Ve406 Applied Regression Analysis using R

1 Introduction

1.1 Course Profile

1.1.1 Course Information

- Course Description:

  This course provides an introduction to the process and procedures of statistical modelling. We will explore real data sets, examine various models for the data, assess the validity of their assumptions, and determine which conclusions we can make, if any. In this course you will learn how to program in R and how to use R for effective data analysis.

- Learning Outcomes:

  After successful completion of this course, you should be able to
  1. Explore data graphically.
  2. Select appropriate models.
  3. Implement those models in R.
  4. Examine those models critically.
  5. Interpret the results of those models to non-statisticians.

- Who should take this class?

  The prerequisite for this class is computer/programming knowledge at the level of Vg101 (or above), and statistics knowledge at the level of Ve401 (or above). Both undergraduates and graduate students are welcome to take the course.

1.1.2 Contact Information

- Instructor:

  Tong Zhu

- Lectures:

  Tuesday (06:20pm – 08:55pm) in E3-102
  Thursday (06:20pm – 08:00pm) in E3-102

- Office Hours:

  Thursday (12:00pm – 14:00pm) in JI-Building 442

- Email:

  tong.zhu@protonmail.com
  When sending an email related to this course please include the tag [Ve406] in the subject e.g. Subject: [Ve406] Special Request

- Teaching Assistant/s:
  Wu Zheng

1.1.3 Grading Policy

- Assignment:
  15% There will be 5 assignments in the form of problem sets.
- Lab/Workshop:
  10% There will be 4 labs/Workshops.
- Project:
  25% There will be a project in the form of a challenge.
- Exam:
  50% There will be two exams: Midterm 25% Final 25%
- Quiz (Optional):
  15% Quizzes will be given frequently in class.
- For those who attempt all quizzes, their grade is whichever is the higher of:
  0. 25% ALW + 25% Proj + 0% Quiz + 50% Exam
  1. 25% ALW + 10% Proj + 15% Quiz + 50% Exam
  2. 25% ALW + 40% Proj + 15% Quiz + 20% Exam
- For this course, the grade will be curved to achieve a median grade of "B+".

1.1.4 Project

- Each of you need to be in one and only one 3-member team for the project.
- The project will be graded according to the following three aspects:
  1. Oral Presentation of your model
  2. Poster Presentation of your model
  3. Prediction Accuracy of your model

  each of those three aspects has an equal weight.
- Each member of the same team will receive the same project mark.
- You will be working on real data (TESLA), some of part 1 is here

  but part 2 will not materialise before the due date.

1.1.5 Honour Code

- Honesty and trust are important. Students are responsible for familiarising themselves with what is considered as a violation of honour code.
- Assignments/projects are to be solved by each student individually. You are encouraged to discuss problems with other students, but you are advised not to show your written work to others. Copying someone else's work is a very serious violation of the honour code.
- Students may read resources on the Internet, such as articles on Wikipedia, Wolfram MathWorld or any other forums, but you are not allowed to post the original assignment question online and ask for answers. It is regarded as a violation of the honour code.
- Since it is impossible to list all conceivable instance of honour code violations, the students has the responsibility to always act in a professional manner and to seek clarification from appropriate sources if their or another student'ï£¡ï£¡s conduct is suspected to be in conflict with the intended spirit of the honour code.

1.1.6 Teaching Schedule

Week	Topics	Others
1	Introduction Simple Linear Regression Least-Squares and Maximum Likelihood	 A1R
2	Diagnostics Transformation	 W1R
3	Workshop 1 Multiple Linear Regression Diagnostics	 A1D A2R
4	**National Holiday**	
5	Inference Categorical regressors Collinearity	W1D W2R
6	Variable Selections Influential points and Outliers	A2D A3R

7	Workshop 2	W2D
	Heteroskedasticity	
	Correlated Noise	W3R
8	Nonlinear Regression	A3D
	Midterm Exam	A4R
9	Workshop 3	W3D
	Logistic Regression	
	Poisson Regression	W4R
10	Nonparametric Regression	A4D
	Generalized Linear Model	A5R
11	Workshop 4	W4D
	Estimating Equations	
	Generalised Additive Models	W5R
12	Principal Component Analysis	A5D
	Factor Analysis	Poster
13	Mixture Models	
	Survival Analysis (Optional)	
	Project Presentation	
14	**Final Exam**	

1.1.7 Textbook

- Simon J. Sheather (2010)
  A Modern Approach to Regression with R

- Grolemund and Wickham (2016)
  R for Data Science
- Myers et al. (2010)
  Generalized Linear Models with Applications in Engineering and the Sciences
- Fox. (2015)
  Applied Regression Analysis and Generalized Linear Models

1.2 R language

1.2.1 Python Vs R

Data Science Wars: Python Vs R

Q: Python probably needs no introduction to this audience, however, what is R?

- There are two unusual things regarding its origin.
- Firstly, it also comes from a quiet and small place,

  University of Auckland, New Zealand

- Secondly, it was created by two statisticians instead of a computer scientist

  Ross Ihaka and Robert Gentleman

It is a programming language for statistics/data science.

- Release Year
  1991
- Inspiration
  C
- Purpose
  Emphasises productivity and code readability.

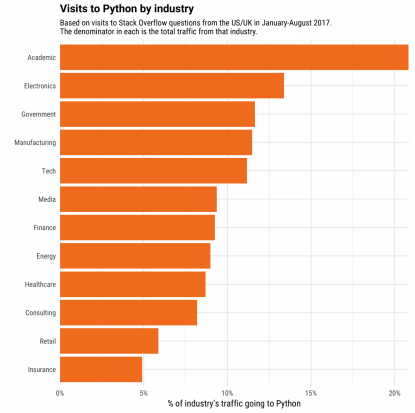
- Release Year
  1995
- Inspiration
  S
- Purpose
  Focuses on better, user friendly data analysis, statistics and graphical models.

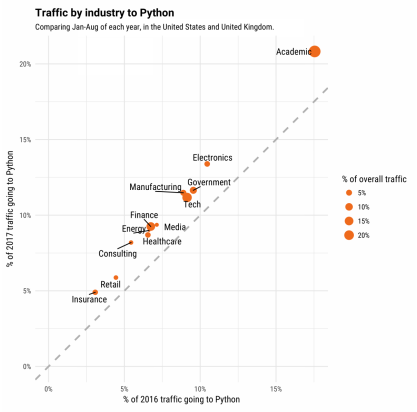
The S language was a commercial software developed at the famous Bell Laboratories by John Chambers and his collaborators Rick Becker, Allan Wilks and Duncan Temple Lang over the years from 1975 to 1998.

- Usability
  Coding and debugging is easier to do. Any piece of functionality is always written the same way.
- Ease of Learning
  Python's focus on readability and simplicity makes that its learning curve is relatively low and gradual.

- Usability
  Statistical models can be written with only a few lines. The same piece of functionality can be written in several ways.
- Ease of Learning
  Require only a minimum knowledge of computing at start, then a steep learning curve later on. However, R is easy to learn for experience programmers in C.

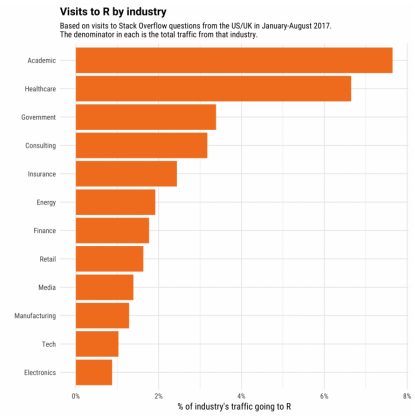
- Community
  Python is used by programmers that want to delve into data analysis or apply statistical techniques, and by developers that turn to data scientists.

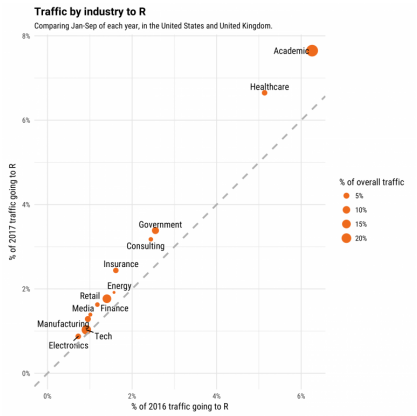




- Community

  Traditionally, R had been used primarily in academic and research institutes. However, R has expanded into the enterprise market.





In 2015, Microsoft closed its acquisition of Revolution Analytics, a commercial provider of and services for R. Revolution has made R enterprise-ready with speed and scalability for the largest data warehouses and Hadoop systems.
https://www.cio.com/article/2906456/data-analytics/microsoft-closes-acquisition-of-r-software-and-services-provider.html

- Usage

  Python is generally used when the data analysis tasks need to be integrated with web apps or incorporated into a production database.

- Good for

  Implementing algorithms for production use.
  Easy to share your work with other developers.

- Usage

  R is mainly used when the data analysis tasks require standalone computing

- Good for

  Beginners in statistics.
  Easy to interact with.

- So if you are good at programming and you want to learn statistics,

  R

  is the way to go, especially, if you begin with something standard.

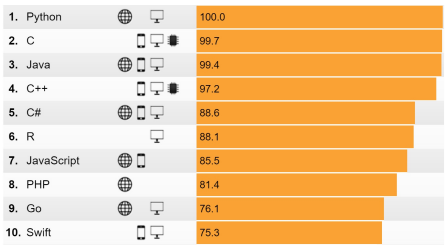
- However, if you are a member of a research and development team,

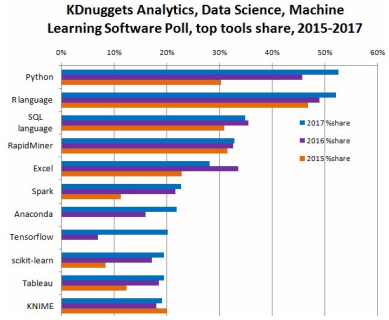
  Python

  is the way to go when integrating with web apps is important.

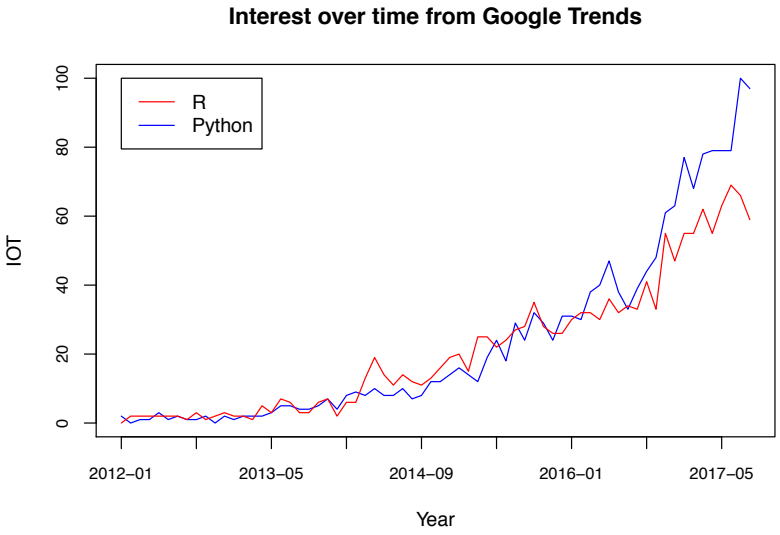
Q: Who is winning at the moment?

2017 IEEE top programming languages





Often R is used to conduct statistical analysis, graph data, and inspect large data sets, and Python is used for machine learning/mining algorithms that are only a portion of a large project.

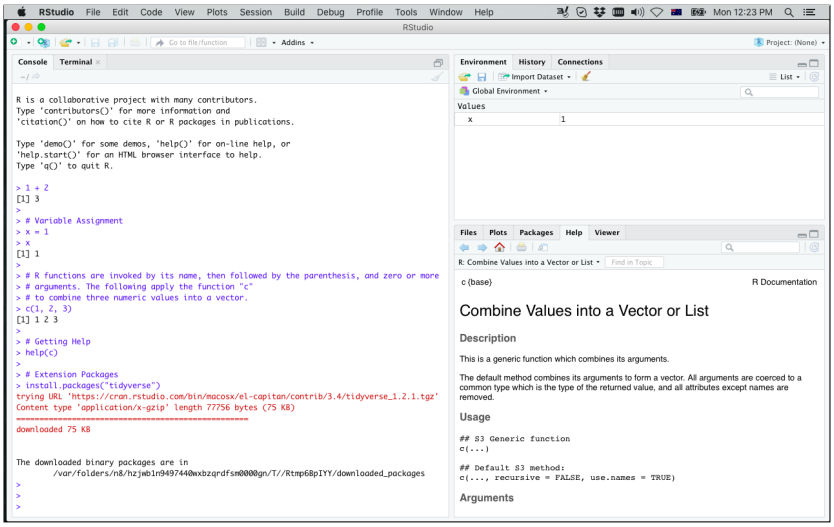


Q: Why using R in this course?

Numbers, IOT, represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means there was not enough data for this term.

- New to data science and statistics in general.
- Working independently
- More resources when comes to regression analysis

- R can be downloaded from HERE
- RStudio is an IDE for R, and can be downloaded from HERE



IDE stands for integrated development environment.

1.2.2 Data Format

Data Type

- In contrast to C, R sets the type based on the given value or expression.

```
> 3.14     # numeric, double-precision real
```

```
[1] 3.14
```

```
> TRUE     # logical, TRUE/FALSE
```

```
[1] TRUE
```

```
> "Hello" # Character
```

```
[1] "Hello"
```

```
> is.logical(TRUE); is.logical("TRUE")
```

```
[1] TRUE
[1] FALSE
```

This setup is similar to Python.

- Other data type exists, but often not explicitly specified or used

```
> 7L                                    # integer
```

```
[1] 7
```

```
> is.integer(7L); is.integer(7)
```

```
[1] TRUE
[1] FALSE
```

```
> 2+3i                                          # complex
```

```
[1] 2+3i
```

```
> sqrt(-1); sqrt(as.complex(-1))          # R in R
```

```
[1] NaN
# Warning message:
# In sqrt(-1) : NaNs produced
[1] 0+1i
```

NaN stands for Not a Number

Data Structure

- Assignment

  ```
  > x = 1
  ```

  ```
  > x <- 1
  ```

  ```
  > 1 -> x
  ```

  note all of above statements store the value 1 under the name `x`.

  ```
  > y = TRUE
  > z = "TRUE"
  ```

  ```
  > class(x); class(y); class(z)     # Object Classes
  ```

  ```
  [1] "numeric"
  [1] "logical"
  [1] "character"
  ```

The "=" assignment is preferred when specifying names.

- Vector

  ```
  > c(1, 2, 3)
  ```

```
[1] 1 2 3
```

```
> x.vec = c(.Last.value, x)      # Special Variable
> x.vec
```

```
[1] 1 2 3 1
```

```
> 1:12                           # Sequence
```

```
[1]  1  2  3  4  5  6  7  8  9 10 11 12
```

Q: What do you think the result of the following statement is?

```
> v.vec = 5:-5

> v.vec
```

```
 [1]  5  4  3  2  1  0 -1 -2 -3 -4 -5
```

- A more general sequence can be generated

```
> y.vec = seq(
+   from = 1,              # starting
+   by = 1,                # increment
+   length.out = 16)       # number of elements
```

Q: What do you think the result of the following statement is?

```
> tmp = seq(as.Date('2018-09-10'),
+           by = 14, length.out = 7)

> tmp[1:3] # assess elements of a vector by indexing
```

```
[1] "2018-09-10" "2018-09-24" "2018-10-08"
```

```
> lab = tmp[-2] # deselecting an element
>
> lab
```

```
[1] "2018-09-10" "2018-10-08" "2018-10-22"
[4] "2018-11-05" "2018-11-19" "2018-12-03"
```

Watch out for the recycling rule in R

Q: What do you think the result of the following statement is?

```
> c(1, 2, 3, 4) + c(1, 2)
```

Recycling Rule

Vectors occurring in the same expression need not all be of the same length. If they are not, the value of the expression is a vector with the same length as the longest vector which occurs in the expression. Shorter vectors in the expression are recycled as often as need be (perhaps fractionally) until they match the length of the longest vector. In particular a constant is simply repeated.

$$\begin{bmatrix}1\\2\\3\\4\end{bmatrix} + \begin{bmatrix}1\\2\end{bmatrix} \longrightarrow \begin{bmatrix}1\\2\\3\\4\end{bmatrix} + \begin{bmatrix}1\\2\\1\\2\end{bmatrix} = \begin{bmatrix}2\\4\\4\\6\end{bmatrix}$$

- Categorical Variable

```
> z.vec = c("hard",
+           "really-hard",
+           "extremely-hard",
+           "bite-your-head-off-hard",
+           "extremely-hard", "really-hard", "hard")

> z.fac = factor(z.vec, order = TRUE,
+                levels = c(
+                  "hard",
+                  "really-hard",
+                  "extremely-hard",
+                  "bite-your-head-off-hard"))

> class(z.fac); mode(z.fac)     # The storage mode
```

```
[1] "ordered" "factor"
[1] "numeric"
```

- By running the factor statement, R has assigned integers to each level

```
> str(z.fac)            # Structure
```

```
 Ord.factor w/ 4 levels "hard"<"really-hard"<..:
 1 2 3 4 3 2 1
```

```
> z.fac[1]>=z.fac[2]    # Comparison of 1st & 2nd
```

```
[1] FALSE
```

```
> table(z.vec)
```

```
z.vec
bite-your-head-off-hard
                      1
         extremely-hard
                      2
                   hard
                      2
            really-hard
                      2
```

```
> table(z.fac)
```

```
z.fac
                   hard
                      2
             really-hard
                      2
          extremely-hard
                      2
bite-your-head-off-hard
                      1
```

Q: What happens when you mix types inside a vector?

```
> x = 1; y = TRUE; z = "TRUE";
> u.vec = c(x,y); class(u.vec)
> u.vec = c(x,z); class(u.vec)
> u.vec = c(x,z.fac); class(u.vec)
> u.vec = c(y,z.fac); class(u.vec)
> u.vec = c(y,z.fac); class(u.vec)
```

- R does so-called implicit coercion to mixed types, the coercion rule goes

$$\text{logical}- > \text{integer}- > \text{numeric}- > \text{complex}- > \text{character}$$

```
> u.vec = c(x,y); class(u.vec)
```

```
[1] "numeric"
```

```
> u.vec = c(x,z); class(u.vec)
```

```
[1] "character"
```

$$\text{logical}- > \underbrace{\text{factor}}_{\text{integer}} - > \text{numeric}- > \text{complex}- > \text{character}$$

- R effectively assigns an integer to each level of a factor,

```
> u.vec = c(x,z.fac); class(u.vec)
```

```
[1] "numeric"
```

```
> u.vec = c(y,z.fac); class(u.vec)
```

```
[1] "integer"
```

```
> u.vec = c(z,z.fac); class(u.vec)
```

```
[1] "character"
```

- Matrix

```
> A = matrix(
+        y.vec,                    # the data elements
+        nrow = 4,                 # number of rows
+        byrow = TRUE)             # fill matrix by rows
```

- Matrices are special vectors in R.

```
> class(A)
```

```
[1] "matrix"
```

```
> mode(A)
```

```
[1] "numeric"
```

```
> attributes(A)
```

```
$dim
[1] 4 4
```

- A matrix is stored as a vector with a dimension specification.

Matrices are a special vectors in R. They are stored as vector with a dimension specification.

- Indexing

```
> A
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
```

```
> A[2,4]
```

```
[1] 8
```

```
> A[3,]
```

```
[1]  9 10 11 12
```

```
> A[c(1,3),c(2,4)]
```

```
     [,1] [,2]
[1,]    2    4
[2,]   10   12
```

```
> # Empty matrix
> B = matrix(nrow = 2, ncol = 3)
>
> # Single indexing in matrix
> B[1:5] = c("a", "p", "p", "l", "e")
>
> # Default is to fill B by row
> B
```

```
     [,1] [,2] [,3]
[1,] "a"  "p"  "e"
[2,] "p"  "l"  NA
```

```
> B[6] = 17
>
> B # No mixed types in matrix, coerced accordingly
```

```
     [,1] [,2] [,3]
[1,] "a"  "p"  "e"
[2,] "p"  "l"  "17"
```

- List

```
> xyzlist = list(x.vec=x.vec, y=y, z.fac=z.fac)
> xyzlist
```

```
$x.vec
[1] 1 2 3 1

$y
[1] TRUE

$z.fac
[1] hard
[2] really-hard
[3] extremely-hard
[4] bite-your-head-off-hard
[5] extremely-hard
[6] really-hard
[7] hard
4 Levels: hard < ... < bite-your-head-off-hard
```

- List is a special vector, each element of which can be a different class.

```
> class(xyzlist)
```

```
[1] "list"
```

```
> attributes(xyzlist)
```

```
$names
[1] "x.vec" "y"     "z.fac"
```

```
> class(xyzlist$x.vec); class(xyzlist$y)
```

```
[1] "numeric"
[1] "logical"
```

```
> xyzlist$x.vec[1]; xyzlist[[1]][1]   # 1st of 1st
```

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

Q: What do you think the result of the following statement is?

```
> xyzlist[["y"]]; xyzlist[[y]]
```

- Data frame

```
> # Vector by crated by replication statement
> manufacturer = rep("audi", 6)

> displ = c(1.80, 1.80, 2.00, 2.80, 3.10, 1.80)
> cyl = as.integer(c(rep(4,4), rep(6,2)))
> cty = c(18, 21, 20, 21, 16, 18)

> mpg.df = data.frame(manufacturer, displ, cyl, cty)
> attributes(mpg.df)
```

```
$names
[1] "manufacturer"   "displ"   "cyl"   "cty"

$row.names
[1] 1 2 3 4 5 6

$class
[1] "data.frame"
```

- displ stands for engineer displacement/size, volume of all the pistons inside the cylinders.
- cyl stands for the number of cylinders.
- cty miles per gallon in city

- For this tiny data set, asking R to display it explicitly makes sense

```
> mpg.df
```

```
  manufacturer displ cyl cty
1         audi   1.8   4  18
2         audi   1.8   4  21
3         audi   2.0   4  20
4         audi   2.8   4  21
5         audi   3.1   6  16
6         audi   1.8   6  18
```

```
> mpg.df[["cty"]]; mpg.df$cty[5]; mpg.df[[4]][5]
```

```
[1] 18 21 20 21 16 18
[1] 16
[1] 16
```

- Notice the indexing is very similar to list indexing.

```
> is.data.frame(mpg.df)
```

```
[1] TRUE
```

```
> is.list(mpg.df)
```

```
[1] TRUE
```

- Notice data frame can be indexed as if they are matrices, but...

```
> mpg.df[3,1]
```

```
[1] audi
Levels: audi
```

```
> is.matrix(mpg.df)
```

```
[1] FALSE
```

Q: What is the difference between list and data.frame?

Data frames are list as well, but they have a few restrictions:

- All elements of a data frame are vectors
- All elements of a data frame have an equal length
- Cannot have the same name for two different variables

So it is like a matrix that has more than numerical columns.

Conditional and Repetitive Execution

- ```
  > if (is.matrix(mpg.df)) {
  +   print("Data frame and Matrix are the same in R")
  + } else {
  +   print("A data frame is not a matrix")
  + }
  ```

  ```
  [1] "A data frame is not a matrix"
  ```

  ```
  > is.data.frame(mpg.df) & is.list(mpg.df)    # and
  ```

  ```
  [1] TRUE
  ```

  ```
  > is.data.frame(mpg.df) | is.matrix(mpg.df) # or
  ```

  ```
  [1] TRUE
  ```

  ```
  > ! is.matrix(mpg.df)                          # not
  ```

  ```
  [1] TRUE
  ```

Of course, you need to know the above.

- The following functions are often useful

  ```
  > x = 1:10
  > all(x>0)
  ```

  ```
  [1] TRUE
  ```

  ```
  > any(x>9)
  ```

  ```
  [1] TRUE
  ```

  ```
  > x = 1:10; (x = ifelse(x > 5, x, -x))
  ```

  ```
   [1] -1 -2 -3 -4 -5  6  7  8  9 10
  ```

Q: What is `s` after the following `for` loop statement?

```
> s = 0
> for (eps in x) {          # Repetitive execution
+   s = s + eps
+ }
```

- Notice R does not need to use an integer loop variable.

$s = 25$
```

```
> myfunc =
+   function(x) {       # Define myfunc
+     s = 0
+     for (eps in x) {
+       if (eps <= 0) {
+         next          # jump to the end of the loop
+       }
+       s = s + eps
+       if (s > 20) {
+         break         # stop the loop
+       }
+     }
+     s                 # output value
+   }
```

Q: What do you think the result of the following statement is?

```
> x = 1:10; x = ifelse(x > 5, x, -x); myfunc(x)
```

2 Simple Linear Regression

2.1 Overview

- Regression analysis is about statistical models that involve only one

  dependent variable Y

  and one or more independent variables X_j.

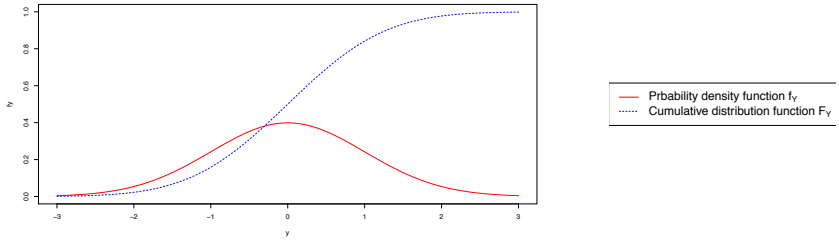
Q: What is the difference between a statistical model and mathematical model?

- It focuses on modelling the distribution of the dependent variable

  $$f_Y(y)$$

  using the independent variables X_j.

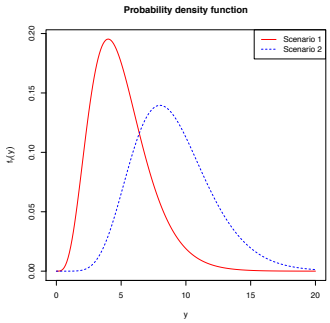
- Recall the distribution of a random variable is usually defined by



It summarise relationships between variables. It is a special kind of mathematical model.

- Non-deterministic, that is, instead of assigning a specific value for some of those variables, it will have probability distributions.
- It describes a *simplified* version of the the data-generating process.

- Depending on the values that the independent variables take,

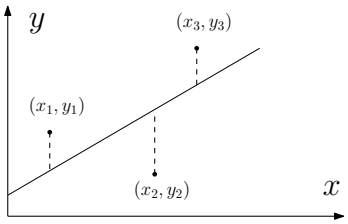


  Y may follow a different distribution. e.g.

  the distribution of height depends on gender

- Roughly speaking, regression is about finding a rule of picking distributions for Y from a space of infinitely many distributions that agrees with the data.

- This view of regression models is probably different from your understanding



- You will see that the two views are equivalent, but one is easier to generalise.
- Regression models usually attempt to address one of two questions:
  - Explaining the dependencies between

    $$\underbrace{\text{the explanatory variables}}_{\text{the independent variables}} \quad \text{and} \quad \underbrace{\text{the response}}_{\text{the dependent variable}}$$

  - Predicting the range of response values given a set of values for

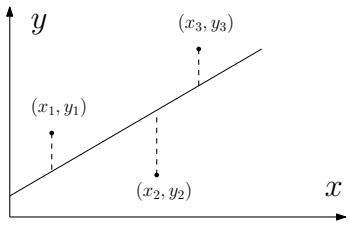
    $$\underbrace{\text{the predictors}}_{\text{the independent variables}}$$

- Independent variables
- Explanatory variables
- predictors
- regressors

- A simple linear model, where both X and Y are continuous

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

is particularly easy to understand and implement



Q: Can you recall how to find the best line? In other words, the estimated

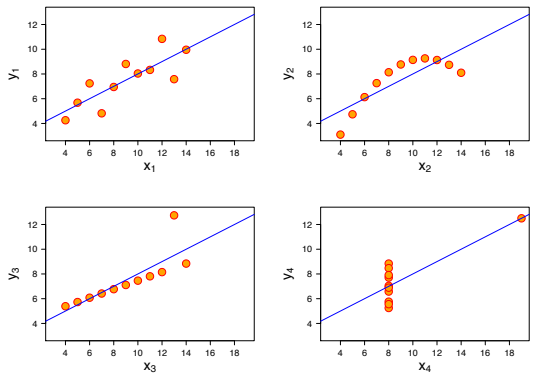
$$\beta_0 \qquad \text{and} \qquad \beta_1$$

- It is even not hard to imagine how one would proceed with finding

$$f_Y$$

$$b_1 = \frac{\sum_i^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_i^n (x_i - \bar{x})^2}$$
$$b_0 = \bar{y} - b_1 \bar{x}$$

Q: But why a linear model? What is exactly complicated with nonlinearity?



The datasets have approximately the same least squares line, as well as nearly identical means, standard deviations, and correlations.

- Note the least squares principle can also be used for

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i \qquad \text{or} \qquad y_i = \frac{\beta_1 x_i}{\beta_2 + x_i} + \varepsilon_i$$

To understand why the linear model, we consider one more approach.

- Consider a random variable Y that follows some known distribution.

$$f_Y$$

Q: Suppose we would like to predict the value of Y. What is the "best" guess?

- Let us denote our guess by

$$m$$

Q: What is a sensible and common way to measure the how good m is ?

- If we don't care about positive more than negative errors, then

$$\text{MSE}(m) = \mathbb{E}\left[(Y - m)^2\right]$$

is the traditional choice, which can be decomposed into

$$\text{MSE}(m) = \text{Var}[Y] + (\mathbb{E}[Y - m])^2$$

Q: Can you recall why the above is true and what does it mean?

MSE stands for mean square error. And

$$\begin{aligned}
\mathbb{E}\left[(Y - m)^2\right] &= \mathbb{E}\left[(Y - \mathbb{E}[Y] + \mathbb{E}[Y] - m)^2\right] \\
&= \mathbb{E}\left[(Y - \mathbb{E}[Y])^2 + 2(Y - \mathbb{E}[Y])(\mathbb{E}[Y] - m) + (\mathbb{E}[Y] - m)^2\right] \\
&= \mathbb{E}\left[(Y - \mathbb{E}[Y])^2\right] + 2(\mathbb{E}[Y] - m)\,\mathbb{E}[Y - \mathbb{E}[Y]] + \mathbb{E}\left[(\mathbb{E}[Y] - m)^2\right] \\
&= \text{Var}[Y] + 0 + (\mathbb{E}[Y] - m)^2 \\
&= \text{Var}[Y] + (\mathbb{E}[Y] - m)^2
\end{aligned}$$

Q: What is bias in statistics?

$$\text{Bias}\left(\hat{\theta}, \theta\right) = \mathbb{E}\left[\hat{\theta} - \theta\right]$$

where $\hat{\theta}$ is often an estimator, and θ is a population parameter, thus fixed.

- So we have the simplest form of the bias-variance decomposition,

$$\text{MSE}(m) = \text{Var}[Y] + (\mathbb{E}[Y - m])^2$$

which confirms the expected value is the best single number for predicting Y

$$m = \mathbb{E}[Y] = \int_{-\infty}^{\infty} y f_Y(y)\, dy$$

- Note changing our prediction, m, does nothing to the true distribution of Y.

When other criterion is used, similar things can be done but we might need a bit more calculus

- Now imagine we have two random variables, say

$$X \quad \text{and} \quad Y$$

- Suppose we also know X in the sense we know

$$f_X,$$

and we would like to use this knowledge to improve our prediction of Y.

Q: What is the "best" guess m now?

- It is clear that our guess should be a function of x,

$$m(x)$$

- Again we would like to make MSE as small as possible

$$\mathbb{E}\left[\left(Y - m(X)\right)^2\right]$$

- We use conditional expectations to reduce

$$\mathbb{E}\left[\left(Y - m(X)\right)^2\right]$$

back to the problem we already solved

$$\mathbb{E}\left[\left(Y - m(X)\right)^2\right] = \mathbb{E}_X\left[\mathbb{E}_Y\left[\left(Y - m(X)\right)^2 \mid X\right]\right]$$

- For each possible value x, the best value is just the conditional mean

$$m(x) = \mu(x) = \mathbb{E}\left[Y \mid X = x\right]$$

which is known as the true regression function.

- Now we have reached the part that requires us to use a linear approximation,

$$\begin{aligned} f(x) &= f(a) + f'(a)(x - a) + \cdots \\ &= f(a) - f'(a) + f'(a)x + \cdots \end{aligned}$$

- We restrict the regression function to have the linear form

$$m(x) = \beta_0 + \beta_1 x$$

and find the choice of β_0 and β_1 so that it is the best under this restriction.

- This is achieved by simply minimising MSE with respect to β_0 and β_1

$$\text{MSE}(\beta_0, \beta_1) = \mathbb{E}\left[\left(Y - (\beta_0 + \beta_1 X)\right)^2\right]$$

Q: Can you see why the following choice is the best?

$$\beta_0 = \mathbb{E}\left[Y\right] - \beta_1 \mathbb{E}\left[X\right] \qquad \text{where} \quad \beta_1 = \frac{\text{Cov}\left[X, Y\right]}{\text{Var}\left[X\right]}$$

- Notice means play no role in β_1, only the variance and covariance mattered.

It is clear from the law of total expectation

$$\mathbb{E}[Y] = \mathbb{E}_X[\mathbb{E}_Y[Y \mid X]] = \mathbb{E}_X[\beta_0 + \beta_1 X] = \beta_0 + \beta_1 \mathbb{E}[X]$$

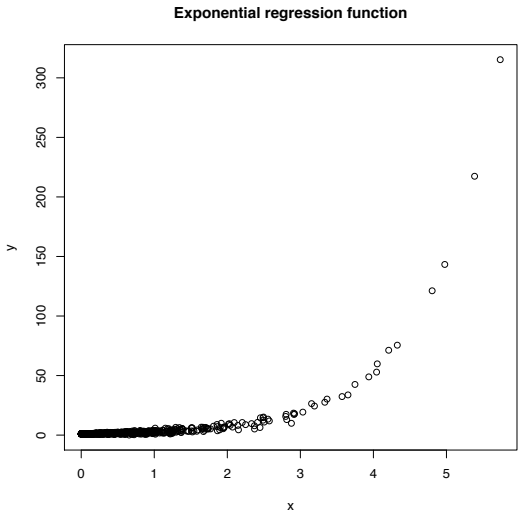
However, β_1 is less clear, using the above

$$\begin{aligned}
\text{MSE}(\beta_1) &= \mathbb{E}\left[\left(Y - (\mathbb{E}[Y] - \beta_1 \mathbb{E}[X] + \beta_1 X)\right)^2\right] \\
&= \mathbb{E}\left[\left((Y - \mathbb{E}[Y]) + \beta_1 (X - \beta_1 \mathbb{E}[X])\right)^2\right] \\
&= \mathbb{E}\left[(Y - \mathbb{E}[Y])^2\right] + \beta_1^2 \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] \\
&\qquad + 2\beta_1 \mathbb{E}\left[(Y - \mathbb{E}[Y])(X - \mathbb{E}[X])\right] \\
&= \text{Var}[Y] + 2\beta_1 \text{Cov}[X, Y] + \beta_1^2 \text{Var}[X]
\end{aligned}$$

This means a big slope β_1 leads X and Y fluctuate together for a fixed $\text{Var}[X]$.

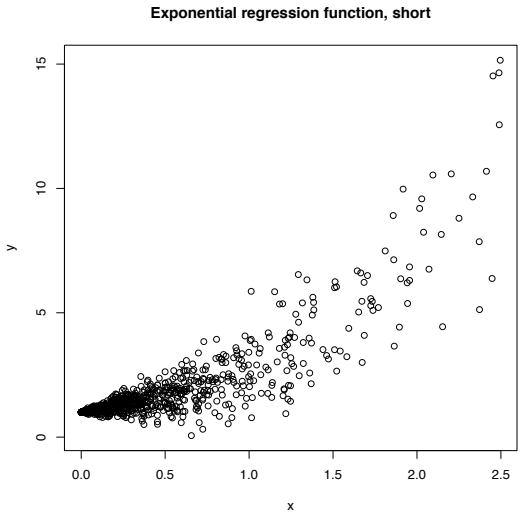
- It is the best amongst linear predictors, but it can be far from the truth. e.g.

$$\mathbb{E}[Y \mid X] = e^x$$



- We did not assume the relationship between X and Y is linear, but

$$\mu(x) = \mathbb{E}[Y \mid X] = \mu(x_0) + (x - x_0)\mu'(x_0) + \frac{1}{2}(x - x_0)^2 \mu''(x_0) + \cdots$$



- So using linear approximation as a justification for teaching and using linear model globally is a weak argument.
- So it was done in the past mainly for computational reasons.
- That said, linear models are very useful for illustrative purposes.

- Recall regression is about finding a rule of picking distributions for Y from a space of infinitely many distributions that agrees with the data.
- If we know everything about X and Y, then the "best" guess is

$$m = \beta_0 + \beta_1 x$$

where

$$\beta_0 = \mathbb{E}[Y] - \beta_1 \mathbb{E}[X] \qquad \text{and} \qquad \beta_1 = \frac{\text{Cov}[X, Y]}{\text{Var}[X]}$$

Q: What do we do when we don't have f_X and f_Y?

- It is natural to consider the unbiased estimates for the expectations, variance and covaraince to obtain estimates for β_0 and β_1

$$b_0 = \bar{y} - b_1 \bar{x} \qquad \text{where} \qquad b_1 = \frac{c_{xy}}{s_x^2}$$

Q: What do you notice?

This is the same as OLS estimates.

2.2 Basics

Q: What does it mean when we say a sequence of random variables

$$\{X_1, X_2, \ldots, X_n\}$$

is independent and identically distributed (i.i.d.)?

- We will treat the data you have on the response,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

as just one realisation of the sequence

$$\{Y_1, Y_2, \cdots, Y_n\}$$

Q: Does this sequence necessarily satisfy the i.i.d. condition?

$$\{Y_1, Y_2, \cdots, Y_n\}$$

- Until the end of part I, we will only consider independent Y_i.
- It was said that some men are more likely to have daughters than others e.g.

a deep-sea diver, a fighter pilot and a heavy smoker

- If you prefer sons, easy! just become US president.

The Facts

- The 45 US presidents from George Washington to Donald Trump have had a total of 158 children, comprising 91 sons and only 67 daughters.

about 1.4 sons for very daughter

- Two studies of deep-sea divers revealed that the men had a total of 190 children, comprising 65 sons and 125 daughters:

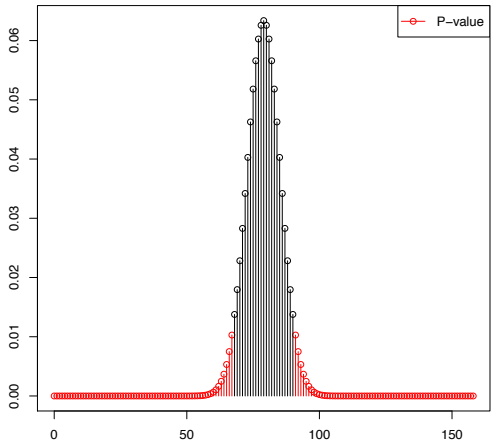
about 1.9 daughters for every son

- Let consider testing the two hypotheses one at time.

```
> # President ----------------------------
> rm(list = ls()) # Clean environment
> # Suppose the chance is 50-50
> # for a president as well
> p = 1/2
>
> # Total number of trials
> n = 158
>
> # All possible values 0 to 158 daughters
```

```
> x = 0:n
>
> # Binomial density/mass function
> fX = dbinom(x, size = n, prob = p)
> # Produce a plot of the density function
> # With the p-value region highlighted in red
```

- Recall P-value is the probability of getting a result at least as extreme as 65 daughters under $H_0 : p = 0.5$.




```
> p.lowerTail = pbinom(67, size = n, prob = p)
> 2 * p.lowerTail                # P-value
```

```
[1] 0.06694264
```

```
> # Lower tail
> tmpL = fX[x <= 67] # As extreme as 67

> # Upper tail probability
> x.upperTail = 1 + qbinom(
+   p.lowerTail, size = n, prob = p,
+   lower.tail = FALSE)

> tmpU = fX[x >= x.upperTail]

> # Middle
> M = x[ x>67 & x<x.upperTail]
> tmpM = fX[ x>67 & x<x.upperTail ]
```

```
> # Actual plotting
> plot(x, fX, type = "n", xlab = "", ylab = "")

> lines(0:67, tmpL, type = "h", col = "red")

> points(0:67, tmpL, col = "red")

> lines(x.upperTail:n, tmpU,
+       type = "h", col = "red")
> points(x.upperTail:n, tmpU, col = "red")

> lines(M,tmpM, type = "h")
> points(M, tmpM)

> legend("topright", legend = "P-value",
+        col ="red", lty = 1, pch = 1)
```

- If we do the same to

```
> # Deep-sea divers -----------------------
> 2 * pbinom(65, size = 190, prob = 0.5)
```

```
[1] 1.603136e-05
```

- This is a little more than one chance in 100 thousand.
- The following is what we can say:

  We conclude that it is extremely unlikely that this observation could have occurred by chance, if the deep-sea divers had equal probabilities of having sons and daughters. The data are not compatible with H_0.

Q: How about the president case?

```
> 2 * p.lowerTail
```

```
[1] 0.06694264
```

- For the president case, the following is what we can say:

  We conclude that there is no real evidence that presidents are more likely to have sons than daughters or vice versa. The observations are compatible with the possibility that there is no difference.

Q: Does this mean presidents are equally like to have sons and daughters?

No, the observations are also compatible with the possibility that there is a difference. We just don't have enough evidence either way.

Q: Back to the deep-sea divers case, does p-value of 1.603136e-05 say about the actual probability of a diver having a daughter instead of a son?

- Let p denote the probability of a deep-sea diver has a daughter, and X be the number of daughters out of 190 children that deep-sea divers have

$$X \sim \text{Binomial}(190, p)$$

Q: Which single value would you say p is equal to?

Common-sense suggests to use

$$p = \frac{\text{number of daughters observed}}{\text{total number of children}} = \frac{125}{190} = 0.658$$

Definition

The process of using data to suggest a value for a parameter is called estimation.

- The value suggested is called the estimate of the parameter. It is clear that the estimate should depend on the data.
- An estimator is a function of the data, that gives estimates of the parameter

$$\hat{\theta} = h(X)$$

- Notice $\hat{\theta}$ is also random, its randomness is inherited from the data.
- The distribution for $\hat{\theta}$ is called the sampling distribution of the estimator.
- Consider the following

```
> # Function ------------------------------
> myp_func = function(p){
+   n = 190
+   p125 = pbinom(125, size = n, prob = p)
+   p124 = pbinom(124, size = n, prob = p)
+   p125 - p124 # dbinom(125, size = n, prob = p)
+ }

> myp_func(p = 0.5); myp_func(p = 0.6)
```

```
[1] 3.972689e-06
[1] 0.01576121
```

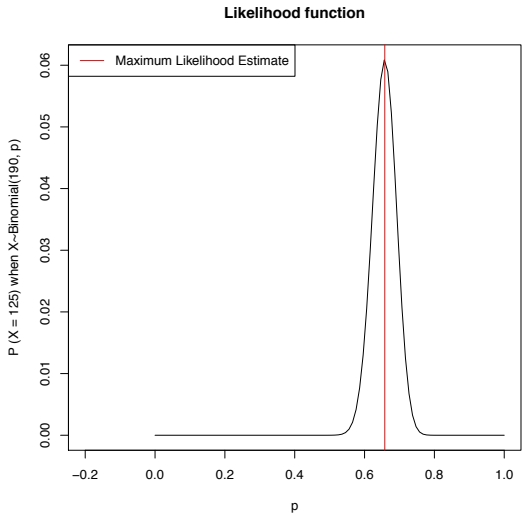
- Having $p = 0.6$ still looks unlikely, but it is almost 4000 times more likely.

```
> n
```

```
[1] 158
```

- It is not hard to understand the best choice of p in this case is MLE.

Q: What is MLE?




```
> # many functions in R can take vector input
> pbinom(125, size = n, prob = c(0.5,0.6))
```

```
[1] 0.9999960 0.9567501
```

Thus the tidy function can actually take vector input as well

```
> myp_func = function(p){
+   n = 190
+   p125 = pbinom(125, size = n, prob = p)
+   p124 = pbinom(124, size = n, prob = p)
+   p125 - p124 # dbinom(125, size = n, prob = p)
+ }

> # Create a vector contains a sequence of numbers
> pvec = seq(0, 1, length.out = 100)
>
> lvec = myp_func(pvec)
```

```
> plot(pvec, lvec,
+      type = "l",
+      xlab = "p",
+      ylab = "P(X = 125) when X~Binomial(190, p)",
+      main = "Likelihood function",
+      xlim = c(-0.2,1)
+      )

> phat = 125/190 # common-sense

> abline(v = phat, col = "red") # MLE
>
> legend("topleft",
+        legend = "Maximum Likelihood Estimate",
+        lty = 1, col = 2)
```

Q: Is there any way to attach some kind of "strength" to out estimate?

$$\hat{p} = \frac{125}{190}$$

- Because there clearly is a difference between the following scenarios
  - 125 daughters out of 190 children
  - 1250 daughters out of 1900 children

  despite of having the same MLE $\hat{p} = \frac{25}{28}$.
- I have been hiding things from you.

```
> # confidence interval: Clopper-Pearson
> binom.test(125, n = 190, p = 0.5)
```

```
        Exact binomial test

data:  125 and 190
number of successes = 125, number of trials = 190, p-value = 1.603e-05
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.5857408 0.7250337
sample estimates:
probability of success
            0.6578947
```

- When people say

  > We are 95% confident that the true probability of having a daughter is between $(0.586, 0.725)$ for a male deep-sea diver.

  they actually meant

  > The method used to obtain the interval
  >
  > $$(0.586, 0.725)$$
  >
  > will contain the true probability 95% of time if it is to be repeated.

```
> install.packages("binom")
> binom::binom.confint(125, n = 190)
```

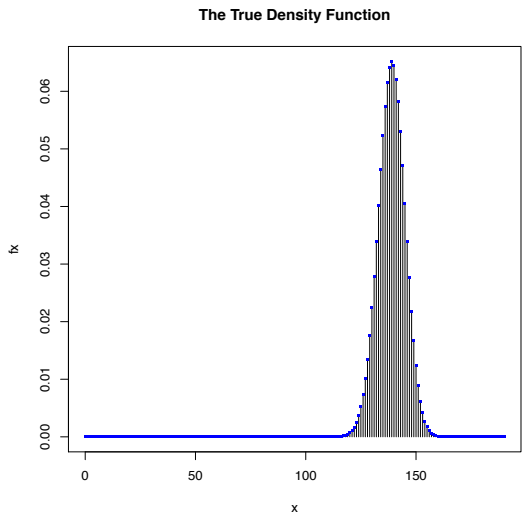
```
         method   x   n      mean     lower     upper
1  agresti-coull 125 190 0.6578947 0.5878351 0.7216963
2     asymptotic 125 190 0.6578947 0.5904374 0.7253521
3          bayes 125 190 0.6570681 0.5895875 0.7236296
4        cloglog 125 190 0.6578947 0.5857332 0.7205325
5          exact 125 190 0.6578947 0.5857408 0.7250337
6          logit 125 190 0.6578947 0.5876377 0.7218476
7         probit 125 190 0.6578947 0.5882529 0.7225372
8        profile 125 190 0.6578947 0.5886447 0.7229128
9            lrt 125 190 0.6578947 0.5886470 0.7229437
10     prop.test 125 190 0.6578947 0.5852078 0.7240821
11        wilson 125 190 0.6578947 0.5879068 0.7216245
```

Q: Can you understand what the following piece of R code is about?

```
> rm(list = ls())
> # Simulation study on CI, estimation, and testing
> num = 1e3       # number of repetition
> n = 190         # number of trial
> # Generate a true parameter use uniform(0,1)
> p = runif(1)
> p
```

```
[1] 0.730913
```

```
> x = 0:190
> fx = dbinom(x, size = n, prob = p)
>
> plot(x, fx, type = "h",
+       main = "The True Density Function")
>
> points(x, fx, pch = ".", col = "blue", cex = 3)
```



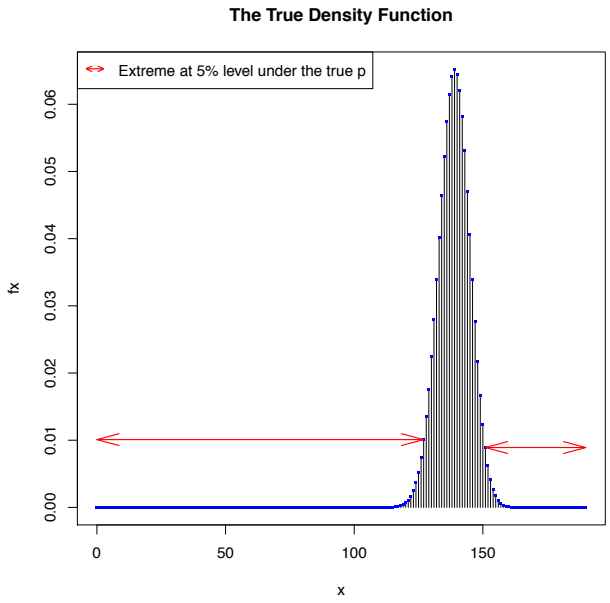
- Let us indicate unusual events under the true p on the graph.

```
> # Indicate unlikely events at 5% level under true
> x_unlikely_lower =
+   qbinom(0.025, size = n, prob = p)

> yL = dbinom(x_unlikely_lower,
+             size = n, prob = p)

> arrows(x0 = 0, y0 = yL,
+        x1 = x_unlikely_lower, y1 = yL,
+        angle = 15, col = 2, code = 3, lwd = 1.2)

> x_unlikely_upper =
+   qbinom(0.025, size = n, prob = p,
+          lower.tail = FALSE)
> yU = dbinom(x_unlikely_upper, size = n, prob = p)
> arrows(x0 = x_unlikely_upper, y0 = yU,
+        x1 = n, y1 = yU,
+        angle = 15, col = 2, code = 3, lwd = 1.2)
```




```
> legend("topleft", lwd = NA, lty = NA, legend =
+          "Extreme at 5% level under the true p",
+        x.intersp = 0)
```

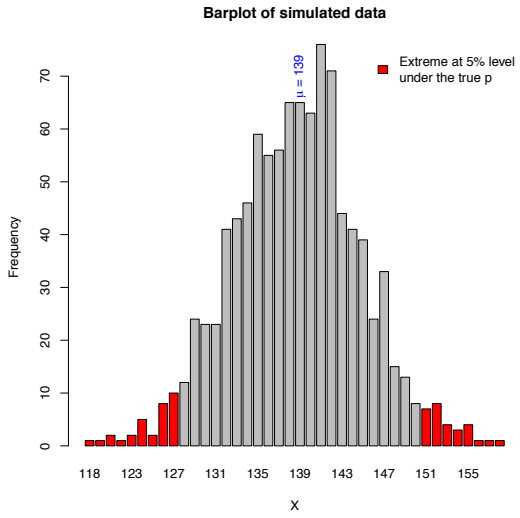
```
> par(font = 5) #change font to get arrows

> legend("topleft", legend = NA,
+        lwd = 1, lty = NA,
+        pch = 171,
+        col = 2,
+        bty = "n",
+        pt.cex=1.3)

> par(font = 1) #change back to common font
```

Q: A confidence interval is a type of interval estimate that may fail with a given probability, what exactly is this probability? What is actually to be repeated?

```
> X = rbinom(num, size = n, prob = p)
```




```
> # Sorting the data into a table of counts
> counts = table(X)

> nonzero.counts = names(counts)
> head(nonzero.counts)
```

```
[1] "118" "120" "121" "122" "123" "124"
```

```
> tmp = as.character(x_unlikely_lower)
> xL.index = which(nonzero.counts == tmp)
```

```
> tmp = as.character(x_unlikely_upper)
> xU.index = which(nonzero.counts == tmp)

> tmp = length(counts)
> index.vec = rep(1, tmp)

> index.vec[1:xL.index] = 2
> index.vec[xU.index:tmp] = 2


> cols.vec = c("grey", "red")[index.vec]

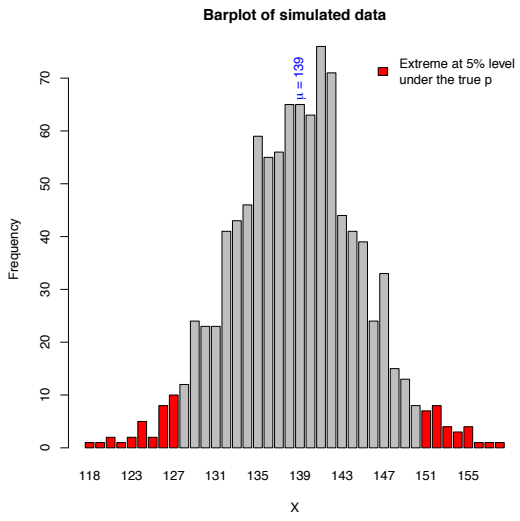
> x.mean = round(n*p)
> xm.index = which(nonzero.counts == x.mean)

> barpos =
+   barplot(counts, col = cols.vec,
+           xlab = "X", ylab = "Frequency",
+           main = "Barplot of simulated data")

> text(barpos[xm.index],
+      counts[[xm.index]] + 5,
+      bquote(mu~"="~.(x.mean)),
+      col = 4, srt = 90)

> legend("topright", legend =
+          "Extreme at 5% level\nunder the true p",
+        fill = 2, bty = "n")
```

Q: What do you think will happen to the C.I. based on those x in the red bars?



