Assignment_2

Dan Boguslavsky & Nadav Livneh

5/24/2020

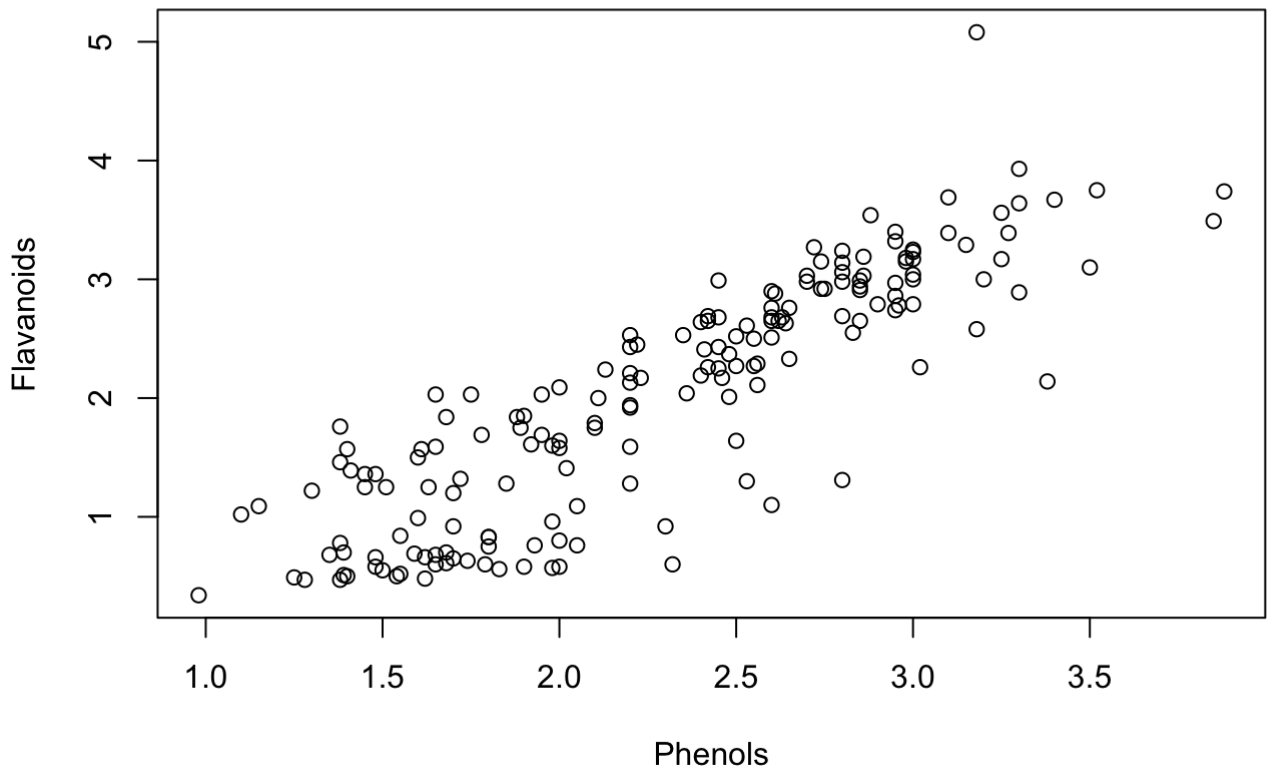
```
rm(list=ls())
```

#Question 1

```
#install.packages('rattle.data')
library('rattle.data')
wine<-rattle.data::wine
?wine
```

#1.a.

```
plot(Flavanoids~Phenols,data = wine)
```



It seems that indeed, Flavanoids and Phenols have some linear relationship between them.

#1.b. The model: Flavanoids = β0 + β1*Phenols + error.

We don’t assume any assumptions. In this course our goal is to make good predictions using correlations and relationships between features, but not describing a phenomenon or infer a cause, so the assumptions from econometrics class are not relevant.

#1.c. we want to minimize the sum of residual squares, so to get those expressions we need to derivate the sum by β0 and β1 and equal the equations to '0'. Then solving the equations, and find β0 and β1.

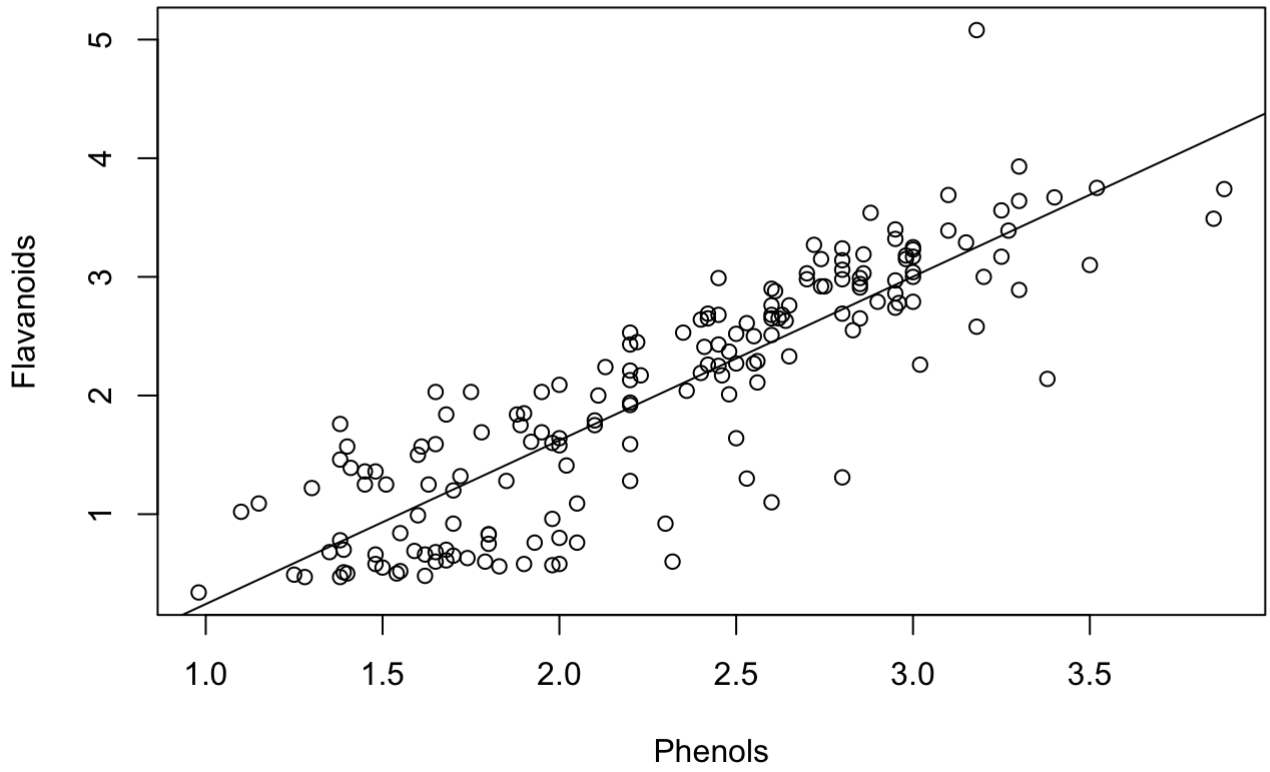
We assumed nothing to solve this minimization problem and to get the linear coefficients. Assumptions are required only if we want to infer a cause with those coefficients.

#1.d.

```
lm.Flavanoids.Phenols<- lm(wine$Flavanoids~wine$Phenols)
#Estimation resaults:
predict(lm.Flavanoids.Phenols)
```

```
##         1         2         3         4         5         6         7         8
## 2.7259358 2.5189592 2.7259358 4.1747719 2.7259358 3.3744624 2.3119826 2.4499670
##         9        10        11        12        13        14        15        16
## 2.7259358 2.9743077 2.9329124 1.8980295 2.4499670 3.1398890 3.4158578 2.7949280
##        17        18        19        20        21        22        23        24
## 2.7259358 2.9329124 3.4158578 2.5879514 3.0019046 2.1877967 2.4637655 2.2843857
##        25        26        27        28        29        30        31        32
## 2.3533779 2.4913623 2.7949280 2.1739982 2.9329124 2.5189592 3.0019046 2.8087264
##        33        34        35        36        37        38        39        40
## 2.2015951 2.9329124 2.1050060 2.5879514 2.4499670 2.2429904 2.1739982 3.0019046
##        41        42        43        44        45        46        47        48
## 3.2088812 2.2429904 3.3468656 2.5051608 3.0019046 2.7949280 3.3468656 3.1398890
##        49        50        51        52        53        54        55        56
## 2.6569436 2.8363233 2.6155483 2.2429904 4.2161672 3.0019046 2.4499670 2.9467108
##        57        58        59        60        61        62        63        64
## 3.2778734 3.0019046 3.5538421 1.5944638 1.6910529 1.6496575 1.7600451 3.6918265
##        65        66        67        68        69        70        71        72
## 1.4702778 2.2015951 2.9743077 1.7738435 2.3533779 1.4150841 0.3802011 2.9329124
##        73        74        75        76        77        78        79        80
## 1.4564794 3.4158578 3.5262453 1.0839215 1.5530685 1.2357044 1.4840763 2.7673311
##        81        82        83        84        85        86        87        88
## 2.2015951 1.8980295 1.6220607 1.1391153 1.8980295 1.8980295 1.3184950 1.5116732
##        89        90        91        92        93        94        95        96
## 1.5530685 1.8980295 1.0701231 0.8631465 0.7665574 2.2429904 3.0295015 2.3119826
##        97        98        99       100       101       102       103       104
## 1.0701231 2.3809748 3.7194234 2.7949280 1.9394248 0.8631465 2.3947733 2.3119826
##       105       106       107       108       109       110       111       112
## 1.8980295 1.1805106 1.1391153 0.7665574 2.1188045 2.6431452 3.2502765 2.3809748
##       113       114       115       116       117       118       119       120
## 1.2770997 2.2843857 2.3947733 2.2567889 1.5944638 1.6220607 1.1115184 1.6220607
##       121       122       123       124       125       126       127       128
## 2.8639202 3.2502765 1.8980295 2.4775639 2.8087264 2.4499670 2.6431452 1.8014404
##       129       130       131       132       133       134       135       136
## 1.9256263 1.7600451 0.9459372 0.6561699 0.4491933 1.2081075 1.6220607 1.0977200
##       137       138       139       140       141       142       143       144
## 0.7665574 1.3322934 1.0977200 2.0636107 0.9873325 0.7941543 1.0011309 1.6220607
##       145       146       147       148       149       150       151       152
## 0.7665574 0.9321387 0.2146199 1.2081075 1.5254716 0.8079528 0.7941543 0.9045418
##       153       154       155       156       157       158       159       160
## 1.8980295 1.3460919 0.9045418 1.2633013 1.3460919 1.4840763 2.7259358 2.4499670
##       161       162       163       164       165       166       167       168
## 2.0360138 1.3874872 1.1391153 0.7803559 0.7251621 0.6285731 1.2081075 0.9045418
##       169       170       171       172       173       174       175       176
## 1.0011309 1.5944638 0.5871777 0.7803559 1.1805106 1.1805106 1.3460919 1.0563247
##       177       178
## 1.1391153 1.6910529
```

```
#Plot + regression line:
plot(Flavanoids~Phenols,data = wine)
abline(coef(lm.Flavanoids.Phenols)[1:2])
```



#1.e.

```
#Slope coefficient:
coef(lm.Flavanoids.Phenols)[2:2]
```

```
## wine$Phenols
##     1.379844
```

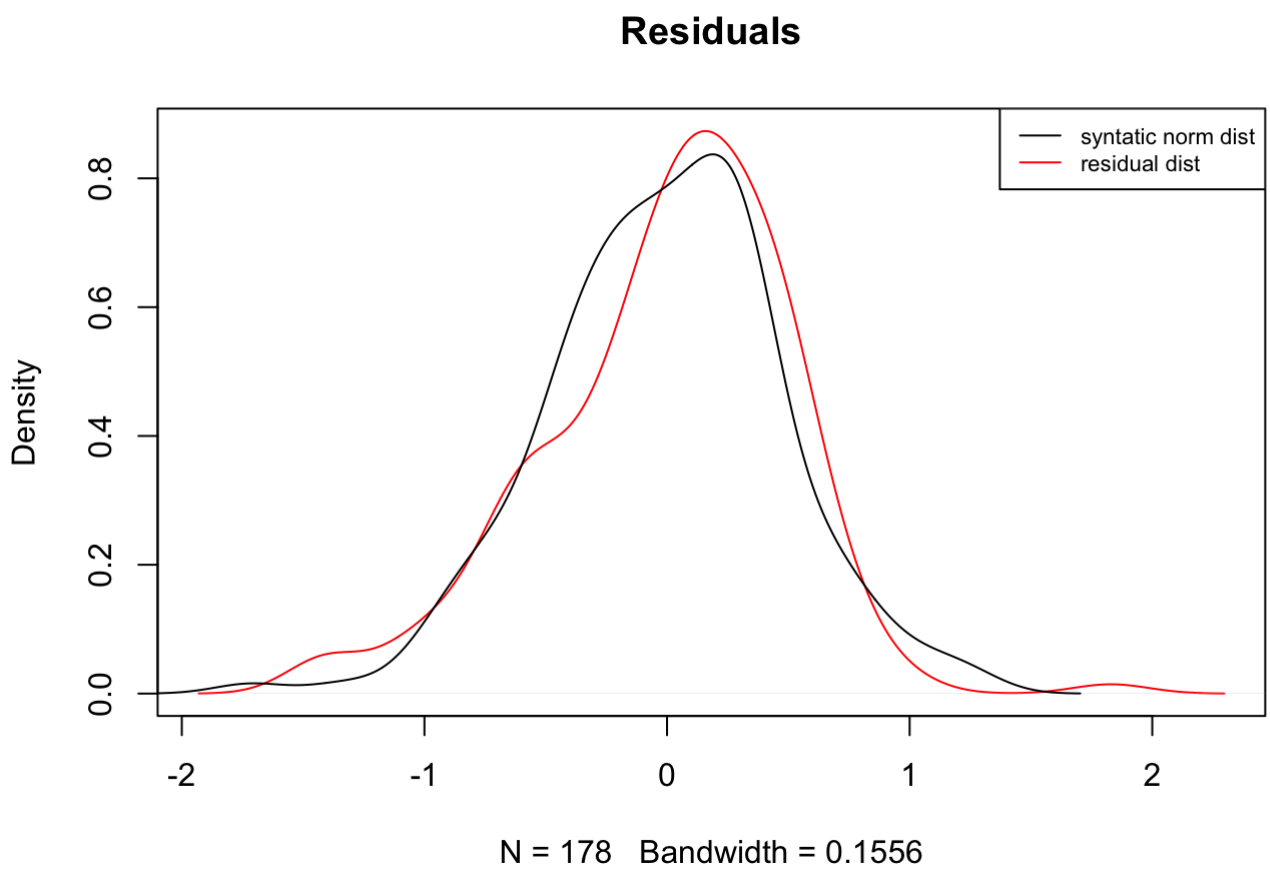
```
summary(lm.Flavanoids.Phenols)
```

```
##
## Call:
## lm(formula = wine$Flavanoids ~ wine$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$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
```

The slope coefficient is significant. We base it on the t-value of the effect - which is 22.82. That indicates a very significant effect. We can also be assisted by the “star” code on the right of the “summary” command - "***" means a significance level of under 0.001% for the effect to be insignificant.

