

Revisión 2023

Iván Arriola - 55366796

Explicativo sobre la prueba:

Esta prueba es de carácter *INDIVIDUAL*, cada estudiante debe completarla y no interactuar con el resto de los estudiantes, si tienen dudas nos pueden preguntar a Mauro o a mí durante la prueba. Pueden consultar durante la prueba el libro del curso (<https://r4ds.had.co.nz/>), las slides, actividades y deberes del curso pero ningún otro material adicional. Sin embargo *NO ESTA PERMITIDO* utilizar otro material que el sugerido ni las respuestas, comentarios o sugerencias de otras personas que no sean los docentes del curso, cualquier apartamiento de esto invalidará la prueba.

Por favor completá tu nombre y CI en el YAML del archivo donde dice author: "NOMBRE Y CI:". Los archivos y la información necesaria para desarrollar la prueba se encuentran en Eva en la pestaña Revisión_2023. La revisión debe quedar en tu repositorio del curso GitHub en una carpeta que se llame Revision_2023 con el resto de las actividades y tareas del curso. Parte de los puntos de la prueba consisten en que tu revisión sea reproducible y tu repositorio de GitHub esté bien organizado como se indica en el Ejercicio 1.

La prueba dura 2 horas con posibilidad de extensión de media hora adicional.

Hay puntos parciales por lo que si algunas de las respuestas de código te queda incompleta agregá en el entorno de código `eval=FALSE` para que no se evalúe el mismo y evitar posibles errores.

EJERCICIO 1 (GitHub y entrega de la Revisión)

(10 puntos)

Esta pregunta es sobre el uso de GitHub y es la forma que van a entregar la prueba. Recordar que para que tengas la última versión de tu repositorio debes hacer `pull` a tu repositorio para no generar inconsistencias y antes de terminar subir tus cambios con `commit` y `push`.

- 1.a)** (1 Punto) En tu repositorio del curso creá una carpeta que se llame Revision_2023 (sin tilde).
- 1.b)** (1 Punto) Asegurate que soy colaboradora del mismo y si no lo soy mandame la invitación, mi usuario es natydasilva.
- 1.c)** (1 Punto) Subí el archivo .Rmd de esta revisión y los necesarios para reproducir el documento. Actualizá el repositorio regularmente durante prueba para que todo ande bien, asegurate al final de la prueba que el archivo Rmd compila.
- 1.d)** (3 Puntos) Asegurate que tu prueba sea reproducible cuando clone tu repositorio. Para ello deberás subir los archivos necesarios para la reproducibilidad (los datos van a ser necesarios). Poné los datos en la carpeta Datos de tu repositorio.
- 1.e)** (2 Puntos) A parte de subir el archivo al repositorio debés mandarme por correo (natalia.dasilva@fcea.edu.uy) el .Rmd y el .pdf de tu versión final. Escribime en el asunto del correo Rev23_STAT_NT y en el cuerpo tu usuario de GitHub para que sea más sencillo encontrar tu repositorio con la prueba.
- 1.f)** (2 Puntos) Presentá tu código de forma ordenada así como las respuestas a cada pregunta.

Ejercicio 2 (R Base)

(35 Puntos)

