Visualizing Data and Exploratory Data analysis using R

Part 1: Univariate datasets

Ziv Shkedy et al. (2020)

Part 1: Introduction

Location, Spread and Shape in univariate data

- In the first part we focus on descriptive measures, numerical and graphical, to characterize and visualize the features of a particular univariate distribution.
- The following three main factors are usually used to specify a paticular distribution:
    - Location
    - Spread
    - Shape
- Each of these control different characteristics of a distribution.

R datasets for illustraions

- In order to simplify the usage of slides, the data we used for illustrations are R datasets.
- We give a short description of each data in the relevant slides.

- More details can be found with `help(dataset)` or (for datasets of the first part) in

    - The singers data: singers.
    - The airquality data: airquality.
    - The cars data: mtcars.
    - The Old Faithful Geyser Data: oldfaithful

What do we cover in this part ?

- How to tell the story in a univariate data using R ?
- A very short introduction to `ggplot2`.
    - Layers and components of a graphical deaply.
    - How to develop a cool figure ?
    - Components of visualization.
- Patterns in Univariate datasets: location speard and shape.

ggplot2 and lattice?

- The following basic graphical functions are used for visualization:
  - `plot()`
  - `lines()`
  - `hist()`
  - `boxplot()`
  - `qqnorm()`
- The following `lattice` graphical functions are used for visualization:
  - `dotplot()`
  - `histogram()`
  - `bwplot()`
  - `qqmath()`

ggplot2 and lattice?

- The following ggplot2 graphical functions are used for visualization:
    - ggplot()
    - geom_point()
    - theme_bw()
    - geom_smooth()
    - geom_histogram()
    - geom_boxplot()
    - geom="density"
    - geom_violin()
    - facet_wrap()
    - qplot()
    - stat_summary()
    - geom_density_ridges() + theme_ridges()

Online references

- Basic Skills in Visualising Data and Exploratory Data Analysis Using R - An interactive online book for the course: BookVD.
- Book: R Graphics Cookbook, 2nd edition by Winston Chang: RGraphics.
- Website: From Data to Viz: Viz.

The `xaringan` package

- Slides were produced using the `xaringan` package.
- Online book chapter about the `xaringan` package: xaringan.

Part 2: `ggplot2`: a short introduction

Focus: Layers

The layers of a `ggplot2` figure

- A key idea behind `ggplot2` is that it allows to easily building up a complex plot layer by layer.
- Each layer adds an extra level of information to the plot.
- In that way we can build sophisticated plots tailored to the problem at hand.

The `mtcars` data

- The data gives information about fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

```
head(mtcars)
```

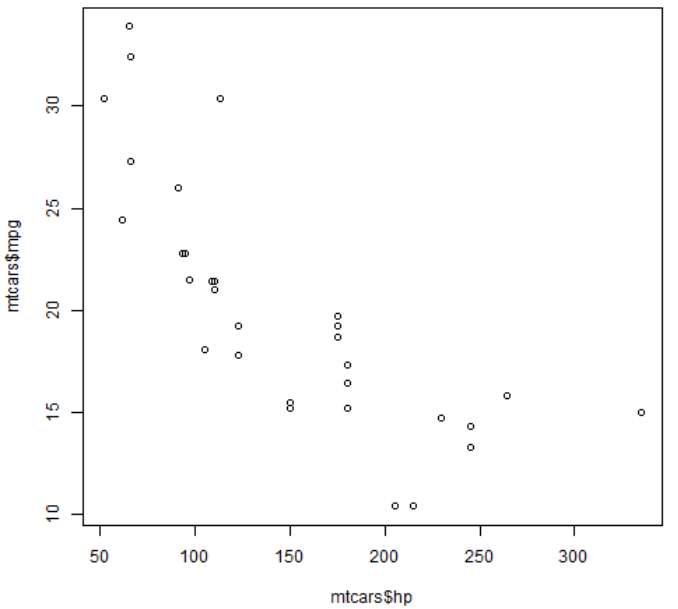
```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 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
```

- For our example: `mpg`, `hp` and `cyl`.

The `mtcars` data

- Bivariate data (miles/(US) per gallon, horsepower).
- Basic scatterplot in R: mgp Vs. hp.
- `plot()`.
- Our aim: fit a linear regression model for mpg with Horsepower as predictor.

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



Linear regression

- A simple linear regression model in R.

```
fit.lm<-lm(mtcars$mpg~mtcars$hp)
```

- Fitted model

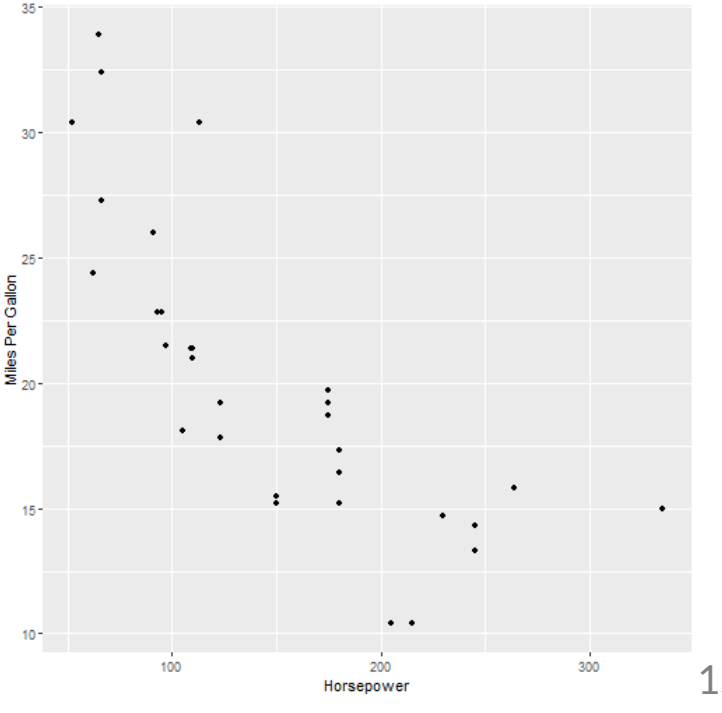
```
summary(fit.lm)
```

```
##
## Call:
## lm(formula = mtcars$mpg ~ mtcars$hp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7121 -2.1122 -0.8854  1.5819  8.2360
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 30.09886    1.63392  18.421  < 2e-16 ***
## mtcars$hp   -0.06823    0.01012  -6.742 1.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.863 on 30 degrees of freedom
## Multiple R-squared:  0.6024,    Adjusted R-squared:  0.5892
## F-statistic: 45.46 on 1 and 30 DF,  p-value: 1.788e-07
```

First layer: basic plot

- Basic plot using `ggplot2`.
- Scaterplot and labels.

```
library(ggplot2)
gg <- ggplot(mtcars, aes(hp, mpg)) +
        geom_point() +
        labs(x = "Horsepower",y= "Mile
gg
```



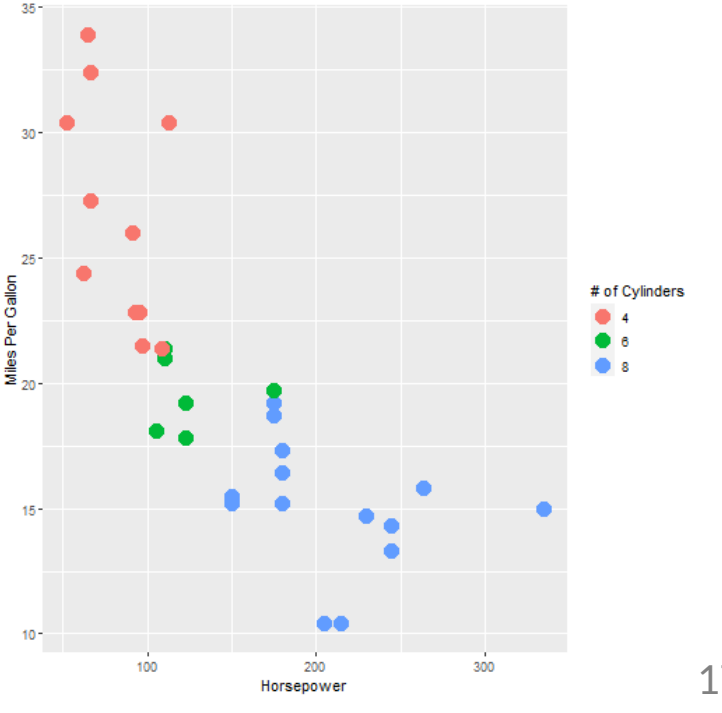
What is `aes()`?

- `aes()`: generate aesthetic mappings that describe how variables in the data are mapped to visual properties (Aesthetics) of geoms.
- `aes()` is a quoting function. This means that its inputs are quoted to be evaluated in the context of the data.
- `aes(hp, mpg)`: use the vraiables `mpg` and `hp` in the plot.
- ggplot2 Cheat Sheet: ggplot2.

Second layer: add Cylinders information

- Add colors to the data points (by Cylinders number).
- `aes(color=as.factor(cyl)), size=5`.

```
library(ggplot2)
gg <- ggplot(mtcars, aes(hp, mpg)) +
        geom_point(aes(color=as.factor
        labs(x = "Horsepower",y= "Mile
gg
```



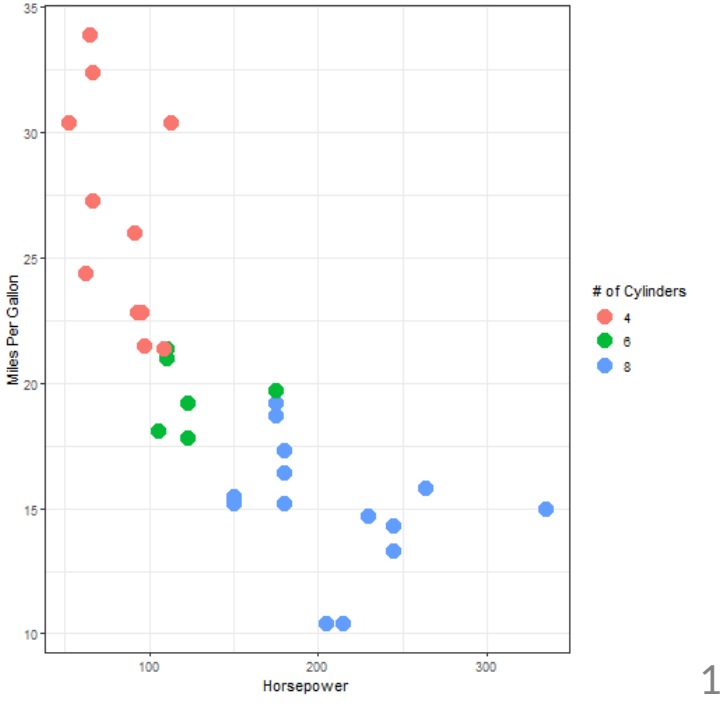
What is geom ?

- A ggplot2 `geom` tells the plot how do we want to display the data in R.
- For example, `geom_bar()` makes a bar chart.
- `geom_point(aes(color=as.factor(cyl)), size=5)`: produce a scaterplot with points.
- `aes(color=as.factor(cyl))`: use different colors acording to the `cyl` levels.
- ggplot2 Cheat Sheet: ggplot2.

Third layer: change the background

- Change the theme (to make the figure clear).
- `theme_bw()`.

