPy, ScikitLearn, Matplotlib, Keras, Tensorflow y Pytorch, o R, con los ecosistemas de paquetes Tidyverse y Tidymodels, entre muchos otros.

Las opciones para crear un panel de aplicación estadística a día de hoy son incontables, tanto gratuitas como de pago. Pese a todas las opciones, la solución ha sido desarrollar

dicho cuadro de mandos con R, un lenguaje de programación para computación estadística y visualización gráfica, y como entorno de desarrollo integrado (IDE) RStudio. Este entorno nos aporta el hecho de tener un uso muy ampliamente extendido, con una comunidad enorme detrás, y facilita visualmente muchas tareas como la adición de librerías o la importación de datos, que amplían la funcionalidad existente, a parte de estar bajo licencia de software libre. Teniendo presente lo anterior, se ha hecho uso de este lenguaje debido a la versatilidad que posee, ya que permitía tanto un tratamiento en profundidad de los datos como la posibilidad de crear interfaces operativas.

De entre las distinguidas librerías usadas en el trabajo cabe destacar las siguientes

- **renv**, para hacer el proyecto más aislado, portátil y reproducible pues, crea una biblioteca privada de paquetes R, por lo que puede actualizar y cambiar las versiones de los paquetes en un proyecto sin preocuparse por romper otros proyectos. Captura el estado de las paquetes R dentro de un archivo de bloqueo, de forma que puede compartir y colaborar más fácilmente en proyectos con otros, y asegurarse de que todos trabajen desde una base común. Al guardar el estado de la biblioteca en el archivo de bloqueo, permite restaurar la biblioteca R exactamente como se especifica en dicho archivo.
- **flexdashboard**, que es una de las librerías que se encarga de generar la interfaz gráfica con la que el cliente va a interactuar. Ha facilitado la creación empleando Rmarkdown de los paneles interactivos, siendo compatible con una amplia variedad de componentes como el uso de la barra de navegación y librerías. Usa diseños basados en filas y columnas flexibles fáciles de especificar, diseños basados en storyboards para presentar secuencias de visualizaciones y comentarios relacionados, diseños basados en columnas de fichas, que muestra conjunto de pestañas, diseños basados en múltiples páginas. Los componentes se redimensionan de forma inteligente para llenar el navegador.
- **Shiny**, la librería fundamental para la interacción con servidor en R, que se encarga de generar la interfaz gráfica con la que el cliente va a interactuar, dotando al tablero creado por flexdashboard de visualizaciones dinámicas diferentes. La aplicación shiny tiene tres componentes esenciales: un objeto de interfaz usuario, una función de servidor y una llamada a la aplicación. El objeto de la interfaz de usuario controla el diseño y la apariencia, y la función de servidor contiene las instrucciones que la computadora necesita para construir la aplicación, recibiendo parámetros input y output contenidos en una función de render, patrón que se repetirá siempre que se quiera realizar procesamiento en el servidor y generar una salida interactiva. Los input traerán aquellos datos que dependen de la interacción del usuario, y el output se generará a partir de estos como salida. Esto conectará con la salida presente en el usuario. Finalmente, la llamada a la aplicación crea objetos de aplicación shiny a partir de un par explícito de interfaz de usuario/servidor.
- **Shinyjs**, ha permitido realizar operaciones JavaScript sin tener que saber nada de JavaScript como ocultando o mostrando un elemento.
- **DT**, para la representación de tablas interactivas.

- **Tidyverse**, es una colección de paquetes diseñados para la ciencia de datos. Este ecosistema incluye muchos otros paquetes con un uso más especializado. La mayoría se cargan automáticamente con “library(tidyverse)”, los paquetes más habitualmente usados se cargan con esa propia llamada a “library(”.”)”, aunque la instalación de tidyverse incluye paquetes adicionales que es necesario cargar aparte con “library”. Los usados a destacar en la aplicación son:
 - **ggplot2**, basados en la llamada “layered grammar of graphics”, propuesta por Hadley Wickham como aplicación y extensión de la “grammar of graphics” definida por Wilkinson [20].
 - **plotly**, para pasar de gráficos estáticos a interactivos a través de la biblioteca de gráficos JavaScript de código abierto. No forma parte del núcleo de Tidyverse.
 - **dplyr**, que proporciona una “gramática para la manipulación y operaciones con”data.frames”.
 - **tydr**, que permite ordenar datos sin depurar para obtener objetos de datos en R (“tbl_df”, muy parecidos a “data.frame”) en formato ordenado.
 - **readxl**, para la importación de los archivos con extensión “.xlsx”
 - **openxlsx**, para escribir datos en excel.
- **Tidymodels**, que es una colección de paquetes para el modelado y aprendizaje automático utilizando los principios de tidyverse, ha permitido montar la aplicación de la asignatura de inteligencia artificial. Paquetes usados que forman parte de este ecosistema son:
 - **rsample**, para validar los modelos por métodos de resampling.
 - **recipes**, ayuda a realizar ingeniería de características para preparar las datos para el modelado.
 - **parsnip**, es el que se encarga de la especificación de los modelos, llamando internamente a paquetes dedicados a aplicar los distintos tipos de modelos, dependiendo de los motores empleados.
 - **tune**, cuyo objetivo es facilitar el ajuste de hiperparámetros para tener un buen ajuste del modelo.
 - **workflows**, para combinar todos los pasos del preprocesado y modelado en un único objeto.
 - **dials**, que contiene las infraestructuras para crear y administrar las métricas deseadas.
 - **yardstick**, para la evaluación del modelo, dotando de métricas que podemos aplicar tras el aprendizaje, pudiendo dejarlas predeterminadas
- **sampling**, para los estimadores de Horvitz-Thompson y Hajek así como la obtención de muestras.
- **car**, para el test de levene.
- **DescTools**, es una colección extensa de diversas funciones estadísticas básicas y contenedores básicos que no están disponibles en el sistema básico R para una descripción eficiente de los datos. ayudó con el test de los signos y el test de Siegel-Tukey.
- **KScorrect**, para la corrección de lilliefors en el test de kolmogorov-Smirnov de

bondad de ajuste.

- **BSDA**, para el uso del test de los signos.
- **PMCMRplus**, para las comparaciones multiples.

Capítulo 3

Aplicación desarrollada

En primer lugar, este capítulo presenta el principal resultado del trabajo, enfocado en el desarrollo de una aplicación para el alumnado del grado en estadística. Cabe destacar que en dicho grado no se han cursado asignaturas destinadas al desarrollo de un proyecto ni la creación de un producto software. No es el objetivo del mismo el de cubrir la formación que en una ingeniería informática se tendría de aspectos de dirección y planificación del mismo, elección de una metodología de desarrollo de proyectos ni llevar a cabo dentro de la misma de forma exhaustiva fases del ciclo de vida de un proyecto software, estudiados en ingeniería del software, como elicitación de requisitos, análisis, diseño e implementación, posiblemente organizados temporalmente mediante metodologías ágiles como SCRUM. Al igual que tampoco se han estudiado asignatura que tengan que ver con usabilidad o con gestión de la configuración de un proyecto. No obstante, pese a las limitaciones que esos aspectos podrían imponer sobre nuestro desarrollo, también partíamos de muchas fortalezas, incluyendo el haber recibido una formación completa en materias de estadística y matemáticas, objeto del proyecto, y haber trabajado con muchas de las herramientas software que nos planteamos emplear en este desarrollo.

Como se ha introducido en apartados anteriores de la memoria, y pese a no haber estudiado materias relacionadas con usabilidad, el enfoque del trabajo ha tenido en todo momento un trasfondo orientado a satisfacer las necesidades de un usuario final, y en ese sentido fue fundamental conocer la base de los potenciales interesados al cursar cada una de las materias del grado. Con todo esto en mente, nos embarcamos en el proyecto y fuimos desarrollando una aplicación visual estructurada en cursos académicos.

A lo largo de este capítulo, en lugar de orientar la memoria a un público que vea diseños de los elementos software interconectados que formarán parte de la aplicación, lo que se va a tratar es de presentar una muestra de las principales funcionalidades que se han proporcionado, y se acompañará a las explicaciones y las capturas de pantalla ilustrando la misma de explicaciones más técnicas para permitir al lector entender los distintos tipos de elementos tecnológicos incorporados en la aplicación, con algunos fragmentos de código ilustrando el uso de los mismos, de forma que potenciales lectores interesados no solo en utilizar sino también en extender estas herramientas puedan tener una guía sobre cómo trabajar con según qué tipos de elementos. Por ello, no se encuentran detalladas en este capítulo ni mucho menos todas las funcionalidades, sino una muestra de ellas que puedan resultar de interés no únicamente funcional sino también tecnológico, quedando la aplicación completa a disposición para poder consultar y probar toda la funcionalidad

proporcionada.

3.1. Esqueleto de la aplicación

La aplicación toma la forma de un cuadro de mando, desplegado como una aplicación web, con una serie de opciones de menú con diferentes menús desplegables que constan de varias opciones cada uno, y cuyo destino de cada opción lleva a la carga en la misma página principal de las funcionalidades solicitadas. Para ello empleo un tablero de flexdashboard¹ junto con la interactividad y dinamismo de shiny, permitiendo un mecanismo para la introducción de controles visuales, y de salidas que refrescan su contenido cada vez que se modifican los valores seleccionados por el usuario mediante los controles permitiendo así hacer la aplicación aún más atractiva.

Una aplicación de este tipo puede generarse de diversas maneras en [3] se comenta un poco acerca de las opciones de crear cuadros de mando, pero para esta se ha optado por partir de un archivo con extensión .Rmd (que es el tablero flexdashboard integrado con shiny) al que en algunas ocasiones se le añadirá el código necesario para la creación de una asignatura y en otras se le anexionará archivos que le dotarán de contenido al tablero de flexdashboard, este último recurso ha sido el más usado pues así se evita la extensión de un archivo distribuyéndola en varias.

Por tanto uno de las primeras librerías que ha de instalarse es flexdashboard

```
install.packages("flexdashboard")
library(flexdashboard)
```

Posteriormente, acceder a la plantilla de flexdashboard desde la interfaz de Rstudio. A continuación se muestra una batería de imágenes, con recuadros numerados que indican donde clicar, siendo el color rojo el primer clic, el color azul el segundo clic y el color verde el tercer clic, en vez de dejar una descripción.

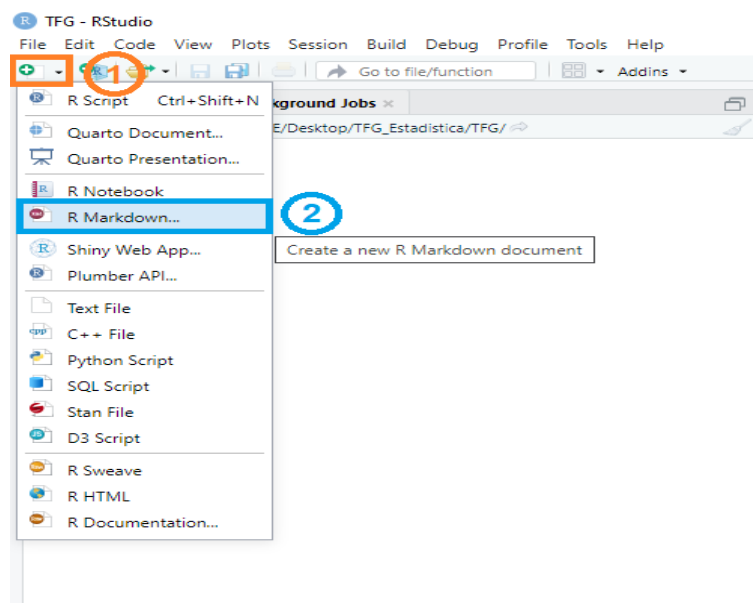


Figura 3.1: Accediendo a plantilla flexdashboard 1ª parte.

¹Para más información acudir a: [14]

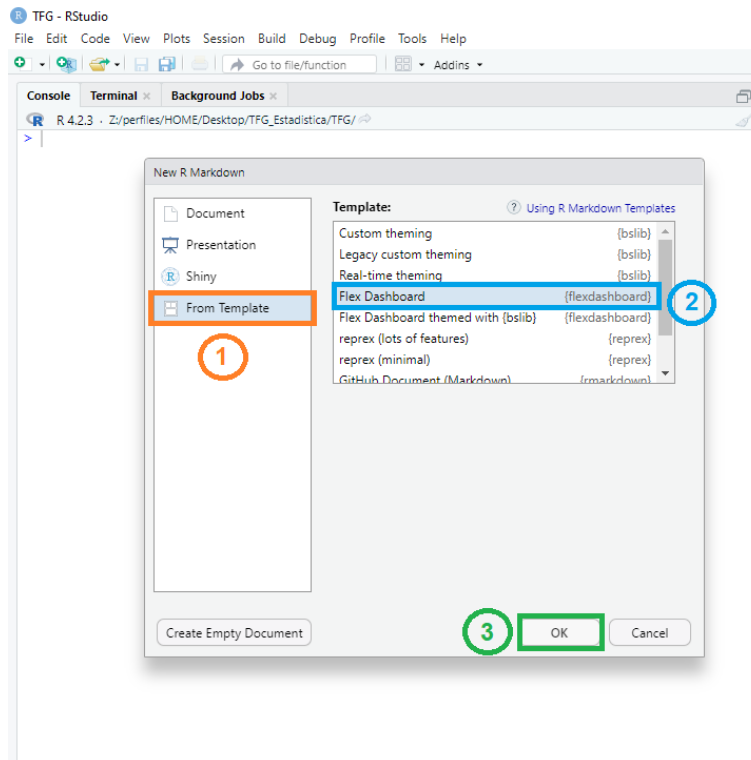


Figura 3.2: Accediendo a plantilla flexdashboard 2ª parte.

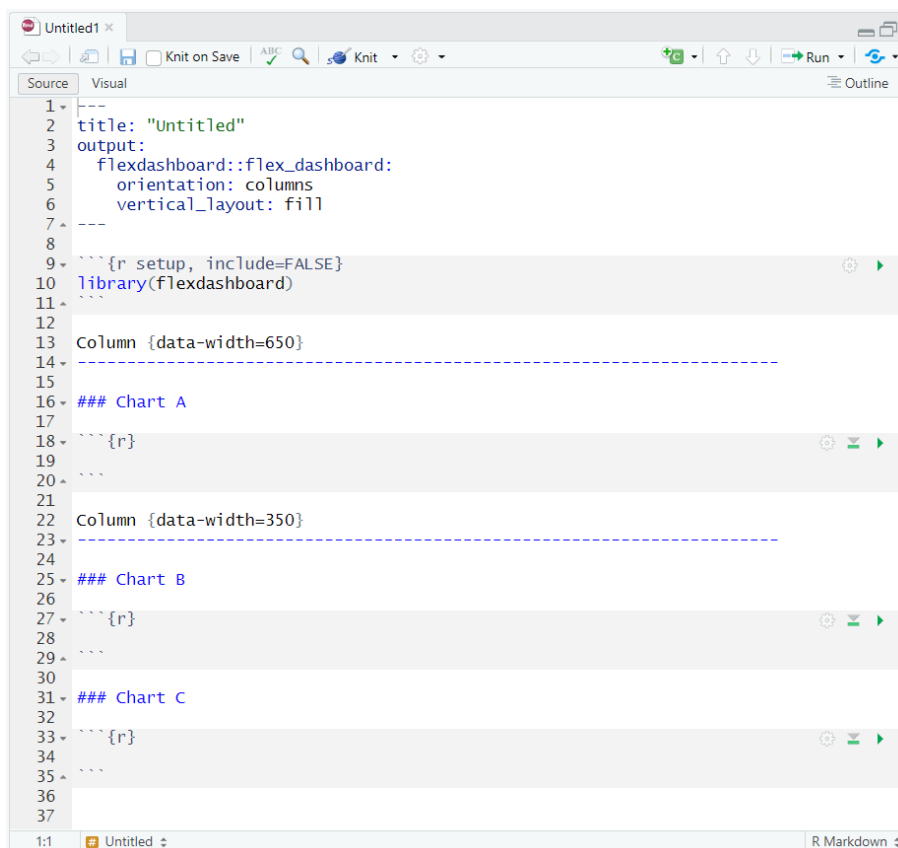


Figura 3.3: Plantilla flexdashboard.

Como consecuencia de la necesidad de interactividad, no se puede generar un informe final en R que sea obtenido, transformado a Markdown y finalmente generado un código html o latex/pdf estático, sino que requerirá la ejecución de código R cada vez que se cambie la entrada por parte del usuario. Por ello, requerirá un servidor R para ejecutar una aplicación Shiny.

Para poder hacer uso de estos mecanismos debemos instalar el paquete, y posteriormente cargarlo, como de costumbre con cualquier otro paquete de R, mediante:

```
install.packages("shiny")  
library(shiny)
```

E indicar en la cabecera YAML el parámetro “runtime: shiny”.

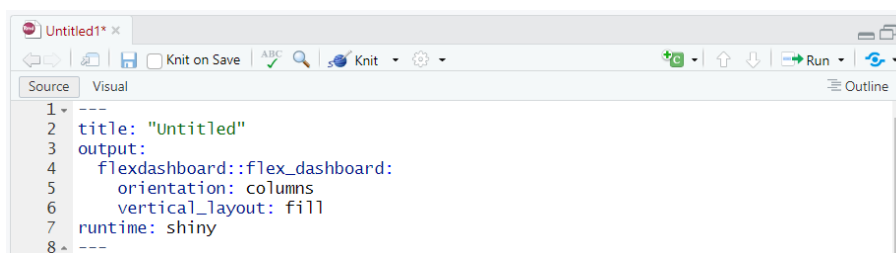


Figura 3.4: Plantilla flexdashboard interactiva.

La plantilla de la Figura 3.4 ha sido modificada para tener una barra de navegación con subpáginas y en cada subpágina tener una trama diferente, y no tener una visualización monótona que cansa la vista del usuario. En la Figura 3.5 puede observarse el esqueleto en el cual se apoya todo el trabajo. En el mismo, cada opción de menú se representa mediante un “#”, las opciones de un mismo menú se agrupan con “data-navmenu”, y cada subpágina se representa mediante tres “###”.

```

1 ---
2 title: "Aplicación estadística"
3 output:
4   flexdashboard::flex_dashboard:
5     source_code: embed
6     navbar:
7       - { title: "Memoria",
8         href: "https://drive.google.com/file/d/1Ebivg1L7m1pZHbmayT5-NVPWVN3C2mD3/view?usp=sharing",
9         icon: "fa-file-pdf-o", target: "_blank"}
10 runtime: shiny
11 ---
12 {r_setup, include=FALSE}
13 library(flexdashboard)
14 library(shiny)
15 ---
16
17 # Introducción {data-navmenu="Inicio" .storyboard}
18
19 ### Presentación
20
21 ### Importación de datos
22
23 # cálculo de probabilidad {data-navmenu="1er curso" .storyboard}
24
25 ### Combinatoria de números
26
27 ### Modelos de variables aleatorias discretas
28
29 ### Vectores aleatorios discretos
30
31 # Estadística descriptiva {data-navmenu="1er curso" data-orientation=rows}
32
33 # Teoría de la probabilidad I {data-navmenu="1er curso" data-orientation=rows}
34
35 # Muestreo estadístico {data-navmenu="2º curso" data-orientation=rows}
36
37 # Teoría de la probabilidad II {data-navmenu="2º curso" data-orientation=rows}
38
39 # Ampliación de inferencia estadística {data-navmenu="3er curso" data-orientation=rows}
40
41 # Analisis multivariante {data-navmenu="3er curso" data-orientation=rows}
42
43 # Modelos lineales {data-navmenu="3er curso" data-orientation=rows}
44
45 # Diseño de experimentos {data-navmenu="4º curso" data-orientation=rows}
46
47 # Inteligencia artificial y estadística {data-navmenu="4º curso" data-orientation=rows}
48

```

Figura 3.5: Esqueleto de la aplicación.

Tras esta última figura, puede observarse el empleo de elementos como el “data-orientation=rows” o el “.storyboard”. El primero establece la orientación de la página a filas, pues por defecto usa la orientación predeterminada de “columnas”. Por su parte, el segundo elemento es para crear un diseño de guión gráfico. Como puede verse también, hay hasta 5 opciones principales de menú que conducirán a sendas páginas, que como vemos con los “data-navmenu” se corresponden con la sección “Inicio” y los 4 cursos académicos del grado. Finalmente, se describirán a continuación con mayor detalle las funcionalidades, empezando por el menú descrito en este esqueleto y que se verán al comienzo de la sección 3.2.

3.2. Esquema general

El esquema general de los contenidos de la aplicación puede apreciarse desde la barra de navegación principal de color azul que trae por defecto el tablero de flexdashboard. En el extremo izquierdo está situado el nombre de la aplicación, que en este caso es “Aplicación estadística”, y las cinco agrupaciones de páginas que contienen subpáginas relacionadas

con el nombre de la agrupación, como se han detallado en el esqueleto anterior. Por último, en el extremo derecho hay colocado dos textos. El primero “Memoria” con el icono de un archivo pdf que contiene la memoria de este trabajo, a través de un enlace que siempre está disponible. El segundo “Source Code”, junto con el icono “</>” que contiene el código del esqueleto de la aplicación, pero no del resto de páginas y subpáginas que son aplicaciones independientes las cuales son llamadas mediante varios argumentos. A continuación puede observarse la barra de navegación.

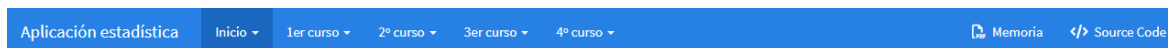


Figura 3.6: Barra de navegación.

El código para desarrollar esta barra de navegación es el indicado en el esqueleto anteriormente comentado, que puede apreciarse en la Figura 3.5.

Las agrupaciones de las páginas visualizadas de izquierda a derecha, así como su contenido son las siguientes:

- **Inicio**, contiene una introducción clasificada en dos apartados, un texto de bienvenida y agradecimiento por hacer uso de la aplicación y una texto explicativo sobre la importación de los datos a la misma.

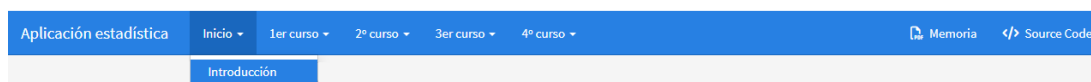


Figura 3.7: Pestaña Inicio. Opciones de menú incluidas.

Como código a destacar en esta sección es la colocación de una galería de imágenes de forma que vaya deslizándose una imagen para dar paso a la siguiente tras clicar en su correspondiente lugar

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0,961624443	6	18	26									
2	0,771186596	8	17	26									
3	0,199579031	9	12	22									
4	0,23200587	9	12	27									
5	0,578656236	7	11	28									
6	0,782350115	6	15	24									
7	0,491724472	3	11	26									
8	0,428410787	1	18	22									
9	0,651758839	2	12	30									
10	0,952554855	5	16	23									
11	0,116403757	1	13	25									
12	0,997249118	2	12	29									
13	0,516302463	6	19	26									
14	0,406754596	8	18	25									
15	0,849572251	10	17	25									
16	0,30648424	8	11	29									
17	1ª	2ª	3ª	4ª									
18	Columna	Columna	Columna	Columna									
19													
20													
21													
22													

Figura 3.8: Galería de imágenes vista usuario.

```

1 library(shiny)
2 library(slickR)
3 |
4 ui <- fluidPage(
5   slickRoutput("slickr", width = "90%")
6 )
7
8 server <- function(input, output) {
9
10  output$slickr <- renderslickR({
11    imgs <- list.files("imagenes/", pattern=".png", full.names = TRUE)
12    slickR(imgs,width = "80%")
13  })
14
15 }
16
17 shinyApp(ui = ui, server = server)

```

Figura 3.9: Código R de galería de imágenes.

Como se explicó al final del inicio de este capítulo,³ en esta aplicación, los bloques y subbloques están separados y se integran dinámicamente en un dashboard, haciendo posible presentar funcionalidades de cada bloque y subbloque, como esta galería de imágenes, en una aplicación de muestra. Por lo que se irán exponiendo algunas figuras sobre contenidos del bloque/subbloque y sobre códigos de aplicación shiny. De las figuras sobre códigos, se comentarán aquellos aspectos que no hayan aparecidos anteriormente, aunque siendo esta la primera figura con código recordaré lo comentado en la sección 2.2: que en toda aplicación de shiny hay tres componentes esenciales.

- El objeto de interfaz usuario que se corresponde con el "ui" controlando el diseño y la apariencia, que en este y en todos los demás casos está generando un estilo de página "fluidPage()" (ya explicado en el apartado anterior) y dentro de ella está ejecutando y creando un hueco para el slickROutput, que en la función servidor le indicaremos que queremos que muestre.
- Posteriormente, la función de servidor que se corresponde con el "server" contiene las instrucciones que la computadora necesita para construir la aplicación. La función "server" de esta miniaplicación va a contener la generación de una salida interactiva a través de una función de render, patrón que se repetirá siempre que se quiera realizar procesamiento en el servidor y generar una salida interactiva que previamente se conecta con el input del usuario mediante el "output\$slickr".
- Finalmente, la llamada a la aplicación "shinyApp(ui,server)" crea objetos de aplicación shiny a partir del par explícito de interfaz de usuario/servidor.

No se entrará en grandes detalles sobre la estructura de una aplicación shiny.²

- **1er curso**, contiene el nombre de tres asignaturas impartidas de primer curso del grado en estadística, estando dos de ellas deshabilitadas (estadística descriptiva y teoría de la probabilidad I).

²Para documentarse sobre una aplicación shiny, acudir a: <https://shiny.posit.co/r/articles/start/basics/>
[3]

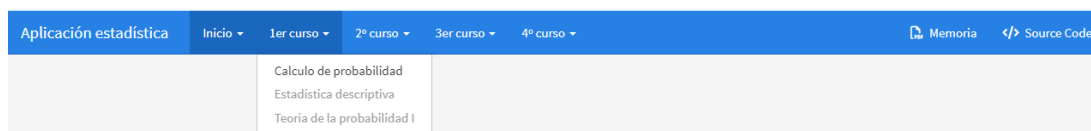


Figura 3.10: Pestaña 1^{er} curso. Opciones de menú incluidas.

- **2º curso**, contiene el nombre de dos asignaturas impartidas de segundo curso del grado en estadística y solo estando una operativa (teoría de la probabilidad II)

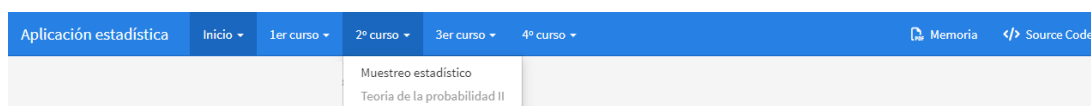


Figura 3.11: Pestaña 2º curso. Opciones de menú incluidas.

- **3er curso**, contiene el nombre de tres asignaturas impartidas de tercer curso del grado en estadística, estando dos de ellas desactivadas (análisis multivariante y modelos lineales)

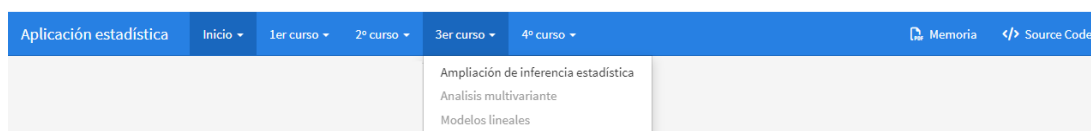


Figura 3.12: Pestaña 3^{er}. Opciones de menú incluidas. curso.

- **4º curso**, contiene el nombre de dos asignaturas impartidas de cuarto curso del grado en estadística y estando una de ella inactiva (diseño de experimentos)

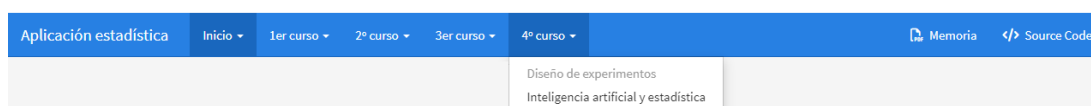


Figura 3.13: Pestaña 4º. Opciones de menú incluidas. curso.

3.3. Descripción curso a curso

En esta sección se van a ir describiendo en cierto detalle todos aquellos elementos que se consideren únicos, a nivel tecnológico, dentro del trabajo, y aquellos que se consideren más interesantes desde el punto de vista funcional, tratando de evitar el contenido repetitivo para el lector, que siempre podrá acudir a la aplicación para poder aprovechar toda la funcionalidad.

Aunque antes de entrar en materia puede observarse que en todas las pestañas hay asignaturas deshabilitadas esto nos sirve para ilustrar posibles planes futuros de expansión de la funcionalidad, que pueden llevarse a cabo próximamente pero aún no se encuentren implementadas. Esto se ha conseguido gracias a las funciones “tags”³ de shiny junto con

³Para más información sobre la función tags, ver: <https://shiny.posit.co/r/articles/build/tag-glossary/> [3]

los conocimientos de la asignatura de bases de datos que permiten el manejo de objetos del documento. Por tanto, observando qué nombre por defecto se le ha puesto al atributo “class” de las subpáginas por medio de la inspección de elementos (para ello se clica con el segundo botón del ratón y se presiona sobre “Inspect Element”), se busca en la venta “Elements” dicho nombre, y mediante lenguaje javascript se capturan los elementos necesarios y se modifican sus atributos:

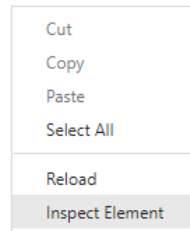


Figura 3.14: Ventana tras clic del segundo botón del ratón en la aplicación.

