14Graficos_cheatsheet.R

moka

2023-04-21

```
# Autor: Monika Avila Marquez, Ph.D.
# Fecha: 12.04.2023
# Objetivo: Data plotting
# Referencia: Basado en R Programming Fundamentals, StanfordOnline XDFS112
```

```
library(dplyr)
# Limpiar el espacio de trabajo
rm(list=ls())
```

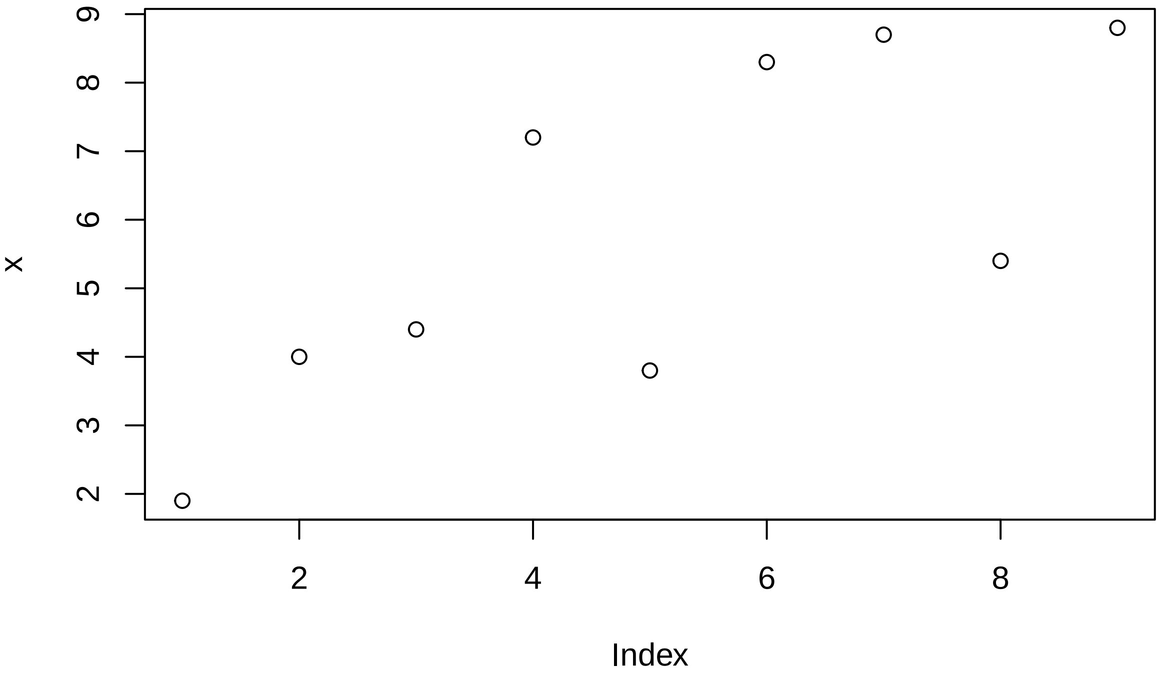
```
# Configurar el directorio
```

```
midirectorio<-setwd("~/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/14Graficos")
midirectorio
```

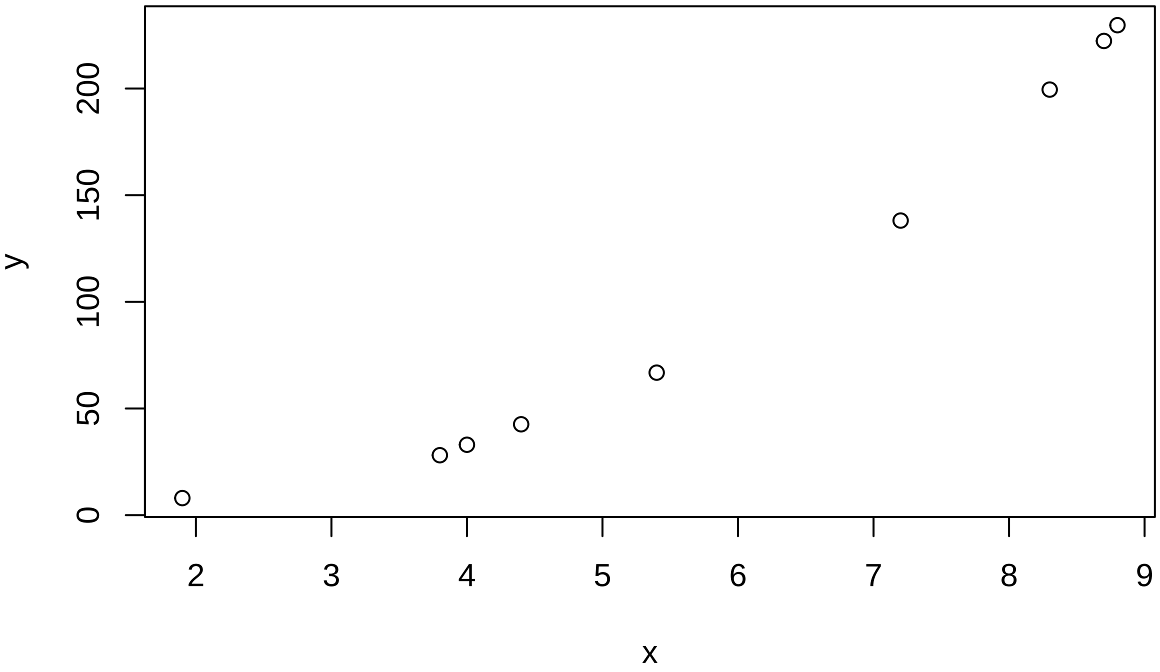
```
## [1] "/Users/moka/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/14Graficos"
```

```
# Funcion plot.
```

