Master en Big Data. Fundamentos Matemáticos del Análisis de Datos (FMAD).

Tarea 2

Departamento de Matemática Aplicada

Curso 2021-22. Última actualización: 2021-09-24

# Instrucciones preliminares

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.4 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.1 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(ggplot2)  
library(nycflights13)

* Empieza abriendo el proyecto de RStudio correspondiente a tu repositorio personal de la asignatura.
* En todas las tareas tendrás que repetir un proceso como el descrito en la sección *Repite los pasos Creando un fichero Rmarkdown para esta práctica* de la *Práctica00*. Puedes releer la sección *Practicando la entrega de las Tareas* de esa misma práctica para recordar el procedimiento de entrega.

# Ejercicio 1. Simulando variables aleatorias discretas.

**Apartado 1:** La variable aleatoria discreta tiene esta tabla de densidad de probabilidad (es la variable que se usa como ejemplo en la Sesión ):

$$
\begin{array}{|c|c|c|c|c|c|c|}
\hline
\text{valor de }X1 & 0 & 1 & 2 & 3 \\
\hline
\text{Probabilidad de ese valor }P(X = x\_i) & \dfrac{64}{125} &
\dfrac{48}{125}& \dfrac{12}{125} & \dfrac{1}{125}\rule{0mm}{6mm} \\[3mm]
\hline
\end{array}
$$

Calcula la media y la varianza teóricas de esta variable.

X1=c(0:3)  
pX1=c(64/125,48/125,12/125,1/125)  
(tableX1=data.frame(X1,pX1=pX1))

## X1 pX1  
## 1 0 0.512  
## 2 1 0.384  
## 3 2 0.096  
## 4 3 0.008

#Media teórica  
(mu\_t=sum(X1\*pX1))

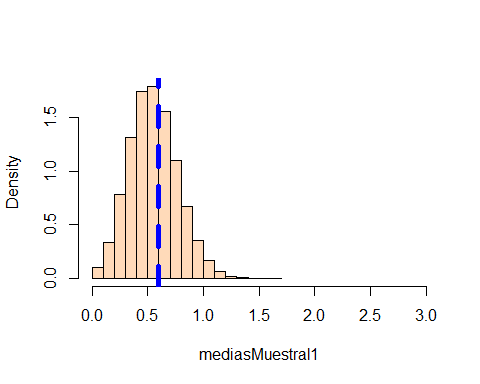
## [1] 0.6

#Varianza teórica  
(var\_t=sum(((X1-mu\_t)^2)\*pX1))

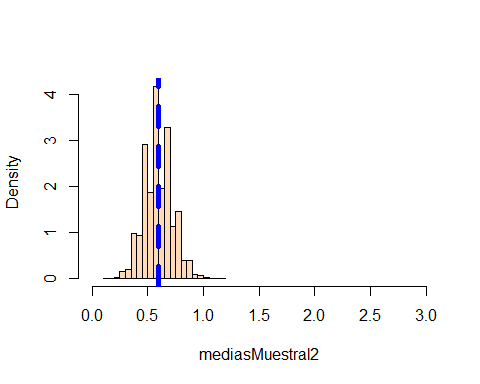
## [1] 0.48

**Apartado 2:** Combina sample con replicate para simular cien mil muestras de tamaño 10 de esta variable . Estudia la distribución de las medias muestrales como hemos hecho en ejemplos previos, ilustrando con gráficas la distribución de esas medias muestrales. Cambia después el tamaño de la muestra a 30 y repite el análisis.

set.seed(2021)  
k=100000  
n=10  
mediasMuestral1 = replicate(k, {  
 muestra = sample(X1, n, replace = TRUE,prob=pX1)  
 mean(muestra)  
})  
  
#Histograma de mediamuestral1 (tamaño 10)  
hist(mediasMuestral1, breaks = 20, main="",  
 col="peachpuff", probability = TRUE, xlim=range(X1))  
abline(v = mu\_t, lty=2, lwd=5, col="blue")



m=30  
mediasMuestral2 = replicate(k, {  
 muestra = sample(X1, m, replace = TRUE,prob=pX1)  
 mean(muestra)  
})  
  