```
> res = binom.test(X[1], n = n, p = p)
>
> res; p
```

```
        Exact binomial test

data:  X[1] and n
number of successes = 124, number of trials = 190, p-value = 0.01733
alternative hypothesis: true probability of success is not equal to 0.7315841
95 percent confidence interval:
 0.5803205 0.7200957
sample estimates:
probability of success
            0.6526316

[1] 0.7315841
```

```
> names(res)
```

```
[1] "statistic"   "parameter"   "p.value"
[4] "conf.int"    "estimate"    "null.value"
[7] "alternative" "method"      "data.name"
```

```
> res$p.value
```

```
[1] 0.01732913
```

```
> res$conf.int[1:2]
```

```
[1] 0.5803205 0.7200957
```

```
> res.df = data.frame(
+   CIL = double(), CIU = double())

> for (i in 1:num){
+   res = binom.test(X[i], n = n, p = p)
+   res.df[i,1:2] = res$conf.int[1:2]
+ }
```

```
> head(res.df, 2)
```

```
       CIL       CIU
1 0.5803205 0.7200957
2 0.6184566 0.7544657
```

```
> res.df[1,1] < p
```

```
[1] TRUE
```

```
> res.df[1,1] < p & res.df[1,2] > p
```

```
[1] FALSE
```

```
> n_ci_contain_true =
+   sum(res.df[, 1] < p & res.df[, 2] > p)

> (rate = 1 - n_ci_contain_true/num)
```

```
[1] 0.044
```

- What does the law of large numbers say?

Law of Large Numbers

Suppose that X_1, X_2, ..., X_n all have the same expected value

$$\mathbb{E}[X_i] = \mu,$$

the same variance

$$\text{Var}[X_i] = \sigma^2,$$

and zero covariance with each other

$$\text{Cov}[X_i, X_j] = 0 \qquad \text{where} \quad i \neq j$$

In particular, if the X_i are i.i.d., then the following holds

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \to \mu \qquad \text{when} \quad n \to \infty$$

- The law of large number is the justification for using the following earlier

$$\hat{\beta}_0 = \bar{Y}_n - \hat{\beta}_1 \bar{X}_n$$

- It is easy to understand intuitively, but it was often misunderstood.
- Consider the following pieces of R code.

```
> # LLN ----------------------------------------
> rm(list=ls())
> n = 1e4 # final sample size
> lambda = 3

> # Consider Poisson random variables
> xpois = rpois(n, lambda)
> xexp = lambda # True mean for Poisson
```

- Let us investigate $\bar{X}$ as n increases

```
> # sample size at various stages
> nseq = 1:n

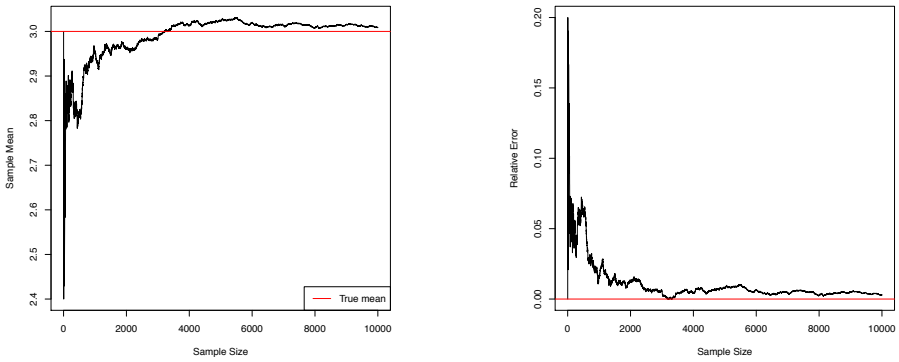

> # sample mean at various stage
> xcbar = cumsum(xpois) / nseq

> # Relative error
> error = abs(xcbar-xexp) / abs(xexp)

> plot(nseq, xcbar, type = "l",
+      xlab = "Sample Size", ylab = "Sample Mean")
>
> abline(h = xexp, col = 2)
>
> legend("bottomright", legend = "True mean",
+        lty = 1, col = 2)

> plot(nseq, error, type = "l",
+      xlab = "Sample Size", ylab = "Relative Error")
>
> abline(h = 0, col = 2)
```

Q: What do you expect to see?



Q: Do you think it is due to not having a large enough sample size?

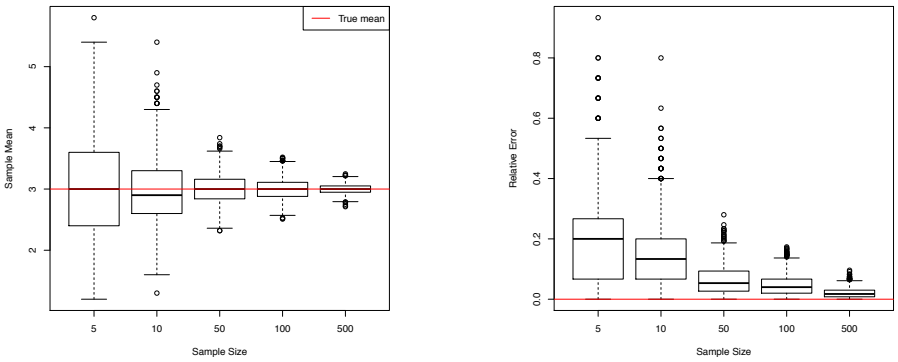
```
> # A better look at LLN ---------------------------
> sample.size.vec = c(5, 10, 50, 100, 500)
> ncases = length(sample.size.vec)          # 5 cases
> num = 1000                    # number of repetitons
> xbar.vec = double()     # x bar for all simulations
> error.vec = double()    # error for all simulations
> n.vec = integer()       # sample size for each case
```

```
> for(j in 1:ncases){
+   # Current sample size
+   n = sample.size.vec[j]

+   s.vec = double(num)  # all x bar for this n
+   e.vec = double(num)  # all error for this n

+   # Repeat num number of times
+   for (i in 1:num){
+       x = rpois(n, lambda)
+       s.vec[i] = sum(x) / n
+       e.vec[i] = abs(s.vec[i] - xexp) / abs(xexp)
+   }

+   # Store simulation results
+   xbar.vec = c(xbar.vec, s.vec)
+   error.vec = c(error.vec, e.vec)
+   n.vec = c(n.vec, rep(sample.size.vec[j], num))
+
+ }
```



- Note how we mange to reduce the spread of the distribution to 0 effectively.

$$\lim_{n\to\infty} \Pr\left(\left|\bar{X}_n - \mu\right| > \varepsilon\right) = 0 \qquad \text{for any} \quad \varepsilon > 0$$

```
> x.df =
+   data.frame(xbar = xbar.vec,
+              error = error.vec, n = n.vec)
>
> boxplot(xbar~n, data = x.df,
+         xlab = "Sample Size",
+         ylab = "Sample Mean")
>
```

```
> abline(h = xexp, col = 2)
>
> legend("topright", legend = "True mean",
+        lty = 1, col = 2)
>
> boxplot(error~n, data = x.df,
+         xlab = "Sample Size",
+         ylab = "Relative Error")
>
> abline(h = 0, col = 2)
```

Q: What does the central limit theorem say?

Central Limit Theorem

Suppose X_1, X_2, ..., X_n are i.i.d. with mean μ and variance σ^2, then

$$\sqrt{n}\frac{\bar{X}_n - \mu}{\sigma}$$

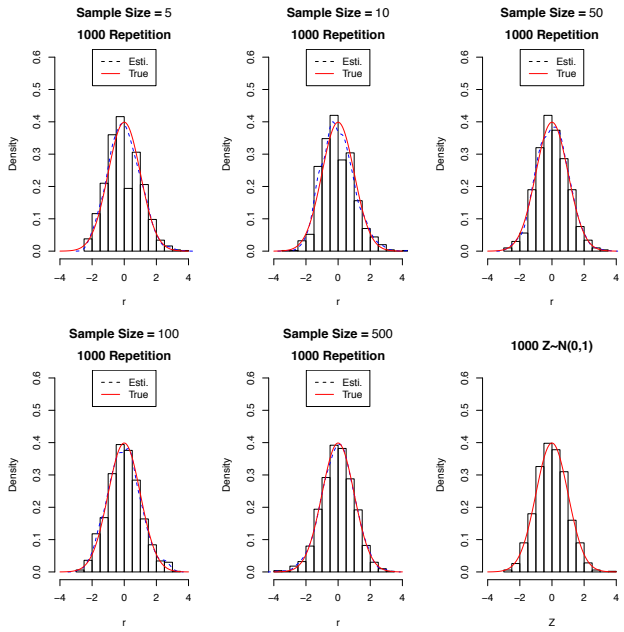
becomes more and more likely a standard normal random variable as $n \to \infty$

$$Z \sim \text{Normal}(0, 1)$$

in the sense that for every real number z

$$\lim_{n\to\infty} \Pr\left(\sqrt{n}\frac{\bar{X}_n - \mu}{\sigma} \leq z\right) = \Pr\left(Z \leq z\right)$$

- Intuitively, it means the sequence of distributions can be better and better described by a normal distribution as n becomes bigger and bigger.




```
> # CLT ---------------------------------------------
> sample.size.vec
```

```
[1]   5  10  50 100 500
```

```
> lambda; xexp
```

```
[1] 3
[1] 3
```

```
> xvar = lambda # True variance for Poisson

> sqrt.n.vec = sqrt(n.vec) # last for loop

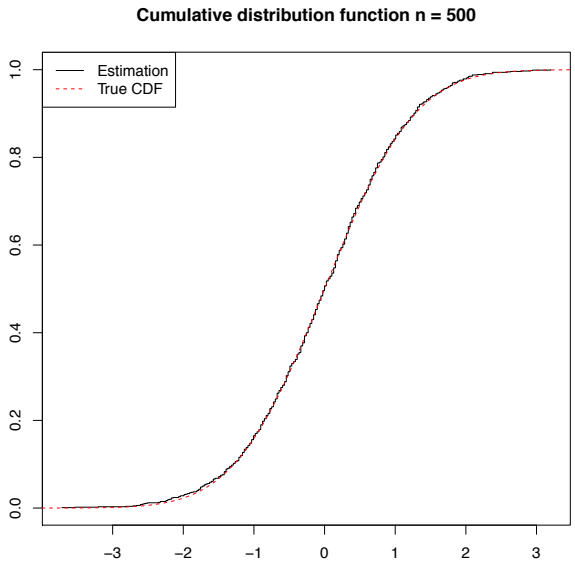
> top = xbar.vec - xexp
> bottom = sqrt(xvar)
> r.vec = sqrt.n.vec * top / bottom

> x.hist = seq(-4, 4, length.out = 100)
> par(mfrow = c(2,3))


> for (eps in sample.size.vec){
+   ss = r.vec[n.vec == eps]   # subset by n
+
```

```
+    tname =                        # title
+      bquote(bold(atop(
+        "Sample Size ="~.(eps), "1000 Repetition")))
+
+    hist(ss, freq = FALSE, xlab = "r", main = tname,
+         ylim = c(0, 0.6), xlim = c(-4, 4))
+
+    # Kernel Density Estimation
+    lines(density(ss), col = 4, lty = 2)
+    # True Density function
+    lines(x.hist, dnorm(x.hist), col = 2, lty = 1)
+
+    legend("top", col = c(1,2), lty = c(2,1),
+           legend = c("Estimation","True "))
+ }
```

- We can ask R to use sample quantiles to estimate the distribution function.




```
> true.norm = rnorm(num, mean = 0, sd = 1)
>
> hist(true.norm, freq = FALSE,
+      ylim = c(0, 0.6), xlim = c(-4, 4),
+      xlab = "Z", main = "1000 Z~N(0,1)")
>
> lines(x.hist, dnorm(x.hist), col = 2, lty = 1)
>
> par(mfrow = c(1,1))
```

```
> sample_quantile = sort(r.vec[n.vec == 500])
> sample_cdf = (1:num) / num
>
> plot(sample_quantile, sample_cdf, type = "s",
+      xlab = "", ylab = "", main =
+        "Cumulative distribution function n = 500")
> lines(x.hist, pnorm(x.hist), col = 2, lty = 2)
> legend("topleft", lty = c(1, 2), col = c(1, 2),
+        legend = c("Estimation", "True CDF"))
```

Definition

An estimator is consistent if

$$\hat{\theta}_n \to \theta \qquad \text{when} \quad n \to \infty$$

- The law of large numbers states $\bar{X}_n = \dfrac{1}{n}\sum_{i=1}^{n}$ is a consistent estimator of

  $$\mathbb{E}[X_i] \qquad \text{where } X_i \text{ are i.i.d.}$$

  as well as being an unbiased estimator of it.
- The central limit theorem states the sampling distribution is asymptotically normal with the variance $\dfrac{\sigma^2}{n}$, thus the standard error of $\hat{\theta} = \bar{X}$,

  $$\text{SE} = \frac{\sigma}{\sqrt{n}} \to 0 \qquad \text{when} \quad n \to \infty$$

2.3 LSE and MLE

- Consider two continuous random variables X and Y, e.g. height and weight.
- Suppose we have obtained n data points

  $$(x_1, y_1), \quad (x_2, y_2), \quad \cdots, \quad (x_n, y_n)$$

  and would like to construct a simple linear regression model

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

  where β_0 and β_1 are fixed but unknown, and ε is the random error term.
- For notational reasons, it is better to treat the observations

  $$x_1, x_2, \cdots x_n$$

  as realisations of n random variables, one for each observation

  $$X_1, X_2, \cdots X_n$$

- Similarly, $Y_1, Y_2, \cdots, Y_n$ denote random variables for $y_1, y_2, \cdots, y_n$.
- Note β_0 and β_1 are unknown to us, thus the realised errors are not observed,

$$e_i = y_i - (\beta_0 + \beta_1 x_i)$$

despite having observed x_i and y_i for all i.

- However, if we have some estimates of β_0 and β_1, then we can estimate it by

$$\hat{e}_i = y_i - (b_0 + b_1 x_i)$$

where b_0 and b_1 are estimates of β_0 and β_1, respectively.

- The estimate $\hat{e}_i$ is known as the residual, and the term

$$\hat{y}_i = b_0 + b_1 x_i$$

is known as the predicted/fitted value.

- Therefore, the observed y_i is the sum of the two components

$$y_i = \hat{y}_i + \hat{e}_i$$

- Recall either by the least square principle

$$(b_0, b_1) = \underset{(\beta_0, \beta_1)}{\text{Arg min}} \sum_{i=1}^{n} e_i^2$$

or by MSE criterion and using unbiased sample estimates, we have

$$b_0 = \bar{y} - b_1 \bar{x} \qquad \text{and} \qquad b_1 = \frac{c_{xy}}{s_x^2}$$

as the estimates of β_0 and β_1, respectively.

- In order to determine the properties of the two estimators, i.e.

unbiasedness and consistency

we have to refine our assumptions.

- Three common model assumptions for simple linear regression are:

1. The conditional mean is linear for all i, i.e.

$$\mathbb{E}\left[Y_i \mid X_i = x_i\right] = \beta_0 + \beta_1 x_i$$

2. The error term

$$\varepsilon_i = Y_i - \beta_0 - \beta_1 X_i$$

has a zero mean and a constant but unknown variance for all i, i.e.

$$\mathbb{E}\left[\varepsilon_i \mid X_i\right] = 0 \qquad \text{and} \qquad \text{Var}\left[\varepsilon_i \mid X_i\right] = \sigma^2$$

3. The error terms are uncorrelated with X_i for all i, i.e.

$$\text{Cov}\left[\varepsilon_i, X_i\right] = 0$$

and are uncorrelated to each other for all $i \neq j$, i.e.

$$\text{Cov}\left[\varepsilon_i, \varepsilon_j\right] = 0$$

Definition

The bias of an estimator $\hat{\theta}$ for a fixed unknown parameter θ is given by

$$\text{Bias}\left(\hat{\theta}, \theta\right) = \mathbb{E}\Big[\hat{\theta} - \theta\Big]$$

and the estimator is said to be unbiased if its bias is always zero.

- Consider our estimator for β_1,

$$\hat{\beta}_1 = \frac{\frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2} \qquad \text{where} \quad \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad \bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

Q: How can we show it is unbiased?

It can be show that $\hat{\beta}_1$ is a random variable in terms of X_i and ε_i for all i, thus

$$\mathbb{E}\left[\hat{\beta}_1\right] = \mathbb{E}_X\Big[\mathbb{E}_\varepsilon\left[\hat{\beta}_1 \mid X_1 = x_1, \dots X_n = x_n\right]\Big]$$

Specifically, we will show in a minute that

$$\mathbb{E}\left[\hat{\beta}_1\right] = \beta_1 + \mathbb{E}\left[\frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\varepsilon_i}{W}\right] \qquad \text{where} \quad W = \sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2$$

$$= \beta_1 + \mathbb{E}_X\left[\sum_{i=1}^{n}\frac{X_i - \bar{X}}{W}\mathbb{E}_\varepsilon\left[\varepsilon_i \mid X_1 = x_1, \dots X_n = x_n\right]\right] = \beta_1$$

$$
\begin{aligned}
\hat{\beta}_1 &= \frac{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\left(Y_i-\bar{Y}\right)}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2} \\
&= \frac{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\left(\beta_0+\beta_1 X_i+\varepsilon_i-\frac{1}{n}\sum_{i=1}^{n}\beta_0+\beta_1 X_i+\varepsilon_i\right)}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2} \\
&= \frac{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\left[\beta_1\left(X_i-\bar{X}\right)+\left(\varepsilon_i-\bar{\varepsilon}\right)\right]}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2} \\
&= \frac{\beta_1\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2+\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\left(\varepsilon_i-\bar{\varepsilon}\right)}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2} \\
&= \beta_1+\frac{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\varepsilon_i-\bar{\varepsilon}\sum_{i=1}^{n}\left(X_i-\bar{X}\right)}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2} \\
&= \beta_1+\frac{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\varepsilon_i-\bar{\varepsilon}\left(n\bar{X}-n\bar{X}\right)}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2} = \beta_1+\frac{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2}
\end{aligned}
$$

Definition

An estimator $\hat{\theta}_n$ for a fixed unknown parameter θ is said to be consistent if

$$\hat{\theta}_n \to \theta \qquad \text{as} \qquad n \to \infty$$

- Consider our estimator for β_1 again and notice it depends on n,

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}\left(X_i-\bar{X}_n\right)\left(Y_i-\bar{Y}_n\right)}{\sum_{i=1}^{n}\left(X_i-\bar{X}_n\right)^2} \qquad \text{where} \qquad \bar{X}_n = \frac{1}{n}\sum_{i=1}^{n}X_i, \quad \bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n}Y_i$$

Q: How can we show it is consistent?

Consider the sample variance for n observations,

$$\begin{aligned} s_x^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x}_n)^2 \\ &= \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i^2-2x_i\bar{x}_n+(\bar{x}_n)^2\right) \\ &= \frac{n}{n-1}\left(\frac{1}{n}\sum_{i=1}^{n}x_i^2-2\bar{x}_n\frac{1}{n}\sum_{i=1}^{n}x_i+(\bar{x}_n)^2\right)=\frac{n}{n-1}\left(\overline{x_n^2}-(\bar{x}_n)^2\right) \end{aligned}$$

Consider the sample covariance

$$\begin{aligned} c_{xy} &= \frac{1}{n-1}\sum_{i}^{n}(x_i-\bar{x}_n)(y_i-\bar{y}_n) \\ &= \frac{1}{n-1}\left(\sum_{i}^{n}x_iy_i-n\bar{x}_n\bar{y}_n\right) \\ &= \frac{1}{n-1}\left(\sum_{i}^{n}x_iy_i-\bar{x}_n\sum_{i}^{n}y_i\right) \\ &= \frac{1}{n-1}\left(\sum_{i}^{n}x_i(\beta_0+\beta_1x_i+r_i)-\bar{x}_n\sum_{i}^{n}(\beta_0+\beta_1x_i+r_i)\right) \\ &= \frac{n}{n-1}\left(\beta_0\bar{x}_n+\beta_1\overline{x_n^2}+\overline{(xr)}_n-\beta_0\bar{x}_n-\beta_1(\bar{x}_n)^2-\bar{x}_n\bar{r}_n\right) \\ &= \beta_1\frac{n}{n-1}\left(\overline{x_n^2}-(\bar{x}_n)^2\right)+\frac{n}{n-1}\overline{(xr)}_n-\frac{n}{n-1}\bar{x}_n\bar{r}_n \end{aligned}$$

According to model assumptions errors have constant $\mathbb{E}[\varepsilon_i]=0$ and variance, and are not correlated with each other, so the law of large numbers says

$$\bar{r}_n \to \mathbb{E}[\varepsilon_i]=0 \qquad \text{when} \quad n\to\infty$$

And ε_i is not correlated with X_i either, so $X_i\varepsilon_i$ are uncorrelated with $X_j\varepsilon_j$

$$\begin{aligned} \operatorname{Cov}[X_i\varepsilon_i, X_j\varepsilon_j] &= \mathbb{E}[X_i\varepsilon_iX_j\varepsilon_j]-\mathbb{E}[X_i\varepsilon_i]\,\mathbb{E}[X_j\varepsilon_j] \\ &= \mathbb{E}[X_i\varepsilon_iX_j]\,\mathbb{E}[\varepsilon_j]-\mathbb{E}[X_i]\,\mathbb{E}[\varepsilon_i]\,\mathbb{E}[X_i]\,\mathbb{E}[\varepsilon_i]=0 \qquad \text{for} \quad i\neq j \end{aligned}$$

It is not difficult to see the it has zero mean and a constant variance,

$$\begin{aligned} \mathbb{E}[X_i\varepsilon_i] &= \mathbb{E}[X_i]\cdot\mathbb{E}[\varepsilon_i]=0 \\ \operatorname{Var}[X_i\varepsilon_i] &= \operatorname{Var}[X_i]\operatorname{Var}[\varepsilon_i]+\operatorname{Var}[X_i]\,\mathbb{E}[\varepsilon_i]^2+\operatorname{Var}[\varepsilon_i]\,\mathbb{E}[X_i]^2 \end{aligned}$$

So the law of large numbers guarantees the following, as $n\to\infty$, thus $\hat{\beta}_1\to\beta_1$

$$\overline{(xr)}_n \to 0 \qquad \text{when} \quad n\to\infty$$

It follows that $\hat{\beta}_0=\bar{y}_n-\hat{\beta}_1\bar{x}_n=\beta_0+\beta_1\bar{x}_n+\bar{r}_n-\hat{\beta}_1\bar{x}_n\to\beta_0 \qquad \text{when} \quad n\to\infty$

- In order to do inferences, i.e.

  hypothesis testing, confidence interval, and prediction interval

we have to make slightly stronger assumptions about the error terms ε_i:

1. The conditional mean is linear for all i, i.e.

$$\mathbb{E}\left[Y_i \mid X_i = x_i\right] = \beta_0 + \beta_1 x_i$$

2. The error term

$$\varepsilon_i = Y_i - \beta_0 - \beta_1 X_i$$

has a zero mean and a constant but unknown variance for all i, i.e.

$$\mathbb{E}\left[\varepsilon_i \mid X_i\right] = 0 \qquad \text{and} \qquad \text{Var}\left[\varepsilon_i \mid X_i\right] = \sigma^2$$

3. The error terms ε_i are independent of X_i, and of each other, for all i.
4. The error terms ε_i follow a normal distribution.

Q: Why do you think we need the additional assumption?

$$\begin{aligned} f_{\varepsilon_i}(e_i) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{e_i^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(y_i - (\beta_0 + \beta_1 x_i)\right)^2}{2\sigma^2}\right) \end{aligned}$$

Q: What is the significance of modifying assumption 3.?

- Being independent means the likelihood function given the observed data is

$$\mathcal{L}(\beta_0, \beta_1, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(y_i - (\beta_0 + \beta_1 x_i)\right)^2}{2\sigma^2}\right)$$

from which we can derive the maximum likelihood estimates of $(\beta_0, \beta_1, \sigma^2)$

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}}{\text{Arg}\max}\, \mathcal{L}(\boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ denotes a vector of all unknown parameters.

- It is usually easier to work with the negative log-likelihood,

$$\ell(\boldsymbol{\theta}) = -\ln \mathcal{L}(\boldsymbol{\theta})$$

for which we minimise instead of maximising with respect to $\boldsymbol{\theta}$ to obtain

$$\boldsymbol{\theta}_{MLE}$$

- In this case, it can be shown that the negative log-likelihood is given by

$$\begin{aligned} \ell(\beta_0, \beta_1, \sigma^2) &= \sum_{i=1}^{n} \left(\frac{1}{2}\ln(2\pi) + \ln\sigma + \frac{(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}\right) \\ &= \frac{n}{2}\ln(2\pi) + n\ln\sigma + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 \end{aligned}$$

Q: What are the maximum likelihood estimates of β_0 and β_1?

- The maximum likelihood estimates of β_0 and β_1 are identical to the previous estimates we had, for which we have shown them to unbiased and consistent.
- By differentiating ℓ with respect to σ and setting it to zero, we have

$$\frac{n}{\sigma} - \frac{1}{\sigma^3}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 = 0$$

solving and using the estimates for β_0 and β_1, we have the following estimate

$$\sigma^2_{MLE} = \frac{1}{n}\sum_{i=1}^{n}\hat{e}_i^2$$

- However, the estimator that gives σ^2_{MLE} can be shown to have a small bias,

$$\mathbb{E}\left[\sigma^2_{MLE}\right] = \frac{n-2}{n}\sigma^2$$

- So the following estimate is unbiased and can also be shown to be consistent

$$\sigma^2_u = \frac{n}{n-2}\sigma^2_{MLE} = \frac{1}{n-2}\sum_{i=1}^{n}\hat{e}_i^2$$

- Recall we have shown the estimator for β_1 can be written as

$$\hat{\beta}_1 = \beta_1 + \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}_n\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}_n\right)^2}$$

Q: Can you see $\hat{\beta}_1$ follows a normal distribution with mean of β_1 and variance of

$$\frac{\sigma^2}{(n-1)s_x^2} \qquad \text{where} \quad s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

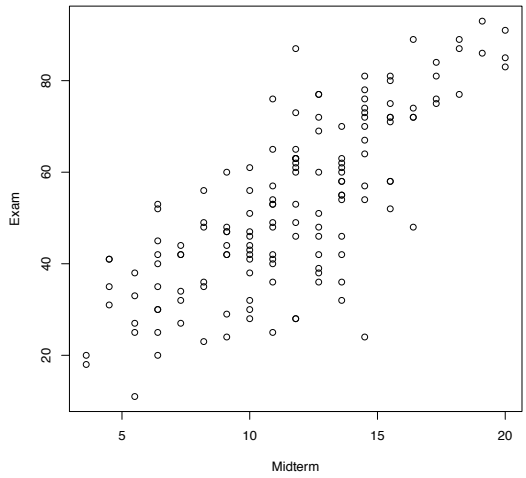
conditional on the observed data $X_1 = x_1, X_2 = x_2, \cdots X_n = x_n$.

It is essentially a sum of normal random variables, thus normal, for the variance,

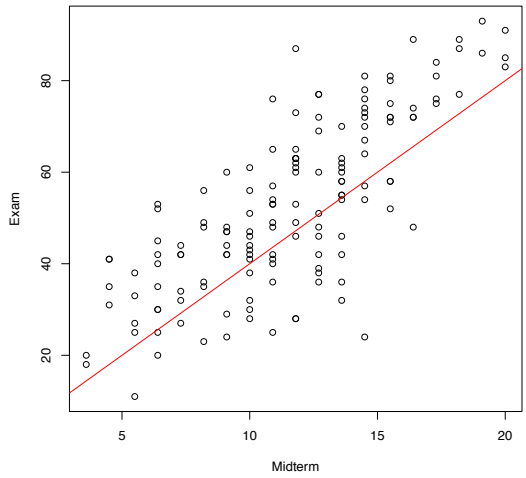
$$\begin{aligned}
\text{Var}\left[\hat{\beta}_1 \mid X_1 = x_1, \cdots X_n = x_n\right] &= \text{Var}\left[\beta_1 + \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}_n\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}_n\right)^2} \mid X_1 = x_1, \cdots X_n = x_n\right] \\
&= \text{Var}\left[\sum_{i=1}^{n}\frac{\left(X_i - \bar{X}_n\right)}{W}\varepsilon_i \mid X_1 = x_1, \cdots X_n = x_n\right] \\
&= \sum_{i=1}^{n}\frac{\left(X_i - \bar{X}\right)^2}{W^2}\text{Var}\left[\varepsilon_i \mid X_1 = x_1, \cdots X_n = x_n\right] \\
&= \frac{\sigma^2}{W}
\end{aligned}$$

- This means, if σ^2 is somehow known, we can use the normal distribution to do hypothesis testing and construct confidence interval for β_1, if σ^2 is not known, then we can use a t-distribution to do so.
- We have only discussed β_1, however, similarly ideas can be applied to β_0.

Q: What do you think the relationship between Midterm and Final Exam is?



- The straight line $y = 4x$ seems to be reasonable to my eye.



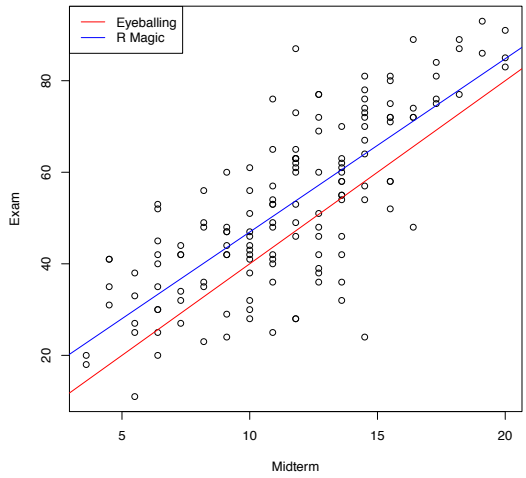

```
> # Load data locally
```

```
> course.df = read.table("~/Desktop/course.txt",
+                        header = TRUE)
>
> plot(Exam~Midterm, data = course.df)
>
> abline(a = 0, b = 4, col = "red")
```

- Of course, R can "automatically" construct the simple linear model for us

```
> course.lm = lm(Exam~Midterm, data = course.df)
>
> abline(course.lm, col = "blue")
>
> legend("topleft", legend =
+          c("Eyeballing", "R Magic"),
+        lty = 1, col = c(2, 4))
```

- Eyeballing is not too far from the truth!



- Of course, we can confirm what R did by manually computing the estimates.

```
> x = course.df$Midterm
>
> y = course.df$Exam

> beta_1_hat = cov(x, y) / var(x)

> xbar = mean(x); ybar = mean(y)

> beta_0_hat = ybar - beta_1_hat * xbar
```

```
> beta_0_hat; beta_1_hat
```

```
[1] 9.084463
[1] 3.785924
```

```
> beta_0_hat; beta_1_hat
```

```
[1] 9.084463
[1] 3.785924
```

```
> summary(course.lm)
```

```
Call:
lm(formula = Exam ~ Midterm, data = course.df)

Residuals:
    Min      1Q  Median      3Q     Max
-39.980  -6.471   0.826   8.575  33.242

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)   9.0845     3.2204   2.821  0.00547 **
Midterm       3.7859     0.2647  14.301  < 2e-16 ***
---
Signif. codes:  0 `***` 0.001 `**` 0.01 `*` 0.05 `.` 0.1 ` ` 1

Residual standard error: 12.05 on 144 degrees of freedom
Multiple R-squared:  0.5868,    Adjusted R-squared:  0.5839
F-statistic: 204.5 on 1 and 144 DF,  p-value: < 2.2e-16
```

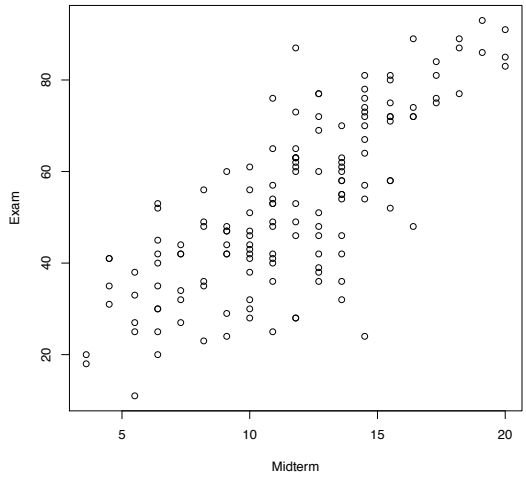
Q: Why can we not trust the above results at this stage?

2.4 Diagnostics

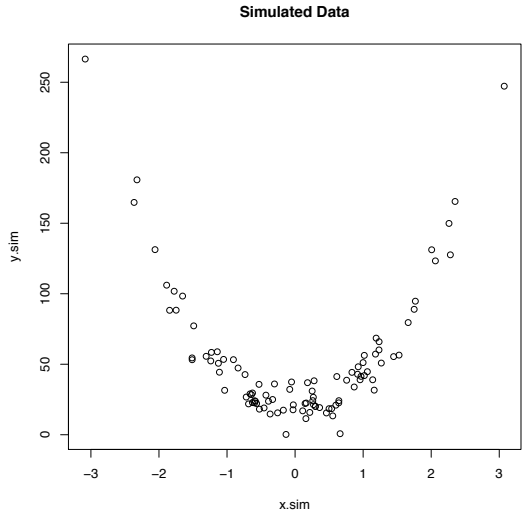
Q: How can we check assumption 1. graphically? What do you expect to see?

1. The conditional mean of the response is linear for all i, i.e.

$$\mathbb{E}\left[Y_i \mid X_i = x_i\right] = \beta_0 + \beta_1 x_i$$



- The following plot of response and regressor provides evidence of nonlinearity



Q: However, do we know for sure that assumption 1. is not adequate?

- Alternatively, nonlinearity is also evident in a plot of

$$y_i \quad \text{Vs} \quad \hat{y}_i$$

Q: What do you expect to see in such plots if assumption 1. is fine?

But we again expect points to be scattered around the diagonal line instead of forming a pattern that is systematically off the diagonal line if the linearity assumption is fine.

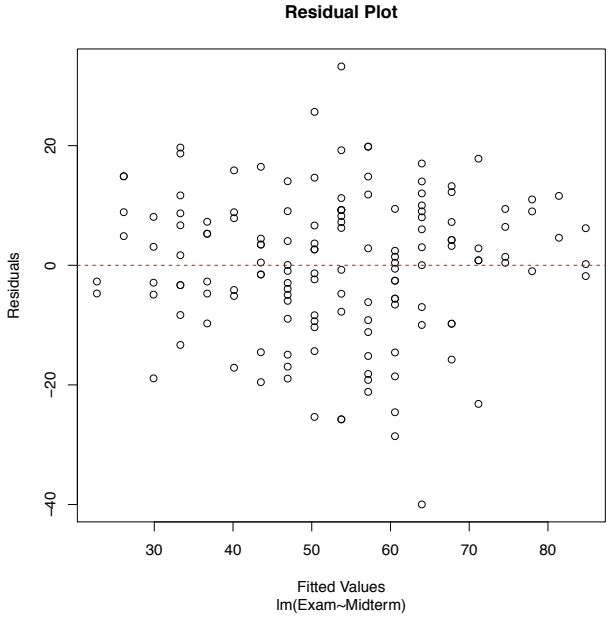
- In practice, to avoid the visual distraction of a sloping pattern, we plot

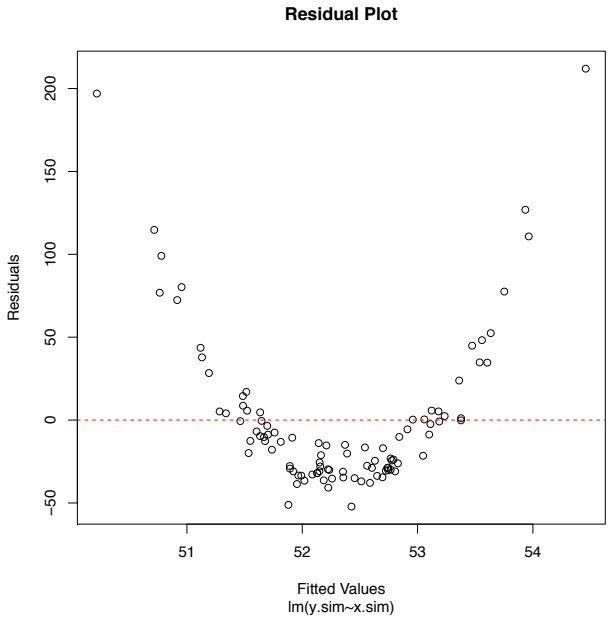
$$\hat{e}_i \quad \text{Vs} \quad \hat{y}_i$$

Q: What do you expect to see in such plots if assumption 1. is fine?

- Having non-flat band of points suggests that 1. might be inadequate.

```
> plot(fitted.values(course.lm),
+      residuals(course.lm),
+      xlab = "Fitted Values", ylab = "Residuals",
+      main = "Residual Plot",
+      sub = "lm(Exam~Midterm)")
>
> abline(a = 0, b = 0, lty = 2, col = "red")
```





Q: How can we check assumption 2. graphically? What do you expect to see?

2. The error has a zero mean and a constant but unknown variance for all i, i.e.

$$\mathbb{E}\left[\varepsilon_i \mid X_i\right] = 0 \qquad \text{and} \qquad \operatorname{Var}\left[\varepsilon_i \mid X_i\right] = \sigma^2$$

Q: If assumption 2. is satisfied, what do you expect to see in a plot of

$$e_i \qquad \text{Vs} \qquad \hat{y}_i$$

- However, e_i are not directly observed, we have to consider the residuals

$$\hat{e}_i = y_i - \hat{y}_i = \beta_0 + \beta_1 x_i + e_i - b_0 - b_1 x_i = (\beta_0 - b_0) + (\beta_1 - b_1)\, x_i + e_i$$

- Notice e_i and $\hat{e}_i$ are not the same, however, it can be shown that

$$\text{Var}\left[Y_i - \hat{Y}_i \mid X_i\right] = \sigma^2 \left[1 - \frac{1}{n} - \frac{\left(X_i - \bar{X}_n\right)^2}{\sum_{i=1}^{n} \left(X_i - \bar{X}_n\right)^2}\right]$$

Q: So what do you expect to see in a plot of $\hat{e}_i$ and x_i if assumption 2. is fine?

Given the followings

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X} \qquad \text{and} \qquad \hat{\beta}_1 = \beta_1 + \sum_{i=1}^{n} \frac{\left(X_i - \bar{X}\right) \varepsilon_i}{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)^2}$$

Since $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$, we have

$$\begin{aligned}\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X} &= \frac{1}{n}\sum_{i=1}^{n}\left(\beta_0 + \beta_1 X_i + \varepsilon_i\right) - \bar{X}\left(\beta_1 + \sum_{i=1}^{n}\frac{\left(X_i - \bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right) \\ &= \beta_0 + \bar{\varepsilon} - \sum_{i=1}^{n}\frac{\bar{X}\left(X_i - \bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\end{aligned}$$

from which, it is clear that

$$\mathbb{E}\left[\hat{\beta}_0 \mid X\right] = \beta_0 + \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\varepsilon_i \mid X\right] - \sum_{i=1}^{n}\frac{\bar{X}\left(X_i - \bar{X}\right)}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\left[\varepsilon_i \mid X\right] = \beta_0$$

According to the following property of variance

$$\operatorname{Var}\left[\sum_{i}^{n} a_i X_i\right] = \sum_{i}^{n} a_i^2 \operatorname{Var}\left[X_i\right] + \sum_{1 \le i < j \le n} 2 a_i a_j \operatorname{Cov}\left[X_i, X_j\right]$$

we have,

$$\begin{aligned}\operatorname{Var}\left[\hat{\beta}_0 \mid X\right] &= \frac{1}{n^2}\sum_{i=1}^{n}\operatorname{Var}\left[\varepsilon_i \mid X\right] + \sum_{i=1}^{n}\frac{\bar{X}^2\left(X_i - \bar{X}\right)^2}{\left(\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2\right)^2}\operatorname{Var}\left[\varepsilon_i \mid X\right] \\ &\qquad - \frac{2}{n}\sum_{i=1}^{n}\frac{\bar{X}\left(X_i - \bar{X}\right)}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\operatorname{Cov}\left[\varepsilon_i, \varepsilon_j\right] \\ &= \frac{\sigma^2}{n} + \frac{\bar{X}^2\sigma^2}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2} - \frac{2\bar{X}\sigma^2}{n\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\sum_{i=1}^{n}\left(X_i - \bar{X}\right) \\ &= \sigma^2\left(\frac{1}{n} + \frac{\bar{X}^2}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right)\end{aligned}$$

and

$$\begin{aligned}\operatorname{Var}\left[Y_i - \hat{Y}_i \mid X\right] &= \operatorname{Var}\left[\left(\beta_0 - \hat{\beta}_0\right) + \left(\beta_1 - \hat{\beta}_1\right)X_i + \varepsilon_i \mid X\right] \\ &= \operatorname{Var}\left[\hat{\beta}_0 \mid X\right] + X_i^2 \operatorname{Var}\left[\hat{\beta}_1 \mid X\right] + \operatorname{Var}\left[\varepsilon_i \mid X\right] \\ &\quad + 2X_i \operatorname{Cov}\left[\hat{\beta}_0, \hat{\beta}_1 \mid X\right] \\ &\quad + 2\operatorname{Cov}\left[\left(\left(\beta_0 - \hat{\beta}_0\right) + \left(\beta_1 - \hat{\beta}_1\right)X_i\right), \varepsilon_i \mid X\right]\end{aligned}$$

For the covariances, recall the following properties of covariance

$$\begin{aligned}\operatorname{Cov}[X+a, Y+b] &= \operatorname{Cov}[X, Y] \\ \operatorname{Cov}[aX+bY, cW+dV] &= ac\operatorname{Cov}[X, W] + ad\operatorname{Cov}[X, V] \\ &\quad + bc\operatorname{Cov}[Y, W] + bd\operatorname{Cov}[Y, V]\end{aligned}$$

Substituting $\hat{\beta}_0$ and $\hat{\beta}_1$ into the covariance $\operatorname{Cov}\left[\hat{\beta}_0, \hat{\beta}_1 \mid X\right]$, we have

$$\begin{aligned}&\operatorname{Cov}\left[\left(\beta_0 + \bar{\varepsilon} - \sum_{i=1}^{n} \frac{\bar{X}\left(X_i - \bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right), \left(\beta_1 + \sum_{i=1}^{n} \frac{\left(X_i - \bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right) \mid X\right] \\ &= \operatorname{Cov}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i - \sum_{i=1}^{n} \frac{\bar{X}\left(X_i - \bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right), \left(\sum_{i=1}^{n} \frac{\left(X_i - \bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right) \mid X\right] \\ &= \frac{1}{n}\operatorname{Cov}\left[\left(\sum_{i=1}^{n}\varepsilon_i\right), \left(\sum_{i=1}^{n} \frac{\left(X_i - \bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right) \mid X\right] \\ &\qquad - \operatorname{Cov}\left[\left(\sum_{i=1}^{n} \frac{\bar{X}\left(X_i - \bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right), \left(\sum_{i=1}^{n} \frac{\left(X_i - \bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right) \mid X\right] \\ &= \frac{\sigma^2}{n}\sum_{i=1}^{n} \frac{\left(X_i - \bar{X}\right)}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2} - \bar{X}\sigma^2\sum_{i=1}^{n} \frac{\left(X_i - \bar{X}\right)^2}{\left(\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2\right)^2} \\ &= \frac{\sigma^2}{n}\cdot 0 - \frac{\bar{X}\sigma^2}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2} = \frac{-\bar{X}\sigma^2}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\end{aligned}$$

Using the property $\operatorname{Cov}[X, Y] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$ and $\mathbb{E}[\varepsilon_i \mid X] = 0$, thus

$$\begin{aligned}&\operatorname{Cov}\left[\left(\left(\beta_0 - \hat{\beta}_0\right) + \left(\beta_1 - \hat{\beta}_1\right)X_i\right), \varepsilon_i \mid X\right] \\ &= \mathbb{E}\left[\varepsilon_i\left(\left(\beta_0 - \hat{\beta}_0\right) + \left(\beta_1 - \hat{\beta}_1\right)X_i\right) \mid X\right] \\ &= -\mathbb{E}\left[\varepsilon_i\hat{\beta}_0 \mid X\right] - X_i\mathbb{E}\left[\varepsilon_i\hat{\beta}_1 \mid X\right] \\ &= -\mathbb{E}\left[\varepsilon_i\left(\bar{Y} - \hat{\beta}_1\bar{X}\right) \mid X\right] - X_i\mathbb{E}\left[\varepsilon_i\hat{\beta}_1 \mid X\right] \\ &= -\mathbb{E}\left[\varepsilon_i\bar{Y} \mid X\right] - \left(X_i - \bar{X}\right)\mathbb{E}\left[\varepsilon_i\hat{\beta}_1 \mid X\right]\end{aligned}$$

Continuing, we have

$$\begin{aligned}
\mathbb{E}\left[\varepsilon_i \bar{Y} \mid X\right] &= \mathbb{E}\left[\varepsilon_i \left(\beta_0 + \beta_1 \bar{X} + \bar{\epsilon}\right) \mid X\right] \\
&= 0 + 0 + \frac{1}{n}\mathbb{E}\left[\varepsilon_i \sum_{i=1}^{n} \varepsilon_i\right] \\
&= \frac{\sigma^2}{n}
\end{aligned}$$

since

$$\begin{aligned}
\operatorname{Var}\left[\varepsilon_i \mid X\right] &= \mathbb{E}\left[\varepsilon_i^2 \mid X\right] - \left(\mathbb{E}\left[\varepsilon_i \mid X\right]\right)^2 \\
&= \mathbb{E}\left[\varepsilon_i^2 \mid X\right] \\
&= \sigma^2
\end{aligned}$$

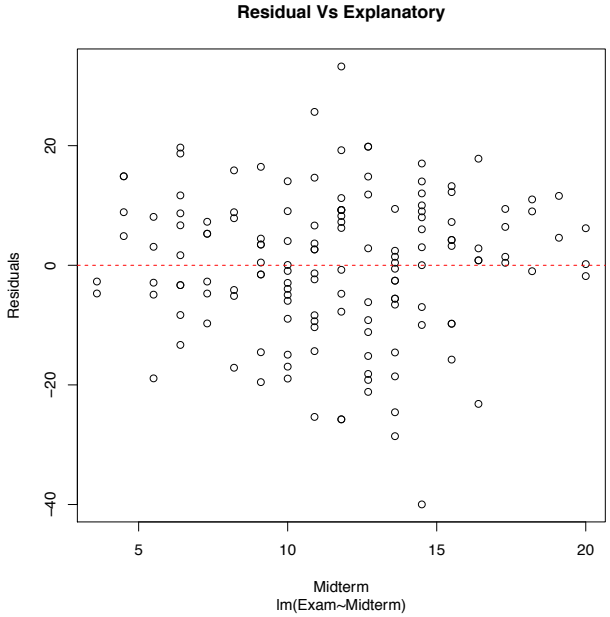
$$\begin{aligned}
\operatorname{Cov}\left[\varepsilon_i, \varepsilon_j \mid X\right] &= \mathbb{E}\left[\varepsilon_i \varepsilon_j \mid X\right] - \mathbb{E}\left[\varepsilon_i \mid X\right] \mathbb{E}\left[\varepsilon_j \mid X\right] \\
&= \mathbb{E}\left[\varepsilon_i \varepsilon_j \mid X\right] \\
&= 0 \qquad \text{for} \quad i \neq j
\end{aligned}$$

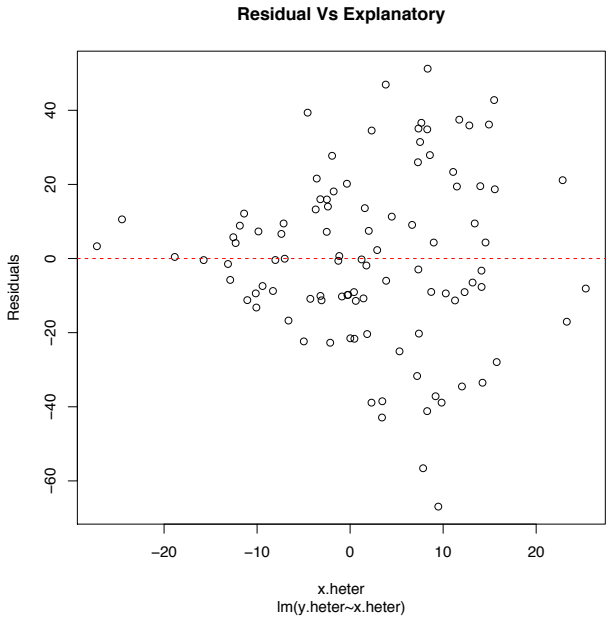
and

$$\mathbb{E}\left[\varepsilon_i \hat{\beta}_1 \mid X\right] = \mathbb{E}\left[\varepsilon_i \beta_1 + \varepsilon_i \sum_{i=1}^{n} \frac{\left(X_i - \bar{X}\right)\varepsilon_i}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2} \mid X\right] = \frac{\sigma^2\left(X_i - \bar{X}\right)}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}$$

Now putting everything together, we have the stated result,

$$\begin{aligned}
\operatorname{Var}\left[Y_i - \hat{Y}_i \mid X\right] &= \operatorname{Var}\left[\hat{\beta}_0 \mid X\right] + X_i^2 \operatorname{Var}\left[\hat{\beta}_1 \mid X\right] + \operatorname{Var}\left[\varepsilon_i \mid X\right] \\
&\quad + 2X_i \operatorname{Cov}\left[\hat{\beta}_0, \hat{\beta}_1 \mid X\right] - 2\mathbb{E}\left[\varepsilon_i \bar{Y} \mid X\right] \\
&\quad - 2\left(X_i - \bar{X}\right)\mathbb{E}\left[\varepsilon_i \hat{\beta}_1 \mid X\right] \\
&= \sigma^2\left(\frac{1}{n} + \frac{\bar{X}^2}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right) + \frac{\sigma^2 X_i^2}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2} + \sigma^2 \\
&\quad - \sigma^2 \frac{2X_i\bar{X}}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2} - \sigma^2\frac{2}{n} - \sigma^2 \frac{2\left(X_i - \bar{X}\right)^2}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2} \\
&= \sigma^2\left(1 - \frac{1}{n} - \frac{\left(X_i - \bar{X}\right)^2}{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}\right)
\end{aligned}$$





Q: How can we check the third assumption?

3. The errors are independent of X_i, and of each other.

- You might be tempted to use the fact the following must be zero if 3. is true

$$\text{Cov}\left[X_i, \varepsilon_i\right] = 0$$

and thus the sample covariance shall not be too far away from zero, that is,

$$\sum_{i=1}^{n} \left(e_i - \bar{e}_n\right)\left(x_i - \bar{x}_n\right) = \sum_{i=1}^{n} e_i \left(x_i - \bar{x}_n\right) \approx 0$$

which is correct, however, you might wrongly jump to the conclusion to test

$$\sum_{i=1}^{n} \left(\hat{e}_i - \bar{\hat{e}}_n\right)\left(x_i - \bar{x}_n\right) = \sum_{i=1}^{n} \hat{e}_i \left(x_i - \bar{x}_n\right)$$

Q: Why is this not going to provide any information about $\text{Cov}\left[X_i, \varepsilon_i\right]$?

- Recall b_0 and b_1 are solutions to the following estimation equations

$$\sum_{i=1}^{n} \left(y_i - b_0 - b_1 x_i\right) = 0 \implies \sum_{i=1}^{n} \hat{e}_i = 0$$
$$\sum_{i=1}^{n} \left(y_i - b_0 - b_1 x_i\right)\left(x_i\right) = 0 \implies \sum_{i=1}^{n} \hat{e}_i x_i = 0$$

Q: Why does the sample covariance between residual and X is always 0?

$$\sum_{i=1}^{n} \hat{e}_i x_i - \bar{x}\sum_{i=1}^{n} \hat{e}_i - \bar{\hat{e}}\sum_{i=1}^{n}(x_i - \bar{x}) = 0 \implies \sum_{i=1}^{n} \left(\hat{e}_i - \bar{\hat{e}}\right)\left(x_i - \bar{x}_n\right) = 0$$

so checking the sample covariance is not a detection tool for the dependence.

- In practice, the assumption on the dependence of ε on X is intimately linked to assumption 1., so we don't run additional diagnostics for it.

$$y_i = \mathbb{E}\left[Y_i \mid X_i = x_i\right] + \varepsilon_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Q: How about the part says that errors are independent?

Q: Can we rely on residuals to detect correlated errors?

Q: Will residuals be independent if errors are independdent?

$$\sum_{i=1}^{n} \hat{e}_i = 0$$

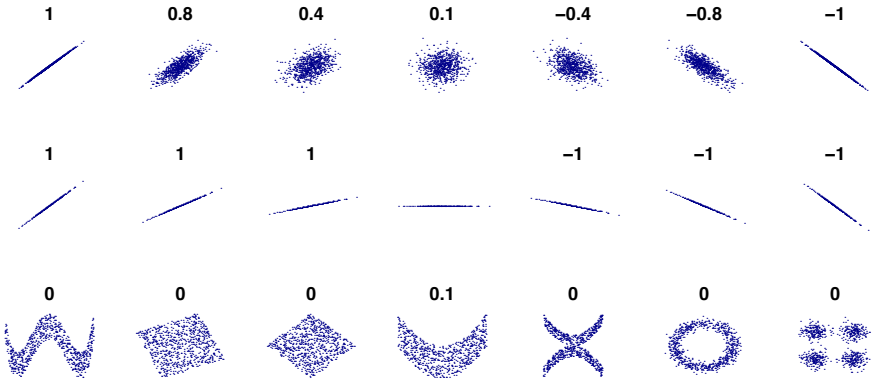
- We expect the sample covariance between $\hat{e}_i$ and $\hat{e}_j$ to be small but nonzero.
- Given all the residuals except one, we can always work out the last one, thus they must be correlated. However, the correlation will disappear as $n \to \infty$.
- So, when there is enough data, we do not expect to see a large correlation

$$\text{between residual } \hat{e}_i \text{ and previous residual } \hat{e}_{i-1}$$

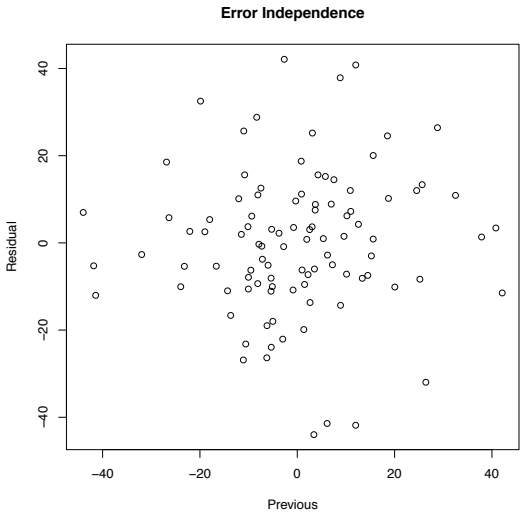
Q: How can we detect excessive correlation in the residuals, thus dependence?

Q: How about nonlinear dependence between $\hat{e}_i$ and $\hat{e}_{i-1}$?

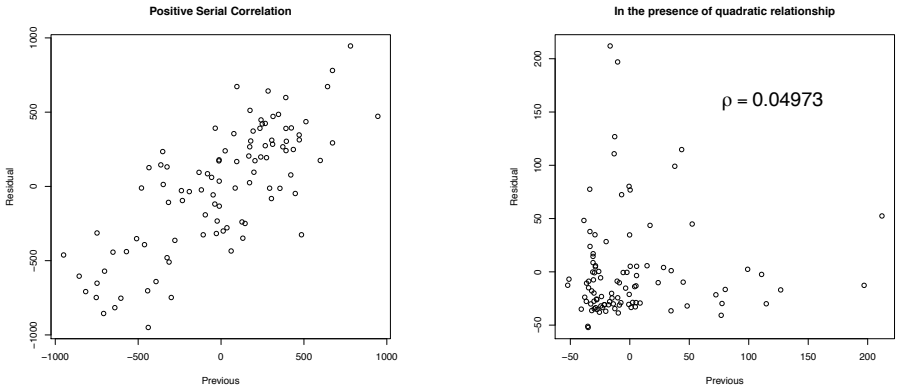
- Plotting residuals is a way to detect severe dependence, especially nonlinear.



- Recall Pearson correlation coefficient is only a measure of linear correlation.
- We are hoping for random scatter, i.e. no particular linear/nonlinear pattern.



- A clear pattern in the plot indicates 1. or 3. is violated.



- Of course, we need to check not only just neighbouring residuals in general,

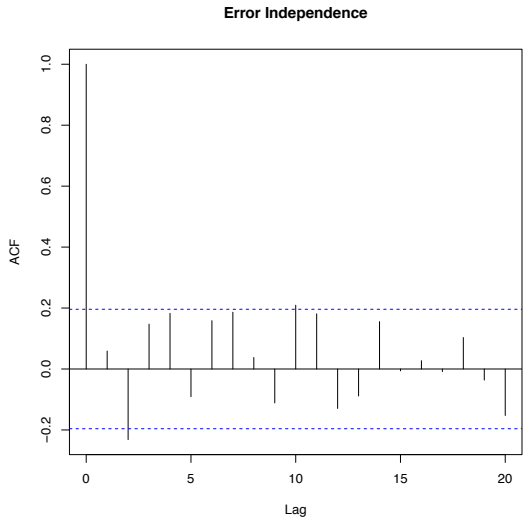
$$\hat{R}(k) = \frac{(n-1)\sum_{i=1}^{n-k}\left(\hat{e}_i - \frac{1}{n}\sum_{i=1}^{n}\hat{e}_i\right)\left(\hat{e}_{i+k} - \frac{1}{n}\sum_{i=1}^{n}\hat{e}_i\right)}{(n-k)\sum_{i=1}^{n}\left(\hat{e}_i - \frac{1}{n}\sum_{i=1}^{n}\hat{e}_i\right)^2}$$

  which is an estimate of autocorrelation function.

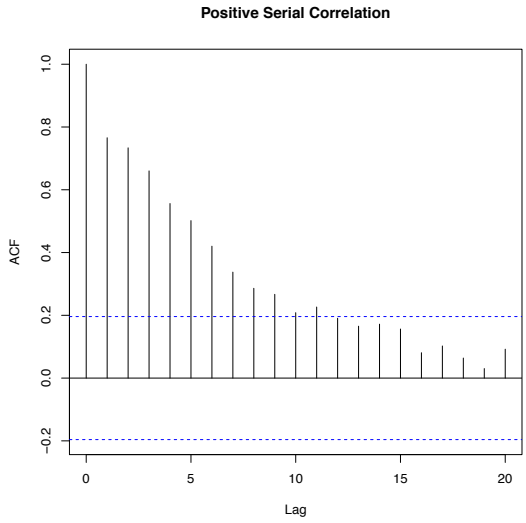
- Under our choice of $\hat{\beta}_0$ and $\hat{\beta}_1$, this estimate can be simplified to

$$\hat{R}(k) = \frac{(n-1)\sum_{i=1}^{n-k}\hat{e}_i\hat{e}_{i+k}}{(n-k)\sum_{i=1}^{n}\hat{e}_i^2}$$

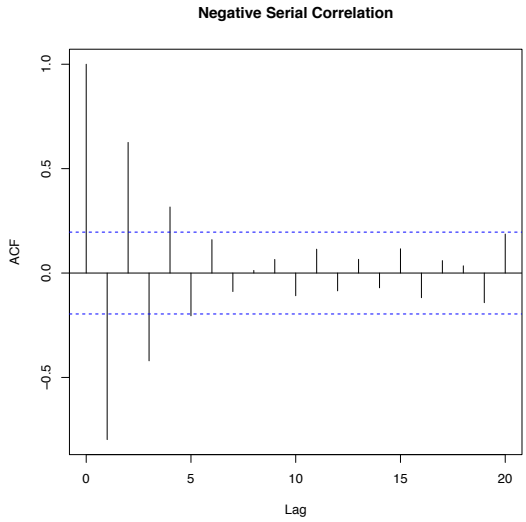
- R can easily compute and plot this estimate for various k,



- We expect the estimates to be small and patternless if errors are independent



- If not, then we have evidence that errors are not independent.



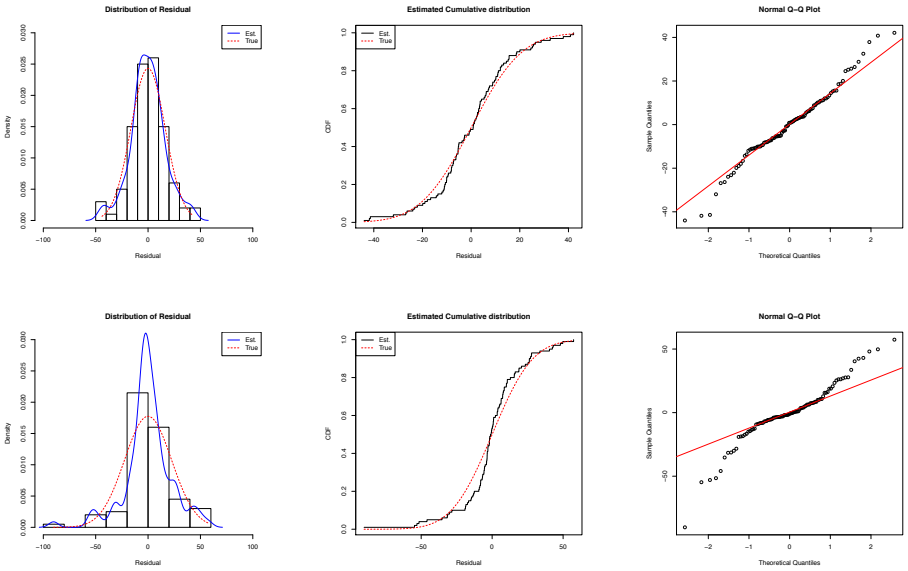

