Maestría en Computo Estadístico
Programación y análisis de algoritmos
Tarea 1
28 de agosto de 2020
Enrique Santibáñez Cortés
Repositorio de Git: Tarea 1, IE.

1. Asigna a una variable x el valor de 17. Posteriormente, crea un vector y con los valores $[2, 4, 6, 10, 100]$. Multiplica esos vectores por componente y guarda el resultado en un objeto z. Calcula la suma de todos los elementos en z.

RESPUESTA

```
# Creamos la variable x:
x <- 17

# Creamos el vector y:
y <- c(seq(2,6,2),10,100)

# Multiplicamos esos vectores y creamos el vector z:
z <- x*y

# Calculamos la suma de z:
sum(z)
```

```
## [1] 2074
```

Es decir, la suma total de z es 2074. ■

2. Define dos vectores con los siguientes datos: s incluye los strings "lun", "mar", "mier", "jueves", "viernes" y "sabado". El vector n incluye los valores $[90, 70, 30, 50, 5, 10]$. Une estos dos vectores de manera columnas en una matriz con 5 renglones y 2 columnas y guárdalo en un nuevo objeto llamado $datos_sem$.

RESPUESTA

```
# Definimos el vector s:
s <- c("lun", "mar", "mier", "jueves", "viernes","sabado")

# Definamos el vector n:
n <- c(90, 70, 30, 50, 5, 10)

# Creamos la matrix de tamaño 6x2:
datos_sem <- matrix(c(s,n),nrow = 6,ncol = 2)
datos_sem # imprimimos el resultado
```

```
##      [,1]      [,2]
## [1,] "lun"     "90"
## [2,] "mar"     "70"
## [3,] "mier"    "30"
## [4,] "jueves"  "50"
## [5,] "viernes" "5"
```

```
## [6,] "sabado"  "10"
```

3. Crea la siguiente data frame

Edad	sexo	altura	peso
21	m	181	69
35	f	173	58
829	m	171	75
2	e	166	60

Calcula el máximo y el mínimo en la columna de edad. Al parecer, hubo algunos problemas en la transcripción de la información. Genera una variable que contenga los resultados de la verificación lógica de edad debajo de 20 y arriba de 80. Usa esta variable para poner el valor de NA en las observaciones correspondientes. Crear el índice de masa corporal (IMC) IMC=Peso en kg/Altura en metros. Guarda los resultados en la variable BMI y agregala a la dataframe. Redondea los valores obtenidos.

RESPUESTA

```r
library(tidyverse) # Cargamos esta libraria, para ocupar ggplot, tidyr and dplyr.
library(latex2exp) # Legendas de las gráficas.
# Datos del dataframe:
edad <- c(21, 35, 829, 2)
sexo <- c("m", "f", "m", "e")
altura <- c(181, 173, 171, 166)
peso <- c(89, 58, 75, 60)
# Creamos el dataframe:
df_ejer3 <- data.frame(edad=edad, sexo=sexo, altura=altura, peso=peso)
df_ejer3
```

```
##   edad sexo altura peso
## 1   21    m    181   89
## 2   35    f    173   58
## 3  829    m    171   75
## 4    2    e    166   60
```

Calculamos el máximo de la columna edad:

```r
# máximo
max(df_ejer3$edad)
```

```
## [1] 829
```

Ahora, calculamos el mínimo:

```r
min(df_ejer3$edad)
```

```
## [1] 2
```

Validación de la variable edad conforme al intervalo (20, 80), donde $TRUE$: si la edad debajo de 20 o arriba de 80, $FALSE$: no cumple la condición anterior:

```r
# verificación de la edad:
df_ejer3 <- df_ejer3 %>%
  mutate(veri_edad=ifelse(edad<=20|edad>=80, T, F))
df_ejer3
```

```
##   edad sexo altura peso veri_edad
## 1   21    m    181   89     FALSE
## 2   35    f    173   58     FALSE
## 3  829    m    171   75      TRUE
## 4    2    e    166   60      TRUE
```