La colocación del código es al final del archivo .Rmd, de forma que cargue todos los objetos y se puedan acceder a los elementos. El código que se ejecutó para ello es el siguiente:

```
fluidRow(
  tags$script(type="module",
    "document.getElementsByClassName('dropdown-item')[2].href='#'
    document.getElementsByClassName('dropdown-item')[3].href='#'
    document.getElementsByClassName('dropdown-item')[5].href='#'
    document.getElementsByClassName('dropdown-item')[7].href='#'
    document.getElementsByClassName('dropdown-item')[8].href='#'
    document.getElementsByClassName('dropdown-item')[9].href='#'
    document.getElementsByClassName('dropdown-item')[2].parentNode.className='disabled'
    document.getElementsByClassName('dropdown-item')[3].parentNode.className='disabled'
    document.getElementsByClassName('dropdown-item')[5].parentNode.className='disabled'
    document.getElementsByClassName('dropdown-item')[7].parentNode.className='disabled'
    document.getElementsByClassName('dropdown-item')[8].parentNode.className='disabled'
    document.getElementsByClassName('dropdown-item')[9].parentNode.className='disabled'"
  )
)
```

La opción de añadir ‘type=“module”’ es para declarar la secuencia a un nivel superior; sin esta acción, no es posible lograr que la opción de menú quede correctamente deshabilitada.

3.3.1. Primer curso

Cuando se accede al grado en estadística, una de las primeras asignaturas que se cursan en el primer cuatrimestre es el *Cálculo de Probabilidades*. Algunos de los principales contenidos que se tratan en esta asignatura son los siguientes:

- Las estructuras combinatorias básicas, de forma que cuentan cuántos elementos hay en ciertas estructuras formadas a partir de un conjunto finito.
- Los modelos de variables aleatorias discretos, pues en la práctica hay situaciones experimentales que aparecen frecuentemente y se ajustan razonablemente bien a algunos de los modelos.

- Cálculos con vectores aleatorios discretos.

Todos estos contenidos, así como las bases para su aplicación están descritas en la sección 2.1.1.

Como funcionalidades desarrolladas dentro de la asignatura de *Cálculo de Probabilidades*, cabe destacar:

- Las estructuras combinatorias con un ejemplo práctico. Para utilizar estas estructuras, solo hay que seleccionar la operación deseada e introducir el/los parámetro/s correspondiente/s y la aplicación devolverá el resultado. Las estructuras combinatorias disponibles son:
 - Permutación sin repetición.
 - Permutación con repetición.
 - Permutación circular.
 - Permutación de longitud n .
 - Combinación sin repetición.
 - Todas las combinaciones posibles de N elementos.

Una vista previa de este subapartado es la Figura 3.15.

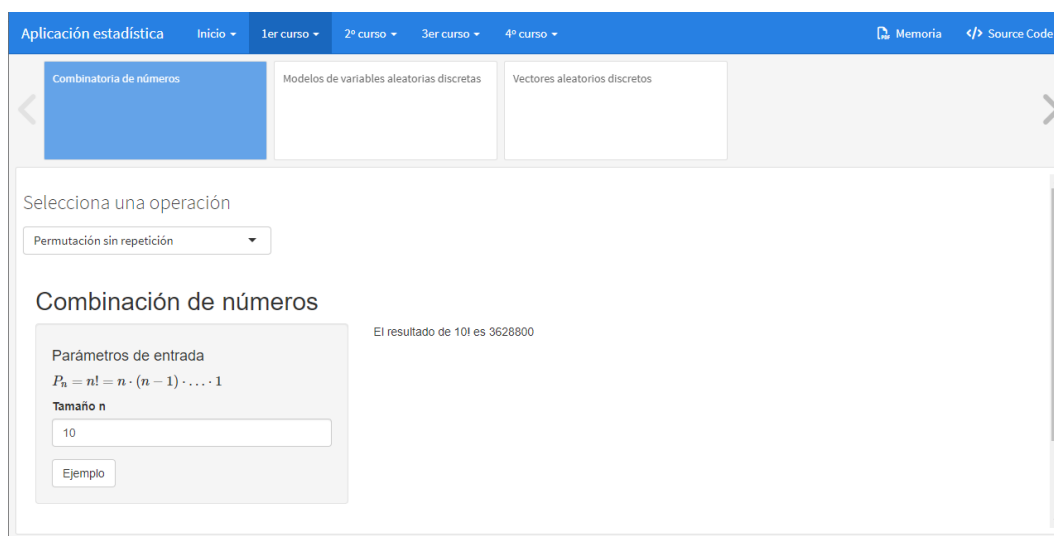


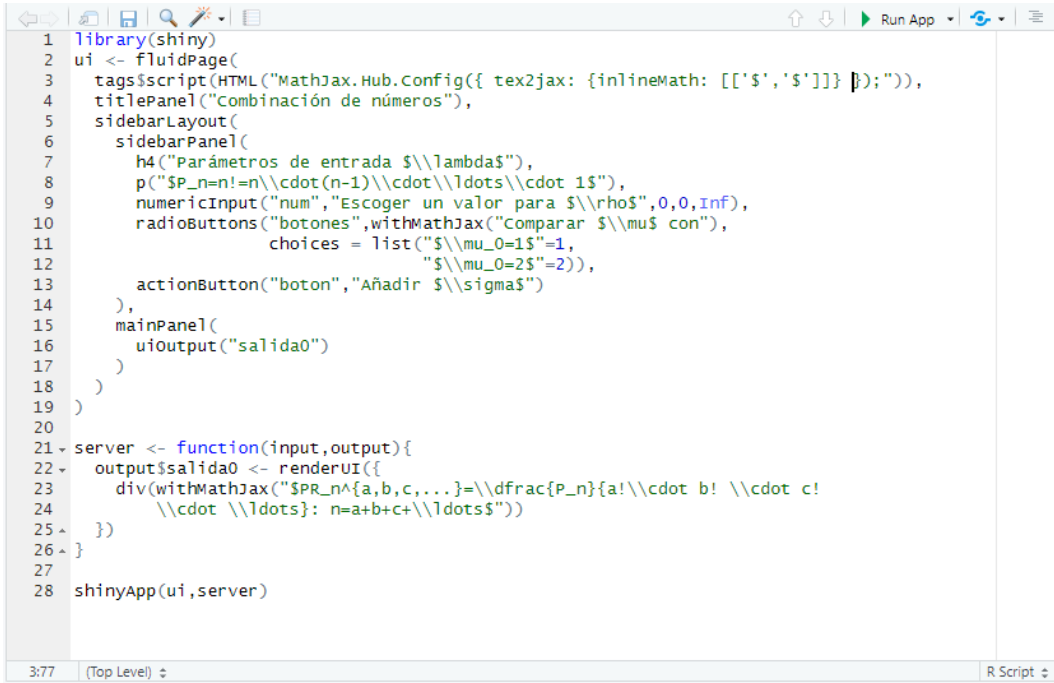
Figura 3.15: Vista previa del diseño de los n° combinatorios.

En esta pantalla si el usuario quisiera obtener el resultado de alguna de las estructuras combinatorias anteriormente mencionadas, solo tendría que clicar en "Selecciona una operación" y posteriormente introducir el valor correspondiente en su casilla. Si quisiera obtener un ejemplo práctico que le sirve de ayuda sobre esa estructura combinatoria pincharía en el botón "Ejemplo" y obtendría el resultado.

En la Figura 3.15 la estructura que se muestra en el bloque superior es la que se genera mediante el ".storyboard" ilustrado en la Figura 3.5, y da una medida de secuencialidad, de modo que es esperable que el usuario vaya pasando por los contenidos de izquierda a derecha si quiere realizar un avance progresivo en su aprendizaje.

Códigos a destacar, en esta subsección son:

- El uso de l atex tanto en la interfaz usuario, ya sea en una etiqueta (exceptuando las opciones de un "selectInput"), t ıtulo o texto como en el servidor. Una muestra de c odigo es la que se proporciona en la Figura 3.16,



```

1 library(shiny)
2 ui <- fluidPage(
3   tags$script(HTML("MathJax.Hub.Config({ tex2jax: {inlineMath: [['$','$']] } });")),
4   titlePanel("Combinaci n de n umeros"),
5   sidebarLayout(
6     sidebarPanel(
7       h4("Par metros de entrada  $\lambda$ "),
8       p("SP_n=n!=n! * (n-1)! * ... * 1"),
9       numericInput("num", "Escoger un valor para  $\rho$ ", 0, 0, Inf),
10      radioButtons("botones", withMathJax("Comparar  $\mu$  con"),
11        choices = list(" $\mu_0=1$ ",
12          " $\mu_0=2$ "),
13      actionButton("boton", "A adir  $\sigma$ ")
14    ),
15    mainPanel(
16      uiOutput("salida0")
17    )
18  )
19 )
20
21 server <- function(input, output){
22   output$salida0 <- renderUI({
23     div(withMathJax("SP_n^{a,b,c,...} = \frac{P_n}{a! * b! * c! * ...} : n=a+b+c+..."))
24   })
25 }
26 }
27
28 shinyApp(ui, server)

```

Figura 3.16: C odigo R para incluir latex en shiny.

En la Figura 3.16 vemos que se establece la herramienta particular como una p gina fluida (adaptable en tama o din amicamente) con una disposici n en columna lateral y principal, lo que realiza sidebarLayout y sus correspondientes panel sidebar y main. A modo de muestra, se puede observar un ejemplo de entrada de usuario, con el numericInput, radioButtons y actionButtons con sem ntica l atex incluida en ellas.

La instrucci n

```
tags$script(
  HTML(
    "MathJax.Hub.Config({ tex2jax: {inlineMath: [['$','$']] } });"))
```

a ade un script que configura el modo matem tico en l nea, pues por defecto solo est  configurado el modo matem tico resaltado que usa dos s mbolos d lar para abrir y dos para cerrar, y en vez de usa una barra diagonal deben usarse dos.

- El bot n "Ejemplo" que si se clica sobre  l se mostrar /ocultar  un ejemplo, inici ndose siempre en oculto. Esto ha sido posible con la ayuda de la librer a shinyjs. Unas instrucciones m nimas para la ejecuci n de esta funcionalidad pueden observarse en la siguiente Figura 3.17.

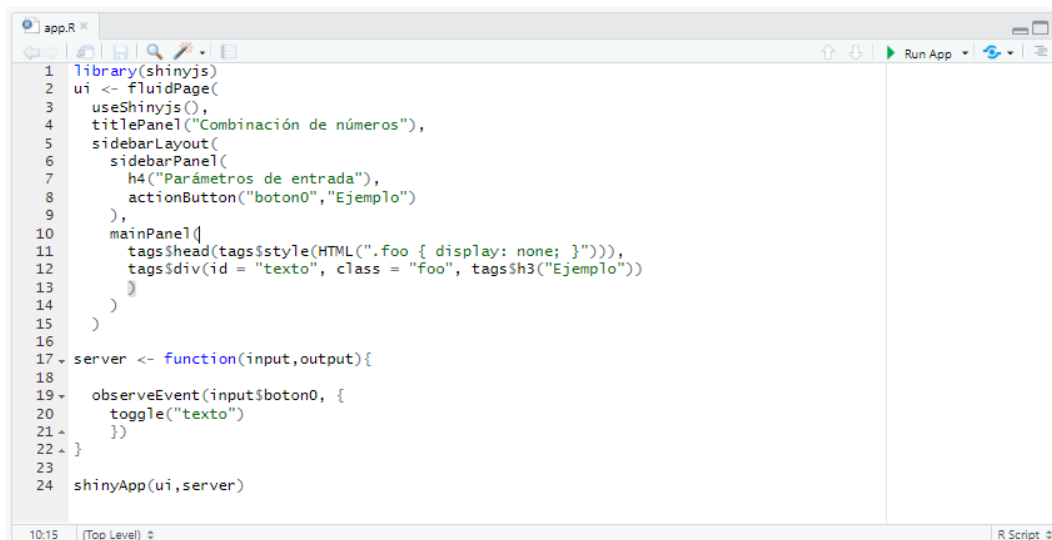


Figura 3.17: Código R de botón que oculta contenido, iniciándose en oculto.

La función "useShinyjs()" dota a la página que se crea de interactividad de forma que se le puedan modificar atributos a los objetos ya creados. Para inicializar el texto como oculto hay que decirselo, esto es lo que hace el siguiente trozo de código

```

tags$head(
  tags$style(
    HTML(".foo { display: none; }"))
)

```

que crea en la cabecera de la página una instrucción de estilo que contenga el atributo "class" del objeto que se desea modificar seguido de "display:none" propiedad para indicarle que inicie en oculto el contenido

- La salida del resultado de la operación es dinámica, es decir, si el valor del parámetro cambia, también cambia su salida, en la siguiente Figura 3.18 puede verse más claramente, teniendo en cuenta que en la Figura 3.10 se calculó el valor para 10! y ahora está para 9! además de clicar sobre el botón "Ejemplo".

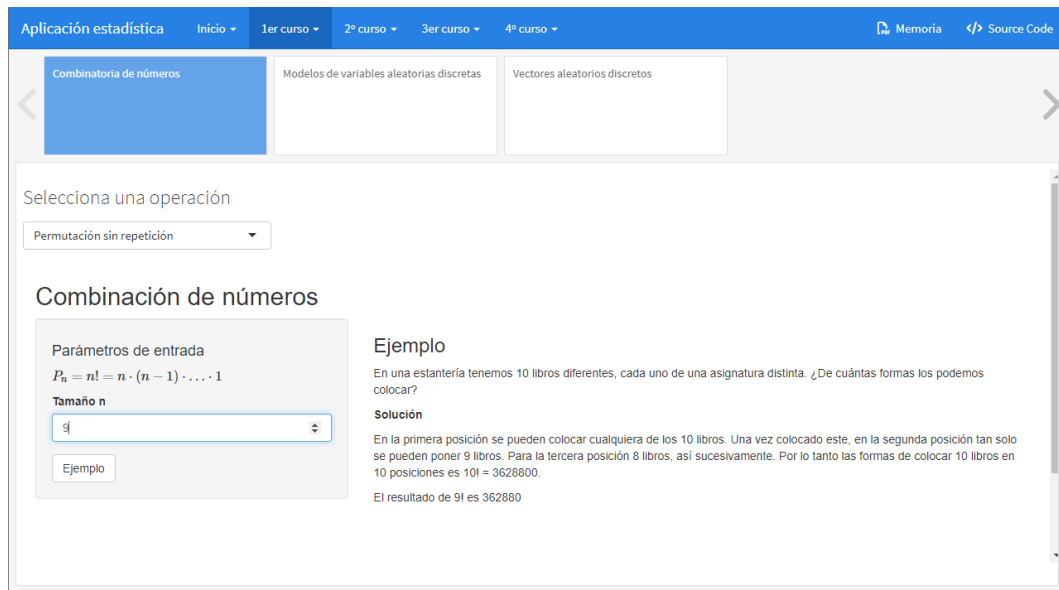


Figura 3.18: Vista previa del diseño de los n° combinatorios cambiando el parámetro.

Instrucciones de código como muestra para lograr dicha interactividad se muestran en la siguiente Figura 3.19

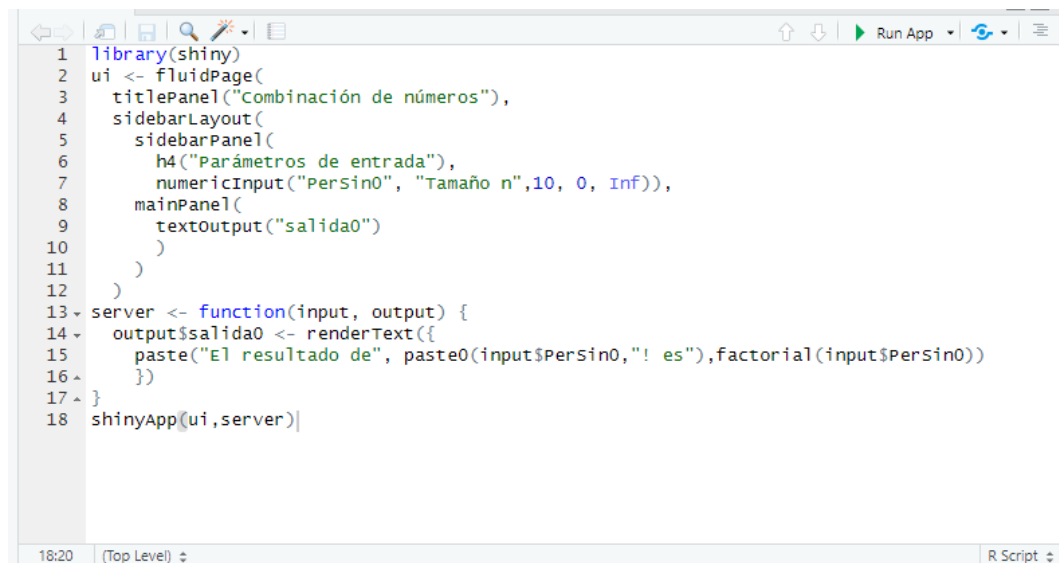


Figura 3.19: Código R para obtener salida dinámica.

- La captura de varios números introducidos por el usuario a través de un "textInput" y convertirlos en un vector numérico pues, por defecto se almacenan como carácter y no se pueden operar con ellos, para ello, hay que realizar un pequeño tratamiento de datos. En la siguiente Figura 3.20 puede apreciarse una pequeña aplicación de cómo es este tratamiento de los datos, para pasar de carácter a numérico a través de un "textInput".

```

1 library(shiny)
2 ui <- fluidPage(
3   titlePanel("Combinación de números"),
4   sidebarLayout(
5     sidebarPanel(
6       h4("Parámetros de entrada"),
7       textInput("text0","Introduce números separados por una coma (,)")
8     ),
9     mainPanel(
10      uiOutput("salida0")
11    )
12  )
13 )
14
15 server <- function(input,output){
16
17   output$salida0 <- renderUI({
18     a=as.numeric(strsplit(c(input$text0),",")[[1]])
19     if (input$text0=="" || any(is.na(a))){
20       div("Escribe números separados por una coma", input$text)
21     }else if(is.numeric(a)){
22       div("Has escrito bien los números","",paste(a, collapse = ", "),"suman ",sum(a))
23     }
24   })
25 }
26
27 shinyApp(ui,server)

```

Figura 3.20: Código R para cambiar la clase de un textInput.

La instrucción

```
a = as.numeric(
  strsplit(c(input$text0),",")[[1]])
```

captura mediante el “input\$text0” lo que el usuario escribe en un vector cadena de longitud uno. Al indicarle al usuario que separe por comas, la función “strsplit” divide la cadena en subcadenas con argumentos divididos, haciendo divisiones, con el segundo argumento que en este caso es la coma (“,”), de forma que almacena en una lista un vector carácter de longitud igual a la cantidad de número que haya introducidos correctamente separados por comas. Finalmente con la función “as.numeric” convierte el vector carácter en vector numérico.

- Los diferentes modelos de variables aleatorios discretos (uniforme, bernouilli, binomial, etc.) así como sus momentos, varianzas y probabilidades, los cuales fueron introducidos en la sección 2.1.1

Una imagen que permite ver el diseño de este subbloque es la Figura 3.21

Aplicación estadística Inicio 1er curso 2º curso 3er curso 4º curso Memoria Source Code

Combinatoria de números Modelos de variables aleatorias discretas Vectores aleatorios discretos

Selecciona un modelo
Modelo uniforme

Modelos de variables aleatorias

Parámetros de entrada
Valores de $x_{(1)}$ y $x_{(N)}$
1 10 500

Momento de orden:
2

Probabilidad igual a:
1

Probabilidad menor o igual que:
1

Probabilidad mayor o igual que:
1

Show 10 entries Search:

	Valor	Probabilidad
1	1	0.1
2	2	0.1
3	3	0.1

Showing 1 to 10 of 10 entries Previous 1 Next

El valor esperado es 5.5
La varianza es 8.25
El momento de orden 2 es 38.5
La probabilidad de que sea igual a 1 es 0.1
La probabilidad de que sea menor o igual que 1 es 1
La probabilidad de que sea mayor o igual que 1 es 1
La probabilidad de que sea menor que 2 es 0.1
La probabilidad de que sea mayor que 2 es 0.8
La probabilidad de que esté entre [1 , 5] es 0.4
La probabilidad de que esté entre (1 , 5) es 0.4
La probabilidad de que esté entre [1 , 5] es 0.5
La probabilidad de que esté entre (1 , 5] es 0.3

Figura 3.21: Vista previa del diseño de los modelos aleatorios.

En esta pantalla si el usuario quisiera obtener resultados sobre los modelos anteriormente mencionadas, solo tendría que clicar en "Selecciona un modelo" y posteriormente introducir el valor del parámetro correspondiente en su casilla. En este caso, como se trata de una variable uniforme solo tendría que proporcionar el valor del parámetro $x_{(1)}$ que se corresponde con el valor más pequeño del conjunto de datos y el valor $x_{(N)}$ el más grande del conjunto de datos. Por defecto, están definidos los valores 1 y 10, por lo que obtiene una tabla con la probabilidad de cada valor, continua con el valor esperado, la varianza, el momento de orden 2 o el que se desee, como las distintas probabilidades del conjunto proporcionado. Finalmente se encuentra la representación de las funciones de probabilidad y de distribución, que pueden observarse en cierta forma en la Figura 3.22.

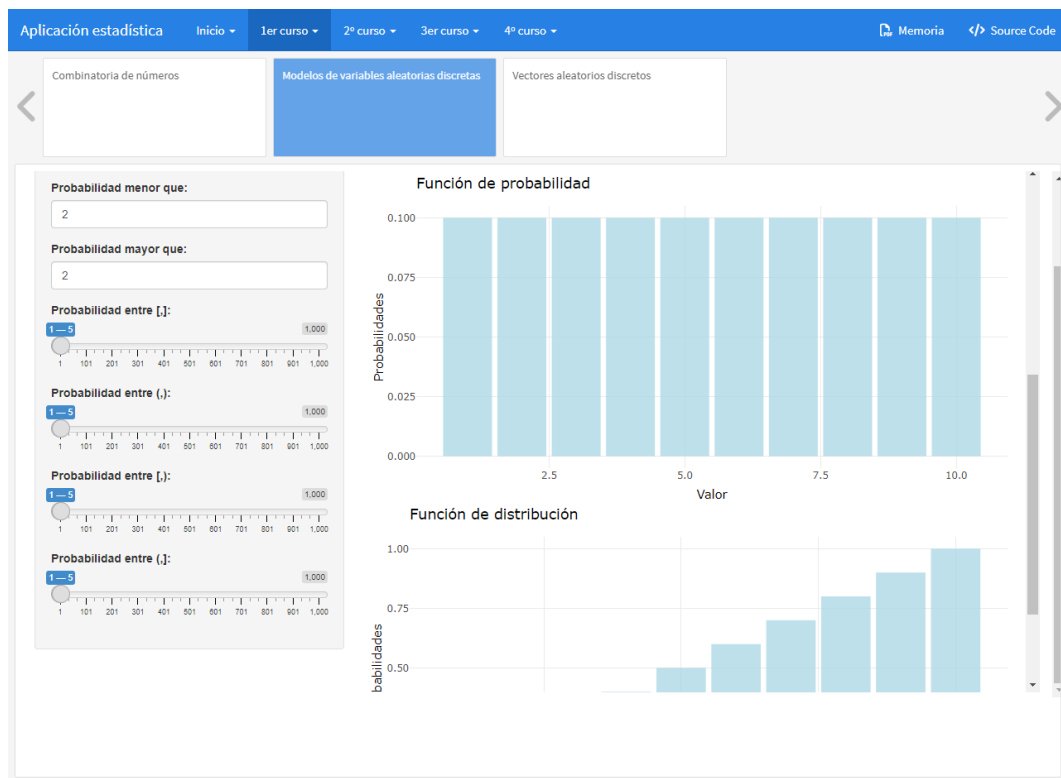


Figura 3.22: Vista previa del diseño de función de probabilidad.

Códigos a destacar de esta parte son:

- La captura de un archivo externo para su posterior manipulación. Para conseguir capturar un archivo externo hay que darle la posibilidad al usuario de que pueda subir dicho archivo a la aplicación y posteriormente generar el código para su manipulación.
- El uso de plotly para obtener gráficos interactivos. Instrucciones de código como muestra para ilustrar estas distinguidas características se observan en la siguiente imagen

```

1 library(shiny)
2 library(plotly)
3 library(readxl)
4 ui <- fluidPage(
5   titlePanel("Combinación de números"),
6   sidebarLayout(
7     sidebarPanel(
8       h4("Parámetros de entrada"),
9       helpText("Cargue un archivo '.xlsx' con los valores de la variable en la primera
10      columna y la probabilidad de cada valor en la segunda columna, sin encabezado"),
11       fileInput("arch0", "Selecciona el excel", accept=".xlsx"),
12       mainPanel(
13         uiOutput("salida0")
14       )
15     )
16   )
17 server <- function(input, output) {
18   path <- reactive({input$sarch0})
19   tabl<-reactive({
20     file <- path()
21     ext <- tools::file_ext(file$datapath)
22     req(file)
23     validate(need(ext == "xlsx", "Por favor cargue un archivo xlsx"))
24     tabla<-read_excel(file$datapath,col_names = c("Valor","Probabilidad"))
25   })
26   output$salida0 <-renderUI({
27     if(!is.null(path())){
28       div(
29         renderDataTable({tabl()}),
30         renderText({paste("El valor esperado es", round(sum(tabl()[1]*tabl()[2]),4))}),
31         renderPlotly({
32           p=ggplot(data=tabl(), aes(x=Valor,y=Probabilidad)) +
33             geom_bar(stat="identity",fill="lightblue",alpha=0.8) + theme_minimal() +
34             labs(title="Función de probabilidad", x="Valor", y = "Probabilidades")
35         })
36       )
37     } else {
38       p("Por favor seleccione un archivo")
39     }
40   })
41 }
42 shinyApp(ui,server)

```

Figura 3.23: Código R para obtener graficos interactivos y manipular archivos externos.

- Con respecto al cálculo con vectores aleatorios, las funcionalidades de la aplicación son el calculo de independencia, de las distribuciones marginales, de sus momentos esperados, varianzas, así como la relación entre ambos.

En la Figura 3.24 puede observarse el diseño que se muestra cuando un usuario accede a este subbloque.

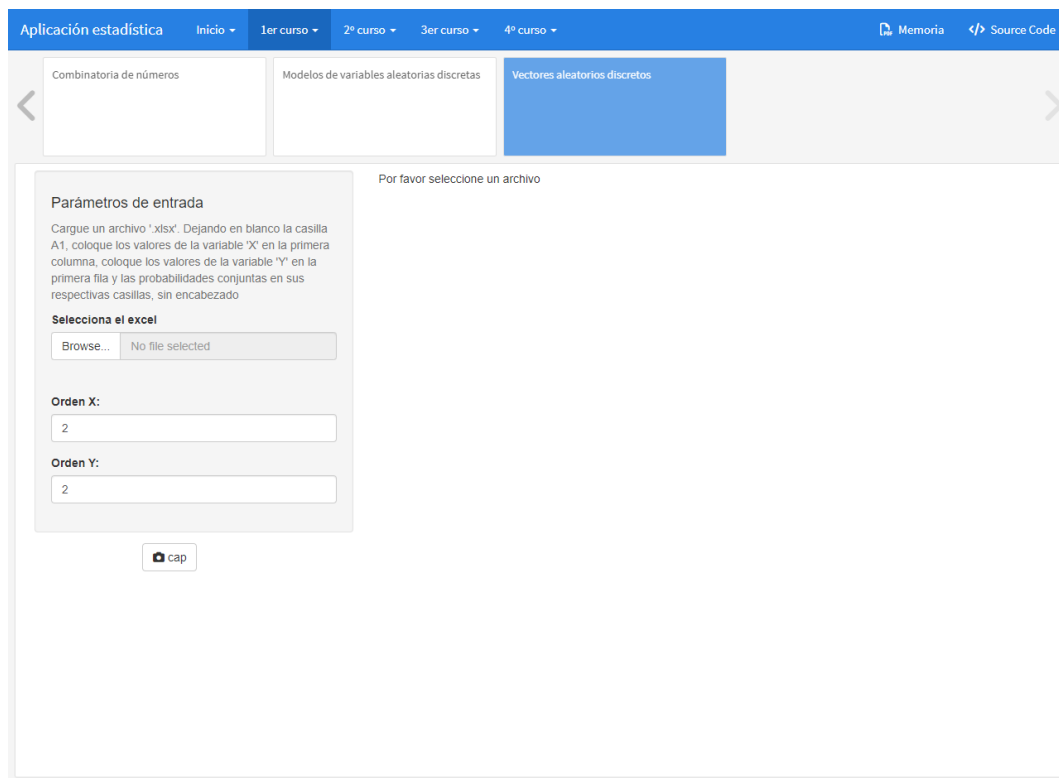


Figura 3.24: Vista previa del diseño de vectores aleatorios.

Para que la aplicación devuelva resultados, el usuario debe importar un archivo con extensión ".xlsx" en la forma explicada en el texto justo debajo de "Parámetro de entrada" que se encuentra en el "sidebarPanel". Una vez importado el archivo la aplicación devolverá resultados con el diseño que se muestra en la Figura 3.25.

Figura 3.25: Vista previa del diseño de vectores aleatorios importando archivo.

Código a destacar en este subbloque es la realización de una fotografía sobre la parte que se está visualizando, si es que el usuario desea almacenar la información que proporciona la aplicación. Una pequeña aplicación como muestra de esta acción es la siguiente Figura 3.26 sobre código R.

```

1 library(shinyscreenshot)
2 library(lorem)
3 ui <- fluidPage(
4   titlePanel("Combinación de números"),
5   sidebarLayout(
6     sidebarPanel(
7       tags$head(tags$style(HTML("#foto{position:fixed; top:200px;}")),
8         h4("Parámetros de entrada"),
9         div(id="foto",screenshotButton(label="cap"))
10    ),
11    mainPanel(
12      uiOutput("salida0")
13    )
14  )
15 )
16
17 server <- function(input,output){
18
19   output$salida0 <- renderUI({
20     ipsum(17)
21   })
22 }
23
24 shinyApp(ui,server)

```

Figura 3.26: Código R para obtener captura de pantalla.

Para conseguir que el botón de captura de pantalla se desplace por la página hay que agregarle la propiedad de "position:fixed" de forma que ocupe siempre el mismo lugar de la página y la propiedad "top:200px" es para añadirle un margen superior de 200 píxeles por cuestiones de presentación.

3.3.2. Segundo curso

Cuando se accede a las asignaturas del segundo curso una de ellas es *Muestreo Estadístico* donde la idea que justifica esta asignatura es la de extraer una parte o selección de elementos más reducida de una población que generalmente es muy grande o con un gran número de elementos, lo que dificulta su estudio exhaustivo y llevar a cabo el análisis de las características de la población completa mediante la información suministrada por dicha selección más pequeña.

