Universidad Torcuato Di Tella

LABORATORIO PARA EL ANÁLISIS DE DATOS ECONÓMICOS Y FINANCIEROS

Trabajo Práctico

FECHA DE ENTREGA: 20/10/2025

Titulares de la Cátedra :

IAN BOUNOS
MARTIN ESCOBAR

Alumnos :

TOMAS SCHVARTZMAN
LUCAS VYHÑAK
SANTIAGO SARKISSIAN

Ejercicio de Análisis de Datos

Para este ejercicio elegimos usar el dataset de exportaciones argentinas de septiembre de 2025 provisto por Operadores de Comercio Exterior (OCE) de ARCA: https://www.afip.gob.ar/operadoresComercioExterior/informacionAgregada/informacion-agregada.asp

En el siguiente link está la opción de descarga y además contiene una tabla con la información de cada columna del archivo, que usaremos luego para renombrar las columnas

Cargamos los paquetes

```
library(tidyverse)
library(dplyr)
library(readxl)
```

Importamos el dataset

Lo llamamos "expo". Usamos **`read_delim`** en vez de **`read_csv`** porque en este dataset, el separador es una comilla simple en vez de una coma.

- Ponemos la ruta
- **`delim = "´"`**, para indicar que el separador es este
- Al probar, nos dimos cuenta que había filas que tenían muchos espacios de más por delante y por detrás, así que los sacamos usando **`trim_ws = TRUE`**.

Renombramos las columnas

Notamos que las columnas del dataset tenían nombres no tan fáciles de escribir, por lo que usamos la tabla que está en link para renombrar las columnas con *rename* (las últimas tres las dejamos iguales porque son claras).

Limpiamos el dataset

Primero usamos **`glimpse`** para verificar qué tipo de datos hay en cada columna. Vemos que todas son **<chr>**, por lo que hay que cambiar las numéricas a números para poder operar.

Abrimos un pipe y modificamos (**`expo <- expo %>%`**)

- Usamos **`filter`** para quedarnos solo con las exportaciones, esto es, en la columna "tipo" que diga "E" (o "e"), ya que hay "I" (de importaciones) o "---" (de separadores).
- Usamos **`!is.na`** para mantener las filas en donde el valor de la columna "**`monto_fob_dolares`**" no está vacía (! significa no).
- Ahora usamos mutate para convertir varias columnas (para eso usamos *across*) de texto a número. Para hacerlo más fácil, usamos matches, así seleccionamos todas las columnas que contengan las palabras que pusimos. Luego usamos **`as.numeric`** para convertir el texto (chr) en número (dbl).

```
expo <- expo %>%
  filter(tipo %in% c("E", "e"),
  !is.na(monto_fob_dolares)) %>%
  mutate(across(matches("monto|kilos|precio|cantidad"), as.numeric))
```

Chequeamos con **`view`** a ver si está todo bien.

Pregunta 1: ¿cuáles fueron los 10 países a los que más se exportó en septiembre de 2025?

Notar lo siguiente: los países aparecen por un código, así que tenemos que relacionar cada país con su respectivo código.
Para eso descargamos el "código de países" del INDEC, que viene en formato .xls

- Primero lo importamos. Usamos **`read_excel`** y salteamos una fila porque no sirve.

Ahora unimos cada código con su país en "expo": usamos **`left_join`**
Chequeamos que esté todo bien y respondemos la pregunta con una tabla.

`expo %>%`

- **`filter(!is.na(`NOMBRE DE USO COMÚN`)`** : descarta todas las filas en las que el nombre de país esté vacía (NA). Ponemos la columna entre ` porque tiene espacios y tildes.
- **`!`NOMBRE DE USO COMÚN` %in% c("América", "Asia", "África", "Europa", "Oceanía", "Última actualización: octubre de 2015."))`** : eliminamos entradas que no son países.

- **`group_by(`NOMBRE DE USO COMÚN`)`** : agrupamos las filas por país.
- **`summarise(valor_total=sum(monto_fob_dolares, na.rm = TRUE)/1000000)`** : calculamos la suma de **`monto_fob_dolares`** por país (ignorando los NA con **`na.rm = TRUE`**). Lo guardamos en una nueva columna llamada "valor_total" y dividimos por 1.000.000 para que sea mejor visualmente.
- **`arrange(desc(valor_total)) %>% head(10)`** : ordenamos de mayor a menor y tomamos los 10 primeros (o sea los más altos).

Pregunta 2: ¿Cuáles fueron los productos (según código ncm) que más se exportaron en septiembre de 2025?

Ahora armamos otra tabla con el mismo procedimiento. Los resultados son códigos que pertenecen a estos productos.

