CONTENIDO

1. Gráficos ggplot2

Una **gráfica de dispersión** puede ser usada para datos en la forma de *parejas ordenadas* de números. El resultado será un montón de puntos "dispersos" alrededor del plano.

- Si la tendencia general es que los puntos suban a la derecha de la gráfica, entonces decimos que hay una **correlación positiva** (Forma un ángulo de 45 grados) ente las dos variables medidas..
- Si los puntos caen a la izquierda de la gráfica, decimos que hay una **correlación negativa** (ángulo de -45 grados).
- Si no hay tendencia general, entonces **No hay correlación**
- Si la tendencia no es muy pronunciada – esto es, los puntos están dispersos ampliamente – entonces decimos que las variables están **débilmente correlacionadas.**
- Si la correlación es más pronunciada, decimos que las variables están **fuertemente correlacionadas.**

```
library(ggplot2)
setwd("C:\\Users\\81799\\OneDrive\\Documentos\\ESFM_CLASES\\Servicio Social ARTF\\Machine Learni
auto <- read.csv("data/tema2/auto-mpg.csv",
                 header = TRUE,
                 stringsAsFactors = FALSE)
head(auto)
```

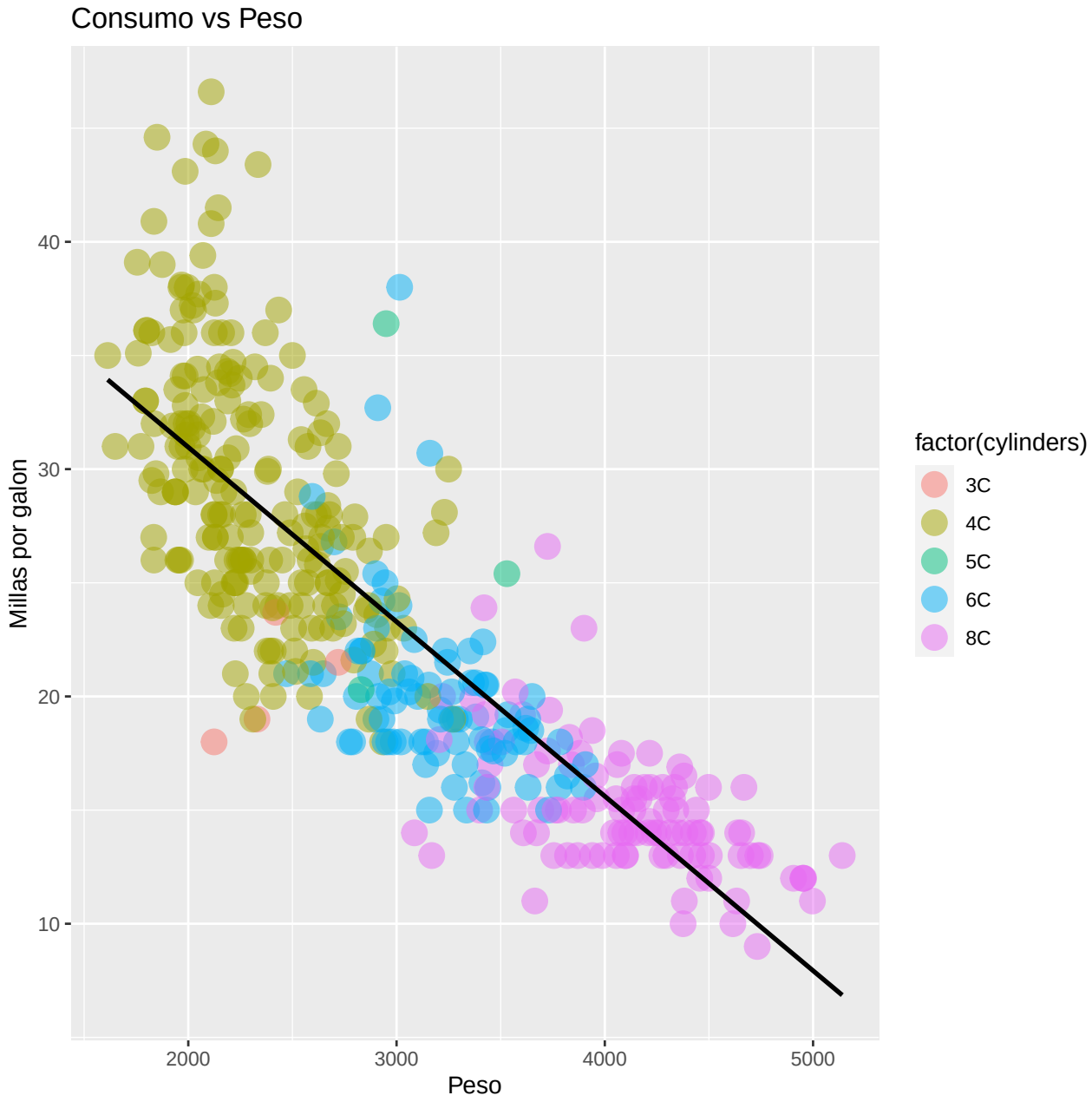
```
##   No mpg cylinders displacement horsepower weight acceleration model_year
## 1  1  28         4          140         90   2264         15.5         71
## 2  2  19         3           70         97   2330         13.5         72
## 3  3  36         4          107         75   2205         14.5         82
## 4  4  28         4           97         92   2288         17.0         72
## 5  5  21         6          199         90   2648         15.0         70
## 6  6  23         4          115         95   2694         15.0         75
##               car_name
## 1 chevrolet vega 2300
## 2     mazda rx2 coupe
## 3        honda accord
## 4     datsun 510 (sw)
## 5         amc gremlin
## 6          audi 100ls
```

```
auto$cylinders <- factor(auto$cylinders,
                         levels = c(3,4,5,6,8),
                         labels = c("3C", "4C", "5C", "6C", "8C"))
```

