UNIVERSITÄT
BERN

R COURSE

LAURENT EXCOFFIER

Day 5 Data analysis and descriptive statistics

Descriptive statistics

Descriptive (or summary) statistics are the first figures used to represent nearly every dataset. They also form the foundation for much more complicated computations and analyses. Thus, in spite of being composed of simple methods, they are essential to the analysis process.

Most important functions in R:

```
• mean(x)        # (sample) mean of the vector x
• median(x)      # median of x
• variance(x)    # variance of x
• sd(x)          # standard deviation of x
• min(x)         # minimum of x
• max(x)         # maximum of x
• cov(x,y)       # covariance of x and y
```

Descriptive statistics

Similar functions exist for matrices or data frames.

E.g., ***colMeans*** calculates the mean for each column:

```
data(iris)
colMeans(iris)
Error in colMeans(iris) : 'x' must be numeric
```

We get error because not all columns of ***iris*** are numeric.

```
str(iris)
```

Shows that 5th column is of type factor.

We therefore have to remove the 5th column:

```
colMeans(iris[,-5])     # calculates the mean for each column
```

Apply function to columns/rows

We can also use the function ***apply*** to apply a function to each column/row of a matrix.

```
apply(X, MARGIN, FUN)
```

where

X is a matrix (or an array)

MARGIN a vector giving the subscripts which the function will be applied over. For a matrix 1 indicates rows, 2 indicates columns, c(1,2) indicates rows and columns.

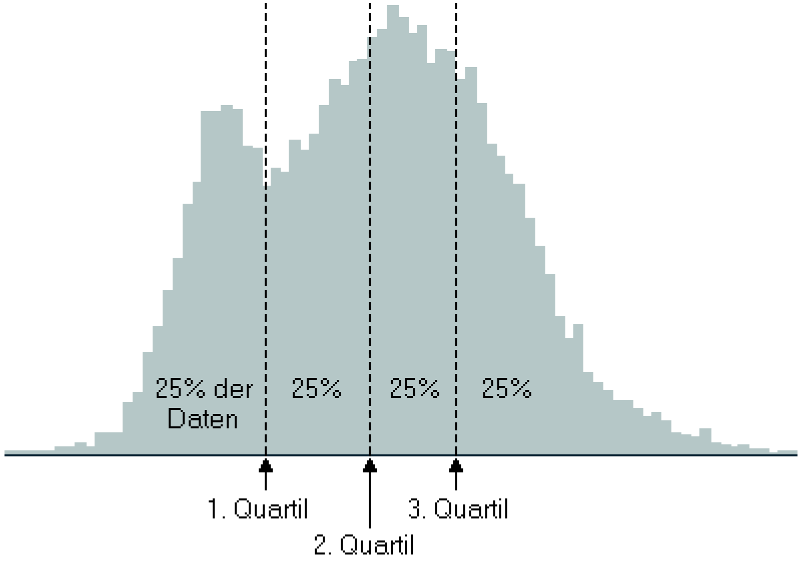
FUN the function being applied

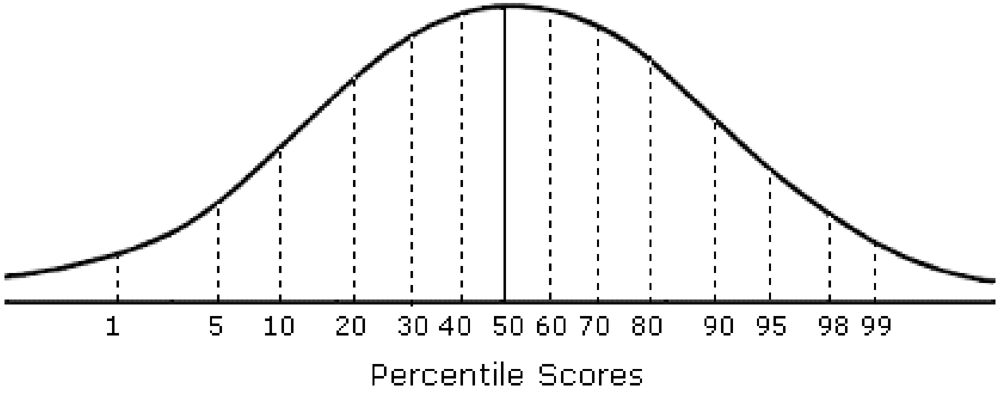
```
mat<-as.matrix(iris[,-5])
apply(mat, 2, mean)    # equivalent to colMeans(mat)
```

Quantiles/Percentiles

Quantiles divide ordered data into *k* essentially equal-sized subsets: the *k*-quantiles are the data values marking the boundaries between these subsets.

The *p*-percentile is defined as the data point below which p % of the distribution lie.



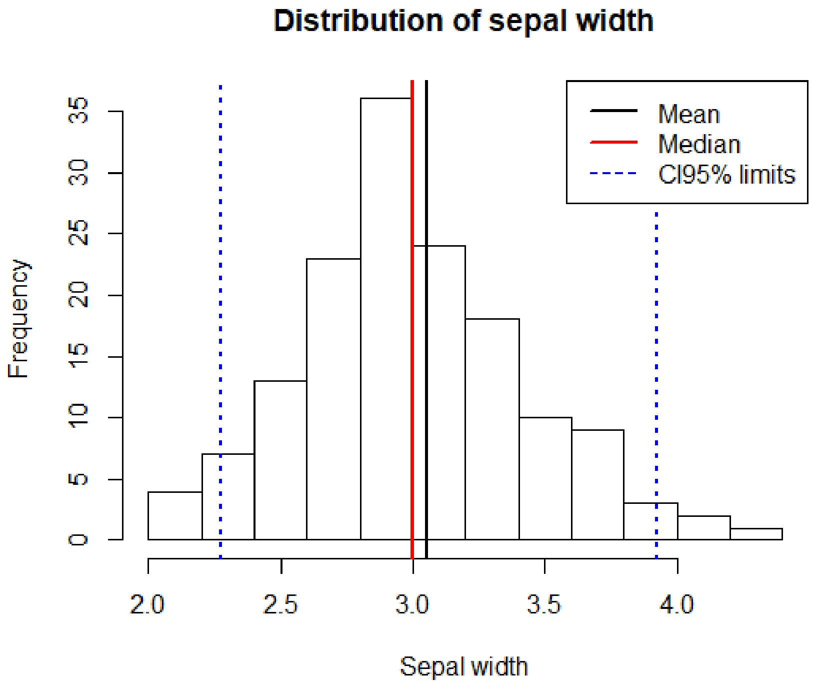


```
quantile(x,probs = c(p1,p2,p3,p4))  # returns the p1, … , p4 percentiles
```

Quantiles/Percentiles

Example:

```
hist(iris$Sepal.Width)
Sep.quant = quantile(iris$Sepal.Width,prob=c(0.025,0.975))
Sep.quant
   5%   95%
2.345 3.800
abline(v = Sep.quant, col=”blue”, lty=3)
```