#Histograma de mediamuestral1 (tamaño 30)  
hist(mediasMuestral2, breaks = 20, main="",  
 col="peachpuff", probability = TRUE, xlim=range(X1))  
abline(v = mu\_t, lty=2, lwd=5, col="blue")



**Apartado 3:** La variable aleatoria discreta tiene esta tabla de densidad de probabilidad:

$$
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{valor de }X2 & 0 & 1 & 2 \\
\hline
\text{Probabilidad de ese valor }P(X = x\_i) & \dfrac{1}{2} &
\dfrac{1}{4}& \dfrac{1}{4}\rule{0mm}{6mm} \\[3mm]
\hline
\end{array}
$$

Suponemos que y son independientes. ¿Qué valores puede tomar la suma ? ¿Cuál es su tabla de probabilidad?

X2=c(0:2)  
pX2=c(1/2,1/4,1/4)  
(tableX2=data.frame(X2,pX2))

## X2 pX2  
## 1 0 0.50  
## 2 1 0.25  
## 3 2 0.25

#Media teórica  
(mu\_t1=sum(X2\*pX2))

## [1] 0.75

#Varianza teórica  
(var\_t1=sum(((X2-mu\_t1)^2)\*pX2))

## [1] 0.6875

(pos=merge(tableX1$X1,tableX2$X2) %>%  
   
   
#Valores de la suma X1 +X2 con tabla de probabilidad   
mutate(posSum=x+y,prob=rep(tableX1$pX1,times=3)\*rep(tableX2$pX2,each=4)))

## x y posSum prob  
## 1 0 0 0 0.256  
## 2 1 0 1 0.192  
## 3 2 0 2 0.048  
## 4 3 0 3 0.004  
## 5 0 1 1 0.128  
## 6 1 1 2 0.096  
## 7 2 1 3 0.024  
## 8 3 1 4 0.002  
## 9 0 2 2 0.128  
## 10 1 2 3 0.096  
## 11 2 2 4 0.024  
## 12 3 2 5 0.002

pos$posSum=as.factor(pos$posSum)  
  
pos %>%  
 group\_by(posSum) %>%  
 summarise(prob=sum(prob))

## # A tibble: 6 x 2  
## posSum prob  
## <fct> <dbl>  
## 1 0 0.256  
## 2 1 0.32   
## 3 2 0.272  
## 4 3 0.124  
## 5 4 0.026  
## 6 5 0.002

**Apartado 4:** Calcula la media teórica de la suma . Después usa sample y replicate para simular cien mil *valores* de esta variable suma. Calcula la media de esos valores. *Advertencia:* no es el mismo tipo de análisis que hemos hecho en el segundo apartado.

#Media teórica de la suma X1 y X2  
(mu\_t2=sum(as.numeric(paste(pos$posSum))\*pos$prob))

## [1] 1.35

set.seed(2021)  
k=100000  
n1=1  
  
mediasMuestral3 = replicate(k, {  
 muestraX1 = sample(X1, n1, replace = TRUE,prob=pX1)  
 muestraX2 = sample(X2, n1, replace = TRUE,prob=pX2)  
 mean(muestraX1) + mean(muestraX2)  
})  
  
#Media con k valores de medias muestrales  
(mediaFinal=mean(mediasMuestral3))

## [1] 1.35143

# Ejercicio 2. Datos limpios

* Descarga el fichero de este enlace

<https://gist.githubusercontent.com/fernandosansegundo/471b4887737cfcec7e9cf28631f2e21e/raw/b3944599d02df494f5903740db5acac9da35bc6f/testResults.csv>

* Este fichero contiene las notas de los alumnos de una clase, que hicieron dos tests cada semana durante cinco semanas. La tabla de datos no cumple los principios de *tidy data* que hemos visto en clase. Tu tarea en este ejercicio es explicar por qué no se cumplen y obtener una tabla de datos limpios con la misma información usando *tidyR*.  
  **Indicación:** lee la ayuda de la función separate de *tidyR*.

test=read\_csv("./data/datatestResults.csv")

## Rows: 200 Columns: 9

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (2): name, gender\_age  
## dbl (7): id, test\_number, week1, week2, week3, week4, week5

##   
## 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.

Para hacer la lista tidy, se han agrupado las columnas week1, week2… en una columna week que contiene unicamente el numero de la semana en cuestión. Además, se han separado los valores de sexo y edad en columnas diferentes.

testTidy=test %>%   
 pivot\_longer(week1:week5, names\_to = 'week') %>%   
 separate(gender\_age, c("gender", "age")) %>%   
 separate(week,c('x','week'), sep=4) %>%   
 mutate(x=NULL)   
  
#Aquí se cambian el número de semana y la edad de tipo chr a tipo numérico  
testTidy$week=as.numeric(testTidy$week)  
testTidy$age=as.numeric(testTidy$age)  
testTidy

## # A tibble: 1,000 x 7  
## name id gender age test\_number week value  
## <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Jacob 108 m 20 1 1 8  
## 2 Jacob 108 m 20 1 2 5  
## 3 Jacob 108 m 20 1 3 7  
## 4 Jacob 108 m 20 1 4 5  
## 5 Jacob 108 m 20 1 5 6  
## 6 Jacob 108 m 20 2 1 2  
## 7 Jacob 108 m 20 2 2 2  
## 8 Jacob 108 m 20 2 3 4  
## 9 Jacob 108 m 20 2 4 0  
## 10 Jacob 108 m 20 2 5 3  
## # ... with 990 more rows

# Ejercicio 3. Lectura de R4DS.

Contnuando con nuestra *lectura conjunta* de este libro, si revisas el índice verás que hemos cubierto (holgadamente en algún caso) el contenido de los Capítulos 6, 8, 9, 10 y 11. Todos esos Capítulos son relativamente ligeros. Por eso esta semana conviene detenerse un poco en la lectura de los Capítulos 7 y 12, que son los más densos en información. Y como motivación os proponemos un par de ejercicios, uno por cada uno de esos capítulos.

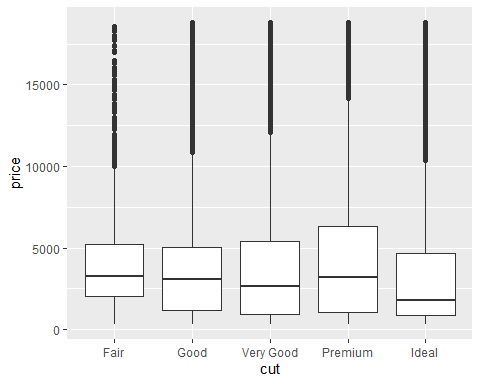
* Haz el [ejercicio 2 de la Sección 7.5.1.1 de R4DS](https://r4ds.had.co.nz/exploratory-data-analysis.html#exercises-17). Las ideas de esa sección son importantes para nuestro trabajo de las próximas sesiones.

Como se puede observar, hay una relacion existente entre el precio y el quilate, ya que, por lo general, a mayor kilate tendriamos un mayor precio.

ggplot(diamonds, aes(x = carat, y = price)) +  
 geom\_point()

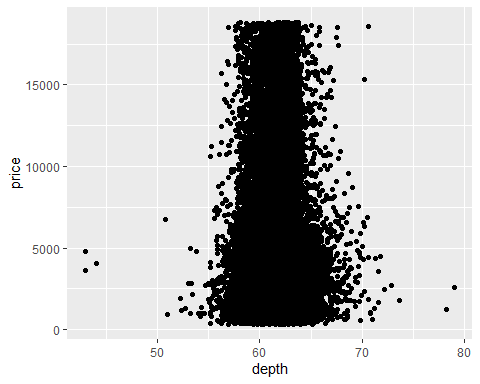
 Como se puede ver, el valor de la variable corte no influye en el precio del diamante y no hay tendencias observables.