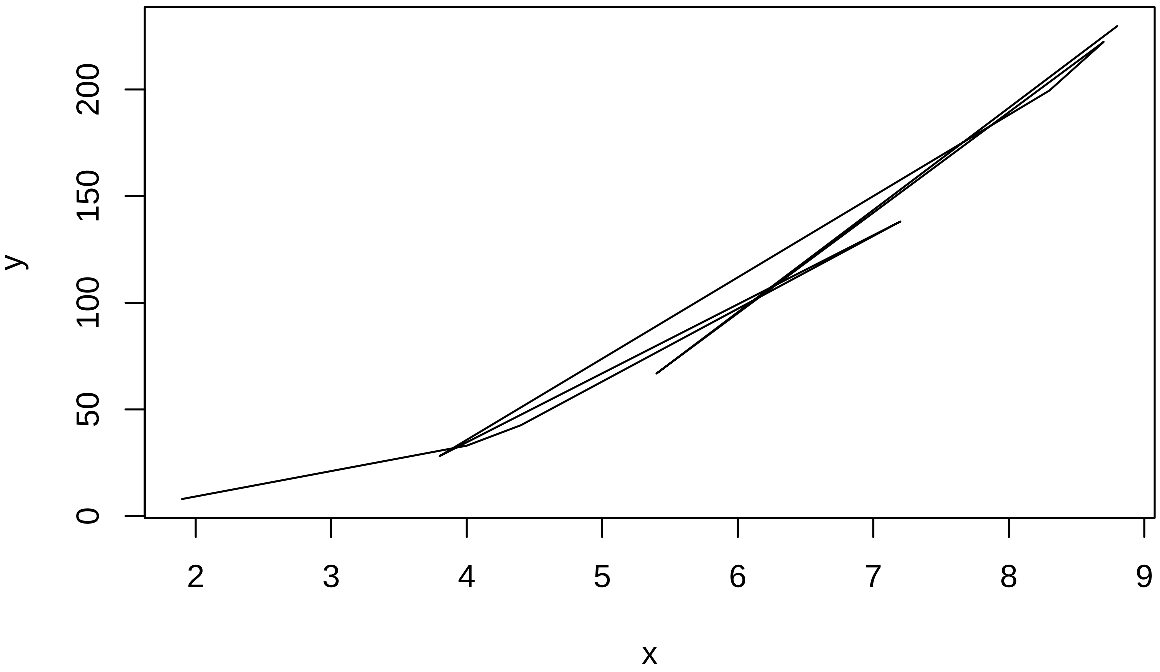
```
x<-c(1.9,4.0,4.4,7.2,3.8,8.3,8.7,5.4,8.8)
plot(x) # we only gave one vector, amnd R takes the order as given in the vector
```



```
y<-c(8,33,42.6,138.1,28.1,199.5,222.3,66.8,229.7)
plot(x,y) # En este caso tenemos
```



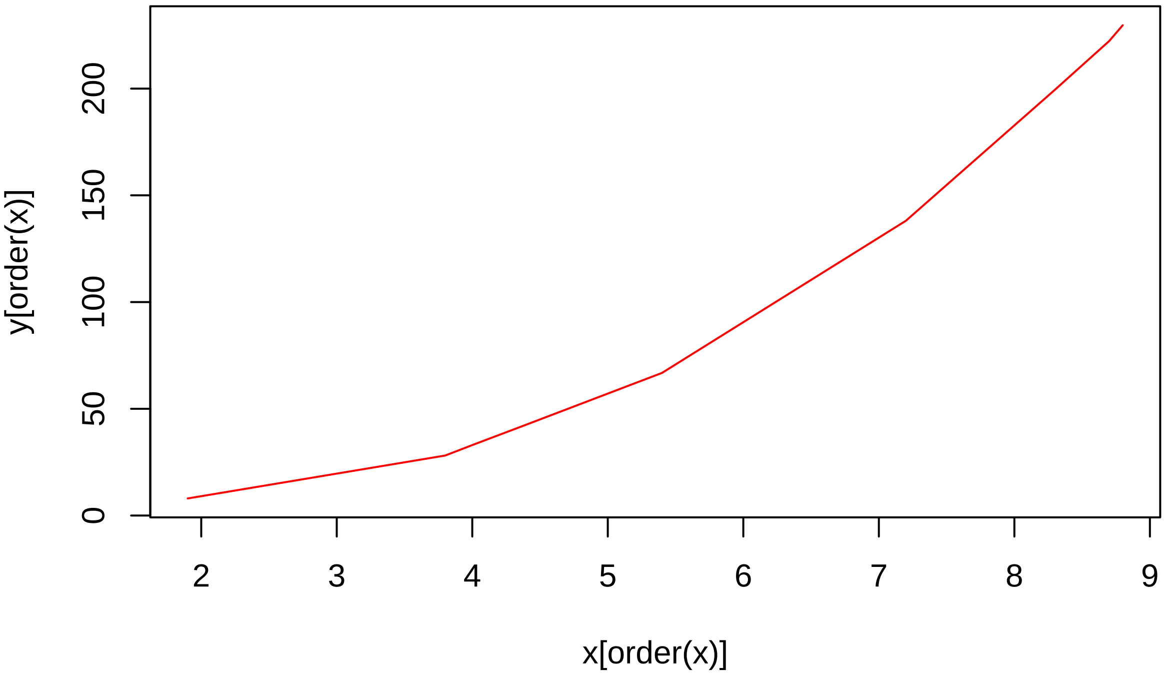
```
plot(x,y,type="l") # En este caso tenemos un grafico que no tiene sentido porque el orden que R esta to
```