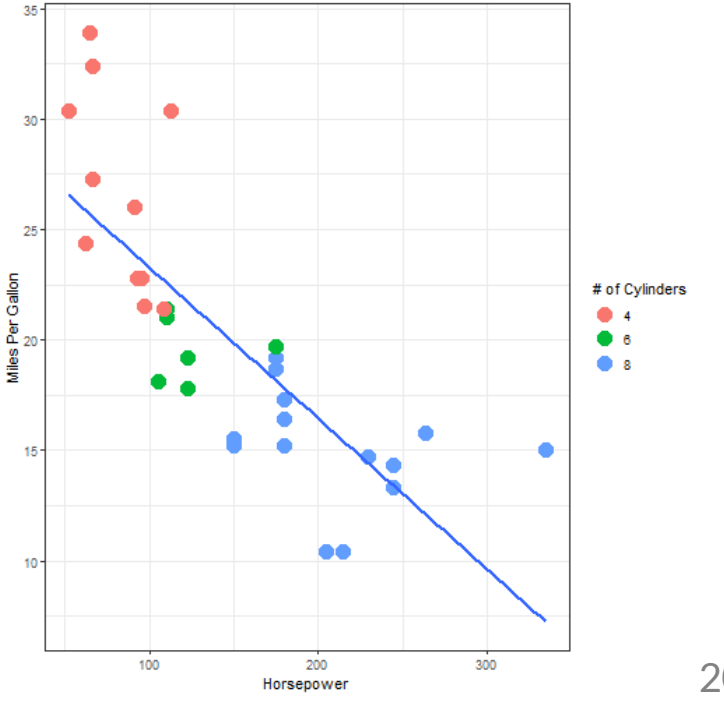
```
library(ggplot2)
gg <- ggplot(mtcars, aes(hp, mpg)) +
        geom_point(aes(color=as.factor
        theme_bw()
gg
```



Fourth layer: add the regression line

- Add a regression line.
- `geom_smooth(method="lm", se=FALSE)`

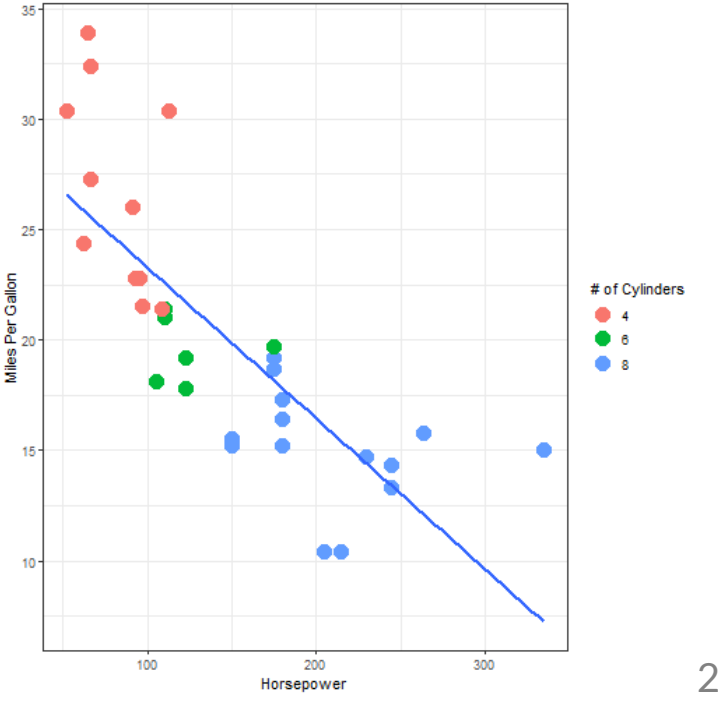
```
library(ggplot2)
gg <- ggplot(mtcars, aes(hp, mpg)) +
        geom_point(aes(color=as.factor
        labs(x = "Horsepower",y= "Mile
gg
```



Fourth layer: lines by line

- ggplot(data,aes(variables)).
- geom_point(size).
- aes(color the data points).
- geom_smooth(add regression line).
- labs(add labels).

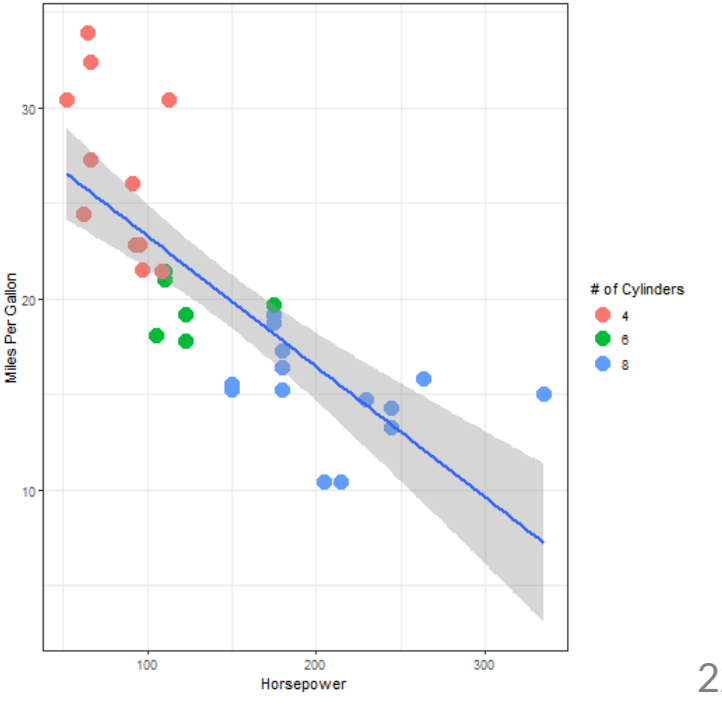
```
library(ggplot2)
gg <- ggplot(mtcars, aes(hp, mpg)) +
        geom_point(aes(color=as.factor
        labs(x = "Horsepower",y= "Mile
gg
```



Fourth layer: add C.Is

- Add C.Is to the regression line.
- `geom_smooth(method="lm", se=TRUE)`

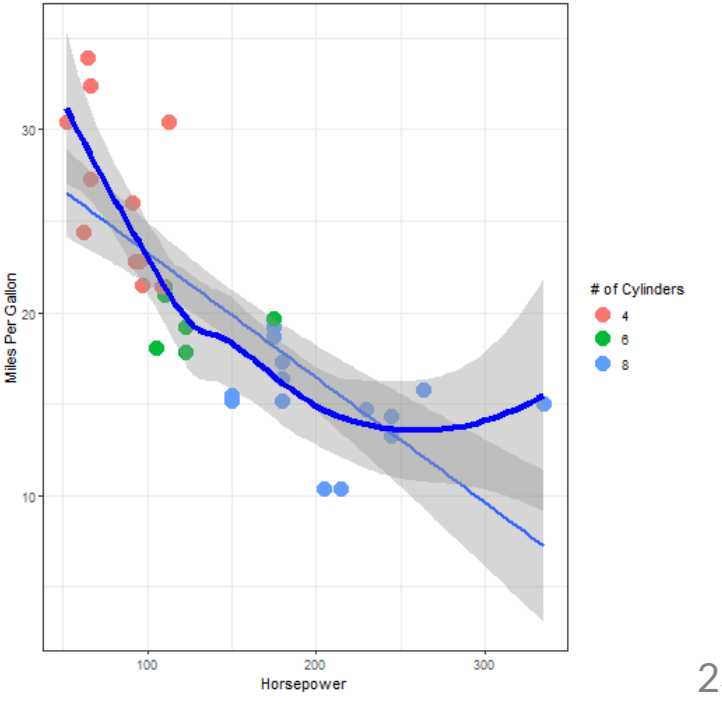
```
library(ggplot2)
gg <- ggplot(mtcars, aes(hp, mpg)) + g
        geom_smooth(method="lm", se=TR
        labs(x = "Horsepower",y= "Mile
gg
```



Fifth layer: add a smoother

- Add C.I to the regression line.
- `geom_smooth(method="loess"", se=TRUE)`

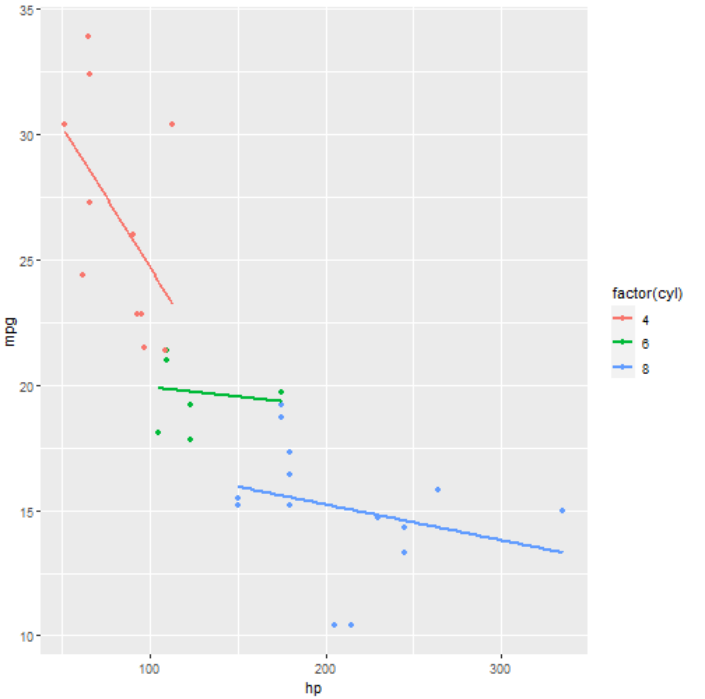
```
library(ggplot2)
gg <- ggplot(mtcars, aes(hp, mpg)) +ge
        geom_smooth(method="lm", se=TR
        labs(x = "Horsepower",y= "Mile
gg
```



Visualize patterns across cylinders

- How the dependence between mpg and hp with Cylinder numbers.

```
qplot(hp,mpg,data = mtcars, colour = f
geom_smooth(method = "lm",se = F)
```



Part 3: Location

Focus: the center of the distribution

Graphical and numerical summaries

YouTube tutorial: calculating Mean, Median, Range, Minimum and Maximum using R studio

For a short online YouTube tutorial, by BIO-RESEARCH, about the mean, median etc, using R see YTVD4.

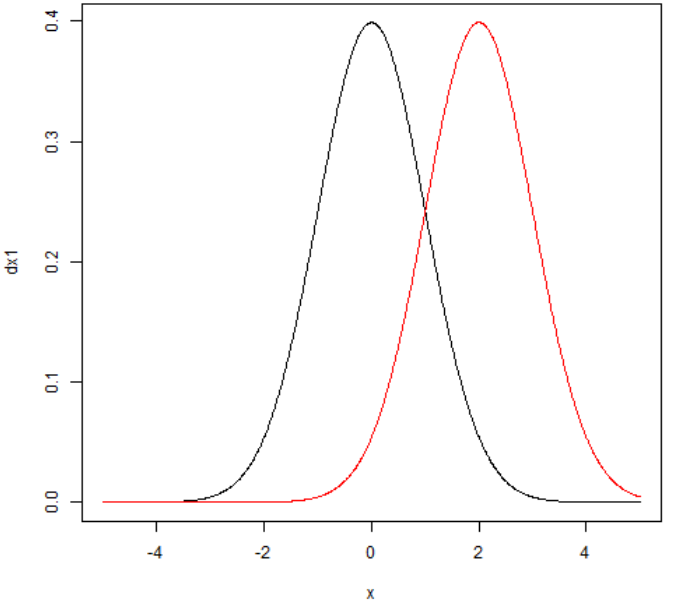
YouTube tutorial: dotplot

- For a short online YouTube tutorial, by ramstatvid, about dotplot using the \texttt{lattice} package see YTVD5.
- For a short online YouTube tutorial, by ramstatvid, about dotplot using the \texttt{gg2plot} package see YTVD6

Location

- Location is the center of the distribution.
- The figure presents distributions with different locations.
- Two normal densities with mean equal to 0 (the black line) and mean equal to 2 (the red line).

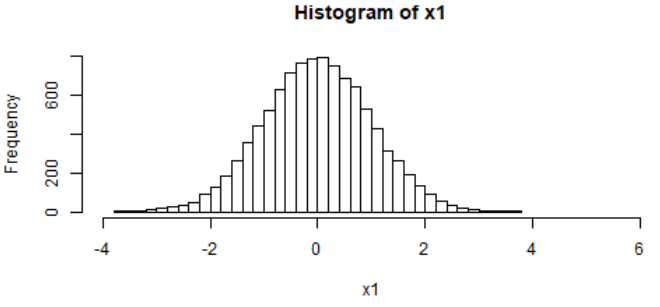
```
x<-seq(from=-5,to=5,length=1000)
dx1<-dnorm(x,0,1)
dx2<-dnorm(x,2,1)
plot(x,dx1,type="l")
lines(x,dx2,col=2)
```

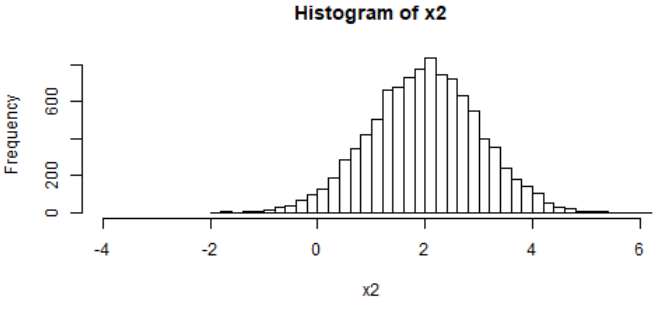


Location

- Histograms for two $random\ samples$ drawn from normal distribution with the same variance but different mean.
- Both histograms show that the data are symmetric around the sample mean but the histogram of x_2 is located to the right relative to the histogram of x_1.