ggplot(diamonds, aes(x = cut, y = price)) +  
 geom\_boxplot()

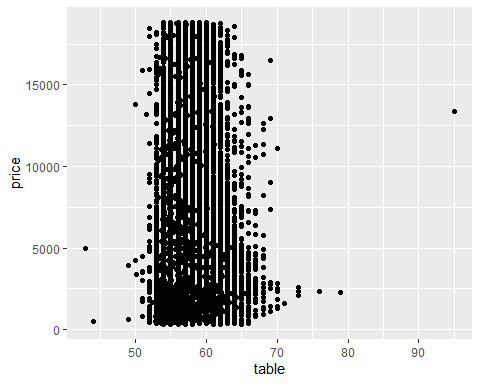


No hay relación ninguna entre depth y precio, para un valor cualquiera de depth tenemos un rango extremadamente amplio de precios.No hay tendencias observadas.

ggplot(diamonds, aes(x = depth, y = price)) +  
 geom\_point()

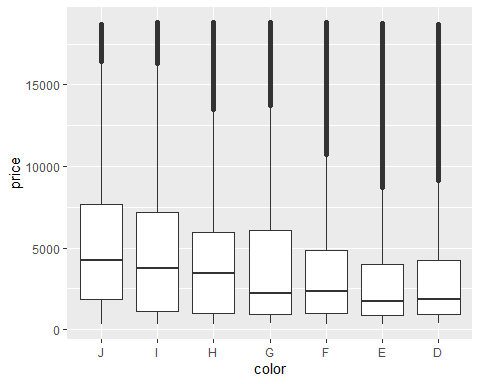
 No hay relación ninguna entre table y precio, para un valor cualquiera de table tenemos un rango extremadamente amplio de precios. No hay tendencias observadas.

ggplot(diamonds, aes(x = table, y = price)) +  
 geom\_point()

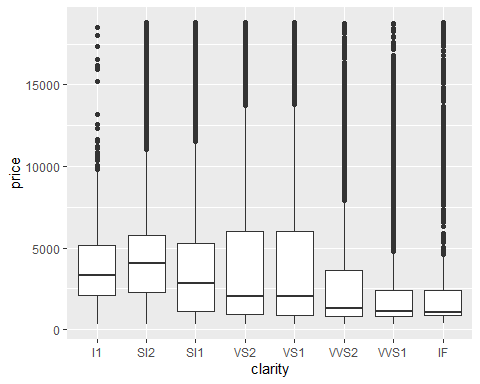


Primero ordenamos el color en orden ascendente en el que J es el menos preferible y D el más preferible. Tras esto llegamos a la conclusión de que no hay una relación clara entre precio y color, y solo podemos observar una mayor variabilidad cuanto más nos acercamos a J.

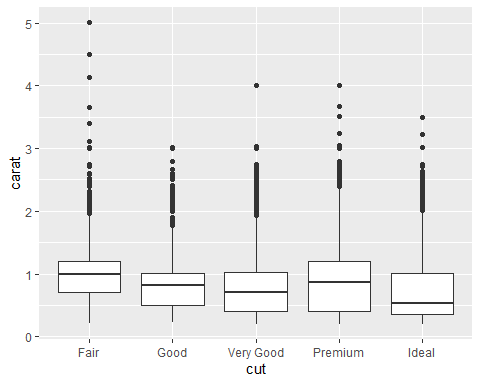
diamonds %>%  
 mutate(color = fct\_rev(color)) %>%  
 ggplot(aes(x = color, y = price)) +  
 geom\_boxplot()

 No hay relación entre claridad y precio, a destacar está que la variabilidad del precio aumenta en las categorías centrales de la claridad y disminuye hacia ambos extremos.

ggplot(data = diamonds) +  
 geom\_boxplot(mapping = aes(x = clarity, y = price))

