# Boxplots

Gaston Sanchez

**Learning Objectives**

- Learn how to read boxplots
- Learn about the `boxplot()` function
- How to graph boxplots with `ggplot2`

## Introduction

Quantitative variables can be summarized using two groups of measures: 1) center, and 2) spread. Just like there are various measures of center (e.g. average, median, mode), we also have several measures of spread or variability:

- range
- interquartile range
- standard deviation (and variance)

In this tutorial we'll use the data set `mtcars` that comes in R.

```
head(mtcars)
```

```
##                   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
```

Let's analyze the variable `mpg` miles per gallon.

The function `summary()` produces basic summary statistics: the five-number summary, plus the mean:

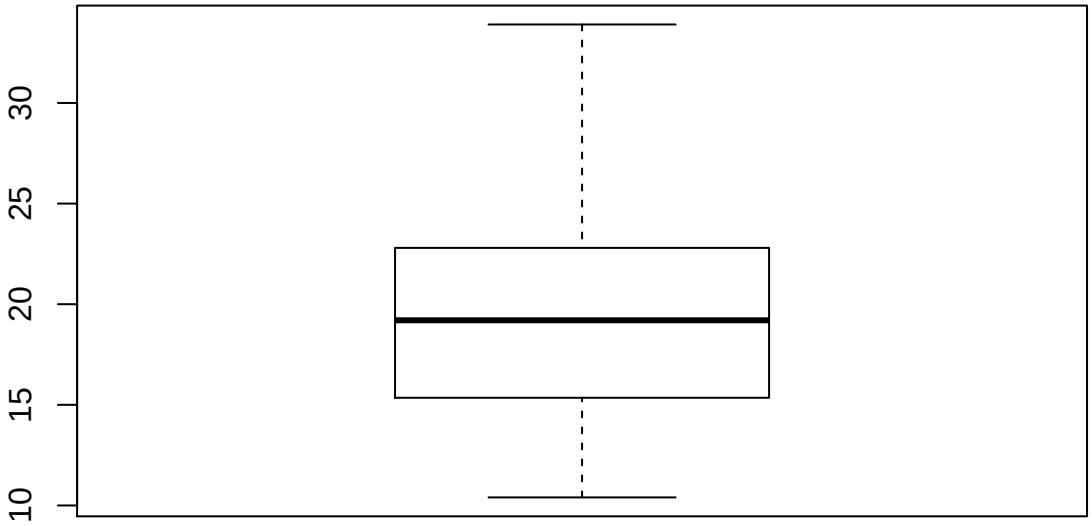
```
summary(mtcars$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.40   15.42   19.20   20.09   22.80   33.90
```

A boxplot, or more formally box-and-whisker plot, is based on the five-number summary: minimum, 1st quartile, median, 3rd quartile, and maximum.

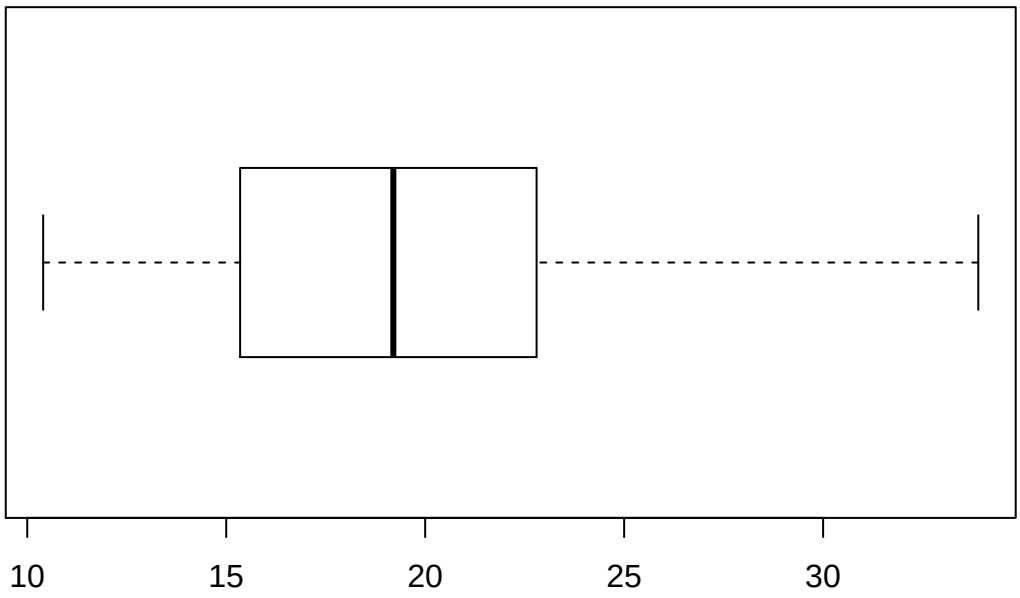
R has the built-in function `boxplot()` that allows you to make boxplots. You just need to pass it a vector, and R will graph a boxplot vertically oriented:

```r
boxplot(mtcars$mpg)
```