2.a) (8 Puntos) En cada uno de los siguientes casos tenés que arreglar algún problema común cuando subdividimos `data.frame`

```
mtcars[mtcars$cyl == 8, ]
```

```
# mtcars[mtcars$cyl = 8, ]
```

```
##          mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Hornet Sportabout 18.7  8 360.0 175 3.15 3.440 17.02  0  0   3   2
## Duster 360       14.3  8 360.0 245 3.21 3.570 15.84  0  0   3   4
## Merc 450SE       16.4  8 275.8 180 3.07 4.070 17.40  0  0   3   3
## Merc 450SL       17.3  8 275.8 180 3.07 3.730 17.60  0  0   3   3
## Merc 450SLC      15.2  8 275.8 180 3.07 3.780 18.00  0  0   3   3
## Cadillac Fleetwood 10.4  8 472.0 205 2.93 5.250 17.98  0  0   3   4
## Lincoln Continental 10.4  8 460.0 215 3.00 5.424 17.82  0  0   3   4
## Chrysler Imperial 14.7  8 440.0 230 3.23 5.345 17.42  0  0   3   4
## Dodge Challenger 15.5  8 318.0 150 2.76 3.520 16.87  0  0   3   2
## AMC Javelin      15.2  8 304.0 150 3.15 3.435 17.30  0  0   3   2
## Camaro Z28       13.3  8 350.0 245 3.73 3.840 15.41  0  0   3   4
## Pontiac Firebird 19.2  8 400.0 175 3.08 3.845 17.05  0  0   3   2
## Ford Pantera L   15.8  8 351.0 264 4.22 3.170 14.50  0  1   5   4
## Maserati Bora    15.0  8 301.0 335 3.54 3.570 14.60  0  1   5   8
```

```
mtcars[-c(1:5), ]
```

```
# mtcars[-1:5, ] # ayuda el - tiene que aparecer en tu prop
```

```
##          mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Valiant       18.1  6 225.0 105 2.76 3.460 20.22  1  0   3   1
## Duster 360    14.3  8 360.0 245 3.21 3.570 15.84  0  0   3   4
## Merc 240D     24.4  4 146.7  62 3.69 3.190 20.00  1  0   4   2
## Merc 230      22.8  4 140.8  95 3.92 3.150 22.90  1  0   4   2
## Merc 280      19.2  6 167.6 123 3.92 3.440 18.30  1  0   4   4
## Merc 280C     17.8  6 167.6 123 3.92 3.440 18.90  1  0   4   4
## Merc 450SE    16.4  8 275.8 180 3.07 4.070 17.40  0  0   3   3
## Merc 450SL    17.3  8 275.8 180 3.07 3.730 17.60  0  0   3   3
## Merc 450SLC   15.2  8 275.8 180 3.07 3.780 18.00  0  0   3   3
## Cadillac Fleetwood 10.4  8 472.0 205 2.93 5.250 17.98  0  0   3   4
## Lincoln Continental 10.4  8 460.0 215 3.00 5.424 17.82  0  0   3   4
## Chrysler Imperial 14.7  8 440.0 230 3.23 5.345 17.42  0  0   3   4
## Fiat 128      32.4  4  78.7  66 4.08 2.200 19.47  1  1   4   1
## Honda Civic   30.4  4  75.7  52 4.93 1.615 18.52  1  1   4   2
## Toyota Corolla 33.9  4  71.1  65 4.22 1.835 19.90  1  1   4   1
## Toyota Corona 21.5  4 120.1  97 3.70 2.465 20.01  1  0   3   1
## Dodge Challenger 15.5  8 318.0 150 2.76 3.520 16.87  0  0   3   2
## AMC Javelin   15.2  8 304.0 150 3.15 3.435 17.30  0  0   3   2
## Camaro Z28    13.3  8 350.0 245 3.73 3.840 15.41  0  0   3   4
## Pontiac Firebird 19.2  8 400.0 175 3.08 3.845 17.05  0  0   3   2
## Fiat