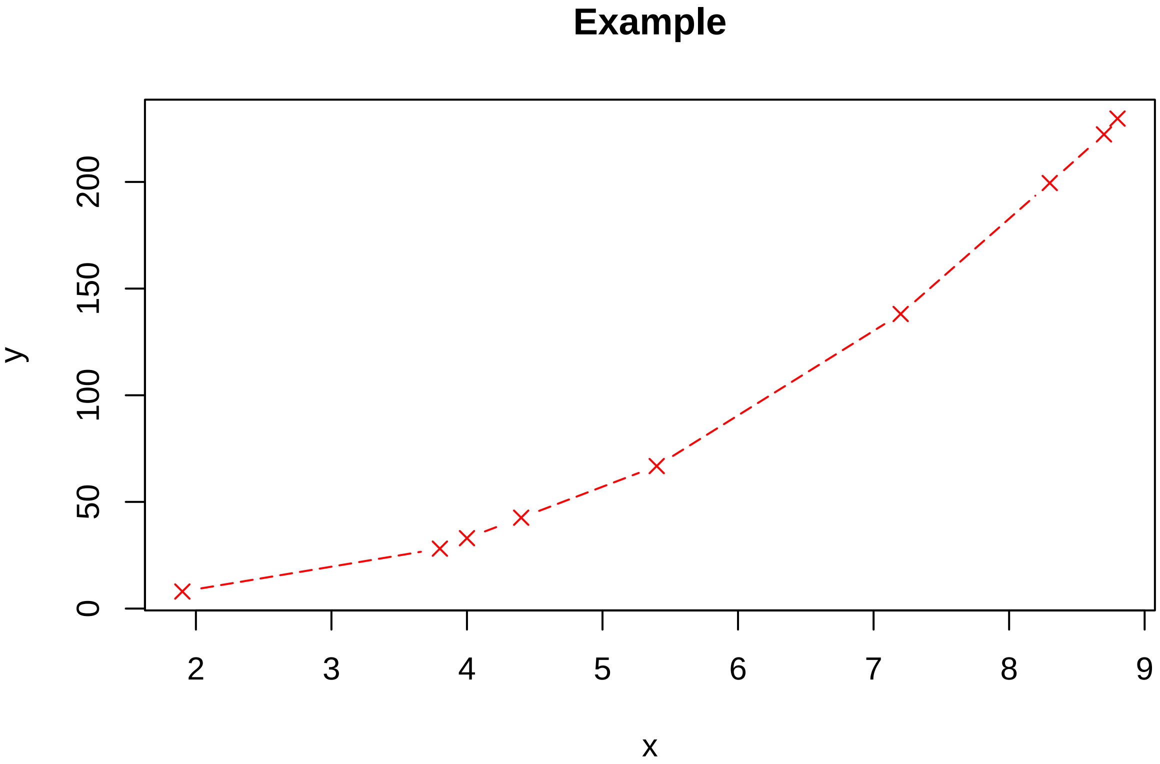
```
# Entonces necesitamos ordenar los valores
order(x)
```

```
## [1] 1 5 2 3 8 4 6 7 9
```

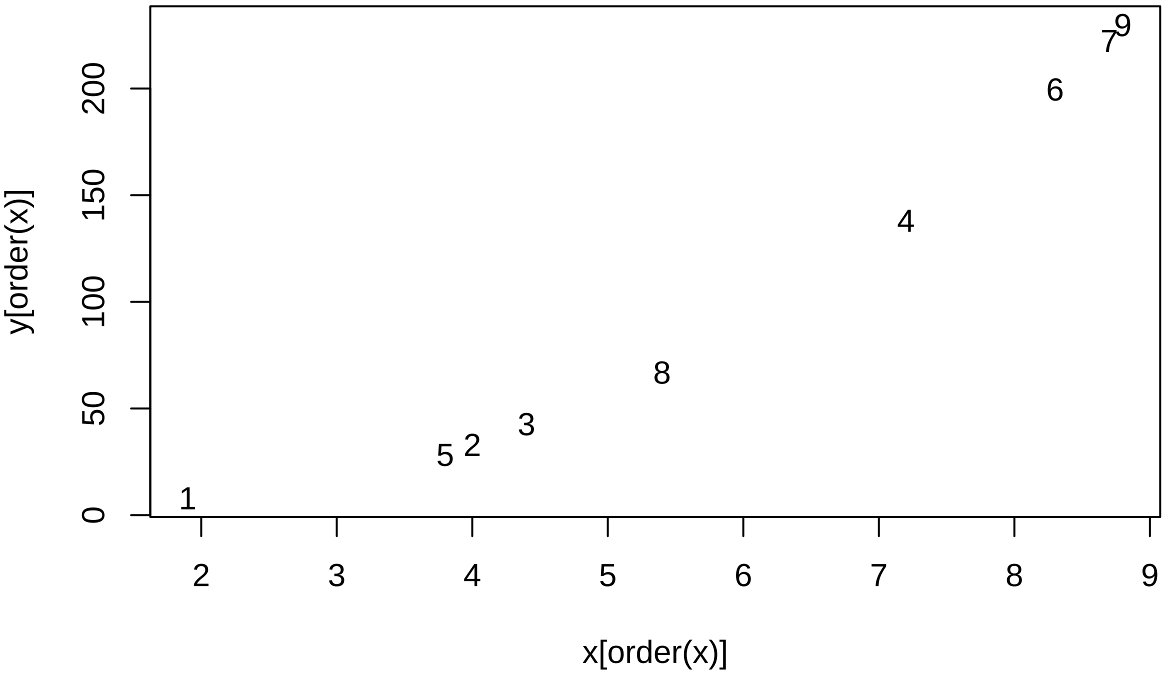
```
plot(x[order(x)],y[order(x)],type='l',col="red")
```



```
# Hay muchas mas opciones
?plot
plot(x[order(x)],y[order(x)],type='b',col="red",xlab="x",ylab="y",main="Example",lty=5,pch=4)
```



```
plot(x[order(x)],y[order(x)],type='n')
text(x[order(x)],y[order(x)],order(x))
```