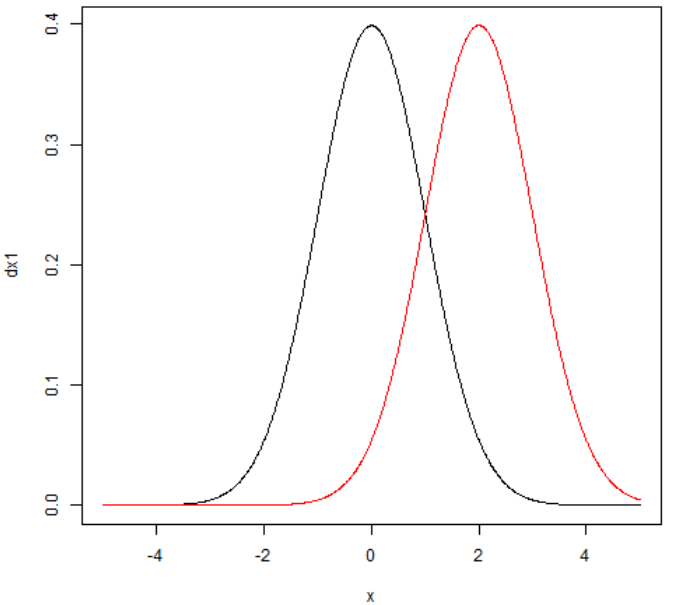
```
x1 <- rnorm(10000, 0, 1)
x2 <- rnorm(10000, 2, 1)
par(mfrow = c(2, 1))
hist(x1, col = 0, nclass = 50, xlim =
hist(x2, col = 0, nclass = 50, xlim =
```





Graphical displays for location

- How can we visualize the distribution ?
- How can we visualize a shift in location ?
- Numerical summaries for location (look at the book): mean. median, trimmed mean...



The singer dataset

- The singer datset (the R object `singer`) is a data frame giving the heights of singers in the New York Choral Society.
- The variables are named `height` (inches) and `voice.part` which is the voice group of the singer
    - Alto.
    - Sporano.
    - Tenor.
    - Bass.
- Each voice group is subdivied into two groups, high voice and low voice (for example Bass1 and Bass2 and the lower and higher Bass voices, respectivly).
- Main focus: the distribution of the singers' `height`.

The singer dataset

- The data:

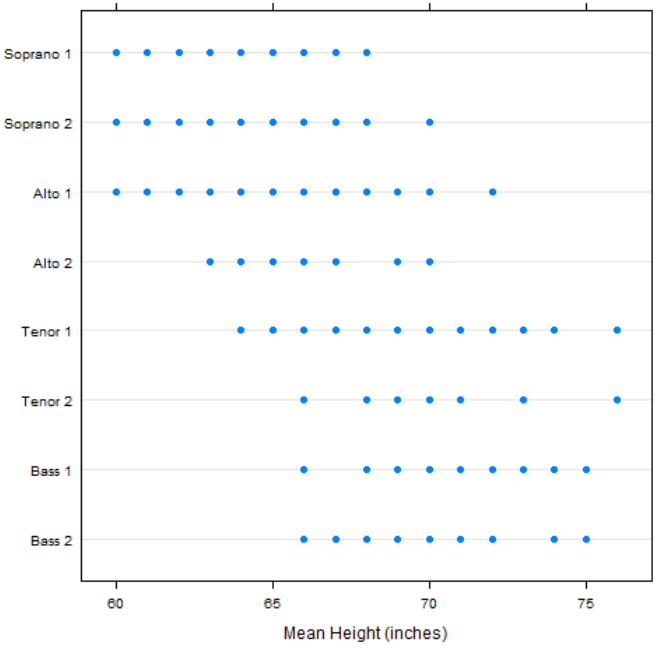
```
head(singer)
```

```
##   height voice.part
## 1     64  Soprano 1
## 2     62  Soprano 1
## 3     66  Soprano 1
## 4     65  Soprano 1
## 5     60  Soprano 1
## 6     61  Soprano 1
```

Stripplot (`lattice`)

- Stripplot: plots the data of each voice group in a different strip.
- `voice.part~height`.
- A clear main pattern in the data:
  - It is easy to distinguish between women (Sopranos and Altos) and men (Tenors and Basses).
  - Women are clearly shorter than men.

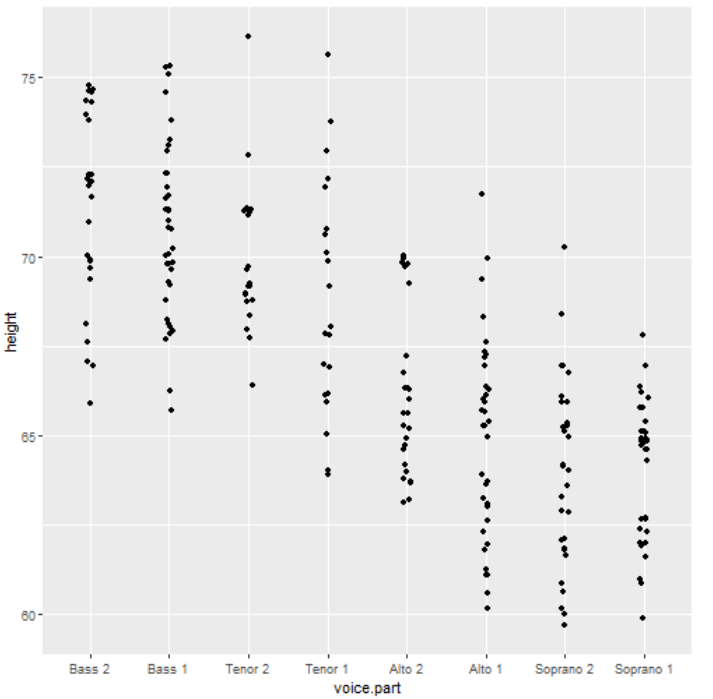
```
dotplot(singer$voice.part~singer$heigh
        aspect=1,
        xlab="Mean Height (inches)")
```



Stripplot (`ggplot2`): first layer

- An equivalent stripplot can be produced using the `ggplot()` package.
- The function `geom_jitter`: observatios with the same values will be ploted side by side and will not overlap so sample size would be seen as well in the plot.

```
ggplot(singer,
       aes(voice.part,height)) +
       geom_jitter(position = position
```



Mean by voice group

- In order to get a better insight of other patterns, we can summarize the distribution of each group with the sample mean.
- In R this can be done using the function `tapply()`.
- To calculate mean of height by the voice group in the singer dataset we use

```
attach(singer)
tapply(singer$height,singer$voice.part,mean)
```

```
##    Bass 2    Bass 1   Tenor 2   Tenor 1    Alto 2    Alto 1 Soprano 2 Soprano 1
##  71.38462  70.71795  69.90476  68.90476  66.03704  64.88571  63.96667  64.25000
```

Mean by voice group

- The group means point on the pattern that was already detected: on average men are taller than women.
- In addition, we can see that within each gender group, singers with lower voice are taller than singers with higher voice.
- For example, the average of the two bass groups (71.38 and 70.71 for Bass 1 and Bass 2 respectively) are higher than the average in the tenor groups (69.90 and 68.90 for Tenor 1 and Tenor 2 respectively).

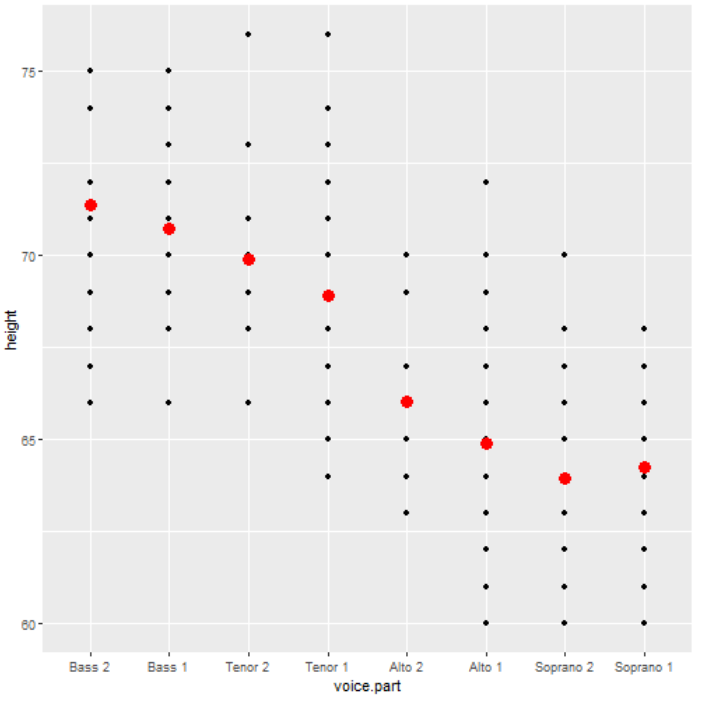
Mean by voice group

- Among women, the sopranos are shorter, on average, than the altos.
- Within each voice group (all except the sopranos), the singers with lower voices (the second voice group Bass 2, Tenor 2 and Alto 2) are taller than the singers with the higher voices (the first group Bass 1, Tenor 1 and Alto 1).
- For example the mean of the Bass 2 group (71.38) is higher than the mean of the Bass 1 group (70.71)

Stripplot (`ggplot2`): second layer

- The Figure shows the same information as in slide 34 with the addtion of the mean for each voice group.
- `stat_summary()`.
- `fun.y = "mean"`.

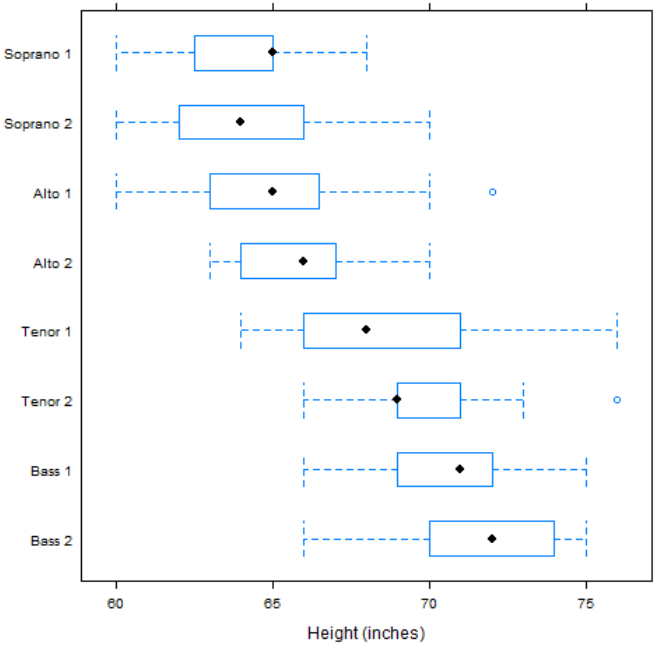
```
ggplot(singer, aes(voice.part,height))
geom_point() +
stat_summary(geom = "point", fun.y = "
```



Boxplot (`lattice`)

- Graphical display of the location of each distribution is the box plot.
- The location of each group is summarized by the median (the dot inside the box).
- Other aspects of this plot will be discussed in later chapters).
- Note that the function `bwplot()` is a part of the `lattice` package.

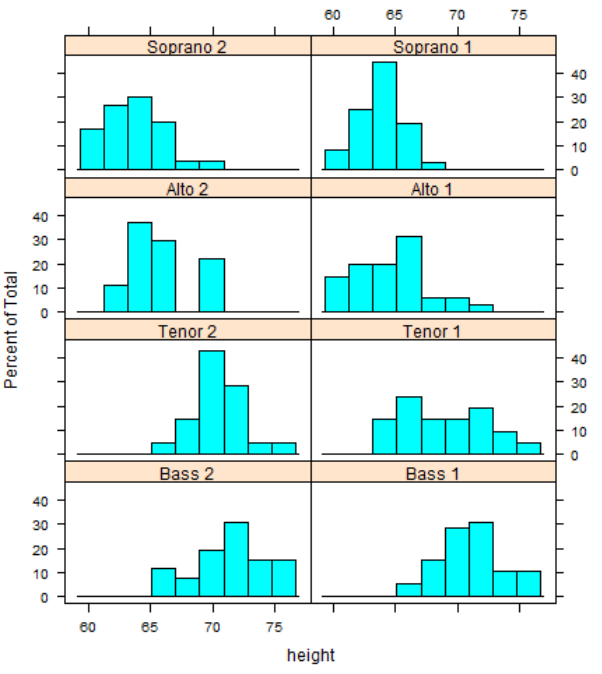
```
bwplot(as.factor(singer$voice.part)~ s
       data=singer,
       aspect=1,
       xlab="Height (inches)")
```



Multiway histogram (`lattice`)

- The multiway histogram presents the distributions of heights across the voice groups.
- Note how the distribution of height is shiffted from left to right across the voice levels.