X1-9     27.3  4  79.0  66 4.08 1.935 18.90  1  1   4   1
## Porsche 914-2 26.0  4 120.3  91 4.43 2.140 16.70  0  1   5   2
## Lotus Europa  30.4  4  95.1 113 3.77 1.513 16.90  1  1   5   2
## Ford Pantera L 15.8  8 351.0 264 4.22 3.170 14.50  0  1   5   4
## Ferrari Dino  19.7  6 145.0 175 3.62 2.770 15.50  0  1   5   6
## Maserati Bora  15.0  8 301.0 335 3.54 3.570 14.60  0  1   5   8
## Volvo 142E    21.4  4 121.0 109 4.11 2.780 18.60  1  1   4   2
```

```
mtcars[mtcars$cyl <= 6 ,] # mtcars[mtcars$cyl <= 6]
```

```
##      mpg  cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0    6 160.0 110 3.90 2.620 16.46 0 1    4    4
## Mazda RX4 Wag  21.0    6 160.0 110 3.90 2.875 17.02 0 1    4    4
## Datsun 710     22.8    4 108.0  93 3.85 2.320 18.61 1 1    4    1
## Hornet 4 Drive 21.4    6 258.0 110 3.08 3.215 19.44 1 0    3    1
## Valiant        18.1    6 225.0 105 2.76 3.460 20.22 1 0    3    1
## Merc 240D      24.4    4 146.7  62 3.69 3.190 20.00 1 0    4    2
## Merc 230       22.8    4 140.8  95 3.92 3.150 22.90 1 0    4    2
## Merc 280       19.2    6 167.6 123 3.92 3.440 18.30 1 0    4    4
## Merc 280C      17.8    6 167.6 123 3.92 3.440 18.90 1 0    4    4
## Fiat 128       32.4    4  78.7  66 4.08 2.200 19.47 1 1    4    1
## Honda Civic    30.4    4  75.7  52 4.93 1.615 18.52 1 1    4    2
## Toyota Corolla 33.9    4  71.1  65 4.22 1.835 19.90 1 1    4    1
## Toyota Corona  21.5    4 120.1  97 3.70 2.465 20.01 1 0    3    1
## Fiat X1-9      27.3    4  79.0  66 4.08 1.935 18.90 1 1    4    1
## Porsche 914-2  26.0    4 120.3  91 4.43 2.140 16.70 0 1    5    2
## Lotus Europa   30.4    4  95.1 113 3.77 1.513 16.90 1 1    5    2
## Ferrari Dino   19.7    6 145.0 175 3.62 2.770 15.50 0 1    5    6
## Volvo 142E     21.4    4 121.0 109 4.11 2.780 18.60 1 1    4    2
```

```
mtcars[mtcars$cyl == 4 | mtcars$cyl == 2, ] # el 4 y el 2 tienen que aparecer en tu propuesta de arregl
```

```
##      mpg  cyl  disp  hp drat   wt  qsec vs am gear carb
## Datsun 710     22.8    4 108.0  93 3.85 2.320 18.61 1 1    4    1
## Merc 240D      24.4    4 146.7  62 3.69 3.190 20.00 1 0    4    2
## Merc 230       22.8    4 140.8  95 3.92 3.150 22.90 1 0    4    2
## Fiat 128       32.4    4  78.7  66 4.08 2.200 19.47 1 1    4    1
## Honda Civic    30.4    4  75.7  52 4.93 1.615 18.52 1 1    4    2
## Toyota Corolla 33.9    4  71.1  65 4.22 1.835 19.90 1 1    4    1
## Toyota Corona  21.5    4 120.1  97 3.70 2.465 20.01 1 0    3    1
## Fiat X1-9      27.3    4  79.0  66 4.08 1.935 18.90 1 1    4    1
## Porsche 914-2  26.0    4 120.3  91 4.43 2.140 16.70 0 1    5    2
## Lotus Europa   30.4    4  95.1 113 3.77 1.513 16.90 1 1    5    2
## Volvo 142E     21.4    4 121.0 109 4.11 2.780 18.60 1 1    4    2
```