El contenido programado o las funcionalidades consta principalmente de la obtención de muestras y de estimaciones de parámetros cuya fundamentación se encuentra en la sección 2.1.2. Una imagen que muestra el diseño de este bloque es la Figura 3.27.

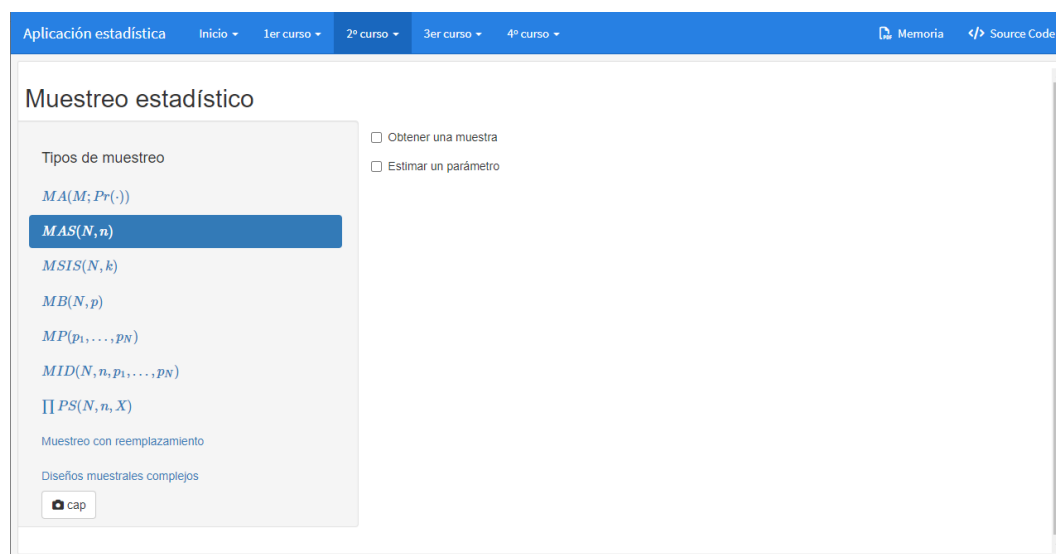


Figura 3.27: Vista previa del diseño de muestreo estadístico. Parte 1.

En la Figura 3.27 aparecen en el lado izquierdo los distintos tipos de diseños muestrales en los que el usuario podría acceder mediante clics del ratón y en lado derecho que permanece invariable en todos los diseños hasta que se clic en sus dos opciones disponibles “Obtener un muestra” y “Estimar un parámetro” que se despliegan nuevas funcionalidades con contenido común en todos los diseños pero cuyo código que no es visible es diferente, por ejemplo, si el usuario clic en la casilla de verificación de “Obtener una muestra” el diseño que se vería sería el mostrado en la Figura 3.28.

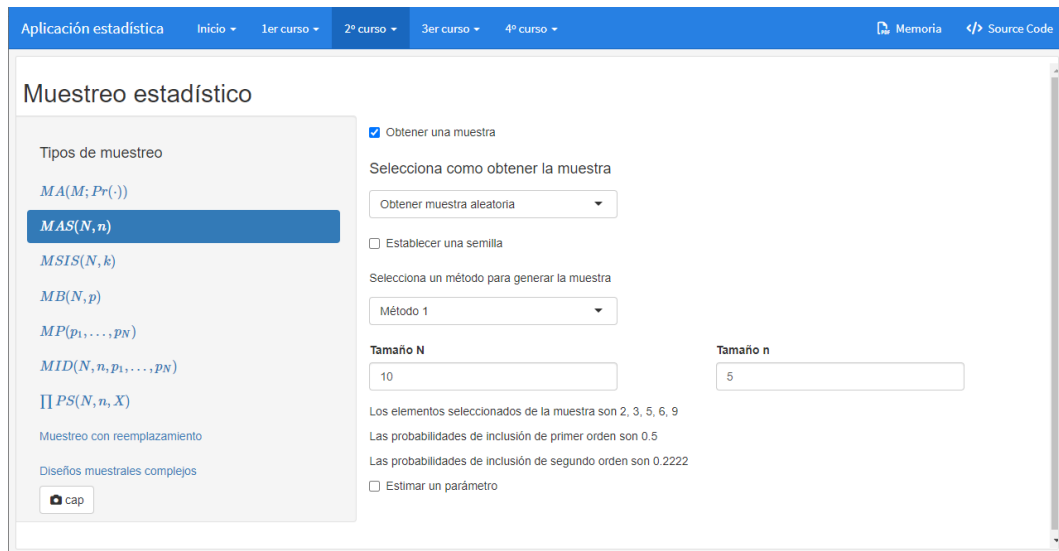


Figura 3.28: Vista previa del diseño de la asignatura de muestreo estadístico. Parte 2.

Opciones comunes si se clicará en cualquier otro diseño son:

- El desplegable “Selecciona como obtener la muestra” que solo tiene dos opciones disponibles, “Obtener una muestra aleatoria” y “Obtener muestra de un archivo”. El primero ya tiene por defecto programado según el diseño en el que se esté accediendo, la obtención de los elementos que van a componer la muestra mediante el método que se elija y los parámetros que se le fijen, que en este caso como se encuentra en un diseño $MAS(N, n)$ siendo $n = 5$ y $N = 10$, y el “Método 1” como obtención de muestra la aplicación ya puede realizar la obtención de muestra y dar los elementos que la componen, así como sus probabilidades de inclusión de primer y segundo orden. Nótese que si estuviera en otro diseño muestral, los parámetros introducidos como N , n y “Método 1” no serían válidos, porque estos no son comunes a otros diseños, son específicos del diseño $MAS(N, n)$. La segunda opción del desplegable “Selecciona como obtener la muestra” concede permiso al usuario de subir un archivo con extensión “.xlsx” en la forma descrita en el texto justo encima del “fileInput” y mediante el método de obtención de muestra y el tamaño de la misma visualiza la muestra en una tabla y describe los elementos que componen la muestra y las probabilidades de inclusión de primer y segundo orden. Así como descargar el conjunto de muestra en un archivo “.xlsx”
- La casilla de verificación, “Establecer una semilla”, que al clicar en ella se despliega un “textInput” y un botón “Aceptar”. Esta opción permite generar una muestra fijando una semilla. Para conseguir establecer una semilla solo se permite introducir números sin ningún tipo de carácter adicional como espacios o comas, en ese caso después de clicar en el botón “Aceptar” mostraría una notificación en amarillo con el siguiente contenido “Le notifico que no ha establecido una semilla”. Si los números se han introducido correctamente, por ejemplo 113, se mostraría una notificación de color azul con el siguiente contenido “Le notifico que se ha establecido una semilla, con valor, 113”.

Si el usuario clicara en la casilla de verificación de “Estimar un parámetro” el diseño que se vería sería el mostrado en la Figura 3.29

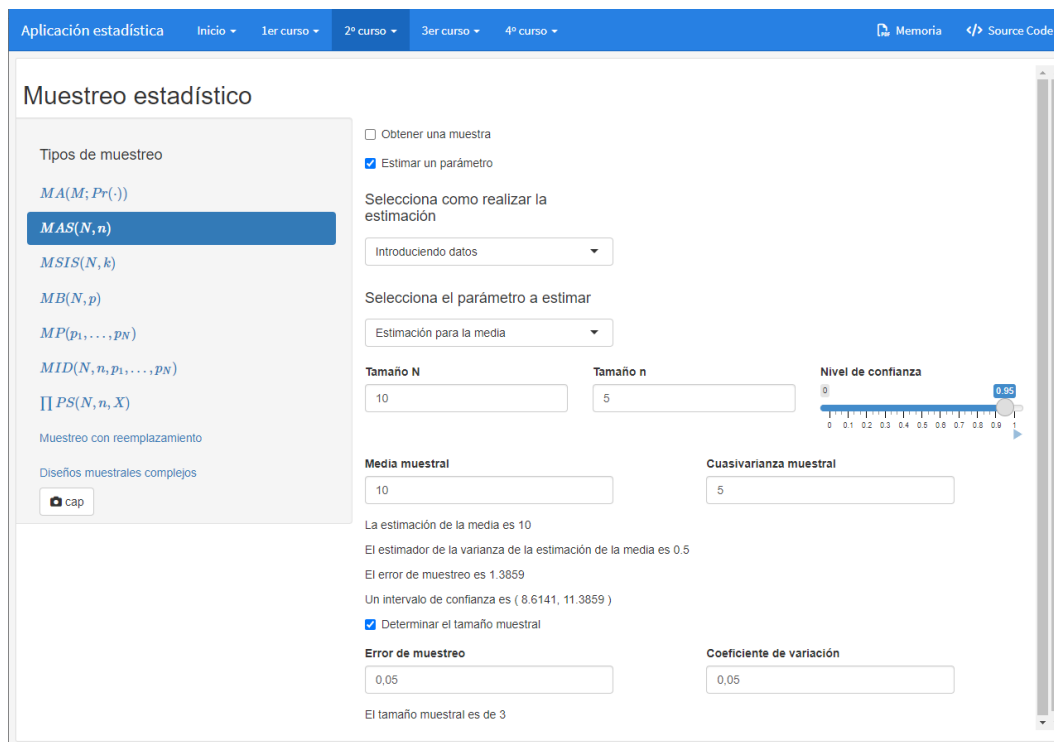


Figura 3.29: Vista previa del diseño de muestreo estadístico. Parte 3.

Opciones comunes si se clicará en cualquier otro diseño son:

- El desplegable “Selecciona como realizar la estimación” que también contiene dos opciones, una primera, “Introduciendo datos” solo se encuentra disponible para el diseño $MAS(N, N)$ en los demás diseños muestrales aparece “Próximamente...” con un fondo gris, indicando que estará disponible en un futuro próximo. Y la segunda opción “Adjunto archivo” que contiene la configuración disponible para subir un archivo con extensión “.xlsx”, pero en la forma descrita en el texto que hay justo encima del fileInput. Al ser una estimación los archivo “.xlsx” que se adjunten deberán contener una solo columna con los valores muestral y con encabezado o dos columnas siendo esta segunda, las probabilidades de inclusión de primer orden en caso de que se pidan.
- El desplegable “selecciona el parámetro a estimar” pero no siendo común las opciones disponibles, en algunos diseños vendrán más que otros dependiendo de su aplicación.

Como códigos a destacar:

- El uso de los paneles condicionales para renderizar diferentes secciones según se vayan confirmando las distintas decisiones a elegir⁴
- Botones de descarga de archivo, como la descarga de los datos muestrales para su posterior tratamiento en la estimación. En la Figura 3.30 puede observarse la construcción de una pequeña aplicación.

⁴Para ver un ejemplo, acudir a: <https://shiny.posit.co/r/gallery/dynamic-user-interface/conditionalpanel-demo/> [3]

```

1 library(shiny)
2 library(openxlsx)
3 ui <- fluidPage(
4   titlePanel("Muestreo estadístico"),
5   navlistPanel(
6     tabPanel(
7       "Diseño muestral 2",
8       fluidRow(
9         column(12, downloadButton("descar_dat", "Descargar"))
10      ),
11    )
12  )
13 )
14 )
15 )
16 server <- function(input,output){
17   muestra_arch <-data.frame(valor=trunc(runif(10,min = 1,max = 10)),
18     Probabilidad=runif(10))
19
20   output$descar_dat <- downloadHandler(
21     filename = function() {
22       paste("Muestra", ".xlsx", sep = "")
23     },
24     content = function(file) {
25       write.xlsx(muestra_arch, file, row.names = FALSE)
26     }
27   )
28 }
29
30 shinyApp(ui,server)

```

Figura 3.30: Código R de botón para descargar de datos.

- Botones para mostrar notificaciones de que se han establecido ciertos valores. En la Figura 3.31 puede observarse el código R para su ejecución.

```

1 library(shiny)
2 ui <- fluidPage(
3   titlePanel("Muestreo estadístico"),
4   navlistPanel(
5     tabPanel(
6       "Diseño muestral 1",
7       fluidRow(
8         column(6,textInput("valor","Escribe un numero")),
9         column(6, actionButton("boton", "Pulsa"))
10      )
11    )
12  )
13 )
14 )
15 server <- function(input,output){
16
17   observeEvent(input$boton, {
18     sem=as.numeric(input$valor)
19     if(is.na(sem)){
20       showNotification(p("Le notifico que no has escrito un numero"), type="warning")
21     }else{
22       showNotification(p("Le notifico que has escrito el número|", input$valor), type="message")
23     }
24   })
25
26 }
27

```

Figura 3.31: Código R de aviso de notificación.

- Uso de los colores, pues debido a la extensión de la programación y al tiempo limitado se deja como acciones futuras ciertos contenidos que no han sido posible programarse. En la Figura 3.32 puede observarse el código R para su aplicación.

```
1 library(shiny)
2 ui <- fluidPage(
3   titlePanel("Muestreo estadístico"),
4   navlistPanel(
5     tabPanel(
6       "Diseño muestral 2",
7       fluidRow(
8         column(12, style = "background-color:#b5bac9;", h3("Próximamente..."))
9       ),
10    ),
11    tabPanel(
12      "Diseño muestral 1",
13      fluidRow(
14        column(6, style = "background-color:#AED6F1;", h3("celeste")),
15        column(6, style="background-color:#82E0AA", h3("verde claro"))
16      ),
17      fluidRow(
18        column(12, style = "background-color:#F5B7B1;", h3("rojo claro")),|
19      ),
20    )
21  )
22 )
23 )
24 )
25 server <- function(input,output){
26
27 }
28
29 shinyApp(ui,server)
```

Figura 3.32: Código R para obtener color de fondo.

3.3.3. Tercer curso

Entre las asignaturas de tercer curso se encuentra *Ampliación de Inferencia Estadística* que entre sus contenidos se encuentran los estudios de contrastes de hipótesis que ayudan a determinar si las afirmaciones realizadas sobre una población son ciertas o no. Usualmente se recogen técnicas no paramétricas (es decir, no se conoce la distribución de la variable objeto de estudio) aunque también se recogen técnicas paramétricas, pues en ocasiones se presentan resultados donde la distribución resulta ser conocida y no acaba ahí el problema sino que se hacen hipótesis sobre sus parámetros (como que el valor de la media valga un cierto valor) y por tanto, para terminar de completar el problema se requieren dichas técnicas.

Este contenido de los contrastes de hipótesis paramétricos y no paramétricos, cuya base teórica se encuentra en la sección 2.1.3, es el que se ha programado. A continuación se muestra una imagen como vista previa de este apartado en la Figura 3.33.



Figura 3.33: Vista previa del diseño de ampliación de inferencia estadística. Parte 1.

El diseño de esta sección tiene una configuración de “tabPanel” creando un panel de pestañas de tal forma que permite al usuario ver a simple vista, ante que problema de contraste de hipótesis se encuentra, si se trata de un contraste para una muestra, para dos muestras independientes o dependientes o más de dos muestras. Posteriormente, una vez identificado esta primera cuestión, el patrón del diseño se repite prácticamente en todas las pestañas disponibles, con un texto informativo sobre la disposición del excel que ha de adjuntarse, un “fileInput” que permite importar el archivo que se quiere analizar, un “selecInput” para seleccionar el contraste a realizar sobre el conjunto de datos importado, el contraste descrito para ayudar al usuario que contraste va a realizar, un texto informativo como regla de validación para rechazar o no una hipótesis nula, la realización de una captura de pantalla para almacenar los datos que ha obtenido del contraste, así como para demostrar que ha realizado el contraste y por último la parte que no es visible, hasta que se importan los datos en la aplicación, que son la visualización del conjunto de datos importados, para comprobar que los datos se han leído correctamente, así como sus métricas, para el rechazo o no de la hipótesis nula.

En la Figura 3.34 puede verse el diseño de la parte no visible hasta que el usuario adjunto el archivo excel.

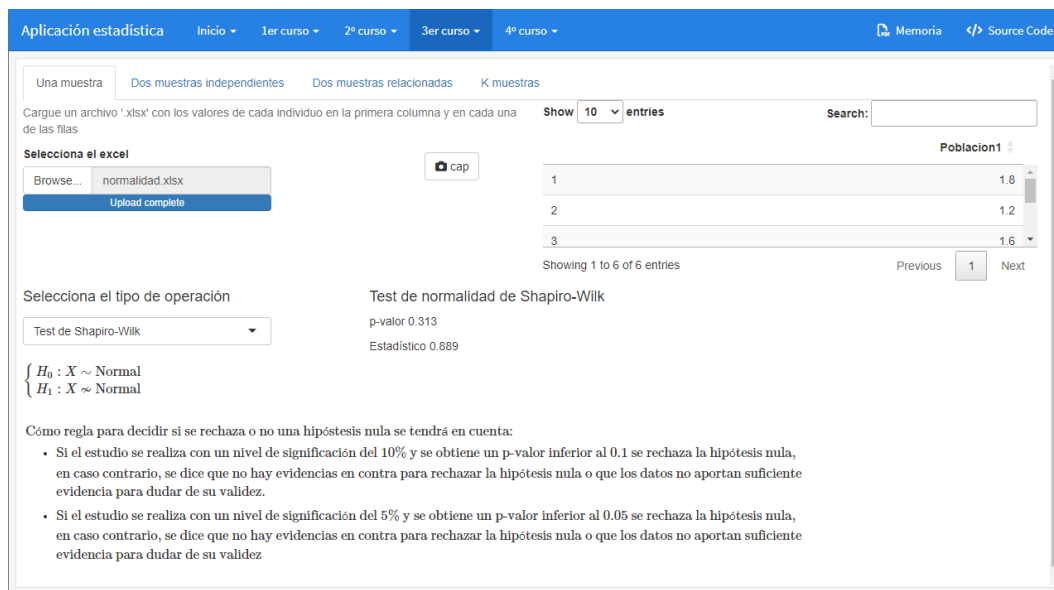


Figura 3.34: Vista previa del diseño de ampliación de inferencia estadística. Parte 2

3.3.4. Cuarto curso

En las asignaturas de cuarto curso está como optativa *Inteligencia Artificial y Estadística*, la cual permite adquirir los conocimientos para llevar a cabo este proyecto reunificando las diversas materias del grado en estadística en una aplicación estadística inteligente.

Como abarcar todos los contenidos supone extender demasiado esta memoria se va a exponer, como resumen y en representación de lo que se estudia en la asignatura, parte del proyecto final. Cuya finalidad consistía en reunir los conocimientos aprendidos en la misma para crear una aplicación. Algo similar es lo que se ha hecho en este trabajo pero siendo el proyecto final de *Inteligencia Artificial y Estadística* menos extenso, tal que así es posible recogerse en este trabajo, aunque omitiéndose muchas técnicas de visualización, depuración y modelación de datos, los cuales se han trabajado ampliamente en la aplicación estadística o puede observarse en en el código del apéndice.

Lo que se presenta del proyecto final de *Inteligencia Artificial y Estadística* es una pequeña aplicación que clasifica y calcula según los datos introducidos el tipo de índice de masa corporal (IMC). Se especifica si se posee un índice de masa corporal con bajo peso, normal, obeso o sobrepeso.

Para ello, se siguieron los pasos típicos de la ciencia del dato mencionados en la sección 2.1.4 donde se definió el problema, se hizo un proceso de búsqueda de datos desde distintas fuentes⁵

Una vez que los datos fueron obtenidos, se empleo el ecosistema Tidyverse desde R, que posee una colección de paquetes orientados a la importación, manipulación, exploración y visualización de datos y que se utiliza exhaustivamente en ciencia de datos.

Al ser el conjunto de datos un archivo “.csv” separados por comas, se llamó a la función ‘read_csv’ del paquete readr, ya cargado debido a que pertenece a tidyverse. Se continuó

⁵Algunas fuentes de datos consultadas fueron datos.gob, data.europa, archive.ics.uci.edu, datahub.io/search, etc.

con la recodificación de las variables ya que el mismo R no interpretaba bien la codificación de las variables debido a que los datos no estaban depurados.

Posteriormente se pasó al proceso de manipulación y ordenación con el paquete “dplyr”, conduciendo a procedimientos de limpieza de los valores, que implican realizar eliminación o generación de nuevos datos. Este proceso es relevante ya que sin datos eficientes y veraces todos los procesos posteriores serán erróneos o poco eficaces. Tal que así, se agruparon observaciones comunes con poca frecuencia de datos con varias variables coincidentes para que estuvieran bien representadas, al igual que se eliminaron aquellas que no podían agruparse a otras observaciones debido a la gran cantidad de datos desconocidos que poseía.

A continuación se pasó al proceso de Visualización, para tratar de entender los datos que tenemos, dentro del Análisis Exploratorio de Datos y para tratar de transmitir conclusiones que ya hemos observado a partir de los datos, de modo que podamos facilitar esa transmisión o comunicación de conclusiones a una audiencia.

Se continuó llevando a cabo el proceso de modelización. Para ello nos apoyamos en el ecosistema Tidymodels. Específicamente se hicieron dos modelos, uno de regresión y otro de clasificación. En ambos casos se hizo una partición del conjunto de datos en entrenamiento y prueba. Empleando, para ello, la función ‘initial_split’ junto con el el argumento ‘strata’, que mantiene la misma proporción del conjunto total, de cada una de las clases, en cada subconjunto (entrenamiento y prueba), en el caso de clasificación, y en el caso de regresión dividía la variable objetivo en cuatro intervalos similares para ambos subconjuntos.

Por otro lado, se especificó el tipo de modelo a utilizar. En los dos casos se ha hecho un modelo sencillo pero con suficientes variables predictoras. Se estableció un árbol de decisión, con la función ‘decision_tree’ y se fijó como motor “rpart”. Para el modelo de regresión además se preparó una receta con secuencias canalizables de pasos de ingeniería de características para preparar los datos para el modelado, lo cual no era necesario para el modelo de clasificación.

A continuación se creó un workflow. En este workflow incorporamos la receta (que ya lleva incluida la fórmula que nos dice que predecir y a partir de qué variables) y la especificación del modelo anterior.

Como los árboles de decisión son un tipo de modelo muy susceptibles a cambios de la muestra elegida, en lugar de establecer unos parámetros concretos en el modelo especificado antes, a estos parámetros se les marcan con la función ‘tune’ que deja los valores abiertos a un proceso de ajuste de hiperparámetros. Para ello, necesitamos apoyarnos en el proceso de validación cruzada generando múltiples conjuntos de análisis y validación a partir del conjunto de entrenamiento. Específicamente se hizo validación cruzada con 10 pliegues, por lo que los datos se dividían en 10 subconjuntos. Por cada una de estas 10 iteraciones uno de los conjuntos se toma como validación y el otro como análisis. Finalmente se realiza la media aritmética de los resultados de las métricas de cada iteración para obtener un único resultado. Este proceso de validación cruzada, se repite, al emplearse dentro de un proceso de ajuste de hiperparámetros, para una serie de combinaciones de valores de los parámetros que habíamos marcado con tune. Este proceso de ajuste lo lleva a cabo ‘tune_grid’, que además de las particiones de validación cruzada y el workflow explicado antes, esta función necesita recibir que combinaciones de valores de los parámetros (o bien que cantidad de las mismas) deben probarse en ese proceso iterativo.

En nuestro caso particular, hemos configurado nuestro proceso de dos formas distintas, mediante el paquete “dials”, específicamente un ‘grid_regular’ (que nos permite controlar esa configuración) y un grid aleatorio, cuyas mejores métricas se pueden observar en las tablas A.1, A.2, A.3, A.4, A.5 y A.6 del Apéndice A. De estas métricas se sacaron las mejores métricas y se compararon. Del grid_regular se sacaron dos métricas tanto del “rsq” como del “rmse” dado que la mejor seleccionada por “select_best” tenía una “min_n=2”, en ambos casos, lo que podía dar lugar a un sobreajuste, y por tanto, se optó por tomar otra con el mismo coste de complejidad pero con una “min_n=11” y también en ambos casos. En clasificación no hubo inconvenientes la función “select_best” seleccionó la mejor métrica.

Con la mejor configuración de parámetros encontrada para cada modelo, lo siguiente que hacemos es la finalización del workflow, que aplica esos mejores parámetros encontrados a los parámetros que inicialmente habíamos marcado con ‘tune’.

Teniendo ya los modelos ajustados por el proceso anterior, en los nuevos workflows, que empleamos llevar a cabo el proceso completo de entrenamiento, predicción y evaluación sobre la partición inicial “clásica” de entrenamiento y prueba (la que obtuvimos con ‘inital_split’). Esto último lo lleva a cabo ‘last_fit’.

Del objeto generado por ‘last_fit’ se puede extraer información útil sobre las métricas obtenidas, bien manualmente o con las funciones ‘collect_metrics’ y ‘collect_predictions’. En el modelo de regresión se obtuvo como mejor combinación de parámetros los generados por el grid aleatorio no siendo así en el caso de clasificación, por una centésima. En la predicción para nuevos datos se dejó la combinación de parámetros del grid aleatorio

Finalmente, deben comunicarse y presentarse los resultados, tratando de ofrecer mucha información, con pocas aportaciones. Esta es la parte que se muestra en la aplicación. La predicción del IMC o nivel de obesidad con las respuesta que aporta el usuario y según el modelo que elija. En la Figura 3.35 puede verse el diseño y las funcionalidades. A través de “Source Code” puede el usuario puede ver el diseño de página de esta sección y todos los pasos antes descrito.

Figura 3.35: Vista previa del diseño de inteligencia artificial y estadística.

Capítulo 4

Conclusiones

Este capítulo va a presentar una recopilación de las principales contribuciones realizadas a lo largo del trabajo. A continuación se resumirán las lecciones aprendidas. Finalmente se comentarán algunas posibles vías para continuar ampliando o extendiendo el trabajo realizado.

4.1. Aportaciones

Las principal aportación que he realizado ha sido el diseño y desarrollo de una aplicación software didáctica para el aprendizaje en el ámbito de la estadística, con la herramienta de software R y como entorno de desarrollo integrado (IDE) RStudio. Como segunda aportación he realizado un repaso completo a los contenidos cursados a lo largo de todo el grado, aunque en esta memoria se ha mostrado un pequeño subconjunto de todo lo analizado. Las aportaciones estadísticas en la aplicación han sido la programación de:

1. Estructuras combinatorias. Cálculos de permutaciones, de subconjuntos y todas las combinaciones posibles de un conjunto finito de elementos, describiendo tanto la fórmula como un ejemplo.
2. Modelo aleatorio y clásicos de variables aleatorias discretas (uniforme, bernouilli, binomial, poisson, geométrica e hipergeométrica). En este caso, se obtienen sus momentos esperados, varianzas, probabilidades, funciones de distribución y funciones de probabilidad.
3. Vectores aleatorios discretos. A partir de una tabla de probabilidad conjunta se obtienen las distribuciones marginales, los momentos y varianzas, así como el grado de relación entre ambos con la covarianza y el coeficiente de correlación y si son independientes o no.
4. Obtención de muestras según diseños muestrales aleatorios o clásicos (muestreo aleatorio simple, muestreo sistemático, muestreo de bernouilli, muestreo de poisson, muestreo de midzuno y el diseño *PPS*) a partir de un archivo excel importado o indicando el tamaño de la población o las probabilidades de selección de cada individuo, según el caso. Además, se lleva a cabo la obtención de estimadores del total poblacional, de la media poblacional, del parámetro razón, de razón, de diferencia y de regresión, así como intervalos de confianza según los diseños muestrales dados.
5. Solución a contrastes de hipótesis paramétrica y no paramétricas a partir de un archivo excel importado, así como la descripción de las hipótesis en cada caso y una

regla para decidir si se rechaza o no la hipótesis nula.

6. Aplicación de muestra de las herramientas de interacción y modelización que se pueden llegar a realizar con la asignatura de inteligencia artificial y estadística.
7. Provisión al usuario final de aspectos adicionales como la visualización de gráficos interactivos, la generación dinámica de las fórmulas y expresiones en LaTeX o la posibilidad de realizar capturas de pantalla de los resultados obtenidos con las distintas herramientas, entre otros.

4.2. Reflexiones

El diseño y desarrollo de una aplicación software didáctica para el aprendizaje en el ámbito de la estadística me ha supuesto varios retos personales. El primero de ellos es el de medir si de verdad he adquirido los conceptos que se enseñan en las asignaturas, al tener que programar funciones para obtener resultados deseados sin tener que mirar los apuntes. En los casos en los que no disponía de ese dominio al afrontar alguna funcionalidad, el trabajo me ha brindado una nueva oportunidad para volver a aprenderlo, madurarlo y ponerlo en práctica. Otro desafío afrontado ha sido el de ser capaz de diseñar una aplicación para un usuario externo, que principalmente es un estudiante, dotándole de herramientas útiles, usables e informativas para facilitar el estudio de la estadística, implicando el plantearme muchas cuestiones didácticas y prácticas que me eran ajenas y eran esenciales para esa utilidad perseguida..

Es un trabajo que ha requerido mucho esfuerzo y dedicación, necesitando compaginar el conocimiento de la teoría y práctica de las asignaturas (con una comprensión mucho mayor de la misma que cuando cursé las mismas) con la capacidad técnica y funcional para diseñar y presentar la aplicación, con saberse la teoría y la práctica de las asignaturas pues también entra en juego el diseño y presentación de la aplicación de tal forma que se consiga que sea intuitiva, moderna, accesible para todos los usuarios y al alcance de los clics del ratón, así como didáctica que sirva de ayuda al usuario. No obstante, creo que el esfuerzo ha merecido mucho la pena, independientemente de la valoración objetiva que se pueda hacer del trabajo por parte del lector, por el aprendizaje teórico-práctico que me han proporcionado todos estos meses de trabajo, y la satisfacción de irme demostrando a mí mismo el crecimiento experimentado.

Por otro lado, por supuesto soy consciente de que habrá aspectos que mejorar y mucho que poder ampliar, pero sinceramente creo que esta aplicación puede ser muy útil para los usuarios objetivo de la misma.

4.3. Trabajo futuro

Como espero haber podido transmitir a lo largo de la memoria, se planteó un trabajo consistente en diseñar y desarrollar una aplicación útil para el estudio de la estadística, recopilando una batería de funcionalidades extraídas de diversas asignaturas del grado. En este sentido, se fue seleccionando una muestra de funcionalidades de asignaturas de los distintos cursos.