```
> # Generating autocorrelated data
> y.psc.vec = double(n)
> y.nsc.vec = double(n)
>
> y.psc.vec[1] = rnorm(1, mean = msl[1])
> y.nsc.vec[1] = rnorm(1, mean = msl[1])
>
> for (i in 2:n) {
+   y.psc.vec[i] =
+     rnorm(1, mean = msl[i] + 0.8 * y.psc.vec[i-1])
>
+   y.nsc.vec[i] =
+     rnorm(1, mean = msl[i] - 0.8 * y.nsc.vec[i-1])
+ }
>
> psc.lm = lm(y.psc.vec~x.new.vec)
> nsc.lm = lm(y.nsc.vec~x.new.vec)


# Plotting ehat_i against ehat_{i-1}, and acf
> plot(psc.lm$residuals[-n],psc.lm$residuals[-1],
+      xlab = "Previous", ylab = "Residual",
+      main = "Positive Serial Correlation")
> plot(lmlist[[1]]$residuals[-n],
+      lmlist[[1]]$residuals[-1],
+      xlab = "Previous", ylab = "Residual",
+      main = "Error Independence")
> plot(nsc.lm$residuals[-n],nsc.lm$residuals[-1],
+      xlab = "Previous", ylab = "Residual",
+      main = "Negative Serial Correlation")
```

```
>
> acf(psc.lm$residuals,
+     main = "Positive Serial Correlation")
> acf(lmlist[[1]]$residuals,
+     main = "Error Independence")
> acf(nsc.lm$residuals,
+     main = "Negative Serial Correlation")
```

- Normality can be checked by plotting the estimated Vs assumed distribution.

4. The errors follow the normal distribution of $\mathrm{N}\left(0, \sigma^2\right)$.




```
> tmp = seq(-100, 100, length.out = 100)
> res = residuals(lmlist[[1]])
> # res = residuals(msl.lm)
>
> sigma = sqrt(sum(res^2) / (n-2))
> # Density function
> hist(res, probability = TRUE,
+      xlim = c(-100, 100), ylim = c(0, 0.03),
+      xlab = "Residual",
+      main = "Distribution of Residual")
>
> lines(density(res), col = "blue")
>
> lines(tmp, dnorm(tmp, sd = sigma),
+       col ="red", lty = 2)
>
```

```
> legend("topright", legend = c("Est.", "True"),
+        lty = c(1,2), col = c(4, 2))


> # Distribution function
> sample_quantile = sort(res); sample_cdf = (1:n)/n
>
> tmp = seq(min(sample_quantile),
+           max(sample_quantile), length.out = 100)
>
> plot(sample_quantile, sample_cdf,
+      xlab = "Residual", ylab = "CDF",
+      main = "Estimated Cumulative distribution",
+      type = "s")
>
> lines(tmp, pnorm(tmp, sd = sigma),
+       col = 2, lty = 2)
>
> legend("topleft", legend = c("Est.", "True"),
+        lty = c(1, 2), col = c(1, 2))


> # Quantile-Quantile Normal plot
> qqnorm(res)
> qqline(res, col = "red")
```

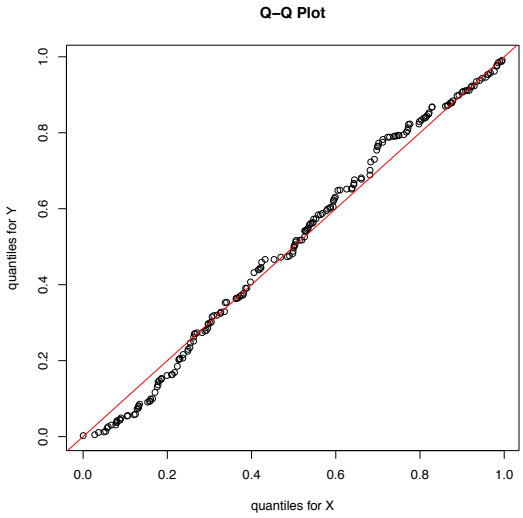
- In general, we can compare the quantiles of two arbitrary distributions

```
> num = 200 # number of observation


> # Compare samples from the same distribution
> x = runif(num, min = 0, max = 1)
> y = runif(num, min = 0, max = 1)


> qqplot(x, y, main = "Q-Q Plot",
+        xlab = "quantiles for x",
+        ylab = "quantiles for y")
>
> abline(a = 0, b = 1, col = 2)
```

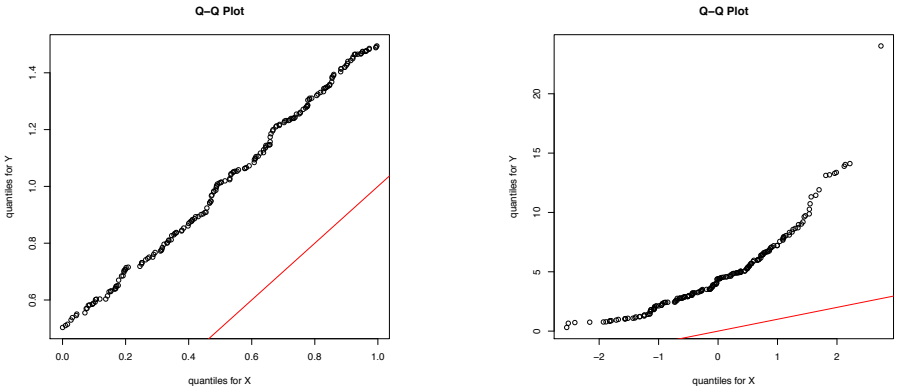
Q: What do you expect to see if X and Y follow the same distribution?



Q: What do you expect to see in the Q-Q plot if we have the following instead?

```
> # The same shape, but different center
> x = runif(num, min = 0, max = 1)
> y = runif(num, min = 0.5, max = 1.5)
> qqplot(x, y, main = "Q-Q Plot",
+        xlab = "quantiles for X",
+        ylab = "quantiles for Y")
> abline(a = 0, b = 1, col = 2)

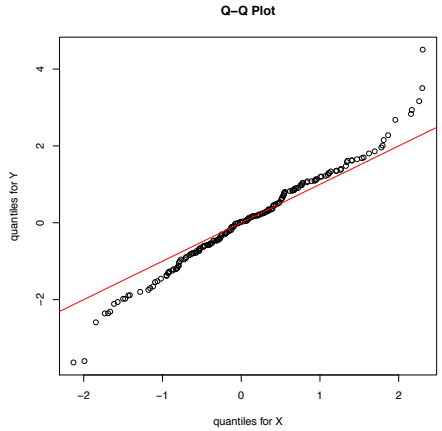
> # One is Skewed to the right, one symmetric
> x = rnorm(num)
> y = rchisq(num, df = 5)
> qqplot(x, y, main = "Q-Q Plot",
+        xlab = "quantiles for X",
+        ylab = "quantiles for Y")
> abline(a = 0, b = 1, col = 2)
```

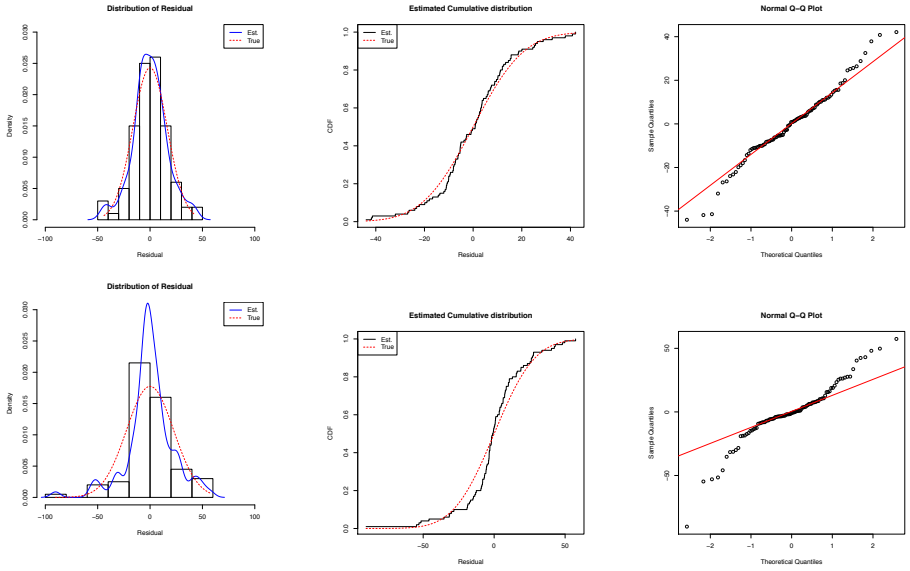


Q: How about the following?

```
> # One has longer tail than the other
> x = rnorm(num)
> y = rt(num, df = 5)


> qqplot(x, y, main = "Q-Q Plot",
+         xlab = "quantiles for X",
+         ylab = "quantiles for Y")
> abline(a = 0, b = 1, col = 2)
```





2.5 Transformation

Q: How should we proceed if one of assumptions is violated?

1. The conditional mean is linear for all i, i.e.

$$\mathbb{E}\left[Y_i \mid X_i = x_i\right] = \beta_0 + \beta_1 x_i$$

2. The error term

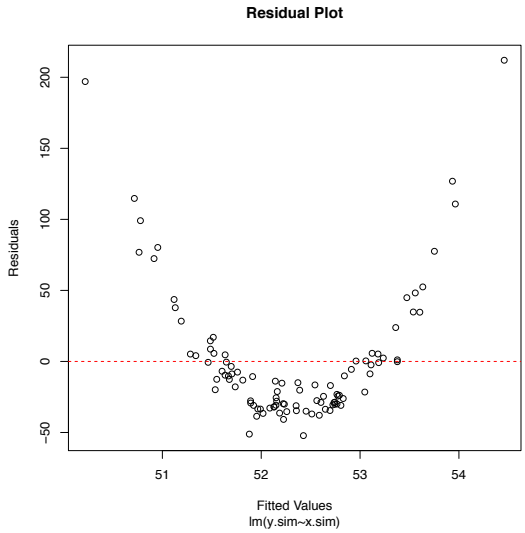
$$\varepsilon_i = Y_i - \beta_0 - \beta_1 X_i$$

has a zero mean and a constant but unknown variance for all i, i.e.

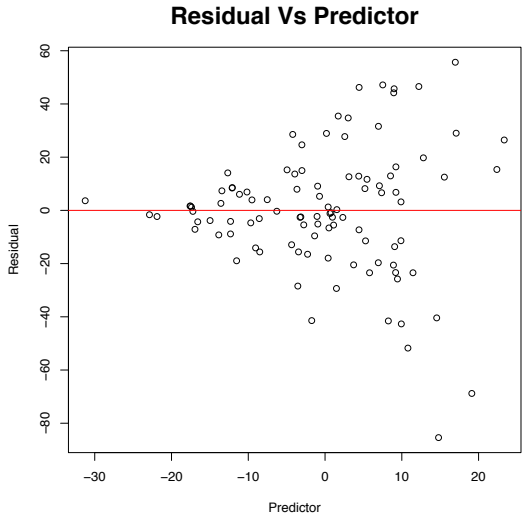
$$\mathbb{E}\left[\varepsilon_i \mid X_i\right] = 0 \qquad \text{and} \qquad \operatorname{Var}\left[\varepsilon_i \mid X_i\right] = \sigma^2$$

3. The error terms ε_i are independent of X_i, and of each other, for all i.
4. The error terms ε_i follow a normal distribution.

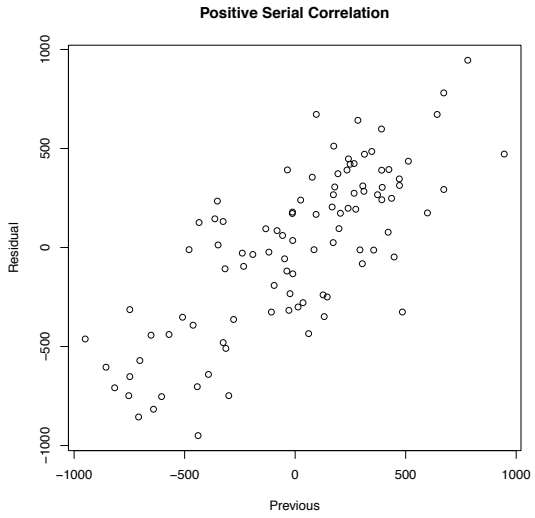
- Nonlinearity, e.g.



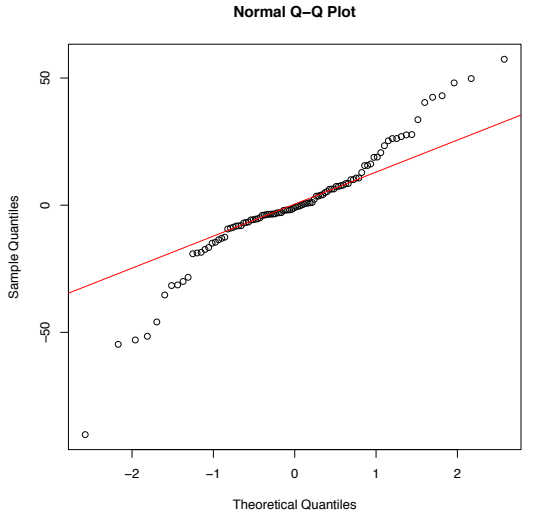
- Heteroskedasticity, e.g.



- Autocorrelation, e.g.



- Nonnormality , e.g.



- There are three approaches:

1. Transformation on the variables
2. Switch to advanced Models
3. Do nothing!

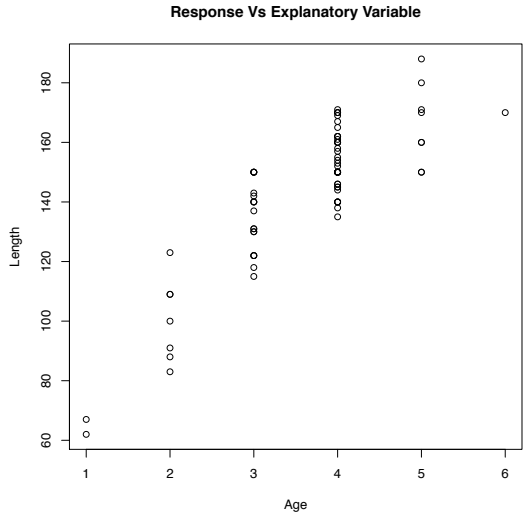
- Transformation is often used when assumption 1., 2. or 4. are in doubt.
- Consider the following dataset,

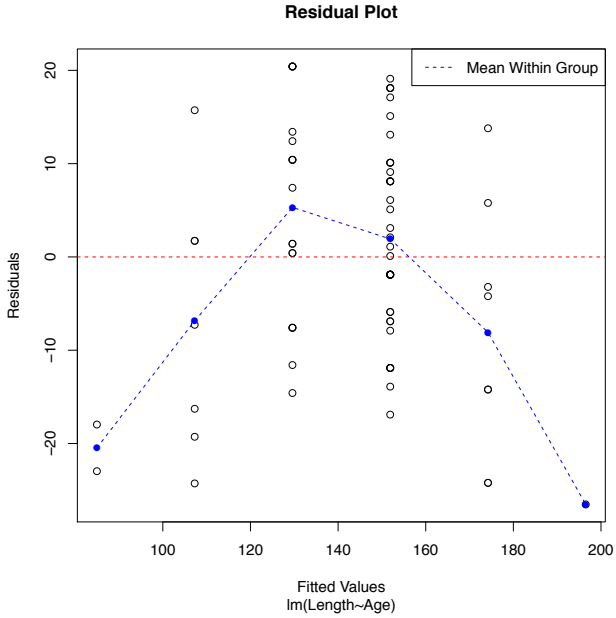
`Age`	Integer age of the fish
`Length`	Integer length of the fish in mm

which is about 78 bluegills that were captured from Lake Mary, Minnesota.

```
> # Load data locally
> bluegill.df = read.table("~/Desktop/bluegill.txt",
+                          header = TRUE)


> plot(Length~Age, data = bluegill.df)
> title(main = "Response Vs Explanatory Variable")
```






```
> model = bluegill.LM # For reusing the code
> # Basic residual Plot
> plot(fitted.values(model), residuals(model),
+      xlab = "Fitted Values", ylab = "Residuals",
+      main = "Residual Plot",
+      sub = "lm(Length~Age)")
> abline(a = 0, b = 0, lty = 2, col = "red")
> # Computing mean residual within group
> FV.group = factor(bluegill.df$Age)
> res.mean = tapply(residuals(model),FV.group, mean)
> age.unique = as.numeric(levels(FV.group))
> tmp = data.frame(Age = age.unique)
> x.tmp = predict(model, tmp)
> y.tmp = as.numeric(res.mean)
> # Adding the means, and a legend
> points(x.tmp, y.tmp, col = "blue", pch = 16)
> lines(x.tmp, y.tmp, col = "blue", lty = 2)
> legend("topright", lty = 2, col = 4,
+        legend = c("Mean Within Group"))
```

- The curvature suggests to us that the conditional mean might be

$$\mathbb{E}\left[Y_i \mid X_i = x_i\right] = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$$

```
> bluegill.quad.LM = lm(Length~Age+I(Age^2),
+                       data = bluegill.df)
```

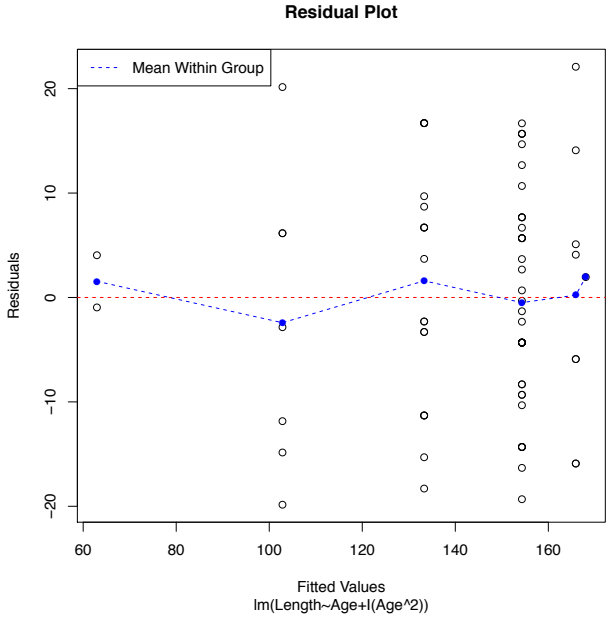
```
> model = bluegill.quad.LM # for reusing the code
> summary(model)
```

```
Call:
lm(formula = Length ~ Age + I(Age^2), data = bluegill.df)

Residuals:
    Min      1Q  Median      3Q     Max
-19.846  -8.321  -1.137   6.698  22.098

Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept)   13.622     11.016   1.237     0.22
Age           54.049      6.489   8.330 2.81e-12 ***
I(Age^2)      -4.719      0.944  -4.999 3.67e-06 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 10.91 on 75 degrees of freedom
Multiple R-squared:  0.8011,    Adjusted R-squared:  0.7958
F-statistic: 151.1 on 2 and 75 DF,  p-value: < 2.2e-16
```



- Consider the following dataset,

`year`	Year of manufacture
`price`	Price in Australian dollars

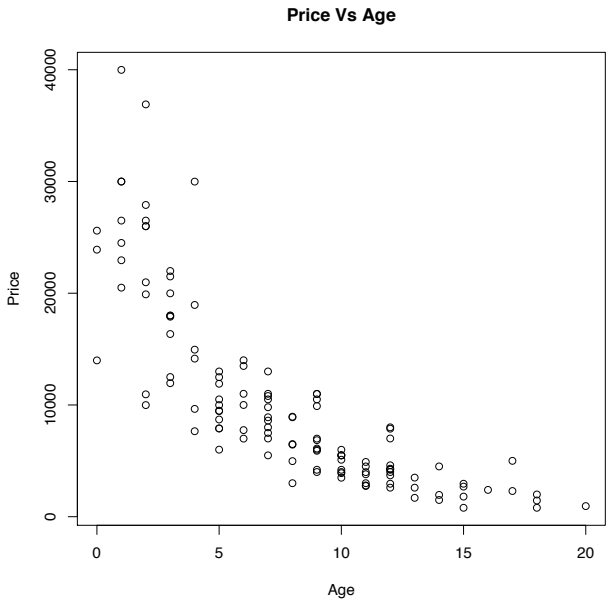
which is about 123 Mazda cars taken from a Melbourne newspaper in 1991.

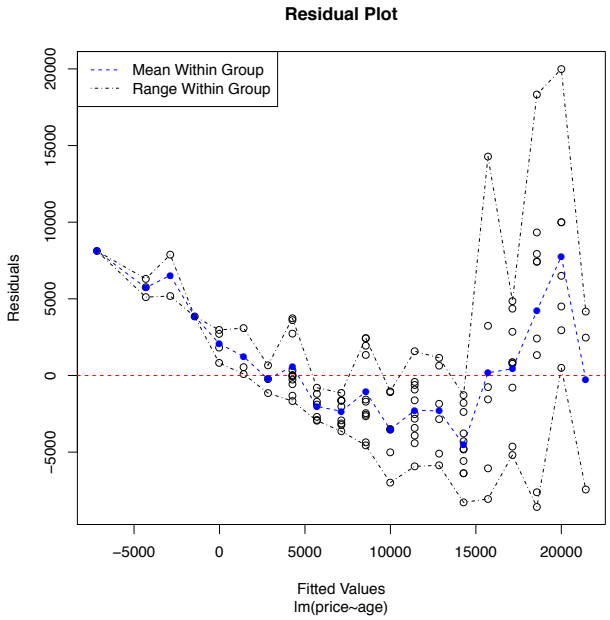
```
> oc.df = read.table("~/Desktop/old_car.txt",
+                    header = TRUE)
> str(oc.df)
```

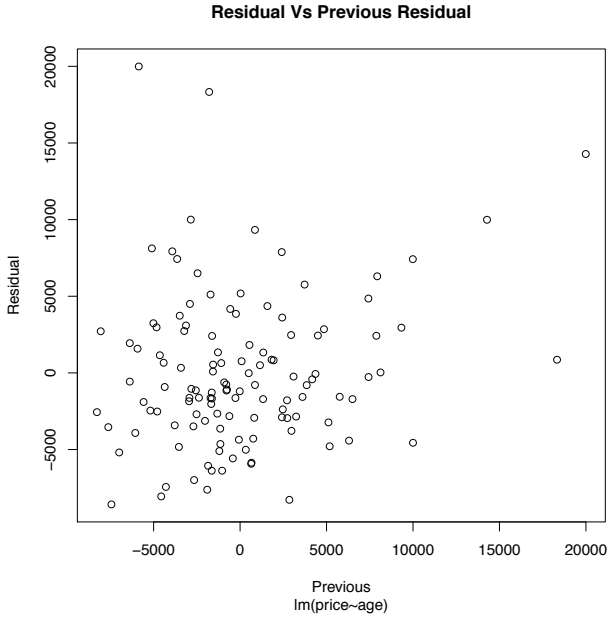
```
'data.frame':    123 obs. of  2 variables:
 $ year : int  79 82 83 ...
```

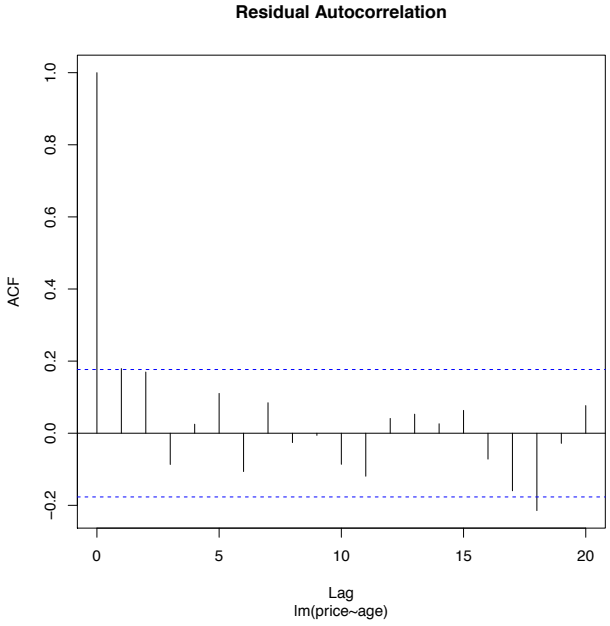
```
 $ price: int  2950 5900 2999 ...
```

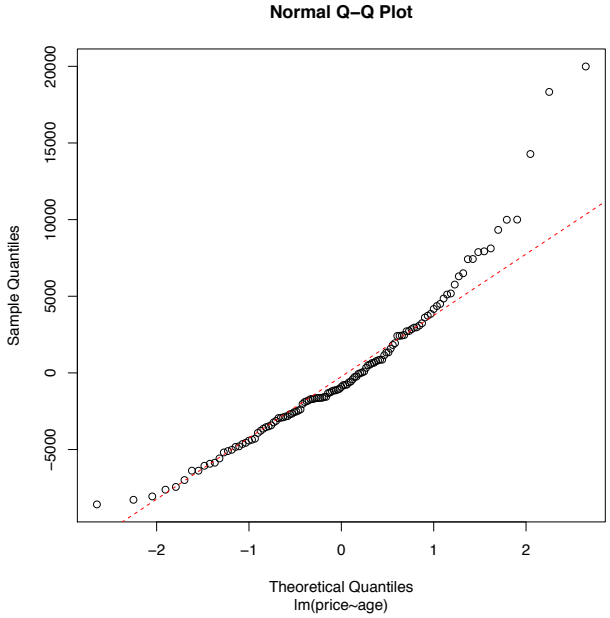
```
> # Create a new variable
> age = 91 - oc.df$year
```






Residual Vs Previous Residual
Residual
Previous
lm(price~age)
−5000
0
5000
10000
15000
20000


Residual Autocorrelation
ACF
Lag
lm(price~age)
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20




```
> ## Construct a preliminary model
> oc.LM = lm(price~age, data = oc.df)
> model = oc.LM
>
> ## Basic Residual Plot
> plot(fitted.values(model), residuals(model),
+      xlab = "Fitted Values", ylab = "Residuals",
+      main = "Residual Plot",
+      sub = "lm(price~age)")
> abline(a = 0, b = 0, lty = 2, col = "red")
>
> # Create groups according to fitted values
> FV.group = factor(age)
> age.unique = as.numeric(levels(FV.group))
>
> # Estimated Conditional Mean
> res.mean = tapply(residuals(model),FV.group, mean)


> # Adding Points and Lines
> tmp = data.frame(age = age.unique)
> x.tmp = predict(model, tmp)
> y.tmp = as.numeric(res.mean)
>
> points(x.tmp, y.tmp, col = "blue", pch = 16)
> lines(x.tmp, y.tmp, col = "blue", lty = 2)
>
```

```
> # Max and Min values
> res.max = tapply(residuals(model),FV.group, max)
> res.min = tapply(residuals(model),FV.group, min)
>
> # Adding Lines
> y.tmp = as.numeric(res.max)
> lines(x.tmp, y.tmp, lty = 4)
>
> y.tmp = as.numeric(res.min)
> lines(x.tmp, y.tmp, lty = 4)


> ## Residual Vs Previous Residual
> res = residuals(model)
> n = length(res)
>
> plot(res[-n], res[-1],
+      xlab = "Previous", ylab = "Residual",
+      main = "Residual Vs Previous Residual",
+      sub = "lm(price~age)")
>
> ## ACF Plot
> acf(res, main = "Residual Autocorrelation",
+     sub = "lm(price~age)")
>
> ## QQ Normal Plot
> qqnorm(res, sub = "lm(price~age)")
> qqline(res, col = 2, lty = 2)
```

Q: Why is using p-value to judge whether polynomial is needed a bad idea?

```
> oc.quad.LM = lm(price~age+I(age^2), data = oc.df)
> model = oc.quad.LM
> summary(model)
```

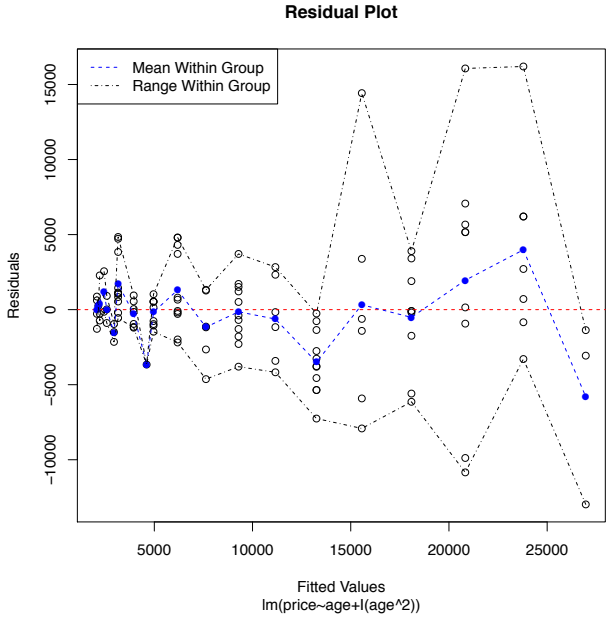
```
Call:
lm(formula = price ~ age + I(age^2), data = oc.df)

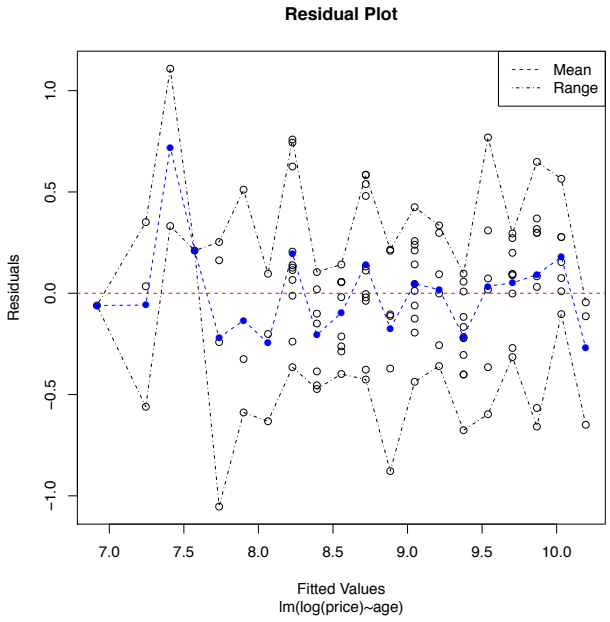
Residuals:
     Min       1Q   Median       3Q      Max
-12974.9  -1442.4   -187.3   1241.6  16200.0

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) 26964.90    1066.35  25.287  < 2e-16 ***
age         -3283.16     271.81 -12.079  < 2e-16 ***
I(age^2)      108.27      15.16   7.142 7.68e-11 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 4164 on 120 degrees of freedom
Multiple R-squared:  0.7514,    Adjusted R-squared:  0.7473
F-statistic: 181.4 on 2 and 120 DF,  p-value: < 2.2e-16
```

Q: What should we look at instead?

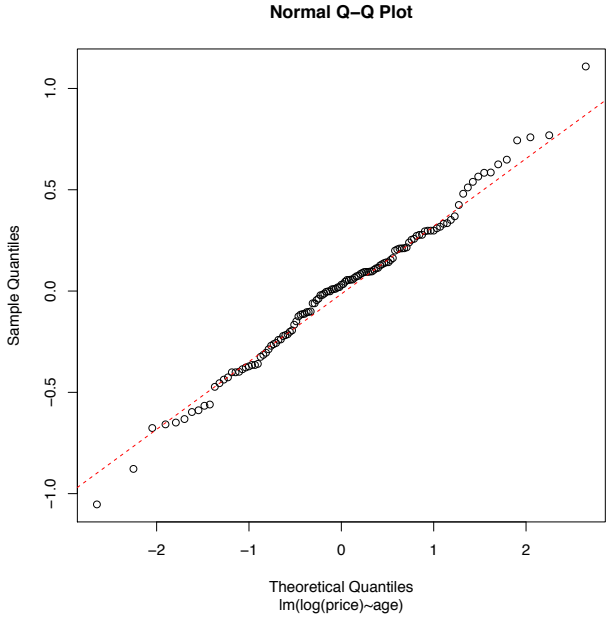


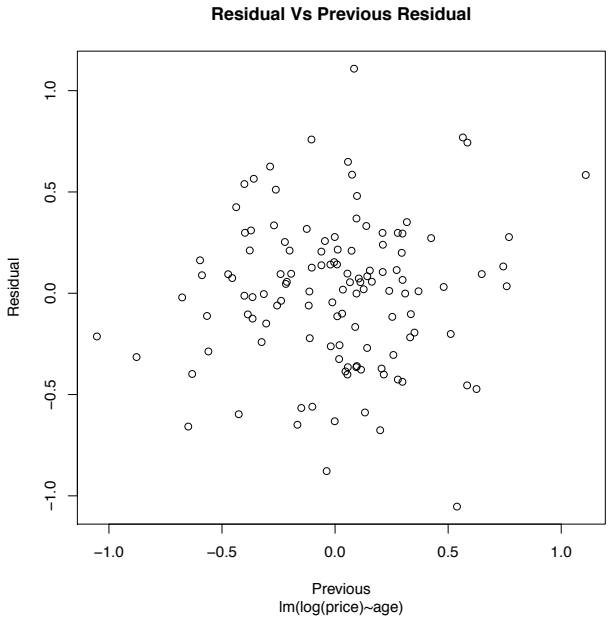


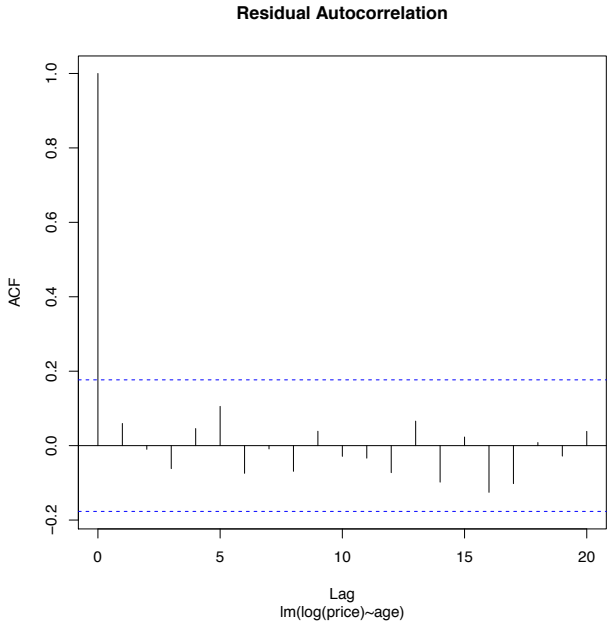

```
> oc.log.LM = lm(log(price)~age, data = oc.df)
> model = oc.log.LM # For reusing the code
>
> plot(fitted.values(model), residuals(model),
```

```
+      xlab = "Fitted Values", ylab = "Residuals",
+      main = "Residual Plot",
+      sub = "lm(log(price)~age)")
> abline(a = 0, b = 0, lty = 2, col = "red")
>
> # Computing mean residual within group
> FV.group = factor(age)
> res.mean = tapply(residuals(model), FV.group, mean)
> age.unique = as.numeric(levels(FV.group))
> tmp = data.frame(age = age.unique)
> x.tmp = predict(model, tmp)
> y.tmp = as.numeric(res.mean)
>
> # Adding the means, and a legend
> points(x.tmp, y.tmp, col = "blue", pch = 16)


> x.sort = sort(x.tmp) # To have not backward lines
> index = as.integer(names(sort(x.tmp)))
> lines(x.sort, y.tmp[index], col = "blue", lty = 2)
>
> # Computing residual range within group
> res.max = tapply(residuals(model),FV.group, max)
> res.min = tapply(residuals(model),FV.group, min)
>
> y.tmp = as.numeric(res.max)
> lines(x.sort, y.tmp[index], lty = 4)
>
>
> y.tmp = as.numeric(res.min)
> lines(x.sort, y.tmp[index], lty = 4)
>
>
> legend("topright", lty = c(2,4), col = c(4,1),
+        legend = c("Mean", "Range"))
```


Normal Q–Q Plot
Sample Quantiles
Theoretical Quantiles
lm(log(price)~age)
−2
−1
0
1
2
−1.0
−0.5
0.0
0.5
1.0


Residual Vs Previous Residual
Residual
Previous
lm(log(price)~age)
−1.0
−0.5
0.0
0.5
1.0



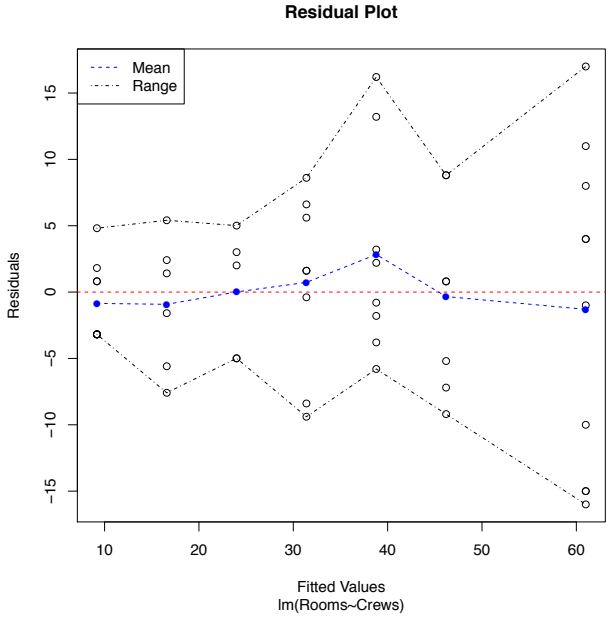
- In this case, it seems the natural logarithmic transformation not only solves the nonlinearity issue but solves the heteroskedasticity/nonnormality issue
- In general, when the variance is greater for larger fitted values than it is for smaller fitted values, a natural logarithmic transformation of the response variable often solves the heteroskedasticity issue.
- However, we cannot possibly expect log-transform or polynomial terms to always be the remedy, we need a more general and systematic method.
- To illustrate what can go wrong, consider the following dataset,

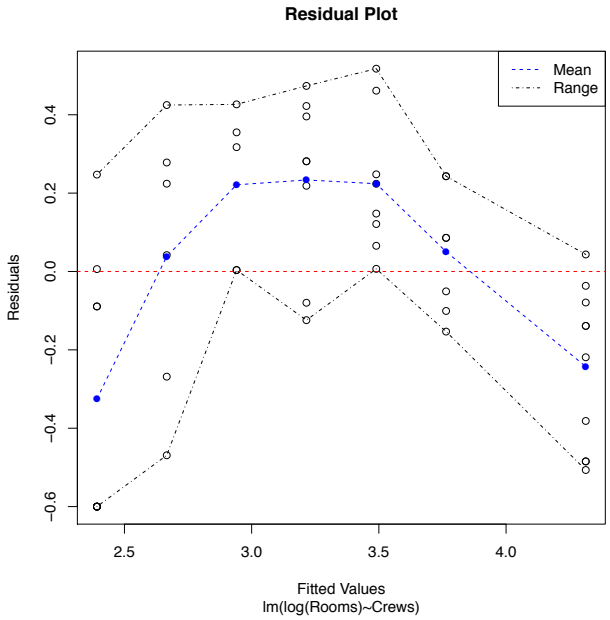
  `Crews` Number of works on the job

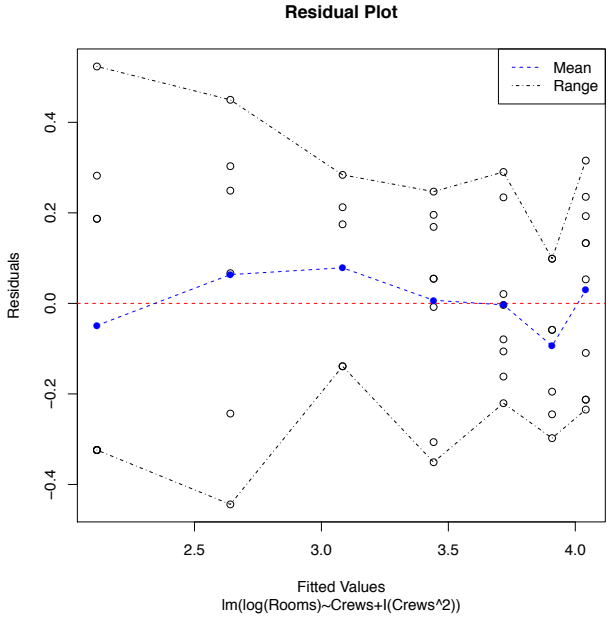
  `Rooms` Number of offices need to be cleaned

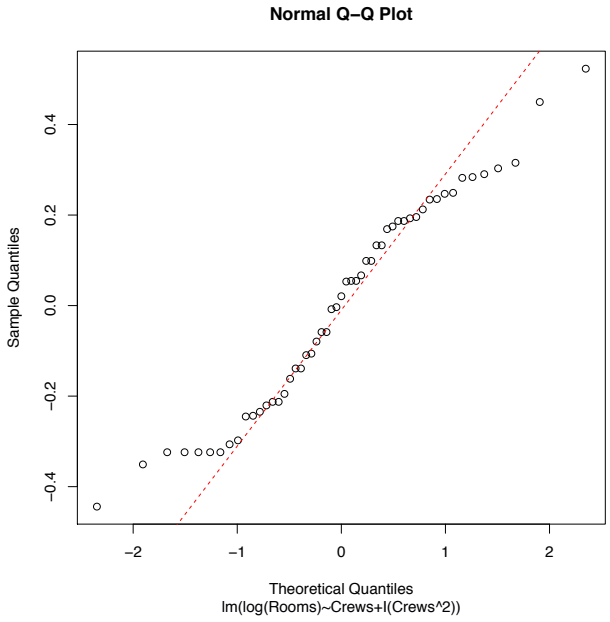
  which is about 53 cases that a maintenance company did in the past.

