Assignment 2

Group 61, Ikrame Zirar, Mohammed Majeed, Sergio Alejandro Gutierrez Maury

2023-03-09

Excersice 1

A) The dataset "treeVolume" contains a response variable, namely "Volume", and several explanatory variables, including "type", "height", and "diameter". To investigate the impact of tree type on volume, we conducted ANOVA using "Volume" as the response variable and "type" as the sole explanatory variable. The p-value from the ANOVA table indicates that there is no significant effect of tree type on tree volume.

```
# Load the dataset
tree_data <- read.csv("treeVolume.txt", header = TRUE, sep = "")
# Perform t-test
model_aov <- aov(volume ~ type, data = tree_data)
summary(model_aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## type         1    380     380     1.9   0.17
## Residuals   57  11395     200
```

We conducted a t-test to compare the means of these two sample groups.the p-value of the t-test indicates that the type of tree does not have a significant impact on its volume.

```
# Perform t-test
t_test <- t.test(volume ~ type, data = tree_data)
t_test
```

```
##
##  Welch Two Sample t-test
##
## data:  volume by type
## t = -1, df = 53, p-value = 0.2
## alternative hypothesis: true difference in means between group beech and group oak is not equal to 0
## 95 percent confidence interval:
##  -12.33   2.17
## sample estimates:
## mean in group beech   mean in group oak
##                30.2                35.2
```

The output of aggregate gives us the estimated volumes for the two tree types

```r
# Estimate the volumes for the two tree types
aggregate(tree_data$volume, by = list(tree_data$type), mean)
```

```
##   Group.1    x
## 1   beech 30.2
## 2     oak 35.2
```

b) To include diameter and height as explanatory variables into the analysis and investigate whether the influence of diameter and height on volume is similar for both tree types.

```r
# Fit a linear model with diameter and height as explanatory variables
model_lm <- lm(volume ~ diameter + height + type, data = tree_data)
summary(model_lm)
```

```
##
## Call:
## lm(formula = volume ~ diameter + height + type, data = tree_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.186 -2.140 -0.087  1.721  7.701
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -63.7814     5.5129  -11.57  2.3e-16 ***
## diameter      4.6981     0.1645   28.56  < 2e-16 ***
## height        0.4172     0.0752    5.55  8.4e-07 ***
## typeoak      -1.3046     0.8779   -1.49     0.14
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.24 on 55 degrees of freedom
## Multiple R-squared:  0.951,  Adjusted R-squared:  0.948
## F-statistic:  355 on 3 and 55 DF,  p-value: <2e-16
```

The below two ANOVA tables to investigate whether the influence of diameter and height on volume is similar for both tree types. In both cases, the interaction term is not significant, indicating that the influence of diameter and height on volume is similar for both tree types.

```r
# Perform ANOVA to investigate the influence of diameter on volume for both tree types
model_aov_diameter <- aov(volume ~ diameter * type, data = tree_data)
summary(model_aov_diameter)
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## diameter       1  10827   10827  666.80 <2e-16 ***
## type           1     45      45    2.79   0.10
## diameter:type  1     10      10    0.59   0.45
## Residuals     55    893      16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Perform ANOVA to investigate the influence of height on volume for both tree types
model_aov_height <- aov(volume ~ height * type, data = tree_data)
summary(model_aov_height)
```