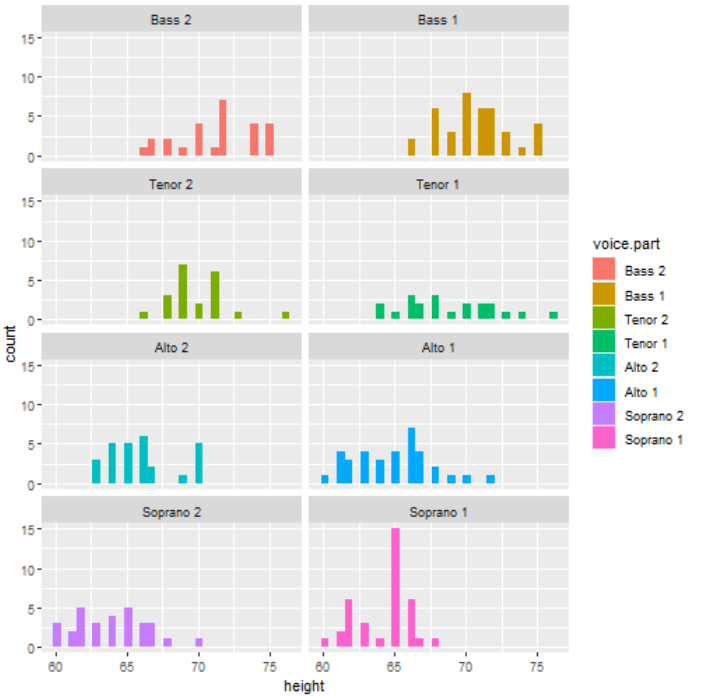
```
histogram(~ singer$height | singer$voi
        data=singer, layout = c(2, 4),
          aspect = 0.5,xlab = "height"
```



Multiway histogram (ggplot2)

- An equivalent multiway histogram can be produce with the ggplot2 package.
- Addtional two layers in the basic plot:
  - The first layer spacifys the plot type `histogram()`.
  - The second layer indicates the factor for the plot partions.
  - The function `facet_wrap(factor)`.

```
ggplot(singer, aes(height,fill = voice
geom_histogram() +
facet_wrap(~voice.part,ncol = 2)
```



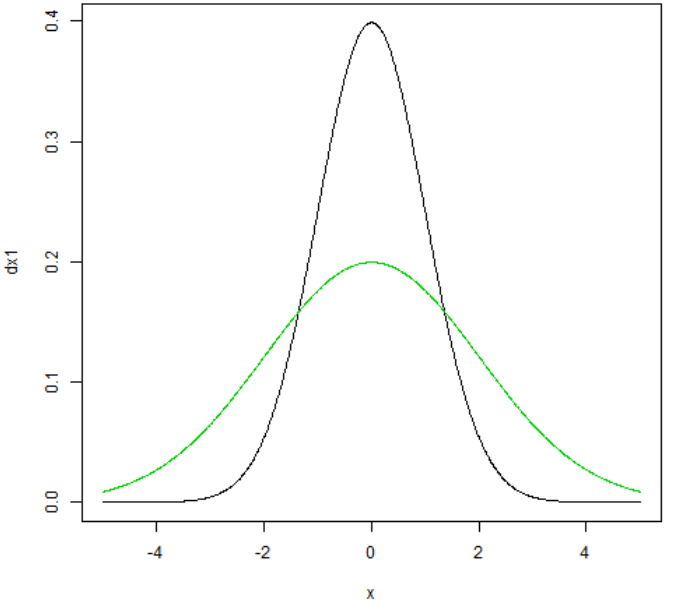
Part 4: Spread

Focus: Variability around the center of the distribution

Spread

- Two normal densities that have different variability (or spread).
- The density with the black line has variance 1 and density with the green line has variance 2.

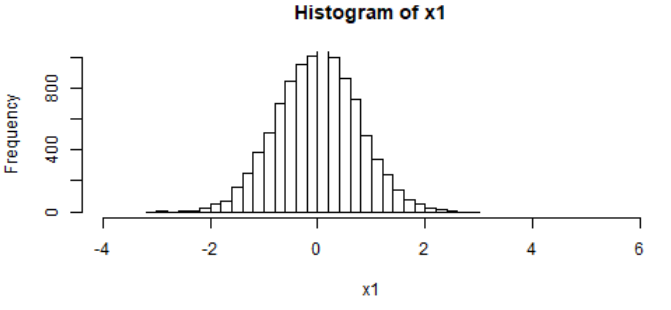
```
dx3<-dnorm(x,0,2)
plot(x,dx1,type="l")
lines(x,dx3,col=3)
```

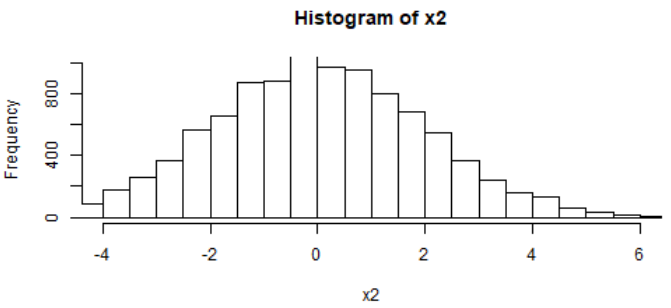


Spread

- Two samples were drawn from normal distribution with mean equal to 0 but with different variance.
- The two distributions have the same shape, both histograms are symmetric around 0 as expected.
- The spread in the histogram of x_2 is much higher than the spread in the histogram of x_1.

```
x1 <- rnorm(10000, 0, 0.75)
x2 <- rnorm(10000, 0, 2)
par(mfrow = c(2, 1))
hist(x1, col = 0, nclass = 25, xlim =
hist(x2, col = 0, nclass = 50, xlim =
```





Main concepts

- Up till now we summarized the distribution of the data with location estimators.
- In this chapter we focus on the spread.
- We want to measure how close the data are to each other and how concentrate the data around the center of the distribution.

- Numerical summaries for spread (see in the book):

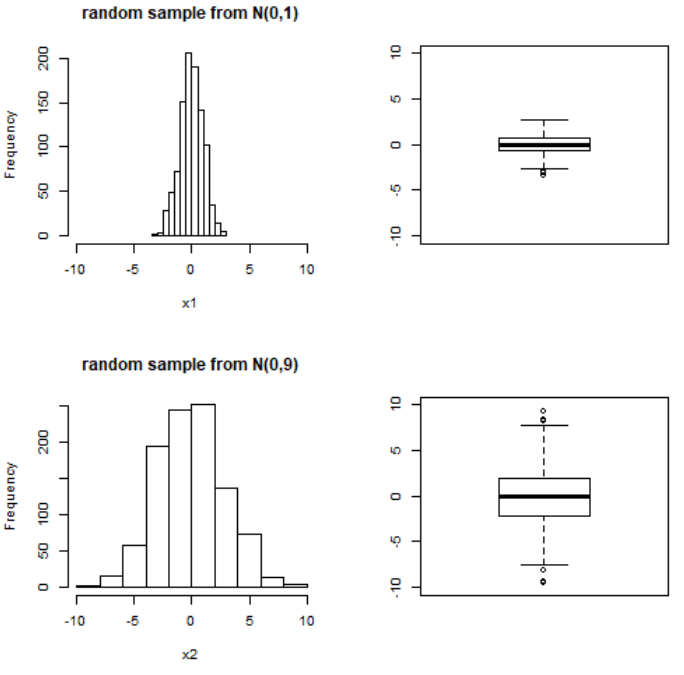
    - The sample variance.
    - The fourth-spared.
    - The MAD as measures

- Graphical displays:

    - boxplot.
    - violin plot.

Boxplot: A graphical display for spread and location

- Boxplot is a graphical display which shows the location, the spread and the shape of the distribution.
- The location is summarized by the median, the spread is summarized by the fourth-spread which is simply the length of the box in the boxplot.
- Inside the box: 50% of the data.



Boxplot: A graphical display for spread and location

- The upper and lower adjacent values in the boxplot are given by

$$\text{Upper adjacent value} = Min\{max(X), Q_3 + 1.5(Q_3 - Q_1)\}$$

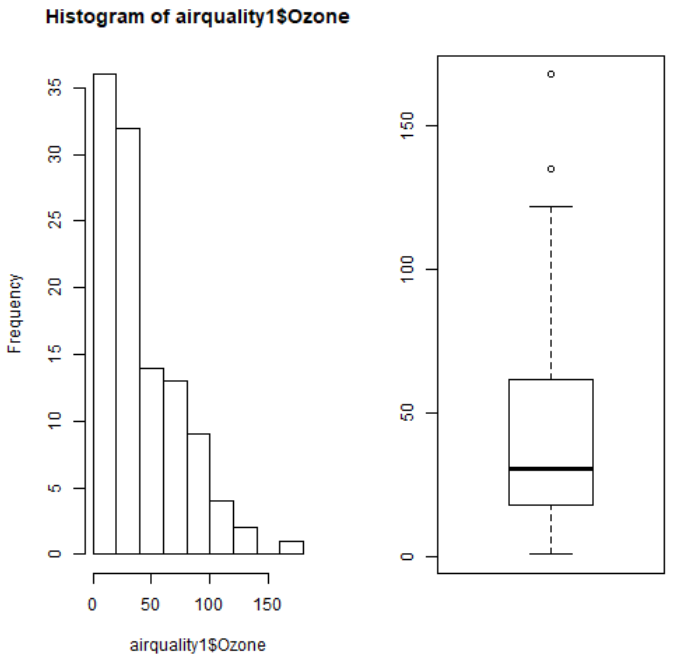
$$\text{Lower adjacent value} = max\{min(X), Q_1 - 1.5(Q_3 - Q_1)\}$$

- The upper and lower adjacent values are used to identify extrime values.
- Observations higher than the upper adjacent value or smaller than the lower adjacent value are considered to be outliers.

Example: the airqiality data

- Daily air quality measurements in New York, May to September 1973.
- The histogram and boxplot for the airquality data: Ozone level.
- A skewed distribution with few outliers at the upper tail (histogram).
- In the boxplot these outliers can be identified above the upper adjacent value.

```
par(mfrow=c(1,2))
airquality1<-na.omit(airquality)
hist(airquality1$Ozone)
boxplot(airquality1$Ozone)
```



YouTube tutorial: Boxplot in R

- For a short online YouTube tutorials:
  - by Data Science Tutorials, about boxplot using the `ggplot2` package see YTVD8.
  - by LawrenceStats, about boxplot using the `ggplot2` package see YTVD9

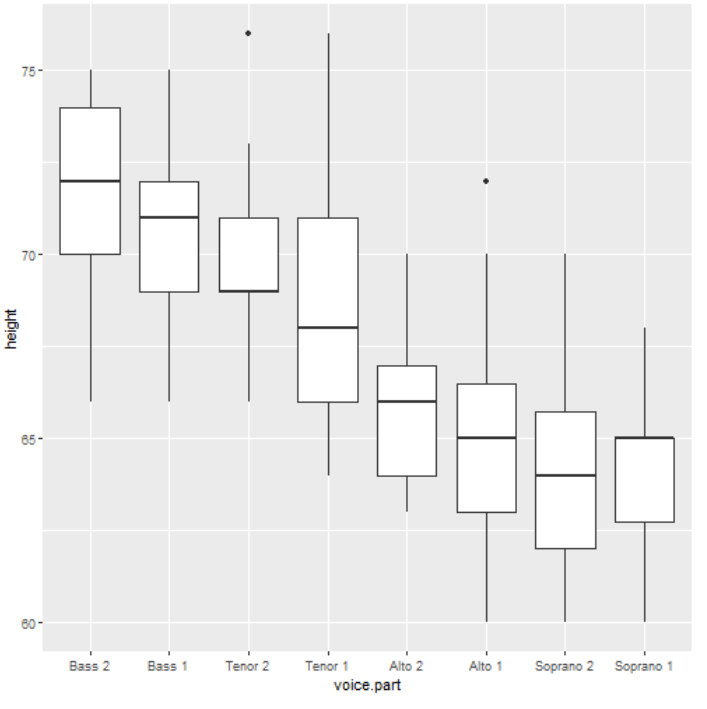
Web tutorial: Advanced boxplots in R

- Example for advanced boxplots in R using the `ggplot2` package and code to produce the plots can be found in the R Graph Gallery website here WAVD2.