```
> cleaning.df = read.table("~/Desktop/cleaning.txt",
+                          header = TRUE)
>
> cleaning.LM = lm(Rooms~Crews, data = cleaning.df)
```


Residual Plot
Mean
Range
Residuals
15
10
5
0
−5
−10
−15
10
20
30
40
50
60
Fitted Values
lm(Rooms~Crews)


Residual Plot
Mean
Range
Residuals
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.5
3.0
3.5
4.0
Fitted Values
lm(log(Rooms)~Crews)





1. The conditional mean is linear for all i, i.e.

$$\mathbb{E}\left[Y_i \mid X_i = x_i\right] = \beta_0 + \beta_1 x_i$$

2. The error has a zero mean and a constant but unknown variance for all i, i.e.

$$\mathbb{E}\left[\varepsilon_i \mid X_i\right] = 0 \qquad \text{and} \qquad \text{Var}\left[\varepsilon_i \mid X_i\right] = \sigma^2$$

4. The error terms ε_i follow a normal distribution.

- Recall the response is given by

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

- Therefore, a compact way to state the assumptions is

$$Y_i \mid X_i \sim \text{Normal}\left(\beta_0 + \beta_1 x_i, \sigma^2\right)$$

for all i.

- When we detect a possible violation of those three assumptions,

$$Y_i \mid X_i \sim \text{Normal}\left(\beta_0 + \beta_1 x_i, \sigma^2\right)$$

it means we have evidence that $Y \mid X$ does not follow a normal distribution.

- The idea is to assume the response can be transformed by

$$Y_i^*(\lambda) = \begin{cases} \dfrac{Y_i^\lambda - 1}{\lambda}, & \text{if} \quad \lambda \neq 0; \\ \ln Y_i, & \text{if} \quad \lambda = 0. \end{cases}$$

so that the new response is conditionally normal, that is

$$Y_i^*(\lambda) \mid X_i \sim \text{Normal}\left(\beta_0 + \beta_1 x_i, \sigma^2\right)$$

Q: Can you see the transformation when $\lambda = 0$ is the limiting case of $\lambda \neq 0$?

Q: Do you notice it implicitly assumes Y is positive?

- For responses that might be negative, the following was proposed

$$Y_{i(\lambda_1,\lambda_2)}^* = \begin{cases} \dfrac{(Y_i + \lambda_2)^{\lambda_1} - 1}{\lambda_1}, & \text{if} \quad \lambda \neq 0; \\ \ln\left(Y_i + \lambda_2\right), & \text{if} \quad \lambda = 0. \end{cases}$$

the idea is essentially the same, so we will consider the positive case here.

Q: How can we determine which λ value to use?

- Since $Y_i^* \mid X_i$ is assumed to be normal,

$$f_{Y_i^* \mid X_i}(y_i^* \mid x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(y_i^* - \beta_0 - \beta_1 x_i\right)^2}{2\sigma^2}\right)$$

with the independence assumption, the joint density is readily available, and

$$\mathcal{L}\left(\beta_0, \beta_1, \sigma^2\right) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i^* - \beta_0 - \beta_1 x_i\right)^2\right)$$

Q: Do you remember the change of variable technique?

$$\mathcal{L}\left(\lambda, \beta_0, \beta_1, \sigma^2\right) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}\left(y_i^*(\lambda) - \beta_0 - \beta_1 x_i\right)^2}{2\sigma^2}\right) |J(\mathbf{y};\lambda)|$$

where $J(\mathbf{y};\lambda)$ is the Jacobian of the transformation in terms of $\mathbf{y}$.

Q: What is the Jacobian in this case?

- Notice the likelihood function with λ is proportional to the one without λ.

$$\mathcal{L}\left(\beta_0, \beta_1, \sigma^2\right) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i^* - \beta_0 - \beta_1 x_i\right)^2\right)$$

- So the maximum likelihood estimates of β_0, β_1 and σ^2 takes the usual form.
- Thus, for a given λ value, we can compute

$$b_0^* = \bar{y}^* - b_1^*\bar{x} \qquad \text{where} \quad \beta_1^* = \frac{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)\left(y_i^* - \bar{y}^*\right)}{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}$$

and

$$\sigma_u^2 = \frac{n}{n-2}\sigma_{MLE}^2 = \frac{1}{n-2}\sum_{i=1}^{n}\left(y_i^* - \hat{y}_i^*\right)^2$$

where

$$\hat{y}_i^* = b_0^* + b_1^* x_i$$

Q: How can we find λ?

- The negative log-likelihood function is given by

$$\ell(\lambda) = \frac{n}{2}\ln(2\pi) + \frac{n}{2}\log\sigma_{MLE}^2 + \frac{1}{2\sigma_{MLE}^2}\sum_{i=1}^{n}\left(y_i^* - \hat{y}_i^*\right)^2 - (\lambda - 1)\sum_{i=1}^{n}\ln y_i$$

- Notice terms in red also depend on λ, and no closed form exists for MLE of

$$\lambda$$

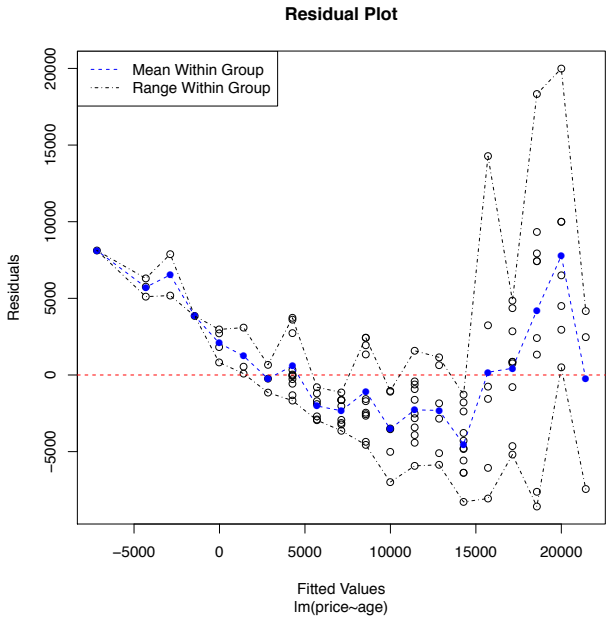
however, numerically we can determine λ reasonably well.

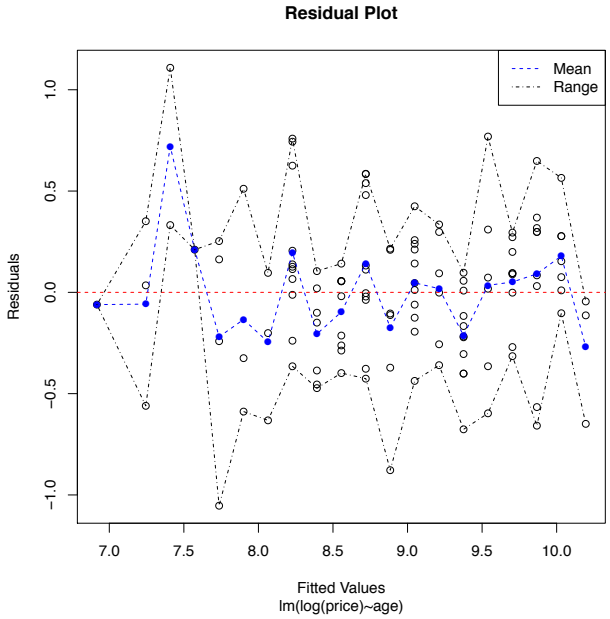
- Using the λ value that maximises the likelihood to transform Y is known as

Box-Cox transformation

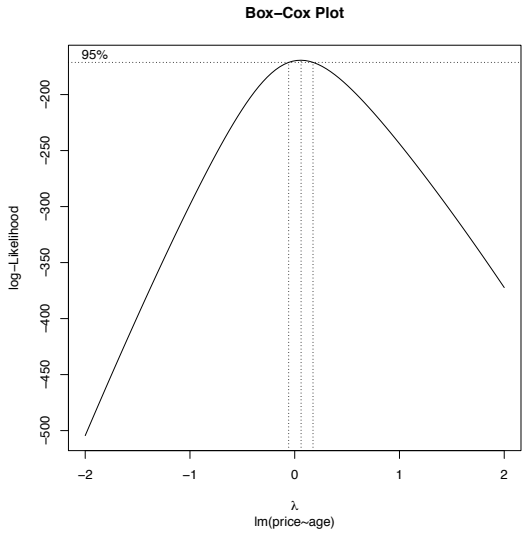
- Recall the following two models for the old Mazda cars dataset

```
> oc.LM = lm(price~age, data = oc.df)
> oc.log.LM = lm(log(price)~age, data = oc.df)
```





Q: What does the following Box-Cox plot suggest?



- Imagine cases in which both variables, X and Y, need to be transformed.

$$Y_i^* \mid X_i^* \sim \text{Normal}\left(\beta_0 + \beta_1 x_i^*, \sigma^2\right)$$

that is, there are surely cases that the Box-Cox procedure cannot handle.

Q: Given two normal random variables, X and Y, what is the distribution of

$$Y \mid X$$

- It can be shown the conditional distribution is always normal,

$$Y \mid X \sim \text{Normal}\left(\mu_Y + \rho_{XY}\frac{\sigma_Y}{\sigma_X}\left(x - \mu_X\right), \sigma_Y^2\left(1 - \rho_{XY}^2\right)\right)$$

Q: What does it mean in terms of simple linear regression and transformation?

- Combining this with the Box-Cox procedure, we have a systematic way to fix assumption 1, 2 and 4 by transforming both X and Y.

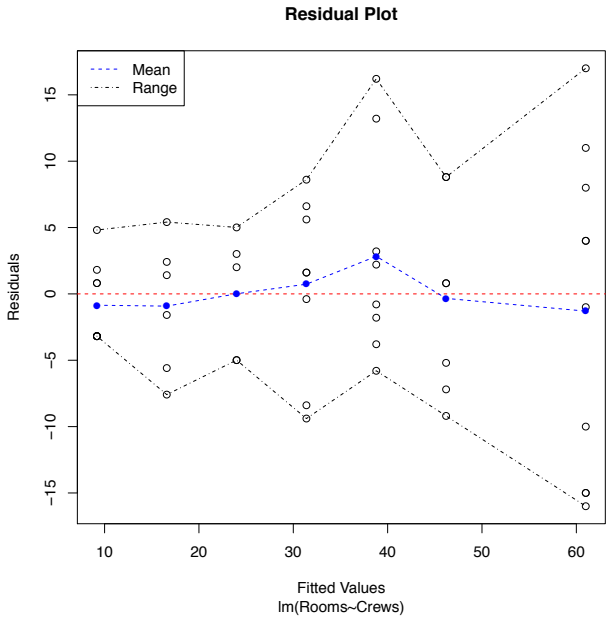
- Let us illustrate that by applying it to the following dataset

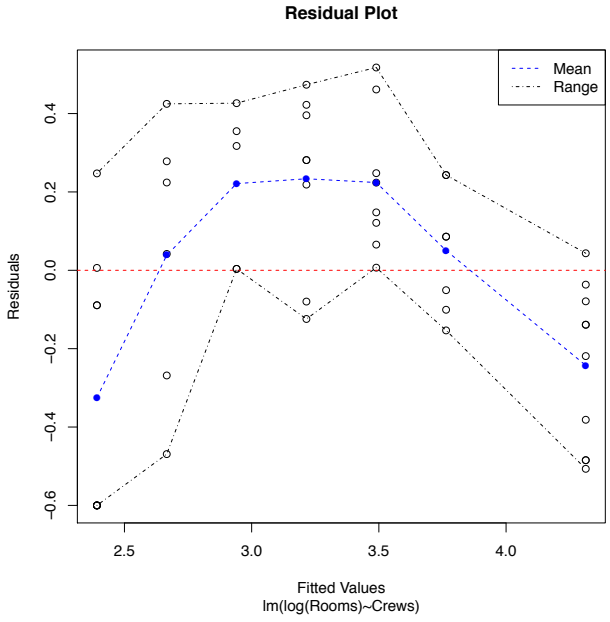
`Crews`	Number of works on the job
`Rooms`	Number of offices need to be cleaned

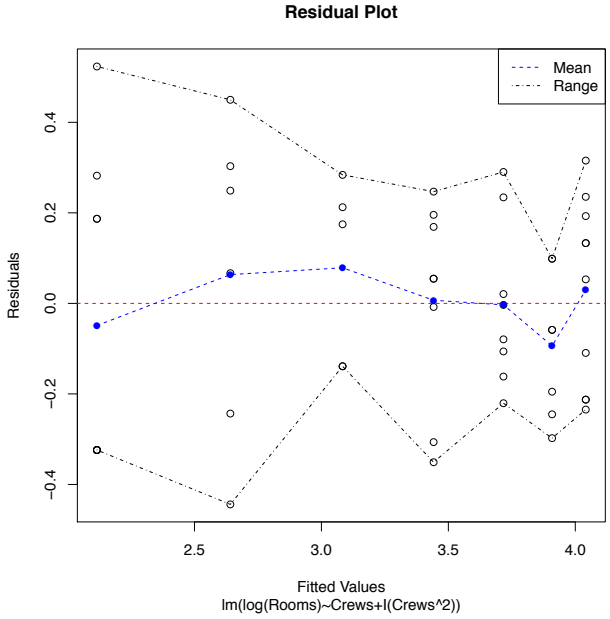
- Recall we considered the following three models for this dataset,

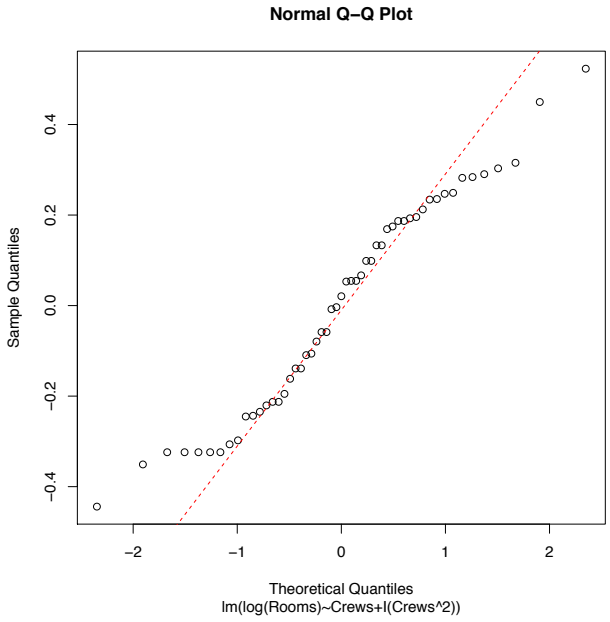
```
> cleaning.LM = lm(Rooms~Crews, data = cleaning.df)
>
> cleaning.log.LM =
+   lm(log(Rooms)~Crews, data = cleaning.df)
>
> cleaning.log.sq.LM =
+   lm(log(Rooms)~Crews+I(Crews^2),
+      data = cleaning.df)
```

none of which is satisfactory!









- The numerical optimisation and the Box-Cox plot were done by

```
> # Box-Cox procedure is provided by this library
> library(MASS)
> bc = boxcox(oc.LM)
```

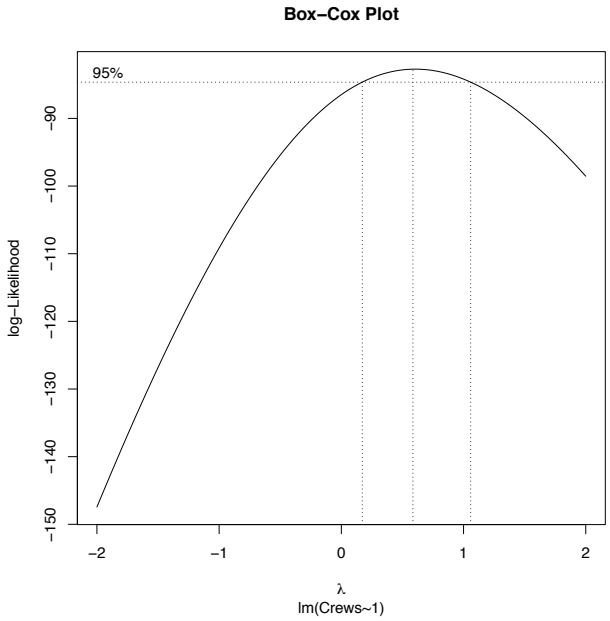
```
> title("Box-Cox Plot", sub = "lm(price~age)")
```

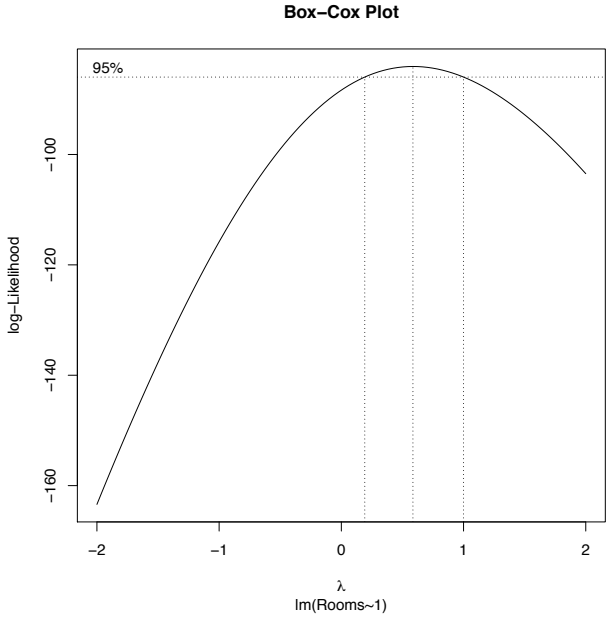
- For the cleaning dataset, we consider the best transformation for `Crews`

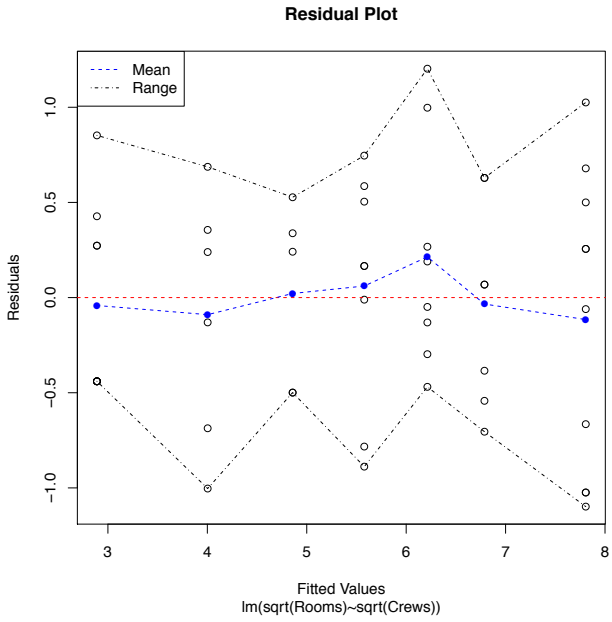
```
> x.LM = lm(Crews~1, data = cleaning.df)
> bc = boxcox(x.LM)
> title("Box-Cox Plot", sub = "lm(Crews~1)")
```

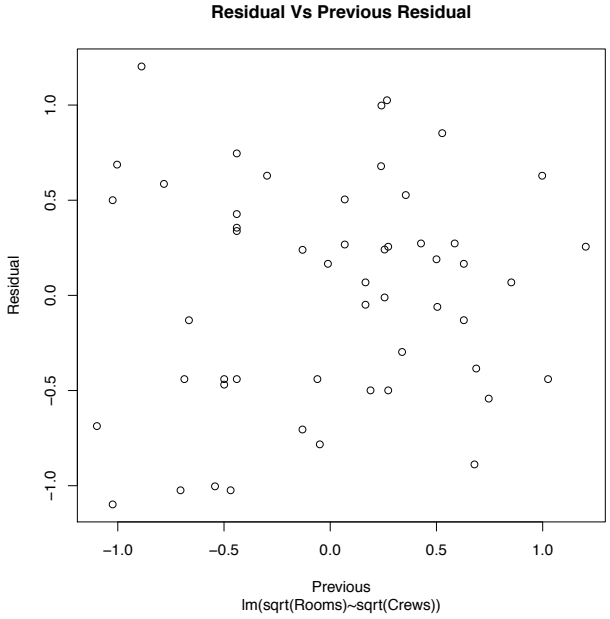
as well considering the best transformation for `Rooms`

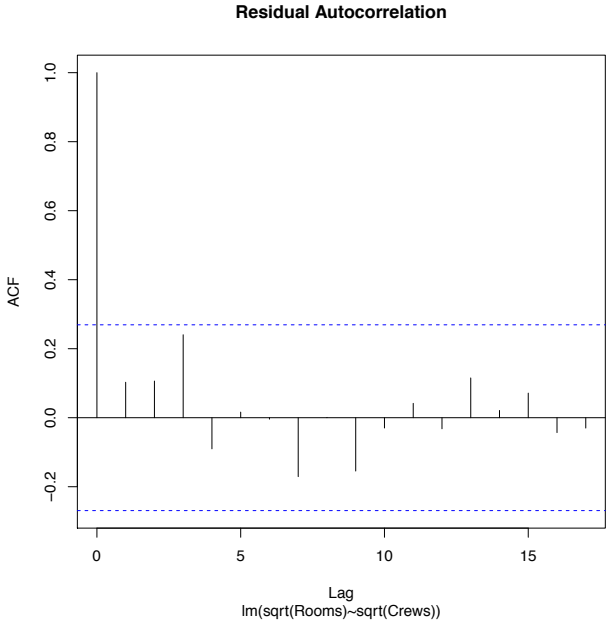
```
> y.LM = lm(Rooms~1, data = cleaning.df)
> bc = boxcox(y.LM)
> title("Box-Cox Plot", sub = "lm(Rooms~1)")
```

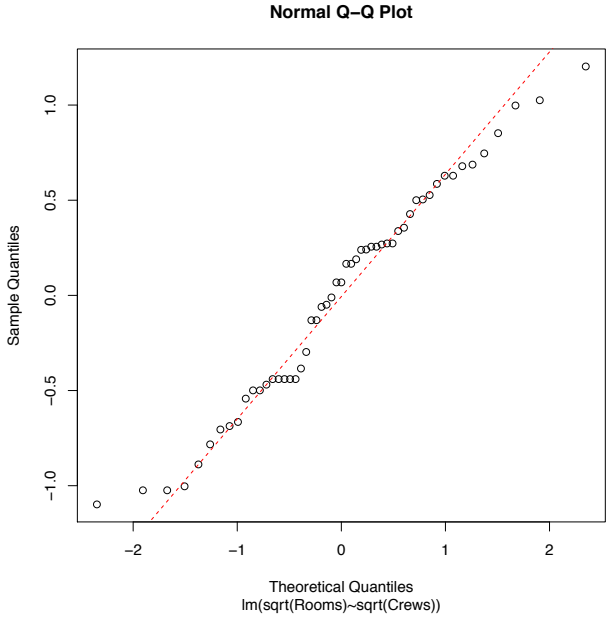



Box–Cox Plot
95%
log–Likelihood
−100
−120
−140
−160
−2
−1
0
1
2
λ
lm(Rooms~1)


Residual Plot
Mean
Range
Residuals
1.0
0.5
0.0
−0.5
−1.0
3
4
5
6
7
8
Fitted Values
lm(sqrt(Rooms)~sqrt(Crews))


Residual Vs Previous Residual
Residual
Previous
lm(sqrt(Rooms)~sqrt(Crews))


Residual Autocorrelation
ACF
Lag
lm(sqrt(Rooms)~sqrt(Crews))



2.6 Inference

A request from an insurance Company

- An insurance company wants to predict the damage (in $) to a home in a particular area if a fire occurs. The damage, and distance (in miles) from the fire station were recorded for 15 house fires in the area of interest.

```
> fire.df = read.table("~/Desktop/fire.txt",
+                      header = TRUE)

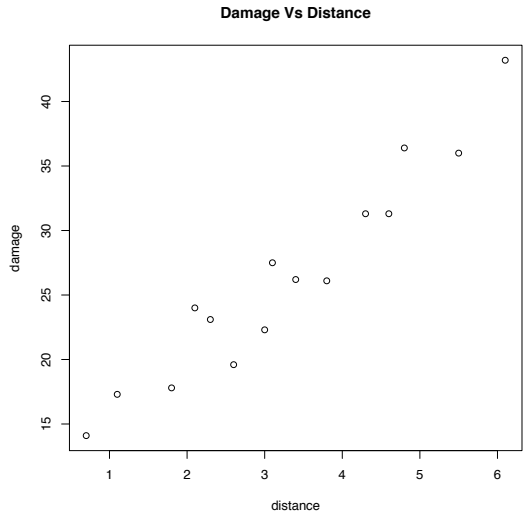
> str(fire.df)
```

```
'data.frame':   15 obs. of  2 variables:
 $ distance: num  3.4 2.6 1.8 4.3 4.6 ...
 $ damage  : num  26.2 19.6 17.8 31.3 31.3 ...
```

- We are required to predict the damage for house fires that are 1 and 4 miles from the fire station.

- Visualisation

```
> with(fire.df, plot(distance, damage,
+       main = "Damage Vs Distance"))
```



- Running a simple linear regression model

```
> fire.LM = lm(damage~distance, data = fire.df)
```

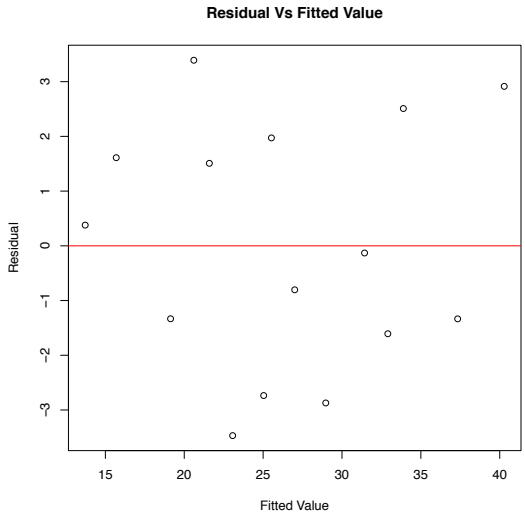
- Checking assumptions by looking at residuals

```
> fit = fire.LM$fitted.values
> res = fire.LM$residuals

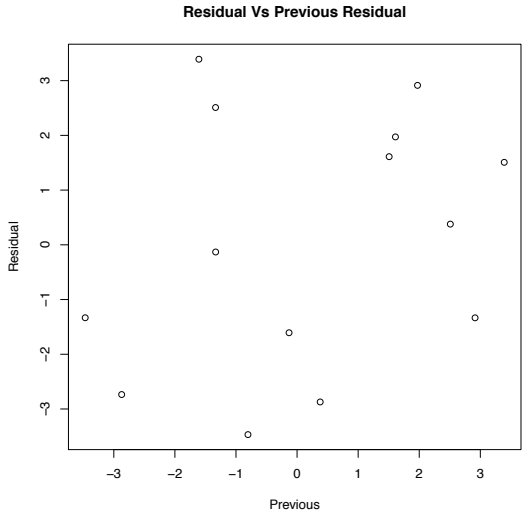
> # Residual Plot
> plot(fit, res, main = "Residual Vs Fitted Value",
+      xlab = "Fitted Value", ylab = "Residual")
> abline(h = 0, col = "red")

> # Correlation Plot
> plot(res[-nrow(fire.df)], res[-1],
+      xlab = "Previous", ylab = "Residual",
+      main = "Residual Vs Previous Residual")

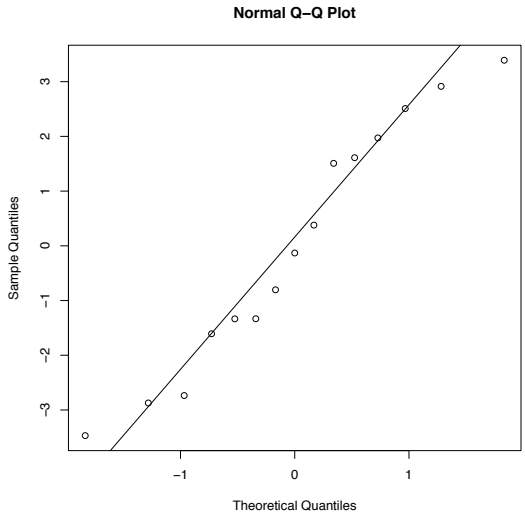
> # QQ plot
> qqnorm(res); qqline(res)
```



- There is no indication of nonlinearity, correlation or non-constant variance.



- There is no indication of autocorrelation.



- There is no indication of non-normality.
- Since the data size n is really small, we have to be careful with normality

```
> # Shapiro-Wilk
> shapiro.test(res)
```

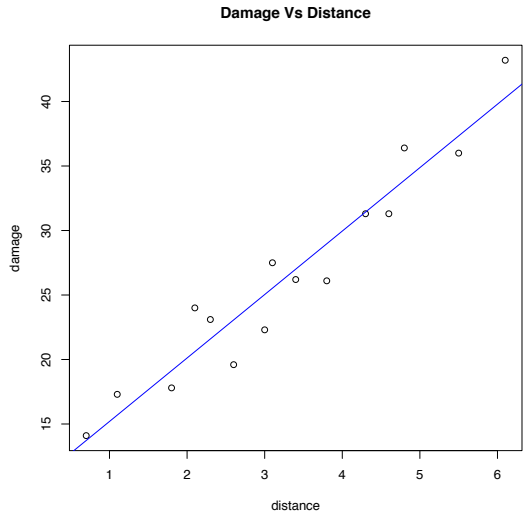
```
        Shapiro-Wilk normality test

data:  res
W = 0.94671, p-value = 0.4743
```

- Notice formal tests on normality are rather strict, use them if n is small.
- Since all the assumptions seem to be satisfied, we can now do inference.

```
> with(fire.df, plot(distance, damage,
+                    main = "Damage Vs Distance"))
> abline(fire.LM, col = "blue")
```

- So not only we can trust the line, but we can do other meaningful things...



- In general, the distribution of an estimator $\hat{\theta}$ is known as the sampling distribution of $\hat{\theta}$ to emphasis the fact that it is a random variable due to resampling, and the standard deviation of of the estimator is known as the standard error.
- Consider a sample of normal random variable X with mean μ and variance σ^2,

  $$\{x_1, x_2, \cdots x_n\}$$

  then it can be shown, where $\text{s. e.} = \sigma/\sqrt{n}$ is the standard error,

  $$\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim \text{Normal}(0, 1)$$

- Suppose we would like to construct a

  $$(1 - \alpha)100\%$$

  confidence interval for μ, and z^* is the value such that

  $$\Pr[Z \leq z^*] = 1 - \alpha/2 \qquad \text{where} \quad Z \sim \text{Normal}(0, 1)$$

- Then, we have

$$\Pr\left[-z^* \leq \frac{\bar{X}-\mu}{\sigma/\sqrt{n}} \leq z^*\right] = 1-\alpha$$

$$\Pr\left[\bar{X} - z^*\sigma/\sqrt{n} \leq \mu \leq \bar{X} + z^*\sigma/\sqrt{n}\right] = 1-\alpha$$

from which the $(1-\alpha)100\%$ confidence interval for μ in this case should be

$$\left(\bar{X} - z^*\sigma/\sqrt{n}, \bar{X} + z^*\sigma/\sqrt{n}\right)$$

Of course, in practice, σ^2 is not known, and is estimated using the following

$$S_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2$$

which can be shown to be unbiased. When S_X is used instead of the true σ,

$$\frac{\bar{X}-\mu}{S_X/\sqrt{n}} \sim \mathrm{t}_{n-1}$$

In a similar fashion,

$$\Pr\left[-t^* \leq \frac{\bar{X}-\mu}{S_x/\sqrt{n}} \leq t^*\right] \implies \Pr\left[\bar{X} - t^*S_X/\sqrt{n} \leq \mu \leq \bar{X} + t^*S_X/\sqrt{n}\right]$$

$$\implies \left(\bar{X} - t^*S_X/\sqrt{n}, \bar{X} + t^*S_X/\sqrt{n}\right)$$

- Now back to simple linear regression, let us denote

$$s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

you should have shown

$$\hat{\beta}_1 \sim \text{Normal}\left(\beta_1, \frac{\sigma^2}{(n-1)s_x^2}\right) \implies \frac{\hat{\beta}_1 - \beta_1}{\sigma/\sqrt{(n-1)s_x^2}} \sim \text{Normal}\,(0,1)$$

when σ^2 is not known, then it is usually estimated by

$$\sigma_u^2 = \frac{1}{n-2}\sum_{i=1}^{n}\hat{e}_i^2$$

and it can be shown that

$$\frac{\hat{\beta}_1 - \beta_1}{\sigma_u/\sqrt{(n-1)s_x^2}} \sim \mathrm{t}_{n-2}$$

Unless otherwise specified, i.e. σ^2 is known, the standard error of β_1 referrs to

$$\text{se}\left[\hat{\beta}_1\right] = \frac{\sigma_u}{\sqrt{(n-1)s_x^2}}$$

Similarly, we have

$$\frac{\hat{\beta}_0 - \beta_0}{\sigma\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{(n-1)s_x^2}}} \sim \text{Normal}\,(0,1)$$

and

$$\frac{\hat{\beta}_0 - \beta_0}{\text{se}\left[\hat{\beta}_0\right]} \sim \text{t}_{n-2}$$

where

$$\text{se}\left[\hat{\beta}_0\right] = \sigma_u\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{(n-1)s_x^2}}$$

```
> (fire.sm = summary(fire.LM))
```

```
Call:
lm(formula = damage ~ distance, data = fire.df)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4682 -1.4705 -0.1311  1.7915  3.3915

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)  10.2779     1.4203   7.237 6.59e-06 ***
distance      4.9193     0.3927  12.525 1.25e-08 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.316 on 13 degrees of freedom
Multiple R-squared:  0.9235,    Adjusted R-squared:  0.9176
F-statistic: 156.9 on 1 and 13 DF,  p-value: 1.248e-08
```

```
> names(fire.sm)
```

```
 [1] "call"          "terms"         "residuals"     "coefficients"  "aliased"
 [6] "sigma"         "df"            "r.squared"     "adj.r.squared" "fstatistic"
[11] "cov.unscaled"
```

Residual standard error is given by

$$\sigma_u = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}\hat{e}_i^2}$$

```
> (coeff.m = fire.sm$coefficients)
```

```
             Estimate Std. Error   t value     Pr(>t)
(Intercept) 10.277929  1.4202778  7.236562 6.585564e-06
distance     4.919331  0.3927477 12.525421 1.247800e-08
```

```
> class(coeff.m)
```

```
[1] "matrix"
```

- Confidence interval for the parameters, e.g. for $\hat{\beta}_1$

```
> b1_hat = coeff.mat[2,1]
>
> b1_se = coeff.mat[2,2]
>
> (b1_ci = b1_hat + c(-1, 1) * b1_se *
+   qt(0.975, fire.LM$df.residual))
```

```
[1] 4.070851 5.767811
```

- In practice, we use the following to obtain the confidence interval

```
> confint(fire.LM, "(Intercept)", level = 0.95)
```

```
                2.5 %   97.5 %
(Intercept) 7.209605 13.34625
```

```
> confint(fire.LM, "distance", level = 0.95)
```

```
            2.5 %   97.5 %
distance 4.070851 5.767811
```

- The following gives the predictions according to our model.

```
> pred.df = data.frame(distance = c(1,4))
> predict(fire.LM, pred.df)
```

```
       1        2
15.19726 29.95525
```

- Recall the optimal prediction of a random variable is the mean. Thus

$$\hat{y}^*_{n+1} \mid X_{n+1} = b_0 + b_1 x_{n+1} \qquad \text{where} \quad \begin{aligned} b_0 &= \bar{y} - b_1 \bar{x} \\ b_1 &= \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})} \end{aligned}$$

the star is here to distinguish it from the fitted value that uses $n+1$ observations.

2.7 Prediction

Q: How can we construct a "confidence interval" for our prediction?

$$(\hat{y}^*_{n+1} - c, \hat{y}^*_{n+1} + c)$$

Q: How can we find the constant c for a given significant level?

Q: Why R has two types of "confidence interval" for prediction?

```
> pred.df = data.frame(distance = c(1,4))
> predict(fire.LM, pred.df, interval = "confidence")
```

```
       fit       lwr       upr
1 15.19726 12.87092 17.52360
2 29.95525 28.52604 31.38446
```

```
> predict(fire.LM, pred.df, interval = "predict")
```

```
       fit       lwr       upr
1 15.19726  9.67879 20.71573
2 29.95525 24.75100 35.15951
```

- Now consider the random variable conditioning on $X_1, X_2, \cdots, X_{n+1}$,

$$\hat{Y}_{n+1} = \hat{\beta}_0 + \hat{\beta}_1 X_{n+1}$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ are the sources of randomness.

- Recall we have derived that

$$\hat{\beta}_1 = \beta_1 + \frac{\sum_{i=1}^{n} (X_i - \bar{X}_n)\, \varepsilon_i}{\sum_{i=1}^{n} (X_i - \bar{X}_n)^2}$$

and

$$\begin{aligned}
\hat{\beta}_0 &= \bar{Y}_n - \hat{\beta}_1 \bar{X}_n \\
&= \beta_0 + \beta_1 \bar{X}_n + \bar{\varepsilon}_n - \beta_1 \bar{X}_n - \frac{\bar{X}_n \sum_{i=1}^{n} (X_i - \bar{X}_n)\, \varepsilon_i}{\sum_{i=1}^{n} (X_i - \bar{X}_n)^2} \\
&= \beta_0 + \bar{\varepsilon}_n - \frac{\bar{X}_n \sum_{i=1}^{n} (X_i - \bar{X}_n)\, \varepsilon_i}{\sum_{i=1}^{n} (X_i - \bar{X}_n)^2}
\end{aligned}$$

- Therefore, the estimated conditional mean is given by

$$\hat{Y}_{n+1} = \beta_0 + \beta_1 X_{n+1} + \bar{\varepsilon}_n + \frac{\left(X_{n+1} - \bar{X}_n\right) \sum_{i=1}^{n} \left(X_i - \bar{X}_n\right) \varepsilon_i}{\sum_{i=1}^{n} \left(X_i - \bar{X}_n\right)^2}$$

  from which, we can conclude $\hat{Y}_{n+1}$ is a normal random variable with

$$\mathbb{E}\left[\hat{Y}_{n+1} \mid X_1, \ldots, X_n, X_{n+1}\right] = \beta_0 + \beta_1 X_{n+1}$$

- Recall the formula for the variance of linear combination of random variables

$$\operatorname{Var}\left[\sum_{i=1}^{n} a_i W_i\right] = \sum_{i=1}^{n} a_i^2 \operatorname{Var}\left[W_i\right] + 2 \sum_{1 \le i < j \le n} a_i a_j \operatorname{Cov}\left[W_i, W_j\right]$$

  Using which, we have

$$\begin{aligned}
&\operatorname{Var}\left[\hat{Y}_{n+1} \mid X_1, \ldots, X_n, X_{n+1}\right] \\
&= \sum_{i=1}^{n} \frac{1}{n^2}\sigma^2 + \frac{\left(X_{n+1} - \bar{X}_n\right)^2 \sum_{i=1}^{n} \left(X_i - \bar{X}_n\right)^2 \sigma^2}{\left(\sum_{i=1}^{n} \left(X_i - \bar{X}_n\right)^2\right)^2} \\
&\qquad + 2 \sum_{1 \le i < j \le n} \frac{1}{n} \frac{\left(X_{n+1} - \bar{X}_n\right)\left(X_j - \bar{X}_n\right)}{\sum_{\ell=1}^{n} \left(X_\ell - \bar{X}_n\right)^2} \operatorname{Cov}\left[\varepsilon_i, \varepsilon_j\right] \\
&= \frac{\sigma^2}{n} + \frac{\left(X_{n+1} - \bar{X}_n\right)^2 \sigma^2}{\sum_{i=1}^{n} \left(X_i - \bar{X}_n\right)^2} + \frac{2\sigma^2}{n} \frac{\sum_{i=1}^{n} \left(X_i - \bar{X}_n\right)}{\sum_{i=1}^{n} \left(X_i - \bar{X}_n\right)^2} \\
&= \sigma^2 \left(\frac{1}{n} + \frac{\left(X_{n+1} - \bar{X}_n\right)^2}{\sum_{i=1}^{n} \left(X_i - \bar{X}_n\right)^2} \right) + 0 \\
&= \frac{\sigma^2}{n-1} \left(\frac{n-1}{n} + \frac{\left(X_{n+1} - \bar{X}_n\right)^2}{\frac{1}{n-1}\sum_{i=1}^{n} \left(X_i - \bar{X}_n\right)^2} \right) \\
&= \frac{\sigma^2}{n-1} \left(\frac{n-1}{n} + \frac{\left(X_{n+1} - \bar{X}_n\right)^2}{S_X^2} \right)
\end{aligned}$$

- The variance of $\hat{Y}^*_{n+1}$ can be derived based on the variance of $\hat{\beta}_1$,

$$\text{Var}\left[\hat{Y}^*_{n+1} \mid X_1, X_2, \ldots, X_n, X_{n+1}\right] = \frac{\sigma^2}{n-1}\left(\frac{n-1}{n} + \frac{(x_{n+1} - \bar{x}_n)^2}{s_x^2}\right)$$

  and the sampling distribution of $\hat{Y}^*_{n+1}$ is normal given σ^2.

- Assuming our model is correct,

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

  the variance of Y_{n+1} conditional on X_{n+1} is simply

$$\text{Var}\left[Y_{n+1} \mid X_{n+1}\right] = \sigma^2$$

Q: What does the following variance capture?

$$\text{Var}\left[\hat{Y}^*_{n+1} + \varepsilon_{n+1} \mid X_1, X_2, \ldots, X_{n+1}\right] = \frac{\sigma^2}{n-1}\left(n - \frac{1}{n} + \frac{(x - \bar{x})^2}{s_x^2}\right)$$

Q: Can you distinguish various random processes in simple linear regression?

Q: Can you distinguish the two classes of variability in the response variable?

$$\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2 = \underbrace{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}_{\text{ESS}} + \underbrace{\sum_{i=1}^{n}\hat{e}_i^2}_{\text{RSS}}$$

- Subtract and add a term of $\hat{y}_i$

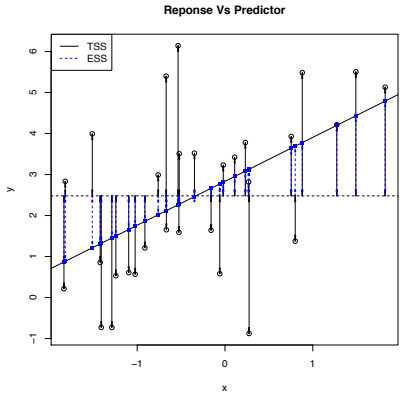
$$\begin{aligned}
\sum_{i=1}^{n}(y_i - \bar{y})^2 &= \sum_{i=1}^{n}(y_i - \hat{y}_i + \hat{y}_i - \bar{y})^2 \\
&= \sum_{i=1}^{n}\left((y_i - \hat{y}_i)^2 + 2(y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) + (\hat{y}_i - \bar{y})^2\right) \\
&= \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + 2\sum_{i=1}^{n}\hat{e}_i(b_0 + b_1 x_i - \bar{y}) + \sum_{i=1}^{n}\hat{e}_i^2 \\
&= \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}\hat{e}_i^2 \\
&\quad + 2\left(b_0\sum_{i=1}^{n}\hat{e}_i + b_1\sum_{i=1}^{n}\hat{e}_i x_i - \bar{y}\sum_{i=1}^{n}\hat{e}_i\right)
\end{aligned}$$

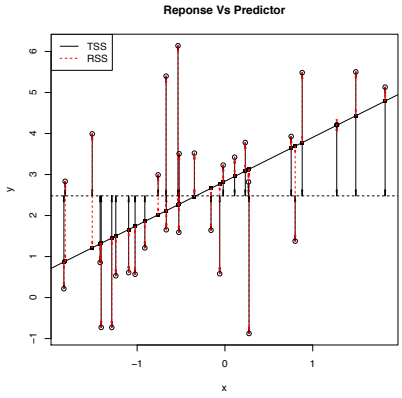
- Recall the two equations from setting the first derivatives equal to zero means

$$\sum_{i=1}^{n}\hat{e}_i = 0 \qquad \text{and} \qquad \sum_{i=1}^{n}\hat{e}_i x_i = 0$$

  thus

$$\text{TSS} = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}\hat{e}_i^2$$

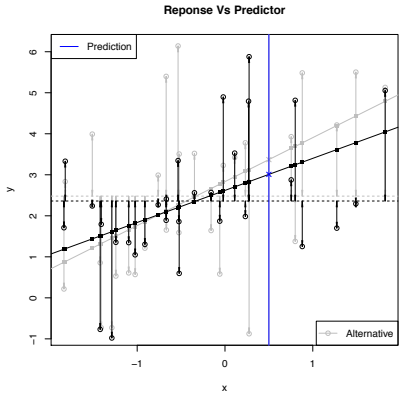


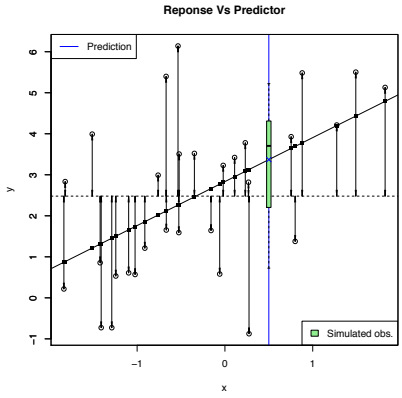


- The coefficient of determination

$$R^2 = \frac{\text{ESS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

- The variability in the value of Y^*_{n+1} has two layers given $\{x_1, x_2, \cdots, x_{n+1}\}$.





Reporting

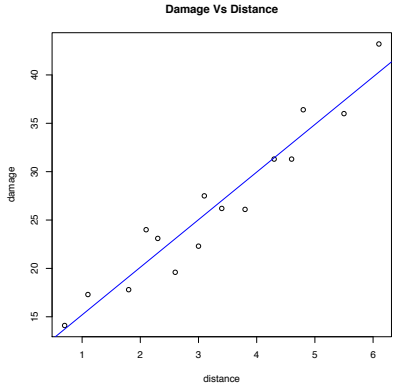
- When using regression in a project, you report should consists of two sections

1. Technical Notes
   - Exploratory Analysis
   - Model Specification
   - Checking Assumptions
   - Statistical Inference
2. Executive Summary
   - Overall quality of the model
   - Explaining the relationship between the response and the predictors
   - Making predictions
   - Addressing specific questions the project is about

Technical Notes

- Exploratory Analysis:

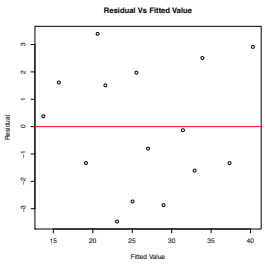
  The scatter plot of fire damage versus distance shows a strong, increasing, linear relationship. The greater the distance from the fire station, the greater the mean amount of damage that is caused by the house fire.

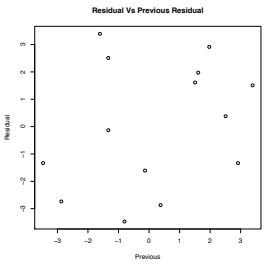


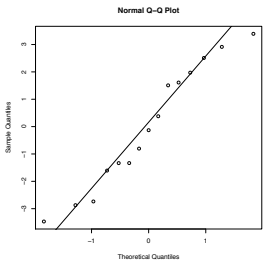
Technical Notes

- Checking Assumptions:

  The observations appear to be independent. The plot of residuals versus fitted values shows random scatter about 0. The Normal Q-Q plot shows the points lying close to the straight line indicating that the errors could have a normal distribution. The Shapiro-Wilk test provides no evidence against the hypothesis that the errors have a normal distribution (P-value = 0.4743).







Technical Notes

- Statistical Inference:

The F-test for regression provides extremely strong evidence against the hypothesis that distance from the fire station is not related to the amount of damage (P-value $= 1.248 \times 10^{-8}$). The Multiple R^2 is 0.9235 indicating that 92% of the variation in damage is explained by the variation in distance from the fire station, so the model should be very accurate for prediction. We have extremely strong evidence against the hypothesis that the intercept is equal to 0 (P-value $= 6.59 \times 10^{-6}$). We have extremely strong evidence against the hypothesis that the slope coefficient associated with distance is equal to 0 (P-value $= 1.25 \times 10^{-8}$).