#1.f.

```
plot(density(lm.Flavanoids.Phenols$residuals),main = "Residuals",col="red")
legend("topright", legend=c("syntatic norm dist", "residual dist"),
           col=c("black","red"),lty=1, cex=0.7)
set.seed(256)
lines(density(rnorm(1:178,0,0.5)))
```




```
shapiro.test(lm.Flavanoids.Phenols$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lm.Flavanoids.Phenols$residuals
## W = 0.96766, p-value = 0.0003786
```

We add a normal distribution line to compare it with the residual distribution. Additionally, we add a Shapiro-Wilk test in order to verify significantly if the residual distribution is normal. You can notice the similarity in the plot and the p-value of the Shapiro-Wilk test is 0.0003786 (extremely significant).

#1.g.

```
#β0,β1 calculation:
numerator<-c()
denominator<-c()
residuals.vec<-c()
mean_x<-mean(wine$Phenols)
mean_y<-mean(wine$Flavanoids)
for(i in 1:NROW(wine)){
  x_i<-wine$Phenols[i]
  y_i<-wine$Flavanoids[i]
  numerator<-c(numerator,((x_i-mean_x)*(y_i-mean_y)))
  denominator<-c(denominator,((x_i-mean_x)^2))
}#close_for
b.1 <-sum(numerator)/sum(denominator)
b.0 <- mean_y - b.1*mean_x
#Residuals calculation:
for(i in 1:NROW(wine)){
  y_i<-wine$Flavanoids[i]
  x_i<-wine$Phenols[i]
  y_hat<-x_i*b.1+b.0
  res<-((y_i-y_hat)*(y_i-y_hat))
  residuals.vec<-c(residuals.vec,res)
}#close_for
RSS <-sum(residuals.vec)
#R_squ calculation:
numerator<-c()
denominator<-c()
for(i in 1:NROW(wine)){
  y_i<-wine$Flavanoids[i]
  x_i<-wine$Phenols[i]
  y_hat<-x_i*b.1+b.0
  numerator<-c(numerator,((y_i-y_hat)^2))
  denominator<-c(denominator,((y_i-mean_y)^2))
}#close_for
R_sqr <-(1-(sum(numerator)/sum(denominator)))
b.0
```

```
## [1] -1.137627
```

```
b.1
```

```
## [1] 1.379844
```

```
RSS
```

```
## [1] 44.59583
```

```
R_sqr
```

```
## [1] 0.74747
```

```
#1.
print("#our computaion:")
```

```
## [1] "#our computaion:"
```

```
b.0
```

```
## [1] -1.137627
```

```
print("#from summary(lm):")
```

```
## [1] "#from summary(lm):"
```

```
lm.Flavanoids.Phenols$coefficients[1]
```

```
## (Intercept)
##   -1.137627
```

```
#2.
print("#our computaion:")
```

```
## [1] "#our computaion:"
```

```
b.1
```

```
## [1] 1.379844
```

```
print("#from summary(lm):")
```

```
## [1] "#from summary(lm):"
```

```
lm.Flavanoids.Phenols$coefficients[2]
```

```
## wine$Phenols
##     1.379844
```

```
#3.
print("#our computaion:")
```

```
## [1] "#our computaion:"
```

```
RSS
```

```
## [1] 44.59583
```

```
print("#from summary(lm):")
```

```
## [1] "#from summary(lm):"
```

```
sum(lm.Flavanoids.Phenols$residuals^2)
```

```
## [1] 44.59583
```

```
#4.
print("#our computaion:")
```

```
## [1] "#our computaion:"
```

```
R_sqr
```

```
## [1] 0.74747
```

```
print("#from summary(lm):")
```

```
## [1] "#from summary(lm):"
```

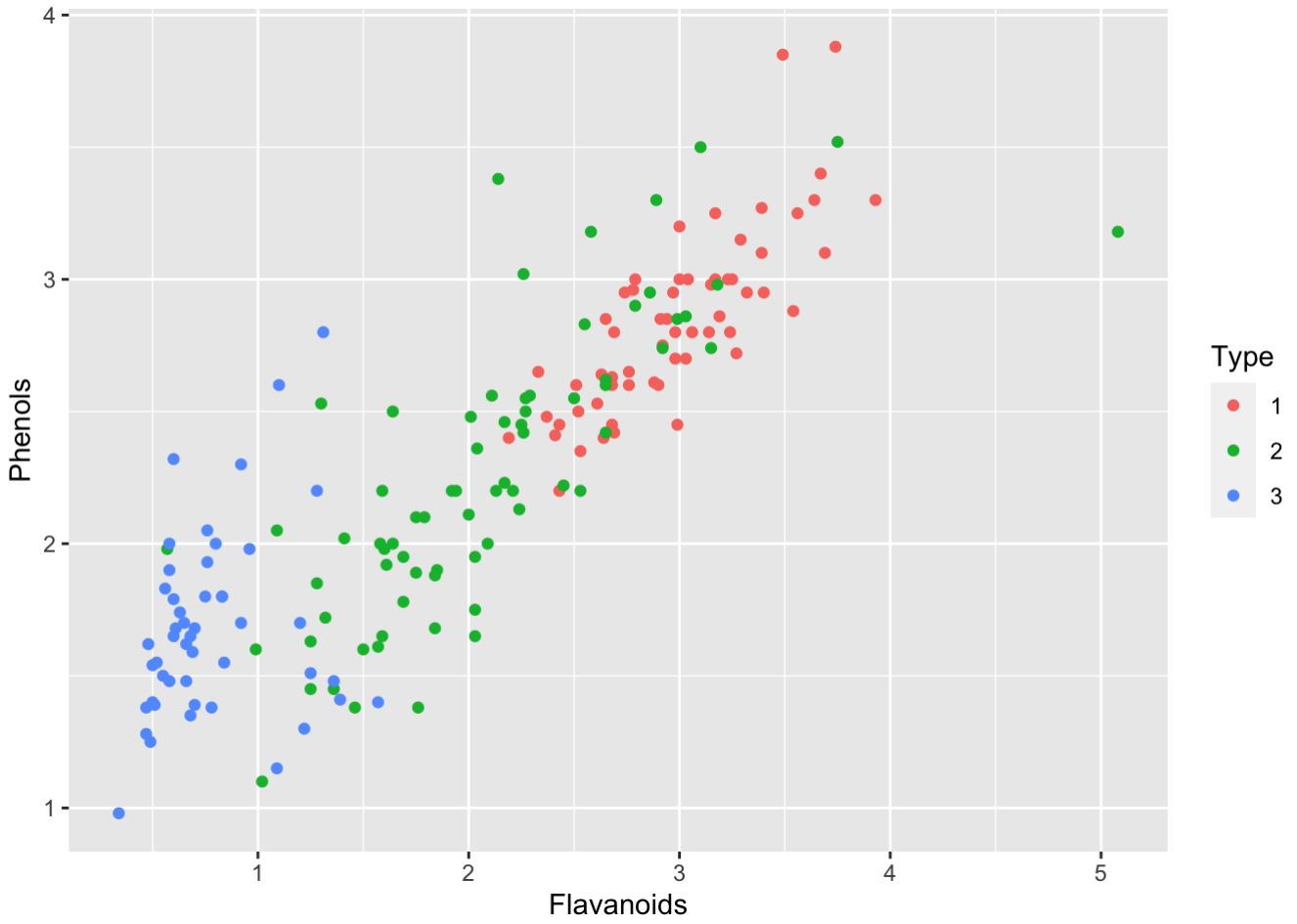
```
summary(lm.Flavanoids.Phenols)$r.squared
```

```
## [1] 0.74747
```

We can see that our computations are the same as the summary of the model presents.

#1.h.

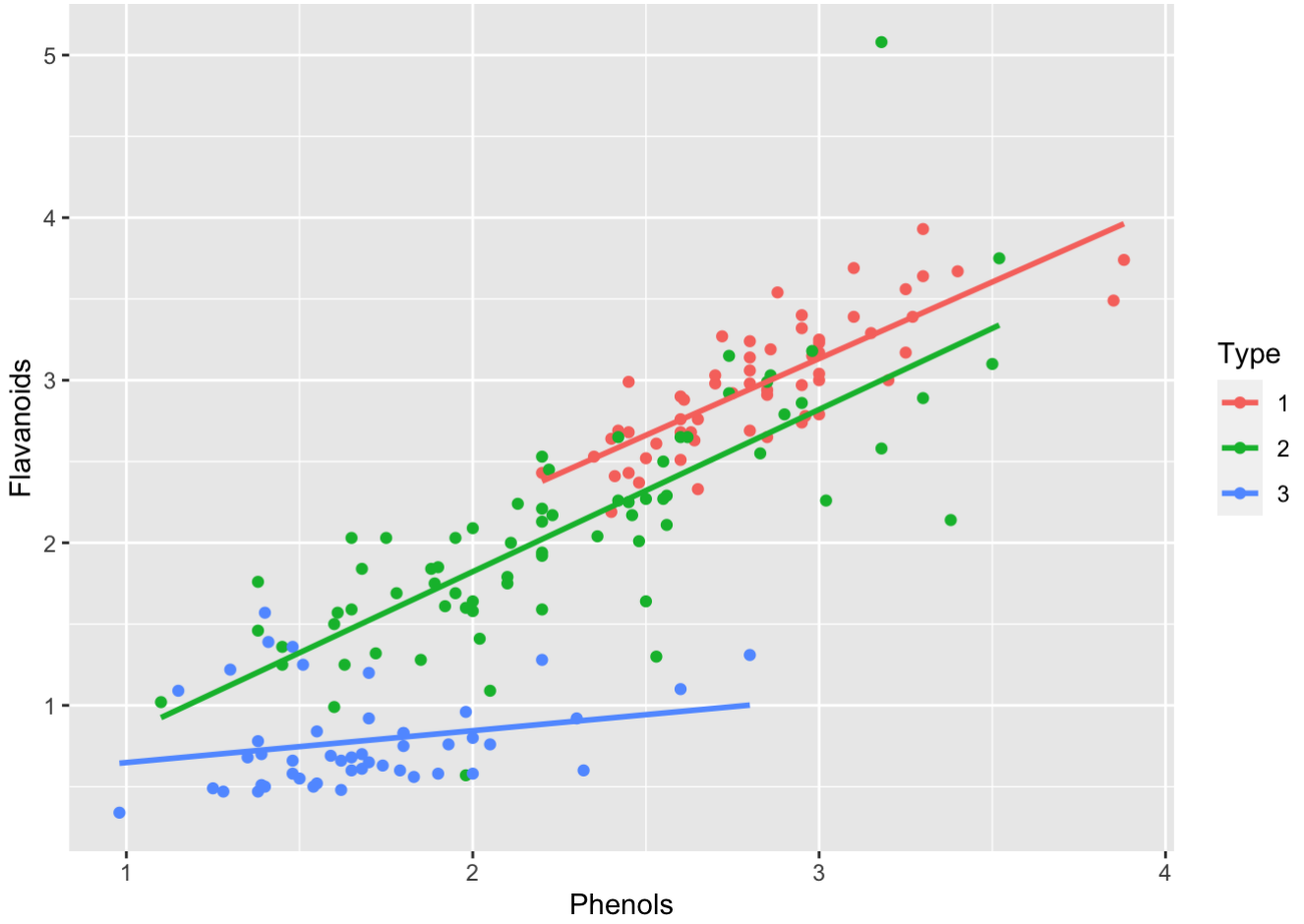
```
library("ggplot2")
ggplot(wine, aes(x=Flavanoids, y=Phenols, color = Type)) + geom_point()
```



#1.i.

```
type_1 <- subset(wine,Type==1)
type_2 <- subset(wine,Type==2)
type_3 <- subset(wine,Type==3)
lm_type_1 <- lm(type_1$Flavanoids ~ type_1$Phenols)
lm_type_2 <- lm(type_2$Flavanoids ~ type_2$Phenols)
lm_type_3 <- lm(type_3$Flavanoids ~ type_3$Phenols)
ggplot(wine, aes(x=Phenols, y=Flavanoids, color = Type)) + geom_point() + geom_smooth
(method = lm, se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



#1.j.

```
coef(lm_type_1)
```

```
##    (Intercept) type_1$Phenols
##      0.3052778      0.9425829
```

```
coef(lm_type_2)
```

```
##    (Intercept) type_2$Phenols
##     -0.1727831      0.9976780
```

```
coef(lm_type_3)
```

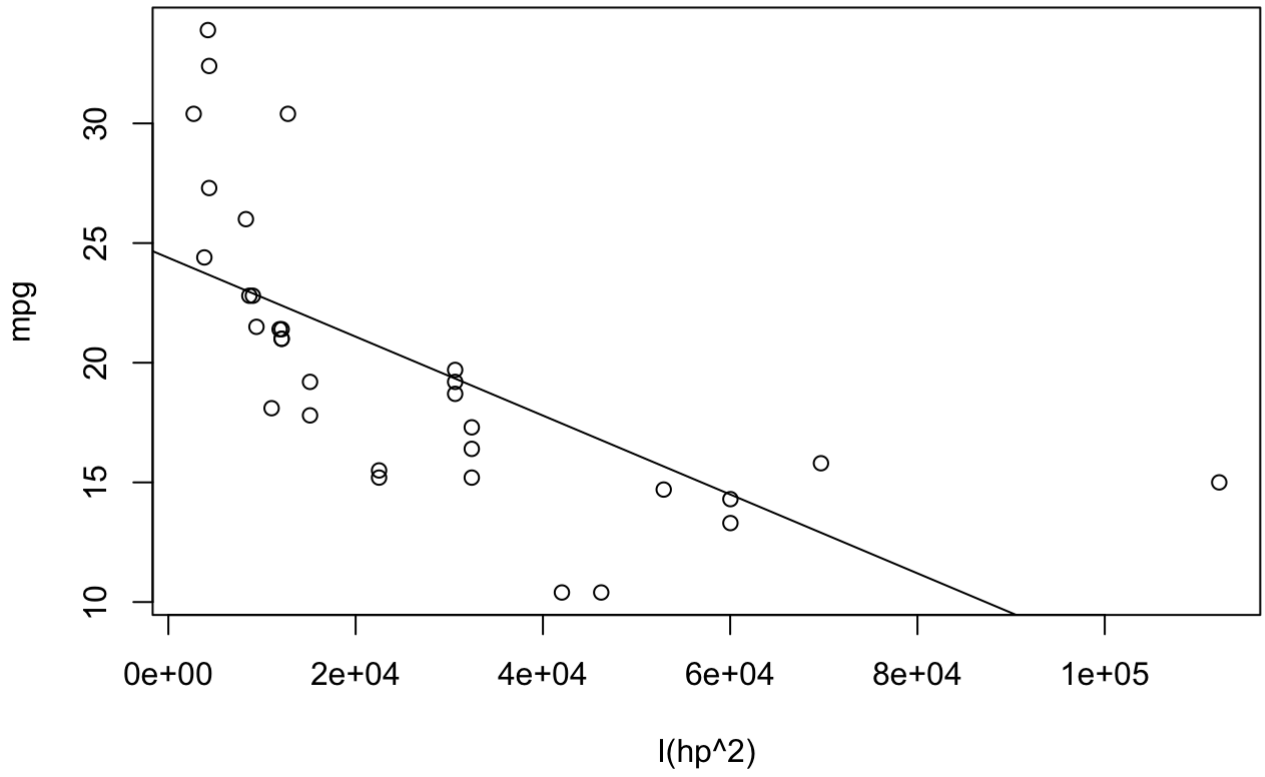
```
##    (Intercept) type_3$Phenols
##      0.4516892      0.1964373
```

#Question 2

```
mtcars<-mtcars
?mtcars
```

#2.a.