Ahora bien, por más herramientas que fuimos incorporando, la cantidad de contenidos del grado a analizar, madurar e incorporar, con aspectos de interacción, cálculo, recopilación

de ejemplos, visualización, presentación, etc., haría que cualquier muestra que tomáramos nos implicara dejar mucha funcionalidad fuera del trabajo.

Así, es inmediato pensar que la primera posible extensión de este trabajo sería completar de manera exhaustiva los contenidos de todas las asignaturas de todo el grado. No es para nada una tarea baladí, pero sin duda sí muy enriquecedora, y me gustaría continuar con este proyecto en el futuro, o facilitar los medios para que otros futuros compañeros puedan ampliar esta aplicación.

Por otro lado, sería interesante pulir algunos detalles y subir el proyecto a shinyapps, una plataforma gratuita para poner a disposición la aplicación, aunque con una cuota reducida de tráfico y recursos de computación, como por ejemplo dotar a la aplicación de instrucciones o una guía que explique cómo la aplicación obtiene los resultados, dotarla aún más de contenido didáctico programando otras asignaturas o añadiendo a las que ya están aspectos que puedan ser de ayuda al usuario. Así como de actualizar librerías que queden obsoletas.

Para continuar este proyecto es necesario conocer el lenguaje de programación R y de las librerías de las que dispone, con especial atención a los aspectos interactivos de R Shiny y los ecosistemas tidyverse y tidymodels, entre otros.

Apéndice A

Apéndice: Tablas

Este apéndice recopila algunas tablas que se mencionan en el documento, de las mejores combinaciones de parámetros en procesos de ajuste de regresión y clasificación

A.1. Ajuste hiperparámetros regresión grid aleatorio

Tabla A.1: Métricas rsq con grid aleatorio. Modelo regresión

cost_complexity	tree_depth	min_n	.metric	.estimator	mean	n	std_err	.config
0.0000000	13	3	rsq	standard	0.9888833	10	0.0025567	Preprocessor1_Model06
0.0001205	15	12	rsq	standard	0.9853473	10	0.0027743	Preprocessor1_Model01
0.0003035	6	16	rsq	standard	0.9810145	10	0.0027349	Preprocessor1_Model08
0.0000141	9	26	rsq	standard	0.9806476	10	0.0021922	Preprocessor1_Model03
0.0000005	10	30	rsq	standard	0.9788397	10	0.0026442	Preprocessor1_Model07

Tabla A.2: Métricas rmse con grid aleatorio. Modelo regresión

cost_complexity	tree_depth	min_n	.metric	.estimator	mean	n	std_err	.config
0.0000000	13	3	rmse	standard	0.9023734	10	0.0999615	Preprocessor1_Model06
0.0001205	15	12	rmse	standard	1.0459539	10	0.0981750	Preprocessor1_Model01
0.0003035	6	16	rmse	standard	1.2018259	10	0.0876201	Preprocessor1_Model08
0.0000141	9	26	rmse	standard	1.2192760	10	0.0688396	Preprocessor1_Model03
0.0000005	10	30	rmse	standard	1.2706342	10	0.0745986	Preprocessor1_Model07

Tabla A.3: Métricas rsq con grid regular. Modelo regresión

cost_complexity	tree_depth	min_n	.metric	.estimator	mean	n	std_err	.config
0.0e+00	8	2	rsq	standard	0.9878522	10	0.0026003	Preprocessor1_Model12
0.0e+00	8	2	rsq	standard	0.9878518	10	0.0026003	Preprocessor1_Model11
3.2e-06	8	2	rsq	standard	0.9878322	10	0.0026217	Preprocessor1_Model13
3.2e-06	8	11	rsq	standard	0.9875325	10	0.0025914	Preprocessor1_Model28
0.0e+00	8	11	rsq	standard	0.9875324	10	0.0025909	Preprocessor1_Model26

Tabla A.4: Métricas rmse con grid regular. Modelo regresión

cost_complexity	tree_depth	min_n	.metric	.estimator	mean	n	std_err	.config
3.2e-06	8	2	rmse	standard	0.9450787	10	0.0998126	Preprocessor1_Model13
0.0e+00	8	2	rmse	standard	0.9452990	10	0.0989206	Preprocessor1_Model12
0.0e+00	8	2	rmse	standard	0.9453126	10	0.0989210	Preprocessor1_Model11
3.2e-06	8	11	rmse	standard	0.9608879	10	0.0947964	Preprocessor1_Model28
0.0e+00	8	11	rmse	standard	0.9609312	10	0.0947629	Preprocessor1_Model26

A.2. Ajuste hiperparámetros clasificación

Tabla A.5: Métricas con grid aleatorio. Modelo clasificación

cost_complexity	tree_depth	min_n	.metric	.estimator	mean	n	std_err	.config
0.0000102	14	19	roc_auc	hand_till	0.9743874	10	0.0021337	Preprocessor1_Model08
0.0000000	11	26	roc_auc	hand_till	0.9714166	10	0.0030547	Preprocessor1_Model01
0.0009264	10	33	roc_auc	hand_till	0.9706660	10	0.0031737	Preprocessor1_Model04
0.0000027	6	23	roc_auc	hand_till	0.9696639	10	0.0027618	Preprocessor1_Model02
0.0000000	8	31	roc_auc	hand_till	0.9693811	10	0.0030717	Preprocessor1_Model07

Tabla A.6: Métricas con grid regular. Modelo de clasificación

cost_complexity	tree_depth	min_n	.metric	.estimator	mean	n	std_err	.config
0.0000000	8	11	roc_auc	hand_till	0.9741448	10	0.0040521	Preprocessor1_Model06
0.0000000	8	11	roc_auc	hand_till	0.9741448	10	0.0040521	Preprocessor1_Model12
0.0000032	8	11	roc_auc	hand_till	0.9741448	10	0.0040521	Preprocessor1_Model18
0.0005623	8	11	roc_auc	hand_till	0.9741448	10	0.0040521	Preprocessor1_Model24
0.0000000	8	2	roc_auc	hand_till	0.9629587	10	0.0047615	Preprocessor1_Model05

Apéndice B

Apéndice: Código R de la aplicación

En este apéndice se recoge el código de la aplicación desarrollada, así como el de sus aplicaciones shiny independientes, que se encuentran en carpetas separadas, por si se quiere consultar.

Cada subsección de este índice llevará por nombre, la carpeta a la que pertenecen ó el nombre del archivo. Cuando sea un archivo aparecerá su extensión “.Rmd” ó “.R”.

B.1. AplicacionEstadística.Rmd

A continuación se recoge el código .Rmd llamado esqueleto en esta memoria, al que se le asocian las diversas aplicaciones shiny independientes de la distintas asignaturas, así como la aplicación para generar la galería de imágenes.

Para ejecutar el siguiente código ha de descomentarse los inicio y finales de los chunks.

```
---
title: "Aplicación estadística"
output:
  flexdashboard::flex_dashboard:
    source_code: embed
    navbar:
      - { title: "Memoria",
href:
  "https://drive.google.com/file/d/1EgjStM4Aq-k-abWjRoy5BE-wPKI__00j/view?usp=sharing",
    icon: "fa-file-pdf-o", target: "_blank"}
runtime: shiny
---
```

```
#```{r eval=FALSE, echo=TRUE}
library(flexdashboard)
library(tidyverse)
library(knitr)
library(DT)
library(rsample)
library(recipes)
library(tidymodels)
#```
```

```
# Introducción {data-navmenu="Inicio" .storyboard}
```

Presentación

```
#```${r}
h1("¡Bienvenido a la aplicación web!")
#````
```

```
$\\[0.3cm]$\
```

Esperamos que encuentre una plataforma útil, práctica, conveniente y fácil de usar.

En esta aplicación web podrás acceder a una gran cantidad de recursos y herramientas que te ayudarán con tus objetivos personales y profesionales. Esta aplicación web es fácil de usar y está diseñada para ser intuitiva y accesible para todos los usuarios. Uno de los objetivos es que disfrute al máximo de la experiencia mientras navegas por las herramientas y recursos utilizando una interfaz sencilla y minimalista.

La idea principal de este proyecto es la de proporcionar una aplicación software que ayude a los estudiantes de estadística a asentar los conceptos aprendidos en el grado, analizando conceptos explicados en diversas asignaturas y ejercicios propuestos para las mismas y dotando de una batería de utilidades visuales que permitan al estudiantado comprender mejor los conceptos y resolver los ejercicios planteados proporcionando los datos al sistema y obteniendo los resultados correspondientes.

Así que sin más preámbulos, ¡gracias por acceder a esta aplicación web!. Esperamos que sea disfrute mientras navega.

Importación de datos

Cada vez que se requiera importar datos a la aplicación aparecerá un texto informativo que explicará como ha de importarse los datos y en que formato. Es muy importante leer y comprender este texto porque a partir de los datos importados se obtendrán los resultados

El texto informativo será como el siguiente:

```
#```${r}
helpText("Cargue un archivo '.xlsx' con los valores de la
          variable en la primera columna y con encabezado")
#````
```

Como el archivo que se importa es un Excel (con extensión .xlsx) se

mostrará una pequeña ilustración de que se entiende por columnas, y encabezado.

```
#```${r}
fluidRow(
  style = " overflow: auto; height: calc(70vh - 150px) !important;",
  shinyAppDir(
    file.path("IMPORT"), # enter valid file path here
    options=list(
      width="100%", height="100%",
      style="display:flex;"
    )
  )
)

#```${r}

# Calculo de probabilidad {data-naumenu="1er curso" .storyboard}

#```${r}
eval(parse("CP/index.R", encoding="UTF-8"))
#```${r}
### Combinatoria de números

#```${r}
fluidRow(responsive = FALSE,
  column(12,
    style = "overflow: auto; height: calc(70vh - 20px) !important;",
    selectInput("basic0", h3("Selecciona una operación"),
      choices = list("Permutación sin repetición"=1,
        "Permutación con repetición"=2,
        "Permutación circular"=3,
        "Permutación de longitud n"=4,
        "Combinación sin repetición"=5,
        "Todas las combinaciones posibles"=6),
      selected = "Permutación sin repetición"),
    renderUI({
      uiComb(as.numeric(input$basic0))
    })
  )
)
#```${r}

### Modelos de variables aleatorias discretas

#```${r}
```

```

fluidRow(responsive = FALSE,
  column(12,
    style = "overflow: auto; height: calc(70vh - 20px) !important;",
    selectInput("basic1", h3("Selecciona un modelo"),
      choices = list(
        "Modelo aleatorio"=1,
        "Modelo uniforme"=2,
        "Modelo bernoulli"=3,
        "Modelo binomial"=4,
        "Modelo poisson"=5,
        "Modelo geométrico"=6,
        "Modelo hipergeométrico"=7),
      selected = "Modelo Uniforme"),
    renderUI({
      uiMod(as.numeric(input$basic1))
    })
  )
)
#` ``

### Vectores aleatorios discretos

#` ``{r}
runVec(1)
#` ``

# Estadística descriptiva {data-naumenu="1er curso" data-orientation=rows}

#` ``{r}
h2("Próximamente...")
#` ``

# Teoría de la probabilidad I {data-naumenu="1er curso" data-orientation=rows}

#` ``{r}
h2("Próximamente...")
#` ``

# Muestreo estadístico {data-naumenu="2º curso" data-orientation=rows}

#` ``{r}
# eval(parse("ME/index.R", encoding="UTF-8"))
#` ``

#` ``{r}
# runMe(1)
fluidRow(
  style = " overflow: auto; height: calc(95vh - 45px) !important;",

```



```

shinyAppDir(
  file.path("ME"), # enter valid file path here
  options=list(
    width="100%", height="100%",
    style="display:flex;"
  )
)
)
)
#` ``
# Teoría de la probabilidad II {data-naumenu="2º curso" data-orientation=rows}
#` ``{r}
eval(parse("TP/index.R", encoding="UTF-8"))
#` ``
#` ``{r}
runTp2(1)
#` ``
# Ampliación de inferencia estadística {data-naumenu="3er curso" data-orientation=rows}
#` ``{r}
# runMe(1)
fluidRow(
  style = " overflow: auto; height: calc(95vh - 45px) !important;",
  shinyAppDir(
    file.path("AIE"), # enter valid file path here
    options=list(
      width="100%", height="100%",
      style="display:flex;"
    )
  )
)
)
#` ``
# Analisis multivariante {data-naumenu="3er curso" data-orientation=rows}
#` ``{r}
h2("Próximamente...")
#` ``
# Modelos lineales {data-naumenu="3er curso" data-orientation=rows}
#` ``{r}
h2("Próximamente...")
#` ``

```

```

# Diseño de experimentos {data-naumenu="4º curso" data-orientation=rows}

#```{r}
h2("Próximamente...")
#```

# Inteligencia artificial y estadística {data-naumenu="4º curso" data-orientation=rows}

#```{r results='hide'}
obesity <- read_csv("IAE/ObesityDataSet_raw_and_data_sinthetic.csv")

obesity$Gender<-parse_factor(obesity$Gender)
# levels(obesity$Gender) #para ver las etiquetas
obesity$Gender = fct_recode(obesity$Gender,
                           "Mujer" = "Female", "Hombre" = "Male")
# levels(obesity$Gender) #para ver las etiquetas

obesity$family_history_with_overweight <-
  parse_factor(obesity$family_history_with_overweight)
# levels(obesity$family_history_with_overweight)#para ver las etiquetas
obesity$family_history_with_overweight <-
  fct_recode(obesity$family_history_with_overweight, "si" = "yes")
# levels(obesity$family_history_with_overweight)#para ver las etiquetas

obesity$FAVC<-parse_factor(obesity$FAVC, levels = c("yes", "no"))
# levels(obesity$FAVC)#para ver las etiquetas
obesity$FAVC<-fct_recode(obesity$FAVC, "si" = "yes")
# levels(obesity$FAVC)#para ver las etiquetas

obesity$CAEC<-parse_factor(
  obesity$CAEC,
  levels=c("no", "Sometimes", "Frequently", "Always"))
# levels(obesity$CAEC)#para ver las etiquetas
obesity$CAEC<-fct_recode(
  obesity$CAEC, "no"="no", "algunas veces" = "Sometimes",
  "frecuentemente" = "Frequently", "siempre" = "Always")
# levels(obesity$CAEC)#para ver las etiquetas

obesity$SMOKE<-parse_factor(obesity$SMOKE, levels = c("yes", "no"))
# levels(obesity$SMOKE)#para ver las etiquetas
obesity$SMOKE<-fct_recode(obesity$SMOKE, "si" = "yes")
# levels(obesity$SMOKE)#para ver las etiquetas

obesity$SCC<-parse_factor(obesity$SCC, levels = c("yes", "no"))
# levels(obesity$SCC)#para ver las etiquetas
obesity$SCC<-fct_recode(obesity$SCC, "si" = "yes")
# levels(obesity$SCC)#para ver las etiquetas

```

```

obesity$CALC<-parse_factor(
  obesity$CALC,
  levels=c("no", "Sometimes", "Frequently", "Always"))
# levels(obesity$CALC)#para ver las etiquetas
obesity$CALC<-fct_recode(
  obesity$CALC, "no"="no", "algunas_veces" = "Sometimes",
  "frecuentemente" = "Frequently", "siempre" = "Always")
# levels(obesity$CALC)#para ver las etiquetas

obesity$MTRANS<-parse_factor(obesity$MTRANS)
# levels(obesity$MTRANS)#para ver las etiquetas
obesity$MTRANS <- fct_recode(
  obesity$MTRANS,
  "transporte_publico" = "Public_Transportation",
  "caminando" = "Walking",
  "automovil" = "Automobile",
  "moto" = "Motorbike",
  "bicicleta" = "Bike")
# levels(obesity$MTRANS)#para ver las etiquetas

obesity$NObeyesdad<-parse_factor(obesity$NObeyesdad,
  levels=c("Insufficient_Weight",
  "Normal_Weight",
  "Overweight_Level_I",
  "Overweight_Level_II",
  "Obesity_Type_I",
  "Obesity_Type_II",
  "Obesity_Type_III"))
# levels(obesity$NObeyesdad)#para ver las etiquetas
obesity$NObeyesdad<-fct_recode(obesity$NObeyesdad,
  "bajo peso" = "Insufficient_Weight",
  "normal" = "Normal_Weight",
  "sobrepeso I" = "Overweight_Level_I",
  "sobrepeso II" = "Overweight_Level_II",
  "obesidad I" = "Obesity_Type_I",
  "obesidad II" = "Obesity_Type_II",
  "obesidad III" = "Obesity_Type_III")
# levels(obesity$NObeyesdad)#para ver las etiquetas

obesity$Age <- trunc(obesity$Age)
obesity <- obesity %>% filter(Age<=27 & Age>=16)

obesity$FCVC <- trunc(obesity$FCVC)
obesity$FCVC <- parse_factor(as.character(obesity$FCVC),
  levels = c("1","2","3"))
obesity$FCVC<-fct_recode(obesity$FCVC, "nunca" = "1",
  "algunas_veces" = "2",

```

```

                                "siempre" = "3")
conver<-function(x){
  for(i in 1:length(x)){
    if(x[i]>=2 & x[i]<2.5){
      x[i]=1;
    }
    else if(x[i]>=2.5 & x[i]<3){
      x[i]=3;
    }
  }
  return(x)
}

obesity$NCP <- fct_lump(parse_factor(as.character(obesity$NCP)), n = 3)
# levels(obesity$NCP)#para ver las etiquetas
obesity$NCP<- fct_recode(obesity$NCP,
                          "entre_1_y_2"="1", "tres"="3", "mas_de_3"="4")
# levels(obesity$NCP)#para ver las etiquetas
obesity$NCP<- fct_relevel(obesity$NCP,
                          c("entre_1_y_2", "tres", "mas_de_3", "Other"))
# levels(obesity$NCP)#para ver las etiquetas

obesity$CH20 <- fct_lump(parse_factor(as.character(obesity$CH20)), n = 3)
obesity$CH20<- fct_recode(obesity$CH20,
                          "menos_de_11"="1",
                          "entre_11_y_21"="2",
                          "mas_de_21"="3")
# levels(obesity$CH20)#para ver las etiquetas
obesity$CH20<- fct_relevel(
  obesity$CH20,
  c("menos_de_11", "entre_11_y_21", "mas_de_21", "Other"))
# levels(obesity$CH20)#para ver las etiquetas

obesity$FAF <- fct_lump(parse_factor(as.character(obesity$FAF)), n = 4)
obesity$FAF<- fct_recode(obesity$FAF,
                          "no_tengo"="0",
                          "1_o_2_dias"="1",
                          "2_o_4_dias"="2",
                          "4_o_5_dias"="3")
# levels(obesity$FAF)#para ver las etiquetas
obesity$FAF<- fct_relevel(
  obesity$FAF,
  c("no_tengo", "1_o_2_dias", "2_o_4_dias", "4_o_5_dias"))
# levels(obesity$FAF)#para ver las etiquetas

obesity$TUE <- fct_lump(parse_factor(as.character(obesity$TUE)), n = 3)
obesity$TUE <- fct_recode(obesity$TUE,

```

```

        "0_2_horas"="0",
        "3_5_horas"="1",
        "mas_de_5_horas"="2")
# levels(obesity$TUE)#para ver las etiquetas
obesity$TUE <- fct_relevel(obesity$TUE , c("0_2_horas","3_5_horas"))
# levels(obesity$TUE)#para ver las etiquetas

obesidad <- obesity %>%
  rename(genero = Gender, edad = Age, altura = Height, peso = Weight,
         familiar_sobrepeso = family_history_with_overweight,
         consume_hipercaloricos= FAVC, consume_vegetales=FCVC,
         comidas_principales = NCP, consume_entre_comidas = CAEC,
         fuma = SMOKE, consume_agua_dia = CH20, controla_calorias = SCC,
         frecuencia_actividad_fisica = FAF, tiempo_en_consolas = TUE,
         consume_alcohol= CALC, transporte = MTRANS,
         nivel_obesidad=NObesidad)

nivel=function(x){
  if(x<18.5){
    y="bajo peso";
    return(y)
  } else if(x>=18.5 & x<25){
    y="normal";
    return(y)
  } else if(x>=25 & x<27){
    y="sobrepeso I";
    return(y)
  } else if(x>=27 & x<30){
    y="sobrepeso II";
    return(y)
  } else if(x>=30 & x<35){
    y="obesidad I";
    return(y)
  } else if(x>=35 & x<40){
    y="obesidad II";
    return(y)
  } else if(x>=40){
    y="obesidad III";
    return(y)
  }
}

IMC <- obesidad %>%
  transmute(imc = peso/(altura*altura),
            nivel_obesidad)
IMC<- IMC %>%

```

```

mutate(real=map_chr(IMC$imc,nivel))

IMC$real<-parse_factor(IMC$real,
                      levels = c("bajo peso","normal","sobrepeso I",
                                   "sobrepeso II", "obesidad I",
                                   "obesidad II", "obesidad III"))

IMC<- IMC %>%
  mutate(coincidi=IMC$real==IMC$nivel_obesidad)

IMCT <- IMC %>% group_by(coincidi)%>%
  count()

IMCf <- IMC %>% filter(coincidi=="FALSE")
#Los quito del estudio

obesidadq <- obesidad
obesidadq$coincidi=IMC$coincidi

obesidadq <-obesidadq %>%
  filter(coincidi==TRUE) %>%
  select(-coincidi)
obesidad <- obesidad %>% mutate(IMC=peso/(altura*altura))
obesidadq <- obesidadq %>% mutate(IMC=peso/(altura*altura))

obesidadn <- obesidad %>%
  select(-c(fuma, controla_calorias,IMC)) %>%
  filter(consume_entre_comidas!="no", comidas_principales!="mas_de_3",
         consume_alcohol!="siempre", consume_alcohol!="frecuentemente",
         transporte!="bicicleta",transporte!="moto",
         transporte!="caminando")

obesidadn$frecuencia_actividad_fisica <- fct_recode(
  obesidadn$frecuencia_actividad_fisica,
  "mas_de_2_dias"="2_o_4_dias", "mas_de_2_dias"="4_o_5_dias")

obesidadn$tiempo_en_consolas <- fct_recode(
  obesidadn$tiempo_en_consolas, "mas_de_3_horas"="3_5_horas",
  "mas_de_3_horas"="mas_de_5_horas")

obesidadn$consume_entre_comidas <- fct_recode(
  obesidadn$consume_entre_comidas, "casi_siempre"="frecuentemente",
  "casi_siempre"="siempre")

obesidadqn <- obesidadq %>%
  select(-c(fuma, controla_calorias, nivel_obesidad)) %>%
  filter(consume_entre_comidas!="no", comidas_principales!="mas_de_3",
         consume_alcohol!="siempre", consume_alcohol!="frecuentemente",

```

```

    transporte!="bicicleta",transporte!="moto",
    transporte!="caminando")

obesidadqn$frecuencia_actividad_fisica <- fct_recode(
  obesidadqn$frecuencia_actividad_fisica, "mas_de_2_dias"="2_o_4_dias",
  "mas_de_2_dias"="4_o_5_dias")

obesidadqn$tiempo_en_consolas <- fct_recode(
  obesidadqn$tiempo_en_consolas, "mas_de_3_horas"="3_5_horas",
  "mas_de_3_horas"="mas_de_5_horas")

obesidadqn$consume_entre_comidas <- fct_recode(
  obesidadqn$consume_entre_comidas, "casi_siempre"="frecuentemente",
  "casi_siempre"="siempre")

# Arbol de regresion
set.seed(123)
obesidadqn_split_reg <- obesidadqn %>%
  initial_split(prop = 0.6, strata = "IMC")

# obesidadqn_split_reg
test_reg <- testing(obesidadqn_split_reg)
train_reg <- training(obesidadqn_split_reg)
set.seed(123)
val_folds_reg <- vfold_cv(train_reg, strata = IMC)

# Arbol de clasificación
set.seed(123)
obesidadn_split_clas <- obesidadn %>%
  initial_split(prop = 0.6, strata = "nivel_obesidad")
# obesidadn_split_clas
test_clas <- testing(obesidadn_split_clas)
train_clas <- training(obesidadn_split_clas)

test_clas %>%
  count(nivel_obesidad) %>%
  mutate(prop = n/sum(n))
train_clas %>%
  count(nivel_obesidad) %>%
  mutate(prop = n/sum(n))

set.seed(123)
val_folds_clas <- vfold_cv(train_clas, strata=nivel_obesidad)

# Receta
### regresión
obesidadqn_rec_reg <-

```

```

recipe(IMC ~ ., data = train_reg) %>%
# < #>% # eliminar variables con varianza cero
step_novel(all_nominal()) %>% # prepara datos de prueba para manejar
#niveles de factor nunca antes vistos
step_unknown(all_nominal()) %>% # clasifica los datos categóricos
#faltantes (NA) como "desconocidos"
step_impute_median(all_numeric(), -all_outcomes())# reemplaza las
#observaciones numéricas faltantes con la mediana
# summary(obesidadqn_rec_reg) #receta

### clasificación
obesidadqn_rec_clas <-
  recipe(nivel_obesidad ~ ., data = train_clas) %>%
  step_nzv(all_predictors()) %>% # eliminar variables con varianza cero
  # step_novel(all_nominal())%>% # prepara datos de prueba para manejar
  #niveles de factor nunca antes vistos
  # step_unknown(all_nominal())%>% # clasifica los datos categóricos
  #faltantes (NA) como "desconocidos"
  step_impute_median(all_numeric(), -all_outcomes())# reemplaza las
#observaciones numéricas faltantes con la mediana
# summary(obesidadqn_rec_clas) #receta

# Modelo
### regresion
tree_mod_reg <- decision_tree(min_n = tune(), tree_depth = tune(),
                             cost_complexity = tune()) %>%
  set_engine("rpart") %>% #, num.threads = cores
  set_mode("regression")

### clasificacion
tree_mod_clas <- decision_tree(min_n = tune(), tree_depth = tune(),
                              cost_complexity = tune()) %>%
  set_engine("rpart") %>% #, num.threads = cores
  set_mode("classification") #modelo

# Workflow

### regresion
obesidadqn_wflow_reg <-
  workflow() %>%
  add_recipe(obesidadqn_rec_reg) %>%
  add_model(tree_mod_reg)
  #workflow

### clasificacion
obesidadqn_wflow_clas <-
  workflow() %>%

```



```

add_recipe(obesidadqn_rec_clas) %>% #workflow
# add_formula(nivel_obesidad ~ .)%>%
add_model(tree_mod_clas)

### Regression

load("metricas.RData")
# grid_reg <- grid_regular(cost_complexity(),tree_depth(),
# min_n(),levels = 5,filter = min_n<=20 & tree_depth<=10)

# obesidadqn_res_reg_aleat <- obesidadqn_wflow_reg%>%
# tune_grid(resamples = val_folds_reg, grid=10)

metric_aleat_reg <- obesidadqn_res_reg_aleat %>% collect_metrics()

# obesidadqn_res_reg <- obesidadqn_wflow_reg%>%
# tune_grid(resamples = val_folds_reg, grid=grid_reg)

metric_regular_reg <- obesidadqn_res_reg %>% collect_metrics()

best_rmse_reg_aleat <- obesidadqn_res_reg_aleat %>%
  show_best(metric = "rmse",n=5)
best_rsqr_reg_aleat <- obesidadqn_res_reg_aleat %>%
  show_best(metric = "rsqr",n=5)

best0_rmse_reg <- obesidadqn_res_reg_aleat %>%
  select_best(metric = "rmse")
best0_rsqr_reg <- obesidadqn_res_reg_aleat %>%
  select_best(metric = "rsqr")

best_rmse_reg <- obesidadqn_res_reg %>%
  show_best(metric = "rmse",n=5)
best_rsqr_reg <- obesidadqn_res_reg %>%
  show_best(metric = "rsqr",n=5)

best1_rmse_reg <- obesidadqn_res_reg %>%
  select_best(metric = "rmse")
best1_rsqr_reg <- obesidadqn_res_reg %>%
  select_best(metric = "rsqr")
best2_rmse_reg <- best_rmse_reg %>%
  slice(4) %>%
  select(cost_complexity,tree_depth, min_n, .config)
best2_rsqr_reg <- best_rsqr_reg %>%
  slice(4) %>%
  select(cost_complexity,tree_depth, min_n, .config)

```

```

# best_tree_reg <- tibble(cost_complexity=0.000000001, tree_depth=15,
# min_n=2, .config = "Preprocessor1_Model07")

### clasificacion

# grid_clas <- grid_regular(min_n(), tree_depth(), cost_complexity(),
# levels = 5, filter = min_n<=20 & tree_depth<=10)

# obesidadqn_res_clas <- obesidadqn_wflow_clas %>%
#   tune_grid(resamples = val_folds_clas, grid=grid_clas)
#
# obesidadqn_res_clas_aleat <- obesidadqn_wflow_clas %>%
#   tune_grid(resamples = val_folds_clas, grid=10)

# save(obesidadqn_res_reg_aleat,
#      obesidadqn_res_reg,
#      obesidadqn_res_clas,
#      obesidadqn_res_clas_aleat,
#      file="metricas.RData")

metric_aleat_clas <- obesidadqn_res_clas_aleat %>% collect_metrics()
metric_regular_clas <- obesidadqn_res_clas %>% collect_metrics()

# library(openxlsx)
# write.xlsx(metric_regular_clas, "metric_regular_clas.xlsx")

best_rocauc_clas_aleat <- obesidadqn_res_clas_aleat %>%
  show_best(metric = "roc_auc", n=5)
best0_clas <- obesidadqn_res_clas_aleat %>%
  select_best(metric = "roc_auc")

best_rocauc_clas <- obesidadqn_res_clas %>%
  show_best(metric = "roc_auc", n=5)
best1_clas <- obesidadqn_res_clas %>%
  select_best(metric = "roc_auc")
# best2_clas <- best_rocauc_clas %>%
# slice(5) %>% select(cost_complexity, tree_depth, min_n, .config)

# save(best_rmse_reg_aleat,
#      best_rsqa_reg_aleat,
#      best_rmse_reg,
#      best_rsqa_reg,
#      best_rocauc_clas_aleat,
#      best_rocauc_clas,
#      file="best_metric.RData")

# best_tree_clas <- tibble(cost_complexity=0.000000001, tree_depth=15,

