Statistics for the Sciences - Part 2

Instructor: Demetrio Labate

September 24, 2023

4 Analysis of Variance

Analysis of Variance

Analysis of Variance or ANOVA is a class of widely used techniques for statistical analysis.

The term **analysis of variance** was introduced by R.A. Fisher in 1920 while conducting the analysis of agronomical data
cf. Fisher's book, "Statistical Methods for Research Workers"

The ANOVA technique is formulated for testing the difference among several means.

More technically, ANOVA is designed to split the variability of data into two parts: systematic factors and random factors.

The systematic factors have a statistical influence on the given data set, while the random factors do not.

Analysis of Variance

There are two main types of ANOVA: one-way and two-way.

- A **one-way ANOVA** evaluates the impact of a single factor on a single response variable. It determines whether all the samples are the same. The one-way ANOVA is used to determine whether there are any statistically significant differences between the means of three or more independent (unrelated) groups.
- A **two-way ANOVA** has two independent variable. For example, a two-way ANOVA allows a company to compare worker productivity based on two independent variables, such as salary and skill set. It is utilized to observe the interaction between the two factors and tests the effect of two factors at the same time.

One-way ANOVA

The one-way analysis of variance (ANOVA), also known as one-factor ANOVA, is an extension of independent two-samples t-test for comparing means in a situation where there are **more than two groups.**

In one-way ANOVA, the data is organized into several groups base on one single grouping variable, also called **factor** variable.

Example.

There is a group of patients who are suffering from a medical condition.
They are being given three different treatments (treatment = factor variable) that have the same functionality, i.e., to cure fever.

The ANOVA test can be applied to evaluate the effectiveness of each treatment and choose the best among them.

One-way ANOVA

Assumptions of ANOVA test:

1. observations are obtained independently and randomly from the population defined by the factor levels;
2. data of each factor level are normally distributed;
3. population variances must be equal, i.e., homoscedastic; homogeneity of variance means that the deviation of scores (measured by the range or standard deviation for example) is similar between populations; Levene's test can be used to check this property.

One-way ANOVA

The one-way ANOVA test generalizes the 2-population hypothesis testing problem to the multi-population setting.

Suppose we collected k independent random samples from k normally distributed population $N(\mu_j, \sigma_j)$, $j = 1, \ldots, k$ with the same variance.

We want to solve the following **hypothesis testing problem**:

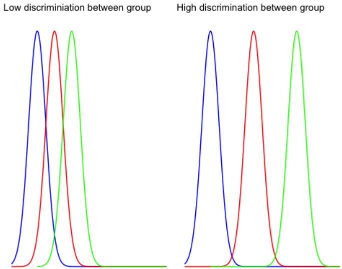
$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k$$
$$H_1 : \mu_i \neq \mu_j, \text{ for some } i, j$$

The **F-statistic** is used to test if the data are from significantly different populations, i.e., different sample means (since the rest is assumed to be equal).

The F-statistic measures the ratio of the **among-group variability** over the **within-group variability.**

One-way ANOVA

The **among-group variability** reflects the differences between the groups inside all of the population.



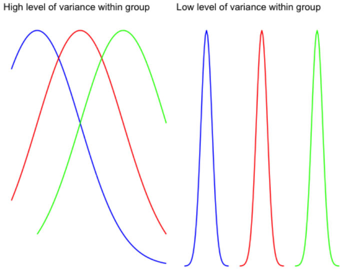
Low discriminiation between group

High discrimination between group

The left plot shows very little variation among the three groups and it is very likely that the three means tends to the overall mean i.e., mean for the three groups.
The right plot shows three distributions far apart with minimal overlap. There is a high chance the difference between the total mean and the groups mean will be large.

One-way ANOVA

The **within group variability** reflects the differences within each individual group: variation comes from individual observations.



The plots show the distributions of three different groups. On the left: variance is large and observations may be very different than the group means. On the right: individual groups have low variability and observations are close to the group means. In the first case, the F-test will decrease, meaning that you tend to accept the null hypothesis. In the second case, the F-test will increase and you tend to favor the alternative hypothesis.

One-way ANOVA - Mathematical formulation

Let X be a random variable defined for k subgroups of a population.

We assume that the observations from the k groups are independent and normally distributed, with the normal distribution $N(\mu_j, \sigma)$, $j = 1, \dots, k$ having the same variance σ^2.

Group	1	2	3	...	k	
	X_{11}	X_{12}	X_{13}	...	X_{1k}	
	X_{21}	X_{22}	X_{23}	...	X_{2k}	
	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	
	$X_{n_1 1}$	$X_{n_2 2}$	$X_{n_3 3}$	...	$X_{n_k k}$	
Mean	$\bar{X}_1$	$\bar{X}_2$	$\bar{X}_3$	...	$\bar{X}_k$	$\bar{X}$

$\bar{X}_j = \frac{1}{n_j} \sum_{i=1}^{n_i} X_{i,j}$ is the mean of observations in the j-th group

$\bar{X} = \frac{1}{N} \sum_{j=1}^{k} \sum_{i=1}^{n_j} X_{i,j}$ is the mean of all observations

$N = \sum_{j=1}^{k} n_j$

One-way ANOVA - Mathematical formulation

Definitions:

- Total sum of squares: $\mathrm{SST} = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (X_{i,j} - \bar{X})^2$
- Within group sum of squares: $\mathrm{SSW} = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (X_{i,j} - \bar{X}_j)^2$
- Sum of squares among groups: $\mathrm{SSA} = \sum_{j=1}^{k} n_j (\bar{X}_j - \bar{X})^2$

We have that $SST = SSA + SSW$

SSA is a measure of how widely dispersed the group averages $\bar{X}_j$ are about their center $\bar{X}$.

Thus, SSA is the basis for a test statistic for accepting the alternative hypothesis that the population means are not all the same.
To be useful, it has to be compared to SSW which is a measure of the inherent variability of the data.

One-way ANOVA - Mathematical formulation

Theorem

SSA and SSW are independent random variable.
In addition, $\frac{SSW}{\sigma^2} \sim \chi^2(df = N - k)$ and, if $H_0 : \mu_1 = \cdots = \mu_k$ is true, $\frac{SSA}{\sigma^2} \sim \chi^2(df = k - 1)$ and

$$F = \frac{N - k}{k - 1} \frac{SSA}{SSW}$$

is F-distributed with $k - 1$ degrees of freedom in the numerator and $N - k$ degrees of freedom in the denominator.

We define

- Mean squares among groups: $\text{MSA} = \frac{SSA}{k-1}$
- Mean squares within groups: $\text{MSW} = \frac{SSW}{N-k}$

Hence we can write $F = \frac{MSA}{MSW}$, which is called the **variance ratio**.

One-way ANOVA - Mathematical formulation

Intuitively, values of the variance ratio F close to 1 support H_0; values of the variance ratio F sufficiently larger than 1 support H_1.

We use F as the **test statistic** for the ANOVA test.

For a given significance level α, let $f_\alpha(r_1, r_2)$ be the $100(1-\alpha)$th percentile of the F distributions with r_1 degrees of freedom in the numerator and r_2 degrees of freedom in the denominator.

We reject H_0 is $F > f_\alpha(k-1, N-k)$.
Otherwise, we accept H_0.

One-way ANOVA - Mathematical formulation

Example. *In an experiments, $k = 4$ treatments are considered to address a medical condition. Each group sample size is $n = 10$, so that $N = 40$ is the total combined sample size.*

The summary statistics for samples of observations results in the following data: $SSW = 224.39, SSA = 37.65$.

We want to test the hypothesis

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$$
$$H_1 : \mu_i \neq \mu_j, \textit{ for some } i, j$$

using significance level $\alpha = 0.05$.

We compute the test statistic $F = \frac{N-k}{k-1}\frac{SSA}{SSW} = \frac{40-4}{4-1}\frac{37.65}{224.39} = 2.013$

We compute the critical value:

$f_\alpha = \texttt{qf}(0.95, \texttt{df1} = 3, \texttt{df2} = 36) = 2.866266.$

Conclusion: Since it $F < f_\alpha$, we accept H_0, that is, we cannot conclude that there is a difference in population means.

We can also compute the p-value:

p-value $= P(F > 2.013) = 1 - \texttt{pf}(2.013, 3, 36) = 0.129$

Conclusion: Since the p-value is below 0.05, we accept H_0.

One-way ANOVA

R example

`PlantGrowth` is a built-in R dataset containing the weight of plants obtained under a control and two different treatment conditions.

```
> my_data <- PlantGrowth
> print(my_data)
          weight    group
 1        4.17      ctrl
 2        5.58      ctrl
 3        5.18      ctrl
 ...      ...       ...
 10       5.14      ctrl
 11       4.81      trt1
 ...      ...       ...
 20       4.69      trt1
 21       6.31      trt2
 ...      ...       ...
 30       5.26      trt2
```

One-way ANOVA

We can inspect the format of the file as follows

```
> str(my_data)
'data.frame':  30 obs.  of 2 variables:
$ weight:  num 4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53
5.33 5.14 ...
$ group :  Factor w/ 3 levels "ctrl","trt1",..:  1 1
1 1 1 1 1 1 1 1 ...
```

The report shows that the table includes a column of numerical values and another column containing the labels describing the factor levels.

In this case, there are 3 levels associated with the “group” factor

One-way ANOVA

In R terminology, the column “group” is called **factor** and the different categories (`ctr, trt1, trt2`) are named factor levels.

```
> levels(my_data$group)
[1] "ctrl" "trt1" "trt2"
```

We compute summary statistics, that is, count, we list mean and standard deviation by groups using the `dplyr` package.

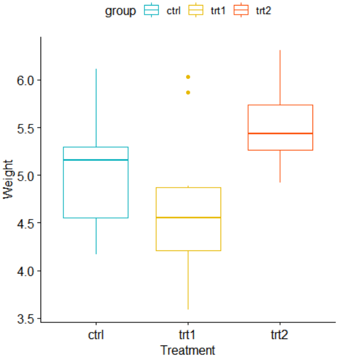
```
library(dplyr)
group_by(my_data, group) %>%
summarise(count = n(),mean = mean(weight, na.rm =
TRUE), sd = sd(weight, na.rm = TRUE))
```

	group	*count*	*mean*	*sd*
1	*ctrl*	10	5.03	0.583
2	*trt*1	10	4.66	0.794
3	*trt*2	10	5.53	0.443

One-way ANOVA

We can visualize summary statistics using boxplots.
For that, we install the ggpubr R package

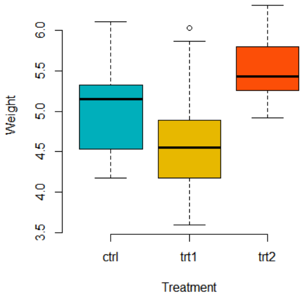
```
> install.packages("ggpubr")
> library("ggpubr")
> ggboxplot(my_data, x="group", y="weight", color="group",
palette=c("#00AFBB","#E7B800","#FC4E07"), order =
c("ctrl","trt1","trt2"), ylab="Weight",xlab="Treatment")
```



One-way ANOVA

Alternatively, we can use the command `boxplot`

```
> boxplot(weight ~ group, data = my_data, xlab =
"Treatment", ylab = "Weight", frame = FALSE, col =
c("#00AFBB", "#E7B800", "#FC4E07"))
```



One-way ANOVA

The R function `aov()` can be used to compute the ANOVA test

```
> res.aov <- aov(weight ~ group, data = my_data)
> summary(res.aov)
```

```
             Df   SumSq   MeanSq   Fvalue   Pr(> F)
group         2   3.766   1.8832   4.8461    0.0159   *
Residuals    27  10.492   0.3886
```

Interpretation: Since the p-value is less than the significance level 0.05, then there are significant differences between the groups; this is highlighted with ∗ in the model summary.

Note that:

- $Fvalue = \frac{MeanSq-group}{MeanSq-Residuals} = \frac{1.8832}{0.3886} = 4.8461$
- p-value $=$ `1-pf(4.8461,df1=2,df2=27)` $= 0.0159$
- $Fvalue = 4.8461 > \texttt{qf}(0.95, \texttt{df1} = 2, \texttt{df2} = 27) = 3.354131$ confirming that we can reject H_0

One-way ANOVA

Alternatively, we can use the R function `lm()`

```
> model = lm(weight ~ group, data = my_data)
> anova(model)
```

```
            Df   SumSq   MeanSq   Fvalue   Pr(> F)
 group       2   3.766   1.8832   4.8461    0.0159   *
 Residuals  27  10.492   0.3886
```

The output table is the same as the one we obtained above.

One-way ANOVA

In the one-way ANOVA test, a significant p-value indicates that some of the group means are different; however we *do not know which pairs of groups are different.*

To determine which pairs of groups are different, we can perform multiple pairwise-comparison.

The **Tukey HSD** (Tukey Honest Significant Differences) performs multiple pairwise-comparison between the means of groups and is useful to determine if the mean difference between specific pairs of group are statistically significant.

R function: `TukeyHSD()`

One-way ANOVA

```
> TukeyHSD(res.aov)
Tukey multiple comparisons of means
95% family-wise confidence level

Fit:  aov(formula = weight ~ group, data = my_data)

 $group
                 diff         lwr       upr      padj
 trt1 − ctrl   −0.371  −1.0622161 0.3202161 0.3908711
 trt2 − ctrl    0.494  −0.1972161 1.1852161 0.1979960
 trt2 − trt1    0.865   0.1737839 1.5562161 0.0120064
```

diff: difference between means of the two groups
lwr, upr: lower and the upper endpoints of 95% CI
p adj: p-value after adjustment for the multiple comparisons

Interpretation: only the difference between trt2 and trt1 is significant with an adjusted p-value of 0.012.

One-way ANOVA

The ANOVA test assumes that:

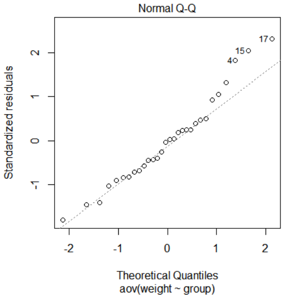
1. the data are normally distributed;
2. the variance across groups are homogeneous.

We will examine the validity of such assumptions on the data of the example.

One-way ANOVA

To check the normality assumption, we can display the Q-Q plot of residuals where the quantiles of the residuals are plotted against the quantiles of the normal distribution.

```
> plot(res.aov, 2)
```



As points fall approximately along this reference line, the plot indicates that the normality assumptions is acceptable.

Note: Points 17, 15, 4 are detected as outliers.

One-way ANOVA

We can also run the Shapiro-Wilk normality test

```
> aov_residuals <- residuals(object = res.aov)
> shapiro.test(x = aov_residuals)
```

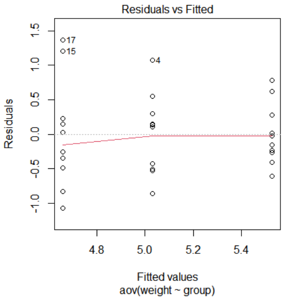
```
Shapiro-Wilk normality test
data:  aov_residuals
W = 0.96607, p-value = 0.4379
```

Since the p-value of the test is above 0.05, the test finds no indication that the normality assumption is violated.

One-way ANOVA

To check the homogeneity of variance assumption, we can display the residuals versus fits plot.

```
> plot(res.aov, 1)
```



As there is no evident relationships between residuals and fitted values (the mean of each groups), we can assume the homogeneity of variances.

Note: Points 17, 15, 4 are detected as outliers.

One-way ANOVA

To test the homogeneity assumption, we can run the Levene's test.

The function `leveneTest()` is available in the car package (you might need to install the package first).

```
> library(car)
> leveneTest(weight ~ group, data = my_data)
```

```
Levene's Test for Homogeneity of Variance (center =
median)
          Df   Fvalue   Pr(> F)
 group     2   1.1192    0.3412
          27
```

Since the p-value is above 0.05, there is no evidence to suggest that there is significant difference in variance across groups. Hence, we can assume the homogeneity of variances in the different treatment groups.

One-way ANOVA

If the assumptions required by the ANOVA test are not met, we can use a *non-parametric alternative*.

This test is called: **Kruskal-Wallis rank sum test**

```
> kruskal.test(weight ~ group, data = my_data)
```

```
Kruskal-Wallis rank sum test

data:  weight by group
Kruskal-Wallis chi-squared = 7.9882, df = 2, p-value =
0.01842
```

The p-value less than 0.05 indicates that there is a significant difference across groups.

One-way ANOVA - RStudio with CSV files

To perform a single factor ANOVA using RStudio, you need to set up a table with two or more columns.

Although it is possible to enter the data directly into the script, in practical applications you might want to load the data from a CSV file, probably one created using Excel or another spreadsheet software.
You can use the command `read.csv()` to load the table.

The format of the table typically consists of one column containing the continuous data to be analyzed and one or more columns containing values that assign the row to one or more groups.

Note: the names used to assign a row to a given group must be **exactly** the same in every row.

One-way ANOVA - RStudio with CSV files

In the following example, we consider the results of an experiment comparing 72 measurements of the amplitude of the response of cockroach eyes stimulated by pulses of red, green and blue light having the same brightness and duration.

```
> ExpData <- read.csv("C:/ma4310/color-example.csv")
```

```
> print(ExpData)
       color   response
 1       red       1.90
 2       red       2.60
...      ...
 25    green       9.10
...      ...
 48    green        6.8
 49     blue        7.6
...      ...
 72     blue        6.8
```

One-way ANOVA - RStudio with CSV files

We next exploring the structure of the data file.

```
> str(ExpData)
'data.frame':  72 obs.  of 3 variables:
$ color :  chr "red" "red" "red" "red" ...
$ response:  num 1.9 2.6 3.4 0.8 5.3 1.5 4.5 2.6 1.16 ...
```

Since the color vector is a character, *we convert it into a factor*.

```
> ExpData$color <- factor(ExpData$color, levels =
c("red","green","blue"),labels = c("RED", "GREEN","BLUE"))
```

Now, the data are organized into a numerical vector and a factor vector with 3 levels.

```
> str(ExpData)
'data.frame':  72 obs.  of 3 variables:
$ color :  Factor w/ 3 levels "RED","GREEN",...:  1 1 1 1
1 1 1 1 1 ...
$ response:  num 1.9 2.6 3.4 0.8 5.3 1.5 4.5 2.6 1.16 ...
```

One-way ANOVA - RStudio with CSV files

For a more detailed analysis of the data file, we can use the command `Summarize` from the FSA library.

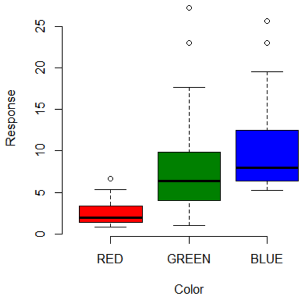
```
> install.packages("FSA")
> library(FSA)
> Summarize(response ~ color, data=ExpData,digits=3)
```

```
     color   n    mean     sd   min     Q1  median     Q3   max
1      RED  24   2.492  1.546  0.80  1.405     2.0   3.40   6.7
2    GREEN  24   8.530  6.978  1.00  4.000     6.4   9.85  27.2
3     BLUE  24  10.632  5.976  5.27  6.600     8.0  11.75  25.6
```

One-way ANOVA - RStudio with CSV files

Here is the boxplot

```
> boxplot(response ~ color, data = ExpData, xlab =
"Color", ylab = "Response", frame = FALSE, col =
c("#FF0000", "#008000", "#0000FF"))
```



One-way ANOVA - RStudio with CSV files

We compute the ANOVA test under the null hypothesis is that there is no difference between the mean response to red light and the mean response to green light. Note: ∗ ∗ ∗ indicates that p-value is less than 0.001.

```
> res.aov <- aov(response ~ color, data = ExpData)
> summary(res.aov)
```

```
            Df   SumSq   MeanSq   Fvalue      Pr(> F)
 color       2   857.2    428.6    14.81   4.44e − 06   ∗ ∗ ∗
 Residuals  69  1996.4     28.9
```

Interpretation:

As the p-value is less than the significance level 0.05, we conclude that there are significant differences between the 3 colors

One-way ANOVA - RStudio with CSV files

```
> TukeyHSD(res.aov)
Tukey multiple comparisons of means
95% family-wise confidence level

Fit:  aov(formula = response ~ color, data = ExpData)

 $group
                   diff        lwr        upr        padj
 GREEN − RED     6.038750   2.319372   9.758128   0.0006628
 BLUE − RED      8.140417   4.421039  11.859795   0.0000049
 BLUE − GREEN    2.101667  −1.617711   5.821045   0.3709119
```

Interpretation: only the differences GREEN-RED and BLUE-RED are significant but the difference BLUE-GREEN is not.

Two-way ANOVA

Two-way ANOVA test is used to evaluate simultaneously the effect of **two grouping variables** on a response variable.

The grouping variables are also known as **factors**.

The different categories or groups of a factor are called **levels**. The number of levels can vary between factors.

The level combinations of factors are called **cell**.

1. When the sample sizes within cells are equal, we have the so-called **balanced design**. In this case, the standard two-way ANOVA test can be applied.
2. When the sample sizes within each level of the independent variables are not the same, we have the case of **unbalanced designs** and the ANOVA test should be handled differently.

Two-way ANOVA

R Example: we will use the built-in R data set named `ToothGrowth`.

It contains data from a study evaluating the effect of vitamin C on tooth growth in Guinea pigs.

The experiment has been performed on 60 pigs, where each animal received one of **three dose levels** of vitamin C (0.5, 1, and 2 mg/day) by one of **two delivery methods**, (orange juice, coded as OJ, or ascorbic acid, coded as VC).

Tooth length was measured to test the hypothesis if tooth length depends on dose and delivery method.

Note: An experiment where we test a combination of the levels of multiple factors is called a **factorial treatment structure** or a **factorial design**.

Two-way ANOVA

```
> my_data <- ToothGrowth
> print(my_data)
       len    supp   dose
 1     4.2    VC     0.5
 2     11.5   VC     0.5
 ...   ...    ...    ...
 11    16.5   VC     1.0
 ...   ...    ...    ...
 21    23.6   VC     2.0
 ...   ...    ...    ...
 31    15.2   OJ     0.5
 ...   ...    ...    ...
 41    19.7   OJ     1.0
 ...   ...    ...    ...
 51    25.5   OJ     2.0
 ...   ...    ...    ...
 60    23.0   OJ     2.0
```

`len` = tooth legth; `supp` = delivery method; `dose` = dose level

Two-way ANOVA

We can check the data structure

```
> levels(my_data$group)
NULL
> str(my_data)
'data.frame':  60 obs.  of 3 variables:
$ len :  num 4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7...
$ supp:  Factor w/2 levels "OJ","VC":2 2 2 2 2 2 2 2 2...
$ dose:  num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5...
```

From the output above, R considers dose as a numeric variable. Hence, we convert it to a *factor* variable as follows:

```
> my_data$dose <- factor(my_data$dose, levels = c(0.5, 1,
2),labels = c("D0.5", "D1.0", "D2.0"))
```

Two-way ANOVA

We can list the data again

```
> print(my_data)
        len     supp    dose
 1      4.2     VC      D0.5
 2      11.5    VC      D0.5
 ...    ...     ...     ...
 11     16.5    VC      D1.0
 ...    ...     ...     ...
 21     23.6    VC      D2.0
 ...    ...     ...     ...
 31     15.2    OJ      D0.5
 ...    ...     ...     ...
 41     19.7    OJ      D1.0
 ...    ...     ...     ...
 51     25.5    OJ      D2.0
 ...    ...     ...     ...
 60     23.0    OJ      D2.0
```

Two-way ANOVA

We can generate the frequency table to display one factor/group (dose level) vs the other factor/group (delivery method).

```
>table(my_data$supp, my_data$dose)
```

	$D0.5$	$D1.0$	$D2.0$
OJ	10	10	10
VC	10	10	10

Note in the table above:

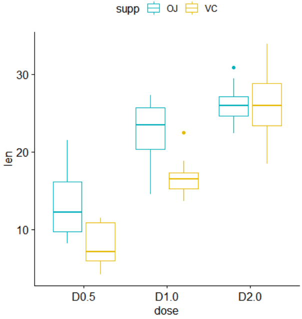
1. Factorial treatment structure: we observe all possible combinations of the levels of the factors.
2. Balanced design: the sample size is the same in each cell.

Two-way ANOVA

We can visualize the data using boxplots

Here are the plots using the ggpubr library.

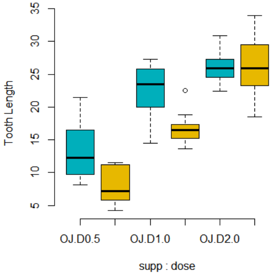
```
# Plot tooth length ("len") by groups ("dose")
# Color box plot by a second group:  "supp"
>library("ggpubr")
>ggboxplot(my_data, x = "dose", y = "len", color =
"supp", palette = c("#00AFBB", "#E7B800"))
```



Two-way ANOVA

Here are the plots using the built-in boxplot command.

```
>boxplot(len ~ supp * dose, data=my_data, frame =
FALSE, col = c("#00AFBB", "#E7B800"), ylab="Tooth
Length")
```



Two-way ANOVA

We next compute the two-way ANOVA test in R using the R function `aov()`

We use the function `summary.aov()` to summarize the analysis of variance model.

```
>res.aov2 <- aov(len ~ supp + dose, data = my_data)
>summary(res.aov2)
```

	Df	*SumSq*	*MeanSq*	*Fvalue*	$Pr(>F)$	
supp	1	205.4	205.4	14.02	0.000429	$***$
dose	2	2426.4	1213.2	82.81	$< 2e - 16$	$***$
Residuals	56	820.4	14.7			

Table shows that both supp and dose are **statistically significant**.

Conclusion: both delivery methods and the dose of vitamin C have a significant impact on the mean tooth length.

