Lecture 1: Introduction

STAT471/571/701

Linda Zhao

11/26/2016

Logistics

Todo:

Read: Chapter 2 and 3

Today:

- Read data into R
- Summary statistics
- Displaying the data

Regression Overview

- Model specification
- LS estimates and properties
- R-squared and RSE
- Confidence intervals for coef.
- Prediction intervals
- Caution about reverse regression
- Appendices

Billion Dollar Billy Beane

Read this article

- Is Billy Beane worth the predicted 12 million dollars?
- Will a team perform better when they are paid more?

Questions

1. How does payroll relate to performance (`avgwin`)?
2. Is Billy Beane worth 12 million dollars, as argued in the article?
3. Given a team with $payroll = .84$, on average what would be the mean winning percentage and average wins?

Packages in R

R's greatest blessing (and curse) is its package ecosystem. (insert CRAN link here)

We've developed a custom package for this class - install it if you haven't yet!

```r
if(!require("devtools")) {
    install.packages("devtools")
}
devtools::install_github("stillmatic/stat471utils")
```

Our Data

Data: `ml_pay`: consists of winning records and the payroll of all 30 ML teams from 1998 to 2014. There are 162 games in each season.

- `payroll`: total pay up to 2014 in billion dollars
- `avgwin`: average winning percentage for the span of 1998 to 2014
- `p2014`: total pay in 2014
- `X2014`: number of games won in 2014.
- `X2014.pct`: percent of games won in 2014. We only need one of the two from above

Load the data

Our custom tooling makes this easy for you:

```r
library(stat471)
datapay <- stat471::ml_pay
```

Try `str(datapay)` now! \ You can also do `help(ml_pay)` to see the documentation

Getting help

R has some good built-in help functions:

```
??read.csv
help(read.csv)  # The argument needs to be a function or da
apropos("read") # List all the functions with "read" as par
args(read.csv)  # List all the arguments to read.csv
str(read.csv)
```

And of course, Google is your friend.

Understanding the data

Before you do any analysis it is always wise to take a quick look at the data to spot anything abnormal.

Find the structure the data and have a quick summary

```
class(datapay)
str(datapay) # make sure the variables are correctly defin
summary(datapay) # a quick summary of each variable
```

Understanding the data

Get a table view

```
fix(datapay)     # need to close the window to move on
View(datapay)
```

Look at the first six rows or first few rows

```
head(datapay) # or tail(datapay)
head(datapay, 2) # first two rows
```

Understanding the data

Find the size of the data or the numbers or rows and columns

```
dim(datapay)
```

Get the variable names

```
names(datapay) # It is a variable by itself
```

Subsetting the data

We often want to work with a subset which includes relevant variables:

```r
datapay[1,1] # payroll for team one
datapay$payroll # call variable payroll
datapay[, 1] # first colunm
datapay[, c(1:3)] # first three columns
```

Or update what the columns are named, e.g. rename "Team.name.2014" to "team"

```r
names(datapay)[3]="team"
```

Missing values

In R, missing values are treated as “NA”, a special type. This is how we can check:

```r
sum(is.na(datapay))
```

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

Note that we cleaned this dataset beforehand!

Exploratory data analysis - stats

We will concentrate on three variables: `payroll`, `avgwin` and `team`

```
mean(datapay$payroll)
```

```
## [1] 1.23844
```

```
sd(datapay$payroll)
```

```
## [1] 0.4269762
```

```
quantile(datapay$payroll)
```