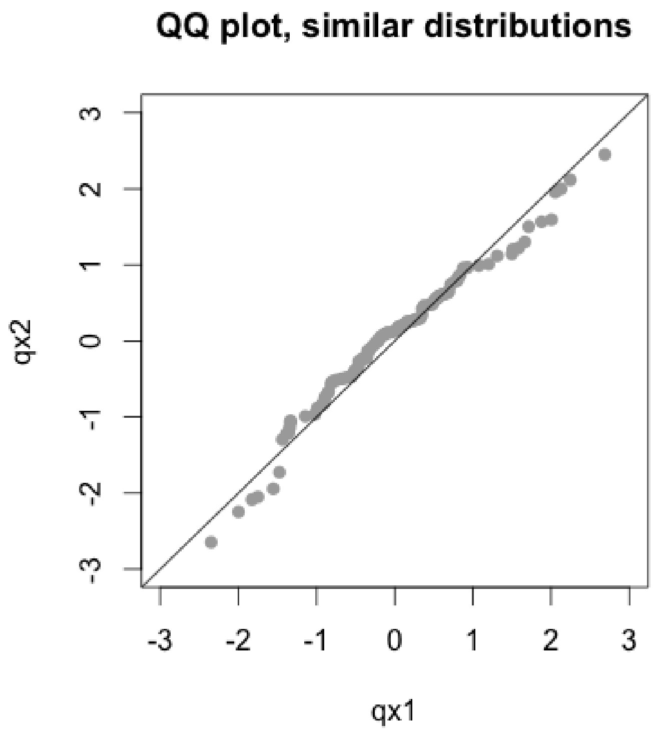
Summary statistics

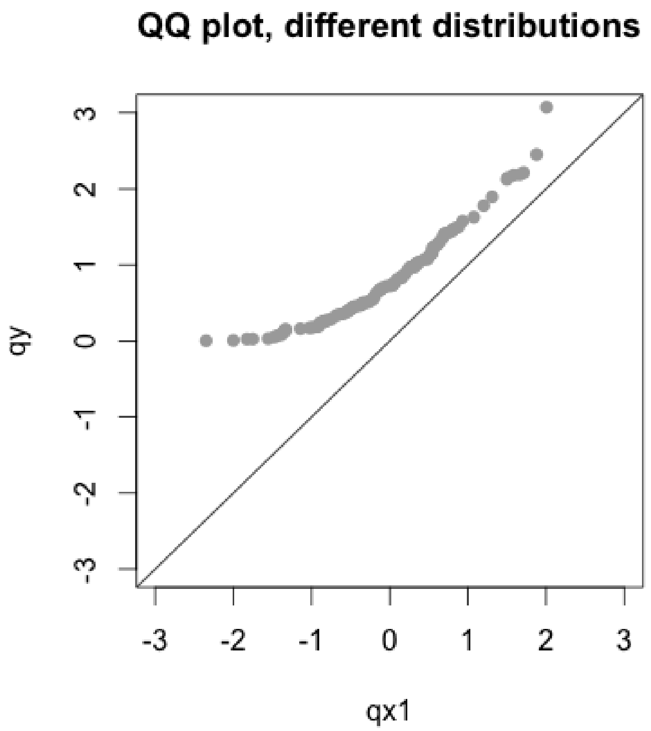
We can get a set of summary statistics for each column of a data frame with the function ***summary***.

```
summary(iris)

 Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

QQ Plots

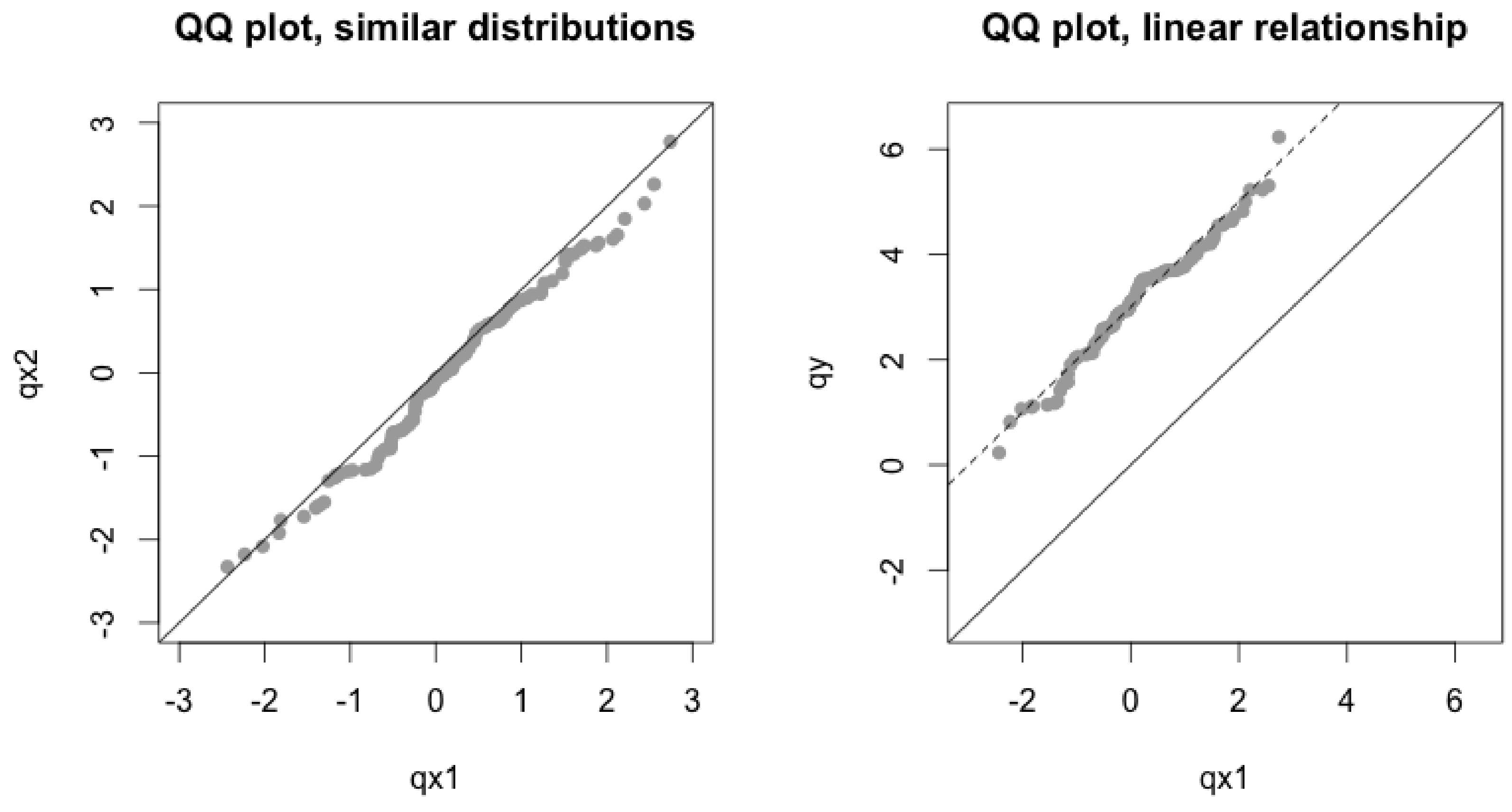
- A QQ plot is a plot of the quantiles of two distributions against each other.
- The pattern of points in the plot is used to compare the two distributions.
- If the two distributions being compared are similar, the points in the QQ plot will approximately lie on the line y = x.





QQ-Plots

If the distributions are linearly related, the points in the Q–Q plot will approximately lie on a line, but not necessarily on the line y = x.



QQ Plots in R

In ***R***, QQ-plots can be done “by hand”:

```
x1 <- rnorm(100,0,1)
x2 <- rnorm(100,0,1)

qx1 <- quantile(x1,prob=seq(0,1,by=0.01))
qx2 <- quantile(x2,prob=seq(0,1,by=0.01))

plot(qx1,qx2,main="QQ plot",pch = 16,col="DARKGRAY")
abline(0,1)  #Since quantiles are from the normal distribution
```

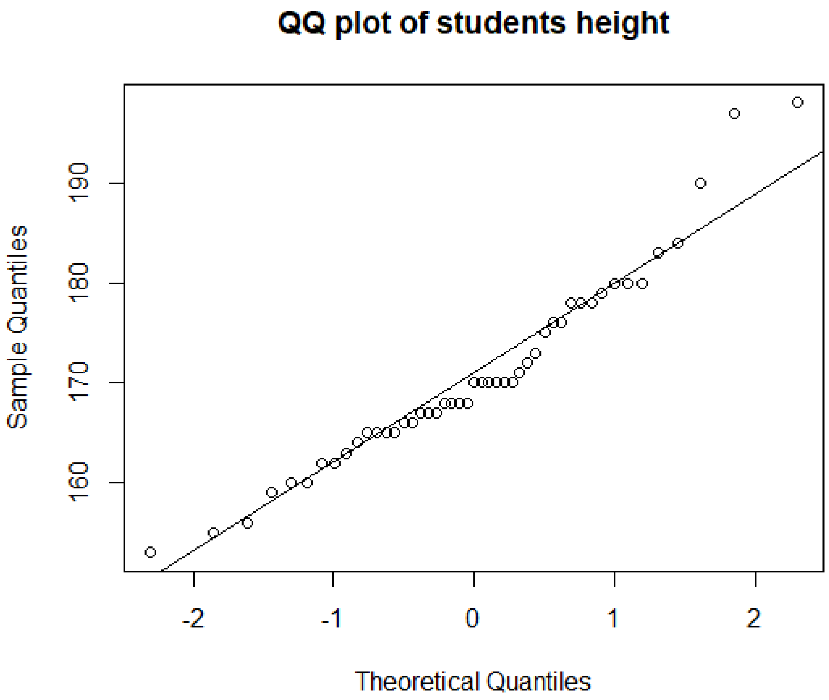
… or automatically with the commands:

```
qqplot(x1,x2)    # qqplot of variables x and y

qqnorm(x1)      # qqplot of standard normal distr. and variable x
qqline(x1)      # adds the diagonal line to a qqnorm-plot
```

QQ-Plots Example

We want to see if the student’s weights are normally distributed

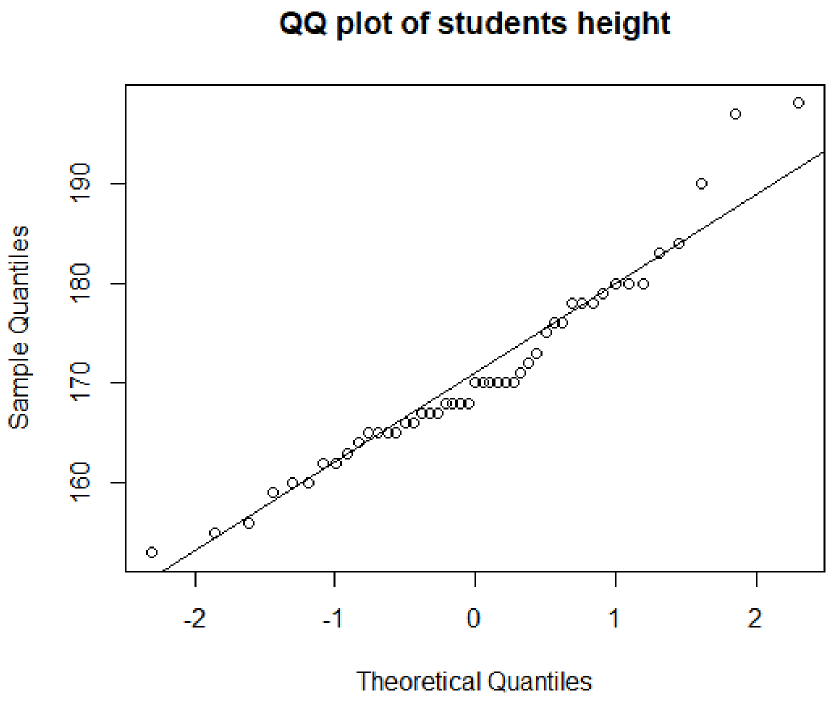
```
# read the student heights
ds2016 = read.table("StudentData2016.txt", na.strings = "?", header=T)
heights=ds2016$Height
qqnorm(heights, main="QQ plot of students height")
qqline(heights)
```

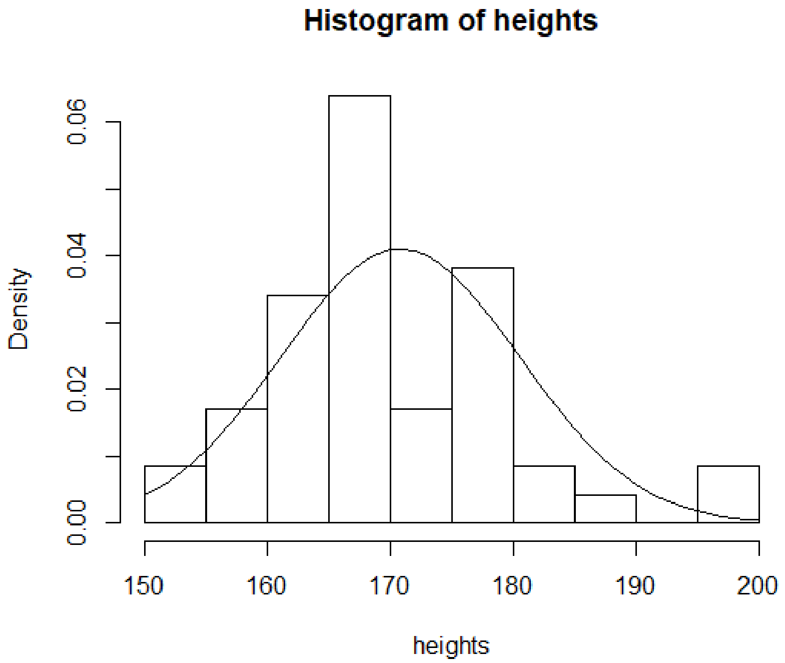


QQ-Plots Example

We want to see if the student’s weights are normally distributed