```
Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept)  10.2779     1.4203   7.237 6.59e-06 ***
distance      4.9193     0.3927  12.525 1.25e-08 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.316 on 13 degrees of freedom
Multiple R-squared:  0.9235,    Adjusted R-squared:  0.9176
F-statistic: 156.9 on 1 and 13 DF,  p-value: 1.248e-08
```

Executive Summary

Our model explains 92% of the variation in house fire damage and should therefore be a very accurate model for prediction. We have extremely strong evidence that as the distance from the fire station increases, the average amount of damage increases. We estimate that if the house next to the fire station catches fire, the mean fire damage will be between \$7,200 and \$13,300. We estimate that for each additional mile from the fire station, the mean fire damage increases by between \$4,100 and \$5,800. Using our model, we predict that if a new fire occurs in a house that is 1 mile from the fire station, the damage will be between \$9,700 and \$20,700. The mean damage for house fires that are 1 mile from the fire station will be between \$12,900 and \$17,500. For a house that is 4 miles from the fire station, we predict the damage will be between \$24,800 and \$35,200. The mean damage for house fires that are 4 miles from the fire station will be between \$28,500 and \$31,400.

3 Multiple Linear Regression

3.1 Matrix Form

- In multiple linear regression, we explore the relation between the response

$$Y_i \qquad i = 1, 2, \ldots, n$$

and two or more predictors/explanatory variables, k of them in general,

$$X_{i1}, \quad X_{i2}, \qquad \ldots \qquad X_{ij}, \qquad \ldots \qquad X_{ik} \qquad i = 1, 2, \ldots, n$$

where we are assuming that we observed n cases, and

1. The conditional mean of the response is given by the following for all i

$$\mathbb{E}\left[Y_i \mid X_{i1}, X_{i2}, \ldots, X_{ik}\right] = \mathbb{E}\left[Y_i \mid \mathbf{X}_i\right] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$$

2. The errors have zero mean and constant variance

$$\mathbb{E}\left[\varepsilon_i \mid \mathbf{X}\right] = 0 \quad \text{and} \quad \operatorname{Var}\left[\varepsilon_i \mid \mathbf{X}\right] = \sigma^2 \qquad \text{where} \quad \varepsilon_i = Y_i - \mathbb{E}\left[Y_i \mid \mathbf{X}_i\right]$$

3. The errors are independent of $\mathbf{X}_i$, and of each other, for all i.
4. The errors follow a normal distribution.

- In matrix form, we have

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1k} \\ 1 & x_{21} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}, \text{ and } \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Q: How to interpret the regression coefficients β_0, β_1, …, β_n now?

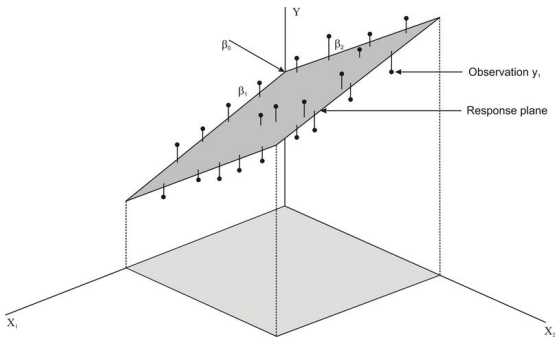
- Suppose there are only two predictors

$$m(x_{i1}, x_{i2}) = \mathbb{E}\left[Y_i \mid X_{i1} = x_{i1}, X_{i2} = x_{i2}\right] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$$

$$\implies \frac{\partial m}{\partial x_{i1}} = \beta_1; \qquad \frac{\partial m}{\partial x_{i2}} = \beta_2; \qquad m(0,0) = \beta_0$$

The regression coefficient represents the increase in the mean response associated with a unit increase in the predictor variable, *provided other predictor variables are held fixed.*

- The regression coefficients in multiple regression are sometimes called partial regression coefficients to emphasise that their interpretation requires that the other variables should be held fixed.



- Notice the essential idea is the same, but we are fitting a plane/hyperplane instead of a line to the data by estimating the parameters in

$$\mathbb{E}\left[Y \mid \mathbf{X}\right]$$

- Most of what we have done can be extended with a bit of linear algebra

$$\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\mathbf{e}}$$

we choose $\hat{\boldsymbol{\beta}} = \mathbf{b}$ that minimises residual sum of squares, which is given by

$$\begin{aligned} f(b_0, b_1, \ldots, b_k) = \hat{\mathbf{e}}^{\mathrm{T}}\hat{\mathbf{e}} &= (\mathbf{y} - \mathbf{X}\mathbf{b})^{\mathrm{T}} (\mathbf{y} - \mathbf{X}\mathbf{b}) \\ &= \left(\mathbf{y}^{\mathrm{T}} - \mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\right) (\mathbf{y} - \mathbf{X}\mathbf{b}) \\ &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{y}^{\mathrm{T}}\mathbf{X}\mathbf{b} - \mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{b} \end{aligned}$$

- Notice the all terms are scalars, thus equal

$$\mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \left(\mathbf{y}^{\mathrm{T}}\mathbf{X}\mathbf{b}\right)^{\mathrm{T}} = \text{scalar} = \mathbf{y}^{\mathrm{T}}\mathbf{X}\mathbf{b}$$

- Hence the function we need to minimise is given by

$$f(\mathbf{b}) = \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{b}$$

- Recall to minimise a function,

$$f(\mathbf{b}) = \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{b}$$

we set the gradient to zero,

$$\nabla f = 0 - 2\mathbf{X}^{\mathrm{T}}\mathbf{y} + 2\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{b}$$

Setting this to zero, we have

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

- Hence, the fitted value is given by

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \mathbf{P}\mathbf{y}$$

and the residual can be found using

$$\hat{\mathbf{e}} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{P})\,\mathbf{y}$$

- With more linear algebra, we have

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}) = \boldsymbol{\beta} + \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\varepsilon}$$

which means it is unbiased as expected,

$$\mathbb{E}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right] = \boldsymbol{\beta} + \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbb{E}\left[\boldsymbol{\varepsilon} \mid \mathbf{X}\right] = \boldsymbol{\beta}$$

- The variance is given by

$$\begin{aligned} \mathrm{Var}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right] &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathrm{Var}\left[\boldsymbol{\varepsilon} \mid \mathbf{X}\right]\left(\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\right)^{\mathrm{T}} \\ &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\sigma^2\mathbf{I}\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} = \sigma^2\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \end{aligned}$$

- With the normal assumption, we see

$$\hat{\boldsymbol{\beta}} \sim \text{Normal}\left(\boldsymbol{\beta}, \sigma^2 \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\right)$$

- For the estimation σ^2, we only need adjust the scalar a bit

$$\hat{\sigma}_u^2 = \frac{1}{n-k-1}\hat{\mathbf{e}}^{\mathrm{T}}\hat{\mathbf{e}} \qquad \text{where} \quad \hat{\mathbf{e}} = (\mathbf{I}-\mathbf{P})\,\mathbf{y}$$

  so that it is unbiased as well as being consistent as before.
- Considering the residual under resampling, it can be shown

$$\hat{\mathbf{e}} = (\mathbf{I}-\mathbf{P})\,\mathbf{Y} = (\mathbf{I}-\mathbf{P})\,(\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon})$$

  is an unbiased and consistent estimator of the error $\mathbf{e}$, and the variance is

$$\begin{aligned}\text{Var}\left[\hat{\mathbf{e}} \mid \mathbf{X}\right] &= \text{Var}\left[(\mathbf{I}-\mathbf{P})\,(\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}) \mid \mathbf{X}\right] \\ &= (\mathbf{I}-\mathbf{P})\,\text{Var}\left[\boldsymbol{\varepsilon} \mid \mathbf{X}\right](\mathbf{I}-\mathbf{P})^{\mathrm{T}} \\ &= (\mathbf{I}-\mathbf{P})\,\sigma^2\mathbf{I}\,(\mathbf{I}-\mathbf{P})^{\mathrm{T}} = \sigma^2\,(\mathbf{I}-\mathbf{P})\end{aligned}$$

- Thus with the normal assumption, we have

$$\hat{\mathbf{e}} \sim \text{Normal}\left(\mathbf{0}, \sigma^2\,(\mathbf{I}-\mathbf{P})\right)$$

3.2 Diagnostics

Q: How can we check assumption 1. graphically? What do you expect to see?

1. The conditional mean of the response is given by the following for all i

$$\begin{aligned}\mathbb{E}\left[Y_i \mid X_{i1}, X_{i2}, \ldots, X_{ik}\right] &= \mathbb{E}\left[Y_i \mid \mathbf{X}_i\right] \\ &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}\end{aligned}$$

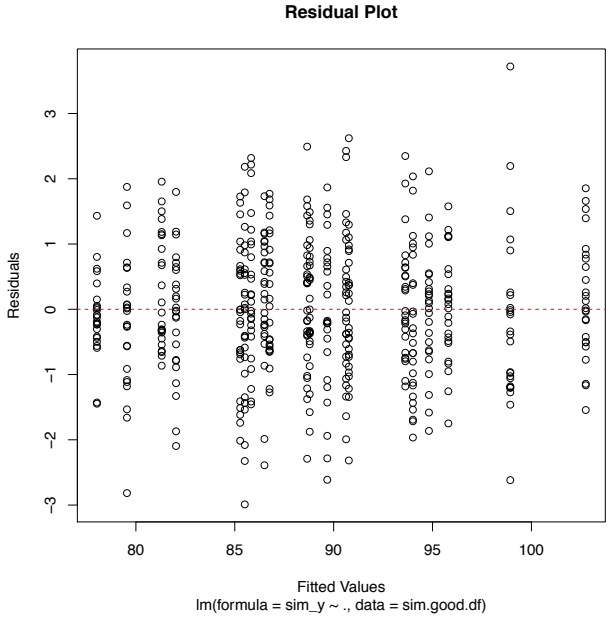
- Since the conditional mean is not available, but the fitted value

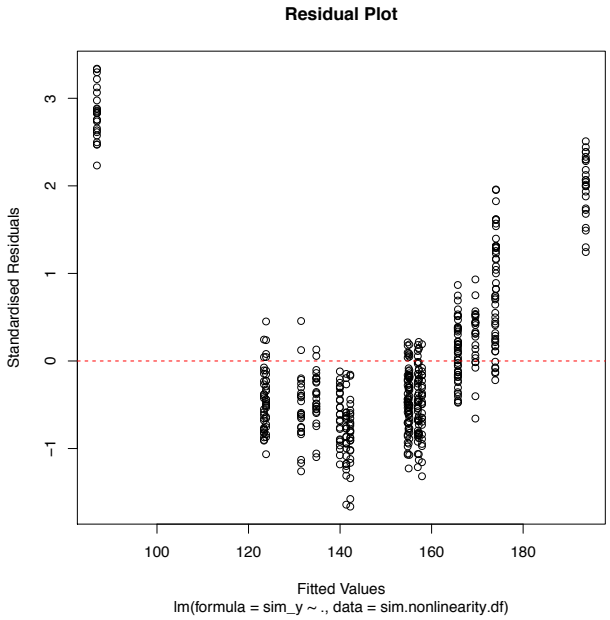
$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_k x_{ik}$$

  our estimate of it is, thus y_i is expected to be scattered around it if 1. holds.
- This is hardly different from the case of simple linear regression, we look at

$$\text{the residual plot:} \qquad \hat{e}_i \qquad \text{Vs} \qquad \hat{y}_i$$

  instead of y_i Vs $\hat{y}_i$ to avoid the visual distraction of a sloping line.
- Having a random scatter of points round x-axis means we have no evidence against 1.. If there is a pattern, then there is evidence against 1..





Q: How can we check assumption 2. graphically? What do you expect to see?

2. The errors have zero mean and constant variance for all i

$$\mathbb{E}\left[\varepsilon_i \mid \mathbf{X}\right] = 0 \quad \text{and} \quad \text{Var}\left[\varepsilon_i \mid \mathbf{X}\right] = \sigma^2$$

- Recall for simple linear regression, we have

$$\text{Var}\left[Y_i - \hat{Y}_i \mid X_i\right] = \sigma^2 \left(1 - \frac{1}{n} - \frac{(x_i - \bar{x})^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}\right)$$

- For multiple linear regression, we have derived the following earlier

$$\text{Var}\left[\mathbf{Y} - \hat{\mathbf{Y}} \mid \mathbf{X}\right] = \sigma^2 (\mathbf{I} - \mathbf{P})$$

Q: What does this formula suggest?

- Internally studentised residual/standardised residual is given by

$$\hat{e}'_i = \frac{\hat{e}_i}{\hat{\sigma}_u \sqrt{1 - p_{ii}}}$$

where p_{ii} is the ith diagonal element of $\mathbf{P}$ and

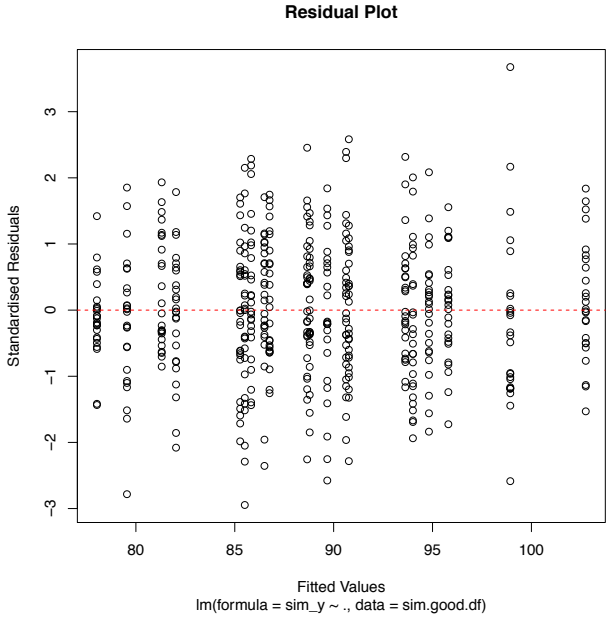
$$\hat{\sigma}_u^2 = \frac{1}{n-k-1} \hat{\mathbf{e}}^{\mathrm{T}} \hat{\mathbf{e}}$$

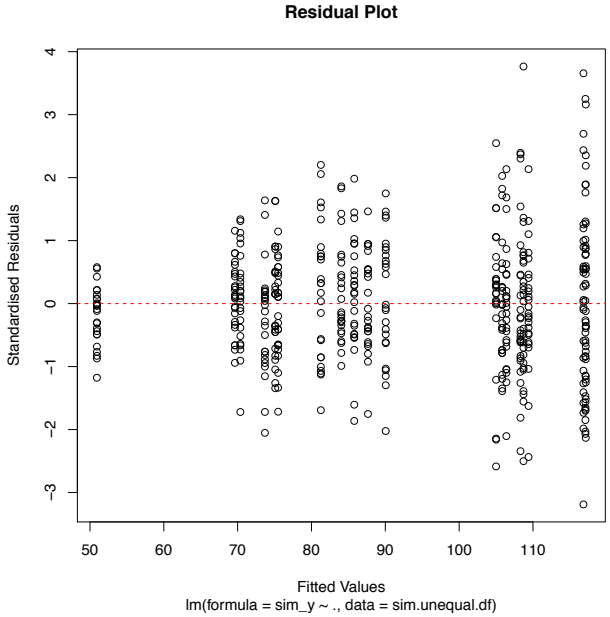
$$= \frac{1}{n-k-1} \mathbf{y}^{\mathrm{T}} (\mathbf{I} - \mathbf{P}) \mathbf{y}$$

Q: What do you expect to see in a plot of

$$\hat{e}'_i \qquad \text{VS} \qquad \hat{y}_i$$

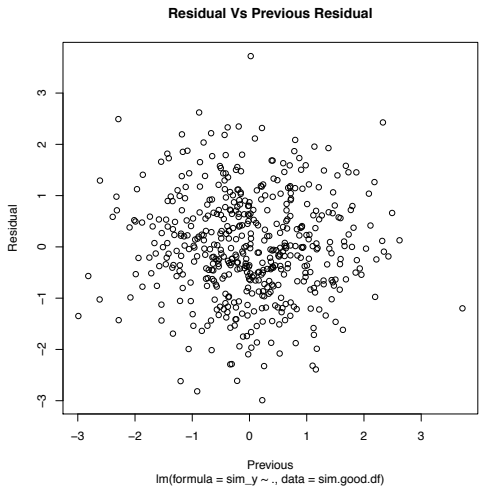
if assumption 1. and 2. are satisfied?

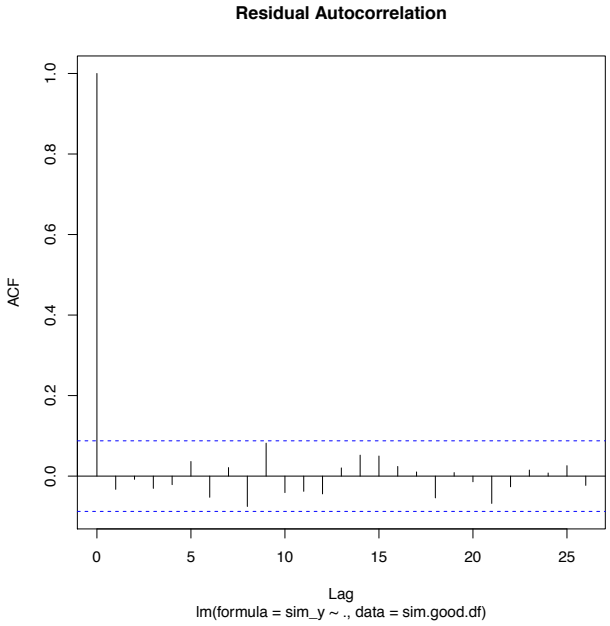


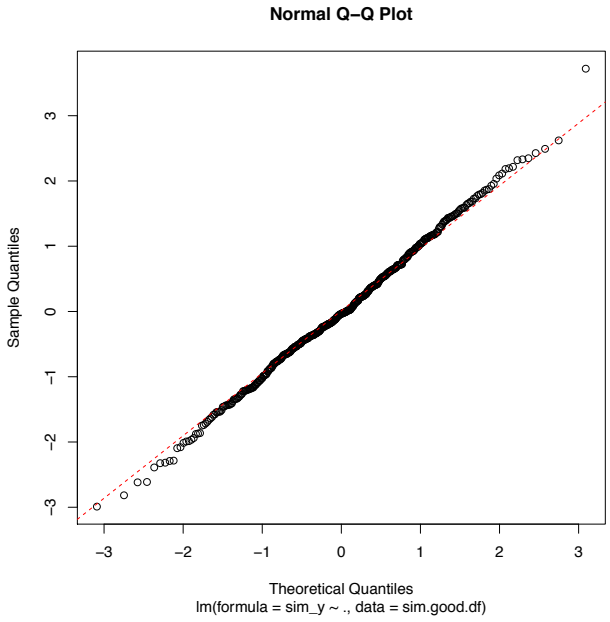


Q: How about assumption 3. and 4.?

- It is clear that we can reply on the same diagnostic tools to for 3. and 4.







Transistor Study

- A study on transistor gain between emitter and collector in an integrated circuit device was done in early days, the following variables are collected.

`dit`	Drive-in time in minutes
`dose`	ions $\times 10^{14}$
`gain`	measured in hFE

- It is interested to determine whether
  1. drive-in time or dose influences gain linearly
  2. drive-in time influences gain for a given dose linearly
  3. dose influences gain for a given drive-in time linearly
  4. there is any reason to use both drive-in time and dose to explain gain

- After loading the data

```
> # install.packages("xlsx")
> library(xlsx)
> trans.df = read.xlsx(
+   file = "~/Desktop/transistor.xlsx",
+   sheetIndex = 1)
```

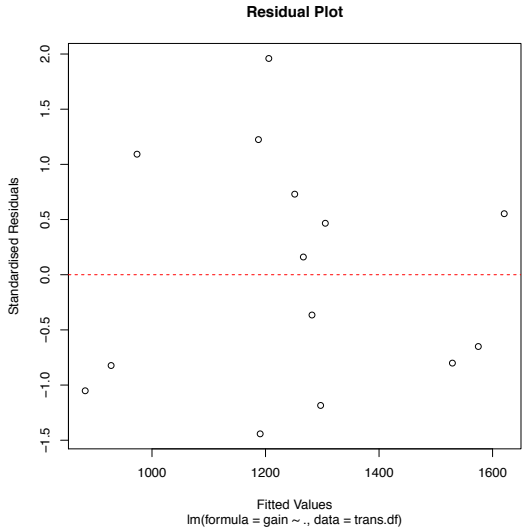
- We consider the full model, that is, including all the regression variables

```
> trans.LM = lm(gain~., data = trans.df)

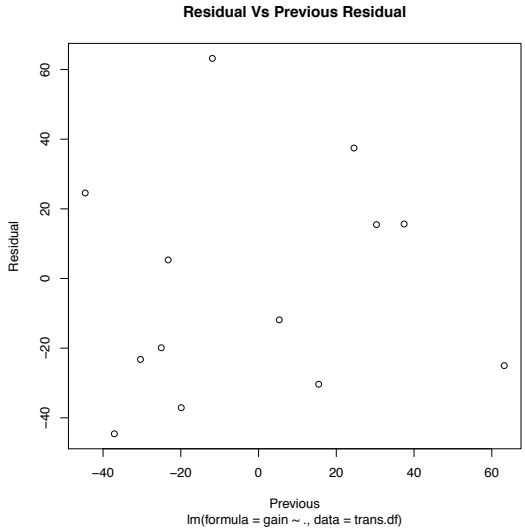
> model = trans.LM              # Reusable
>
> fvs = fitted.values(model)    # fitted values
> res = residuals(model)        # residuals
> sres = rstandard(model)       # standardised
```

- We must check assumptions before we can use it to answer those questions.

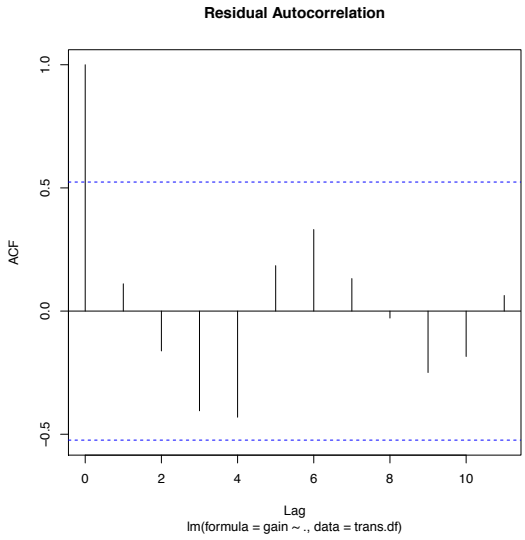
- There is no indication of nonlinearity, correlation or non-constant variance.



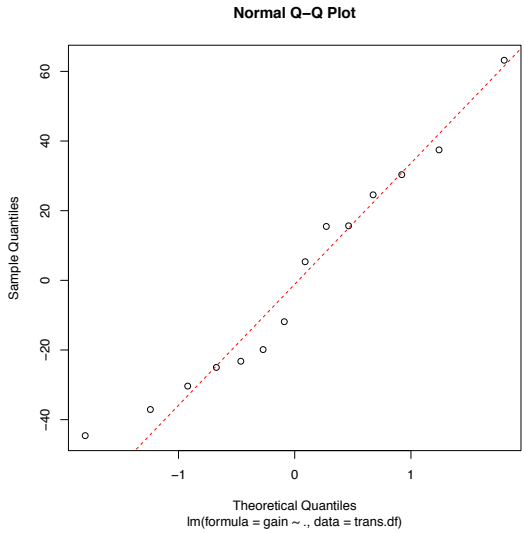
- There is no indication that the residuals are correlated.



- There is no indication that the residuals are correlated.



- There is no clear indication that normality is violated.



- Given there is no indication of any violation of the assumptions, we proceed

  `> summary(model)`

```
Call:
lm(formula = gain ~ ., data = trans.df)

Residuals:
    Min      1Q  Median      3Q     Max
-44.584 -24.565  -3.266  22.330  63.201

Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept) -520.0767   192.1071  -2.707  0.02039 *
dit           10.7812     0.4743  22.730 1.35e-10 ***
dose        -152.1489    36.6754  -4.149  0.00162 **
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 34.93 on 11 degrees of freedom
Multiple R-squared:  0.9798,    Adjusted R-squared:  0.9762
F-statistic: 267.2 on 2 and 11 DF,  p-value: 4.742e-10
```

Q: What is the base for answering question 1.?

Does drive-in time or dose influence gain linearly?

Q: Any evidence against the claim that neither `dit` nor `dose` influences `gain`?

Q: Any one remember how F-test works?

- Recall the underlying hypothesis of F-statistics is that all slopes are zero

$$\beta_1 = \beta_2 = \cdots = \beta_k = 0$$

- If the p-value corresponding to a F-statistics is small, then we have evidence against the regression coefficients being all zero.

Q: Can you remember how to derive F-test?

- Under the above hypothesis and the model assumptions, F-statistics follows

$$F(d_1 = k, d_2 = n - k - 1)$$

where k is the number of predictors, and n is the number of observations.

- We can work out the F-statistics manually by using R as calculator

```
> n = nrow(trans.df)
> n
```

```
[1] 14
```

```
> y.bar = mean(trans.df$gain)

> tmp = trans.df$gain - y.bar

> yss = sum(tmp^2)

> rse = sigma(trans.LM)^2

> f0 = ( yss - (n - 2 - 1) * rse ) / ( 2* rse)
>
> f0
```

```
[1] 267.177
```

```
> 1 - pf(f0, 2, n - 2 -1)
```

```
[1] 4.741632e-10
```

- It verifies the F-statistics and the extremely small p-value in the summary,

```
> summary(model)
```

```
Call:
lm(formula = gain ~ ., data = trans.df)

Residuals:
    Min      1Q  Median      3Q     Max
-44.584 -24.565  -3.266  22.330  63.201

Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept) -520.0767   192.1071  -2.707  0.02039 *
dit           10.7812     0.4743  22.730 1.35e-10 ***
dose        -152.1489    36.6754  -4.149  0.00162 **
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 34.93 on 11 degrees of freedom
Multiple R-squared:  0.9798,    Adjusted R-squared:  0.9762
F-statistic: 267.2 on 2 and 11 DF,  p-value: 4.742e-10
```

- Hence we have extremely strong evidence against the claim that neither `dit` nor `dose` influences `gain` in this case, therefore, we favour the conclusion either drive-in time or dose, or both influence gain linearly.

Q: What is the base for answering question 2. and 3.?

Does drive-in time influence gain for a given dose linearly?
Does dose influence gain for a given drive-in time linearly?

- Recall we have derived the following

$$\hat{\boldsymbol{\beta}} \sim \text{Normal}\left(\boldsymbol{\beta}, \sigma^2 \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\right)$$

from which t-test can be derived to be used as the bases.

- Individually, for the jth predictor, the t-test has the hypothesis that

$$\beta_j = 0$$

- If the p-value corresponding to a t-statistics is small, then we have evidence against the corresponding predictor not influencing the response linearly.

```
> summary(model)
```

```
Call:
lm(formula = gain ~ ., data = trans.df)

Residuals:
    Min      1Q  Median      3Q     Max
-44.584 -24.565  -3.266  22.330  63.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -520.0767   192.1071  -2.707  0.02039 *
dit           10.7812     0.4743  22.730 1.35e-10 ***
dose        -152.1489    36.6754  -4.149  0.00162 **
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 34.93 on 11 degrees of freedom
Multiple R-squared:  0.9798,    Adjusted R-squared:  0.9762
F-statistic: 267.2 on 2 and 11 DF,  p-value: 4.742e-10
```

- We have extremely strong evidence against the claim that `dit` does not influence `gain` for a fixed value of `dose`, therefore, we favour the conclusion the drive-in time influence gain linearly while holding the does constant.
- Similarly, we have very strong evidence against the claim that `dose` does not influence `gain` for a fixed value of `dit`, therefore, we favour the conclusion dose influence gain linearly while holding the drive-in time constant.

Q: What is the base for answering question 4.?

Is there any reason to use both drive-in time and dose to explain gain?

Q: Can we rely on F-test?

Q: How about t-test?

Q: Is there anything else in summary that can be used to address this question?

```
> summary(model)
```

```
Call:
lm(formula = gain ~ ., data = trans.df)

Residuals:
    Min      1Q  Median      3Q     Max
-44.584 -24.565  -3.266  22.330  63.201

Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept) -520.0767   192.1071  -2.707  0.02039 *
dit           10.7812     0.4743  22.730 1.35e-10 ***
dose        -152.1489    36.6754  -4.149  0.00162 **
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 34.93 on 11 degrees of freedom
Multiple R-squared:  0.9798,    Adjusted R-squared:  0.9762
F-statistic: 267.2 on 2 and 11 DF,  p-value: 4.742e-10
```

- Essentially, this question could be answered by comparing the submodel

  `> trans.dose.LM = lm(gain~dose, data = trans.df)`

  and the submodel

  `> trans.dit.LM = lm(gain~dit, data = trans.df)`

  with the full model

  `> trans.LM = lm(gain~., data = trans.df)`

- The Adjusted R-squared in the summary is a crude to do the comparison.
- Recall R-squared, which is also known as the coefficient of determination, is

$$R^2 = \frac{\text{ESS}}{\text{TSS}} = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^{n} \hat{e}_i^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

  which gives the proportional of variability explained by the model.

- The Adjusted R-squared is given by

$$\bar{R}^2 = 1 - \left(1 - R^2\right) \frac{n-1}{n-k-1}$$

Q: Given R-squared can be understood as

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \hat{e}_i^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n} \hat{e}_i^2}{\frac{1}{n}\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

why adjusted R-squared is simply correcting the biased MLE.

- This alteration allows $\bar{R}^2$ to take model complexity into account.
- A submodel might have a larger $\bar{R}^2$, if so, it is preferred over the full model.

Q: What is the difference between R^2 and the adjusted R^2?

- R^2 is a measure of goodness of fit when all the assumptions are satisfied.
- However, large R^2 does not mean:

1. the assumptions are satisfied.
2. a better predictive model.
3. a better model across different datasets.
4. a better model when the models have different number of parameters.

- The adjusted R^2 is a relative measure to address 4., and it cannot be

1. interpreted alone.
2. used for two models that have different responses.

- Like F-test and t-test in the summary, both R^2 and $\bar{R}^2$ are only meaningful if all model assumptions are satisfied.
- However, it can be shown there is evidence against the two submodels for `trans.df` satisfying the models assumptions, thus we prefer the full model.

3.3 Polynomial

- Under multiple linear regression,

1. The conditional mean of the response is given by

$$\begin{aligned}\mathbb{E}\left[Y_i \mid X_{i1}, X_{i2}, \ldots, X_{ik}\right] &= \mathbb{E}\left[Y_i \mid \mathbf{X}_i\right] \\ &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}\end{aligned}$$

which means there is no curvature in conditional mean.

- For a a general region, there is no reason to expect it is a flat hyperplane.
- Using polynomials of x_{*j} is the easiest way to model nonlinearity, we assume

$$\mathbb{E}\left[Y_i \mid \mathbf{X}_i\right] = \beta_0 + P_1^{d_1} + P_2^{d_2} + \cdots + P_k^{d_k}$$

where $P_j^{d_j} = \sum_{\ell=1}^{d_j} \beta_{j\ell} x_{ij}^{\ell}$ is a d_jth degree polynomial of the predictor x_{*j}.

- This is often known as polynomial regression, which is linear in parameters.
- In terms of estimation and inference, we only modify the matrix $\mathbf{X}_{n\times(k+1)}$ to

$$\mathbf{X}^*_{n\times(m+1)} = \begin{bmatrix} 1 & x_{11} & x_{11}^2 & \cdots & x_{1k}^{d_k} \\ 1 & x_{21} & x_{21}^2 & \cdots & x_{2k}^{d_k} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & x_{n1} & x_{n1}^2 & \cdots & x_{nk}^{d_k} \end{bmatrix}, \qquad \text{where} \quad m = \sum_{j=1}^{k} d_j,$$

the rest remains the same

$$\mathbf{y} = \mathbf{X}^*\boldsymbol{\beta} + \mathbf{e}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}, \text{ and } \quad \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

- The following illustrates when and how to used polynomial regression,

```
> sim.df = read.csv(
+    file = "~/Desktop/sim_poly_class.csv.bz2")

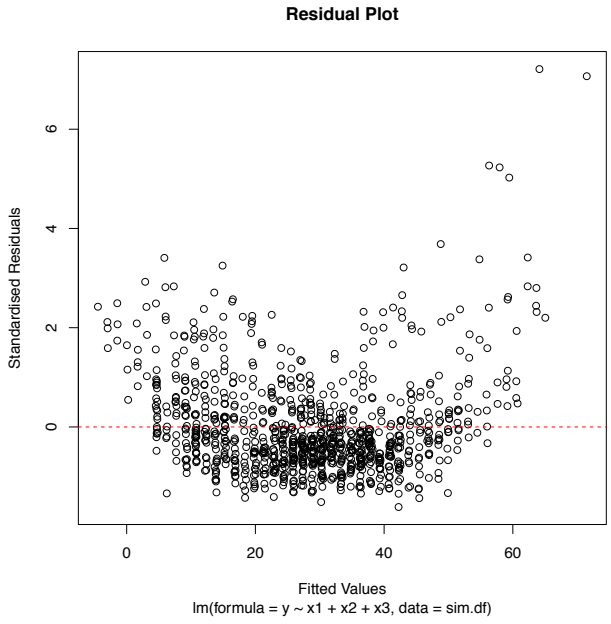
> str(sim.df)
```

```
'data.frame':   1000 obs. of  4 variables:
 $ y : num  4.31 16.09 10.67 3.24 33.5 ...
 $ x1: int  1 1 1 1 1 1 1 1 1 1 ...
 $ x2: int  3 6 5 3 9 9 3 7 5 5 ...
 $ x3: num  -0.3706 -1.5564 -0.5157 -1.5024 -0.0638
```

- Since there are only 3 regressors, we construct the full multiple linear model

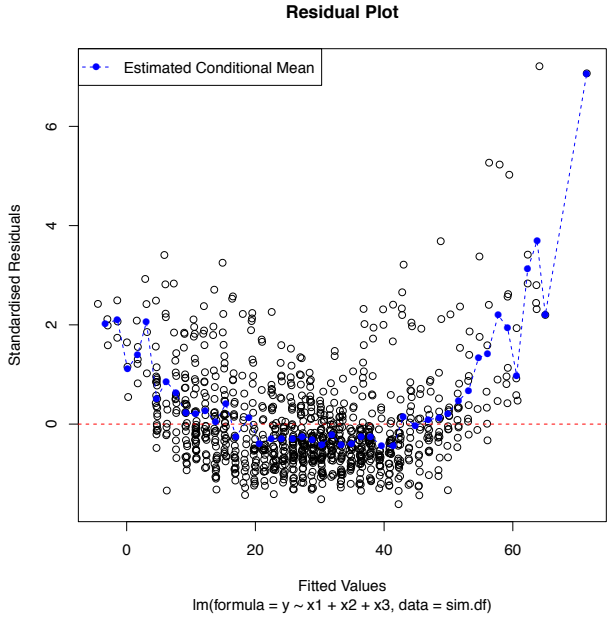
```
> sim.LM = lm(y~x1+x2+x3, data = sim.df)
```

- The residual plot indicates nonlinearity or error is correlated with fitted value.



- The nonlinearity is clearer if we add estimated conditional means to the plot

```
> c.mean.plot =
+   function(tmp.x, tmp.y, n.each, pos = "topleft"){
+
+   n.x = round(length(unique(tmp.x)) / n.each)
+   x.group = cut(tmp.x, breaks = n.x, labels = FALSE)
+
+   x.vec = tapply(tmp.x, x.group, mean)
+   y.vec = tapply(tmp.y, x.group, mean)
+
+   points(x.vec, y.vec, col = "blue", pch = 16)
+   lines(x.vec, y.vec, col = "blue", lty = 2)
+
+   legend(pos, "Estimated Conditional Mean",
+          col = "blue", lty = 2, pch = 16)
+
+ }
```



- The residual plot only indicates a potential problem,

$$\hat{e}'_i \qquad \text{Vs} \qquad \hat{y}_i$$

it provides no information regarding the exact nature of the problem.

Q: Suppose the conditional mean is given by or can be approximated by

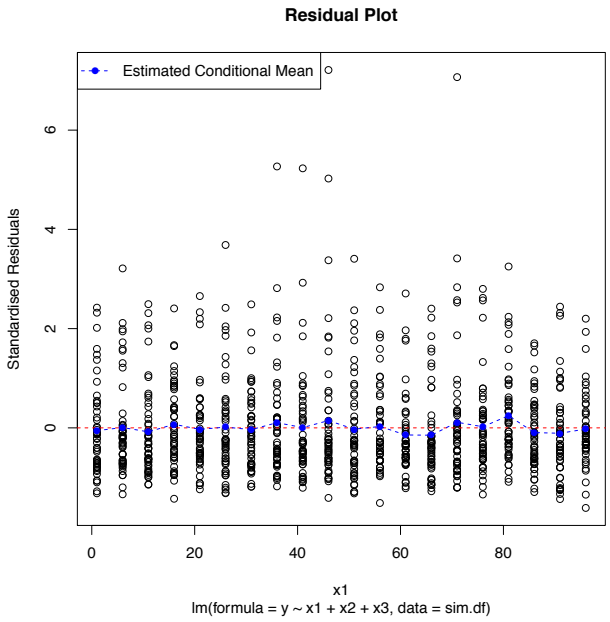
$$\mathbb{E}\left[Y_i \mid \mathbf{X}_i\right] = \beta_0 + P_1^{d_1} + P_2^{d_2} + \cdots + P_k^{d_k}$$

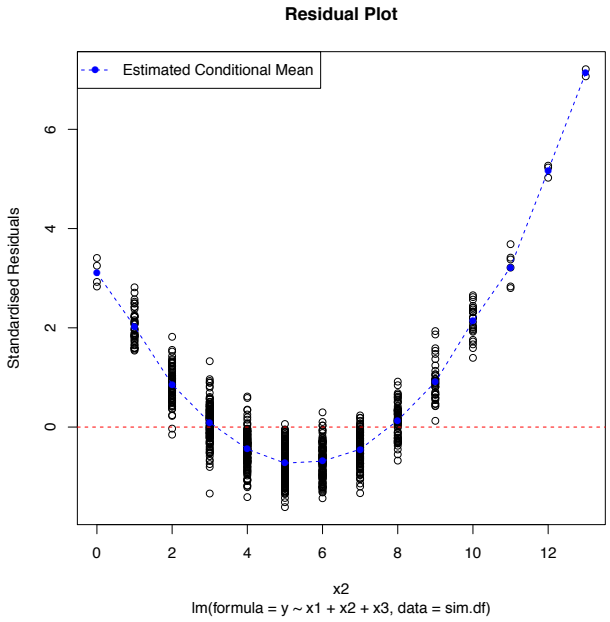
what do you expect to see in the residual plot of

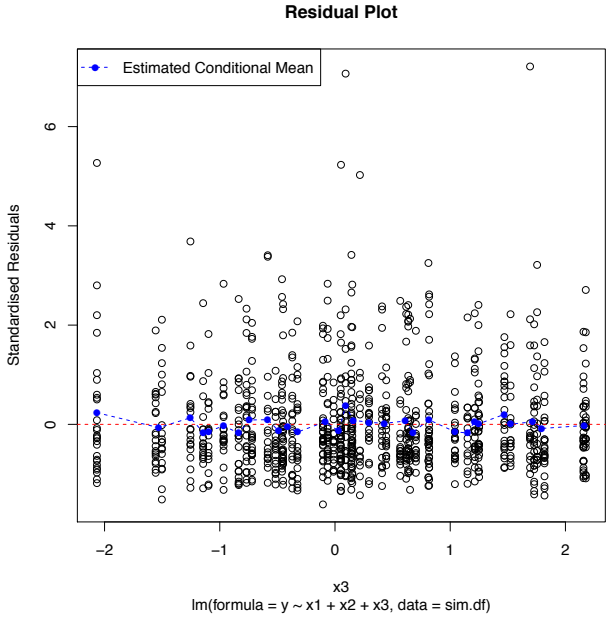
$$\hat{e}'_i \qquad \text{Vs} \qquad x_{ij}$$

that is, the standardised residual against one of the regressors?

- This is the simplest way to determine whether a polynomial term is needed.
- So we usually produce residual plots of both kinds as a standard practice.







Q: What do those residual plots show us?

- It seems there is nothing left in the residual that can be explained by

  `x1` and `x3`.

- However, the residual is clearly not independent of

  `x2`.

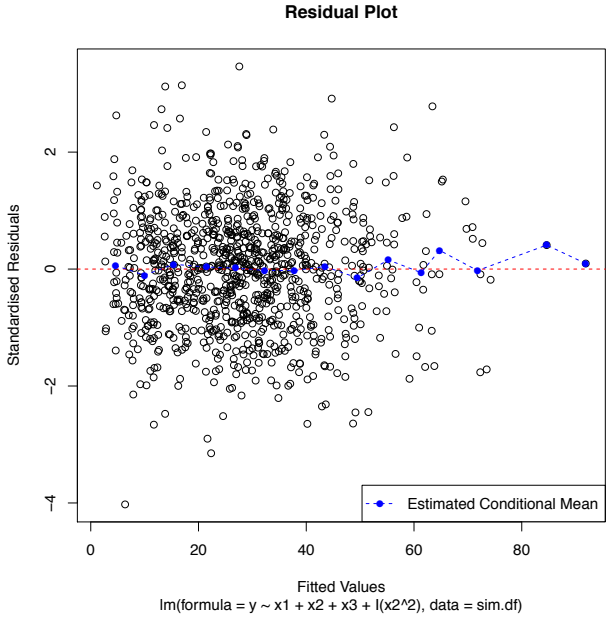
- One of many possibilities for seeing such residual plots is

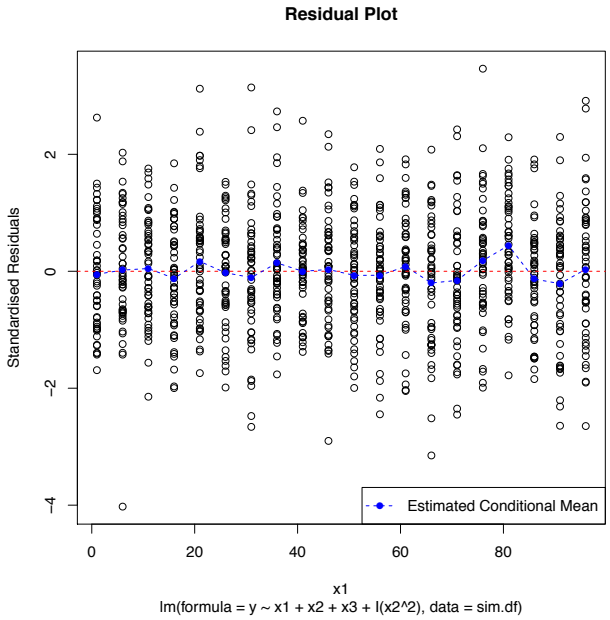
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \underbrace{\beta_4 X_2^2 + \varepsilon^*}_{\varepsilon}$$

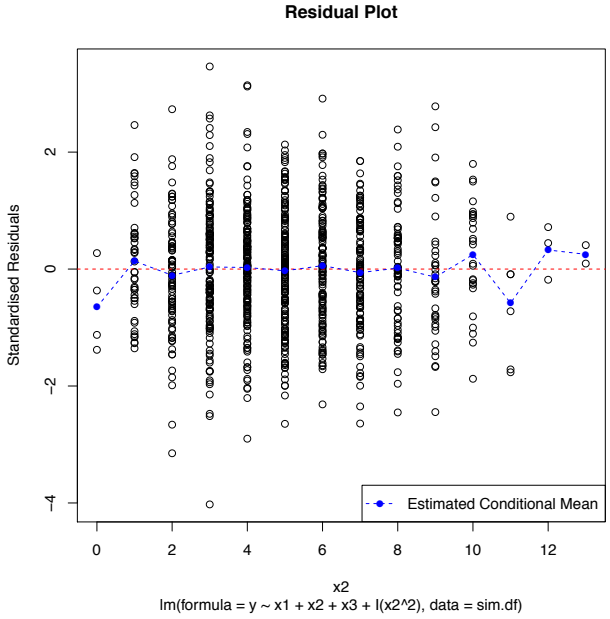
- Based on those residual plots, we consider the following model next

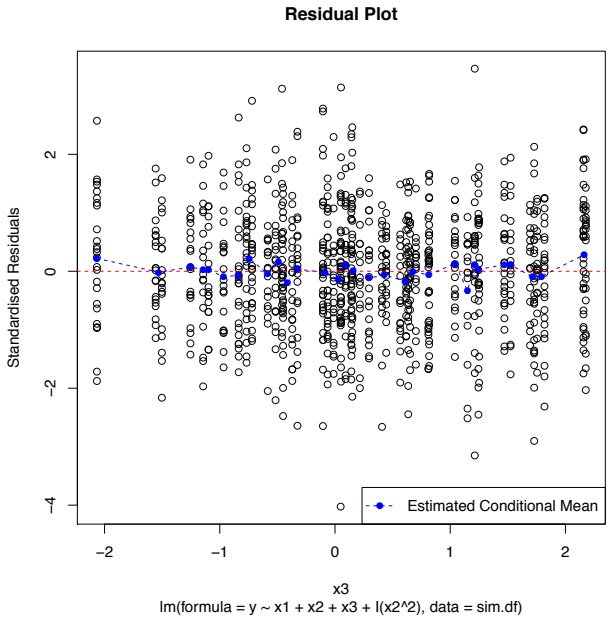
```
> sim.quad.LM = lm(y~x1+x2+x3+I(x2^2),data = sim.df)
```

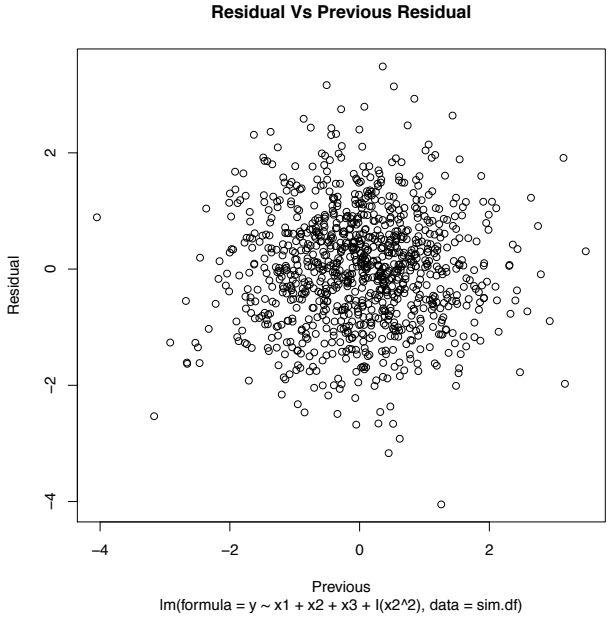
  which seems to satisfy all the model assumptions.

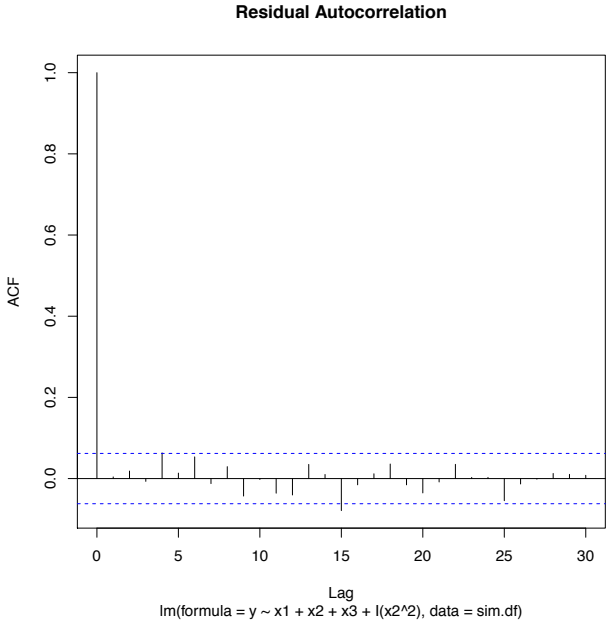

Residual Plot
Standardised Residuals
Fitted Values
lm(formula = y ~ x1 + x2 + x3 + I(x2^2), data = sim.df)
Estimated Conditional Mean

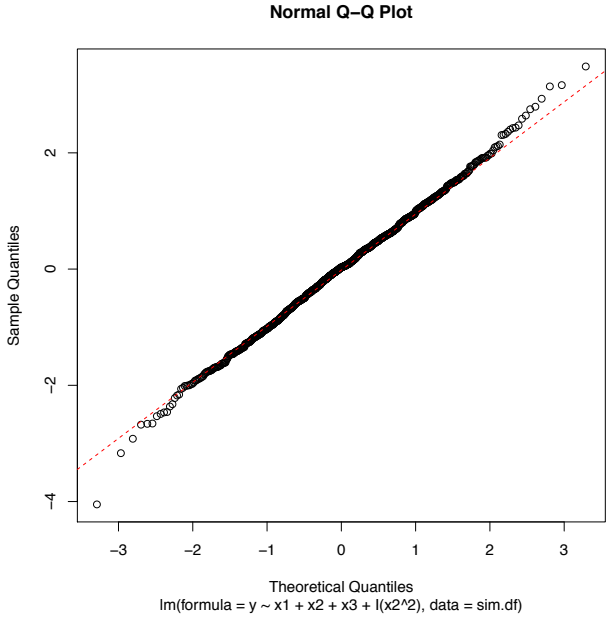

Residual Plot
Standardised Residuals
x1
lm(formula = y ~ x1 + x2 + x3 + I(x2^2), data = sim.df)
Estimated Conditional Mean


Residual Plot
Standardised Residuals
x2
lm(formula = y ~ x1 + x2 + x3 + I(x2^2), data = sim.df)
Estimated Conditional Mean


Residual Plot
Standardised Residuals
x3
lm(formula = y ~ x1 + x2 + x3 + I(x2^2), data = sim.df)
Estimated Conditional Mean


Residual Vs Previous Residual
Residual
Previous
lm(formula = y ~ x1 + x2 + x3 + I(x2^2), data = sim.df)


Residual Autocorrelation
ACF
Lag
lm(formula = y ~ x1 + x2 + x3 + I(x2^2), data = sim.df)



- Since we have no evidence against all assumptions being satisfied, we trust

```
> summary(sim.quad.LM)
```

```
Call:
lm(formula = y ~ x1 + x2 + x3 + I(x2^2), data = sim.df)

Residuals:
    Min      1Q  Median      3Q     Max
-4.0491 -0.6691  0.0280  0.6326  3.4861

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) 0.277649   0.146033   1.901   0.0576 .
x1          0.300215   0.001106 271.321  < 2e-16 ***
x2          0.238286   0.052575   4.532 6.54e-06 ***
x3          0.155625   0.030654   5.077 4.58e-07 ***
I(x2^2)     0.397486   0.004636  85.741  < 2e-16 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 1.008 on 995 degrees of freedom
Multiple R-squared:  0.9948,    Adjusted R-squared:  0.9947
F-statistic: 4.73e+04 on 4 and 995 DF,  p-value: < 2.2e-16
```

Q: What can you learn from comparing this with the output of

```
> summary(sim.LM)
```

```
> summary(sim.LM)
```

```
Call:
lm(formula = y ~ x1 + x2 + x3, data = sim.df)

Residuals:
    Min      1Q  Median      3Q     Max
-4.6987 -1.8885 -0.7724  1.0437 20.8883