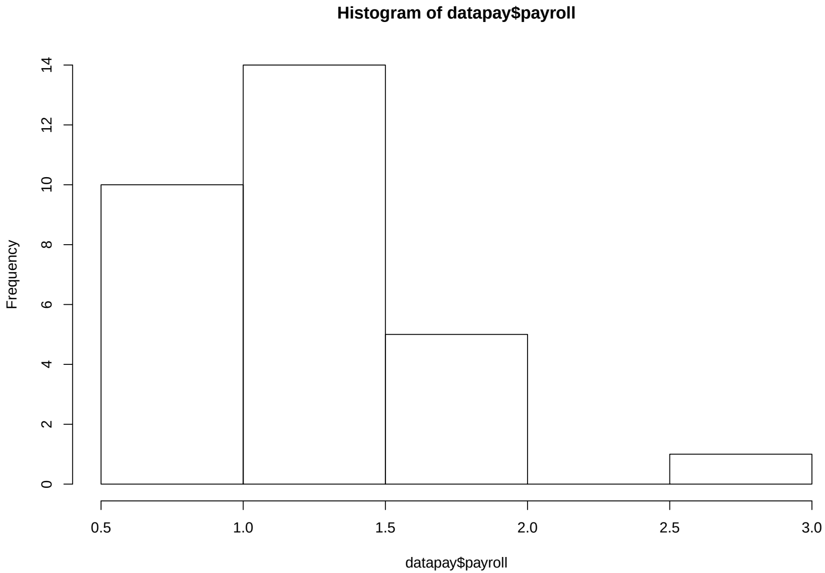
```
##        0%       25%       50%       75%      100%
## 0.6678019 0.9721439 1.1451168 1.4265249 2.7032482
```

Find the team name with the max payroll

```
datapay$team[which.max(datapay$payroll)]
```

Exploratory data analysis - plots

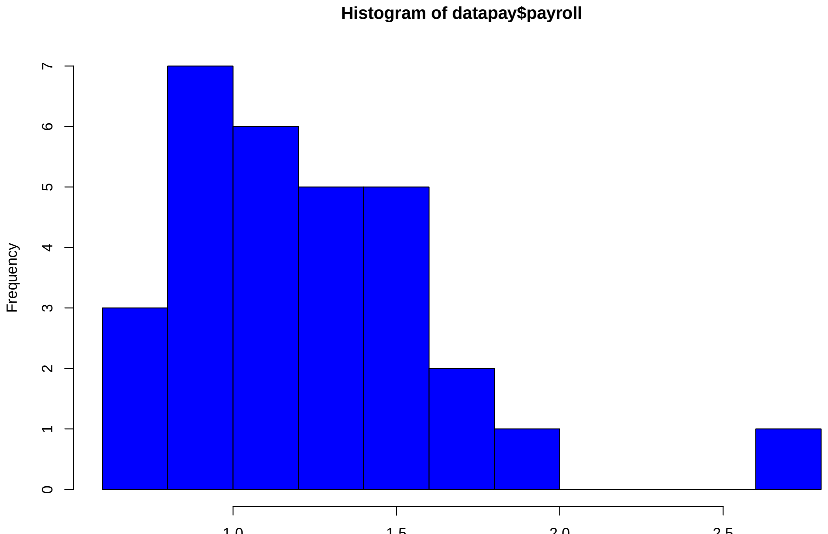
```
hist(datapay$payroll, breaks=5)
```



Exploratory data analysis - plots

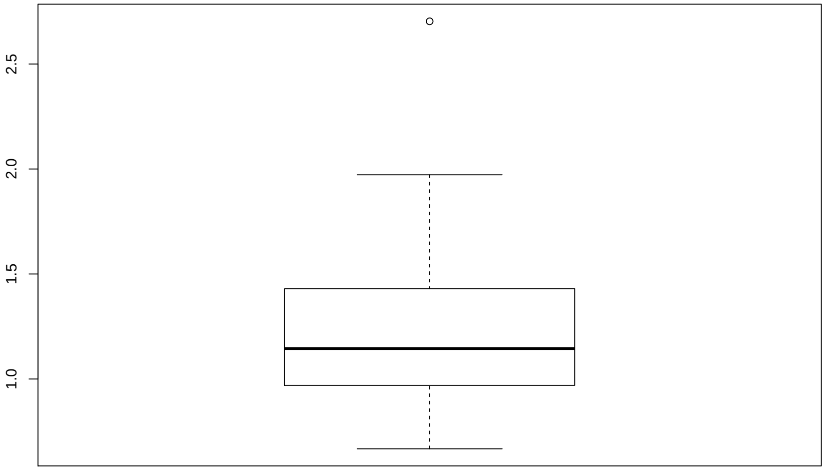
Larger number of classes to see the details

```
hist(datapay$payroll, breaks=10, col="blue")
```