```
hist(heights, freq = F)
xseq=seq(150, 200, 1)
lines(xseq, dnorm(xseq, mean=mean(heights), sd=sd(heights)))
```

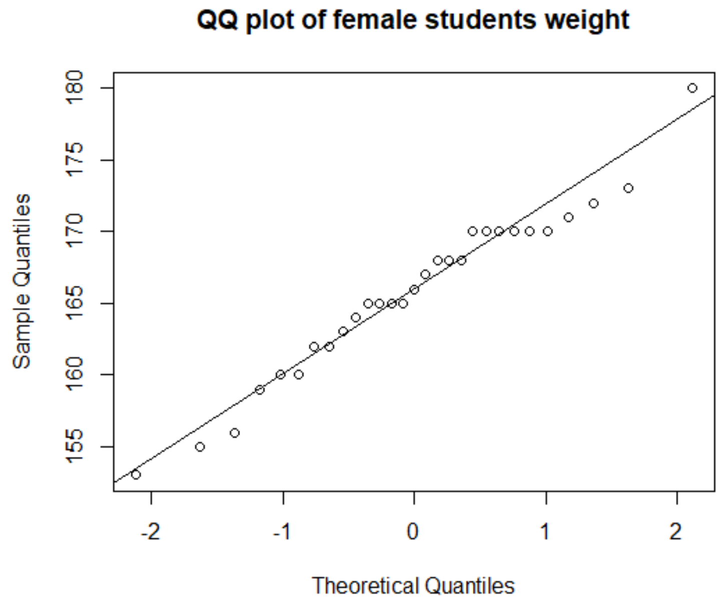


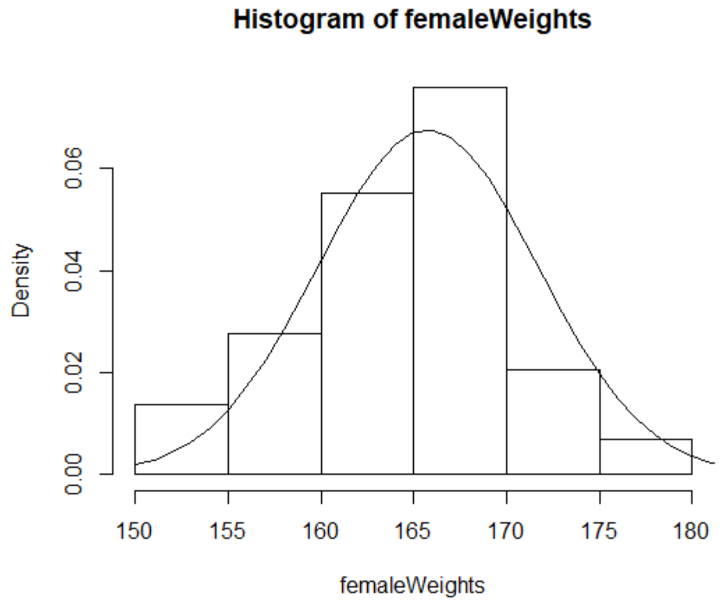


QQ-Plots Example

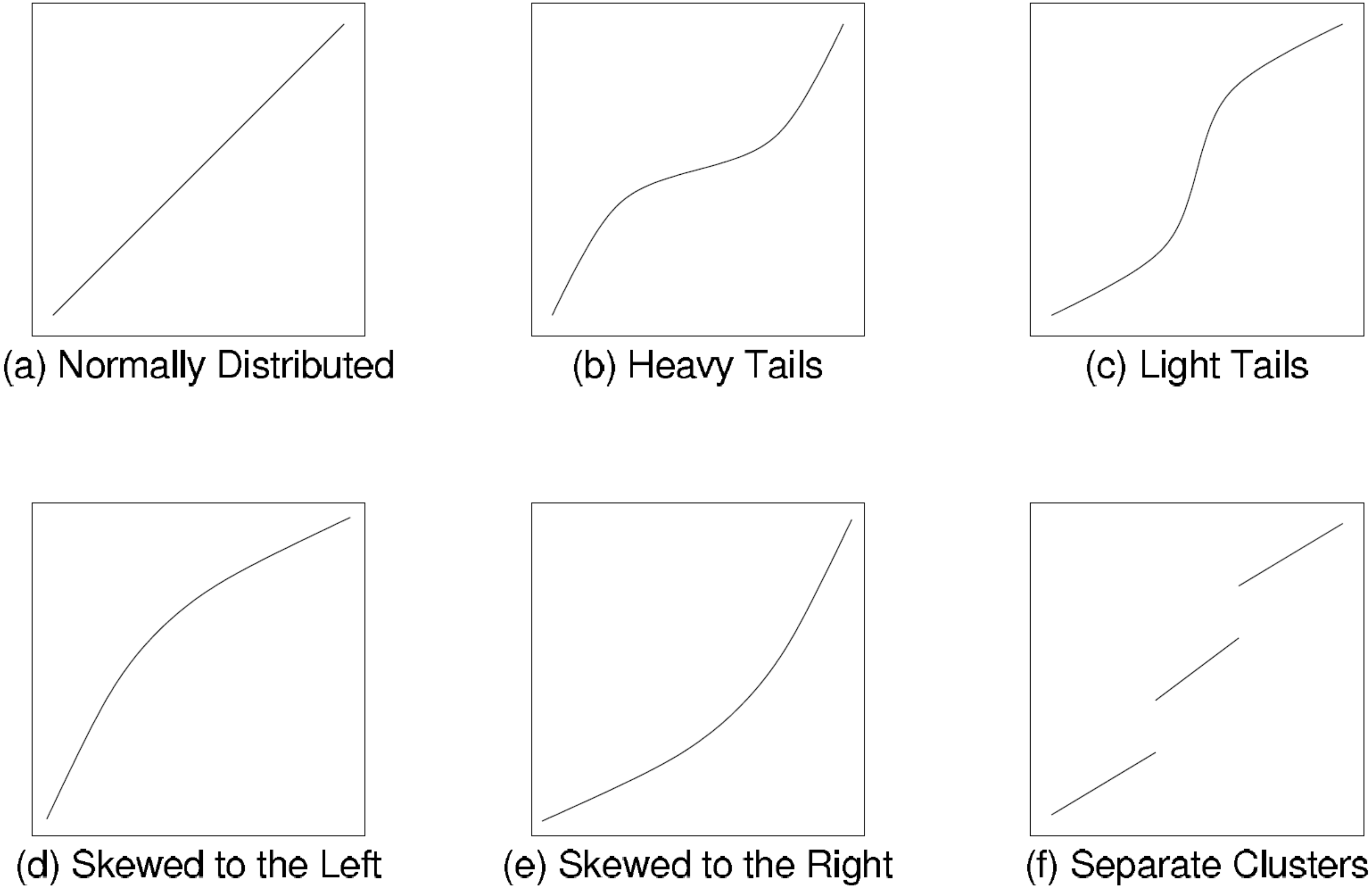
Let’s check the female weights

```
#Get female weights
femaleWeights=ds2016$Height[ds2016$Sex=="W"]
qqnorm(femaleWeights, main="QQ plot of female students weight")
qqline(femaleWeights)

hist(femaleWeights, freq = F)
xseq=seq(150, 200, 1)
lines(xseq, dnorm(xseq, mean=mean(femaleWeights), sd=sd(femaleWeights)
)))
```





QQ-Plots: Comparison with normal distribution



(a) Normally Distributed

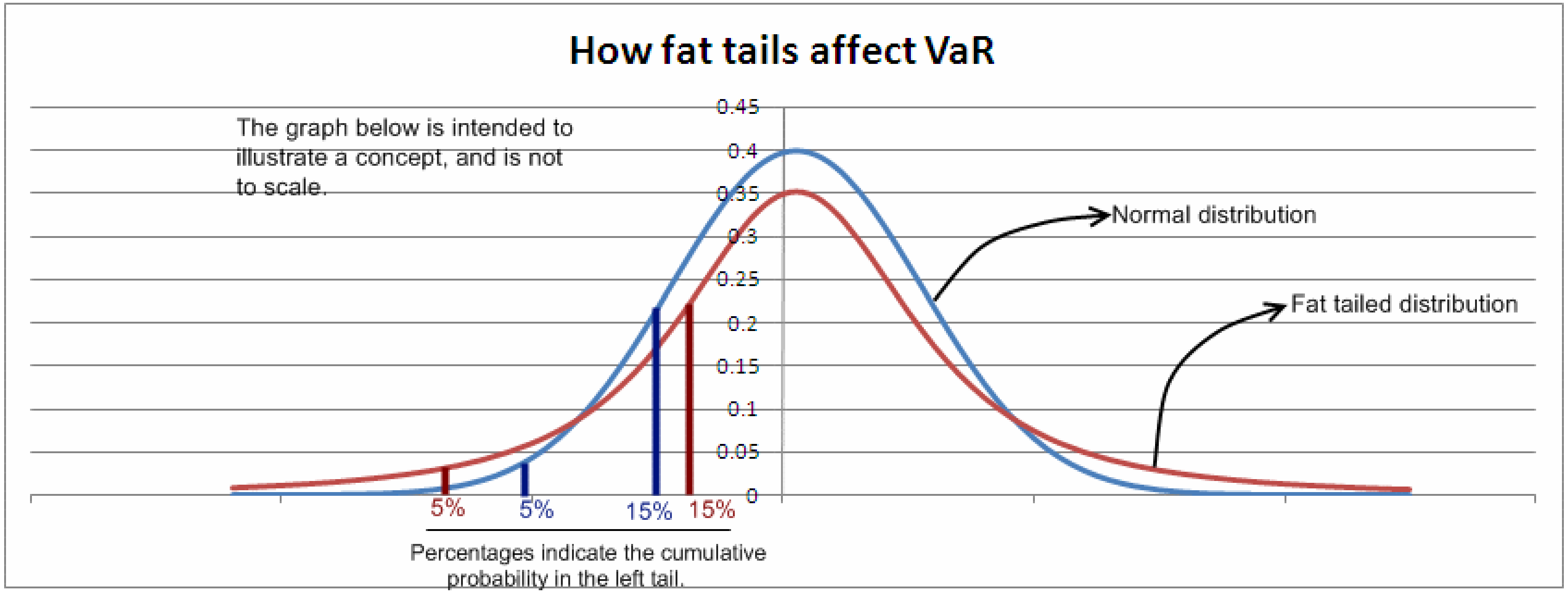
(b) Heavy Tails

(c) Light Tails

(d) Skewed to the Left

(e) Skewed to the Right

(f) Separate Clusters



How fat tails affect VaR

The graph below is intended to illustrate a concept, and is not to scale.

Percentages indicate the cumulative probability in the left tail.

Contingency tables

A contingency table (also referred to as cross tabulation or cross tab) is a type of table in a matrix format that displays the (multivariate) frequency distribution of the variables.

Example:

```
my.table <- table(ds2016$Sex,ds$Smoking)
my.table
prop.table(my.table)
```

	0	**1**	**2**
M	10	6	2
F	25	2	2

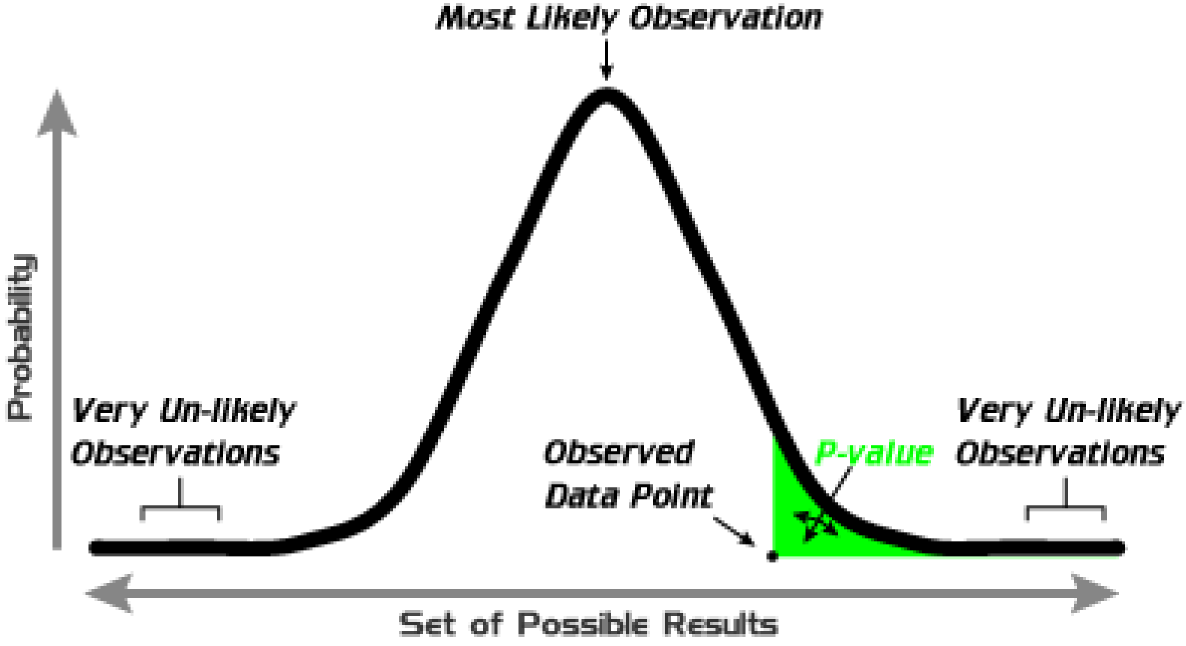
	0	**1**	**2**
M	0.21276596	0.12765957	0.04255319
F	0.53191489	0.04255319	0.04255319

Statistical tests

1. State the null and alternative hypotheses.
2. Consider the statistical assumptions being made about the sample.
3. Decide which test is appropriate and state test statistic T.
4. Derive the distribution of the test statistic under the null hypothesis
5. Select a significance level below which the null hypothesis will be rejected. Common values are 5% and 1%.
6. Compute the observed value t_{obs} of the test statistic T.
7. From the statistic calculate a probability of the observation under the null hypothesis (the p-value).
8. Reject the null hypothesis if and only if the p-value is less than the significance level threshold.

Statistical tests: p-values

In statistical significance testing the p-value is the probability of obtaining a test statistic **at least as extreme as or equal to the one that was observed** under the null hypothesis. One often rejects the null hypothesis when the p-value is less than the predetermined significance level which is often 0.05.



A p-value (shaded green area) is the probability of an observed (or more extreme) result arising by chance

Example: χ^2 Square Test (goodness of fit)

It tests a null hypothesis stating that the frequency distribution of certain events observed in a sample is consistent with a particular theoretical distribution. The events considered must be mutually exclusive and have total probability 1.

A common case for this is where the events each cover an outcome of a categorical variable. A simple example is the hypothesis that an ordinary six-sided die is "fair", i.e., all six outcomes are equally likely to occur.

χ^2 Square Test (goodness of fit)

The test statistic is given by:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

χ^2 = Pearson's cumulative test statistic, which asymptotically approaches a χ^2 distribution.

O_i = an observed frequency;

E_i = an expected (theoretical) frequency, asserted by the null hypothesis;

n = the number of cells in the table.

Unfair or fair die?

```
die.unfair <- sample(c(rep(1:6,each=3),6),size=1000,replace = TRUE)
# simulate rolls of a loaded die, 6 occurs more often than expected
prop.table(table(die.unfair)

    1     2     3     4     5     6
0.161 0.158 0.159 0.154 0.149 0.219
```

It is difficult to tell whether die is unfair or not. Could be random fluctuation around the expectation.

```
chisq.test(table(die.unfair), p=rep(1/6, each=6))
Chi-squared test for given probabilities

data:  (table(die.unfair))
X-squared = 20.264, df = 5, p-value = 0.001115
```

Unfair or fair die?

```
die.unfair <- sample(c(rep(1:6,each=3),6),size=1000,replace = TRUE)
# simulate rolls of a loaded die, 6 occurs more often than expected
prop.table(table(die.unfair)

    1     2     3     4     5     6
0.161 0.158 0.159 0.154 0.149 0.219
```

It is difficult to tell whether die is unfair or not. Could be random fluctuation around the expectation.

```
chisq.test(table(die.unfair), p=rep(1/6, each=6))
Chi-squared test for given probabilities

data:  (table(die.unfair))
X-squared = 20.264, df = 5, p-value = 0.001115
```

Unfair or fair die?

```
die.unfair <- sample(c(rep(1:6,each=3),6),size=1000,replace = TRUE)
# simulate rolls of a loaded die, 6 occurs more often than expected
prop.table(table(die.unfair)

    1     2     3     4     5     6
0.161 0.158 0.159 0.154 0.149 0.219
```

It is difficult to tell whether die is unfair or not. Could be random fluctuation around the expectation.

```
chisq.test(table(die.unfair), p=rep(1/6, each=6))
Chi-squared test for given probabilities

data:  (table(die.unfair))
X-squared = 20.264, df = 5, p-value = 0.001115
```

Another Example

Consider the random numbers picked by the students (Random). If the numbers have been picked randomly, they should be uniformly distributed. We can test this using the χ^2 test implementation in R.