2.b) (5 Puntos) ¿Porqué `mtcars[1:30]` da un error? y ¿Cómo esto es distinto de `mtcars[1:30,]`?

Porque `mtcars` es una data frame de dos variables, porque el operador `[]` necesita de dos parametros, uno que indique filas y otro columnas deseadas. Cuando se ejecuta `mtcars[1:30]`, no se machean la cantidad de parametros necesarios con los pasados en el operador y eso da error. Sin embargo, cuando se ejecuta `mtcars[1:30,]`, estamos dejando explícitamente el segundo parametro como indefinido, lo cual el operador `[]` interpreta como que queremos todas las columnas de `mtcars` y ahí estarían definidas las filas y columnas deseadas y todo el código se ejecuta de la manera deseada.

2.c) (10 Puntos) Usando un for loop recodificó los valores del siguiente vector `c(2, 8, 5, 7, 2, 3, 1, NA)` al que deberás llamar `v_orig` como 1 es 'Azul', 2 es 'Verde', 3 es 'Amarillo', 4 es 'Rojo', 5 es 'Violeta', NA es NA y el resto como 'Otro'. El vector resultantes deberá ser `c("Verde", "Otro", "Violeta", "Otro", "Verde", "Amarillo", "Azul", NA)` al que debes guardar en el objeto `v_recod`

```
v_ord <- c(2, 8, 5, 7, 2, 3, 1, NA)
v_recod <- c()
for (v in v_ord) {
  if(is.na(v)) {
    v_recod <- c(v_recod, NA)
  }
}
```

```

} else if(v == 1) {
  v_recod <- c(v_recod, "Azul")
} else if(v == 2) {
  v_recod <- c(v_recod, "Verde")
} else if(v == 3) {
  v_recod <- c(v_recod, "Amarillo")
} else if(v == 4) {
  v_recod <- c(v_recod, "Rojo")
} else if(v == 5) {
  v_recod <- c(v_recod, "Violeta")
} else{
  v_recod <- c(v_recod, "Otro")
}
}

```

Ahora incorporará lo anterior en una función que sirva para cualquier vector numérico y si el vector no es numérico que frene e indique un error con sentido.

```

f_recod <- function(vector){
  if(!is.numeric(vector)) {
    stop('Esta función necesita valores numéricos y me diste un objeto de la clase: ', class(v)[1])
  }
  vector_recod <- c()
  for (v in vector) {
    if(is.na(v)) {
      vector_recod <- c(vector_recod, NA)
    } else if(v == 1) {
      vector_recod <- c(vector_recod, "Azul")
    } else if(v == 2) {
      vector_recod <- c(vector_recod, "Verde")
    } else if(v == 3) {
      vector_recod <- c(vector_recod, "Amarillo")
    } else if(v == 4) {
      vector_recod <- c(vector_recod, "Rojo")
    } else if(v == 5) {
      vector_recod <- c(vector_recod, "Violeta")
    } else{
      vector_recod <- c(vector_recod, "Otro")
    }
  }
  return(vector_recod)
}

```

2.d) (7 Puntos) Implementá una función que extraiga la diagonal principal de una matriz. Debes comprobar que se comporta como la función `diag(x)`, donde `x` es una matriz.

```

diagonalPrincipal <- function(m) {
  coordMin <- min(dim(m))
  diagonal <- c()
  for (i in 1:coordMin) {
    diagonal <- c(diagonal, m[i, i])
  }
  return(diagonal)
}

```

```
a <- sample(1:100, 3*3)
b <- sample(1:100, 7*2)

matrixA <- matrix(a, 3, 3)
matrixB <- matrix(b, 7, 2)

identical(diagonalPrincipal(matrixA), diag(matrixA))
```

```
## [1] TRUE
```

```
identical(diagonalPrincipal(matrixB), diag(matrixB))
```

```
## [1] TRUE
```

2.e) (5 Puntos) ¿Qué propiedad importante de `&&` hace funcionar a `f1`?

```
f1 <- function(x) {
  !is.null(x) && length(x) == 1 && x > 0
}
```

¿Qué es diferente con este código? Porqué este comportamiento no es deseable aquí?

```
f1 <- function(x) {
  !is.null(x) & length(x) == 1 & x > 0
}
```

Ejercicio 3 (tidyverse)

(35 Puntos)

En este ejercicio usaremos los datos del **Estudio Longitudinal de Bienestar en el Uruguay** (IECON) utilizados en la Actividad 12 de clase. Estos datos consisten en un relevamiento longitudinal representativo de los niños que concurren al sistema de educación primaria pública.