Two-way ANOVA

The above fitted model is an **additive model.**
It assumes that the two factor variables are independent.

In general, the two factors might interact to create an synergistic effect.
This is called **two-way ANOVA with interaction effect**.

To solve this modified ANOVA test, we need replace the plus symbol (+) by an asterisk (*), as follow.

```
# These two calls are equivalent
>res.aov3 <- aov(len ~ supp * dose, data = my_data)
>res.aov3 <- aov(len ~ supp + dose + supp:dose, data
= my_data)
```

Two-way ANOVA

```
> summary(res.aov3)
```

	Df	*SumSq*	*MeanSq*	*Fvalue*	$Pr(> F)$	
supp	1	205.4	205.4	15.572	0.000231	***
dose	2	2426.4	1213.2	92.000	$< 2e - 16$	***
supp : *dose*	2	108.3	54.2	4.107	0.021860	*
Residuals	54	712.1	13.2			

The table shows that the two factors (supp and dose) as well as their interaction are statistically significant.

Specifically:
- the p-value of supp is 0.000429, hence the levels of supp are associated with significant different tooth length;
- the p-value of dose is below 2e-16, hence the levels of dose are associated with significant different tooth length;
- the p-value for the interaction between supp and dose is 0.021860, hence the relationships between dose and tooth length depends on the supp method.

Two-way ANOVA

As the ANOVA test is significant, we compute the Tukey HSD test for performing pairwise-comparison between the means of groups. We don't need to perform the test for the "supp" variable because it has only two levels, which have been already proven to be significantly different by ANOVA test. Therefore, we will run the Tukey HSD test only for the factor variable "dose".

```
> TukeyHSD(res.aov3, which = "dose")
```

```
Tukey multiple comparisons of means
95% family-wise confidence level
Fit: aov(formula = len ~ supp * dose, data = my_data)
 $dose
                  diff         lwr         upr        padj
 D1.0 − D0.5     9.130    6.362488   11.897512   0.0e + 00
 D2.0 − D0.5    15.495   12.727488   18.262512   0.0e + 00
 D2.0 − D1.0     6.365    3.597488    9.132512   2.7e − 06
```

diff: difference between means of the two groups
lwr, upr: lower and the upper endpoints of 95% CI
p adj: p-value after adjustment for the multiple comparisons

Two-way ANOVA

The ANOVA test assumes that:

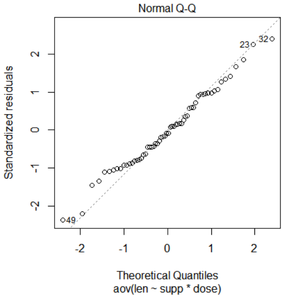
1. the data are normally distributed;
2. the variance across groups are homogeneous.

We will examine the validity of such assumptions on the data of the example.

Two-way ANOVA

To check the normality assumption, we cdisplay the Q-Q plot of residuals where the quantiles of the residuals are plotted against the quantiles of the normal distribution.

```
> plot(res.aov3, 2)
```



As points fall approximately along this reference line, the plot indicates that the normality assumptions is acceptable.

Note: Points 23, 32, 49 are detected as outliers.

Two-way ANOVA

We can also run the Shapiro-Wilk normality test

```
>aov_residuals <- residuals(object = res.aov3)
>shapiro.test(x = aov_residuals)
```

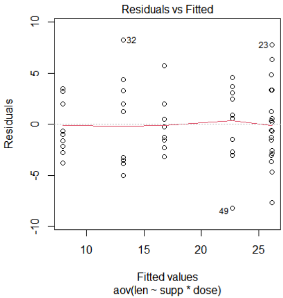
```
Shapiro-Wilk normality test
data:  aov_residuals
W = 0.98499, p-value = 0.6694
```

Since the p-value of the test is above 0.05, the test finds no indication that the normality assumption is violated.

Two-way ANOVA

To check the homogeneity of variance assumption, we can display the residuals versus fits plot:

```
>plot(res.aov3, 1)
```



As there is no evident relationships between residuals and fitted values (the mean of each groups), we can assume the homogeneity of variances.

Note: Points 23, 32, 49 are detected as outliers.

Two-way ANOVA

To test the homogeneity assumption, we can run the Levene's test.

The function `leveneTest()` is available in the car package.

```
>library(car)
>leveneTest(len ~ supp*dose, data = my_data)
```

```
Levene's Test for Homogeneity of Variance (center =
median)
          Df   Fvalue   Pr(> F)
  group    5   1.7086    0.1484
          54
```

Since the p-value is above 0.05, there is no evidence to suggest that there is significant difference in variance across groups. Hence, we can assume the homogeneity of variances in the different treatment groups.

Two-way ANOVA - Unbalanced design

In example above, the experiment has equal numbers of subjects in each group.

An **unbalanced design** has unequal numbers of subjects in each group.

There are three fundamentally different ways to run an ANOVA in an unbalanced design.
They are known as Type-I, Type-II and Type-III sums of squares.
The three methods give the same result when the design is balanced. However, when the design is unbalanced, they don't give the same results.

Two-way ANOVA - Unbalanced design

Type I is also called *sequential sum of squares*. Because of the sequential nature and the fact that the two main factors are tested in a particular order, this type of sums of squares will give different results for unbalanced data depending on which main effect is considered first.

Type II tests for each main effect after the other main effect and assumes *no significant interaction*. Computationally, this method is equivalent to running a type I analysis with different orders of the factors.

Type III tests for the presence of a main effect after the other main effect and interaction. This approach is therefore valid in the presence of *significant interactions*.

Two-way ANOVA - Unbalanced design

The function `Anova()` in the car package can be used to compute two-way ANOVA test for unbalanced designs. You might need to first install (`install.packages(\car")`)

```
library(car)
> my_anova <- aov(len ~ supp*dose, data = my_data)
> Anova(my_anova, type = "III")
```

```
Response:  len
               SumSq   Df   Fvalue      Pr(> F)
 (Intercept)  889.35    1  53.3438  1.090e − 09   * * *
 supp         227.15    1  13.6246    0.0005073   * * *
 dose         711.88    1  42.6991  2.028e − 08   * * *
 supp : dose   88.92    1   5.3335    0.0246314   *
 Residuals    933.63   56
```

Note: we apply type III anova since we consider a model *with interaction*.

ANOVA with Blocks

Blocks are used in an analysis of variance to account for suspected variation from factors other than the treatments.

The **randomized complete block design**, for instance, is a standard design for agricultural experiments in which similar experimental units are grouped into blocks or replicates.

Assume that we can divide our experimental units into r groups, also known as **blocks**, containing g experimental units each. Think for example of an agricultural experiment at r different locations having g different plots of land each. Hence, a block is given by a location and an experimental unit by a plot of land.

The experimental units should be as similar as possible within the same block, but can be very different between different blocks. This design allows us to fully remove the between-block variability. For the analysis of a randomized complete block design, the block is treated as a factor in our model.

ANOVA with Blocks

Example. At six different locations (**factor block**), three plots of land were available.
Three varieties of oat (**factor variety**), i.e., Goldenrain, Marvellous and Victory varieties, were randomized to them, individually per location.
The response was yield (in 0.25lbs per plot)

We want to test the *hypothesis that the yield is the same for all oat varieties* against the alternative hypothesis that the yield is not the same for all oat varieties.

We will use two-way ANOVA to solve the problem.

ANOVA with Blocks

```
> block <- factor(rep(1:6, times = 3))
> variety <- factor(rep(c("Goldenrain", "Marvellous",
"Victory"), each = 6))
> yield <- c(133.25, 113.25, 86.75, 108, 95.5,
90.25,129.75, 121.25, 118.5, 95, 85.25, 109,143,
87.25, 82.5, 91.5, 92, 89.5)
> oat.variety <- data.frame(block, variety, yield)
> str(oat.variety)
```

```
'data.frame': 18 obs. of 3 variables:
$ block: Factor w/ 6 levels ”1”,”2”,”3”,”4”,..: 1 2 3 4 5 6 1 2 3 ...
$ variety: Factor w/ 3 levels ”Goldenrain”,”Marvellous”,..: 1 1 1 1
1 1 2 2 2 2 ...
$ yield: num 133.2 113.2 86.8 108 95.5 ...
```

ANOVA with Blocks

Note that the experiment has a balanced design with the same number of experimental units in each cell

```
> table(oat.variety$block, oat.variety$variety)
```

	Goldenrain	*Marvellous*	*Victory*
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	1	1	1

ANOVA with Blocks

```
> aov.model <- aov(yield ~ block + variety, data =
oat.variety)
> summary(aov.model)
```

	Df	*SumSq*	*MeanSq*	*Fvalue*	*Pr*(> *F*)	
block	5	3969	793.8	5.280	0.0124	*
variety	2	447	223.3	1.485	0.2724	
Residuals	10	1503	150.3			

Conclusion: we accept the null hypothesis that the 3 varieties of oat give the same yield.
The table also shows that there is a statistically significant difference among the blocks, suggesting that different locations have statistically significant different yields (even though this is of no interest in the experiment).

Note. It is good practice to write the block factor first; in case of unbalanced data, we would get the effect of variety adjusted for block.

ANOVA with Blocks

Example (from https://rcompanion.org/handbook). Brendon Small, Coach McGuirk, and Melissa Robbins are 3 instructors teaching different nutrition education programs. During their course, they have their students keep diaries of what they eat for a week and then calculate the daily sodium intake in milligrams. They want to *test if the mean sodium intake is the same among classes.*
They suspect though that the town of residence may have some effect on sodium intake since each town has varying income, ethnic makeup, and other demographic factors. Therefore they have recorded the town each student is from, and they would like to use this as a blocking variable.

ANOVA with Blocks

```
Input = ("
Instructor        Town            Sodium
'BrendonSmall'    Squiggleville   1200
'BrendonSmall'    Squiggleville   1400
'BrendonSmall'    Squiggleville   1350
'BrendonSmall'    Metalocalypse   950
...               ...
'CoachMcGuirk'    Squiggleville   1100
'CoachMcGuirk'    Squiggleville   1200
...               ...
'MelissaRobins'   Metalocalypse   900
...               ...
'MelissaRobins'   Squiggleville   1400
'MelissaRobins'   Metalocalypse   1100
")
```

```
> Data =
read.table(textConnection(Input),header=TRUE)
```

ANOVA with Blocks

```
> str(Data)
'data.frame':  60 obs.  of 3 variables:
$ Instructor:  chr "Brendon Small" "Brendon Small"
"Brendon Small" "Brendon Small" ...
$ Town :  chr "Squiggleville" "Squiggleville"
"Squiggleville" "Metalocalypse" ...
$ Sodium :  int 1200 1400 1350 950 1400 1150 1300 1325
1425 1500 ...
```

ANOVA with Blocks

We redefine Instructor and Town as factors.

```
> Data$Instructor =
factor(Data$Instructor,levels=unique(Data$Instructor))
> Data$Town = factor(Data$Town,levels=unique(Data$Town))
```

```
> str(Data)
'data.frame':  60 obs.  of 3 variables:
$ Instructor:  Factor w/ 3 levels "Brendon Small",..:  1 1
1 1 1 1 1 1 1 1 ...
$ Town :  Factor w/ 2 levels "Squiggleville",..:  1 1 1 2
1 1 1 2 2 1 ...
$ Sodium :  int 1200 1400 1350 950 1400 1150 1300 1325
1425 1500 ...
```

ANOVA with Blocks

```
> table(Data$Instructor, Data$Town)
```

	Squiggleville	*Metalocalypse*
BrendonSmall	11	9
CoachMcGuirk	14	6
MelissaRobins	6	14

The table shows that data are unbalanced, with a different number of experiments for each cell.

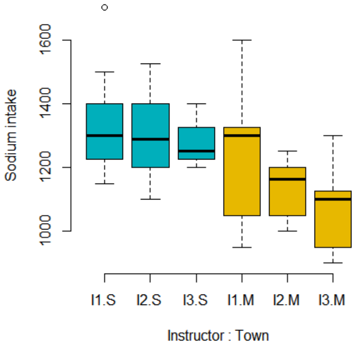
ANOVA with Blocks

Before generating the boxplot, for the benefit of visualization, we relabel the factor levels using shorter symbols.

```
> Data$Town = factor(Data$Town,levels=unique(Data$Town),
labels=c("S","M"))
> Data$Instructor =
factor(Data$Instructor,levels=unique(Data$Instructor),
labels=c("I1","I2","I3"))
```

ANOVA with Blocks

```
> boxplot(Sodium   Instructor * Town, data=Data,
frame = FALSE, col = rep(c("#00AFBB",
"#E7B800"),each=3), ylab="Sodium intake")
```



ANOVA with Blocks - Unbalanced design

We use the function `Anova()` in the car package to compute **two-way ANOVA test for unbalanced design**.

```
library(car)
> my_anova <- aov(Sodium ~ Town + Instructor, data =
Data)
> Anova(my_anova, type = "II")
```

```
Response:  Sodium
               SumSq  Df    Fvalue    Pr(> F)
 Town         329551   1   15.9332  0.0001928   * * *
 Instructor   148944   2    3.6006  0.0338033   *
 Residuals   1158261  56
```

Conclusion: there is statistically significant difference among Instructors; also Town has a statistically significant impact (hence it was useful to use the town as a block factor)

ANOVA with Blocks - Unbalanced design

In this example, the Anova function is useful to compute the ANOVA test of an unbalanced design experiment.

If we use the standard solution for the balanced design setting, we do get a different and slightly less accurate result

```
> summary(my_anova)
               Df     SumSq    MeanSq    Fvalue      Pr(> F)
 Town           1    470753    470753    22.760    1.35e − 05    * * *
 Instructor     2    148944     74472     3.601        0.0338    *
 Residuals     56   1158261     20683
```

ANOVA with Blocks - Unbalanced design

Remark: we observed above that it is important to write the block factor first in the aov function in order to get the effect of variety adjusted for block. *If we change the order of the factors, the standard (balanced) solution changes.*
However, the Anova function is not affected by changing the order of the two factors in aov() function.

```
> my_anova <- aov(Sodium ~ Town + Instructor, data =
Data)
> summary(my_anova)
              Df     SumSq   MeanSq   Fvalue    Pr(> F)
 Instructor    2    290146   145073    7.014   0.001913   **
 Town          1    329551   329551   15.933   0.000193   ***
 Residuals    56   1158261    20683
```

ANOVA with Blocks - Unbalanced design

Next, we run the Tuckey HSD test to perform pairwise comparisons between the instructors

```
> TukeyHSD(my_anova, which = "Instructor")
Tukey multiple comparisons of means
95% family-wise confidence level

Fit:  aov(formula = Sodium ~ Instructor + Town, data =
Data)
```

$Instructor

	diff	*lwr*	*upr*	*padj*
$I2 - I1$	−41.25	−150.7432	68.24317	0.6381712
$I3 - I1$	−163.75	−273.2432	−54.25683	0.0019215
$I3 - I2$	−122.50	−231.9932	−13.00683	0.0248622

Repeated Measures ANOVA

A **Repeated Measures** experimental design is one in which measurements of the same variable are made on the same subject on two or more different occasions

For example, you might measure running speed before, one week into, and three weeks into a program of exercise. Because individuals would start with different running speeds, it is better to analyze using a two-way anova, with "individual" as one of the factors, rather than lumping everyone together and analyzing with a one-way anova.

Sometimes the repeated measures are repeated at different places rather than different times, such as the hip abduction angle measured on the right and left hip of individuals. Repeated measures experiments are often done without replication, although they could be done with replication.

Repeated Measures ANOVA

The simplest setting for repeated measures design is the one where, in addition to the treatment variable, one additional variable (a factor) is considered to isolate its contribution to the total variability among the observations. This additional factor is usually referred to as **subjects.** We refer to this setting as a

single-factor repeated measures design.

In a repeated measures design, one of main effects is usually uninteresting and the test of its null hypothesis may not be reported.

For examples, if the goal is to determine whether a particular exercise program affects running speed, there would be little point in testing whether subjects differed from each other in their average running speed; only the change in running speed over time would be of interest.

Repeated Measures ANOVA

Example A study examined diamondback rattlesnakes in a "rattlebox," a box with a lid that would slide open and shut every 5 minutes. At first, the snake would rattle its tail each time the box opened. After a while, the snake would become habituated to the box opening and stop rattling its tail. Researchers counted the number of box openings until a snake stopped rattling; fewer box openings means the snake was more quickly habituated. They repeated this experiment on each snake on four successive days, which we treat as a nominal variable for this example.

The measurement variable is trials to habituation, and the two nominal variables are day (1 to 4) and snake ID (D1,...,D6). This is a repeated measures design, as the measurement variable is measured repeatedly on each snake. It is analyzed using a two-way anova.

Repeated Measures ANOVA

```
> Input = ("
Day          Snake   Openings
1            D1        85
1            D2       107
1            D3        61
...          ...
2            D1        58
...          ...
3            D1        15
...          ...
4            D1        57
...          ...
4            D6        16
")
```

```
>Data = read.table(textConnection(Input),header=TRUE)
```

Repeated Measures ANOVA

```
> str(Data)
’data.frame’:  24 obs.  of 3 variables:
$ Day :  int 1 1 1 1 1 1 2 2 2 2 ...
$ Snake :  chr "D1" "D2" "D3" "D4" ...
$ Openings:  int 85 107 61 22 40 65 58 51 60 41 ...
```

We convert Day into a factor variable

```
> Data$Day = as.factor(Data$Day)
> Data$Snake =
factor(Data$Snake,levels=unique(Data$Snake))
> str(Data)
’data.frame’:  24 obs.  of 3 variables:
$ Day :  Factor w/ 4 levels "1","2","3","4":  1 1 1 1
1 1 2 2 2 2 ...
$ Snake :  Factor w/ 6 levels "D1","D2","D3",..:  1 2
3 4 5 6 1 2 3 4 ...  ...
$ Openings:  int 85 107 61 22 40 65 58 51 60 41 ...
```

ANOVA with Blocks - Unbalanced design

We use the function `Anova()` in the car package to compute **two-way ANOVA test for unbalanced design**.

```
library(car)
> my_anova <- aov(Openings ~ Day + Snake, data =
Data)
> Anova(my_anova, type = "II")
```

```
Response:  Openings
              SumSq   Df   Fvalue   Pr(> F)
 Day          4877.8   3   3.3201   0.04866  *
 Snake        3042.2   5   1.2424   0.33818
 Residuals    7346.0  15
```

Conclusion: there is a statistically significant difference among Days.

Repeated Measures ANOVA

Next, we run the Tuckey HSD test to perform pairwise comparisons between the Dayss

```
> TukeyHSD(my_anova, which = "Day")
Tukey multiple comparisons of means
95% family-wise confidence level

Fit:  aov(formula = Openings ~ Day + Snake, data = Data)
 $Day
              diff        lwr         upr       padj
 2 − 1  −16.333333  −53.15762   20.490954  0.5896276
 3 − 1  −28.833333  −65.65762    7.990954  0.1529601
 4 − 1  −38.000000  −74.82429   −1.175713  0.0420317
 3 − 2  −12.500000  −49.32429   24.324287  0.7636046
 4 − 2  −21.666667  −58.49095   15.157621  0.3596726
 4 − 3   −9.166667  −45.99095   27.657621  0.8886209
```

There is a statistically significant difference in the pair 4-1.

Nested ANOVA

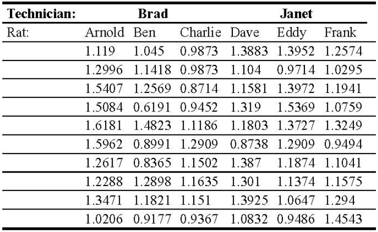
Nested ANOVA is used when you have one measurement variable and more than one nominal variable, and the nominal variables are nested, that is, they form subgroups within groups.
It tests whether there is significant variation in means among groups and among subgroups within group.
Nested analysis of variance is an extension of one-way ANOVA in which each group is divided into subgroups

Assumptions: like all ANOVA tests, it assumes that the observations within each subgroup are normally distributed and have equal standard deviations.

Nested ANOVA

Example. A laboratory studies uptake of fluorescently labeled protein in rat kidneys. We want to know whether the two technicians, Brad and Janet, are performing the procedure consistently. So Brad and Janet randomly chose 3 rats each, and each technician measured protein uptake in each rat.

If Brad and Janet had measured protein uptake only once on each rat, you would have one measurement variable (protein uptake) and one nominal variable (technician) and you would analyze it with one-way anova. However, rats are expensive and measurements are cheap, so Brad and Janet measured protein uptake at several random locations in the kidney of each rat:



Technician:	**Brad**			**Janet**		
Rat:	Arnold	Ben	Charlie	Dave	Eddy	Frank
	1.119	1.045	0.9873	1.3883	1.3952	1.2574
	1.2996	1.1418	0.9873	1.104	0.9714	1.0295
	1.5407	1.2569	0.8714	1.1581	1.3972	1.1941
	1.5084	0.6191	0.9452	1.319	1.5369	1.0759
	1.6181	1.4823	1.1186	1.1803	1.3727	1.3249
	1.5962	0.8991	1.2909	0.8738	1.2909	0.9494
	1.2617	0.8365	1.1502	1.387	1.1874	1.1041
	1.2288	1.2898	1.1635	1.301	1.1374	1.1575
	1.3471	1.1821	1.151	1.3925	1.0647	1.294
	1.0206	0.9177	0.9367	1.0832	0.9486	1.4543

Nested ANOVA

Because there are several observations per rat, the identity of each rat is a **nominal variable**. The values of this variable (the identities of the rats) are nested under the technicians; rat A is only found with Brad, and rat D is only found with Janet. In this case, it's a **two-level nested anova**; the technicians are groups, and the rats are subgroups within the groups. If the technicians had looked at several random locations in each kidney and measured protein uptake several times at each location, you'd have a three-level nested anova, with kidney location as subsubgroups within the rats.

Note: if the subgroups, subsubgroups, etc. are distinctions with some interest (fixed effects, variables), rather than random, you should not use a nested anova. For example, Brad and Janet could have looked at protein uptake in two male rats and two female rats apiece. In this case you would use a two-way anova to analyze the data, rather than a nested anova.

Nested ANOVA

```
Input = ("
Tech      Rat  Protein
Janet     1    1.119
Janet     1    1.2996
...       ...
Janet     2    1.045
...       ...
Janet     3    0.9367
Brad      4    1.3883
...       ...
Brad      5    1.3952
Brad      6    1.4543
")
```

```
> Data =
read.table(textConnection(Input),header=TRUE)
```

Nested ANOVA

```
> str(Data)

'data.frame':  60 obs.  of 3 variables:
$ Tech :  chr "Janet" "Janet" "Janet" "Janet" ...
$ Rat :  int 1 1 1 1 1 1 1 1 1 1 ...
$ Protein:  num 1.12 1.3 1.54 1.51 1.62 ...
```

We redefine Rat and Tech as factors.

```
> Data$Rat = factor(Data$Rat,levels=unique(Data$Rat))
> Data$Tech = factor(Data$Tech,levels=unique(Data$Tech))
> str(Data)
'data.frame':  60 obs.  of 3 variables:
$ Tech :  Factor w/ 2 levels "Janet","Brad":  1 1 1 1 1 1
1 1 1 1 ...
$ Rat :  Factor w/ 6 levels "1","2","3","4",..:  1 1 1 1 1
1 1 1 1 1 ...
$ Protein:  num 1.12 1.3 1.54 1.51 1.62 ...
```

Nested ANOVA

The aov function in R allows you to specify an error component to the model. We will use this error to manage the Rat factor.

```
> my_anova = aov(Protein ~ Tech + Error(Rat),
data=Data)
> summary(my_anova)
```

```
Error: Rat
             Df   SumSq    MeanSq    Fvalue   Pr(> F)
 Tech         1   0.0384   0.03841    0.268     0.632
 Residuals    4   0.5740   0.14349
Error: Within
             Df   SumSq    MeanSq    Fvalue   Pr(> F)
 Residuals   54    1.946   0.03604
```

Conclusion: the test found that there is no statistically significant difference between the technicians.

Nested ANOVA

If we want to test whether the technicians or the rats are a statistically significant factor in the measured protein intake, we can run a standard two-way ANOVA.

```
> my_anova2 = aov(Protein ~ Tech + Rat, data=Data)
> summary(my_anova2)
```

```
           Df  SumSq   MeanSq   Fvalue  Pr(> F)
Tech        1  0.0384  0.03841  1.066   0.30649
Rat         4  0.5740  0.14349  3.982   0.00666  **
Residuals  54  1.9460  0.03604
```

5 Linear Regression

Linear Regression

In some applications, we are interested in obtaining a simple model that explains the relationship between two or more variables.

For example, suppose that we are interested in studying the relationship between the income of parents and the income of their children in a certain country. Even though many factors can impact the income of a person, we conjecture that children from wealthier families tend to become wealthier when they grow up. Here, we consider two variables:

1. The family income x, defined as the average income of parents at a certain period.
2. The child income y, defined as his/her average income at a certain period (e.g, age).

Linear Regression

To examine the relationship between the two variables, we collect some data

$$(x_i, y_i), \qquad i = 1, \ldots, n$$

where y_i is the average income of the i-th child and x_i is the average income of his/her parents.
We are often interested in finding a simple model. A linear model is the simplest model that we can define:

$$y_i \approx \beta_1 x_i + \beta_0$$

As there are other factors that impact each child's future income, so we might write

$$y_i = \beta_1 x_i + \beta_0 + \epsilon_i$$

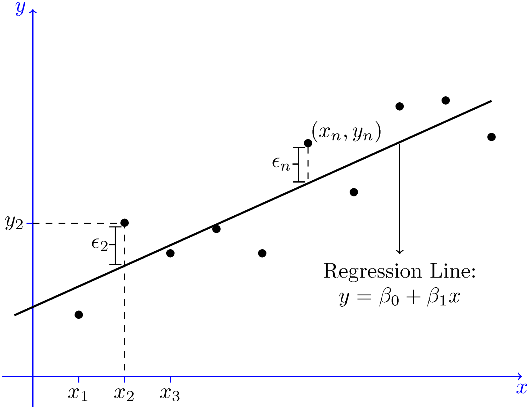
where ϵ_i is modeled as a random variable.