Boxplot for the singers data

- A boxplot for the singers' height by voice group that was produced using the function `geom_boxplot()`.

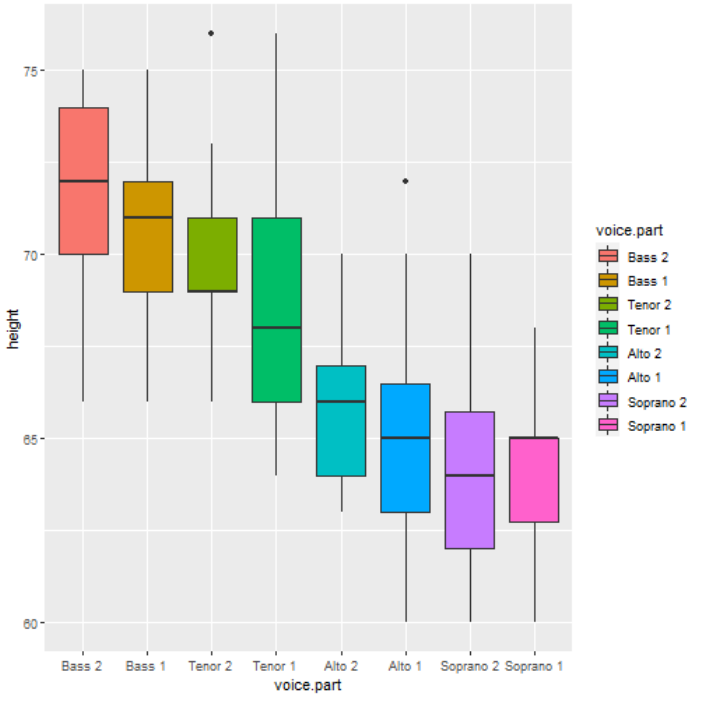
```
ggplot(singer, aes(voice.part,height))
```



Boxplot for the singers data

- The same boxplot in which colors (by group) are added to the boxplot using the argument `fill=voice.part`.
- Note that the object `voice.part` is a factor.

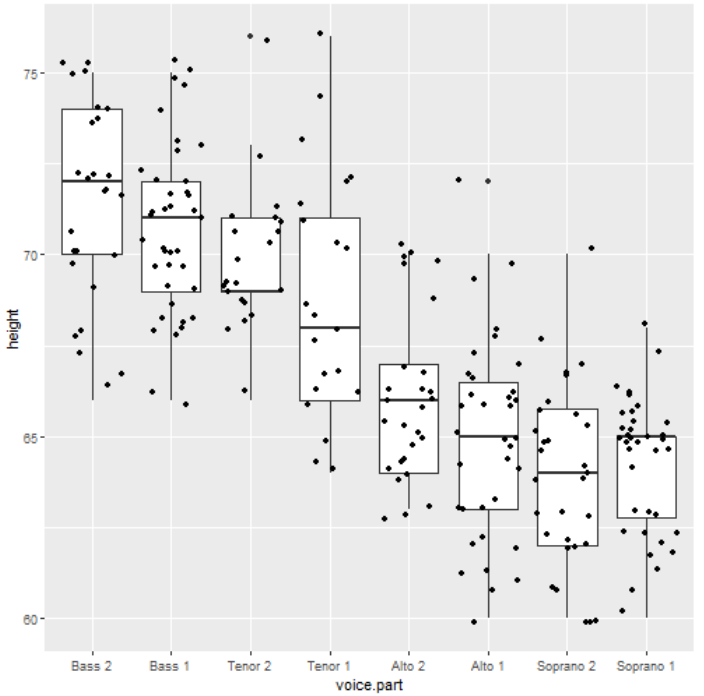
```
ggplot(singer, aes(voice.part,height,f
```



Boxplot for the singers data

- The data are added to the boxplot using the argument `geom = c("boxplot", "jitter")`.

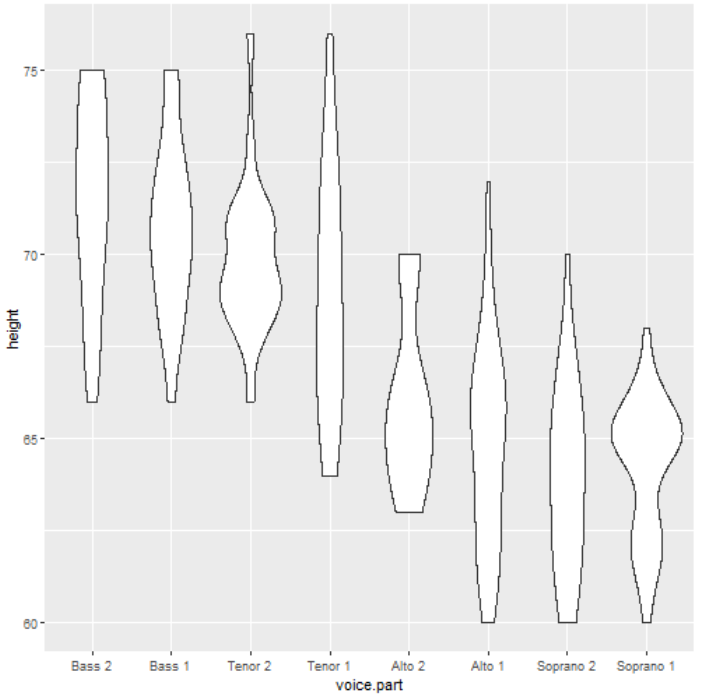
```
qplot(voice.part, height, data = singe
```



Violin plot for the singers data

- When the argument `geom_violin()` is used instead of `geom_boxplot()` the boxplot become a violin plot.

```
ggplot(singer, aes(voice.part,height))
```

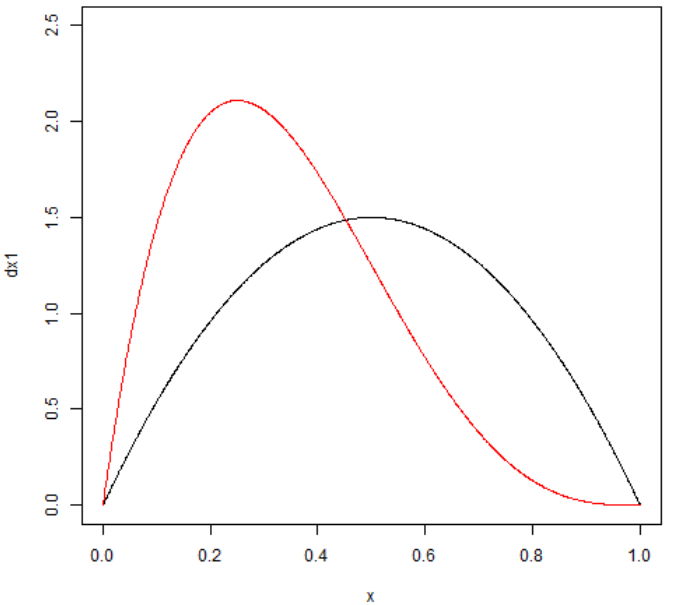


Part 5: Shape

Focus: How does the distribution look like ?

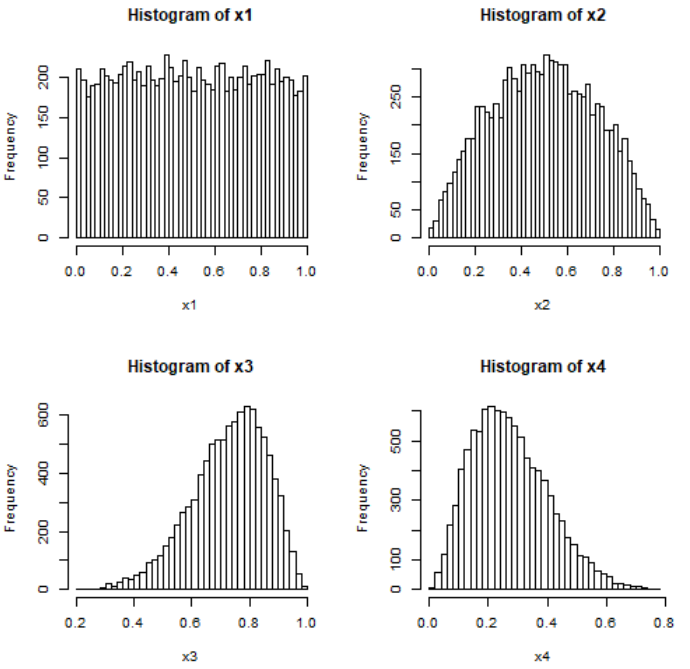
Shape

- Two beta densities having diffent shapes.
- Black line: $Beta(2, 2)$, red line: $Beta(2, 4)$.



Shape

- 4 samples (each with 10000 observations) that were drawn from different distributions.
- x_1 and x_2: samples were drawn from symmetric distributions.
- The distributions of x_3 is skewed to the left and the distribution of x_4 to the right.

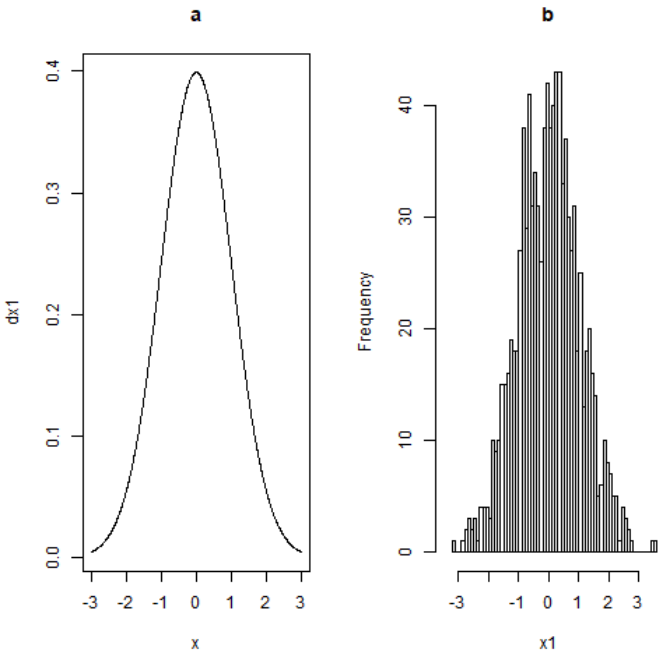


Density and density estimate

- So far we used histogram to visualize the shape of the distribution of the observations in the sample.
- In this chapter we discuss density estimates as a method to estimate and visualized the distribution in the population.

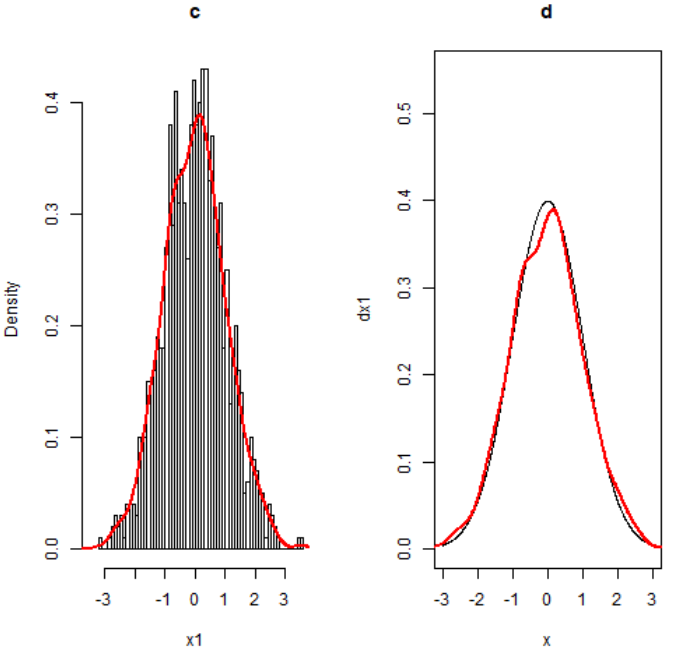
Density and density estimate

- A density function of $N(0, 1)$ that represents the distribution of a random variable in the population.
- Suppose that we draw a random sample of size n from the population.
- The histogram can be used to visualize the shape of the distribution.
- It is an estimate for the density in the population.