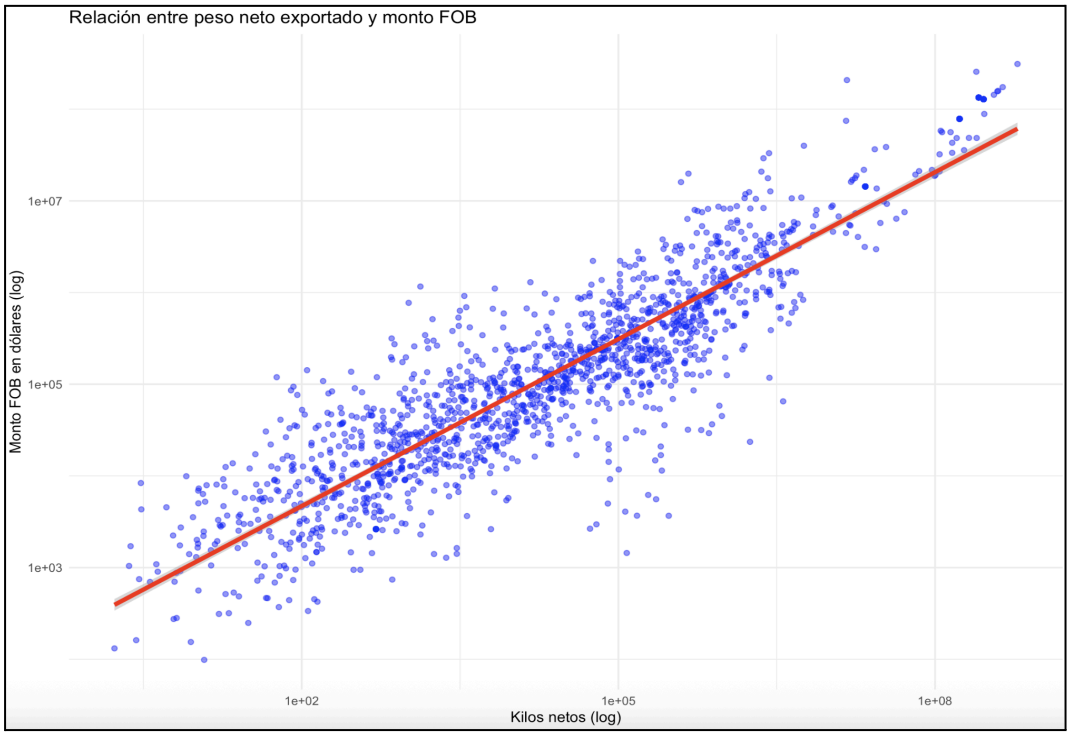
Código NCM	Descripción
2709.00.10	Petróleo crudo
1201.90.00	Soja
1507.10.00	Aceite de soja en bruto
8704.21.90	Camiones diésel con peso total entre 5 y 20 toneladas
2304.00.10	Harina y pellets de soja
2710.12.59	Fuel oil y otros aceites minerales pesados
0202.30.00	Carne bovina congelada (deshuesada)
1005.90.10	Maíz amarillo duro
1001.99.00	Trigo y morcajo (salvado)
0306.17.10	Camarones y langostinos congelados

Pregunta 3: ¿Existe relación entre cantidad importada y monto FOB?

Un monto FOB (Free On Board) es el valor de la mercadería puesta en el medio de transporte, en el puerto de salida, sin incluir fletes ni seguros internacionales. Sería lo que vale la mercadería antes de contar los costos de transporte.

Para responder esto, primero creamos una tabla nueva con los datos relevantes, los cuales son **`fecha`**, **`codigo_cm`** (estos dos para no perder las identificaciones de las exportaciones), **`kilos_netos`**, **`monto_fab_dolares`**.

Ahora hacemos un gráfico que explique lo que pide la pregunta. En el gráfico pasamos las variables a una escala logarítmica para mejorar la visualización de los datos, e incluimos una recta de mejor ajuste para ver la tendencia general de los datos. A continuación vemos el gráfico:



En el gráfico vemos que hay una clara relación creciente entre el monto FOB en dólares y el peso neto exportado (en kilos) lo que nos indica que, a mayor peso exportado, el monto FOB (el valor de la mercadería antes de los costos de transporte) en dólares aumenta. Económicamente es algo lógico ya que las exportaciones de gran volumen (por ejemplo, commodities agrícolas, minerales o combustibles) representan los mayores montos en dólares.

Ejercicio de Análisis econométrico con datos de gapminder

Parte 1. Ingreso por persona

Antes de empezar, cargamos el dataset y lo nombramos gapminder:

```
gapminder <-
read_csv("/Users/tommy/Downloads/gapminder.csv")
```

Inciso 1

Filtramos los datos que necesitamos y armamos la tabla **`ingreso_argentina`**. Con eso creamos el gráfico usando **`ggplot()`**, y todos los comandos que se ven en el código, para que nos quede de la siguiente manera:

En el gráfico vemos que hay una tendencia creciente en el ingreso por persona en la Argentina, aunque no constante ya que se observan distintas caídas como por ejemplo en el año 1974 y 1998.

Inciso 2

Separamos los datos entre el entrenamiento (train) y lo que testeamos (test). Usamos los últimos 10 años para testear, y el resto para entrenamiento.

Luego nombramos las variables independientes (**`x_train`** y **`x_test`**) y las variables dependientes (**`y_train`** e **`y_test`**) de nuestros modelos. Con eso, estimamos los modelos para los datos de train en el orden solicitado, y pedimos resumen con los datos relevantes, utilizando el comando **lm()**. Veamos qué resultados nos devuelve cada regresión:

`mod_lineal` (Modelo lineal):

```
Residuals:
    Min      1Q  Median      3Q     Max
-2509.1  -887.6  -213.4  1023.3  2288.1

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -171079.77   32497.28  -5.264 5.42e-06 ***
x_train          92.95      16.41   5.663 1.52e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1243 on 39 degrees of freedom
Multiple R-squared:  0.4513,    Adjusted R-squared:  0.4372
F-statistic: 32.07 on 1 and 39 DF,  p-value: 1.521e-06
```