Linear Regression

Our goal is to obtain the values of β_0 and β_1 resulting in the smallest errors. That, we want to find the line in the $x - y$ plane

$$\hat{y}(x) = \beta_1 x + \beta_0$$

that 'best' fits our data points. Such line is the **regression line**.



Linear Regression Model

The **linear regression model** is

$$Y = \beta_1 X + \beta_0 + \epsilon$$

Since ϵ is a random variable, Y is also a random variable. The variable X is called the **predictor** or the **explanatory variable**, and the random variable Y is called the **response variable**.

We have data points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ and our goal is to find the ‘best’ values for β_0 and β_1 resulting in the line that provides the ‘best’ fit for the data points. The observed values of the random variables Y are

$$y_i = \beta_1 x_i + \beta_0 + \epsilon_i.$$

We model the terms ϵ_i’s as independent and zero-mean normal random variables, $\epsilon_i \sim N(0, \sigma^2)$.

Statistical method to find the regression line

We assume that the x_i's are observed values of a random variable X, so the model is

$$Y = \beta_1 X + \beta_0 + \epsilon,$$

where $\epsilon \sim N(0, \sigma^2)$.
By taking the expectation on both sides

$$E[Y] = \beta_1 E[X] + \beta_0 + E[\epsilon] = \beta_1 E[X] + \beta_0$$

Hence $\boxed{\beta_0 = E[Y] - \beta_1 E[X]}$
From the covariance, observing that X and ϵ are independent

$$\begin{aligned} cov(X, Y) &= cov(X, \beta_1 X + \beta_0 + \epsilon) \\ &= \beta_1\, cov(X, X) + \beta_0\, cov(X, 1) + cov(X, \epsilon) \\ &= \beta_1\, cov(X, X) = \beta_1 var(X) \end{aligned}$$

Hence $\boxed{\beta_1 = \dfrac{cov(X, Y)}{var(X)}}$

Statistical method to find the regression line

To compute β_0, β_1 from the observed pairs $(x_1, y_1), \ldots, (x_n, y_n)$, we introduce the notation

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i, s_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2, s_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

so we can estimate β_0, β_1 as

$$\boxed{\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}, \quad \hat{\beta}_1 = \frac{s_{xy}}{s_{xx}}}$$

and express the regression line as

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x.$$

For each $i = 1, \ldots, n$, the quantity $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ is the predicted value of y_i using the regression formula and the error in this prediction, called a **residual**, is

$$e_i = y_i - \hat{y}_i.$$

Computation of the regression line

Example. Consider the following observed values of (x_i, y_i):

$$(1,3),(2,4),(3,8),(4,9)$$

Find the estimated regression line based on the observed data.

Solution. We have

$$\bar{x} = \frac{1}{4}(1+2+3+4) = 2.5, \quad \bar{y} = \frac{1}{4}(3+4+8+9) = 6$$

$$s_{xx} = (1-2.5)^2 + (2-2.5)^2 + (3-2.5)^2 + (4-2.5)^2 = 5$$

$$s_{xy} = (1-2.5)(3-6) + (2-2.5)(4-6) + (3-2.5)(8-6) + (4-2.5)(9-6)$$

Hence

$$\hat{\beta}_1 = \frac{s_{xy}}{s_{xx}} = \frac{11}{5} = 2.2, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x} = 6 - (2.2)(2.5) = 0.5$$

and the regression line is

$$\hat{y} = 0.5 + 2.2x$$

Computation of the regression line

The fitted values are given by

$$\hat{y}_i = 0.5 + 2.2x_i$$

and the residuals are

$$\begin{aligned} e_1 &= y_1 - \hat{y}_1 = 3 - (0.5 + (2.2)(1)) = 0.3 \\ e_2 &= y_2 - \hat{y}_2 = 4 - (0.5 + (2.2)(2)) = -0.9 \\ e_3 &= y_3 - \hat{y}_3 = 8 - (0.5 + (2.2)(3)) = 0.9 \\ e_4 &= y_4 - \hat{y}_4 = 9 - (0.5 + (2.2)(4)) = -0.3 \end{aligned}$$

We have that $e_1 + e_2 + e_3 + e_4 = 0$

It is true in general that $\sum_i e_i = 0$.

Coefficient of Determination

Our linear regression model for regression is

$$Y = \hat{\beta}_0 + \hat{\beta}_1 X + \epsilon, \tag{1}$$

where $\epsilon \sim N(0, \sigma^2)$ is a random variable independent of X.
X is the only variable that we observe, so we estimate Y using X as

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

and the error in our estimate is

$$\epsilon = Y - \hat{Y}.$$

The randomness in Y comes from X and ϵ. From (1) we get

$$Var(Y) = \hat{\beta}_1^2 Var(X) + Var(\epsilon),$$

showing that the variation in Y is the sum of a part dependent on $Var(X)$ and a part which is the variance of the error ϵ.

Coefficient of Determination

If $Var(\epsilon)$ is small, then Y is close to $\hat{Y}$, so our regression model will be successful in estimating Y. Hence the quantity

$$\beta_1^2 \frac{Var(X)}{Var(Y)}$$

describes the portion of variance of Y that is explained by variation in X.
Using the formula for β_1, we observe that

$$\beta_1^2 \frac{Var(X)}{Var(Y)} = \frac{(cov(X,Y))^2}{(Var(X))^2} \frac{Var(X)}{Var(Y)} = \frac{(cov(X,Y))^2}{Var(X)\,Var(Y)} = \rho^2,$$

which is the **correlation coefficient**.
Interpretation: if X and Y are highly correlated ($|\rho|$ close to 1), then Y is well approximated by a linear function, that is, $Y \approx \hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$.

Coefficient of Determination

In practice, we only have access to the observed pairs (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n), so we estimate ρ^2 from the observed data. We define the **coefficient of determination** r^2 as

$$r^2 = \frac{\left(\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})\right)^2}{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2} = \frac{s_{xy}^2}{s_{xx}\, s_{yy}},$$

where

$$s_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2, \;\; s_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2, \;\; s_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

We have that $0 \leq r^2 \leq 1$ with larger values indicating that our linear model $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ is a good fit for the data.

$r = sqrt(r^2)$ is called the **Pearson correlation coefficient.**

Computation of the regression line and r^2

Example. Consider the following observed values of (x_i, y_i):

$$(1,3),(2,4),(3,8),(4,9)$$

We computed the estimated regression line $\hat{y} = 0.5 + 2.2\,x$.

We found that $s_{xx} = 5$ and $s_{xy} = 11$.
A similar calculation gives $s_{yy} = 26$.
Thus

$$r^2 = \frac{s_{xy}^2}{s_{xx}\,s_{yy}} = \frac{(11)^2}{(5)\,(26)} = 0.931$$

$$r = \sqrt{r^2} = 0.965$$

Correlation Coefficients

We can use R to compute the Pearson correlation coefficient.

```
> x=c(1,2,3,4)
> y=c(3,4,8,9)
> cor(x, y, method ="pearson")
[1] 0.9647638
```

Alternative correlation coefficients:

```
> cor(x, y, method ="spearman")
[1] 1
> cor(x, y, method ="kendall")
[1] 1
```

Method of Least Squares

There is a different method to estimate β_0 and β_1 in the equation of the regression line. This method will result in the same estimates as before; however, it is based on a different idea.

Suppose that we have data points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. Consider the model

$$\hat{y} = \beta_0 + \beta_1 x$$

The residuals are given by

$$e_i = y_i - \hat{y}_i = y_i - \beta_0 - \beta_1 x_i, \quad i = 1, \dots, n$$

and the **sum of the squared errors** is given by

$$E(\beta_0, \beta_1) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2.$$

To find the best fit for the data, we find the values of $\hat{\beta}_0, \hat{\beta}_1$ such that the error function $E(\beta_0, \beta_1)$ is minimized.

Method of Least Squares

To minimize the error function $E(\beta_0, \beta_1)$, we compute the partial derivatives with respect to β_0 and β_1, and set them to zero

$$\frac{\partial E}{\partial \beta_0} = \sum_{i=1}^{n} 2(-1)(y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\frac{\partial E}{\partial \beta_1} = \sum_{i=1}^{n} 2(-x_i)(y_i - \beta_0 - \beta_1 x_i) = 0$$

It follows that

$$\sum_{i=1}^{n} y_i = n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i, \quad \sum_{i=1}^{n} x_i y_i = \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2$$

By solving these equations for β_0, β_1, we obtain

$$\boxed{\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}, \quad \hat{\beta}_1 = \frac{s_{xy}}{s_{xx}}}$$

Linear Regression using R

The basic command in R to compute a regression model is `lm` which takes the variables in the format

```
lm([target variable] ~ [predictor variables], data =
                    [data source])
```

Data can be read from an Excel or csv file

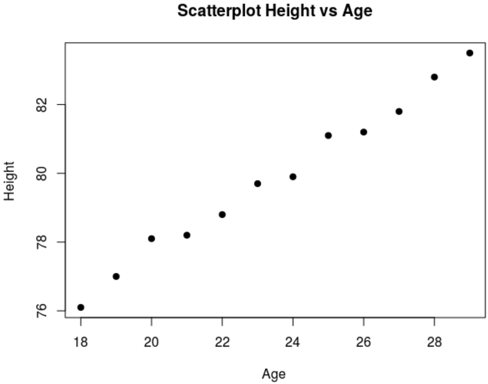
Example. We want to use a linear regression model to describe the relationship between the height of a child and its age. We collect the following measurements for children between the age of 18 and 29 months and the data are reported in the table below.

Age (months)	18	19	20	21	22	23	24	25	26	27	28	29
Height (cm)	76.1	77	78.1	78.2	78.8	79.7	79.9	81.1	81.2	81.8	82.8	83.5

Linear Regression using R

We start by loading the data and displaying them in a scatterplot

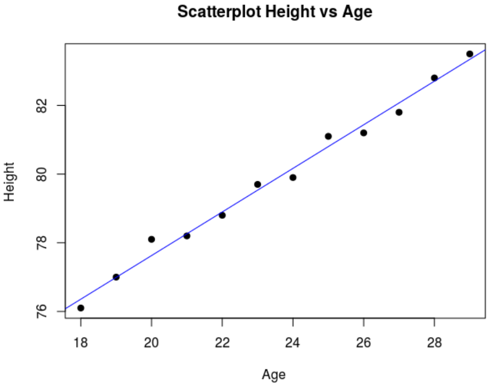
```
> age <-c(18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29)
> height <-c(76.1, 77.0, 78.1, 78.2, 78.8, 79.7, 79.9,
81.1, 81.2, 81.8, 82.8, 83.5)
> plot(age, height, main="Scatterplot Height vs Age",
xlab="Age ", ylab="Height ", pch=19)
```



Linear Regression using R

We next compute and display the regression line

```
> abline(lm(height ~ age), col = "blue")
```



Linear Regression using R

We can display the parameters of the regression equation with the following command line

```
> print(lm(height ~ age))
```

```
Call:
lm(formula = height ~ age)


Coefficients:
(Intercept)   age
64.928       0.635
```

Hence the regression model is

$$\hat{y} = 64.928 + 0.635\,x$$

Linear Regression using R

We can display full information of the linear regression model with the following command line

```
> summary(lm(height ~ age))

Call:
lm(formula = height ~ age)
Residuals:
Min           1Q     Median    3Q      Max
-0.27238 -0.24248 -0.02762 0.16014 0.47238

Coefficients:
.           Estimate   Std. Error t value Pr(>|t|)
(Intercept)  64.9283       0.5084  127.71   < 2e-16 ***
age           0.6350       0.0214   29.66   4.43e-11 ***

Residual standard error:  0.256 on 10 degrees of freedom
Multiple R-squared:  0.9888, Adjusted R-squared:  0.9876
F-statistic:  880 on 1 and 10 DF, p-value:  4.428e-11
```

Linear Regression using R

Comments about the R summary table.

Residuals. In the R summary of the lm function, you can see descriptive statistics about the residuals of the model. They are very informative about the quality of the fit of the model. The closewr to zero, the better the fit.

Coefficient of determination. There are two different R-squared, one multiple and one adjusted. The multiple is the Coefficient of determination that we discussed.
One problem with this coefficient is that it cannot decrease as you add more independent variables to your model, it will continue increasing as you make the model more complex, even if these variables don't add anything to your predictions. For this reason, the adjusted R-squared is useful if you are adding more than one variable to the model.
R-squared is about 0.99 in this example, showing that the model can explain 99% of the total variability.

Linear Regression using R

If data are stored in an Excel file, we can import them into R

```
> ageandheight <- read_csv("ageandheight.csv")
> lmHeight = lm(height ~ age, data = ageandheight)
> summary(lmHeight)

Call:
lm(formula = height ~ age, data = ageandheight)
Residuals:
Min          1Q     Median     3Q      Max
-0.27238 -0.24248 -0.02762 0.16014 0.47238

Coefficients:
.           Estimate   Std. Error t value Pr(>|t|)
(Intercept)   64.9283     0.5084   127.71   < 2e-16 ***
age           0.6350      0.0214    29.66   4.43e-11 ***

Residual standard error:  0.256 on 10 degrees of freedom
Multiple R-squared:  0.9888, Adjusted R-squared:  0.9876
F-statistic:  880 on 1 and 10 DF, p-value:  4.428e-11
```

Assessing Linear Regression

The coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$ of the linear regression are both calculated from the data and they are subject to error.

If the true model is

$$Y = \beta_0 + \beta_1 X,$$

then $\hat{\beta}_0$ and $\hat{\beta}_1$ are point estimators for the true coefficients.

We want to assess the uncertainty of the estimators.

For that, we will examine the sampling distribution of $\hat{\beta}_0$ and $\hat{\beta}_1$

Assessing Linear Regression

Proposition. Under the assumption that the error $\epsilon_i \sim N(0, \sigma^2)$, where σ is known, the sampling distribution of $\hat{\beta}_1$ and $\hat{\beta}_0$ is normal with

$$E[\hat{\beta}_1] = \beta_1 \quad \text{and} \quad var(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

and

$$E[\hat{\beta}_0] = \beta_0 \quad \text{and} \quad var(\hat{\beta}_0) = \sigma^2\left(\frac{1}{n} + \frac{(\bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right).$$

Hence the $(1-\alpha)100$ percent confidence interval of β_1 and β_0 are

$$\hat{\beta}_1 \pm z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}} \quad \text{and} \quad \hat{\beta}_0 \pm z_{\frac{\alpha}{2}} \sigma \sqrt{\frac{1}{n} + \frac{(\bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

Assessing Linear Regression

Proof.

Under the assumption that the error $\epsilon_i \sim N(0, \sigma^2)$, we have the responses Y_i are also normally distributed $Y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$.

We observe that $\hat{\beta}_1$ can be written as

$$\hat{\beta}_1 = \frac{s_{xy}}{s_{xx}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{\sum_{i=1}^n (x_i - \bar{x}) y_i}{\sum_{i=1}^n (x_i - \bar{x})^2}.$$

Hence, we write the estimator $\hat{\beta}_1$ as a linear combination of $Y_1, \ldots, Y_n$:

$$\hat{\beta}_1 = \sum_{i=1}^n c_i Y_i,$$

where $c_i = \frac{(x_i - \bar{x})}{\sum_{j=1}^n (x_j - \bar{x})^2}$ and we have the following observations:

$\sum_{i=1}^n c_i = 0$, since $\sum_{i=1}^n (x_i - \bar{x}) = \sum_{i=1}^n x_i - n\bar{x} = n\bar{x} - n\bar{x}$

$\sum_{i=1}^n c_i x_i = 1$, since $\frac{\sum_{i=1}^n (x_i - \bar{x}) x_i}{\sum_{j=1}^n (x_j - \bar{x})^2} = \frac{\sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})}{\sum_{j=1}^n (x_j - \bar{x})^2} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\sum_{j=1}^n (x_j - \bar{x})^2}$

Assessing Linear Regression

Using the above observations,

$$E[\hat{\beta}_1] = \sum_{i=1}^{n} c_i E[Y_i] = \sum_{i=1}^{n} c_i E[\beta_0 + \beta_1 x_i] = \beta_0 \sum_{i=1}^{n} c_i + \beta_1 \sum_{i=1}^{n} c_i x_i = \beta_1$$

For the variance:

$$var(\hat{\beta}_1) = \sum_{i=1}^{n} c_i^2 \sigma^2 = \sigma^2 \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{(\sum_{j=1}^{n} (x_j - \bar{x})^2)^2} = \frac{\sigma^2}{\sum_{j=1}^{n} (x_j - \bar{x})^2}$$

This shows that $\hat{\beta}_1 \sim N(\beta_1, \frac{\sigma^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2})$.

We next consider the estimator $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}$. We can write

$$\hat{\beta}_0 = \frac{1}{n} \sum_{i=1}^{n} Y_i - \sum_{i=1}^{n} c_i Y_i \bar{x} = \sum_{i=1}^{n} k_i Y_i,$$

showing that $\hat{\beta}_0$ is a linear combination of the random variables Y_i where $k_i = \frac{1}{n} - c_i \bar{x}$. The rest of the proof is left for exercise. □

Assessing Linear Regression

In the last Proposition, we have assumed that the variance of the noise σ^2 is known.
In practice, σ^2 is unknown and must be estimated from then data.
For that, we replace σ^2 with the mean squared error:

$$\hat{\sigma}^2 = MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}$$

so that we estimate the variance of $\hat{\beta}_1$ as

$$s^2(\hat{\beta}_1) = \frac{MSE}{s_{xx}} = \frac{1}{(n-2)} \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

The quantity

$$s(\hat{\beta}_1) = \sqrt{\frac{1}{(n-2)} \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}.$$

is called the **standard error** of the estimate β_1.

Assessing Linear Regression

Similarly, since σ^2 is unknown, we estimate the variance of $\hat{\beta}_0$ as

$$s^2(\hat{\beta}_0) = \hat{\sigma}^2\left(\frac{1}{n} + \frac{(\bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right) = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}(\frac{1}{n} + \frac{(\bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}).$$

The **standard error** of the estimate β_0 is

$$s(\hat{\beta}_0) = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}\left(\frac{1}{n} + \frac{(\bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)}$$

Replacing σ^2 with $\hat{\sigma}^2$ changes the sampling distribution so that $\hat{\beta}_0$ and $\hat{\beta}_1$ are associated with a **t-distribution with** $n-2$ **degrees of freedom**.

Assessing Linear Regression

Proposition. Under the assumption that the error $\epsilon_i \sim N(0, \sigma^2)$, where σ is unknown, the sampling distribution of $\hat{\beta}_1$ and $\hat{\beta}_0$ follow a t-distribution with

$$E[\hat{\beta}_1] = \beta_1 \quad \text{and} \quad var(\hat{\beta}_1) = \frac{\hat{\sigma}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}.$$

and

$$E[\hat{\beta}_0] = \beta_0 \quad \text{and} \quad var(\hat{\beta}_0) = \hat{\sigma}^2 \left(\frac{1}{n} + \frac{(\bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right).$$

The $(1 - \alpha)100$ percent **confidence interval of β_1 and β_0** are

$$\hat{\beta}_1 \pm t_{\frac{\alpha}{2}, n-2} \frac{\hat{\sigma}}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}} \quad \text{and} \quad \hat{\beta}_0 \pm t_{\frac{\alpha}{2}, n-2}\, \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(\bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

Assessing Linear Regression - Confidence interval

Example.

In the example about the relationship between height and age in children, we found that

$$\hat{\beta}_0 = 64.9283, \quad \hat{\beta}_1 = 0.6350$$

The table also reports the standard errors

$$s(\hat{\beta}_0) = 0.5084, \quad s(\hat{\beta}_1) = 0.0214$$

Hence the 95% confidence intervals of β_0 and β_1 are

$$\hat{\beta}_0 \pm t_{\frac{\alpha}{2},n-2}\, s(\hat{\beta}_0) = 64.9283 \pm (2.228)(0.5084) = [63.7956, 66.0610]$$

$$\hat{\beta}_1 \pm t_{\frac{\alpha}{2},n-2}\, s(\hat{\beta}_1) = 0.6350 \pm (2.228)(0.0214) = [0.5873, 0.6827]$$

(Note: I used $t_{0.025,10} = 2.228$)

Assessing Linear Regression - Hypothesis testing

We test hypotheses about the slope and intercept of the regression model by computing an appropriate test statistic. Below, we assume that σ^2 is unknown and that we the estimate value $\hat{\sigma}^2$.

- **Hypothesis testing for the slope** $\hat{\beta}_1$.

Test statistic: $W_1 = \frac{\hat{\beta}_1 - \beta_{1,0}}{s(\hat{\beta}_1)}$

- **Hypothesis testing for the intercept** $\hat{\beta}_0$.

Test statistic: $W_0 = \frac{\hat{\beta}_0 - \beta_{0,0}}{s(\hat{\beta}_0)}$

W_0, W_1 satisfy a t distribution with $n-2$ degrees of freedom.

Hypothesis testing is then carried out in the usual way.

1. Two-tailed test. Rejection region: $|W| > t_{\frac{\alpha}{2},n-2}$
2. Lower tailed test. Rejection region: $W < -t_{\alpha,n-2}$
3. Upper tailed test. Rejection region: $W > t_{\alpha,n-2}$

Assessing Linear Regression - Hypothesis testing

Example.
We examine again the relationship between height and age in children from the example above and consider the following test

- $H_0 : \beta_1 = 0$
- $H_1 : \beta_1 \neq 0$

We calculate the test statistic

$$W = \frac{\hat{\beta}_1}{s(\beta_1)} = \frac{0.6350}{0.0214} = 29.6729$$

and next apply a two-tailed test.
Since $W = 29.6729 > t_{0.025,10} = 2.228$, then we reject H_0 with significance level $\alpha = 0.05$.

The R table report that the p-value is $4.43 \cdot 10^{-11}$.

Assessing Linear Regression - Hypothesis testing

Example: house selling price and taxes

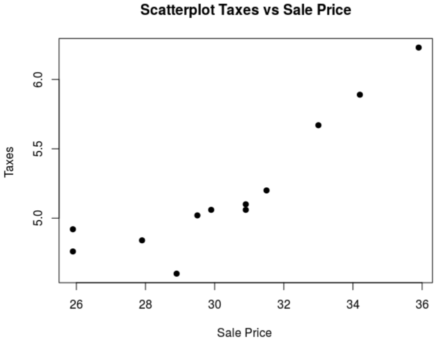
Sale price/k	25.9	29.5	27.9	25.9	29.9	29.9	30.9	28.9	35.9	31.5	31.0	30.9
Taxes/k	4.92	5.02	4.54	4.56	5.06	3.89	5.90	5.60	5.83	5.30	6.27	5.96

Independent variable X: Sale Price
Dependent variable Y: Taxes

Assessing Linear Regression - Hypothesis testing

We display a scatterplot

```
>sale <- c(25.9, 29.5, 27.9, 25.9, 29.9, 34.2, 30.9, 28.9,
35.9, 31.5, 33.0, 30.9)
>tax <- c(4.92, 5.02, 4.84, 4.76, 5.06, 5.89, 5.10, 4.60,
6.23, 5.20, 5.67, 5.06)
> plot(sale, tax, main="Scatterplot Taxes vs Sale Price",
xlab="Sale Price ", ylab="Taxes ", pch=19)
```



Assessing Linear Regression - Hypothesis testing

```
> summary(lm(tax ~ sale))

Call:
lm(formula = tax ~ sale)

Residuals:
Min       1Q      Median    3Q     Max
-0.3875 -0.1605 -0.0291 0.1619 0.3586

Coefficients:
.           Estimate   Std.Error   t value  Pr(>|t|)
(Intercept)   0.88259     0.68394    1.290    0.226
sale          0.14204     0.02242    6.336    8.51e-05 ***

Residual standard error:  0.2277 on 10 degrees of freedom
Multiple R-squared:  0.8006, Adjusted R-squared:  0.7806
F-statistic:  40.14 on 1 and 10 DF, p-value:  8.506e-05
```

Assessing Linear Regression - Hypothesis testing

We consider the following test

- $H_0 : \beta_1 = 0$
- $H_1 : \beta_1 \neq 0$

From the table: $\hat{\beta}_1 = 0.14204$, $s(\beta_1) = 0.02242$.
We calculate the test statistic

$$W = \frac{\hat{\beta}_1}{s(\beta_1)} = \frac{0.14204}{0.02242} = 6.3354$$

and next apply a two-tailed test. Note that df$=n-2=10$.

Since $W = 6.3354 > t_{0.025,10} = 2.228$, then we reject H_0 with significance level $\alpha = 0.05$.

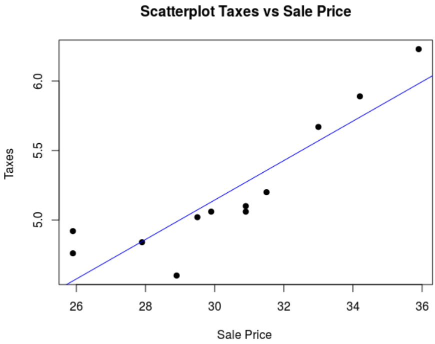
The R table report that the p-value is $8.506 \cdot 10^{-5}$.