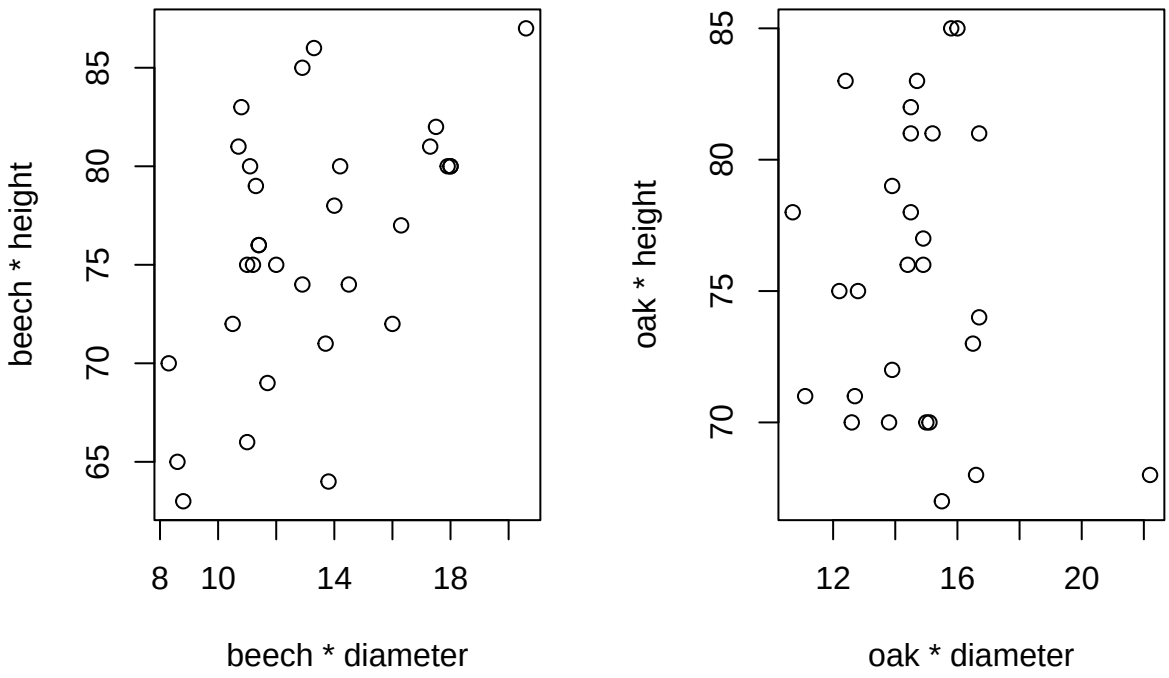
```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## height       1   2188    2188   14.30 0.00039 ***
## type         1    431     431    2.82 0.09883 .
## height:type  1    742     742    4.85 0.03183 *
## Residuals   55   8413     153
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

c)

The correlation coefficient is 0.519, indicating a moderate positive linear relationship between diameter and height of beech trees. The p-value is less than 0.05, indicating that there is strong evidence of a significant correlation between diameter and height of beech trees.

Whereas the correlation coefficient of oak trees is -0.116, indicating a weak negative linear relationship between diameter and height of oak trees. The p-value is greater than 0.05, indicating that there is not enough evidence to reject the null hypothesis of no correlation between diameter and height of oak trees.

```r
par(mfrow=c(1, 2))
plot(tree_data[tree_data$type == "beech", "diameter"],
     tree_data[tree_data$type == "beech", "height"],
     xlab =  "beech * diameter", ylab = "beech * height")
plot(tree_data[tree_data$type == "oak", "diameter"],
     tree_data[tree_data$type == "oak", "height"],
     xlab =  "oak * diameter", ylab = "oak * height")
```




```
cor.test(tree_data[tree_data$type == "beech", "diameter"],
         tree_data[tree_data$type == "beech", "height"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  tree_data[tree_data$type == "beech", "diameter"] and tree_data[tree_data$type == "beech", "hei
## t = 3, df = 29, p-value = 0.003
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.202 0.738
## sample estimates:
##   cor
## 0.519
```

```
cor.test(tree_data[tree_data$type == "oak", "diameter"],
         tree_data[tree_data$type == "oak", "height"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  tree_data[tree_data$type == "oak", "diameter"] and tree_data[tree_data$type == "oak", "height"
## t = -0.6, df = 26, p-value = 0.6
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.469  0.269
## sample estimates:
##    cor
## -0.116
```

Using the results from b), we can investigate how diameter, height, and type influence volume. To predict the volume for a tree with the overall average diameter and height.

```
# Calculate the overall average diameter and height
avg_diameter <- mean(tree_data$diameter)
avg_height <- mean(tree_data$height)

# Predict the volume for a tree with the overall average diameter and height
predict(model_lm, newdata = data.frame(diameter = avg_diameter, height = avg_height, type = "beech"), in
```

```
##    fit lwr  upr
## 1 33.2  32 34.4
```

d) It seems like there may be a natural relationship between the volume of a tree and its height and diameter. One possible transformation to consider is taking the logarithm of both height and diameter to create new variables, which may better capture the relationship with volume.

Both models have high R-squared values, indicating that they explain a large proportion of the variation in the response variable. However, the first model has a slightly higher R-squared value of 0.977 compared to the second model's (with no transformation) R-squared value of 0.951. This suggests that the first model may be a slightly better fit for the data.