```
#Lets see if we can make a linear relation when we make the following model (When Fro
ss Horsepower is in polynomial of order 2 relation with Miles per gallon:
lm.mpg.hp_2<-lm(mpg~I(hp^2),data = mtcars)
plot(mpg~I(hp^2),data = mtcars)
abline(coef(lm.mpg.hp_2)[1:2])
```



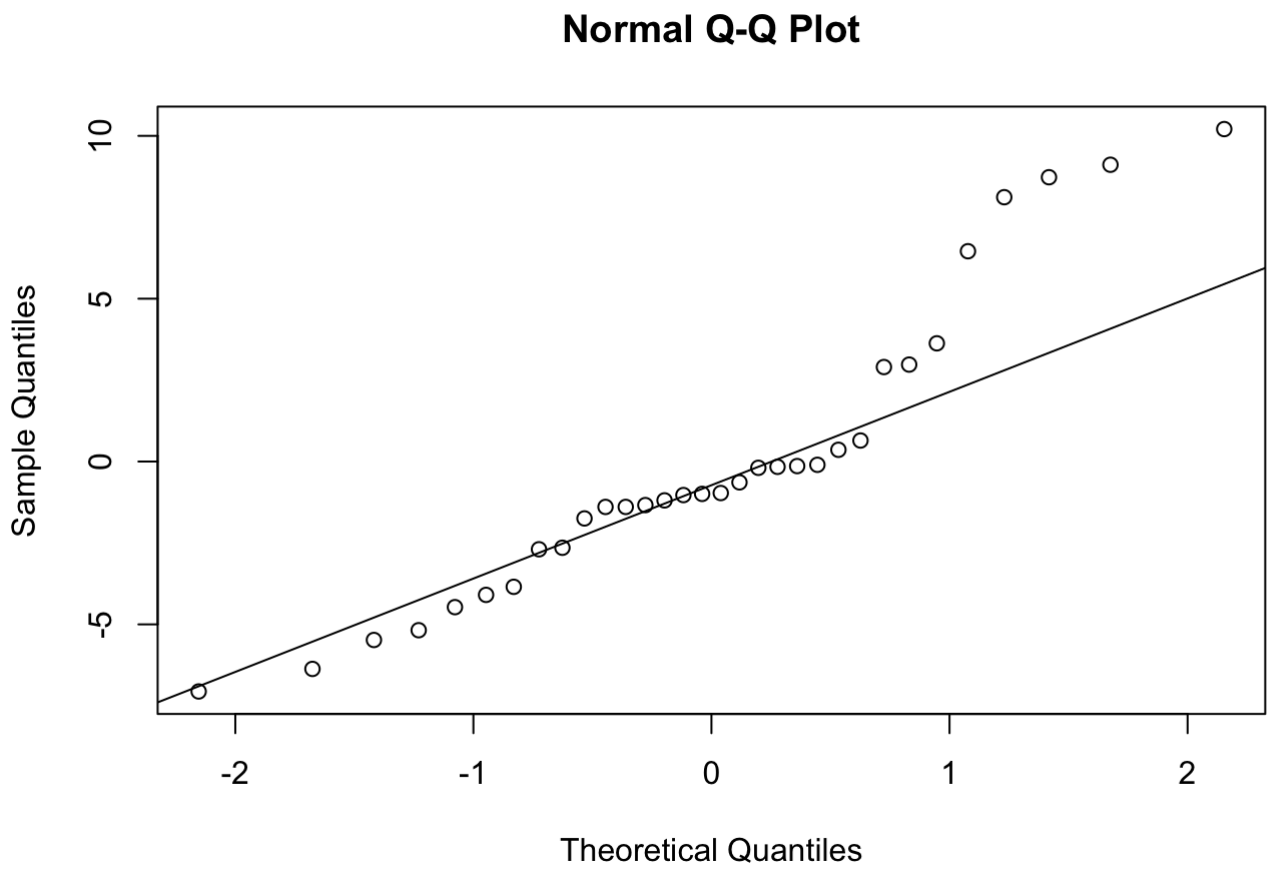

```
summary(lm.mpg.hp_2)
```

```
##
## Call:
## lm(formula = mpg ~ I(hp^2), data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.0605 -2.6591 -0.9808  1.2091 10.2078
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.439e+01  1.197e+00  20.371  < 2e-16 ***
## I(hp^2)     -1.649e-04  3.384e-05  -4.871 3.35e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.578 on 30 degrees of freedom
## Multiple R-squared:  0.4417, Adjusted R-squared:  0.423
## F-statistic: 23.73 on 1 and 30 DF,  p-value: 3.35e-05
```

The coefficients are significant by 99.9% so we might say that the effect is significant by using a polynomial of order 2 relation. Although, when plotting the relation - does it seems that the linear line fits the plot? Not likely… It looks like the variables have some other kind of relationship that fits better.

Let's try another approach to examine the fitting of this model by making a QQPLOT of the residuals.

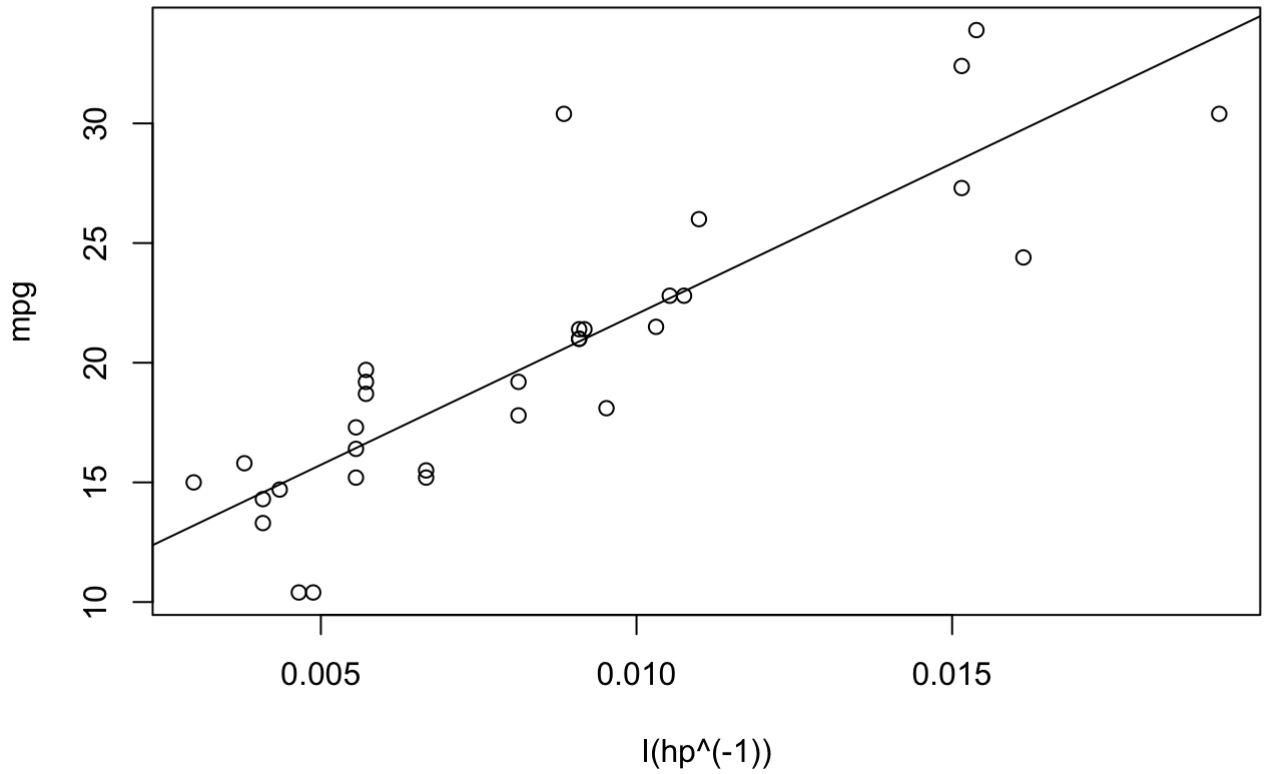
```
qqnorm(lm.mpg.hp_2$residuals); qqline(lm.mpg.hp_2$residuals)
```



Clearly, the dots are not aligned along the line on the top right corner.

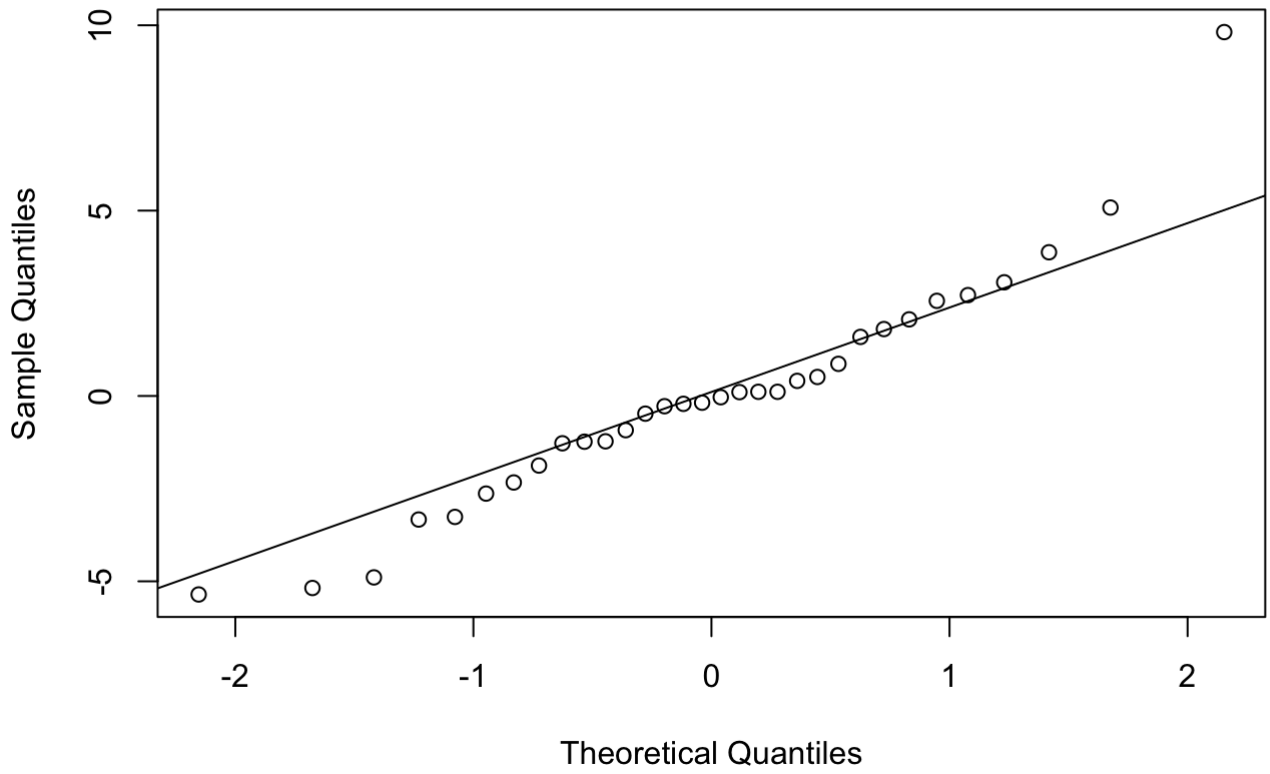
Though, it does seem like the relation of the variables is of y=1/x. Let's try it:

```
lm.mpg.hp_min_1<-lm(mpg~I(hp^(-1)),data = mtcars)
plot(mpg~I(hp^(-1)),data = mtcars)
abline(coef(lm.mpg.hp_min_1)[1:2])
```



This plot seems much better. We will test it with the QQPLOT as well:

```
qqnorm(lm.mpg.hp_min_1$residuals); qqline(lm.mpg.hp_min_1$residuals)
```



This time the dots are organized a lot better, but still not perfect.

#2.b.

```
library("ggplot2")
library(multcomp)
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: survival
```

```
## Loading required package: TH.data
```

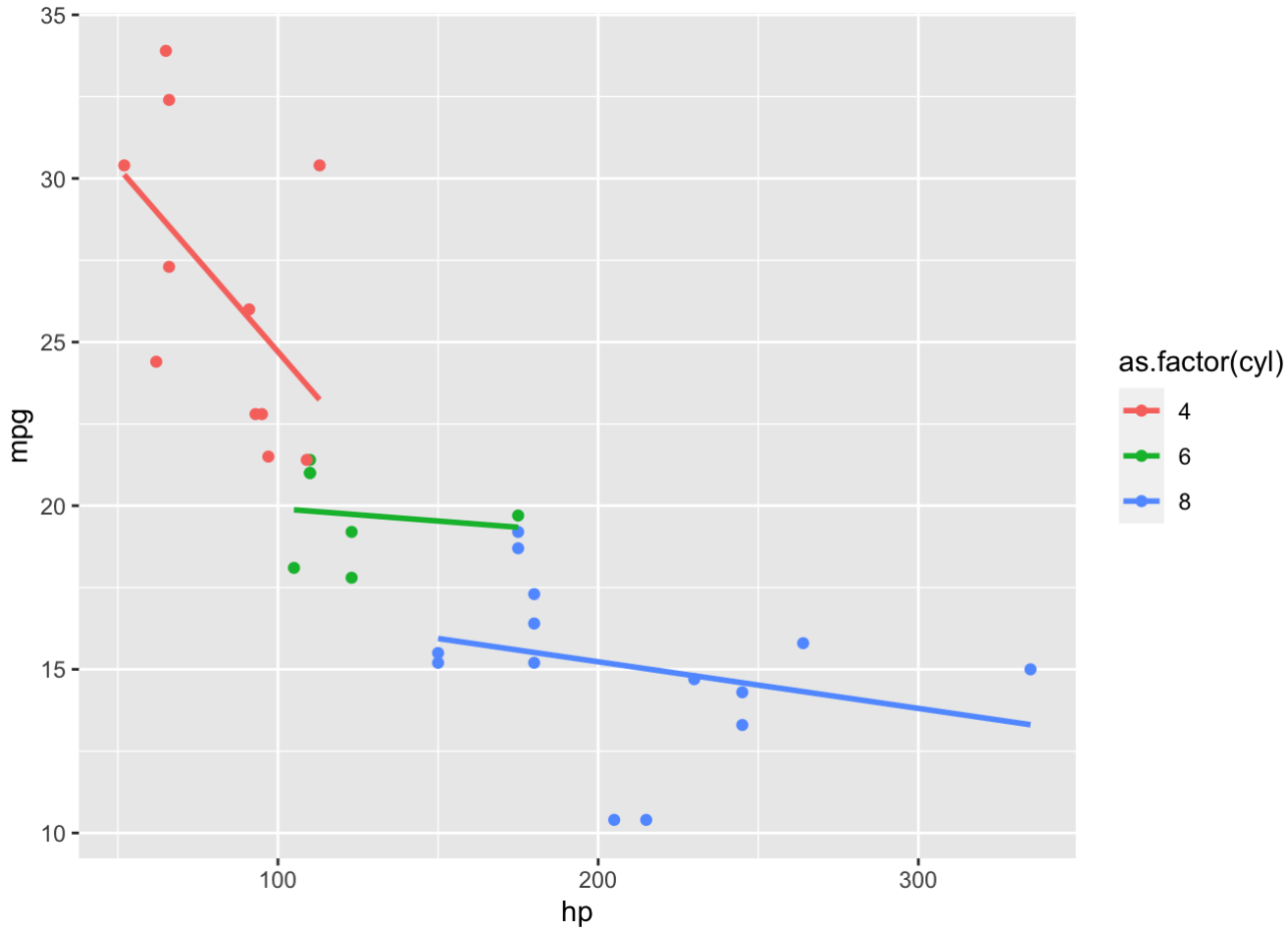
```
## Loading required package: MASS
```