Assessing Linear Regression - Confidence interval

The 95% confidence interval of β_1 is

$$\hat{\beta}_1 \pm t_{0.025,10}\, s(\beta_1) = 0.14204 \pm (2.228)(0.02242) = [0.09209, 0.19199]$$

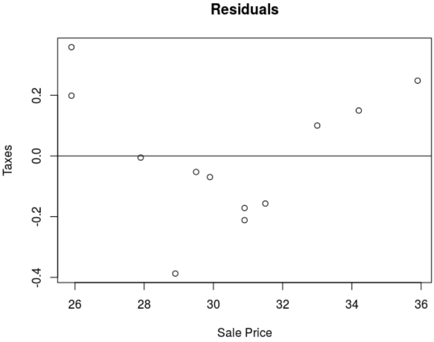
```
> abline(lm(tax ~ sale), col = "blue")
```



Assessing Linear Regression - Residuals

We can use R to analyze residuals $e_i = y_i - \hat{y}_i$

```
> housesale.lm = lm(tax ~ sale)
> housesale.res = resid(housesale.lm)
> plot(sale, housesale.res, xlab="Sale Price",
ylab="Taxes", main="Residuals")
> abline(0, 0) # the reference line
```

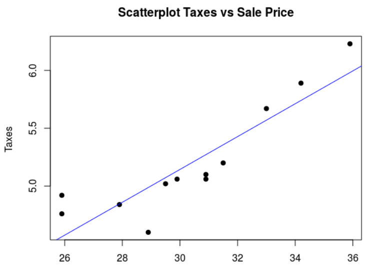


Assessing Linear Regression - Prediction Interval

For a given value of x, the interval estimate of the dependent variable y is called the **prediction interval**.

In this example, the prediction interval, is the expected value of the taxes, for a given sale price.

```
> housesale.lm = lm(tax ~ sale)
> newdata = data.frame(sale=40)
> predict(housesale.lm, newdata, interval="predict")
.     fit      lwr      upr
1 6.564139 5.849629 7.278649
```



Assessing Linear Regression - Correlation

The command `cor.test` is used to test for correlation between paired samples. It returns both the correlation coefficient and the p-value of the correlation.

```
> cor.test(sale, tax, method = "pearson")
```

```
Pearson's product-moment correlation

data: sale and tax
t = 6.3358, df = 10, p-value = 8.506e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.6594867 0.9703672
sample estimates:
cor
0.8947449
```

p-value is the same as the one reported by the `lm` command.

Multiple Linear Regression

In the above discussion, our model had only one predictor (explanatory variable), x.
We can consider models with more than one explanatory variable.

For example, suppose that we would like to have a model to predict house prices based on square footage, age, number of bedrooms, etc.
Here, the response variable y is the house price and our goal is to have a linear model

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon$$

where $x_1, x_2, \ldots, x_k$ are the explanatory variables (square footage, age, number of bedrooms, etc).

Such a model is a **multiple linear regression model**.

It is possible to extend the method of least squares to this case to compute estimates of $\beta_0, \beta_1, \ldots, \beta_k$.

Multiple Linear Regression

Estimates for the parameters $\beta_0, \beta_1, \ldots, \beta_k$ of the multiple regression equation

$$Y = \beta_0 + \beta_1 X_1 + \beta_1 2 X_2 + \cdots + \beta_k X_k + \epsilon$$

can be obtained using the method of **least squares**.
That is, we determine the coefficients β_i by minimizing the sum of the squared deviations of the observed values y_j of Y (= the sum of residuals).

Given the observations

$$y_j = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \cdots + \beta_k x_{kj} + \epsilon_j, \quad j = 1, \ldots, N,$$

the sum of residuals is written as

$$\sum_j \epsilon_j^2 = \sum_j \left(y_j - \beta_0 - \beta_1 x_{1j} - \beta_2 x_{2j} - \cdots - \beta_k x_{kj}\right)^2$$

Multiple Linear Regression

Using the notation

$$E(\beta_0, \ldots, \beta_k) = \sum_{j=1}^{N} \epsilon_j^2$$

the solution of the least squares problem for the multiple regression equation requires to solve the $k + 1$ equations

$$\frac{\partial E}{\partial \beta_i} = 0, \quad i = 0, 1, \ldots k.$$

Similarly to the linear case, this leads to the equations

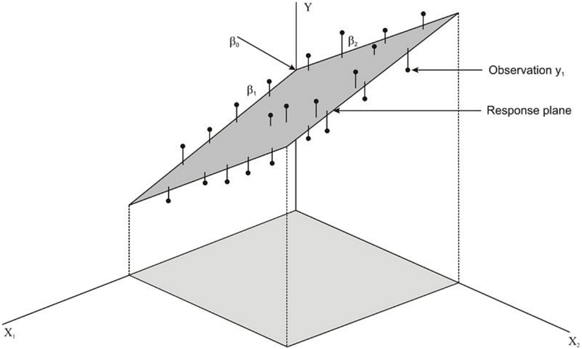
$$\sum_{j=1}^{N} (-1)(y_j - \beta_0 - \beta_1\, x_{1j} - \cdots - \beta_k\, x_{kj}) = 0$$

$$\sum_{j=1}^{N} (-x_{ij})(y_j - \beta_0 - \beta_1\, x_{1j} - \cdots - \beta_k\, x_{kj}) = 0, \quad i = 1, \ldots k.$$

Multiple Linear Regression

In the 3-variable case, for example, by minimizing the sum of the square deviations $\sum_j \epsilon_j^2$, we estimate the parameters $\beta_0, \beta_1, \beta_2$ determining the **regression plane**

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$



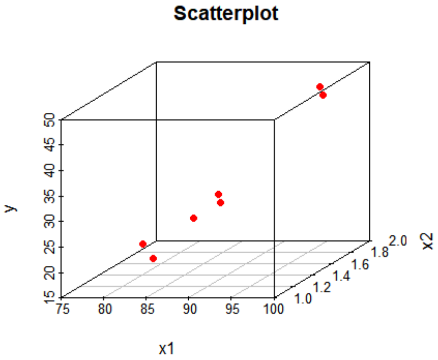
Multiple Linear Regression

Example: Suppose we want to predict the amount of water consumed by football players during practice. The football coach notices that the water consumption tends to be influenced by the time that the players are on the field and by the temperature. He measures the average water consumption, temperature, and practice time for seven practices and records the following data:

Temperature (F)	Practice time (h)	Water consumption (oz)
75	1.85	16
83	1.25	20
85	1.5	25
85	1.75	27
92	1.15	32
97	1.75	48
99	1.6	48

Multiple Linear Regression

```
x1 <- c(75,83,85,85,92,97,99)
x2 <- c(1.85,1.25,1.5,1.75,1.15,1.75,1.6)
y <- c(16,20,25,27,32,48,48)
library(scatterplot3d)
plot3d <- scatterplot3d(x1,x2,y,angle=45, scale.y=0.9,
+ pch=16, color ="red", main ="Scatterplot")
```



Multiple Linear Regression

```
dataset = cbind.data.frame(x1,x2,y)
relation <- lm(y ~ x1+x2, data = dataset)
print(relation)
```

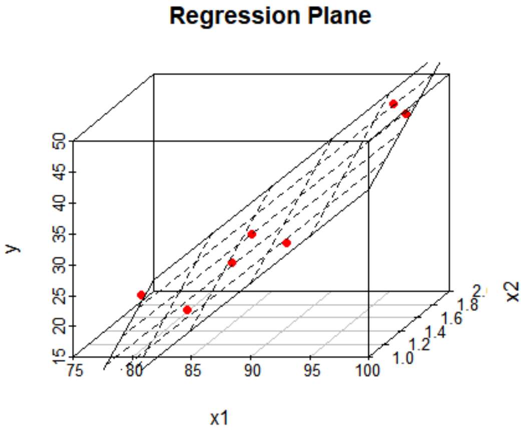
```
Call:
lm(formula = y ~ x1+x2, data = dataset)

Coefficients:
(Intercept)          x1           x2
-121.655          1.512       12.532
```

The multilinear regression solution is

$$\hat{y} = -121.655 + 1.512\, x_1 + 12.532\, x_2$$

Multiple Linear Regression

```
plot3d <- scatterplot3d(x1,x2,y,angle=55, scale.y=0.7,
+ pch=16, color ="red", main ="Regression Plane")
plot3d$plane3d(relation, lty.box = "solid")
```



Multiple Linear Regression

```
print(summary(relation))
Call:
lm(formula = y ~ x1 + x2, data = dataset)

Residuals:
1      2       3       4       5      6      7
1.0441 0.4642 -0.6935 -1.8264 0.1061 1.0252 -0.1197

Coefficients:
.            Estimate   Std. Error t value Pr(>|t|)
(Intercept) -121.65500    6.54035 -18.601 4.92e-05 ***
x1             1.51236    0.06077  24.886 1.55e-05 ***
x2            12.53168    1.93302   6.483  0.00292 **

Residual standard error: 1.245 on 4 degrees of freedom
Multiple R-squared: 0.9937, Adjusted R-squared: 0.9905
F-statistic: 313.2 on 2 and 4 DF, p-value: 4.027e-05
```

Multiple Linear Regression

Interpretation

The regression coefficients in the multilinear regression solution

$$\hat{y} = -121.655 + 1.512\,x_1 + 12.532\,x_2$$

tells us something about the relationship between the predictor variable and the predicted outcome.

The temperature coefficient of 1.51 tells us that for every degree increase in temperature, we predict there to be an increase of 1.512 ounces of water consumed, if we hold the practice time constant. Similarly, for every one-hour increase in practice time, we predict players will consume an additional 12.532 ounces of water, if we hold the temperature constant.

Based on the value of 0.9937 for the Multiple R-squared, we can conclude that approximately 99% of the variance in the outcome variable Y can be explained by the variance in the combined predictor variables.

Multiple Linear Regression

Also for multiple linear regression you can test the strength of the linear relationship between Y and the independent variables.

To test the null hypothesis that $\beta_i = \beta_{i0}$, $i = 1, \dots, k$, we compute the test statistic

$$t = \frac{\hat{\beta}_i - \beta_{i0}}{s_{\hat{\beta}_i}}$$

where the degrees of freedom is $n - k - 1$ and $s_{\hat{\beta}_i}$ is the standard deviation of $\hat{\beta}_i$.

In the example above, we may want to test

$$H_0 : \beta_1 = 0 \quad vs. \quad H_1 : \beta_1 \neq 0$$

using significance level $\alpha = 0.05$

Multiple Linear Regression

In the output table computed above using R, we have

$$t = \frac{\hat{\beta}_1}{s_{\hat{\beta}_1}} = \frac{1.51236}{0.06077} = 24.8866$$

Hence the p-value is

```
> 2*(1-pt(24.8866,df=4))= 1.547493e-05
```

showing that we can reject the null hypothesis.

A similar computation can be carried out to test the hypothesis

$$H_0 : \beta_2 = 0 \quad vs. \quad H_1 : \beta_2 \neq 0$$

Multiple Linear Regression

We can compute the confidence intervals of the parameters β_i as we did for the linear regression.

A $100(1-\alpha)\%$ confidence intervals of β_i is given by

$$\hat{\beta}_i \pm t_{\frac{\alpha}{2},n-k-1}\, s(\hat{\beta}_i)$$

For instance, in the exmaple above, a 95% confidence intervals of β_1 is

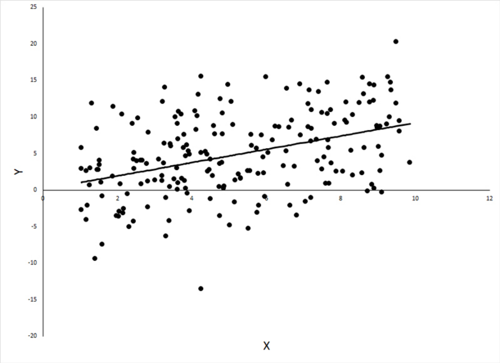
$$\hat{\beta}_1 \pm t_{0.025,4}\, s(\hat{\beta}_1) = 1.512 \pm (2.776)(0.061) = [1.343, 1.681]$$

Note $t_{0.025,4} = qt(1-0.05/2, 4) = 2.776445$

Non-Linear Regression

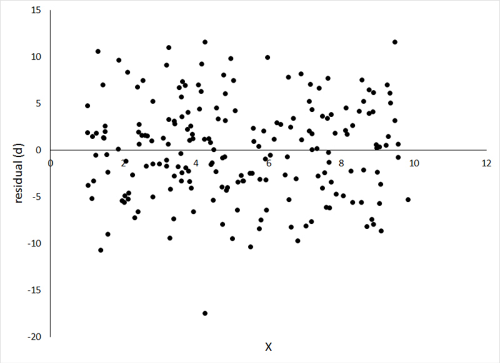
The linear regression model works under the assumption of a **linear** relationship between the independent (X) and dependent (Y) variables.

Below is a hypothetical, linear relationship between X and Y, showing the regression line passing through the observations:



Non-Linear Regression

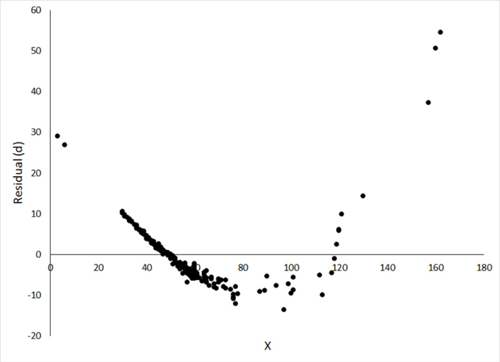
If we plot the residuals against X, we produce the residual plot:



The plot is what we expect to see, with points distributed along the horizontal line $Y = 0$ and the distribution of the points around the line remaining constant w.r. to X.

Non-Linear Regression

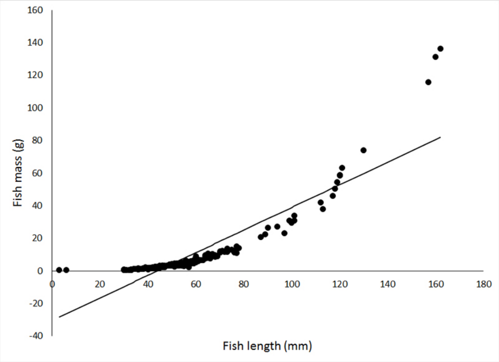
What if your residual plot looks like this?



The pattern of residuals seen above is the result of trying to fit a straight line to a non-linear relationship.

Non-Linear Regression

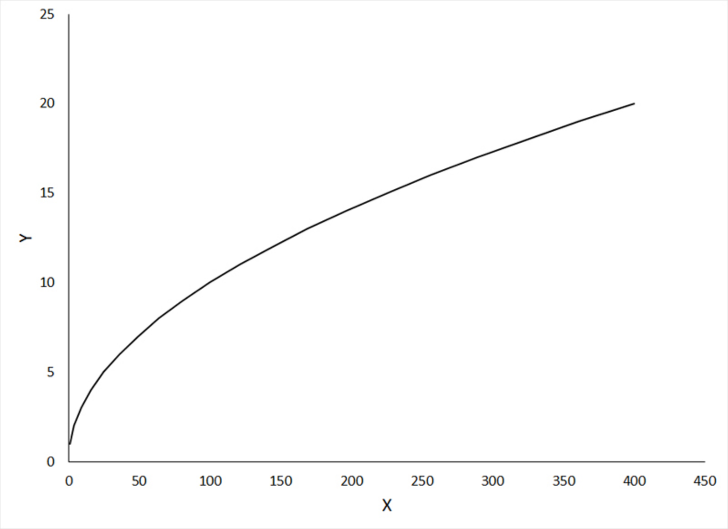
The data corresponding to the residual plot are plotted below. They show sunfish mass vs sunfish length.



It is visually clear that the regression line is not a good approximation of the data.

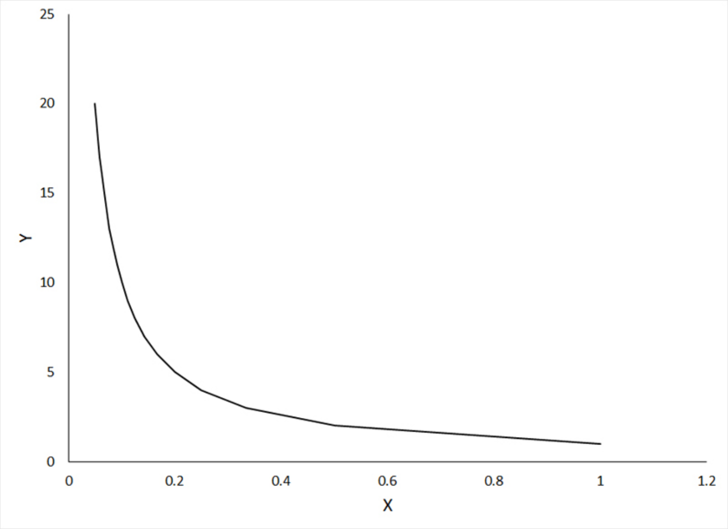
Non-Linear Regression

Data **transformations** can applied to transform a non-linear relationship Y vs X into a linear relationship.



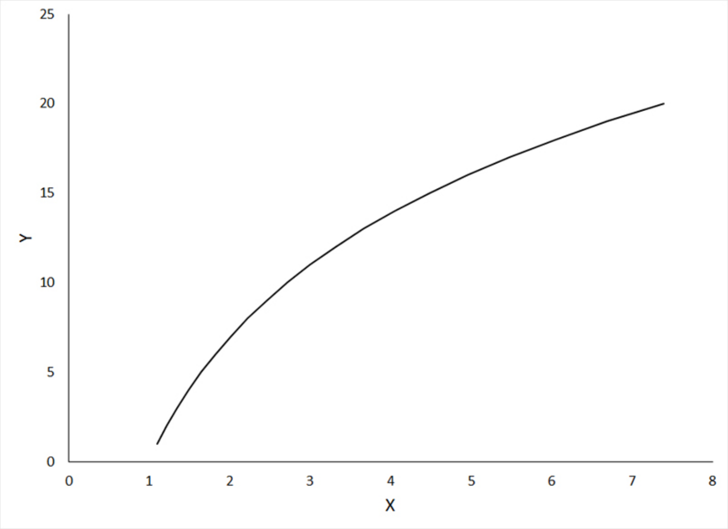
Non-Linear Regression

Data **transformations** can applied to transform a non-linear relationship Y vs X into a linear relationship.



Non-Linear Regression

Data **transformations** can applied to transform a non-linear relationship Y vs X into a linear relationship.

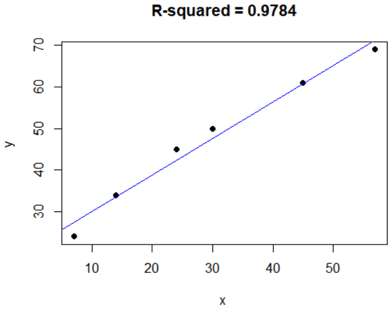


Non-Linear Regression

Example. Suppose we have the following data:

$$X = (7, 14, 24, 30, 45, 57)$$
$$Y = (24, 34, 45, 50, 61, 69)$$

```
> x <-c(7, 14, 24, 30, 45, 57)
> y <-c(24, 34, 45, 50, 61, 69)
> plot(x, y, main="R-squared = 0.9784", xlab="x ", ylab="y
", pch=19); abline(lm(y ~ x), col = "blue")
```

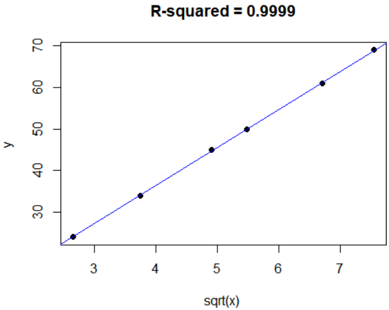


Non-Linear Regression

We now apply the transformation

$$x \mapsto \sqrt{x}$$

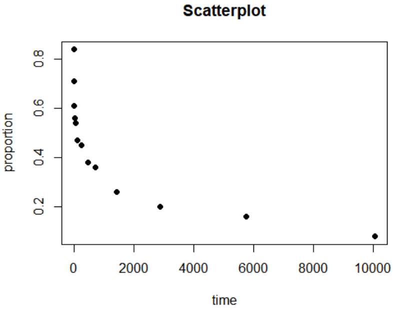
```
> x <-c(7, 14, 24, 30, 45, 57)
> y <-c(24, 34, 45, 50, 61, 69)
> plot(sqrt(x), y, main="R-squared = 0.9999",
xlab="sqrt(x) ", ylab="y ", pch=19);abline(lm(y ~
sqrt(x)), col = "blue")
```



Non-Linear Regression

Example. During a memory retention experiment, 13 subjects were asked to memorize a list of disconnected items. The subjects were then asked to recall the items at various times up to a week later. The proportion of items y correctly recalled at various times x (time in minutes) since the list was memorized were recorded.

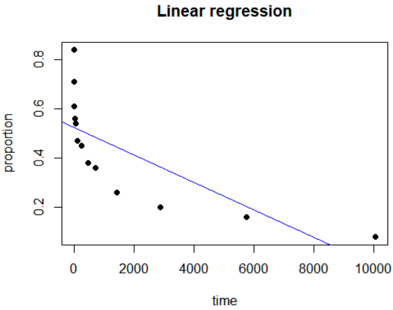
```
x <- c( 1, 5, 15, 30, 60, 120, 240, 480, 720, 1440, 2880, 5760, 10080)
y <- c(0.84, 0.71, 0.61, 0.56, 0.54, 0.47, 0.45, 0.38, 0.36, 0.26, 0.20, 0.16, 0.08)

plot(x, y, main="Scatterplot", xlab=" time ", ylab=" proportion ", pch=19)
```



Non-Linear Regression

Clearly, a linear regression model is not a satisfactory fit to the data

```
> plot(x, y, main="Linear regression", xlab=" time ",
ylab=" proportion ", pch=19)
> abline(lm(y ~ x), col = "blue")
> print(summary(lm(y ~ x)))
...
Multiple R-squared:  0.5709, Adjusted R-squared:  0.5318
```

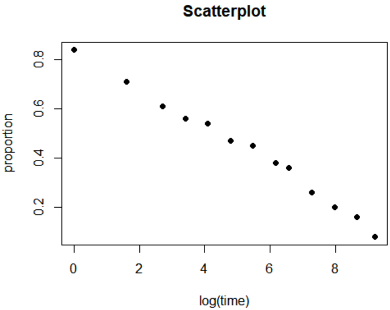


Non-Linear Regression

We now apply the transformation

$$x \mapsto \log x$$

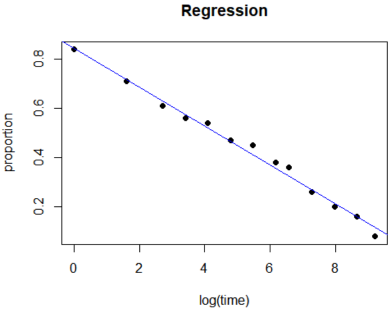
```
> plot(log(x), y, main="Linear regression", xlab="
log(time) ", ylab=" proportion ", pch=19)
```



Non-Linear Regression

The analysis below shows that this transformation is very effective

```
> plot(log(x), y, main="Linear regression", xlab="
log(time) ", ylab=" proportion ", pch=19)
> abline(lm(y ~ log(x)), col = "blue")
> print(summary(lm(y ~ log(x))))
...
Multiple R-squared:  0.9899, Adjusted R-squared:  0.989
```



Non-Linear Regression

R includes an environment for general curve fitting using the least squares method

```
nls {stats} R Documentation
Nonlinear Least Squares

Description

Determine the nonlinear (weighted) least-squares estimates
of the parameters of a nonlinear model.

Usage

nls(formula, data, start, control, algorithm, trace, subset,
weights, na.action, model,lower, upper, ...)
```

Non-Linear Regression

The Michaelis-Menten Kinetics model is a very popular kinetics model, used for modeling enzyme kinetics in biochemistry. The model describes the rate of enzymatic reactions by relating the reaction rate to the concentration of a substrate:

$$V = \frac{V_M S}{K + S}$$

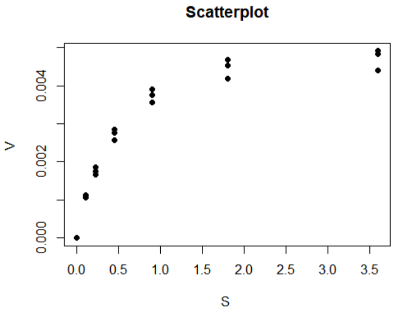
where

- V is the rate of the enzymatic reaction
- S is the concentration of the substrate
- V_M is the maximum rate achieved by the system
- K is the Michaelis coefficient

Non-Linear Regression

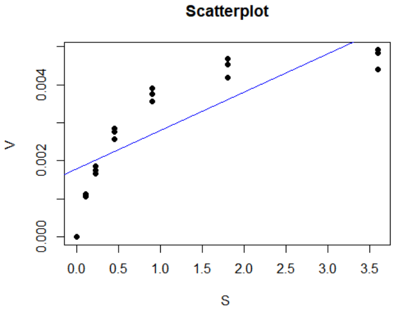
Here is a set of data that we would like to fit to the Michaelis Menten model:

```
> S <- c(3.6, 1.8, 0.9, 0.45, 0.225, 0.1125, 3.6, 1.8, 0.9, 0.45, 0.225, 0.1125, 3.6, 1.8, 0.9,
0.45, 0.225, 0.1125, 0)
> V <- c(0.004407692, 0.004192308, 0.003553846, 0.002576923, 0.001661538, 0.001064286,
0.004835714, 0.004671429, 0.0039, 0.002857143,0.00175, 0.001057143, 0.004907143,
0.004521429,0.00375, 0.002764286, 0.001857143, 0.001121429,0)
> plot(S, V, main="Scatterplot", xlab="S ", ylab="V ", pch=19)
```



Non-Linear Regression

Clearly, a linear regression model works poorly in this case



```
> summary(lm(y ~ x))

Call:
lm(formula = y ~ x)
.....
Multiple R-squared:  0.6799, Adjusted R-squared:  0.6611
```

Non-Linear Regression

```
> data = cbind.data.frame(S,V)
> mm.model.nls <- nls(V ~ Vm*S/(K+S), data=data, start =
list(K=max(data$V)/2, Vm=max(data$V)))
> summary(mm.model.nls)

Formula:  V ~ Vm * S/(K + S)

Parameters:
.       Estimate  Std. Error  t value Pr(>|t|)
K       0.4398016 0.0311612   14.11 8.1e-11 ***
Vm      0.0054252 0.0001193   45.47 < 2e-16 ***

Residual standard error:  0.0001727 on 17 degrees of
freedom

Number of iterations to convergence:  7
Achieved convergence tolerance:  4.666e-07
```

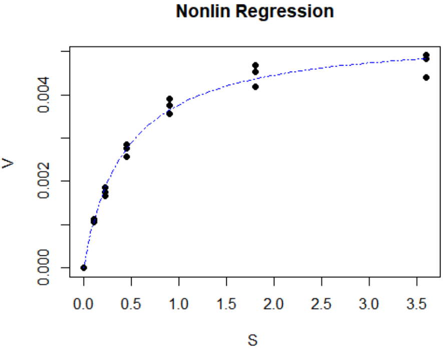
Non-Linear Regression

To calculating the R-squared value, we proceed as follows

```
> (RSS.p <- sum(residuals(m)^2)) # Residual sum of squares
[1] 5.071155e-07
> (TSS <- sum((V - mean(V))^2))
[1] 4.203714e-05
> 1 - (RSS.p/TSS)
[1] 0.9879365
```

Non-Linear Regression

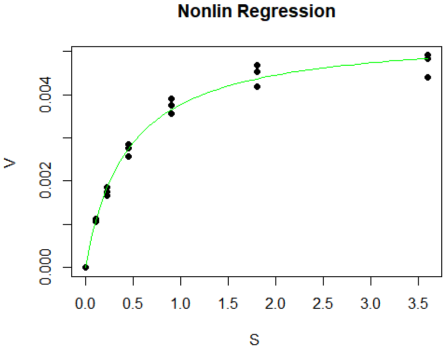
```
> plot(S, V, main="Nonlin Regression", xlab="S ", ylab="V
", pch=19)
> s <- seq(from = 0, to = 3.6, length = 50)
> K =coef(mm.model.nls)[1];Vm =coef(mm.model.nls)[2]
> lines(s,Vm*s/(K+s),lty=4,col = "blue")
```



Non-Linear Regression

One can use the predict function to plot the fitting curve directly from the model

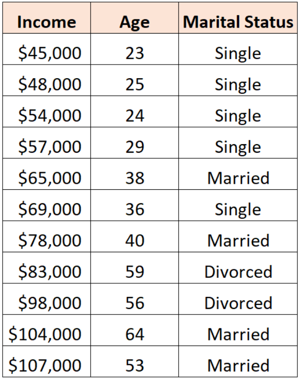
```
> plot(S, V, main="Nonlin Regression", xlab="S ", ylab="V
", pch=19)
> s <- seq(from = 0, to = 3.6, length = 50)
> lines(s, predict(mm.model.nls, list(S = s)), col =
"green")
```



Qualitative independent variables in Regression Models

It is common to use **dummy variables** as explanatory variables in regression models, if qualitative independent variables are likely to influence the outcome variable.