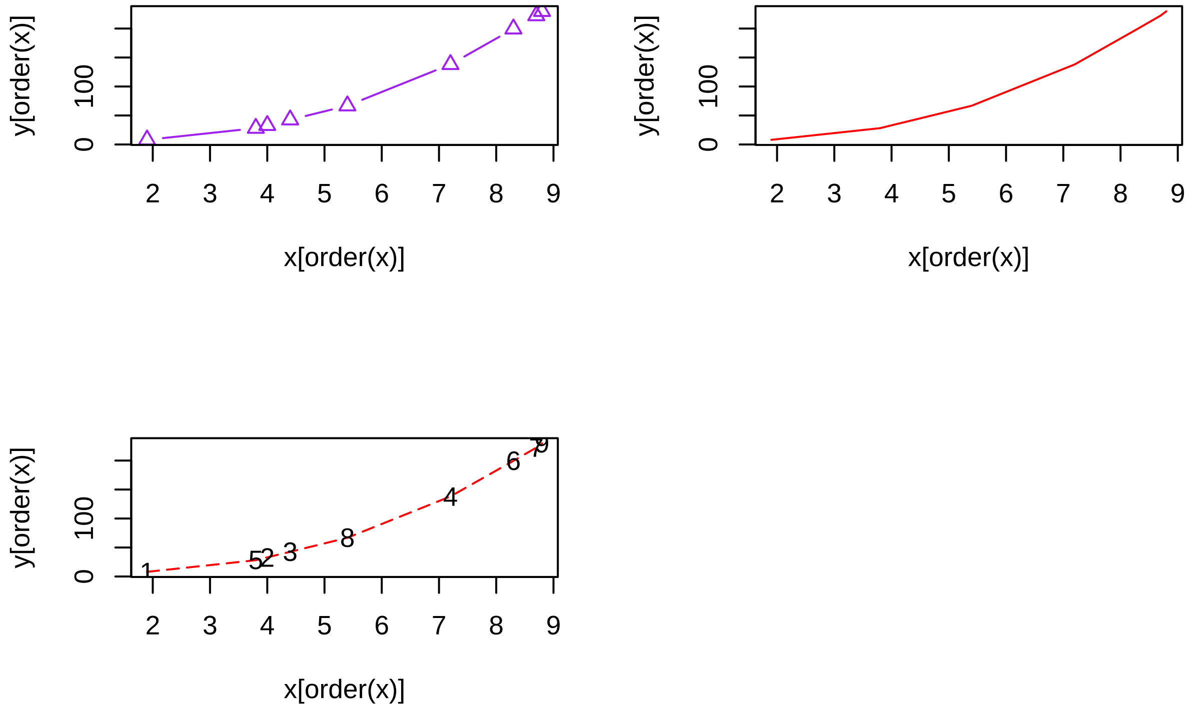


```
# Grid of plots
```

```
par(mfrow=c(2,2))
plot(x[order(x)],y[order(x)],type='b',col="purple",pch=2)
plot(x[order(x)],y[order(x)],type='l',col="red",pch=5)
plot(x[order(x)],y[order(x)],type='l',col="red",pch=5,lty=5)
text(x,y,1:9)
par("mfrow")
```