Gráfica

```
ggplot(auto,aes( weight, mpg ))+geom_point(alpha = 1/2 , size = 5,
aes(color=factor(cylinders)))+
labs(x="Peso", y="Millas por galon", title = "Consumo vs Peso")+
  geom_smooth(method = "lm", se=FALSE, col = "black" )
```

```
## `geom_smooth()` using formula 'y ~ x
```



1.1. Gráficos de linea

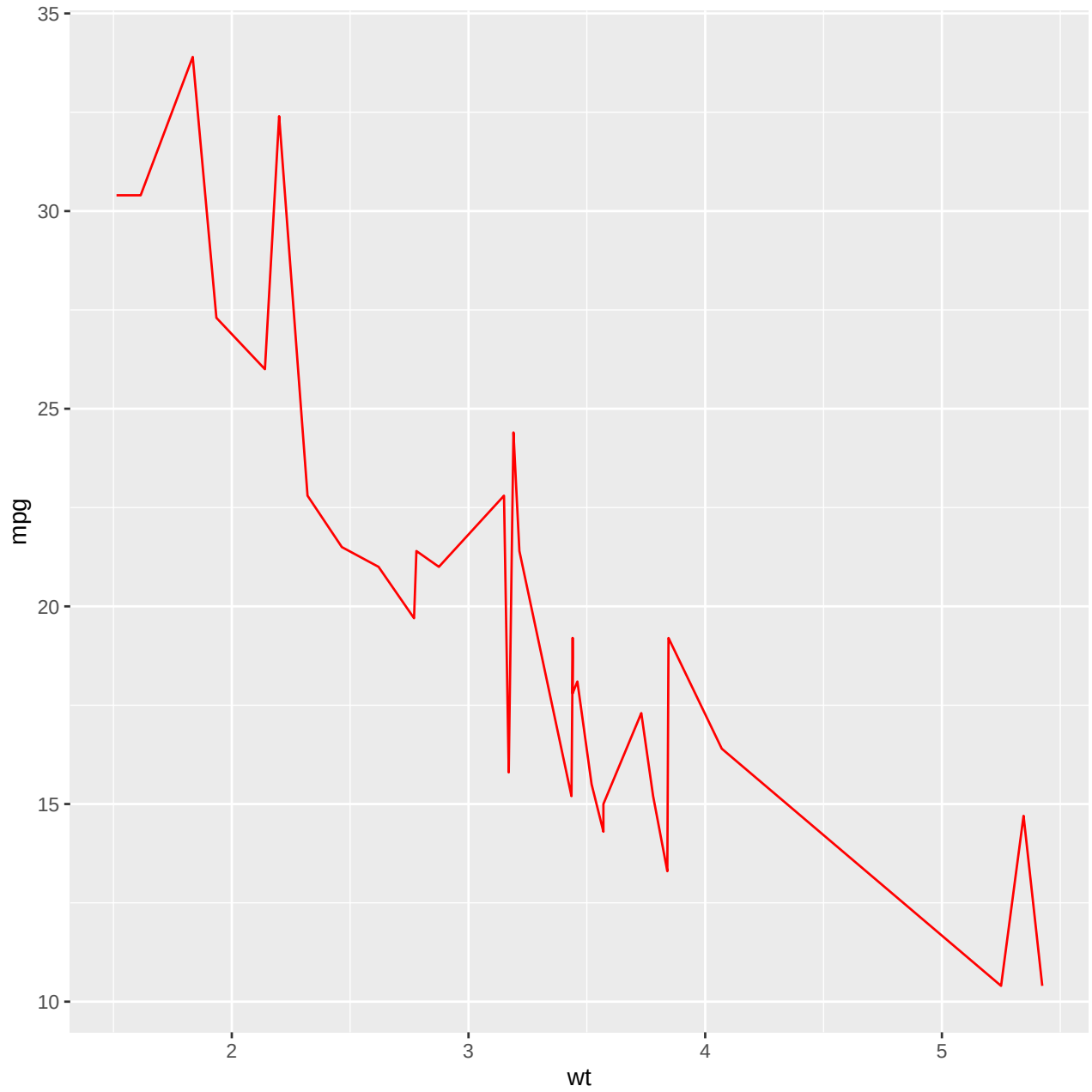
Los gráficos de lineas permiten visualizar los cambios a lo largo de un rango continuo, como el tiempo o la distancia. La visualización del cambio con un gráfico de líneas permite ver de una sola vez la tendencia general y comparar simultáneamente varias tendencias.

```
library(ggplot2)
setwd("C:\\Users\\81799\\OneDrive\\Documentos\\ESFM_CLASES\\Servicio Social ARTF\\Machine Learni
mtcars <- read.csv("data/tema7/mtcars.csv",stringsAsFactors = FALSE)
head(mtcars)
```

