

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ciencias de la Computación e Informática
Diseño de Experimentos

Laboratorio 1

2025



Autor

Brandon Trigueros Lara C17899

David González Villanueva C13388

Parte A - Ejercicio R.Rproj

A1

1_1

Al ejecutar la línea 16

```
16 head(mtcars) # lines from start of data set
```

Obtenemos en la consola de R studio lo siguiente

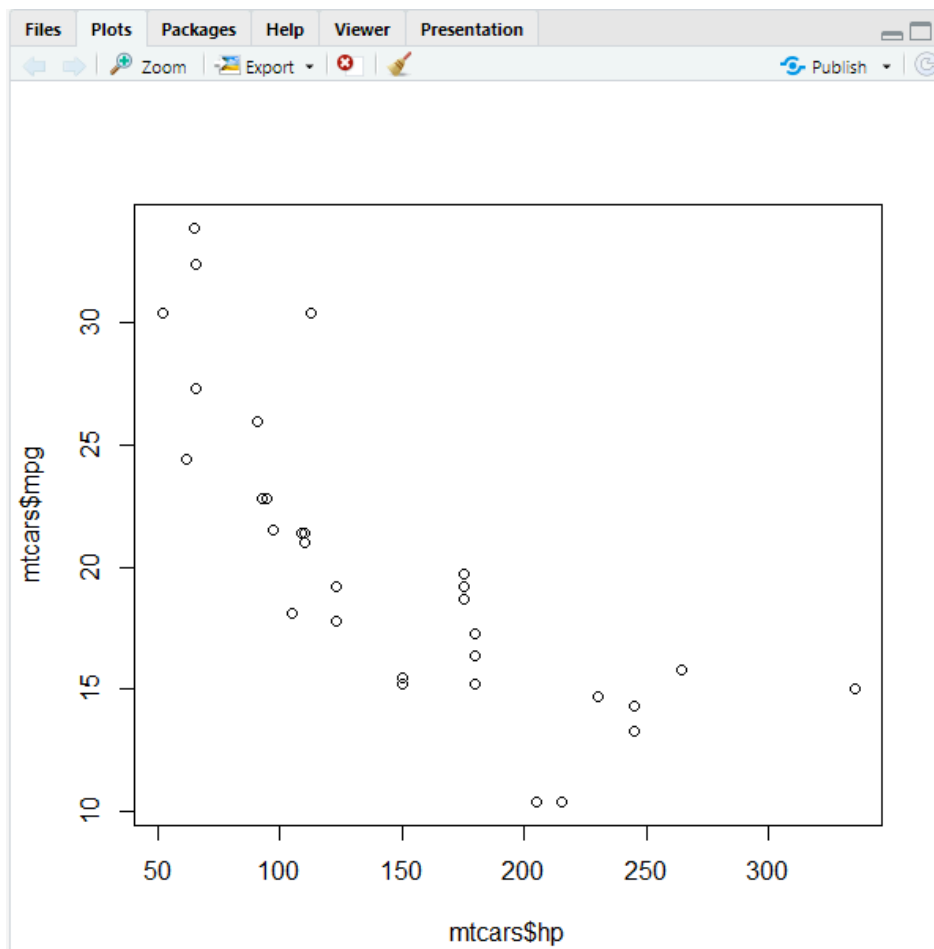
```
> head(mtcars) # lines from start of data set
      mpg  cyl  disp  hp  drat   wt   qsec  vs  am  gear  carb
Mazda RX4         21.0   6  160 110  3.90  2.620 16.46  0   1    4    4
Mazda RX4 Wag     21.0   6  160 110  3.90  2.875 17.02  0   1    4    4
Datsun 710        22.8   4  108  93  3.85  2.320 18.61  1   1    4    1
Hornet 4 Drive    21.4   6  258 110  3.08  3.215 19.44  1   0    3    1
Hornet Sportabout 18.7   8  360 175  3.15  3.440 17.02  0   0    3    2
Valiant           18.1   6  225 105  2.76  3.460 20.22  1   0    3    1
> |
```

Esto me muestra los primeros 6 registros del dataset con sus 11 columnas.

Con la línea 19

```
19 plot(mtcars$hp,mtcars$mpg)
```

Se genera un gráfico en la sección de plots



El gráfico es un scatterplot entre las variables mpg (miles per gallon) y hp (horse power).

Al ejecutar las líneas 22 a 26

```
22 LETTERS
23 letters
24 month.abb
25 month.name
26 pi
27
```

Lo cual nos imprime en la consola de R lo siguiente

```
> LETTERS
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
> month.abb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
> month.name
[1] "January" "February" "March" "April" "May" "June" "July" "August" "September" "October" "November" "December"
> pi
[1] 3.141593
> |
```

Vemos que R tiene variables incorporadas que podemos invocar (letras en mayúscula, letras en minúscula, meses abreviados, meses con nombre completo, valor de pi). Algunos son arreglos y también hay valores numéricos como PI.

1_3

La línea 30

```
30 lots.of.letters[,"letters"] # returns the "letters" column
```

Lo que imprime en consola es

```
letters
1      a
2      b
3      c
4      d
5      e
6      f
7      g
8      h
9      i
10     j
11     k
12     l
13     m
14     n
15     o
16     p
17     q
18     r
19     s
20     t
21     u
22     v
23     w
24     x
25     y
26     z
> |
```

Lo cual es la columna “letters” del data frame lots.of.letters.

La línea 31

```
31 lots.of.letters[3:8,2] # returns values from column 2 for rows 3 through 8
```

Me va a dar

```
> lots.of.letters[3:8,2] # returns values from column 2 for rows 3 through 8
[1] "c" "d" "e" "f" "g" "h"
> |
```

En el comando indicamos que queremos en el lado de las filas solo aquellas que van de la 3 a la 8 y la segunda columna. Por eso obtenemos de la columna 2 (letras en minúscula), de la tercera a la octava letra.

En la línea 32.

```
32 lots.of.letters[3:8,2:3]
```

Especifiqué un rango en tanto en el lado de las filas como en el lado de las columnas.

```
> lots.of.letters[3:8,2:3]
  letters position
3      c         3
4      d         4
5      e         5
6      f         6
7      g         7
8      h         8
> |
```

Por lo que obtengo la submatriz que conforma el rango de filas 3 a 8 y el rango de columnas 2 a 3.

En la línea 33

```
33 lots.of.letters[lots.of.letters$position > 10,"letters"]
```

Que genera

```
> lots.of.letters[lots.of.letters$position > 10,"letters"]
[1] "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
> |
```

Tenemos la columna de las letra minúsculas y en las filas tenemos una condición, recuperar aquellas cuya posición es mayor a 10.

En la línea 36

```
lots.of.letters[LETTERS=="R","letters"]
```

Tenemos básicamente lo mismo, una condición lógica usada como selector.

```
> lots.of.letters[LETTERS=="R","letters"]
[1] "r"
```

Similar para la línea 37

```
37 lots.of.letters[LETTERS=="R" | LETTERS=="T","letters"]
```

Que me da

```
> lots.of.letters[LETTERS=="R" | LETTERS=="T","letters"]
[1] "r" "t"
```

Solo que esta vez tenemos un '|' que significa operador or.

1_4

La línea 59

```
59 paste(I.am.character,"adishes")
```

Me da

```
> paste(I.am.character,"adishes")
[1] "I like R adishes"
```

Mientras que la línea

```
60 paste0(I.am.character,"adishes")
```

Me da

```
60 paste0(I.am.character,"adishes")
```

Podemos ver que paste y paste0 se diferencian en cómo manejan el **separador entre elementos** (sep). paste() separa elementos con un espacio (sep = " ") mientras que paste0() no agrega separador (sep = "").

1_5

Al ejecutar la línea 23

```
23 paste(I.am.a.vector,I.am.also.a.vector) # maybe not what you expected
```

Obtenemos la siguiente salida:

```
> paste(I.am.a.vector,I.am.also.a.vector) # maybe not what you expected
[1] "twas did" "brillig gyre" "and and" "the gimble" "slithey in" "toves the" "twas wabes"
```

Luego al ejecutar la línea 24

```
24 c(I.am.a.vector,I.am.also.a.vector) # end for end combination.
```

Obtenemos:

```
> c(I.am.a.vector,I.am.also.a.vector) # end for end combination.
[1] "twas" "brillig" "and" "the" "slithey" "toves" "did" "gyre" "and" "gimble" "in" "the" "wabes"
```

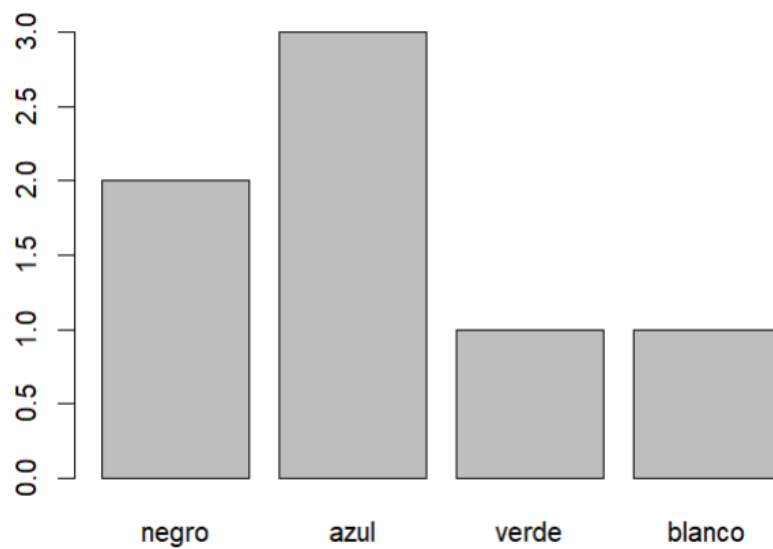
Lo que parece suceder es que la función paste() combina los elementos de los vectores originales que están en la misma posición en un mismo elemento del vector de salida. Y por otro lado la función c lo que genera es una concatenación de los dos vectores tomando cada uno como un elemento diferente, produciendo una combinación end for end como lo indica el archivo. ¿Pero por qué sucede eso?

1_7

La línea 19

```
19 barplot(table(I.am.a.factor)) # extra credit
```

Produce un gráfico



En el eje x tenemos los diferentes niveles del factor y en la y la frecuencia con la que estos aparecen.

1_8

La línea 34

```
34 matrix.transposed <- t(I.am.a.matrix)
```

Genera la siguiente matriz transpuesta

```
> matrix.transposed
      [,1]      [,2]
[1,] "twas"    "brillig"
[2,] "and"     "the"
[3,] "slithey" "toves"
> |
```

1_10

Con la línea 16

```
16 I.am.a.dataframe <- data.frame(I.am.a.vector,I.am.also.a.vector,many.months)
```

Se crea un dataframe

```
      I.am.a.vector I.am.also.a.vector many.months
1             1             twas             Jan
2             2             brillig           Feb
3             3             and             Mar
4             4             the             Apr
5             5             slitley          May
6             6             toves           Jun
> |
```

1_12

Con la línea 23

```
23 medianTime <- median(Chickweight$Time)
```

Se crea la variable

```
> medianTime
[1] 10
```

A2

Vector:

- Unidimensional.
- Almacena elementos de un solo tipo.
- No tiene atributos (más allá de nombres, si se asignan).

Dataframe:

- Bidimensional (tabla con filas y columnas).
- Cada columna es un vector y puede tener su propio tipo.
- Permite nombres para filas y columnas, facilitando la organización de datos.

Parte B - EjercicioR_2.Rproj

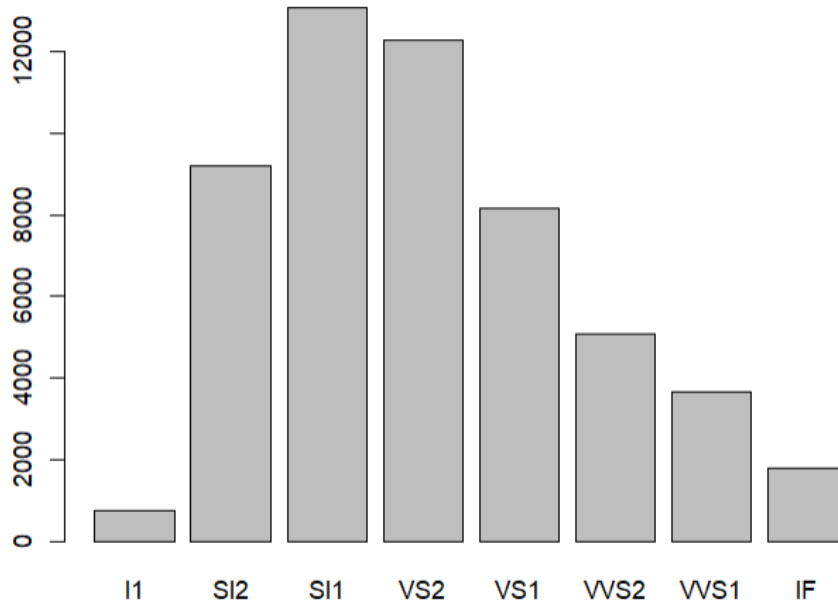
B1

3_2

Con la línea 38 a 41.

```
38 diamonds %>%
39   select(clarity) %>%
40   table() %>% # Put data in appropriate format
41   barplot()
42
```

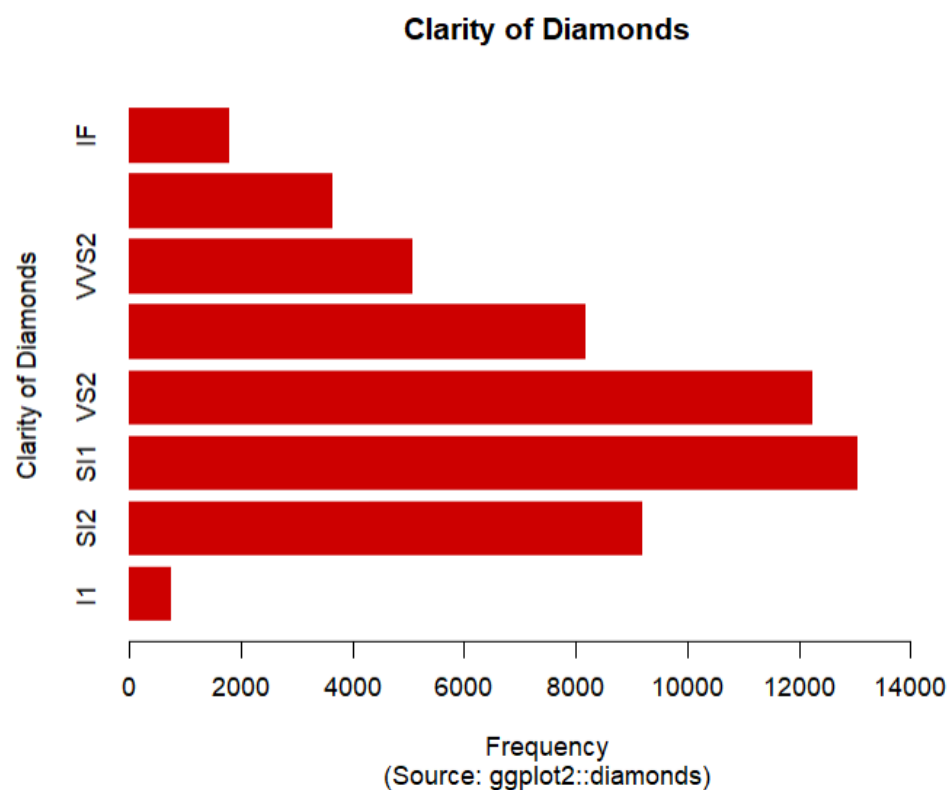
Se genera el siguiente gráfico sobre la claridad de los diamantes



A partir de la línea 51

```
50 # Add options to plot
51 diamonds %>%
52   select(clarity) %>%
53   table() %>% # Put data in appropriate format
54   barplot(
55     main = "Clarity of Diamonds",
56     sub = "(Source: ggplot2::diamonds)",
57     horiz = T, # Draw horizontal bars
58     ylab = "Clarity of Diamonds", # Flip axis labels
59     xlab = "Frequency",
60     xlim = c(0, 15000), # Limits for X axis
61     border = NA, # No borders on bars
62     col = "#CD0000" # red3
63   )
```

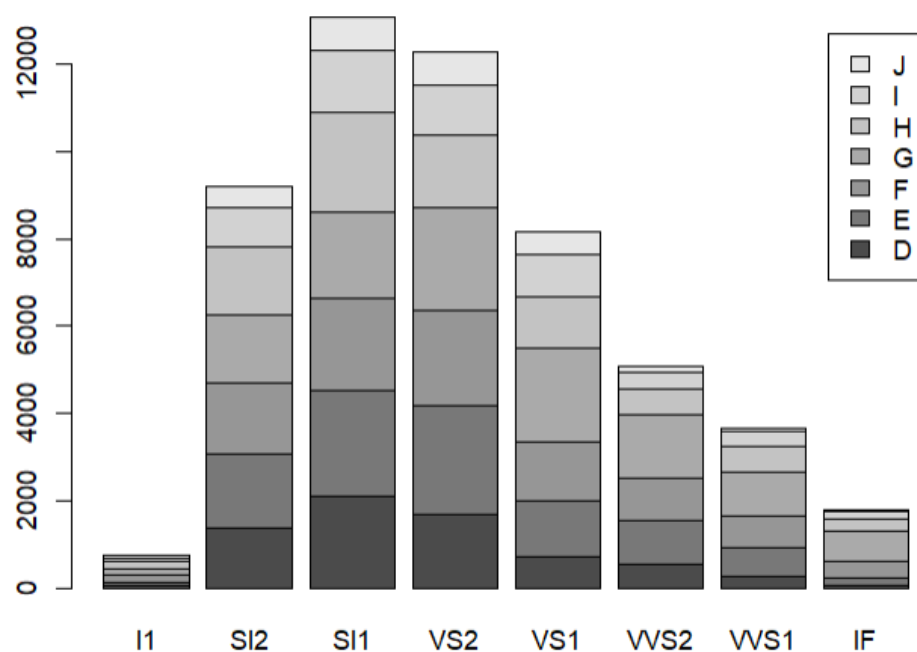
Se puede ver que es posible generar gráficos con varias opciones, nos genera



Con las líneas

```
78 # Stacked bars: step 1: create graph
79 df %>%
80   barplot(legend = rownames(.)) # Draw plot w/legend
```

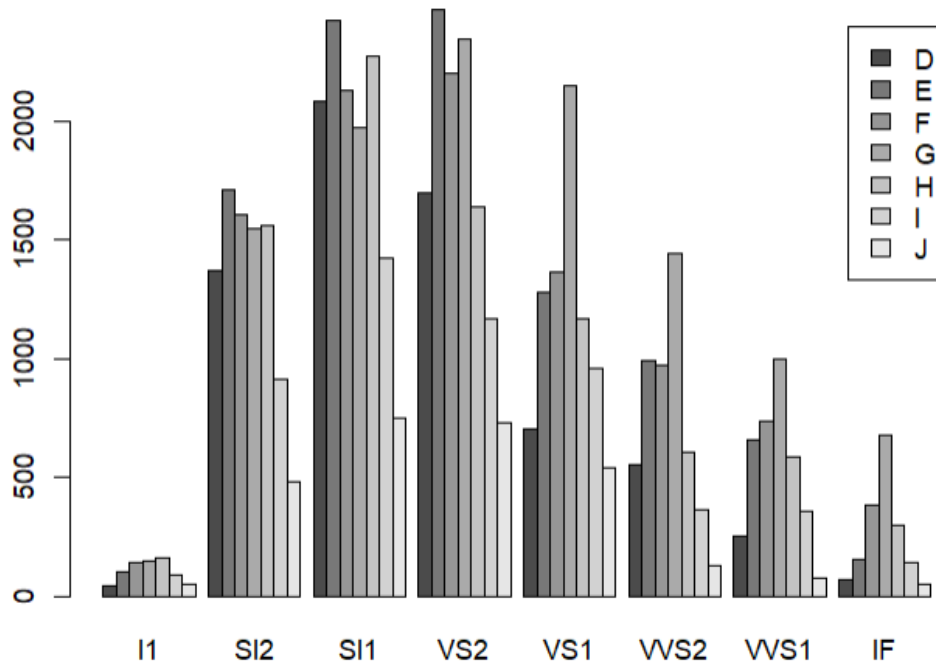
Se genera el gráfico



Con las líneas

```
83 df %>%  
84   barplot(  
85     legend = rownames(.), # Dot is placeholder for pipe  
86     beside = T # Put bars next to each other  
87   )
```

Se genera el gráfico

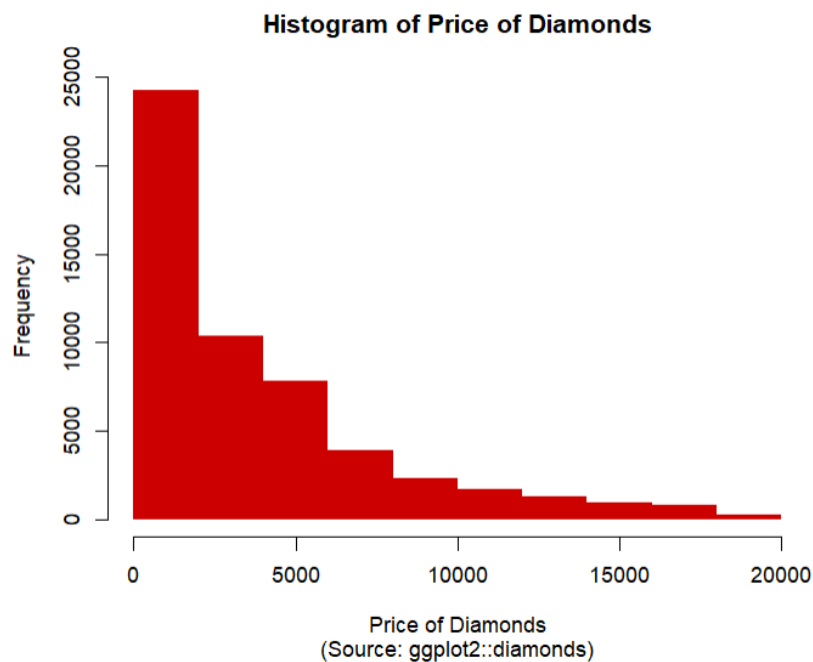


3_3

Con el código que comienza en la línea 24

```
24 hist(diamonds$price,  
25   breaks = 7, # Suggest number of breaks  
26   main = "Histogram of Price of Diamonds",  
27   sub = "(Source: ggplot2::diamonds)",  
28   ylab = "Frequency",  
29   xlab = "Price of Diamonds",  
30   border = NA, # No borders on bars  
31   col = "#CD0000" # red3  
32 )
```

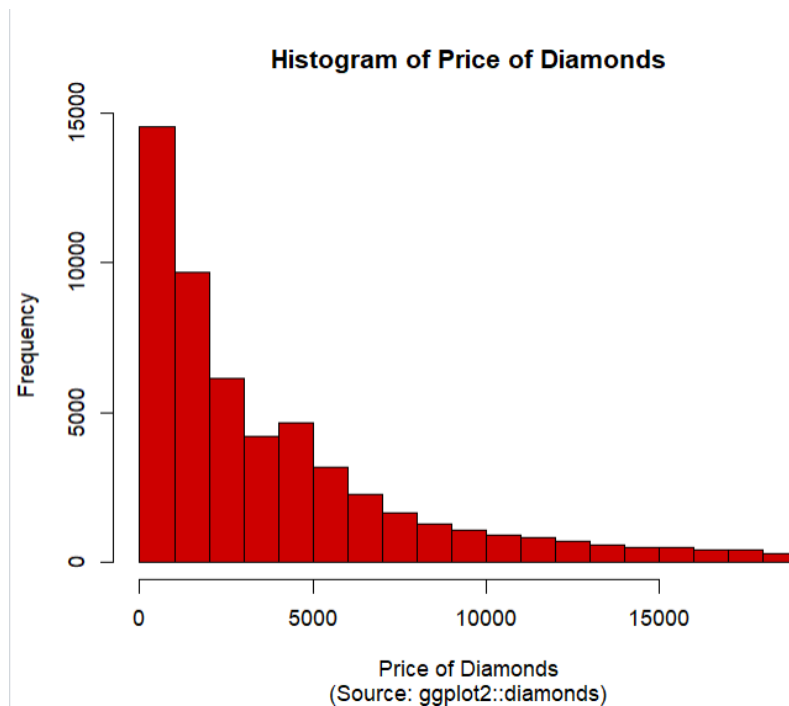
Obtenemos el siguiente histograma



Con el código siguiente de la línea 34

```
34 hist(diamonds$price,  
35     main = "Histogram of Price of Diamonds",  
36     sub  = "(Source: ggplot2::diamonds)",  
37     ylab = "Frequency",  
38     xlab = "Price of Diamonds",  
39     col  = "#CD0000" # red3  
40 )
```

Tenemos el siguiente histograma

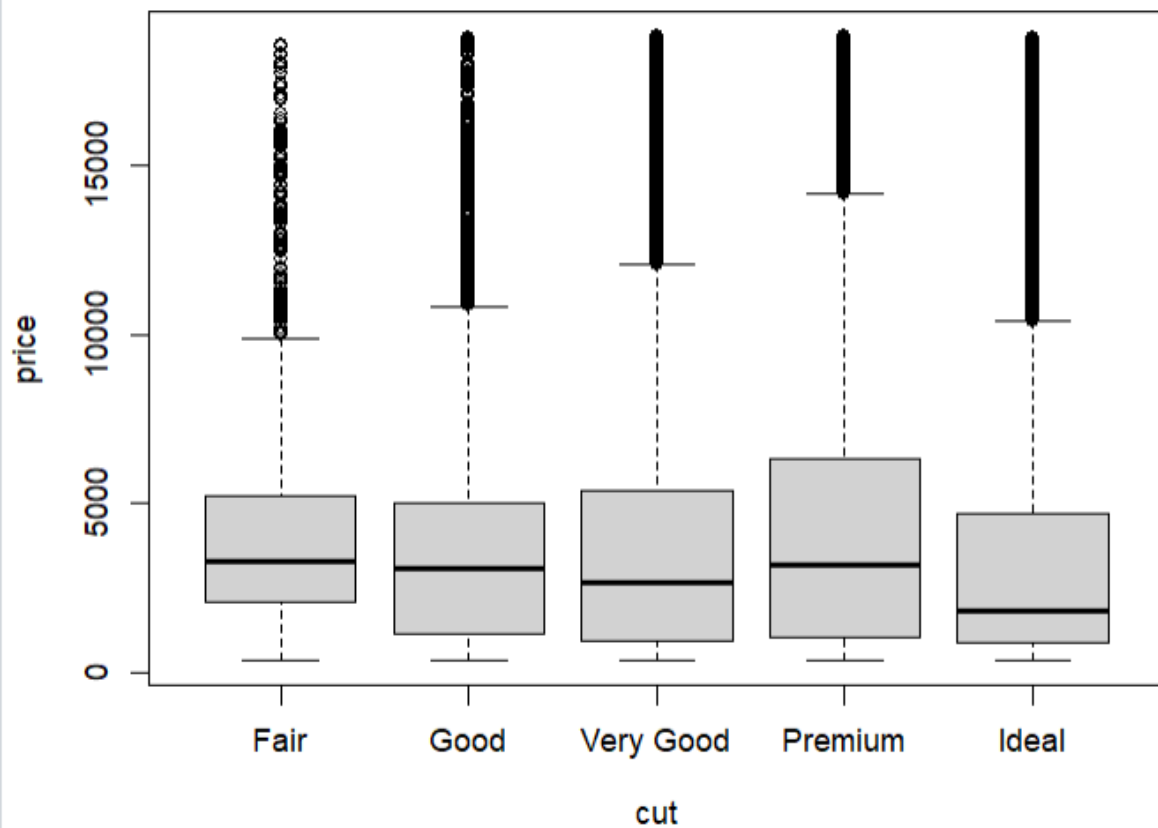


3_4

La línea 46

```
48 diamonds %>%  
49   select(cut, price) %>%  
50   plot()
```

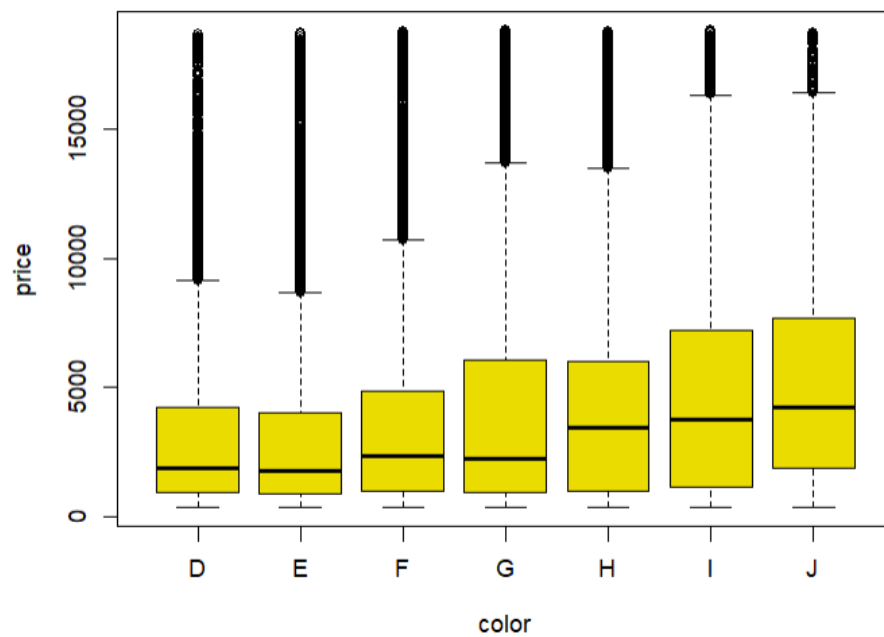
Me genera gráficos de cajas y bigotes por grupo



El siguiente bloque

```
54 diamonds %>%  
55   select(color, price) %>%  
56   boxplot(  
57     price ~ color, # Tilde indicates formula  
58     data = .,      # Dot is placeholder for pipe  
59     col = "#EEEE00"  
60   )
```

Me genera un gráfico de cajas y bigotes con boxplot

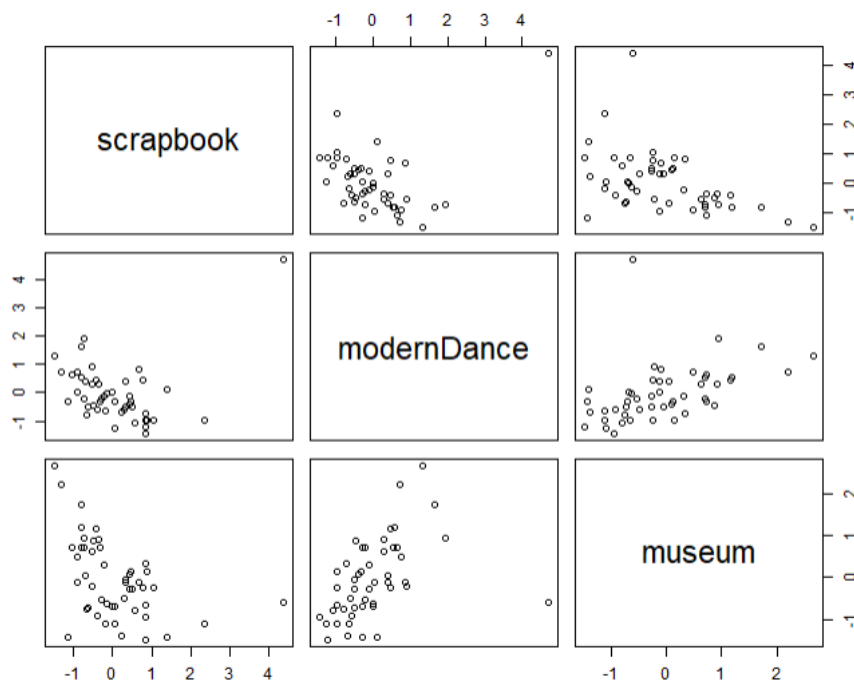


3_5

La línea

```
27 df %>%
28   select(scrapbook,modernDance, museum) %>%
29   plot()
```

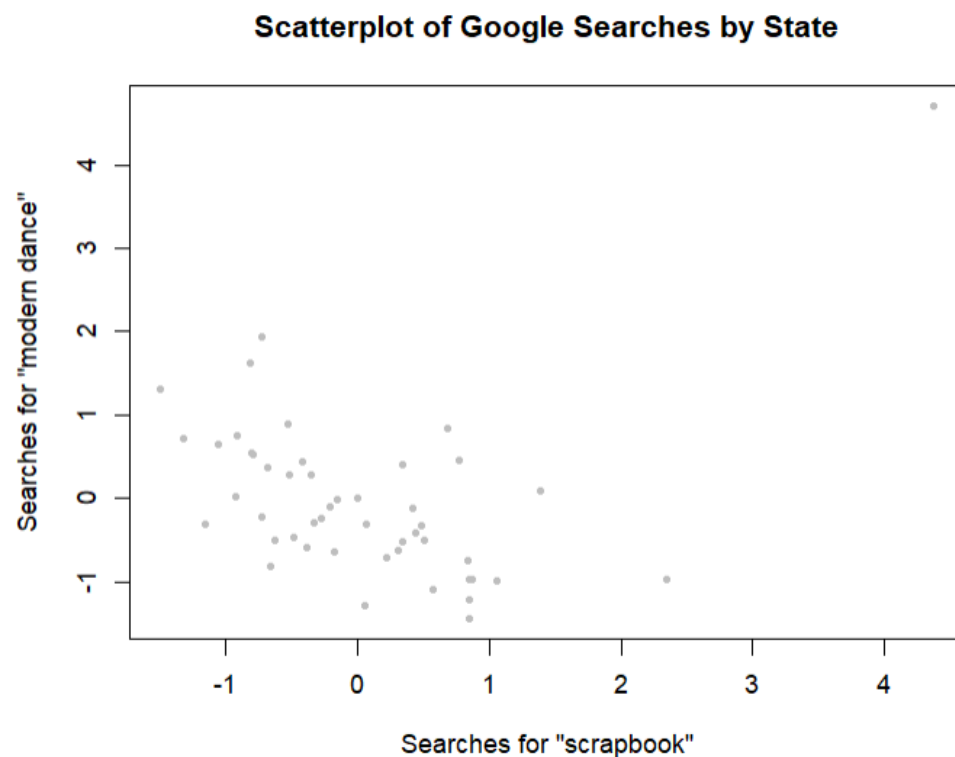
Produce el gráfico



Con las líneas

```
32 df %>%  
33   select(scrapbook,modernDance) %>%  
34   plot(  
35     main = "Scatterplot of Google Searches by State",  
36     xlab = "Searches for \"scrapbook\"",  
37     ylab = "Searches for \"modern dance\"",  
38     col = "gray", # Color of points  
39     pch = 20,      # "Plotting character" (small circle)  
40   )
```

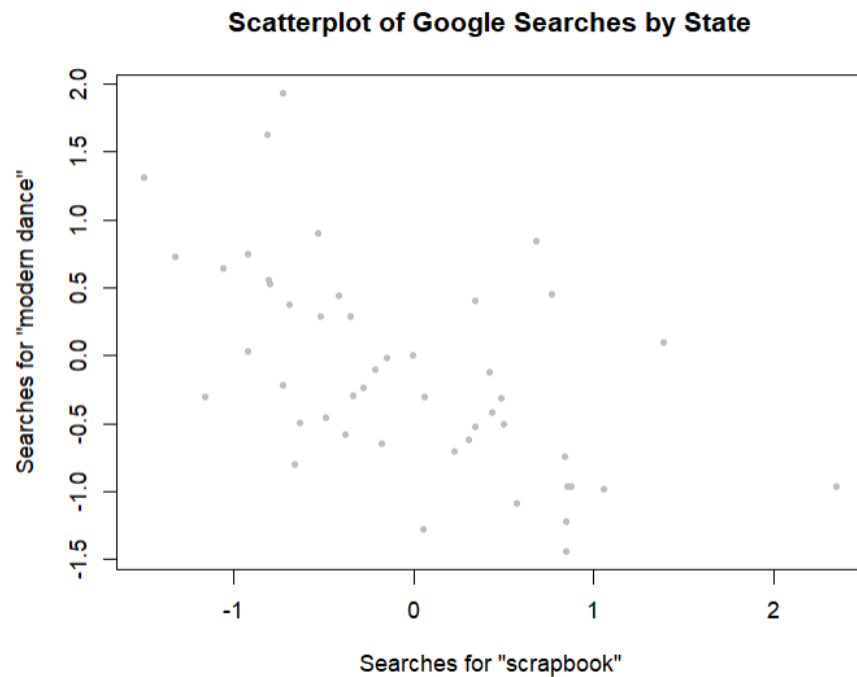
Se obtiene el gráfico de dispersión



Las líneas

```
58 df %>%  
59   select(scrapbook:modernDance) %>%  
60   filter(scrapbook < 4) %>% # filter out outlier  
61   plot(  
62     main = "Scatterplot of Google Searches by State",  
63     xlab = "Searches for \"scrapbook\"",  
64     ylab = "Searches for \"modern dance\"",  
65     col = "gray", # Color of points  
66     pch = 20,      # "Plotting character" (small circle)  
67   )
```

Me producen el gráfico de dispersión

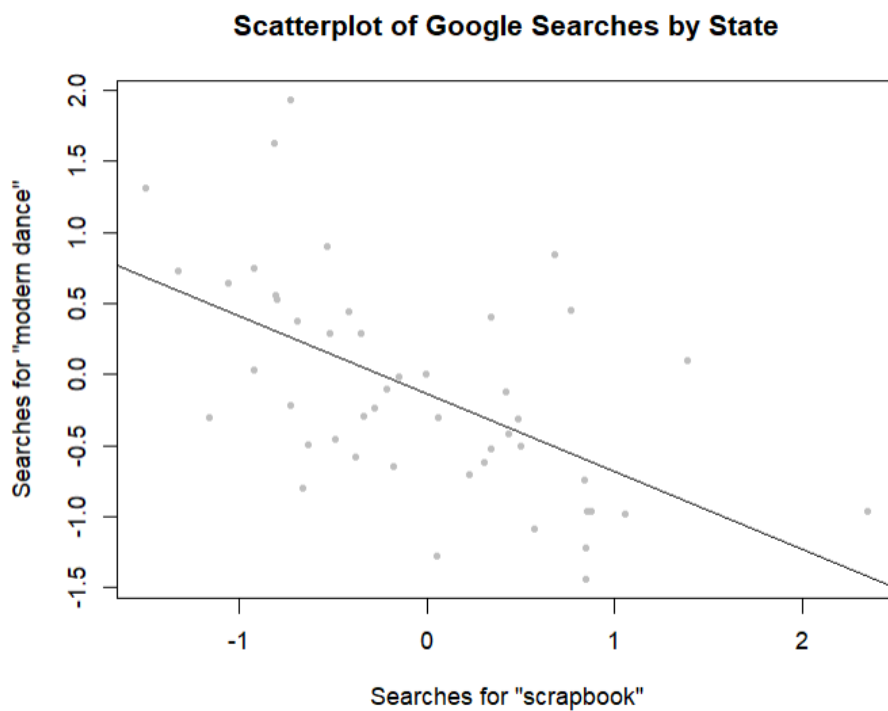


Con menos outliers

Las líneas

```
70 df %>%
71   filter(scrapbook < 4) %>% # filter out outlier
72   lm(modernDance ~ scrapbook, data = .) %>%
73   abline()
```

Me van a generar el siguiente gráfico

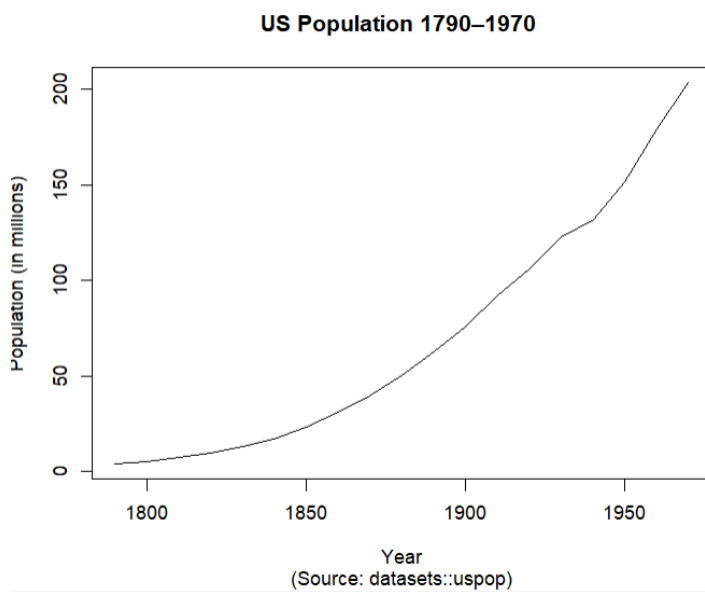


3_6

Con las líneas

```
24 # Plot with options
25 uspop %>%
26   plot(
27     main = "US Population 1790-1970 ",
28     sub = "(Source: datasets::uspop)",
29     xlab = "Year",
30     ylab = "Population (in millions)",
31   )
32
```

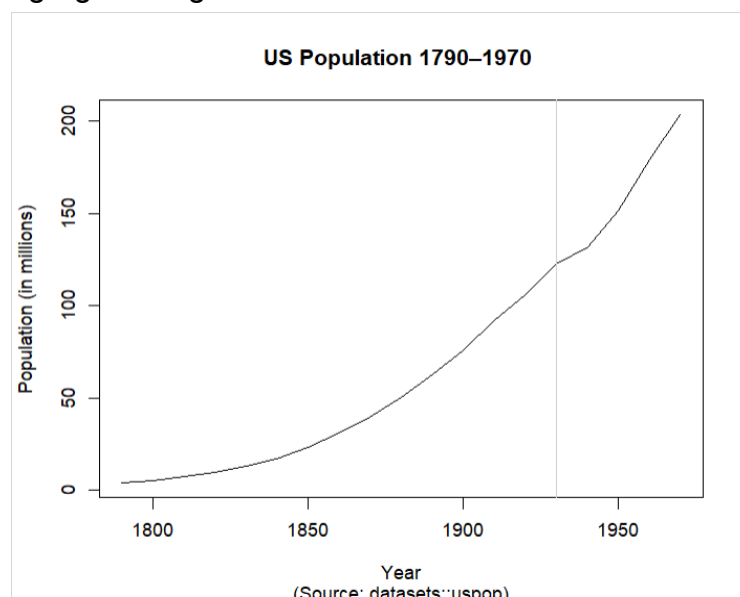
Resulta el gráfico de línea



La línea

```
33 abline(v = 1930, col = "lightgray")
```

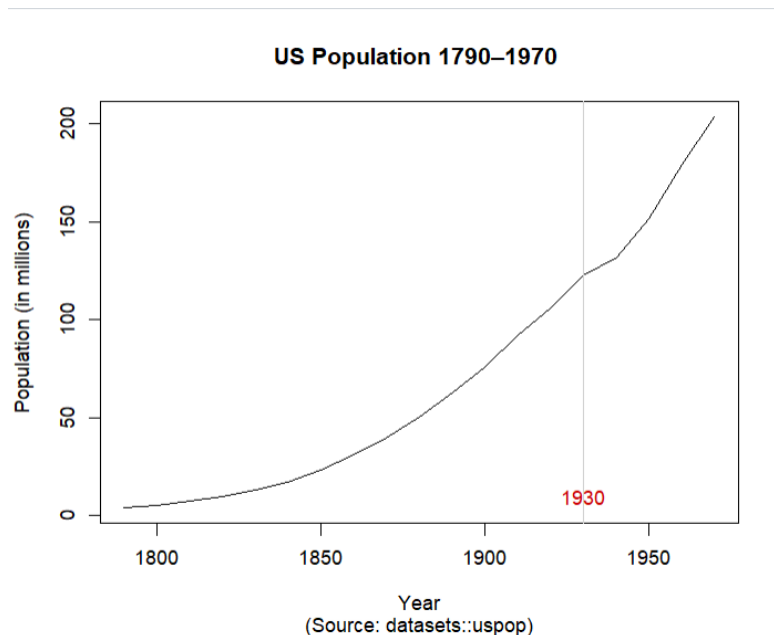
Agrega a mi gráfico una línea en el valor 1930



La línea 34

```
34 text(1930, 10, "1930", col = "red3")
```

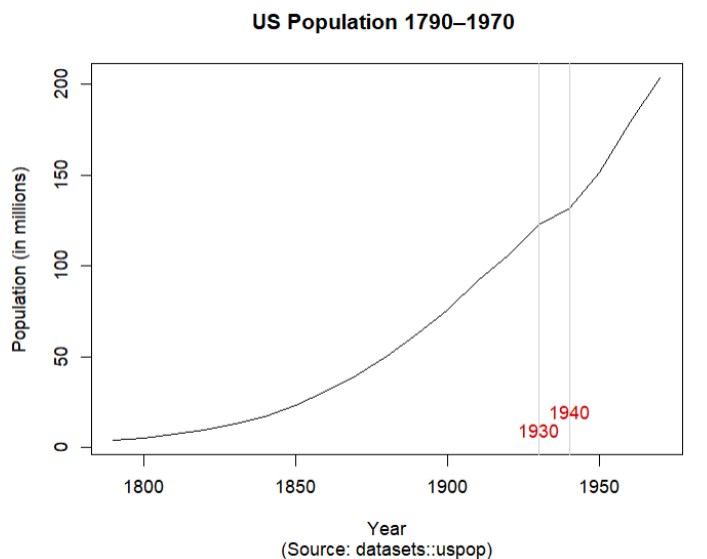
Agrega un label al gráfico



Similarmente las siguiente líneas

```
36 abline(v = 1940, col = "lightgray")
37 text(1940, 20, "1940", col = "red3")
38
```

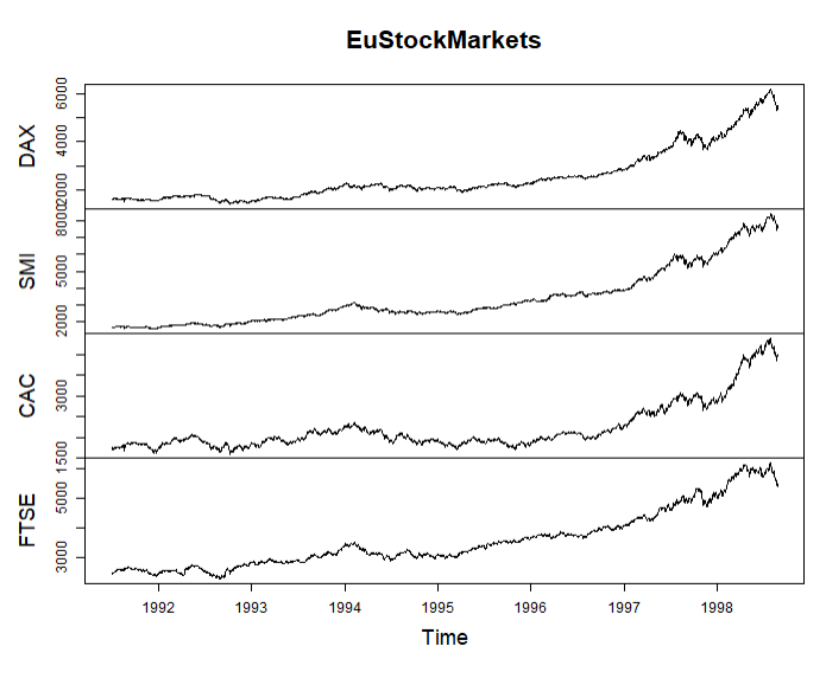
Me producen



La línea 56

```
56 plot(EuStockMarkets) # Stacked windows
```

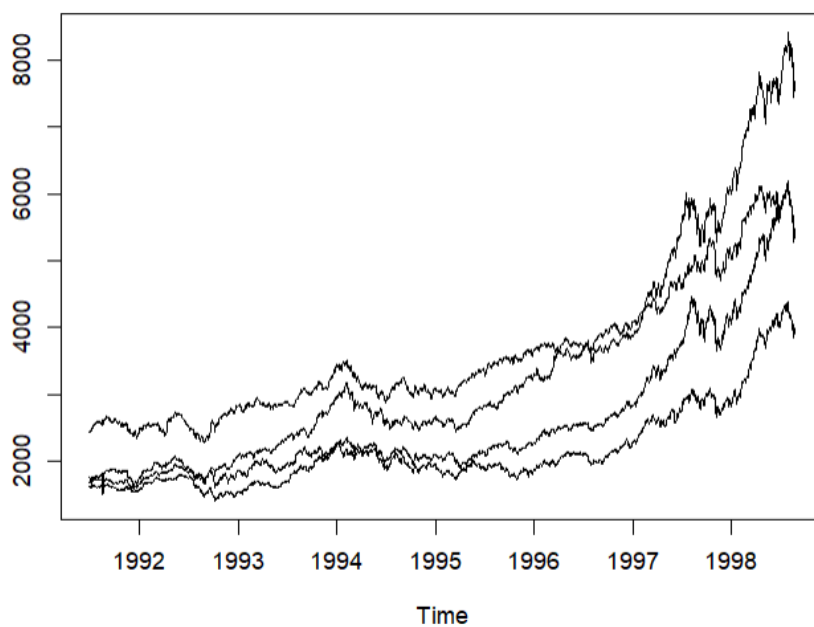
Me da un gráfico con ventanas apiladas



Por el contrario la línea 58

```
58 ts.plot(EuStockMarkets) # One window
```

Me da lo mismo pero en una sola ventana



B2

5_1

En la línea 57

```
57 summary(df)
```

Obtengo el resultado

```
> summary(df)
  state_code      region      psychRegions
Length:48      Midwest :12 Friendly and Conventional :24
Class :character Northeast: 9 Relaxed and Creative      :10
Mode  :character South   :16 Temperamental and Uninhibited:14
      West    :11
```

```
> |
```

5_2

Con la línea

```
25 df %>% # Summary for one variable
26   select(entrepreneur) %>%
27   summary()
28
```

Me produce el resultado

```
+ summary()
  entrepreneur
Min.   :-1.79900
1st Qu.: -0.59675
Median :  0.05650
Mean    :  0.03448
3rd Qu.:  0.51700
Max.    :  2.54700
```

```
> |
```

5_3

La línea 23

```
22 ct <- table(df$region, df$psychRegions)
23 ct
```

Me da el resultado

```
      Friendly and Conventional Relaxed and Creative Temperamental and Uninhibited
Midwest                11                0                1
Northeast               0                0                9
South                  10                2                4
West                   3                 8                0
```

```
> |
```

La línea 32

```
32 ct %>%
33   prop.table(1) %>% # 1 is for row percentages
34   round(2) %>%
35   multiply_by(100)
```

Me da el resultado

```

      Friendly and Conventional Relaxed and Creative Temperamental and Uninhibited
Midwest                92                0                8
Northeast              0                0               100
South                 62               12               25
West                  27               73                0
> |

```

La línea 38

```

38 ct %>%
39   prop.table(2) %>% # 2 is for columns percentages
40   round(2) %>%
41   multiply_by(100)
42

```

Me da el resultado

```

      Friendly and Conventional Relaxed and Creative Temperamental and Uninhibited
Midwest                46                0                7
Northeast              0                0               64
South                 42               20               29
West                  12               80                0
> |

```

Finalmente la línea 44

```

44 ct %>%
45   prop.table() %>% # No argument for total percentages
46   round(2) %>%
47   multiply_by(100)
48

```

Me da como resultado

```

44 ct %>%
45   prop.table() %>% # No argument for total percentages
46   round(2) %>%
47   multiply_by(100)
48

```