Exploratory data analysis - plots

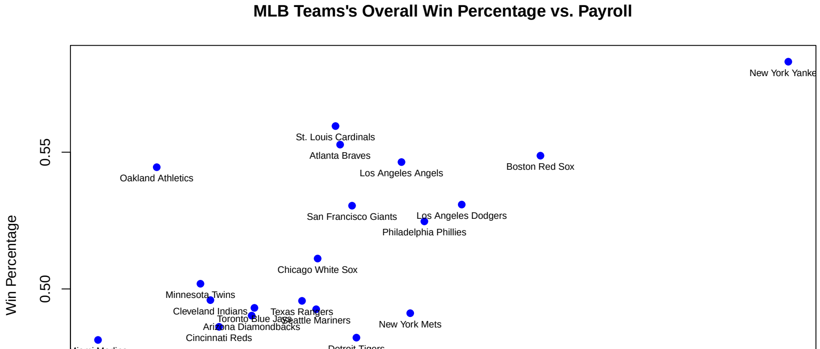
```
boxplot(datapay$payroll)
```



Exploratory data analysis - plots

Explore the relationship between payroll (x) and avgwin (y), with a scatter plot.

```
## $ind
## integer(0)
##
## $pos
## integer(0)
```



Linear models

We fit linear regressions in R with the following syntax:

```
myfit0 <- lm(avgwin~payroll, data=datapay)    # avgwin is g
```

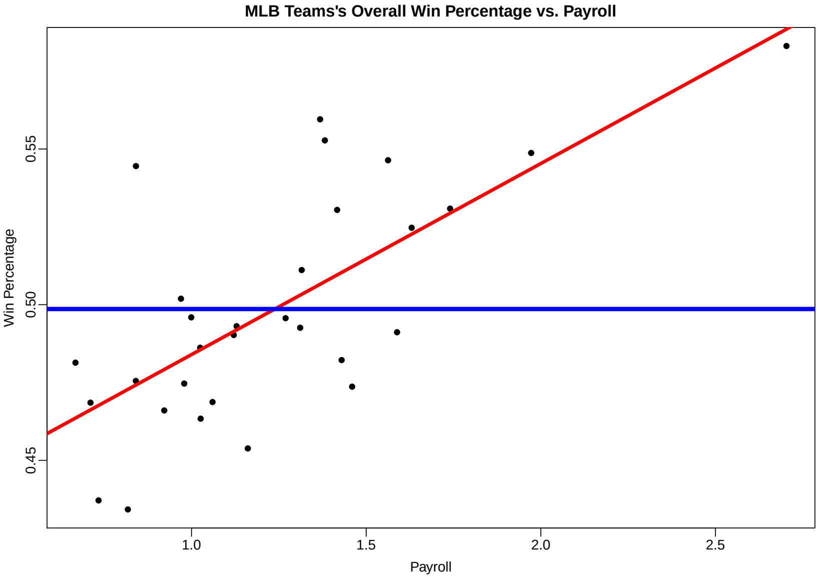
Linear models - evaluating

```r
names(myfit0)# it outputs many statistics
str(myfit0) # myfit0 is a list

summary(myfit0)   # it is another object that is often used
results <- summary(myfit0)
names(results)
str(results)
```

Linear models - plotting

Scatter plot with the LS line added



Your thoughts

Here is how the article concludes that Beane is worth as much as the GM in Red Sox.

By looking at the above plot, Oakland A's win pct is more or less same as that of Red Sox, so based on the LS equation, the team should have paid 2 billion.

Do you agree?

Linear models - mathematical evaluation

RSS and RSE:

```
myfit0 <- lm(avgwin~payroll, data=datapay)
RSS <- sum((myfit0$res)^2) # residual sum of squares
RSE <- sqrt(RSS/myfit0$df) # residual standard error
TSS <-sum((datapay$avgwin-mean(datapay$avgwin))^2) # total
Rsquare <- (TSS-RSS)/TSS    # Percentage reduction of the 

Rsquare <- (cor(datapay$avgwin, myfit0$fitt))^2 # Square o
```

We can also get these results from the summary output

```
RSE=summary(myfit0)$sigma
Rsquare=summary(myfit0)$r.squared
```

Inference

Under the model assumptions:

1. y_i is i.i.d, and normally distributed
2. the mean of y given x is linear
3. the var of y does not depend on x

THEN we have nice properties about the LS estimates b_1, b_0.