Coefficients:
            Estimate Std. Error t value Pr(>t)
```

```
(Intercept) -9.322246   0.271399 -34.349   <2e-16 ***
x1           0.298398   0.003203  93.176   <2e-16 ***
x2           4.585967   0.040203 114.071   <2e-16 ***
x3           0.066189   0.088685   0.746    0.456
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.918 on 996 degrees of freedom
Multiple R-squared:  0.9561,    Adjusted R-squared:  0.956
F-statistic:  7233 on 3 and 996 DF,  p-value: < 2.2e-16
```

- We could wrongly drop `x3` if we didn't do residual analysis first!
- Transformations can be used in as before to alleviate non-constant variance and non-normality in multiple linear regression or polynomial models.
- Consider the following dataset,

`Temperature`	temperature standard deviation during production
`Density`	density of the final product
`Rate`	rate of production
`Defective`	average number of defects per 1000 produced

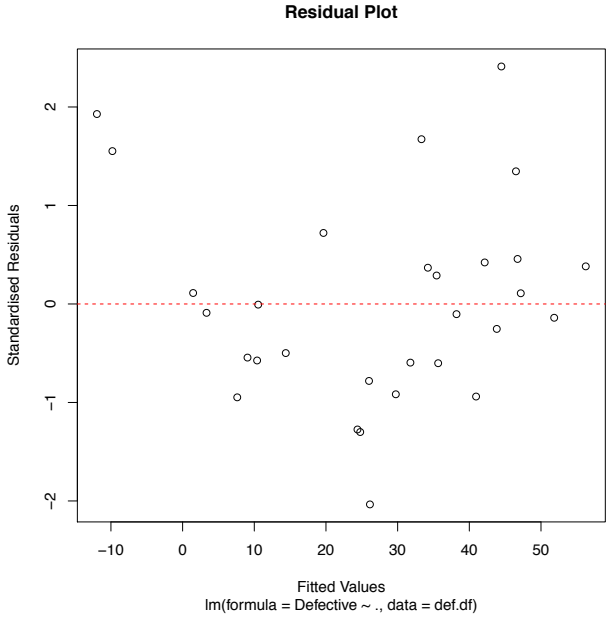
  which is gathered for quality control in production.

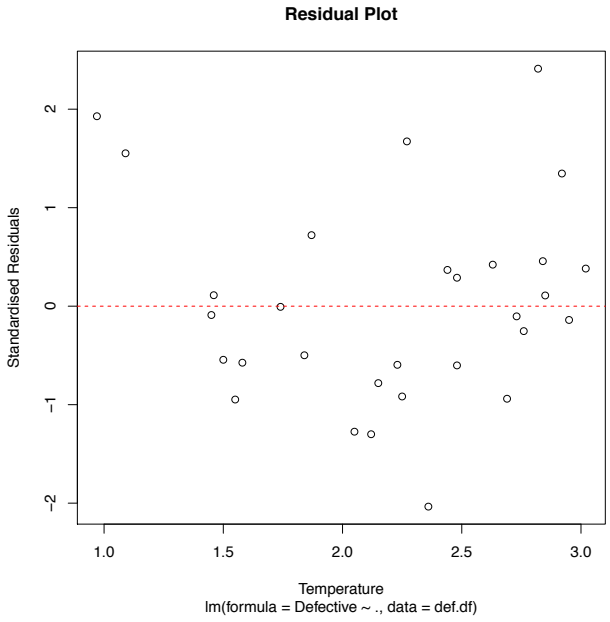
```
> str(def.df)
```

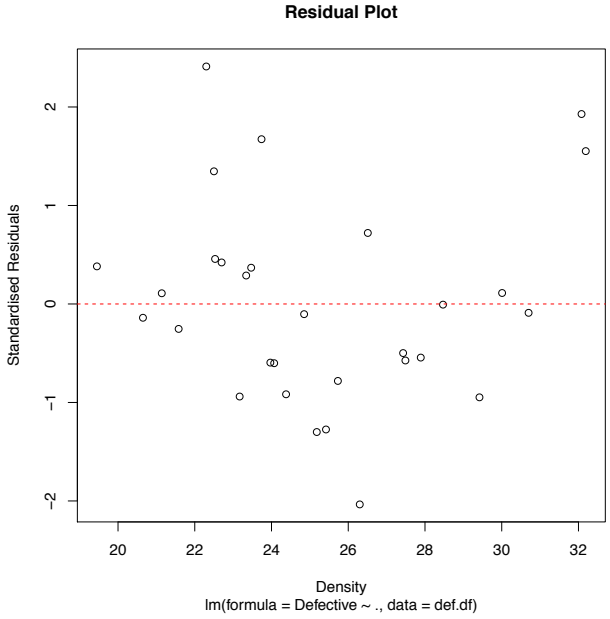
```
'data.frame':   30 obs. of  4 variables:
 $ Temperature: num  0.97 2.85 2.95 2.84 1.84 2.05 1.5 2.48 2.23 3.02 ...
 $ Density    : num  32.1 21.1 20.6 22.5 27.4 ...
 $ Rate       : num  178 254 273 273 211 ...
 $ Defective  : num  0.2 47.9 50.9 49.7 11 15.6 5.5 37.4 27.8 58.7 ...
```

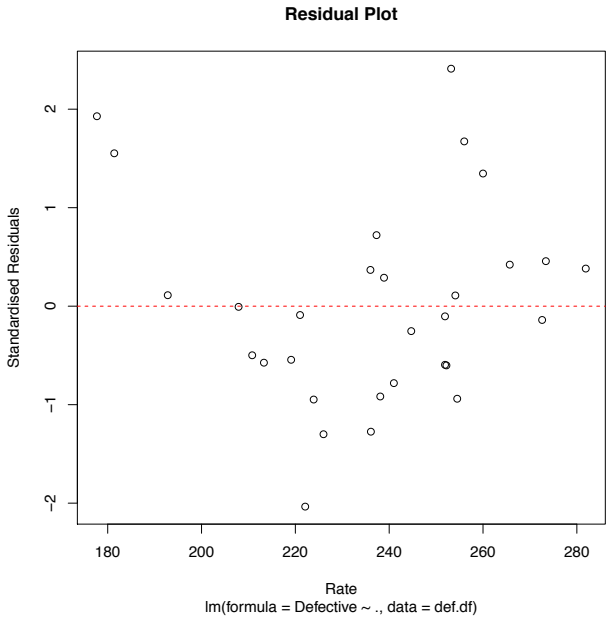
- We start with the multiple linear regression model with all the variables

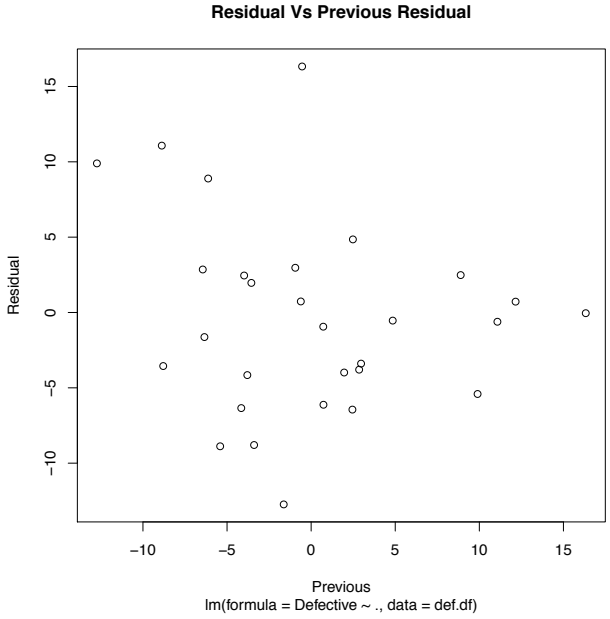
```
> def.LM = lm(Defective~., data = def.df)
```

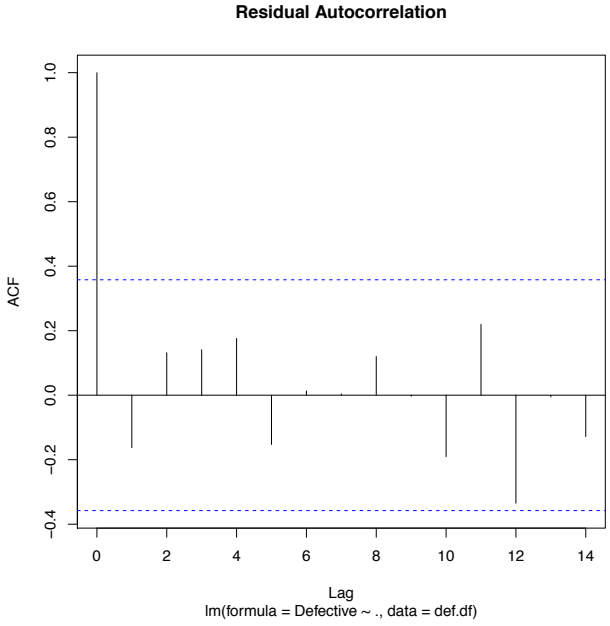

Residual Plot
Standardised Residuals
Fitted Values
lm(formula = Defective ~ ., data = def.df)

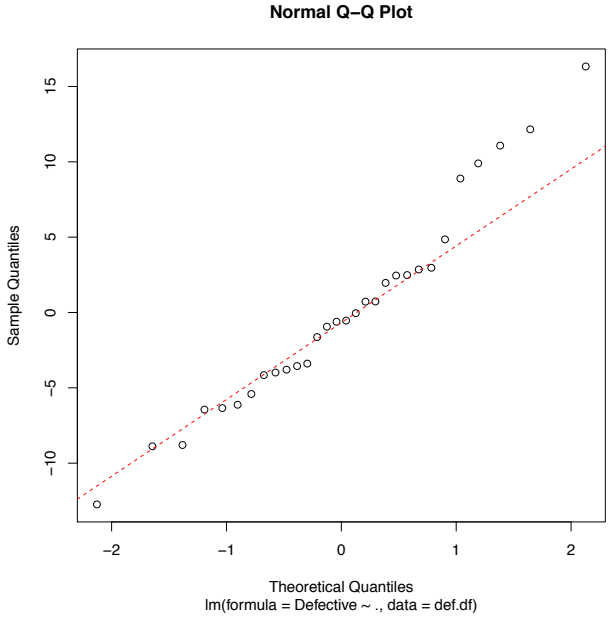

Residual Plot
Standardised Residuals
Temperature
lm(formula = Defective ~ ., data = def.df)

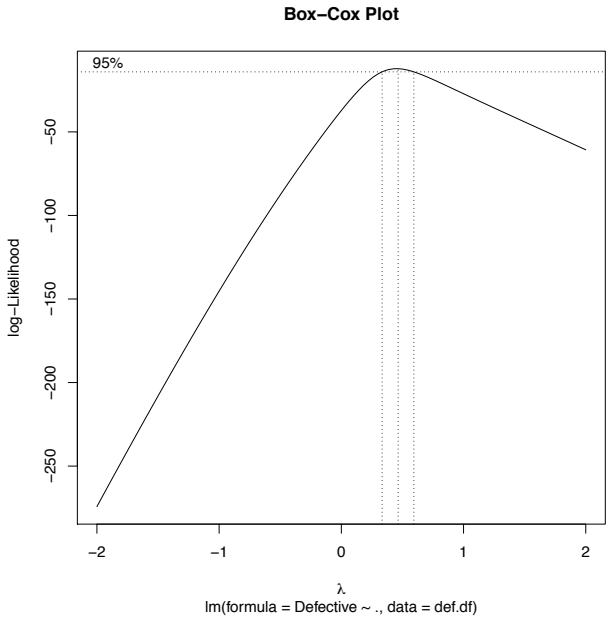

Residual Plot
Standardised Residuals
Density
lm(formula = Defective ~ ., data = def.df)

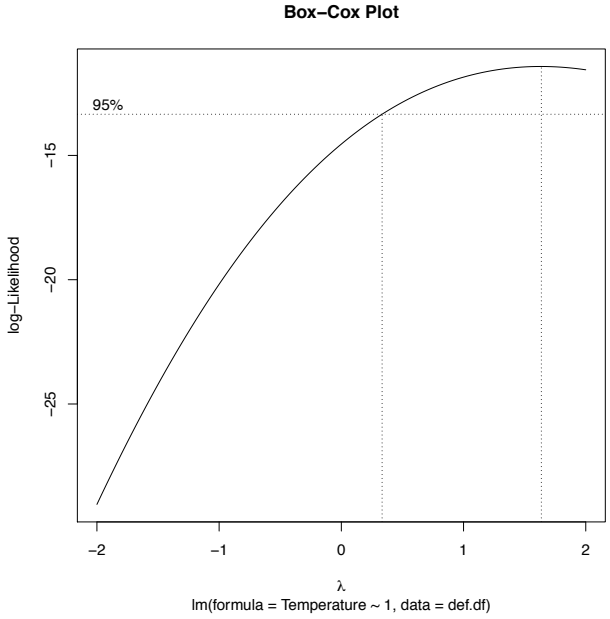

Residual Plot
Standardised Residuals
Rate
lm(formula = Defective ~ ., data = def.df)

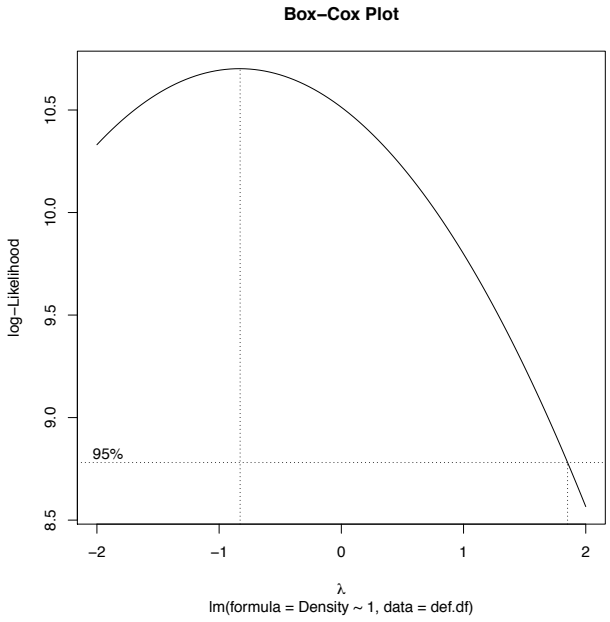

Residual Vs Previous Residual
Residual
Previous
lm(formula = Defective ~ ., data = def.df)

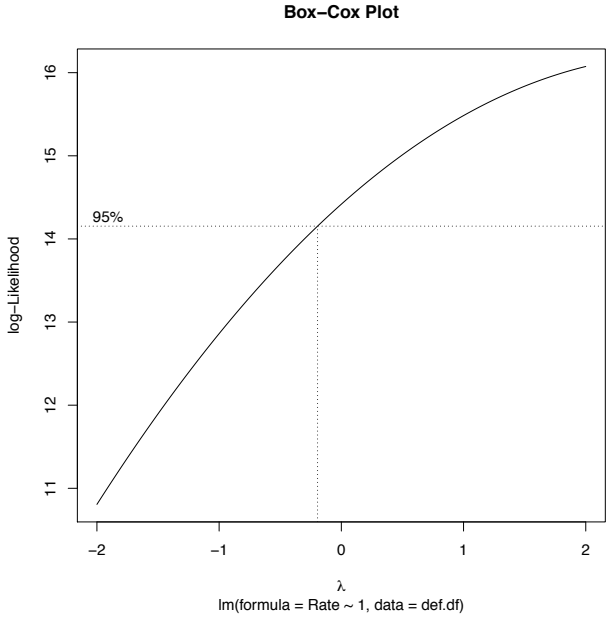

Residual Autocorrelation
ACF
Lag
lm(formula = Defective ~ ., data = def.df)

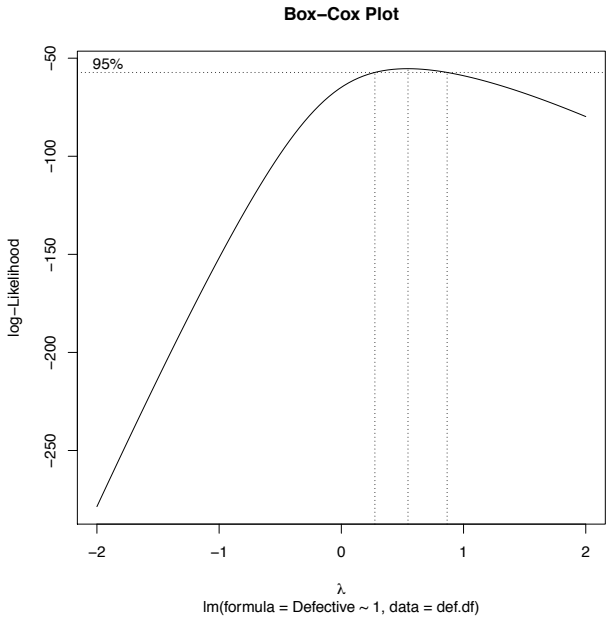

Normal Q–Q Plot
Sample Quantiles
Theoretical Quantiles
lm(formula = Defective ~ ., data = def.df)


Box–Cox Plot
95%
log–Likelihood
λ
lm(formula = Defective ~ ., data = def.df)


Box–Cox Plot
95%
log–Likelihood
−15
−20
−25
−2
−1
0
1
2
λ
lm(formula = Temperature ~ 1, data = def.df)


Box–Cox Plot
log–Likelihood
10.5
10.0
9.5
9.0
8.5
95%
−2
−1
0
1
2
λ
lm(formula = Density ~ 1, data = def.df)





Q: We don't have any serious issue, but what do those Box-Cox plots say?

- Therefore, we consider the following model instead

```
> def.bc.y.LM=lm(2*sqrt(Defective)-2~.,data=def.df)
```

- Running the residual analysis again reveals no evidence of any violation, thus

```
> summary(def.bc.y.LM)
```

```
Call:
lm(formula = 2 * sqrt(Defective) - 2 ~ ., data = def.df)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2029 -0.5700 -0.1543  0.6828  2.2790

Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept)  9.18594   10.52802   0.873   0.3909
Temperature  3.13033    1.32451   2.363   0.0259 *
Density     -0.58333    0.23907  -2.440   0.0218 *
Rate         0.02580    0.02086   1.237   0.2273
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 1.135 on 26 degrees of freedom
Multiple R-squared:  0.943,     Adjusted R-squared:  0.9365
F-statistic: 143.5 on 3 and 26 DF,  p-value: 2.713e-16
```

```
> summary(def.bc.y.no.rate.LM)
```

```
Call:
lm(formula = 2 * sqrt(Defective) - 2 ~ . - Rate, data = def.df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.96888 -0.86248 -0.02937  0.57578  2.58125

Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept)  17.1226     8.4271   2.032   0.0521 .
Temperature   3.5661     1.2892   2.766   0.0101 *
Density      -0.6939     0.2239  -3.099   0.0045 **
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 1.147 on 27 degrees of freedom
Multiple R-squared:  0.9397,    Adjusted R-squared:  0.9352
F-statistic: 210.3 on 2 and 27 DF,  p-value: < 2.2e-16
```

Q: Given the above model also satisfies all assumptions, should we retain `Rate`?

- Suppose there is actually one more variable collected by the study

`am_pm`	Morning is indicated by 1, and Afternoon by 0.

Q: How can we interpret each of the following parameters?

$$\sqrt{\texttt{Defective}} = \beta_0 + \beta_1 \texttt{Temperature} + \beta_2 \texttt{Density} + \beta_3 \texttt{Rate} + \beta_4 \texttt{am\_pm} + \varepsilon$$

Q: How can we interpret β_5^*?

$$\sqrt{\texttt{Defective}} = \beta_0^* + \beta_1^* \texttt{Temperature} + \beta_2^* \texttt{Density} + \beta_3^* \texttt{Rate} + \beta_4^* \texttt{am\_pm} + \beta_5^* \texttt{am\_pm*Temperature} + \varepsilon$$

- Suppose there is another categorical variable collected by the study

`location`	CHN, UK and USA

- Let us just investigate the two categorical variables,

```
> def.interaction.LM =
+   lm(Defective~am_pm*location, data = def.df)
```

Q: What do you think the following coefficients represent?

```
> summary(def.interaction.LM)
```

```
Call:
lm(formula = Defective ~ am_pm * location, data = def.df)

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           33.750      9.892   3.412  0.00229 **
am_pmPM               -5.467     12.771  -0.428  0.67243
locationUK            -6.710     13.272  -0.506  0.61777
locationUSA          -14.517     12.771  -1.137  0.26689
am_pmPM:locationUK    -1.973     17.879  -0.110  0.91303
am_pmPM:locationUSA   26.483     18.061   1.466  0.15554
```

Q: Can you construct a two-way table for the 6 categories from the output?

- The interaction is the difference between column/row differences

$$(\mu_{ij} - \mu_{1j}) - (\mu_{i1} - \mu_{11}) = \mu_{ij} - \mu_{i1} - \mu_{1j} + \mu_{11}$$

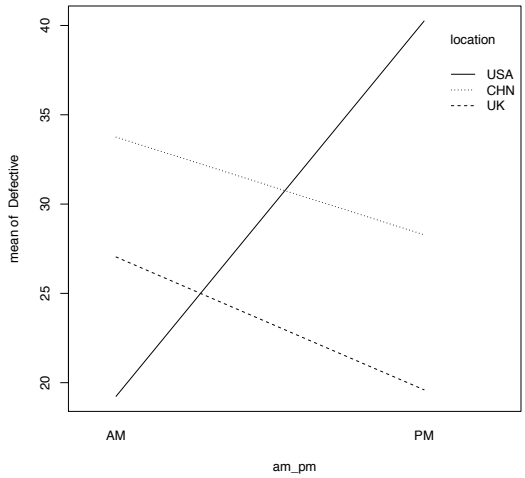
Q: Can you see the R summary output naturally decomposes the mean μ_{ij}?

	CHN	UK	USA
AM	$\mu_{11} = 33.750$	$\mu_{12} = 27.040$	$\mu_{13} = 19.233$
FM	$\mu_{21} = 28.283$	$\mu_{22} = 19.600$	$\mu_{23} = 40.249$

The conditional mean can be decomposed in to the following

$$\mu_{ij} = \mu_{11} + \underbrace{(\mu_{i1} - \mu_{11})}_{\text{row mean effect}} + \underbrace{(\mu_{1j} - \mu_{11})}_{\text{col mean effect}} + \underbrace{(\mu_{ij} - \mu_{i1} - \mu_{1j} + \mu_{11})}_{\text{interaction}}$$

```
> # A graphical representation of the two-way table
> with(def.df, interaction.plot(
+   am_pm, location, Defective))
```



- Of course, the full model now will include all 5 regressors

`Temperature`	temperature standard deviation during production
`Density`	density of the final product
`Rate`	rate of production
`Defective`	average number of defects per 1000 produced
`am_pm`	Morning is indicated by 1, and Afternoon by 0.
`location`	CHN, UK and USA

- In R, this can be specified as the following

```
> def.full.LM = lm(Defective~am_pm*location*Rate
+                    +am_pm*location*Density
+                    +am_pm*location*Temperature,
+                    data = def.df)
```

Q: What does the above model correspond to?

This model is the model that has different slopes and different intercepts for each one of the 6 categories.

3.4 Multicollinearity

- Let us go back to simple linear regression for a moment,

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

- Recall the standard error of $\hat{\beta}_1$ is given by

$$\text{Var}\left[\hat{\beta}_1 \mid x_1, x_2, \ldots, x_n\right] = \frac{\sigma^2}{(n-1)s_x^2}$$

$$\implies \text{SE}\left(\hat{\beta}_1\right) = \frac{\sigma}{\sqrt{(n-1)s_x^2}} = \frac{\sigma}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{1/2}}$$

from which we see the "accuracy" of our model is partly determined by the amount of scatter about the true regression line, measured by σ, but also by

"configuration"

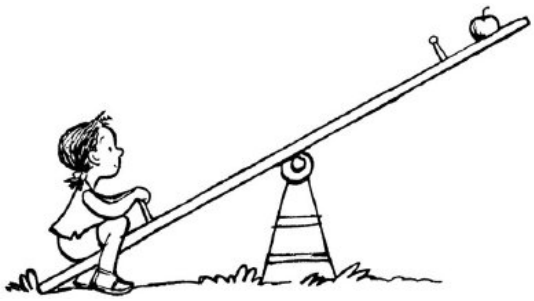
of observed x_i, that is, the spread of the observed x_i.

- If the x_i values are spread out, the estimated regression line is well

"supported"

and changes a little under resampling.

- On the other hand, if x_i's are bunched up, then it is not well supported,



and the regression line is "unstable" very much like a seesaw.

- Consider the following simulation to see the stability problem

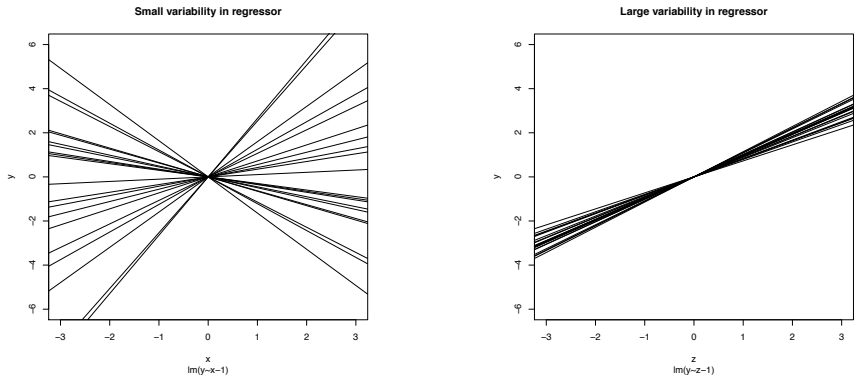
```
> set.seed(1)
> x.vec = rnorm(10, 0, 1)
> z.vec = 10*x.vec

> tmp = seq(-3, 3, length.out = 10)
> plot(tmp, 2*tmp, type = "n")

> replicate(20, {
+     y.vec = rnorm(10, mean = x.vec, sd = 3)
+     abline(lm(y.vec~x.vec-1))
+     })

> plot(tmp, 2*tmp, type = "n")
>
> replicate(20, {
+     y.vec = rnorm(10, mean = z.vec, sd = 3)
+     abline(lm(y.vec~z.vec-1))
+   })
```

- On the left, we have $\sum_{i=1}^{10}(x_i - \bar{x})^2 = 60.9$, and 20 simulated sets of $\{y_i\}$,



while $\sum_{i=1}^{10} (z_i - \bar{z})^2 = 0.609$ on the right.

Q: What happens the stability problem when having two predictors X_1 and X_2?

- Intuitively, the spread are still important in the stability of the regression,

$$\sum_{i=1}^{n} (x_{i1} - \bar{x}_1)^2 \qquad \text{and} \qquad \sum_{i=1}^{n} (x_{i2} - \bar{x}_2)^2$$

Q: Can you see why the correlation between X_1 and X_2 also matters?

- If there is a strong linear relationship between X_1 and X_2, we tends to have

"a knife edge"

support for the regression plane, which is also not stable intuitively.

- On the other hand, if the predictors are uncorrelated (or orthogonal), then they will be well spread out and support the fitted plane better.
- This intuition can be confirmed by explicitly working out

$$\text{Var}\left[\hat{\beta}_j \mid \mathbf{X}\right] \qquad \text{for } j = 1, 2$$

- When having only two predictors X_1 and X_2,

$$y = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{e}$$

the variance of the estimator β_j can be shown to take the following form

$$\text{Var}\left[\hat{\beta}_j \mid \mathbf{X}\right] = \frac{1}{1 - r^2} \cdot \frac{\sigma^2}{(n-1)s_{x_j}^2} \qquad \text{for } j = 1, 2$$

where r is the correlation coefficient between the two predictors, and

$$s_{x_j}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_{ij})^2$$

denotes the sample variance of X_j as usual.

- The term $\dfrac{1}{1 - r^2}$ in the variance can be used to detect the stability problem.

Q: Can you see that this term is a ratio between two related variances?

- In general,

$$y = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_k x_k + \hat{e}$$

the variance can be shown to take the following form

$$\text{Var}\left[\hat{\beta}_j \mid \mathbf{X}\right] = \frac{1}{1 - R_j^2} \cdot \frac{\sigma^2}{(n-1)s_{x_j}^2} \qquad \text{for } j = 1, 2, \ldots, k$$

where R_j^2 is the Multiple R-squared obtained from the regression

$$x_j = \hat{\gamma}_0 + \sum_{\ell \neq j} \hat{\gamma}_\ell x_\ell + \hat{\nu}$$

- The term $\dfrac{1}{1-R_j^2}$, which is a generalisation of $\dfrac{1}{1-r^2}$, is used to detect the stability problem, and is known as the jth variance inflation factor (VIF).

- To fully understand the stability problem, consider the data matrix

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1k} \\ 1 & x_{21} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_k \end{bmatrix}$$

- Multicollinearity occurs when the columns of the data matrix are almost

linearly dependent

that is, for $\alpha_i \in \mathbb{R}$ that are not simultaneously zero, the following is true

$$\alpha_0 \mathbf{1} + \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots \alpha_k \mathbf{x}_k = \mathbf{0}$$

- In the presence of numerical errors, it can occur if

1. One or more of the predictors has/have very little variation.
2. One or more of the predictors has/have very large mean.
3. Two or more of the predictors have a linear relationship.

- We can think of multicollinearity caused by the first two ways as inessential,

1. One or more of the predictor variables has/have very little variation.
2. One or more of the predictor variables has/have very large mean.

since we can remove it by standardising the data

$$x_{ij}^c = \frac{x_{ij} - \bar{x}_j}{\left(\dfrac{1}{n-1} \displaystyle\sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2 \right)^{1/2}} \quad \text{or} \quad x_{ij}^c = \frac{x_{ij} - \bar{x}_j}{\left(\displaystyle\sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2 \right)^{1/2}}$$

- The "Essential" multicollinearity is what remains after standardisation, and is caused by almost linear relationships between the predictor variables.

$$\alpha_0 \mathbf{1} + \alpha_1 \mathbf{x}_1^c + \alpha_2 \mathbf{x}_2^c + \cdots \alpha_k \mathbf{x}_k^c = \mathbf{0}$$

Q: How can we detect multicollinearity and identify the source of the problem?

$$\text{Var}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right] = \sigma^2 \left(\mathbf{X}^{\mathrm{T}} \mathbf{X}\right)^{-1}$$

- Eigenvalue and eigenvector problem

$$\mathbf{A}_{n\times n}\mathbf{x} = \lambda\mathbf{x} \qquad \text{where} \quad \mathbf{x} \neq \mathbf{0}$$

- Recall if $\mathbf{A}$ possesses n linearly independent eigenvectors, then

$$\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$$

- It can be shown if $\mathbf{A}$ is a normal, that is

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{A}\mathbf{A}^{\mathrm{T}}$$

then the matrix can be decomposed as

$$\mathbf{A} = \mathbf{Q}\mathbf{D}\mathbf{Q}^{\mathrm{T}} \qquad \text{where} \qquad \mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I} = \mathbf{Q}\mathbf{Q}^{\mathrm{T}}$$

- It is clear that $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is normal, thus

$$\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{Q}\mathbf{D}\mathbf{Q}^{\mathrm{T}}$$

- Now consider the inverse

$$\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} = \left(\mathbf{Q}\mathbf{D}\mathbf{Q}^{\mathrm{T}}\right)^{-1} = \mathbf{Q}\mathbf{D}^{-1}\mathbf{Q}^{\mathrm{T}} =$$

- Therefore, the jth diagonal element of $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$ is $\sum_{\ell=1}^{k+1} q_{j\ell}^2/\lambda_\ell$ and

$$\mathrm{Var}\left[\hat{\beta}_j \mid \mathbf{X}\right] = \sigma^2 \sum_{\ell=1}^{k+1} q_{j\ell}^2/\lambda_\ell \qquad \text{where} \quad \sum_{\ell=1}^{k+1} q_{j\ell}^2 = 1$$

which means it can only be usually big if one of λ_ℓ is usually small.

- To identify which the source of the problem, consider

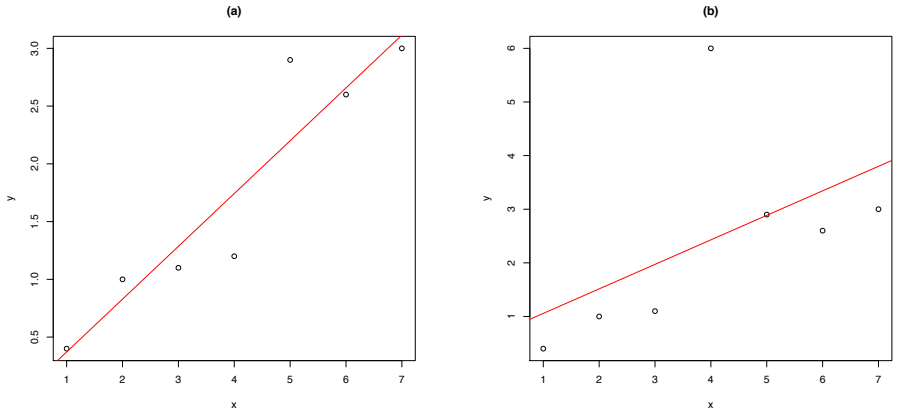
$$\begin{aligned}
\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{q}_\ell &= \lambda_\ell\mathbf{q}_\ell \\
\mathbf{q}_\ell^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{q}_\ell &= \mathbf{q}_\ell^{\mathrm{T}}\lambda_\ell\mathbf{q}_\ell \\
\left(\mathbf{X}\mathbf{q}_\ell\right)^{\mathrm{T}}\mathbf{X}\mathbf{q}_\ell &= \lambda_\ell\mathbf{q}_\ell^{\mathrm{T}}\mathbf{q}_\ell \\
\left\|\mathbf{X}\mathbf{q}_\ell\right\|^2 &= \lambda_\ell \approx 0
\end{aligned}$$

which means the eigenvector corresponding to a very small eigenvalue tell us which variables are almost linearly dependent.

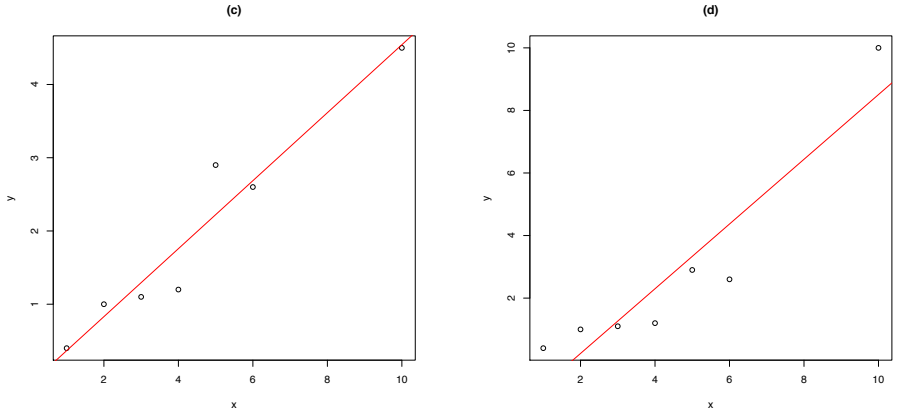
3.5 Unusual points

- Outliers are points with extreme response values $y_i \mid x_i$, so possibly large $\hat{e}_i$.
- High leverage points are points with extreme x_{ij}-values relative to others.

Q: Do we have any outlier or high leverage point in the following?



- Notice the difference between the following two cases.

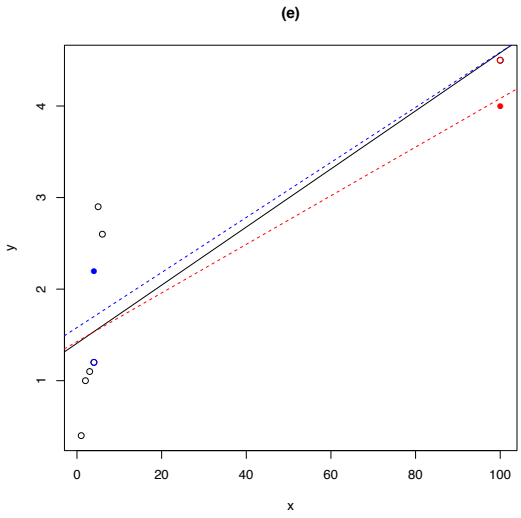


Thus no large residuals.

- High leverage points do not necessarily have large residuals, so that it is occasionally difficult to recognise them from a residual plot.

- Outliers need to be discussed because outliers can distort the regression.

Q: Why do we care about high leverage points? Consider the following



- Having different response values for high leverage points

$$y_i$$

will alter the regression surface more in comparison to low leverage points.

- This observation leads to a common detection and quantification method.
- Notice the fitted values are always on the regression surface,

$$\hat{y}_i$$

so if they change values, regression surface will change.

- Recall we derived the following when we started multiple regression

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \mathbf{P}\mathbf{y}$$

where $\mathbf{P}$ is known as the project or hat matrix.

- The leverage score is defined as the partial derivative

$$\frac{\partial \hat{y}_i}{\partial y_i} = [\mathbf{P}]_{ii} = p_{ii} \qquad \text{since} \qquad \hat{y}_i = \sum_{j=1}^{n} p_{ij} y_j$$

- The leverage score is entirely depended on the design matrix $\mathbf{X}$, not on $\mathbf{y}$.
- For the model having one predictor x_i and no intercept, it can be shown

$$\frac{\partial \hat{y}_i}{\partial y_i} = p_{ii} = \frac{x_i^2}{\sum_{j=1}^{n} x_j^2}$$

- When we add back the intercept, we have

$$\frac{\partial \hat{y}_i}{\partial y_i} = p_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2}$$

- Recall residuals have the following variance formula

$$\text{Var}\left[\hat{e}_i \mid \mathbf{X}\right] = (1 - p_{ii})\sigma^2$$

  based on which we use the following

$$\hat{e}'_i = \frac{\hat{e}_i}{\hat{\sigma}\sqrt{1 - p_{ii}}}$$

  is known as the Internally studentised residual/standardised residual.

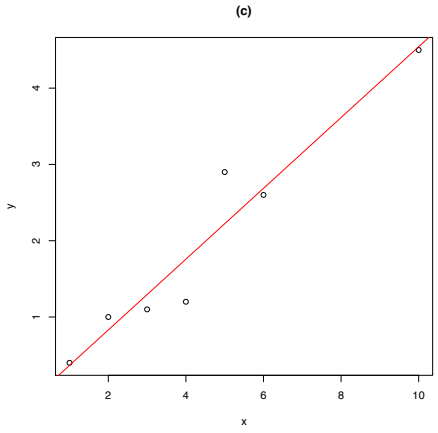
- The implication is that high leverage points tend to have smaller variances.
- Externally studentised residuals/studentised residual is defined as

$$\hat{e}^*_i = \frac{\hat{e}_i}{\hat{\sigma}(-i)\sqrt{1 - p_{ii}}}$$

  which offers some protection from the case that ith point is an outlier.

Q: Can you figure out why p_{ii} is between 0 and 1?

- Intuitively, the leverage score p_{ii} measures the "the amount of support" that the ith data point provided to the regression surface.
- A large leverage score for the ith point means the ith point is high leverage, and regression surface alters significantly if y_i takes a different value.
- A high leverage point is not necessarily influential, or an influential point



- An influential point is a point whose deletion would significantly alter the regression surface. There are several detection or quantification methods:

1. Standardised difference in coefficients

$$\frac{\hat{\beta}_j - \hat{\beta}_j(-i)}{\text{SE}\left(\hat{\beta}_j\right)}$$

where $\hat{\beta}_j(-i)$ is the estimate of β_j after the ith data point has been deleted.

2. Standardised difference in fitted values

$$\frac{\hat{Y}_i - \hat{Y}_i(-i)}{\text{SE}\left(\hat{Y}_i\right)}$$

- The standard errors are based on an estimate of σ without the ith data.

3. Cook's distance, D_i, is based on the idea of confidence ellipsoid

$$\left\{ \boldsymbol{\beta}\colon \frac{\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)^{\mathrm{T}} \mathbf{X}^{\mathrm{T}}\mathbf{X} \left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)}{(k+1)\hat{\sigma}^2} \leq F_{k+1,n-k-1}(\alpha) \right\}$$

which gives $100 \cdot (1-\alpha)\%$ confidence ellipsoid for $\boldsymbol{\beta}$.

- The idea is to define

$$D_i = \frac{\left(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}(-i)\right)^{\mathrm{T}} \mathbf{X}^{\mathrm{T}}\mathbf{X} \left(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}(-i)\right)}{(k+1)\hat{\sigma}^2}$$

to quantify the whether $\hat{\boldsymbol{\beta}}(-i)$ is essentially the same as $\hat{\boldsymbol{\beta}}$ by comparing

$$D_i \qquad \text{with} \qquad F_{k+1,n-k-1}(\alpha)$$

3.6 Case Study

Yield Study of a chemical process

- The following data were gathered in the course of studying the yield of a particular chemical process.

`yield`	measures the effectiveness of a synthetic procedure
`conversion`	a percentage of desired over undesired product(s)
`flow`	measures the speed of the reaction process
`ratio`	a ratio of between two chemicals

- Chemical theory predicts `yield` is related to the reciprocal of `ratio`.
- Theory also predicts `yield` is related to `conversion*flow`.
- We are interested in the relation between the yield and the other variables.
- In particular, we would like to see whether the data support the two theories.

- In practice, after loading the data

```
> chem_pro.df = read.table("~/Desktop/chem_pro.csv",
+                          sep = ",", header = TRUE)
```

we should always check the data frame for issues

```
> str(chem_pro.df)
```

```
'data.frame':   44 obs. of  4 variables:
 $ yield     : num  55.5 54.8 52.2 50.4 49.3 ...
 $ conversion: num  11.8 11.9 12.1 12.1 12 ...
 $ flow      : num  119 105 97 101 44 ...
 $ ratio     : Factor w/ 40 levels "0.036","0.089",..: 16 2 3 6 5 1 7 9 11 24 ...
```

- In this case, somehow `ratio` is a `Factor`,

```
> class(chem_pro.df$ratio)
```

```
[1] "factor"
```

- It is often a result of typos or having inconsistent data format in the data file.
- We look into the values that this factor variable can take

```
> levels(chem_pro.df$ratio)
```

```
 [1] "0.036" "0.089" "0.094" "0.097"
 [5] "0.1"   "0.108" "0.113" "0.116"
 [9] "0.123" "0.126" "0.135" "0.136"
[13] "0.143" "0.152" "0.153" "0.155"
[17] "0.16"  "0.161" "0.164" "0.166"
[21] "0.169" "0.17"  "0.18"  "0.183"
[25] "0.184" "0.188" "0.192" "0.194"
[29] "0.195" "0.197" "0.201" "0.211"
[33] "0.215" "0.221" "0.222" "0.223"
[37] "0.225" "0.229" "0.233" "0>163"
```

- Identifying the observation, which is clearly a typo,

```
> ratio_typo = which(chem_pro.df$ratio == "0>163")
> ratio_typo
```

```
[1] 32
```

- Correcting the typo by first converting it into a character variable

```
> chem_pro.df$ratio = as.character(chem_pro.df$ratio)
```

- Reassigning the value `0.163` instead of `"0>163"`

```
> chem_pro.df$ratio[ratio_typo] = "0.163"
```

- Coercing it into a double precision numeric variable

```
> chem_pro.df$ratio = as.double(chem_pro.df$ratio)
> chem_pro.df$ratio
```

```
 [1] 0.155 0.089 0.094 0.108 0.100 0.036 0.113
 [8] 0.123 0.135 0.183 0.166 0.221 0.192 0.188
[15] 0.201 0.153 0.194 0.097 0.136 0.143 0.116
[22] 0.195 0.160 0.164 0.197 0.233 0.211 0.222
[29] 0.223 0.229 0.170 0.163 0.153 0.180 0.126
[36] 0.152 0.184 0.225 0.169 0.161 0.197 0.201
[43] 0.221 0.215
```

Q: Is there any more obvious error in the dataset?

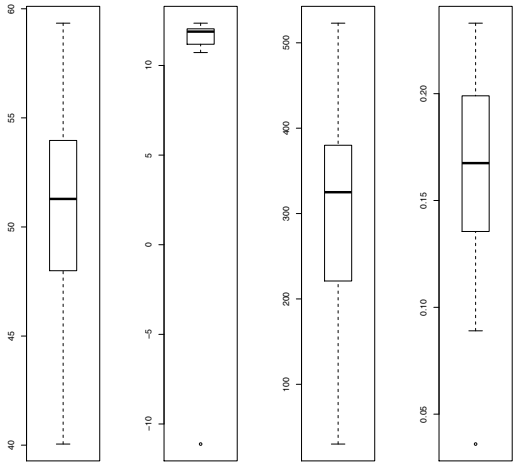
```
> summary(chem_pro.df)
```

```
     yield          conversion          flow           ratio        
 Min.   :40.05   Min.   :-11.12   Min.   : 30.0   Min.   :0.0360  
 1st Qu.:48.05   1st Qu.: 11.19   1st Qu.:221.5   1st Qu.:0.1358  
 Median :51.28   Median : 11.89   Median :325.0   Median :0.1675  
 Mean   :50.68   Mean   : 11.11   Mean   :291.9   Mean   :0.1658  
 3rd Qu.:53.91   3rd Qu.: 12.04   3rd Qu.:380.0   3rd Qu.:0.1980  
 Max.   :59.34   Max.   : 12.36   Max.   :523.0   Max.   :0.2330  
```

- Boxplot as well as summary is particularly useful to identify unusual values

```
> par(mfrow = c(1,4))
> lapply(chem_pro.df, boxplot) # do for every column
> par(mfrow = c(1,1))
```

- We should take a note and report later when there is any suspicion of error in the data, since every data point matters when there are only 44 of them.
- This procedure of correcting obvious typos and reporting potential errors in the dataset is often known as data cleaning.



- There seems to be 1 unusually small value in variable 2 and one in variable 4.
- Identifying both of them

```
> names(chem_pro.df)
```

```
[1] "yield"      "conversion" "flow"       "ratio"
```

```
> conversion_typo = which(
+   chem_pro.df$conversion <= -10)
>
> ratio_unusual = which(
+   chem_pro.df$ratio <= 0.05)
>
> conversion_typo; ratio_unusual
```

```
[1] 44
[1] 6
```

- From the description of the data, `conversion` can only be between 0 and 1, but observation 6, the unusually small `ratio` value, might just be unusual.
- According to the summary,

```
> summary(chem_pro.df$conversion)
```

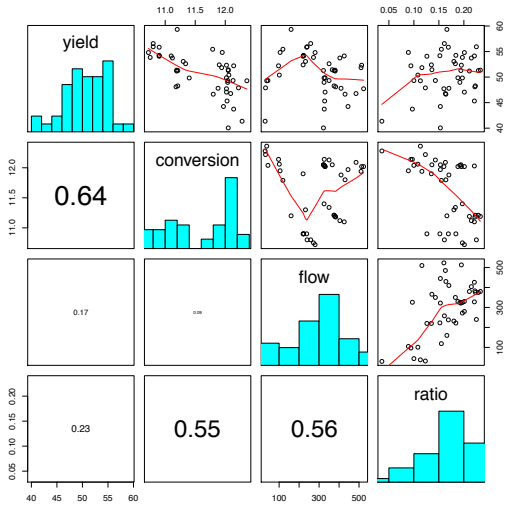
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-11.12   11.19   11.89   11.11   12.04   12.36
```

  it is likely that the minus sign is a typo.
- So we modify the 44th observation and keep record of it

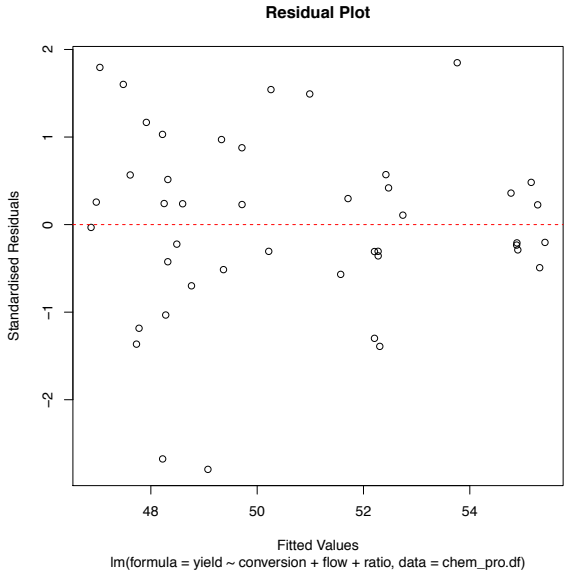
```
> chem_pro.df$conversion[conversion_typo] =
+   - chem_pro.df$conversion[conversion_typo]
```

- We have to keep observation 6 on watch list, it may be a high leverage point.
- Observations can be unusual in terms of relationships amongst variables, so plotting two variables at a time is useful, which is known as a pairs plot

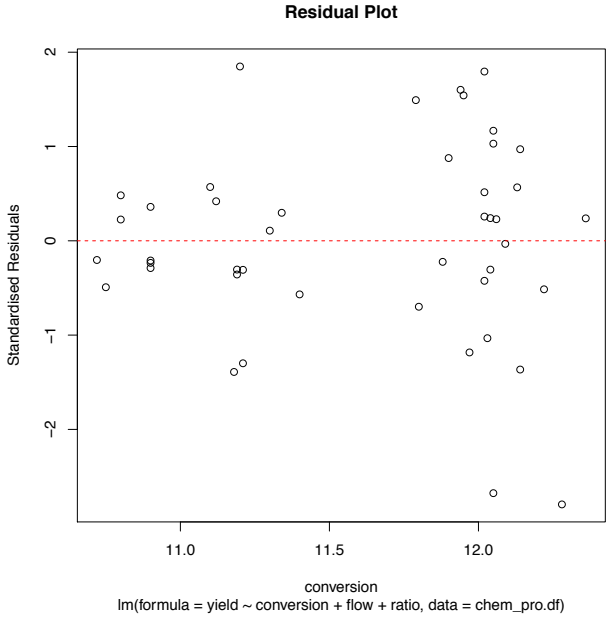
```
> ?pairs # See the help document for how to use it
```

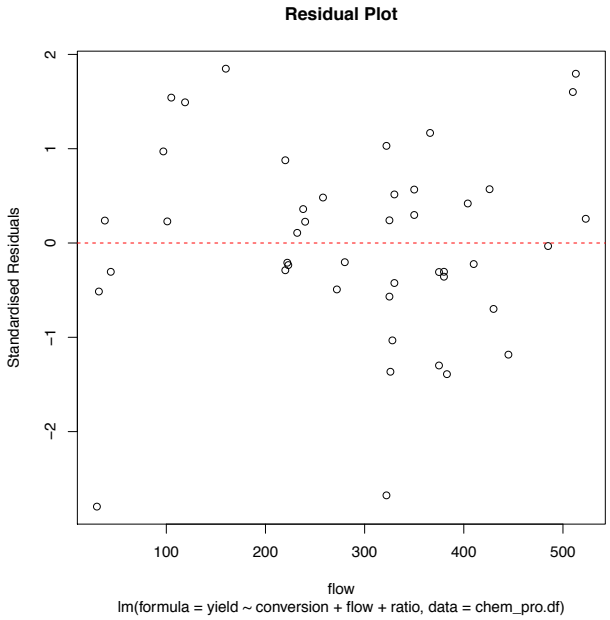


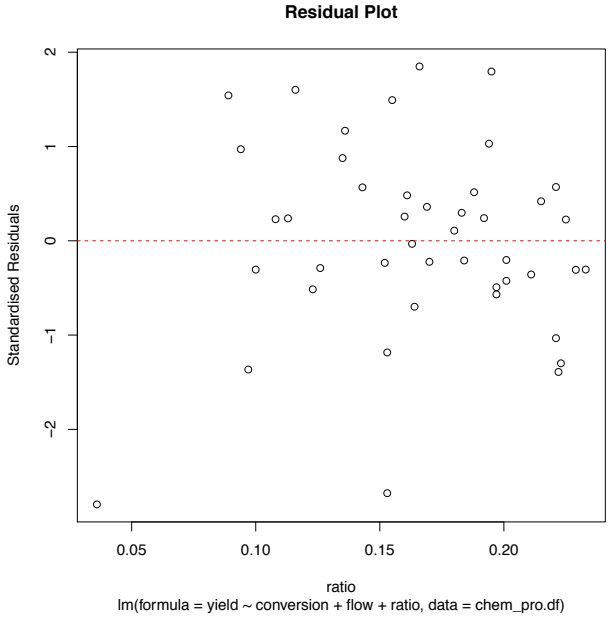
- No more unusual point other than the 44th, and 6th seems to be consistent.
- It is also useful for seeing the spread and detecting strong linear relationships.
- We have no evidence against the multiple linear regression model being valid.

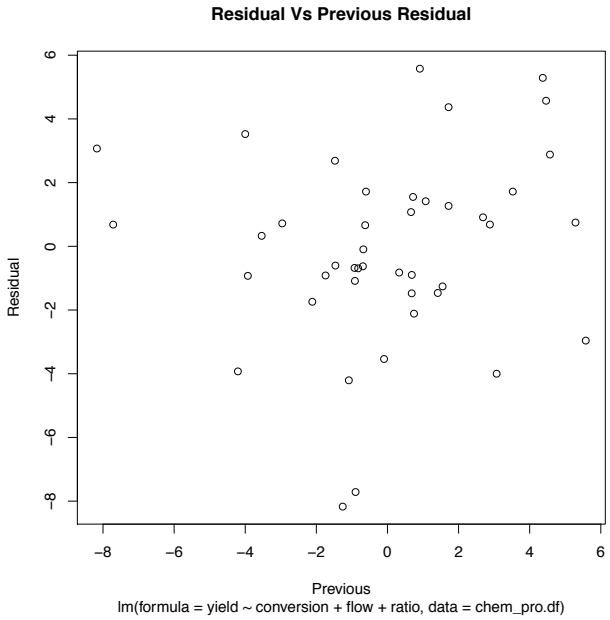


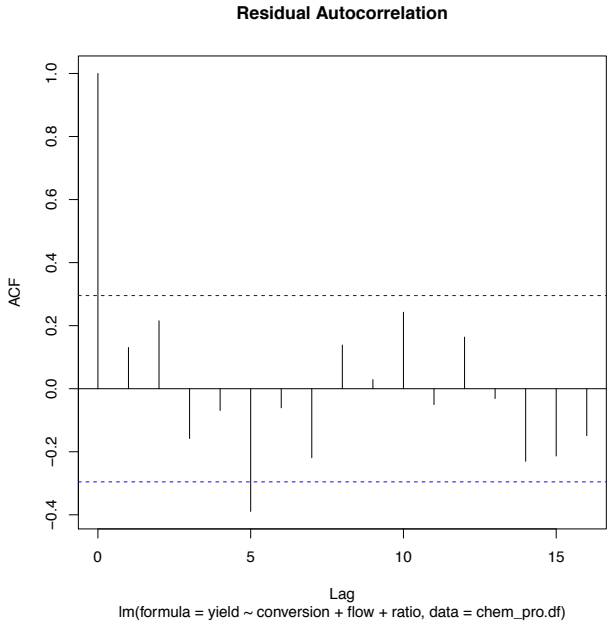
This simple model will be used as a yardstick for comparison if it is valid.

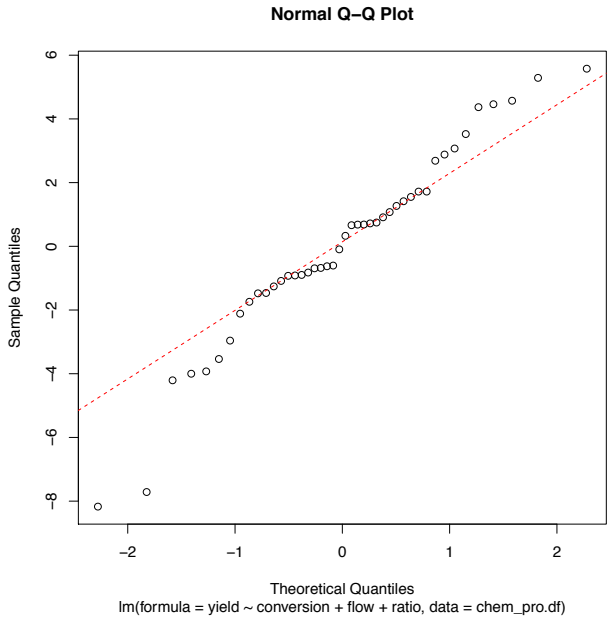

Residual Plot
Standardised Residuals
2
1
0
−1
−2
11.0
11.5
12.0
conversion
lm(formula = yield ~ conversion + flow + ratio, data = chem_pro.df)


Residual Plot
Standardised Residuals
2
1
0
−1
−2
100
200
300
400
500
flow
lm(formula = yield ~ conversion + flow + ratio, data = chem_pro.df)


Residual Plot
Standardised Residuals
2
1
0
−1
−2
0.05
0.10
0.15
0.20
ratio
lm(formula = yield ~ conversion + flow + ratio, data = chem_pro.df)


Residual Vs Previous Residual
Residual
6
4
2
0
−2
−4
−6
−8
−8
−6
−4
−2
0
2
4
6
Previous
lm(formula = yield ~ conversion + flow + ratio, data = chem_pro.df)





- The normal QQ plot is the only plot that deserves a second look since

```
> nrow(chem_pro.df)
```

```
[1] 44
```

```
> model
```

```
Call:
lm(formula = yield ~ conversion + flow + ratio, data = chem_pro.df)

Coefficients:
(Intercept)   conversion         flow        ratio
 113.011270    -5.189435    -0.006983    -0.055762
```

```
> res = residuals(model)
>
> shapiro.test(res)
```

```
        Shapiro-Wilk normality test

data:  res
W = 0.96238, p-value = 0.1597
```

Q: Given we have no evidence against the model being valid,

```
> summary(model)
```

```
Call:
lm(formula = yield ~ conversion + flow + ratio, data = chem_pro.df)

Residuals:
    Min      1Q  Median      3Q     Max
-8.1715 -1.3101  0.1171  1.5926  5.5769

Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept) 113.011270  14.631935   7.724 1.88e-09 ***
conversion   -5.189435   1.149651  -4.514 5.49e-05 ***
flow         -0.006983   0.004467  -1.563    0.126
ratio        -0.055762  15.755755  -0.004    0.997
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 3.118 on 40 degrees of freedom
Multiple R-squared:   0.46,     Adjusted R-squared:  0.4195
F-statistic: 11.36 on 3 and 40 DF,  p-value: 1.591e-05
```

what can we conclude from the above summary output?

- Of course, being a valid model doesn't mean it is the best model.
- One aspect of a model that is not addressed by its validity is the

  "stability"

  of the regression surface.
- This can be partly addressed by looking at the variance inflation factor (VIF)

$$\frac{1}{1-R_j^2}$$

- If it is big, e.g. greater than 10, then we will have to reconsider our model.

Q: What does it mean to have a VIF bigger than 10?

- It can be shown the jth VIF is just the jth diagonal element of

$$\mathbf{R}^{-1}$$

  where $\mathbf{R}$ is the sample correlation matrix for the k predictors.

- Recall the jth variance inflation factor is given by

$$\frac{1}{1 - R_j^2}$$

which is the first factor of

$$\text{Var}\left[\hat{\beta}_j \mid \mathbf{X}\right] = \frac{1}{1 - R_j^2} \frac{\sigma^2}{(n-1)s_{x_j}^2}$$

which is given by the jth diagonal element of the following

$$\text{Var}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right] = \sigma^2 \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$$

- Recall the sample Pearson's correlation is given by

$$\begin{aligned} r_{ij} &= \frac{\displaystyle\sum_{\ell=1}^{n} (x_{\ell i} - \bar{x}_i)(x_{\ell j} - \bar{x}_j)}{\sqrt{\displaystyle\sum_{\ell=1}^{n} (x_{\ell i} - \bar{x}_i)^2}\sqrt{\displaystyle\sum_{\ell=1}^{n} (x_{\ell j} - \bar{x}_j)^2}} \\ &= \frac{1}{n-1}\frac{\displaystyle\sum_{\ell=1}^{n} (x_{\ell i} - \bar{x}_i)(x_{\ell j} - \bar{x}_j)}{\sqrt{\dfrac{1}{n-1}\displaystyle\sum_{\ell=1}^{n} (x_{\ell i} - \bar{x}_i)^2}\sqrt{\dfrac{1}{n-1}\displaystyle\sum_{\ell=1}^{n} (x_{\ell j} - \bar{x}_j)^2}} \\ &= \frac{1}{n-1}\sum_{\ell=1}^{n} \frac{(x_{\ell i} - \bar{x}_i)}{s_i}\frac{(x_{\ell j} - \bar{x}_j)}{s_j} \end{aligned}$$

- In matrix notation, suppose the usual data matrix is partitioned such that

$$\mathbf{X} = \begin{bmatrix} \mathbf{1} & \mathbf{X}_{-1} \end{bmatrix}$$

then

$$\mathbf{R} = \frac{1}{n-1}\mathbf{X}_{cs}^{\mathrm{T}}\mathbf{X}_{cs}$$

where $\mathbf{X}_{cs} = \mathbf{C}\mathbf{X}_{-1}\mathbf{D}^{-1}$ with $\mathbf{C} = \mathbf{I} - \dfrac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}$ and $\mathbf{D} = \operatorname{diag}(s_1, s_2, \ldots, s_k)$.