```
# fit a linear model with the transformed variables
transformed_model <- lm(log(volume) ~ log(tree_data$height) + log(tree_data$diameter) + type, data=tree_
# print the summary of the model to check the results
summary(transformed_model);
```

```
##
## Call:
## lm(formula = log(volume) ~ log(tree_data$height) + log(tree_data$diameter) +
##     type, data = tree_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16830 -0.04261 -0.00212  0.04817  0.12936
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -6.7750     0.5061  -13.39  < 2e-16 ***
## log(tree_data$height)      1.1445     0.1232    9.29  7.3e-13 ***
## log(tree_data$diameter)    1.9924     0.0501   39.79  < 2e-16 ***
## typeoak                    0.0178     0.0192    0.92     0.36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0702 on 55 degrees of freedom
## Multiple R-squared:  0.977,  Adjusted R-squared:  0.976
## F-statistic:  773 on 3 and 55 DF,  p-value: <2e-16
```

```r
# fit a linear model with the transformed variables
model <- lm(volume ~ tree_data$height + tree_data$diameter + type, data=tree_data)
# print the summary of the model to check the results
summary(model);
```

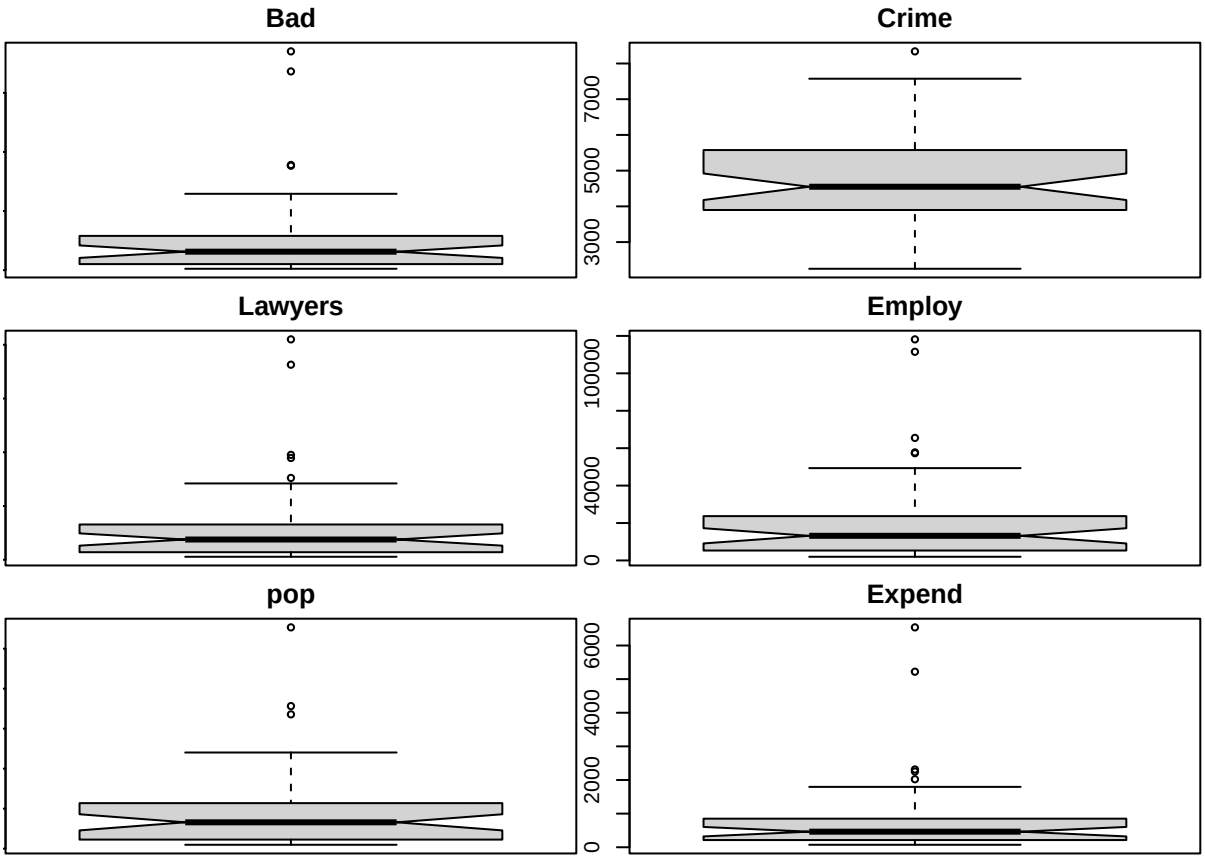
```
##
## Call:
## lm(formula = volume ~ tree_data$height + tree_data$diameter +
##     type, data = tree_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.186 -2.140 -0.087  1.721  7.701
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -63.7814     5.5129  -11.57  2.3e-16 ***
## tree_data$height      0.4172     0.0752    5.55  8.4e-07 ***
## tree_data$diameter    4.6981     0.1645   28.56  < 2e-16 ***
## typeoak              -1.3046     0.8779   -1.49     0.14
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.24 on 55 degrees of freedom
## Multiple R-squared:  0.951,  Adjusted R-squared:  0.948
## F-statistic:  355 on 3 and 55 DF,  p-value: <2e-16
```

