REASEARCH METHODS (COMP 5003) SPRING 2020

HOMEWORK #2

REPORT OF LAB WORK

BY

GBENGA S. AGUNSOYE

AND

UBOHO VICTOR

SUBMITTED TO

DR. LIN LI

DEPARTMENT OF COMPUTER SCIENCE

PRAIRIE VIEW A&M UNIVERSITY

TEXAS.

Lab#1

1c. Yes, the mean is close to expectation because in is distributed over 0.5

1d. Summary shows the min, 1st quartile, median, mean, 3rd quartile and the maximum

Min. 1st Qu. Median Mean 3rd Qu. Max.

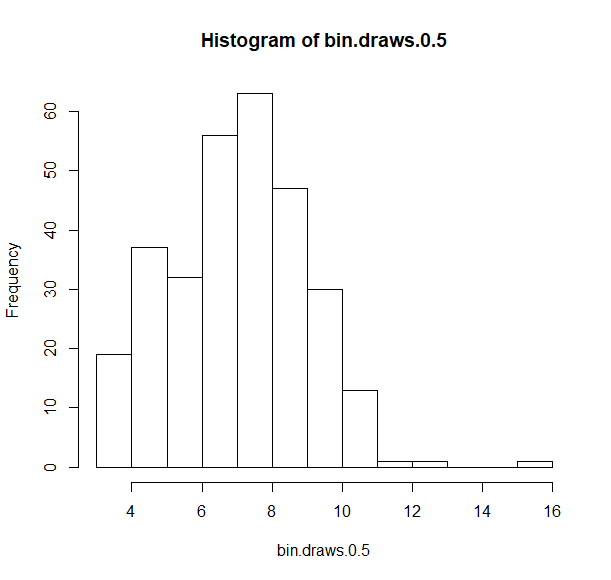
3.00 6.00 8.00 7.51 9.00 16.00

1e. Yes, the result was formatted differently. It is now treated as a character vector

Length Class Mode

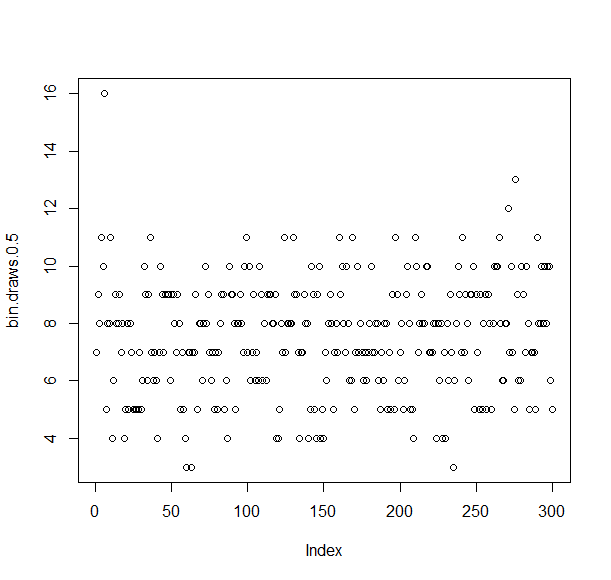
300 character character

2a.

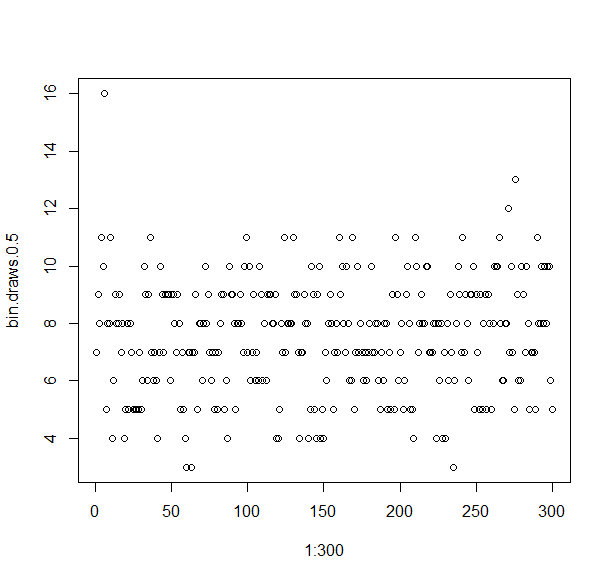


2b. Yes, it does.

2c. The plot describes the value in vector bin.draws.0.5 against its index



2d. Yes it does.



3a.

1. The probability that a standard normal random variable exceeds 1.644854 is 0.04999996. This is approximately 0.05. Hence, yes, it is close.

Lab #2

2a. Dimension = 97x9. Pros.dat has 97 rows and 9 columns.

First 6 rows and all columns

lcavol lweight age lbph svi lcp gleason pgg45 lpsa

1 -0.5798185 2.769459 50 -1.386294 0 -1.386294 6 0 -0.4307829

2 -0.9942523 3.319626 58 -1.386294 0 -1.386294 6 0 -0.1625189

3 -0.5108256 2.691243 74 -1.386294 0 -1.386294 7 20 -0.1625189

4 -1.2039728 3.282789 58 -1.386294 0 -1.386294 6 0 -0.1625189

5 0.7514161 3.432373 62 -1.386294 0 -1.386294 6 0 0.3715636

6 -1.0498221 3.228826 50 -1.386294 0 -1.386294 6 0 0.7654678

> pros.dat[1:6, ]

Last 6 rows and all columns

pros.dat[92:97, ]

lcavol lweight age lbph svi lcp gleason pgg45 lpsa

92 2.532903 3.677566 61 1.3480732 1 -1.386294 7 15 4.129551

93 2.830268 3.876396 68 -1.3862944 1 1.321756 7 60 4.385147

94 3.821004 3.896909 44 -1.3862944 1 2.169054 7 40 4.684443

95 2.907447 3.396185 52 -1.3862944 1 2.463853 7 10 5.143124

96 2.882564 3.773910 68 1.5581446 1 1.558145 7 80 5.477509

97 3.471966 3.974998 68 0.4382549 1 2.904165 7 20 5.582932

2b

head(pros.dat)

lcavol lweight age lbph svi lcp gleason pgg45 lpsa

1 -0.5798185 2.769459 50 -1.386294 0 -1.386294 6 0 -0.4307829

2 -0.9942523 3.319626 58 -1.386294 0 -1.386294 6 0 -0.1625189

3 -0.5108256 2.691243 74 -1.386294 0 -1.386294 7 20 -0.1625189

4 -1.2039728 3.282789 58 -1.386294 0 -1.386294 6 0 -0.1625189

5 0.7514161 3.432373 62 -1.386294 0 -1.386294 6 0 0.3715636

6 -1.0498221 3.228826 50 -1.386294 0 -1.386294 6 0 0.7654678

tail(pros.dat)

lcavol lweight age lbph svi lcp gleason pgg45 lpsa

92 2.532903 3.677566 61 1.3480732 1 -1.386294 7 15 4.129551

93 2.830268 3.876396 68 -1.3862944 1 1.321756 7 60 4.385147

94 3.821004 3.896909 44 -1.3862944 1 2.169054 7 40 4.684443

95 2.907447 3.396185 52 -1.3862944 1 2.463853 7 10 5.143124

96 2.882564 3.773910 68 1.5581446 1 1.558145 7 80 5.477509

97 3.471966 3.974998 68 0.4382549 1 2.904165 7 20 5.582932

2c. pros.dat has names assigned to the columns while the rownames are numbers 1 – 97

|  |
| --- |
| rownames(pros.dat)  [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15" "16" "17" "18"  [19] "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30" "31" "32" "33" "34" "35" "36"  [37] "37" "38" "39" "40" "41" "42" "43" "44" "45" "46" "47" "48" "49" "50" "51" "52" "53" "54"  [55] "55" "56" "57" "58" "59" "60" "61" "62" "63" "64" "65" "66" "67" "68" "69" "70" "71" "72"  [73] "73" "74" "75" "76" "77" "78" "79" "80" "81" "82" "83" "84" "85" "86" "87" "88" "89" "90"  [91] "91" "92" "93" "94" "95" "96" "97"  > colnames(pros.dat)  [1] "lcavol" "lweight" "age" "lbph" "svi" "lcp" "gleason" "pgg45" "lpsa" |
|  |
| |  | | --- | | > | |

2d. pros.dat.sub = as.matrix(pros.dat[,c('lcavol', 'lweight')])

> dim(pros.dat.sub)

[1] 97 2

|  |
| --- |
| head(pros.dat.sub)  lcavol lweight  1 -0.5798185 2.769459  2 -0.9942523 3.319626  3 -0.5108256 2.691243  4 -1.2039728 3.282789  5 0.7514161 3.432373  6 -1.0498221 3.228826 |
| Yes, R automatically assigned column names to pros.dat.sub |
| |  | | --- | | > | |

2e. pros.dat.sub <- transform(pros.dat.sub, lcadensity = lweight / lcavol)

> head(pros.dat.sub)

lcavol lweight lcadensity

1 -0.5798185 2.769459 -4.776424

2 -0.9942523 3.319626 -3.338817

3 -0.5108256 2.691243 -5.268418

4 -1.2039728 3.282789 -2.726631

5 0.7514161 3.432373 4.567873

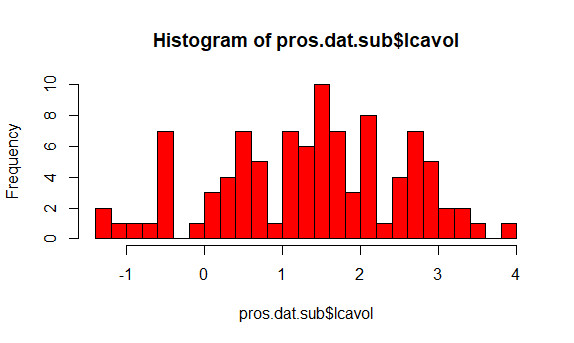
6 -1.0498221 3.228826 -3.075593

2f. lcadensity= pros.dat.sub[, 'lcadensity']

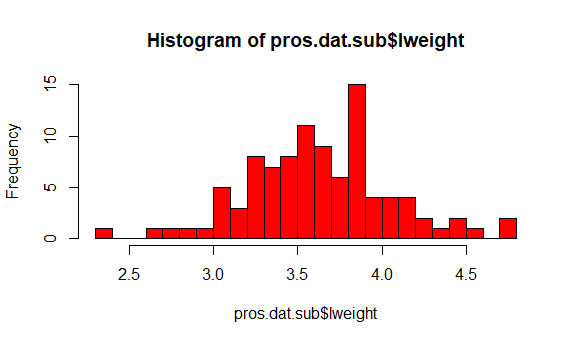
pros.dat = cbind(pros.dat,lcadensity)

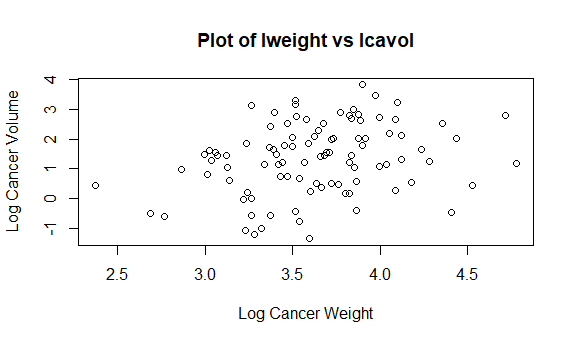
head(pros.dat)

3a. The lcavol histogram shows that the data is normally distributed



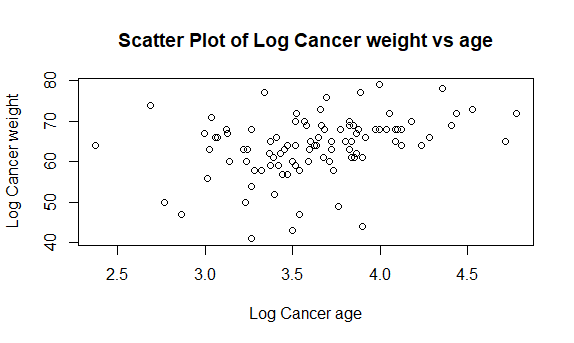
Histogram of lweight is also normally distributed



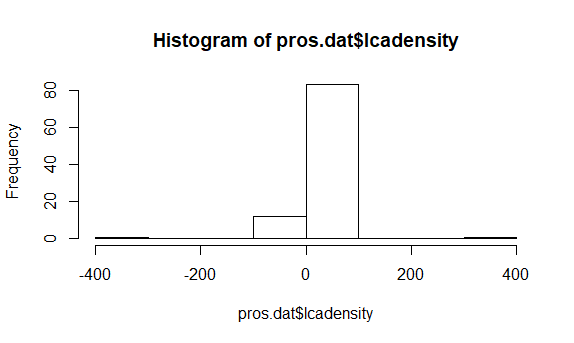


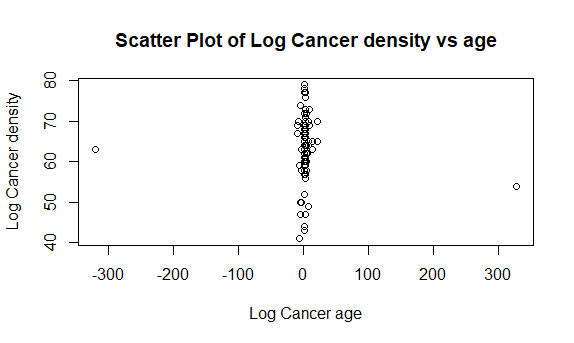
The Scatterplot shows a non linear relationship between lweight and lcavol.

3b. Scatter plot of Log cancer weight vs age



3c. histogram of log cancer density and scatter plot of density vs age





3d. pros.dat = pros.dat[,-10]

5a. pros.dat.svi.sd = vector(length=ncol(pros.dat))

> i = 1

> pros.dat.svi.sd[i] = sd(pros.dat.svi[,i])

> pros.dat.svi.sd[i]

[1] 0.6707867

5b. pros.dat.no.svi.sd = vector(length=ncol(pros.dat))

> i = 1

> pros.dat.no.svi.sd[i] = sd(pros.dat.no.svi[,i])

> pros.dat.no.svi.sd[i]

[1] 1.068573

5c.

|  |
| --- |
| cat("pros.dat.svi.sd","\t", "pros.dat.no.svi.sd", "\n")  pros.dat.svi.sd pros.dat.no.svi.sd  > for( i in 1:ncol(pros.dat)){  + pros.dat.svi.sd[i] = sd(pros.dat.svi[,i])  + pros.dat.no.svi.sd[i] = sd(pros.dat.no.svi[,i])  + cat(pros.dat.svi.sd[i],"\t", pros.dat.no.svi.sd[i], "\n")  + }  0.6707867 1.068573  0.3275689 0.4479291  7.871588 7.310591  1.354526 1.478201  0 0  1.04529 1.03794  0.6015852 0.7088414  25.73445 25.06676  0.9251229 0.9646403 |
|  |
| |  | | --- | | > | |

5d. The lpsa variable exhibited large mean difference of 31.17794 relative to standard deviation of 0.66769

Lab3

2a. sapply(pros.dat, mean)

lcavol lweight age lbph svi lcp gleason

1.3500096 3.6289427 63.8659794 0.1003556 0.2164948 -0.1793656 6.7525773

pgg45 lpsa

24.3814433 2.4783869

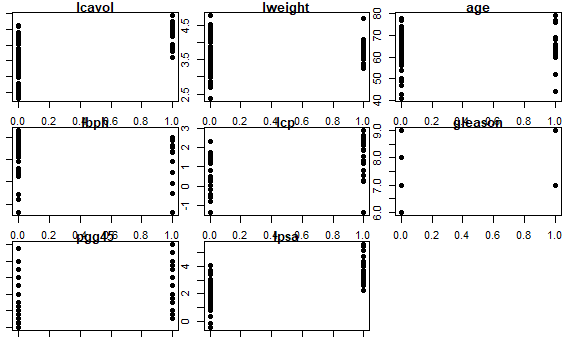
2b. par(mfrow=c(3, 3))

myPlot<-function(index){ plot(pros.dat2[,index] ~

pros.dat$svi,

main=names(pros.dat2)[index],pch=16,xlab="SVI",ylab=names(pros.dat2)[index])}

lapply(1:8,FUN=myPlot)



2c.

3a. Rio is a list object and the dimension is 11538 x 12. Each row represents each athlete in the competition and provides the class of prizes received. The information variable is missing some data.

3b.

Lab4

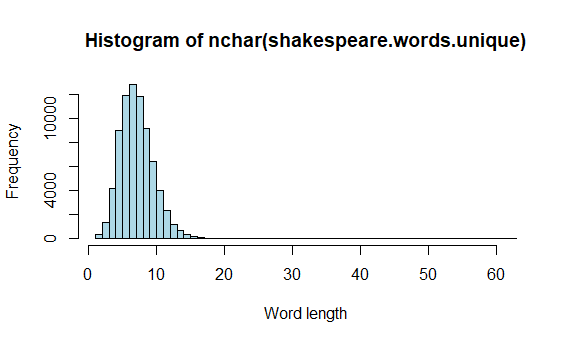
2a. There are 147838 lines. There are 70 chars in the longest line. An average of 37.6 characters per line. There are 17743 empty lines(zero characters).

2b. The length of shakespeare.lines with non-empty string is 130095

2c. Shakespeare.all has 5690994 has compared to 5560900 in shakespeare.lines. The difference 130094, is the number of non-empty strings. It does makes sense.

2d. There are 1370375 word in shakespeare.words There are 76171 unique words in shakespeare.words.unique

2e.



2f. No, I do not recognize any because they are old Shakespeare words with modern day equivalent. The fourth longest word is Zwagger with equivalent Swagger, which mean “force by blustering language, bully”

3a. No. There are 76172 unique words in shakespeare.wordtab.

“thou” appeared 4522 times

“rumour” appeared 7 times

“gloomy” appeared 3 times

“assassination” appeared 1 times

3b. once = 41843

Twice = 10756

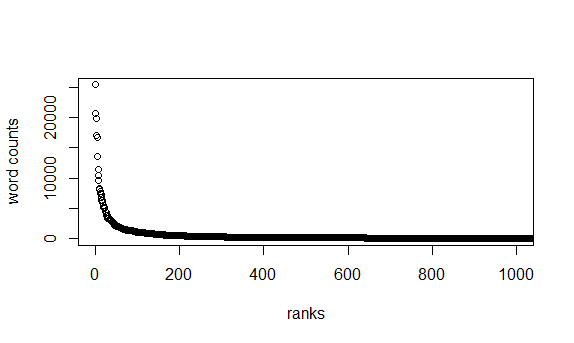
At least 10 times = 8187

More 100 times = 975

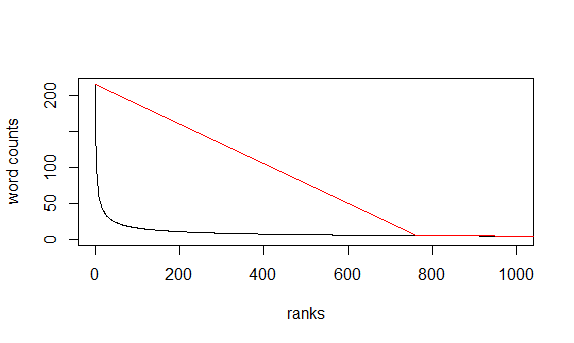
3c. 1st = “”, 2nd = “the”, and 3rd = “I”

3d. All empty strings removed from shakespeare.words and sorted

3e.



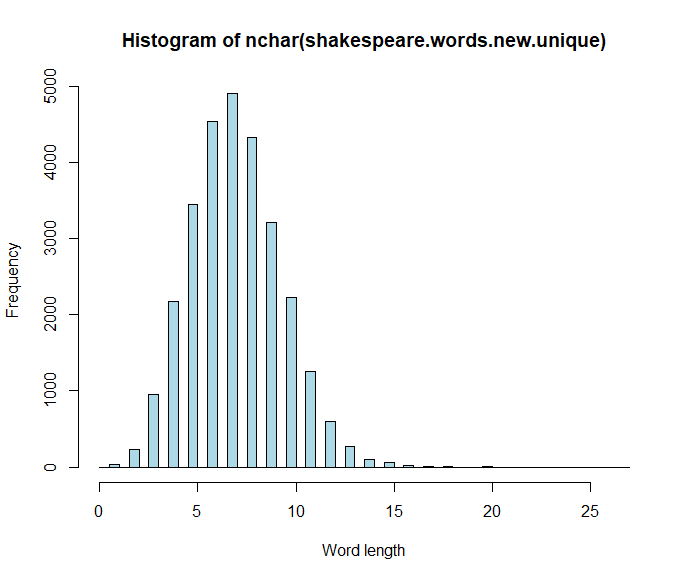
The plot shows Zipf’s law in action.



4a. Regular expression performed to split words and punctuation

4b. We observed that shakespeare.words.new has more words when compared to shakespeare.words because more words ensue from identifying punctuations and treating them separately. Also the length of shakespeare.wordtab.new is over 100% less than shakespeare.wordtab, because all occurrences of a word are group together and counted. This made the size of the table smaller in shakespeare.wordtab.new.

4c.



The histogram is spatial, shows normal distribution of word-length and there is a shift to the right.

The top 5 longest words in Shakespeare.words.new.unique and their indices are

12966 zwounds

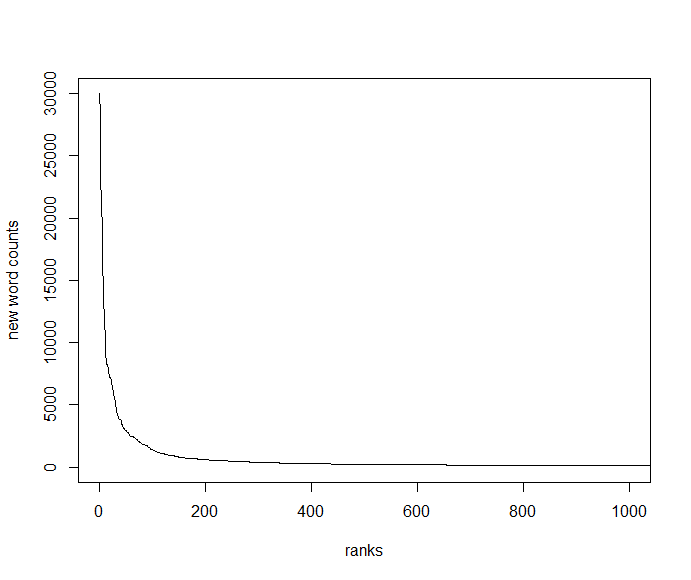
18839 zwaggered

12768 zounds

12498 zone

19903 zodiacs

4d.



The plot of new word count against their rank shows that Zipf’s law holds.

5a. From the new output of Shakespeare.lines, empty spaces in each string is removed as seen in line 19 to 23.

5b. “THE SONNETS” in Shakespeare.lines 23, 67 and “VENUS AND ADONIS” in 66, 128535, and 128555.

Lab #6

1a. Huber(1) = 1; Huber(4) = 7

1b. huber(3,2) = 8; huber(3,4) = 9

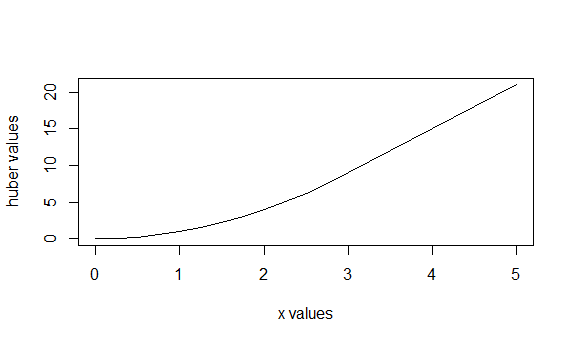
1c. huber(3) = 5

1d. huber(a=1, x=3) = 5; huber(1, 3) = 1;

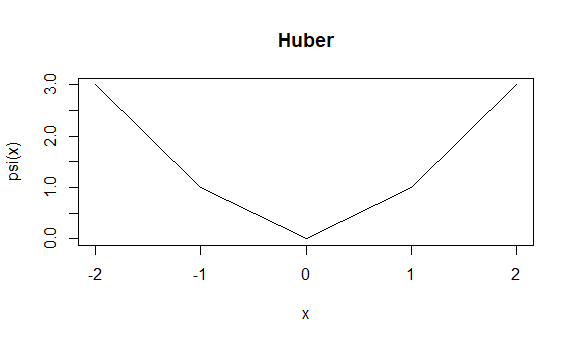
The default parameters are overwritten in the first call and parameters are explicitly specified, hence the order does not matter. However in the second call to Huber function, the parameters supply are treated based on function definition, hence, x=1 and a=3 is implied.

1e. Huber vectorized

Challenge

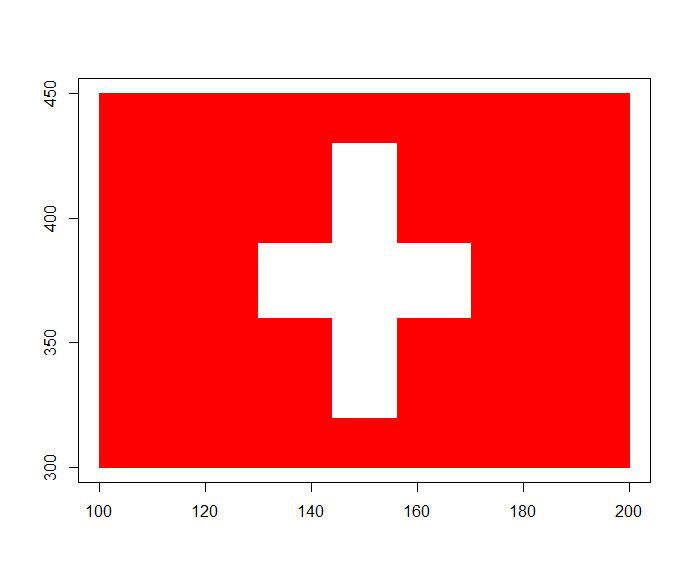
`

2a.



2c.

Flag of Switzerland



3a. The value of x.squared remained unchanged 999.

3b. local variable a in the Huber.mod function remained unchanged as well for different iterations.

3c. Result of Huber.sloppy is 6.75 because the global variable a was absorbed and used as parameter, since the local variable is not explicitly defined. For variant global “a” variable, the result changes and setting a to a string, returns the length of the string to the function when called.

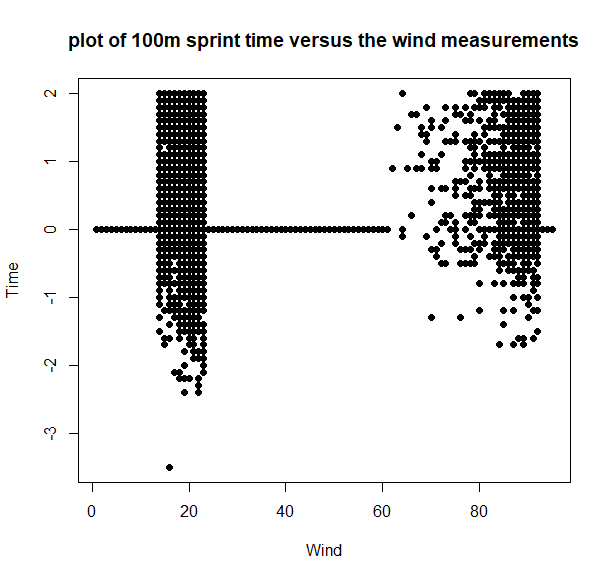
3d. Global definition of variable a is overwritten by the local definition with the <- assignment operator.

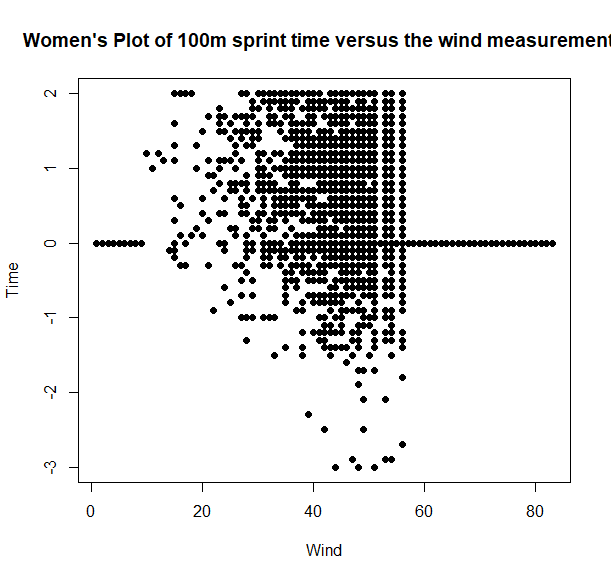
3e. The variable b was initialized and assigned to a at run time. Also, the assignment operator, gives the new variable b a global and scope. Making it both accessible locally and globally

#LAB 8

1d. NA was introduced for fields that are empty

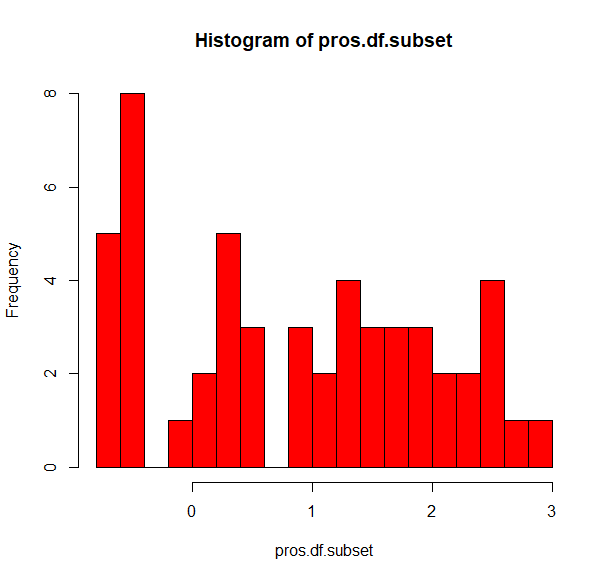
1e.



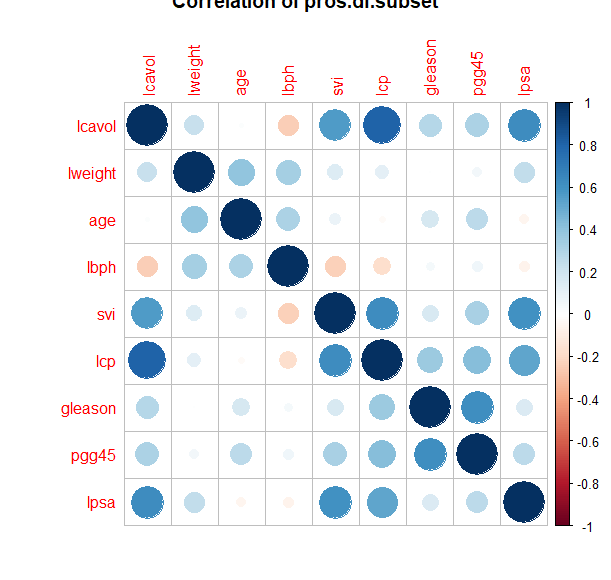


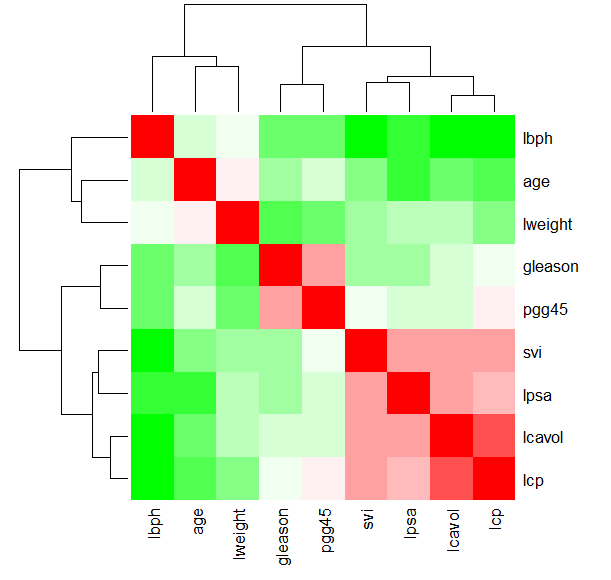
Lab #9

1a.



1b.

  
1b. Challenge



1c. Although the coefficients are close, they different for each model as shown

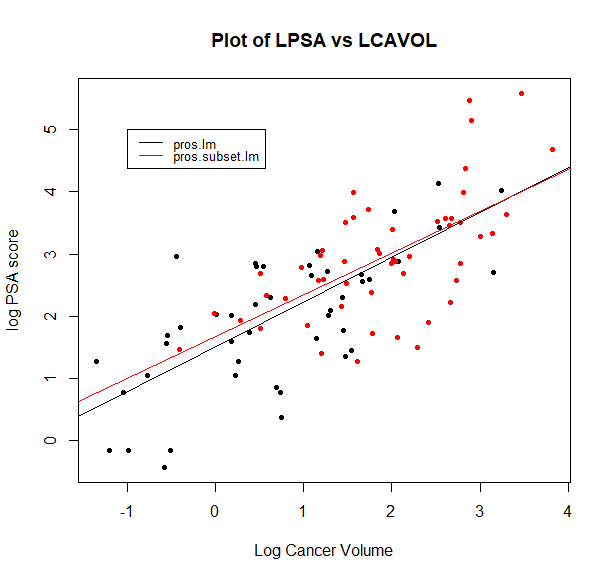
coef(pros.lm) - (Intercept) pros.df$lcavol

1.5072975 0.7193204

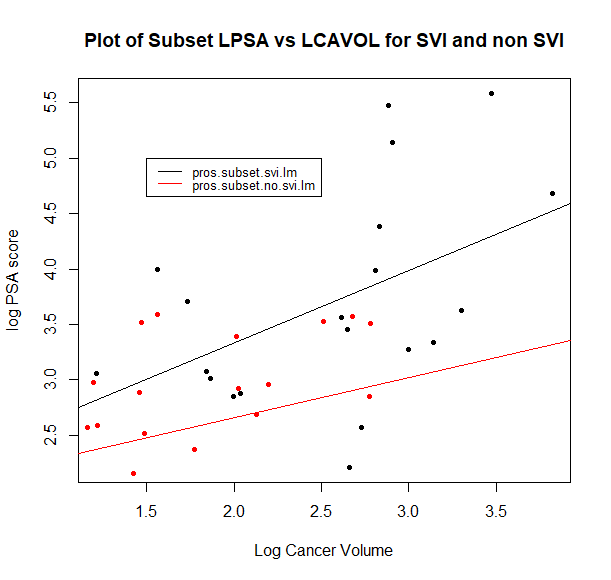
coef(pros.subset.lm) - (Intercept) all\_pros.df.subset$lcavol

1.6695707 0.6725807

1d.

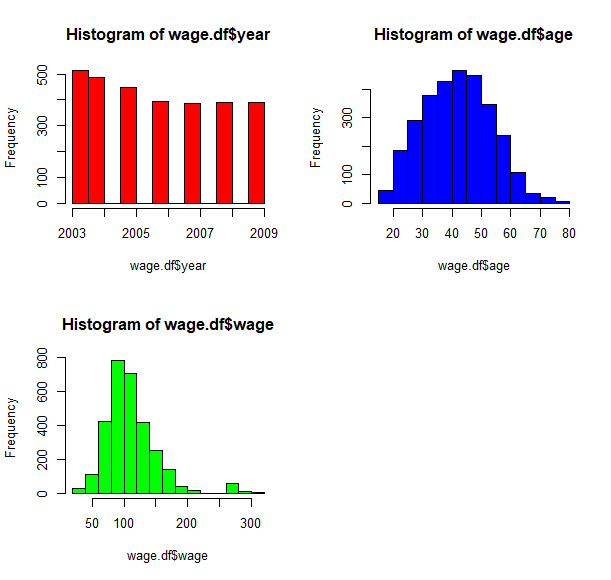


1e.



2a. The dataset was not as described in the lab question, hence, we performed some preprocessing of extra column alongside removal of descriptions provided at the top of the dataset.

2b.



2c.

3a. The coefficient for year and age are;

coef(wage.lm)

(Intercept) wage.df$year wage.df$age

-2318.5309186 1.1968236 0.6992032

The standard display by the model summary:

Estimate Std. Error

wage.df$year 0.3685

wage.df$age 0.0647

The p-value is shown below:

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -2318.5309 739.1385 -3.137 0.00172 \*\*

wage.df$year 1.1968 0.3685 3.247 0.00118 \*\*

wage.df$age 0.6992 0.0647 10.808 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

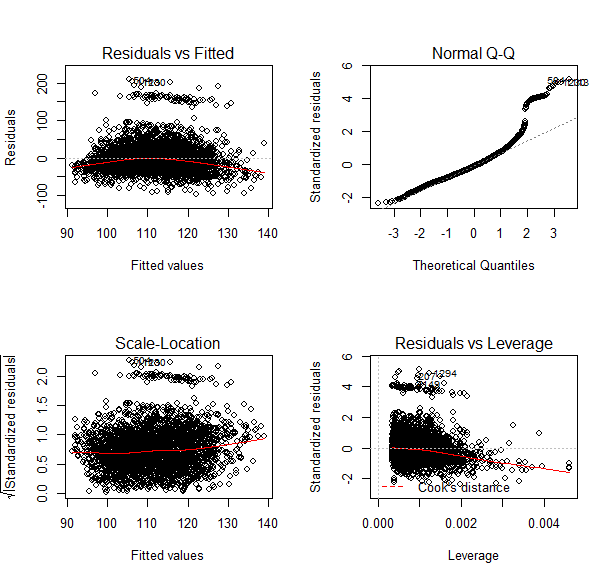
Residual standard error: 40.86 on 2997 degrees of freedom

Multiple R-squared: 0.04165, Adjusted R-squared: 0.04101

F-statistic: 65.12 on 2 and 2997 DF, p-value: < 2.2e-16

3b. wage.sum is a list. We convert it to a dataframe to extract the standard error.

3c. Yes, there are group of points away from the main bulk along the x-ais as shown below:



Yes, from normal Q-Q the standardized residuals lie along the line y=x

Challenge: Histogram of age is close.

3d. Summary of the new model for wage less than 250

Call:

lm(formula = wage.df.lt250$wage ~ wage.df.lt250$year + wage.df.lt250$age,

data = wage.df.lt250)

Residuals:

Min 1Q Median 3Q Max

-90.001 -21.690 -2.905 18.518 112.226

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -2.126e+03 5.715e+02 -3.721 0.000203 \*\*\*

wage.df.lt250$year 1.102e+00 2.850e-01 3.866 0.000113 \*\*\*

wage.df.lt250$age 5.656e-01 4.986e-02 11.345 < 2e-16 \*\*\*

---

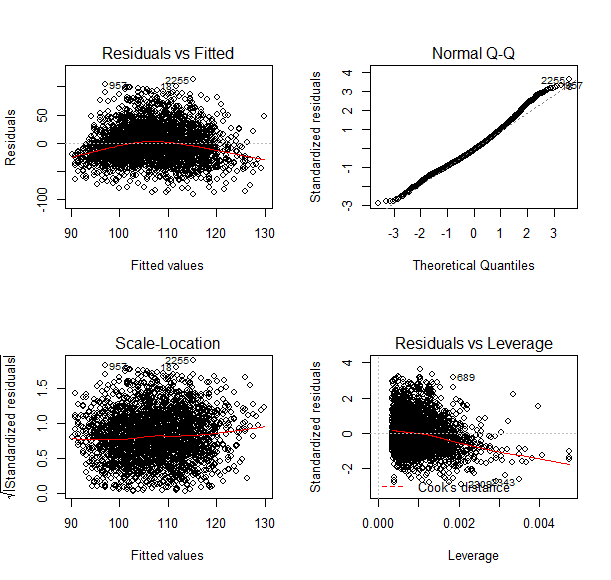
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 31.17 on 2918 degrees of freedom

Multiple R-squared: 0.04812, Adjusted R-squared: 0.04747

F-statistic: 73.75 on 2 and 2918 DF, p-value: < 2.2e-16

Problems from the previous plot is resolved here



3e. Need data to predict on

4a. summary(wage.glm)

Call:

glm(formula = as.factor(wage.high$wage) ~ wage.high$year + wage.high$age,

family = "binomial", data = wage.high)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.62154 0.08071 0.15233 0.26155 0.54395

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.280e+03 1.111e+03 -1.153 0.249

wage.high$year 6.415e-01 5.541e-01 1.158 0.247

wage.high$age -4.873e-02 8.377e-02 -0.582 0.561

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 18.654 on 78 degrees of freedom

Residual deviance: 16.188 on 76 degrees of freedom

AIC: 22.188

Number of Fisher Scoring iterations: 8

4b. with category variable education, we have the following summary

Call:

glm(formula = as.factor(wage.high$wage) ~ wage.high$year + wage.high$age +

wage.high$education, family = "binomial", data = wage.high)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.21118 0.00002 0.00003 0.05688 0.80391

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -8.255e+02 1.280e+04 -0.064 0.949

wage.high$year 4.232e-01 5.941e-01 0.712 0.476

wage.high$age -5.688e-02 8.573e-02 -0.663 0.507

wage.high$education3. Some College 1.650e+00 1.673e+04 0.000 1.000

wage.high$education4. College Grad -1.773e+01 1.274e+04 -0.001 0.999

wage.high$education5. Advanced Degree 1.177e+00 1.340e+04 0.000 1.000

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 18.654 on 78 degrees of freedom

Residual deviance: 11.949 on 73 degrees of freedom

AIC: 23.949

Number of Fisher Scoring iterations: 20

4c.

LAB #14

2a. We observed that there are 67 training observations and 30 test observations

3a. Both train and test set contained 1500 records