```

```

#           min_n=2, .config = "Preprocessor1_Model07")

# finalizar workflow

### Regresion

final_tree_workflow_reg0 <- obesidadqn_wflow_reg %>%
  finalize_workflow(best0_rmse_reg)
final_tree_workflow_reg1 <- obesidadqn_wflow_reg %>%
  finalize_workflow(best1_rmse_reg)
final_tree_workflow_reg2 <- obesidadqn_wflow_reg %>%
  finalize_workflow(best2_rmse_reg)

final_tree_workflow_reg0_rsqu <- obesidadqn_wflow_reg %>%
  finalize_workflow(best0_rsqu_reg)
final_tree_workflow_reg1_rsqu <- obesidadqn_wflow_reg %>%
  finalize_workflow(best1_rsqu_reg)
final_tree_workflow_reg2_rsqu <- obesidadqn_wflow_reg %>%
  finalize_workflow(best2_rsqu_reg)

# final_tree_workflow_reg <- obesidadqn_wflow_reg %>%
#           finalize_workflow(best_tree_reg)

### Clasificacion

final_tree_workflow_clas0 <- obesidadqn_wflow_clas %>%
  finalize_workflow(best0_clas)
final_tree_workflow_clas1 <- obesidadqn_wflow_clas %>%
  finalize_workflow(best1_clas)
# final_tree_workflow_clas2 <- obesidadqn_wflow_clas %>%
#           finalize_workflow(best2_clas)

# final_tree_workflow_clas <- obesidadqn_wflow_clas %>%
#           finalize_workflow(best_tree_clas)

### regresion
# tree_wf_fit_reg <- final_tree_workflow_reg %>%
#           fit(data = train_reg)

tree_wf_fit_reg0 <- final_tree_workflow_reg0 %>%
  fit(data = train_reg)
tree_wf_fit_reg1 <- final_tree_workflow_reg1 %>%
  fit(data = train_reg)
tree_wf_fit_reg2 <- final_tree_workflow_reg2 %>%

```

```

        fit(data = train_reg)

#clasificacion
# tree_wf_fit_clas <- final_tree_workflow_clas%>%
#           fit(data = train_clas)

tree_wf_fit_clas0 <- final_tree_workflow_clas0 %>%
  fit(data = train_clas)
tree_wf_fit_clas1 <- final_tree_workflow_clas1 %>%
  fit(data = train_clas)
# tree_wf_fit_clas2 <- final_tree_workflow_clas2%>%
#           fit(data = train_clas)

### regresion
# tree_fit_reg <- tree_wf_fit_reg%>%
#           extract_fit_parsnip()
tree_fit_reg0 <- tree_wf_fit_reg0 %>%
  extract_fit_parsnip()
tree_fit_reg1 <- tree_wf_fit_reg1 %>%
  extract_fit_parsnip()
tree_fit_reg2 <- tree_wf_fit_reg2 %>%
  extract_fit_parsnip()

### clasifiacion
# tree_fit_clas <- tree_wf_fit_clas%>%
#           extract_fit_parsnip()
tree_fit_clas0 <- tree_wf_fit_clas0 %>%
  extract_fit_parsnip()
tree_fit_clas1 <- tree_wf_fit_clas1 %>%
  extract_fit_parsnip()
# tree_fit_clas2 <- tree_wf_fit_clas2%>%
#           extract_fit_parsnip()

###regresion
# tree_last_fit_reg <- final_tree_workflow_reg%>%
#           last_fit(obesidadqn_split_reg)

tree_last_fit_reg0 <- final_tree_workflow_reg0 %>%
  last_fit(obesidadqn_split_reg)
tree_last_fit_reg1 <- final_tree_workflow_reg1 %>%
  last_fit(obesidadqn_split_reg)
tree_last_fit_reg2 <- final_tree_workflow_reg2 %>%
  last_fit(obesidadqn_split_reg)

# tree_last_fit_reg%>% collect_metrics()
tree_last_fit_reg0 %>% collect_metrics()

```

```

tree_last_fit_reg1 %>% collect_metrics()
tree_last_fit_reg2 %>% collect_metrics()

###clasificacion

# tree_last_fit_clas <- final_tree_workflow_clas%>%
#     last_fit(obesidadn_split_clas)

tree_last_fit_clas0 <- final_tree_workflow_clas0 %>%
    last_fit(obesidadn_split_clas)
tree_last_fit_clas1 <- final_tree_workflow_clas1 %>%
    last_fit(obesidadn_split_clas)
# tree_last_fit_clas2 <- final_tree_workflow_clas2%>%
#     last_fit(obesidadn_split_clas)

# tree_last_fit_clas%>% collect_metrics()
tree_last_fit_clas0 %>% collect_metrics()
tree_last_fit_clas1 %>% collect_metrics()
# tree_last_fit_clas2%>% collect_metrics()

###regresion
# prediccion_2 <- predict(tree_fit_reg, new_data = test_reg,
# type = "numeric")
prediccion0_2 <- predict(tree_fit_reg0, new_data = test_reg,
    type = "numeric")
prediccion1_2 <- predict(tree_fit_reg1, new_data = test_reg,
    type = "numeric")
prediccion2_2 <- predict(tree_fit_reg2, new_data = test_reg,
    type = "numeric")

###clasificacion

# prediccion_1 <- predict(tree_fit_clas, new_data = test_clas,
# type = "class")
prediccion0_1 <- predict(tree_fit_clas0, new_data = test_clas,
    type = "class")
prediccion1_1 <- predict(tree_fit_clas1, new_data = test_clas,
    type = "class")
# prediccion2_1 <- predict(tree_fit_clas2, new_data = test_clas,
# type = "class")

# ``

# ``{r}
fluidRow(
  column(3,

```

```

        sliderInput("Edad",label = "Introduce tu edad",
                    min=1, max=100, value = 16,animate = TRUE)),
column(3,
      selectInput("Genero",label = "Selecciona tu género",
                  choices = c("Mujer", "Hombre"))),
column(3,selectInput("Vegetales",label = "¿Consumes vegetales?",
                    choices = c("nunca", "algunas veces",
                                "siempre"))),
column(3,selectInput("Familiar",label = "¿Familiar con sobrepeso?",
                    choices = c("si", "no")))
)
fluidRow(
  column(3,
        sliderInput("Peso",label = "Introduce tu peso", min=0,
                    max=150, value = 50, step = 0.01,animate = TRUE)),
  column(3,
        selectInput("NCP",label = "¿Número de comidas principales?",
                    choices = c("entre 1 y 2", "tres",
                                "mas de 3", "Otro"))),
  column(3,
        selectInput("CEC",label = "¿Consumes entre comidas?",
                    choices = c("no", "algunas veces",
                                "casi siempre"))),
  column(3,
        selectInput("Hipercaloricos",
                    label = "¿Consumes hipercalóricos?",
                    choices = c("si", "no")))
)
fluidRow(
  column(3,
        sliderInput("Altura",label = "Introduce tu altura", min=0,
                    max=3, value = 1.5, step = 0.01,animate = TRUE)),
  column(3,
        selectInput("Agua",label = "¿consume agua al día?",
                    choices = c("menos de 1l", "entre 1l y 2l",
                                "mas de 2l", "Otro"))),
  column(3,
        selectInput("Actividad",label = "¿Frecuencia actividad física?",
                    choices = c("no tengo", "1 o 2 dias",
                                "mas de 2 dias", "Otro"))),
  column(3,
        selectInput("Consolas",label = "¿Tiempo en consolas?",
                    choices = c("de 0 a 2 horas", "mas de 3 horas",
                                "Otro")))
)
fluidRow(

```

```

column(3,
  selectInput("Alcohol",label = "¿Consume alcohol?",
    choices = c("no", "algunas veces",
      "frecuentemente", "siempre"))),
column(3,
  selectInput("Transporte",label = "¿Como suele desplazarse?",
    choices = c("transporte publico", "caminando",
      "automovil", "moto", "bicicleta")),
column(3,
  selectInput("select3", "Elección del arbol:",
    choices = c("Regresión", "Clasificación"))),
column(3,
  actionButton("myboton0", "Calcular",
    class = "btn btn-success"))
)
fluidRow(
  column(12, textOutput("mytext1"))
)

datasetInput3 <- reactive({
  switch(input$select3,
    "Clasificación" = obesidadn,
    "Regresión" = obesidadqn
  ) })

myvalor_clas <- reactiveValues(data1=NULL)
myvalor_reg <- reactiveValues(data1=NULL)

observeEvent(input$myboton0, {
  edad <- input$Edad
  genero <- input$Genero
  altura <- input$Altura
  peso <- input$Peso
  familiar_sobrepeso <- input$Familiar
  consume_hipercaloricos <-input$Hipercaloricos

  consume_vegetales<-NULL
  if(input$Vegetales=="algunas veces"){
    consume_vegetales<-"algunas_veces"
  } else {
    consume_vegetales<-input$Vegetales
  }

  comidas_principales <- NULL
  if(input$NCP=="entre 1 y 2"){
    comidas_principales<-"entre_1_y_2"
  } else if(input$NCP=="tres"){

```

```
comidas_principales<-"tres"
} else if(input$NCP=="mas de 3"){
  comidas_principales<-"mas_de_3"
} else if(input$NCP=="Otro"){
  comidas_principales<-"Other"
}

consume_entre_comidas <- NULL
if(input$CEC=="algunas veces"){
  consume_entre_comidas<-"algunas_veces"
} else if(input$CEC!="no"){
  consume_entre_comidas<-"no"
} else if(input$CEC!="casi siempre"){
  consume_entre_comidas<-"casi_siempre"
}

consume_agua_dia <- NULL
if(input$Agua=="menos de 1l"){
  consume_agua_dia<-"menos_de_1l"
} else if(input$Agua=="entre 1l y 2l"){
  consume_agua_dia<-"entre_1l_y_2l"
} else if(input$Agua=="mas de 2l"){
  consume_agua_dia<-"mas_de_2l"
} else if(input$Agua=="Otro"){
  consume_agua_dia<-"Other"
}

frecuencia_actividad_fisica <- NULL
if(input$Actividad=="no tengo"){
  frecuencia_actividad_fisica<-"no_tengo"
} else if(input$Actividad=="1 o 2 dias"){
  frecuencia_actividad_fisica<-"1_o_2_dias"
} else if(input$Actividad=="mas de 2 dias"){
  frecuencia_actividad_fisica<-"mas_de_2_dias"
} else if(input$Actividad=="Otro"){
  frecuencia_actividad_fisica<-"Other"
}

tiempo_en_consolas <- NULL
if(input$Consolas=="de 0 a 2 horas"){
  tiempo_en_consolas<-"0_2_horas"
} else if(input$Consolas=="mas de 3 horas"){
  tiempo_en_consolas<-"mas_de_3_horas"
} else if(input$Consolas=="Otro"){
  tiempo_en_consolas<-"Other"
}
```



```

consume_alcohol<-NULL
if(input$Alcohol=="algunas veces"){
  consume_alcohol<-"algunas_veces"
} else {
  consume_alcohol<-input$Alcohol
}

transporte<-NULL
if(input$Transporte=="transporte publico"){
  transporte<-"transporte_publico"
} else {
  transporte<-input$Transporte
}

nuevo <- tibble(edad, genero, altura, peso, familiar_sobrepeso,
               consume_hipercaloricos, consume_vegetales,
               comidas_principales, consume_entre_comidas,
               consume_agua_dia, frecuencia_actividad_fisica,
               tiempo_en_consolas, consume_alcohol, transporte)
if(colnames(datasetInput3())[15]!="IMC"){
  prediccion_3 <- predict(tree_fit_clas0, new_data = nuevo,
                          type = "class")
  myvalor_clas$data1 <- prediccion_3$.pred_class

} else {
  prediccion_4 <- predict(tree_fit_reg0, new_data = nuevo,
                          type = "numeric")
  myvalor_reg$data1 <- prediccion_4$.pred
}

})

output$mytext1 <- renderText({
  if(colnames(datasetInput3())[15]=="IMC" & !is.null(myvalor_reg$data1)) {
    return(paste("El IMC es de:",myvalor_reg$data1,sep=" "))
  } else if(colnames(datasetInput3())[15]!="IMC" &
            !is.null(myvalor_clas$data1)) {
    return(paste("El nivel de obesidad es de:",
                myvalor_clas$data1,sep=" "))
  } else if(colnames(datasetInput3())[15]=="IMC" &
            is.null(myvalor_reg$data1)) {
    return("Calcule su IMC")
  } else if(colnames(datasetInput3())[15]!="IMC" &
            is.null(myvalor_clas$data1)) {
    return("Calcule su nivel de obesidad")}
})
#`´´´

#`´´´{r}
fluidRow(
tags$script(type="module",

```

```

"document.getElementsByClassName('dropdown-item')[2].href='#'
document.getElementsByClassName('dropdown-item')[3].href='#'
document.getElementsByClassName('dropdown-item')[5].href='#'
document.getElementsByClassName('dropdown-item')[7].href='#'
document.getElementsByClassName('dropdown-item')[8].href='#'
document.getElementsByClassName('dropdown-item')[9].href='#'
document.getElementsByClassName('dropdown-item')[2].parentNode.className='disabled'
document.getElementsByClassName('dropdown-item')[3].parentNode.className='disabled'
document.getElementsByClassName('dropdown-item')[5].parentNode.className='disabled'
document.getElementsByClassName('dropdown-item')[7].parentNode.className='disabled'
document.getElementsByClassName('dropdown-item')[8].parentNode.className='disabled'
document.getElementsByClassName('dropdown-item')[9].parentNode.className='disabled')
)
#`´´´

```

B.1.1.1. IMPORT

Esta sección se corresponde con la carpeta IMPORT de la aplicación estadística, la cual contiene un archivo “app.R”

B.1.1.1.1. app.R

```

library(shiny)
library(slickR)

ui <- fluidPage(
  slickROutput("slickr", width = "90%")
)

server <- function(input, output) {

  output$slickr <- renderSlickR({
    imgs <- list.files("imagenes/", pattern=".png", full.names = TRUE)
    slickR(imgs,width = "80%")
  })

}

shinyApp(ui = ui, server = server)

```

B.1.2. CP

Esta subsección contiene cuatro apartados, los cuales corresponden, a los cuatro archivos de códigos de la carpeta CP.

B.1.2.1. index.R

```

source("CP/comb.R")
source("CP/mod.R")
source("CP/vect.R")

```

B.1.2.2. comb.R

```

library(shiny)
library(shinyjs)

```

```

uiComb <- function(n){
  ui <- switch (n,
    fluidPage(
      withMathJax(),
      tags$script(HTML("MathJax.Hub.Config({
        tex2jax: {inlineMath: [['$','$']]
      });")),
      useShinyjs(),
      titlePanel("Combinación de números"),
      sidebarLayout(
        sidebarPanel(
          h4("Parámetros de entrada"),
          p(withMathJax("$P_n=n!=n\\cdot(n-1)\\cdot\\ldots\\cdot 1$")),
          numericInput("PerSin0", "Tamaño n",
            10, 0, Inf),
          actionButton("boton0", "Ejemplo")
        ),
        mainPanel(
          tags$head(tags$style(HTML(".foo { display: none; }"))),
          tags$div(id = "texto", class = "foo", tags$h3("Ejemplo"),
            tags$p("En una estantería tenemos 10 libros diferentes, cada
              uno de una asignatura distinta. ¿De cuántas formas los podemos
              colocar?"), tags$strong("Solución"), tags$p("En la
              primera posición se pueden colocar cualquiera de los 10 libros.
              Una vez colocado este, en la segunda posición tan solo se
              pueden poner 9 libros. Para la tercera posición 8 libros, así
              sucesivamente. Por lo tanto las formas de colocar 10 libros en
              10 posiciones es  $10! = 3628800$ .")),
          textOutput("salida0")
        )
      )
    ),
    fluidPage(
      withMathJax(),
      tags$script(HTML("MathJax.Hub.Config({
        tex2jax: {inlineMath: [['$','$']]
      });")),
      useShinyjs(),
      titlePanel("Combinación de números"),
      sidebarLayout(
        sidebarPanel(
          h4("Parámetros de entrada"),
          p(withMathJax("$PR_n^{a,b,c,...}=
            \\frac{P_n}{a! \\cdot b! \\cdot c! \\cdot \\ldots}:
            n=a+b+c+\\ldots$")),
          numericInput("PerCon0", "Tamaño n",
            10, 0, Inf),
          helpText("Escriba el tamaño de cada subgrupo separado por comas,
            por ejemplo, 1,2,3"),
          textInput("grupo", "Introduzca el tamaños de los subgrupos,
            a, b, c, ..."),
          actionButton("boton1", "Ejemplo")
        ),
        mainPanel(
          tags$head(tags$style(HTML(".foo { display: none; }"))),
          tags$div(id = "texto0", class = "foo", tags$h3("Ejemplo"),
            tags$p("En el palo de señales de un barco se pueden izar

```

```

    tres banderas rojas, dos azules y cuatro verdes. ¿Cuántas
    señales distintas pueden indicarse con la colocación de
    las nueve banderas?"),
    tags$p(tags$strong("Solución")),
    tags$p("Primero formemos grupos con los elementos de la misma
    clase. El primero es formado por las 3 banderas rojas, el
    segundo por las 2 banderas azules y el ultimo por las 4
    verdes. Si a,b,c denotan el número de banderas en cada grupo
    tenemos que"),
    tags$p(withMathJax("$$a=3, \\; b=2, \\; c=4, \\; n=a+b+c=9$$")),
    tags$p("Las permutaciones de estos elementos cumplen los
    siguiente. Sí entran todos los elementos. Sí importa el orden.
    Sí se repiten los elementos"),
    tags$p("Luego se pueden realizar 1260 combinaciones")),
    uiOutput("salida1")
  )
)
),
fluidPage(
  withMathJax(),
  tags$script(HTML("MathJax.Hub.Config({
    tex2jax: {inlineMath: [['$','$']]
  });")),
  useShinyjs(),
  titlePanel("Combinación de números"),
  sidebarLayout(
    sidebarPanel(
      h4("Parámetros de entrada"),
      p(withMathJax("$PC_n=P_{n-1}=(n-1)!$")),
      numericInput("PerCir0", "Tamaño n",
        10, 0, Inf),
      actionButton("boton3", "Ejemplo")
    ),
    mainPanel(
      tags$head(tags$style(HTML(".foo { display: none; }"))),
      tags$div(id = "texto1", class = "foo", tags$h3("Ejemplo"),
        tags$p("Determine cuantas pulsera circulares pueden
        hacerse con 7 bolas de diferentes colores (los colores
        no se repiten)."), tags$p(tags$strong("Solución")),
        tags$p("Dos pulseras se consideran iguales si es posible
        rotar una de ellas de manera que coincida con la otra.
        Supongamos, por ejemplo, que dos pulseras contienen el
        color rojo. Para comprobar si son iguales o no,
        colocamos ambas pulseras de tal manera que el color
        rojo aparezca en la misma posición en ambas pulseras
        (por ejemplo arriba del todo). Si tras hacer esto el
        resto de bolas aparecen en la misma posición, entonces
        las pulseras son iguales y en caso contrario son
        distintas."), tags$p("Teniendo en cuenta lo anterior,
        podemos contar el número de pulseras diferentes que
        se pueden hacer con n bolas de n colores diferentes.
        Para ello fijamos una posición (por ejemplo, arriba) y
        en ella ponemos un color rojo (por ejemplo, rojo).
        Las restantes n-1 posiciones se rellenan con los n-1
        colores restantes. El orden es importante (pues
        diferentes ordenaciones dan lugar a pulseras distintas)
        y no hay repeticiones, por tanto hay (n-1)!

```

```

        pulseras diferentes.")),
      textOutput("salida2")
    )
  )
),
fluidPage(
  withMathJax(),
  tags$script(HTML("MathJax.Hub.Config({
    tex2jax: {inlineMath: [['$','$']]
  });")),
  useShinyjs(),
  titlePanel("Combinación de números"),
  sidebarLayout(
    sidebarPanel(
      h4("Parámetros de entrada"),
      p(withMathJax("$P_{N,n}=N(N-1)\\cdot\\ldots\\cdot (N-n+1)$")),
      numericInput("Perlong0", "Tamaño N", 10, 1, Inf),
      numericInput("Perlong1", "Tamaño n", 4, 1, Inf),
      actionButton("boton4", "Ejemplo")
    ),
    mainPanel(
      tags$head(tags$style(HTML(".foo { display: none; }"))),
      tags$div(id = "texto2", class = "foo", tags$h3("Ejemplo"),
        tags$p("Si no se permiten repeticiones. ¿Cuántos números
de tres cifras se pueden formar con los seis dígitos 2,
3, 5, 6, 7, 9 ?."), tags$p(tags$strong("Solución")),
        tags$p("Hay 6 dígitos diferentes, no pueden repetirse e
importa el orden. Se trata de permutaciones de longitud
3 de un conjunto de 6 elementos,
          por tanto hay  $6 \times 5 \times 4 = 120$ ."),
        uiOutput("salida3")
      )
    )
  ),
fluidPage(
  withMathJax(),
  tags$script(HTML("MathJax.Hub.Config({
    tex2jax: {inlineMath: [['$','$']]
  });")),
  useShinyjs(),
  titlePanel("Combinación de números"),
  sidebarLayout(
    sidebarPanel(
      h4("Parámetros de entrada"),
      p(withMathJax("$C_m^n = \\binom{m}{n} = \\frac{m!}{n! \\cdot (m-n)!}$")),
      numericInput("Comb0", "Tamaño N",
        10, 0, Inf),
      numericInput("Comb1", "Tamaño n",
        10, 0, Inf),
      actionButton("boton5", "Ejemplo")
    ),
    mainPanel(
      tags$head(tags$style(HTML(".foo { display: none; }"))),
      tags$div(id = "texto3", class = "foo", tags$h3("Ejemplo"),
        tags$p("Un examen tipo test se compone de 15 cuestiones.
Todas las cuestiones tienen el mismo valor y las

```

```

      respuestas incorrectas no restan nota. Determine el
      número de formas posibles en las que un alumno puede
      obtener la calificación de 6 puntos sobre 10."),
      tags$p(tags$strong("Solución")), tags$p("Para conseguir
      6 puntos debe responder correctamente 9 de las 15
      preguntas. No importa el orden en que se respondan las
      preguntas y no pueden repetirse las respuestas
      (combinaciones). Hay 5005 formas de responder")),
      uiOutput("salida4")
    )
  )
),
fluidPage(
  withMathJax(),
  tags$script(HTML("MathJax.Hub.Config({
    tex2jax: {inlineMath: [['$','$']]
  });")),
  useShinyjs(),
  titlePanel("Combinación de números"),
  sidebarLayout(
    sidebarPanel(
      h4("Parámetros de entrada"),
      p(withMathJax("$2^{n}$")),
      numericInput("TodComb0", "Tamaño n",
        10, 0, Inf),
      actionButton("boton6", "Ejemplo")
    ),
    mainPanel(
      tags$head(tags$style(HTML(".foo { display: none; }"))),
      tags$div(id = "texto4", class = "foo", tags$h3("Ejemplo"),
        tags$p("Un químico posee en su laboratorio una balanza
        que tiene las siguientes pesas: 1 gr., 5 gr., 10 gr.,
        25 gr. y 50 gr. ¿Cuántas pesadas diferentes puede
        hacer?"), tags$p(tags$strong("Solución")),
        tags$p("Suponemos que para cada pesada puede usarse
        cada una de las pesas disponibles a lo sumo una vez.
        El orden en que las pesas son seleccionadas no importa.
        Entonces, cada posible pesada se corresponde con un
        único subconjunto del conjunto de pesas {1, 5, 10,
        25, 50}. Por ejemplo a una pesada de 31 gr. le
        corresponde el subconjunto {1, 5, 25} y a una pesada
        de 15 gr. el subconjunto {5, 10}. El conjunto {1, 5, 10,
        25, 50} tiene  $2^5$ , "= 32 subconjuntos,
        luego hay 32 posibles pesadas diferentes (admitiendo
        que el conjunto vacío; corresponde con una pesada de
        0 gr.)")),
      uiOutput("salida5")
    )
  )
)
)

server <- switch (n,
  function(input, output) {
    observeEvent(input$boton0, {
      toggle("texto")
    })
  }
)

```

```

output$salida0 <- renderText({
  paste("El resultado de", paste0(input$PerSin0,"! es"),
        factorial(input$PerSin0))
})
},
function(input, output){
Pr<-function(n,k){
  if (n==sum(k)){
    Pr<-paste0(factorial(n)/prod(factorial(k)))
  }
  else{
    Pr<-paste0(
      "La suma de los subgrupos no coincide con el tamaño ", n)
  }
  return(Pr)
}
observeEvent(input$boton1, {
  toggle("texto0")
})
output$salida1 <- renderUI({
  a=as.numeric(strsplit(c(input$grupo),",")[[1]])
  n=input$PerCon0
  if (n==sum(a)){
    div("El resultado de ",
        withMathJax("$PR_{",input$PerCon0,"}^{",input$grupo,"}$"),
        " es ", Pr(input$PerCon0,a))
  } else{Pr(input$PerCon0,a)}
})
},
function(input, output) {
  observeEvent(input$boton3, {
    toggle("texto1")
  })
  output$salida2 <- renderText({
    paste("El resultado de",
          paste0("(" ,input$PerCir0,"-1)! es"),
          factorial(input$PerCir0-1))
  })),
function(input, output) {
  observeEvent(input$boton4, {
    toggle("texto2")
  })
  output$salida3 <- renderUI({
    if(input$Perlong0<input$Perlong1){
      div("El tamaño", withMathJax("$n$"), "no puede ser mayor que",
          withMathJax("$N$"))
    } else{
      n=max(cumprod(c(input$Perlong0:(input$Perlong0-input$Perlong1+1))))
      div("El resultado de ",
          withMathJax("$P_{",input$Perlong0,",",input$Perlong1,"}$"),
          "es",n)}
  })),
function(input, output) {
  observeEvent(input$boton5, {
    toggle("texto3")
  })
  output$salida4 <- renderUI({

```

```

    div("El resultado de",
        withMathJax("$\\dbinom{" ,input$Comb0,"}{" ,input$Comb1,"}$"),
        "es", choose(input$Comb0,input$Comb1))
  }},
function(input, output) {
  observeEvent(input$boton6, {
    toggle("texto4")
  })
  output$salida5 <- renderUI({
    div("El resultado de",
        withMathJax("$2^{",input$TodComb0,"}$"),2^(input$TodComb0))
  })
}
)
shinyApp(ui,server)
}

```

B.1.2.3. mod.R

```

library(shiny)
library(readxl)
library(ggplot2)
library(plotly)

uiMod <- function(n){
  ui <- switch (n,
    fluidPage(
      titlePanel("Modelos de variables aleatorias"),
      sidebarLayout(
        sidebarPanel(
          h4("Parámetros de entrada"),
          helpText("Cargue un archivo '.xlsx' con los valores
de la variable en la primera columna y la
probabilidad de cada valor en la segunda columna,
sin encabezado"),
          fileInput("arch0",
            "Selecciona el excel", accept=".xlsx"),
          numericInput("moment0",
            "Momento de orden:", 2, 0, Inf),
          numericInput("Probigual0",
            "Probabilidad igual a:", 1, 0, Inf),
          numericInput("Probmenorigual0",
            "Probabilidad menor o igual que:",
            1, 0, Inf),
          numericInput("Probmayorigual0",
            "Probabilidad mayor o igual que:",
            1, 0, Inf),
          numericInput("Probmenor0",
            "Probabilidad menor que:", 1, 0, Inf),
          numericInput("Probmayor0",
            "Probabilidad mayor que:", 1, 0, Inf),
          sliderInput("Probentrececer0",
            "Probabilidad entre [,]:", min = 0,
            max = 1000, value = c(0,5)),
          sliderInput("Probentreabi0",
            "Probabilidad entre (,):", min = 0,
            max = 1000, value = c(0,5)),
          sliderInput("Probentrececerabi0",

```



```

        "Probabilidad entre [,):", min = 0,
        max = 1000, value = c(0,5)),
    sliderInput("Probentreabicer0",
        "Probabilidad entre (,):", min = 0,
        max = 1000, value = c(0,5)),
  ),
  mainPanel(
    uiOutput("salida0")
  )
),
fluidPage(
  withMathJax(),
  tags$script(HTML("MathJax.Hub.Config({
    tex2jax: {inlineMath: [['$','$']]
  });")),
  titlePanel("Modelos de variables aleatorias"),
  sidebarLayout(
    sidebarPanel(
      h4("Parámetros de entrada"),
      sliderInput("n0",
        withMathJax("Valores de  $x_{(1)}$  y  $x_{(N)}$ "),
        1,500,value = c(1,10),step=1,animate=TRUE),
      numericInput("moment1",
        "Momento de orden:", 2, 0, Inf),
      numericInput("Probigual1",
        "Probabilidad igual a:", 1, 0, Inf),
      numericInput("Probmenorigual1",
        "Probabilidad menor o igual que:",
        1, 0, Inf),
      numericInput("Probmayorigual1",
        "Probabilidad mayor o igual que:",
        1, 0, Inf),
      numericInput("Probmenor1",
        "Probabilidad menor que:", 2, 0, Inf),
      numericInput("Probmayor1", "
        Probabilidad mayor que:", 2, 0, Inf),
      sliderInput("Probentreacer1",
        "Probabilidad entre [,):", min = 1,
        max = 1000, value = c(1,5)),
      sliderInput("Probentreabili",
        "Probabilidad entre (,):", min = 1,
        max = 1000, value = c(1,5)),
      sliderInput("Probentreacerabi1",
        "Probabilidad entre [,):", min = 1,
        max = 1000, value = c(1,5)),
      sliderInput("Probentreabicer1",
        "Probabilidad entre (,):", min = 1,
        max = 1000, value = c(1,5)),
    ),
    mainPanel(
      uiOutput("salida1")
    )
  )
),
fluidPage(
  titlePanel("Modelos de variables aleatorias"),