Se recoge información referente a múltiples dimensiones del bienestar de los niños en la muestra así como también del resto de los integrantes de sus hogares, entre los que se destacan:

- situación nutricional.
- habilidades cognitivas y no cognitivas.
- imaginación; razonamiento y sentimientos.
- actividades de ocio e interacción social.
- logros educativos.
- ingresos.
- calidad de la vivienda y bienestar subjetivo, entre otros.

La información es relevada en Olas, la primera fue en 2004 a 3000 niños del primer grado de escuela. En las distintas olas se los sigue en el tiempo a los mismos niños (por eso es longitudinal).

- De **personas**, con información referente al niño y personas del hogar donde reside.
- De **hogares**, con información más enfocada al niño, sus capacidades de relacionamiento, emociones, amigos, etc.

Vamos a usar datos de la tercer ola (2012). La base de personas contiene información sobre las personas del hogar del niño que salió en la muestra. Contiene la variable `nform` (número de formulario) que permite identificar al **hogar** (es una *key*). Para cada hogar existe la variable `nper` (número de persona) que le asigna un número a cada **persona del hogar**.

La base de hogares contiene información de los hogares.

Tiene la variable `nform` (número de formulario) que permite identificar al hogar y la variable `nper` (número de persona encuestada) que permite identificar a la persona del hogar que fue encuestada.

Todas las respuestas a este problema deberán ser respondidas con funciones de algún paquete de tidyverse

Todas las transformaciones y recodificaciones de variables que se hicieron en la Actividad 12 están en el conjunto de datos que vamos a trabajar para personas.

Trabajaremos solamente con las siguientes variables y sus nombres:

- Número de formulario: `nform`
- Número de persona: `nper`
- Sexo: `sexo`
- Situación conyugal: `sit.conyugal`
- Nivel educativo que cursa o cursó: `nivel.educ`
- Relacion de parentezco con el jefe/a: `parent.jefe`
- Edad: `edad`
- Sexo: `sexo`
- Ocupación principal: `dedicacion.p`
- Tipo de ocupación principal: `tipo.p`
- Tareas de su ocupación principal: `tareas.p`
- Jubilaciones pensiones: `jub.pen`
- Transferencia de otros hogares: `transf.pais`
- Transferencias externas: `transf.ext`
- Monto asignación familiar: `afam`
- Beneficio en especia: `benef.esp`

```
library(here)
```

```
## here() starts at T:/Curso_STAT_NT_Arriola
```

```
library(magrittr)
```

```
library(ggplot2)
```

3.a) (1 Punto) Usando la función `read_csv` del paquete `readr` cargá la base de datos `persona_recode_ed.csv` que se encuentra disponible en el EVA y a estos datos nombralos `personas`. Usá la función `read_csv()` y `here()` para asegurar la reproducibilidad.

```
(personas <- readr::read_csv(here("Datos", "personas_recode_ed.csv")))
```

```
## Rows: 10447 Columns: 15
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## chr (2): edad, dedicacion.p
```

```
## dbl (13): nform, nper, sit.conyugal, nivel.educ, parent.jefe, sexo, tipo.p, ...
```

```
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## # A tibble: 10,447 x 15
```

```
##   nform nper sit.conyugal nivel.educ parent.jefe edad      sexo dedicacion.p
```

```
##   <dbl> <dbl>         <dbl>         <dbl>         <dbl> <chr>    <dbl> <chr>
```

```
## 1     1     1           1           4           3 [12,15)    2 <NA>
```

```
## 2     1     2           1           2           1 [30, +)    1 trabaja
```

```
## 3     1     3           3           2           2 [30, +)    2 trabaja
```

```
## 4     1     4           3           4           3 [15, 18)   2 estudia
```

```
## 5     1     5           1           2           3 [6,12)     2 <NA>
```

```
## 6     1    13           1          NA           3 [0-3)      1 <NA>
```

```
## 7      2      1          1          5          3 [15, 18)      2 estudia
## 8      2      3          2          3          2 [30, +)      2 trabaja
## 9      2      4          1          3          3 [15, 18)      1 estudia
## 10     2     13          2          3          1 [30, +)      1 trabaja
## # i 10,437 more rows
## # i 7 more variables: tipo.p <dbl>, tareas.p <dbl>, jub.pen <dbl>,
## #   transf.pais <dbl>, transf.ext <dbl>, afam <dbl>, benef.esp <dbl>
```