You can set the argument `horizontal = TRUE` to get a boxplot horizontally oriented:

```r
boxplot(mtcars$mpg, horizontal = TRUE)
```




In fact, you can actually store the output of `boxplot()`, for example:

```r
bb = boxplot(mtcars$mpg)
```

The object `bb` is an object of class `"boxplot"` which contains various elements:

```r
bb
```

```
## $stats
##       [,1]
```

```
## [1,] 10.40
## [2,] 15.35
## [3,] 19.20
## [4,] 22.80
## [5,] 33.90
##
## $n
## [1] 32
##
## $conf
##          [,1]
## [1,] 17.11916
## [2,] 21.28084
##
## $out
## numeric(0)
##
## $group
## numeric(0)
##
## $names
## [1] "1"
```

The first element `stats` contains the five-number summary:

```
# five number summary
bb$stats
```

```
##       [,1]
## [1,] 10.40
## [2,] 15.35
## [3,] 19.20
## [4,] 22.80
## [5,] 33.90
```

```
# compare to summary()
summary(mtcars$mpg)
```

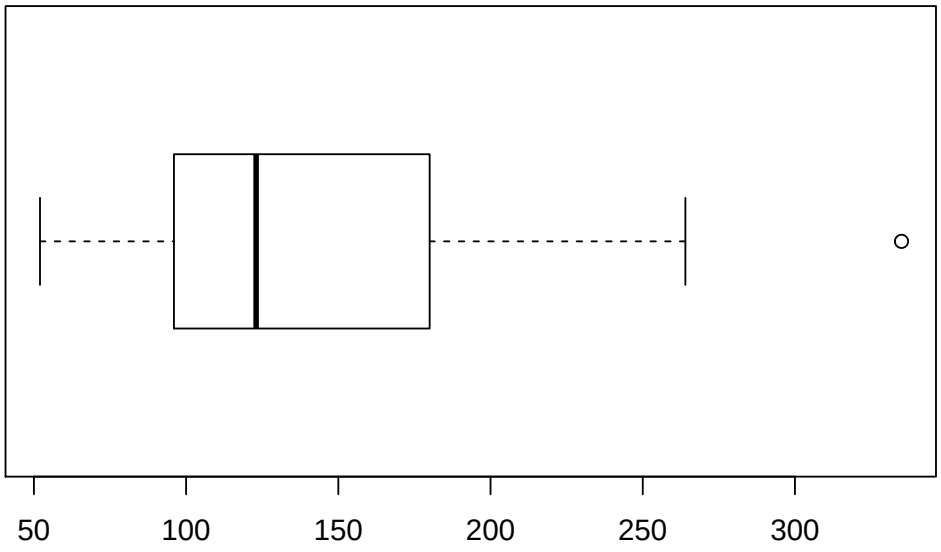
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.40   15.42   19.20   20.09   22.80   33.90
```

### Fences

The function `boxplot()` has an argument `range`. This argument determines how far the plot whiskers extend out from the box. By default `range = 1.5`, this means that the whiskers extend to the most extreme data point which is no more than 1.5 times the interquartile

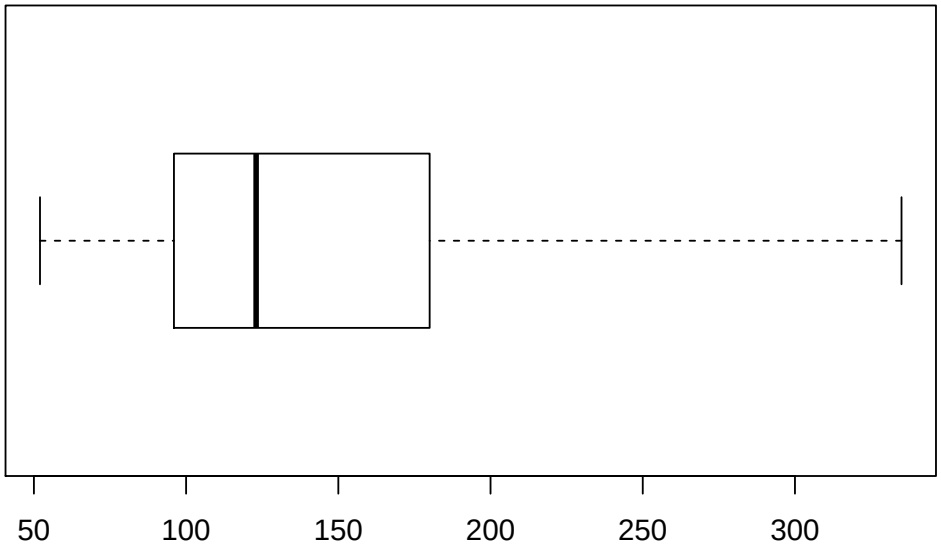
range from the box.