For instance, consider the following dataset where we would like to use age and marital status to predict income:



Income	Age	Marital Status
$45,000	23	Single
$48,000	25	Single
$54,000	24	Single
$57,000	29	Single
$65,000	38	Married
$69,000	36	Single
$78,000	40	Married
$83,000	59	Divorced
$98,000	56	Divorced
$104,000	64	Married
$107,000	53	Married

Dummy variables in Regression Models

To use marital status as a predictor variable in a regression model, we must convert it into a dummy variable.

Since the marital status is a categorical variable that can take on $k = 3$ different values ("Single", "Married", or "Divorced"), we need to create $k - 1 = 3 - 1 = 2$ dummy variables. Hence we use the model:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i}$$

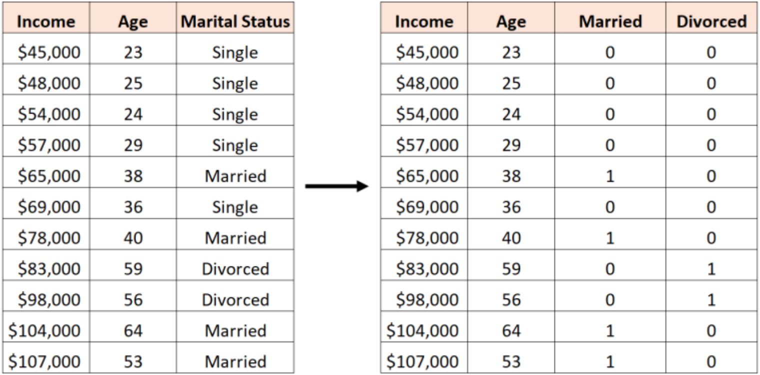
where x_{1i} is the non-categorical variable (Income) and

- $x_{2i} = 1$ if "Married", $x_{2i} = 0$ if not "Married"
- $x_{3i} = 1$ if "Divorced", $x_{13} = 0$ if not "Divorced"

Note that:
the configuration $x_{2i} = 0$, $x_{3i} = 0$ corresponds to "Single"

Dummy variables in Regression Models

This is the transformation of the data table:



Income	Age	Marital Status
$45,000	23	Single
$48,000	25	Single
$54,000	24	Single
$57,000	29	Single
$65,000	38	Married
$69,000	36	Single
$78,000	40	Married
$83,000	59	Divorced
$98,000	56	Divorced
$104,000	64	Married
$107,000	53	Married

Income	Age	Married	Divorced
$45,000	23	0	0
$48,000	25	0	0
$54,000	24	0	0
$57,000	29	0	0
$65,000	38	1	0
$69,000	36	0	0
$78,000	40	1	0
$83,000	59	0	1
$98,000	56	0	1
$104,000	64	1	0
$107,000	53	1	0

Dummy variables in Regression Models

Here is the analysis of the problem using R

```
> df <- data.frame(income=c(45000, 48000, 54000,
57000, 65000, 69000, 78000, 83000, 98000, 104000,
107000), age=c(23, 25, 24, 29, 38, 36, 40, 59, 56,
64, 53),status=c(’Single’, ’Single’, ’Single’,
’Single’,’Married’, ’Single’, ’Married’,
’Divorced’,’Divorced’, ’Married’, ’Married’))
```

Create dummy variables:

```
> married <- ifelse(df$status == ’Married’, 1, 0)
> divorced <- ifelse(df$status == ’Divorced’, 1, 0)
```

Create data frame to use for regression

```
> df_reg <- data.frame(income = df$income, age =
df$age, married = married, divorced = divorced)
```

Dummy variables in Regression Models

```
> model <- lm(income ~ age + married + divorced, data
= df_reg)
> summary(model)
```

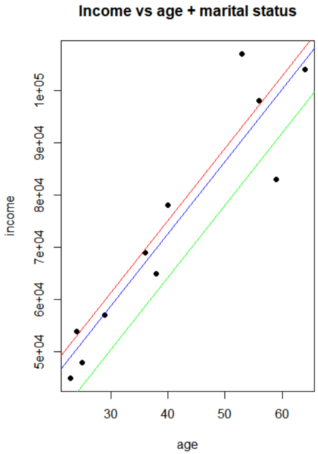
```
Call:
lm(formula = income ~ age + married + divorced, data = df_reg)

Coefficients:
             Estimate  Std.Error   tvalue  Pr(> |t|)
(Intercept)   14276.1    10411.5    1.371    0.21266
age            1471.7      354.4    4.152    0.00428   **
married        2479.7     9431.3    0.263    0.80018
divorced      −8397.4    12771.4   −0.658    0.53187
```

The solution of the regression problem is:

income = 14276.1+1471.7*(age)+2479.7*(married)−8397.4*(divorced)

Dummy variables in Regression Models

```
> plot(age,income, main="Income vs age + marital status",
xlab="age", ylab="income", pch=19)
> abline(lm(income~age,data=df_reg),col="blue")
> abline(lm(income+2479.7~age,data=df_reg),col="red")
> abline(lm(income-8397.4~age,data=df_reg),col="green")
```



Dummy variables in Regression Models

Interpretation:

- Age: Each one year increase in age is associated with an average increase of \$1,471.70 in income. Since the p-value (.004) is less than .05, age is a statistically significant predictor of income.
- Married: A married individual, on average, earns \$2,479.70 more than a single individual. Since the p-value (0.800) is not less than .05, this difference is not statistically significant.
- Divorced: A divorced individual, on average, earns \$8,397.40 less than a single individual. Since the p-value (0.532) is not less than .05, this difference is not statistically significant.
- Since both dummy variables were not statistically significant, we could drop marital status as a predictor from the model.

Dummy variables in Regression Models

It is possible to include the effect of interaction of the categorical variables with the non-categorical ones.

In this case, the regression model becomes

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \beta_4 x_{1i} x_{2i} + \beta_5 x_{1i} x_{3i}$$

Note that we do not include the cross term $x_{2i} x_{3i}$ since this product is always zero

Observe that the contribution due to the interaction will affect also the slope, not only the intercept

Dummy variables in Regression Models

Here is the modified R script

```
> model <- lm(income ~ age + married + divorced +
age:married + age:divorced, data = df_reg)
> summary(model)
```

Call:
lm(formula = income ~ age + married + divorced + age:married + age:divorced, data = df_reg)

Coefficients:

```
                 Estimate   Std.Error    tvalue   Pr(> |t|)
 (Intercept)       9143.1     20256.1     0.451     0.67066
 age               1659.0       728.4     2.278     0.0717.
 married           6741.1     27343.8     0.247      0.8151
 divorced        368856.9    211102.8     1.747      0.1410
 age : married     −169.5       816.2    −0.208      0.8437
 age : divorced   −6659.0      3725.1    −1.788      0.1339
```

Logistic regression

Linear regression is not an appropriate model if the predictor variable Y is a dichotomous variable.

The **logistic regression** is a method used for fitting a regression curve

$$y = f(x)$$

when y is a categorical variable.

The typical use of this model is predicting y given a set of predictors x.

The predictors can be continuous, categorical or a mix of both.

Logistic regression

Consider a predictor variable Y taking values in the set $\{0, 1\}$

If we let $p = P(Y = 1)$ then

- the quantity $\frac{p}{1-p}$ can take values on $[0, \infty)$
- the quantity $\ln \frac{p}{1-p}$ can take values on $(-\infty, \infty)$

Thus we have the **logistic regression model**

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x$$

which can be also written as

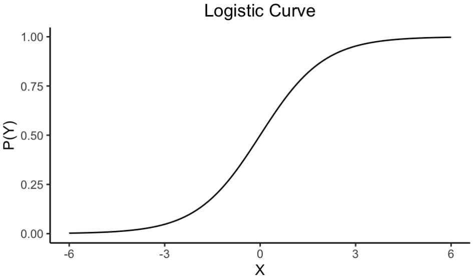
$$p = \frac{\exp(\beta_0 + \beta_1 x)}{1 + \exp(\beta_0 + \beta_1 x)}$$

where the right hand side is a sigmoid function. The transformation $p \rightarrow \ln \frac{p}{1-p}$ is called the **logit transformation**

Logistic regression

With a simple manipulation

$$P(Y) = p = \frac{\exp(\beta_0 + \beta_1 x)}{1 + \exp(\beta_0 + \beta_1 x)} = \frac{1}{1 + \exp(-\beta_0 - \beta_1 x)}$$



To solve the logistic regression problem, we do not use least squares minimization but *maximum likelihood estimation* with the assumption that $Y_i \sim binom(1, p = \frac{\exp()}{1+\exp()})$

Logistic regression

Let $\ln\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$
Hence $\ln\left(\frac{p(x+1)}{1-p(x+1)}\right) = \beta_0 + \beta_1(x+1)$ and the **log odds ratio** is

$$\beta_1 = \ln\left(\frac{p(x+1)}{1-p(x+1)}\right) - \ln\left(\frac{p(x)}{1-p(x)}\right) = \ln\left(\frac{\frac{p(x+1)}{1-p(x+1)}}{\frac{p(x)}{1-p(x)}}\right)$$

The **odds ratio** is

$$e^{\beta_1} = \frac{\frac{p(x+1)}{1-p(x+1)}}{\frac{p(x)}{1-p(x)}}$$

Interpretation: for an increase of 1 of the explanatory variable x, the odds increases by a factor of e^{β_1}

Logistic regression

Note about terminology.

In probability, given a probability p, we define

$$\textbf{odds for success} = \frac{p}{1-p} = \frac{\text{probability of success}}{\text{probability of failure}}$$

Given two probabilities p_1 and p_2 (not necessarily adding up to 1, we use the following terminology

- **relative risk** $= \frac{p_1}{p_2}$
- **odds ratio** $= \frac{\frac{p_1}{1-p_1}}{\frac{p_2}{1-p_2}} = \left(\frac{p_1}{p_2}\right) \frac{1-p_2}{1-p_1}$

Interpretation: the odds ratio measures how much greater or smaller the odda are for a subject possessing a risk factor to experience a particular outcome.

Note that relative risk and odds ratio are very close if p_1 and p_2 are both small (e.g., probability of diseases).

Logistic regression

Solution of logistic regression using R

Example. The in-built data set `mtcars` in R describes different models of a car with their various engine specifications. In this data set, the transmission mode (automatic or manual) is described by the column `am` which is a binary value (0 or 1). We can create a logistic regression model between the columns ”am” and the other columns ”hp”, ”wt”, ”cyl”, associated with horse power, weight and cylinders, respectively.

```
> input <- mtcars[,c("am","cyl","hp","wt")]
> input
                   am   cyl    hp     wt
 MazdaRX4           1     6   110  2.620
 MazdaRX4Wag        1     6   110  2.875
 Datsun710          1     4    93  2.320
 Hornet4Drive       0     6   110  3.215
 ...              ...   ...   ...
```

Logistic regression

We start by considering a simple logistic regression model where we examine the effect of the weight `wt` on the transmission mode `am`. We use the `glm` function to create the logistic regression model.

```
> am.data = glm(formula = am ~ wt, data = input, family =
binomial)
> print(summary(am.data))
```

```
Call:
glm(formula = am ~ wt, family = binomial, data = input)

Coefficients:
               Estimate   Std.Error   zvalue   Pr(>|z|)
 (Intercept)     12.040       4.510    2.670    0.00759   **
 wt              -4.024       1.436   -2.801    0.00509   **
```

Since the p-value in the last column for the variables "wt" is less than 0.05, the weight "wt" has a significant impacts on the "am" value in this regression model.

Logistic regression

The logistic regression is

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x = 12.040 - 4.024x$$

Hypothesis testing shows that we can reject the null hypothesis that $\beta_1 = 0$ since the p-value is less than 0.05.

We can also exponentiate the coefficients to compute the odds-ratio

```
> exp(coef(am.data))
 (Intercept)                    wt
 1.694596e + 05   1.788183e − 02
```

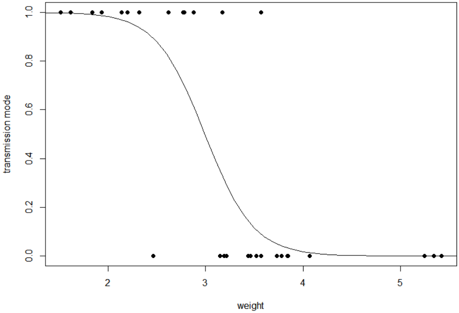
Hence the odds ratio is

$$e^{\beta_1} = 1.788183e - 02$$

Logistic regression

To display the logistic regression, we proceed as follows

```
> wt <-input$wt
> am <-input$am
> plot(wt, am, pch = 16, xlab = "weight", ylab =
"transmission mode")
> xwt <- seq(1, 6, 0.1)
> yam <- predict(am.data, list(wt = xwt),type="response")
> lines(xwt, yam)
```



Logistic regression

We can compute the 95% confidence interval of β_0 and β_1 as follows

```
> confint(am.data, level=0.95)
                    2.5%       97.5%
 (Intercept)    5.213795   23.628911
 wt            -7.698930   -1.833365
```

Similarly, we can compute the 95% confidence interval of e^{β_0} and e^{β_1} as follows

```
> exp(confint(am.data, level=0.95))
                        2.5%             97.5%
 (Intercept)  1.837902e + 02   1.827703e + 10
 wt           4.533119e - 04   1.598747e - 01
```

Logistic regression

We can run the `anova` function on the model to analyze the table of deviance to assess the goodness of fit.

The difference between the null deviance and the residual deviance shows how our model is doing against the null model (a model with only the intercept). The wider this gap, the better.

```
> anova(am.data, test="Chisq")
```

Analysis of Deviance Table
Model: binomial, link: logit
Response: am
Terms added sequentially (first to last)

	Df	*DevianceResid.*	*Df*	*Resid.Dev*	*Pr(> Chi)*	
NULL			31	43.230		
wt	1	24.054	30	19.176	$9.369e-07$	$***$

Logistic regression

While there is no exact equivalent to the coefficient of determination R2 of linear regression, there are some pseudo R squared measures playing a similar role for the logistic regression.

```
> library(pscl)
> pR2(am.data)
fitting null model for pseudo-r2
       llh     llhNull        G2   McFadden       r2ML       r2CU
 -9.588042  -21.614866 24.053648   0.556414   0.528424   0.713123
```

- llh: The log-likelihood from the fitted model
- llhNull: The log-likelihood from the intercept-only restricted model
- G2: Minus two times the difference in the log-likelihoods
- McFadden: McFadden's pseudo r-squared
- r2ML: Maximum likelihood pseudo r-squared
- r2CU: Cragg and Uhler's pseudo r-squared

Multiple Logistic regression

In the example above, we have considered a simple logistic regression model where the weight `wt` is the only the effect on the transmission mode `am`. We can modified the model to also consider the effect of the variables `cyl` and `hp`.

The multiple logistic equation is the multivariate equation

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \ldots \beta_k x_k$$

where there are multiple predictor variables $x_1, \ldots, x_k$
We have

$$p = \frac{\exp(\beta_0 + \beta_1 x_1 + \ldots \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \ldots \beta_k x_k)} = \frac{1}{1 + \exp(-\beta_0 - \beta_1 x_1 - \ldots \beta_k x_k)}$$

Multiple Logistic regression

The solution of the multiple logistic regression In R requires a relatively simple modification in the use of the command `glm`

Let us re-examine the example where are trying to predict the transmission mode `am` of a car.
We now modify the regression model to consider not only the effect of the weight `wt` but also the effect of the variables `cyl` and `hp`.

Multiple Logistic regression

We use the `glm` function to create the multiple logistic regression model.

```
> am.data = glm(formula = am ~ cyl + hp + wt, data =
input, family = binomial)
> print(summary(am.data))
```

```
Call:
glm(formula = am ~ cyl + hp + wt, family = binomial, data = input)
Coefficients:
               Estimate   Std.Error   zvalue   Pr(> |z|)
 (Intercept)   19.70288     8.11637    2.428      0.0152   *
 cyl            0.48760     1.07162    0.455      0.6491
 hp             0.03259     0.01886    1.728      0.0840   .
 wt            -9.14947     4.15332   -2.203      0.0276   *
```

Again, the first column reports the values of the logistic regression coefficients; the other columns contain information about the goodness of the fit

Logistic regression

The logistic regression is

$$\ln\left(\frac{p}{1-p}\right) = 19.703 + 0.488 \text{ cyl} + 0.033 \text{ hp} - 9.149 \text{ wt}$$

From the table, the solution of the hypothesis testing problem $H_0 : \beta_i = 0$ vs $H_0 : \beta_i \neq 0$ shows that we can reject H_0 for β_3 since the corresponding p-value is less than 0.05 but not for β_1 and β_2.

We can exponentiate the coefficients to compute the odds-ratios

```
> exp(coef(am.data))
 (Intercept)            cyl            hp             wt
 3.604568e+08   1.628400e+00   1.033129e+00   1.062760e-04
```

Hence the odds ratios are

$$e^{\beta_1} = 1.628400e+00,\ e^{\beta_2} = 1.033129e+00,\ e^{\beta_3} = 1.062760e-04$$

Logistic regression

We can compute the 95% confidence interval of β_0, β_1, β_2, and β_3 as follows

```
> confint(am.data, level=0.95)
```

```
                     2.5%        97.5%
(Intercept)    8.555361001  44.25163736
cyl           −1.532997981   3.12047408
hp             0.003320913   0.08838958
wt           −21.363156479  −3.48150672
```

We can also compute the pseudo R2 `> pR2(am.data)`

fitting null model for pseudo-r2

```
       llh        llhNull          G2    McFadden        r2ML        r2CU
−4.9207458    −21.6148666  33.3882416   0.7723444   0.6477389   0.8741409
```

Multiple Logistic regression

Example. A researcher is interested in how GRE (Graduate Record Exam scores), GPA (grade point average) and prestige of the undergraduate institution, affect admission into graduate school. The response variable, admit/don't admit, is a binary variable.

```
> mydata <-
read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
## view the first 6 rows of the data
> head(mydata)
```

	admit	*gre*	*gpa*	*rank*
1	0	380	3.61	3
2	1	660	3.67	3
3	1	800	4.00	1
4	1	640	3.19	4
5	0	520	2.93	4
6	1	760	3.00	2

Multiple Logistic regression

This dataset has a binary response (outcome, dependent) variable called `admit` taking values in $\{0, 1\}$.

There are three predictor variables: `gre`, `gpa` and `rank`.

We will treat the variables `gre` and `gpa` as continuous.

The variable `rank` is a categorical variable which takes on the values 1 through 4. Institutions with a rank of 1 have the highest prestige, while those with a rank of 4 have the lowest.

Multiple Logistic regression

```
> str(mydata)
'data.frame':  400 obs.  of 4 variables:
$ admit:  int 0 1 1 1 0 1 1 0 1 0 ...
$ gre :  int 380 660 800 640 520 760 560 400 540 700 ...
$ gpa :  num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 ...
$ rank :  int 3 3 1 4 4 2 1 2 3 2 ...
```

First, we convert rank to a factor to indicate that rank should be treated as a categorical variable.

```
> mydata$rank <- factor(mydata$rank)
> str(mydata)
'data.frame':  400 obs.  of 4 variables:
$ admit:  int 0 1 1 1 0 1 1 0 1 0 ...
$ gre :  int 380 660 800 640 520 760 560 400 540 700 ...
$ gpa :  num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 ...
$ rank :  Factor w/ 4 levels "1","2","3","4":  3 3 1 4 4 2
1 2 3 2 ...
```

Multiple Logistic regression

```
> mylogit <- glm(admit ~ gre + gpa + rank, data = mydata,
family = "binomial")
> summary(mylogit)

Call:
glm(formula = admit ~ gre + gpa + rank, family =
"binomial", data = mydata)
Coefficients:
                Estimate   Std.Error   zvalue   Pr(> |z|)
 (Intercept)   −3.989979    1.139951   −3.500    0.000465   * * *
 gre            0.002264    0.001094    2.070    0.038465   *
 gpa            0.804038    0.331819    2.423    0.015388   *
 rank2         −0.675443    0.316490   −2.134    0.032829   *
 rank3         −1.340204    0.345306   −3.881    0.000104   * * *
 rank4         −1.551464    0.417832   −3.713    0.000205   * * *
```

Multiple Logistic regression

Interpretation:

According to the table, both `gre` and `gpa` are statistically significant, as are the three terms for `rank`.
The logistic regression coefficients give the change in the log odds of the outcome for a one unit increase in the predictor variable.

- For every one unit change in `gre`, the log odds of admission (versus non-admission) increases by 0.002.
- For a one unit increase in `gpa`, the log odds of being admitted to graduate school increases by 0.804.
- The indicator variables for `rank` have a different interpretation. For example, having attended an undergraduate institution with rank of 2, versus an institution with a rank of 1, changes the log odds of admission by -0.675.

Multiple Logistic regression

You can also exponentiate the coefficients and interpret them as odds-ratios.

```
> exp(coef(mylogit))
 (Intercept)          gre          gpa        rank2        rank3        rank4
  0.0185001    1.0022670    2.2345448    0.5089310    0.2617923    0.2119375
```

Now we can say that for a one unit increase in `gre`, the odds of being admitted to graduate school (versus not being admitted) increase by a factor of 1.00; that for a one unit increase in `gpa`, the odds of being admitted to graduate school (versus not being admitted) increase by a factor of 2.23.

Poisson regression

Poisson regression is used to model count variables.

As for the logistic model, we develop a model for count data using a link function.

A Poisson regression model is given by the function

$$y = \exp(\beta_0 + \beta_1 x + + \cdots + \beta_k x_k)$$

Equivalently

$$\ln y = \beta_0 + \beta_1 x + + \cdots + \beta_k x_k$$

Poisson regression

Example. We want to predict the number of awards earned by students at one high school using as predictors the type of program in which the student was enrolled (e.g., vocational, general or academic) and the score on their final exam in math.

In this example, `num_awards` is the outcome variable and indicates the number of awards earned by students at a high school in a year, `math` is a continuous predictor variable and represents students' scores on their math final exam, and `prog` is a categorical predictor variable with three levels indicating the type of program in which the students were enrolled. It is coded as 1 = "General", 2 = "Academic" and 3 = "Vocational".