Density and density estimate

- A second approach to estimate the distribution of the population is to use a smooth version of the histogram , i.e., a density estimate.
- The density estimate for our example is shown (in red) panel c and d.



YouTube tutorial: Creating density plots and enhancing it with the `ggplot2` package

- A short online YouTube tutorial by LawrenceStats, about density plot using the `ggplot2` package see YTVD10.

Web tutorial: the ridgeline chart

- A Web tutorial about the ridgeline chart using the `ggplot2` and `ggridges` package is given in the the R Graph Gallery website WAVD4.

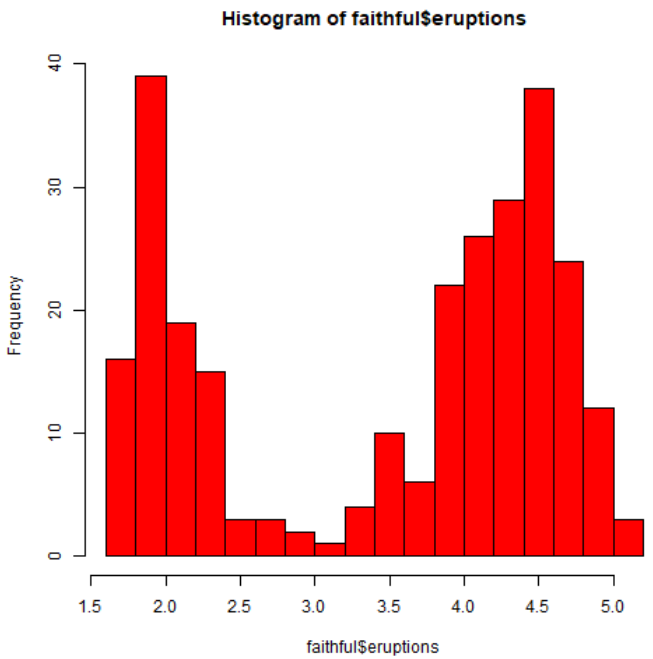
The old faithful data

- The data gives information about waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.
- Two variables:
    - `eruptions`: numeric, Eruption time in mins.
    - `waiting`: numeric, Waiting time to next eruption (in mins)

The old faithful data

- Histogram of eruptions time.
- A bi-model.
- `hist()`: basic graphical function in R.

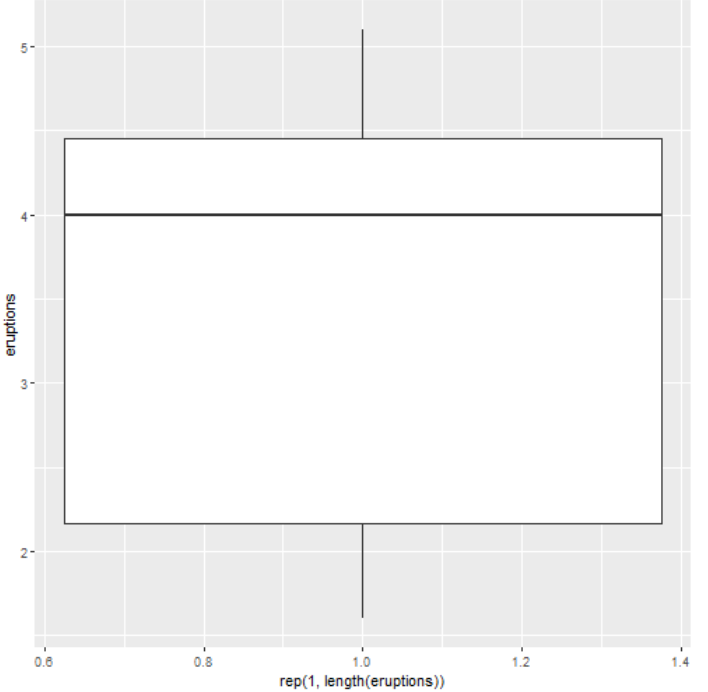
```
hist(faithful$eruptions,nclass=20,col=
```



The old faithful data

- A failure of the boxplot to capture this feature (the bi-model) of the data .

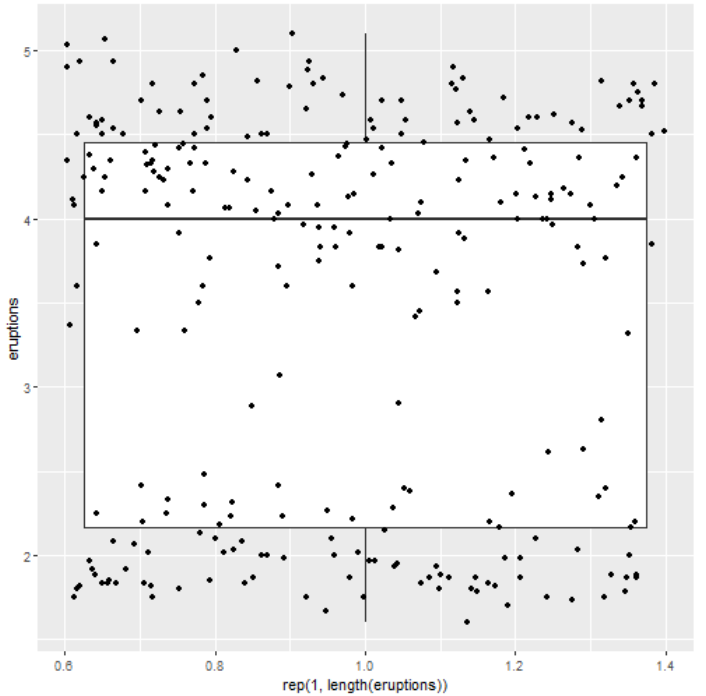
```
qplot(rep(1,length(eruptions)),eruptio
```



The old faithful data

- We add the data to the boxplot, using the option `geom = c("boxplot", "jitter")`.
- We identify the two parts of the eruptions time distribution.

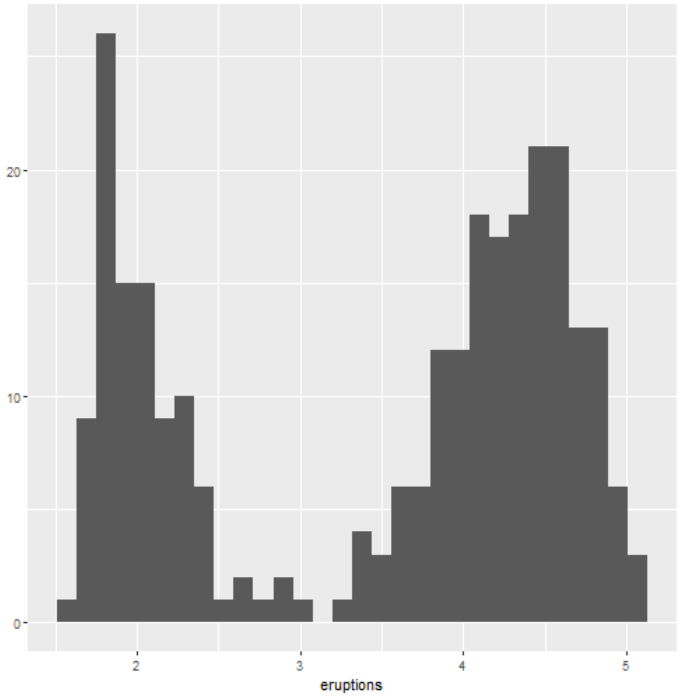
```
qplot(rep(1,length(eruptions)),eruptio
```



The old faithful data

- The histogram is able to capture the shape of the distribution.

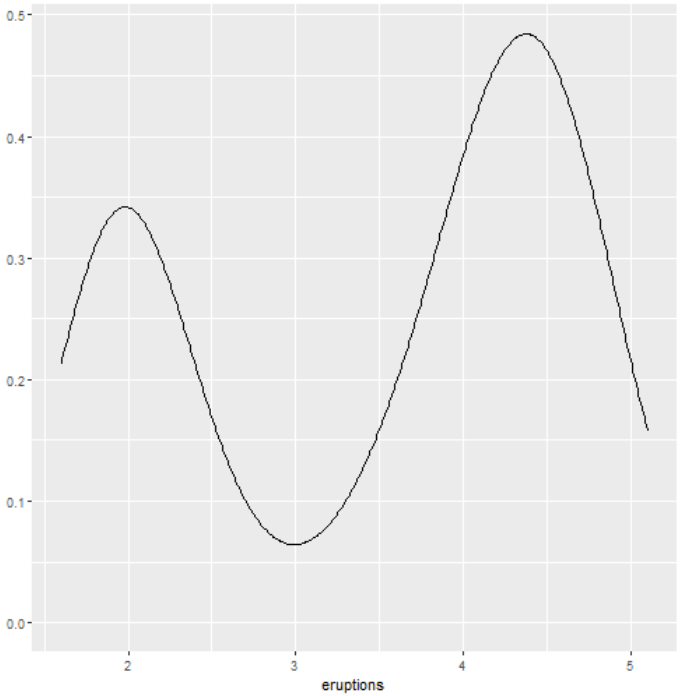
```
qplot(eruptions, data=faithful, geom="
```



The old faithful data

- A density estimate was produced using the opion `geom="density"`) provides a smooth estimate of the distribuon.

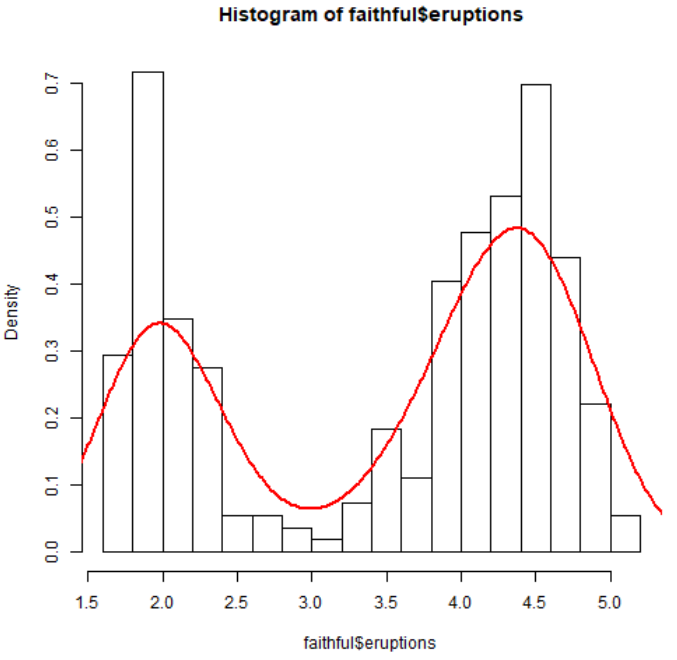
```
qplot(eruptions, data=faithful, geom="
```



The old faithful data

- We can plot both histogram and density in the same plot.

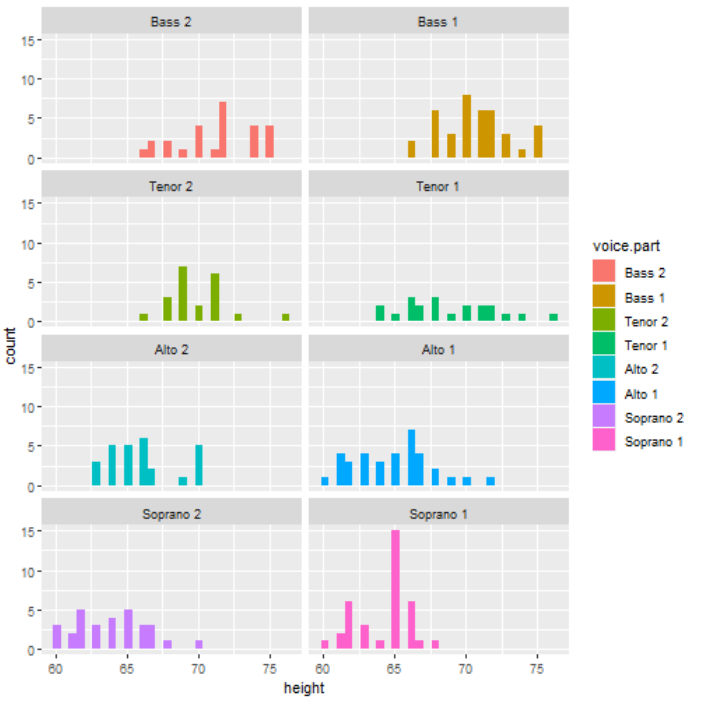
```
hist(faithful$eruptions,nclass=15,prob
dx<-density(faithful$eruptions)
lines(dx$x,dx$y,lwd=2,col=2)
```



The singer data

- We use density plots to visualize the shit of the distribution of the singers' height across the voice part groups.
- The histograms ravel the shifts within and between the voice groups.