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

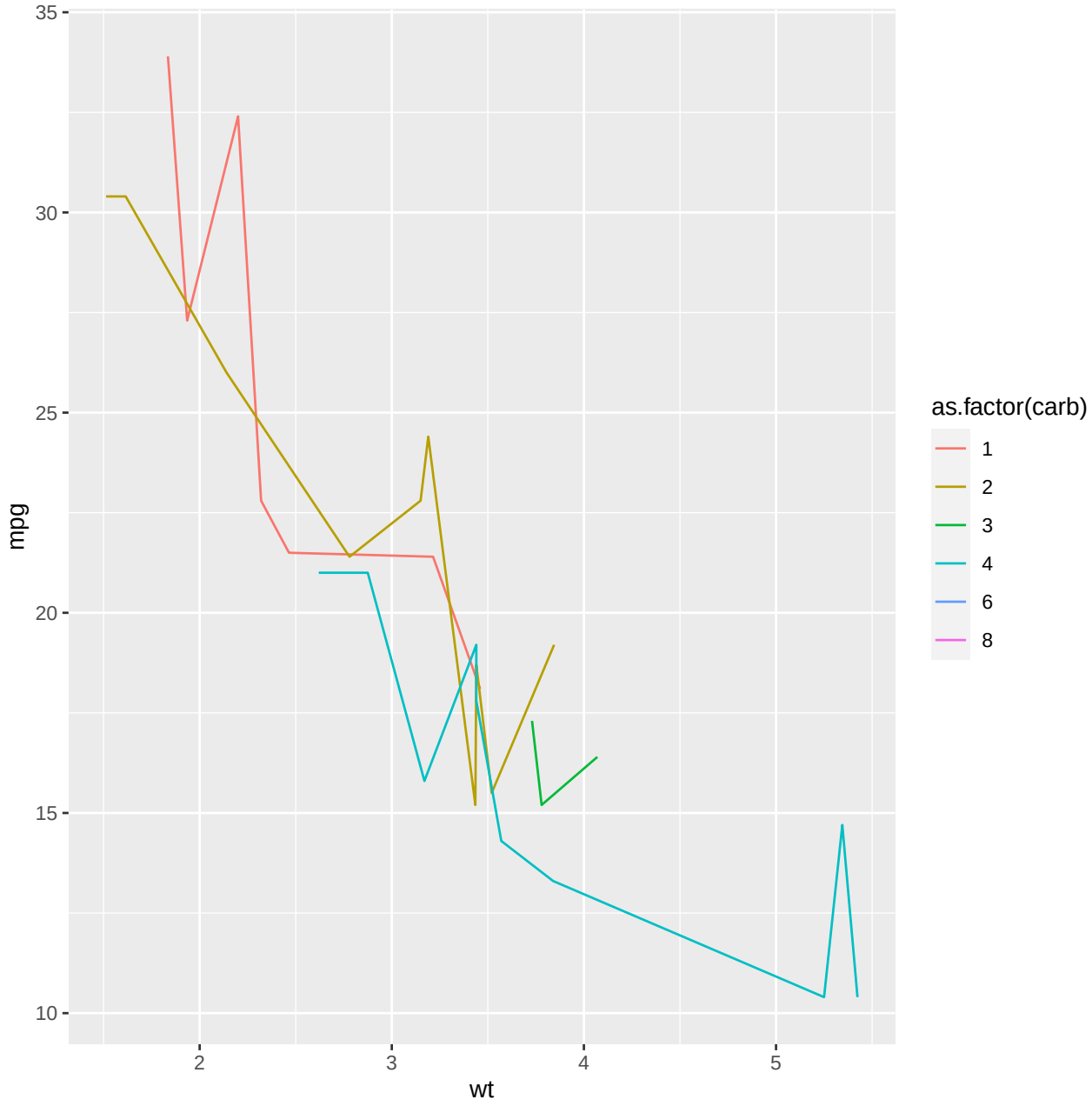
Gráfica

```
ggplot(mtcars , aes(wt, mpg ))+geom_line(color = "red")
```



Gráfica agrupando por categoría

```
ggplot(mtcars , aes(wt, mpg ))+
  geom_line(aes(color = as.factor(carb)) )
```



Lo anterior son las lineas agrupadas en el número de carburadores o el número de tipo de carburador.

1.2. Gráfico de barras.

Un gráfico de barras es una forma de representar gráficamente un conjunto de datos o valores mediante barras *rectangulares* de longitud proporcional a los valores representados. Los gráficos de barras pueden ser usados para comparar cantidades de una variable en diferentes momentos o diferentes variables para el mismo momento.