- t intervals and t tests for beta's
- use RSE to estimate the true sigma.

Subset analysis

```
data1=datapay[, -(21:37)] # take X1998 to X2014 out
data2=data1[, sort(names(data1)[21-37])] # sort the col nar
names(data2)
```

```
##  [1] "avgwin"    "p1998"     "p1999"     "p2000"     "p2
##  [6] "p2002"     "p2003"     "p2004"     "p2005"     "p2
## [11] "p2007"     "p2008"     "p2009"     "p2011"     "p2
## [16] "p2013"     "p2014"     "payroll"   "team"      "X1
## [21] "X1999.pct" "X2000.pct" "X2001.pct" "X2002.pct" "X2
## [26] "X2004.pct" "X2005.pct" "X2006.pct" "X2007.pct" "X2
## [31] "X2009.pct" "X2010.pct" "X2011.pct" "X2012.pct" "X2
## [36] "X2014.pct"
```

Confidence intervals

```
summary(myfit0)  # Tests and CI for the coefficients
confint(myfit0)  # Pull out the CI for the coefficients
```

Create new data, and feed it to our trained model, to find a 95% CI:

```
new_team <- data.frame(payroll=c(.841))
CImean <- predict(myfit0, new_team, interval="confidence",
```

Prediction intervals

```
CIpred <- predict(myfit0, new_team, interval="prediction",
CIpred
```

A 95 prediction interval varies from 0.474 to 0.531, for a team like the Oakland A's.

But their `avewin` is 0.5445, and somewhat outside of this interval …

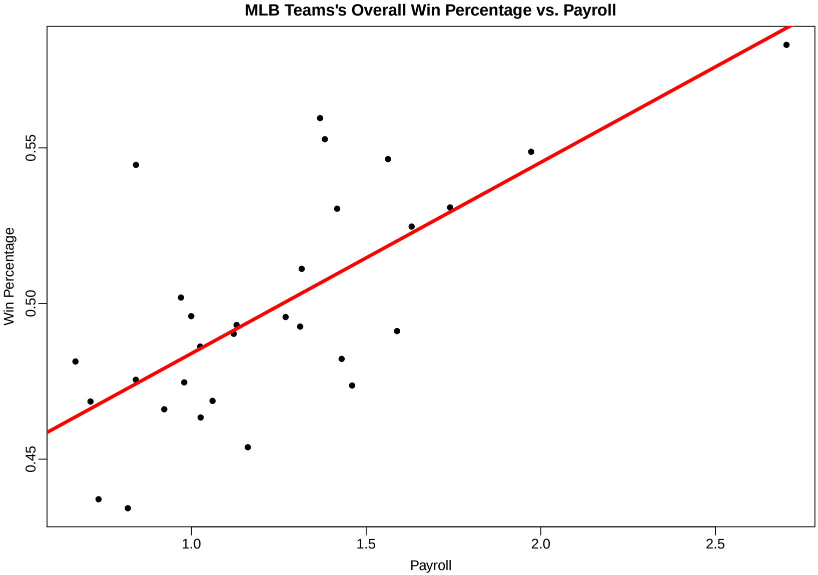
Prediction intervals - 99%

A 99% prediction interval would contain .5445!

```
CIpred_99 <- predict(myfit0, new_team, interval="prediction
CIpred_99
```

Reverse Regression

Recall this scatterplot:



Reverse Regression

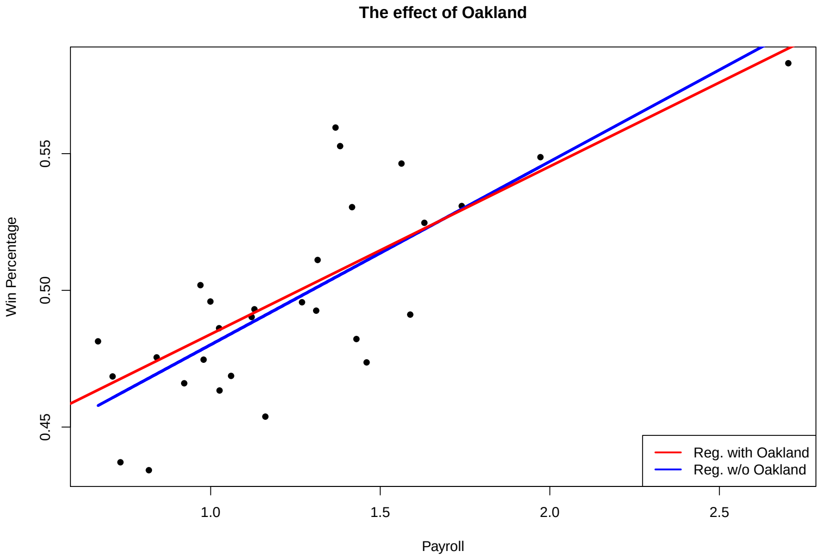
Add LS line:

```
##
## Call:
## lm(formula = payroll ~ avgwin, data = datapay)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76735 -0.18705 -0.03714  0.16633  0.78427
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.7784     0.7697  -3.610  0.00118 **
## avgwin        8.0563     1.5396   5.233 1.47e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
## Residual standard error: 0.309 on 28 degrees of freedom
## Multiple R-squared:  0.4944,  Adjusted R-squared:  0.4763
```

Reverse Regression

We may want to get the LS equation w/o Oakland first.

Scatter plot with both LS lines



Reverse Regression

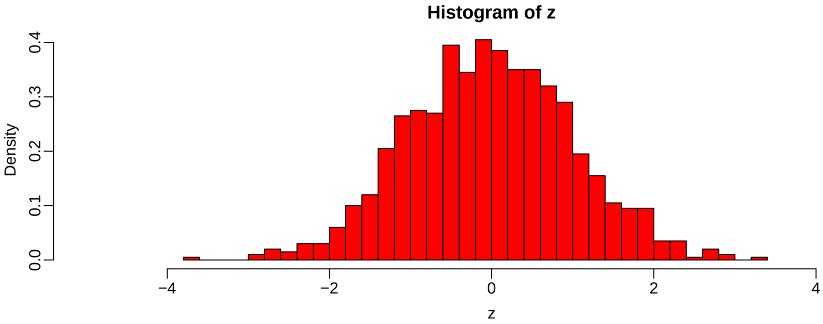
```r
subdata1 <- datapay[-19,]
myfit3 <- lm(avgwin~payroll, data=subdata1)
summary(myfit3)

plot(subdata$payroll, subdata$avgwin, pch=16,
     xlab="Payroll", ylab="Win Percentage",
     main="The effect of Yankees")
abline(myfit3, col="blue", lwd=3)
abline(myfit0, col="red", lwd=3)
legend("bottomright", legend=c("Reg. All", "Reg. w/o Yankee
       lty=c(1,1), lwd=c(2,2), col=c("red","blue"))
```