```
# default boxplot
# (whiskers may not extend to the most extreme data points)
boxplot(mtcars$hp, horizontal = TRUE)
```




A value of zero (e.g. `range = 0`) causes the whiskers to extend to the data extremes.

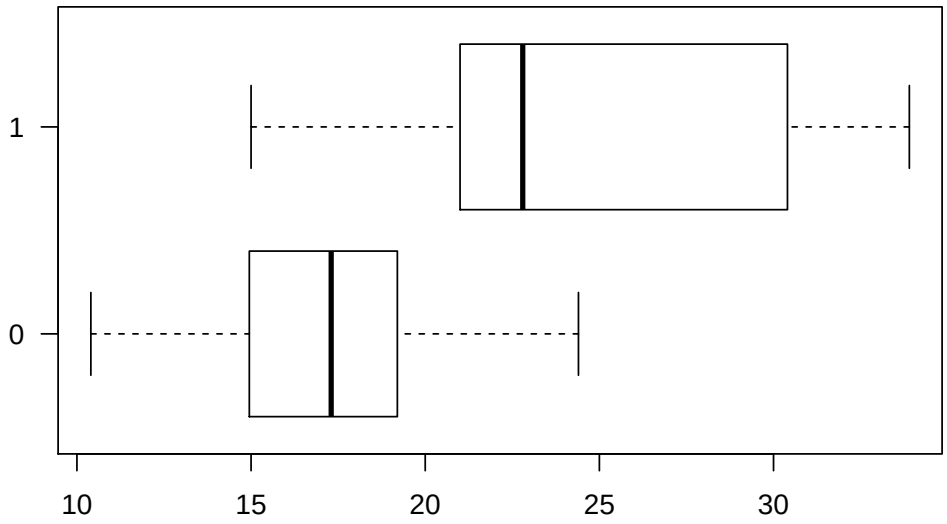
```
# boxplot with unmodified whiskers
# (whiskers extend to the most extreme data points)
boxplot(mtcars$hp, horizontal = TRUE, range = 0)
```



### Formulas with `boxplot()`

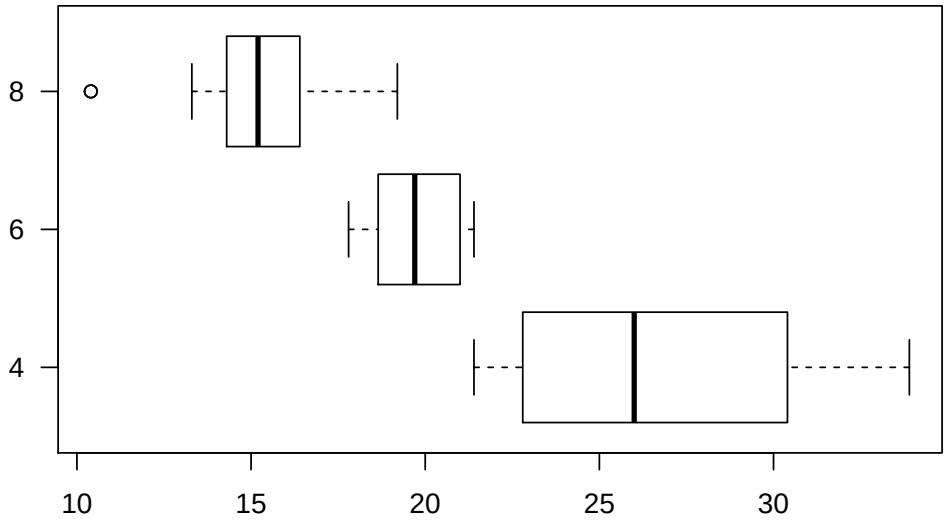
An interesting feature of `boxplot()` is that you can pass R formulas. For example the variable (column) `am` refers to the automatic transmission. This variable takes two values: 0 if a car is automatic, 1 if the transmission is manual (stick).