```
st.rand <- table(ds2016$Random); st.rand # table with obs. freqs
chisq.test(st.rand,p = rep(1/length(st.rand), length(st.rand)))
```

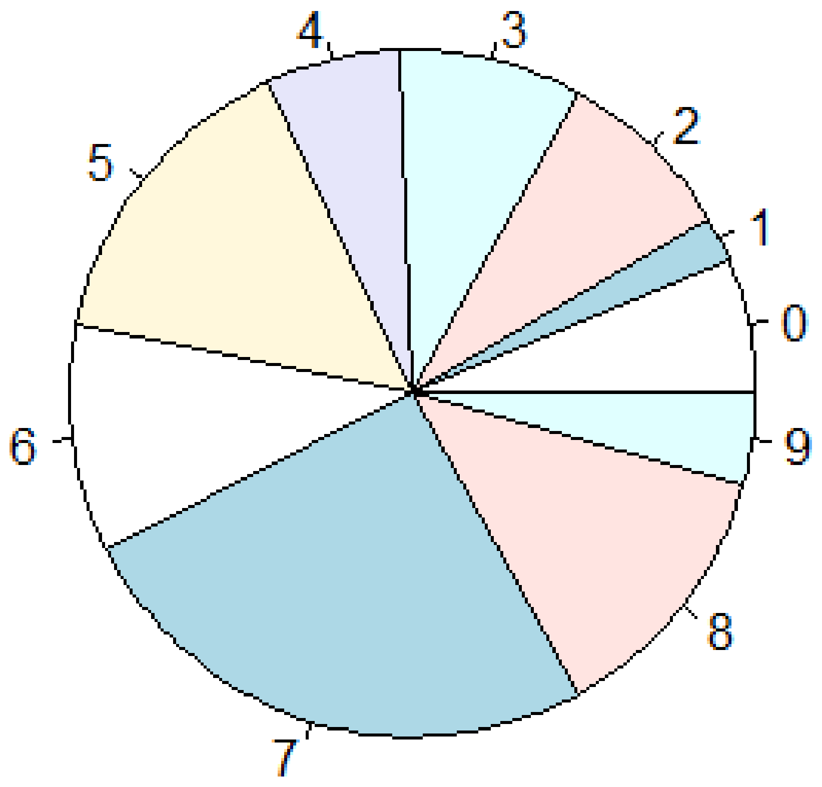
output:

```
 0  1  2  3  4  5  6  7  8  9
 3  1  4  4  3  7  5 12  6  2

   Chi-squared test for given probabilities

data:  rand.choice
X-squared = 18.745, df = 9, p-value = 0.02745
```

p-value is extremely small => we can reject the hypothesis that the random numbers are uniformly distributed

Distribution of chosen random numbers

```
pie(st.rand)
```



Student’s t-test

A class of tests in which the test statistic follows the t-distribution if the null-hypothesis is true.

Can be used to test whether two samples are significantly different from each other.

Assumption: normally distributed data

One sample tests: determines if a mean has a specific value
Two sample tests: compares the means of two samples

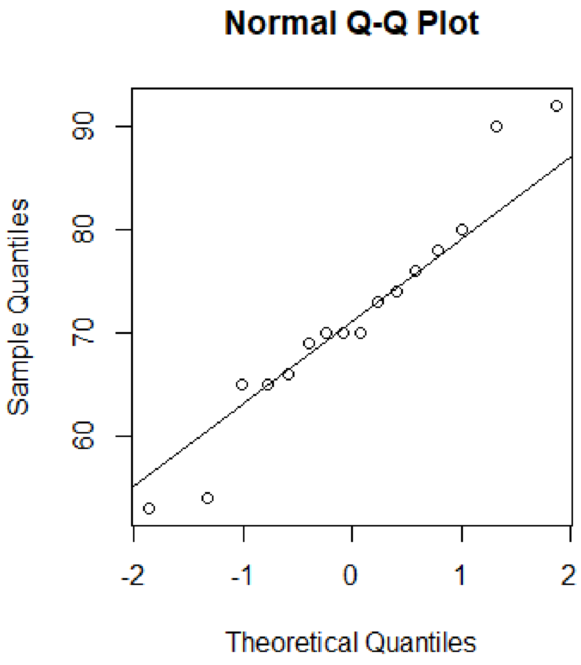
1. Unpaired: independent samples
2. Paired: dependent samples (e.g., tumor size before and after treatment)

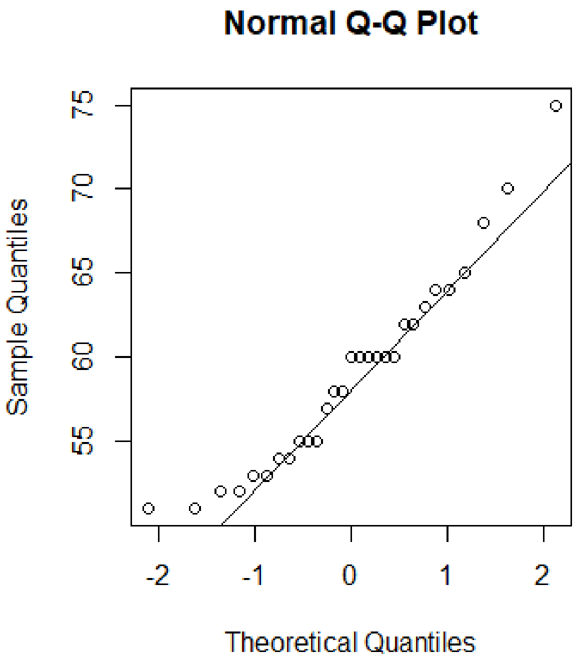
Example: t-test

We want to know whether the (mean) weight of males and females differs significantly from each other.

First, we need to check if data are normally distributed:

```
ind.M = which(ds2016$Sex == "M")
qqnorm((ds2016$Weight)[ind.M]); qqline((ds2016$Weight)[ind.M])
qqnorm((ds2016$Weight)[-ind.M]); qqline((ds2016$Weight)[-ind.M])
```





Example: t-test

Next, we can perform a t-test to determine the p-value of our data under the null hypothesis that the two samples are drawn from normal distributions with equal means.

```
t.test((ds2016$Weight)[ind.M],(ds2016$Weight)[-ind.M],paired=FALSE)

   Welch Two Sample t-test

data:  (ds2016$Weight)[ind.M] and (ds2016$Weight)[-ind.M]
t = 4.381, df = 20.322, p-value = 0.0002794
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  6.587134 18.537866
sample estimates:
mean of x mean of y 
  71.5625   59.0000 
```

Example: t-test

Next, we can perform a t-test to determine the p-value of our data under the null hypothesis that the two samples are drawn from normal distributions with equal means.

```
   Welch Two Sample t-test

data:  (ds2016$Weight)[ind.M] and (ds2016$Weight)[-ind.M]
t = 4.381, df = 20.322, p-value = 0.0002794
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  6.587134 18.537866
sample estimates:
mean of x mean of y 
  71.5625   59.0000
```

name of test that is used

Example: t-test

Next, we can perform a t-test to determine the p-value of our data under the null hypothesis that the two samples are drawn from normal distributions with equal means.

```
t.test((ds2016$Weight)[ind.M],(ds2016$Weight)[-ind.M],paired=FALSE)

   Welch Two Sample t-test

data:  (ds2016$Weight)[ind.M] and (ds2016$Weight)[-ind.M]
t = 4.381              value of test statistic
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  6.587134 18.537866
sample estimates:
mean of x mean of y 
  71.5625   59.0000 
```

Example: t-test

Next, we can perform a t-test to determine the p-value of our data under the null hypothesis that the two samples are drawn from normal distributions with equal means.

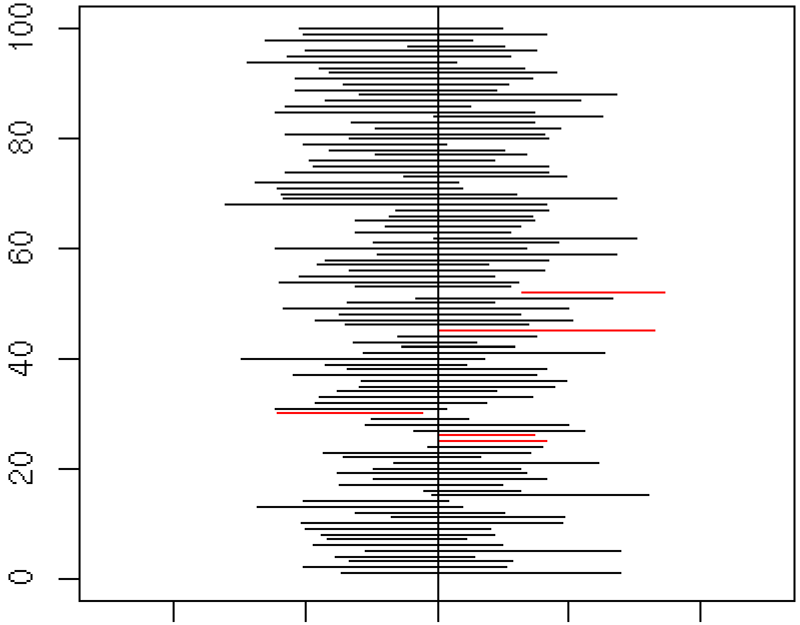
```
t.test((ds2016$Weight)[ind.M],(ds2016$Weight)[-ind.M],paired=FALSE)