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

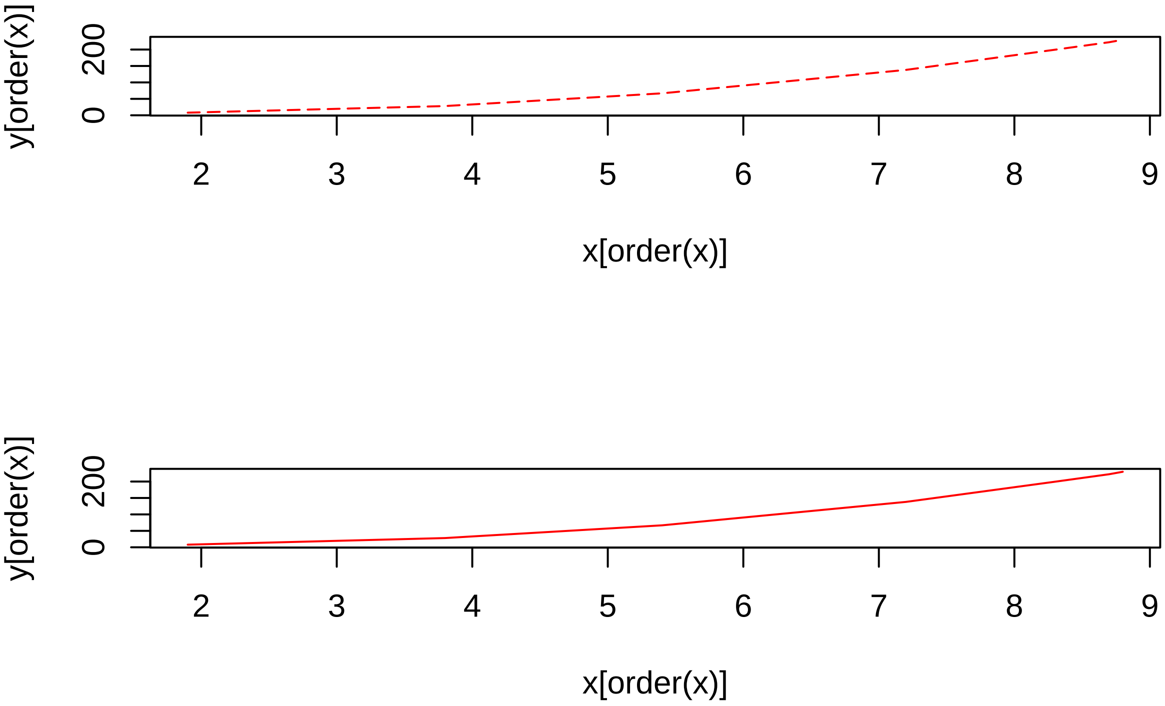
```
?par
par(mfrow=c(1,1))
```



```
plot(x[order(x)],y[order(x)],type='l',col="red",pch=5,lty=5)

par(mfrow=c(1,2))
plot(x[order(x)],y[order(x)],type='l',col="red",pch=5,lty=5)
plot(x[order(x)],y[order(x)],type='l',col="red",pch=5)

par(mfcol=c(2,1))
plot(x[order(x)],y[order(x)],type='l',col="red",pch=5,lty=5)
plot(x[order(x)],y[order(x)],type='l',col="red",pch=5)
```



```r
#The function par has the following default parameters
par()
```

```
## $xlog
## [1] FALSE
##
## $ylog
## [1] FALSE
##
## $adj
## [1] 0.5
##
## $ann
## [1] TRUE
##
## $ask
## [1] FALSE
##
## $bg
## [1] "transparent"
##
## $bty
## [1] "o"
##
## $cex
## [1] 1
##
## $cex.axis
## [1] 1
##
## $cex.lab
## [1] 1
##
## $cex.main
## [1] 1.2
##
## $cex.sub
## [1] 1
##
## $cin
## [1] 0.15 0.20
##
## $col
## [1] "black"
##
## $col.axis
## [1] "black"
##
## $col.lab
## [1] "black"
##
## $col.main
## [1] "black"
##
```

```
## $col.sub
## [1] "black"
##
## $cra
## [1] 10.8 14.4
##
## $crt
## [1] 0
##
## $csi
## [1] 0.2
##
## $cxy
## [1]   0.2125095 116.7980488
##
## $din
## [1] 6.5 4.5
##
## $err
## [1] 0
##
## $family
## [1] ""
##
## $fg
## [1] "black"
##
## $fig
## [1] 0.0 1.0 0.0 0.5
##
## $fin
## [1] 6.50 2.25
##
## $font
## [1] 1
##
## $font.axis
## [1] 1
##
## $font.lab
## [1] 1
##
## $font.main
## [1] 2
##
## $font.sub
## [1] 1
##
## $lab
## [1] 5 5 7
##
## $las
## [1] 0
##
```

```
## $lend
## [1] "round"
##
## $lheight
## [1] 1
##
## $ljoin
## [1] "round"
##
## $lmitre
## [1] 10
##
## $lty
## [1] "solid"
##
## $lwd
## [1] 1
##
## $mai
## [1] 1.02 0.82 0.82 0.42
##
## $mar
## [1] 5.1 4.1 4.1 2.1
##
## $mex
## [1] 1
##
## $mfcol
## [1] 2 1
##
## $mfg
## [1] 2 1 2 1
##
## $mfrow
## [1] 2 1
##
## $mgp
## [1] 3 1 0
##
## $mkh
## [1] 0.001
##
## $new
## [1] FALSE
##
## $oma
## [1] 0 0 0 0
##
## $omd
## [1] 0 1 0 1
##
## $omi
## [1] 0 0 0 0
##
```

```
## $page
## [1] TRUE
##
## $pch
## [1] 1
##
## $pin
## [1] 5.26 0.41
##
## $plt
## [1] 0.1261538 0.9353846 0.4533333 0.6355556
##
## $ps
## [1] 12
##
## $pty
## [1] "m"
##
## $smo
## [1] 1
##
## $srt
## [1] 0
##
## $tck
## [1] NA
##
## $tcl
## [1] -0.5
##
## $usr
## [1]   1.624   9.076  -0.868 238.568
##
## $xaxp
## [1] 2 9 7
##
## $xaxs
## [1] "r"
##
## $xaxt
## [1] "s"
##
## $xpd
## [1] FALSE
##
## $yaxp
## [1]   0 200   4
##
## $yaxs
## [1] "r"
##
## $yaxt
## [1] "s"
##
```

```
## $ylbias
## [1] 0.2
```

```
# More about plot function
```

```
# If we want to know the different plot functions, we can use apropos.
apropos("plot")
```

```
##  [1] ".rs.api.savePlotAsImage"                ".rs.replayNotebookPlots"                ".rs.reticula
##  [4] ".rs.reticulate.matplotlib.onLoaded"     ".rs.reticulate.matplotlib.showHook"     "assocplot"
##  [7] "barplot"                                "barplot.default"                        "biplot"
## [10] "boxplot"                                "boxplot.default"                        "boxplot.matr
## [13] "boxplot.stats"                          "bwplot"                                 "cdplot"
## [16] "coplot"                                 "densityplot"                            "fluxplot"
## [19] "fourfoldplot"                           "interaction.plot"                       "lag.plot"
## [22] "matplot"                                "monthplot"                              "mosaicplot"
## [25] "plot"                                   "plot"                                   "plot.default
## [28] "plot.design"                            "plot.ecdf"                              "plot.functio
## [31] "plot.new"                               "plot.spec.coherency"                    "plot.spec.ph
## [34] "plot.stepfun"                           "plot.ts"                                "plot.window"
## [37] "plot.xy"                                "preplot"                                "qqplot"
## [40] "recordPlot"                             "replayPlot"                             "savePlot"
## [43] "screeplot"                              "spineplot"                              "stripplot"
## [46] "sunflowerplot"                          "termplot"                               "ts.plot"
## [49] "xyplot"
```

```
# For time series object, we use plot.ts()
```

```
# Histograms
```

```
load("~/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/datos/procesados/trafico1.Rdata")
head(trafico)
```

```
##   year month date_of_month day_of_week     autos
## 1 2000     1             1           6  9084.165
## 2 2000     1             2           7  8005.351
## 3 2000     1             3           1 11363.158
## 4 2000     1             4           2 13030.962
## 5 2000     1             5           3 12559.135
## 6 2000     1             6           4 12465.477
```

```
str(trafico)
```

```
## 'data.frame':    5479 obs. of  5 variables:
##  $ year         : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
##  $ month        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ date_of_month: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ day_of_week  : int  6 7 1 2 3 4 5 6 7 1 ...
##  $ autos        : num  9084 8005 11363 13031 12559 ...
```

```
attach(trafico)
```

