Practical Statistics for Experimental Biologists in RStudio

Caroline Walter

August 2020

Contents

1 Background

This course will be reviewing some practical statistics work in the R computing language useful in experimental biology.

Below are the necessary libraries for the statistical work we will be doing.

```
#Loading libraries
library(caret)
library(tidyverse)
library(dplyr)
library(RDocumentation)
library(psych)
library(car)
library(pwr)
```

2 Contingency Tables

Contingency tables (also called crosstabs or two-way tables) are used in statistics to summarize the relationship between several categorical variables. A contingency table is a special type of frequency distribution table, where two variables are shown simultaneously.

- create an example data set

```
example_data <- data.frame(x1 = sample(letters[1:5], 20, replace=TRUE),
                           x2 = sample(LETTERS[1:5], 20, replace = TRUE))
```

- have a look at the data set

```
print.data.frame(example_data)
```

```
##    x1 x2
## 1   d  D
## 2   c  D
## 3   c  A
## 4   b  A
## 5   c  E
## 6   c  E
## 7   c  D
## 8   e  C
## 9   b  E
## 10  a  D
## 11  e  D
## 12  e  E
## 13  a  B
## 14  d  D
## 15  c  C
## 16  a  E
## 17  a  C
## 18  b  C
## 19  a  D
## 20  d  C
```

- create a table

```
my_table_0 <- table(example_data$x1, example_data$x2)
print.table(my_table_0)
```

```
##
##     A B C D E
##   a 0 1 1 2 1
##   b 1 0 1 0 1
##   c 1 0 1 2 2
##   d 0 0 1 2 0
##   e 0 0 1 1 1
```

- if we want to have row and column totals

```
my_table_01<- addmargins(my_table_0)
print.table(my_table_01)
```

```
##
##         A  B  C  D  E Sum
##   a     0  1  1  2  1   5
##   b     1  0  1  0  1   3
##   c     1  0  1  2  2   6
##   d     0  0  1  2  0   3
##   e     0  0  1  1  1   3
##   Sum   2  1  5  7  5  20
```

- convert it to a dataframe

```
my_table_1 <- as.data.frame.matrix(my_table_0)
```

- have a look at the table

```
print.data.frame(my_table_1)
```

```
##   A B C D E
## a 0 1 1 2 1
## b 1 0 1 0 1
## c 1 0 1 2 2
## d 0 0 1 2 0
## e 0 0 1 1 1
```

- to have a table of proportions based on rows, and convert it to a dataframe

```
my_table_2 <- prop.table(my_table_0, margin = 1) %>%
  as.data.frame.matrix()
```

- have a look at the table

```
print.data.frame(my_table_2, digits = 2)
```

```
##      A   B    C    D    E
## a 0.00 0.2 0.20 0.40 0.20
## b 0.33 0.0 0.33 0.00 0.33
## c 0.17 0.0 0.17 0.33 0.33
## d 0.00 0.0 0.33 0.67 0.00
## e 0.00 0.0 0.33 0.33 0.33
```

- to have a table of proportions based on columns

```
my_table_3 <- prop.table(my_table_0, margin=2)%>%
  as.data.frame.matrix()
```

- have a look at the table

```
print.data.frame(my_table_2, digits = 2)
```

```
##      A   B    C    D    E
## a 0.00 0.2 0.20 0.40 0.20
## b 0.33 0.0 0.33 0.00 0.33
## c 0.17 0.0 0.17 0.33 0.33
## d 0.00 0.0 0.33 0.67 0.00
## e 0.00 0.0 0.33 0.33 0.33
```

3 Confidence Intervals and P-values

- P-values quantify the probability of data being as or more extreme than the data in hand, were the null hypothesis true.

- In statistics, the p-value is the probability of obtaining results as extreme as the observed results of a statistical hypothesis test, assuming that the null hypothesis is correct.
- The p-value is used as an alternative to rejection points to provide the smallest level of significance at which the null hypothesis would be rejected.
- A smaller p-value means that there is stronger evidence in favor of the alternative hypothesis.
- The confidence interval (CI) is a range of values that's likely to include a population value with a certain degree of confidence. It is often expressed a % whereby a population means lies between an upper and lower interval.
- The 95% confidence interval is a range of values that you can be 95% certain contains the true mean of the population. As the sample size increases, the range of interval values will narrow, meaning that you know that mean with much more accuracy compared with a smaller sample.
- Confidence Intervals are more informative than p-values. We can reject a null hypothesis if it lies outside of the confidence interval.

4 The Correlation Coefficient (r)

The main result of a correlation is called the correlation coefficient ("r"). It ranges from -1 to 1. The closer r is to -1 or 1, the more closely the two variables are (inversely) related. If r is close to 0, it means there is no relationship between the variables.

Correlation coefficients are used in statistics to measure how strong a relationship is between two variables.

- When r=0, the two variables are independent.
- When r= 1, the two variables have a strong positive correlation or relationship.
- When r = -1, the two variables have a strong negative correlation/relationship.

5 The Coefficient of Determination (r squared)

R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.
- Correlation explains the strength of th relationship between an independent and dependent variable, r-squared explains to what extent the variance of one variable explains the variance of the second variable.

R-squared is always between 0 and 1. It is commonly interpreted as the fraction of variance in y explained by x (or the other way around). It is a goodness-of-fit measure for linear regression models. This statistics indicates a percentage of the variance in the dependent variable that the independent variables explain collectively. T-squared measures the strength of the relationship between your model and the dependent variable on a convenient 0-100% scale.

After fitting a linear regression model to the data (by using a least-squares regression line), we need to determine how well the model fits the data.

6 Sensitivity and Specificity

In medical diagnosis, test sensitivity is the ability of a test to correctly identify those with the disease (true positive rate), whereas test specificity is the ability of the test to correctly identify those without the disease (true negative rate).

For the AIP test: AIP (Acute Intermediate Porphyria) is a rare metabolic disorder.

- Sensitivity: (AIP example) the probability of testing positive given that the subject has the disease

$Sensitivity = p(test^{+}|disease^{+}) = 0.82$

$p(test^{+}|disease^{+}) = \frac{TP}{TP+FN} = \frac{82}{82+18} = 0.82$

- Specificity: (AIP example) the probability of a negative test given that the subject does not have the disease

$Specificity = p(test^{-}|disease^{-}) = 0.963$

- here are two helpful youtube videos:
  - https://www.youtube.com/watch?v=9f5XgjWpzi0
  - https://www.youtube.com/watch?v=UsOv0DcXk6w

```
devtools::install_github("datacamp/RDocumentation")

data01 <- factor(c("A", "B", "B", "B"))
data02 <- factor(c("A", "B", "B", "B"))

ref01 <- factor(c("B", "B", "B", "B"))
ref02 <- factor(c("B", "A", "B", "B"))
```

7 Positive Predictive Value (PPV)

Positive Predictive Value is the probability that subjects with a positive screening test truly have the disease

- TP: True Positive
- FP: False Positive
- TN: True Negative
- FN: False Negative

$p(disease^{+}|test^{+}) = \frac{TP}{TP+FP} = \frac{82}{82+36{,}996} = 0.0022$

- Even if you test positive, the probablity of you having AIP is still very low.
- PPV is often far less than sensitivity in screening tests for rare diseases.

```
table(data01, ref01)
```

```
##        ref01
## data01 B
##      A 1
##      B 3
```

```r
sensitivity(data01, ref01)
```

```
## [1] 0.75
```

```r
posPredValue(data01, ref01)
```

```
## [1] NA
```

```r
table(data02, ref02)
```

```
##       ref02
## data02 A B
##      A 0 1
##      B 1 2
```

```r
sensitivity(data02, ref02)
```

```
## [1] 0
```

```r
posPredValue(data02, ref02)
```

```
## [1] 0
```

```r
data03 <- factor(c("A", "B", "B", "B"))
data04 <- factor(c("B", "B", "B", "B"))

ref03 <- factor(c("B", "B", "B", "B"), levels = c("A", "B"))
ref04 <- factor(c("B", "A", "B", "B"))
```

8 Negative Predictive Value (NPV)

Negative predictive value is the probability that subjects with a negative screening test truly do not have the disease.

$p(disease^{-}|test^{-})$

```r
table(data03, ref03)
```

```
##       ref03
## data03 A B
##      A 0 1
##      B 0 3
```

```r
specificity(data03, ref03)
```

```
## [1] 0.75
```

```r
negPredValue(data03, ref03)
```

```
## [1] NA
```

```r
table(data04, ref04)
```

```
##       ref04
## data04 A B
##      B 1 3
```

```r
specificity(data04, ref04)
```

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

```r
negPredValue(data04, ref04)
```

```
## [1] NaN
```

```r
if(!isTRUE(all.equal(sensitivity(data01, ref01), .75))) stop("error in sensitivity test 1")
if(!isTRUE(all.equal(sensitivity(data02, ref02), 0))) stop("error in sensitivity test 2")
ref03 <- factor(c("B", "B", "B", "B"))
if(!is.na(sensitivity(data02, ref03, "A"))) stop("error in sensitivity test3")

options(show.error.messages = FALSE)
test1 <-try(sensitivity(data02, as.character(ref03)))
if(grep("Error", test1) != 1)
  stop("error in sensitivity calculation - allowed non-factors")
options(show.error.messages = TRUE)

ref03 <- factor(c("B", "B", "B", "B"), levels = c("A", "B"))

if(!isTRUE(all.equal(specificity(data03, ref03), .75))) stop("error in specificity test 1")

if(!isTRUE(all.equal(specificity(data04, ref04), 1.00))) stop("error in specificity test 2")

if(!is.na(specificity(data01, ref01, "A"))) stop("error in specificity test3")

options(show.error.messages = FALSE)
test1 <-try(specificity(data04, as.character(ref03)))
if(grep("Error", test1) != 1)
  stop("error in specificity calculation - allowed non-factors")
options(show.error.messages = TRUE)
```

9 Prevalence

Prevalence is the fraction of individuals in a population who have a disease.