```

```

sidebarLayout(
  sidebarPanel(
    h4("Parámetros de entrada"),
    sliderInput("p0", "Valor de p",0,1,0.5,
               step=0.01,animate=TRUE),
    numericInput("moment2",
                "Momento de orden:", 2, 0, Inf)
  ),
  mainPanel(
    uiOutput("salida2")
  )
),
fluidPage(
  titlePanel("Modelos de variables aleatorias"),
  sidebarLayout(
    sidebarPanel(
      h4("Parámetros de entrada"),
      sliderInput("p1", "Valor de p",0,1,0.5,
                  step=0.01,animate=TRUE),
      sliderInput("n1", "Valor de N",1,500,1,
                  step=1,animate=TRUE),
      numericInput("moment3",
                  "Momento de orden:", 2, 0, Inf),
      numericInput("Probigual2",
                  "Probabilidad igual a:", 1, 0, Inf),
      numericInput("Probmenorigual2",
                  "Probabilidad menor o igual que:",
                  1, 0, Inf),
      numericInput("Probmayorigual2",
                  "Probabilidad mayor o igual que:",
                  1, 0, Inf),
      numericInput("Probmenor2",
                  "Probabilidad menor que:", 1, 0, Inf),
      numericInput("Probmayor2",
                  "Probabilidad mayor que:", 1, 0, Inf),
      sliderInput("Probentreocer2",
                  "Probabilidad entre [,]:", min = 0,
                  max = 1000, value = c(0,5)),
      sliderInput("Probentreabi2",
                  "Probabilidad entre (,):", min = 0,
                  max = 1000, value = c(0,5)),
      sliderInput("Probentreocerabi2",
                  "Probabilidad entre [,):", min = 0,
                  max = 1000, value = c(0,5)),
      sliderInput("Probentreabicer2",
                  "Probabilidad entre (,]:", min = 0,
                  max = 1000, value = c(0,5)),
    ),
    mainPanel(
      uiOutput("salida3")
    )
  )
),
fluidPage(
  titlePanel("Modelos de variables aleatorias"),
  sidebarLayout(

```

```

sidebarPanel(
  h4("Parámetros de entrada"),
  sliderInput("p2", "Valor de p",0,500,1,
    step=0.01,animate=TRUE),
  sliderInput("n2", "Valor de N",1,500,2,
    step=1,animate=TRUE),
  numericInput("moment4", "Momento de orden:",
    2, 0, Inf),
  numericInput("Probigual3",
    "Probabilidad igual a:", 1, 0, Inf),
  numericInput("Probmenorigual3",
    "Probabilidad menor o igual que:",
    1, 0, Inf),
  numericInput("Probmayorigual3",
    "Probabilidad mayor o igual que:",
    1, 0, Inf),
  numericInput("Probmenor3",
    "Probabilidad menor que:", 1, 0, Inf),
  numericInput("Probmayor3",
    "Probabilidad mayor que:", 1, 0, Inf),
  sliderInput("Probentrececer3",
    "Probabilidad entre [,]:" , min = 0,
    max = 1000, value = c(0,5)),
  sliderInput("Probentreabi3",
    "Probabilidad entre (,):" , min = 0,
    max = 1000, value = c(0,5)),
  sliderInput("Probentrececerabi3",
    "Probabilidad entre [,,):" , min = 0,
    max = 1000, value = c(0,5)),
  sliderInput("Probentreabicer3",
    "Probabilidad entre (,,):" , min = 0,
    max = 1000, value = c(0,5)),
),
mainPanel(
  uiOutput("salida4")
)
),
fluidPage(
  titlePanel("Modelos de variables aleatorias"),
  sidebarLayout(
    sidebarPanel(
      h4("Parámetros de entrada"),
      sliderInput("p3", "Valor de p",0,1,0.5,
        step=0.01,animate=TRUE),
      sliderInput("n3", "Valor de N",1,500,1,
        step=1,animate=TRUE),
      numericInput("moment5",
        "Momento de orden:", 2, 0, Inf),
      numericInput("Probigual4",
        "Probabilidad igual a:", 1, 0, Inf),
      numericInput("Probmenorigual4",
        "Probabilidad menor o igual que:",
        1, 0, Inf),
      numericInput("Probmayorigual4",
        "Probabilidad mayor o igual que:",
        1, 0, Inf),

```

```

        numericInput("Probmenor4",
                    "Probabilidad menor que:", 1, 0, Inf),
        numericInput("Probmayor4",
                    "Probabilidad mayor que:", 1, 0, Inf),
        sliderInput("Probentreecer4",
                    "Probabilidad entre [,]:" , min = 0,
                    max = 1000, value = c(0,5)),
        sliderInput("Probentreabi4",
                    "Probabilidad entre (,):" , min = 0,
                    max = 1000, value = c(0,5)),
        sliderInput("Probentreecerabi4",
                    "Probabilidad entre [,,):" , min = 0,
                    max = 1000, value = c(0,5)),
        sliderInput("Probentreabicer4",
                    "Probabilidad entre (,):" , min = 0,
                    max = 1000, value = c(0,5)),
    ),
    mainPanel(
        uiOutput("salida5")
    )
),
fluidPage(
    titlePanel("Modelos de variables aleatorias"),
    sidebarLayout(
        sidebarPanel(
            h4("Parámetros de entrada"),
            sliderInput("n4", "Valor de n",1,500,1,
                        step=1,animate=TRUE),
            sliderInput("n5", "Valor de N1",1,500,1,
                        step=1,animate=TRUE),
            sliderInput("n6", "Valor de N",1,500,1,
                        step=1,animate=TRUE),
            numericInput("moment6",
                        "Momento de orden:" , 2, 0, Inf),
            numericInput("Probigual5",
                        "Probabilidad igual a:" , 1, 0, Inf),
            numericInput("Probmenorigual5",
                        "Probabilidad menor o igual que:" ,
                        1, 0, Inf),
            numericInput("Probmayorigual5",
                        "Probabilidad mayor o igual que:" ,
                        1, 0, Inf),
            numericInput("Probmenor5",
                        "Probabilidad menor que:" , 1, 0, Inf),
            numericInput("Probmayor5",
                        "Probabilidad mayor que:" , 1, 0, Inf),
            sliderInput("Probentreecer5",
                        "Probabilidad entre [,]:" , min = 0,
                        max = 1000, value = c(0,5)),
            sliderInput("Probentreabi5",
                        "Probabilidad entre (,):" , min = 0,
                        max = 1000, value = c(0,5)),
            sliderInput("Probentreecerabi5",
                        "Probabilidad entre [,,):" , min = 0,
                        max = 1000, value = c(0,5)),
            sliderInput("Probentreabicer5",

```

```

        "Probabilidad entre (,]:", min = 0,
        max = 1000, value = c(0,5)),
    ),
    mainPanel(
      uiOutput("salida6")
    )
  )
)

server <- switch (n,
  function(input, output) {
    path <- reactive({input$aarch0})
    tabl<-reactive({
      file <- path()
      ext <- tools::file_ext(file$datapath)
      req(file)
      validate(need(ext == "xlsx",
        "Por favor cargue un archivo xlsx"))
      tabla<-read_excel(file$datapath,
        col_names = c("Valor",
          "Probabilidad"))
    })
    output$salida0 <-renderUI({
      if(!is.null(path())){
        div(
          renderDataTable({tabl()}),
          renderText({
            paste("El valor esperado es",
              round(sum(tabl()[1]*tabl()[2]),4))}),
          renderText({
            paste("La varianza es",
              round(sum((tabl()[1]^2)*tabl()[2])-
                (sum(tabl()[1]*tabl()[2]))^2,4)
            )),
          renderText({
            paste("El momento de orden ",
              input$moment0," es ",
              round(sum((tabl()[1]^input$moment0)*
                tabl()[2]),4))}),
          renderText({
            paste("La probabilidad de que sea igual a ",
              input$Probigual0," es ",
              round(sum(tabl()[
                tabl()$Valor==input$Probigual0,
                ],[2]),4))}),
          renderText({
            paste(
              "La probabilidad de que sea menor o igual que ",
              input$Probmenorigual0," es ",
              round(sum(tabl()[tabl()$Valor<=input$Probmenorigual0],[2]),
                4))}),
          renderText({
            paste(
              "La probabilidad de que sea mayor o igual que ",
              input$Probmayorigual0," es ",
              round(sum(tabl()[tabl()$Valor>=input$Probmayorigual0],[2]),

```

```

4))}),
  renderText({
    paste("La probabilidad de que sea menor que ",
          input$Probmenor0, " es ",
    round(sum(tabl()[tabl()$Valor<input$Probmenor0,][,2]),
          4))}),
  renderText({
    paste("La probabilidad de que sea mayor que ",
          input$Probmayor0, " es ",
    round(sum(tabl()[tabl()$Valor>input$Probmayor0,][,2]),
          4))}),
  renderText({
    a=strsplit(paste0(input$Probentrececer0,
                      collapse = ", "),"[1]
    b=as.numeric(a[1])
    c=as.numeric(a[2])
    paste("La probabilidad de que esté entre [",
          a[1], ",", a[2], "] es ",
    round(sum(tabl()[tabl()$Valor<=c,][,2])-
          sum(tabl()[tabl()$Valor<=b,][,2]),
          4))}),
  renderText({
    a=strsplit(paste0(input$Probentreabi0,
                      collapse = ", "),"[1]
    b=as.numeric(a[1])
    c=as.numeric(a[2])
    paste("La probabilidad de que esté entre (",
          a[1], ",", a[2], ") es ",
    round(sum(tabl()[tabl()$Valor<c,][,2])-
          sum(tabl()[tabl()$Valor<b,][,2]),
          4))}),
  renderText({
    a=strsplit(paste0(input$Probentreabicer0,
                      collapse = ", "),"[1]
    b=as.numeric(a[1])
    c=as.numeric(a[2])
    paste("La probabilidad de que esté entre [",
          a[1], ",", a[2], ") es ",
    round(sum(tabl()[tabl()$Valor<=c,][,2])-
          sum(tabl()[tabl()$Valor<b,][,2]),
          4))}),
  renderText({
    a=a=strsplit(
      paste0(input$Probentrececerabi0,
            collapse = ", "),"[1]
    b=as.numeric(a[1])
    c=as.numeric(a[2])
    paste("La probabilidad de que esté entre (",
          a[1], ",", a[2], "] es ",
    round(sum(tabl()[tabl()$Valor<c,][,2])-
          sum(tabl()[tabl()$Valor<=b,][,2]),
          4))}),
  renderPlotly({
    p=ggplot(data=tabl(),
             aes(x=Valor,y=Probabilidad)) +
      geom_bar(stat="identity",fill="lightblue",
              alpha=0.8) + theme_minimal() +

```

```

        labs(title="Función de probabilidad",
             x="Valor", y = "Probabilidades")
    }),
    renderPlotly({
        p=ggplot(data=tabl(),
                 aes(x=Valor,
                     y=cumsum(Probabilidad))) +
        geom_bar(stat="identity",fill="lightblue",
                 alpha=0.8) + theme_minimal() +
        labs(title="Función de distribución",
             x="Valor", y = "Probabilidades")
    })
)
} else {
    p("Por favor seleccione un archivo")
}
})
},
function(input, output) {
    tabl<-reactive({
        a=strsplit(
            paste0(input$n0, collapse = ","),", ") [[1]]
        b=as.numeric(a[1])
        d=as.numeric(a[2])
        e=length(c(b:d))
        tabla<-data.frame(
            Valor=c(b:d),
            Probabilidad= round(rep(1/(d-b+1),e),8))
    })
    output$salida1 <-renderUI({
        div(
            renderDataTable({tabl()}),options=list(digits=8)),
            renderText({
                paste("El valor esperado es",
                    round(sum(tabl()[1]*tabl()[2]),4))),
            renderText({
                paste("La varianza es",
                    round(sum((tabl()[1]^2)*tabl()[2])-
                        (sum(tabl()[1]*tabl()[2]))^2,4))),
            renderText({
                paste("El momento de orden ",
                    input$moment1," es ",
                    round(sum((tabl()[1]^input$moment1)*tabl()[2]),
                        4))),
            renderText({
                paste("La probabilidad de que sea igual a ",
                    input$Probigual1," es ",
                    round(sum(tabl()[tabl()$Valor==input$Probigual1],[2]),
                        4))),
            renderText({
                paste("La probabilidad de que sea menor o igual
                    que ",input$Probmenorigual1," es ",
                    round(sum(tabl()[tabl()$Valor<=input$Probmenorigual1],[2]),
                        4))),
            renderText({
                paste("La probabilidad de que sea mayor o igual que ",
                    input$Probmayorigual1," es ",

```

```

round(sum(tabl()[tabl()$Valor>=input$Probmayorigual1,][,2]),
4))}),
  renderText({
    paste("La probabilidad de que sea menor que ",
          input$Probmenor1," es ",
          round(sum(tabl()[tabl()$Valor<input$Probmenor1,][,2]),
                4))}),
    renderText({
      paste("La probabilidad de que sea mayor que ",
            input$Probmayor1," es ",
            round(sum(tabl()[tabl()$Valor>input$Probmayor1,][,2]),
                  4))}),
      renderText({
        a=strsplit(
          paste0(input$Probentrececer1,
                collapse = ", ")," ")[[1]]
        b=as.numeric(a[1])
        c=as.numeric(a[2])
        paste("La probabilidad de que esté entre [",
              a[1],"",a[2],"] es ",
              round(sum(tabl()[tabl()$Valor<=c,][,2])-
                    sum(tabl()[tabl()$Valor<=b,][,2]),4))
      }),
      renderText({
        a=strsplit(
          paste0(input$Probentreabi1,
                collapse = ", ")," ")[[1]]
        b=as.numeric(a[1])
        c=as.numeric(a[2])
        paste("La probabilidad de que esté entre (",
              a[1],"",a[2],") es ",
              round(sum(tabl()[tabl()$Valor<c,][,2])-
                    sum(tabl()[tabl()$Valor<b,][,2]),4))
      }),
      renderText({
        a=strsplit(
          paste0(input$Probentrececerabi1,
                collapse = ", ")," ")[[1]]
        b=as.numeric(a[1])
        c=as.numeric(a[2])
        paste("La probabilidad de que esté entre [",
              a[1],"",a[2],") es ",
              round(sum(tabl()[tabl()$Valor<=c,][,2])-
                    sum(tabl()[tabl()$Valor<b,][,2]),4))
      }),
      renderText({
        a=strsplit(
          paste0(input$Probentreabicer1,
                collapse = ", ")," ")[[1]]
        b=as.numeric(a[1])
        c=as.numeric(a[2])
        paste("La probabilidad de que esté entre (",
              a[1],"",a[2],"] es ",
              round(sum(tabl()[tabl()$Valor<c,][,2])-
                    sum(tabl()[tabl()$Valor<=b,][,2]),4))
      }),
      renderPlotly({

```



```

        p=ggplot(data=tabl(),
                aes(x=Valor,y=Probabilidad)) +
        geom_bar(stat="identity",fill="lightblue",
                alpha=0.8) + theme_minimal() +
        labs(title="Función de probabilidad",
             x="Valor", y = "Probabilidades")
    }),
    renderPlotly({
        p=ggplot(data=tabl(),
                aes(x=Valor,y=cumsum(Probabilidad))) +
        geom_bar(stat="identity",fill="lightblue",
                alpha=0.8) + theme_minimal() +
        labs(title="Función de distribución",
             x="Valor", y = "Probabilidades")
    })
    )
})
},
function(input, output) {
    tabl<-reactive({
        tabla<-data.frame(
            Valor=c(0:1),
            Probabilidad=c(1-input$p0,input$p0))
    })
    output$salida2 <-renderUI({
        div(
            renderDataTable({tabl()}),
            renderText({paste("El valor esperado es",
                               sum(tabl()[1]*tabl()[2]))}),
            renderText({
                paste("La varianza es",
                      sum((tabl()[1]^2)*tabl()[2])-
                          (sum(tabl()[1]*tabl()[2]))^2)},
            renderText({
                paste("El momento de orden ",
                      input$moment2, " es ",
                      sum((tabl()[1]^input$moment2)*tabl()[2]))}),
            renderPlotly({
                p=ggplot(data=tabl(), aes(x=Valor,
                                           y=Probabilidad)) +
                geom_bar(stat="identity",fill="lightblue",
                        alpha=0.8) + theme_minimal() +
                labs(title="Función de probabilidad",
                     x="Valor", y = "Probabilidades")
            }),
            renderPlotly({
                p=ggplot(data=tabl(),
                        aes(x=Valor,y=cumsum(Probabilidad))) +
                geom_bar(stat="identity",fill="lightblue",
                        alpha=0.8) + theme_minimal() +
                labs(title="Función de distribución",
                     x="Valor", y = "Probabilidades")
            })
        })
    })
},
function(input, output) {

```

```

tabl<-reactive({
  Probabilidad=NULL
  for(i in 0:input$n1){
    Probabilidad=c(Probabilidad,
                  dbinom(i,input$n1,input$p1))
  }
  tabla<-data.frame(
    Valor=c(0:input$n1),
    Probabilidad=round(Probabilidad,8))
})
output$salida3 <-renderUI({
  div(
    renderDataTable({tabl()},options=list(digits=8)),
    renderText({
      paste("El valor esperado es",
            round(sum(tabl()[1]*tabl()[2]),4)}),
    renderText({
      paste("La varianza es",
            round(sum((tabl()[1]^2)*tabl()[2])-
                  (sum(tabl()[1]*tabl()[2]))^2,4)}),
    renderText({
      paste("El momento de orden ",
            input$moment3, " es ",
            round(sum((tabl()[1]^input$moment3)*tabl()[2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea igual a ",
            input$Probigual2, " es ",
            round(sum(tabl()[tabl()$Valor==input$Probigual2],[2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea menor o igual que ",
            input$Probmenorigual2, " es ",
            round(sum(tabl()[tabl()$Valor<=input$Probmenorigual2],[2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea mayor o igual que ",
            input$Probmayorigual2, " es ",
            round(sum(tabl()[tabl()$Valor>=input$Probmayorigual2],[2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea menor que ",
            input$Probmenor2, " es ",
            round(sum(tabl()[tabl()$Valor<input$Probmenor2],[2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea mayor que ",
            input$Probmayor2, " es ",
            round(sum(tabl()[tabl()$Valor>input$Probmayor2],[2]),
                  4)}),
    renderText({
      a=strsplit(paste0(input$Probentrececer2,
                       collapse = ", ", " "))[1]
      b=as.numeric(a[1])
      c=as.numeric(a[2])
      paste("La probabilidad de que esté entre [",
            a[1],",",a[2],"] es ",

```

```

        round(sum(tabl()[tabl()$Valor<=c,][,2])-
              sum(tabl()[tabl()$Valor<=b,][,2]),4)
      ),
      renderText({
        a=strsplit(paste0(input$Probentreabi2,
                          collapse = ", "),"")[[1]]
        b=as.numeric(a[1])
        c=as.numeric(a[2])
        paste("La probabilidad de que esté entre (",
              a[1],"",a[2],") es ",
              round(sum(tabl()[tabl()$Valor<c,][,2])-
                    sum(tabl()[tabl()$Valor<b,][,2]),4)
            ),
      ),
      renderText({
        a=strsplit(paste0(input$Probentreacerabi2,
                          collapse = ", "),"")[[1]]
        b=as.numeric(a[1])
        c=as.numeric(a[2])
        paste("La probabilidad de que esté entre [",
              a[1],"",a[2],") es ",
              round(sum(tabl()[tabl()$Valor<=c,][,2])-
                    sum(tabl()[tabl()$Valor<b,][,2]),4)
            ),
      ),
      renderText({
        a=strsplit(paste0(input$Probentreabicer2,
                          collapse = ", "),"")[[1]]
        b=as.numeric(a[1])
        c=as.numeric(a[2])
        paste("La probabilidad de que esté entre (",
              a[1],"",a[2],") es ",
              round(sum(tabl()[tabl()$Valor<c,][,2])-
                    sum(tabl()[tabl()$Valor<=b,][,2]),4)
            ),
      ),
      renderPlotly({
        p=ggplot(data=tabl(), aes(x=Valor,
                                  y=Probabilidad)) +
          geom_bar(stat="identity",fill="lightblue",
                  alpha=0.8) + theme_minimal() +
          labs(title="Función de probabilidad",
               x="Valor", y = "Probabilidades")
      ),
      renderPlotly({
        p=ggplot(data=tabl(),
                  aes(x=Valor,y=cumsum(Probabilidad))) +
          geom_bar(stat="identity",fill="lightblue",
                  alpha=0.8) + theme_minimal() +
          labs(title="Función de distribución",
               x="Valor", y = "Probabilidades")
      )
    )
  )
}
},
function(input, output) {
  tabl<-reactive({
    Probabilidad=NULL
    for(i in 0:input$n2){
      Probabilidad=c(Probabilidad,dpois(i,input$p2))
    }
  })
}

```

```

}
tabla<-data.frame(
  Valor=c(0:input$n2),
  Probabilidad=round(Probabilidad,8))
})
output$salida4 <-renderUI({
  div(
    renderDataTable({tabl()}),
    renderText({
      paste("El valor esperado es",
            round(sum(tabl()[1]*tabl()[2]),4)}),
    renderText({
      paste("La varianza es",
            round(sum((tabl()[1]^2)*tabl()[2])-
                  (sum(tabl()[1]*tabl()[2]))^2,4)}),
    renderText({
      paste("El momento de orden ",
            input$moment4, " es ",
            round(sum((tabl()[1]^input$moment4)*tabl()[2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea igual a ",
            input$Probigual3, " es ",
            round(sum(tabl()[tabl()$Valor==input$Probigual3],[,2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea menor o igual que ",
            input$Probmenorigual3, " es ",
            round(sum(tabl()[tabl()$Valor<=input$Probmenorigual3],[,2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea mayor o igual que ",
            input$Probmayorigual3, " es ",
            round(sum(tabl()[tabl()$Valor>=input$Probmayorigual3],[,2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea menor que ",
            input$Probmenor3, " es ",
            round(sum(tabl()[tabl()$Valor<input$Probmenor3],[,2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea mayor que ",
            input$Probmayor3, " es ",
            round(sum(tabl()[tabl()$Valor>input$Probmayor3],[,2]),
                  4)}),
    renderText({
      a=strsplit(paste0(input$Probentrececer3,
                        collapse = ", ", " "))[1]
      b=as.numeric(a[1])
      c=as.numeric(a[2])
      paste("La probabilidad de que esté entre [",
            a[1],",",a[2],"] es ",
            round(sum(tabl()[tabl()$Valor<=c],[,2])-
                  sum(tabl()[tabl()$Valor<=b],[,2]),4)
            ),
    renderText({
      a=strsplit(paste0(input$Probentreabi3,

```

```

                                collapse = ", ",", ")[[1]]
    b=as.numeric(a[1])
    c=as.numeric(a[2])
    paste("La probabilidad de que esté entre (",
          a[1],",",a[2],") es ",
          round(sum(tabl()[tbl()$Valor<c,][,2])-
                sum(tabl()[tbl()$Valor<b,][,2]),4)
    ),
    renderText({
      a=strsplit(paste0(input$Probentrecerabi3,
                        collapse = ", ",", ")[[1]]
      b=as.numeric(a[1])
      c=as.numeric(a[2])
      paste("La probabilidad de que esté entre [",
            a[1],",",a[2],") es ",
            round(sum(tabl()[tbl()$Valor<=c,][,2])-
                  sum(tabl()[tbl()$Valor<b,][,2]),4)
      ),
      renderText({
        a=strsplit(paste0(input$Probentreabicer3,
                          collapse = ", ",", ")[[1]]
        b=as.numeric(a[1])
        c=as.numeric(a[2])
        paste("La probabilidad de que esté entre (",
              a[1],",",a[2],") es ",
              round(sum(tabl()[tbl()$Valor<c,][,2])-
                    sum(tabl()[tbl()$Valor<=b,][,2]),4)
        ),
        renderPlotly({
          p=ggplot(data=tabl(),
                   aes(x=Valor,y=Probabilidad)) +
            geom_bar(stat="identity",fill="lightblue",
                    alpha=0.8) + theme_minimal() +
            labs(title="Función de probabilidad",
                 x="Valor", y = "Probabilidades")
        ),
        renderPlotly({
          p=ggplot(data=tabl(),
                   aes(x=Valor,y=cumsum(Probabilidad))) +
            geom_bar(stat="identity",fill="lightblue",
                    alpha=0.8) + theme_minimal() +
            labs(title="Función de distribución",
                 x="Valor", y = "Probabilidades")
        )
      )
    )
  },
  function(input, output) {
    tabl<-reactive({
      Probabilidad=NULL
      for(i in 0:input$n3){
        Probabilidad=c(Probabilidad,dgeom(i,input$p3))
      }
      tabla<-data.frame(
        Valor=c(0:input$n3),
        Probabilidad=round(Probabilidad,8)
      )
    })
  }
}

```

```

output$salida5 <-renderUI({
  div(
    renderDataTable({tabl()}),
    renderText({
      paste("El valor esperado es",
            round(sum(tabl()[1]*tabl()[2]),4)}),
    renderText({
      paste("La varianza es",
            round(sum((tabl()[1]^2)*tabl()[2])-
                  (sum(tabl()[1]*tabl()[2]))^2,4)}),
    renderText({
      paste("El momento de orden ",
            input$moment5, " es ",
            round(sum((tabl()[1]^input$moment5)*tabl()[2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea igual a ",
            input$Probigual4, " es ",
            round(sum(tabl()[tabl()$Valor==input$Probigual4,][,2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea menor o igual que ",
            input$Probmenorigual4, " es ",
            round(sum(tabl()[tabl()$Valor<=input$Probmenorigual4,][,2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea mayor o igual que ",
            input$Probmayorigual4, " es ",
            round(sum(tabl()[tabl()$Valor>=input$Probmayorigual4,][,2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea menor que ",
            input$Probmenor4, " es ",
            round(sum(tabl()[tabl()$Valor<input$Probmenor4,][,2]),
                  4)}),
    renderText({
      paste("La probabilidad de que sea mayor que ",
            input$Probmayor4, " es ",
            round(sum(tabl()[tabl()$Valor>input$Probmayor4,][,2]),
                  4)}),
    renderText({
      a=strsplit(paste0(input$Probentrece4,
                        collapse = ", "),"")[[1]]
      b=as.numeric(a[1])
      c=as.numeric(a[2])
      paste("La probabilidad de que esté entre [",
            a[1],",",a[2],"] es ",
            round(sum(tabl()[tabl()$Valor<=c,][,2])-
                  sum(tabl()[tabl()$Valor<=b,][,2]),4)
            ),
    renderText({
      a=strsplit(paste0(input$Probentreabi4,
                        collapse = ", "),"")[[1]]
      b=as.numeric(a[1])
      c=as.numeric(a[2])
      paste("La probabilidad de que esté entre (",
            a[1],",",a[2],") es ",

```

```

        round(sum(tabl()[tabl()$Valor<c,][,2])-
              sum(tabl()[tabl()$Valor<b,][,2]),4)
      ),
      renderText({
        a=strsplit(paste0(input$Probentrececerabi4,
                          collapse = ", "),"")[[1]]
        b=as.numeric(a[1])
        c=as.numeric(a[2])
        paste("La probabilidad de que esté entre [",
              a[1],"",a[2],"") es ",
              round(sum(tabl()[tabl()$Valor<=c,][,2])-
                    sum(tabl()[tabl()$Valor<b,][,2]),4)
        ),
      renderText({
        a=strsplit(paste0(input$Probentreabicer4,
                          collapse = ", "),"")[[1]]
        b=as.numeric(a[1])
        c=as.numeric(a[2])
        paste("La probabilidad de que esté entre (",
              a[1],"",a[2],"") es ",
              round(sum(tabl()[tabl()$Valor<c,][,2])-
                    sum(tabl()[tabl()$Valor<=b,][,2]),4)
        ),
      renderPlotly({
        p=ggplot(data=tabl(),
                 aes(x=Valor,y=Probabilidad)) +
          geom_bar(stat="identity",fill="lightblue",
                  alpha=0.8) + theme_minimal() +
          labs(title="Función de probabilidad",
               x="Valor", y = "Probabilidades")
      },
      renderPlotly({
        p=ggplot(data=tabl(),
                 aes(x=Valor,y=cumsum(Probabilidad))) +
          geom_bar(stat="identity",fill="lightblue",
                  alpha=0.8) + theme_minimal() +
          labs(title="Función de distribución",
               x="Valor", y = "Probabilidades")
      })
    )
  })
},
function(input, output) {
  tabl<-reactive({
    Probabilidad=NULL
    for(i in 0:input$n4){
      Probabilidad=c(Probabilidad,
                    dhyper(i,input$n5,
                           input$n6-input$n5,
                           input$n4))
    }
    tabla<-data.frame(
      Valor=c(0:input$n4),
      Probabilidad=round(Probabilidad,8))
  })
  output$salida6 <-renderUI({
    div(

```

```

renderDataTable({tabl()}),
renderText({
  paste("El valor esperado es",
        round(sum(tabl()[1]*tabl()[2]),4)}),
renderText({
  paste("La varianza es",
        round(sum((tabl()[1]^2)*tabl()[2])-
                (sum(tabl()[1]*tabl()[2]))^2,4)}),
renderText({
  paste("El momento de orden ",
        input$moment6, " es ",
        round(sum((tabl()[1]^input$moment6)*tabl()[2]),
                4)}),
renderText({
  paste("La probabilidad de que sea igual a ",
        input$Probigual5, " es ",
        round(sum(tabl()[tabl()$Valor==input$Probigual5],[,2]),
                4)}),
  renderText({
    paste("La probabilidad de que sea menor o igual que ",
          input$Probmenorigual5, " es ",
          round(sum(tabl()[tabl()$Valor<=input$Probmenorigual5],[,2]),
                  4)}),
  renderText({
    paste("La probabilidad de que sea mayor o igual que ",
          input$Probmayorigual5, " es ",
          round(sum(tabl()[tabl()$Valor>=input$Probmayorigual5],[,2]),
                  4)}),
  renderText({
    paste("La probabilidad de que sea menor que ",
          input$Probmenor5, " es ",
          round(sum(tabl()[tabl()$Valor<input$Probmenor5],[,2]),
                  4)}),
  renderText({
    paste("La probabilidad de que sea mayor que ",
          input$Probmayor5, " es ",
          round(sum(tabl()[tabl()$Valor>input$Probmayor5],[,2]),
                  4)}),
  renderText({
    a=strsplit(paste0(input$Probentrececer5,
                      collapse = ", "),"[1]")
    b=as.numeric(a[1])
    c=as.numeric(a[2])
    paste("La probabilidad de que esté entre [",
          a[1],",",a[2],"] es ",
          round(sum(tabl()[tabl()$Valor<=c],[,2])-
                sum(tabl()[tabl()$Valor<=b],[,2]),4)
    )}),
  renderText({
    a=strsplit(paste0(input$Probentreabi5,
                      collapse = ", "),"[1]")
    b=as.numeric(a[1])
    c=as.numeric(a[2])
    paste("La probabilidad de que esté entre (",
          a[1],",",a[2],") es ",
          round(sum(tabl()[tabl()$Valor<c],[,2])-
                sum(tabl()[tabl()$Valor<b],[,2]),4)
    )