3.b) (5 Puntos) La variable `sexo` tiene tres valores, recodificala usando la función `case_when` para que el 1 sea Hombre el 2 sea Mujer y 9 sea NS/NC (no sabe). Guardá los nuevos datos en `personas_reco`. La nueva variable nombrala `sexo_reco`

```
(
  personas <- personas %>%
    dplyr::mutate(
      sexo_reco = dplyr::case_when(
        sexo == 1 ~ "Hombre",
        sexo == 2 ~ "Mujer",
        sexo == 9 ~ "NS/NC"
      )
    )
)

## # A tibble: 10,447 x 16
##   nform nper sit.conyugal nivel.educ parent.jefe edad      sexo dedicacion.p
##   <dbl> <dbl>      <dbl>      <dbl>      <dbl> <chr>    <dbl> <chr>
## 1     1     1          1          4          3 [12,15)      2 <NA>
## 2     1     2          1          2          1 [30, +)      1 trabaja
## 3     1     3          3          2          2 [30, +)      2 trabaja
## 4     1     4          3          4          3 [15, 18)      2 estudia
## 5     1     5          1          2          3 [6,12)       2 <NA>
## 6     1    13          1         NA          3 [0-3)        1 <NA>
## 7     2     1          1          5          3 [15, 18)      2 estudia
## 8     2     3          2          3          2 [30, +)      2 trabaja
## 9     2     4          1          3          3 [15, 18)      1 estudia
## 10    2    13          2          3          1 [30, +)      1 trabaja
## # i 10,437 more rows
## # i 8 more variables: tipo.p <dbl>, tareas.p <dbl>, jub.pen <dbl>,
## #   transf.pais <dbl>, transf.ext <dbl>, afam <dbl>, benef.esp <dbl>,
## #   sexo_reco <chr>
```

3.c) (7 Puntos) Usando funciones de `dplyr` respondé ¿Cuál es la proporción de personas según `sexo_reco`?

```
(
  personas %>% dplyr::select(sexo_reco) %>% dplyr::group_by(sexo_reco) %>%
    dplyr::summarise(
      prop = dplyr::n() / dplyr::n()
    )
)

## # A tibble: 3 x 2
##   sexo_reco prop
##   <chr>      <dbl>
## 1 Hombre      1
## 2 Mujer       1
## 3 NS/NC       1
```

Reporté una tabla (con `xtable`), La tabla debe contener tres columnas (Sexo, Conteo y Proporción). Guardé el objeto generado con nombre `tabla`. Recordé poner en el chunk de código `results='asis'` y adentro del entorno `options(xtable.comment = FALSE)` para que salga la tabla en el pdf.

3.d) (5 Puntos) Nos gustaría tener la variable **departamento** que indica el departamento al cuál pertenece el hogar de la muestra así como la variable **a4** que indica propiedad de la vivienda y **a4a** que indica el monto del alquiler.

Tenemos esta información en la **base de Hogares** y una variable de mapeo que es **nform**, la cuál indica el **número de formulario** y vale lo mismo para todos los miembros del hogar.

Nos interesa mantener todas las observaciones de la base de personas, y “pegarle” el departamento, **a4** y **a4a**.

La base de hogares es `hogares_reco.csv` con alguna función de `dplyr` relacioné estas dos `data.frames` y al nuevo objeto nombralo `presonas_hogar`

```
(hogares <- readr::read_csv(here("Datos", "hogares_reco.csv")))
```

```
## Rows: 2151 Columns: 397
## -- Column specification -----
## Delimiter: ","
## chr (302): dpto_cod, fecha, a1, a2, a3, a4, a4a, a5, a8, a92, a93, a96, a97,...
## dbl (95): nform, nperenc, codigo, a4b, a5a, a6, a7, a91, a94, a95, a917, b5...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## # A tibble: 2,151 x 397
##   nform nperenc dpto_cod fecha codigo a1 a2 a3 a4 a4a a4b a5
##   <dbl> <dbl> <chr> <chr> <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1 1 3 Montevi~ 07 0~ 10 2 1 1995 2 <NA> NA 1
## 2 2 3 Montevi~ 15 0~ 10 1 1 1994 7 <NA> NA 1
## 3 3 3 Montevi~ 15 0~ 10 2 1 1998 7 <NA> NA 1
## 4 4 3 Montevi~ 12 0~ 17 2 1 1995 7 <NA> NA 1
## 5 5 3 Montevi~ 26 D~ 0 2 1 1994 7 <NA> NA 1
## 6 6 3 Montevi~ 16 N~ 1 1 3 2005 2 <NA> NA 1
## 7 8 2 Montevi~ 09 0~ 10 1 3 2000 7 <NA> NA 1
## 8 9 3 Montevi~ 22 S~ 15 2 1 1999 2 <NA> NA 1
## 9 10 3 Montevi~ 04 0~ 15 1 1 2001 4 <NA> NA 1
## 10 11 3 Montevi~ 22 S~ 15 1 1 2004 4 <NA> NA 1
## # i 2,141 more rows
## # i 385 more variables: a5a <dbl>, a6 <dbl>, a7 <dbl>, a8 <chr>, a91 <dbl>,
## # a92 <chr>, a93 <chr>, a94 <dbl>, a95 <dbl>, a96 <chr>, a97 <chr>,
## # a98 <chr>, a99 <chr>, a910 <chr>, a911 <chr>, a912 <chr>, a913 <chr>,
## # a914 <chr>, a915 <chr>, a916 <chr>, a917 <dbl>, a918 <chr>, a919 <chr>,
## # b1 <chr>, b2 <chr>, b3a <chr>, b3b <chr>, b4 <chr>, b5 <dbl>, b6 <dbl>,
## # b7 <chr>, b8 <chr>, b9a <chr>, b9b <chr>, b10 <chr>, b11 <chr>, ...
```

```
(
  personas_hogar <- hogares %>% dplyr::select(nform, dpto_cod, a4, a4a) %>%
    dplyr::right_join(personas, by="nform")
)
```

```
## # A tibble: 10,447 x 19
##   nform dpto_cod a4 a4a nper sit.conyugal nivel.educ parent.jefe edad
##   <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr>
## 1 1 Montevideo 2 <NA> 1 1 4 3 [12,1~
## 2 1 Montevideo 2 <NA> 2 1 2 1 [30, ~
```



```
## 3      1 Montevideo 2      <NA>      3      3      2      2 [30, ~
## 4      1 Montevideo 2      <NA>      4      3      4      3 [15, ~
## 5      1 Montevideo 2      <NA>      5      1      2      3 [6,12)
## 6      1 Montevideo 2      <NA>     13      1      NA      3 [0-3)
## 7      2 Montevideo 7      <NA>      1      1      5      3 [15, ~
## 8      2 Montevideo 7      <NA>      3      2      3      2 [30, ~
## 9      2 Montevideo 7      <NA>      4      1      3      3 [15, ~
## 10     2 Montevideo 7      <NA>     13      2      3      1 [30, ~
## # i 10,437 more rows
## # i 10 more variables: sexo <dbl>, dedicacion.p <chr>, tipo.p <dbl>,
## #   tareas.p <dbl>, jub.pen <dbl>, transf.pais <dbl>, transf.ext <dbl>,
## #   afam <dbl>, benef.esp <dbl>, sexo_reco <chr>
```