Agreamos las NaN en donde la edad esta fuera de rango del intervalo (20,80):

```r
# Agreamos las NaN en donde la edad esta fuera de rango:
df_ejer3 <- df_ejer3 %>%
  mutate(edad=ifelse(veri_edad, NaN, edad))
df_ejer3
```

```
##   edad sexo altura peso veri_edad
## 1   21    m    181   89     FALSE
## 2   35    f    173   58     FALSE
## 3  NaN    m    171   75      TRUE
## 4  NaN    e    166   60      TRUE
```

Creamos el índice de masa corporal (redondeando a 1 decimal):

```r
# Creamos la variable BMI= índice de masa corporal:
df_ejer3 <- df_ejer3 %>%
  mutate(BMI = round(peso/(altura/100),1))
df_ejer3
```

```
##   edad sexo altura peso veri_edad  BMI
## 1   21    m    181   89     FALSE 49.2
## 2   35    f    173   58     FALSE 33.5
## 3  NaN    m    171   75      TRUE 43.9
## 4  NaN    e    166   60      TRUE 36.1
```

4. Genera una secuencia de -5 a 5 en incrementos de 0.01. Grafique la función $Y = x^2$ donde X es la secuencia previamente generada. Compara la función a: $Y = -2 + x^2$, $y = 5x^2$?

RESPUESTA

Creamos la secuencia:

```r
x <- seq(-5, 5, 0.01)
```

Generamos un dataframe que contenga las tres funciónes, donde y_1: es la función $Y = x^2$, y_2: es la función $Y = -2 + x^2$ y y_3: es la función $Y = 5x^2$:

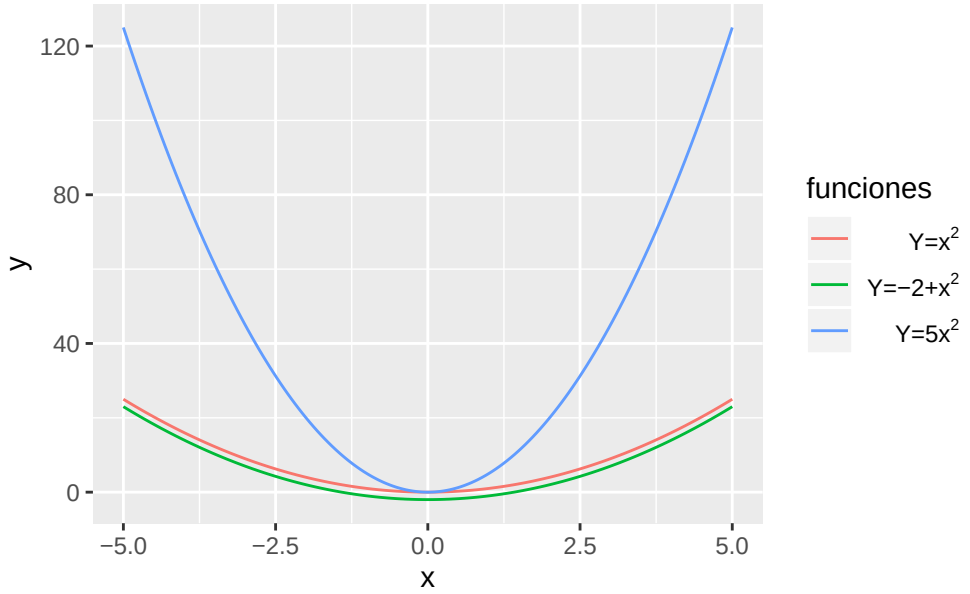
```r
# Creamos un dataframe
graficas <- data.frame(x=x)
# Creamos las 3 funciones:
graficas <- graficas %>%
  mutate(y_1 = x**2,
```

```
        y_2 =-2+x**2,
        y_3 = 5*(x**2))
# Modificamos el formato del data frame:
graficas_gat <- gather(graficas, key="funciones", value="y", 2:4)
head(graficas_gat)
```

```
##       x funciones       y
## 1 -5.00       y_1 25.0000
## 2 -4.99       y_1 24.9001
## 3 -4.98       y_1 24.8004
## 4 -4.97       y_1 24.7009
## 5 -4.96       y_1 24.6016
## 6 -4.95       y_1 24.5025
```