```
##
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
lm_dif_sl_co <-lm(mpg ~ as.factor(cyl) + as.factor(cyl)*hp, data = mtcars)
ggplot(mtcars, aes(x=hp, y=mpg, color=as.factor(cyl))) + geom_point() + geom_smooth(m
ethod = lm, se = FALSE)
```

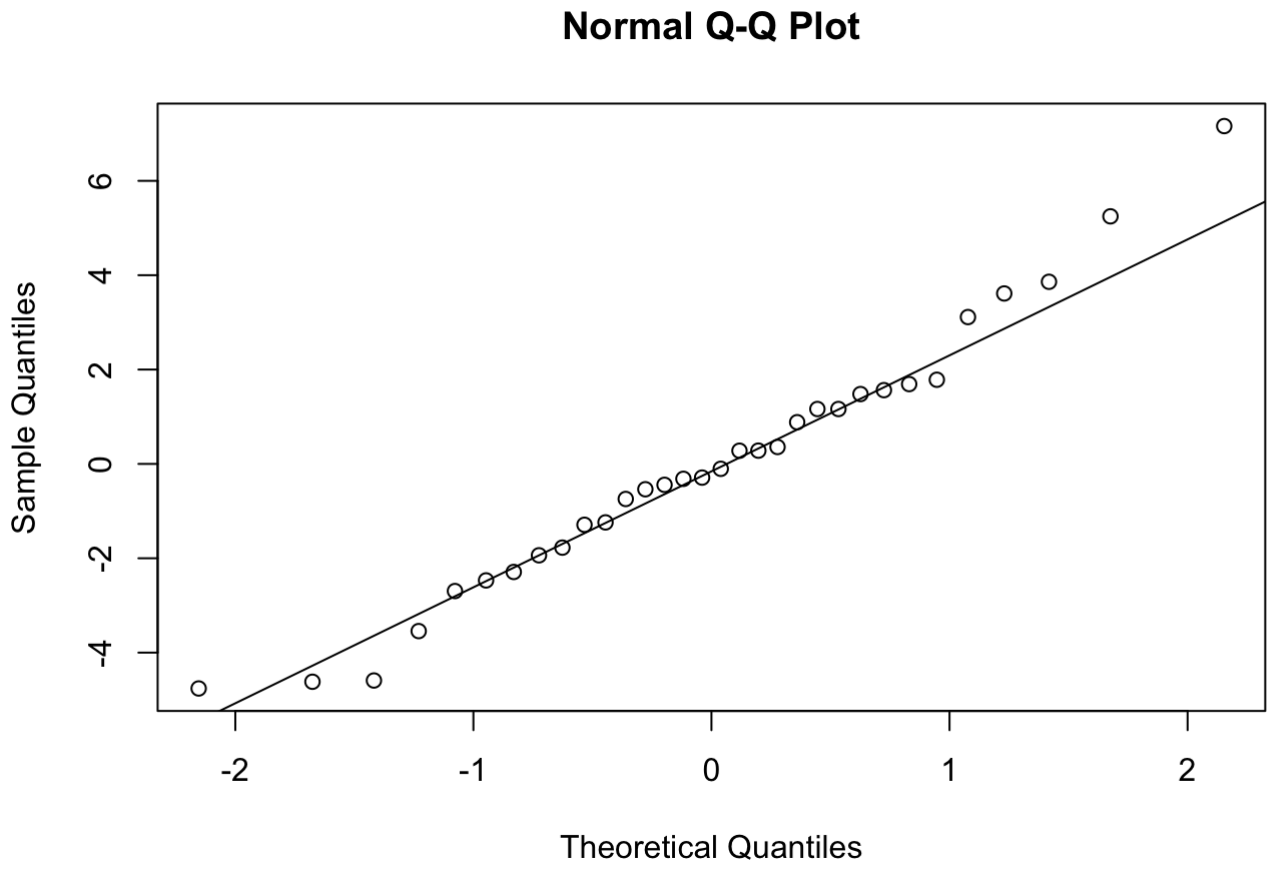
```
## `geom_smooth()` using formula 'y ~ x'
```




```
summary(lm_dif_sl_co)
```

```
##
## Call:
## lm(formula = mpg ~ as.factor(cyl) + as.factor(cyl) * hp, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7600 -1.8152 -0.1971  1.5012  7.1606
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         35.98303    3.88908   9.252 1.04e-09 ***
## as.factor(cyl)6    -15.30917    7.43456  -2.059  0.04962 *
## as.factor(cyl)8    -17.90295    5.25961  -3.404  0.00216 **
## hp                  -0.11278    0.04575  -2.465  0.02061 *
## as.factor(cyl)6:hp   0.10516    0.06848   1.536  0.13672
## as.factor(cyl)8:hp   0.09853    0.04862   2.026  0.05310 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.028 on 26 degrees of freedom
## Multiple R-squared:  0.7882, Adjusted R-squared:  0.7475
## F-statistic: 19.35 on 5 and 26 DF,  p-value: 5.019e-08
```

```
qqnorm(lm_dif_sl_co$residuals); qqline(lm_dif_sl_co$residuals)
```




```
hyp.test.mat1<- matrix(c(0,0,0,1,-1,0), nrow =  1)
hyp.test.mat2<- matrix(c(0,0,0,1,0,-1), nrow =  1)
hyp.test.mat3<- matrix(c(0,0,0,0,1,-1), nrow =  1)
hyp.test1 <- glht(lm_dif_sl_co, linfct=hyp.test.mat1)
summary(hyp.test1)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = mpg ~ as.factor(cyl) + as.factor(cyl) * hp, data = mtcars)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0  -0.2179     0.1047  -2.081   0.0474 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
hyp.test2 <- glht(lm_dif_sl_co, linfct=hyp.test.mat2)
summary(hyp.test2)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = mpg ~ as.factor(cyl) + as.factor(cyl) * hp, data = mtcars)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0 -0.21131    0.09297  -2.273   0.0315 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
hyp.test3 <- glht(lm_dif_sl_co, linfct=hyp.test.mat3)
summary(hyp.test3)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = mpg ~ as.factor(cyl) + as.factor(cyl) * hp, data = mtcars)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0 0.006631   0.053560   0.124    0.902
## (Adjusted p values reported -- single-step method)
```

The summary of the hypotheses test indicates that the slope of (cyl==4) group is different significantly from the other two groups, but the (cyl==6) group slope isn't different from the (cyl==8) group slope.

#2.c.

```
library(data.table)
mt_db <- as.data.table(mtcars)
auto_db <- mt_db[mt_db$am==0]
auto_db$eng <- ifelse (auto_db$vs==0, " v-shaped", " straight")
summary(lm(wt~as.factor(eng) , data = auto_db))
```

```
##
## Call:
## lm(formula = wt ~ as.factor(eng), data = auto_db)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72929 -0.45408 -0.04429  0.24571  1.31992
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.1943     0.2463  12.969 3.04e-10 ***
## as.factor(eng) v-shaped   0.9098     0.3099   2.936  0.00924 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6516 on 17 degrees of freedom
## Multiple R-squared:  0.3364, Adjusted R-squared:  0.2974
## F-statistic: 8.618 on 1 and 17 DF,  p-value: 0.009238
```

The hypothesis is wrong. The intercept is the weight of straight engine type and the v-shape coefficient is the additional weight of v-shape type cars to the intercept. It is negative so it means that v-shape type cars weight less than straight engine type cars, within the automatic cars group.

#2.d.

```
#install.packages('multcomp')
library(multcomp)
#First we run the model:
lm.mpg.disp<-lm(mpg~disp*I(disp>200), data = mtcars)
#Now lets make the test:
hyp.test.mat<- matrix(c(0,-1,0,1), nrow =  1)
hyp.test <- glht(lm.mpg.disp, linfct=hyp.test.mat)
summary(hyp.test)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = mpg ~ disp * I(disp > 200), data = mtcars)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0   0.2320     0.0365   6.357 7.04e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

We can see that the effects are significantly different(t value of 6.357), meaning the relationship is indeed changing when the Displacement is larger than 200.

#2.e.

```
lm.qsec.gear.drat<-lm(qsec~gear+drat, data = mtcars)
summary(lm.qsec.gear.drat)
```

```
##
## Call:
## lm(formula = qsec ~ gear + drat, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5501 -0.8906 -0.1583  0.8180  4.9530
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.0350     2.0736   8.215 4.66e-09 ***
## gear         -1.3116     0.5776  -2.271   0.0307 *
## drat          1.5711     0.7970   1.971   0.0583 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.695 on 29 degrees of freedom
## Multiple R-squared:  0.1581, Adjusted R-squared:  0.09999
## F-statistic: 2.722 on 2 and 29 DF,  p-value: 0.08253
```

The Number of forward gears (gear), does indeed affect the time it takes a car to pass 1/4 mile (qsec) when we control the Rear axle ratio (drat). [An addition gear *reduece* the time in 1.3116] We can see that the effect is significant with t-value of 2.271.

#Question 3 #3.a.

```
library(data.table)
wine_db <- as.data.table(rattle.data::wine)
wine_db$is_1 <- ifelse(wine_db$Type==1, 1, 0)
suppressWarnings(
step(glm(is_1~., data = wine_db, family = binomial))
)
```

```
## Start:  AIC=32
## is_1 ~ Type + Alcohol + Malic + Ash + Alcalinity + Magnesium +
##     Phenols + Flavanoids + Nonflavanoids + Proanthocyanins +
##     Color + Hue + Dilution + Proline
##
##                    Df   Deviance AIC
## - Type              2 6.0957e-09  28
## - Alcohol           1 1.0327e-09  30
## - Malic             1 1.0327e-09  30
## - Ash               1 1.0327e-09  30
## - Alcalinity        1 1.0327e-09  30
## - Magnesium         1 1.0327e-09  30
## - Phenols           1 1.0327e-09  30
## - Flavanoids        1 1.0327e-09  30
## - Nonflavanoids     1 1.0327e-09  30
## - Proanthocyanins   1 1.0327e-09  30
## - Color             1 1.0327e-09  30
## - Hue               1 1.0327e-09  30
## - Dilution          1 1.0327e-09  30
## - Proline           1 1.0327e-09  30
## <none>                1.0327e-09  32
##
## Step:  AIC=28
## is_1 ~ Alcohol + Malic + Ash + Alcalinity + Magnesium + Phenols +
##     Flavanoids + Nonflavanoids + Proanthocyanins + Color + Hue +
##     Dilution + Proline
##
##                    Df   Deviance AIC
## - Proanthocyanins   1 6.2329e-09  26
## - Phenols           1 6.2341e-09  26
## - Nonflavanoids     1 6.2458e-09  26
## - Magnesium         1 6.3019e-09  26
## - Color             1 6.3280e-09  26
## - Hue               1 6.4177e-09  26
## - Malic             1 6.4546e-09  26
## - Flavanoids        1 6.9301e-09  26
## - Dilution          1 6.9423e-09  26
## - Alcohol           1 1.0887e-08  26
## - Ash               1 1.2463e-08  26
## - Proline           1 1.4416e-08  26
## - Alcalinity        1 2.1390e-08  26
## <none>                6.0957e-09  28
##
## Step:  AIC=26
## is_1 ~ Alcohol + Malic + Ash + Alcalinity + Magnesium + Phenols +
##     Flavanoids + Nonflavanoids + Color + Hue + Dilution + Proline
##
##                   Df   Deviance AIC
## - Nonflavanoids    1 6.2717e-09  24
## - Magnesium        1 6.3123e-09  24
## - Phenols          1 6.4438e-09  24
## - Malic            1 6.4548e-09  24
## - Hue              1 6.6462e-09  24
## - Color            1 6.8688e-09  24
## - Dilution         1 6.9833e-09  24
## - Flavanoids       1 7.3415e-09  24
## - Alcohol          1 1.1836e-08  24
```

```
## - Ash            1 1.3148e-08  24
## - Proline        1 1.5792e-08  24
## - Alcalinity     1 2.2806e-08  24
## <none>             6.2329e-09  26
##
## Step:  AIC=24
## is_1 ~ Alcohol + Malic + Ash + Alcalinity + Magnesium + Phenols +
##     Flavanoids + Color + Hue + Dilution + Proline
##
##              Df   Deviance AIC
## - Magnesium   1 6.3623e-09  22
## - Phenols     1 6.4511e-09  22
## - Malic       1 6.6502e-09  22
## - Hue         1 6.7187e-09  22
## - Color       1 6.8716e-09  22
## - Flavanoids  1 7.3998e-09  22
## - Dilution    1 7.6295e-09  22
## - Alcohol     1 1.1972e-08  22
## - Proline     1 1.6781e-08  22
## - Alcalinity  1 2.3060e-08  22
## - Ash         1 2.7199e-08  22
## <none>          6.2717e-09  24
##
## Step:  AIC=22
## is_1 ~ Alcohol + Malic + Ash + Alcalinity + Phenols + Flavanoids +
##     Color + Hue + Dilution + Proline
##
##              Df   Deviance AIC
## - Phenols     1 6.4520e-09  20
## - Hue         1 6.7210e-09  20
## - Malic       1 6.7790e-09  20
## - Color       1 7.0710e-09  20
## - Flavanoids  1 7.5110e-09  20
## - Dilution    1 8.0750e-09  20
## - Alcohol     1 1.2241e-08  20
## - Proline     1 2.4526e-08  20
## - Alcalinity  1 2.8782e-08  20
## - Ash         1 3.3571e-08  20
## <none>          6.3620e-09  22
##
## Step:  AIC=20
## is_1 ~ Alcohol + Malic + Ash + Alcalinity + Flavanoids + Color +
##     Hue + Dilution + Proline
##
##              Df   Deviance AIC
## - Color       1 7.1700e-09  18
## - Malic       1 7.5400e-09  18
## - Hue         1 7.6000e-09  18
## - Dilution    1 8.2600e-09  18
## - Flavanoids  1 8.2700e-09  18
## - Alcohol     1 1.6670e-08  18
## - Proline     1 3.2360e-08  18
## - Ash         1 1.2190e-07  18
## - Alcalinity  1 2.0783e-06  18
## <none>          6.4500e-09  20
##
## Step:  AIC=18
## is_1 ~ Alcohol + Malic + Ash + Alcalinity + Flavanoids + Hue +
```

```
##     Dilution + Proline
##
##              Df Deviance    AIC
## - Hue         1    0.000 16.000
## - Flavanoids  1    0.000 16.000
## - Malic       1    0.000 16.000
## - Dilution    1    0.000 16.000
## - Alcohol     1    0.000 16.000
## - Proline     1    0.000 16.000
## - Ash         1    0.000 16.000
## <none>             0.000 18.000
## - Alcalinity  1   10.412 26.412
##
## Step:  AIC=16
## is_1 ~ Alcohol + Malic + Ash + Alcalinity + Flavanoids + Dilution +
##     Proline
##
##              Df Deviance    AIC
## - Flavanoids  1   0.0000 14.000
## - Dilution    1   0.0000 14.000
## - Malic       1   0.0000 14.000
## - Alcohol     1   0.0000 14.000
## - Proline     1   0.0000 14.000
## <none>            0.0000 16.000
## - Ash         1   5.2607 19.261
## - Alcalinity  1  12.0813 26.081
##
## Step:  AIC=14
## is_1 ~ Alcohol + Malic + Ash + Alcalinity + Dilution + Proline
##
##              Df Deviance    AIC
## - Malic       1   0.0000 12.000
## - Alcohol     1   0.0000 12.000
## - Proline     1   0.0000 12.000
## <none>            0.0000 14.000
## - Ash         1   6.3496 18.350
## - Dilution    1   9.8204 21.820
## - Alcalinity  1  16.2813 28.281
##
## Step:  AIC=12
## is_1 ~ Alcohol + Ash + Alcalinity + Dilution + Proline
##
##              Df Deviance    AIC
## <none>            0.0000 12.000
## - Proline     1   5.4619 15.462
## - Alcohol     1   8.3859 18.386
## - Ash         1   9.9134 19.913
## - Dilution    1  13.4267 23.427
## - Alcalinity  1  18.4407 28.441
```

```
##
## Call:  glm(formula = is_1 ~ Alcohol + Ash + Alcalinity + Dilution +
##     Proline, family = binomial, data = wine_db)
##
## Coefficients:
## (Intercept)      Alcohol          Ash   Alcalinity     Dilution      Proline
##  -853.80202     46.43607    109.66384    -13.24030     61.01085      0.08283
##
## Degrees of Freedom: 177 Total (i.e. Null);  172 Residual
## Null Deviance:       226.1
## Residual Deviance: 1.368e-08     AIC: 12
```