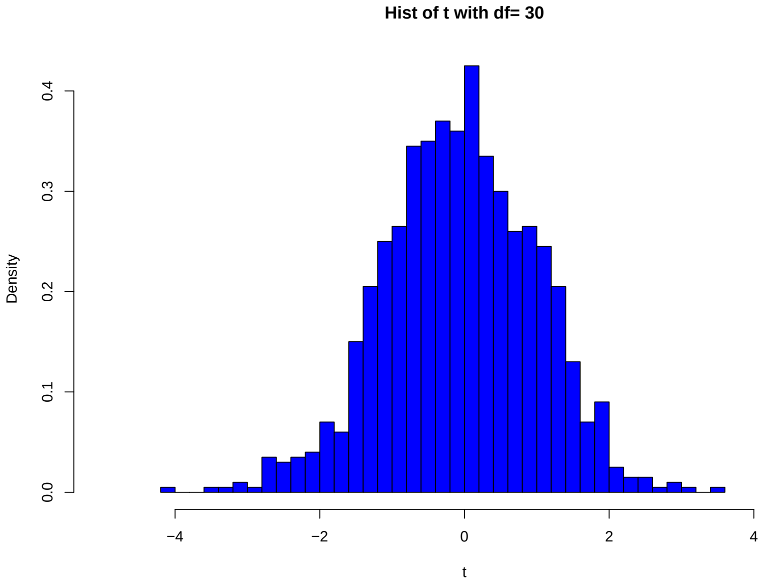
End

Appendix 1: z vs t

Difference between z and t with $df = n$. The distribution of z is similar to that t when df is large, say 30.



Appendix 1



Appendix 2: R-squared

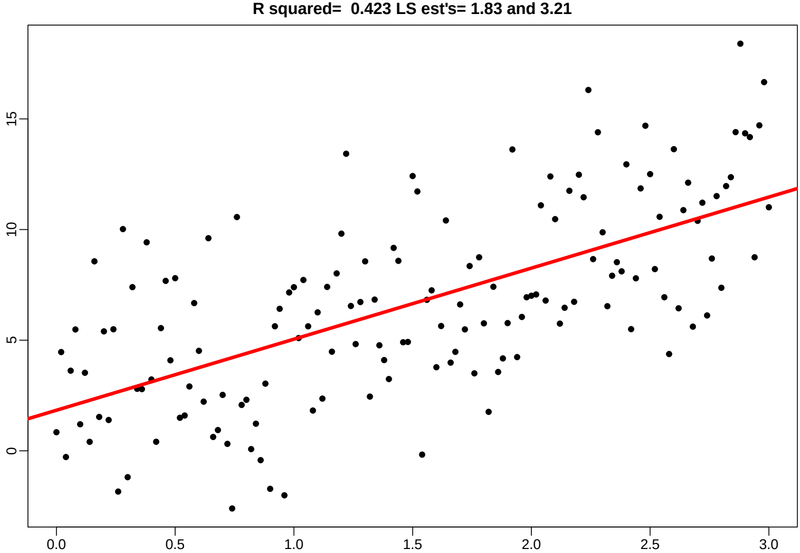
Case I: a perfect model between X and Y but it is not linear

R-squared=.837 here y=x^3 with no noise!

Appendix 2

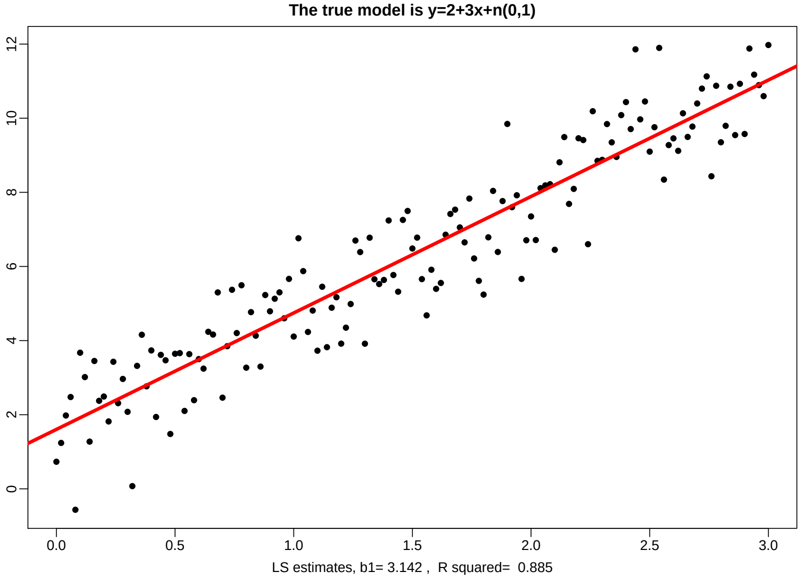
Case II: a perfect linear model between X and Y but with noise.

Here $y = 2 + 3x + \epsilon$, ϵ is iid $N(0, \sigma = 9)$.