- If a subject's sibling has AIP, there is a 50% chance that they do too.

$prevalence = p(disease^{+}) = 0.50$

```
set.seed(0934)
Data.All.df.2008 <- data.frame(FSA = sample(c("N8N", "N8R", "B3L", "P1H"), 50, T),
                               Lyme = sample(0:1, 50, T),
                               stringsAsFactors = F)
```

- First 10 observations:

```
head(Data.All.df.2008)
```

```
##   FSA Lyme
## 1 N8N    1
## 2 P1H    1
## 3 N8N    0
## 4 P1H    0
## 5 N8N    1
## 6 N8N    1
```

- Prevalence can be calculated as the number of positive diagnoses divided by the total number of observations, i.e. `sum(Lyme)/n()`.

```
Data.All.df.2008 %>%
  group_by(FSA) %>%
  summarise(Prevalence = sum(Lyme)/n())
```

```
## # A tibble: 4 x 2
##   FSA   Prevalence
##   <chr>      <dbl>
## 1 B3L        0.778
## 2 N8N        0.571
## 3 N8R        0.583
## 4 P1H        0.467
```

10 Frequentist vs. Bayesian Statistics

- Frequentist Statistics (aka classical statistics) focuses on likelihood. It avoids calculations involving prior odds, and therefore yields results that are prone to misinterpretation due to the base rate fallacy. Frequentist statistics is used heavily in biological research. It can still be useful and informative if you know exactly what to watch out for
  - Base Rate Fallacy: If presented with related base rate information (generic, general information) and specific information (information pertaining only to a certain case), the mind tends to ignore the former and focus of the latter. (Individuals tend to focus on the specific information pertaining to that single case, instead of focusing on the generic information given.)
  - Iron Law of Frequentist Statistics: Never compute the probability of a hypothesis.
  - Frequentist: $p(data|hypothesis)$
- Bayesian Statistics explicitly accounts for prior odds. It focuses on computing posterior probabilities and requires prior information that is often hard to quantify. It is central to the modern machine learning and more advanced areas of quantitative biology. Experimental researchers in biology tend not to use Bayesian statistics.
  - Bayesian: $p(hypothesis|data)$

11 Standard Error of the Mean (SEM)

The standard error of the mean (SEM) is a statistical term that measures the accuracy with which a sample distribution represents a population by using the standard deviation. In statistics, a sample mean deviates from the actual mean of a population - this deviation is the standard error of the mean (SEM)

$SEM = \sqrt{q(1-q)/N}$

```
Input =("
Stream                      Fish
 Mill_Creek_1                 76
 Mill_Creek_2                102
 North_Branch_Rock_Creek_1    12
 North_Branch_Rock_Creek_2    39
 Rock_Creek_1                 55
 Rock_Creek_2                 93
 Rock_Creek_3                 98
 Rock_Creek_4                 53
 Turkey_Branch               102
")

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

- calculate standard error manually

```
sd(Data$Fish, na.rm=TRUE) /
  sqrt(length(Data$Fish[!is.na(Data$Fish)]))
```

```
## [1] 10.69527
```

- install psych package in console and use describe function from package for standard error. This function also works on dataframes.

```
describe(Data$Fish,
         type=2)
```

```
##    vars n mean    sd median trimmed  mad min max range  skew kurtosis   se
## X1    1 9   70 32.09     76      70 34.1  12 102    90 -0.65    -0.69 10.7
```

12 Fisher's Exact Test

Fisher's exact test is a statistical significance test used in the analysis of contingency tables and in place of chi square test in 2 by 2 tables, especially in cases of small samples. The test is useful for categorical data that result from classifying objects in two different ways; it is used to examine the significance of the association (contingency) between the two kinds of classification.

Example: A British woman claimed to be able to distinguish whether milk or tea was added to the cup first. To test, she was given 8 cups of tea, in four of which milk was added first. The null hypothesis is that there is no association between the true order of pouring and the woman's guess, the alternative hypothesis is that there is a positive association (that the odds ratio is greater than 1).

Syntax: fisher.test(x, y = NULL, workspace = 200000, hybrid = FALSE, hybridPars = c(expect = 5, percent = 80, Emin = 1), control = list(), or = 1, alternative = "two.sided", conf.int = TRUE, conf.level = 0.95, simulate.p.value = FALSE, B = 2000)

```
TeaTasting <-
  matrix(c(3,1,1,3),
         nrow=2,
         dimnames = list(Guess = c("Milk", "Tea"),
                         Truth = c("Milk", "Tea")))
fisher.test(TeaTasting, alternative = "greater")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  TeaTasting
## p-value = 0.2429
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  0.3135693       Inf
## sample estimates:
## odds ratio
##   6.408309
```

13 Gaussian/Normal Distributions

Gaussian distribution ("the normal distribution") is ubiquitous in statistics. This distribution is based on the mean μ and standard deviation σ of the data.

$x \sim Normal(\mu, \sigma^2)$

- The Central Limit Theorem states that the population of all possible samples of size n from a populations= with mean μ and variance (σ squared) approaches a normal distribution when n (sample size) approaches infinity. The central limit theorem makes the normal distribution extremely relevant.

The parameters of a statistical model that has been fit to a large dataset will have lingering uncertainty, but this uncertainty will very often be approximately normally distributed. This is why statisticians so often assume that experimental measurements follow normal distributions.

$\theta =$ model parameter inferred from data $\theta \sim Normal(\mu, \sigma^2)$

The goal of statistical inference is to determine the mean and standard deviation of this distribution μ: best estimate of θ, denoted θ σ: lingering uncertainty in θ: affects confidence interval

- Example: Here, this normal/gaussian distribution results shows that the percentage of students scoring an 84% or higher in the college entrance exam is 21.5%

```
pnorm(84, mean = 72, sd = 15.2, lower.tail = FALSE)
```

```
## [1] 0.2149176
```

14 Bernoulli Distribution

Bernoulli Distribution: a discrete distribution having two possible outcomes labeled by n=0 and n=1 in which n=1("success") occurs with probability p and n=0 ("failure") occurs with probability q = 1-p, where 0<p<1. Describes probabilities for a binary variable. (Biased coin example); when the probabilities sum up to 100%. Example: A Biased Coin Biased coins are modeled using a Bernoulli distribution, which describes probabilities for a binary variable.

15 Binomial Tests

A binomial test compares the number of successes observed in a given number of trials with a hypothesized probability of success. The test has the null hypothesis that the real probability of success is equal to some value denoted p, and the alternative hypothesis that is not equal to p. The test can also be performed with a one-sided alternative hypothesis that the real probability of success is either greater than p or that it is less than p.

A one-tailed test with a significance level of 0.05 will be used. You roll the die 300 times and throw a total of 60 sixes. We cannot reject the null hypothesis that the probability of rolling a six is 1/6. This means that there is no evidence to prove that the die is not fair.

- Syntax: `binom.test(nsuccesses, ntrials, p)`

```
binom.test(60, 300, 1/6, alternative = "greater")
```

```
##
##  Exact binomial test
##
## data:  60 and 300
## number of successes = 60, number of trials = 300, p-value = 0.07299
## alternative hypothesis: true probability of success is greater than 0.1666667
## 95 percent confidence interval:
##  0.1626847 1.0000000
## sample estimates:
## probability of success
##                    0.2
```

16 Chi Square Tests

Chi Square tests are used to determine if two categorical variables have a significant correlation between them. The two variables are selected from the same population. (male/female, red/green, yes/no). The chi-square statistic is commonly used for testing relationships between categorical variables.

$\chi^2 = \Sigma \frac{(obersverd-expected)^2}{expected}$

- Syntax: `chisp.test(data)`
- importing data, have a look at the table

```
data_frame<- read.csv("https://goo.gl/j6lRXD")
table(data_frame$treatment, data_frame$improvement)
```

```
##
##                 improved not-improved
##   not-treated         26           29
##   treated             35           15
```

- Example A: Here, we get a chi-squared value of 5.5569. Since we get a p-value less than the significance level of 0.05, we reject the null hypothesis and conclude that the two variables are in fact dependent.

```r
chisq.test(data_frame$treatment, data_frame$improvement, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  data_frame$treatment and data_frame$improvement
## X-squared = 5.5569, df = 1, p-value = 0.01841
```

- Example B: Using the "mtcars" R package, we run a chi-squared test. Here, we get a high chi-squared value and a p-value of less than the 0.05 significance level. Therefore, we can reject the null hypothesis and conclude that carb and cyl have a significant relationship.

```r
data("mtcars")
table(mtcars$carb, mtcars$cyl)
```

```
##
##      4 6 8
##   1  5 2 0
##   2  6 0 4
##   3  0 0 3
##   4  0 4 6
##   6  0 1 0
##   8  0 0 1
```

```r
chisq.test(mtcars$carb, mtcars$cyl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  mtcars$carb and mtcars$cyl
## X-squared = 24.389, df = 10, p-value = 0.006632
```

17 Z-Tests

Two-Proportions Z-Test: is used to compare two observed proportions.
Example: We have two groups of individuals: * Group A with lung cancer; n=500 * Group B, healthy individuals; n=500

The number of smokers in each group is as follows: * Group A with lung cancer; n=500, 490 smokers, pA = 490/500 = 98 * Group B, healthy individuals; n=500, 400 smokers, pB = 400/500 = 80

In this setting: * The overall proportion of smokers is $\frac{490+400}{500+500} = 89$ * The overall proportion of non-smokers is $q = 1 - p = 11$

Formula of the test statistic: The test statistic (also knows as z-test) can be calculated as follows: $z = \frac{pA-pB}{\sqrt{\frac{pq}{nA}+\frac{pq}{nB}}}$ where, * pA is the proportion observed in group A with size nA * pB is the proportion observed in group B with size nB * p and q are the overall proportions * If |z| < 1.96, then the difference is not significant at 5% * If |z| >= 1.96, then the difference is significant at 5% * The significance level (p-value) corresponding to the z-statistic can be read in the z-table. Lets see how to compute it in R.

** The Fisher Exact Test is an excellent non-parametric technique for comparing proportions, when the two independent samples are small in size.

Syntax: prop.test(x, n, p=NULL, alternative = "two.sided", correct=TRUE)

- x: a vector of counts of successes
- n: a vector of count trials
- alternative: a character string specifying the alternativ hypothesis
- correct: a logical indicating whether Yates' continuity correction should be applied where possible

```
res <- prop.test(x=c(490, 400), n=c(500, 500))
res
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(490, 400) out of c(500, 500)
## X-squared = 80.909, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.1408536 0.2191464
## sample estimates:
## prop 1 prop 2
##   0.98   0.80
```

The function returns the value of Pearson's chi-squared test statistic, a p-value, a 95% confidence interval, and an estimated probability of success (the proportion of smokers in the two groups.)

If you want to test whether the observes proportion of smokers in group A (pA) is less than the observed proportion of smokers in group B (pB):

```
prop.test(x = c(490, 400), n = c(500, 500),
           alternative = "less")
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(490, 400) out of c(500, 500)
## X-squared = 80.909, df = 1, p-value = 1
## alternative hypothesis: less
## 95 percent confidence interval:
##  -1.0000000  0.2131742
## sample estimates:
## prop 1 prop 2
##   0.98   0.80
```

If you want to test whether the observed proportion of smokers in group A (pA) is greater than the observed proportion of smokers in group (pB):

```r
prop.test(x = c(490, 400), n = c(500, 500),
              alternative = "greater")
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(490, 400) out of c(500, 500)
## X-squared = 80.909, df = 1, p-value < 2.2e-16
## alternative hypothesis: greater
## 95 percent confidence interval:
##  0.1468258 1.0000000
## sample estimates:
## prop 1 prop 2
##   0.98   0.80
```

The p-value of the test is 2.36310^{-19}, which is less than the significance level alpha = 0.05. We can conclude that the proportion of smokers is significantly different in the two groups with a p-value = 2.36310^{-19}

18 T-Tests

A t-test is a statistical test which is used to compare the mean of two groups of samples. It is therefore to evaluate whether the means of the two sets of data are statistically significantly different from each other.

The `t.test()` function produces a variety of t-tests. Unlike most statistical packages, the default assumes unequal variance and applies the Welsh degrees of freedom modification.

A one-sample t-test is used to compare the mean of a population with a theoretical value. An unpaired two sample t-test is used to compare the mean of two independent samples. A paired t-test is used to compare the means between two related groups of samples. The formula of a t-test depends on the mean and standard deviation of the data being compared.

- One-Sample t-test: based on the theoretical mean(μ), the set of values with size n, the mean μ, and the standard deviation σ. The degrees of freedom is found with (df=n-1).

** Degrees of Freedom are the number of independent values that a statistical analysis can estimate. You also think of it as the number of values that are free to vary as you estimate parameters.

$t = \frac{m-\mu}{\frac{\sigma}{\sqrt{n}}}$

- Independent two-sample t-test Let A and B represent the two groups to compare. Let mA and mB represent the means of groups A and B, respectively. Let nA and nB represent the sizes of group A and B, respectively.

$t = \frac{mA-mB}{\sqrt{\frac{S^2}{nA}+\frac{S^2}{nB}}}$

S^2 is an estimator of the common variance of the two samples.

- Welch's T-Tests (or unequal variances t-test), is a two-sample location test which is used to test the hypothesis that the two populations have unequal means.

- independent two-sample t-test, where y is a numeric and x is a binary factor

```
x=rnorm(10)
y=rnorm(10)
t.test(y,x)  #Welch's t-test
```

```
##
##  Welch Two Sample t-test
##
## data:  y and x
## t = 0.055054, df = 12.403, p-value = 0.957
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.9643282  1.0145094
## sample estimates:
##    mean of x    mean of y
##  0.018022788 -0.007067778
```

- one sample t-test, with a null hypothesis (Ho) $\mu = 3$

```
t.test(y,mu=3)
```

```
##
##  One Sample t-test
##
## data:  y
## t = -7.1566, df = 9, p-value = 5.328e-05
## alternative hypothesis: true mean is not equal to 3
## 95 percent confidence interval:
##  -0.9245609  0.9606065
## sample estimates:
##  mean of x
## 0.01802279
```

- Mann-Whitney-Wilcoxon Test Two data samples are independent if they come from distinct populations and the samples do not affect each other. Using the Mann-Whitney-Wilcoxon Test, we can decide whether the population distribution are identical without assuming them to follow the normal distribution.

Example: At the 0.05 significance level, we conclude that the gas mileage data of manual and automatic transmissions in R package "mtcars" are nonidentical populations.

```
mtcars$mpg
```

```
##  [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
wilcox.test(mpg ~ am, data=mtcars)
```

```
## Warning in wilcox.test.default(x = c(21.4, 18.7, 18.1, 14.3, 24.4, 22.8, :
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  mpg by am
## W = 42, p-value = 0.001871
## alternative hypothesis: true location shift is not equal to 0
```

19 ANOVA tests (Analysis of Variance)

ANOVA is a statistical test for estimating how a quantitative dependent variable changes according to the levels of one or more categorical independent variables. ANOVA tests whether there is a difference in means of the groups at each level of the independent variable.

Example: You can check the levels of poison with the following. You should see three character values because you convert them in factor with the mutate verb.

```
PATH<- "https://raw.githubusercontent.com/guru99-edu/R-Programming/master/poisons.csv"
df <- read.csv(PATH) %>%
  select(-X) %>%
  mutate(poison = factor(poison, ordered = TRUE))
glimpse(df)
```

```
## Rows: 48
## Columns: 3
## $ time   <dbl> 0.31, 0.45, 0.46, 0.43, 0.36, 0.29, 0.40, 0.23, 0.22, 0.21, ...
## $ poison <ord> 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 1, 1, 1, 1, 2, 2, 2, 2, ...
## $ treat  <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", ...
```

```
levels(df$poison)
```

```
## [1] "1" "2" "3"
```

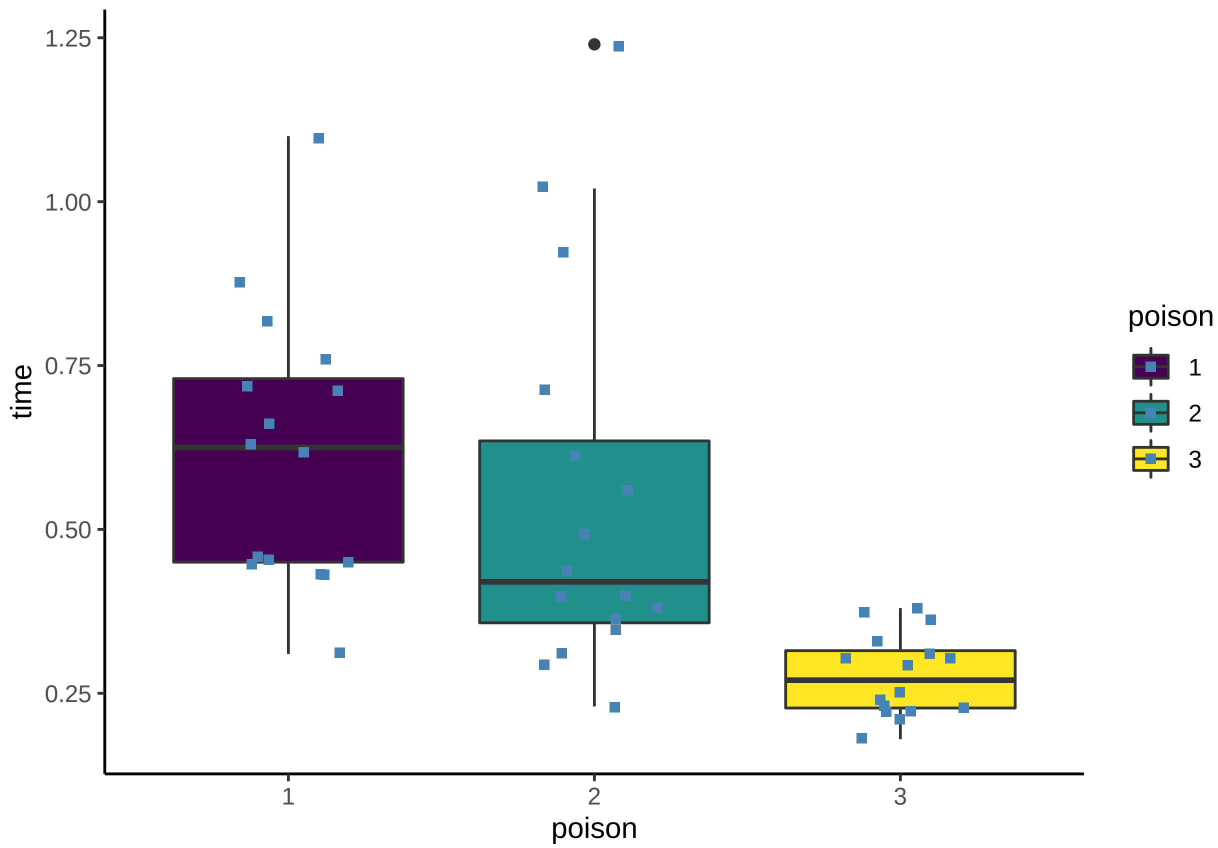
- Compute the mean and the standard deviation

```
df %>%
  group_by(poison)%>%
  summarise(
    count_poison = n(),
    mean_time = mean(time, na.rm = TRUE),
    sd_time = sd(time, na.rm=TRUE)
  )
```

```
## # A tibble: 3 x 4
##   poison count_poison mean_time sd_time
##   <ord>         <int>     <dbl>   <dbl>
## 1 1                16     0.618  0.209
## 2 2                16     0.544  0.289
## 3 3                16     0.276  0.0623
```

- Graphically check if there is a difference between the distribution.

```r
ggplot(df, aes(x = poison, y = time, fill = poison)) +
  geom_boxplot() +
  geom_jitter(shape = 15,
              color = "steelblue",
              position = position_jitter(0.21)) +
  theme_classic()
```



* Run the one-way ANOVA test with the command aov.

- Syntax: `aov(formula, data)`

```r
anova_one_way <- aov(time~poison, data = df)
summary(anova_one_way)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## poison       2  1.033  0.5165   11.79 7.66e-05 ***
## Residuals   45  1.972  0.0438                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- ANOVA two-way test

```r
anova_two_way <- aov(time~poison + treat, data = df)
summary(anova_two_way)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)    
## poison       2 1.0330  0.5165   20.64 5.7e-07 ***
## treat        3 0.9212  0.3071   12.27 6.7e-06 ***
## Residuals   42 1.0509  0.0250                    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

20 Assessing Goodness-of-Fit in a Regression Model

Linear Regression identifies the equation that produces the smallest difference between all of the observed values and their fitted values. Linear regression finds the smallest sum of squared residuals that is possible for the dataset (least-squares regression line).

R-Squared is the percentage of the dependent variable variation that a linear model explains:

$R^2 = \frac{\text{variance explained by model}}{\text{total variance}}$

- 0% or an r-squared close to 0.000 represents a model that does not explain any of the variation in the response (dependent) variable around its mean. The mean of the dependent variable predicts the dependent variable as well as the regression model.
- 100% or an r-squared of 1.00 represents a model that explains all of the variation in the response variable around its mean.
- Example: We apply the `lm` function to a formula that describes the variable `eruptions` by the variable `waiting`, and save the linear regression model in a new variable 'eruption.lm'.

```
eruption.lm = lm(eruptions ~ waiting, data=faithful)
summary(eruption.lm)$r.squared
```

```
## [1] 0.8114608
```

```
summary(eruption.lm)
```

```
## 
## Call:
## lm(formula = eruptions ~ waiting, data = faithful)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.29917 -0.37689  0.03508  0.34909  1.19329 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.874016   0.160143  -11.70   <2e-16 ***
## waiting      0.075628   0.002219   34.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4965 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108 
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

The coefficient of determination (r^2) of the simple linear regression model for the data set below is 0.81146. This means with an r-squared of 0.81146, then approximately 81.15% of the observed variation can be explained by the model's inputs.

21 Power Analysis

The power of a statistical test is the probability that the test will reject a false null hypothesis. Power analysis allows us to determine the sample size required to detect an effect of a given size with a given degree of confidence. It also allows us to determine the probability of detecting an effect of a given size with a given level of confidence, under sample size constraints.

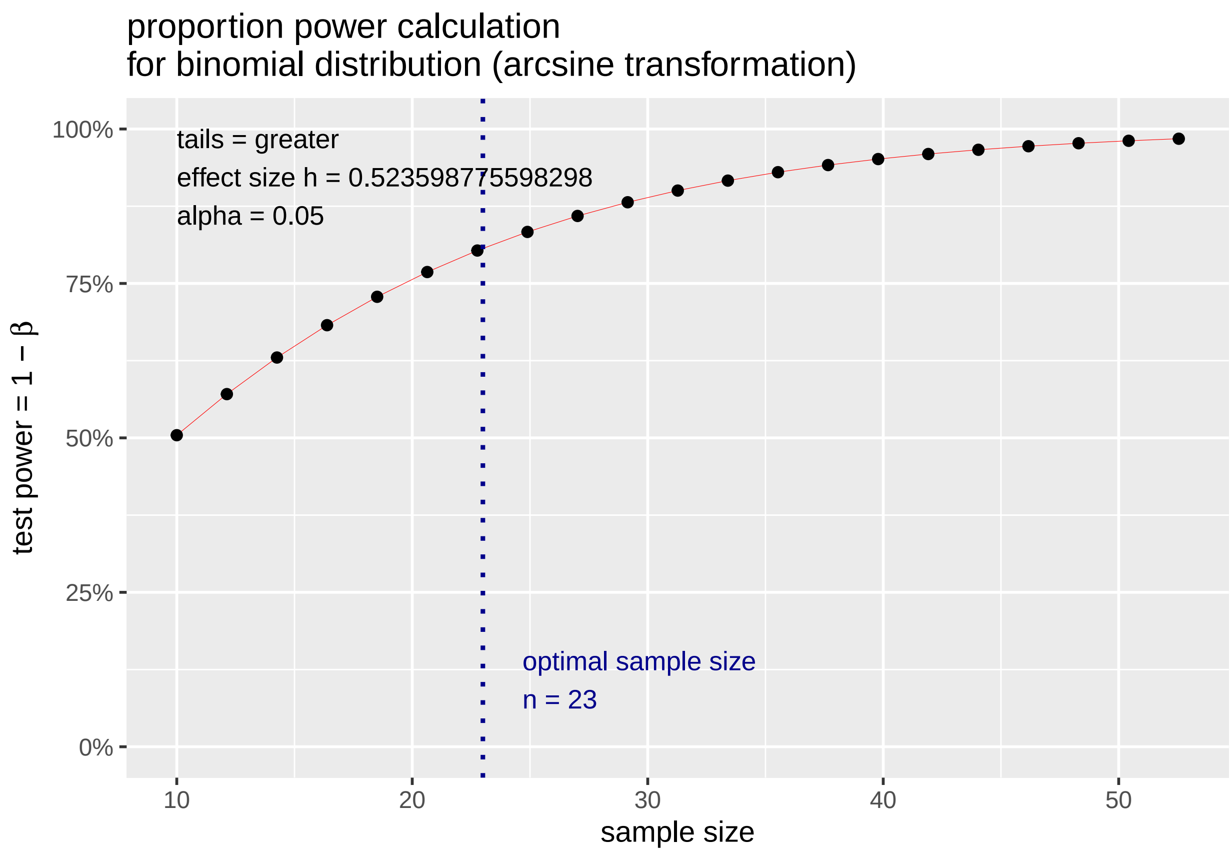
- Example: The function tells us we should flip the coin 22.55126 times, which would round up to 23. Always round sample size estimates up in power analyses. If we are correct that our coin lands heads 75% of the time, we need to flip at least 23 times to have an 80% chance of correctly rejecting the null hypothesis at the 0.05 significance level.
- Available tests (code chunk not evaluated):

```
#  two proportions (equal n)
pwr.2p.test
#two proportions(unequal n)
pwr.2p2n.test
# balanced one-way ANOVA
pwr.anova.test
# chi-square test
pwr.chisq.test
# general linear model
pwr.f2.test
# proportion (one sample)
pwr.p.test
# correlation
pwr.r.test
#  t-tests (one sample, 2 sample, paired)
pwr.t.test
# t-test (two samples with unequal n)
pwr.t2n.test
```

```
pwr.p.test(h=ES.h(p1=0.75, p2=0.50),
           sig.level = 0.05,
           power = 0.80,
           alternative = "greater")
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.5235988
##               n = 22.55126
##       sig.level = 0.05
##           power = 0.8
##     alternative = greater
```

```r
p.out <- pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.50),
                    sig.level = 0.05,
                    power = 0.80,
                    alternative = "greater")
plot(p.out)
```



Example: What is the power of our test if we flip the coin 40 times and lower our Type I error tolerance to 0.01? Notice we leave out the power argument, add n = 40, and change sig.level = 0.01:

The power of our test is now about 84%. If we wish to assume a "two-sided" alternative, we can simply leave it out of the function. Notice how our power estimate drops below 80% when we do this.

```r
pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.50),
           sig.level = 0.01,
           n = 40,
           alternative = "greater")
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.5235988
##               n = 40
##       sig.level = 0.01
##           power = 0.8377325
##     alternative = greater
```

```
pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.50),
           sig.level = 0.01,
           n = 40)
```

```
##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.5235988
##               n = 40
##       sig.level = 0.01
##           power = 0.7690434
##     alternative = two.sided
```

22 Nonlinear Regression

- Load the data

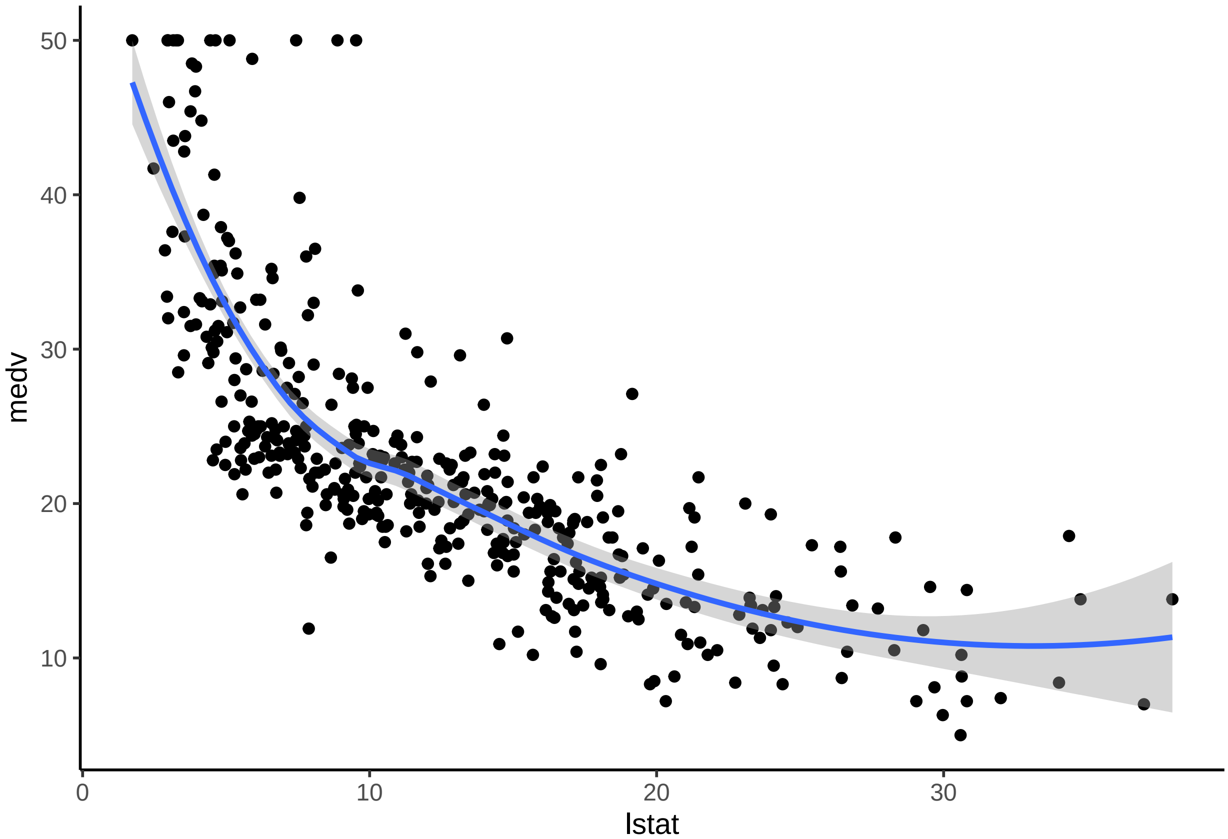
```
theme_set(theme_classic())
data("Boston", package = "MASS")
```

- Split the data into training and test set

```
set.seed(123)
training.samples <- Boston$medv %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data  <- Boston[training.samples, ]
test.data <- Boston[-training.samples, ]
```

- First, visualize the scatter plot of the medv vs lstat variables:

```
ggplot(train.data, aes(lstat, medv) ) +
  geom_point() +
  stat_smooth()
```



The standard linear regression model equation can be written as:

$medv = b_0 + b_1 \times l_{stat}$

Compute the linear regression model: * Build the model

```
model <- lm(medv ~ lstat, data = train.data)
```

- Make predictions

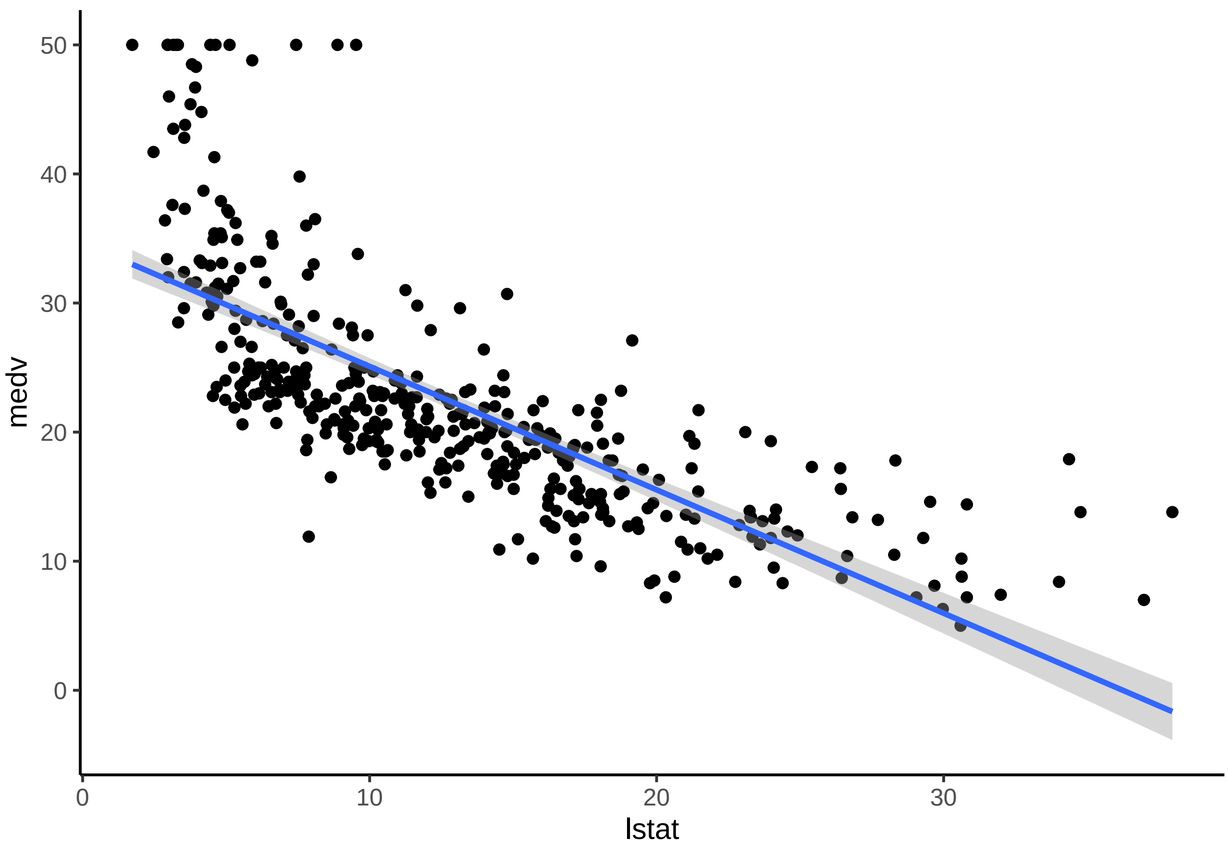
```
predictions <- model %>% predict(test.data)
```

- Model performance

```
data.frame(
  RMSE = RMSE(predictions, test.data$medv),
  R2 = R2(predictions, test.data$medv)
)
```

```
##       RMSE       R2
## 1 6.503817 0.513163
```

```
ggplot(train.data, aes(lstat, medv) ) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ x)
```



23 Polynomial Regression

Polynomial regression: adds polynomial or quadratic terms to the regression equations

```
lm(medv ~ lstat + I(lstat^2), data = train.data)
```

```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2), data = train.data)
##
## Coefficients:
## (Intercept)        lstat   I(lstat^2)
##     42.5736      -2.2673       0.0412
```

```
#or
lm(medv ~ poly(lstat, 2, raw = TRUE), data = train.data)
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 2, raw = TRUE), data = train.data)
##
## Coefficients:
##                 (Intercept)  poly(lstat, 2, raw = TRUE)1
```

```
##                      42.5736                           -2.2673
## poly(lstat, 2, raw = TRUE)2
##                       0.0412
```

```
lm(medv ~ poly(lstat, 6, raw = TRUE), data = train.data) %>%
  summary()
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 6, raw = TRUE), data = train.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1962  -3.1527  -0.7655   2.0404  26.7661
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  7.788e+01  6.844e+00  11.379  < 2e-16 ***
## poly(lstat, 6, raw = TRUE)1 -1.767e+01  3.569e+00  -4.952 1.08e-06 ***
## poly(lstat, 6, raw = TRUE)2  2.417e+00  6.779e-01   3.566 0.000407 ***
## poly(lstat, 6, raw = TRUE)3 -1.761e-01  6.105e-02  -2.885 0.004121 **
## poly(lstat, 6, raw = TRUE)4  6.845e-03  2.799e-03   2.446 0.014883 *
## poly(lstat, 6, raw = TRUE)5 -1.343e-04  6.290e-05  -2.136 0.033323 *
## poly(lstat, 6, raw = TRUE)6  1.047e-06  5.481e-07   1.910 0.056910 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.188 on 400 degrees of freedom
## Multiple R-squared:  0.6845, Adjusted R-squared:  0.6798
## F-statistic: 144.6 on 6 and 400 DF,  p-value: < 2.2e-16
```

- Build the model

```
model <- lm(medv ~ poly(lstat, 5, raw = TRUE), data = train.data)
```

- Make predictions

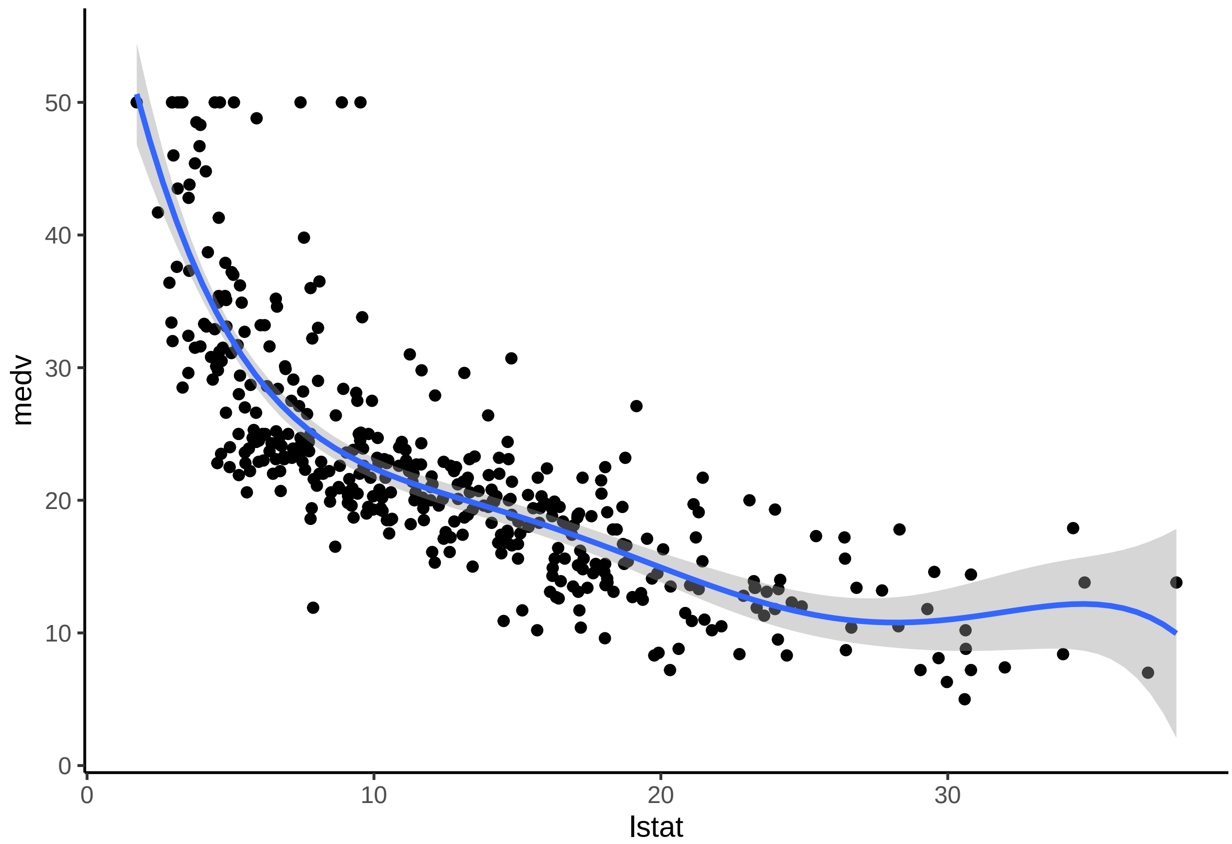
```
predictions <- model %>% predict(test.data)
```

- Model performance

```
data.frame(
  RMSE = RMSE(predictions, test.data$medv),
  R2 = R2(predictions, test.data$medv)
)
```

```
##       RMSE        R2
## 1 5.270374 0.6829474
```

```r
ggplot(train.data, aes(lstat, medv) ) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ poly(x, 5, raw = TRUE))
```



24 Log Transformation in Regression

Log transformation: when you have a non-linear relationship, you can try a logarithmic transformation of the predictor variables:

- Build the model

```r
model <- lm(medv ~ log(lstat), data = train.data)
```

- Make predictions

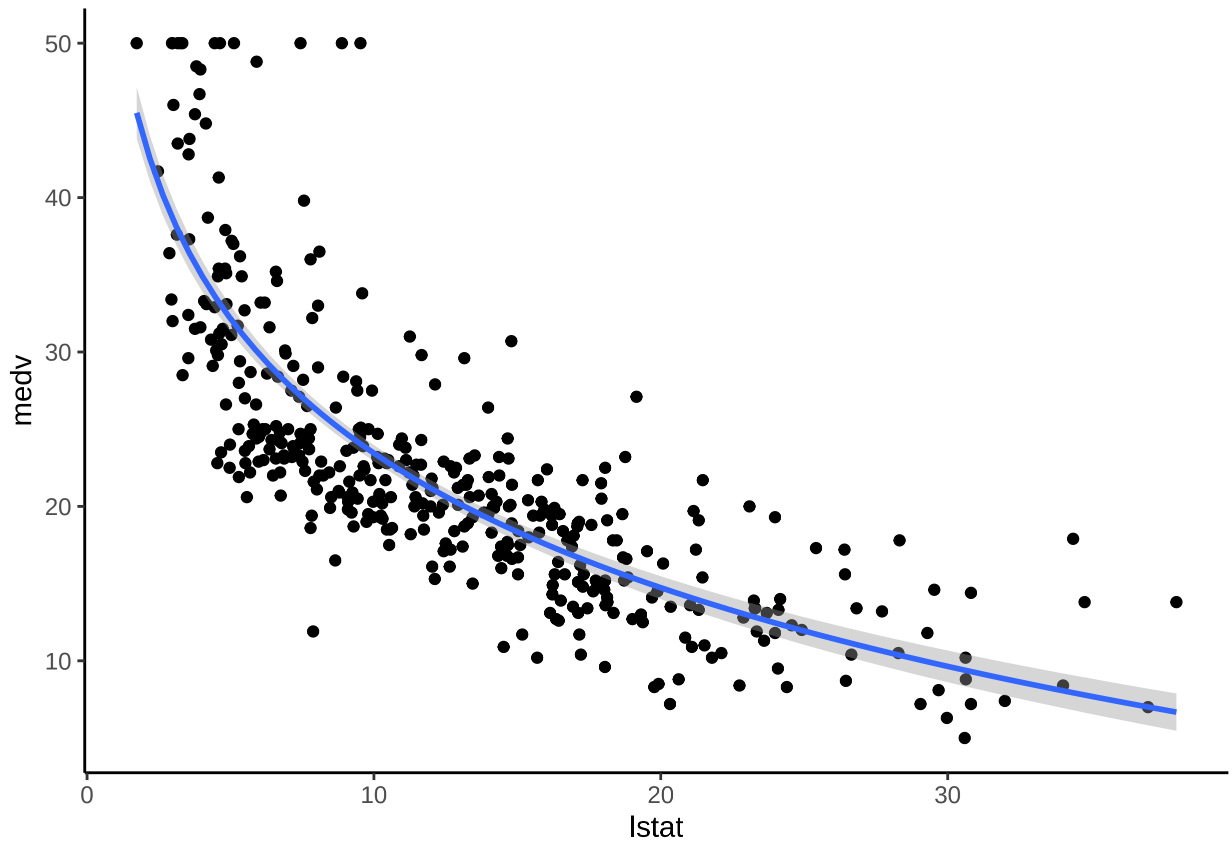
```r
predictions <- model %>% predict(test.data)
```

- Model performance

```r
data.frame(
  RMSE = RMSE(predictions, test.data$medv),
  R2 = R2(predictions, test.data$medv)
)
```

```
##       RMSE        R2
## 1 5.467124 0.6570091
```

```r
ggplot(train.data, aes(lstat, medv) ) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ log(x))
```



R Diagnostic Plots take a look at the dataframe

```r
data(women)
women
```

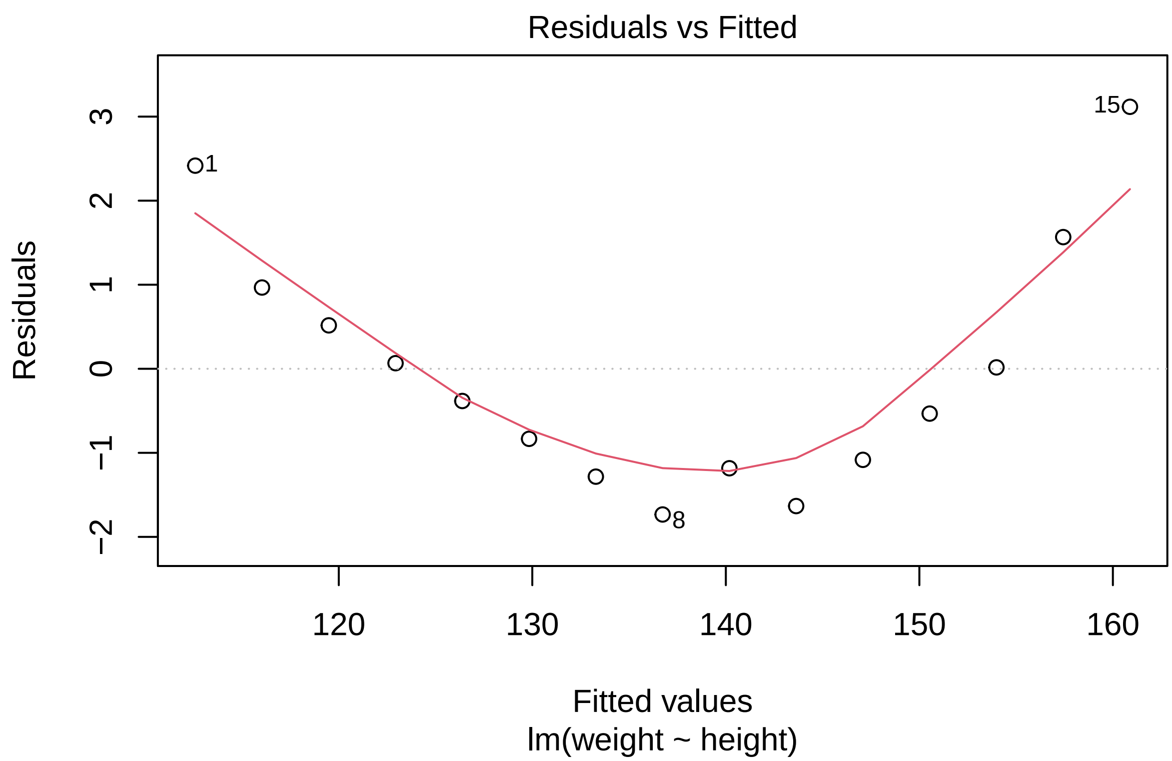
```
##    height weight
## 1      58    115
## 2      59    117
## 3      60    120
## 4      61    123
## 5      62    126
## 6      63    129
## 7      64    132
## 8      65    135
## 9      66    139
## 10     67    142
## 11     68    146
## 12     69    150
## 13     70    154
```

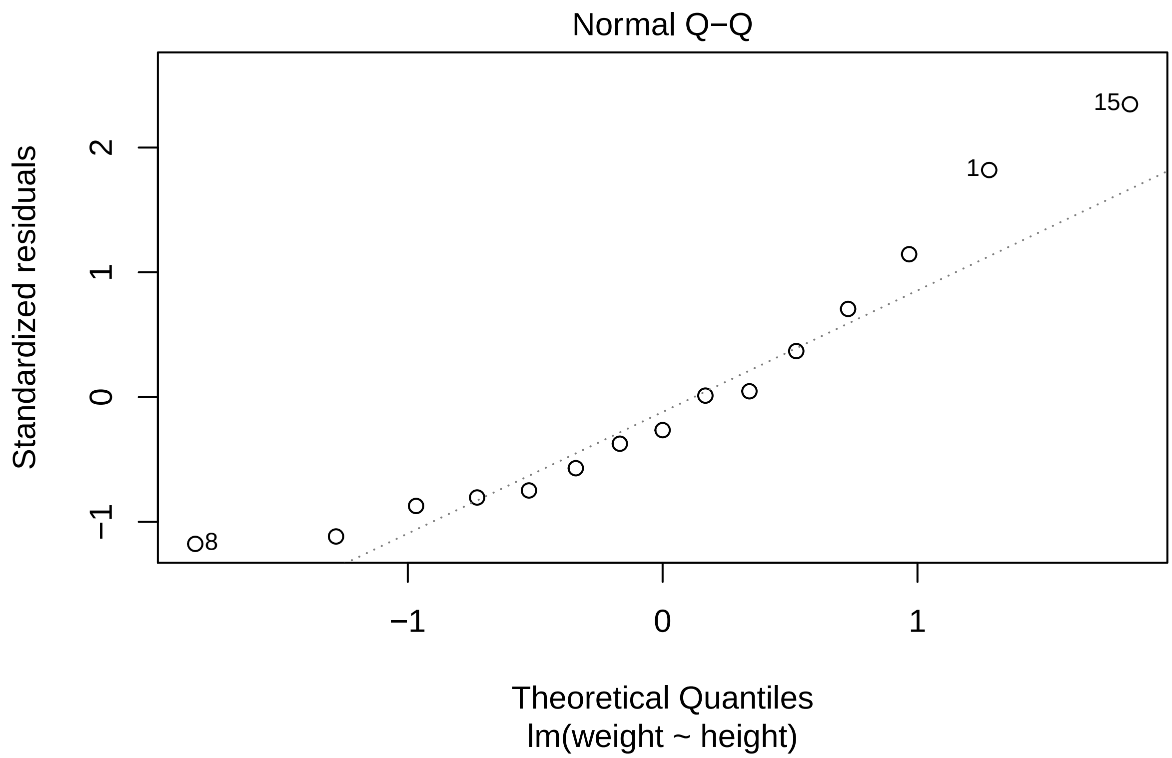
```
## 14      71     159
## 15      72     164
```

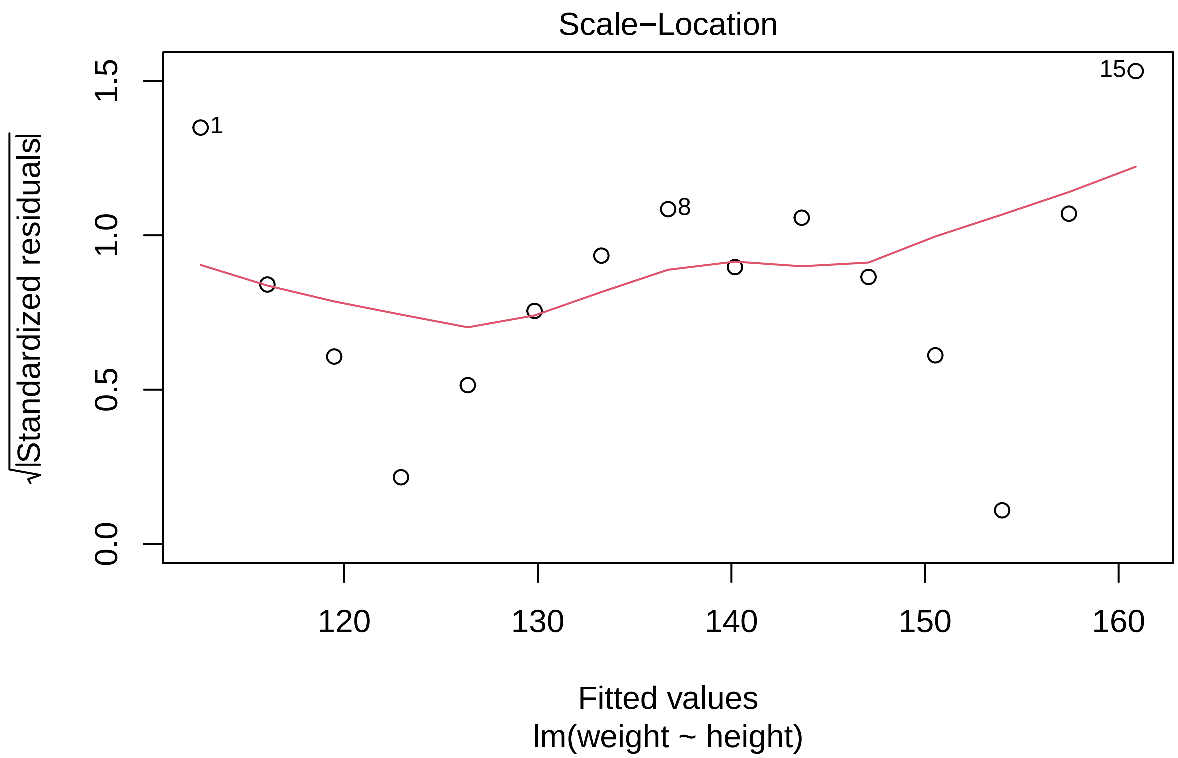
```r
# Load a built-in data called 'women'
```

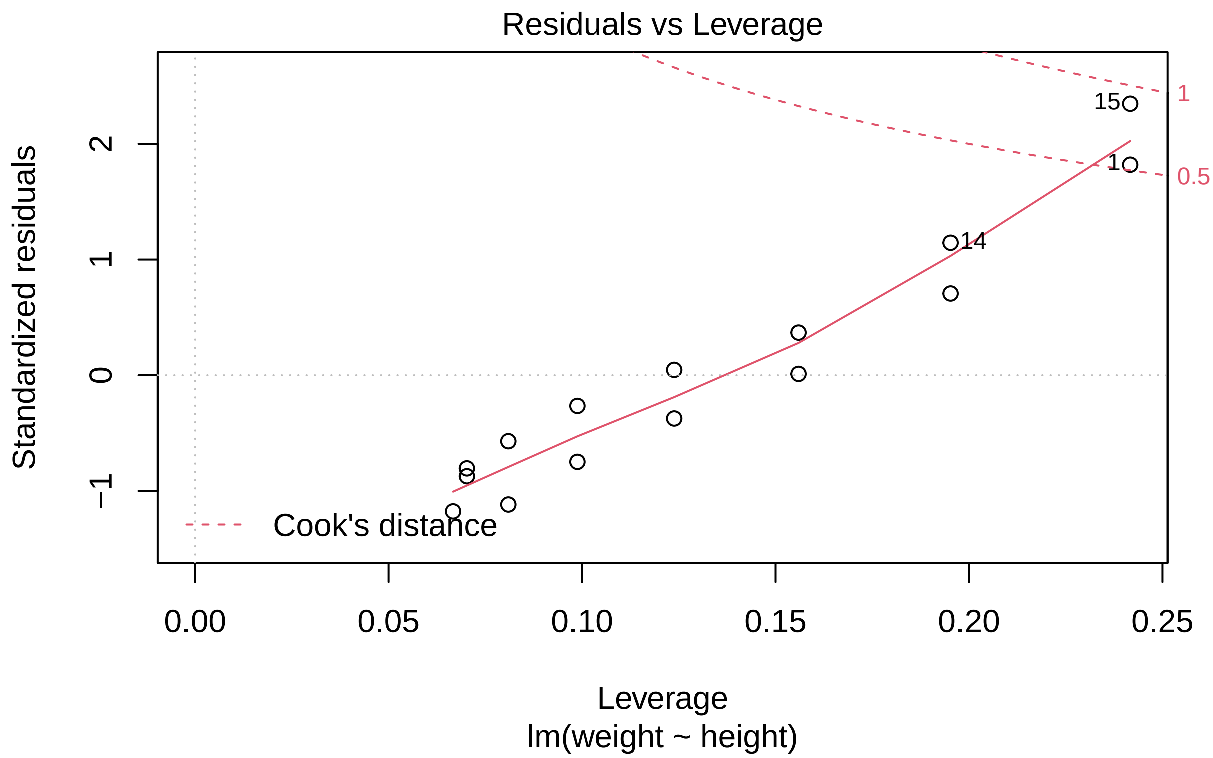
plot this data

```r
fit = lm(weight ~ height, women) # Run a regression analysis
plot(fit)
```



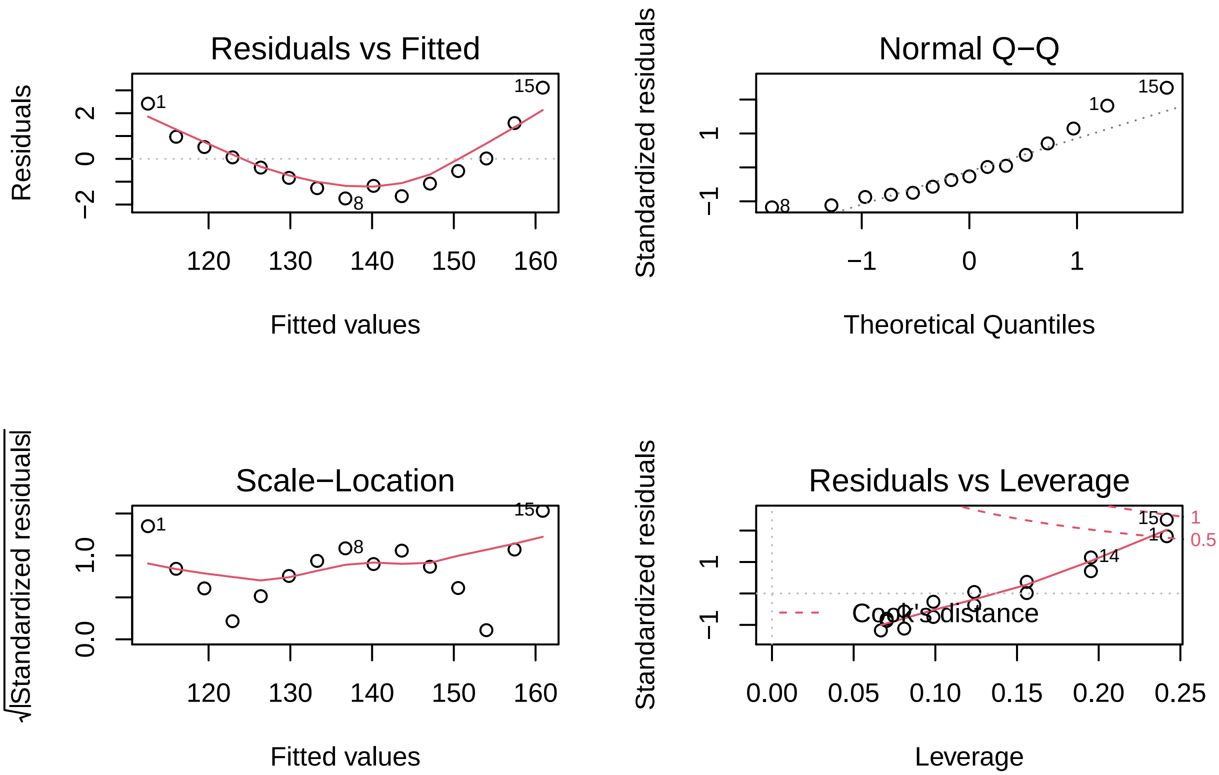

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(weight ~ height)
−1
0
1
2
−1
0
1
15
1
8


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
1
8
15
120
130
140
150
160
Fitted values
lm(weight ~ height)



if you want to look at four plots rather than one by one:

```
par(mfrow=c(2,2)) # Change the panel layout to 2 x 2
plot(fit)
```




```
par(mfrow=c(1,1)) # Change back to 1 x 1
```

Website: https://data.library.virginia.edu/diagnostic-plots/

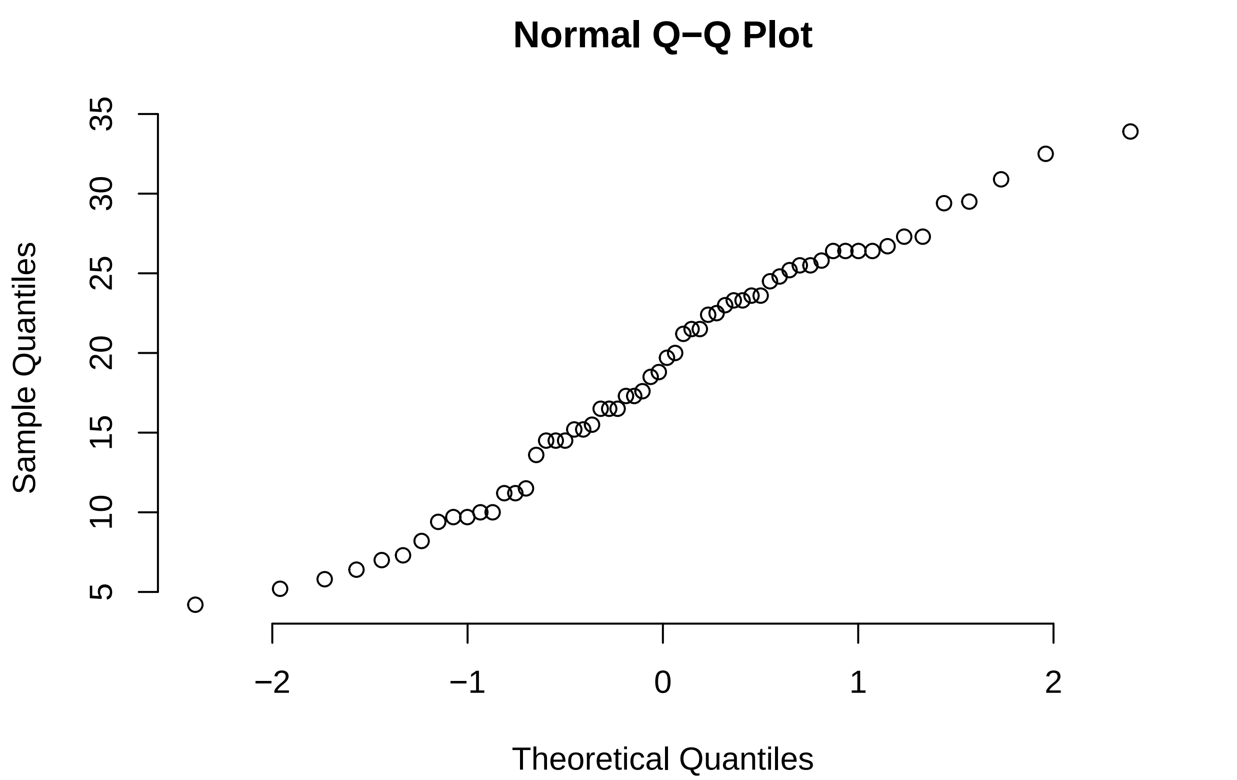
- A) Residuals vs Fitted: When conducting a residual analysis, a "residuals vs fitted lot" is the most frequently created plot. It is a scatter plot of residuals on the y axis and fitted values (estimated responses) on the x-axis. This plot shows if residuals have non-linear patterns. There could be a non-linear relationship between predictor variables and an outcome variable and the pattern could show up in this plot if the model doesn't capture the non-linear relationship. If you find equally spread residuals around a horizontal line without distinct patterns, that is a good indication you don't have non-linear relationships.
- B) Normal Q-Q Plot: This plot shows if residuals are normally distributed. Do residuals follow a straight line well or do they deviate severely? It's good if residuals are lined well on the straight dashed line.
- C) Scale Location Plot: also called Spread_Location plot. This plot shows if residuals are spread equally along the ranges of predictors. This is how you can check the assumption of equal variance. It is good if you see a horizontal line with equally (randomly) spread points.
- D) Residuals vs Leverage Plot: This plot helps us to find influential cases (subjects) if any. Not all outliers are influential in linear regression analysis (whatever outliers mean). Even though data have extreme values, they might not be influential to determine a regression line. That means, the results wouldn't be much different if we either include or exclude the from analysis. They follow the trend in the majority of cases and they don't really matter; they are not influential. On the other hand, some cases could be very influential even if they look to be within a reasonable range

of the values. They could be extreme cases against a regression line and can alter the results if we dont exclude them from analysis. Another way to put it is that they don't get along with the trend in the majority of cases.

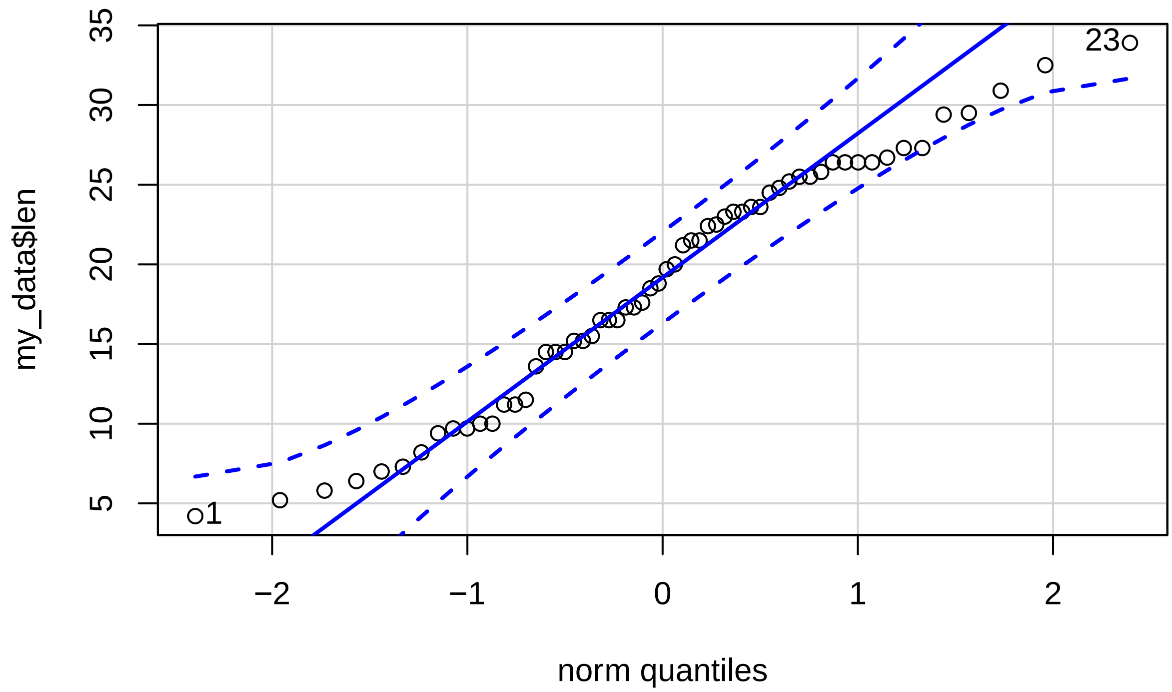
25 QQ Plots

QQ Plots are used to visually test whether data follows an expected distribution. They are used to verify that data used in a t-test is actually normally distributed. QQ plots are not useful on small datasets. As an example, we will use data from the `car` package, so install and load `car` (`install.packages("car")` and `library(car)`). Then:

```
my_data <- ToothGrowth
qqnorm(my_data$len, pch = 1, frame = FALSE)
```




```
qqPlot(my_data$len)
```




```
## [1] 23  1
```