```



```

    }),
    renderText({
      a=strsplit(paste0(input$Probentrecerabi5,
                       collapse = ", "),", ")[[1]]
      b=as.numeric(a[1])
      c=as.numeric(a[2])
      paste("La probabilidad de que esté entre [",
            a[1],",",a[2],"] es ",
            round(sum(tabl()[tabl()$Valor<=c,][,2])-
                  sum(tabl()[tabl()$Valor<b,][,2]),4))
    }),
    renderText({
      a=strsplit(paste0(input$Probentrecerabi5,
                       collapse = ", "),", ")[[1]]
      b=as.numeric(a[1])
      c=as.numeric(a[2])
      paste("La probabilidad de que esté entre (",
            a[1],",",a[2],"] es ",
            round(sum(tabl()[tabl()$Valor<c,][,2])-
                  sum(tabl()[tabl()$Valor<=b,][,2]),4))
    }),
    renderPlotly({
      p=ggplot(data=tabl(),
               aes(x=Valor,y=Probabilidad)) +
        geom_bar(stat="identity",fill="lightblue",
                 alpha=0.8) + theme_minimal() +
        labs(title="Función de probabilidad",
              x="Valor", y = "Probabilidades")
    }),
    renderPlotly({
      p=ggplot(data=tabl(),
               aes(x=Valor,y=cumsum(Probabilidad))) +
        geom_bar(stat="identity",fill="lightblue",
                 alpha=0.8) + theme_minimal() +
        labs(title="Función de distribución",
              x="Valor", y = "Probabilidades")
    })
  )
}
)
shinyApp(ui,server)
}

```

B.1.2.4. vect.R

```

library(shinyscreenshot)
runVec <- function(n){
  ui <- switch (n,
    fluidPage(
      style = "overflow: auto; height: calc(100vh - 50px) !important;",
      sidebarLayout(
        sidebarPanel(
          tags$head(tags$style(
            HTML(
              "#foto{position:fixed; top:450px; margin-left:110px}"))),
          h4("Parámetros de entrada"),

```

```

    helpText("Cargue un archivo '.xlsx'. Dejando en blanco la
casilla A1, coloque los valores de la variable 'X' en la
primera columna, coloque los valores de la variable 'Y' en la
primera fila y las probabilidades conjuntas en sus respectivas
casillas, sin encabezado"),
    fileInput("arch0", "Selecciona el excel", accept=".xlsx"),
    numericInput("moment0", "Orden X:", 2, 0, Inf),
    numericInput("moment1", "Orden Y:", 2, 0, Inf),
    div(id="foto", screenshotButton(label="cap"))
  ),
  mainPanel(
    withMathJax(),
    tags$script(HTML("MathJax.Hub.Config({
      tex2jax: {inlineMath: [['$','$']]
    });")),
    uiOutput("salida")
  )
)
)
)
)
server <- switch (
  n,
  function(input,output){
    path <- reactive({input$arch0})
    tabl<-reactive({
      file <- path()
      ext <- tools::file_ext(file$datapath)
      req(file)
      validate(need(ext == "xlsx",
                    "Por favor cargue un archivo xlsx"))
      tabla<-read_excel(file$datapath)
      colnames(tabla)[1]<-"X/Y"
      tabla<-as.matrix(tabla)
    })
    marginx <- reactive({
      d=matrix(data=NA,nrow = dim(tabl())[1],ncol = 2)
      colnames(d)<-c("X", "P[X=x]")
      d[,1]<-tabl()[,1]
      d[,2]<-apply(tabl()[,-1],1,sum)
      d
    })
    marginy <- reactive({
      d=matrix(data=NA,
              nrow = dim(tabl())[2]-1,ncol = 2)
      colnames(d)<-c("Y", "P[Y=y]")
      d[,1]<-as.numeric(colnames(tabl())[-1])
      d[,2]<-as.numeric(apply(tabl()[,-1],2,sum))
      d
    })
    indep <- reactive({
      a=NULL
      for(i in 1:dim(tabl())[1])
        for(j in 2:dim(tabl())[2]){
          a=c(a,as.numeric(tabl()[i,j]))
        }
      b=NULL
      for(i in 1:dim(marginx())[1]){

```

```

    b=c(b,marginx()[i,2]*marginx()[,2])
  }
  bmatr=matrix(b,nrow = dim(tabl())[1],
              ncol = dim(tabl())[2]-1)
  espxy=(marginx()[,1]%%t(marginx()[,1]))%%t(tabl()[,-1])
  espxy_rs=sum(
    rowSums(
      (marginx()[,1]^(input$moment0)%%
        t(marginx()[,1]^(input$moment1)))*tabl()[,-1]))
  espxy_muxy_rs=sum(
    rowSums(
      ((marginx()[,1]-
        round(sum(marginx()[,1]*marginx()[,2]),4))^
        (input$moment0)%%
        t((marginx()[,1]-round(sum(marginx()[,1]*marginx()[,2]),4)
          )^(input$moment1)))*tabl()[,-1]))
  covxy=round(
    sum(diag(espxy)),4)-
    (round(sum(marginx()[,1]*marginx()[,2]),4)*
      round(sum(marginx()[,1]*marginx()[,2]),4))
  varx=round(
    sum(marginx()[,1]^2*marginx()[,2])-
    sum(marginx()[,1]*marginx()[,2])^2,4)
  vary=round(
    sum(marginx()[,1]^2*marginx()[,2])-
    sum(marginx()[,1]*marginx()[,2])^2,4)
  if(any(a!=b)){
    e=as.numeric(which(bmatr!=tabl()[,-1],arr.ind = TRUE)[1,1])
    f=as.numeric(which(bmatr!=tabl()[,-1],arr.ind = TRUE)[1,2])
    div(
      p("Las variables no son independientes porque",
        withMathJax("$$P[X=" ,marginx()[e,1] ,"; Y=" ,
          marginx()[f,1] ,"]=" ,
          tabl()[e,(f+1)] ,"\neq P[X=" ,marginx()[e,1] ,
            "]\cdot P[Y=" ,marginx()[f,1] ,"]=" ,
            bmatr[e,f] ,"$")),
      p(withMathJax("$E[XY]=" , round(sum(diag(espxy)),4) , "$")),
      p(withMathJax("$E[X^" ,input$moment0 , "Y^" ,input$moment1 ,"]=" ,
        round(espxy_rs,4) , "$")),
      p(withMathJax("$E[(X-\mu_x)^" ,input$moment0 , "(Y-\mu_Y)^" ,
        input$moment1 ,"]=" , round(espxy_muxy_rs,4) ,
        "$")),
      p(withMathJax("$Cov(X,Y)=" , covxy , "$")),
      p(withMathJax(
        "$\rho = \frac{Cov(X,Y)}{\sqrt{Var(X)}\cdot \sqrt{Var(Y)}}=" ,
        round(covxy/(sqrt(varx)*sqrt(vary)),4) , "$")),
    )
  }
  else{
    div(
      p("Las variables son independientes porque el producto de
        las marginales coincide con la conjunta $\forall i,j$"),
      p(withMathJax("$E[XY]=" , round(sum(diag(espxy)),4) , "$")),
      p(withMathJax("$E[X^" ,input$moment0 , "Y" ,input$moment1 ,"]=" ,
        round(espxy_rs,4) , "$")),
      p(withMathJax("$E[(X-\mu_x)^" ,input$moment0 , "(Y-\mu_Y)^" ,
        input$moment1 ,"]=" ,

```

```

        round(sum(diag(espxy_muxy_rs)),4), "$")),
  p(withMathJax("$Cov(X,Y)=", covxy, "$")),
  p(withMathJax(
    "$\rho = \frac{Cov(X,Y)}{\sqrt{Var(X)}\cdot \sqrt{Var(Y)}}=",
    round(covxy/(sqrt(varx)*sqrt(vary)),4), "$")),
  )
}
})
output$salida <-renderUI({
  if(!is.null(path())){
    tagList(
      div(
        p("La funcion de probabilidad conjunta de (X,Y) es"),
        renderDataTable({tabl()}),
        # p(withMathJax("La funcion de probabilidad marginal de X es
        $$$\begin{array}{|c|c|} \hline X & \\ \hline P[X=x] & \\ \hline \end{array}$$$" ),
        #round(sum(tabl()[1,-1]),2), "&", round(sum(tabl()[2,-1]),2), "
        #\begin{array}{|c|c|} \hline P[X=x] & \\ \hline \end{array}$$$" ))
        p("La funcion de probabilidad marginal de X es"),
        renderDataTable({marginx()}),
        p(withMathJax("$E[X]=",
          round(sum(marginx()[,1]*marginx()[,2]),4),"$")),
        p(withMathJax("$E[X^",input$moment0,"]=",
          round(
            sum(marginx()[,1]^input$moment0*marginx()[,2]),
            4),"$")),
        p(withMathJax("$E[X-\mu_X]^",input$moment0,"=",
          round(sum((marginx()[,1]-
            (round(sum(marginx()[,1]*marginx()[,2]),
              4)))^input$moment0*marginx()[,2]),4),
          "$")),
        p(withMathJax("$Var[X]=",
          round(sum(marginx()[,1]^2*marginx()[,2])-
            sum(marginx()[,1]*marginx()[,2])^2,4),"$")),
        p(withMathJax("$\sigma_X=",
          round(sqrt(round(sum(
            marginx()[,1]^2*marginx()[,2])-
            sum(marginx()[,1]*marginx()[,2])^2,4)),4),"$")),
        p("La funcion de probabilidad marginal de Y es"),
        renderDataTable({marginy()}),
        p(withMathJax("$E[Y]=",
          round(sum(marginy()[,1]*marginy()[,2]),4),"$")),
        p(withMathJax("$E[Y^",input$moment1,"]=",
          round(sum(marginy()[,1]^input$moment1*marginy()[,2]),
            4),"$")),
        p(withMathJax("$E[(Y-\mu_Y)^",input$moment1,"=",
          round(sum((marginy()[,1]-
            (round(sum(marginy()[,1]*marginy()[,2]),
              4)))^input$moment1*marginy()[,2]),4),
          "$")),
        p(withMathJax("$Var[Y]=",
          round(sum(marginy()[,1]^2*marginy()[,2])-
            sum(marginy()[,1]*marginy()[,2])^2,4),"$")),
        p(withMathJax("$\sigma_Y=",
          round(sqrt(
            round(sum(marginy()[,1]^2*marginy()[,2])-

```

```

sum(marginy()[,1]*marginy()[,2]^2,4),4),
"$")),
  p(indep())
# renderPlotly({
#   p=ggplot(data=tabl(), aes(x=Valor,y=Probabilidad)) +
#     geom_bar(stat="identity",fill="lightblue",alpha=0.8) +
#     theme_minimal() +
#     labs(title="Función de probabilidad", x="Valor",
#          y = "Probabilidades")
#   }),
# renderPlotly({
#   p=ggplot(data=tabl(), aes(x=Valor,y=cumsum(Probabilidad))) +
#     geom_bar(stat="identity",fill="lightblue",alpha=0.8) +
#     theme_minimal() +
#     labs(title="Función de distribución", x="Valor",
#          y = "Probabilidades")
#   })
  ))
} else {
  p("Por favor seleccione un archivo")
}
})
)
shinyApp(ui,server)
}

```

B.1.3. ME

Esta sección se corresponde con la carpeta MR de la aplicación estadística, la cual contiene un archivo “app.R”

B.1.3.1. app.R

```

library(sampling)
library(openxlsx)
library(tibble)
# library(survey) #Para diseños muestrales complejos
ui <- fluidPage(
  tags$head(tags$style(HTML("#foto{position:fixed; top:510px;}"))),
  titlePanel("Muestreo estadístico"),
  withMathJax(),
  tags$script(HTML("MathJax.Hub.Config({
tex2jax: {inlineMath: [['$','$']]})");)),
  navlistPanel(
    "Tipos de muestreo",
    tabPanel(
      withMathJax("$MA(M;Pr(\\cdot))$"),
      checkboxInput("obt_muestra", "Obtener una muestra",
                    value = FALSE, width = NULL),
      conditionalPanel(
        condition = "input.obt_muestra==1",
        fluidRow(
          column(6,selectInput("select_muestra",
                                h4("Selecciona como obtener la muestra"),
                                choices = list("Obtener muestra aleatoria=1",
                                              "Obtener muestra de un archivo=2"),
                                selected = "Obtener muestra aleatoria")),
          column(6, numericInput("Tamn", "Tamaño n", 5, 0, Inf)),
          checkboxInput("semil", "Establecer una semilla", value = FALSE,
                        width = NULL),
          conditionalPanel(
            condition = "input.semil==1",

```

Esta sección se corresponde con la carpeta AP de la aplicación estadística, la cual contiene un archivo “app.R”

B.1.4.1. app.R

```

library(car)
library(DescTools)
library(KScorrect)
library(PMCMRplus)
library(BSDA)
library(dplyr)
ui <- fluidPage(
  withMathJax(),
  tags$script(HTML("MathJax.Hub.Config({
    tex2jax: {inlineMath: [['$','$']]
  });")),
  tags$head(
    tags$style(
      HTML("#foto{position:fixed; top:105px; margin-left:485px}")),
  tabsetPanel(
    type = "tabs",
    tabPanel(
      "Una muestra",
      fluidRow(
        conditionalPanel(
          condition = "input.oper!=3",
          column(6,
            helpText("Cargue un archivo '.xlsx' con los
              valores de cada individuo en la primera
              columna y en cada una de las filas"),
            fileInput("arch", "Selecciona el excel", accept=".xlsx"),
            div(id="foto",screenshotButton(label="cap"))
          ),
          column(6,uiOutput("salidatabla"))),
        conditionalPanel(
          condition = "input.oper==3",
          column(6, helpText("Cargue un archivo '.xlsx' con los
              valores de cada individuo en la primera
              columna y la proporción esperada en la
              segunda columna"),
            fileInput("arch0", "Selecciona el excel",
              accept=".xlsx"),
            div(id="foto",screenshotButton(label="cap"))),
          column(6,uiOutput("salidatabla0"))
        ),
      # fluidRow(actionButton("imp","imprimir")),
      fluidRow(
        column(4, selectInput("oper",
          h4("Selecciona el tipo de operación"),
          choices = list(
            "Test de Shapiro-Wilk"=1,
            "Test de Student's"=2,
            "Test chi-cuadrado de bondad de ajuste"=3,
            "Test de kolmogorov-Smirnov de bondad de ajuste"=4,
            "Test de Wilcoxon"=5,
            "Test de los signos"=6,
            "Test binomial"=7),
          selected = "Shapiro-Wilk")),
        conditionalPanel(
          condition = "input.oper==2 || input.oper==5 || input.oper==6",
          column(3, textInput("val","Introduce el valor a comparar",

```

```

        value = "0",
        placeholder = "Usa el punto como separador
        decimal por ejemplo 1.7"))
    ),
    conditionalPanel(
      condition = "input.oper!=3 && input.oper!=7",
      column(5, uiOutput("salida0"))
    ),
    conditionalPanel(
      condition = "input.oper==3",
      column(6, uiOutput("salida1"))
    )
  ),
  conditionalPanel(
    condition = "input.oper==1",
    fluidRow(
      column(6, "$\\begin{cases}H_0: X\\sim \\text{Normal} \\|\\|\\|\\|
        H_1: X\\not\\sim \\text{Normal}\\|\\end{cases} \\|\\|\\|\\|")
    )
  ),
  conditionalPanel(
    condition = "input.oper==2",
    fluidRow(
      column(6, "$\\begin{cases}H_0: \\mu_X= \\beta_0\\|\\|\\|\\|
        H_1: \\mu_X\\neq \\beta_0 \\|\\end{cases}$")
    ),
    fluidRow(
      column(4, checkboxInput("contuni",
        "Realizar contraste unilateral")),
      conditionalPanel(
        condition = "input.contuni==1",
        column(4,
          radioButtons("ttestuni",
            "Selecciona el tipo de contraste unilateral",
            choices=list("$H_0:\\mu_X\\geq \\beta_0$=1,
              "$H_0:\\mu_X\\leq \\beta_0$=2))),
        column(4, uiOutput("salida2")))
      )
    ),
  conditionalPanel(
    condition = "input.oper==3",
    fluidRow(
      column(6, radioButtons("tipo", "Tipo de contraste",
        choices = list(
"$\\begin{cases}H_0: X\\sim \\text{Multinomial} \\|\\|\\|\\|
H_1: X\\not\\sim \\text{Multinomial}\\|\\end{cases} \\|\\|\\|\\|=$=1,
"$\\begin{cases}H_0: X\\sim \\text{Poisson}(\\lambda) \\|; \\forall
\\lambda \\in (0, \\infty)\\|\\|\\|\\|
H_1: X\\not\\sim \\text{Poisson}(\\lambda)\\|\\end{cases}$=2)"))
    )
  )
  ),
  conditionalPanel(
    condition = "input.oper==4",
    fluidRow(
      column(6, "$\\begin{cases}H_0: X\\sim \\text{Exp}(\\lambda) \\|\\|\\|\\|
        H_1: X\\not\\sim \\text{Exp}(\\lambda)\\|\\end{cases}$",
        checkboxInput("lilliefor",

```

```

                                "Aplicar corrección de Lilliefors"))
    )
  ),
  conditionalPanel(
    condition = "input.oper==5",
    fluidRow(
      column(6, "$\\begin{cases}H_0: m_X= m_0 \\text{ y distribución simétrica} \\ \\ H_1: m_X\\neq m_0 \\text{ o distribución no simétrica}\\end{cases} \\ \\ \\ \\ [0.5cm] \\text{Si alguna observación coincide con } \\ \\ ; m_0\\ \\ ; \\text{se elimina del estudio}$")
    )
  ),
  conditionalPanel(
    condition = "input.oper==6",
    fluidRow(
      column(6, "$\\begin{cases}H_0: m_X= m_0 \\ \\ \\ H_1: m_X\\neq m_0 \\ \\ \\ \\end{cases} \\ \\ \\ [0.5cm] \\text{La variable } X \\ \\ \\text{ tiene que ser continua en un entorno de la mediana} \\ \\ \\ [0.5cm] P(X>m)=P(X<m)=\\frac{1}{2}\\ \\ ; \\ \\quad P(X=m)=0 \\ \\ \\ [0.5cm] \\text{Si alguna observación coincide con } \\ \\ ; m_0\\ \\ ; \\text{ se elimina del estudio}$")
    ),
    fluidRow(
      column(4,
        checkboxInput("contuni4",
          "Realizar contraste unilateral")),
      conditionalPanel(
        condition = "input.contuni4==1",
        column(4,
          radioButtons("signouni",
            "Selecciona el tipo de contraste unilateral",
            choices=list("$H_0: m_X\\geq m_0$=1",
              "$H_0: m_X\\leq m_0$=2))),
        column(4, uiOutput("salidasignouni"))))
    ),
  conditionalPanel(
    condition = "input.oper==7",
    fluidRow(
      column(4, "$\\begin{cases}H_0: p_X= p_0 \\ \\ \\ H_1: p_X\\neq p_0 \\ \\ \\ \\end{cases}$",
        radioButtons("forma", "Realizar el contraste",
          choices = list(
            "Suponiendo que un porcentaje de los datos sea menor a un cierto valor "=1,
            "Introduciendo el número de éxitos, el tamaño muestral y la probabilidad de éxito "=2))),
      conditionalPanel(
        condition = "input.forma==1",
        column(4, textInput("prob", "Introduce el porcentaje",
          placeholder = "Usa el punto como separador decimal, como por ejemplo, 0.7"),
          textInput("val0", "Introduce el valor a comparar",
            placeholder = "Usa el punto como separador decimal, como por ejemplo, 1.7"),
          column(4, uiOutput("salidabinom1"))
      )
    )
  )
)

```



```

),
conditionalPanel(
  condition = "input.forma==2",
  column(4, textInput("totexit",
    "Introduce el total de acierto"),
    textInput("tamn", "Introduce el tamaño muestral"),
    textInput("prob0", "Introduce el porcentaje",
      placeholder = "Usa el punto como separador
        decimal, como por ejemplo, 0.7")),
  column(4, uiOutput("salidabinom2"))
)
),
fluidRow(
  column(4, checkboxInput("contuni3",
    "Realizar contraste unilateral")),
  conditionalPanel(
    condition = "input.contuni3==1",
    column(4,
      radioButtons("binomuni",
        "Selecciona el tipo de contraste unilateral",
        choices=list("$H_0:p_X\\geq p_0$=1,
          "$H_0:p_X\\leq p_0$=2")),
      column(4, uiOutput("salida8")))
  ),
  fluidRow(
    column(10, "$\\begin{array}{l}\\\\
      \\text{Cómo regla para decidir si se rechaza o no
        una hipótesis nula se tendrá en cuenta:}
      \\\\end{array} \\\\")
  ),
  fluidRow(
    column(10, tags$ul(tags$li("$\\text{Si el estudio se realiza con
      un nivel de significación del 10% y se obtiene un p-valor
      inferior al 0.1 se rechaza la hipótesis nula,}\\\\
      \\text{en caso contrario, se dice que no hay evidencias en contra para rechazar
      la hipótesis nula o que los datos no aportan suficiente}\\\\
      \\text{evidencia para dudar de su validez.}$"))
  ),
  fluidRow(
    column(10, tags$ul(tags$li("$\\text{Si el estudio se realiza con
      un nivel de significación del 5% y se obtiene un p-valor inferior
      al 0.05 se rechaza la hipótesis nula, }\\\\
      \\text{en caso contrario, se dice que no hay evidencias en contra para rechazar
      la hipótesis nula o que los datos no aportan suficiente}\\\\
      \\text{evidencia para dudar de su validez}$"))
  )
),
tabPanel(
  "Dos muestras independientes",
  fluidRow(
    column(6,
      helpText("Introduce los valores de cada variable en cada
        columna con encabezados"),
      fileInput("arch1", "Selecciona el excel", accept=".xlsx"),
      div(id="foto", screenshotButton(label="cap")),
      column(6, uiOutput("salidatabla1"))
    ),

```

```

fluidRow(
  column(4, selectInput("oper0",
    h4("Selecciona el tipo de operación"),
    choices = list(
      "Test de Shapiro-Wilk"=1,
      "Test de Levene"=2,
      "Test de Student's"=3,
      "Test de Fisher"=4,
      "Test de kolmogorov-Smirnov de homogeneidad"=5,
      "Test de Wilcoxon-Mann-Whitney"=6,
      "Test de Siegel-Tukey"=7,
      "Test chi-cuadrado de independencia"=8),
    selected = "Shapiro-Wilk")),
  conditionalPanel(
    condition = "input.oper0==4",
    column(2, checkboxInput("discret",
      "Los datos son continuos")),
    conditionalPanel(
      condition = "input.discret==1",
      column(3,
        textInput("val1",
          "Introduce un valor tal que se le
          asignará 1 si el valor muestral es
          mayor que el valor indicado")))
  ),
  column(3, uiOutput("salida3"))
),
conditionalPanel(
  condition = "input.oper0==1",
  fluidRow(
    column(6, "$\\begin{cases}H_0: X\\sim \\text{Normal} \\\\
      H_1: X\\not\\sim \\text{Normal}\\end{cases} \\\\
      \\begin{cases}H_0: Y\\sim \\text{Normal} \\\\
      H_1: Y\\not\\sim \\text{Normal}\\end{cases}$")
  )
),
conditionalPanel(
  condition = "input.oper0==2",
  fluidRow(
    column(6, "$\\begin{cases}H_0: \\sigma_X^2 = \\sigma_Y^2 \\\\
      \\\\ H_1: \\sigma_X^2 \\neq \\sigma_Y^2 \\\\
      \\end{cases}$")
  )
),
conditionalPanel(
  condition = "input.oper0==3",
  fluidRow(
    column(6, "$\\begin{cases}H_0: \\mu_X = \\mu_Y \\\\
      \\\\ H_1: \\mu_X \\neq \\mu_Y\\end{cases}$")
  )
),
fluidRow(
  column(4, checkboxInput("contuni0",
    "Realizar contraste unilateral")),
  conditionalPanel(
    condition = "input.contuni0==1",
    column(4, radioButtons("ttestuni2mu",

```

```

        "Selecciona el tipo de contraste unilateral",
        choices=list("$H_0: \\mu_X \\geq \\mu_Y$=1,
                    "$H_0: \\mu_X \\leq \\mu_Y$=2))),
        column(4, uiOutput("salida4"))))
    ),
    conditionalPanel(
        condition = "input.oper0==4",
        fluidRow(
            column(6, "$\\begin{cases}H_0: p_X = p_Y \\ \\ \\ H_1: p_X \\ \\ \\ \\neq p_Y\\end{cases}$")
        )
    ),
    conditionalPanel(
        condition = "input.oper0==5",
        fluidRow(
            column(6, "$\\begin{cases}H_0: F_X(x) = F_Y(x) \\ \\ \\ \\forall x \\ \\ \\ H_1: F_X(x) \\ \\neq F_Y(x) \\ \\text{ para alg\\u00fan } x\\end{cases}$",
                checkboxInput("asintotic",
                    "Obtener p-valor asint\u00f3tico"))
        )
    ),
    conditionalPanel(
        condition = "input.oper0==6",
        fluidRow(
            column(6, "$\\begin{cases}H_0: m_X = m_Y \\ \\ \\ \\ H_1: m_X \\ \\neq m_Y \\ \\end{cases}$")
        ),
        fluidRow(
            column(4, checkboxInput("contuni1",
                "Realizar contraste unilateral")),
            conditionalPanel(
                condition = "input.contuni1==1",
                column(4, radioButtons("wiluni",
                    "Selecciona el tipo de contraste unilateral",
                    choices=list("$H_0: m_X \\geq m_Y$=1,
                                "$H_0: m_X \\leq m_Y$=2))),
                column(4, uiOutput("salida5"))))
        ),
    conditionalPanel(
        condition = "input.oper0==7",
        fluidRow(
            column(4, "$\\begin{cases}H_0: \\ \\text{Medidas de dispersi\u00f3n iguales} \\ \\ \\ \\ H_1: \\ \\text{Medidas de dispersi\u00f3n diferentes} \\ \\ \\ \\end{cases}$",
                checkboxInput("contuni2",
                    "Realizar contraste unilateral")),
            conditionalPanel(
                condition = "input.contuni2==1",
                column(4, radioButtons("sieguni",
                    "Selecciona el tipo de contraste unilateral",
                    choices=list("$H_0: m_X \\geq m_Y$=1,
                                "$H_0: m_X \\leq m_Y$=2))),
                column(4, uiOutput("salida6"))
            )
        ),
    ),

```

```

fluidRow(
  column(10,"$\begin{array}{l}\\\\
    \\text{Cómo regla para decidir si se rechaza o no
    una hipótesis nula se tendrá en cuenta:}
    \\end{array} \\\\")
),
fluidRow(
  column(10,tags$ul(tags$li("$\\text{Si el estudio se realiza
con un nivel de significación del 10% y se obtiene un p-valor
inferior al 0.1 se rechaza la hipótesis nula,}\\\\ \\text{en
caso contrario, se dice que no hay evidencias en contra para
rechazar la hipótesis nula o que los datos no aportan
suficiente}\\\\ \\text{evidencia para dudar de su
validez.}$"))
),
fluidRow(
  column(10,tags$ul(tags$li("$\\text{Si el estudio se realiza
con un nivel de significación del 5% y se obtiene un p-valor
inferior al 0.05 se rechaza la hipótesis nula, }\\\\
\\text{en caso contrario, se dice que no hay evidencias en
contra para rechazar la hipótesis nula o que los datos no
aportan suficiente}\\\\ \\text{evidencia para dudar de su
validez.}$"))
)
),
tabPanel(
  "Dos muestras relacionadas",
  fluidRow(
    column(6,
      helpText("Introduce los valores de cada variable
en cada columna con encabezados"),
      fileInput("arch3", "Selecciona el excel",
        accept=".xlsx"),
      div(id="foto",screenshotButton(label="cap"))),
    column(6, uiOutput("salidatabla3")),
  fluidRow(
    column(4, selectInput("oper2",
      h4("Selecciona el tipo de operación"),
      choices = list("Test de McNemar=1))),
    column(4,textInput("val3","Introduce un valor tal que se le
asignará 1 si el valor muestral es menor
que el valor indicado")),
    column(4,uiOutput("salida9"))
  ),
  fluidRow(
    column(10,"$\begin{array}{l}\\\\
    \\text{Cómo regla para
    decidir si se rechaza o no una hipótesis nula se
    tendrá en cuenta:} \\end{array} \\\\")
  ),
  fluidRow(
    column(10,tags$ul(tags$li("$\\text{Si el estudio se realiza
con un nivel de significación del 10% y se obtiene un p-valor
inferior al 0.1 se rechaza la hipótesis nula,}\\\\
\\text{en caso contrario, se dice que no hay evidencias en
contra para rechazar la hipótesis nula o que los datos no
aportan suficiente}\\\\ \\text{evidencia para dudar de su
validez.}$"))
  )
)