3.e) (7 Puntos) Recodificá la variable a4 (propiedad de la vivienda) de `personas_hogar` usando `case_match` guardá los datos en el objeto `personas_recode` como sigue

- 1 prop_vivter_pagando
- 2 prop_vivter
- 3 prop_viv_pagando
- 4 prop_viv
- 5 alquiler
- 6 ocupa_dep
- 7 ocupa_gr
- 8 ocupa_noper
- 9 pension
- 10 otro
- ns/nc "NA"

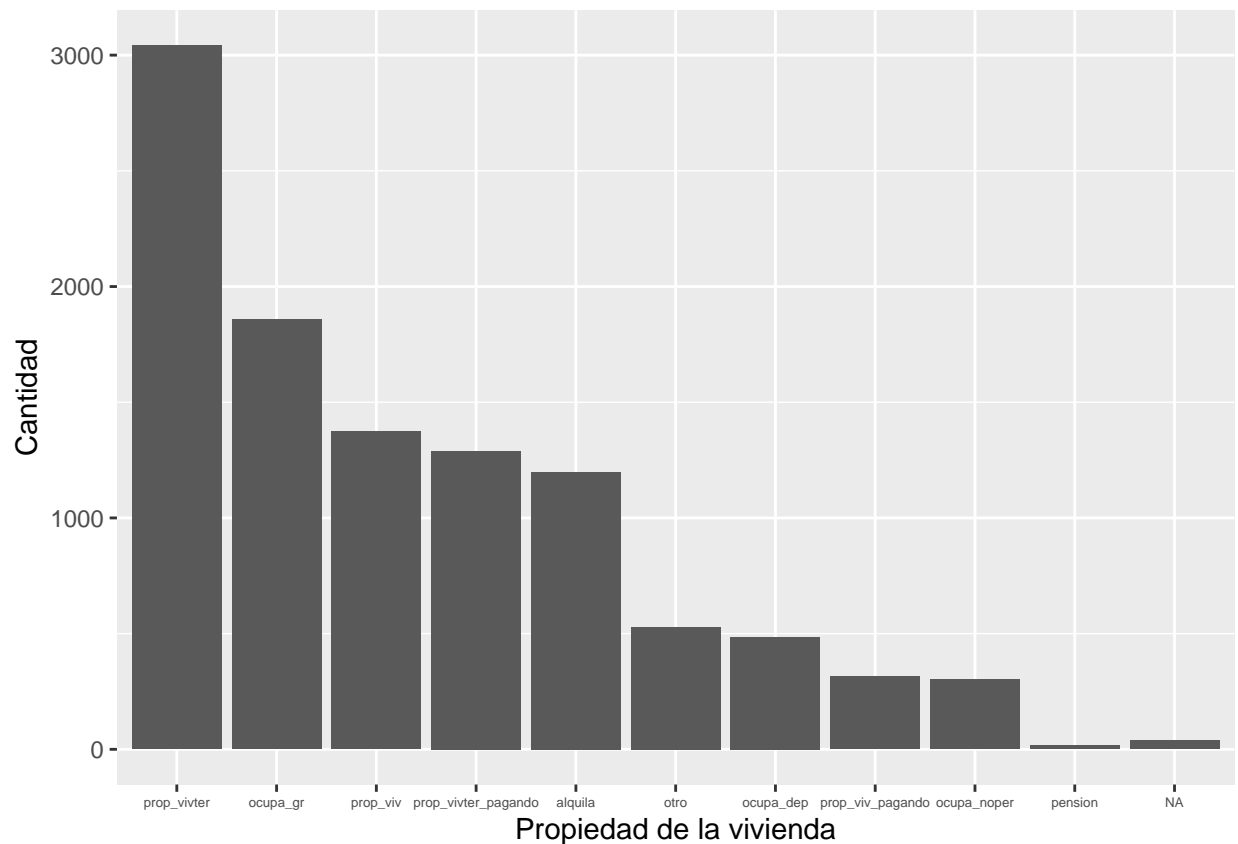
```
(
personas_recode <- personas_hogar %>%
  dplyr::mutate(
    a4 = dplyr::case_match(a4,
      "1" ~ "prop_vivter_pagando",
      "2" ~ "prop_vivter",
      "3" ~ "prop_viv_pagando",
      "4" ~ "prop_viv",
      "5" ~ "alquiler",
      "6" ~ "ocupa_dep",
      "7" ~ "ocupa_gr",
      "8" ~ "ocupa_noper",
      "9" ~ "pension",
      "10" ~ "otro",
      "ns/nc" ~ NA,
    )
  )
)
```

```
## # A tibble: 10,447 x 19
##   nform dpto_cod a4      a4a      nper sit.conyugal nivel.educ parent.jefe edad
##   <dbl> <chr>    <chr> <chr> <dbl>      <dbl>      <dbl>      <dbl> <chr>
## 1      1 Montevideo prop_~ <NA>      1      1      4      3 [12,~
## 2      1 Montevideo prop_~ <NA>      2      1      2      1 [30,~
## 3      1 Montevideo prop_~ <NA>      3      3      2      2 [30,~
## 4      1 Montevideo prop_~ <NA>      4      3      4      3 [15,~
## 5      1 Montevideo prop_~ <NA>      5      1      2      3 [6,1~
## 6      1 Montevideo prop_~ <NA>     13      1      NA      3 [0-3)
```

```
## 7      2 Montevideo ocupa~ <NA>      1      1      5      3 [15,~
## 8      2 Montevideo ocupa~ <NA>      3      2      3      2 [30,~
## 9      2 Montevideo ocupa~ <NA>      4      1      3      3 [15,~
## 10     2 Montevideo ocupa~ <NA>     13      2      3      1 [30,~
## # i 10,437 more rows
## # i 10 more variables: sexo <dbl>, dedicacion.p <chr>, tipo.p <dbl>,
## #   tareas.p <dbl>, jub.pen <dbl>, transf.pais <dbl>, transf.ext <dbl>,
## #   afam <dbl>, benef.esp <dbl>, sexo_reco <chr>
```

Usando `ggplot2` hice un gráfico de barras con la distribución de dicha variable indicando la más frecuente y en caso que las etiquetas no se vean bien proponé alguna forma de mejorarlo visualmente.

```
personas_recode %>% ggplot(
  aes(forcats::fct_infreq(a4))
) +
  geom_bar() +
  scale_y_continuous(name = "Cantidad") +
  scale_x_discrete(name="Propiedad de la vivienda") +
  theme(
    axis.text.x = element_text( size = 5)
  )
```



3.f) (10 Puntos) Mediante un gráfico describí como se distribuye la variable `a4a` por departamento en distintos paneles (4 columnas y dos filas), notá que deberá convertirla a numérica y no considerar los NA.

1. Cambiá el nombre de los ejes apropiadamente.
2. Cambiar la paleta de colores a una paleta apropiada según el tipo de variable, usando el paquete `paletteer` y la función `scale_color_paletteer_d`

3. Cambiá el nombre de la leyenda de color a “Depto” usando las función de tipo `scale_aes_type` apropiada.
4. Cambiá la posición de la leyenda en la parte inferior del gráfico usando `theme`, el título del eje x e y de color `darkgray` y tamaño 12, el título y texto de la leyenda también de tamaño 12 y con color `darkgray`.
5. Incluí un caption a tu figura y una descripción de lo que ves en ella, algo interesante que surja del gráfico.