By using 'step' function we specipicied the best 5 relevant variables. The formula = is_1 ~ Alcohol + Ash + Alcalinity + Dilution + Proline. Meaning the model is: is_1 = β0 x Alcohol + β1 x Ash + β2 x Alcalinity + β3 x Dilution + β4 x Proline

#3.b.

```
glm_is_1 <-glm(formula = is_1 ~ Alcohol + Ash + Alcalinity + Dilution + Proline,
               data = wine_db, family = binomial) # Formula copyed from the step of s
teped_glm with 5 vars.
```

```
## Warning: glm.fit: algorithm did not converge
```

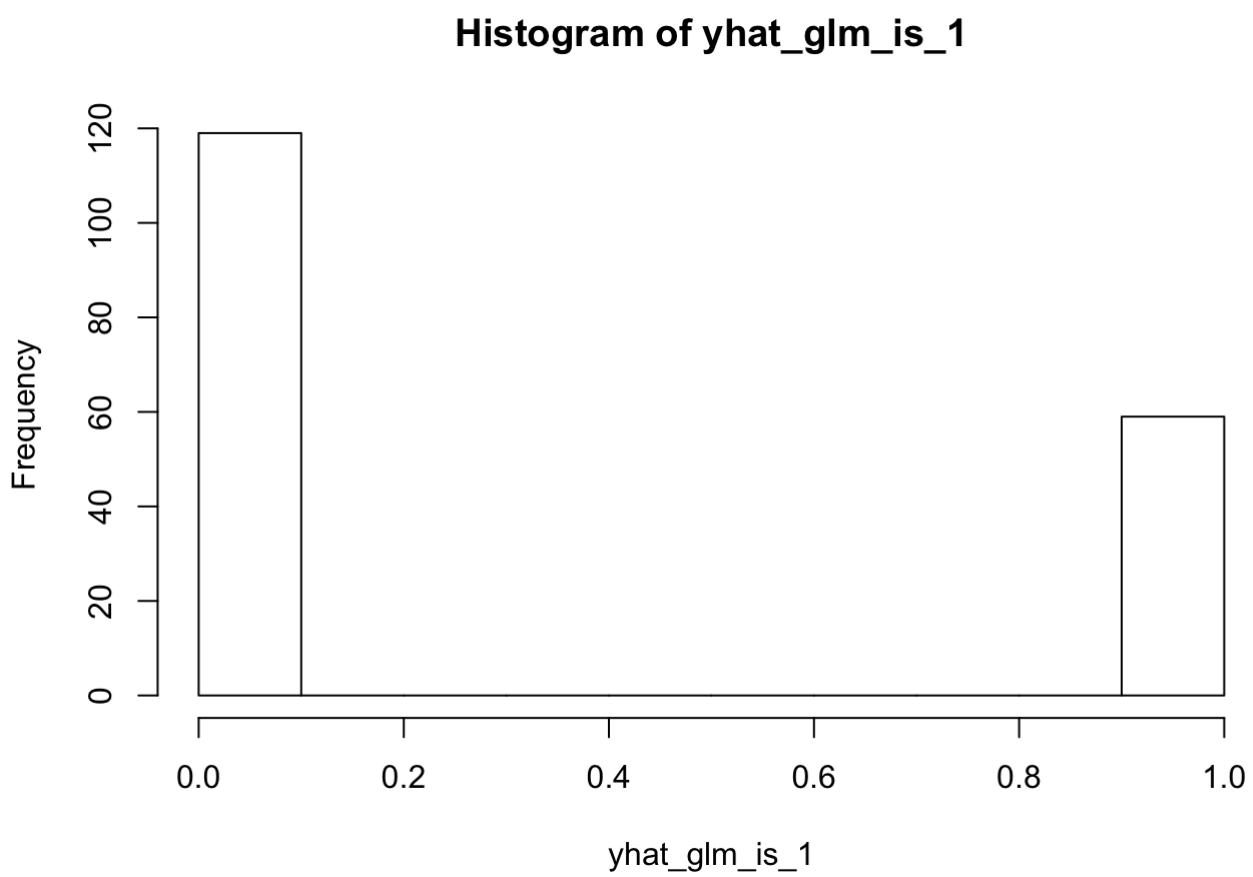
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
                                                # Changes when changing the amount
 of train/test of data.
coef(glm_is_1)
```

```
##   (Intercept)       Alcohol           Ash    Alcalinity      Dilution
## -853.80202082   46.43607397  109.66384017  -13.24029927   61.01084520
##       Proline
##    0.08283345
```

#3.c.

```
yhat_glm_is_1 <- predict(glm_is_1, wine_db, type = "response") ###only for test data
hist(yhat_glm_is_1)
```




```
yhat_glm_is_1_binar <- (yhat_glm_is_1>0.5)*1
print(paste('the mean of is_1 prediction is', mean(yhat_glm_is_1_binar), 'as positiv
 to be 1')) #classification rate???
```

```
## [1] "the mean of is_1 prediction is 0.331460674157303 as positiv to be 1"
```

```
acurate_rate_matrix <- table(true = wine_db$is_1, predicted = yhat_glm_is_1_binar) ##
#only for test data
print(paste('FALSE-POSITVE:',acurate_rate_matrix[1,1],'. FALSE-NEGATIVE:', acurate_ra
te_matrix[2,1], '. TRUE-NEGATIVE:', acurate_rate_matrix[1,2], '. TRUE-POSITIVE:', acu
rate_rate_matrix[2,2]))
```

```
## [1] "FALSE-POSITVE: 119 . FALSE-NEGATIVE: 0 . TRUE-NEGATIVE: 0 . TRUE-POSITIVE: 5
9"
```

```
acurate_rate_matrix
```

```
##     predicted
## true   0   1
##    0 119   0
##    1   0  59
```

```
print(paste('the accuracy rate is',sum(diag(acurate_rate_matrix)) / sum(acurate_rate_
matrix)))
```

```
## [1] "the accuracy rate is 1"
```

```
print(paste('the prcision rate is', Precision <- acurate_rate_matrix[4] / sum(acurate
_rate_matrix[,2])))
```

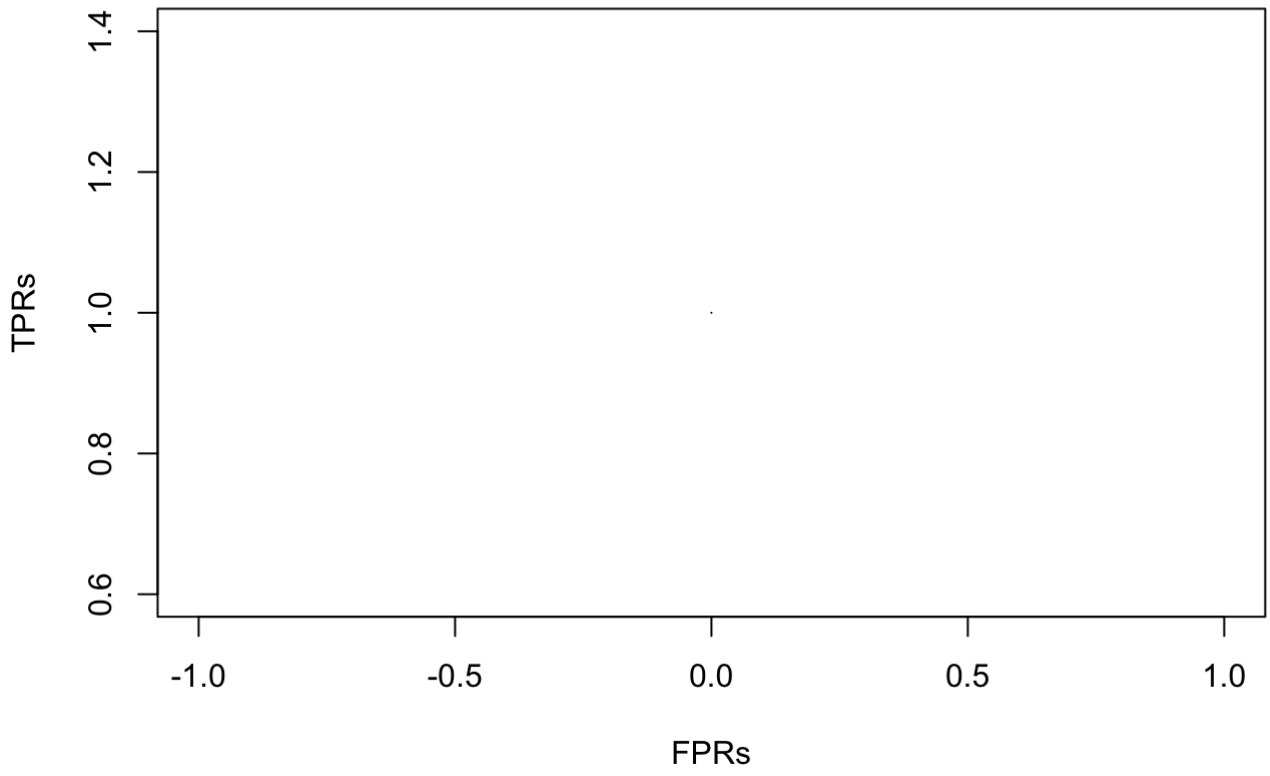
```
## [1] "the prcision rate is 1"
```

```
print(paste('the racall rate is',Recall <- acurate_rate_matrix[4] / sum(acurate_rate_
matrix[2,])))
```

```
## [1] "the racall rate is 1"
```

#3.d.

```
alphas <- seq(0,1,0.01)
TPRs <- numeric(length(alphas))
FPRs <- numeric(length(alphas))
for (i in seq_along(alphas)){
  pr_i <- ifelse(yhat_glm_is_1>alphas[i],1,0)
  CM_i <- table(wine_db$is_1, pr_i)  ###only for test data
  TPRs[i] <- CM_i[4] / sum(CM_i[2,])  # TP/TP+FN - regection from FN (high is good)
  FPRs[i] <- CM_i[3] / sum(CM_i[1,])  # FP/FP+FN - regection from FP (low is good)
}
plot(TPRs~FPRs, type = "l")
```



```
TPRs
```

```
##   [1] NA  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [26]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [51]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [76]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
## [101] NA
```

```
FPRs
```

```
##   [1] NA  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##  [26]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##  [51]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##  [76]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [101] NA
```

In each alpha level we get the same rate of correct predictions.

We will also compare the errors (FP/FN) vectors to find the optimal alpha, assuming we don't have a difference or preference between the types of errors (FP=FN).

```
CON_TPRS_FPRS <- TPRs-FPRs
CON_TPRS_FPRS[1] <- 0     # convert NA to 0 (first arg)
CON_TPRS_FPRS[101] <- 0   # convert NA to 0 (last arg)
max <- c(0,0)
for (i in 1:length(CON_TPRS_FPRS)) {
 if (CON_TPRS_FPRS[i] > max[1]) {
   max[1] <- CON_TPRS_FPRS[i]
   max[2] <- i
 }
}
alphas[max[2]]   # the alpha that gets the highest (TPRs-FPRs).
```

```
## [1] 0.01
```

#3.e. If there are 3 levels (for example) such as in our data - 'wine' DB, we need to generate 2 new variables - 'is_1' for the first type and 'is_2' for the second one. Now we need to run 2 glm's - for each variable separately and to choose the same formula for both. The third will be calculated from those two: "glm_is_3 = 1 - glm_is_1 - glm_is_2". With those three regressions we can estimate the 'chances' of each observation to get any of the levels and check the accuracy of the results. In this format we have only true/false but not positive/negative. We still can decide from which false we prefer to avoid (more than the others). It is the same for 'n' levels - generate 'n-1' new variables and run 'n-1' regressions and so on…

#Question 4

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

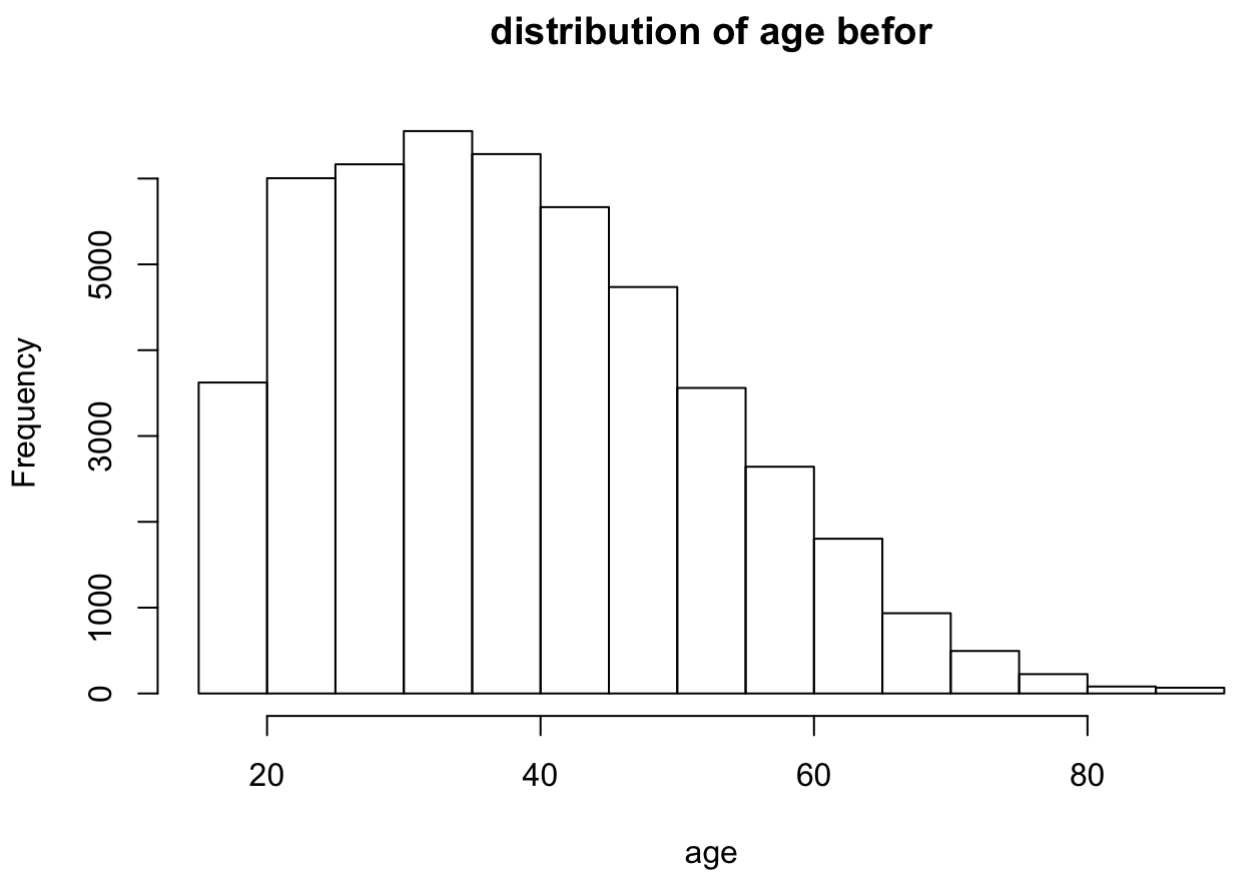
#4.a.

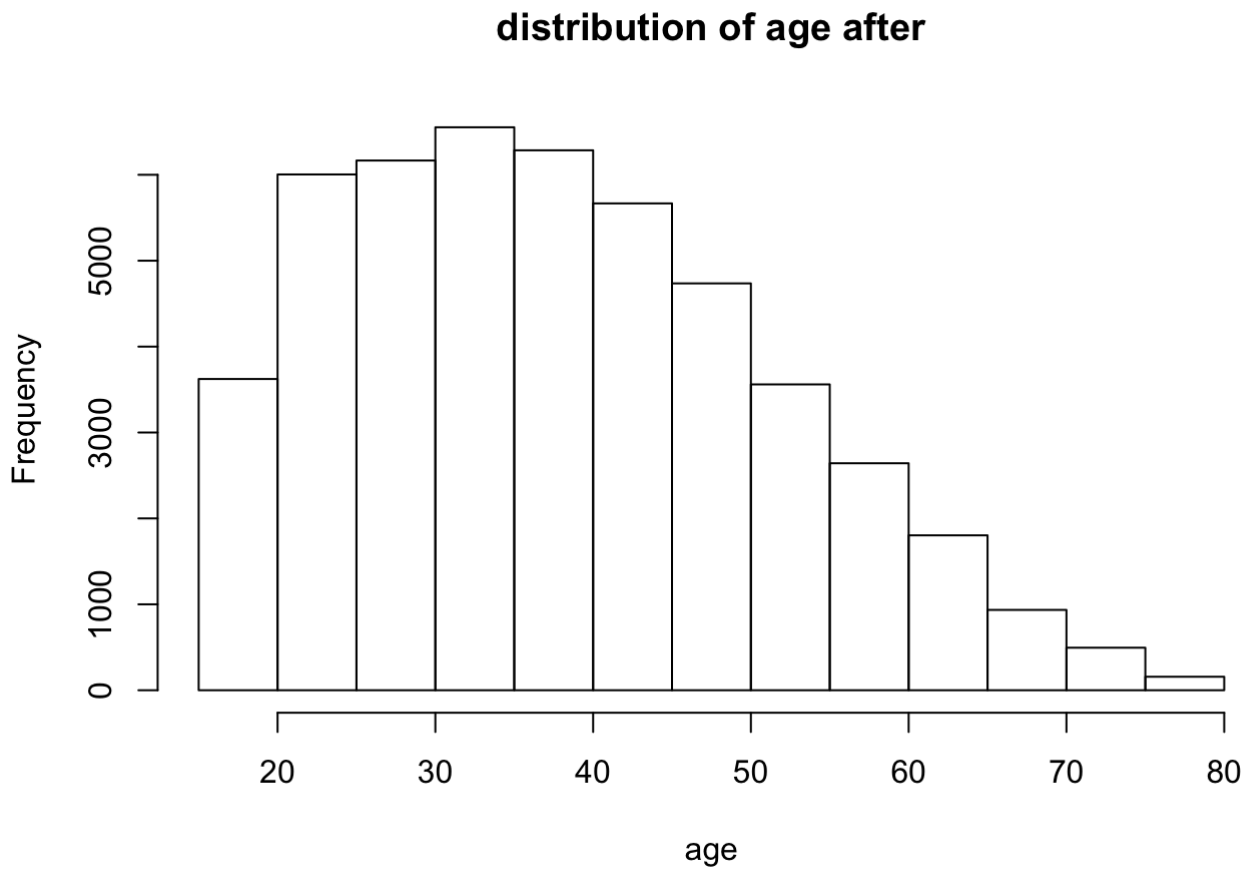
```
for (i in colnames(data)){
  if(class(data[[i]])=="integer"){print(paste("The feature " , i , "is continuous"))}
  else{print(paste("The feature " , i , "is " , class(data[[i]])))}
}
```

```
## [1] "The feature  age is continuous"
## [1] "The feature  workclass is  factor"
## [1] "The feature  education is  factor"
## [1] "The feature  educational.num is continuous"
## [1] "The feature  marital.status is  factor"
## [1] "The feature  race is  factor"
## [1] "The feature  gender is  factor"
## [1] "The feature  hours.per.week is continuous"
## [1] "The feature  income is  factor"
```

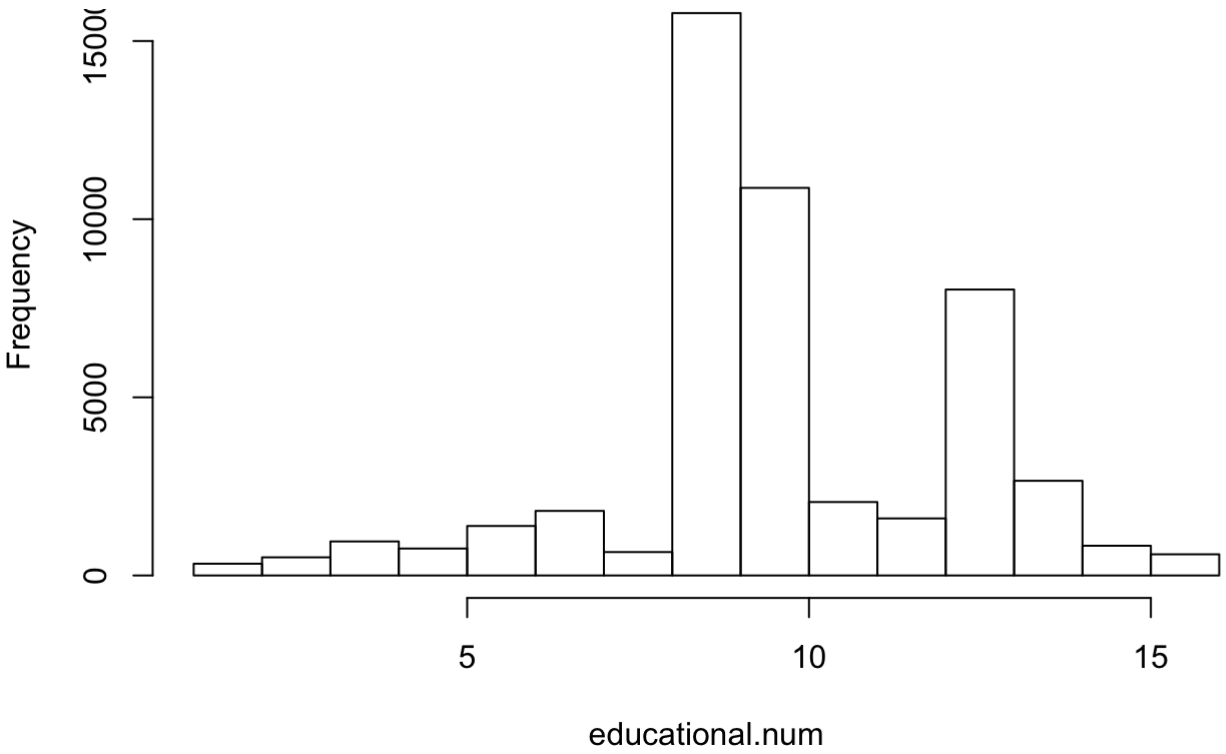
#4.b.

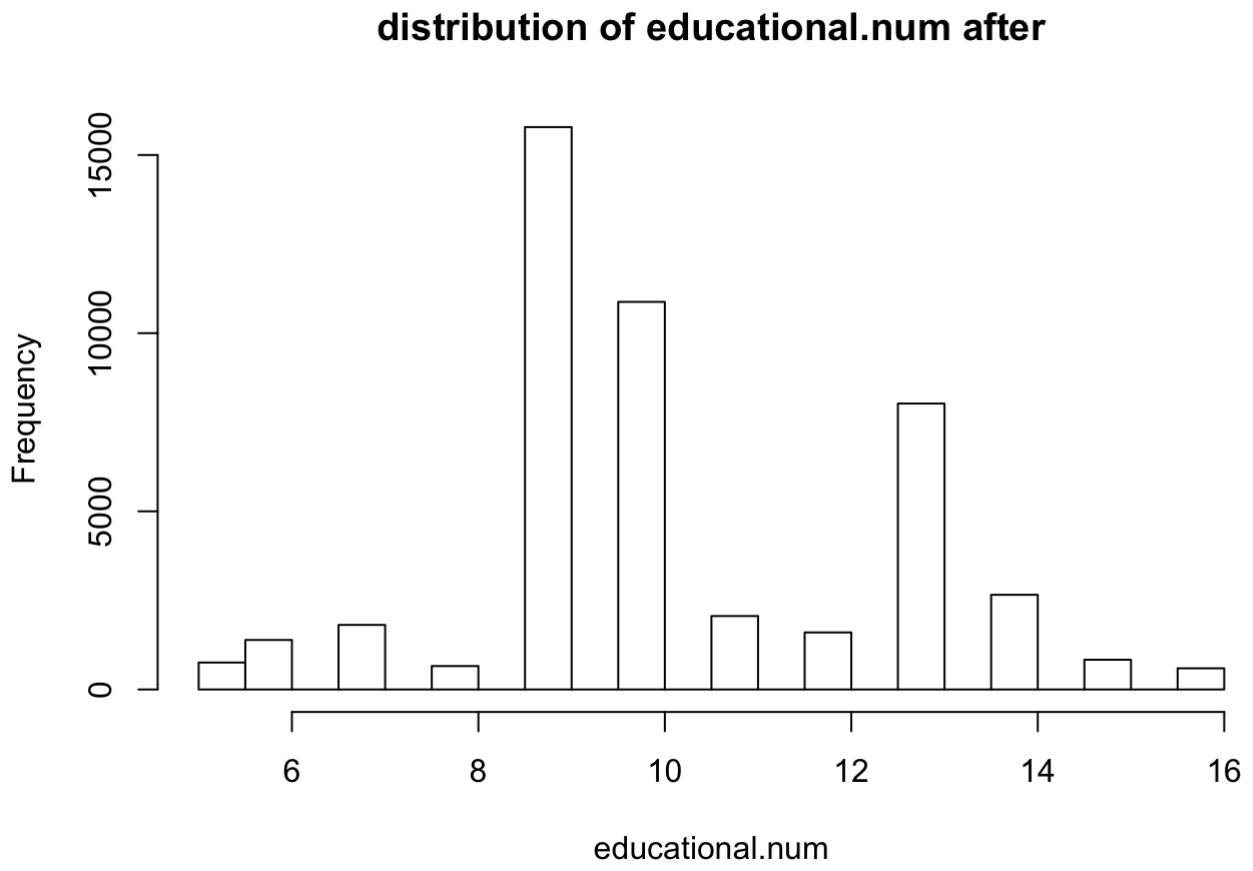
```
for (i in colnames(data)){
  if(class(data[[i]])=="integer"){
    hist(data[[i]],xlab=i,main=paste("distribution of",i, "befor"))
    temp_d <- data[[i]]
    qnt <- quantile(temp_d, probs=c(.25, .75), na.rm = T)
    a <- ifelse (i=="hours.per.week",5,1.5) # the anomality of hpw shoulde be wider s
o it dosen't drop too many data
    H <- a * IQR(temp_d, na.rm = T)
    data[[i]][temp_d < (qnt[1] - H)] <- NA
    data[[i]][temp_d > (qnt[2] + H)] <- NA
    hist(data[[i]],xlab=i,main=paste("distribution of",i, "after"))
    remove(H, qnt, temp_d)
  }#close_if
}#close_for
```

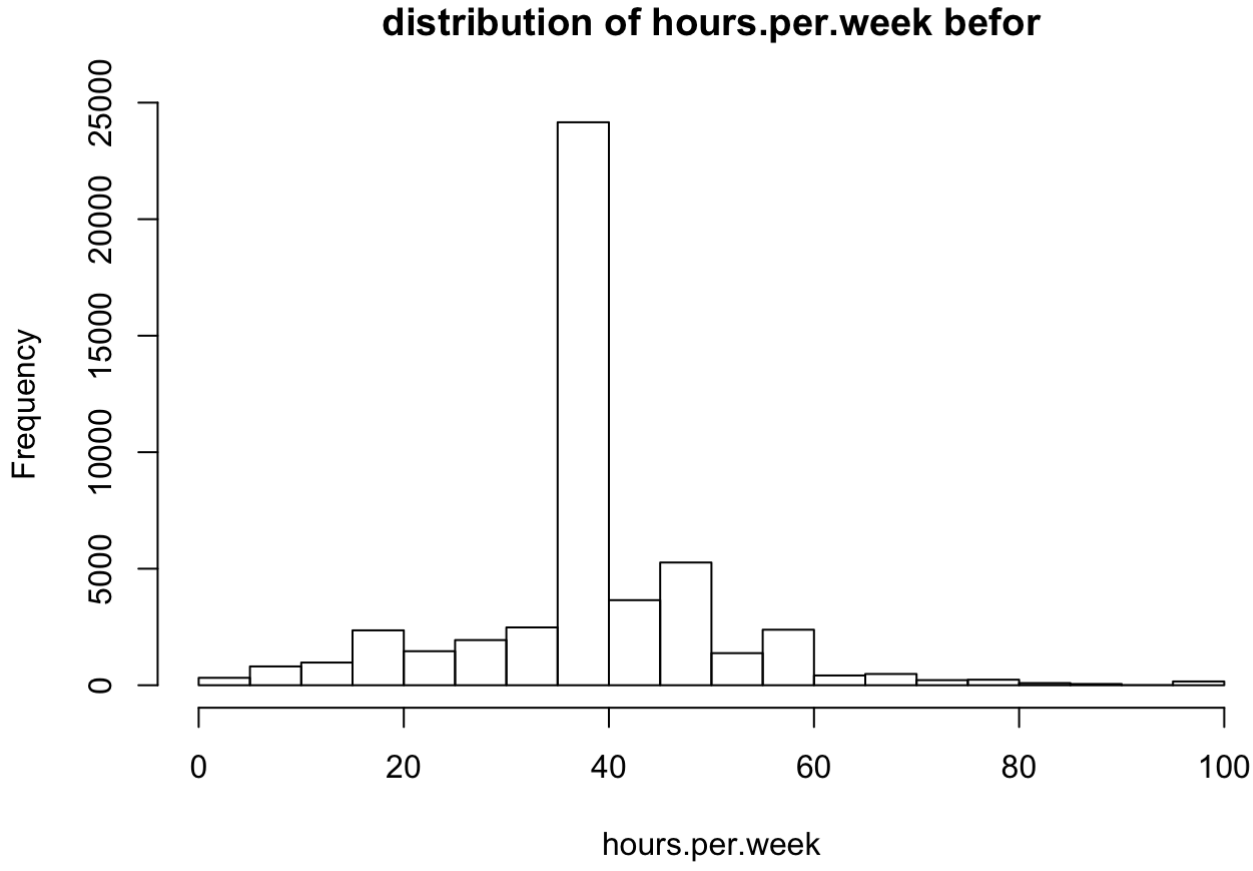

distribution of age befor
Frequency
0
1000
3000
5000
20
40
60
80
age

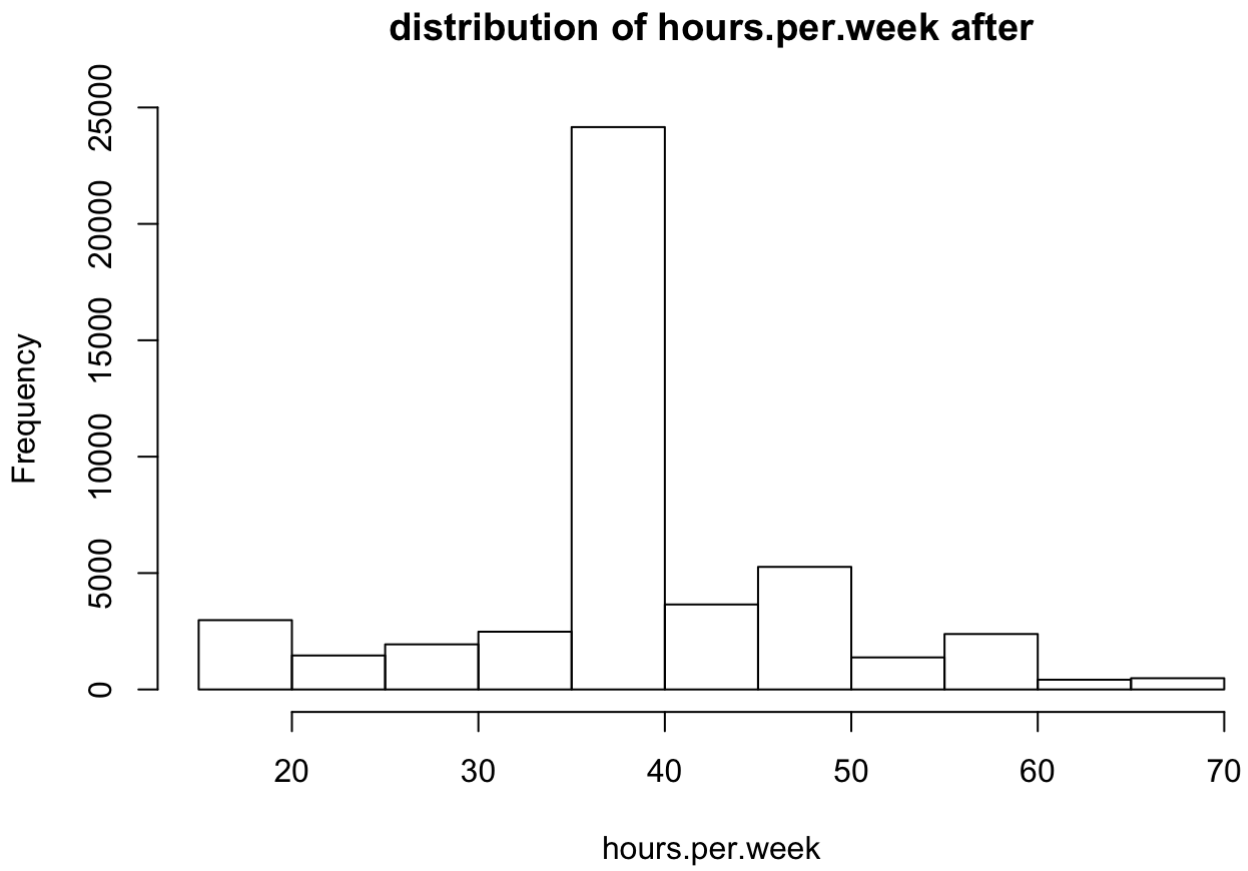

distribution of age after
Frequency
0
1000
3000
5000
20
30
40
50
60
70
80
age


distribution of educational.num befor


Frequency
0
5000
10000
15000
5
10
15
educational.num


distribution of educational.num after
Frequency
0
5000
10000
15000
6
8
10
12
14
16
educational.num


distribution of hours.per.week befor
Frequency
0
5000
10000
15000
20000
25000
0
20
40
60
80
100
hours.per.week