```

```

    ),
    fluidRow(
      column(10, tags$ul(tags$li("$\\text{Si el estudio se realiza
con un nivel de significación del 5% y se obtiene un p-valor
inferior al 0.05 se rechaza la hipótesis nula, }\\\\
\\text{en caso contrario, se dice que no hay evidencias en
contra para rechazar la hipótesis nula o que los datos no
aportan suficiente}\\\\ \\text{evidencia para dudar de su
validez.}$"))
    )
  ),
  tabPanel(
    "K muestras",
    fluidRow(
      column(6,
        helpText("Introduce los valores de cada variable en
cada columna con encabezados"),
        fileInput("arch2", "Selecciona el excel", accept=".xlsx"),
        div(id="foto", screenshotButton(label="cap"))),
      column(6, uiOutput("salidatabla2")),
      fluidRow(
        column(3, selectInput("oper1",
          h4("Selecciona el tipo de operación"),
          choices = list(
            "Test chi-cuadrado de homogeneidad"=1,
            "Test de Kruskal-Wallis (muestras independientes)"=2,
            "Test de Friedman (muestras dependientes)"=3,
            selected = "Test chi-cuadrado de homogeneidad")),
        conditionalPanel(
          condition = "input.oper1==1",
          column(2, checkboxInput("discret0", "Los datos son continuos")),
          conditionalPanel(
            condition = "input.discret0==1",
            column(3, textInput("val2", "Introduce un valor tal que se le
asignará 1 si el valor muestral es mayor
que el valor indicado"))
          )
        ),
        conditionalPanel(
          condition = "input.oper1==2",
          column(3,
            checkboxInput("comp",
              "Comparaciones múltiples
de Kruskal-Nemenyi"))
          ),
        conditionalPanel(
          condition = "input.oper1==3",
          column(3, checkboxInput("comp0",
            "Comparaciones múltiples
de Friedman-Nemenyi"))
          ),
        column(4, uiOutput("salida7"))
      ),
      fluidRow(
        conditionalPanel(
          condition = "input.oper1==1",
          column(12, "$\\begin{cases}H_0: p_1 = \\dots = p_k \\ \\ H_1:"

```



```

)

server <- function(input,output){
  path <- reactive({input$arch})
  tabl <-reactive({
    file <- path()
    ext <- tools::file_ext(file$datapath)
    req(file)
    validate(need(ext == "xlsx", "Por favor cargue un archivo xlsx"))
    tabla<-read_excel(file$datapath,col_types = c("numeric"))
  })
  # observeEvent(input$imp,
  #               screenshot(
  #                 scale = 1,
  #                 filename = "prueba"
  #               ))
  output$salidatabla <- renderUI({
    if(!is.null(path())){
      if(input$oper!=3){
        renderDataTable({tabl()})
      }
    } else {
      p("Por favor seleccione un archivo")
    }
  })
  normal <- reactive({
    datosmuestrales=tabl()[[1]]
    shapiro.test(datosmuestrales)
  })
  ttestbi<- reactive({
    datosmuestrales=tabl()[[1]]
    a=as.numeric(c(input$val))
    t.test(datosmuestrales,mu=a)
  })
  ttestles<- reactive({
    datosmuestrales=tabl()[[1]]
    a=as.numeric(c(input$val))
    t.test(datosmuestrales,mu=a, alternative="less")
  })
  ttestge<- reactive({
    datosmuestrales=tabl()[[1]]
    a=as.numeric(c(input$val))
    t.test(datosmuestrales,mu=a, alternative="greater")
  })
  kolmogo <- reactive({
    datosmuestrales=tabl()[[1]]
    media=mean(datosmuestrales)
    ks.test(datosmuestrales,pexp,1/media)
  })
  lillief <- reactive({
    set.seed(123)
    datosmuestrales=tabl()[[1]]
    LcKS(datosmuestrales,cdf = "pexp",nreps = 2000)
  })
  wilco1mu <- reactive({
    datosmuestrales=tabl()[[1]]
    mediana=as.numeric(input$val)

```

```

pos<-which(datosmuestrales==mediana)
if(length(pos)>0){
  datos=datosmuestrales[-pos]
} else{
  datos=datosmuestrales
}
wilcox.test(datos,mu=mediana)
})
signos <- reactive({
  datosmuestrales=tabl()[[1]]
  mediana=as.numeric(input$val)
  SIGN.test(datosmuestrales,md=mediana)
})
signosless<- reactive({
  datosmuestrales=tabl()[[1]]
  mediana=as.numeric(input$val)
  SIGN.test(datosmuestrales,md=mediana, alternative="less")
})
signosgreater<- reactive({
  datosmuestrales=tabl()[[1]]
  mediana=as.numeric(input$val)
  SIGN.test(datosmuestrales,md=mediana, alternative="greater")
})
testbinom<-reactive({
  if(!is.null(path()) && input$forma==1
    && input$prob!="" && input$val0!=""){
    datosmuestrales=tabl()[[1]]
    comp=as.numeric(input$val0)
    t_X=sum(datosmuestrales<comp)
    tamn=length(datosmuestrales)
    prob=as.numeric(input$prob)
    binom.test(t_X,n=tamn,p=prob)
  }
  else if (input$forma==2 && input$prob0!=""
    && input$totexit!="" && input$tamn!=""){
    prob=as.numeric(input$prob0)
    totalExi=as.numeric(input$totexit)
    tamn=as.numeric(input$tamn)
    binom.test(totalExi,n=tamn,p=prob)
  }
})
testbinomless<-reactive({
  if(!is.null(path()) && input$forma==1){
    datosmuestrales=tabl()[[1]]
    comp=as.numeric(input$val0)
    t_X=sum(datosmuestrales<comp)
    tamn=length(datosmuestrales)
    prob=as.numeric(input$prob)
    binom.test(t_X,n=tamn,p=prob,alternative = "less")
  }
  else{
    prob=as.numeric(input$prob0)
    totalExi=as.numeric(input$totexit)
    tamn=as.numeric(input$tamn)
    binom.test(totalExi,n=tamn,p=prob,alternative = "less")
  }
})

```



```

testbinomgreater<-reactive({
  if(!is.null(path()) && input$forma==1){
    datosmuestrales=tabl()[[1]]
    comp=as.numeric(input$val0)
    t_X=sum(datosmuestrales<comp)
    tamn=length(datosmuestrales)
    prob=as.numeric(input$prob)
    binom.test(t_X,n=tamn,p=prob,alternative = "greater")
  }
  else{
    prob=as.numeric(input$prob0)
    totalExi=as.numeric(input$totexit)
    tamn=as.numeric(input$tamn)
    binom.test(totalExi,n=tamn,p=prob,alternative = "greater")
  }
})
pathchiq <- reactive({input$aarch0})
tablchiq <-reactive({
  file <- pathchiq()
  ext <- tools::file_ext(file$datapath)
  req(file)
  validate(need(ext == "xlsx", "Por favor cargue un archivo xlsx"))
  tabla<-read_excel(file$datapath,col_types = c("numeric","numeric"))
})
chisqbond <- reactive({
  freq=tablchiq()[[1]]
  propor=tablchiq()[[2]]
  chisq.test(freq,p=propor)
})
output$salidatabla0 <- renderUI({
  if(!is.null(pathchiq()) && input$oper==3){
    renderDataTable({tablchiq()})
  } else {
    p("Por favor seleccione un archivo")
  }
})
output$salida0 <- renderUI({
  if(!is.null(path())){
    if(input$oper==1){
      div(h4("Test de normalidad de Shapiro-Wilk"),
        p("p-valor ", round(normal()$p.value,3)),
        p("Estadístico ",round(normal()$statistic,3))
      )
    } else if(input$oper==2){
      div(h4("Prueba t de una muestra"),
        p("p-valor ", round(ttestbi()$p.value,3)),
        p("Estadístico ",round(ttestbi()$statistic,3)))
    }
    else if(input$oper==4 && input$lilliefors==1){
      div(h4("Prueba de bondad de ajuste de Kolmogorov-Smirnov
        corregida por Lilliefors"),
        p("p-valor ", round(lillief()$p.value,3)),
        p("Estadístico ",round(lillief()$D.obs,3)))
    } else if(input$oper==4){
      div(h4("Prueba de Kolmogorov-Smirnov para una muestra"),
        p("p-valor ", round(kolmogo()$p.value,3)),
        p("Estadístico ",round(kolmogo()$statistic,3)))
    }
  }
})

```

```

} else if(input$oper==5){
  div(h4("Prueba exacta de rango con signo de Wilcoxon"),
    p("p-valor ", round(wilcolmu()$p.value,3)),
    p("Estadístico ",round(wilcolmu()$statistic,3)))
} else if(input$oper==6){
  div(h4("Prueba de signos de una muestra"),
    p("p-valor ", round(signos()$p.value,3)),
    p("Estadístico ",round(signos()$statistic,3)))
}

} else {
  p("Por favor seleccione un archivo")
}
})
output$salida1 <-renderUI({
  if(!is.null(pathchiq()) && input$oper==3){
    div(h4("Prueba de chi-cuadrado para probabilidades dadas"),
      p("p-valor ", round(chisqbond()$p.value,3)),
      p("Estadístico ", round(chisqbond()$statistic,3)))
  } else {
    p("Por favor seleccione un archivo")
  }
})
output$salida2 <-renderUI({
  if(!is.null(path())){
    if(input$oper==2 && input$contuni==1 && input$ttestuni==1){
      div(
        h4("Prueba t de una muestra"),
        p("p-valor ", round(ttestles()$p.value,3)),
        p("Estadístico ",round(ttestles()$statistic,3)))
    } else if(input$oper==2 && input$contuni==1 && input$ttestuni==2){
      div(
        h4("Prueba t de una muestra"),
        p("p-valor ", round(ttestge()$p.value,3)),
        p("Estadístico ", round(ttestge()$statistic,3)))
    }
  } else {
    p("Por favor seleccione un archivo")
  }
})
output$salidasignouni <- renderUI({
  if(!is.null(path())){
    if(input$oper==6 && input$contuni4==1 && input$signouni==1){
      div(
        h4("Prueba t de una muestra"),
        p("p-valor ", round(signosgreater()$p.value,3)),
        p("Estadístico ",round(signosgreater()$statistic,3)))
    } else if(input$oper==6 && input$contuni4==1 && input$signouni==2){
      div(
        h4("Prueba t de una muestra"),
        p("p-valor ", round(signosless()$p.value,3)),
        p("Estadístico ",round(signosless()$statistic,3)))
    }
  } else {
    p("Por favor seleccione un archivo")
  }
})

```

```

output$salidabinom1 <- renderUI({
  if(!is.null(path()) && input$forma==1){
    if(input$prob!="" && input$val0!=""){
      div(h4("Prueba binomial exacta"),
        p("p-valor ", round(testbinom()$p.value,3)),
        p("Estadístico ", round(testbinom()$statistic,3)))} else{
      p("Introduce el porcentaje y el valor a comparar")
    }
  }else {
    p("Por favor seleccione un archivo")
  }
})
output$salidabinom2 <- renderUI({
  if(input$forma==2){
    if(input$prob!="" && input$totexit!="" && input$tamn!=""){
      div(h4("Prueba binomial exacta"),
        p("p-valor ", round(testbinom()$p.value,3)),
        p("Estadístico ", round(testbinom()$statistic,3)))} else{
      p("Introduce el total de exitos,
        el tamaño muestral y el porcentaje")
    }
  }else {
    p("Por favor seleccione un archivo")
  }
})
output$salida8 <- renderUI({
  if(input$oper==7 && input$contuni3==1 && input$forma==1 &&
    input$binomuni==1 &&
    !is.null(path()) && input$prob!="" && input$val0!=""){
    div(h4("Prueba binomial exacta"),
      p("p-valor ", round(testbinomgreater()$p.value,3)),
      p("Estadístico ", round(testbinomgreater()$statistic,3)))
  } else if(input$oper==7 && input$contuni3==1 && input$forma==1 &&
    input$binomuni==2 &&
    !is.null(path())&& input$prob!="" && input$val0!=""){
    div(h4("Prueba binomial exacta"),
      p("p-valor ", round(testbinomless()$p.value,3)),
      p("Estadístico ", round(testbinomless()$statistic,3)))
  } else if (input$oper==7 && input$contuni3==1 && input$binomuni==2 &&
    input$totexit!="" && input$tamn!="" && input$prob0!="" &&
    input$forma==2) {
    div(h4("Prueba binomial exacta"),
      p("p-valor ", round(testbinomless()$p.value,3)),
      p("Estadístico ", round(testbinomless()$statistic,3)))
  } else if (input$oper==7 && input$contuni3==1 && input$binomuni==1 &&
    input$totexit!="" && input$tamn!="" && input$prob0!="" &&
    input$forma==2) {
    div(h4("Prueba binomial exacta"),
      p("p-valor ", round(testbinomgreater()$p.value,3)),
      p("Estadístico ", round(testbinomgreater()$statistic,3)))
  } else if(input$oper==7 && is.null(path())
    && input$contuni3==1 && input$forma==1){
    p("Por favor, seleccione un archivo")
  } else if(input$oper==7 && !is.null(path()) && input$contuni3==1 &&
    input$forma==1 && (input$prob=="" || input$val0=="")){
    p("Introduce el porcentaje y el valor a comparar")
  } else if(input$oper==7 && input$forma==2 && input$contuni3==1 &&

```

```

      (input$prob0==" " || input$totexit==" " || input$tamn==" "){
        p("Introduce el total de exitos, el tamaño muestral
          y el porcentaje")
      }
    })
    path2mind <- reactive({input$sarch1})
    tabl2mind <-reactive({
      file <- path2mind()
      ext <- tools::file_ext(file$datapath)
      req(file)
      validate(need(ext == "xlsx", "Por favor cargue un archivo xlsx"))
      tabla<-read_excel(file$datapath)
    })

    output$salidatabla1 <- renderUI({
      if(!is.null(path2mind())){
        renderDataTable({tabl2mind()})
      } else {
        p("Por favor seleccione un archivo")
      }
    })

    normal2mindx <- reactive({
      datos1=tabl2mind()[[1]]
      shapiro.test(datos1)
    })
    normal2mindy <- reactive({
      datos2=tabl2mind()[[2]]
      shapiro.test(datos2)
    })
    levene<-reactive({
      datos1=tabl2mind()[[1]]
      datos2=tabl2mind()[[2]]
      leveneTest(c(datos1,datos2),
                 group=as.factor(rep(c("a","b"),
                                     c(length(datos1),
                                       length(datos2)))), center=mean)
    })
    ttestbi2mind<- reactive({
      datos1=tabl2mind()[[1]]
      datos2=tabl2mind()[[2]]
      t.test(x=datos1,y=datos2,var.equal = TRUE)
    })
    # ttestles2mind<- reactive({
    #   datosmuestrales=tabl()[[1]]
    #   a=as.numeric(c(input$val))
    #   t.test(datosmuestrales,mu=a, alternative="less")
    # })
    # ttestge2mind<- reactive({
    #   datosmuestrales=tabl()[[1]]
    #   a=as.numeric(c(input$val))
    #   t.test(datosmuestrales,mu=a, alternative="greater")
    # })
    fishertest <- reactive({
      if(input$discret==1){
        datos1=tabl2mind()[[1]]
        datos2=tabl2mind()[[2]]

```

```

    comp=as.numeric(input$val1)
    datos1p<-as.integer(datos1>comp)
    datos2p<-as.integer(datos2>comp)
    n=length(datos1p)
    n_tot1=sum(datos1p)
    n_tot2=sum(datos2p)
    fisher.test(matrix(c(n_tot1,n_tot2,n-n_tot1,n-n_tot2),ncol = 2))
  } else{
    datos=table(tabl2mind())
    fisher.test(datos)
  }
})
kolmogo2m<-reactive({
  datos1=tabl2mind()[[1]]
  datos2=tabl2mind()[[2]]
  ks.test(x=datos1,y=datos2)
})
kolmogo2masin<-reactive({
  datos1=tabl2mind()[[1]]
  datos2=tabl2mind()[[2]]
  ks.test(x=datos1,y=datos2,exact = FALSE)
})
wilcoxon <- reactive({
  datos1=tabl2mind()[[1]]
  datos2=tabl2mind()[[2]]
  wilcox.test(datos1,datos2)
})
wilcoxonless <- reactive({
  datos1=tabl2mind()[[1]]
  datos2=tabl2mind()[[2]]
  wilcox.test(datos1,datos2,alternative = "less")
})
wilcoxongreater <- reactive({
  datos1=tabl2mind()[[1]]
  datos2=tabl2mind()[[2]]
  wilcox.test(datos1,datos2,alternative = "greater")
})
siegel <- reactive({
  datos1=tabl2mind()[[1]]
  datos2=tabl2mind()[[2]]
  SiegelTukeyTest(datos1,datos2)
})
siegelless <- reactive({
  datos1=tabl2mind()[[1]]
  datos2=tabl2mind()[[2]]
  SiegelTukeyTest(datos1,datos2,alternative = "less")
})
siegelgreater <- reactive({
  datos1=tabl2mind()[[1]]
  datos2=tabl2mind()[[2]]
  SiegelTukeyTest(datos1,datos2,alternative = "greater")
})
output$salida3 <-renderUI({
  if(!is.null(path2mind())){
    if(input$oper0==1){
      div(
        h4("Test de normalidad de Shapiro-Wilk"),

```

```

    p("p-valor ", colnames(tabl2mind())[1],
      paste(round(normal2mindx()$p.value,3),"",sep = "."),
      "Estadístico ",colnames(tabl2mind())[1],
      round(normal2mindx()$statistic,3)),
    p("p-valor ", colnames(tabl2mind())[2],
      paste(round(normal2mindy()$p.value,3),"",sep="."),
      "Estadístico ",colnames(tabl2mind())[2],
      round(normal2mindy()$statistic,3)))
  } else if(input$oper0==2){
    div(
      h4("Prueba de homogeneidad de varianza
        de Levene (centro = media)"),
      p("p-valor ", round(levene()$`Pr(>F)`[1],3)),
      p("Estadístico ", round(levene()$`F value`[1],3)))
  } else if(input$oper0==3){
    div(
      h4("Prueba t de dos muestras"),
      p("p-valor ", round(ttestbi2mind()$p.value,3)),
      p("Estadístico ", round(ttestbi2mind()$statistic,3)))
  } else if(input$oper0==4){
    if(input$discret==1 &input$val1==""){
      p("Por favor, selecciona un valor a comparar")
    } else if(input$discret==1 &input$val1!=""){
      div(
        h4("Prueba exacta de Fisher para datos de conteo"),
        p("p-valor ", round(fishertest()$p.value,3)))
    } else {
      div(
        h4("Prueba exacta de Fisher para datos de conteo"),
        p("p-valor ", round(fishertest()$p.value,3)))
    }
  } else if(input$oper0==5){
    if(input$asintotic==1){
      div(
        h4("Prueba asintótica de Kolmogorov-Smirnov
          para dos muestras"),
        p("p-valor ", round(kolmogo2masin()$p.value,3)),
        p("Estadístico ", round(kolmogo2masin()$statistic,3)))
    }else{
      div(
        h4("Prueba exacta de Kolmogorov-Smirnov de dos muestras"),
        p("p-valor ", round(kolmogo2m()$p.value,3)),
        p("Estadístico ", round(kolmogo2m()$statistic,3)))
    }
  } else if(input$oper0==6){
    div(
      h4("Prueba exacta de Mann-Whitney-Wilcoxon"),
      p("p-valor ", round(wilcoxon()$p.value,3)),
      p("Estadístico ", round(wilcoxon()$statistic,3)))
  } else if(input$oper0==7){
    div(
      h4("Prueba de Siegel-Tukey para igualdad de la dispersión"),
      p("p-valor ", round(siegel()$p.value,3)),
      p("Estadístico ", round(siegel()$statistic,3)))
  }
} else {
  p("Por favor seleccione un archivo")
}

```

```

}
})
output$salida5 <-renderUI({
  if(!is.null(path2mind()) && input$oper0==6 && input$contuni1==1 &&
    input$wiluni==1){
    div(
      p("p-valor ", round(wilcoxongreater())$p.value,3))
  } else if(!is.null(path2mind()) && input$oper0==6 &&
    input$contuni1==1 && input$wiluni==2){
    div(
      p("p-valor ", round(wilcoxonless())$p.value,3))
  }
})
output$salida6 <-renderUI({
  if(!is.null(path2mind()) && input$oper0==7 && input$contuni2==1 &&
    input$sieguni==1){
    div(
      p("p-valor ", round(siegelgreater())$p.value,3))
  } else if(!is.null(path2mind()) && input$oper0==7 &&
    input$contuni2==1 && input$sieguni==2){
    div(
      p("p-valor ", round(siegelless())$p.value,3))
  }
})
path2rel <- reactive({input$aarch3})
tabl2rel <-reactive({
  file <- path2rel()
  ext <- tools::file_ext(file$datapath)
  req(file)
  validate(need(ext == "xlsx", "Por favor cargue un archivo xlsx"))
  tabla<-read_excel(file$datapath)
})

output$salidatabla3 <- renderUI({
  if(!is.null(path2rel())){
    renderDataTable({tabl2rel()})
  } else {
    p("Por favor seleccione un archivo")
  }
})
mcnemar<-reactive({
  datos<-tabl2rel()
  n1=length(tabl2rel()[[1]])
  comp=as.numeric(input$val3)
  datos_vec=unlist(map(datos,as.vector))
  datos_FT=datos_vec<comp
  grupo=rep(names(tabl2rel()),each=n1)
  mcnemar.test(datos_FT,grupo)
})
output$salida9<- renderUI({
  if(!is.null(path2rel()) && input$oper2==1 && input$val3!=""){
    div(
      h4("Prueba Chi-cuadrado de McNemar con corrección
        de continuidad"),
      p("p-valor ", round(mcnemar())$p.value,3)),
      p("Estadístico ", round(mcnemar())$statistic,3))
  } else if(is.null(path2rel()) && input$oper2==1){

```

```

  p("Por favor, seleccione un archivo")
} else if(!is.null(path2rel()) && input$oper2==1 && input$val3==""){
  p("Por favor, introduzca un valor a comparar")
}
})
pathkmind <- reactive({input$sarch2})
tablkmind <-reactive({
  file <- pathkmind()
  ext <- tools::file_ext(file$datapath)
  req(file)
  validate(need(ext == "xlsx", "Por favor cargue un archivo xlsx"))
  tabla<-read_excel(file$datapath)
})

output$salidatabla2 <- renderUI({
  if(!is.null(pathkmind())){
    renderDataTable({tablkmind()})
  } else {
    p("Por favor seleccione un archivo")
  }
})

fishertestkmu <- reactive({
  if(input$discret0==1){
    datos<-tablkmind()
    colu=dim(datos) [2]
    n1<-unlist(tablkmind() %>%map(na.omit) %>%map(length))
    datos1=tablkmind() [[1]]
    n=length(datos1)
    comp=as.numeric(input$val2)
    datos_vec<-unlist(map(datos,as.vector))
    nombre<-rep(names(tablkmind()),n1)
    names(datos_vec)<-nombre
    datos_01<-as.integer(datos_vec>comp)
    names(datos_01)<-nombre
    yu<-split(datos_01, names(datos_vec))
    aciertos<-unlist(lapply(yu,sum))
    errores<-n-aciertos
    tabla=matrix(c(aciertos,errores),byrow=FALSE,ncol=2)
    colnames(tabla)=c(1,0)
    rownames(tabla)=c("X", "Y", "Z")
    chisq.test(tabla)
  } else{
    aciertos=apply(tablkmind(),2,sum)
    n=length(tablkmind() [[1]])
    errores=n-aciertos
    tabla=matrix(c(aciertos,errores),byrow=FALSE,ncol=2)
    colnames(tabla)=c(1,0)
    rownames(tabla)=names(tablkmind())
    chisq.test(tabla)
  }
})

kruskal<-reactive({
  datos<-tablkmind()
  colu=dim(datos) [2]
  n1<-unlist(tablkmind() %>%map(na.omit) %>%map(length))

```



```

grupo<-rep(1:colu,n1)
datos_vec<-unlist(map(datos,as.vector))
kruskal.test(datos_vec,grupo)
})
compmult<-reactive({
  datos<-tablmind()
  colu=dim(datos)[2]
  n1<-unlist(tablmind() %>%map(na.omit) %>%map(length))
  grupo<-rep(1:colu,n1)
  datos_vec<-unlist(map(datos,as.vector))
  kwAllPairsNemenyiTest(datos_vec,grupo)
})
friedman<- reactive({
  datos<-tablmind()
  colu=dim(datos)[2]
  datos_vec<-unlist(map(datos,as.vector))
  n1<-length(tablmind()[[1]])
  datosmuestra<-matrix(datos_vec,ncol=colu,byrow=FALSE,
                        dimnames=list(1:n1, names(tablmind()))))
  friedman.test(datosmuestra)
})
compmult0<-reactive({
  datos<-tablmind()
  colu=dim(datos)[2]
  datos_vec<-unlist(map(datos,as.vector))
  n1<-length(tablmind()[[1]])
  datosmuestra<-matrix(datos_vec,ncol=colu,byrow=FALSE,
                        dimnames=list(1:n1, names(tablmind()))))
  frdAllPairsNemenyiTest(datosmuestra)
})
output$salida7 <-renderUI({
  if(!is.null(pathkmind())){
    if(input$oper1==1 && input$discret0==1 && input$val2!=""){
      div(
        h4("Prueba de Chi-cuadrado de homogeneidad de Pearson"),
        p("p-valor ", round(fishertestkmu()$p.value,3)),
        p("Estadístico ", round(fishertestkmu()$statistic,3))
      )
    } else if(input$oper1==1 && input$discret0==1 && input$val2==""){
      p("Por favor, introduzca un valor a comparar")
    } else if(input$oper1==1 && input$discret0!=1){
      div(
        h4("Prueba de Chi-cuadrado de homogeneidad de Pearson"),
        p("p-valor ", round(fishertestkmu()$p.value,3)),
        p("Estadístico ", round(fishertestkmu()$statistic,3))
      )
    } else if(input$oper1==2){
      div(
        h4("Prueba de suma de rangos de Kruskal-Wallis"),
        p("p-valor ", round(kruskal()$p.value,3)),
        p("Estadístico ", round(kruskal()$statistic,3))
      )
    } else if(input$oper1==3){
      div(
        h4("Prueba de suma de rangos de Friedman"),
        p("p-valor ", round(friedman()$p.value,3)),
        p("Estadístico ", round(friedman()$statistic,3))
      )
    }
  } else {
    p("Por favor seleccione un archivo")
  }
}

```

```

}
})

output$salidacomp <-renderUI({
  if(!is.null(pathkmind())){
    if(input$oper1==2 && input$comp==1){
      tabla<-compmult()$p.value
      nombre<-names(tablkmind())
      n=length(nombre)
      colnames(tabla)<-nombre[-n]
      rownames(tabla)<-nombre[-1]
      tabla2<-compmult()$statistic
      colnames(tabla2)<-nombre[-n]
      rownames(tabla2)<-nombre[-1]
      div( h4("Prueba de comparación de rango de
              todos los pares de Nemenyi"),
            div("p-valor ", renderTable(round(tabla,3),
                                          rownames=TRUE)),
            div("Estadístico ", renderTable(round(tabla2,3),
                                          rownames=TRUE)))
    }
  } else{
    p("Por favor seleccione un archivo")
  }
})

output$salidacomp0 <-renderUI({
  if(!is.null(pathkmind())){
    if(input$oper1==3 && input$comp0==1){
      tabla<-compmult0()$p.value
      nombre<-names(tablkmind())
      n=length(nombre)
      colnames(tabla)<-nombre[-n]
      rownames(tabla)<-nombre[-1]
      tabla2<-compmult0()$statistic
      colnames(tabla2)<-nombre[-n]
      rownames(tabla2)<-nombre[-1]
      div( h4("Prueba de comparación de rango de todos
              los pares de Nemenyi"),
            div("p-valor ", renderTable(round(tabla,3),
                                          rownames=TRUE)),
            div("Estadístico ", renderTable(round(tabla2,3),
                                          rownames=TRUE)))
    }
  } else{
    p("Por favor seleccione un archivo")
  }
})
}

shinyApp(ui,server)

```

Bibliografía

- [1] JJ Allaire, Yihui Xie, Christophe Dervieux, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. *rmarkdown: Dynamic Documents for R*, 2023. URL <https://CRAN.R-project.org/package=rmarkdown>. R package version 2.22.
- [2] Fernando López Blázquez. *Apuntes de clase de la signatura Cálculo de probabilidades*, 2015.
- [3] Winston Chang, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. *shiny: Web Application Framework for R*, 2022. URL <https://CRAN.R-project.org/package=shiny>. R package version 1.7.4.
- [4] Joaquín García de las Heras. *Apuntes de clase de la signatura Ampliación de Inferencia Estadística*, 2021.
- [5] J.A. Fernández García, F.R. y Mayor Gallego. *Muestreo en poblaciones finitas: Curso básico*. PPU, S.A., Promociones y Publicaciones Universitarias, S.A., 1994.
- [6] JD. Gibbons and S Chakraborti. *Nonparametric Statistical Inference*. Marcel Dekker, 2003.
- [7] Jesús San Martín y Venancio Tomeo Isaís Uña. *Cálculo de probabilidades*. IBERGARCETA PUBLICACIONES, S.L., 2009.
- [8] M.L Lipschutz, S. y Lipson. *Teoría y Problemas de Probabilidad*. McGraw-Hill., 2001.
- [9] Pedro L. Luque-Calvo. *Escribir un Trabajo Fin de Estudios con R Markdown*, 2017.
- [10] Pedro L. Luque-Calvo. *Cómo crear Tablas de información en R Markdown*, 2019.
- [11] Juan Luis Muñoz Reyes, Ana Maria; Moreno Rebollo. *Apuntes de clase de la signatura Muestreo Estadístico.*, 2016-2018.
- [12] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL <https://www.R-project.org/>.
- [13] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA, 2015. URL <http://www.rstudio.com/>.
- [14] Carson Sievert, Richard Iannone, JJ Allaire, and Barbara Borges. *flexdashboard: R Markdown Format for Flexible Dashboards*, 2023. <https://pkgs.rstudio.com/flexdashboard/>, <https://github.com/rstudio/flexdashboard/>.
- [15] M.R. Spiegel. *Teoría y problemas de probabilidad y estadística*. McGraw-Hill., 1992.
- [16] Nicholas Tierney. *R Markdown for Scientists*, 2020.
- [17] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D’Agostino McGowan, Romain François, Garrett Golemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi: 10.21105/joss.01686.

- [18] Hadley Wickham, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*, 2023. URL <https://CRAN.R-project.org/package=ggplot2>. R package version 3.4.2.
- [19] Hadley Wickham, Romain François, Lionel Henry, Kirill Müller, and Davis Vaughan. *dplyr: A Grammar of Data Manipulation*, 2023. URL <https://CRAN.R-project.org/package=dplyr>. R package version 1.1.2.
- [20] Leland Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Springer, 2005.
- [21] Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2023. URL <https://yihui.org/knitr/>. R package version 1.43.