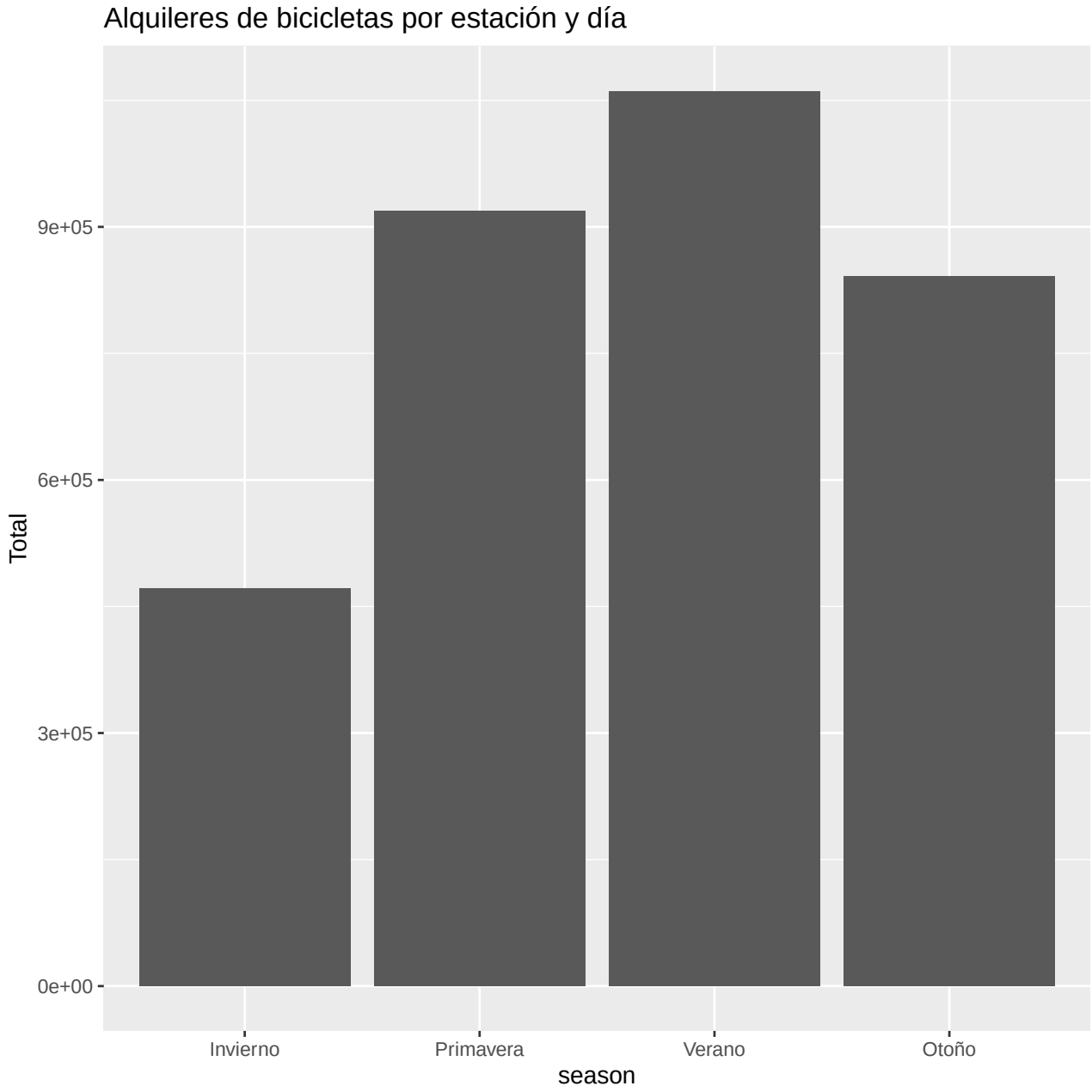
```
library(ggplot2)
setwd("C:\\Users\\81799\\OneDrive\\Documentos\\ESFM_CLASES\\Servicio Social ARTF\\Machine Learni
bike <- read.csv("data/tema7/daily-bike-rentals.csv")
bike$season <-  factor(bike$season, levels = c(1,2,3,4),
                       labels = c("Invierno", "Primavera", "Verano", "Otoño"))
bike$workingday <-  factor(bike$workingday, levels = c(0,1),
                           labels = c("Día libre", "Día de trabajo"))
bike$weathersit <- factor(bike$weathersit , levels = c(1,2,3),
                          labels = c("Despejado", "Nublado", "Lluvia") )
```

Vamos agrupar por season y workingday de modo que tengamos una nueva columna donde nos de el conteo total de lo anterior.

```
library(dplyr)

##
## Attaching package: 'dplyr
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

bike.sum = bike %>% group_by(season, workingday) %>%
  summarise(Total = sum(cnt))

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

bike.sum

## # A tibble: 8 x 3
## # Groups:   season [4]
##   season    workingday      Total
##   <fct>     <fct>           <int>
## 1 Invierno  Día libre      137683
## 2 Invierno  Día de trabajo 333665
## 3 Primavera Día libre      287976
## 4 Primavera Día de trabajo 630613
## 5 Verano    Día libre      312056
## 6 Verano    Día de trabajo 749073
## 7 Otoño     Día libre      262554
## 8 Otoño     Día de trabajo 579059
```