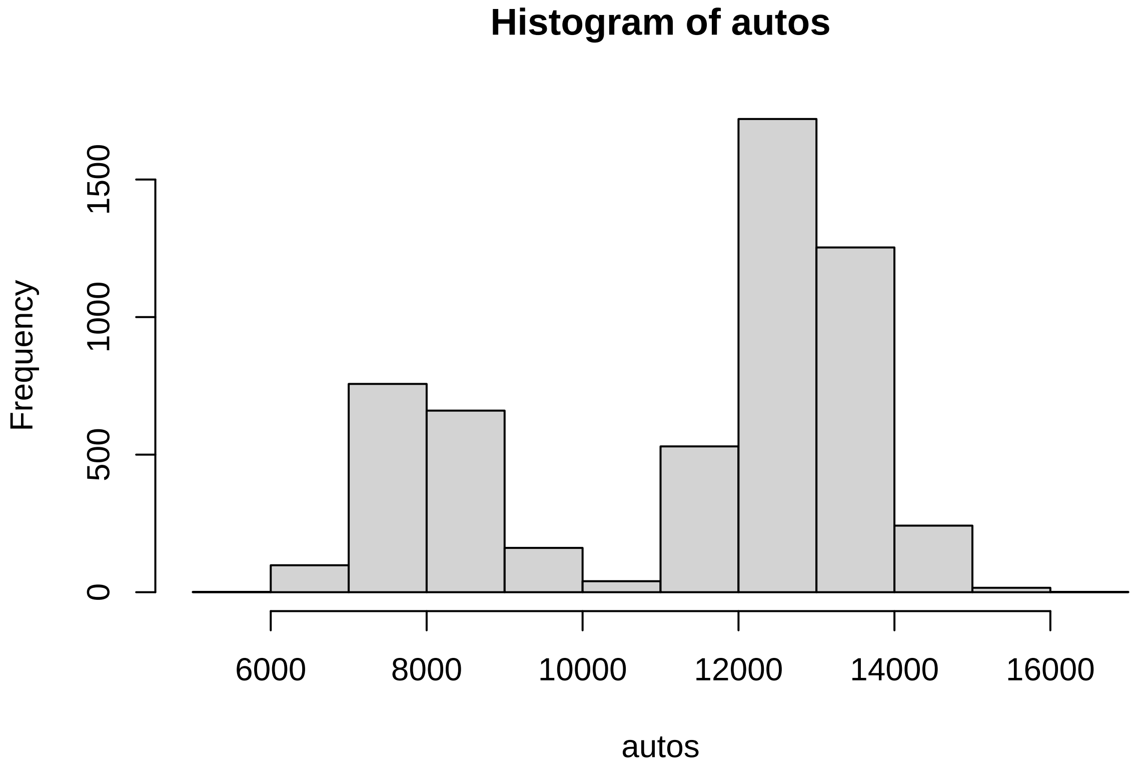
```
## The following objects are masked from trafico (pos = 3):
##
##     autos, date_of_month, day_of_week, month, year

## The following objects are masked from trafico (pos = 4):
##
##     autos, date_of_month, day_of_week, month, year
```