Graficamos las tres funciones:

```
ggplot(data=graficas_gat, aes(x=x, y=y, col=funciones))+
  geom_line()+
  scale_color_discrete(labels = unname(TeX(c("$Y=x^2", "$Y=-2+x^2","Y=5x^2") )))
```



Primero observemos que las tres funciones son parabolas por definición. Si comparamos $Y = x^2$ con $Y = -2 + x^2$ observamos que tiene la misma forma solo que esta trasladada hacía abajo 2 unidades en el eje y, ahora si la comparamos con $Y = 5^2$ observamos que esta función crece más rápido que la función $Y = x^2$ y este cambio es debido a como esta definida la función. ■

5. Carga el conjunto de datos "Boston" de la librería "MASS", que muestra los potenciales parámetros que influyen en los valores de las casas en los suburbios de la ciudad.

a. La mediana del valor de las casas ocupadas en miles está dado por la columna "medv". Obtenga los estadísticos de resumen y comentelos.

b. Muestra la relación entre valor de las casas(columna: medv) e indíce criminal (columna: crim) con un gráfico. Dibuje también una línea en el gráfico que muestre la relación.

RESPUESTA

Cargamos la librería y los datos:

```
library(MASS)
data("Boston")
```

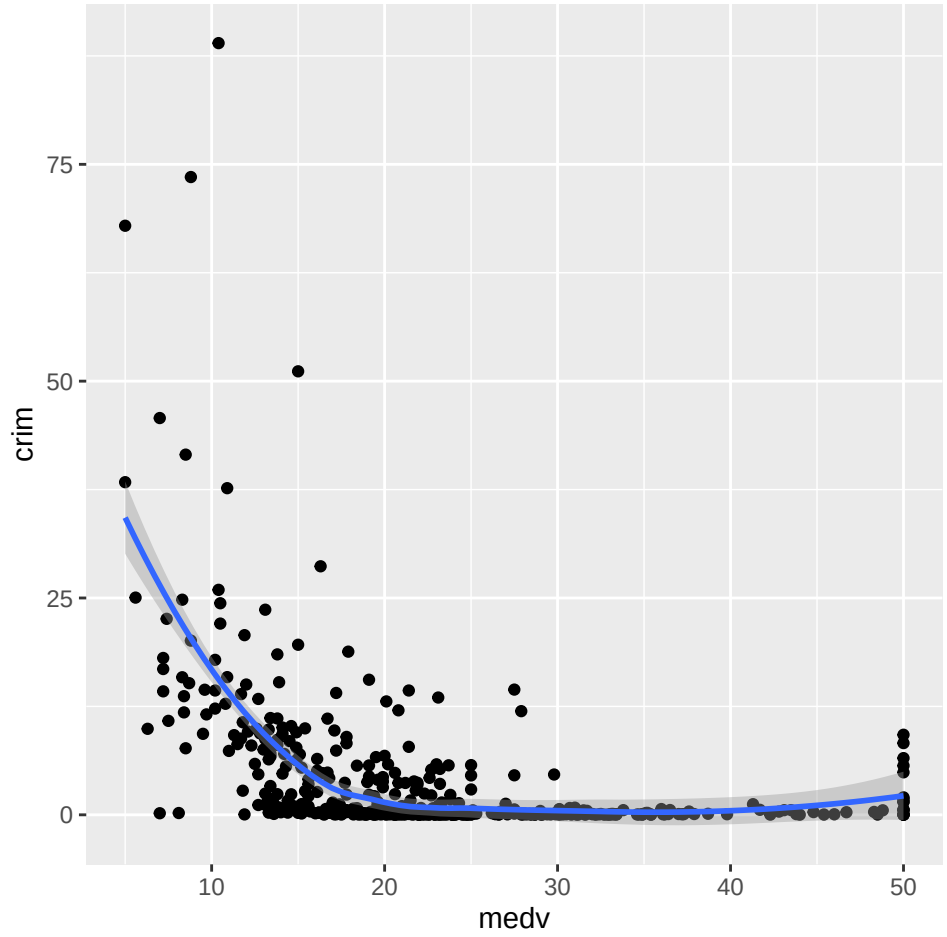
a. Calculamos los estadísticos de resumen de la mediana del valor de las casas:

```
summary(Boston$medv)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   17.02   21.20   22.53   25.00   50.00
```

```
ggplot(data=Boston, aes(x=medv, y=crim))+
  geom_point()+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Calculamos la correlación de estas dos variables:

```
# Correlación:
cor(Boston$crim, Boston$medv)
```

```
## [1] -0.3883046
```

6. Tenemos los datos de 100 billetes reales y 100 falsos. En la base bank2.dat se encuentran los datos de estos. Los primeros registros corresponden a los billetes reales y los segundos a los falso. Las variables son las siguientes:

- $X1$: Ancho,
- $X2$: Altura, medida desde el lado izquierdo
- $X3$: Altura, medida desde el lado derecho
- $X4$: Distancia del marco interior al borde inferior
- $X5$: Distancia del marco interior al borde superior
- $X6$: Tamaño de la diagonal.

Realice un análisis exploratorio donde se puedan observar las diferencias/similitudes entre los diferentes tipos de billetes. Incluya gráficas comparativas para ellos.

```
bank2 <- read.table("bank2.dat", quote="\"", comment.char="",
                    col.names = c("ancho", "altura_iz", "altura_der",
                                  "marco_inf", "marco_sup", "diagonal"))

# Etiquetamos los billetes:
bank2$veri <- c(rep("real", 100), rep("falso",100))

summary(bank2[bank2$veri=="real",])
```

```
##      ancho          altura_iz       altura_der      marco_inf
##  Min.   :213.8   Min.   :129.0   Min.   :129.0   Min.   : 7.200
##  1st Qu.:214.7   1st Qu.:129.7   1st Qu.:129.4   1st Qu.: 7.900
##  Median :215.0   Median :129.9   Median :129.7   Median : 8.250
##  Mean   :215.0   Mean   :129.9   Mean   :129.7   Mean   : 8.305
##  3rd Qu.:215.2   3rd Qu.:130.2   3rd Qu.:130.0   3rd Qu.: 8.800
##  Max.   :215.9   Max.   :131.0   Max.   :131.1   Max.   :10.400
##    marco_sup          diagonal         veri
##  Min.   : 7.700   Min.   :139.6   Length:100
##  1st Qu.: 9.775   1st Qu.:141.2   Class :character
##  Median :10.200   Median :141.5   Mode  :character
##  Mean   :10.168   Mean   :141.5
##  3rd Qu.:10.600   3rd Qu.:141.8
##  Max.   :11.700   Max.   :142.4
```

```
summary(bank2[bank2$veri!="real",])
```

```
##      ancho          altura_iz       altura_der      marco_inf
##  Min.   :213.9   Min.   :129.6   Min.   :129.3   Min.   : 7.40
##  1st Qu.:214.6   1st Qu.:130.1   1st Qu.:130.0   1st Qu.: 9.90
##  Median :214.8   Median :130.3   Median :130.2   Median :10.60
##  Mean   :214.8   Mean   :130.3   Mean   :130.2   Mean   :10.53
##  3rd Qu.:215.0   3rd Qu.:130.5   3rd Qu.:130.4   3rd Qu.:11.40
##  Max.   :216.3   Max.   :130.8   Max.   :131.1   Max.   :12.70
##    marco_sup        diagonal         veri
##  Min.   : 9.10   Min.   :137.8   Length:100
##  1st Qu.:10.68   1st Qu.:139.2   Class :character
##  Median :11.10   Median :139.5   Mode  :character
##  Mean   :11.13   Mean   :139.4
##  3rd Qu.:11.53   3rd Qu.:139.8
##  Max.   :12.30   Max.   :140.6
```

```
ggplot(data = bank2, aes(x=ancho, col=veri))+
  geom_density()
```


```
#ggplot(data = bank2, aes(x=altura_iz, col=veri))+
#  geom_density()

#ggplot(data = bank2, aes(x=altura_der, col=veri))+
#  geom_density()

#ggplot(data = bank2, aes(x=marco_inf, col=veri))+
#  geom_density()

#ggplot(data = bank2, aes(x=marco_sup, col=veri))+
#  geom_density()

#ggplot(data = bank2, aes(x=diagonal, col=veri))+
#  geom_density()
```