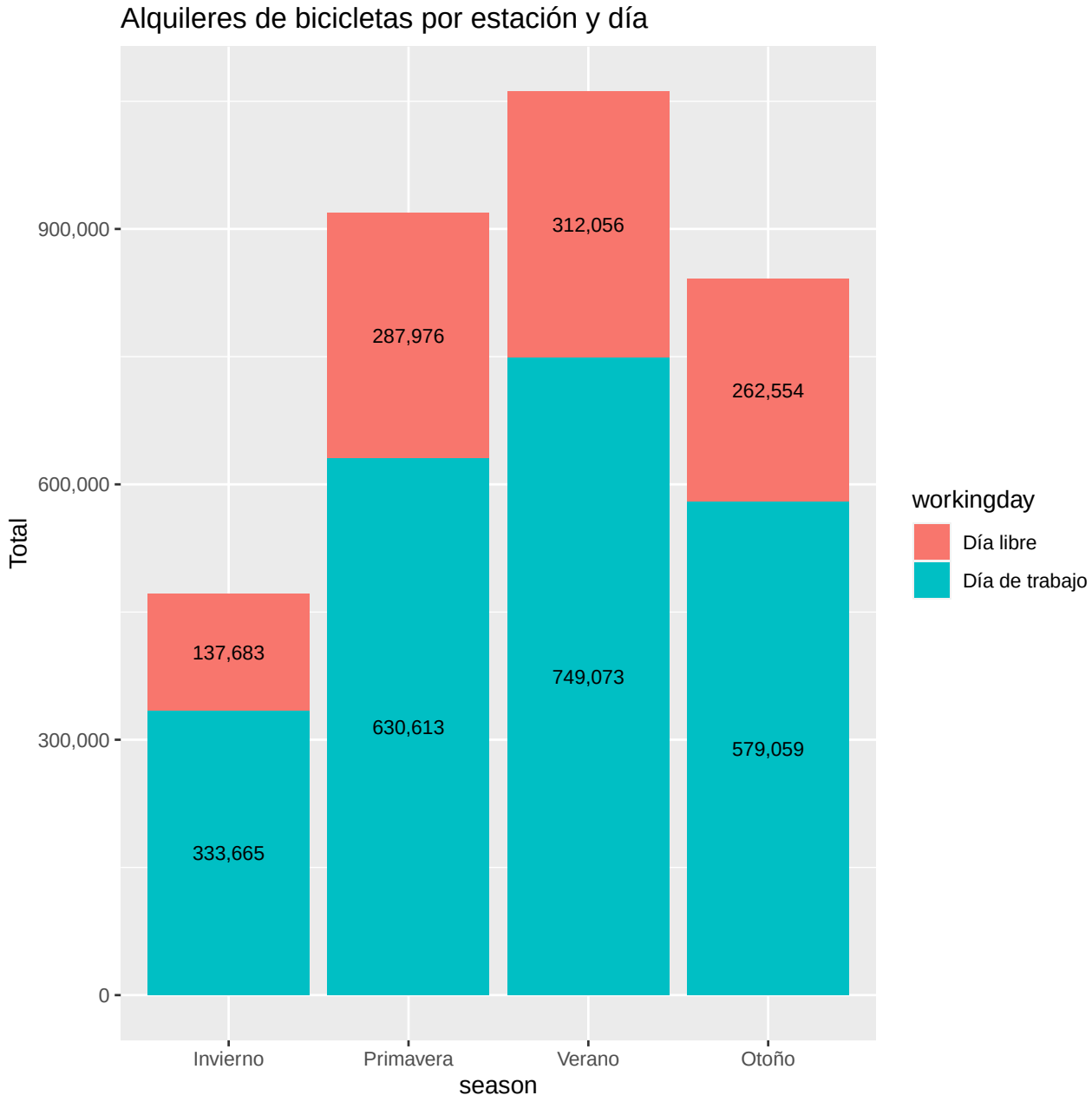
Gráfica

```
ggplot(bike.sum , aes (x=season , y=Total))+
  geom_bar(show.legend = T, stat = "identity")+
  labs(title = "Alquileres de bicicletas por estación y día")
```



en la gráfica no sabemos si es día libre o de trabajo, entonces lo haremos de la siguiente manera:

```
ggplot(bike.sum , aes (x=season , y=Total,fill = workingday,
                       label =scales::comma(Total) ))+
  geom_bar(show.legend = T, stat = "identity")+
  labs(title = "Alquileres de bicicletas por estación y día")+
  scale_y_continuous(labels = scales::comma )+
  geom_text(size = 3, position = position_stack(vjust = 0.5))
```



1.3. Gráficos de distribuciones

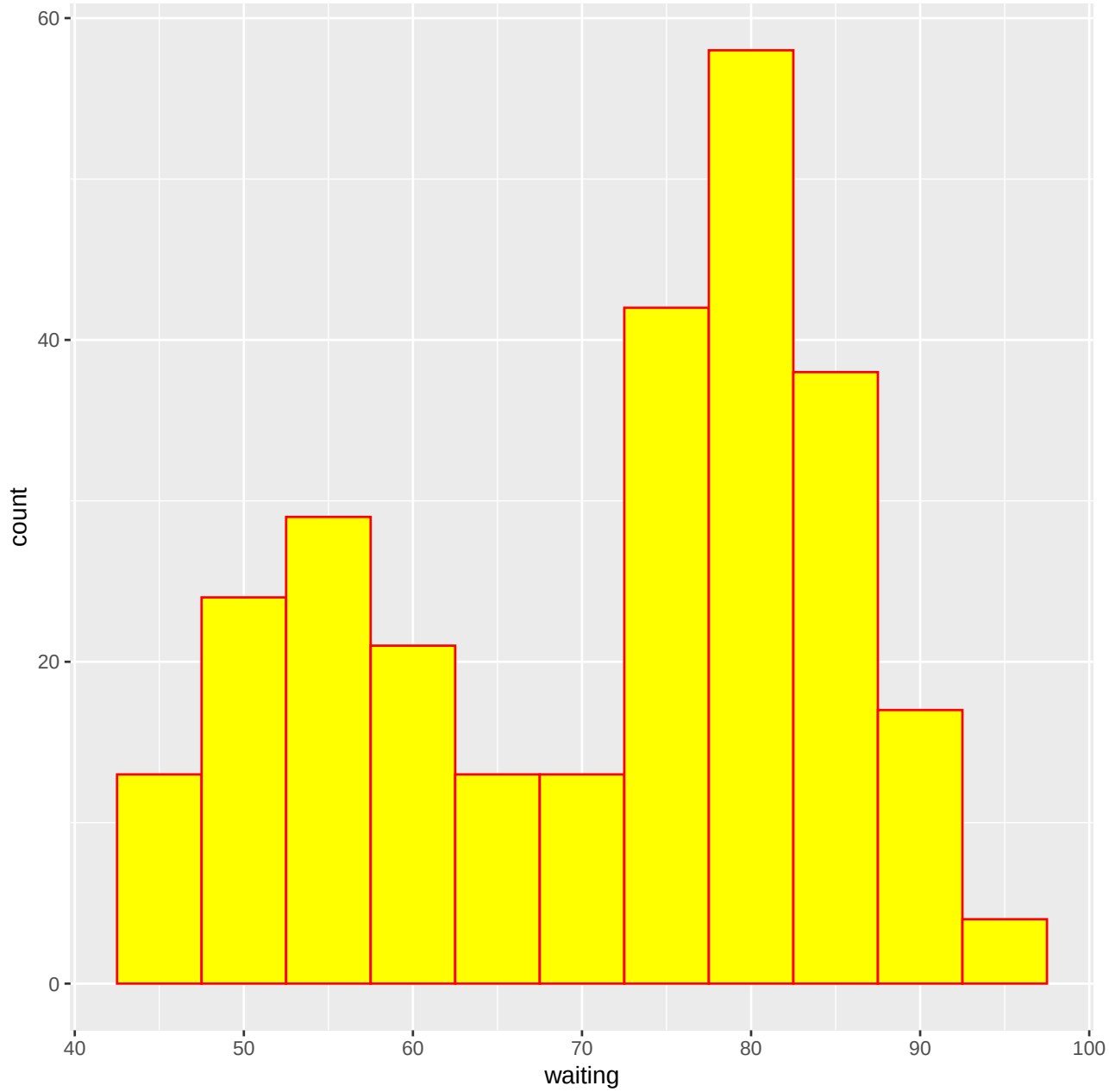
Los histogramas son las representaciones gráficas que mejor nos ayudan a explorar como se distribuyen los elementos de una o más variables cuantitativas. También nos ofrecen un modo muy útil de representar distribuciones a través de las funciones de densidad, las cuales representan una aproximación de la distribución de los datos con una función continua en lugar de divisiones unitarias y discretas, las cuales ayudan a estimar la función de distribución o función de densidad (función de probabilidad).
En este caso veremos como representar los histogramas y las funciones de densidad haciendo uso de la librería *ggplot2*

```
library(ggplot2)
setwd("C:\\Users\\81799\\OneDrive\\Documentos\\ESFM_CLASES\\Servicio Social ARTF\\Machine Learni
geiser <- read.csv("data/tema7/geiser.csv")
head(geiser, 5)

##   X eruptions waiting
```

```
## 1 1     3.600       79
## 2 2     1.800       54
## 3 3     3.333       74
## 4 4     2.283       62
## 5 5     4.533       85
```