`mod_polin2` (Modelo polinómico de segundo grado):

```
Residuals:
    Min      1Q  Median      3Q     Max
-2622.2  -736.2  -146.4  1021.0  1993.9

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  -1.126e+07  5.896e+06  -1.910   0.0637 .
poly(x_train, 2, raw = TRUE)1  1.129e+04  5.956e+03   1.896   0.0655 .
poly(x_train, 2, raw = TRUE)2 -2.829e+00  1.504e+00  -1.881   0.0677 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1205 on 38 degrees of freedom
Multiple R-squared:  0.498,     Adjusted R-squared:  0.4716
F-statistic: 18.85 on 2 and 38 DF,  p-value: 2.059e-06
```

`mod_polin10` (Modelo polinómico de grado 10):

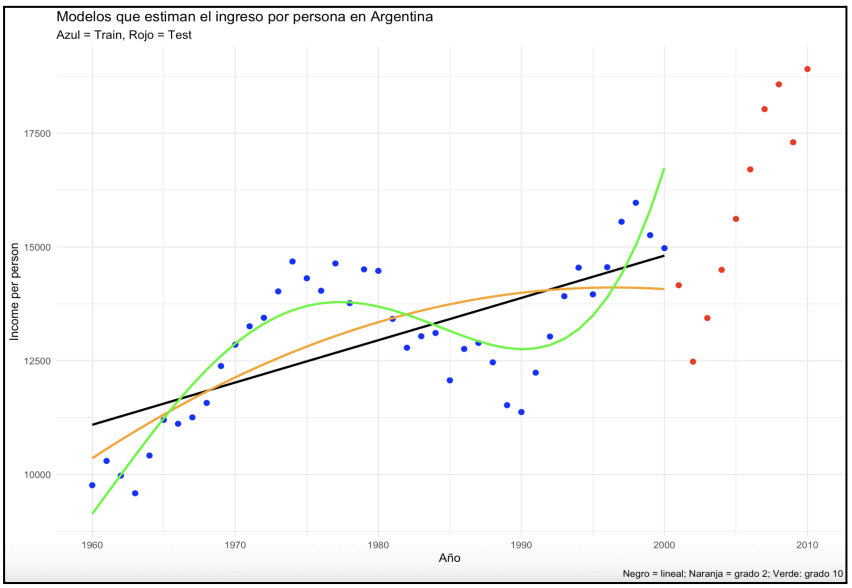
```
Residuals:
     Min       1Q   Median       3Q      Max
-1759.05  -501.08   -24.63   630.00  1352.06

Coefficients: (6 not defined because of singularities)
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                   1.513e+11  6.237e+10   2.426   0.0204 *
poly(x_train, 10, raw = TRUE)1 -2.589e+08  1.063e+08  -2.435   0.0200 *
poly(x_train, 10, raw = TRUE)2  1.500e+05  6.137e+04   2.444   0.0196 *
poly(x_train, 10, raw = TRUE)3 -2.957e+01  1.205e+01  -2.453   0.0191 *
poly(x_train, 10, raw = TRUE)4          NA         NA      NA       NA
poly(x_train, 10, raw = TRUE)5          NA         NA      NA       NA
poly(x_train, 10, raw = TRUE)6          NA         NA      NA       NA
poly(x_train, 10, raw = TRUE)7          NA         NA      NA       NA
poly(x_train, 10, raw = TRUE)8          NA         NA      NA       NA
poly(x_train, 10, raw = TRUE)9   5.970e-21  2.381e-21   2.507   0.0168 *
poly(x_train, 10, raw = TRUE)10         NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 779.7 on 36 degrees of freedom
Multiple R-squared:  0.8008,    Adjusted R-squared:  0.7787
F-statistic: 36.19 on 4 and 36 DF,  p-value: 3.749e-12
```

Ahora, calculamos con el código **`rmse()`** la raíz del error cuadrático medio (RMSE) para medir el desempeño del test, el mejor modelo es el que menor RMSE tenga. A continuación vemos los resultados:

```
   Lineal    Grado2   Grado10
2012.472  3114.474 11311.216
```

Vemos que el modelo con menor RMSE es el lineal y el de mayor RMSE es el polinomio de grado 10, lo cual quiere decir que el modelo polinómico de grado 10 tiene overfitting.

Ahora graficamos los modelos de train, e incluimos puntos de los datos, tanto de train como de test, y el gráfico nos queda de la siguiente manera:



Con este gráfico, vemos que el modelo lineal es el que mejor refleja el crecimiento del ingreso por persona, pero sin las variaciones. El modelo cuadrático refleja un poco más las variaciones y se ajusta mejor al patrón que siguen los datos. Y el modelo de grado 10 se ajusta muy bien a los datos, pero muestra un claro overfitting, como vimos cuando calculamos el RMSE.

Inciso 3.a

Elegimos los países (a parte de argentina), Bolivia, Chile, Paraguay, Uruguay y creamos la tabla **`sudamérica`** con los datos que necesitamos y la convertimos a formato ancho.
Calculamos la matriz de correlaciones que nos queda así:

	Argentina	Bolivia	Chile	Paraguay	Uruguay
Argentina	1.0000000	0.9244884	0.7650413	0.6733786	0.8291831
Bolivia	0.9244884	1.0000000	0.7450691	0.7099295	0.7985954
Chile	0.7650413	0.7450691	1.0000000	0.7377218	0.9407932
Paraguay	0.6733786	0.7099295	0.7377218	1.0000000	0.8555424
Uruguay	0.8291831	0.7985954	0.9407932	0.8555424	1.0000000

La comentaremos al final del inciso (b).