- Instead of the original model,

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + e_i$$

consider $\alpha_0 = \beta_0 + \beta_1\bar{x}_1 + \beta_2\bar{x}_2 + \cdots + \beta_k\bar{x}_k$, then

$$\begin{aligned} y_i &= \alpha_0 + \beta_1(x_{i1} - \bar{x}_1) + \beta_2(x_{i2} - \bar{x}_2) + \cdots + \beta_k(x_{ik} - \bar{x}_k) + e_i \\ &= \alpha_0 + \beta_1\tilde{x}_{i1} + \beta_2\tilde{x}_{i2} + \cdots + \beta_k\tilde{x}_{ik} + e_i \end{aligned}$$

and scaling is done by considering the following

$$x_{ij}^* = \frac{x_{ij} - \bar{x}_j}{s_j} = \frac{\tilde{x}_{ij}}{s_j}$$

which means $\gamma_j = \beta_j s_j$ and $\hat{\gamma}_j = \hat{\beta}_j s_j$ and

$$y_i = \alpha_0 + \gamma_1 x_{i1}^* + \gamma_2 x_{i2}^* + \cdots + \gamma_k x_{ik}^* + e_i$$

- It can be proved by considering the relationship between

$$\operatorname{Var}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right] \qquad \text{and} \qquad \operatorname{Var}\left[\hat{\boldsymbol{\gamma}} \mid \mathbf{X}\right]$$

the later variance which can be written in terms of the correlation matrix $\mathbf{R}$.

- Thus the easiest way to obtain VIF is the following

```
> Xminus1 = chem_pro.df[, -1]
> VIF = diag(solve(cor(Xminus1)))
> VIF
```

```
conversion       flow      ratio
  1.580323   1.606860   2.276409
```

  which shows we have no multicollinearity problem.

- Unusual points also effect the "stability", so need to be checked/reported.
- It can be shown the sum of leverage scores

$$\frac{\partial \hat{y}_i}{\partial y_i} = p_{ii}, \qquad \text{where} \quad \mathbf{P} = \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}$$

  is given by by the rank of $\mathbf{X}$, i.e. it is the number of linearly independent columns in $\mathbf{X}$ when $n \geq k+1$, which means the sum is equal to $k+1$.

- Ideally we wish this "total leverage" of $k+1$ to be shared equally, that is,

$$\frac{k+1}{n}$$

  by the n observations, any particular observation having more than

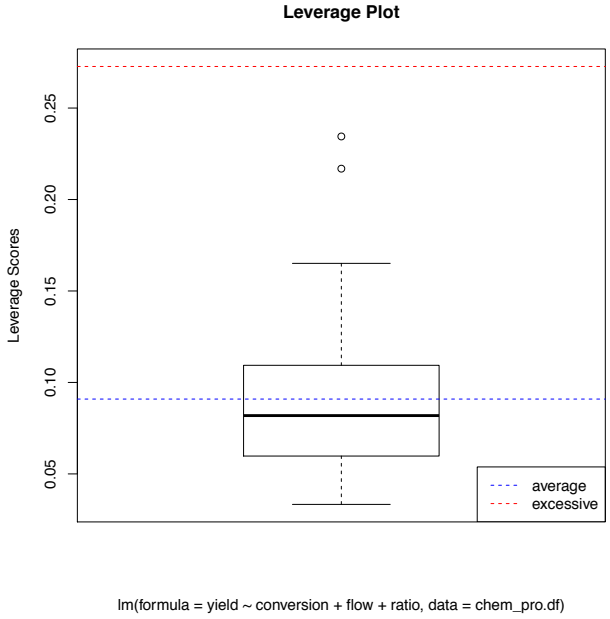
$$\frac{3(k+1)}{n}$$

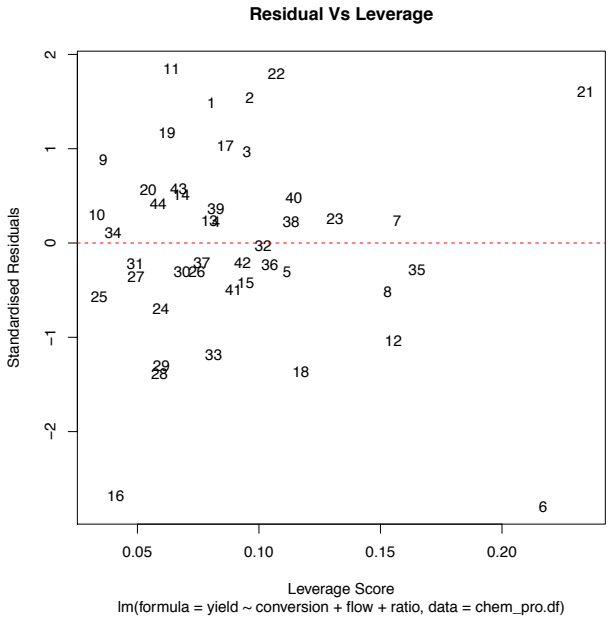
  is deemed to have "excessive leverage" and has the potential of exerting a very large influence over the position of the fitted regression surface.

- We need to report if there is any observation that has "excessive leverage", especially so if it is also an outlier, the regression plane is unstable due to it.

```
> pii.vec = hatvalues(model)
> order(pii.vec, decreasing = TRUE)
```

```
 [1] 21  6 35  7 12  8 23 18 40 38  5 22 36 32  2
[16]  3 15 42 41 17  4 39 33  1 13 37 26 30 14 43
[31] 11 19 29 24 28 44 20 27 31 16 34  9 25 10
```





- Another way to determine whether our model is excessively influenced by a point is to explicitly consider "leave-one-out" influence measures

```
> im = influence.measures(model)
> im # Report and possibly delete influential points
```

```
Influence measures of
         lm(formula = yield ~ conversion + flow + ratio, data = chem_pro.df) :

     dfb.1_  dfb.cnvr dfb.flow dfb.rati   dffit cov.r   cook.d    hat inf
1  -0.15379  0.171205 -0.37075  0.21414  0.4483 0.957 4.87e-02 0.0804
2   0.10348 -0.044898 -0.11423 -0.24833  0.5123 0.958 6.33e-02 0.0962
3  -0.02789  0.062869 -0.12017 -0.07752  0.3146 1.111 2.48e-02 0.0951
4  -0.01039  0.017236 -0.03453 -0.00626  0.0678 1.200 1.18e-03 0.0823
5   0.01094 -0.021314  0.06432  0.00689 -0.1070 1.234 2.93e-03 0.1116
6  -0.31180  0.115894  0.17695  0.97151 -1.6193 0.592 5.41e-01 0.2169   *
7  -0.05076  0.058095 -0.07330  0.03086  0.1015 1.305 2.64e-03 0.1569   *
8   0.10158 -0.115249  0.17135 -0.08036 -0.2165 1.272 1.19e-02 0.1530
:
35 -0.11592  0.110300 -0.03173  0.09279 -0.1270 1.314 4.13e-03 0.1651   *
36 -0.06936  0.067273 -0.00626  0.04110 -0.0790 1.229 1.60e-03 0.1045
37 -0.03747  0.038115  0.01600  0.00120 -0.0594 1.193 9.04e-04 0.0765
38  0.01881 -0.023149 -0.03880  0.03878  0.0797 1.242 1.63e-03 0.1133
39  0.08636 -0.085527 -0.00198 -0.03544  0.1066 1.190 2.91e-03 0.0823
40  0.15475 -0.152393  0.03017 -0.08905  0.1717 1.221 7.51e-03 0.1144
41 -0.10567  0.110095  0.01319  0.00863 -0.1529 1.186 5.96e-03 0.0896
42 -0.04383  0.046000  0.00491  0.00218 -0.0645 1.215 1.06e-03 0.0933
43  0.03744 -0.047960  0.04730  0.02774  0.1516 1.148 5.85e-03 0.0670
44  0.02885 -0.035479  0.02799  0.01741  0.1034 1.155 2.73e-03 0.0585
```

Q: Does the data support the idea that yield depends on the reciprocal of ratio?

```
> recip.LM = lm(yield~ conversion + flow +
+                   I(1/ratio), data = chem_pro.df)

> summary(recip.LM)
```

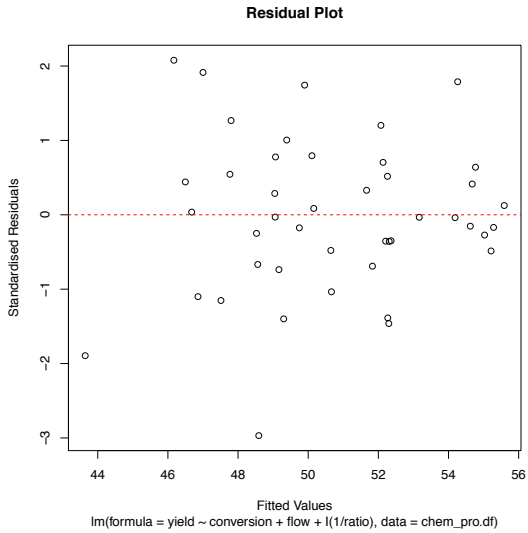
```
Call:
lm(formula = yield ~ conversion + flow + I(1/ratio), data = chem_pro.df)

Residuals:
    Min      1Q  Median      3Q     Max
-8.5413 -1.5016 -0.1017  1.6203  5.6798

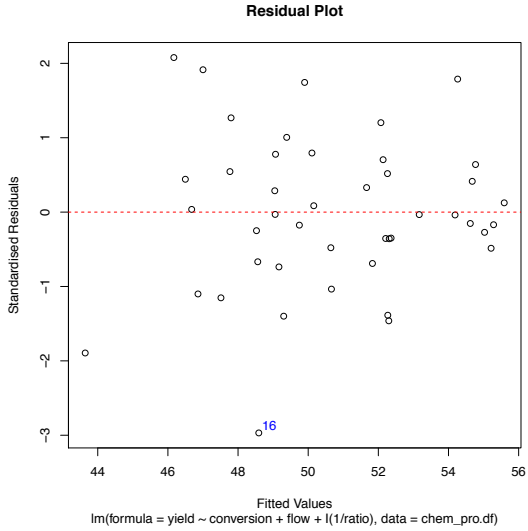
Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept) 105.935729  10.694444   9.906 2.54e-12 ***
conversion   -4.263058   0.961839  -4.432 7.08e-05 ***
flow         -0.011542   0.003921  -2.944  0.00537 **
I(1/ratio)   -0.345478   0.155987  -2.215  0.03253 *
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.943 on 40 degrees of freedom
Multiple R-squared:  0.519,     Adjusted R-squared:  0.4829
F-statistic: 14.39 on 3 and 40 DF,  p-value: 1.663e-06
```

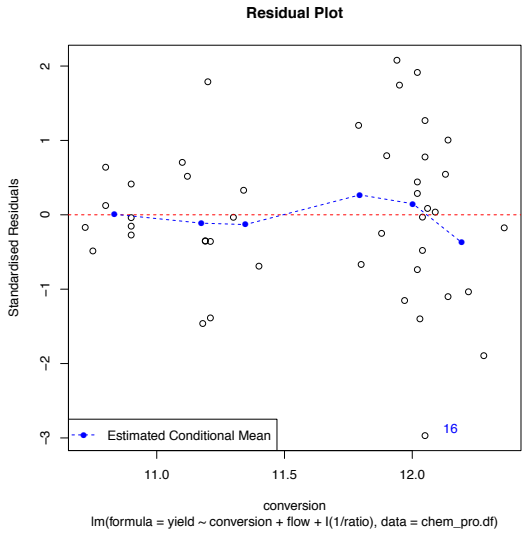
Q: Can we trust this model? Of course, we have to check the residuals first.



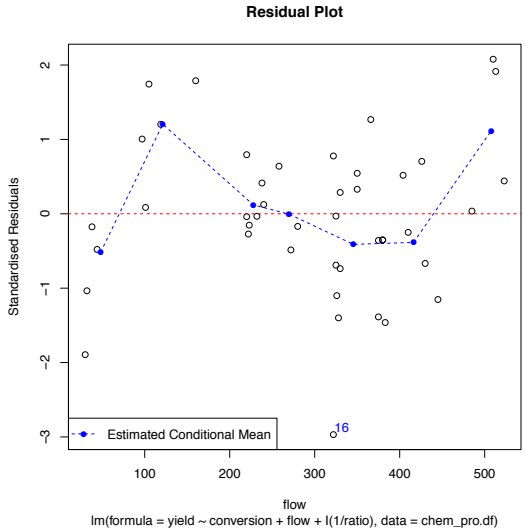

```
> recip_outlier_index = which( sres <  -2.5 )
```



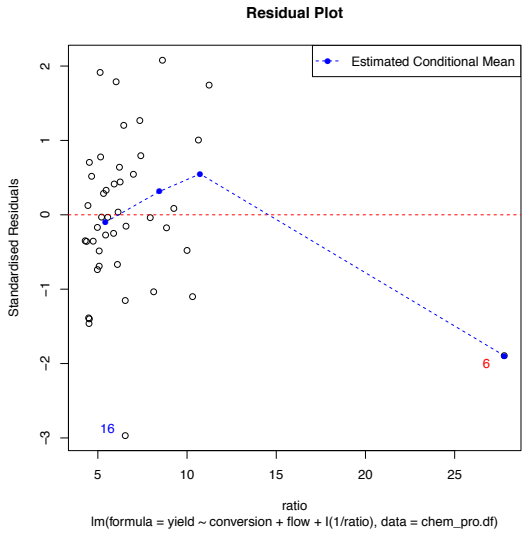
- Notice the outlier, but it seems no transformation is needed for `conversion`.



- It is a different story for `flow`, there seems to be a cubic relationship.



- The curvature is clearly due to the high leverage point.



- Identifying the high leverage point,

```
> recip_hl_index = which(1/chem_pro.df$ratio > 25)
> recip_hl_index
```

```
[1] 6
```

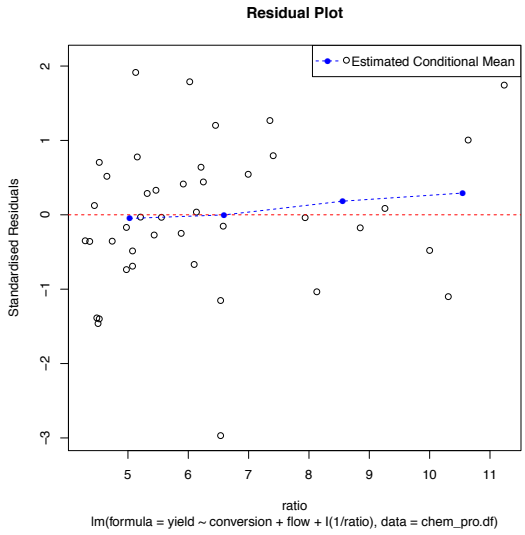
```
> ratio_unusual
```

```
[1] 6
```

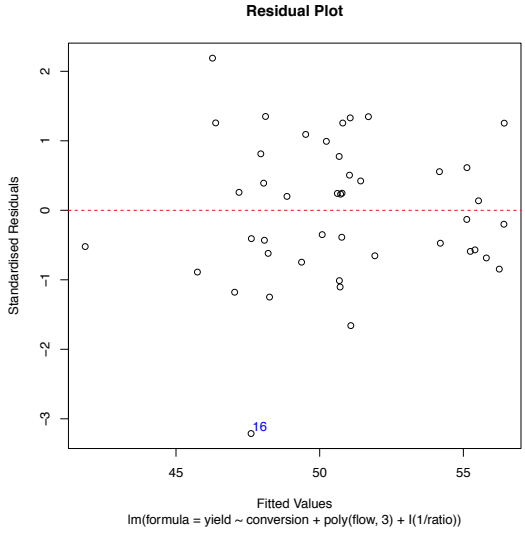
```
> chem_pro.df[ratio_unusual,]
```

```
  yield conversion flow ratio
6 41.36      12.28   30 0.036
```

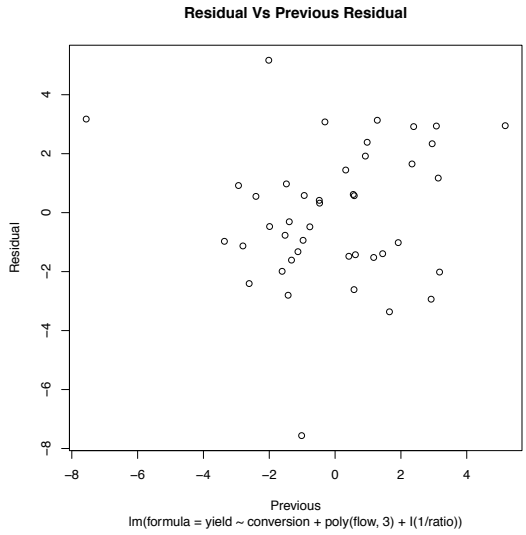
- We can see the effect this point on the trend line by leaving it out.
- No more transformation is need for `ratio`.



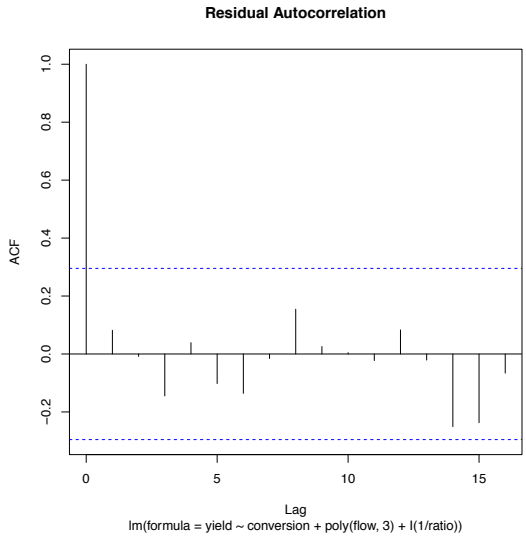

```
> cubic.LM = lm(yield~ conversion + poly(flow,3)
+                 + I(1/ratio), data = chem_pro.df)
```



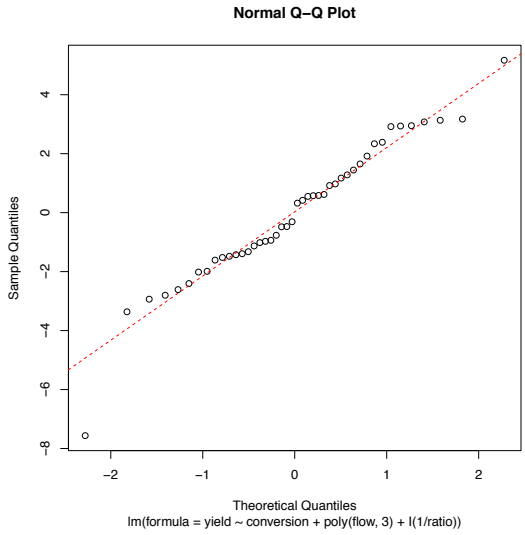
- Other than the outlier, it looks fine.



- We have no evidence against the independence assumption.



- Other than the outlier, QQ-normal plot seems to be fine.




```
> summary(cubic.LM)
```

```
Call:
lm(formula = yield ~ conversion + poly(flow, 3) + I(1/ratio),
    data = chem_pro.df)

Residuals:
   Min     1Q Median     3Q    Max
-7.562 -1.440  0.007  1.497  5.170

Coefficients:
                Estimate Std. Error t value Pr(>t)
(Intercept)     101.6345    10.4408   9.734 7.18e-12 ***
conversion       -4.1717     0.9105  -4.582 4.86e-05 ***
poly(flow, 3)1 -10.4209      3.0096  -3.462 0.001340 **
poly(flow, 3)2   3.7437      3.1710   1.181 0.245090
poly(flow, 3)3  10.0417      2.5762   3.898 0.000382 ***
I(1/ratio)      -0.3644      0.1404  -2.594 0.013386 *
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.469 on 38 degrees of freedom
Multiple R-squared:  0.6783,    Adjusted R-squared:  0.6359
F-statistic: 16.02 on 5 and 38 DF,  p-value: 1.742e-08
```

- This model is reasonably good, its explains 67.83% of variability in `yield`.
- The data support the claim that `yield` relates to `1/ratio` under this model.
- We press on and exam the other claim, that is, `yield` is related to

  `conversion*flow`

```
> prod.LM =
+   lm(yield~ conversion + poly(flow,3) +
+         I(1/ratio) + I(flow*conversion),
+       data = chem_pro.df)
```

- After examining the residuals, it seems all assumptions are satisfied, but

```
> summary(prod.LM)
```

```
Call:
lm(formula = yield ~ conversion + poly(flow, 3) + I(1/ratio) +
    I(flow * conversion), data = chem_pro.df)

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          99.42924   10.67628   9.313 3.08e-11 ***
conversion           -0.18663    4.11546  -0.045  0.96407
poly(flow, 3)1      127.62434  139.06123   0.918  0.36469
poly(flow, 3)2        1.73942    3.75947   0.463  0.64631
poly(flow, 3)3       13.30024    4.17243   3.188  0.00291 **
I(1/ratio)           -0.35397    0.14086  -2.513  0.01646 *
I(flow * conversion) -0.01305    0.01314  -0.993  0.32720
```

- Notice `conversion` and `flow` were highly significant

```
> conversion = chem_pro.df$conversion
>
> flow = poly(chem_pro.df$flow,3)
>
> recip_ratio = 1/chem_pro.df$ratio
>
> prod = chem_pro.df$flow * chem_pro.df$conversion
>
> Xtd = cbind(conversion, flow, recip_ratio, prod)
```

- Recall variance inflation factor of more than 10 is considered to be large,

```
> VIF = diag(solve(cor(Xtd)))
>
> VIF
```

```
conversion          1          2          3 recip_ratio       prod
  32.276768 3169.973656    2.316849    2.853786    1.932279 3132.705871
```

- Eigenvalues of $\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}$ confirms, we definitely have collinearity problem

```
> eigen.stuff = eigen(cor(Xtd))
>
> eigen.stuff$values
```

```
[1] 2.4837616083 1.7021510928 1.0013419291 0.4480934145 0.3644941045 0.0001578508
```

```
> round(eigen.stuff$vectors, 2)
```

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,]  0.28 -0.56  0.03  0.76  0.16  0.07
[2,] -0.55 -0.35  0.00 -0.17  0.20  0.71
[3,]  0.23 -0.57 -0.35 -0.36 -0.61 -0.01
[4,] -0.09  0.21 -0.94  0.18  0.20  0.02
[5,]  0.52 -0.19 -0.04 -0.48  0.68  0.00
[6,] -0.53 -0.40 -0.02 -0.08  0.24 -0.70
```

- It seems we have to go without `conversion*flow`.
- It seems that `conversion*flow` is unnecessary according to this dataset.

- We move back to the cubic model,

```
> cubic.LM = lm(yield~ conversion + poly(flow,3)
+                 + I(1/ratio), data = chem_pro.df)
```

and investigate the effect of unusual points on the model `cubic.LM`.

```
> model$call
```

```
lm(formula = yield ~ conversion + poly(flow, 3) + I(1/ratio), data = chem_pro.df)
```

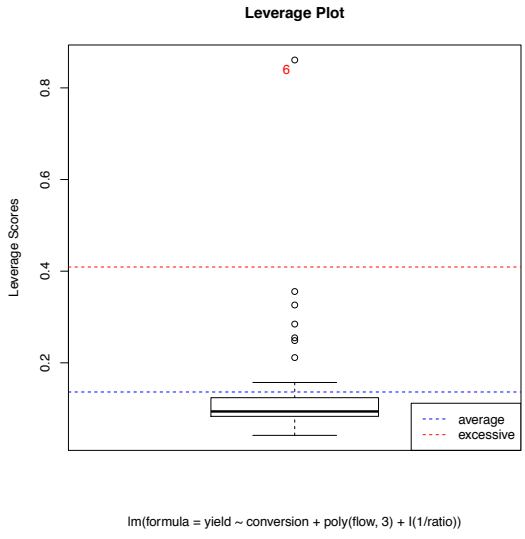
```
> pii.vec = hatvalues(model)
> order(pii.vec, decreasing = TRUE)
```

```
 [1]  6  8 23  7 22 21  5 18 32  1  9 43  3  4 35  2 11 44 20 36 38 12
[23] 42 17 37 16 13 40 41 15 19 14 39 28 26 27 30 33 29 34 24 31 10 25
```

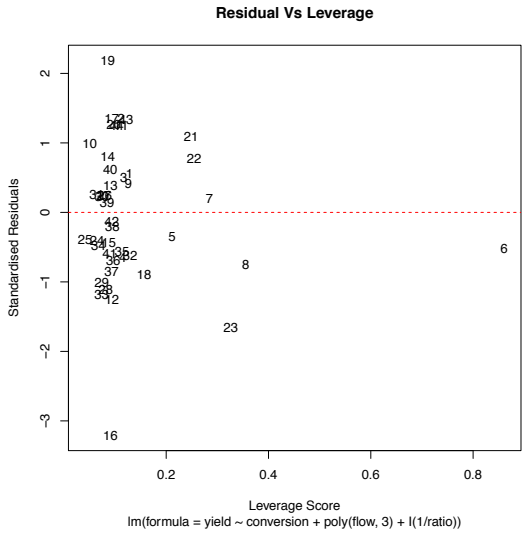
```
> influence.measures(model)
```

```
      dfb.1_ dfb.cnvr dfb.p..3.1 dfb.p..3.2 dfb.p..3.3 dfb.I.1.    dffit cov.r   cook.d
6  -0.184018  0.27195   -0.26389    4.31e-02    0.21870 -1.07461 -1.2852 8.077 0.280687
7  -0.009520  0.01655   -0.09130    6.73e-02   -0.05045 -0.06502  0.1246 1.630 0.002654
8  -0.029285 -0.00373    0.40018   -3.36e-01    0.26729  0.30531 -0.5513 1.667 0.051257
16  0.833470 -0.83617   -0.23336    8.30e-01   -0.01424 -0.08045 -1.1740 0.193 0.171884
```

- Point 6 definitely has excessive leverage.



- Point 16 is an outlier



- But 6 is NOT an outlier despite being picked up by influential measures.
- Recall the following are the rows that are picked up by influential measures

```
      dfb.1_ dfb.cnvr dfb.p..3.1 dfb.p..3.2 dfb.p..3.3 dfb.I.1.    dffit cov.r   cook.d
6  -0.184018  0.27195   -0.26389   4.31e-02    0.21870 -1.07461 -1.2852 8.077 0.280687
7  -0.009520  0.01655   -0.09130   6.73e-02   -0.05045 -0.06502  0.1246 1.630 0.002654
8  -0.029285 -0.00373    0.40018  -3.36e-01    0.26729  0.30531 -0.5513 1.667 0.051257
16  0.833470 -0.83617   -0.23336   8.30e-01   -0.01424 -0.08045 -1.1740 0.193 0.171884
```

- So point 6 is a high leverage influential point, and 16 is an influential outlier.

```
> recip_outlier_index
```

```
16
```

- Refit the model without the two observations,

```
> cubic.rm.LM =
+   lm(yield~conversion
+                   + poly(flow,3) + I(1/ratio),
+                   data = chem_pro.df[-c(6,16),])
>
> summary(cubic.rm.LM)
```

- The current cubic model and the previous model are shown below

```
Call:
lm(formula = yield ~ conversion + poly(flow, 3) + I(1/ratio),
    data = chem_pro.df[-c(6, 16), ])

Coefficients:
              Estimate Std. Error t value Pr(>t)
(Intercept)     96.0543     9.8763   9.726  1.3e-11 ***
conversion      -3.7628     0.9155  -4.110 0.000218 ***
poly(flow, 3)1  -9.2321     2.7876  -3.312 0.002117 **
poly(flow, 3)2   2.0092     2.6767   0.751 0.457747
poly(flow, 3)3   9.0940     2.3355   3.894 0.000410 ***
```

```
I(1/ratio)      -0.2025     0.2828  -0.716 0.478691
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.155 on 36 degrees of freedom
Multiple R-squared:  0.6727,    Adjusted R-squared:  0.6272
F-statistic:  14.8 on 5 and 36 DF,  p-value: 6.765e-08
```

```
Coefficients:
               Estimate Std. Error t value Pr(>t)
(Intercept)    101.6345    10.4408   9.734 7.18e-12 ***
conversion      -4.1717     0.9105  -4.582 4.86e-05 ***
poly(flow, 3)1 -10.4209     3.0096  -3.462 0.001340 **
poly(flow, 3)2   3.7437     3.1710   1.181 0.245090
poly(flow, 3)3  10.0417     2.5762   3.898 0.000382 ***
I(1/ratio)      -0.3644     0.1404  -2.594 0.013386 *
```

- It seems observation 6 and 16 are also problematic for `prod.LM`, however,

```
Call:
lm(formula = yield ~ conversion + poly(flow, 3) + I(1/ratio) +
    I(flow * conversion), data = chem_pro.df[-c(6, 16), ])

Coefficients:
                     Estimate Std. Error t value Pr(>t)
(Intercept)          94.57208    9.93354   9.520 3.02e-11 ***
conversion            0.15272    3.63512   0.042   0.9667
poly(flow, 3)1      118.75295  115.04943   1.032   0.3091
poly(flow, 3)2        0.42501    3.02399   0.141   0.8890
poly(flow, 3)3       12.19298    3.62966   3.359   0.0019 **
I(1/ratio)           -0.20802    0.28194  -0.738   0.4655
I(flow * conversion)  -0.01273    0.01144  -1.113   0.2734
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.148 on 35 degrees of freedom
Multiple R-squared:  0.6839,    Adjusted R-squared:  0.6297
F-statistic: 12.62 on 6 and 35 DF,  p-value: 1.567e-07
```

and eigenvalues/eigenvectors give similar conclusions.

```
> VIF = diag(solve(cor(Xtd[-c(6,16),])));  VIF
```

```
 conversion           1           2           3 recip_ratio        prod
  31.416465 2889.153756    2.177048    2.895338    2.507123 2872.891795
```

- It seems `1/ratio` is only significant because of those two points!

```
> chem_pro_final.LM =
+   lm(yield~conversion + poly(flow,3),
+      data = chem_pro.df[-c(6,16),])
> summary(chem_pro_final.LM)
```

```
Call:
lm(formula = yield ~ conversion + poly(flow, 3), data = chem_pro.df[-c(6, 16), ])

Residuals:
    Min      1Q  Median      3Q     Max
-4.1800 -1.4444  0.2131  1.7172  4.0394

Coefficients:
               Estimate Std. Error t value Pr(>t)
(Intercept)     98.4299     9.2406  10.652 7.98e-13 ***
conversion      -4.0787     0.7968  -5.119 9.77e-06 ***
poly(flow, 3)1  -7.9682     2.1429  -3.718 0.000662 ***
poly(flow, 3)2   1.5336     2.5758   0.595 0.555213
poly(flow, 3)3   8.5939     2.2138   3.882 0.000412 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.141 on 37 degrees of freedom
Multiple R-squared:  0.668,     Adjusted R-squared:  0.6322
F-statistic: 18.61 on 4 and 37 DF,  p-value: 1.844e-08
```

3.7 Heteroskedasticity

- So far, when we have evidence against the assumption of equal variance,

2. The errors have zero mean and constant variance

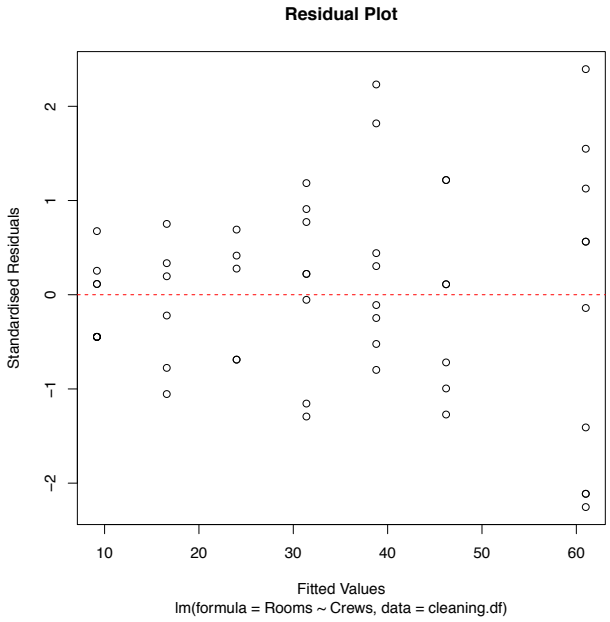
$$\mathbb{E}\left[\varepsilon_i \mid \mathbf{X}_i\right] = 0 \quad \text{and} \quad \text{Var}\left[\varepsilon_i \mid \mathbf{X}_i\right] = \sigma^2 \qquad \text{where} \quad \varepsilon_i = Y_i - \mathbb{E}\left[Y_i \mid \mathbf{X}_i\right]$$

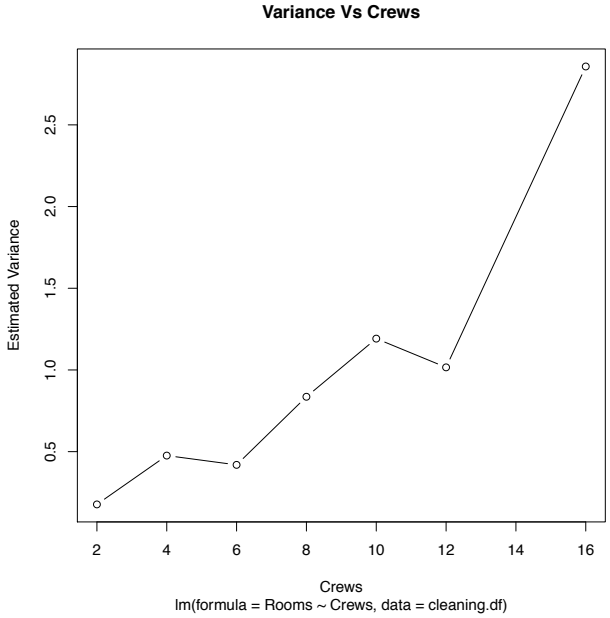
we consider transforming the data. For example, recall the following dataset

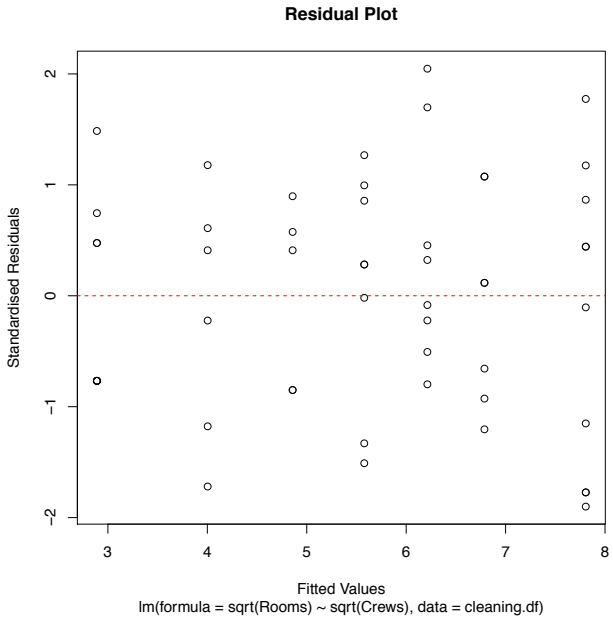
`Crews`	Number of works on the job
`Rooms`	Number of offices need to be cleaned

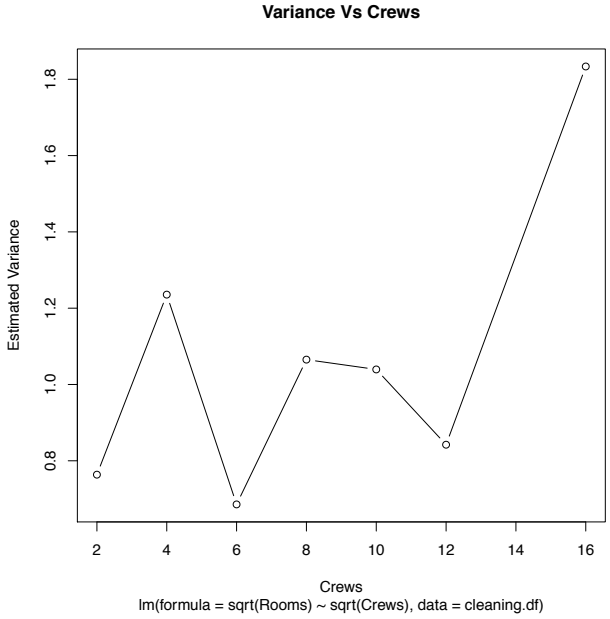
which is about 53 cases that a maintenance company did in the past.

```
> cleaning.df = read.table("~/Desktop/cleaning.txt",
+                          header = TRUE)
>
> cleaning.LM = lm(Rooms~Crews, data = cleaning.df)
```




Variance Vs Crews
Estimated Variance
0.5
1.0
1.5
2.0
2.5
2
4
6
8
10
12
14
16
Crews
lm(formula = Rooms ~ Crews, data = cleaning.df)


Residual Plot
Standardised Residuals
−2
−1
0
1
2
3
4
5
6
7
8
Fitted Values
lm(formula = sqrt(Rooms) ~ sqrt(Crews), data = cleaning.df)



- In the presence of heteroskedasticity, the estimator

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

  is still unbiased and consistent, but is no longer efficient.

- However, the variance of the estimator loses the consistency property as well

$$\begin{aligned}\hat{\mathrm{Var}}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right] &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\hat{\mathrm{Var}}\left[\boldsymbol{\varepsilon} \mid \mathbf{X}\right]\left(\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\right)^{\mathrm{T}} \\ &= \hat{\sigma}_u^2\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \qquad \text{where} \quad \hat{\sigma}_u^2 = \frac{1}{n-k-1}\hat{\mathbf{e}}^{\mathrm{T}}\hat{\mathbf{e}}\end{aligned}$$

- This is particularly problematic if the purpose of the model is to explain since

$$t_j = \frac{\hat{\beta}_j}{\mathrm{SE}\left(\hat{\beta}_j\right)}$$

  where $\mathrm{SE}\left(\hat{\beta}_j\right)$ is the jth main diagonal element of $\hat{\mathrm{Var}}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right]$.

- Recall we have the following under equal variance

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik} + \varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim \mathrm{N}(0, \sigma^2)$$

  and the estimate $\hat{\boldsymbol{\beta}} = \mathbf{b}$ is found by minimising

$$f(\mathbf{b}) = \hat{\mathbf{e}}^{\mathrm{T}}\hat{\mathbf{e}} = \sum_{i=1}^{n}\left(y_i - b_0 - b_1 x_{i1} - \cdots - b_k x_{ik}\right)^2$$

which is given by founding the gradient of the following and setting it to $\mathbf{0}$

$$f(\mathbf{b}) = (\mathbf{y} - \mathbf{X}\mathbf{b})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\mathbf{b}) = \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{b}$$
$$\Longrightarrow \hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

- We could have written the equation differently without affecting $\hat{\boldsymbol{\beta}} = \mathbf{b}$

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik} + \sigma\varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim \mathrm{N}(0, 1)$$

- Now with heteroskedasticity, we have

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik} + \sigma_i\varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim \mathrm{N}(0, 1)$$

- If we scale our variables according to the true values of σ_i, we have

$$\frac{Y_i}{\sigma_i} = \beta_0\frac{1}{\sigma_i} + \beta_1\frac{X_{i1}}{\sigma_i} + \cdots + \beta_k\frac{X_{ik}}{\sigma_i} + \varepsilon_i$$
$$Y_i^* = \beta_0\frac{1}{\sigma_i} + \beta_1 X_{i1}^* + \cdots + \beta_k X_{ik}^* + 1\varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim \mathrm{N}(0, 1)$$

from which we see all the nice properties will be back if y_i^* and x_i^* are used

$$\hat{\mathbf{e}}_w^{\mathrm{T}}\hat{\mathbf{e}}_w = \sum_{i=1}^{n}\left(y_i^* - b_0/\sigma_i - b_1 x_{i1}^* - \cdots - b_k x_{ik}^*\right)^2$$
$$= \sum_{i=1}^{n}\frac{1}{\sigma_i^2}\left(y_i - b_0 - b_1 x_{i1} - \cdots - b_k x_{ik}\right)^2 = \hat{\mathbf{e}}^{\mathrm{T}}\mathbf{W}\hat{\mathbf{e}}$$

where $\mathbf{W}$ denote the diagonal matrix containing the scaling factor $\dfrac{1}{\sigma_i^2}$.

- In terms of original response, the objective function now is given by

$$f(\mathbf{b}) = \hat{\mathbf{e}}_w^{\mathrm{T}}\hat{\mathbf{e}}_w = \hat{\mathbf{e}}^{\mathrm{T}}\mathbf{W}\hat{\mathbf{e}} = (\mathbf{y} - \mathbf{X}\mathbf{b})^{\mathrm{T}}\mathbf{W}(\mathbf{y} - \mathbf{X}\mathbf{b})$$
$$= \mathbf{y}^{\mathrm{T}}\mathbf{W}\mathbf{y} - 2\mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{y} + \mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}\mathbf{b}$$

- Differentiating with respect to $\mathbf{b}$, we have the following gradient,

$$\nabla f = -2\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{y} + 2\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}\mathbf{b}$$

- Hence the following estimator will have the same nice properties as before

$$\hat{\boldsymbol{\beta}}_w = \left(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{y}$$

which is known as the weighted least squares estimator.

- Of course, the true σ_i^2 are unknown in practice, thus we have estimate

$$\mathbf{W}$$

Q: What is the intuitive reason behind the weighted least squares estimator?

- For the maintenance company data,

```
> crews.group = factor(cleaning.df$Crews)
>
> tapply(cleaning.df$Rooms, crews.group, length)
```

```
 2  4  6  8 10 12 16
 9  6  5  8  8  7 10
```

which has many repeated observations, one clearly choice of estimator for

$$\mathbf{W} = \text{diag}\left(\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \cdots, \frac{1}{\sigma_i^2}, \cdots, \frac{1}{\sigma_n^2}\right)$$

is based on the conditional sample variance of the response, that is,

```
> convar = tapply(cleaning.df$Rooms,crews.group,var)
> convar
```

```
       2        4        6        8       10       12       16
 9.00000 24.66667 22.00000 44.12500 62.83929 53.14286 144.01111
```

```
> # vector of corresponding conditional variance
> v.vec = convar[crews.group]

> w.vec = 1/v.vec # vector of diagonal elements of w

> cleaning.WLS = lm(Rooms~Crews, weights = w.vec,
+                   data = cleaning.df)
```

- The above syntax leads to the desired estimate of

$$\hat{\boldsymbol{\beta}}_w = \left(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{y}$$

and as before the fitted value is given by

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}_w$$

- However, the residual analysis should be done using the following residual

$$\hat{\mathbf{e}}_w = \mathbf{W}^{1/2}\hat{\mathbf{e}} \qquad \text{where} \qquad \hat{\mathbf{e}} = \mathbf{y} - \hat{\mathbf{y}}$$

- It is the residual $\hat{\mathbf{e}}_w$, not $\hat{\mathbf{e}}$, that should be used as the estimate of the error

$$\boldsymbol{\varepsilon}$$

that satisfies the model assumptions, i.e.

$$Y_i = \mathbb{E}\left[Y_i \mid \mathbf{X}_i\right] + \sigma_i\varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim \text{N}(0, 1)$$

$$\implies \varepsilon_i = \frac{Y_i - \mathbb{E}\left[Y_i \mid \mathbf{X}_i\right]}{\sigma_i}$$

- In terms of our current model, we obtain $\hat{\mathbf{e}}_w$ by running the following

```
> model = cleaning.WLS
>
> res = residuals(model)
>
> resw = sqrt(w.vec) * res
```

- Using conditional sample variance to estimate σ_i^2, thus the weight matrix

$$\mathbf{W}$$

is common but could fail badly when the groups of similar or identical points

```
> tapply(cleaning.df$Rooms, crews.group, length)
```

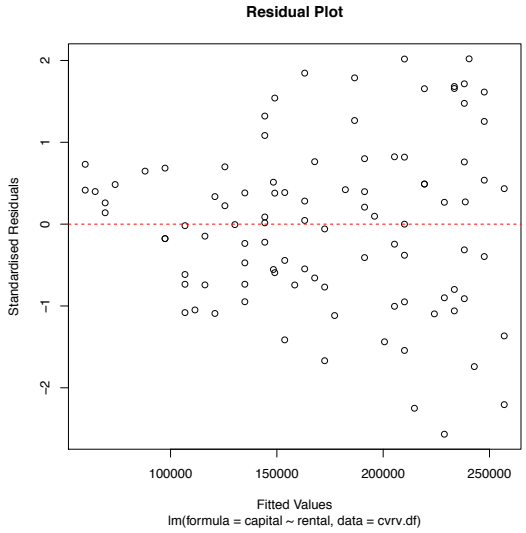
```
 2  4  6  8 10 12 16
 9  6  5  8  8  7 10
```

are too small, say less than 10. So it actually not reliable for this dataset.

- This approach is certainly not reliable for the following dataset

```
> cvrv.df =
+   read.table("~/Desktop/cvrv.csv",
+              sep = ",", header = TRUE)
>
> cvrv.LM = lm(capital~rental, data = cvrv.df)
```

for which the variance is clearly unequal, and `rental` varies continuously.



- Notice the following equality

$$\sigma_i^2 = \text{Var}\left[\varepsilon_i \mid \mathbf{X}\right] = \mathbb{E}\left[\left(\varepsilon_i - \mathbb{E}\left[\varepsilon_i \mid \mathbf{X}\right]\right)^2 \mid \mathbf{X}\right]$$

$$= \mathbb{E}\left[\varepsilon_i^2 \mid \mathbf{X}\right]$$

- This leads us to the following estimate for σ_i^2,

$$\hat{\sigma}_i^2 = \mathbb{E}\left[\hat{e}_i^2 \mid \mathbf{X}\right]$$

  which is a conditional mean of a random variable, that can be observed, and can in turn be estimated by the fitted value of another linear regression.

- Since $\hat{e}_i^2$ might be really small/large in practice, thus people often work with

$$z_i = \ln\left(\hat{e}_i^2\right) = 2\ln|\hat{e}_i| \implies \hat{e}_i^2 = \exp\left(z_i\right)$$

  the log-scale provides extra numerical stability.

- Thus `z` is the auxiliary response,

```
> z = 2 * log(abs(cvrv.LM$residuals))

> # Perform the auxiliary regression
> auxiliary.LM = lm(z~rental, data = cvrv.df)
```

- Transform back to obtain the estimated σ_i^2

```
> v.vec = exp(auxiliary.LM$fitted.values)
>
> w.vec = 1/v.vec
```

- Specify the weights according to the estimated σ_i^2

```
> cvrv.WLS = lm(capital~rental, weights = w.vec,
+               data = cvrv.df)
```

- Of course, this can also be performed for the maintenance dataset, and WLS can be used when we have k predictors in general.

3.8 Correlated Errors

- Consider the following simple example to start our discussion on

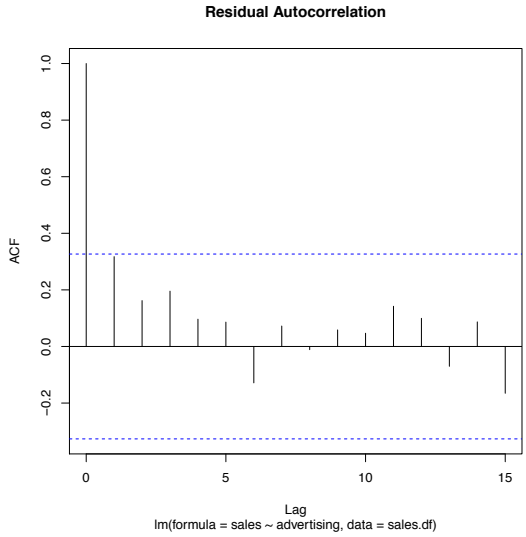
lack of independence

- The data is about sales and advertising expenditure

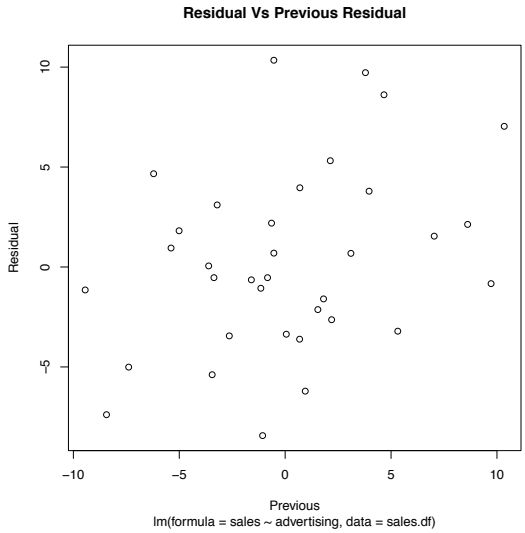
`sales`	Monthly sales of a retailer
`advertising`	Amount spent on advertising this month

```
> sales.df =
+   read.table("~/Desktop/sales.txt",
+              sep = "", header = TRUE)
>
> sales.LM = lm(sales~advertising, data = sales.df)
```

- It is expected that advertising might take a while to come in effect.
- Although we don't have clear evidence from acf that the errors are correlated



- Extra care should be taken when data is collected over time.



- For a simple linear regression model,

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

  where ε_i is independent from ε_j for $i \neq j$, and independent from X_j for all j.
- One of many possibilities that errors can lack independence is the following

$$\mathbb{E}\left[Y_i \mid X_1, X_2, \dots X_n\right] = \beta_0 + \beta_1 x_i + \beta_2 x_{i-1} + \beta_3 x_{i-2} + \cdots$$

  which is known as a distributed lag model since the following no longer holds

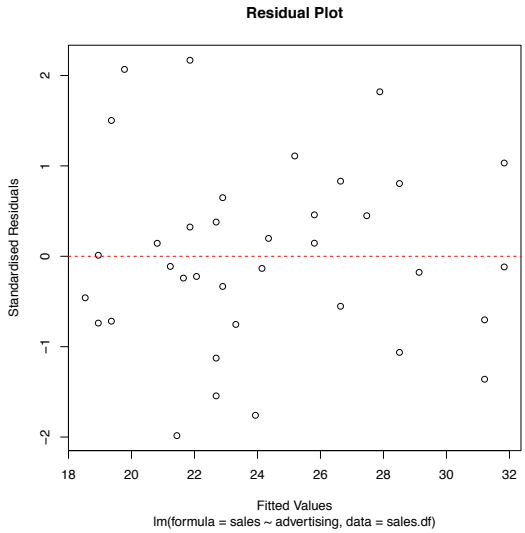
$$\mathbb{E}\left[Y_i \mid X_1, X_2, \dots X_n\right] = \beta_0 + \beta_1 x_i$$

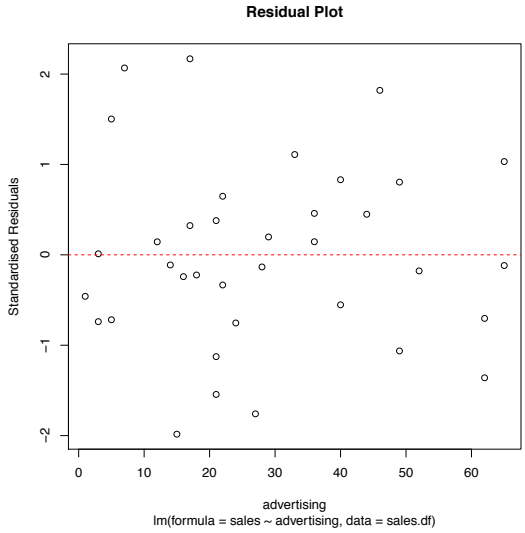
- Let ν_i be independent from ν_j for $i \neq j$, and independent from X_j for all j,

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_i + \beta_2 X_{i-1} + \nu_i \\ &= \beta_0 + \beta_1 X_i + \varepsilon_i \qquad \text{where} \quad \varepsilon_i = \beta_2 X_{i-1} + \nu_i \end{aligned}$$

Q: Why is this a problem? How can we detect it? What do we expect to see?

- In addition to the following two residual plots,





- The lagged predictors should be plotted with the standardised residual,

```
> lag = 1 # This might take other natural numbers

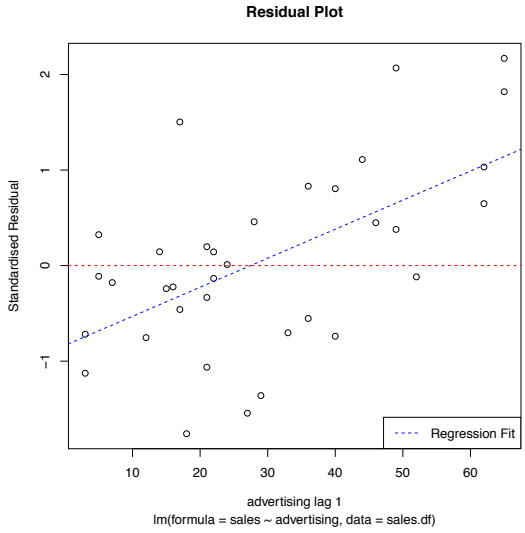
> index = 0:(lag-1) - nrow(sales.df)

> x = sales.df$advertising[index]

> y = rstandard(sales.LM)[-(1:lag)]
```

```
> plot(x, y, xlab = bquote("advertising lag"~.(lag)),
+      ylab = "Standardised Residual",
+      main = "Residual Plot",
+      sub = deparse(model$call))
>
> abline(a = 0, b = 0, lty = 2, col = "red")
> abline(lm(y~x), lty = 2, col = "blue")
> legend("bottomright", "Regression Fit",
+        lty = 2, col = 4)
```

- We can expect a nonrandom pattern in the plot below if ε_i depends on X_{i-1}



- The plot provides evidence that the error is not independent of the predictor,

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_{i-1} + \nu_i$$

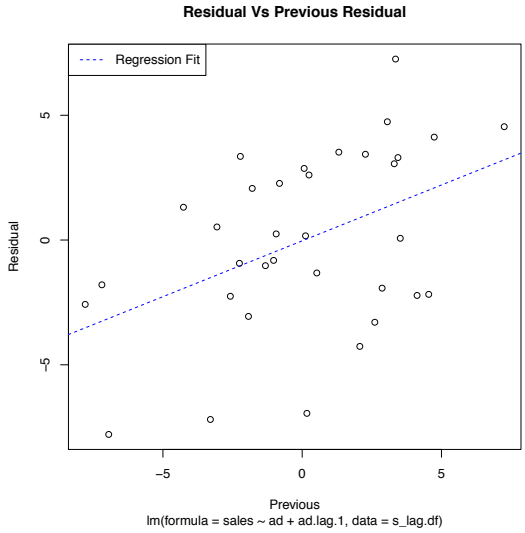
$$= \beta_0 + \beta_1 X_i + \varepsilon_i \qquad \text{where} \quad \varepsilon_i = \beta_2 X_{i-1} + \nu_i$$

- If the above is a valid model, then having the first order lag of the predictor in the model will satisfy independence assumption.