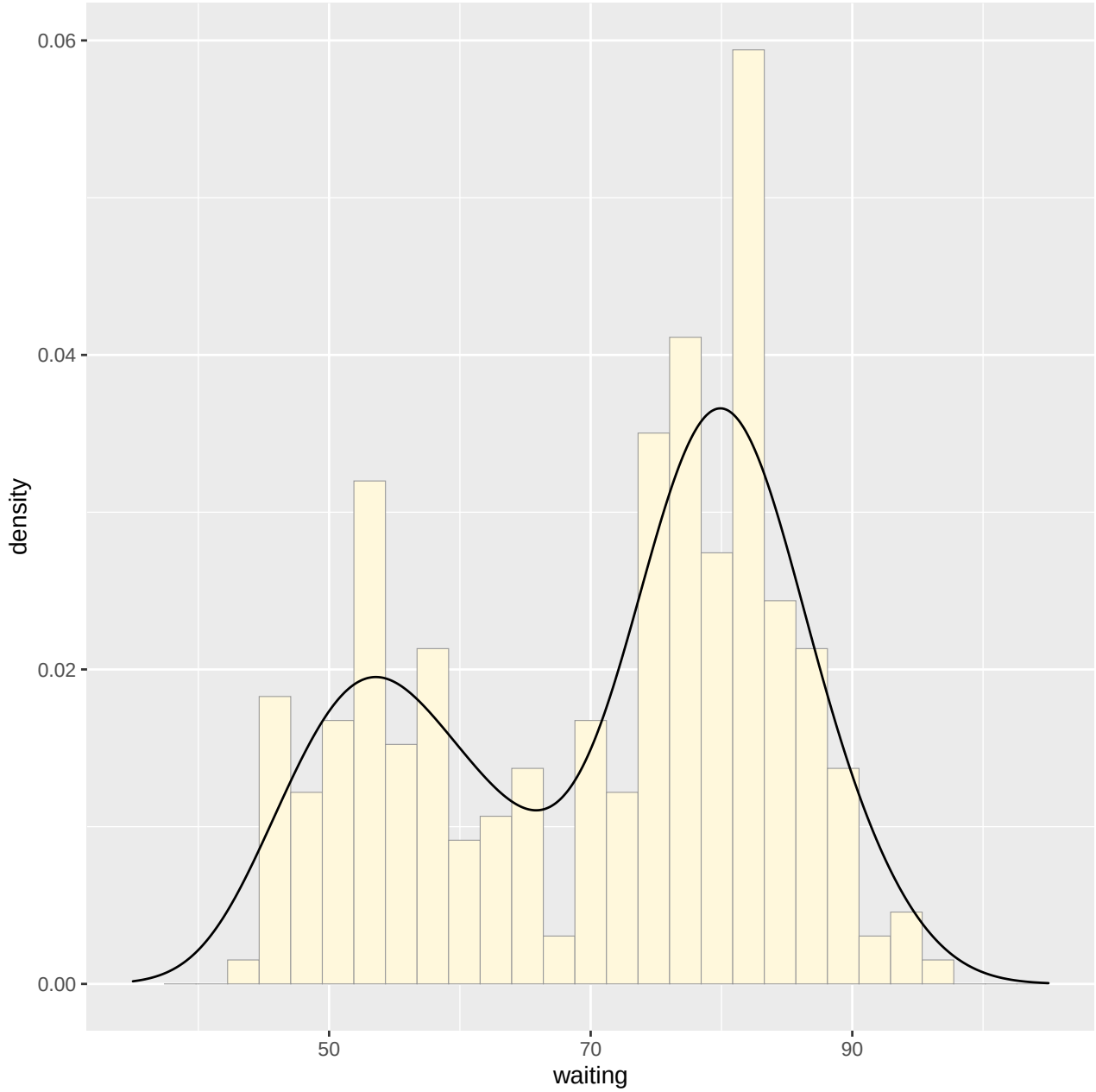
Histograma con frecuencias absolutas

```
ggplot(geiser , aes(x=waiting))+
  geom_histogram(binwidth = 5, fill = "yellow", colour ="red"  )
```



Histograma con frecuencias relativas y función de densidad.

```
ggplot(geiser , aes(x=waiting, y= ..density..))+
  geom_histogram(fill = "cornsilk", color ="grey60", size =0.2  )+
  geom_density()+xlim(35,105)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 1 rows containing missing values (geom_bar).
```

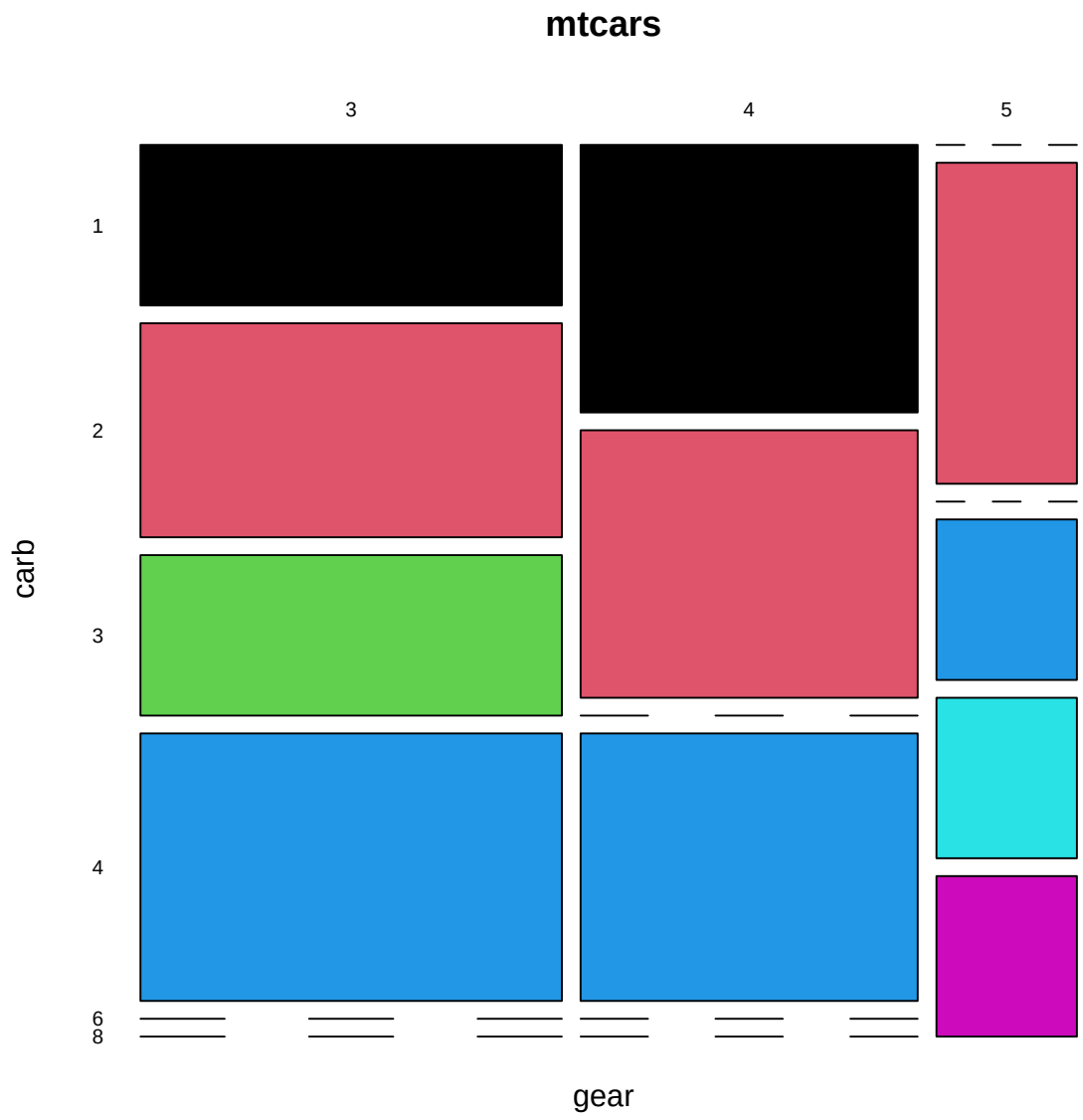


Recodando que el área bajo la función de densidad siempre debe de sumar 1.

1.4. Gráfico de Mosaicos

Los gráficos de mosaico o diagramas de Marimekko son usados para mostrar la relación entre dos variables discretas, ya sean factores o cadenas de texto.
Este tipo de gráfico recibe su nombre porque consiste en una cuadrícula, en la que cada rectángulo representa el número de casos que corresponden a un cruce específico de variables. Entre más casos se encuentren en ese cruce, más grande será el rectángulo.