Inciso 3.b

Calculamos las variaciones porcentuales anuales y armamos la siguiente matriz de correlaciones:

	Argentina	Bolivia	Chile	Paraguay	Uruguay
Argentina	1.0000000	0.2066589	0.1691989	0.1469514	0.5127562
Bolivia	0.2066589	1.0000000	0.1349516	0.2606822	0.2653272
Chile	0.1691989	0.1349516	1.0000000	0.1880664	0.3655066
Paraguay	0.1469514	0.2606822	0.1880664	1.0000000	0.3244240
Uruguay	0.5127562	0.2653272	0.3655066	0.3244240	1.0000000

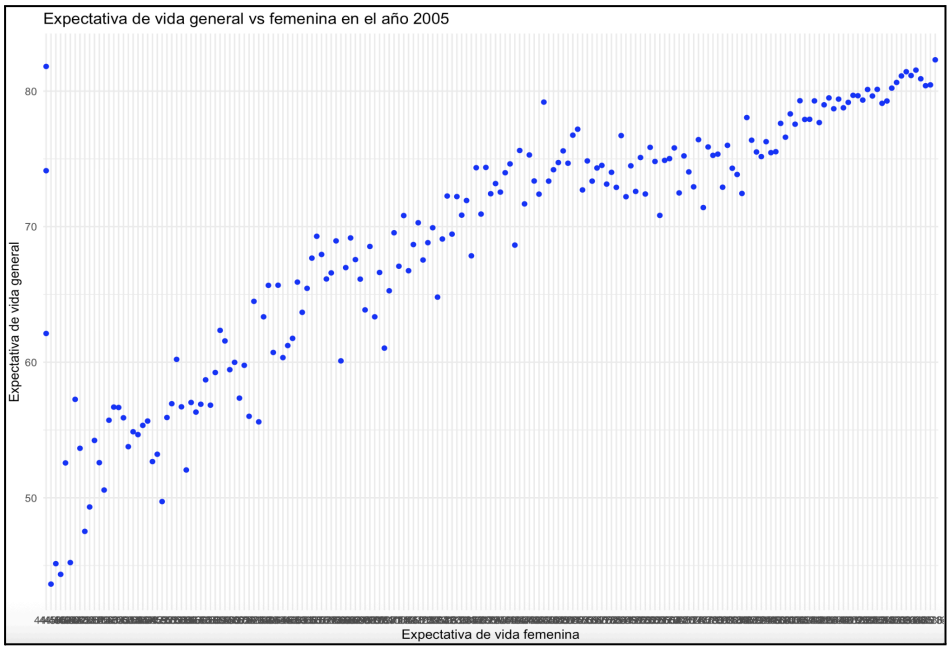
Se puede ver en la primera matriz que las correlaciones entre los ingresos son muy altas, lo que nos dice que a largo plazo, en los países sudamericanos, se observa un crecimiento general muy similar en todos los países. En cambio, en la segunda matriz vemos que las correlaciones son positivas, aunque bajas, lo que nos dice que los países crecen de una manera muy diferente año a año, aunque si todas crecen muy parecido en el largo plazo.

Parte 2: Esperanza de vida y género

Usaremos el año 2005 para toda esta parte.

Inciso 5

Creamos la tabla **`expectativa_vida_2005`** con los datos relevantes para este inciso, y luego creamos el gráfico:



Vemos que hay una relación creciente (prácticamente lineal) entre la expectativa de vida total y la femenina.

Inciso 6

Empezamos estimando la regresión lineal simple de **`life_expectancy`** contra **`life_expectancy_female`** con el código **`lm()`**.

Observamos que hay una relación casi proporcional entre ambas variables, es decir que cuando aumenta la expectativa de vida femenina, la expectativa de vida general lo hace en similar magnitud. El R cuadrado es muy alto (0.9394), lo que nos dice que la expectativa de vida femenina explica casi toda la variabilidad de la expectativa de vida total.

Inciso 7

Para realizar el test de t, vamos a crear una variable de diferencia que nos lo facilita. Primero, convertimos las variables de expectativa de vida a numéricas.

Ahora si, realizamos el test con la siguiente hipótesis:

H_0: $\mu = 0$

H_1: $\mu > 0$

Usamos el comando **`t.test`** para realizar el test. Esto nos devuelve un p-valor de 7.2x(10^-13), lo que es mucho menor a 0.05. Por lo tanto, rechazamos la hipótesis nula. Esto quiere decir que las mujeres tienden a vivir más tiempo que el promedio general de las personas en la mayoría de los países.

Inciso 8

Creamos la tabla **`vida_ingreso_2005`** con los datos necesarios, verificamos que los datos sean numéricos y estimamos la regresión múltiple de la variable **`life_expectancy`** sobre **`life_expectancy_female`** e **`income_per_person`**. Nos da los siguientes datos:

```
Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           8.444e+00  1.293e+00   6.530 6.44e-10 ***
life_expectancy_female 8.538e-01  1.940e-02  44.010  < 2e-16 ***
income_per_person      2.283e-05  1.053e-05   2.168   0.0315 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.285 on 181 degrees of freedom
  (9 observations deleted due to missingness)
Multiple R-squared:  0.9413,    Adjusted R-squared:  0.9406
F-statistic:  1450 on 2 and 181 DF,  p-value: < 2.2e-16
```

El coeficiente de expectativa de vida femenina es de 0.85, lo que quiere decir que cuando aumenta en un año la expectativa de vida femenina, la general aumenta en 0.85 años. Por otro lado, el coeficiente de ingreso es de 2.2x(10^-5). Como un aumento de un dólar es muy irrelevante, el coeficiente nos dice, por ejemplo, que para aumentar la expectativa de vida por un año, el ingreso debe aumentar en, aproximadamente, 44.000 USD. Pasando con el R2, vemos que el valor es de 0.94. Esto nos dice que las variables independientes explican casi toda la variabilidad de la expectativa de vida, y aunque sea muy similar al R2 de la regresión simple, igualmente mejora y cómo es significativo, vale la pena incluir la variable **`income_per_person`** en el modelo.