Poisson regression

```
> mydata <-
read.csv("https://stats.idre.ucla.edu/stat/data/poisson_sim.csv"
```

```
> head(mydata)
 id   num_awards   prog   math
 1           45     0      3   41
 2          108     0      1   41
 3           15     0      3   44
 4           67     0      3   42
 5          153     0      3   40
 6           51     0      1   42
```

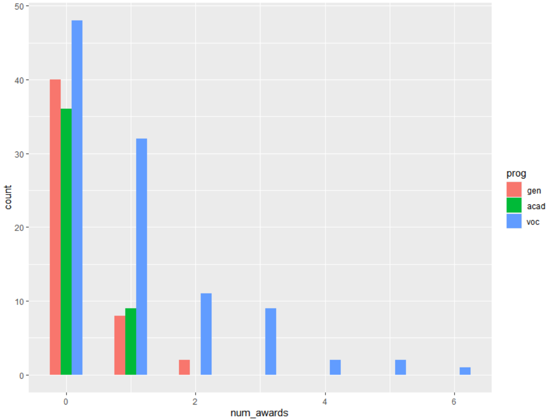
Poisson regression

```
> str(mydata)
'data.frame':  200 obs.  of 4 variables:
$ id :  int 45 108 15 67 153 51 164 133 2 53 ...
$ num_awards:  int 0 0 0 0 0 0 0 0 0 0 ...
$ prog :  int 3 1 3 3 3 1 3 3 3 3 ...
$ math :  int 41 41 44 42 40 42 46 40 33 46 ...
```

```
> mydata$prog =
factor(mydata$prog,levels=unique(mydata$prog),
labels=c("gen","acad","voc"))
> str(mydata)
'data.frame':  200 obs.  of 4 variables:
$ id :  int 45 108 15 67 153 51 164 133 2 53 ...
$ num_awards:  int 0 0 0 0 0 0 0 0 0 0 ...
$ prog :  Factor w/ 3 levels "gen","acad","voc":  1 2 1 1
1 2 1 1 1 1 ...
$ math :  int 41 41 44 42 40 42 46 40 33 46 ...
```

Poisson regression

```
> require(ggplot2)
> ggplot(mydata, aes(num_awards, fill = prog)) +
+ geom_histogram(binwidth=.5, position="dodge")
```



Poisson regression

We can now perform the Poisson model analysis using the `glm` function.

```
> p.model <- glm(num_awards ~ prog + math,
family="poisson", data=mydata)
> summary(p.model)
```

Call:

glm(formula = num_awards ~ prog + math, family = ”poisson”, data = mydata)

Coefficients:

	Estimate	*Std.Error*	*zvalue*	*Pr*(> |*z*|)	
(*Intercept*)	−4.87732	0.62818	−7.764	8.21*e* − 15	$***$
progacad	−0.36981	0.44107	−0.838	0.4018	
progvoc	0.71405	0.32001	2.231	0.0257	$*$
math	0.07015	0.01060	6.619	3.63*e* − 11	$***$

Poisson regression

The table shows that `math` has a statistically significant impact on `num_awards`.

The coefficient for `math` is .07. This means that the expected log count for a one-unit increase in math is .07.

The indicator variable progacad compares prog = "Academic" and prog = "General", the expected log count for prog = "Academic" decreases by about 0.37. The indicator variable progvoc is the expected difference in log count (approx .71) between prog = "Vocational" and the reference group prog = "General".

The information on deviance is also provided. We can use the residual deviance to perform a goodness of fit test for the overall model. The residual deviance is the difference between the deviance of the current model and the maximum deviance of the ideal model where the predicted values are identical to the observed. Therefore, if the residual difference is small enough, the goodness of fit test will not be significant, indicating that the model fits the data.

Goodness-of-Fit Tests

Goodness-of-fit tests are used to compare proportions of levels of a nominal variable to theoretical proportions. Common goodness-of-fit tests are chi-square, G-test (also called likelihood ratio tests), and binomial or multinomial exact tests.

In general, there are no assumptions about the distribution of data for these tests. However, the results of chi-square tests and G-tests can be inaccurate if statistically expected cell counts are low. A rule of thumb is that all statistically expected cell counts should be 5 or greater for chi-square- and G-tests.

Chi-square Test

Assumptions:

- A nominal variable with two or more levels
- Theoretical, typical, or neutral values for the proportions for this variable are needed for comparison
- chi-square and G-test may not be appropriate if there are cells with low expected counts

Hypotheses

1. Null hypothesis: The proportions for the levels for the nominal variable are not different from the theoretical proportions.
2. Alternative hypothesis (two-sided): The proportions for the levels for the nominal variable are different from the theoretical proportions.

Chi-square Test

Example.
A shop owner claims that an equal number of customers come into his shop each weekday. To test this hypothesis, a researcher records the number of customers that come into the shop in a given week and finds the following:

- Monday: 50 customers
- Tuesday: 60 customers
- Wednesday: 40 customers
- Thursday: 47 customers
- Friday: 53 customers

Hence our observations are
$O_1 = 50, O_2 = 60, O_3 = 40, O_4 = 47, O_5 = 53$

Note: total number of customers $= 250$

Chi-square Test

We test the hypothesis

1. H_0: the number of customers is the same every day.
2. H_1: the number of customers is not the same every day.

If the number of customers is the same every day, then the expected observations would be $E_1 = \cdots = E_5 = 50$

To test the hypothesis, we compute the test statistics

$$
\begin{aligned}
X^2 &= \sum_{i=1^5} \frac{(O_i - E_i)^2}{E_i} \\
&= \frac{(50-50)^2}{50} + \frac{(60-50)^2}{50} + \frac{(40-50)^2}{50} + \frac{(47-50)^2}{50} + \frac{(53-50)^2}{50} = 4.36
\end{aligned}
$$

To test the hypothesis at significance level $\alpha = 0.05$, we compute $\chi_{0.05,4}$ = `qchisq(0.95,4)` $= 9.487729$.
Since X^2 is not larger than 9.487729, we cannot reject the null hypothesis. That is, we accept that the number of customers is the same every day.

Chi-square Test

To carry out a Chi-Square Goodness of Fit Test in R we use the function

```
chisq.test(x, p)
```

where:

- x: A numerical vector of observed frequencies.
- p: A numerical vector of expected proportions.

The elements in x are numbers
The expected proportions must add up to 1

Chi-square Test

Solution of the example using R.

```
observed <- c(50, 60, 40, 47, 53)
expected <- c(.2, .2, .2, .2, .2)
```

Note: the expected vector is a set of probabilities adding up to 1

```
> chisq.test(x=observed, p=expected)

Chi-squared test for given probabilities

data:  observed
X-squared = 4.36, df = 4, p-value = 0.3595
```

Conclusion: Since p-value = 0.3595, we do not reject the nulll hypothesis at significance level 0.05

Chi-square Test

Note about R: The total number of observations is 250 Hence, if observations were uniform, there would be 50 observations for each bin.

One can run the Chi-square test in R alternatively as follows

```
> observed <- c(50, 60, 40, 47, 53)
> expected <- c(50, 50, 50, 50, 50)
> chisq.test(x=observed, p=expected,rescale.p = TRUE)

Chi-squared test for given probabilities

data:  observed
X-squared = 4.36, df = 4, p-value = 0.3595
```

Chi-square Test

Example. Here is the distribution of the number of girls per family in a sample of 100 families of 5 children

index	girls	frequency
1	0	5
2	1	12
3	2	28
4	3	33
5	4	17
6	5	5

Do the observed frequencies satisfy a binomial distribution?

Chi-square Test

We test the hypothesis

1. H_0: the number of girls follows a binomial distribution.
2. H_1: the number of girls does not follow a binomial distribution.

The expected frequencies, assuming a probability of 0.5 of having a girl for each of the 5 children, are given by the probabilities

$$p(k) = \binom{5}{i}(0.5)^k(0.5)^{5-k} = \texttt{dbinom}(\texttt{k}, \texttt{size} = 5, \texttt{prob} = 0.5),$$

for $k = 0, \ldots 5$

Chi-square Test

Solution in R

We build a vector with the expected relative frequencies according to the binomial pmf

```
> x <- 0:5
> expected = dbinom(x, size = 5, prob = 0.5)
> expected
[1] 0.03125 0.15625 0.31250 0.31250 0.15625 0.03125
```

Note `> sum(expected)`

```
[1] 1
```

Chi-square Test

```
> observed <-c(5,12,28,33,17,5)
> chisq.test(x=observed, p=expected)
```

```
Chi-squared test for given probabilities

data: observed
X-squared = 3.648, df = 5, p-value = 0.6011
```

Conclusion: Since p-value = 0.6011, we do not reject the nulll hypothesis at significance level 0.05

Chi-square Test

Example. As part of a demographic survey of students in his environmental issues webinar series, Alucard recorded the race and ethnicity of his students. He wants to compare the data for his class to the demographic data of the County.

Race	Alucard's class	County proportion
White	20	0.775
Black	9	0.132
American Indian	9	0.012
Asian	1	0.054
Pacific Islander	1	0.002
Two or more races	1	0.025
Total	41	1.000

Ethnicity	Alucard's class	County proportion
Hispanic	7	0.174
Not Hispanic	34	0.826
Total	41	1.000

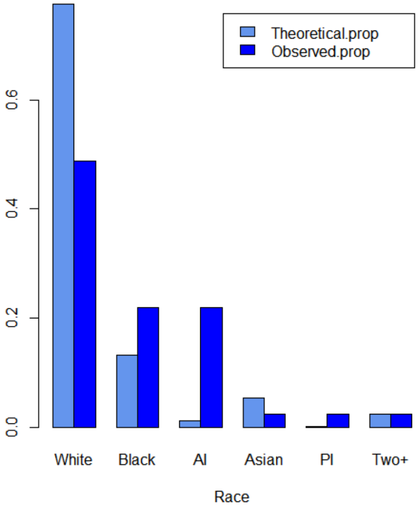
Chi-square Test

To facilitate race data visualization, the vector of counts is converted to proportions, and the theoretical and observed proportions are combined into a table.

```
> observed = c(20, 9, 9, 1, 1, 1)
> theoretical = c(0.775, 0.132, 0.012, 0.054, 0.002,
0.025)
> Observed.prop = observed / sum(observed)
> Theoretical.prop = theoretical
> Observed.prop = round(Observed.prop, 3)
> Theoretical.prop = round(Theoretical.prop, 3)
> XT = rbind(Theoretical.prop, Observed.prop)
> colnames(XT) = c("White", "Black", "AI", "Asian","PI",
"Two+")
> XT
                 White  Black     AI  Asian     PI   Two+
 Theoretical.prop 0.775  0.132  0.012  0.054  0.002  0.025
 Observed.prop    0.488  0.220  0.220  0.024  0.024  0.024
```

Chi-square Test

```
> barplot(XT,beside = T,xlab = "Race",col =
c("cornflowerblue","blue"),legend = rownames(XT))
```



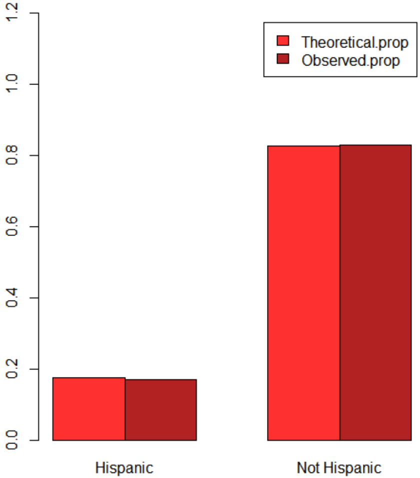
Chi-square Test

For the ethnicity data visualization, we proceed exactly the same way.

```
> observed = c(7, 34)
> theoretical = c(0.174, 0.826)
> Observed.prop = observed / sum(observed)
> Theoretical.prop = theoretical
> Observed.prop = round(Observed.prop, 3)
> Theoretical.prop = round(Theoretical.prop, 3)
> XT = rbind(Theoretical.prop, Observed.prop)
> colnames(XT) = c("Hispanic", "Not Hispanic")
> XT
                 Hispanic Not Hispanic
Theoretical.prop    0.174        0.826
Observed.prop       0.171        0.829
```

Chi-square Test

```
> barplot(XT,beside = T,xlab = "Ethnicity",col =
c("firebrick1","firebrick"),legend = rownames(XT),ylim =
c(0, 1.2))
```



Chi-square Test

Analysis of the race data.

```
> observed = c(20, 9, 9, 1, 1, 1)
> theoretical = c(0.775, 0.132, 0.012, 0.054, 0.002,
0.025)
> chisq.test(x = observed,p = theoretical)
```

```
Chi-squared test for given probabilities

data: observed
X-squared = 164.81, df = 5, p-value < 2.2e − 16
```

Test indicates that observed data do not fit theoretical data.

Chi-square Test

We check expected counts to assess if the test is appropriate.

```
> Test = chisq.test(x = observed, p = theoretical)
> Test$expected
[1] 31.775 5.412 0.492 2.214 0.082 1.025
```

The low expected counts: 0.492, 0.082, and 1.025, suggests that the test may not be valid.

A way to address this problem would be to aggregate some races into the same bin

Chi-square Test

Analysis of the ethnicity data.

```
> observed = c(7, 34)
> theoretical = c(0.174, 0.826)
> chisq.test(x = observed,p = theoretical)
```

```
Chi-squared test for given probabilities

data: observed
X-squared = 0.0030472, df = 1, p-value = 0.956
```

Test indicates that observed data fit theoretical data.

Chi-square Test

We check expected counts to assess if the test is appropriate.

```
> Test = chisq.test(x = observed, p = theoretical)
> Test$expected
[1] 7.134 33.866
```

There are no low expected counts, so there are no concerns about the validity of the test.

Chi-square Test

Post-hoc analysis. If the goodness of fit test is significant, a post-hoc analysis can be conducted to determine which counts differ from their theoretical proportions.

One approach is to look at the standardized residuals from the chi-square analysis. Cells with a standardized residual whose absolute value is greater than 1.96 indicate a cell differing from theoretical proportions. (The 1.96 cutoff is analogous to alpha $=$ 0.05 for a hypothesis test, or 2.58 for alpha $=$ 0.01.)

Chi-square Test

Post-hoc analysis of race data

```
> observed = c(20, 9, 9, 1, 1, 1)
> theoretical = c(0.775, 0.132, 0.012, 0.054, 0.002,
0.025)
> chisq.test(x = observed, p = theoretical)$stdres
[1] -4.403792 1.655440 12.202995 -0.838850 3.209005 -0.0250078
```

Cells with standardized residuals whose absolute value is greater than 1.96 are White, American Indian, and Pacific Islander; hence counts in these cells differ from theoretical proportions.

Chi-square Test

Post-hoc analysis of ethnicity data

```
> observed = c(7, 34)
> theoretical = c(0.174, 0.826)
> chisq.test(x = observed, p = theoretical)$stdres
[1] [1] -0.05520116 0.05520116
```

In this case, the goodness of fit test is not significant. Post-hoc analysis shows that no cells have standardized residuals with absolute value is greater than 1.96.

Chi-square Test of Independence

Recall that two random events A and B are called **independent** if $P(A \cap B) = P(A)P(B)$

Suppose that $O_{i,j}$ is the observed frequency count of events belonging to both i-th category of A and j-th category of B.

Also suppose that $E_{i,j}$ is the corresponding expected count if A and B are independent.

The null hypothesis of the independence assumption is to be rejected if the p-value of the following Chi-squared test statistics

$$X^2 = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

is less than a given significance level α.

Chi-square Test of Independence

Example. The following example examines the students smoking habit against their exercise level.

In the **contingency table** below the column records the students smoking habit ("Heavy", "Never", "Occas", "Regul") while the column records their exercise level ("Freq", "None", "Some").

```
> library(MASS)
> tbl = table(survey$Smoke, survey$Exer)
> tbl
          Freq   None   Some
 Heavy       7      1      3
 Never      87     18     84
 Occas      12      3      4
 Regul       9      1      7
```

Chi-square Test of Independence

Problem: Test the hypothesis whether the students smoking habit is independent of their exercise level at .05 significance level.

We apply the `chisq.test` function to the contingency table tbl

```
> chisq.test(tbl)
Pearson's Chi-squared test
data: tbl
X-squared = 5.4885, df = 6, p-value = 0.4828
```

Conclusion: Since p-value = 0.482, we do not reject the null hypothesis and we accept the hypothesis that the smoking habit is independent of the exercise level of the students.

Chi-square Test of Independence

The R solution from the example above contains a warning message:

In chisq.test(table(survey$Smoke, survey$Exer)):
Chi-squared approximation may be incorrect

The warning message is due to the small cell values in the contingency table.

To avoid such warning, we combine the second and third columns of tbl, and save it in a new table named ctbl.

```
> ctbl = cbind(tbl[,"Freq"], tbl[,"None"] +
tbl[,"Some"])
> ctbl
         [,1]  [,2]
 Heavy      7     4
 Never     87   102
 Occas     12     7
 Regul      9     8
```

Chi-square Test of Independence

We can now compute our revised solution by applying the `chisq.test` function against ctbl.

```
> chisq.test(ctbl)
```

```
Pearson's Chi-squared test
data: ctbl
X-squared = 3.2328, df = 3, p-value = 0.3571
```

Conclusion: Since p-value = 0.3571, we do not reject the null hypothesis. Also in this case we accept the hypothesis that the smoking habit is independent of the exercise level of the students.

Chi-square Test of Independence

Example. On April 14th 1912 the ship the Titanic sank. Only 705 passengers and crew out of the total 2228 population on board survived. Information on 1309 of those on board will be used to demonstrate summarizing categorical variables.

```
> Tdata <-
data.frame(read.csv('C:/Users/dlabate/Desktop/Teaching/ma4310/titanic.csv',header=T,sep=',')) >
str(Tdata)
'data.frame':  1310 obs.  of 14 variables:
$ pclass :  int 1 1 1 1 1 1 1 1 1 1 ...
$ survived :  int 1 1 0 0 0 1 1 0 1 0 ...
$ name :  chr "Allen, Miss.  Elisabeth Walton" "Allison, Master.  Hudson Trevor" "Allison, Miss.
Helen Loraine" "Allison, Mr.  Hudson Joshua Creighton" ...
$ sex :  chr "female" "male" "female" "male" ...
$ age :  num 29 0.917 2 30 25 ...
$ sibsp :  int 0 1 1 1 1 0 1 0 2 0 ...
$ parch :  int 0 2 2 2 2 0 0 0 0 0 ...
$ ticket :  chr "24160" "113781" "113781" "113781" ...
$ fare :  num 211 152 152 152 152 ...
```

Chi-square Test of Independence

We want to test the following hypotheses:

- H_0: Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton) is not associated with survival.
- H_1: Port of Embarkation is associated with survival

We first convert the relevant variables into factors

```
> Tdata$survived=factor(Tdata$survived,
levels=unique(Tdata$survived),
labels=c("survived","died"))
> Tdata$embarked=factor(Tdata$embarked,
levels=unique(Tdata$embarked))
```

Chi-square Test of Independence

Here is the table of the data

```
> tbl = table(Tdata$survived, Tdata$embarked)
> tbl
```

	S	*C*	*Q*
survived	304	150	46
died	610	120	79

We next run the chi-square test of independence

```
> chisq.test(tbl)


Pearson's Chi-squared test


data:  tbl
X-squared = 44.002, df = 2, p-value = 2.787e-10
```

Conclusion: Since p-value is less than 0.05, we reject the null hypothesis that survival is not associated with port of embarkation.

Chi-square Test of Independence

We now want to test the following hypotheses:

- H_0: Passenger Class ($1 = 1$st; $2 = 2$nd; $3 = 3$rd) is not associated with survival.
- H_1: Passenger Class is associated with survival

We first convert the relevant variable into a factor

```
> Tdata$pclass=factor(Tdata$pclass,
levels=unique(Tdata$pclass), labels=c("1","2","3"))
> tbl = table(Tdata$survived, Tdata$pclass)
> tbl
```

	1	2	3
survived	200	119	181
died	123	158	528

Chi-square Test of Independence

We next run the chi-square test of independence

```
> chisq.test(tbl)
Pearson's Chi-squared test

data:  tbl
X-squared = 127.86, df = 2, p-value < 2.2e-16
```

Conclusion: Since p-value is less than 0.05, we reject the null hypothesis that survival is not associated with passenger class.

Chi-square Test of Independence

We can carry out a post-hoc analysis by inspecting the values of the standardized residuals.

Recall that absolute values above 1.96 indicate that the difference from the theoretical value (of independence) is significant with significance level at least 0.05.

```
> chisq.test(tbl)$stdres
                   1           2            3
 survived   10.110480    1.837589   −10.254442
 died      −10.110480   −1.837589    10.254442
```

Chi-square Test of Independence

Comment: A positive result from a chi-squared test, like what we found concerning survival and point of embark, indicates that there is some kind of relationship between two variables but we do not know what sort of relationship it is. Further analysis is needed.

The following table shows port of embarkation vs passenger class.

```
> tbl2 = table(Tdata$pclass, Tdata$embarked)
> tbl2
```

	S	*C*	*Q*
1	177	141	5
2	242	28	7
3	495	101	113

The distribution of passengers among the three passenger classes is very different depending on the port of embarkation: 52% of passengers embarked at C are in class 1 as compared with 19% and 4% of those embarked at S and Q, respectively.

2×2 Contingency Table

Sometimes each of two criteria of classification can be broken down into two categories.
In this case, the result is a 2×2 **Contingency Table**

	1	2
1	a	b
2	c	d

and the test statistic has a closed formula

$$X^2 = \frac{n\,(ad - cb)^2}{(a+c)(b+d)(a+b)(c+d)}$$

where $n = a + b + c + d$

2×2 Contingency Table

Example. Some males and females are randomly surveyed whether they like sushi or not.

```
> counts <- c(19, 24, 18, 21)
> gender <- gl(n = 2, k = 1, length = 4, labels =
c("Male", "Female"))
> interest <- gl(n = 2, k = 2, length = 4, labels =
c("Yes", "No"))
> survey_data <- data.frame(counts, gender, interest)
> survey_data
```

```
     counts  gender  interest
 1       19    Male       Yes
 2       24  Female       Yes
 3       18    Male        No
 4       21  Female        No
```

2 × 2 Contingency Table

We display the data using a Contingency Tables (2 by 2 Case)

```
> cont_table <- xtabs(counts ~ gender + interest)
> cont_table
         interest
gender    Yes  No
Male       19  18
Female     24  21
```

We want test the hypothesis

1. H_0: the interest into sushi is independent of gender
2. H_1: the interest into sushi is dependent of gender

2 × 2 Contingency Table

We apply the Chi square test

```
> chisq.test(cont_table)
```

```
Pearson's Chi-squared test with Yates' continuity correction


data: cont_table
X-squared = 0, df = 1, p-value = 1
```

Conclusion: We accept the hypothesis that the interest into sushi is independent of gender

Note that R applied the Yates' continuity correction, due to the numbers in the table being small.

2 × 2 Contingency Table

The Fisher's Exact Test for Count Data offers an alternative approach to analyze contingency tables with small samples.

```
> fisher.test(cont_table)
```

```
Fisher's Exact Test for Count Data

data: cont_table
p-value = 1
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
0.3537211 2.4128090
sample estimates:
odds ratio
0.9245076
```

6 Non-parametric Statistics

Non-parametric Methods

A statistical method is called **non-parametric** (or **distribution-free**) if it makes no assumption on the population distribution or sample size.

This is in contrast with parametric methods in elementary statistics that assume that the data is quantitative and the population follows a probability distribution, e.g., the normal distribution, or that the sample size is sufficiently large so that we can apply the Centrl Limit Theorem.

Non-parametric methods make fewer assumptions, are more flexible, more robust, and applicable to non-quantitative data. The cost to pay for the fewer assumptions is that, in general, conclusions drawn from non-parametric methods are not as powerful as the parametric ones.

Non-parametric Methods

Nonparametric methods are useful for data that do not meet the assumptions of parametric analyses.

For instance, there are some situations when it is clear that data does not follow a normal distribution:

- data is an ordinal variable or a rank,
- data is definite outliers or
- data has clear limits of detection.

Additional situations where to use nonparametric methods include data that are skewed, non-normal, contain outliers, or, possibly, are censored, where censored data is data where there is an upper or lower limit to values. For example, if ages under 5 are reported as “under 5”.

Non-parametric Methods

A large class of nonparametric tests are **rank-based** tests. Instead of using the numeric values of the dependent variable, the dependent variable is converted into relative ranks.

Example. Suppose we have the heights of eight students in centimeters.

```
> Height = c(110, 132, 137, 139, 140, 142, 142, 145)
```

```
> names(Height) = letters[1:8]
```

```
Height
 a    b    c    d    e    f    g    h
 110  132  137  139  140  142  142  145
```

```
> rank(Height)
 a    b    c    d    e    f    g    h
 1.0  2.0  3.0  4.0  5.0  6.5  6.5  8.0
```

a has the smallest height and so is ranked 1. b has the next smallest height and so is ranked 2. And so on. Note that f and g are tied for spots 6 and 7, and so share a rank of 6.5.

Non-parametric Methods

Note that information about the absolute height values is lost, and only the relative ranking is retained in the ranks.

In fact, the value of a is quite a bit smaller than the others, but its rank is simply 1. That is, if the value of a were changed to 100 or 5 or –10, its rank would remain 1 in this data set.

The advantage of using these rank-based tests is that they don't make many assumptions about the distribution of the data. Instead, their conclusions are based on the relative ranks of values in the groups being tested.

One-sample Wilcoxon Signed-rank Test

The one-sample Wilcoxon test is a rank-based test to compare a set of values to a given default value.

It is useful to test a null hypothesis about a population mean when the z- or t-test are not applicable, e.g., when the sample size is small and the population is grossly non-normally distributed

The Wilcoxon Signed-rank Test assumes that

- The sample is random
- The variable is continuous
- The population is symmetrically distributed about its mean
- The measurement scale is at least interval

For purely ordinal data, the one-sample sign test (to be discussed next) could be used instead.