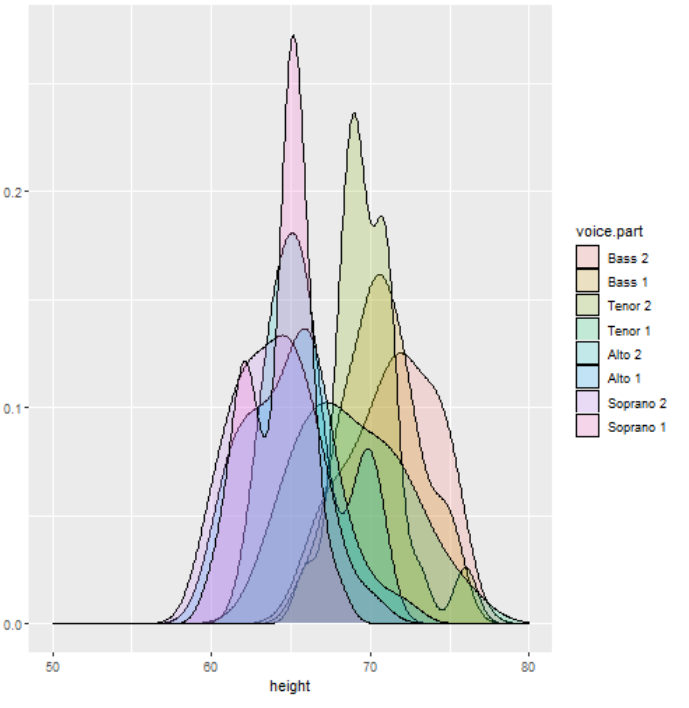
```
ggplot(singer, aes(height,fill = voice
geom_histogram() +
facet_wrap(~voice.part,ncol = 2)
```



The singer data

- In the density plots:
  - The difference between the sopranos and altos (women singers, the densities in the left) and the tenors and basses (men singers, densities in the right) is clearly seen.
  - The difference within each group is more difficult to detect.

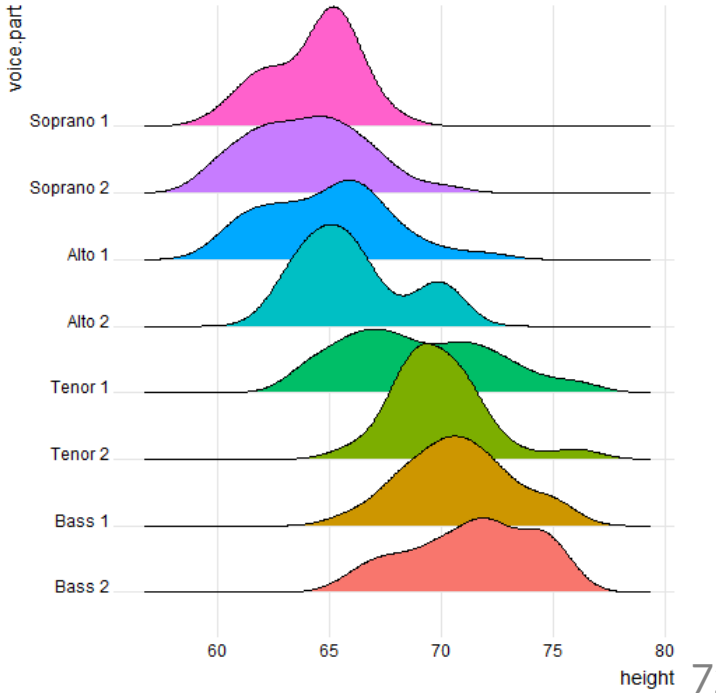
```
qplot(height, data=singer, geom="densi
fill = voice.part, alpha = I(0.2))
```



The singer data

- A ridgeline charts visualizes the difference between the groups and within each voice group.
- Note that the R package `ggridges` should be instaled to produce the plot.

```
library(ggridges)
ggplot(singer, aes(x=height,y=voice.pa
  geom_density_ridges() +
  theme_ridges() +
  theme(legend.position = "none")
```



Part 6: Shape (normal probability plot)

Focus: Normal distribution?

Normal probability plot

- A normal probability plot is a plot in which the quantile of the samples are plotted versus the corresponding quantiles of a standard normal distribution $N(0, 1)$.
- In this chapter we discuss the normal probability plot as a graphical tools to vizualise the shape of a distribution.

- Using histograms and boxplots we are able to investigate the shape of the distribution focusing on the follwing issues:

    - How nearly symmetric the distribution of the data is.
    - Whether the distribution of the data is single-peaked, or whether it is multi-peaked.
    - Whether it is skewed.
    - How far we from a Normal distribution ?

Quantile of $N(\mu, \sigma^2)$: Definition and a simple example

- A qq normal plot is a graphical disply to investigate how now nearly is the sample to a normal distribution. Let $q_{\mu,\sigma}(f)$ be a quantile of $N(\mu, \sigma^2)$, it can be expressed as

$$q_{\mu,\sigma}(f) = \mu + \sigma q_{0,1}(f).$$

- For example, the $2.5 \% $ quantile of the standard normal distribution is -1.96.

```
qnorm(0.025,0,1)
```

```
## [1] -1.959964
```

- For $N(2, 5^2)$ we have

$$q_{2,5}(2.5\%) = 2 + 5 \times -1.96 = -7.8$$

```
qnorm(0.025,2,5)
```

```
## [1] -7.79982
```

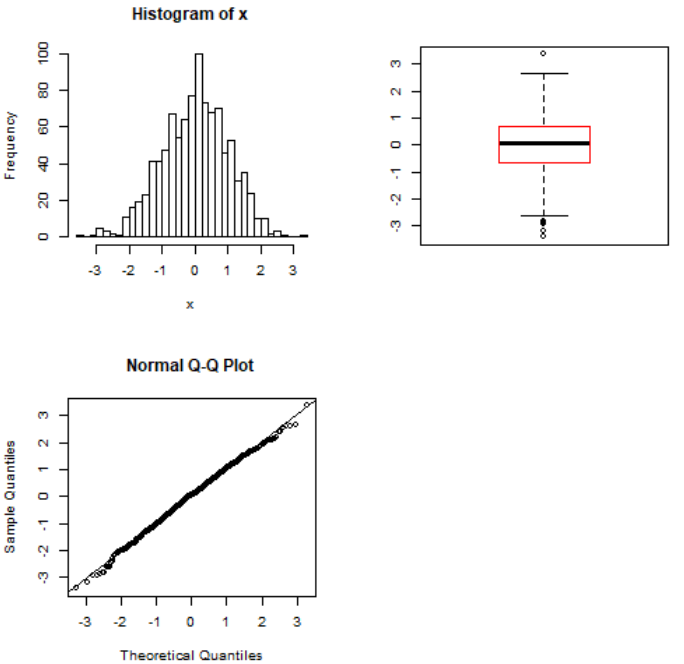
YouTube tutorial: QQ-plots in RStudio

For a short online YouTube tutorials by UTSSC, about normal probability plot using R studio see YTVD12.

A sample from N(0,1)

- We use the R function `qqnorm()` to produce the normal probabilty plot.
- Data were sampled from $N(0, 1)$.
- we expect that all points in the normal probability plot will lay on the on the 45^o line.

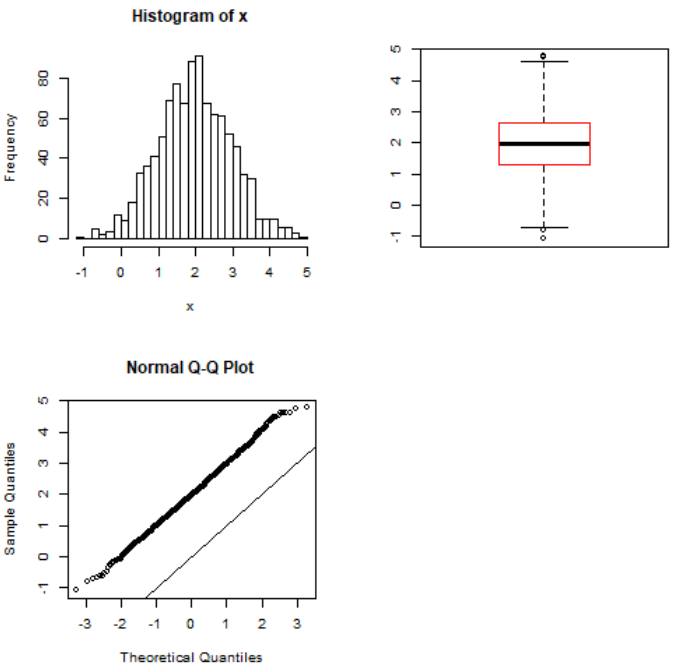
```
x <- rnorm(1000, 0, 1)
par(mfrow = c(2, 2))
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
qqnorm(x)
abline(0, 1)
```



A Sample from N(2,1)

- A sample from $N(2,1)$, i.e., it represents a shift model with the same variabiliy comapre to $N(0,1)$.
- In this case we expect that all points in the normal probability plot will lay above and parallel to the 45^o lines.

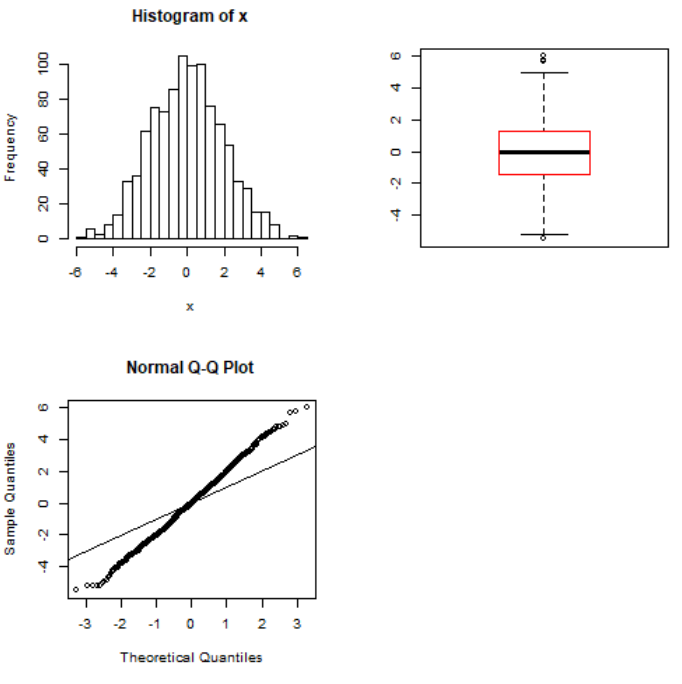
```
par(mfrow = c(2, 2))
x <- rnorm(1000, 2, 1)
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
qqnorm(x)
abline(0, 1)
```



A Sample from N(0,2)

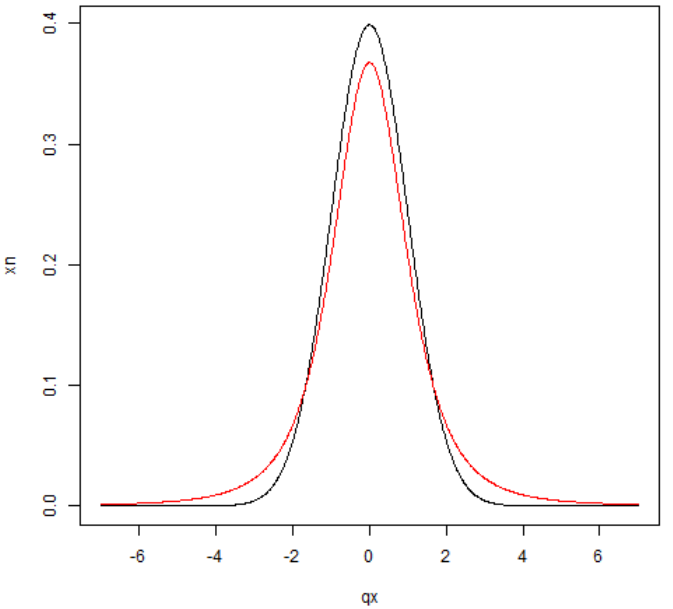
- The sample was drawn from $N(0, 2^2)$ which implies that the mean is the same as $N(0, 1)$ but the variability is higher.
- We expect the points in the normal probability plot to form a straight line with higher slope than 1.

```
par(mfrow = c(2, 2))
x <- rnorm(1000, 0, 2)
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
qqnorm(x)
abline(0, 1)
```



A Sample from $t_{(3)}$

- A $t_{(3)}$ distribution (red line) has the same mean as $N(0, 1)$ but longer tails.
- Note that the two distribution and centered around zero.

