Assignment2

Chemi Goldstein , Dor Kolsky

13 12 2020

Assignment2 :

required packages that must be installed :

```
#install.packages("magrittr")
#install.packages("data.table")
#install.packages("ggplot2")
#install.packages("reshape2")
#install.packages("lubridate")
#install.packages("stringr")
#install.packages("corrplot")
#install.packages("dplyr")
#install.packages("gridExtra")
#install.packages("microbenchmark")
#install.packages("chron")
#install.packages("rattle.data")
#install.packages("ROCit")
#install.packages("lme4")
#install.packages("fitdistrplus")
```

required packes that must be loaded :

```
library(magrittr)
library(data.table)
library(ggplot2)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(stringr)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(microbenchmark)
library(chron)
```

```
##
## Attaching package: 'chron'
```

```
## The following objects are masked from 'package:lubridate':
##
##     days, hours, minutes, seconds, years
```

```
library(rattle.data)
library(ROCit)
library(lme4)
```

```
## Loading required package: Matrix
```

```
library(fitdistrplus)
```

```
## Loading required package: MASS
```

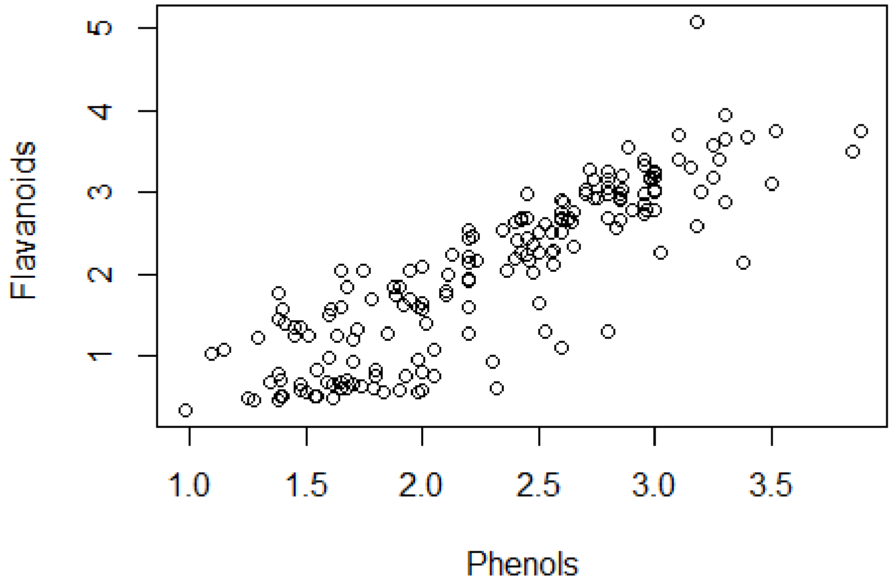
```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: survival
```

#Question 1 : Load the dataset :

```
wine_data <- as.data.table(wine)
?wine

## starting httpd help server ... done
```

A: Present the plotted relation between Flavanoids and Phenols :

```
plot(wine_data$Flavanoids ~ wine_data$Phenols, xlab = "Phenols" , ylab =
"Flavanoids ")
```



based on the visualization of our plot, we can clearly see there IS a positive linear relation between our two variables.

B: A plausible module for their relation is: Flavanoids(i) = β0 + β1*Phenols(i) + error(i).

No assumption need to be made. In this course our goal is to make good predictions using correlations and relationships between features, and not to describe a phenomenons or infer a causes. No assumptions need to be made regarding the OLS model.

C: To arrive to these equations, we took the minimized sum of squares, and afterward made the derivative using once B0, and secondly B1. This calculation, broken down and reorganized, leads us to the equations which we were asked about.

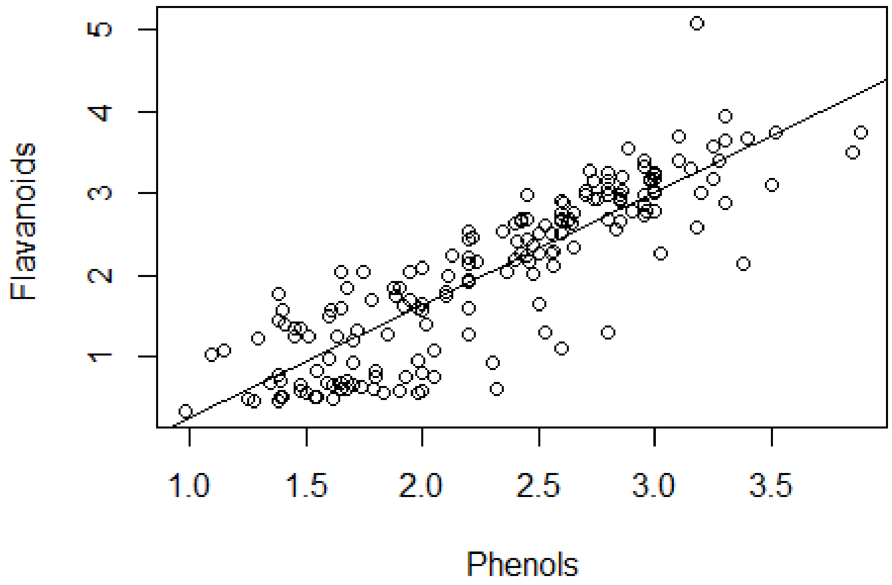
Since we have not used any conclusions from our module yet, no assumptions need to be made. We simply performed mathematical calculations.

D:

```
lm1 <- lm(wine_data$Flavanoids~wine_data$Phenols)
summary(lm1)
```

```
##
## Call:
## lm(formula = wine_data$Flavanoids ~ wine_data$Phenols)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.46361 -0.28305  0.05922  0.37011  1.82972
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.13763    0.14379  -7.912 2.71e-13 ***
## wine_data$Phenols  1.37984    0.06046  22.824  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5034 on 176 degrees of freedom
## Multiple R-squared:  0.7475, Adjusted R-squared:  0.746
## F-statistic: 520.9 on 1 and 176 DF,  p-value: < 2.2e-16
```

```
plot(wine_data$Flavanoids ~ wine_data$Phenols , xlab = "Phenols" , ylab =
"Flavanoids ")
abline(reg = lm1)
```

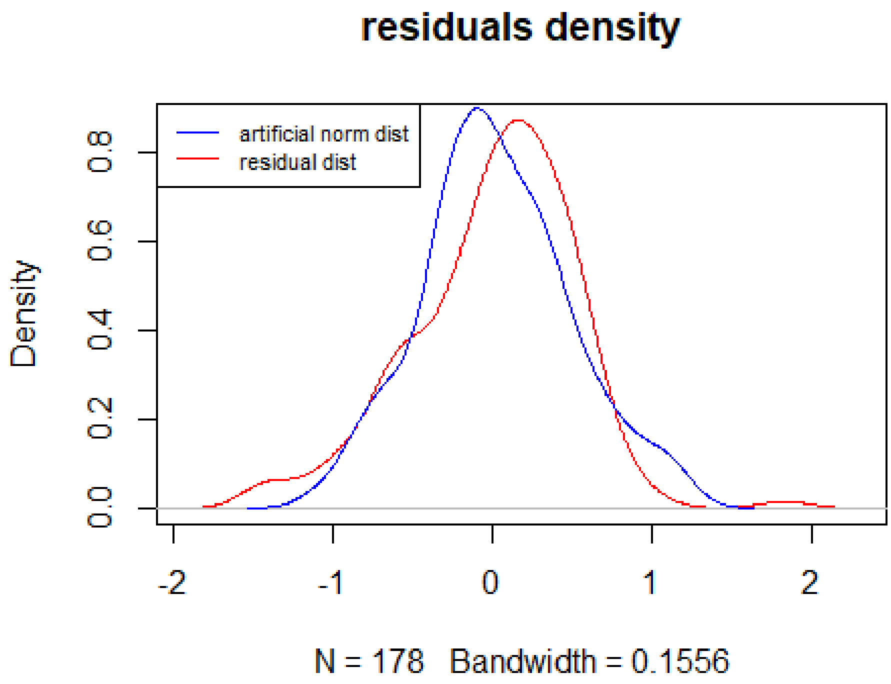


E: Based on the OLS regression data , the slope’s coefficient is 1.37984, meaning that on average, for every additional Phenol-unit, the Flavanoid is increased by 1.37984 units and its significant because its p-value is under 2e-16 which is smaller than alpha (95% as default)

The assumptions we used are are the basic assumptions of OLS model’s. For Example : Linear in parameters , random sampling of observations, epsilon ~ N(0,sigma^2)

F:

```
plot(density(lm1$residuals), main = "residuals density",col="red")
set.seed(1)
lines(density(rnorm(1:178,0,0.5)),col="blue")
legend("topleft", legend=c("artificial norm dist", "residual dist"),
col=c("blue","red"),lty=1, cex=0.7)
```



As we can see, the errors are distributed normally. We can see there is a Gaussian bell curve, which is equivalent to having a normal distribution.

G: Now will make a manual calculation of our model's coeffiences and more :

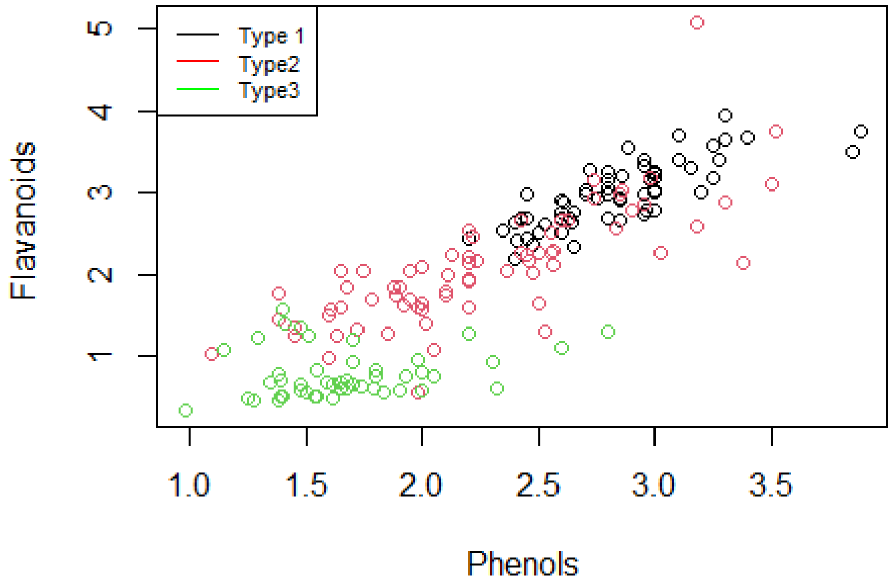
```
X <- model.matrix(~1+Phenols, data = wine_data)
Y <- wine_data$Flavanoids
beta_hat <- solve(t(X) %*% X) %*% (t(X) %*% Y) # The Coefficients
cat( "The Coeffiecients are " , beta_hat)

## The Coeffiecients are  -1.137627 1.379844

Y_hat <- predict(lm1)
R2 <- function(Y, Y_hat){
  numerator <- (Y-Y_hat)^2 %>% sum
  denominator <- (Y-mean(Y))^2 %>% sum
  1-numerator/denominator
}
cat (" The R2 is " , R2(Y,Y_hat))

##  The R2 is  0.74747

RSS <- function (Y , Y_hat){
  numerator <- (Y-Y_hat)^2 %>% sum
  return (numerator)
}
cat (" The RSS is ", RSS(Y,Y_hat))
```

```
##  The RSS is  44.59583
```

Based on the output of the summary we ran earlier, we see that our calculations are accurate.

H :

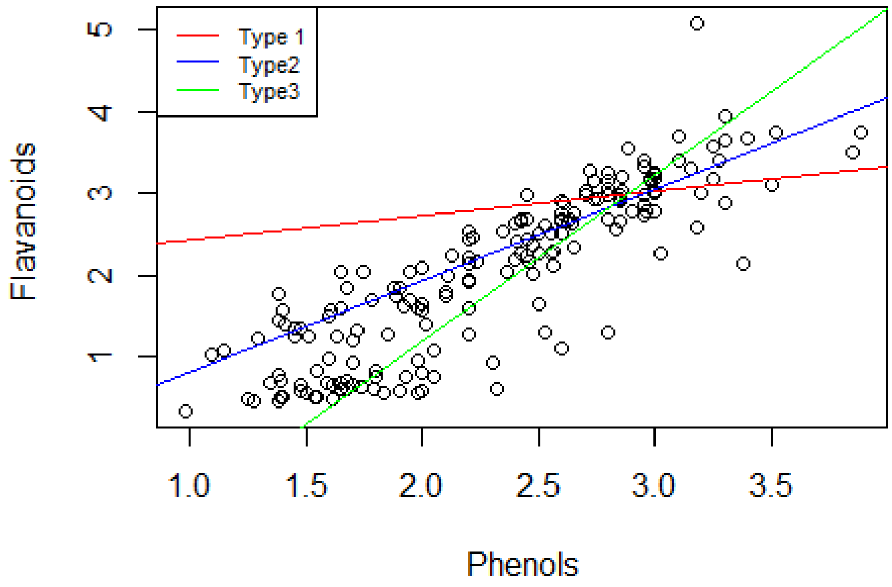
```
plot(wine_data$Flavanoids ~ wine_data$Phenols, xlab = "Phenols" , ylab = 
"Flavanoids " , col = wine_data$Type)
legend("topleft", legend=c("Type 1", "Type2" , "Type3"),
col=c("black","red","green"),lty=1, cex=0.7)
```



i :

```
lm2 <- lm(wine_data$Flavanoids~wine_data$Phenols + factor(wine_data$Type)+
factor(wine_data$Type)*wine_data$Flavanoids)

## Warning in model.matrix.default(mt, mf, contrasts): the response appeared
on the
## right-hand side and was dropped

## Warning in model.matrix.default(mt, mf, contrasts): problem with term 3 in
## model.matrix: no columns are assigned

summary(lm2)

## 
## Call:
## lm(formula = wine_data$Flavanoids ~ wine_data$Phenols + 
factor(wine_data$Type) +
##     factor(wine_data$Type) * wine_data$Flavanoids)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65916 -0.08044 -0.00173  0.09050  0.80846
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                   2.12281    0.13087  16.220
## wine_data$Phenols                             0.30264    0.04511   6.710
## factor(wine_data$Type)2                      -2.43122    0.11534 -21.079
## factor(wine_data$Type)3                      -2.56215    0.10903 -23.500
## wine_data$Flavanoids:factor(wine_data$Type)2  0.81968    0.04397  18.641
## wine_data$Flavanoids:factor(wine_data$Type)3  0.91206    0.10296   8.858
##                                              Pr(>|t|)
## (Intercept)                                   < 2e-16 ***
## wine_data$Phenols                            2.71e-10 ***
## factor(wine_data$Type)2                       < 2e-16 ***
## factor(wine_data$Type)3                       < 2e-16 ***
## wine_data$Flavanoids:factor(wine_data$Type)2  < 2e-16 ***
## wine_data$Flavanoids:factor(wine_data$Type)3 9.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2055 on 172 degrees of freedom
## Multiple R-squared:  0.9589, Adjusted R-squared:  0.9577
## F-statistic:   802 on 5 and 172 DF,  p-value: < 2.2e-16
```

```
plot(wine_data$Flavanoids ~ wine_data$Phenols , xlab = "Phenols" , ylab =
"Flavanoids ")
abline(a=lm2$coefficients[1] , b= lm2$coefficients[2] , col = "red") # TYPE 1
ABLINE
abline(a=lm2$coefficients[1]+lm2$coefficients[3] ,
b=lm2$coefficients[2]+lm2$coefficients[5] , col= "blue") # TYPE 2 ABLINE
abline(a=lm2$coefficients[1]+lm2$coefficients[3]+lm2$coefficients[4] ,
b=lm2$coefficients[2]+lm2$coefficients[5]+lm2$coefficients[6] , col= "green")
# TYPE 3 ABLINE
legend("topleft", legend=c("Type 1", "Type2" , "Type3"),
col=c("red","blue","green"),lty=1, cex=0.7)
```



J: We plotted the Flav levels vs the Phenol levels nd added intercepts and slopes for three different wine types using factor function :

```
cat("Type 1 intercept is the basic intercept with the value of
",lm2$coefficients[1] , "Type 1 slope is the basic slope with the value of" ,
lm2$coefficients[2])
```

```
## Type 1 intercept is the basic intercept with the value of  2.122812 Type 1
slope is the basic slope with the value of 0.3026444
```

```
cat(" Type 2 intercept is type 1 intercept along with addition with the total
value of " , lm2$coefficients[1]+lm2$coefficients[3] , " Type 2 slope is type
2 slope along with addition with the total value of " , lm2$coefficients[2] +
lm2$coefficients[5])
```

```
##  Type 2 intercept is type 1 intercept along with addition with the total
value of  -0.3084119  Type 2 slope is type 2 slope along with addition with
the total value of  1.122322
```

```
cat(" Type 3 intercept is type 2 intercept along with addition with the total
value of " , lm2$coefficients[1]+lm2$coefficients[3]+lm2$coefficients[4] , "
Type 3 slope is type 2 slope along with addition with the total value of " ,
lm2$coefficients[2] + lm2$coefficients[5]+lm2$coefficients[6])
```

```
##  Type 3 intercept is type 2 intercept along with addition with the total
value of  -2.870565   Type 3 slope is type 2 slope along with addition with
the total value of  2.03438
```

The coefficient interpretations: type 1 wine - the average Flav level when the Phenols level is zero is: 2.12; and for every addition of 1 level of phenols, on average, the Flav level is increased by 0.302

type 2 wine - the average Flav level when the Phenols level is zero is: -0.308; and for every addition of 1 level of phenols, on average, the Flav level is increased by 1.122

type 3 wine - the average Flav level when the Phenols level is zero is: -2.87; nd for every addition of 1 level of phenols, on average, the Flav level is increased by 2.034

#Question 2 :

```
adult <-read.csv("https://raw.githubusercontent.com/guru99-edu/R-
Programming/master/adult.csv")[,-1]
```

A:

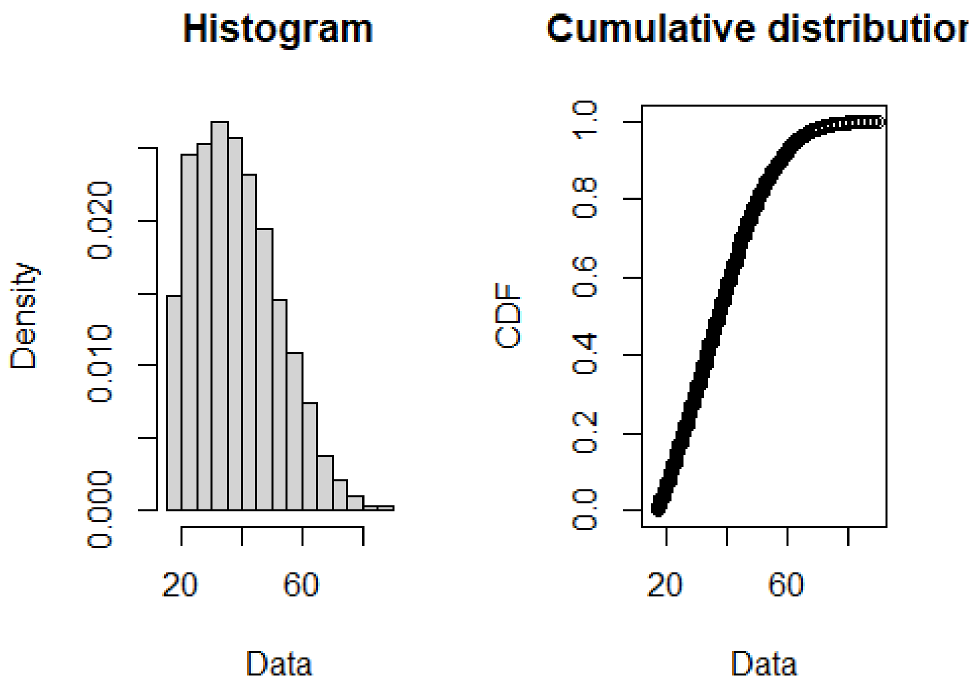
```
lapply(adult , class)

## $age
## [1] "integer"
##
## $workclass
## [1] "character"
##
## $education
## [1] "character"
##
## $educational.num
## [1] "integer"
##
## $marital.status
## [1] "character"
##
## $race
## [1] "character"
##
## $gender
## [1] "character"
##
## $hours.per.week
## [1] "integer"
##
## $income
## [1] "character"
```

The continues variables are : age , educational_num and hours_per_week , the rest are categorical .

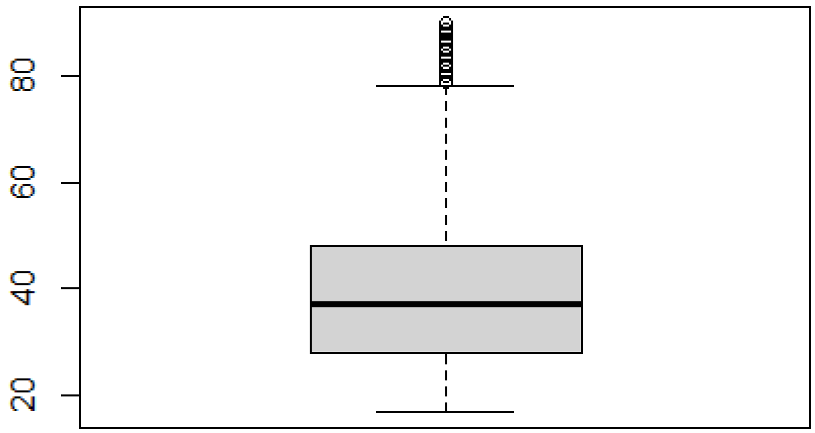
B: In order to plot the density plot we will use the command plotdist and in order to identify outliers we will use boxplot.

Age:

```r
plotdist(adult$age)
```



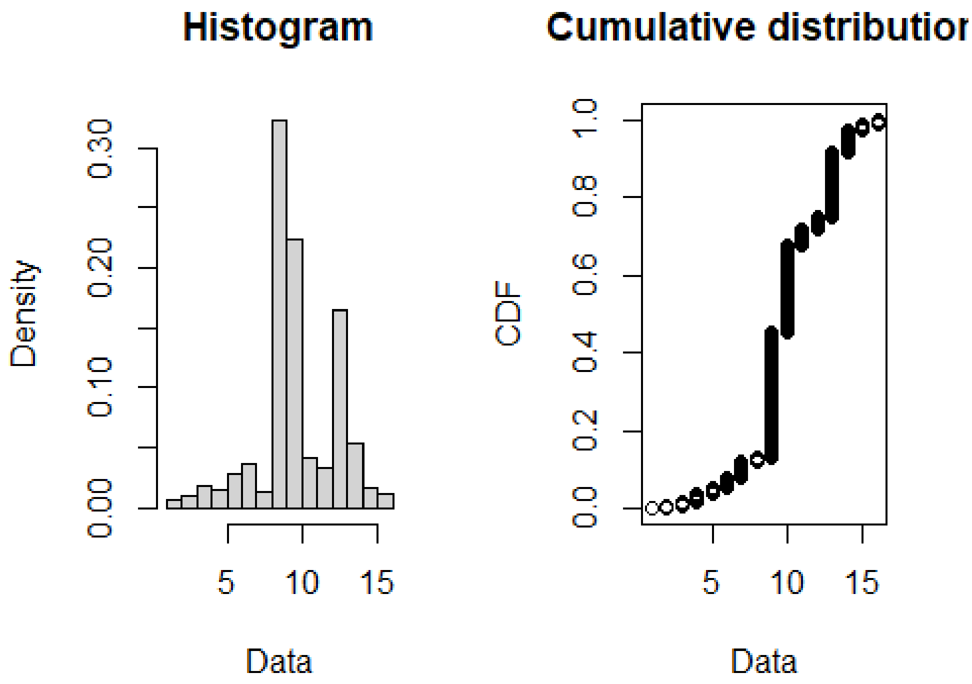
```r
boxplot(adult$age)
```



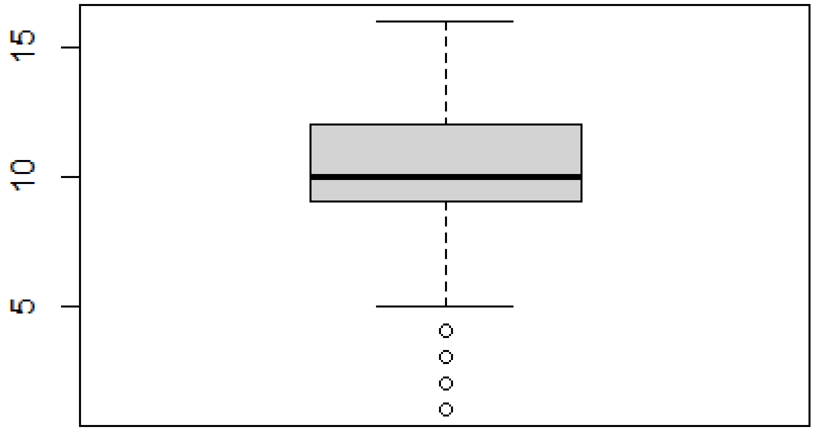
```
Q1 <- quantile(adult$age, probs=c(.25, .75), na.rm = FALSE)
iqr1 <- IQR(adult$age)
up1 <-  Q1[2]+1.5*iqr1 # Upper Range
low1<- Q1[1]-1.5*iqr1 # Lower Range
adult<- subset(adult, adult$age > (Q1[1] - 1.5*iqr1) & adult$age <
(Q1[2]+1.5*iqr1))
```

Educational Num :

```
plotdist(adult$educational.num)
```



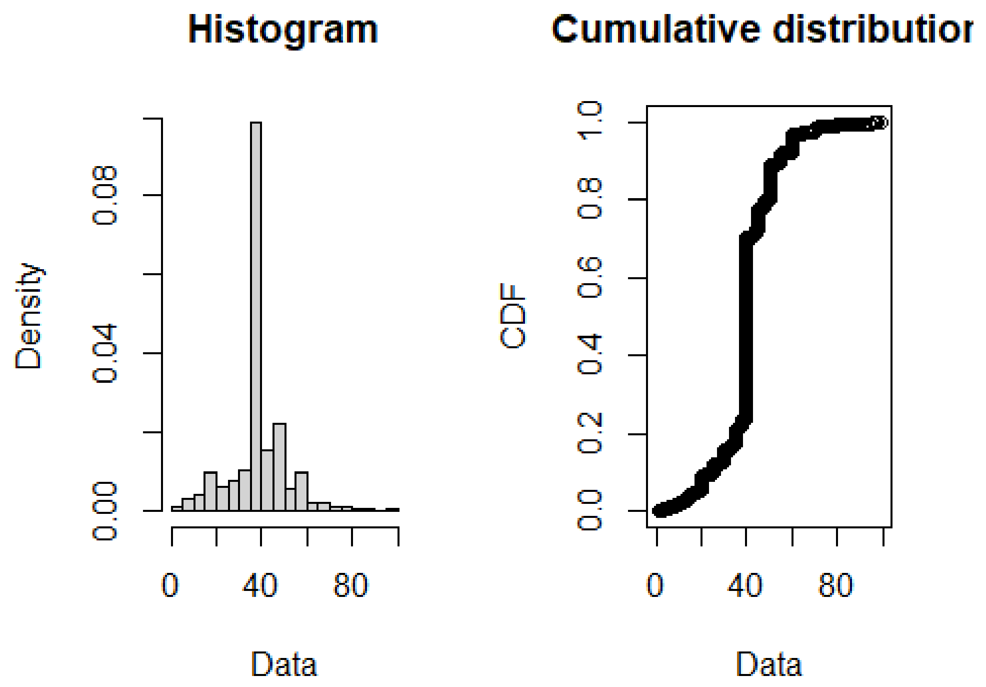
```
boxplot(adult$educational.num)
```



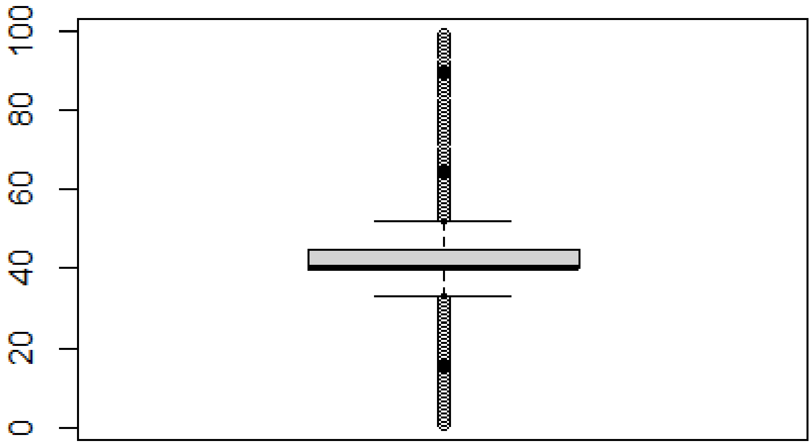
```
Q2 <- quantile(adult$educational.num, probs=c(.25, .75), na.rm = FALSE)
iqr2 <- IQR(adult$educational.num)
up2 <-  Q2[2]+1.5*iqr2 # Upper Range
low2<- Q2[1]-1.5*iqr2 # Lower Range
adult<- subset(adult, adult$educational.num > (Q2[1] - 1.5*iqr2) &
adult$educational.num < (Q2[2]+1.5*iqr2))
```

Hours_Per_Week

```
plotdist(adult$hours.per.week)
```



```
boxplot(adult$hours.per.week)
```



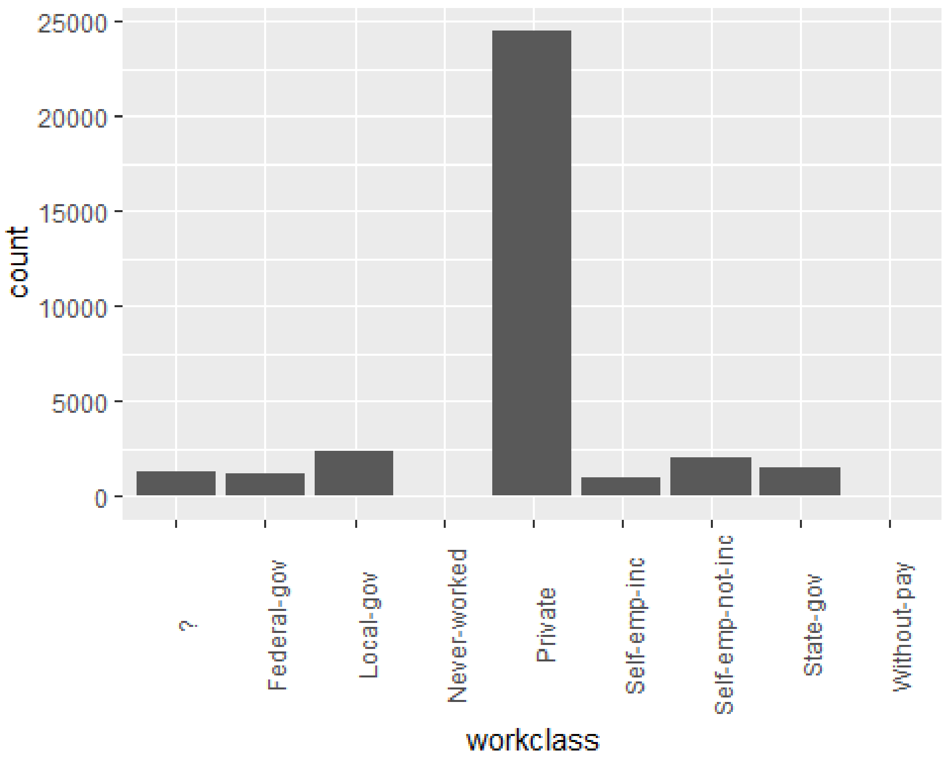
```
Q3 <- quantile(adult$hours.per.week, probs=c(.25, .75), na.rm = FALSE)
iqr3 <- IQR(adult$hours.per.week)
up3 <-  Q3[2]+1.5*iqr3 # Upper Range
low3<- Q3[1]-1.5*iqr3 # Lower Range
adult<- subset(adult, adult$hours.per.week > (Q3[1] - 1.5*iqr3) &
adult$hours.per.week < (Q3[2]+1.5*iqr3))
```

C: We will standardize our data's continuous variables :

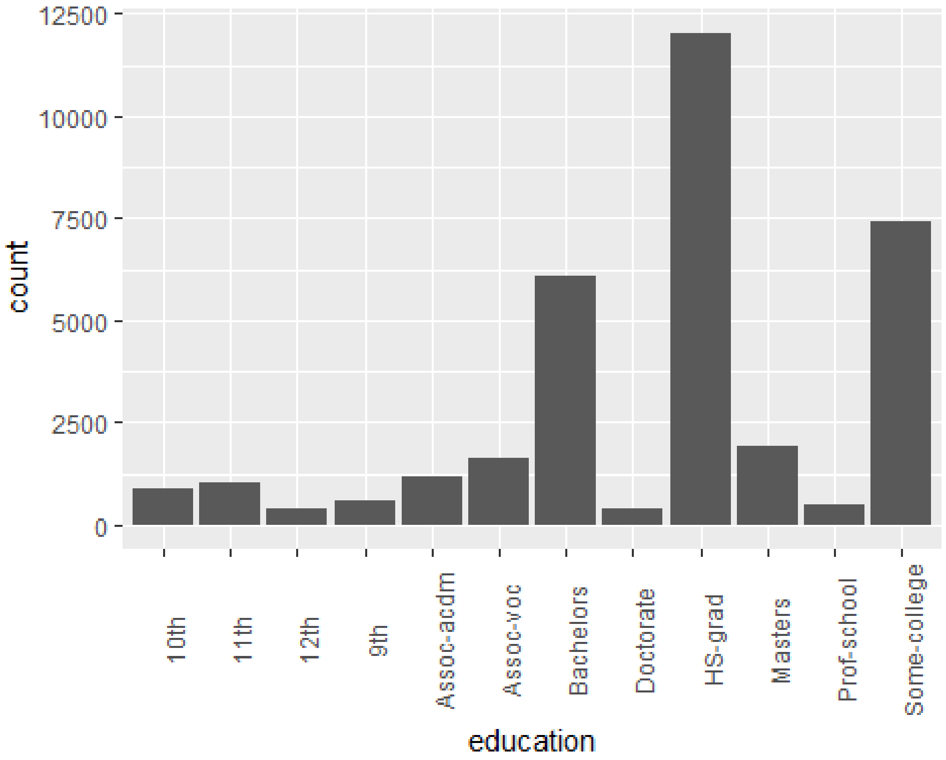
```
adult$age <- scale(adult$age)
adult$educational.num <- scale(adult$educational.num)
adult$hours.per.week <- scale(adult$hours.per.week)
```

D:

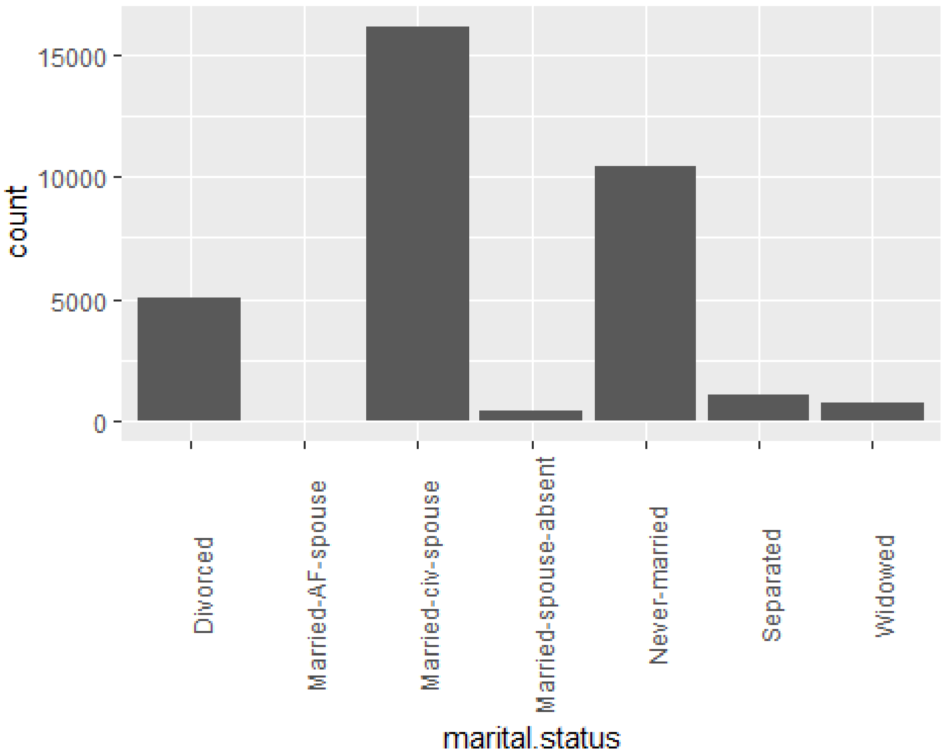
```
ggplot(adult) + geom_bar(aes(x=workclass))+ theme(axis.text.x =
element_text(angle=90))
```



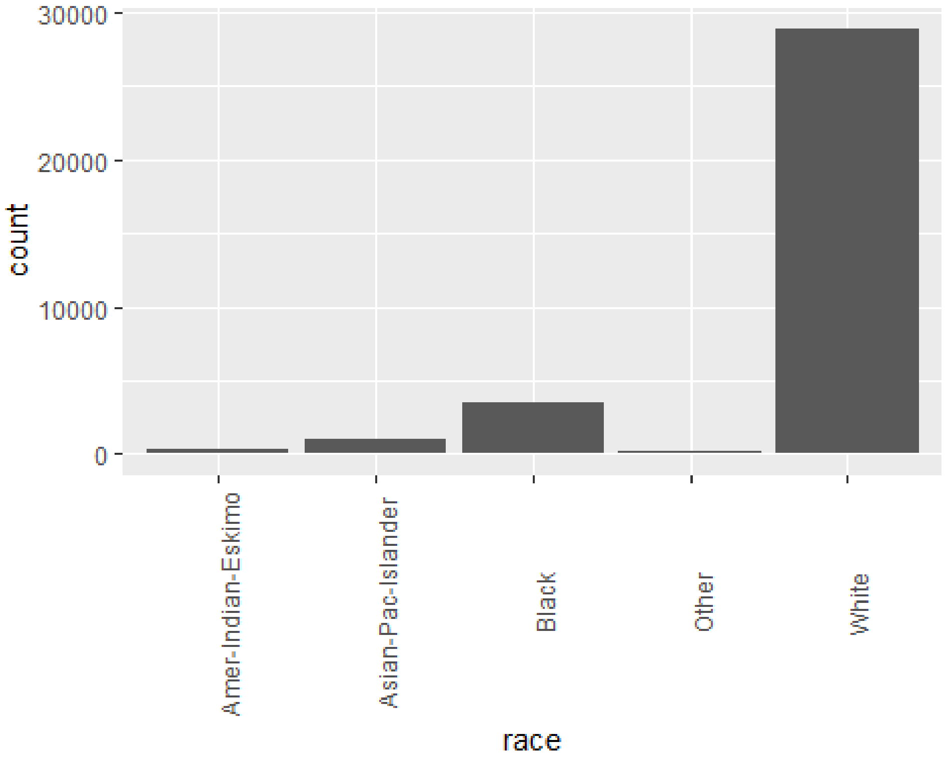
```
ggplot(adult) + geom_bar(aes(x=education))+ theme(axis.text.x =
element_text(angle=90))
```



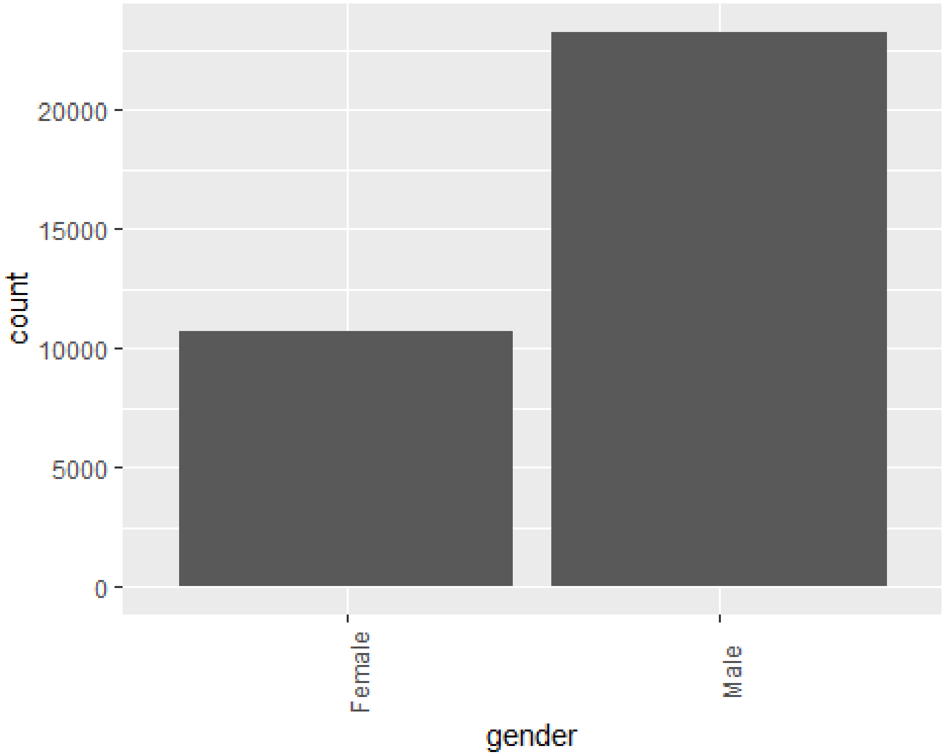
```
ggplot(adult) + geom_bar(aes(x=marital.status)) + theme(axis.text.x =
element_text(angle=90))
```



```
ggplot(adult) + geom_bar(aes(x=race)) + theme(axis.text.x =
element_text(angle=90))
```



```r
ggplot(adult) + geom_bar(aes(x=gender)) + theme(axis.text.x =
element_text(angle=90))
```



E: WorkClass recast :

```
adult$workclass <- as.factor(adult$workclass)
levels(adult$workclass) <- c("Non-Private","Non-Private","Non-Private","Non-
Private","Priavte","Non-Private","Non-Private","Non-Private","Non-Private") #
recast its levels
levels(adult$workclass)

## [1] "Non-Private" "Priavte"
```

educational recast :

```
adult$education <- as.factor(adult$education)
levels(adult$education) <-
c("Highschool","Highschool","Highschool","Highschool","Assosiciate","Assosici
ate","Bachelores","Above BA","HS-grad","Above BA","Above BA","Some-college")
# recast its levels
levels(adult$education)

## [1] "Highschool"   "Assosiciate"  "Bachelores"   "Above BA"     "HS-grad"
## [6] "Some-college"
```

Martial Status recast:

```
adult$marital.status <- as.factor(adult$marital.status)
levels(adult$marital.status) <- c("After-
Marriage","Married","Married","Married","Never-Married","After-
Marriage","After-Marriage") # recast its levels
levels(adult$marital.status)

## [1] "After-Marriage" "Married"        "Never-Married"
```

race recast :

```
adult$race <- as.factor(adult$race)
levels(adult$race) <- c("Other","Other","Black","Other","White") # recast its
levels
levels(adult$race)

## [1] "Other" "Black" "White"
```

F :

```
set.seed(1)
ind<- sample(2, nrow(adult), replace= T , prob = c(0.7,0.3))
train<- adult[ind==1,]
test<- adult[ind==2,]
```

G:

```
glm.1 <- glm(as.factor(income) ~ ., data = train, family = binomial)
summary(glm.1)
```

```
## 
## Call:
## glm(formula = as.factor(income) ~ ., family = binomial, data = train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.6076  -0.6361  -0.2784   0.4876   3.2620  
## 
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                 -3.53580    0.25333 -13.958  < 2e-16 ***
## age                          0.44771    0.02110  21.215  < 2e-16 ***
## workclassPriavte             0.15749    0.04080   3.861 0.000113 ***
## educationAssosiciate         0.20053    0.29189   0.687 0.492067    
## educationBachelores          0.57507    0.37228   1.545 0.122414    
## educationAbove BA            0.80638    0.44920   1.795 0.072628 .  
## educationHS-grad             0.14808    0.16612   0.891 0.372698    
## educationSome-college        0.43433    0.21550   2.015 0.043862 *  
## educational.num              0.65844    0.12246   5.377 7.58e-08 ***
## marital.statusMarried        2.10588    0.06178  34.087  < 2e-16 ***
## marital.statusNever-Married -0.36700    0.07912  -4.638 3.51e-06 ***
## raceBlack                    0.04914    0.11306   0.435 0.663849    
## raceWhite                    0.42706    0.08981   4.755 1.98e-06 ***
## genderMale                   0.21850    0.05094   4.289 1.79e-05 ***
## hours.per.week               0.27179    0.01771  15.351  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 27322  on 23796  degrees of freedom
## Residual deviance: 18756  on 23782  degrees of freedom
## AIC: 18786
## 
## Number of Fisher Scoring iterations: 6
```

The AIC level in this model is 18786, a smaller AIC points to a more explained module.

H:

```
data_test_hat<-predict(glm.1, test, type = "response")


data_test_hat_binar<-(data_test_hat>0.5)*1 #as we learned in our course-book, 
a trivial rule is: the default threshold is 0.5

CM.glm <- table(true= test$income, predicted = data_test_hat_binar)
CM.glm
```

```
##         predicted
## true       0    1
##   <=50K 6867  666
##   >50K  1279 1382

paste("We predicted correctly ", CM.glm[1,1]+CM.glm[2,2],", and we missed ",
CM.glm[1,2]+CM.glm[2,1],".",sep="")

## [1] "We predicted correctly 8249, and we missed 1945."
```

I: To measure accuracy we will use the formula: Accuracy = (TP+TN)/(TP+TN+FP+FN)

```
accuracy<-(sum(diag(CM.glm)) / sum(CM.glm))
paste("The accuracy of the model is: ",accuracy)

## [1] "The accuracy of the model is:  0.80920149107318"
```

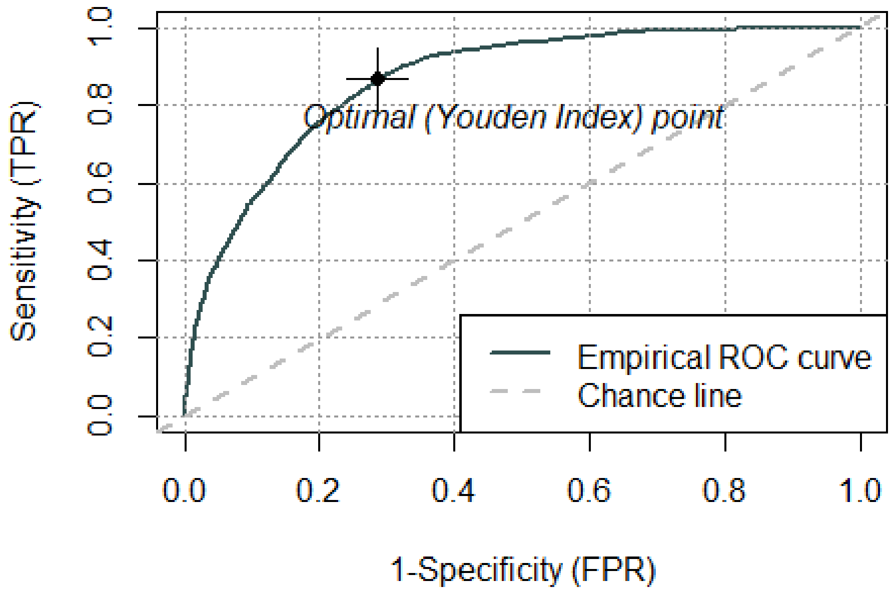
J: The Precision formula is:TP/(TP+FP) The Recall formula is:TP/(TP+FN)

```
Precion <- (CM.glm[4] / sum(CM.glm[,2]))
Recall <- (CM.glm[4] / sum(CM.glm[2,]))
paste("The Precision of the model is: ",Precion)

## [1] "The Precision of the model is:  0.6748046875"

paste("The Recall of the model is: ",Recall)

## [1] "The Recall of the model is:  0.51935362645622"
```

As we learned in class there is a trade-off between Precision and Recall. they come at the expense of one another. We choose which of the two measurements we want to be higher based on the sensitivity we have on False-Positives/False-Negatives. High sensitivity to False-Positives - higher Precision High sensitivity to False-Negatives - higher Recall

K: The ROC Curve is as follows :

```
rocit_obj <-rocit(score=data_test_hat,class=test$income)
plot(rocit_obj)
```



The ROC curve is a plot of the [true positive rate (Recall) VS the False positive rate] for different sensitivity levels. Using this curve, enables us to select the most optimal modules.

I: In order to improve our model we will add interaction between variables to our model:

```
glm.1_improved<-  glm(as.factor(income) ~ . + I(age^2), data = train, family
= binomial)
summary(glm.1_improved)

##
## Call:
## glm(formula = as.factor(income) ~ . + I(age^2), family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4300  -0.6089  -0.2753   0.4983   3.5126
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -3.255302   0.252988 -12.867  < 2e-16 ***
## age                            0.774142   0.029445  26.291  < 2e-16 ***
## workclassPriavte               0.140443   0.040967   3.428 0.000608 ***
## educationAssosiciate           0.079165   0.290565   0.272 0.785274
## educationBachelores            0.468688   0.370346   1.266 0.205678
## educationAbove BA              0.628109   0.447187   1.405 0.160148
## educationHS-grad               0.073363   0.165355   0.444 0.657282
## educationSome-college          0.347267   0.214488   1.619 0.105437
## educational.num              0.656533   0.121877   5.387 7.17e-08 ***
## marital.statusMarried        2.146118   0.061789  34.733  < 2e-16 ***
## marital.statusNever-Married -0.132535   0.079566  -1.666 0.095770 .
## raceBlack                    0.008343   0.113985   0.073 0.941655
## raceWhite                    0.439488   0.090833   4.838 1.31e-06 ***
## genderMale                   0.197529   0.051451   3.839 0.000123 ***
## hours.per.week               0.253816   0.017879  14.196  < 2e-16 ***
## I(age^2)                    -0.348923   0.020211 -17.264  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27322  on 23796  degrees of freedom
## Residual deviance: 18435  on 23781  degrees of freedom
## AIC: 18467
##
## Number of Fisher Scoring iterations: 6
```

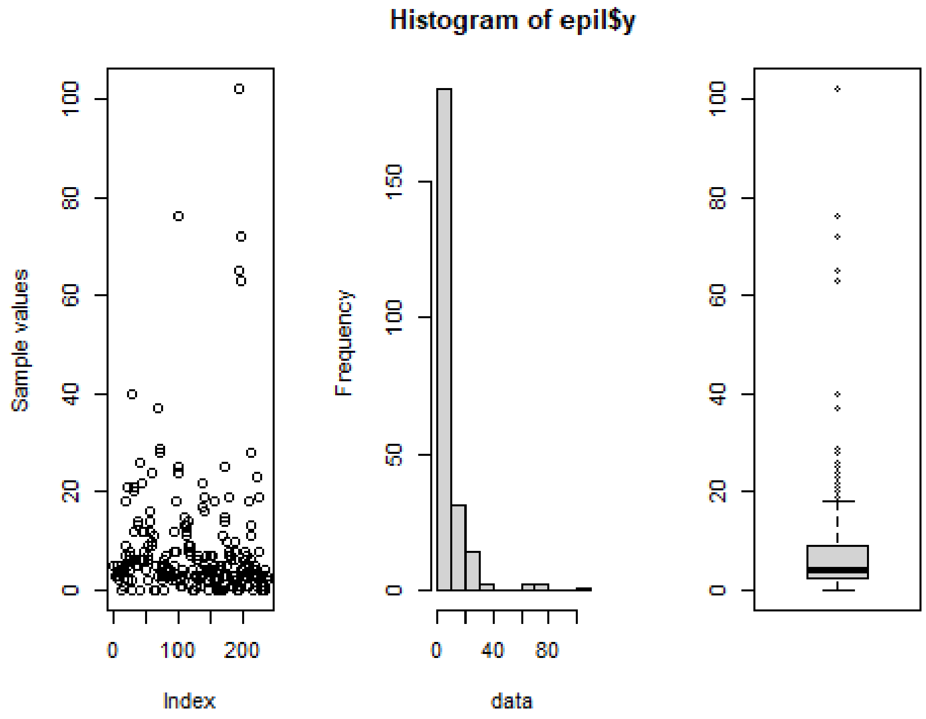
```
data_test_hat2<-predict(glm.1_improved, test, type = "response")
data_test_hat_binar2<-(data_test_hat2>0.5)*1

CM.glm1.improved <- table(true= test$income, predicted =
data_test_hat_binar2)
accuracy2<-(sum(diag(CM.glm1.improved)) / sum(CM.glm1.improved))
paste("The accuracy of the improved model is: ",accuracy2)
```

```
## [1] "The accuracy of the improved model is:  0.815283500098097"
```

Additionally, we can se that the accuracy of this new and improved model, has been increased!

#Question 3 : A:

```
epil <- as.data.table(epil)

par(mfrow=c(1,3))
plot(epil$y,ylab="Sample values")
hist(epil$y, ylab="Frequency " , xlab = "data")
boxplot(epil$y)
```


Histogram of epil$y

Based on the visualization of the graphs we created, we can assume that the most fitting distribution for y is the ‘Poisson distribution’. This distribution expresses the the probability of a given number of events occurring in a fixed interval of time, if these events happen with a known rate. Our ‘y’ is a count data, on which events happen in different time intervals, thus the poisson distribution fits well.

B:

```
glm.3 <- glm(y~age+trt , data = epil , family=poisson)
summary(glm.3)

## 
## Call:
## glm(formula = y ~ age + trt, family = poisson, data = epil)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -4.3628  -2.4087  -1.3791   0.0006  17.8489  
## 
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)    
## (Intercept)   2.533031   0.110638  22.895  < 2e-16 ***
## age          -0.013331   0.003708  -3.596 0.000324 ***
## trtprogabide -0.092514   0.045596  -2.029 0.042460 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
```

```
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2517.8  on 235  degrees of freedom
## Residual deviance: 2502.0  on 233  degrees of freedom
## AIC: 3273.9
##
## Number of Fisher Scoring iterations: 6
```

We created a Poisson GLM regression where the model counts the amount of seizures based on various continuous variables.By doing this, we fit a module assuming y|x ~ Poisson (exp(x'b)) - a poisson distributed with a rate that depends on the predictors .

C: The treatment reduces the seizure frequency, with the rate of e^(-0.092514) seizures per treatment-unit. The findings are significant, with an alpha of - 0.05, since the P-value is 0.042.

D:

```
y_predicted <- glm.3$coefficients[1] + (glm.3$coefficients[2]
*20)+glm.3$coefficients[3]
cat("The predicted y for 20 old who used progabide treatment is " ,
y_predicted , "seizures")
```

```
## The predicted y for 20 old who used progabide treatment is  2.173888
seizures
```

E: The coefficient of ‘age’ is the difference in amount of seizures, on average, based on the patients age. Meaning that on average, the amount of seizures decreases by e^(-0.013331) for each additional year of age.

#Question 4:

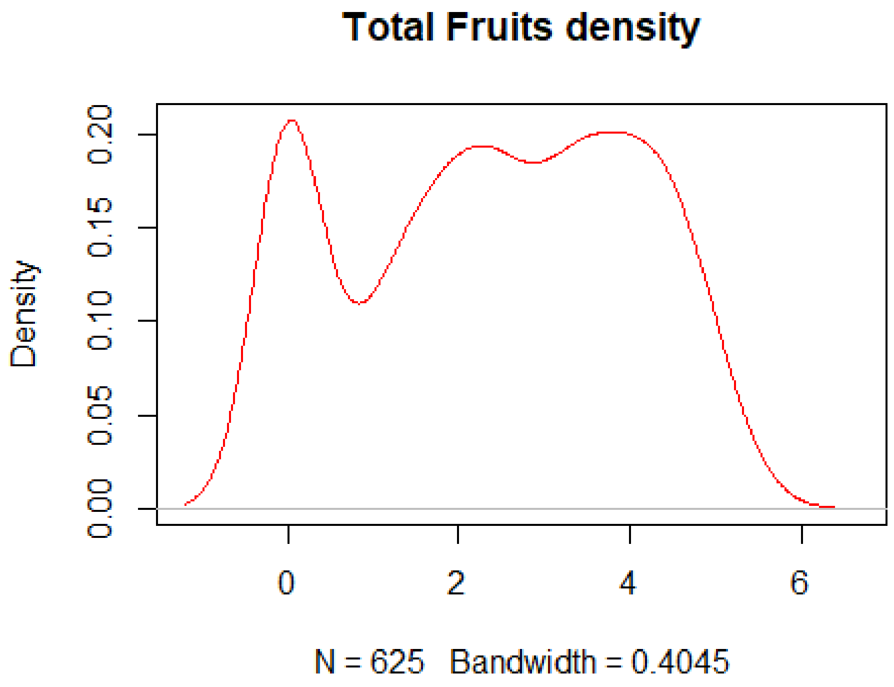
```
arab <- as.data.table(Arabidopsis)
```

Attaching the Dataset :

```
attach(arab)
```

A:

```
arab$gen <- as.factor(arab$gen)
arab$rack <- as.factor(arab$rack)
arab$nutrient <- as.factor(arab$nutrient)
arab$total.fruits <- log1p(arab$total.fruits)

plot(density(arab$total.fruits), main = "Total Fruits density",col="red")
```



As we can see, there is a slight approxamation to the Gaussian distribution. we can some what see the Gauss-bell.

B: We see that there is a dependence relation between the population-region variables. Meaning that one is reliant on the other.

C:

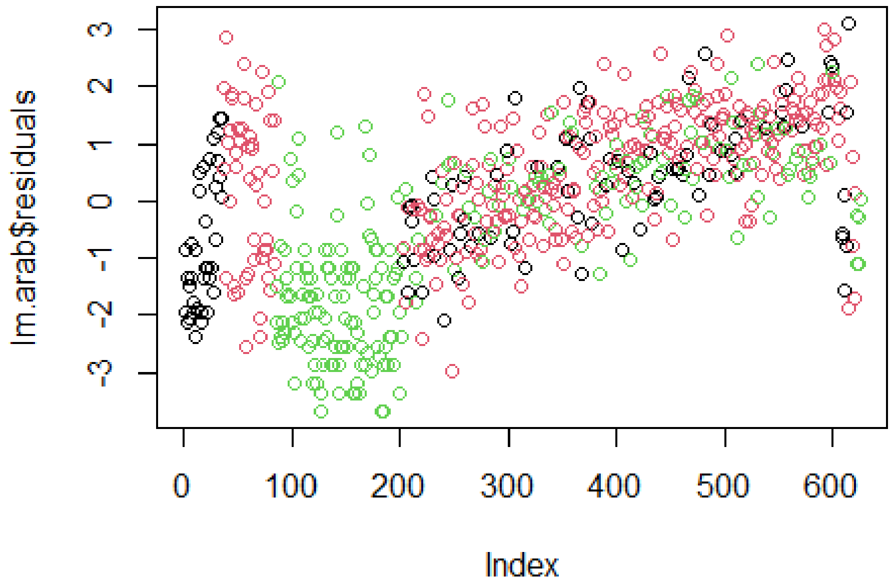
```
X <- model.matrix(~1+rack+nutrient+amd+status, data = arab)
# X is a large matrix we can view it by the command of View(X) we will
represent only the first row of X
X[1,]

##       (Intercept)             rack2         nutrient8      amdunclipped
##                 1                 1                 0                 0
## statusPetri.Plate  statusTransplant
##                 0                 1
```

D:

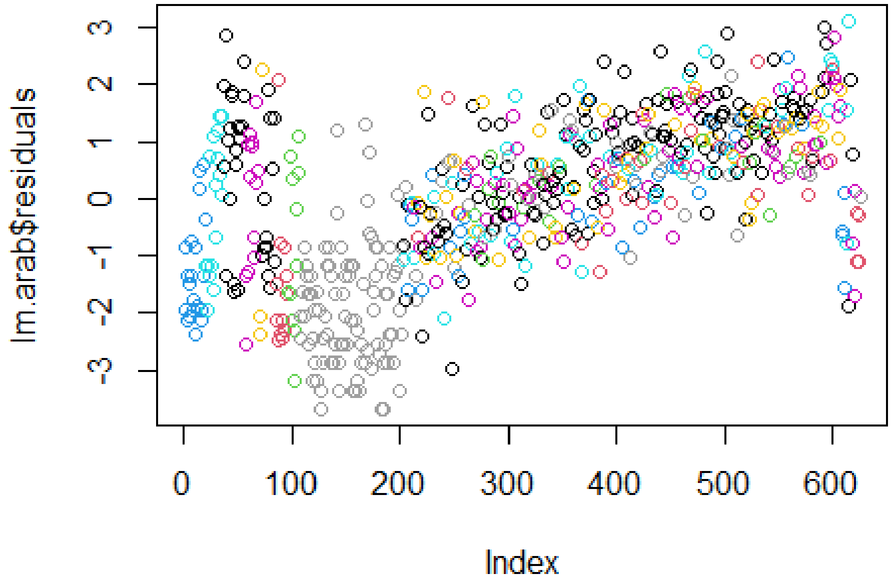
```
lm.arab <- lm(total.fruits~rack+nutrient+amd+status, data=arab)
#we ran the model on all the x's nor the x's on the previous question.
summary(lm.arab)

##
## Call:
## lm(formula = total.fruits ~ rack + nutrient + amd + status, data = arab)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.6835 -1.0418  0.2295  1.1256  3.1183 
## 
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           2.1404     0.1260  16.991  < 2e-16 ***
## rack2                -0.7956     0.1150  -6.917 1.15e-11 ***
## nutrient8             1.2192     0.1150  10.599  < 2e-16 ***
## amdunclipped          0.3239     0.1150   2.818 0.004990 ** 
## statusPetri.Plate    -0.1726     0.1681  -1.027 0.304915    
## statusTransplant     -0.4989     0.1364  -3.659 0.000275 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.435 on 619 degrees of freedom
## Multiple R-squared:  0.2298, Adjusted R-squared:  0.2236 
## F-statistic: 36.94 on 5 and 619 DF,  p-value: < 2.2e-16
```

```
plot(lm.arab$residuals, col=arab$reg)
```



```
plot(lm.arab$residuals, col=arab$popu)
```



Since the model is reliant on specific regions and populations, our errors will not be only measurement errors, but rather also be affected by random-effect and not only fixed effects. For this reason, the linear model is not sufficient, thus we will use a linear mixed model (LMM).

We plotted the residuals of our model, and differentiated each region's and population's residuals by color. As we can see, for each region/population (each color), the observations are clustered, or varianced, at different points.

E: Making LMM Regression :

```
lme.arab1 <- lmer(total.fruits~ 1|gen, data =arab)
lme.arab2 <- lmer(total.fruits~ (1|popu)+(1|reg), data =arab)
lme.arab3 <- lmer(total.fruits ~ (1|arab$reg) +(1|reg/popu) ,data =arab)
```

F:

```
#For LME 1 :
Z1 <- model.matrix(~0 + gen, data = arab)
z1 <- Z1[1,]

Z2 <- model.matrix(~0 + popu + reg, data = arab)
z2 <- Z2[1,]

Z3 <- model.matrix(~0 + reg + reg/popu, data = arab)
z3 <- Z3[1,]
```

```
z1

##  gen4  gen5  gen6 gen11 gen12 gen13 gen14 gen15 gen16 gen17 gen18 gen19 
gen20 
##     1     0     0     0     0     0     0     0     0     0     0     0 
0 
## gen21 gen22 gen23 gen24 gen25 gen27 gen28 gen30 gen34 gen35 gen36 
##     0     0     0     0     0     0     0     0     0     0     0

z2

## popu1.SP popu1.SW popu2.SW popu3.NL popu5.NL popu5.SP popu6.SP popu7.SW 
##        0        0        0        1        0        0        0        0 
## popu8.SP    regSP    regSW 
##        0        0        0

z3

##          regNL          regSP          regSW regNL:popu1.SW regSP:popu1.SW 
##              1              0              0              0              0 
## regSW:popu1.SW regNL:popu2.SW regSP:popu2.SW regSW:popu2.SW regNL:popu3.NL 
##              0              0              0              0              1 
## regSP:popu3.NL regSW:popu3.NL regNL:popu5.NL regSP:popu5.NL regSW:popu5.NL 
##              0              0              0              0              0 
## regNL:popu5.SP regSP:popu5.SP regSW:popu5.SP regNL:popu6.SP regSP:popu6.SP 
##              0              0              0              0              0 
## regSW:popu6.SP regNL:popu7.SW regSP:popu7.SW regSW:popu7.SW regNL:popu8.SP 
##              0              0              0              0              0 
## regSP:popu8.SP regSW:popu8.SP 
##              0              0
```

G:

```
lme.arab4 <- lmer(total.fruits ~ nutrient | reg ,data =arab)

## boundary (singular) fit: see ?isSingular
```

An appropriate LMM model could be one where we checked the slope of the nutrient affect on the total_fruit count. We assume that the variance is different for each region.

H:

```
lme.arab5 <- lmer(total.fruits ~ nutrient | reg/popu ,data =arab)

## boundary (singular) fit: see ?isSingular
```

As we saw in section 2, there is a relation between Population-Regions. Meaning that there is a random effect on population as well (as it is correlated with region), for this reason we ran an LMM on our data, while taking into consideration the impacts of the interactions of region-population

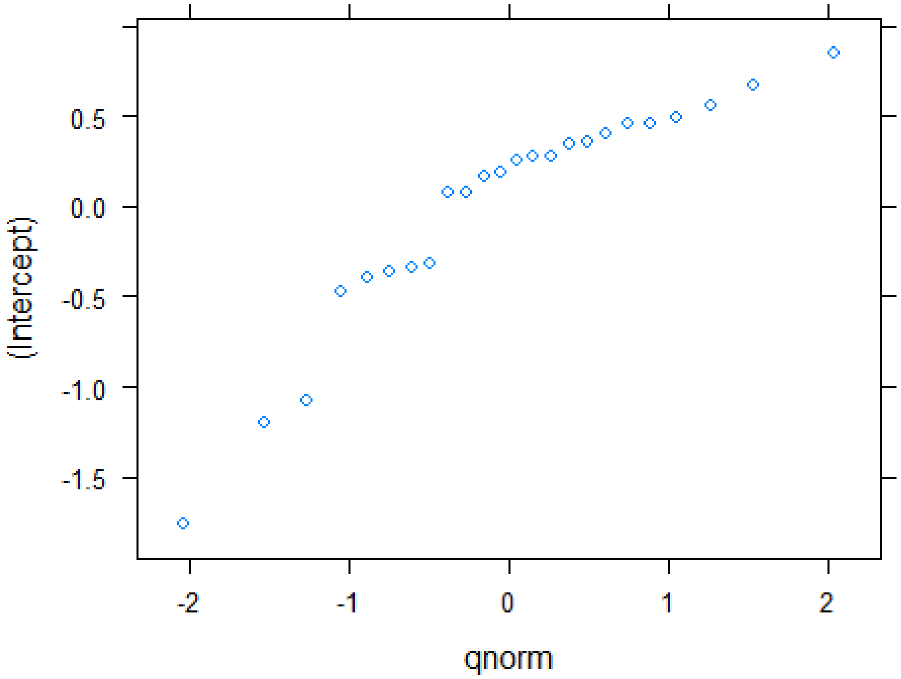
I:

```
anova(lme.arab1,lme.arab2,lme.arab3,lme.arab4,lme.arab5)

## refitting model(s) with ML (instead of REML)

## Data: arab
## Models:
## lme.arab1: total.fruits ~ 1 | gen
## lme.arab2: total.fruits ~ (1 | popu) + (1 | reg)
## lme.arab3: total.fruits ~ (1 | arab$reg) + (1 | reg/popu)
## lme.arab4: total.fruits ~ nutrient | reg
## lme.arab5: total.fruits ~ nutrient | reg/popu
##           npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)    
## lme.arab1    3 2279.2 2292.5 -1136.6   2273.2                         
## lme.arab2    4 2261.3 2279.1 -1126.7   2253.3 19.870  1  8.289e-06 ***
## lme.arab3    5 2263.3 2285.5 -1126.7   2253.3  0.000  1          1    
## lme.arab4    5 2187.6 2209.8 -1088.8   2177.6 75.724  0               
## lme.arab5    8 2138.9 2174.4 -1061.5   2122.9 54.687  3  8.006e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

AIC(lm.arab)

## [1] 2233.127
```

Based on the analysis we performed, we can see that the last model (LM5) is the most accurate in regard to it’s AIC. (it has the lowest AIC of the models) We manually extracted the AIC level of lm.arab (since it is linear)

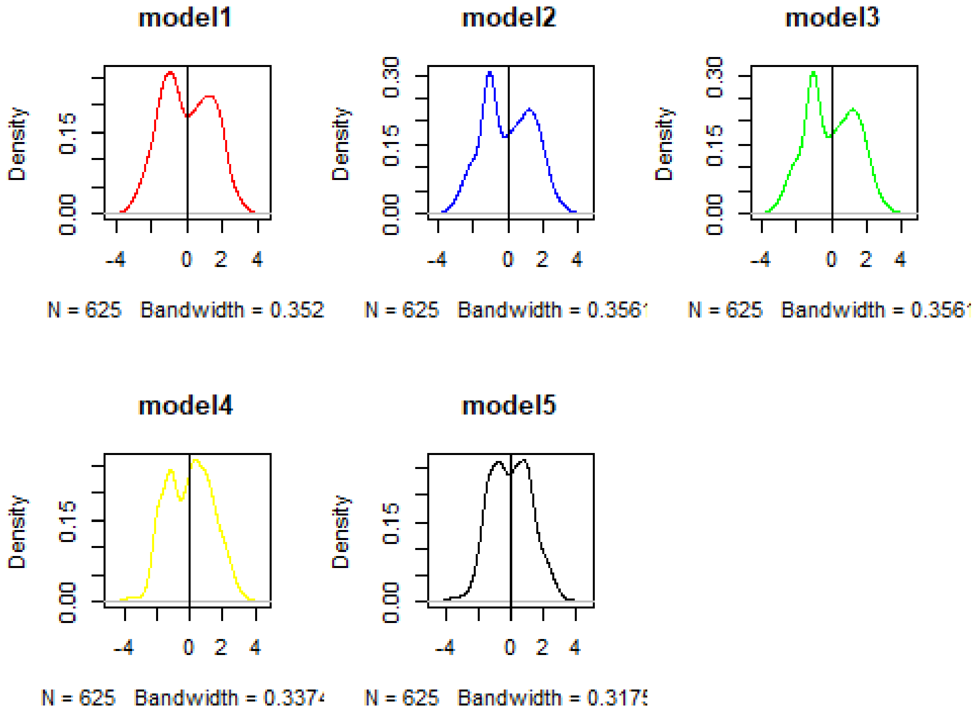
J:

```
plot(ranef(lme.arab1))

## $gen
```



We plotted the density of random genotype effects (intercepts) from model_a section5. We can see that it’s distribution is quite similar to the QQplot’s normal distribution. K:

```r
par(mfrow=c(2,3))
plot(density(residuals(lme.arab1)),col="red", main="model1")
abline(v=0)
plot(density(residuals(lme.arab2)),col="blue", main="model2")
abline(v=0)
plot(density(residuals(lme.arab3)),col="green", main="model3")
abline(v=0)
plot(density(residuals(lme.arab4)),col="yellow", main="model4")
abline(v=0)
plot(density(residuals(lme.arab5)),col="black", main="model5")
abline(v=0)
```



Based on what we see, model 5 is most zero-centered.

```
detach(arab)
```