One-sample Wilcoxon Signed-rank Test

Hypotheses

1. Null hypothesis: The population mean is equal (larger, less) than the default value.
2. Alternative hypothesis: The population man is different (less, larger) than the default value.

One-sample Wilcoxon Signed-rank Test

Let $x_1, \ldots x_N$ be a random sample that we want to test about some unknown population mean μ_0
The Wilcoxon Signed-rank Test is performed as follows

1. Calculates the difference between each observation and the hypothesized mean μ_0:

$$d_i = x_i - \mu_0$$

2. Rank the values d_i from the smallest to the largest in absolute value. If two or more of the $|d_i|$ are equal, assign each tied valu the mean of the rank positions they occupy. For instance, if the three smallest $|d_i|$ are equal, place them in rank position 1,2,3 bu assign each a rank $(1 + 2 + 3)/3 = 2$
3. Assign each rank the sign of the corresponding d_i
4. Compute T_+, the sum of ranks with positive signs, and T_-, the sum of ranks with negative signs.

One-sample Wilcoxon Signed-rank Test

The test statistics of the Wilcoxon Signed-rank Test is

1. T_+ if we are testing $H_0 : \mu \geq \mu_0$ against $H_1 : \mu < \mu_0$
2. T_- if we are testing $H_0 : \mu \leq \mu_0$ against $H_1 : \mu > \mu_0$
3. $T = \min\{T_-, T_+\}$ if we are testing $H_0 : \mu = \mu_0$ against $H_1 : \mu \neq \mu_0$

Critical values of the Wilcoxon Signed-rank Test are tabulated.

In R, for a two-sided test, they are found using
`qsignrank(.025,n) -1`
where n is the sample size and $\alpha = 0.05$.

For instance, for $n = 15$, `qsignrank(.025,15) -1`=25

One-sample Wilcoxon Signed-rank Test

Example. Here are the values of the cardiac output (l/min) measured by thermodilution in a random sample of $n = 15$ patients

$$4.91, 4.10, 6.74, 7.27, 7.42, 7.50, 6.65, 4.64, 5.98, 3.14, 3.23, 5.80, 6.17, 5.39, 5.77$$

We want to test if the population mean is different from 5.05 at significance level $\alpha = 0.05$

1. $H_0 : \mu = 5.05$
2. $H_1 : \mu \neq 5.05$

One-sample Wilcoxon Signed-rank Test

We compute the signed ranks of the $|d_i|$

```
> x <-c(4.91, 4.10, 6.74, 7.27, 7.42, 7.50, 6.65, 4.64,
5.98, 3.14, 3.23, 5.80, 6.17, 5.39, 5.77)
> x-5.05
[1] -0.14 -0.95 1.69 2.22 2.37 2.45 1.60 -0.41 0.93 -1.91
-1.82 0.75 1.12 0.34 0.72
> rank(abs(x-5.05))
[1] 1 7 10 13 14 15 9 3 6 12 11 5 8 2 4
> y=rank(abs(x-5.05))*sign(x-5.05)
> y
[1] -1 -7 10 13 14 15 9 -3 6 -12 -11 5 8 2 4
```

One-sample Wilcoxon Signed-rank Test

We now compute T_- and T_+

```
> Tplus=sum(y*(y>0))
> Tminus=-sum(y*(y<0))
> Tminus
[1] 34
> Tplus
[1] 86
```

The test statistic is $T = \min\{T_-, T_+\} = 34$
and the rejection region is $T < T_{crtitical}$.

Since $T > \texttt{qsignrank}(.025, 15) - 1 = 25$, we accept H_0.

One-sample Wilcoxon Signed-rank Test

Here is the solution in R using the built-in command `wilcox.test`

```
> x <-c(4.91, 4.10, 6.74, 7.27, 7.42, 7.50, 6.65, 4.64,
5.98, 3.14, 3.23, 5.80, 6.17, 5.39, 5.77)
> wilcox.test(x,mu=5.05)
```

```
Wilcoxon signed rank exact test

data: x
V = 86, p-value = 0.1514
alternative hypothesis: true location is not equal to 5.05
```

One-sample Wilcoxon Signed-rank Test

Example. An instructor (Maggie Simpson) is being anonymously rated by the students in the course with scores (the `likert` score) ranging from 1 ("Strongly agree") to 5 ("Strongly disagree"). We want to answers the question, "Are the instructor's scores significantly different from a 'neutral' score of 3?"

Hence, we want test the hypothesis:

- H_0 : The likert score of Maggie Simpson is 3
- H_1 : The likert score of Maggie Simpson is different from 3

One-sample Wilcoxon Signed-rank Test

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE,
text="
 Instructor          Rater   Likert
 'MaggieSimpson'       1       3
 'MaggieSimpson'       2       4
 'MaggieSimpson'       3       5
 'MaggieSimpson'       4       4
 'MaggieSimpson'       5       4
 'MaggieSimpson'       6       4
 'MaggieSimpson'       7       4
 'MaggieSimpson'       8       3
 'MaggieSimpson'       9       2
 'MaggieSimpson'      10       5
")
```

One-sample Wilcoxon Signed-rank Test

We convert the `Likert` variable into a factor

```
> Data$Likert.f = factor(Data$Likert,ordered = TRUE)
```

We diplay the data table

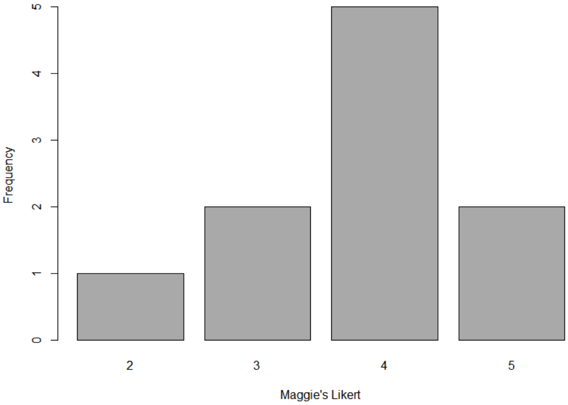
```
> xtabs( ~ Instructor + Likert.f,data = Data)
```

```
                  Likert.f
 Instructor       2 3 4 5
 Maggie Simpson   1 2 5 2
```

One-sample Wilcoxon Signed-rank Test

Here is the bar plot of the data

```
> XT = xtabs(~ Likert.f,data=Data)
> barplot(XT,col="dark gray",xlab="Maggie's
Likert",ylab="Frequency")
```



One-sample Wilcoxon Signed-rank Test

To run the one-sample Wilcoxon signed-rank test in R, we call the `wilcox.test` function

In the `wilcox.test` function, the `mu` option indicates the value of the default value to compare to.

```
> wilcox.test(Data$Likert,mu=3,conf.int=TRUE,
conf.level=0.95)
```

```
Wilcoxon signed rank test with continuity correction

data: Data$Likert
V = 32.5, p-value = 0.04007
alternative hypothesis: true location is not equal to 3
95 percent confidence interval:
3.000044 4.500083
sample estimates:
(pseudo)median
4.000032
```

One-sample Wilcoxon Signed-rank Test

Conclusion:
Since the p-value $= 0.04007$, we reject H_0 (the likert score of Maggie Simpson is 3) and accept H_1 (the likert score of Maggie Simpson is different from 3) at significance leve;l α = 0.05.

From the output of the one-sample Wilcoxon Signed-rank Test we also have that:

> the 95 percent confidence interval of the likert score is [3.000044 4.500083]
>
> the pseudo-median likert score is 4.000032

Sign Test for One-sample Data

The **one-sample sign test** compares the number of observations greater than or less than the default value without accounting for the magnitude of the difference between each observation and the default value.

The test is similar in purpose to the one-sample Wilcoxon signed-rank test, but looks specifically at the **median value**, and is not affected by the distribution of the data.

Hypotheses

1. Null hypothesis: The median of the population from which the sample was drawn is equal to the default value.
2. Alternative hypothesis (two-sided): The median of the population from which the sample was drawn is not equal to the default value.

Sign Test for One-sample Data

Example. An instructor (Maggie Simpson) is being rated by the students in the course with scores (the `likert` score) ranging from 1 (”Strongly agree”) to 5 (”Strongly disagree”). Are the instructor’s scores significantly different from a ‘neutral’ score of 3?

```
> library(nonpar)
> signtest(Data$Likert, m=3, conf.level=0.95, exact=FALSE)
```

```
Exact Sign Test
H0: The population median is = 3
HA: The population median is not equal to 3
B = 7

Significance Level = 0.05
The p-value is 0.07032
There is not enough evidence to conclude that the population
median is different than 3 at a significance level of 0.05

The 95 % confidence interval is [2 , 4].
```

Sign Test for One-sample Data

Alternatively, we can call the SIGN.test command from the BSDA library.

```
> if(!require(BSDA))install.packages("BSDA")
> library(BSDA)
> SIGN.test(Data$Likert, md = 3)
```

```
One-sample Sign-Test

data: Data$Likert
s = 7, p-value = 0.07031
alternative hypothesis: true median is not equal to 3
95 percent confidence interval:
3.000000 4.675556
sample estimates:
median of x
4
```

Two-sample Mann–Whitney U Test

The **two-sample Mann–Whitney U test** is a rank-based test that compares values for two groups. It is equivalent to a two-sample Wilcoxon rank-sum test.

The test assumes that the observations are independent. Hence, it is not appropriate for paired observations or repeated measures data.

Without assumptions about the distribution of the data, this test does not address hypotheses about the medians of the groups. Instead, the test addresses if it is likely that an observation in one group is greater than an observation in the other. This is sometimes stated as testing if one sample has stochastic dominance compared with the other.

Two-sample Mann–Whitney U Test

Assumptions:

- Two-sample data. That is, one-way data with two groups only
- Dependent variable is ordinal, interval, or ratio
- Independent variable is a factor with two levels. That is, two groups
- Observations between groups are independent. That is, not paired or repeated measures data.
- In order to be a test of medians, the distributions of values for each group need to be of similar shape and spread. Otherwise, the test is typically a test of stochastic equality

Hypotheses

1. Null hypothesis: The two groups are sampled from populations with identical distributions.
2. Alternative hypothesis (two-sided): The two groups are sampled from populations with different distributions.

Two-sample Mann–Whitney U Test

Example. We want to compare the ratings scores of two instructors, Pooh and Piglet, who have been rated independently.

```
Data = read.table(header=TRUE,stringsAsFactors=TRUE,text="
 Instructor    Likert
 Pooh             3
 Pooh             5
 Pooh             4
 Pooh             4
 Pooh             4
 Pooh             4
 Pooh             4
 Pooh             4
 Pooh             5
 Pooh             5
 Piglet           2
 Piglet           4
 Piglet           2
 Piglet           2
 Instructor    Likert
 Piglet           1
 Piglet           2
 Piglet           3
 Piglet           2
 Piglet           2
 Piglet           3
")
```

Two-sample Mann–Whitney U Test

We convert the `Likert` variable into a factor

```
Data$Likert.f = factor(Data$Likert, ordered = TRUE)
```

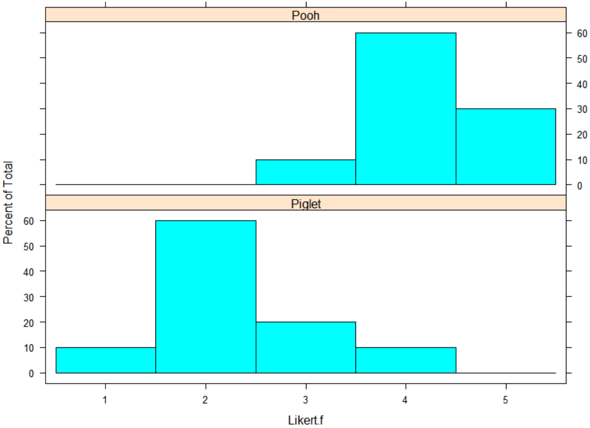
We diplay the data as a table

```
> xtabs( ~ Instructor + Likert.f,data = Data)
             Likert.f
 Instructor   1 2 3 4 5
 Piglet       1 6 2 1 0
 Pooh         0 0 1 6 3
```

Two-sample Mann–Whitney U Test

Here are the histograms of the Likert scores for the two instructors

```
>histogram(~ Likert.f | Speaker,
data=Data,layout=c(1,2))
```



Two-sample Mann–Whitney U Test

To run the Two-sample Mann–Whitney U Test we apply the `wilcox.test` command as follows.

```
> wilcox.test(Likert ~ Instructor,data=Data)
```

```
Wilcoxon rank sum test with continuity correction

data: Likert by Instructor
W = 5, p-value = 0.0004713
alternative hypothesis: true location shift is not equal to 0
```

Conclusion: since p-value = 0.0004713, there is a statistically significant difference between the scores of the two instructors

Two-sample Mann–Whitney U Test

R note

The command `wilcox.test` performs one- and two-sample Wilcoxon tests on vectors of data; the latter is also known as ‘Mann-Whitney’ test.

Usage

```
wilcox.test(x, ...)
```

- method for default
  wilcox.test(x, y = NULL, alternative = c(”two.sided”, ”less”, ”greater”), mu = 0, paired = FALSE, exact = NULL, correct = TRUE, conf.int = FALSE, conf.level = 0.95)
- method for formula
  wilcox.test(formula, data, subset, na.action)

Two-sample Mann–Whitney U Test

Example[Hollander & Wolfe (1973)]
We compare the permeability constants of the human chorioamnion (a placental membrane) at term (x) and between 12 to 26 weeks gestational age (y). The alternative of interest is greater permeability of the human chorioamnion for the term pregnancy.

```
x <- c(0.80, 0.83, 1.89, 1.04, 1.45, 1.38, 1.91,
1.64, 0.73, 1.46)
y <- c(1.15, 0.88, 0.90, 0.74, 1.21)
wilcox.test(x, y, alternative = "greater")
```

Alternatively

```
wilcox.test(permeability ~ group, alternative =
"greater", data=Data)
```

Mood's Median Test for Two-sample Data

The **Mood's median test** compares the medians of two or more groups.

Assumptions:

- One-way data with two or more groups
- Dependent variable is ordinal, interval, or ratio
- Independent variable is a factor with levels indicating groups
- Observations between groups are independent. That is, not paired or repeated measures data.

Hypotheses

1. Null hypothesis: The medians of the populations from which the groups were sampled are equal.
2. Alternative hypothesis (two-sided): The medians of the populations from which the groups were sampled are not equal.

Mood's Median Test for Two-sample Data

Example. We want to compare the ratings scores of two instructors, Pooh and Piglet, who have been rated independently. We want to answer the question, "Are Pooh's scores significantly different from those of Piglet?" Same data as above.

```
> X = Data$Likert[Data$Instructor=="Pooh"]
> Y = Data$Likert[Data$Instructor=="Piglet"]
> library(nonpar)
> mediantest(x = X, y = Y, exact=TRUE)
```

```
Exact Median Test
H0: The 2 population medians are equal.
HA: The 2 population medians are not equal.

Significance Level = 0.05
The p-value is 0.0010825088224469
There is enough evidence to conclude that the population medians
are different at a significance level of 0.05.
```

Two-sample Paired Signed-rank Test

The **two-sample paired signed-rank test** (also called **Wilcoxon paired signed-rank test)** is used to compare values for two groups where each observation in one group is paired with one observation in the other group.

The test is useful, for instance, to compare scores on a pre-test vs. scores on a post-test, or scores or ratings from two speakers, two different presentations, or two groups of audiences when there is a reason to pair observations, such as being done by the same rater.

Because the first step in the calculations is the subtraction of the paired values, one from the other, the data must be at least ordinal in nature.

The test is equivalent to using a one-sample signed-rank test on the difference of the paired values.

Two-sample Paired Signed-rank Test

Assumptions:

- Two-sample paired data. That is, one-way data with two groups only, where the observations are paired between groups.
- Dependent variable is interval or ratio.
- Independent variable is a factor with two levels; that is, two groups.
- For the test to be a test of the median of the differences, the distribution of differences in paired samples needs to be symmetric.

Hypotheses

1. Null hypothesis: The population of the differences of paired values is symmetric around zero.
2. Alternative hypothesis (two-sided): The population of the differences of paired values is not symmetric around zero.

Two-sample Paired Signed-rank Test

Example. We want to compare Likert scores for Pooh between Time 1 and Time 2.

```
Data = read.table(header=TRUE,stringsAsFactors=TRUE,text="
 Instructor   Time   Student   Likert
 Pooh           1       a         1
 Pooh           1       b         4
 Pooh           1       c         3
 Pooh           1       d         3
 Pooh           1       e         3
 Pooh           1       f         3
 Pooh           1       g         4
 Pooh           1       h         3
 Pooh           1       i         3
 Pooh           1       j         3
 Pooh           2       a         4
 Pooh           2       b         5
 Pooh           2       c         4
 Pooh           2       d         5
 Pooh           2       e         4
 Pooh           2       f         5
 Pooh           2       g         3
 Pooh           2       h         4
 Pooh           2       i         3
 Pooh           2       j         4
")
```

Two-sample Paired Signed-rank Test

In the data table, we recorded the identity of the student raters, and Pooh's score for each rater.

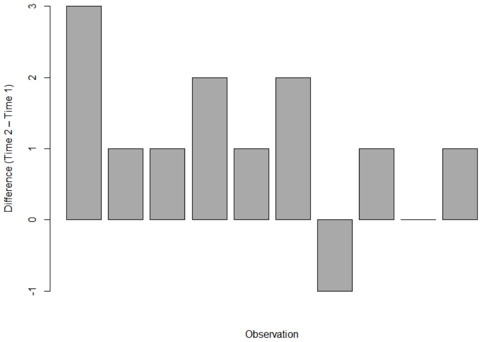
Summary table of the data shows that each student has paired observations.

```
> xtabs( ~ Student + Time,data = Data)
          Time
 Student   1 2
 a         1 1
 b         1 1
 c         1 1
 d         1 1
 e         1 1
 f         1 1
 g         1 1
 h         1 1
 i         1 1
 j         1 1
```

Two-sample Paired Signed-rank Test

Data is arranged in long form. To apply the statistical test, data must be ordered so that observations are paired.

```
> Time.1 = Data$Likert[Data$Time==1]
> Time.2 = Data$Likert[Data$Time==2]
> Difference = Time.2 - Time.1
> barplot(Difference,col="dark gray",
xlab="Observation",ylab="Difference (Time 2 - Time 1)")
```



Two-sample Paired Signed-rank Test

To run the two-sample paired signed-rank test, we call the command `wilcox.test` with the parameter `paired = TRUE`.

```
> wilcox.test(Likert ~ Time,data = Data,paired =
TRUE,conf.int = TRUE,conf.level = 0.95)
```

```
Wilcoxon signed rank test with continuity correction

data: Likert by Time
V = 3.5, p-value = 0.02355
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
-2.000051e+00 -1.458002e-05
sample estimates:
(pseudo)median
-1.000083
```

Two-sample Paired Signed-rank Test

Equivalently, one can run the Two-sample Paired Signed-rank Test with the command `wilcox.test` and the parameter `paired = TRUE` as follows.

```
> wilcox.test(Time.1,Time.2,paired = TRUE,conf.int =
TRUE,conf.level = 0.95)
```

```
Wilcoxon signed rank test with continuity correction

data: Likert by Time
V = 3.5, p-value = 0.02355
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
-2.000051e+00 -1.458002e-05
sample estimates:
(pseudo)median
-1.000083
```

Two-sample Paired Signed-rank Test

Example. In the built-in data set named `immer`, the barley yield in years 1931 and 1932 of the same field are recorded. Without assuming the data to have normal distribution, we want to test at .05 significance level if the barley yields of 1931 and 1932 in data set immer have identical data distributions.

```
> library(MASS)
> head(immer)
    Loc  Var     Y1     Y2
 1   UF    M   81.0   80.7
 2   UF    S  105.4   82.3
 3   UF    V  119.7   80.4
 4   UF    T  109.7   87.2
 5   UF    P   98.3   84.2
 6    W    M  146.6  100.4
```

Two-sample Paired Signed-rank Test

```
> Year1 = immer$Y1
> Year2 = immer$Y2
> wilcox.test(Year1,Year2,paired = TRUE,conf.int =
TRUE,conf.level = 0.95)
```

```
Wilcoxon signed rank test with continuity correction

data: Year1 and Year2
V = 368.5, p-value = 0.005318
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
5.900029 27.499955
sample estimates:
(pseudo)median
18.89997
```

We conclude that, at .05 significance level, the barley yields of 1931 and 1932 from the data set immer are nonidentical populations.

Two-sample Paired Signed-rank Test

NOTE: Under the assumption that data are normally distributed, we could run a paired sample t-test on the same data.

```
> t.test(Year1,Year2,paired = TRUE,conf.int =
TRUE,conf.level = 0.95)
```

```
Paired t-test


data: Year1 and Year2
t = 3.324, df = 29, p-value = 0.002413
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
6.121954 25.704713
sample estimates:
mean difference
15.91333
```

Also in this case we can reject the null hypothesis. The p-value obtained from the paired sample t-test is smaller than the paired signed-rank Test.

Kruskal–Wallis Test

The **Kruskal–Wallis test** is a rank-based test that is similar to the Mann–Whitney U test, but can be applied to one-way data with more than two groups.

Without any assumptions about the distribution of the data, the Kruskal–Wallis test does not address hypotheses about the medians of the groups. Instead, the test addresses if it is likely that an observation in one group is greater than an observation in the other. This is sometimes stated as testing if one sample has stochastic dominance compared with the other.

NOTE: the test assumes that the observations are independent. Hence it is not appropriate for paired observations or repeated measures data.

Kruskal–Wallis Test

Assumptions:

- One-way data with two or more groups
- Dependent variable is ordinal, interval, or ratio
- Independent variable is a factor with two or more levels.
- Observations between groups are independent. That is, not paired or repeated measures data
- In order to be a test of medians, the distributions of values for each group need to be of similar shape and spread. Otherwise, the test is typically a test of stochastic equality.

Hypotheses

1. Null hypothesis: The groups are sampled from populations with identical distributions. Typically, that the sampled populations exhibit stochastic equality.
2. Alternative hypothesis (two-sided): The groups are sampled from populations with different distributions. Typically, that one sampled population exhibits stochastic dominance.

Kruskal–Wallis Test

Interpretation of the Kruskal–Wallis Test.

Significant results indicate that: There was a significant difference in values among groups.
Post-hoc analysis is needed to be able to conclude if there was a significant difference in values between groups A and B, and so on.

Remark. The Mood's median test is also compares multiple populations, specifically itcompare the medians of groups.

There is conflicting information in the literature about the Mann–Whitney and Kruskal–Wallis tests. Some authors state that they test medians, usually adding an assumption that the distributions of the groups need to be of the same shape and spread. If this assumption holds, then, yes, these tests can be thought of as tests of location such as the median.
Without this assumption, these tests compare the stochastic dominance of the groups. Once a rank transformation is applied, stochastic dominance is exhibited simply by the groups with higher values.

Kruskal–Wallis Test

Example. We want to compare the following likert scores.

```
Data = read.table(header=TRUE,stringsAsFactors=TRUE,text="
  Speaker     Likert
  Pooh          3
  Pooh          5
  Pooh          4
  Pooh          4
  Pooh          4
  Pooh          4
  Pooh          4
  Pooh          4
  Pooh          5
  Pooh          5
  Piglet        2
  Piglet        4
  Piglet        2
  Piglet        2
  Piglet        1
  Piglet        2
  Piglet        3
  Piglet        2
  Piglet        2
  Piglet        3
  Tigger        4
  Tigger        4
  Tigger        4
  Tigger        4
  Tigger        5
  Tigger        3
  Tigger        5
  Tigger        4
  Tigger        4
  Tigger        3
")
```

Kruskal–Wallis Test

We order levels of the factor, otherwise R will alphabetize them

```
> Data$Speaker = factor(Data$Speaker,
levels=unique(Data$Speaker))
```

We create a new variable which is the likert scores as an ordered factor

```
Data$Likert.f = factor(Data$Likert, ordered = TRUE)
```

Here is the table orf the data

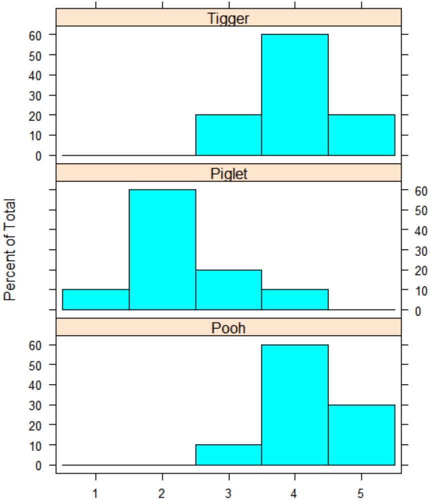
```
> xtabs( ~ Speaker + Likert.f,data = Data)
```

```
          Likert.f
 Speaker  1 2 3 4 5
 Pooh     0 0 1 6 3
 Piglet   1 6 2 1 0
 Tigger   0 0 2 6 2
```

Kruskal–Wallis Test

Here we show bar plots of data by group

```
> library(lattice)
> histogram(~ Likert.f | Speaker,data=Data,layout=c(1,3))
```



Kruskal–Wallis Test

We now run the Kruskal–Wallis Test in R

```
> kruskal.test(Likert ~ Speaker,data = Data)

Kruskal-Wallis rank sum test

data: Likert by Speaker
Kruskal-Wallis chi-squared = 16.842, df = 2, p-value = 0.0002202
```

Since the p-value is less that 0.05, we conclude that we reject the null hypothesis at significance level 0.05.

There is a significant difference between the scores of the 3 speakers.

Kruskal–Wallis Test

If the Kruskal–Wallis test is significant, a **post-hoc analysis** can be performed to determine which groups differ from each other group. The most popular post-hoc test is the **Dunn test**. Because the post-hoc test produces multiple p-values, adjustments to the p-values is made.