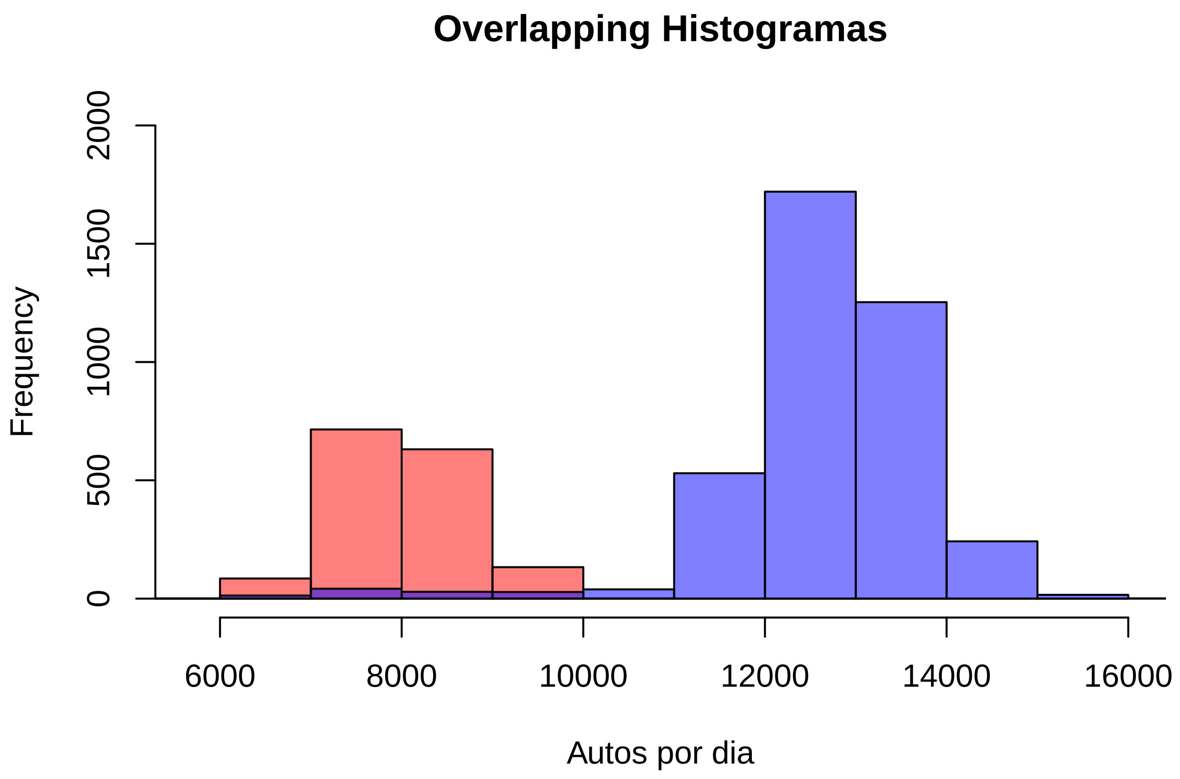
```
summary(trafico)
```

```
##       year          month         date_of_month    day_of_week     autos
##  Min.   :2000   Min.   : 1.000   Min.   : 1.00   Min.   :1    Min.   : 5728
##  1st Qu.:2003   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.:2    1st Qu.: 8740
##  Median :2007   Median : 7.000   Median :16.00   Median :4    Median :12343
##  Mean   :2007   Mean   : 6.523   Mean   :15.73   Mean   :4    Mean   :11350
##  3rd Qu.:2011   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:6    3rd Qu.:13081
##  Max.   :2014   Max.   :12.000   Max.   :31.00   Max.   :7    Max.   :16081
```

```
autosn=as.numeric(trafico$autos)
hist(autos)
```

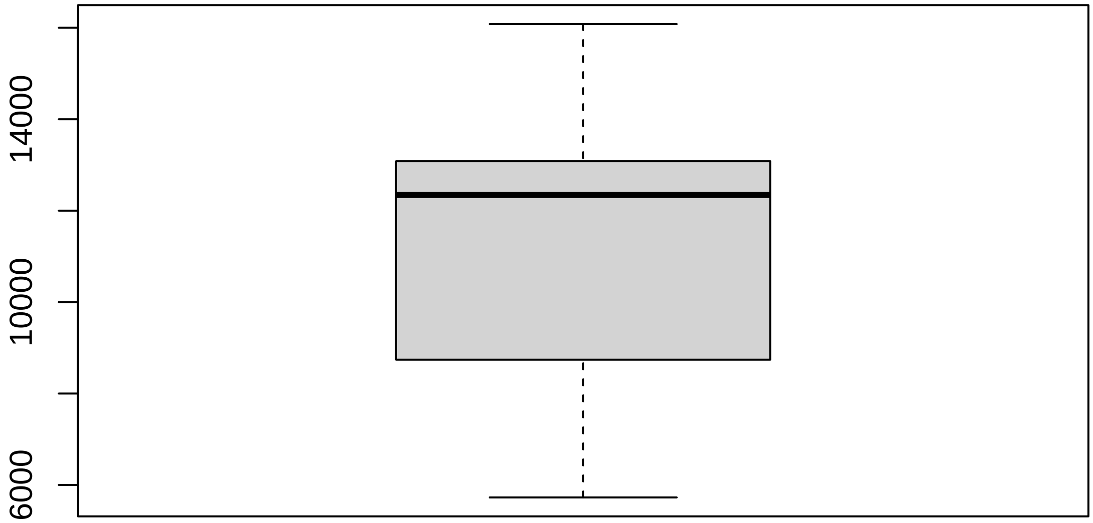


```
# We do two histograms overlaying
# Days of the week, and days of the weekend.
# Weekends: red
hist(autos[day_of_week>5],breaks=seq(5000,16000,by=1000),col=rgb(1,0,0,0.5),xlim=c(5700,16000),ylim=c(0
# Weekdays: mauve
hist(autos[day_of_week<6],breaks=seq(5000,17000,by=1000),col=rgb(0,0,1,0.5),add=T)
```