Appendix 2

Case III: Same as that in Case II, but lower the sigma (σ) to 1



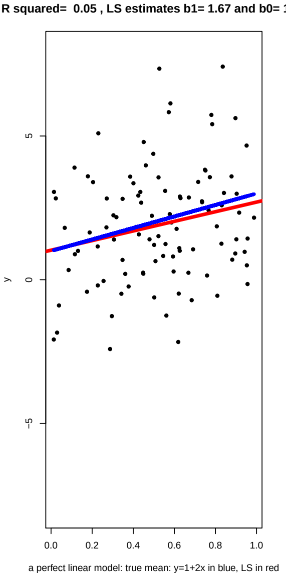
Appendix 3

What do we expect to see even all the model assumptions are met?

1. Variability of the LS estimates β's
2. Variability of the R^2
3. Model diagnosis: through residuals

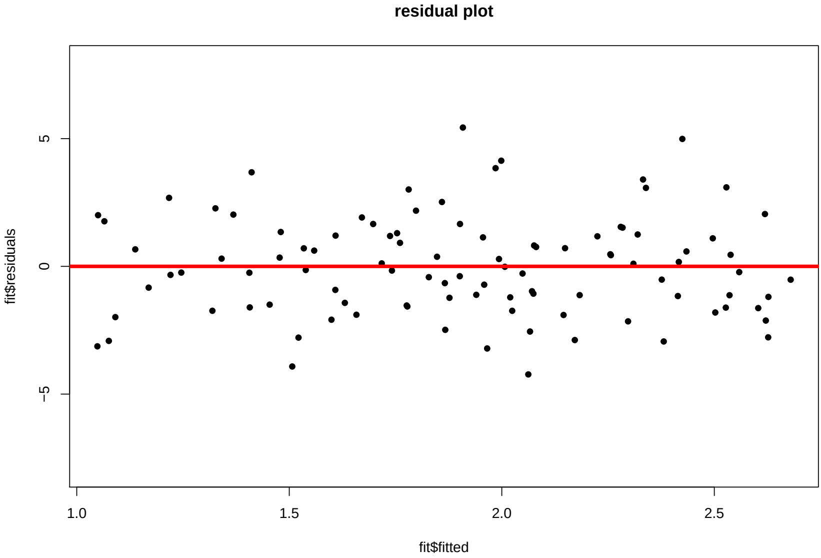
We answer this through simulation

Appendix 3


R squared= 0.05 , LS estimates b1= 1.67 and b0= 1
y
5
0
−5
0.0
0.2
0.4
0.6
0.8
1.0
a perfect linear model: true mean: y=1+2x in blue, LS in red

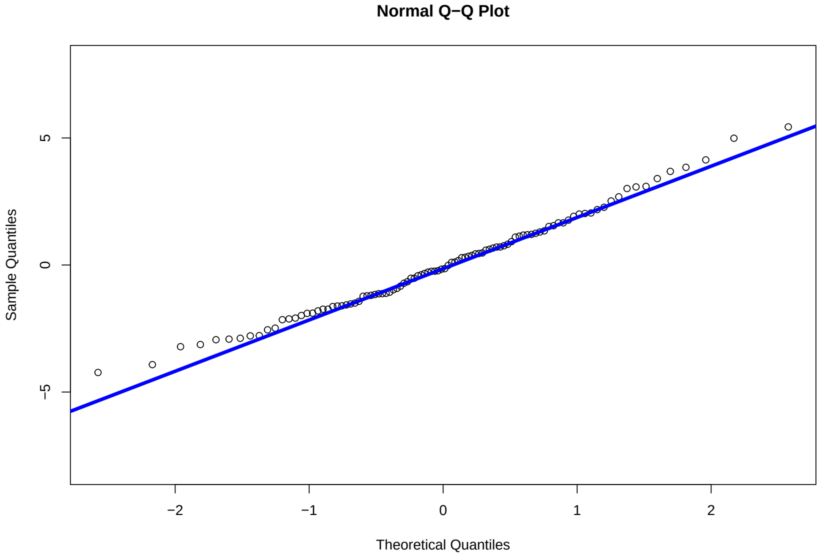
Appendix 3

Residual plot:



Appendix 3

Check normality:


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles