**CSC591**: Foundations of Data Science  
HW2-3-R-project: Combined R mini-projects for topics covered across HW2 and HW3.

Released: **10/06/15**Due: **10/17/16 (23:55pm);** (One day late: -25%; -100% after that).

Student Name: Anush Bonam  
Student ID:200110634

**Notes – Read carefully**

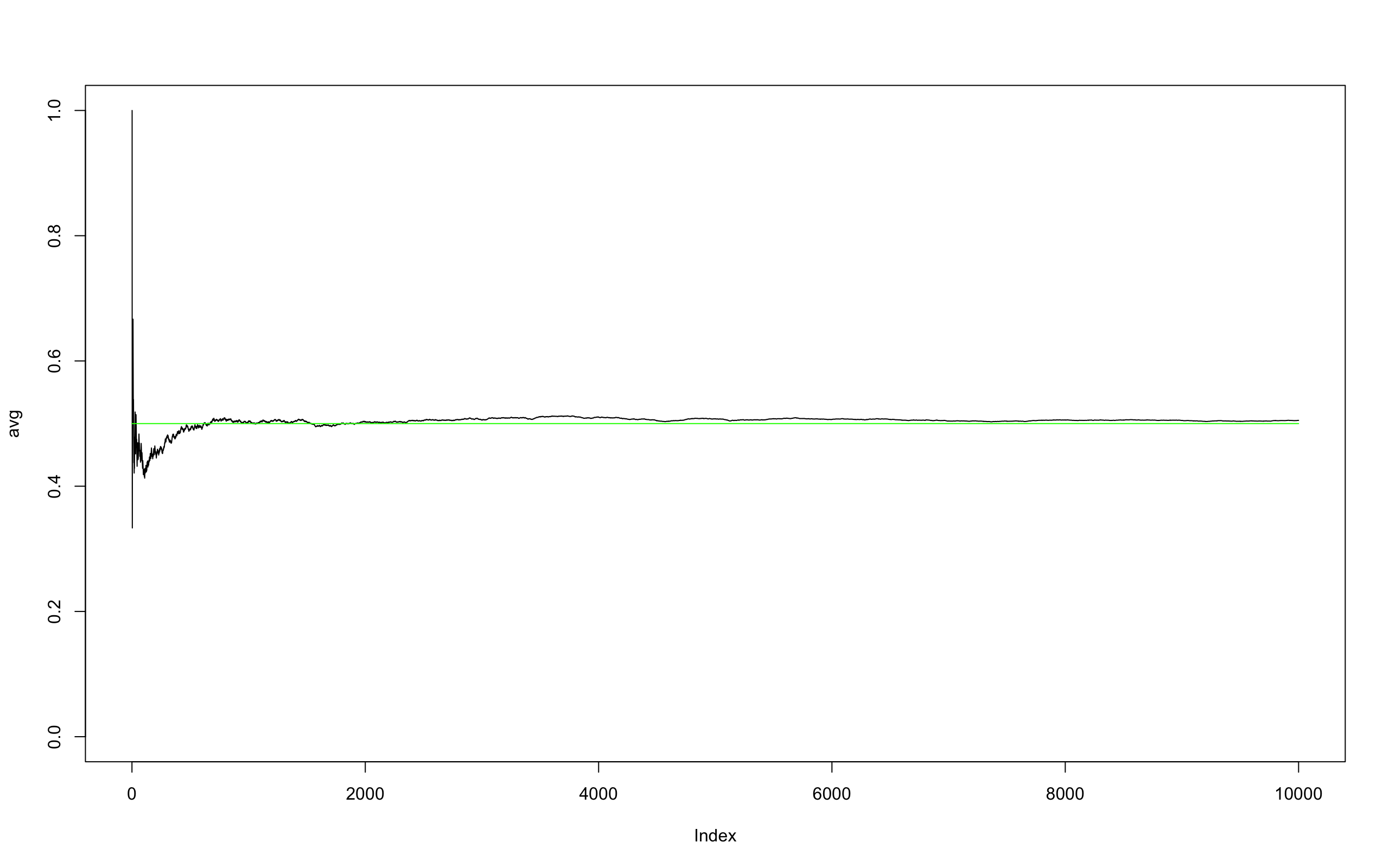
Submit Single zip file (Name it studentID\_hw23\_R.zip); follow naming convention for all files. Submit code with reasonable documentation; if needed individuals will be asked to run their code (no modifications are allowed after submission). **Don’t change the folder structure given to you.**

* Zip file should include one folder and a pdf.
  + Folder name ‘code’:
    - One .txt file, showing instructions on how to run R/python programs; libraries used, etc.; Also answer any question specific items (for example, if question asks you to submit a plot, then include it here). **Call it Readme.txt**
    - **installPackages.R:** (provided to you).In the first line, add all the R packages your code will need.
    - One R file that includes functions for each question. **Call it utils.R**
    - **solution\_hw23.R:** (provided to you) One R file that calls each of these functions. This has already been provided to you. Don’t change the function format in this file. We will run this file to evaluate your results.
  + One PDF file that includes plots and corresponding explanation for each question. Name it **studentID\_hw23.pdf**

**Q1**. Law of large numbers (5 points)  
See the Wikipedia page (<https://en.wikipedia.org/wiki/Law_of_large_numbers)>. Write R program to simulate an experiment involving flipping a coin 10,000 times. Generate a plot (similar to the one shown in Wikipedia page) showing the cumulative proportion of tails.

Note: Write a function CoinFlip() to demonstrate this simulation and save in utils.R.

Sol:

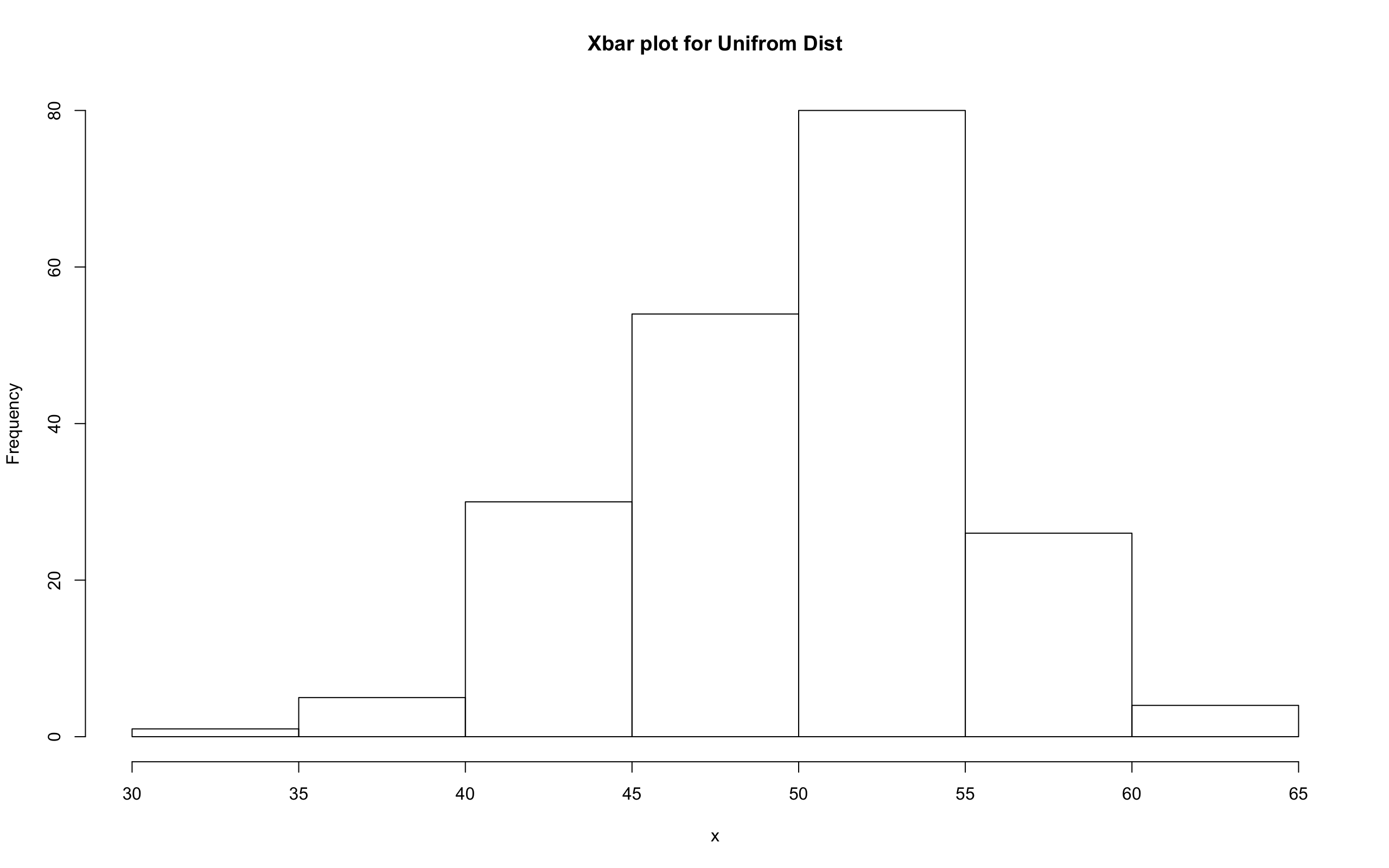


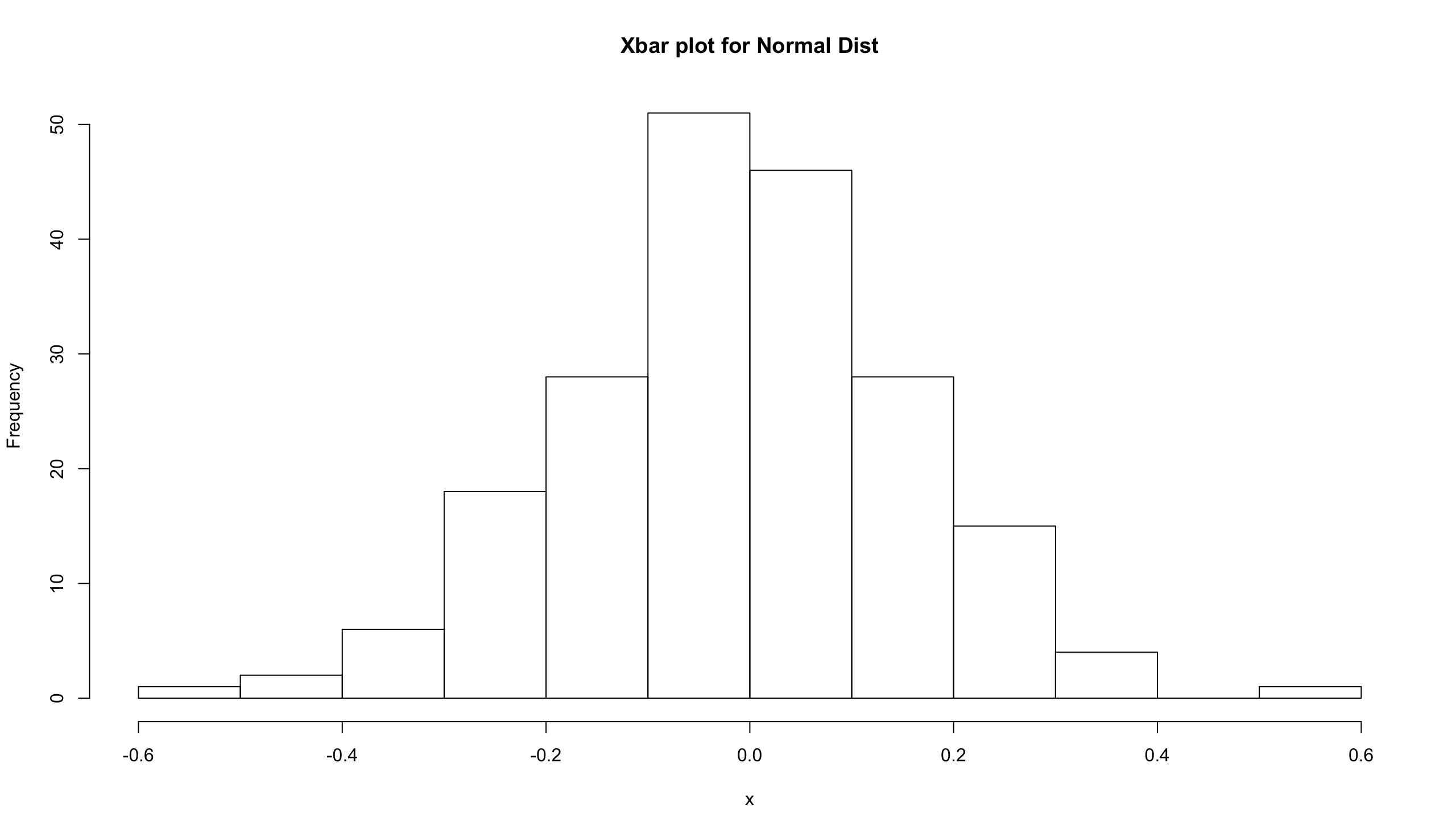
**Q2.** CLT (10 points)  
Implement an R program to demonstrate CLT. At the minimum, your project should implement the following elements.

1. Should take **inputs**: (1) type of population distribution (e.g., uniform, normal, etc), (2) sample size (~30), (3) number of samples (>100).
2. **Output**: Show the plot of sampling distribution

Note: Write a function CLT() (add all arguments necessary) to demonstrate the above question and save it in utils.R.

Sol: Function CLT() implemented in R and saved in utils.R file available in zip file.





**Q3**. Using “hw23R-Advertising.csv” data (from the book) answer (a) and (b) (2x5 = 10 points)

1. Fit simple linear regression (separately) for each covariate. Provide scatter plots with fitted regression line. Which covariate provides best prediction?

Note: Write a function SLR() to answer Q3(a) and save it in utils.R.

1. Fit multiple linear regression model for the data. Show resulting equation. How do you compare the β’s obtained with this model with corresponding β’s found in (a).

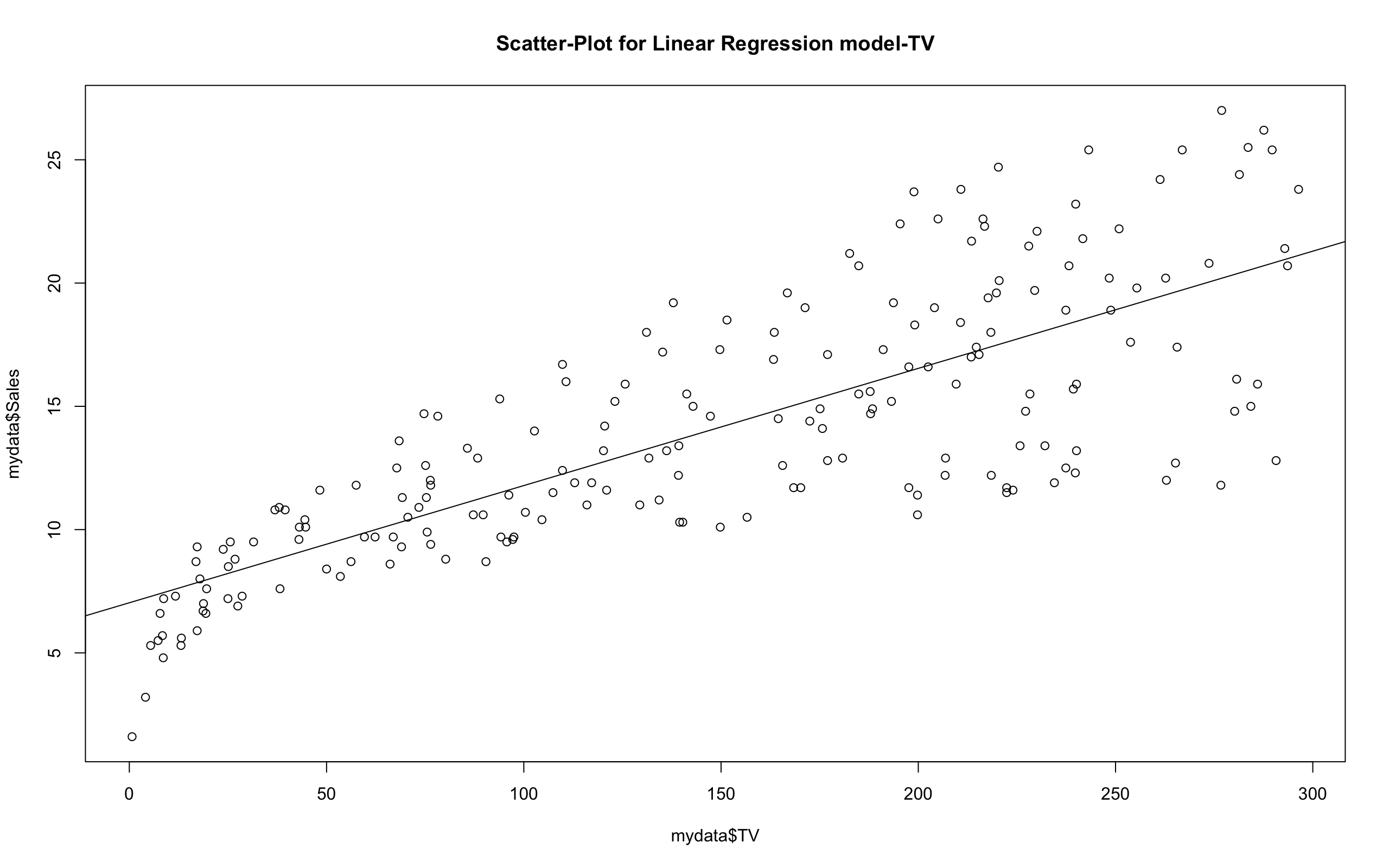
Note: Write a function MLR(),to answer Q3(b) and save it in utils.R.

Sol: (a)

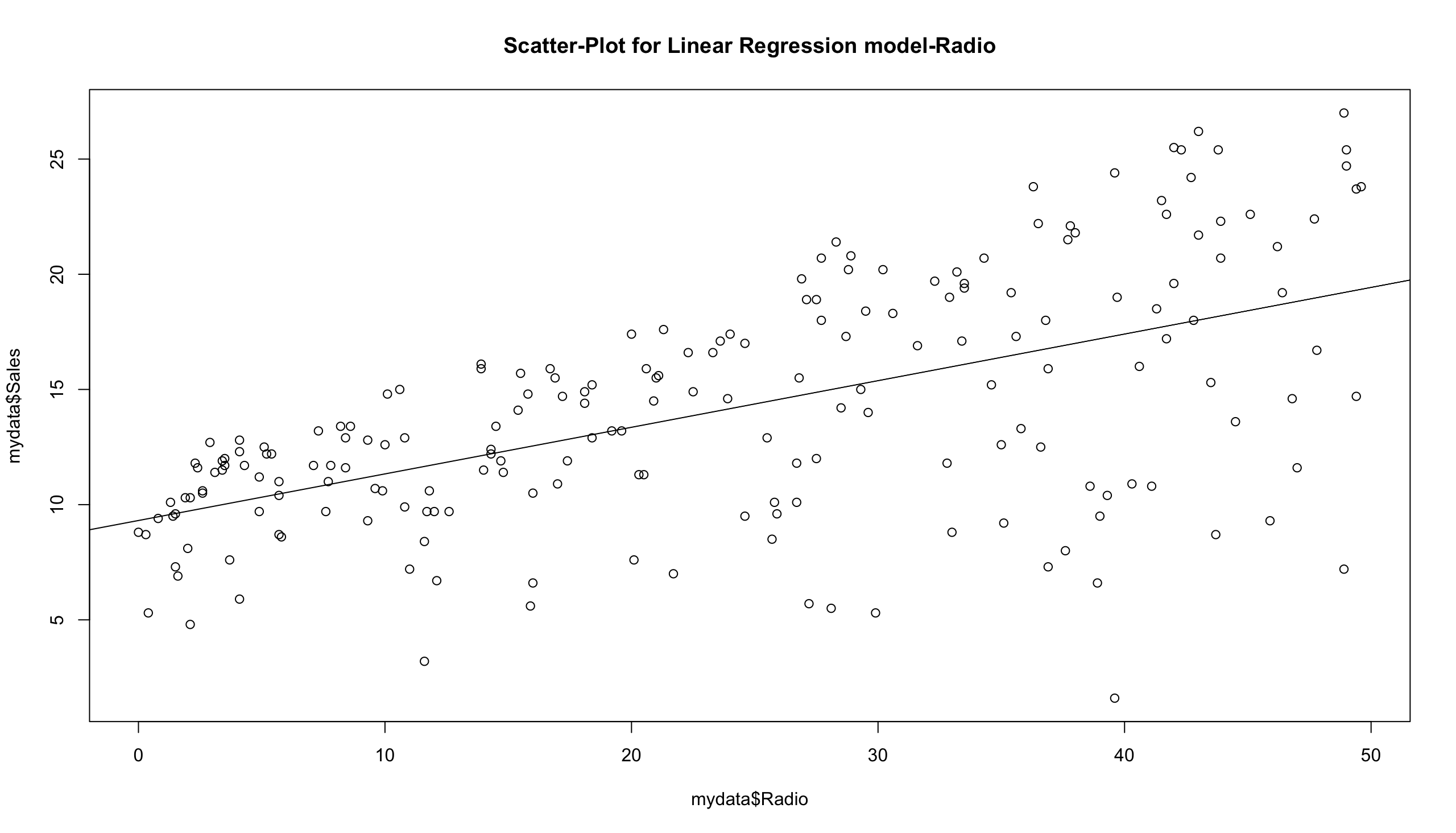
Scatter plots for 1: TV Adv

2: Radio\_Adv

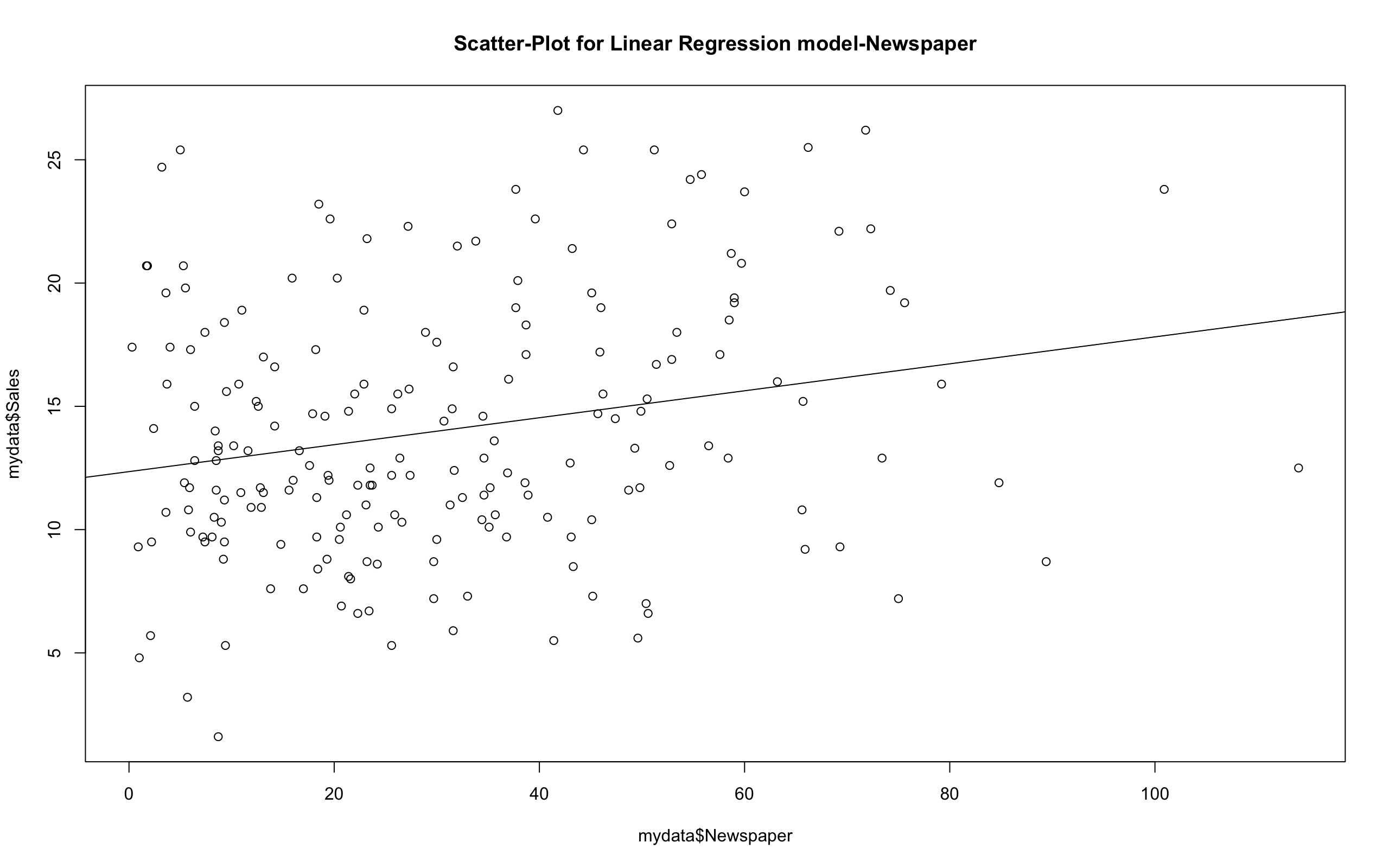
3: Newspaper\_Adv



1



2



3

mydata$TV provides the best prediction. We can say this from the correlation coefficient r found using cor() function in R.

r\_TV = 0.7822244

r\_Radio= 0.5762226

r\_Newspaper= 0.228299

Hence, TV advertising has more correlation with Sales and thus provides more prediction that other variates.

(b)

Estimated βvalues with Simple Linear Regression

β1 = 0.047537

β2 = 0.20250

β3 = 0.05469

Estimated β values with Multiple Linear regression

β1 = 0.045765

β2 = 0.188530

β3 = -0.001037

We can observe that there is a slight decrease in the β values as compared to the Simple Linear Regression models. This can be explained by the effect of other 2 variates on each of the independent variable. If there was a 10% or more change in the β values we can say that there is confounding variable for that respective variate.

**Q4.**  Fit logistic regression model for the dataset (hw23R-q4data.txt). Note that this dataset contains 3 covariates, therefore you should use multiple logistic regression which is straight forward generalization of simple logistic regression (recall simple linear regression vs. multiple linear regression; read documentation of the function you are using in R) (5 points)

Note: Write a function LogisticRegression(), to answer Q4 and save it in utils.R.

Sol: Function LogisticRegression() implemented in R and saved in utils.R file available in zip file.

Call:

glm(formula = mydata$Y ~ mydata$X1 + mydata$X2 + mydata$X3, data = mydata)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.6364 -0.2956 0.0973 0.2493 0.8030

Coefficients:

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

(Intercept) 0.3218658 0.0874576 3.680 0.000490 \*\*\*

mydata$X1 0.0031216 0.0008294 3.764 0.000374 \*\*\*

mydata$X2 0.0042457 0.0014367 2.955 0.004415 \*\*

mydata$X3 0.1485037 0.0453154 3.277 0.001721 \*\*

---

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

(Dispersion parameter for gaussian family taken to be 0.1144623)

Null deviance: 16.5000 on 65 degrees of freedom

Residual deviance: 7.0967 on 62 degrees of freedom

AIC: 50.118

Number of Fisher Scoring iterations: 2

**Q5**. Apply your data science skills to **improve** the model fitted in Q4. In what sense your improved model is **better** than the model found in (3.c). [Note the term “improve”; that is, you still have to use multiple logistic model only]. Show your work. (5 points)

Note: Write a function LogisticRegressionImproved() to answer Q5 and save it in utils.R.

Sol: Function LogisticRegressionImproved() implemented in R and saved in utils.R file available

After plotting each of the independent variate, and eliminating the outliers; we fit a model which is not an improvement from the previous based the summary statistics.

Call:

glm(formula = my\_newdata$Y ~ my\_newdata$X1 + my\_newdata$X2 +

my\_newdata$X3, data = my\_newdata)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.55870 -0.19719 0.01583 0.16727 0.52921

Coefficients:

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

(Intercept) 0.393638 0.091526 4.301 7.34e-05 \*\*\*

my\_newdata$X1 0.005892 0.001091 5.401 1.60e-06 \*\*\*

my\_newdata$X2 0.009112 0.002222 4.101 0.000142 \*\*\*

my\_newdata$X3 0.081693 0.054244 1.506 0.137993

---

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

(Dispersion parameter for gaussian family taken to be 0.06930267)

Null deviance: 14.140 on 56 degrees of freedom

Residual deviance: 3.673 on 53 degrees of freedom

AIC: 15.463

Number of Fisher Scoring iterations: 2