Inciso 9

Para regresar un nuevo modelo de **`life_expectancy`**, elegimos las variables **`children_per_woman`**, **`income_per_person`**, y **`population`**. Los elegimos porque son variables tanto demográficas como económicas, y pueden explicar la expectativa de vida general de las personas. Creamos primero una nueva tabla con los datos que necesitamos y luego creamos el modelo, el cual nos da los siguientes valores:

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         7.869e+01  1.192e+00  65.987  < 2e-16 ***
children_per_woman -3.752e+00  2.789e-01 -13.450  < 2e-16 ***
income_per_person   1.216e-04  2.428e-05   5.008 1.31e-06 ***
population         -5.348e-10  3.091e-09  -0.173    0.863
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.52 on 180 degrees of freedom
  (9 observations deleted due to missingness)
Multiple R-squared:  0.6591,    Adjusted R-squared:  0.6534
F-statistic:   116 on 3 and 180 DF,  p-value: < 2.2e-16
```

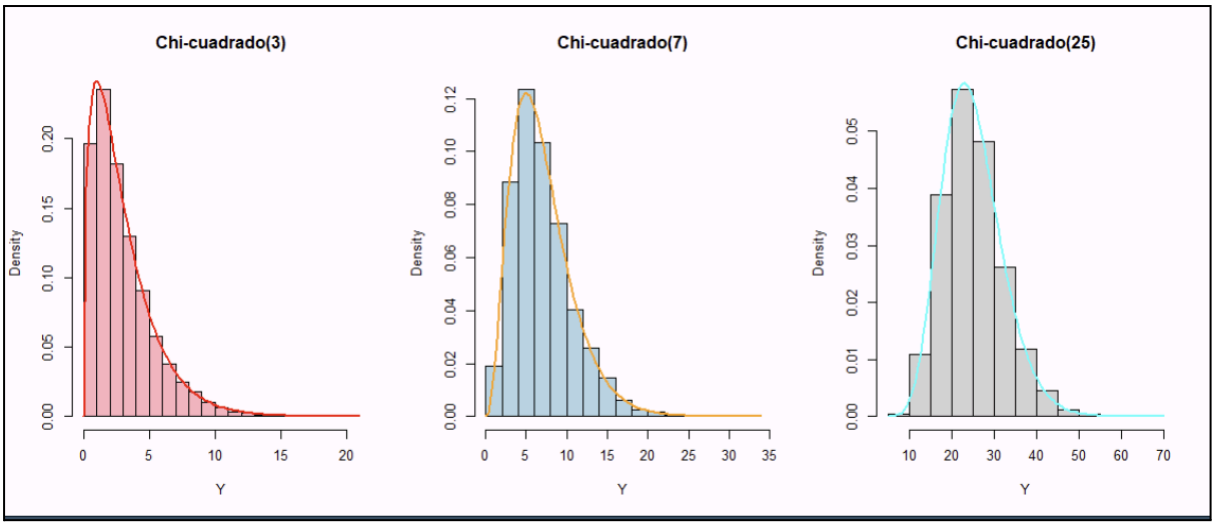
Con estos resultados, vemos que la cantidad de hijos por mujer nos muestra que, con cada hijo nacido, se reduce la esperanza de vida adicional por 3.75 años. Luego, un aumento del ingreso per cápita de 10.000 USD refleja un aumento de 1.2 años en la esperanza de vida. Por último, el coeficiente de población nos muestra que la variable no es significativa ($p = 0.863$), por lo que no tiene una relación estadística directa con la expectativa de vida. Luego, con el R2 vemos que el modelo explica el 66% de la variabilidad de la expectativa de vida, lo que es un nivel moderadamente alto teniendo en cuenta que tenemos tres variables y una de las cuales no es significativa.

Ejercicio de Simulación 1: Demanda con preferencias Cobb-Douglas

Nota: Los comandos "**`View`**" para ver los dataframe están puestos después de un # para poder correr el código sin interrupciones.

Inciso 1)

El ejercicio pide una función de ingreso que siga una distribución chi cuadrado creamos la función **`simular_ingreso`** con **`function`** para n y k genéricos con tal de poder crear resultados distintos cada vez que la utilizamos con distintos parámetros, eso nos va a ayudar a poder comparar entre distintos niveles de k. En cuanto a cómo elegir el k, el k mismo nos dice la cantidad de factores independientes que determinan el ingreso. Hacemos la comparativa con distintos grados de libertad("k") para un mismo "n" grande (elegido arbitrariamente), los definimos como **`yk11`** si tiene un k=11, armamos los gráficos y vemos que la dispersión se asemeja a una distribución normal a medida que aumenta el número de k. Es decir, con un nivel de k cercano al 3 vemos claramente la falta de simetría y una "desigualdad" en los ingresos (es decir, habría mucha gente con ingresos bajos y poca con ingresos altos). Con un nivel de k=25 vemos como se parece a una normal donde hay simetría en la dispersión de los ingresos, la mayoría de los datos coinciden con la media y están distribuidos más simétricamente que en el otro caso.



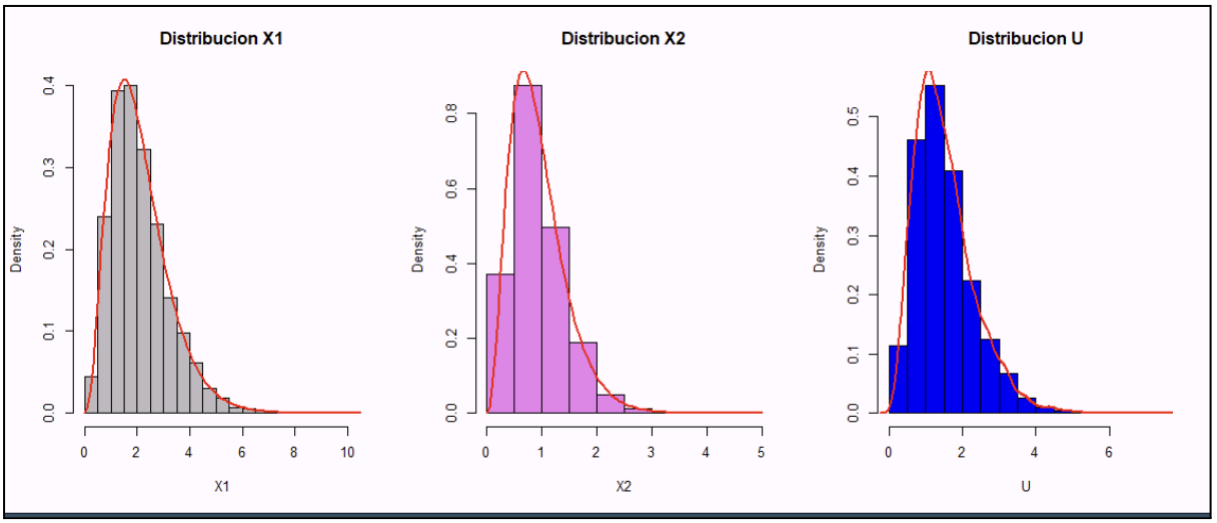
Inciso 2)

El ejercicio nos pide una función de demanda que según los parámetros devuelva las demandas óptimas y la utilidad indirecta. Para esto, definimos una "**`function`**" que incluya los parámetros de interés y definimos al x1, x2, U según las fórmulas dadas por el enunciado, además, agregamos un dataframe para que podamos tener los resultados del ejemplo dados los parámetros elegidos.