```
> sales = sales.df$sales[-(1:lag)]
>
> ad = sales.df$advertising[-(1:lag)]
>
> ad.lag.1 = sales.df$advertising[index]

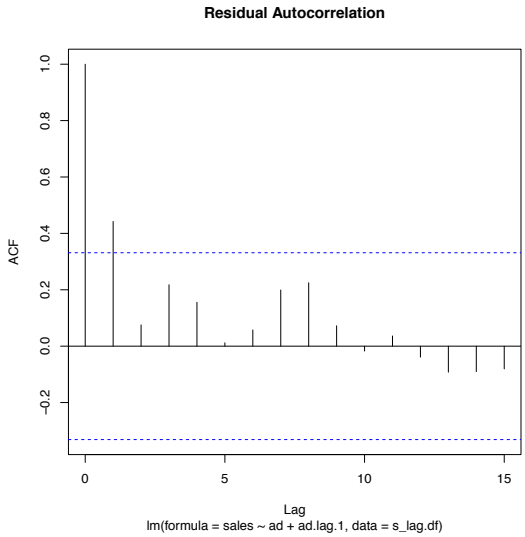
> s_lag.df = data.frame(sales = sales,
+                       ad = ad,
+                       ad.lag.1 = ad.lag.1)
```

- However, it does not solve the lack of independence problem for this dataset.

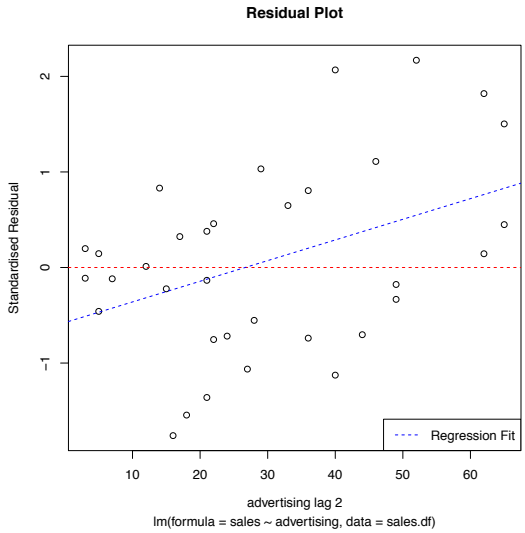


There is a slight but unmistakable linear relationship

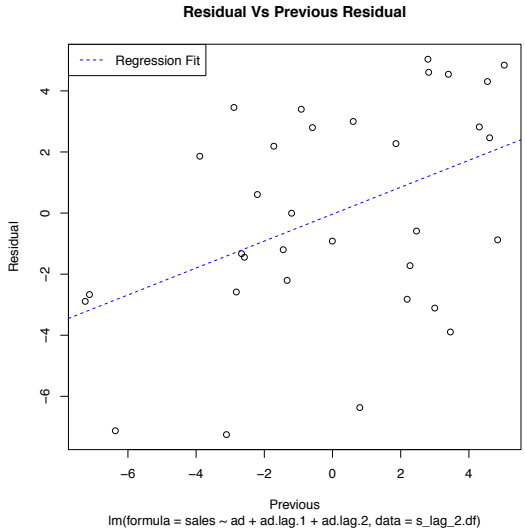
- The linear relationship between residual and previous residual is confirmed by



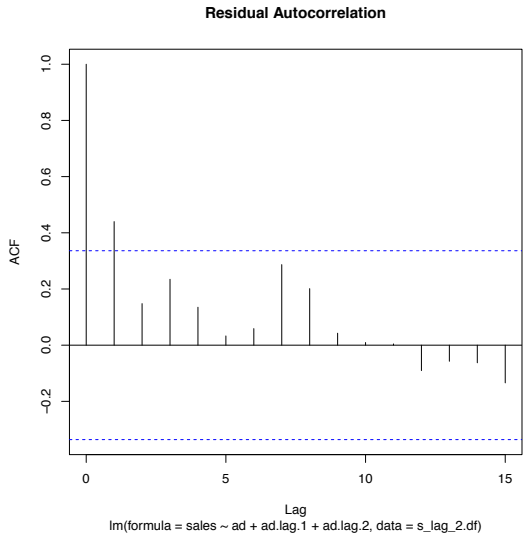
- Looking at the second order lagged values, X_{i-2}, you might be tempted to...



- However, including higher order lagged terms is not helping at all!



- The slight but unmistakable linear trend still presents.



- Another possible and simple way that error can lack independence is having

$$\varepsilon_i = \rho\varepsilon_{i-1} + \nu_i$$

where ν_i are independent and identically distributed for all i

$$\nu_i \sim \mathrm{N}(0, \sigma^2)$$

- This structure in errors is called first-order autoregressive process or AR(1).

Q: How can detect the presence of such structure?

```
> s_lag.LM$call
```

```
lm(formula = sales ~ ad + ad.lag.1, data = s_lag.df)
```

```
> res = residuals(s_lag.LM)
>
> cor(res[-1], res[-length(res)])
```

```
[1] 0.4450734
```

- Of course, we have seen this value

```
> cor(res[-1], res[-length(res)])
```

```
[1] 0.4450734
```

Q: However, can this value be due to chance while the true ρ is zero?

$$\varepsilon_i = \rho\varepsilon_{i-1} + \nu_i$$

- There are 3 tools, the 1st you know, using which depends on the data size.

```
> res.lag.df = data.frame(x = res[-length(res)],
+                         y = res[-1])

> auxiliary.LM = lm(y~x, data = res.lag.df)
```

- Of course, we have to ask whether we can trust the output of

```
> summary(auxiliary.LM)
```

- It can be shown we have no evidence against the auxiliary model being valid

```
> summary(auxiliary.LM)
```

```
Call:
lm(formula = y ~ x, data = res.lag.df)

Residuals:
    Min      1Q  Median      3Q     Max
-6.9895 -2.0044  0.8178  2.5104  5.7853

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) -0.03219    0.56092  -0.057  0.95460
x            0.44763    0.15921   2.812  0.00835 **
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 3.27 on 32 degrees of freedom
Multiple R-squared:  0.1981,    Adjusted R-squared:  0.173
F-statistic: 7.905 on 1 and 32 DF,  p-value: 0.00835
```

- The data size of our current example is arguably too small,

```
> lmtest::dwtest(s_lag.LM, alternative = "two.sided")
```

```
        Durbin-Watson test

data:  s_lag.LM
DW = 1.1039, p-value = 0.004222
alternative hypothesis: true autocorrelation is not 0
```

- All three methods indicate we need AR(1) for our current example.
- In its simplest form, AR(1) is given by the following

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

$$\varepsilon_i = \rho \varepsilon_{i-1} + \nu_i \qquad \text{where} \quad \nu_i \sim \mathrm{N}\left(0, \sigma^2\right)$$

where the errors are usually assumed to be stationary, that is

$$\mathbb{E}\left[\varepsilon_i\right] = 0 \qquad \text{and} \qquad \mathrm{Var}\left[\varepsilon_i\right] = \sigma_\varepsilon^2 \qquad \text{for all } i$$

under which, it can be shown the correlation coefficient is given by

$$r_{i,i-r} = \mathrm{Corr}\left[\varepsilon_i, \varepsilon_{i-r}\right] = \rho^r$$

for $i = 1, 2, \ldots, n$ and $r = 1, 2, \ldots, n-1$, where $i > r$.

Q: What is the distribution of $\varepsilon_i \mid \varepsilon_{i-1}$?

Q: What is this simple AR(1) different from simple linear regression?

The covariance is given by

$$\begin{aligned}\operatorname{Cov}\left[\varepsilon_i, \varepsilon_{i-1}\right] &= \mathbb{E}\left[\varepsilon_i \varepsilon_{i-1}\right] - \mathbb{E}\left[\varepsilon_i\right]\mathbb{E}\left[\varepsilon_{i-1}\right] = \mathbb{E}\left[\left(\rho\varepsilon_{i-1} + \nu_i\right)\varepsilon_{i-1}\right] \\ &= \rho\mathbb{E}\left[\varepsilon_{i-1}^2\right] + \mathbb{E}\left[\nu_i\right]\mathbb{E}\left[\varepsilon_{i-1}\right] = \rho\sigma_\varepsilon^2\end{aligned}$$

- When there are k predictors, and n observations, in general matrix notation,

  $$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

  however, instead of assuming that the errors are independent, we assume

  $$\boldsymbol{\varepsilon} \sim \mathrm{N}\left(\mathbf{0}, \boldsymbol{\Sigma}\right)$$

  where $\boldsymbol{\Sigma}$ is the covariance matrix, and $\mathbf{R}$ is the correlation coefficient matrix

  $$\boldsymbol{\Sigma} = \sigma_\varepsilon^2\mathbf{R} = \frac{\sigma^2}{1-\rho^2}\begin{bmatrix} 1 & \rho & \cdots & \rho^{n-1} \\ \rho & 1 & \cdots & \rho^{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & \cdots & 1 \end{bmatrix}$$

- If σ^2 and ρ are given, the estimation problem can be solved similarly to WLS,

  $$\hat{\boldsymbol{\beta}}_g = \left(\mathbf{X}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{y}$$

  which is known as generalised least squares (GLS) estimator of β.

- Combing AR(1) together with our early distributed lag model,

  $$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_{i-1} + \varepsilon_i$$

  $$\varepsilon_i = \rho\varepsilon_{i-1} + \nu_i \qquad \text{where} \quad \nu_i \sim \mathrm{N}\left(0, \sigma^2\right)$$

  we have the following model,

  $$Y_i = \beta_0(1-\rho) + \beta_1 X_i + (\beta_2 - \rho\beta_1)X_{i-1} - \beta_2\rho X_{i-2} + \rho Y_{i-1} + \nu_i$$

- When σ^2 and ρ are not given, this is a nonlinear optimisation problem

  $$\begin{aligned}\ell\left(\boldsymbol{\beta}, \rho, \sigma^2; \mathbf{X}^*, \mathbf{y}\right) &= -\ln\left(L\left(\boldsymbol{\beta}, \rho, \sigma^2; \mathbf{X}^*, \mathbf{y}\right)\right) \\ &= \frac{n}{2}\ln(2\pi) + \frac{1}{2}\ln\left(\det\boldsymbol{\Sigma}\right) \\ &\quad + \frac{1}{2}\left(\mathbf{y} - \mathbf{X}^*\boldsymbol{\beta}\right)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(\mathbf{y} - \mathbf{X}^*\boldsymbol{\beta}\right)\end{aligned}$$

  which has too be solved numerically.

- R can construct AR models, and solve the optimisation without the details

```
> with(s_lag.df,
+      {s_lag.AR1 = arima(
+        sales, order = c(1, 0, 0),
+        xreg = cbind(ad, ad.lag.1))
+      })
> s_lag.AR1
```

```
Call:
arima(x = sales, order = c(1, 0, 0), xreg = cbind(ad, ad.lag.1))

Coefficients:
         ar1  intercept      ad  ad.lag.1
      0.4966    16.9080  0.1218    0.1391
s.e.  0.1580     1.6716  0.0308    0.0316

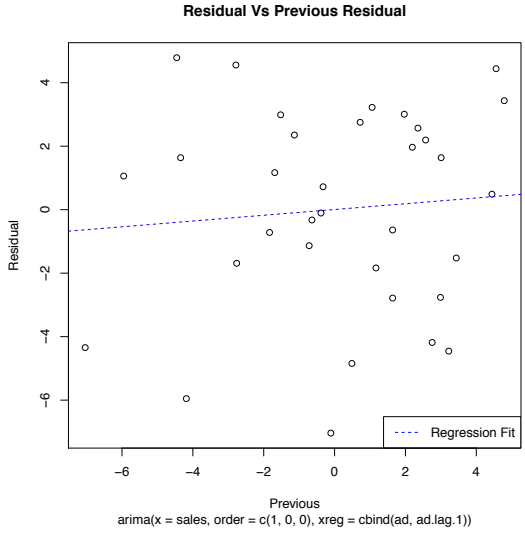
sigma^2 estimated as 9.476:  log likelihood = -89.16,  aic = 188.32
```

```
> s_lag.LM
```

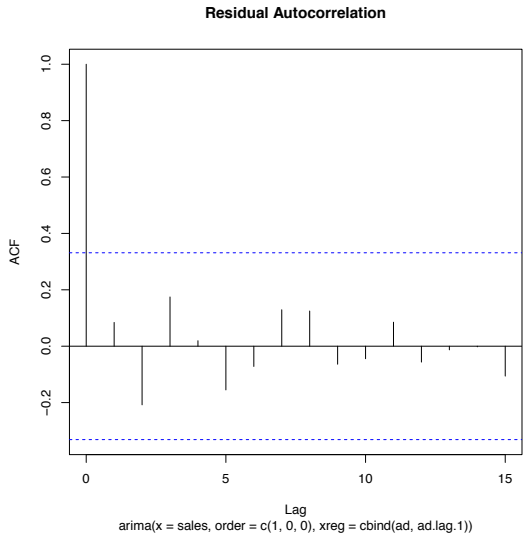
```
Call:
lm(formula = sales ~ ad + ad.lag.1, data = s_lag.df)

Coefficients:
(Intercept)           ad     ad.lag.1
    15.6036       0.1424       0.1665
```

- It seems the new error ν_i are independent and identically distributed,



- It seems the new error ν_i are independent and identically distributed,



- The rest of assumptions seem to be fine, however, the model equation is

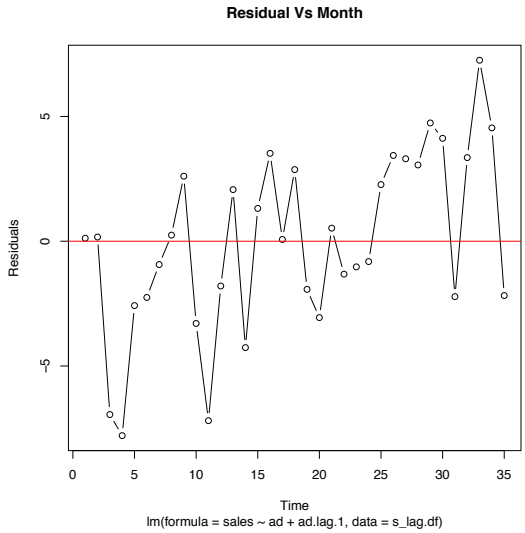
$$Y_i = \beta_0(1-\rho) + \beta_1 X_i + (\beta_2 - \rho\beta_1)X_{i-1} - \beta_2\rho X_{i-2} + \rho Y_{i-1} + \nu_i$$

  more advanced models are harder to interpret, should be avoided if possible.
- In this case, there is actually a simpler model that is reasonable good.
- Consider the following model again

```
> sales_pre.LM = lm(sales~ad+ad1, data = s_pre.df)
>
> res = sales_pre.LM$residuals

> plot(res, type = "b",
+      xlab = "Time", ylab = "Residuals",
+      main = "Residual Vs Month")
>
> abline(h = 0, col = "red")
```

- Notice there seems to be an weak increasing trend



- It is very likely that the data is recorded/sorted according to time, the plot suggests that time might help to explain the error, thus the sales number.

```
> s_lag_m.df = cbind(s_lag.df, m = 1:nrow(s_lag.df))

> s_f.LM = lm(sales~ad+ad.lag.1+m,data = s_lag_m.df)

> summary(s_f.LM) # A peek before checking residuals
```

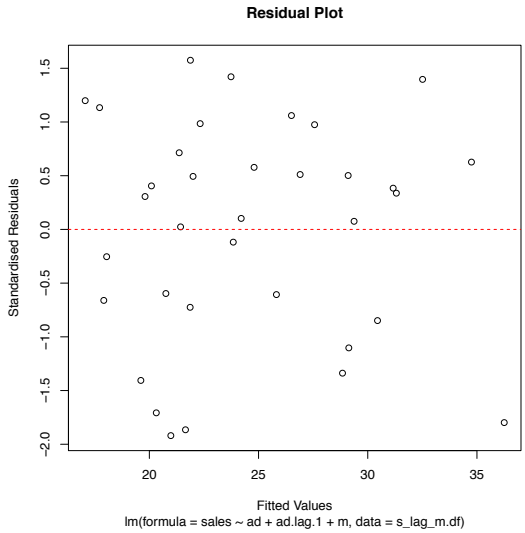
```
Call:
lm(formula = sales ~ ad + ad.lag.1 + m, data = s_lag_m.df)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4840 -2.0409  0.8971  1.9423  4.5192

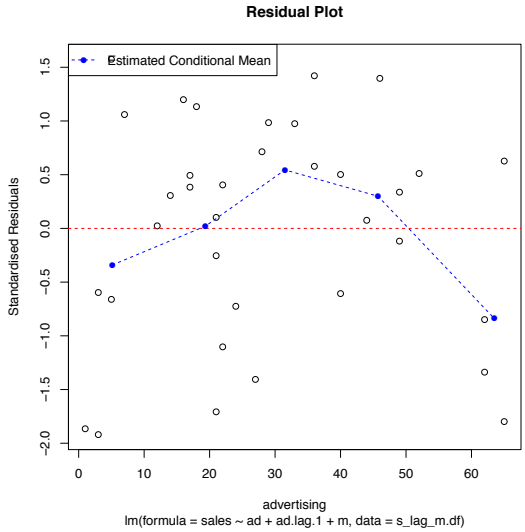
Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) 12.03345    1.48653   8.095 3.84e-09 ***
ad           0.15308    0.02984   5.129 1.48e-05 ***
ad.lag.1     0.15914    0.03052   5.214 1.16e-05 ***
m            0.19323    0.05189   3.724 0.000781 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 3.084 on 31 degrees of freedom
Multiple R-squared:  0.7507,    Adjusted R-squared:  0.7266
F-statistic: 31.12 on 3 and 31 DF,  p-value: 1.769e-09
```

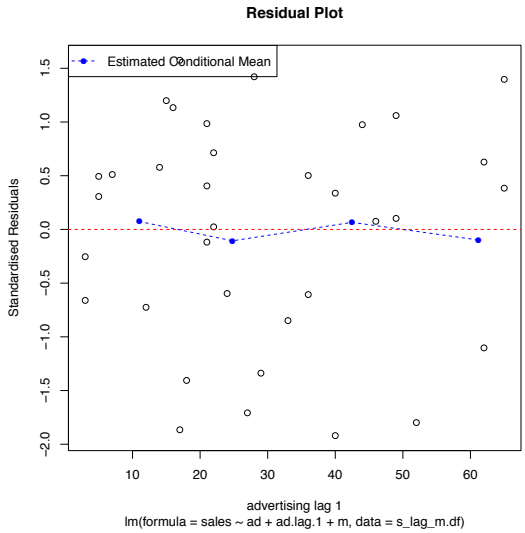
- It seems time, `m`, is highly significant, but we have to check residuals first,



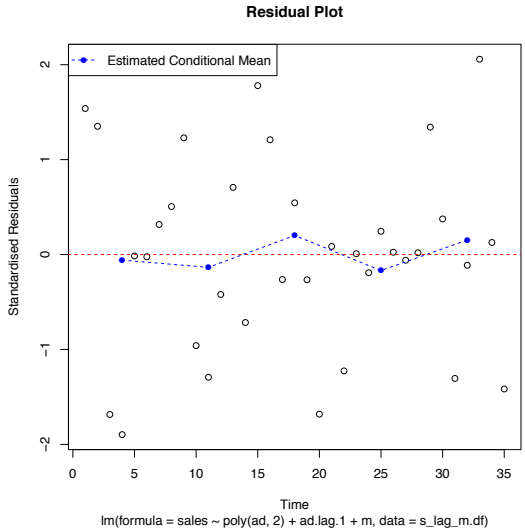
- It seems that we need a polynomial term for `ad`,



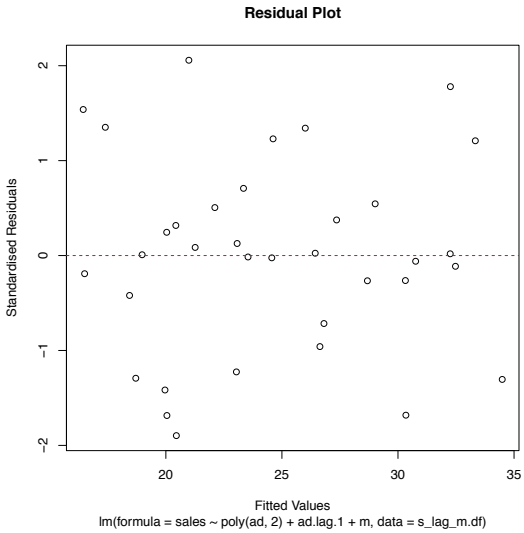
- The lag 1 term seems to be OK.

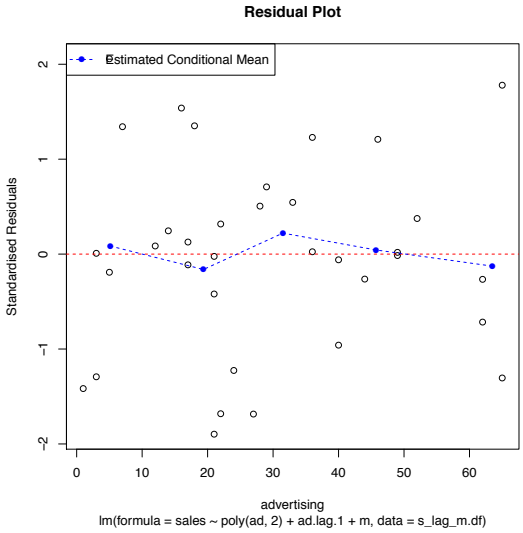


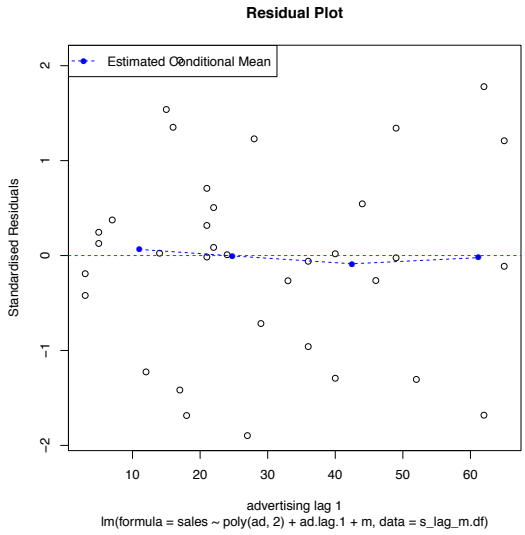
- We also have no evidence against time being linear.

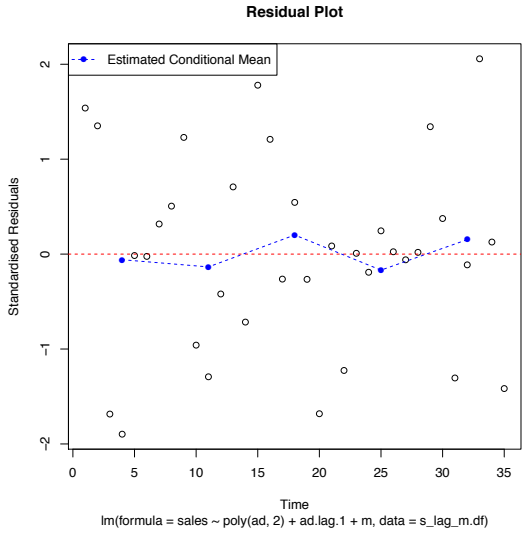


- We have no evidence against linearity from the next 4 plots.

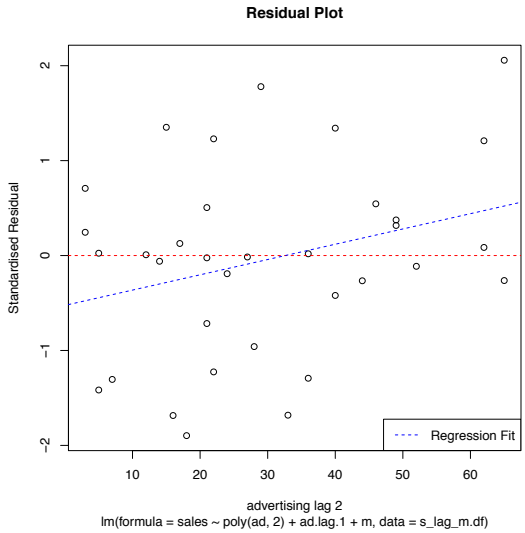

Residual Plot
Standardised Residuals
Fitted Values
lm(formula = sales ~ poly(ad, 2) + ad.lag.1 + m, data = s_lag_m.df)


Residual Plot
Estimated Conditional Mean
Standardised Residuals
advertising
lm(formula = sales ~ poly(ad, 2) + ad.lag.1 + m, data = s_lag_m.df)





- And we have no evidence that errors depend on higher order lagged `ad`



- There are no evidence against the following auxiliary regression being valid

```
> lag = 1
> index = 0:(lag-1) - nrow(s_lag_m.df)
>
> x = s_lag_m.df$ad.lag.1[index]
> y = rstandard(sales_final.LM)[-(1:lag)]
>
> auxiliary.LM = lm(y~x)

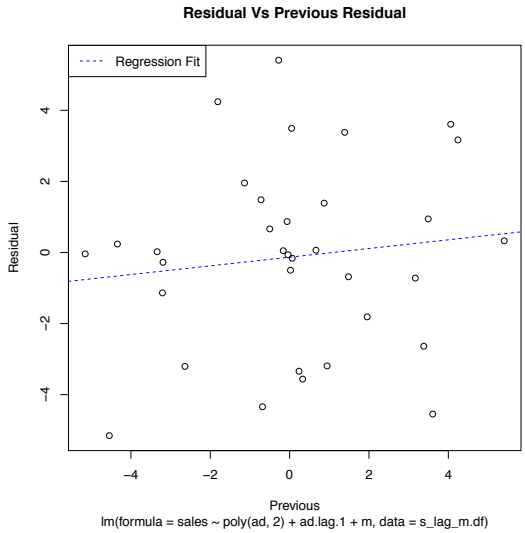
> summary(auxiliary.LM)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) -0.524933   0.317745  -1.652   0.1083
x            0.016100   0.009111   1.767   0.0868 .
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 0.9731 on 32 degrees of freedom
Multiple R-squared:  0.0889,    Adjusted R-squared:  0.06043
F-statistic: 3.123 on 1 and 32 DF,  p-value: 0.08675
```

based on the t-test, we do not pursue a higher oder distributed lag model.

- We have no evidence of more first order autocorrelation,

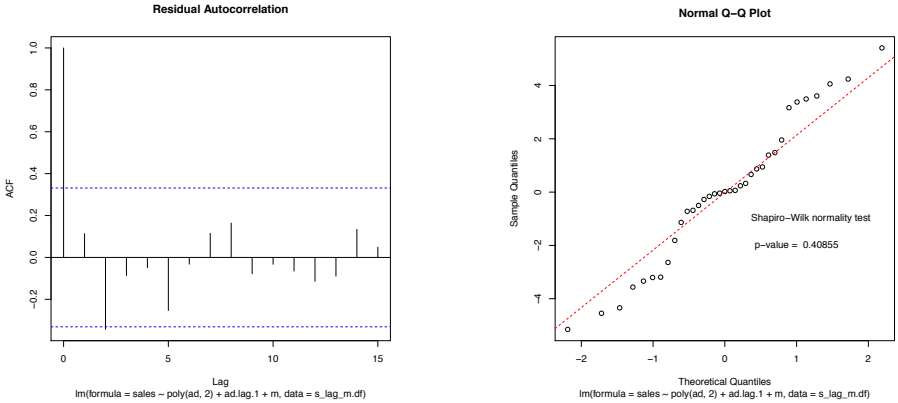



```
> lmtest::dwtest(s_f.LM, alternative = "two.sided")
```

```
        Durbin-Watson test

data:  s_f.LM
DW = 1.6193, p-value = 0.1615
alternative hypothesis: true autocorrelation is not 0
```

- However, the fix for independence might not be as good as the AR(1) model.



- However, given we have only a relatively small number of data points, and we probably should not push for AR(2) without a significant AR(1) term.
- Since the normality assumption seems to be reasonable as well, we procedure

```
> summary(sales_final.LM)
```

```
Call:
lm(formula = sales ~ poly(ad, 2) + ad.lag.1 + m, data = s_lag_m.df)

Residuals:
    Min      1Q  Median      3Q     Max
-5.1523 -1.4739  0.0221  1.4357  5.4090

Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept)  15.40536    1.32144  11.658 1.15e-12 ***
poly(ad, 2)1 16.90116    3.04137   5.557 4.83e-06 ***
poly(ad, 2)2 -7.75778    3.09681  -2.505   0.0179 *
ad.lag.1      0.16483    0.02831   5.823 2.29e-06 ***
m             0.24254    0.05185   4.678 5.77e-05 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.851 on 30 degrees of freedom
Multiple R-squared:  0.7939,    Adjusted R-squared:  0.7664
F-statistic: 28.88 on 4 and 30 DF,  p-value: 6.653e-10
```

Q: If you are the manager, what conclusions can you draw from this model?

3.9 Variable Selection

- So far we have only a handful of regressors/predictors/explanatory variables.

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_k X_{ik} + \varepsilon_i$$

  that is, the above full model has a relatively small k value.

- Variable selection aims to choose a subset of p regressors that is

  "best"

  in some sense out of k regressors, it often deals with a reasonably large k

$$5 \leq k \leq 30$$

- Notice there are various ways to define the "best" model.
- Often the following criteria are used wrongly!

  RSS, R^2, R^2_{adj}, Mallow's C_p, t-test, partial F-test, AIC, BIC

Q: What can go wrong with variable selection?

- If we introduce too few, we are underfitting, and if we introduce too many, we are overfitting, both of which lead to unpleasant consequences.
- The type of consequences that we would try to avoid depends on the reason for building a regression model, there are two purposes for building a model:

1. To explain
2. To predict

Q: Why is it unlikely to find a model that fulfil both purposes?

- There is a bias-variance trade-off,

$$\underbrace{\mathbb{E}\left[\left(Y_{n+1} - \hat{Y}^*_{n+1}\right)^2 \mid \cdot\right]}_{\text{mean squared error}} = \sigma^2 + \underbrace{\text{Var}\left[\hat{Y}^*_{n+1} \mid \cdot\right]}_{\text{variance}} + \underbrace{\mathbb{E}\left[\hat{Y}^*_{n+1} - \mu_{n+1} \mid \cdot\right]^2}_{\text{bias}}$$

  however, the trade-off is not necessarily one-for-one.

- In both cases, we would like to select a model that has a small MSE.

Q: How can we get some idea on MSE?

- Data splitting

  If we have a lot of data, we can divide the data into two parts, the training set and the test set. We use the training set to build models and test set to estimate the MSE of prediction.

- Cross-validation

  If there is not sufficient data, we split the data into several parts. Treating one of them as the test set and estimate MSE, and the rest as the training set. Repeat by treating each of other parts as the test set and the rest as the training set. Take the mean of all of those estimates.

- Bootstrap

  Sample the data with replacement to create a training set, and use the original data set as the test set and estimate the MSE. Repeat this many times and take the mean of all of those estimates.

- The last two are computationally intense, thus were avoided in the old days.
- So when we have the "luxury", use the last three. However, traditionally,
- Residual sum of squares

$$\begin{aligned}\mathrm{RSS} = (n-p-1)\,\sigma_u^2 = \hat{\mathbf{e}}^{\mathrm{T}}\hat{\mathbf{e}} &= \mathbf{y}^{\mathrm{T}}\left(\mathbf{I}-\mathbf{P}\right)\mathbf{y} \\ &= \mathbf{y}^{\mathrm{T}}\left(\mathbf{I}-\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\right)\mathbf{y}\end{aligned}$$

  This simply uses the residual sum of squares as the estimate for MSE. This is usually too optimistic and underestimates the prediction error.

- Coefficient of determination

$$R^2 = \frac{\mathrm{ESS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

  where ESS is the explained sum of squares and TSS the total sum of squares

$$\mathrm{TSS} = \sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2 = \left(\mathbf{y}-\frac{\mathbf{1}^{\mathrm{T}}\mathbf{y}\mathbf{1}}{n}\right)^{\mathrm{T}}\left(\mathbf{y}-\frac{\mathbf{1}^{\mathrm{T}}\mathbf{y}\mathbf{1}}{n}\right)$$

  This is only valid if the two models have the same number of regressors.

- Adjusted coefficient of determination

$$R^2_{\mathrm{adj}} = 1 - \frac{\mathrm{RSS}/(n-p-1)}{\mathrm{TSS}/(n-1)}$$

  This can be agued to correct the bias in R^2 and can be used when the two models have different number of regressors. Notice this equivalent to using

$$\sigma_u^2 = \frac{1}{n-p-1}\hat{\mathbf{e}}^{\mathrm{T}}\hat{\mathbf{e}}$$

  which tends to select a model that suffers from over-fitting.

- Mallow's C_p

$$C_p = \frac{\text{RSS}_p}{\hat{\sigma}^2_{ku}} - n + 2p$$

Before cross-validation/ bootstrap was realistic, it was used as an estimate of MSE and address the issue of overfitting without regression assumptions.

- t-test

This removes 1 regressor at a time from the model according the p-value.

- Partial F-test

This checks the adequacy of the submodel according to the p-value.

- Akaike information criterion

$$\text{AIC} = 2p^* - 2\ln\left(L\left(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2; \mathbf{y}, \mathbf{X}\right)\right)$$

where $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ denote the maximum likelihood estimates of $\boldsymbol{\beta}$ and σ^2, and

$$p^* = p + 2$$

is the number of parameters estimated in the model.

Q: Can you work out whether a better model is indicated by a small or big AIC?

- AIC is based on Kullback-Leibler divergence,

$$I\left(f, g\right) = \int f(z) \ln\left(\frac{f(z)}{g(z;\boldsymbol{\theta})}\right) dz$$

which is a distance measure between two distributions.

- In some sense, it measures the amount information lost when distribution

$$g\left(z;\theta\right)$$

is used to model distribution

$$f(z)$$

- Since $f(z)$ is a density function, we have

$$\begin{aligned} I(f,g) &= \int f(z) \ln\Big(f(z)\Big)\, dz - \int f(z) \ln\Big(g(z;\boldsymbol{\theta})\Big)\, dz \\ &= \mathbb{E}\left[\ln\Big(f(z)\Big)\right] - \mathbb{E}\left[\ln\Big(g(z;\boldsymbol{\theta})\Big)\right] \end{aligned}$$

- Akaike found the likelihood function L of the model can be used to estimate

$$\mathbb{E}\left[\ln\Big(g(z;\boldsymbol{\theta})\Big)\right]$$

and the bias of using $\ln\left(L\left(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2; \mathbf{y}, \mathbf{X}\right)\right)$ is approximately p^*, thus

$$\hat{I}(f,g) = \mathbb{E}\left[\ln\Big(f(z)\Big)\right] - \left(\ln\left(L\left(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2; \mathbf{y}, \mathbf{X}\right)\right) - p^*\right)$$

is approximately unbiased.

- Since the first expectation is a constant with respect to model selection,

$$\text{AIC} = 2p^* - 2\ln\left(L\left(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2; \mathbf{y}, \mathbf{X}\right)\right)$$

  along with the multiple of 2 was proposed by Akaike historically.

- Bayesian information criterion

$$\text{BIC} = p^* \ln n - 2\ln\left(L\left(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2; \mathbf{y}, \mathbf{X}\right)\right)$$

  Similar to AIC, but penalises the number of parameters more severely, since n is the number of observations. Having a small value of BIC means better.

- For variable/model selection purposes, there is no clear choice between

  AIC and BIC.

- BIC is asymptotically consistent as a selection criterion, that is, given a set of models that include the true model, the probability of that BIC will select the correct model approaches 1 as $n \to \infty$, which is not the case for AIC, which tends to choose models that are too complex as $n \to \infty$.

- However, for finite samples, BIC often chooses models that are too simple, because of the heavy penalty on complexity.

Q: So what should we use for variable selection? Do you notice any problem?

- If there are k regressors, there are

$$2^k$$

  possible multiple regression models in total if assume the intercept is needed.

- Traditionally, the following procedures were popular

  Forward selection, Backward elimination, and Stepwise regression

- However, an efficient algorithm (the leaps and bounds procedure) together with modern computing power, means an exhaustive approach is possible.

- If an exhaustive approach cannot find all possible regressions efficiently, we really need to question whether we want to do what we do in the first place.

- Best subset regression finds for each

$$p \in \{0, 1, 2, \ldots, k\}$$

  the subset of size p according to a certain selection criterion.

- The following is a slightly bigger dataset,

```
> evap.df = read.table("~/Desktop/evap.txt",
+                      header = TRUE)
```

```
'data.frame':   46 obs. of  11 variables:
 $ avst : int  84 84 79 81 84 74 73 75 84 86 ...
 $ minst: int  65 65 66 67 68 66 66 67 68 72 ...
 $ maxst: int  147 149 142 147 167 131 131 134 161 169 ...
 $ avat : int  85 86 83 83 88 77 78 84 89 91 ...
 $ minat: int  59 61 64 65 69 67 69 68 71 76 ...
 $ maxat: int  151 159 152 158 180 147 159 159 195 206 ...
 $ avh  : int  95 94 94 94 93 96 96 95 95 93 ...
 $ minh : int  40 28 41 50 46 73 72 70 63 56 ...
 $ maxh : int  398 345 388 406 379 478 462 464 430 406 ...
 $ wind : int  273 140 318 282 311 446 294 313 455 604 ...
 $ evap : int  30 34 33 26 41 4 5 20 31 38 ...
```

the exact nature of study that generated the dataset is not important.

- R has an implementation of the leaps and bounds algorithm

```
> # install.packages("leaps")
> library(leaps)
```

```
> regsubsets.out = regsubsets(
+    evap~.,
+    data = evap.df,
+    nbest = 1,        # 1 best model for each p
+    nvmax = NULL,     # NULL for no limit on p
+    method = "exhaustive")
```

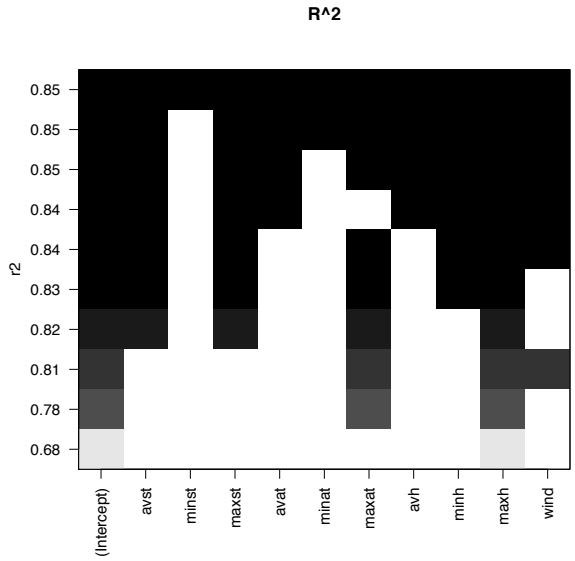
- Regressor(s) needed is/are indicated with `"*"`.

```
> summary(regsubsets.out)
```

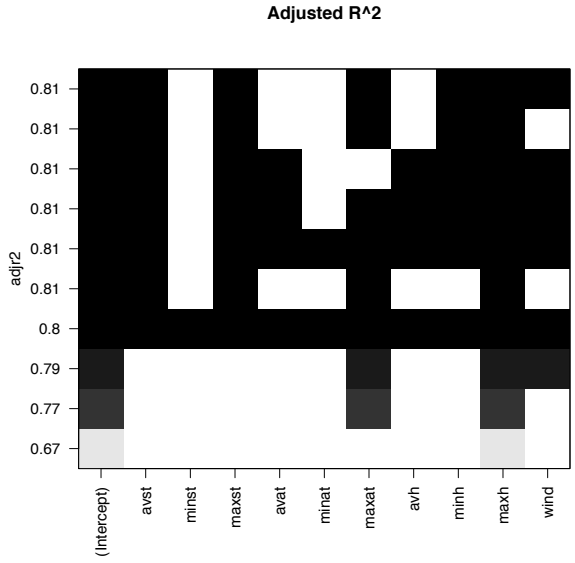
```
Selection Algorithm: exhaustive
          avst minst maxst avat minat maxat avh minh maxh wind
1  ( 1 )  " "  " "   " "   " "  " "   " "   " " " " " "  "*"  " "
2  ( 1 )  " "  " "   " "   " "  " "   "*"   " " " " " "  "*"  " "
3  ( 1 )  " "  " "   " "   " "  " "   "*"   " " " " " "  "*"  "*"
4  ( 1 )  "*"  " "   "*"   " "  " "   "*"   " " " " " "  "*"  " "
5  ( 1 )  "*"  " "   "*"   " "  " "   "*"   " " "*" "*"  "*"  " "
6  ( 1 )  "*"  " "   "*"   " "  " "   "*"   " " "*" "*"  "*"  "*"
7  ( 1 )  "*"  " "   "*"   "*"  " "   " "   "*" "*" "*"  "*"  "*"
8  ( 1 )  "*"  " "   "*"   "*"  " "   "*"   "*" "*" "*"  "*"  "*"
9  ( 1 )  "*"  " "   "*"   "*"  "*"   "*"   "*" "*" "*"  "*"  "*"
10  ( 1 ) "*"  "*"   "*"   "*"  "*"   "*"   "*" "*" "*"  "*"  "*"
```

Q: Why would RSS, R^2, R^2_{adj}, Mallow's C_p, AIC and BIC give the same output?

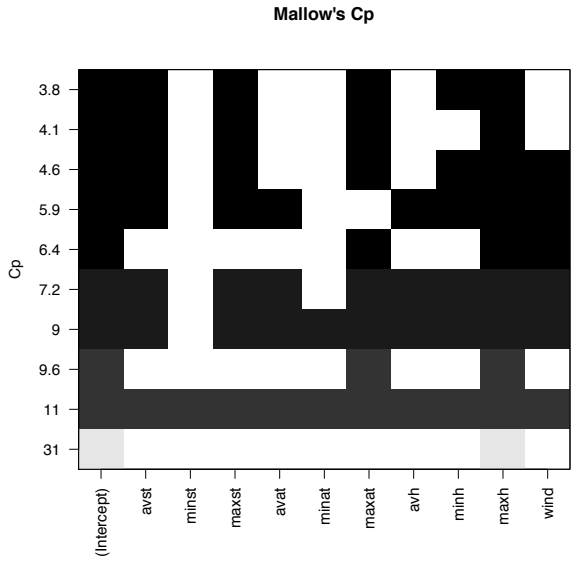
```
> plot(regsubsets.out, scale = "r2", main = "R^2")
```



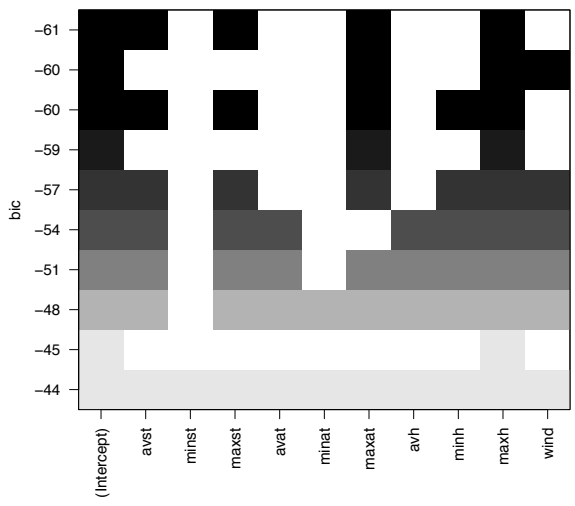

```
> plot(regsubsets.out, scale = "adjr2", main = "Adjusted R^2")
```




```
> plot(regsubsets.out, scale = "Cp", main = "Mallow's Cp")
```




```
> plot(regsubsets.out, main = "BIC")
```



- We should also check residuals for each one of those models, then using

$$R^2_{\text{adj}}, \quad \text{Mallow's } C_p, \quad t\text{-test}, \quad \text{partial } F\text{-test}, \quad \text{AIC}, \quad \text{BIC}$$

to pick the "best" model out of the valid models.

- If some transformation on regressors is used, then

$$t\text{-test}, \quad \text{or} \quad \text{partial } F\text{-test},$$

should NOT be used, since they only work on nested models.

- If the response needs to be transformed, we need redo the variable selection!
- The validity of the model is a priority especially when the goal is to explain.
- For predictive models, the modern approach is very different!

1. First of all, we tends to have plenty of data, and powerful computers, so

$$\text{data splitting, cross-validation or bootstrapping}$$

2. By keeping a subset of the predictors and discarding the rest, it leads us to a model that is interpretable and possibly has a lower MSE than the full model.

- However, it is a discrete process, i.e.

$$\text{variables are either kept or discarded}$$

so it often does not leads to a smaller MSE than the full model.

- Also for predictive models, we don't care about interpretability of the model.

Q: Can we drop half of a variable from the full model?

- This can be achieved if we deviate from the traditional least squares estimate

$$\hat{\boldsymbol{\beta}} = \underset{\mathbf{b}}{\operatorname{argmin}} \left\{ (\mathbf{y} - \mathbf{X}\mathbf{b})^{\mathrm{T}} (\mathbf{y} - \mathbf{X}\mathbf{b}) \right\} = \underset{\mathbf{b}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k} x_{ij} b_j \right)^2 \right\}$$

which have the smallest variance amongst all linear unbiased estimators.

- But modern approaches argue having unbiased estimators is not necessary!

- Shrinkage methods are more continuous in nature, often lead to estimators that are biased but don't suffer as much from high variability.
- Ridge regression shrinks the regression coefficients by imposing a penalty

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \underset{\mathbf{b}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \hat{b}_0 - \sum_{j=1}^{k} x^*_{ij} b_j \right)^2 + \lambda \sum_{j=1}^{k} b_j^2 \right\}$$

and effectively "dropping" variables partly to achieve a small MSE.

Q: Why is this penalty of sum of squares of the partial slope parameters useful?

- Here $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage.
- Notice $\mathbf{X}^*$ is different from the original $\mathbf{X}$, where $\mathbf{X}^*$ has no unit column for the b_0, and are centred and scaled to have zero mean and unit variance, i.e.

$$x^*_{ij} = \frac{x_{i,j+1} - \bar{x}_{j+1}}{s_{j+1}} \qquad \text{for} \quad j = 1, 2, \ldots, k$$

Q: Have you seem something like in other courses?

- According to Lagrange's multiplier, the minimisation problem

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \underset{\mathbf{b}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \hat{b}_0 - \sum_{j=1}^{k} x_{ij}^* b_j \right)^2 + \lambda \sum_{j=1}^{k} b_j^2 \right\}$$

  can be shown to be equivalent to

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \underset{\mathbf{b}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \hat{b}_0 - \sum_{j=1}^{k} x_{ij}^* b_j \right)^2 \right\}$$
$$\text{subject to} \quad \sum_{j=1}^{k} b_j^2 \leq c_\lambda$$

- Notice the minimisation is done with respect to $\mathbf{b}$, which does not include,

$$\hat{b}_0 = \bar{y} = \frac{1}{n} \sum_{i}^{n} y_i$$

- Comparing it with least squares problem,

$$\hat{\boldsymbol{\beta}} = \underset{\mathbf{b}}{\text{argmin}} \left\{ (\mathbf{y} - \mathbf{X}\mathbf{b})^{\mathrm{T}} (\mathbf{y} - \mathbf{X}\mathbf{b}) \right\} = \underset{\mathbf{b}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k} x_{ij} b_j \right)^2 \right\}$$

  and dropping $\hat{b}_0$ since it does not affect $\mathbf{b}$, we have the following

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^{\text{ridge}} &= \underset{\mathbf{b}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \hat{b}_0 - \sum_{j=1}^{k} x_{ij}^* b_j \right)^2 + \lambda \sum_{j=1}^{k} b_j^2 \right\} \\
&= \underset{\mathbf{b}}{\text{argmin}} \left\{ (\mathbf{y} - \mathbf{X}^*\mathbf{b})^{\mathrm{T}} (\mathbf{y} - \mathbf{X}^*\mathbf{b}) + \lambda \mathbf{b}^{\mathrm{T}}\mathbf{b} \right\} \\
\implies \mathbf{0} &= -2\mathbf{X}^{*\mathrm{T}}\mathbf{y} + 2\mathbf{X}^{*\mathrm{T}}\mathbf{X}^*\mathbf{b} + 2\lambda\mathbf{b} \\
\implies \hat{\boldsymbol{\beta}}^{\text{ridge}} &= \left(\mathbf{X}^{*\mathrm{T}}\mathbf{X}^* + \lambda\mathbf{I} \right)^{-1} \mathbf{X}^{*\mathrm{T}}\mathbf{y}
\end{aligned}$$

- We will come back to this later on when we talk about SVD and PCA, and show why $\hat{\boldsymbol{\beta}}^{\text{ridge}}$ has a smaller variance in return for allowing some bias.

- Another popular shrinkage method in modern era is known as Lasso,

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \underset{\mathbf{b}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \hat{b}_0 - \sum_{j=1}^{k} x_{ij}^* b_j \right)^2 \right\}$$
$$\text{subject to} \quad \sum_{j=1}^{k} |b_j| \leq c_\lambda$$

in Lagrangian form, we have

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \underset{\mathbf{b}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \hat{b}_0 - \sum_{j=1}^{k} x_{ij}^{*} b_j \right)^2 + \lambda \sum_{j=1}^{k} |b_j| \right\}$$

- It can be shown λ or c_λ in ridge or Lasso affect the MSE and variance, and are chosen to minimise MSE for a given variance in practice.

$$\underbrace{\mathbb{E}\left[\left(Y_{n+1} - \hat{Y}_{n+1}^{*} \right)^2 \mid \cdot \right]}_{\text{mean squared error}} = \sigma^2 + \underbrace{\operatorname{Var}\left[\hat{Y}_{n+1}^{*} \mid \cdot \right]}_{\text{variance}} + \underbrace{\mathbb{E}\left[\hat{Y}_{n+1}^{*} - \mu_{n+1} \mid \cdot \right]^2}_{\text{bias}}$$

4 Nonlinear Regression

- There are many situations where the response and regressors are related through a known nonlinear function. It leads to a nonlinear regression model.
- We have largely focus on the multiple linear regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$$

but linear regression models in general include polynomial models and others

$$Y = \beta_0 + \beta_1 Z_1 + \beta_2 Z_2 + \cdots + \beta_k Z_k + \varepsilon$$

where Z_j represents any function of the regressors $X_1, X_2, \ldots, X_k$, e.g.

$$X_j^2, \qquad \sqrt{X_j}, \qquad \exp(X_j), \qquad \ln(X_j),$$

- The reason that those models are considered to be linear is that they are

$$\text{linear in the unknown parameters } \beta_j$$

for all $j = 1, 2, \ldots, k$.

- That is, the regression function m is always linear in terms of $\boldsymbol{\beta}$

$$m(\mathbf{X}, \boldsymbol{\beta}) = \mathbb{E}[Y \mid \mathbf{X}]$$

- Of course, this might not always be appropriate, e.g.

$$Y = \beta_1 e^{\beta_2 X} + \varepsilon$$

- In practice, the true nonlinear relationship might be known to be governed

$$Y = m(\mathbf{X}, \boldsymbol{\beta}) + \varepsilon$$

by a differential equation up to some unknown parameters,

$$\boldsymbol{\beta}$$

- Often an uncorrelated normally distributed random error is assumed with

$$\mathbb{E}[\varepsilon \mid \mathbf{X}] = 0 \qquad \operatorname{Var}[\varepsilon \mid \mathbf{X}] = \sigma^2$$