```
sum(is.na(data$hours.per.week))
```

```
## [1] 2249
```

```
sum(is.na(data$educational.num))
```

```
## [1] 1794
```

```
sum(is.na(data$age))
```

```
## [1] 216
```

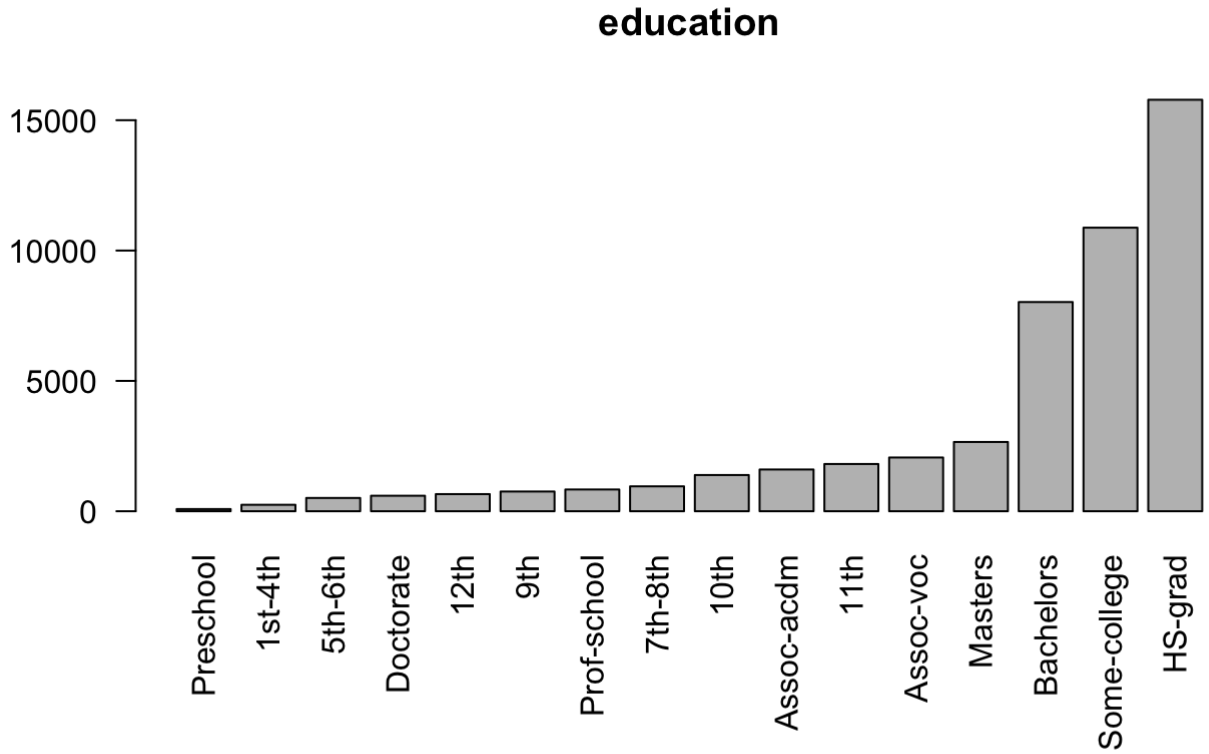
#4.c.

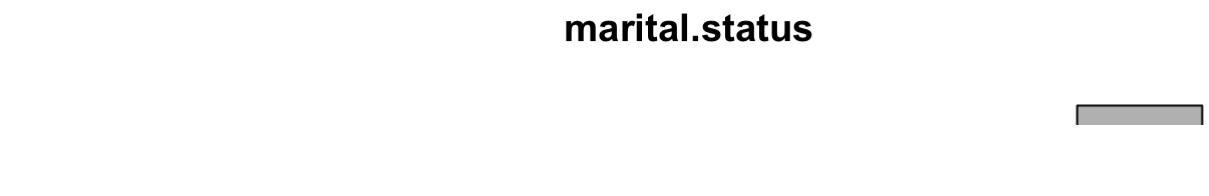
```
for (i in colnames(data)){
  if(class(data[[i]])=="integer"){
    data[[paste(i,"_standardize",sep="")]]<-((data[[i]]-mean(data[[i]], na.rm = T))/s
d(data[[i]], na.rm = T))
  }#close_if
}#close_for
```

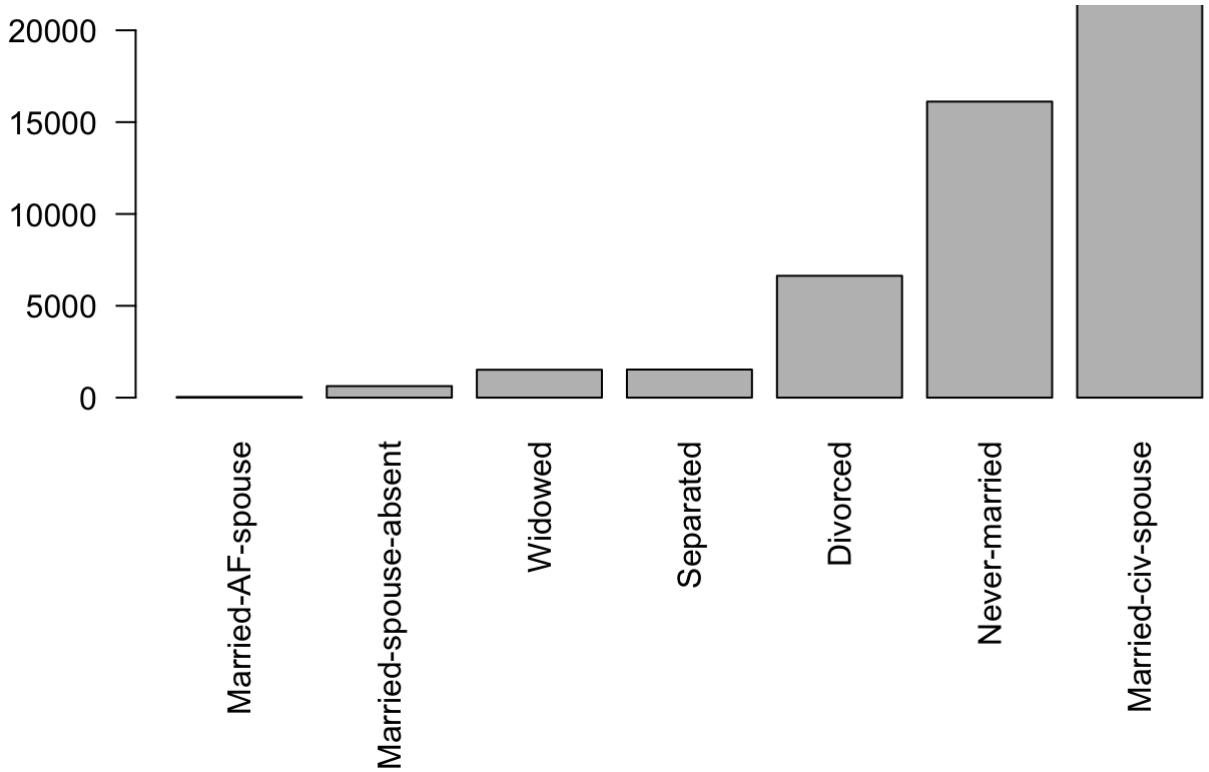
#4.d.

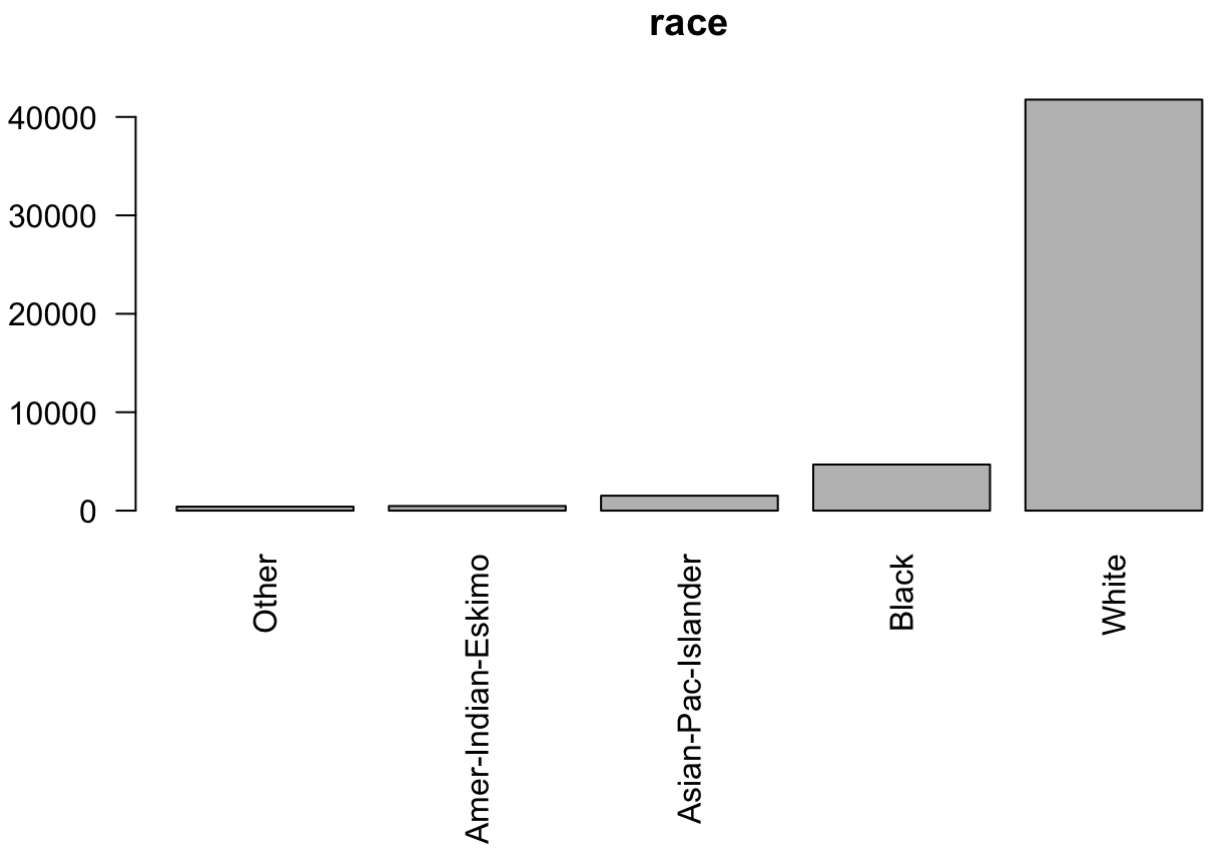
```
par(mar = c(10,4,4,2) + 0.1)
for (i in colnames(data)){
  if(class(data[[i]])=="factor"){
    barplot(table(data[[i]])[order(table(data[[i]]))],las=2,main = i)
      }#close_if
}#close_for
```


workclass
30000
25000
20000
15000
10000
5000
0
Never-worked
Without-pay
Federal-gov
Self-emp-inc
State-gov
?
Local-gov
Self-emp-not-inc
Private

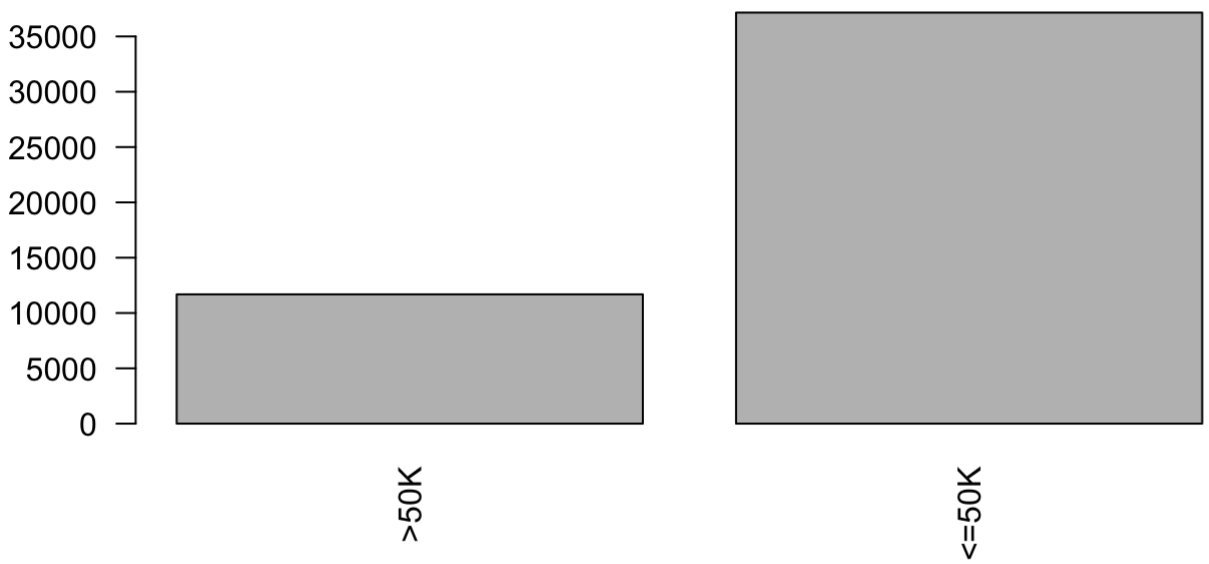

education
15000
10000
5000
0
Preschool
1st-4th
5th-6th
Doctorate
12th
9th
Prof-school
7th-8th
10th
Assoc-acdm
11th
Assoc-voc
Masters
Bachelors
Some-college
HS-grad


marital.status


20000
15000
10000
5000
0
Married-AF-spouse
Married-spouse-absent
Widowed
Separated
Divorced
Never-married
Married-civ-spouse


race
40000
30000
20000
10000
0
Other
Amer-Indian-Eskimo
Asian-Pac-Islander
Black
White


gender
30000
25000
20000
15000
10000
5000
0
Female
Male



#4.e. How can we merge different levels in the data? 1.race: “White”, all others as “Other”. 2.education: This is the hierarchy of education levels: Preschool < 1st-4th < 5th-6th < 7th-8th < 9th < 10th < 11th < 12th < HS-grad < Prof-school < Assoc-acdm < Assoc-voc < Some-college < Bachelors < Masters < Doctorate. As so, we will merge the following factors: Dropped_out(preschool to 12th grade), Advance_deg(Masters & Doctorate), Basic_acdm(Assoc-acdm, Assoc-voc, Some-college) HS-grad(HS-grad, Prof-school) 3.workclass: First, we see that we have an unkwown (“?”) workclass. We will deal with that later. Govermant(Federal-gov, Local-gov, State-gov) Self_imp(Self-emp-inc, Self-emp-not-inc) Non_pay(Never-worked, Without-pay) 4.married: Was-married(Separated, Divorced, Widowed, Married-spouse-absent) Married(Married-AF-spouse, Married-civ-spouse)

```
data$race_merged<-data$race
levels(data$race_merged)<-c("Other","Other","Other","Other","White")
data$education_merged<-data$education
levels(data$education_merged)<-c("Dropped_out","Dropped_out","Dropped_out","Dropped_o
ut","Dropped_out",
                                 "Dropped_out","Dropped_out","Basic_acdm","Basic_acd
m","Bachelors",
                                 "Advance_deg","HS-grad","Advance_deg","Dropped_out",
"HS-grad","Basic_acdm")
data$workclass_merged<-data$workclass
levels(data$workclass_merged)<-c("?","Government","Government","Non_pay","Private","S
elf_imp",
                                 "Self_imp","Government","Non_pay")
data$marital.status_merged<-data$marital.status
levels(data$marital.status_merged)<-c("Was-married","Married","Married","Was-married"
,"Never-married",
                                      "Was-married","Was-married")
```

#4.f.

```
set.seed(256)
in_train <- sample(1:nrow(data), 0.7*nrow(data)) #70%-30%
data_train <- data[in_train, ] # 70%
data_test <- data[-in_train, ] # 30%
```

#4.g.

```
glm.q4<-glm(income~age_standardize + educational.num_standardize + hours.per.week_sta
ndardize +
      race_merged + education_merged + workclass_merged + marital.status_merged + gen
der,
        family = binomial,data=data_train)
summary(glm.q4)
```

```
##
## Call:
## glm(formula = income ~ age_standardize + educational.num_standardize +
##     hours.per.week_standardize + race_merged + education_merged +
##     workclass_merged + marital.status_merged + gender, family = binomial,
##     data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5981  -0.5958  -0.2519   0.2439   3.3334
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -3.54061    0.16082 -22.016  < 2e-16 ***
## age_standardize                    0.46638    0.02060  22.640  < 2e-16 ***
## educational.num_standardize        0.94630    0.03838  24.659  < 2e-16 ***
## hours.per.week_standardize         0.40523    0.01838  22.050  < 2e-16 ***
## race_mergedWhite                   0.22823    0.05271   4.330 1.49e-05 ***
## education_mergedBasic_acdm        -0.08953    0.10889  -0.822   0.4110
## education_mergedBachelors         -0.18568    0.14034  -1.323   0.1858
## education_mergedAdvance_deg       -0.22292    0.16311  -1.367   0.1717
## education_mergedHS-grad           -0.05837    0.09857  -0.592   0.5538
## workclass_mergedGovernment         0.78225    0.11261   6.947 3.74e-12 ***
## workclass_mergedNon_pay           -0.55700    0.83309  -0.669   0.5038
## workclass_mergedPrivate            0.78984    0.10726   7.364 1.79e-13 ***
## workclass_mergedSelf_imp           0.49546    0.11401   4.346 1.39e-05 ***
## marital.status_mergedMarried       2.13563    0.05371  39.763  < 2e-16 ***
## marital.status_mergedNever-married -0.45915    0.06971  -6.587 4.49e-11 ***
## genderMale                         0.08523    0.04585   1.859   0.0631 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 35277  on 31357  degrees of freedom
## Residual deviance: 23393  on 31342  degrees of freedom
##   (2831 observations deleted due to missingness)
## AIC: 23425
##
## Number of Fisher Scoring iterations: 6
```

The ‘AIC’ is 23425. Smaller AIC values indicate that the model is closer to the truth.

#4.h.

```
data_test_hat<-predict(glm.q4, data_test, type = "response")
data_test_hat_binar<-(data_test_hat>0.5)*1
#confusion matrix:
CM.glm.q4 <- table(true= data_test$income, predicted = data_test_hat_binar)
paste("We predicted correctly ", CM.glm.q4[1,1]+CM.glm.q4[2,2],", and we missed ", C
M.glm.q4[1,2]+CM.glm.q4[2,1],".",sep="")
```

```
## [1] "We predicted correctly 10983, and we missed 2455."
```

#4.i. To measure accuracy we will use the formula: Accuracy = (TP+TN)/(TP+TN+FP+FN)

```
acc<-(sum(diag(CM.glm.q4)) / sum(CM.glm.q4))
paste("The accuracy of the model is: ",acc)
```

```
## [1] "The accuracy of the model is:  0.817309123381456"
```

#4.j. The Precision formula is:TP/(TP+FP) The Recall formula is:TP/(TP+FN)

```
Preci <- (CM.glm.q4[4] / sum(CM.glm.q4[,2]))
Rec <- (CM.glm.q4[4] / sum(CM.glm.q4[2,]))
paste("The Precision of the model is: ",Preci)
```

```
## [1] "The Precision of the model is:  0.670042851577717"
```

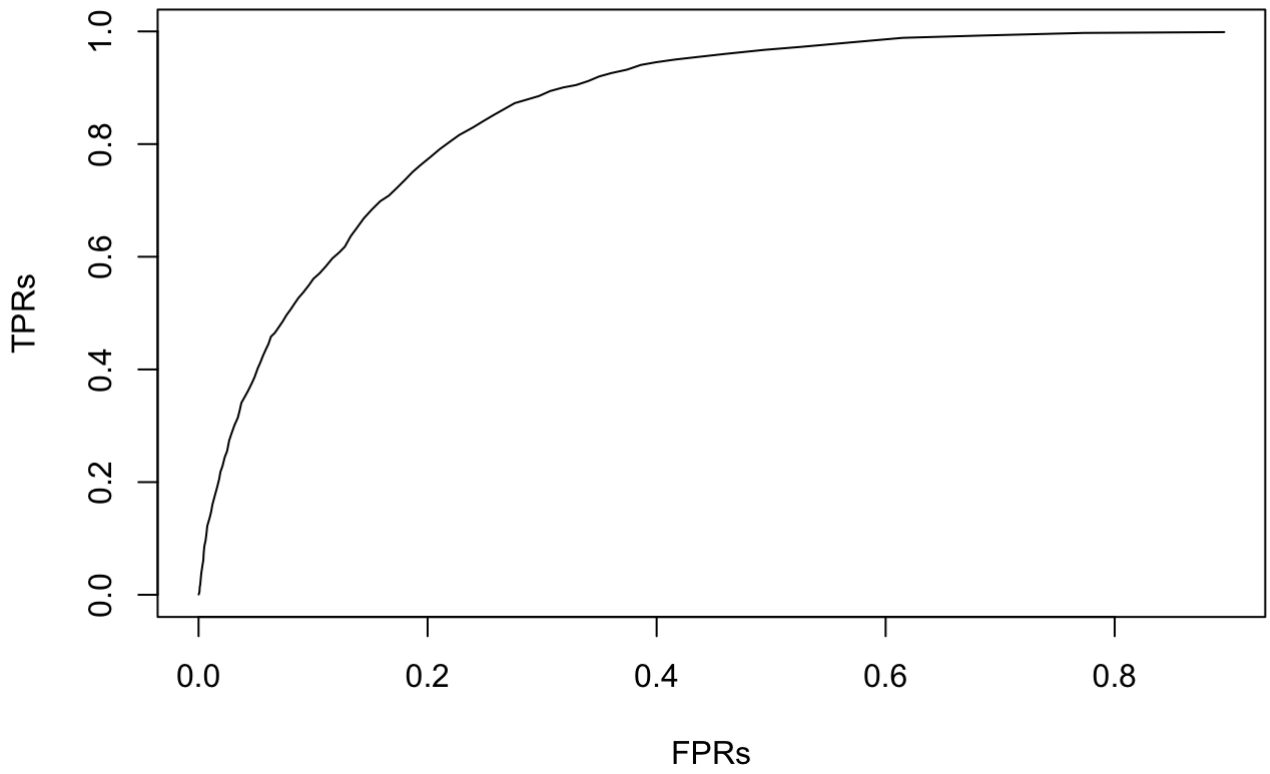
```
paste("The Recall of the model is: ",Rec)
```

```
## [1] "The Recall of the model is:  0.516826923076923"
```

There is a trade-off between Precision and Recall. We can not have them both high at the same time, it depends on what do we find more important to avoid - a false positive or a false negative. Precision is more important than Recall when you would like to have fewer false positives and the other way around with Recall.

#4.k.

```
alphas <- seq(0,1,0.01)
TPRs <- numeric(length(alphas))
FPRs <- numeric(length(alphas))
for (i in seq_along(alphas)){
  pr_i <- ifelse(data_test_hat>alphas[i],1,0)
  CM_i <- table(data_test$income,pr_i) ###only for test data
  TPRs[i] <- CM_i[4] / sum(CM_i[2,]) # TP/TP+FN - regection from FN (high is good)
  FPRs[i] <- CM_i[3] / sum(CM_i[1,]) # FP/FP+FN - regection from FP (low is good)
}
plot(TPRs~FPRs, type = "l",main = "ROC Curve")
```



The ROC curve is a plot of the true positive rate (Recall) against the false positive rate for different threshold levels. By that we can select possibly optimal models.

#4.I. We think we should remove the “education_merged”(factorial education), because it is correlated with the “educational.num” and so less informative. In addition we noticed that a model without it decreases the AIC and the accuracy by a bit.

We will also add the interactions of the age with race (all three combinations) and gender because we believe that males and females or whites and non-whites might start working at different ages. Also we will interact age with the number of work hours per week because we believe that there is additional information gain of the number of hours a person works due to its age.

We will add the interaction of hours per week with the marital status as we believe people with or without a spouse work a different amount of time and also add this relation with the race to help the model see the effect within each race. We will add gender and marital status as interactions as we believe that men and women work different amounts of hours per week.

We will additionally set a relation between the years of education, race and marital status to add an effect of different levels of education within different races and different marital statuses and the same with different genders instead of marital status.

Finally we will also present the effect of non-linear relation on our numerical features. This transformation might be somewhat “incorrect” as we lose the negative values in the data when setting the values in the power of 2, but as we see, that interaction contributes to the accuracy of the model so we will use it -as this was the goal of this question.

```
glm.q4_try <- glm(income ~ age_standardize + educational.num_standardize + hours.per.
week_standardize + race_merged + workclass_merged + marital.status_merged + gender +
I(age_standardize^2) + I(educational.num_standardize^2) + I(hours.per.week_standardiz
e^2) +
age_standardize*race_merged + age_standardize*gender + age_standardize*hours.per.week
_standardize + age_standardize*race_merged*gender + hours.per.week_standardize*marita
l.status_merged + hours.per.week_standardize*gender +  hours.per.week_standardize*rac
e_merged*marital.status_merged + educational.num_standardize*race_merged*marital.stat
us_merged + educational.num_standardize*race_merged*gender,family = binomial,data=dat
a_train)

data_test_hat_try<-predict(glm.q4_try, data_test, type = "response")
data_test_hat_binar_try<-(data_test_hat_try>0.5)*1
CM.glm.q4_try <- table(true= data_test$income, predicted = data_test_hat_binar_try)
acc_try<-(sum(diag(CM.glm.q4_try)) / sum(CM.glm.q4_try))
paste("The accuracy of the model is: ",acc_try)
```

```
## [1] "The accuracy of the model is:  0.824899538621819"
```

```
paste("We gained an additional model accuracy of " ,acc_try-0.818718696269717,".",sep
="")
```

```
## [1] "We gained an additional model accuracy of 0.0061808423521017."
```

Though this might be a very low addition to the accuracy this is the best we could do. Generally we believe that this addition is not too good for the general idea of modeling as we added a very large amount of new features that were not extremely informative. The testing of the accuracy of the model was on a single test set, and by that we think we should say that this model is an example of “Over-fitting”. If we would like to predict a different kind of a test set, we think we should use our initial model as it much simpler and predict similar results.