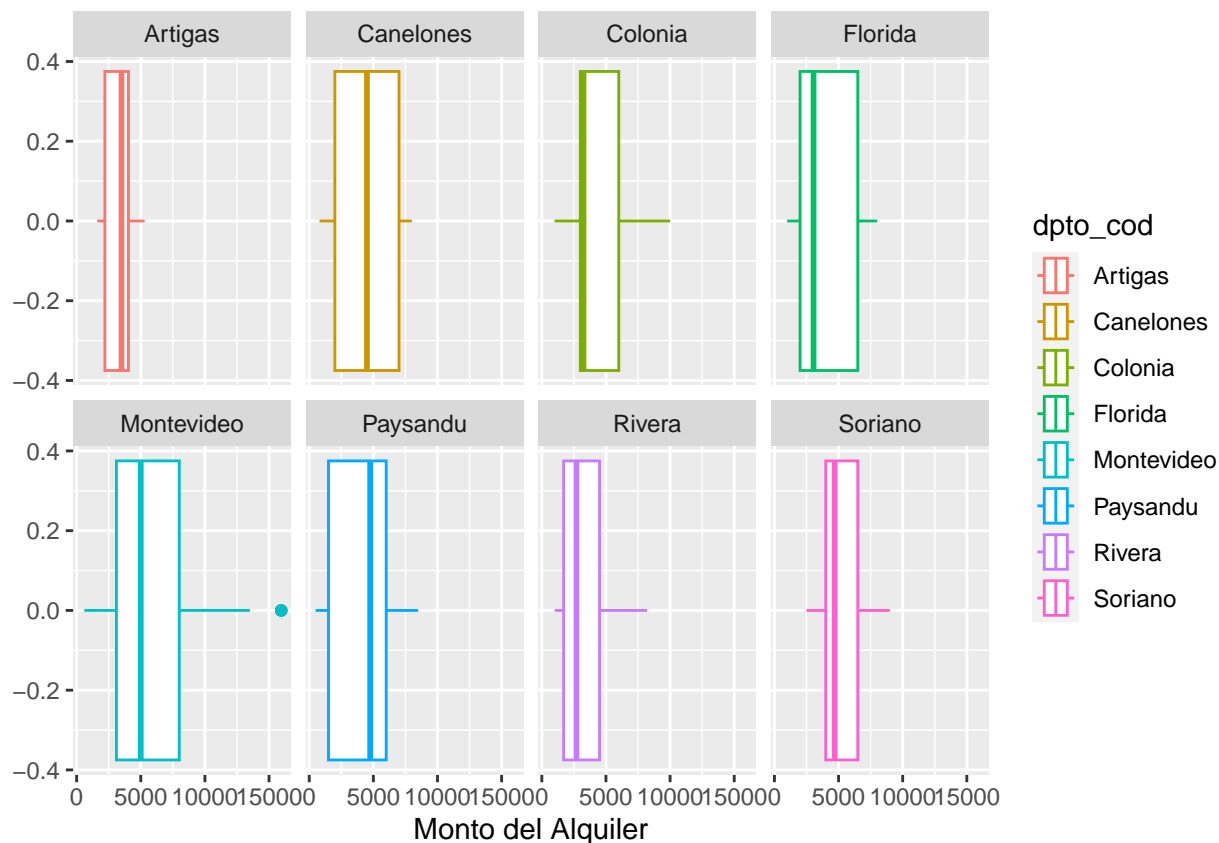
```
library(paletteer)
personas_recode %>% dplyr::select(dpto_cod, a4a) %>%
  dplyr::filter(!is.na(a4a)) %>% dplyr::mutate(a4a=as.numeric(a4a)) %>%
  ggplot() +
  geom_boxplot(
    aes(a4a,
        color = dpto_cod),

  ) +
  facet_wrap(~ dpto_cod, nrow = 2) +
  scale_x_continuous(name = "Monto del Alquiler") +
  theme(

)

## Warning: There was 1 warning in `dplyr::mutate()`.
## i In argument: `a4a = as.numeric(a4a)`.
## Caused by warning:
## ! NAs introducidos por coerción

## Warning: Removed 61 rows containing non-finite values (`stat_boxplot()`).
```



Ejercicio 4 (Popurrí)

(20 Puntos)

4.a) (5 Puntos) Indique cuál de las siguientes opciones NO corresponde a partes de un gráfico estadístico.

Seleccione una:

a) Datos, mapeo estético y sistema de coordenadas

b) Datos, mapeo estético y escalas

c) Datos, mapeo estético y modelo estadístico

d) Datos, mapeo estético y temas

4.b)(5 Puntos) Indica cuál de las opciones describe mejor lo que hace la función

```
func1 <-function(a){
  for(i in 1:a){
    b <- i^2
  }
  print(b)}
}
```

a. Calcula un número al cuadrado b. Calcula una secuencia de números al cuadrado

b. Calcula el producto de a por 2

c. Calcula una secuencia de números al cuadrado para los valores positivos hasta a

4.c) (10 Puntos) Hacer una función llamada `simulaplot` que tenga como argumentos `n` tamaño de muestra, `sigma` el desvío y `forma`. Esta función simula datos para distintos tamaños de muestra con distinta media

y desvío de una distribución normal usando `rnorm` y simula la misma cantidad de observaciones `n` de una distribución con `rgamma` cambiando el parámetro de forma. Como resultado de esta función se hace un diagrama de dispersión (usando `ggplot2`) con `x` los datos simulados normales y en `y` los datos simulados de la gamma. A su vez si el argumento es `sigma` es negativo la función no debe ser evaluada y debe imprimir el mensaje “sigma debe ser negativo”.

```
simulaplot <- function(n, sigma, forma){  
  if(sigma<0) stop("sigma no debe ser negativo")  
  
}
```