```
> library(FSA)
> dunnTest(Likert ~ Speaker,data=Data,method="bh")
```

Dunn (1964) Kruskal-Wallis multiple comparison p-values adjusted with the Benjamini-Hochberg method.

	Comparison	*Z*	*P.unadj*	*P.adj*
1	*Piglet – Pooh*	−3.7702412	0.0001630898	0.0004892695
2	*Piglet – Tigger*	−3.2889338	0.0010056766	0.0015085149
3	*Pooh – Tigger*	0.4813074	0.6302980448	0.6302980448

Conclusion: Piglet vs Pooh and Piglet vs Tigger are statistically different.

Mood's Median Test

Mood's median test compares the medians of two or more groups. The test can be conducted with the `median_test` function in the coin package.

Post-hoc tests: The outcome of Mood's median test tells you if there are differences among the groups, but doesn't tell you which groups are different from other groups. In order to determine which groups are different from others, post-hoc testing can be conducted. The function pairwiseMedianTest in the R companion package can perform the post-hoc tests. It simply passes data for pairs of groups to coin::median_test and produces a table of output.

Mood's Median Test

Example. We want to compare the following likert scores.

```
Data = read.table(header=TRUE,stringsAsFactors=TRUE,text="
   Speaker    Likert
   Pooh         3
   Pooh         5
   Pooh         4
   Pooh         4
   Pooh         4
   Pooh         4
   Pooh         4
   Pooh         4
   Pooh         5
   Pooh         5
   Piglet       2
   Piglet       4
   Piglet       2
   Piglet       2
   Piglet       1
   Piglet       2
   Piglet       3
   Piglet       2
   Piglet       2
   Piglet       3
   Tigger       4
   Tigger       4
   Tigger       4
   Tigger       4
   Tigger       5
   Tigger       3
   Tigger       5
   Tigger       4
   Tigger       4
   Tigger       3
")
```

Mood's Median Test

To apply the test, we need to load the coin library

```
> library(coin)
> media_test(Likert ~ Speaker,data = Data)
```

```
Asymptotic K-Sample Brown-Mood Median Test

data: Likert by Speaker (Pooh, Piglet, Tigger)
chi-squared = 3.248, df = 2, p-value = 0.1971
```

Note: An interesting thing happened with the result here. The test counts how many observations in each group are greater than the global median for all groups together, in this case 4. It then tests **if there is a significant difference in this proportion among groups**. For this data set, however, both Pooh and Tigger have a majority of observations equal to the global median. *Because they are equal to the global median, they are not greater than the global median, and so aren't much different than Piglet's scores on this count. The result in this case is a non-significant p-value.*

Mood's Median Test

The test would come out differently if we were counting observation less than the global median, because Pooh and Tigger have few of these, and Piglet has relatively many.

To achieve that, we will invert the scale we are using. This is really an arbitrary change, but for this test, it can make a difference. Imagine if our original scale interpreted 5 to be the best, and 1 to be the worst. When we designed the survey tool, we could just as easily have made 1 the best and 5 the worst. And then instead of ranking "good" with a 4, the respondents would have marked it 2, and so on. By the way the calculations are done, this arbitrary change in scale will change the results of Mood's median test.

For a 5-point scale, we do this inversion by simply by making a new variable equal to 6 minus the original score.

Mood’s Median Test

```
Data$Likert.inv = 6 - Data$Likert
```

```
> head(Data)
```

	Speaker	*Likert*	*Likert.f*	*Likert.inv*
1	*Pooh*	3	3	3
2	*Pooh*	5	5	1
3	*Pooh*	4	4	2
4	*Pooh*	4	4	2
5	*Pooh*	4	4	2
6	*Pooh*	4	4	2

Note:

```
> median(Data$Likert)
[1] 4
> median(Data$Likert.inv)
[1] 2
```

Mood’s Median Test

In the prior application of the test, using `Data$Likert`, we tested if any population has median above the global median of 4.

Using `Data$Likert.inv`, we now test if any population is above the global median of 2 which is equivalent to test if any population has `Data$Likert` with median below the global median of 4.

```
> median_test(Likert.inv ~ Speaker,data = Data)
```

```
Asymptotic K-Sample Brown-Mood Median Test

data: Likert.inv by Speaker (Pooh, Piglet, Tigger)
chi-squared = 15.306, df = 2, p-value = 0.0004747
```

Now we obtain a significant result, consistent with the observation that ”Piglet” has a median below the global median of 4, as shown in the bar plots.

Mood's Median Test

If Mood's median test is significant, a post-hoc analysis can be performed to determine which groups differ from each other group.

For this we use the `pairwiseMedianTest` function in the `rcompanion` package, which conducts Mood's median test on all pairs of groups from one-way data. Because the post-hoc test will produce multiple p-values, adjustments to the p-values are made.

```
> Data$Speaker =
factor(Data$Speaker,levels=c("Pooh","Tigger","Piglet"))
> library(rcompanion)
> pairwiseMedianTest(Likert.inv ~
Speaker,data=Data,exact=NULL,method="fdr")
```

	Comparison	*p.value*	*p.adjust*
1	*Pooh* – *Tigger* = 0	0.5416	0.541600
2	*Pooh* – *Piglet* = 0	0.0004883	0.001465
3	*Tigger* – *Piglet* = 0	0.001381	0.002072

Nonparametric tests - Summary

- *Hypothesis testing for the mean - one sample.*
  Method: One-sample Wilcoxon Signed-rank Test.
  R-implementation: `wilcox.test`
  (Corresp. parametric test: t-test, R implementation `t.test`)
- *Hypothesis testing for the mean - two samples, independent.*
  Method: Two-sample Mann–Whitney U test.
  R-implementation: `wilcox.test`
  (Corresp. parametric test: t-test, R implementation `t.test`)
- *Hypothesis testing for the mean - two samples, paired.*
  Method: Two-sample Paired Signed-rank test.
  R-implementation: `wilcox.test`
  (Corresp. parametric test: Paired t-test, R implementation `t.test`)
- *Hypothesis testing for the mean - three or more samples.*
  Method: Kruskal-Wallis test.
  R-implementation: `kruskal.test`
  (Corresp. parametric test: ANOVA, R implementation `aov`)

Nonparametric regression

There are different techniques that are considered to be forms of nonparametric, semi-parametric, or robust regression:

- Kendall–Theil regression fits a linear model between one x variable and one y variable using a completely nonparametric approach.
- Rank-based estimation regression is another robust approach.
- Quantile regression is a very flexible approach that can find a linear relationship between a dependent variable and one or more independent variables.
- Local regression fits a smooth curve to the dependent variable and can accommodate multiple independent variables.
- Generalized additive models are a powerful and flexible approach.

Kendall–Theil regression

Kendall–Theil regression (sometimes called Theil–Sen regression) is a completely nonparametric approach to linear regression where there is one independent and one dependent variable. A modified, and preferred, method is named after Siegel.

It is robust to outliers in the dependent variable.

It simply computes all the lines between each pair of points, and uses the median of the slopes of these lines.

The method yields a slope and intercept for the fit line, and a p-value for the slope can be determined as well.

Typically, no measure analogous to r-squared is reported.

Kendall–Theil regression

Example. The following survey reports several measurements collected by 5 instructors for students in their classes related to their nutrition education program. We want to explore the relationship between Sodium and Calories.

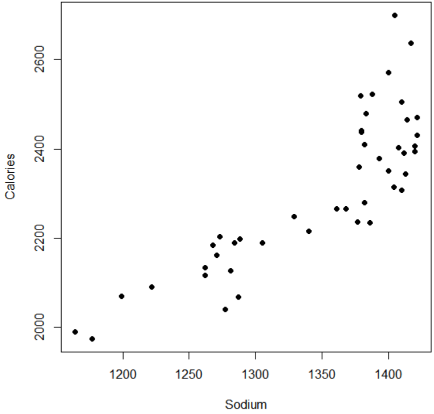
```
Data <- read.csv("C:/Users/student_survey.csv")
> head(Data)
```

	Instructor	*Grade*	*Weight*	*Calories*	*Sodium*	*Score*
1	*BrendonSmall*	6	43	2069	1287	77
2	*BrendonSmall*	6	41	1990	1164	76
3	*BrendonSmall*	6	40	1975	1177	76
4	*BrendonSmall*	6	44	2116	1262	84
5	*BrendonSmall*	6	45	2161	1271	86
6	*BrendonSmall*	6	44	2091	1222	87

Kendall–Theil regression

The inspection of the data shows that the relationship between Calories (y) and Sodium (x) variables is not particularly linear.

```
> plot(Calories ~ Sodium,data=Data,pch=16,ylab =
"Calories", xlab = "Sodium")
```



Kendall–Theil regression

We can try to fit the data using a standard linear regression

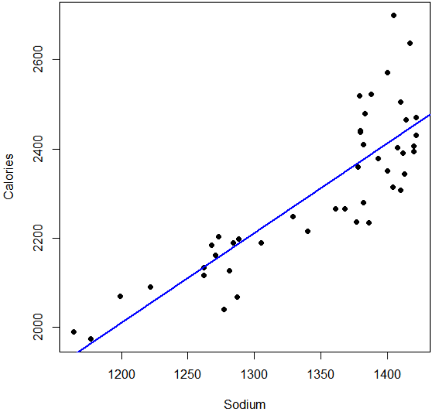
```
> model = lm(Calories ~ Sodium, data = Data)
> summary(model)
```

```
Call:
lm(formula = Calories ~ Sodium, data = Data)

 Min            1Q      Median      3Q        Max
 −149.678   −60.777   −7.621   46.122   277.188

Coefficients:
                 Estimate    Std.Error    tvalue     Pr(> |t|)
 (Intercept)   −398.1311    256.9580    −1.549        0.129
 Sodium            2.0071      0.1905    10.534    1.74e − 13   * * *

Residual standard error: 91.94 on 43 degrees of freedom
Multiple R-squared: 0.7207, Adjusted R-squared: 0.7142
F-statistic: 111 on 1 and 43 DF, p-value: 1.737e-13
```

Kendall–Theil regression

The plot below shows that the linear regression line is not a good fit.

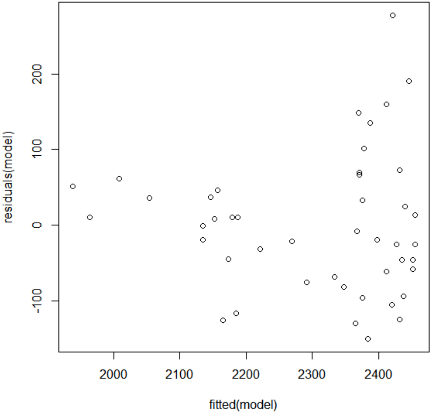
```
> plot(Calories ~ Sodium,data=Data,pch=16,ylab =
"Calories", xlab = "Sodium")
> abline(model,col = "blue",lwd = 2)
```



Kendall–Theil regression

There is a clear nonlinearity in the data which is apparent in the plot of residuals vs. fitted values.

```
> plot(fitted(model),residuals(model))
```



Kendall–Theil regression

We next compute the Kendall–Theil regression.
For that, we apply the `mblm` function in the mblm package.

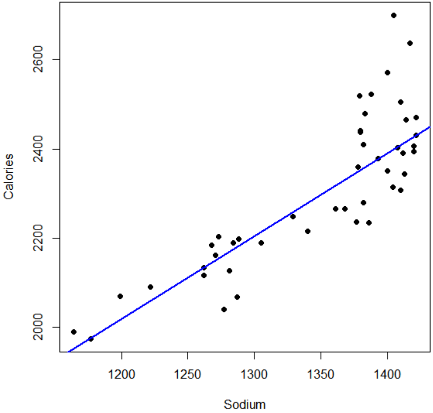
```
> library(mblm)
> model.k = mblm(Calories ~ Sodium, data=Data)
> summary(model.k)
```

```
Call:
mblm(formula = Calories ~ Sodium, dataframe = Data)

Residuals:
 Min           1Q   Median     3Q      Max
 −130.18   −41.27     0.00  48.85   299.56

Coefficients:
              Estimate        MAD   V value   Pr(> |V|)
 (Intercept)  −208.5875  608.4540       230    0.000861   ***
 Sodium          1.8562    0.4381      1035  5.68e − 14   ***

Residual standard error: 93.64 on 43 degrees of freedom.
```

Kendall–Theil regression

Here is the plot of the Kendall–Theil regression line.

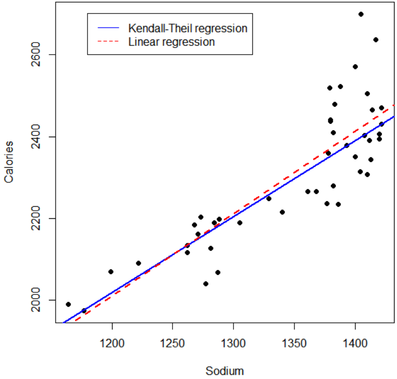
```
> plot(Calories ~ Sodium,data=Data,pch=16,ylab =
"Calories", xlab = "Sodium")
> abline(model.k,col = "blue",lwd = 2)
```



Kendall–Theil regression

Here we compare the regression lines.

```
> plot(Calories ~ Sodium,data=Data,pch=16,ylab = "Calories", xlab = "Sodium")

> abline(model.k,col = "blue",lwd = 2)

> abline(model,col = "red",lty = "dashed",lwd = 2)

> legend(1180, 2700, legend=c("Kendall-Theil regression", "Linear

regression"),col=c("blue","red"), lty=1:2, cex=1)
```



Kendall–Theil regression

Summary result:

The Kendall–Theil regression line is

$$y = -208.5875 + 1.8562\,x$$

as compared with the standard regression line

$$y = -398.1311 + 2.0071\,x$$

The lower value of the slope in the Kendall–Theil regression line is explained with the reduce sensitivity of this method to the larger data variability observed for larger values of the Sodium.

Rank-based regression

Rank-based estimation regression uses estimators and inference that are robust to outliers.

It is implemented using the `rfit` command from the Rfit package.

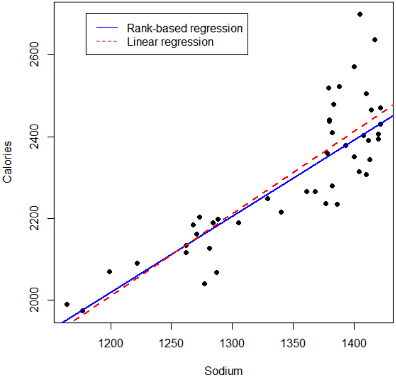
```
> library(Rfit)
> model.r = rfit(Calories ~ Sodium, data = Data)
> summary(model.r)
```

```
Call:
rfit.default(formula = Calories ~ Sodium, data = Data)

Coefficients:
                Estimate  Std.Error   tvalue     p.value
 (Intercept) −213.08411  248.42488  −0.8577      0.3958
 Sodium         1.85981    0.18407  10.1036  6.307e − 13   * * *

Multiple R-squared (Robust): 0.6637139
Reduction in Dispersion Test: 84.86733 p-value: 0
```

Rank-based regression

Here we compare the regression lines.

```
> plot(Calories ~ Sodium,data=Data,pch=16,ylab = "Calories", xlab = "Sodium")

> abline(model.r,col = "blue",lwd = 2)

> abline(model,col = "red",lty = "dashed",lwd = 2)

> legend(1180, 2700, legend=c("Rank-based regression", "Linear regression"),col=c("blue","red"),

lty=1:2, cex=1)
```



Rank-based regression

Summary result:

The Kendall–Theil regression line is

$$y = -213.0841 + 1.8598\,x$$

as compared with the standard regression line

$$y = -398.1311 + 2.0071\,x$$

The lower value of the slope in the rank-based regression line is explained with the reduce sensitivity of this method to the larger data variability observed for larger values of the Sodium.

Quantile regression

While traditional linear regression models the conditional mean of the dependent variable, quantile regression models the conditional median or other quantile. Medians are most common, but for example, if the factors predicting the highest values of the dependent variable are to be investigated, a 95th percentile could be used. Likewise, models for several quantiles, e.g., 25th , 50th, 75th percentiles, could be investigated simultaneously.

In the example reported below, we choose to model the median of dependent variable, which is indicated with the tau = 0.5 option.

Quantile regression makes no assumptions about the distribution of the underlying data, and is robust to outliers in the dependent variable.

It does assume the dependent variable is continuous. However, there are functions for other types of dependent variables.

Quantile regression

Quantile regression is implemented using the `rq` command from the quantreg package.

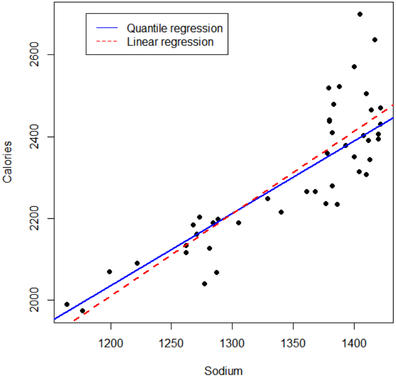
```
> library(quantreg)
> model.q = rq(Calories ~ Sodium, data = Data, tau =
0.5)
> summary(model.q)
```

```
Call: rq(formula = Calories ~ Sodium, tau = 0.5, data = Data)

tau: [1] 0.5

Coefficients:
              coefficients    lowerbd      upperbd
 (Intercept)  −84.12409     −226.58102   134.91738
 Sodium         1.76642        1.59035     1.89615
```

Rank-based regression

Here we compare the regression lines.

```
> plot(Calories ~ Sodium,data=Data,pch=16,ylab = "Calories", xlab = "Sodium")

> abline(model.q,col = "blue",lwd = 2)

> abline(model,col = "red",lty = "dashed",lwd = 2)

> legend(1180, 2700, legend=c("Quantile regression", "Linear regression"),col=c("blue","red"),

lty=1:2, cex=1)
```



Local regression

The basic idea of local regression is to fit a curve to data by averaging, or otherwise summarizing, data points that are next to one another.

Local regression is useful for investigating the behavior of the response variable in more detail than would be possible with a simple linear model.

Local polynomial regression is computed using the function `loess` in the native stats package. It can be used for one continuous dependent variable and up to four independent variables.

The process is essentially nonparametric, and is robust to outliers in the dependent variable.

Local regression

The `loess` function includes several optional parameters

```
loess(formula, data, weights, subset, na.action,
model = FALSE, span = 0.75, enp.target, degree = 2,
parametric = FALSE, drop.square = FALSE, normalize =
TRUE, family = c("gaussian", "symmetric"),method =
c("loess", "model.frame"),control = loess.control()))
```

- subset: an optional specification of a subset of the data to be used.
- span: it controls the degree of smoothing.
- degree: the degree of the polynomials to be used, normally 1 or 2.
- family: if ”gaussian”, fitting is done by least-squares; if ”symmetric”, a re-descending M estimator is used with Tukey’s biweight function.

Local regression

Here we choose a local polynomial regression of order 2, using least squares to fit the data.

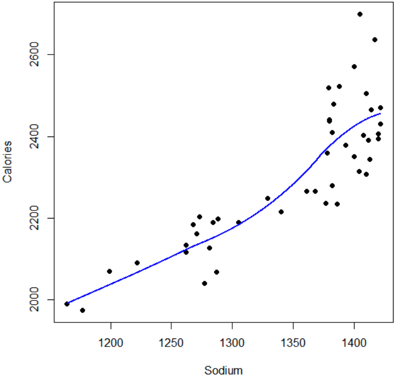
```
> model.l = loess(Calories ~ Sodium,data = Data,span
= 0.75,degree=2,family="gaussian")
> summary(model.l)
```

```
Call: loess(formula = Calories ~ Sodium, data = Data, span =
0.75, degree = 2, family = ”gaussian”)

Number of Observations: 45
Equivalent Number of Parameters: 4.19
Residual Standard Error: 91.97
Trace of smoother matrix: 4.57 (exact)
```

Local regression

To plot the local regression curve, we use the `plotPredy` command from the rcompanion library.

```
> library(rcompanion)
> plotPredy(data = Data, x = Sodium, y = Calories,
model = model.l, xlab = "Sodium", ylab = "Calories")
```



Generalized additive models (GAMs)

This is a flexible and smooth technique to capture nonlinearities in the data.

GAMs are a generalized version of classical Linear Models in which the Predictors X_i depend linearly or nonlinearly on some smooth nonlinear functions like polynomials, splines or step functions.

Given a set of predictors X_i, this method looks for a regression equation of the form

$$f(x) = y_i = \alpha + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i$$

where the function $f_1, \ldots, f_p$ are different nonlinear functions on variables X_p.

Generalized additive models (GAMs)

The gam function in the mgcv package uses smooth functions plus a conventional parametric component.

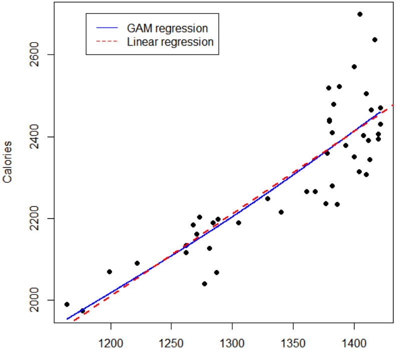
```
> library(mgcv)
> model.g = gam(Calories ~ s(Sodium),data = Data,
family=gaussian())
> summary(model.g)
```

```
Formula:
Calories ~ Sodium

Parametric coefficients:
              Estimate   Std.Error   tvalue   Pr(> |t|)
 (Intercept)   2304.87       13.62    169.2   < 2e − 16   * * *

Approximate significance of smooth terms:
               edf   Ref.df        F   p − value
 s(Sodium)   1.347    1.613    66.65   < 2e − 16   * * *

R-sq.(adj) = 0.718 Deviance explained = 72.6%
GCV = 8811.5 Scale est. = 8352 n = 45
```

Generalized additive models (GAMs)

We use `plotPredy` to plot the result.

```
> library(rcompanion)
> plotPredy(data = Data, x = Sodium, y = Calories, model =
model.g, xlab = "Sodium", ylab = "Calories")
> abline(model,col = "red",lty = "dashed",lwd = 2)
> legend(1180, 2700, legend=c("GAM regression", "Linear
regression"),col=c("blue"."red"). ltv=1:2. cex=1)
```



Generalized additive models (GAMs)

WARNING: The command `gam` without the function `s()` in front of the explanatory variable will simply yield the linear regression (provided the gaussian family is selected)!

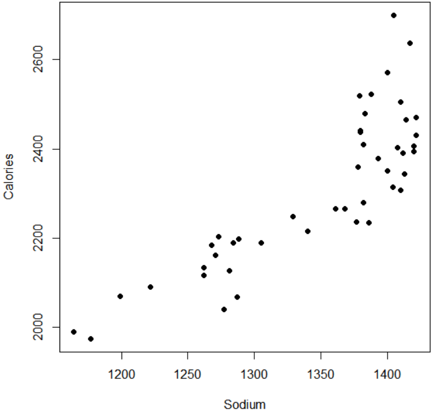
```
> library(mgcv)
> model.g = gam(Calories ~ Sodium,data = Data,
family=gaussian())
> summary(model.g)
```

```
Formula:
Calories ~ Sodium

Parametric coefficients:
                Estimate   Std.Error    tvalue   Pr(> |t|)
 (Intercept)   −398.1311    256.9580    −1.549       0.129
 Sodium           2.0071      0.1905    10.534   1.74e − 13   * * *

R-sq.(adj) = 0.718 Deviance explained = 72.6%
GCV = 8811.5 Scale est. = 8352 n = 45
```

Non-parametric correlation

As we observed above, the inspection of the data shows that the relationship between Calories (y) and Sodium (x) variables is not particularly linear.

```
> plot(Calories ~ Sodium,data=Data,pch=16,ylab =
"Calories", xlab = "Sodium")
```



Non-parametric correlation

We can analyze the correlation

```
> cor.test(Data$Sodium,Data$Calories, method =
"pearson")
```

```
Pearson's product-moment correlation

data: Data$Sodium and Data$Calories
t = 10.534, df = 43, p-value = 1.737e-13
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.7397691 0.9145785
sample estimates:
cor
0.8489548
```

Non-parametric correlation

Are the data from each of the 2 variables (x, y) following a normal distribution?

```
> shapiro.test(Data$Calories)
Shapiro-Wilk normality test

data: Data$Calories
W = 0.98697, p-value = 0.8873


> shapiro.test(Data$Sodium)
Shapiro-Wilk normality test

data: Data$Sodium
W = 0.85661, p-value = 5.441e-05
```

Non-parametric correlation

The **Spearman's rho statistic** is used to estimate a rank-based measure of association.
This test may be used if data do not come from a bivariate normal distribution.

```
> cor.test(Data$Sodium,Data$Calories, method =
"spearman")
```

```
Spearman's rank correlation rho

data: Data$Sodium and Data$Calories
S = 2729.7, p-value = 5.443e-12
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.8201766
```

Non-parametric correlation

Spearman's rho statistic is also used to estimate a **rank-based measure of association.**

This test may be used if the data do not come from a bivariate normal distribution. it only requires that each variable at least be measured on the ordinal scale.

Spearman's correlation determines the strength and direction of the **monotonic relationship** between your two variables rather than the strength and direction of the **linear relationship** between your two variables, which is what Pearson's correlation determines.

Monotonicity is less restrictive than a linear relationship.

A monotonic relationship is a relationship that does one of the following: (1) as the value of one variable increases, so does the value of the other variable; or (2) as the value of one variable increases, the other variable value decreases.