   Welch Two Sample t-test

data:  (ds2016$Weight)[ind.M] and (ds2016$Weight)[-ind.M]
t = 4.381, df = 20.322, p-value = 0.0002794     p-value
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  6.587134 18.537866
sample estimates:
mean of x mean of y
  71.5625   59.0000
```

Confidence intervals

A confidence interval (CI) is a type of interval estimate of a parameter and it is used to indicate the reliability of an estimate.

CIs will be different from sample to sample. How frequently the observed interval contains the parameter is determined by the confidence level or confidence coefficient:

a 95% CI contains the true parameter in 95% of the cases.



Confidence intervals

Example: CI of normally distributed data

We can estimate the mean and the CI using the function ***t.test()***

```
t.test(ds2016$Weight)
```

```
        One Sample t-test

data:  ds2016$Weight
t = 43.091, df = 44, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 60.49833 66.43500
sample estimates:
mean of x
 63.46667
```

Returns an estimate of the mean and a 95% CI.

With probability 95%, the true mean lies between 60.498 and 66.435.

Confidence intervals

We can get CI for different intervals with parameter *conf.level*

```
> t.test(ds2016$Weight)
```

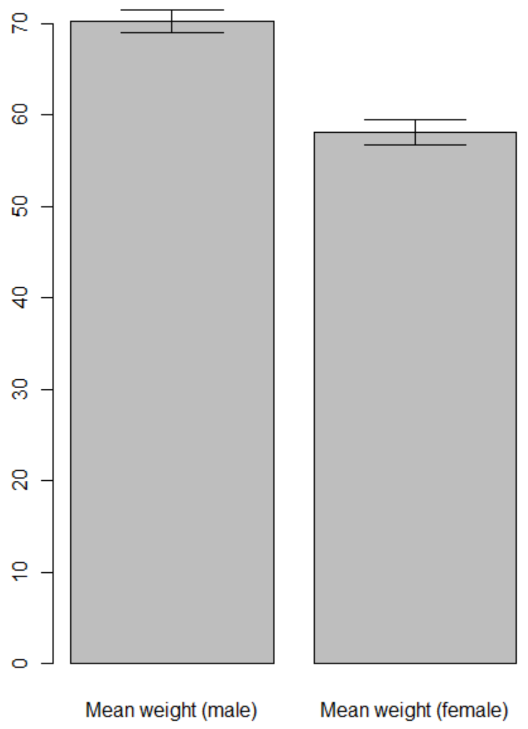
```
	One Sample t-test