Inciso 3)

El ejercicio pide simular eligiendo los parámetros y en 10000 hogares, establecemos los parámetros arbitrariamente, pero con tal de que nos dé un x1 y x2 distintos, una vez establecemos los parámetros para esta simulación en específico lo nombramos "ejemplo1" y establecemos la media de x1 y x2 además de los cuantiles.

La función **`demanda_CD`** toma como entrada el vector de ingresos Y (simulado previamente con k = 7 grados de libertad, almacenado en Yk7). De este modo, R interpreta internamente Y = Yk7 durante la ejecución.



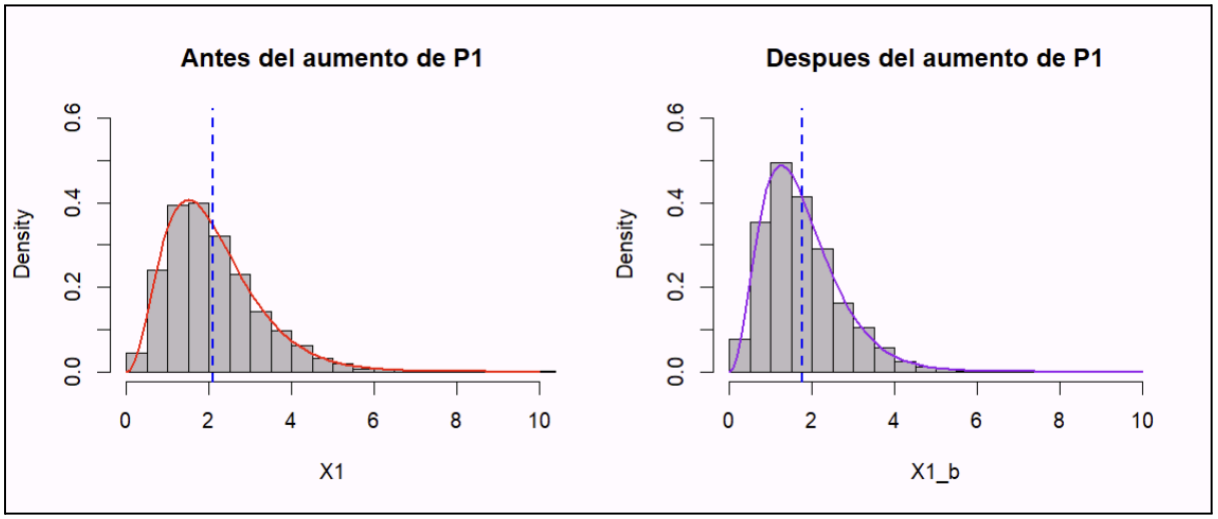
Inciso 4)

El ejercicio pide que creemos un umbral c (mayor que 0) y que calculemos la probabilidad de que el consumo de la población “caiga” por debajo de este umbral, por eso le tomamos la media a la columna de los valores de X1 < c, y lo definimos como pide el enunciado.

Inciso 5)

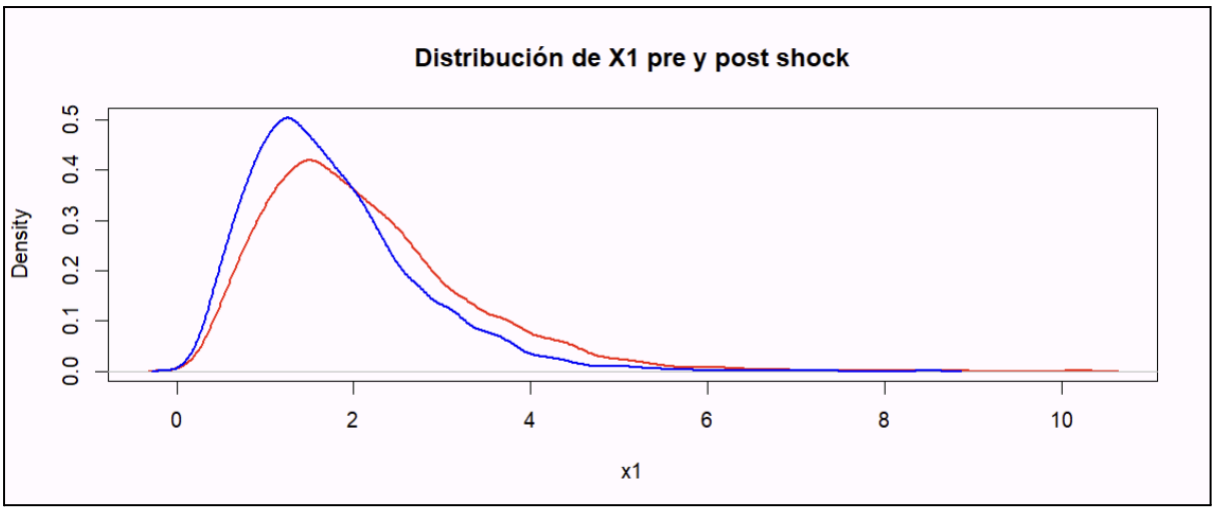
Definimos P1_b como P1 multiplicado por sí mismo más el shock, definimos un nuevo vector de demandas (le agregamos "_b" para denotar que es del shock). Y determinamos "**`ejemplo2`**" al data frame con las demandas y el nuevo precio de shock.

De ahí repetimos la simulación y hacemos histogramas para comparar con las medias siendo las líneas azules punteadas para una mejor comparativa. Vemos que la media antes del shock era mayor, lo cual es totalmente esperable dado que subió el precio del bien 1 y causó que la media de consumo de los hogares haya caído. Resultado que también se ve en los gráficos, donde la función de densidad se concentra más a la "izquierda" que antes.



Inciso 6)

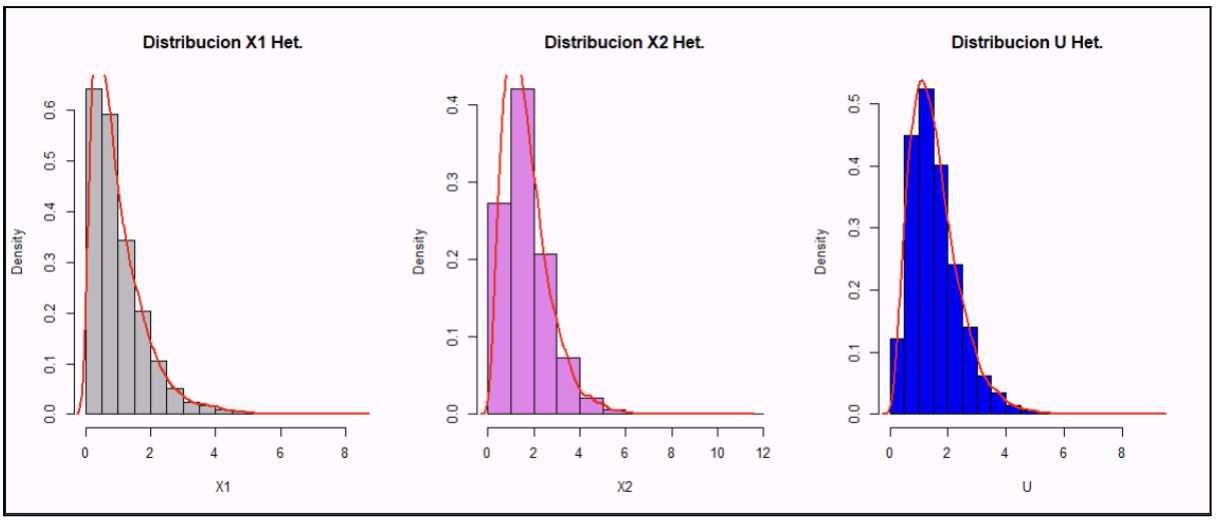