 Buscamos correlación entre quilate (que tiene correlación con precio) y corte.Como se puede observar, la variabilidad del corte ‘fair’ es baja, y su mediana es la mayor de entre todos los cortes. Esto nos lleva a la conclusión de que en un corte ‘fair’ hay más valores de quilate alto. Como en la tabla quilate vs precio vemos que hay una relación lineal (mayor quilate, mayor precio generalmente), podemos concluir que a menor calidad (corte ‘fair’), mayor precio (mayor quilate).

ggplot(diamonds, aes(x = cut, y = carat)) +  
 geom\_boxplot()



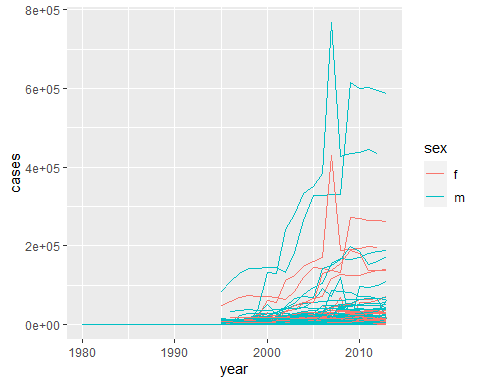
* Haz el [ejercicio 4 de la Sección 12.6.1 de R4DS](https://r4ds.had.co.nz/tidy-data.html#exercises-27). ¡Aprovecha el código previo de esa sección para trabajar con datos limpios!

who\_1=who %>%  
 pivot\_longer(  
 cols = new\_sp\_m014:newrel\_f65,   
 names\_to = "key",   
 values\_to = "cases",   
 values\_drop\_na = TRUE  
 ) %>%   
 mutate(  
 key = stringr::str\_replace(key, "newrel", "new\_rel")  
 ) %>%  
 separate(key, c("new", "var", "sexage")) %>%   
 select(-new, -iso2, -iso3) %>%   
 separate(sexage, c("sex", "age"), sep = 1)

Aquí generamos una gráfica de números de casos vs tiempo. La gráfica tiene en cuenta el país y el sexo de las personas afectadas. Como se puede ver, esto nos proporciona poca visión del impacto real por país, sobre todo en los más afectados.

who\_1 %>%  
 group\_by(country, year, sex) %>%  
 summarise(cases = sum(cases)) %>%  
 unite(country\_sex, country, sex, remove = FALSE) %>%  
 ggplot(aes(x = year, y = cases, group = country\_sex, colour = sex)) +  
 geom\_line()

## `summarise()` has grouped output by 'country', 'year'. You can override using the `.groups` argument.



#Aquí seleccionamos los países con más casos (>500000)  
(CountriesCases=who\_1 %>%   
 group\_by(country) %>%   
 summarise(n=sum(cases)) %>%   
 filter(n>500000) %>%   
 select(country))

## # A tibble: 16 x 1  
## country   
## <chr>   
## 1 Bangladesh   
## 2 Brazil   
## 3 China   
## 4 Democratic People's Republic of Korea  
## 5 Democratic Republic of the Congo   
## 6 Ethiopia   
## 7 India   
## 8 Indonesia   
## 9 Kenya   
## 10 Nigeria   
## 11 Pakistan   
## 12 Philippines   
## 13 Russian Federation   
## 14 South Africa   
## 15 United Republic of Tanzania   
## 16 Viet Nam

Aquí se grafican los casos según el sexo. Además, la separación de tablas dependiendo del país nos permite ver los datos con mucha más claridad.

who\_1 %>%   
 filter(country %in% CountriesCases$country) %>%   
 group\_by(country, year, sex) %>%  
 summarise(cases = sum(cases)) %>%   
 unite(country\_sex, country, sex, remove=FALSE) %>%   
 ggplot(aes(x=year, y=cases, group =country\_sex, colour=sex)) +  
 geom\_line()+  
 facet\_wrap(~country, scales='free\_y')

## `summarise()` has grouped output by 'country', 'year'. You can override using the `.groups` argument.