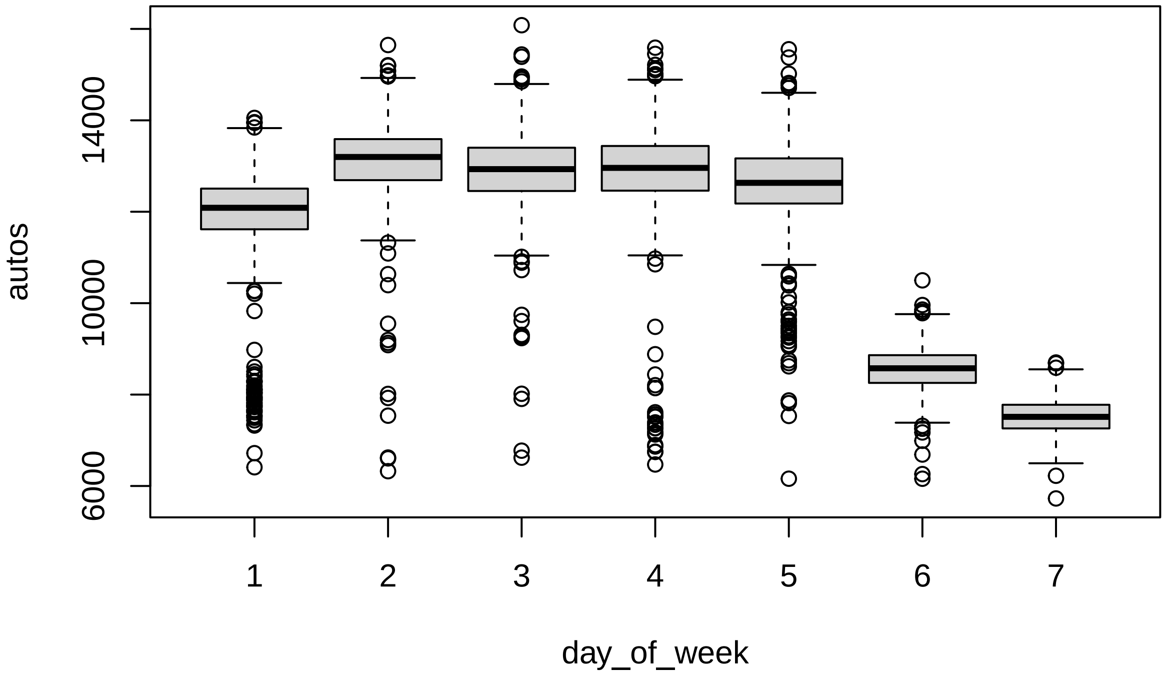


```
# Boxplots
```

```
boxplot(autos)
```



```
boxplot(autos~day_of_week)
```



```
# Barplot
```

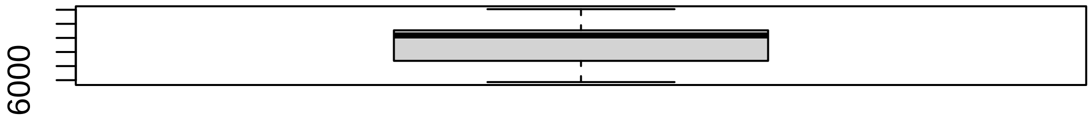
```
meansperday<- trafico %>%
  group_by(day_of_week) %>%
  summarise(ave=mean(autos)) %>%
  arrange()
vectormeans <- meansperday %>% .$ave
vectormeans
```

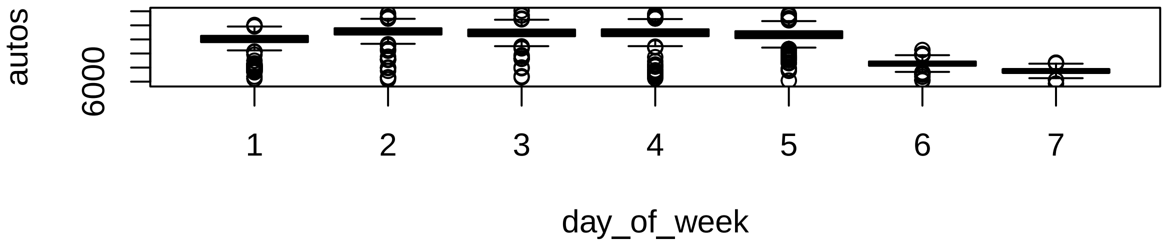
```
## [1] 11897.833 13122.472 12910.729 12845.740 12596.179  8562.539  7518.424
```

```
pdf("weekbars.pdf") # We can add this line to get the pdf file of our graph
bp=barplot(vectormeans,xlab="Day of the week")
axis(at=bp,labels=c("M","T","W","Th","F","Sa","Su"),side=1)
dev.off()
```

```
## RStudioGD
##         2
```

```
dir()
```





```
##  [1] "14Graficos_cheatsheet_files"    "14Graficos_cheatsheet.aux"      "14Graficos_cheatsheet.html"
##  [5] "14Graficos_cheatsheet.R"        "14Graficos_cheatsheet.spin.R"   "14Graficos_cheatsheet.spin.R
##  [9] "14Graficos_cheatsheetLatex.tex" "14Graficos_conceptos.aux"       "14Graficos_conceptos.tex"
## [13] "14Graficos_lab.tex"             "14Graficos_solution.R"          "weekbars.pdf"
```

```
?dev.off

# Note: you can use Rmarkdown to get a report
# No quitar el comentario de la linea inferior. Solamente copiar en la consola para que ejecute
#rmarkdown::render("14Graficos_cheatsheet.R",c("pdf_document","html_document"))
```