```
# boxplots of mpg by transmission
boxplot(mpg ~ am, data = mtcars, horizontal = TRUE, las = 1)
```




A similar boxpot can be produced for the number of cylinders `cyl`

```
# boxplots of mpg by cylinders
boxplot(mpg ~ cyl, data = mtcars, horizontal = TRUE, las = 1)
```

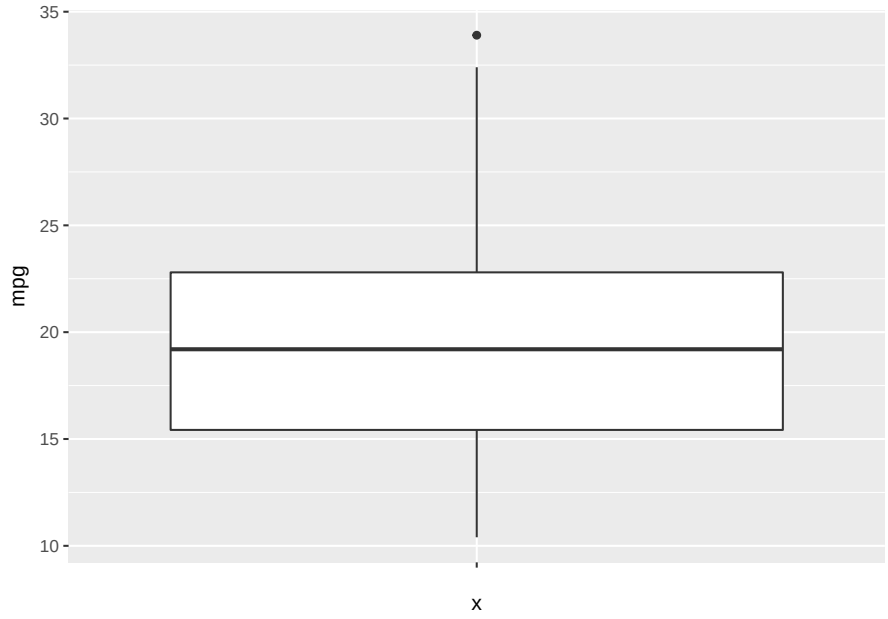


### Boxplots with "ggplot2"

You can also use the package "ggplot2" to graph boxplots. As usual, in order to use the functions in ggplot2 remember to load the package with the library() function.

Let's see a basic example of a boxplot of the variable mpg:

```
ggplot(data = mtcars, aes(x = "", y = mpg)) +
  geom_boxplot()
```

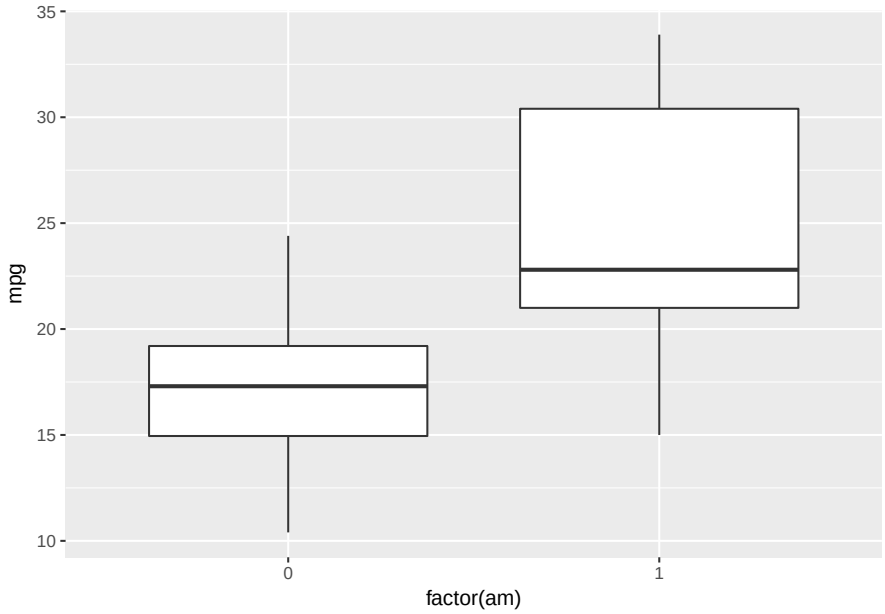



In order to make a boxplot with ggplot() of a single variable, you need to:

- specify the data frame containing the variable of interest, in this example: mtcars
- specify visual attributes x and y via aes()
- x = "" indicates that the x-axis is "empty" in the sense that we don't want ggplot() to map anything in this axis.
- y = mpg indicates that the variable of interest is mpg, and that the boxplot will be constructed with this variable

You can also take a qualitative variable and take it into account when making boxplots of a quantitative variable. For example, say we are interested in the variable am that has to do with the type of transmission: automatic or manual (stick shift). And we want see the distributions of mpg distinguishing between automatic and manual transmissions. Here's how to do that in "ggplot2":

```
ggplot(data = mtcars, aes(x = factor(am), y = mpg)) +
  geom_boxplot()
```




Notice that we need to use the `factor()` function to indicate that `am` should be treated as a categorical variable instead of a numeric-quantitative one.