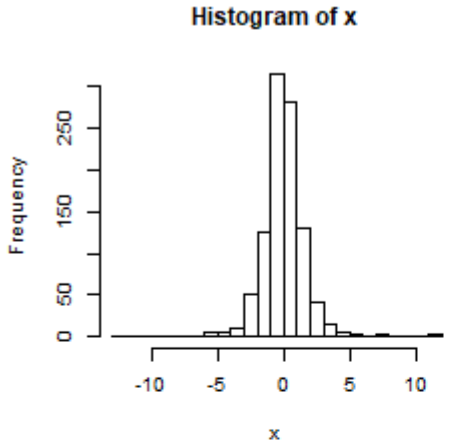


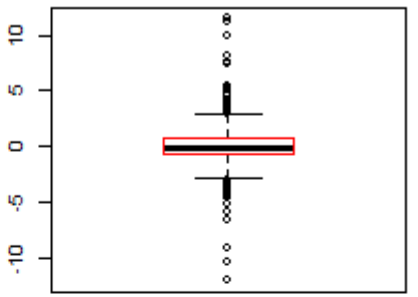
A Sample from $t_{(3)}$

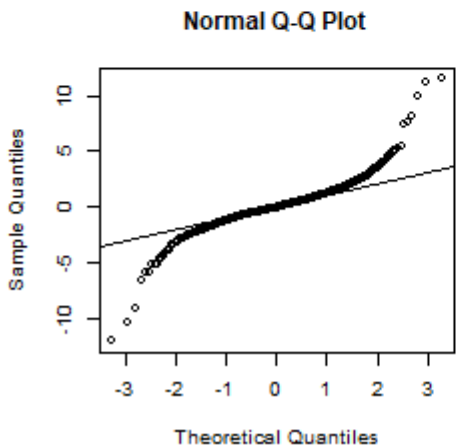
- For the normal probability plot we expect that the points will lay on the 45^o line in the center but with more extreme values .

```
par(mfrow = c(2, 2))
x <- rt(1000, 3)
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
qqnorm(x)
abline(0, 1)
```

A Sample from $t_{(3)}$





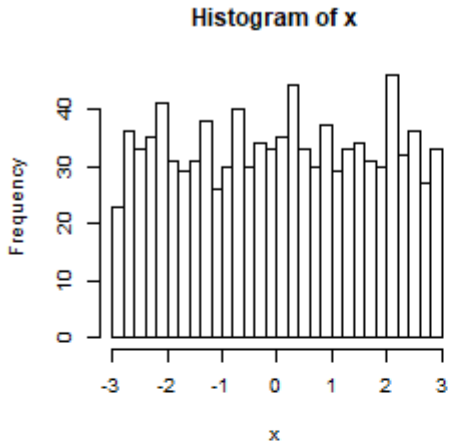


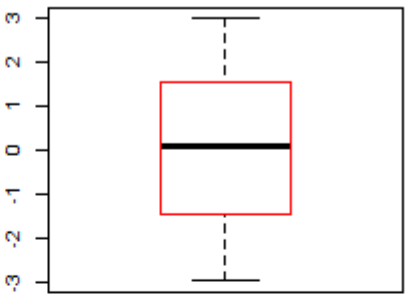
A Sample from $U(-3, 3)$

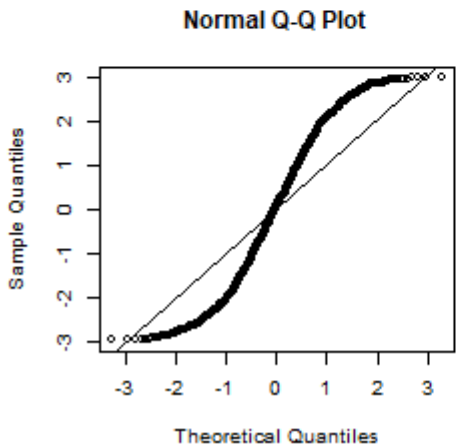
- The data of this example are uniformly distributed across the minimum and maximum values.
- We expect the points in the normal probability plot to cross the 45^o lines and to lay relatively far from the line.

```
x <- runif(1000, -3, 3)
par(mfrow = c(2, 2))
hist(x, nclass = 25, col = 0)
boxplot(x, boxcol = 2, medcol = 1)
qqnorm(x)
abline(0, 1)
```

A Sample from $U(-3, 3)$



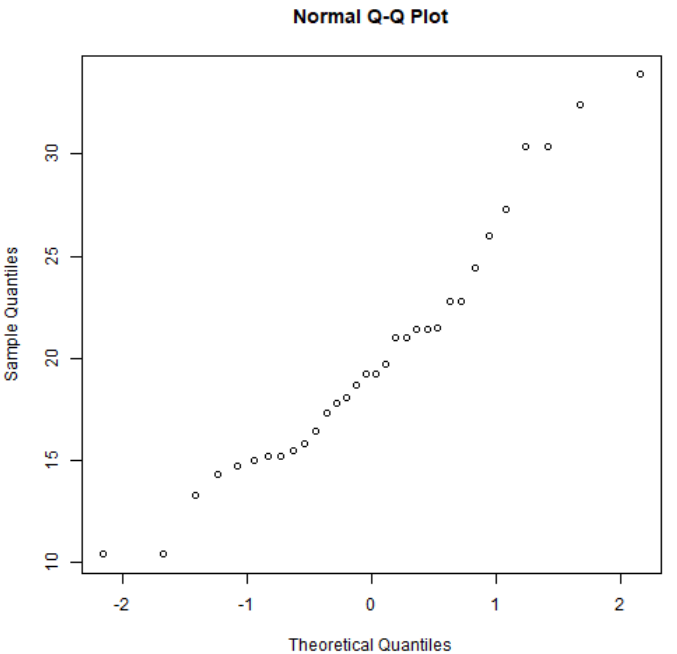




The cars data

- Normal probability plot for for mpg.
- $N(0, 1)$?
- $N(\mu, \sigma^2)$?

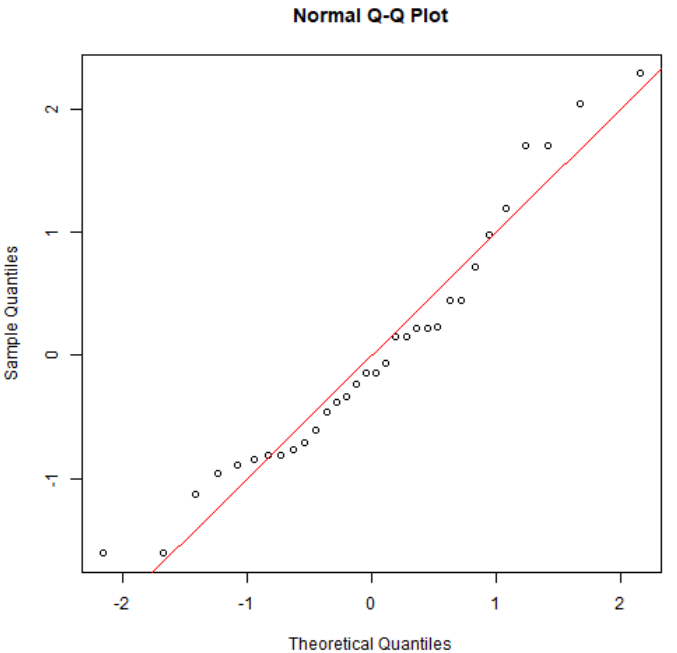
```
qqnorm(mtcars$mpg)
```



The cars data

- Normal probability plot for for the z-score of mpg.
- $N(0,1)$?

```
m.mpg<-mean(mtcars$mpg)
sd.mpg<-sqrt(var(mtcars$mpg))
z<-(mtcars$mpg-m.mpg)/sd.mpg
qqnorm(z)
abline(0,1,col=2)
```

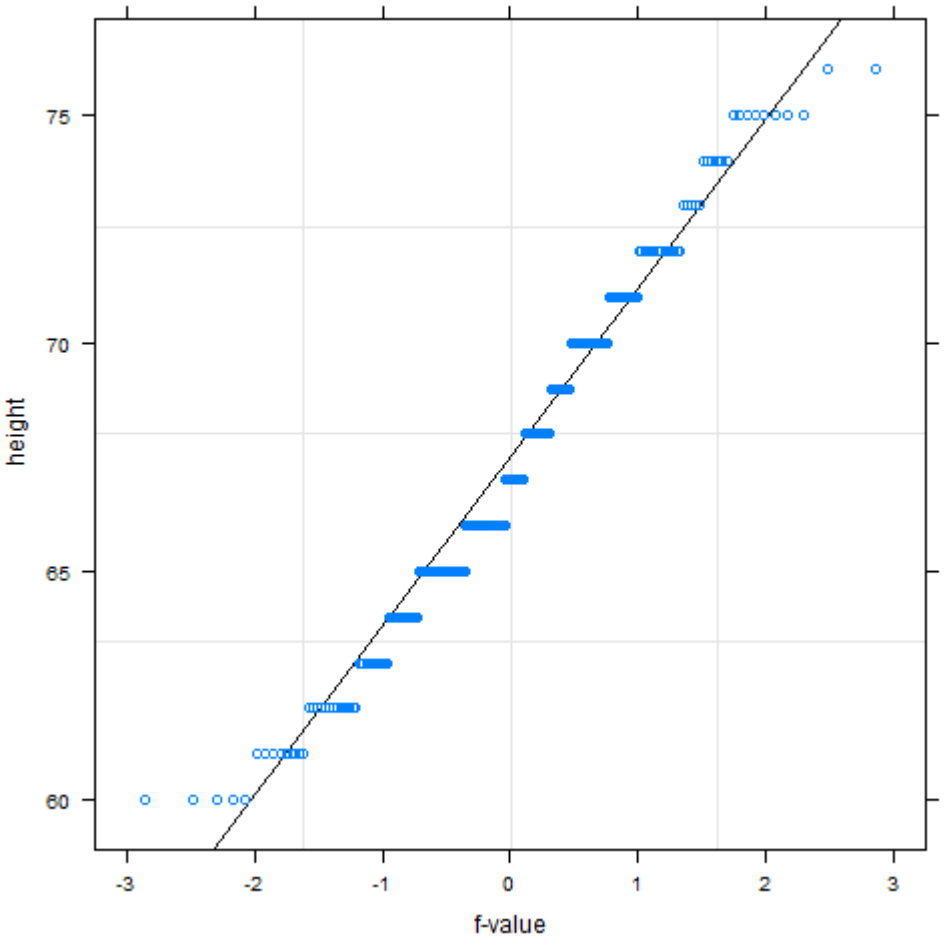


The signer dataset

- Normal probability plot for the height.

```
qqmath(~ height,
        distribution = qnorm,
        data=singer,
           layout=c(1,1),
             prepanel = prepanel.qqmat
            panel = function(x, ...) {
            panel.grid()
            panel.qqmathline(x, ...)
            panel.qqmath(x, ...)
            },
        aspect=1,
        xlab = "f-value",
        ylab="height")
```

The signer dataset



The signer dataset

- Normal probability plot for the height (by voice group).

```
qqmath(~ height | voice.part,
       distribution = qnorm,
       data=singer,
          layout=c(4,2),
            prepanel = prepanel.qqmat
           panel = function(x, ...) {
           panel.grid()
           panel.qqmathline(x, ...)
           panel.qqmath(x, ...)
           },
       aspect=1,
       xlab = "f-value",
       ylab="height")
```

The signer dataset