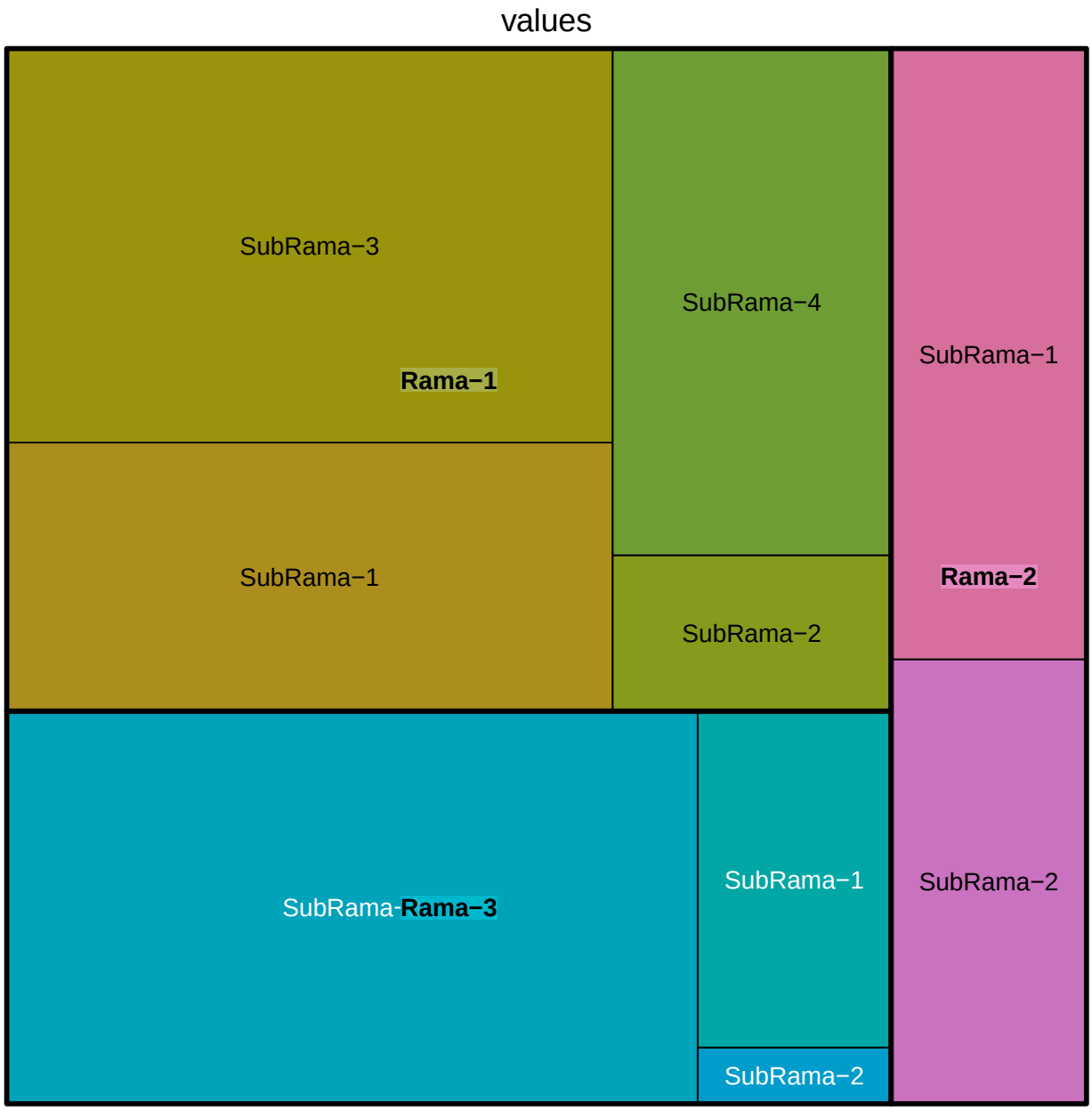
```
mosaicplot(~gear+carb, data = mtcars, color =1:6, las =1 )
```



1.5. Treemap

```
library(treemap)
Rama <- c(rep("Rama-1",4),rep("Rama-2",2),rep("Rama-3",3))
SubRama <- paste("SubRama",c(1,2,3,4,1,2,1,2,3), sep = "-" )
values <- c(15,4,22,13,11,8,6,1,25)
data <- data.frame(Rama, SubRama ,values)
head(data,9)

##      Rama   SubRama values
## 1 Rama-1 SubRama-1     15
## 2 Rama-1 SubRama-2      4
## 3 Rama-1 SubRama-3     22
## 4 Rama-1 SubRama-4     13
## 5 Rama-2 SubRama-1     11
## 6 Rama-2 SubRama-2      8
```

```
## 7 Rama-3 SubRama-1      6
## 8 Rama-3 SubRama-2      1
## 9 Rama-3 SubRama-3     25

treemap(data, index = c("Rama", "SubRama"),
        vSize = "values", type ="index" )
```




```
library(treemap)
setwd("C:\\Users\\81799\\OneDrive\\Documentos\\ESFM_CLASES\\Servicio Social ARTF\\Machine Learni
post <- read.csv("data/tema7/post-data.csv")
head(post)

##     id  views comments              category
## 1 5019 148896       28 Artistic Visualization
## 2 1416  81374       26          Visualization
## 3 1416  81374       26               Featured
## 4 3485  80819       37               Featured
```

```
## 5 3485  80819       37               Mapping
## 6 3485  80819       37          Data Sources

treemap(post, index = c("category", "comments"),
        vSize = "views", type = "index" )
```