data:  ds2016$Weight
t = 43.091, df = 44, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 60.49833 66.43500
```

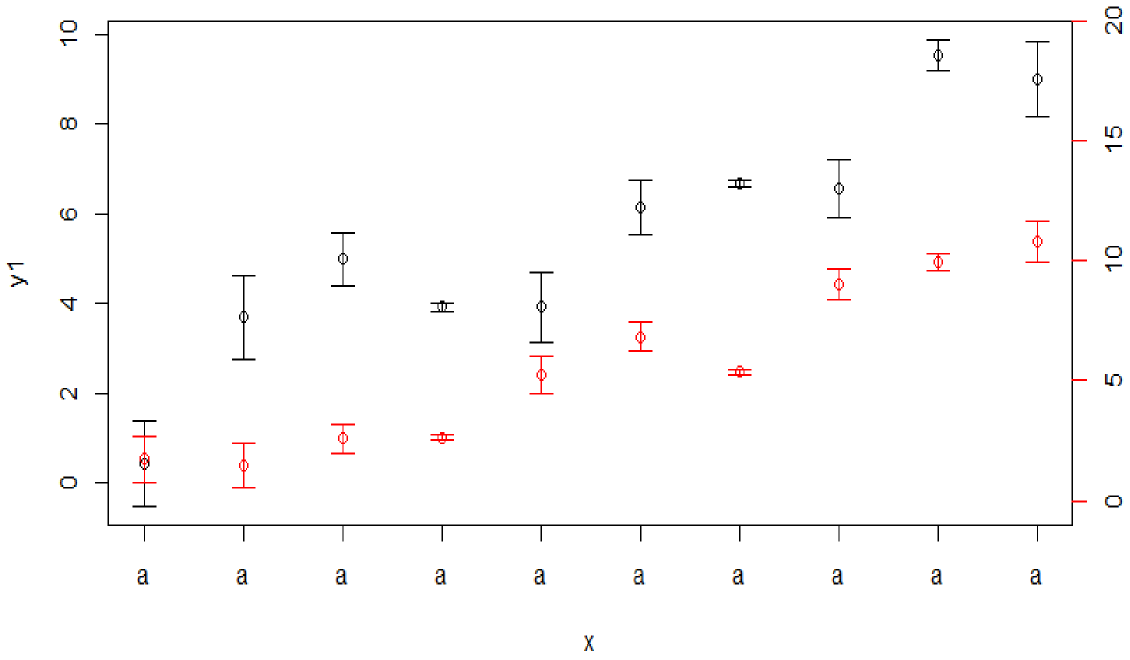
You can access the resulting CI values directly with *$conf.int*

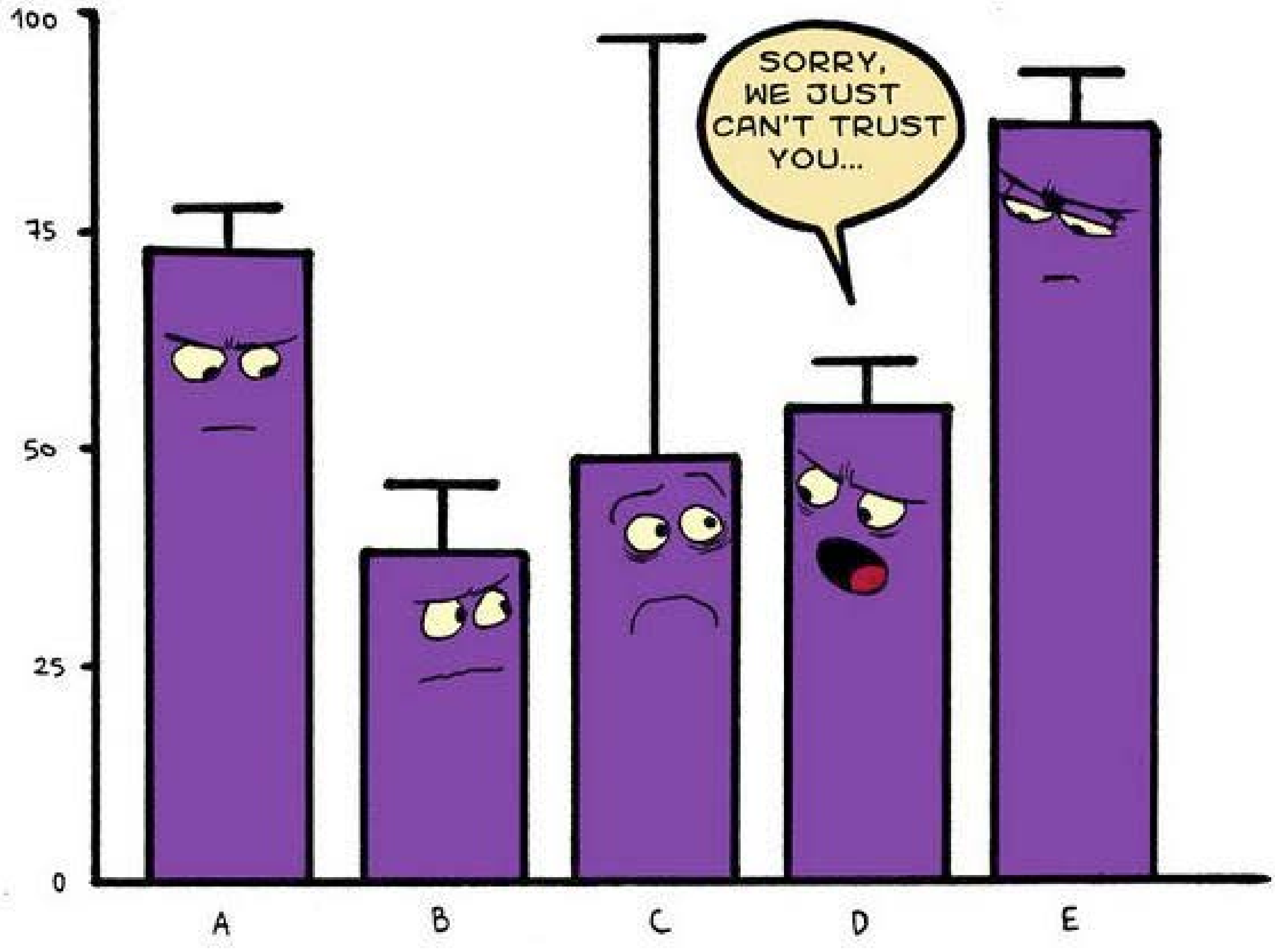
```
> res=t.test(ds2016$Weight, conf.level=0.99)
> res$conf.int
[1] 59.50134 67.43199
attr(,"conf.level")
[1] 0.99
> res$conf.int[c(1,2)]
[1] 59.50134 67.43199
```

Representation of CIs

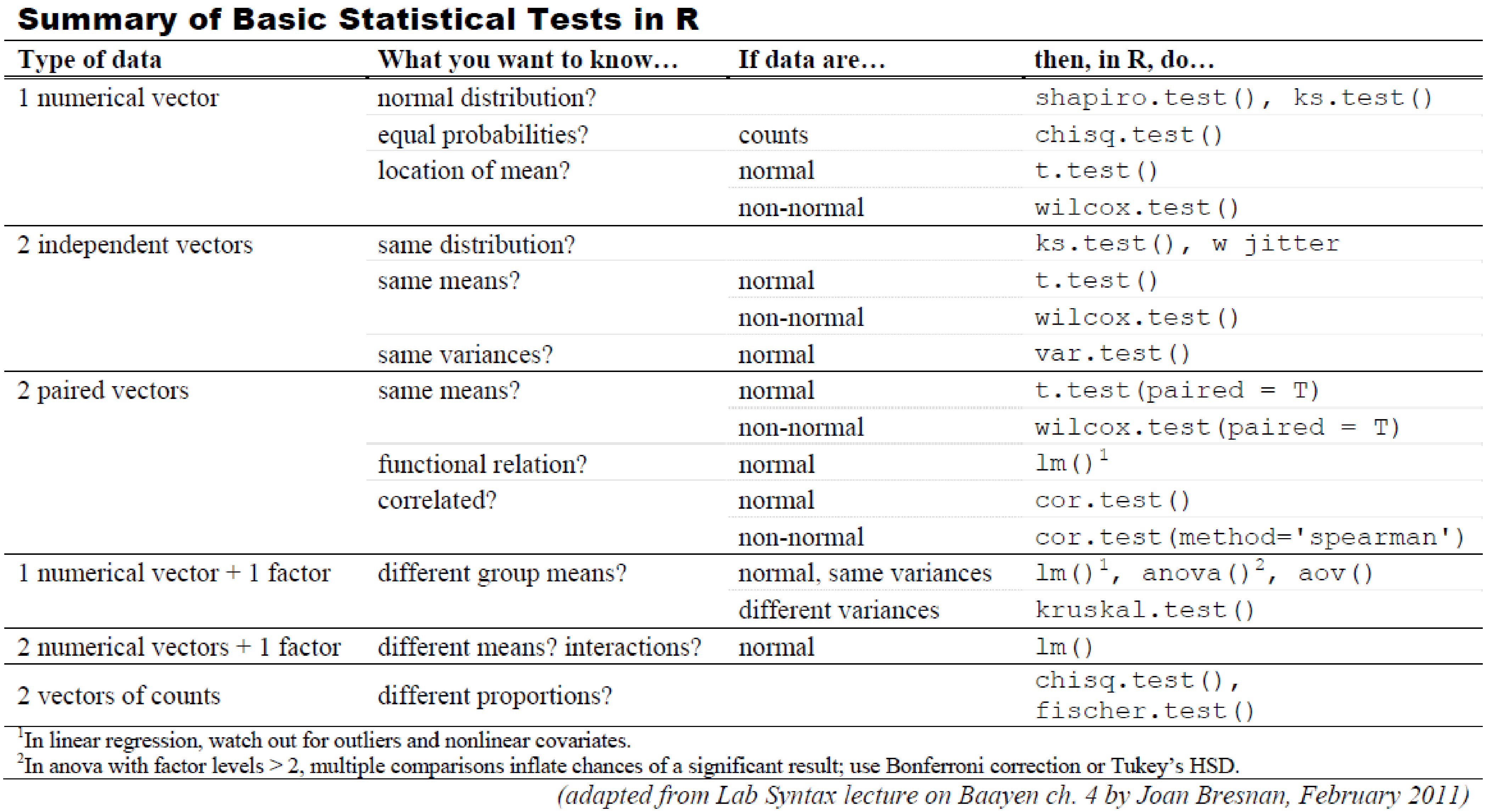
Usually CIs are shown as intervals around a point estimate:






100
75
50
25
0
A
B
C
D
E
SORRY, WE JUST CAN'T TRUST YOU...

R-functions for statistical testing



Summary of Basic Statistical Tests in R

Type of data	What you want to know…	If data are…	then, in R, do…
1 numerical vector	normal distribution?		shapiro.test(), ks.test()
	equal probabilities?	counts	chisq.test()
	location of mean?	normal	t.test()
		non-normal	wilcox.test()
2 independent vectors	same distribution?		ks.test(), w jitter
	same means?	normal	t.test()
		non-normal	wilcox.test()
	same variances?	normal	var.test()
2 paired vectors	same means?	normal	t.test(paired = T)
		non-normal	wilcox.test(paired = T)
	functional relation?	normal	lm() [1]
	correlated?	normal	cor.test()
		non-normal	cor.test(method='spearman')
1 numerical vector + 1 factor	different group means?	normal, same variances	lm() [1], anova() [2], aov()
		different variances	kruskal.test()
2 numerical vectors + 1 factor	different means? interactions?	normal	lm()
2 vectors of counts	different proportions?		chisq.test(), fischer.test()

[1] In linear regression, watch out for outliers and nonlinear covariates.

[2] In anova with factor levels > 2, multiple comparisons inflate chances of a significant result; use Bonferroni correction or Tukey's HSD.

(adapted from Lab Syntax lecture on Baayen ch. 4 by Joan Bresnan, February 2011)