Definimos con una "**`d_`**" a las funciones de densidad de pre-shock y post-shock y graficamos a ambas distribuciones como líneas para poder comparar. Donde se reafirma la conclusión de antes. La función de densidad roja se concentra en valores de consumo de X1 más altos antes del shock que después de una suba de P1. Lo cual tiene sentido, y finalmente, la utilidad indirecta después del shock baja, la utilidad indirecta mide la utilidad alcanzable en equilibrio independientemente de las elecciones de consumo del individuo, lo cual es esperable que haya bajado dado que el máximo de utilidad de los consumidores es claramente menor ahora que P1 subió.

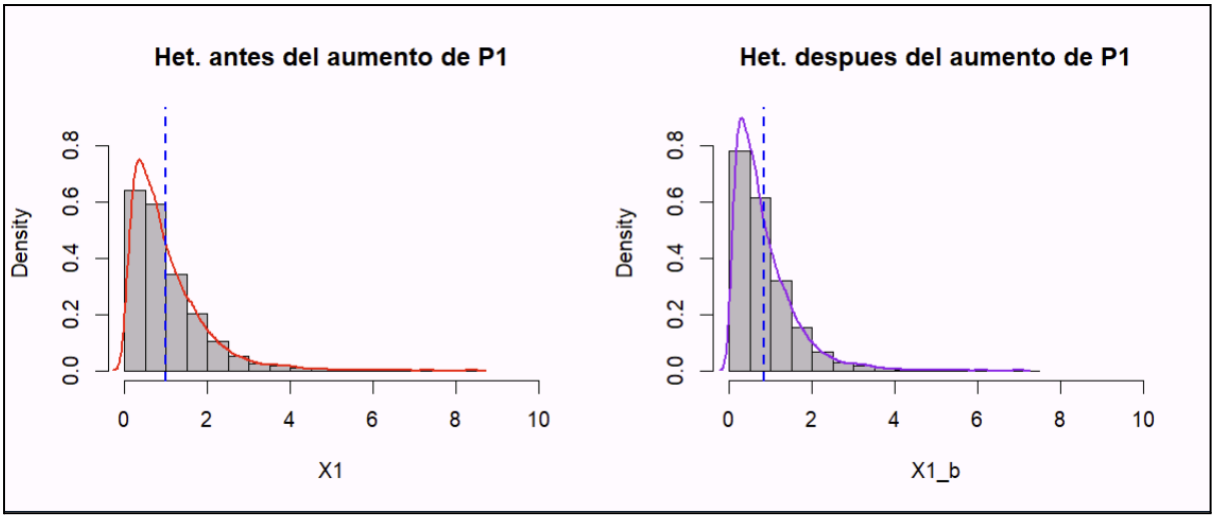


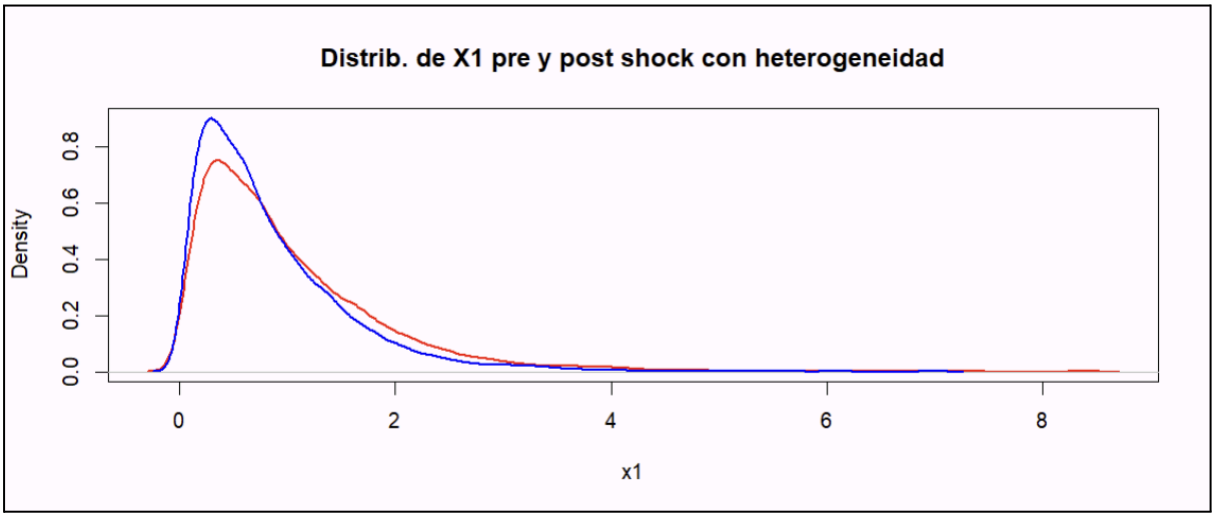
Inciso 7)

Heterogeneidad en las preferencias refiere a que cada hogar no tiene las mismas preferencias por cada bien como antes que arbitrariamente elegimos los valores de alpha1 y alpha2, sino que ahora seguirán una distribución beta, En el caso homogéneo, todos los consumidores destinan la misma proporción del ingreso a cada bien, por lo que el aumento del precio de P1 reduce de forma uniforme el consumo de X1.

En cambio, con heterogeneidad en preferencias, la respuesta al shock de precios es desigual: los hogares con alpha1 altos (más gusto por el bien 1) reducen más su consumo, mientras que los hogares con alpha1 bajos apenas se ven afectados. Esto genera una distribución más dispersa de X1 y una caída promedio mayor de la utilidad indirecta óptima.




Het. antes del aumento de P1
Density
0.0
0.2
0.4
0.6
0.8
0
2
4
6
8
10
X1
Het. despues del aumento de P1
Density
0.0
0.2
0.4
0.6
0.8
0
2
4
6
8
10
X1_b


Distrib. de X1 pre y post shock con heterogeneidad
Density
0.0
0.2
0.4
0.6
0.8
0
2
4
6
8
x1