Excersice 2

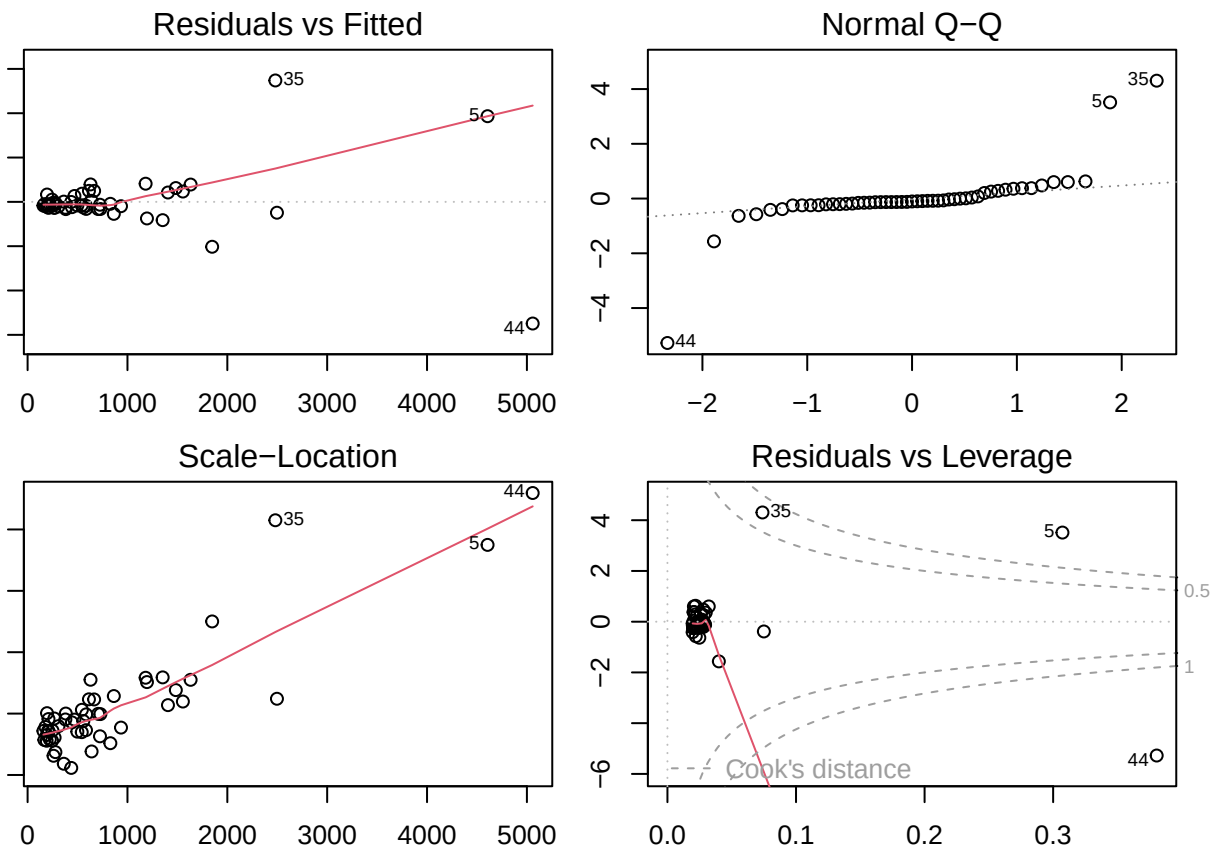
A)

It can be seen from the box plots, that there are some outliers that might be influential, and from the leverage vs residuals plots we can see that, this is indeed the case.

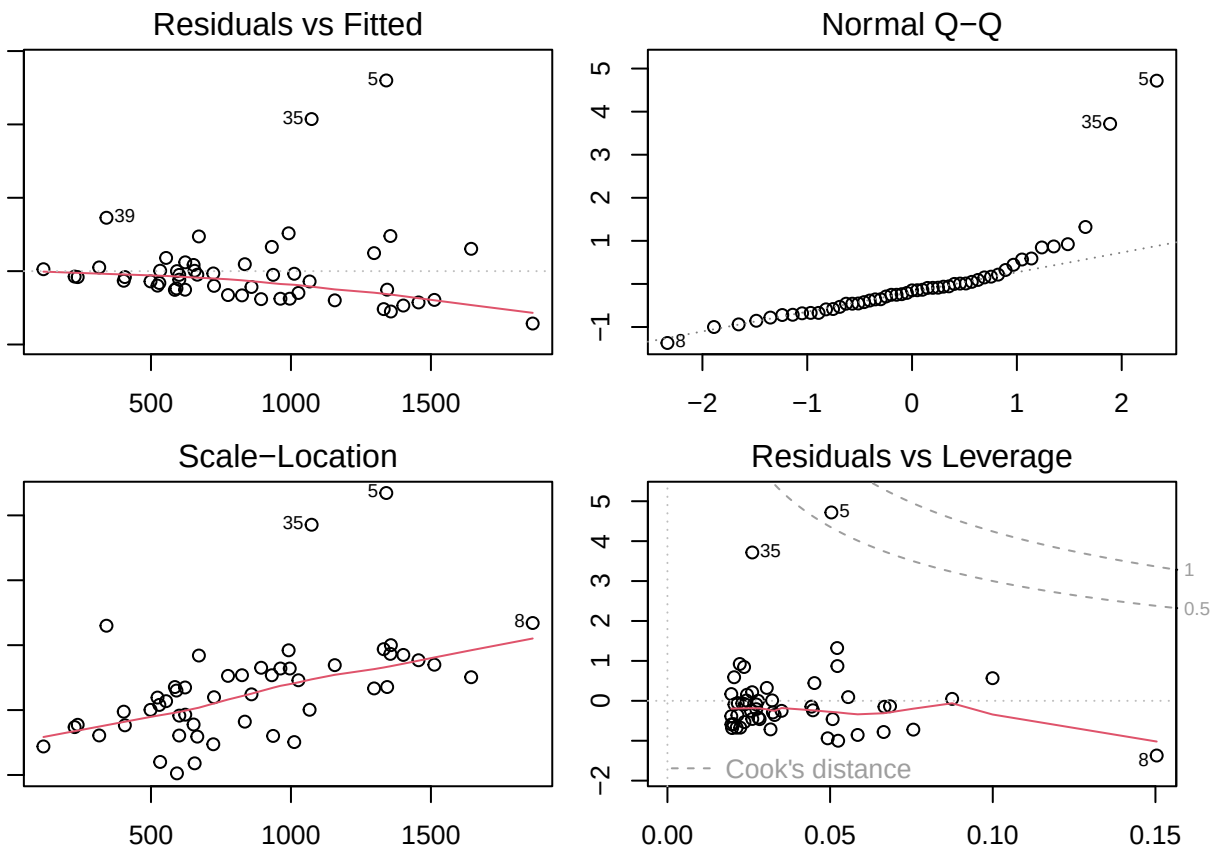
```r
data <- read.csv("expensescrime.txt", header = TRUE, sep = " ")
# scatterplots
par(mfrow=c(3,2), mar=c(0.1,0.1,2,2))

boxplot(data$bad, data = data, notch = TRUE, varwidth = TRUE, main = "Bad")
boxplot(data$crime, data = data, notch = TRUE, varwidth = TRUE, main = "Crime")
boxplot(data$lawyers, data = data, notch = TRUE, varwidth = TRUE, main = "Lawyers")
boxplot(data$employ, data = data, notch = TRUE, varwidth = TRUE, main = "Employ")
boxplot(data$pop, data = data, notch = TRUE, varwidth = TRUE, main = "pop")
boxplot(data$expend, data = data, notch = TRUE, varwidth = TRUE, main = "Expend")
```



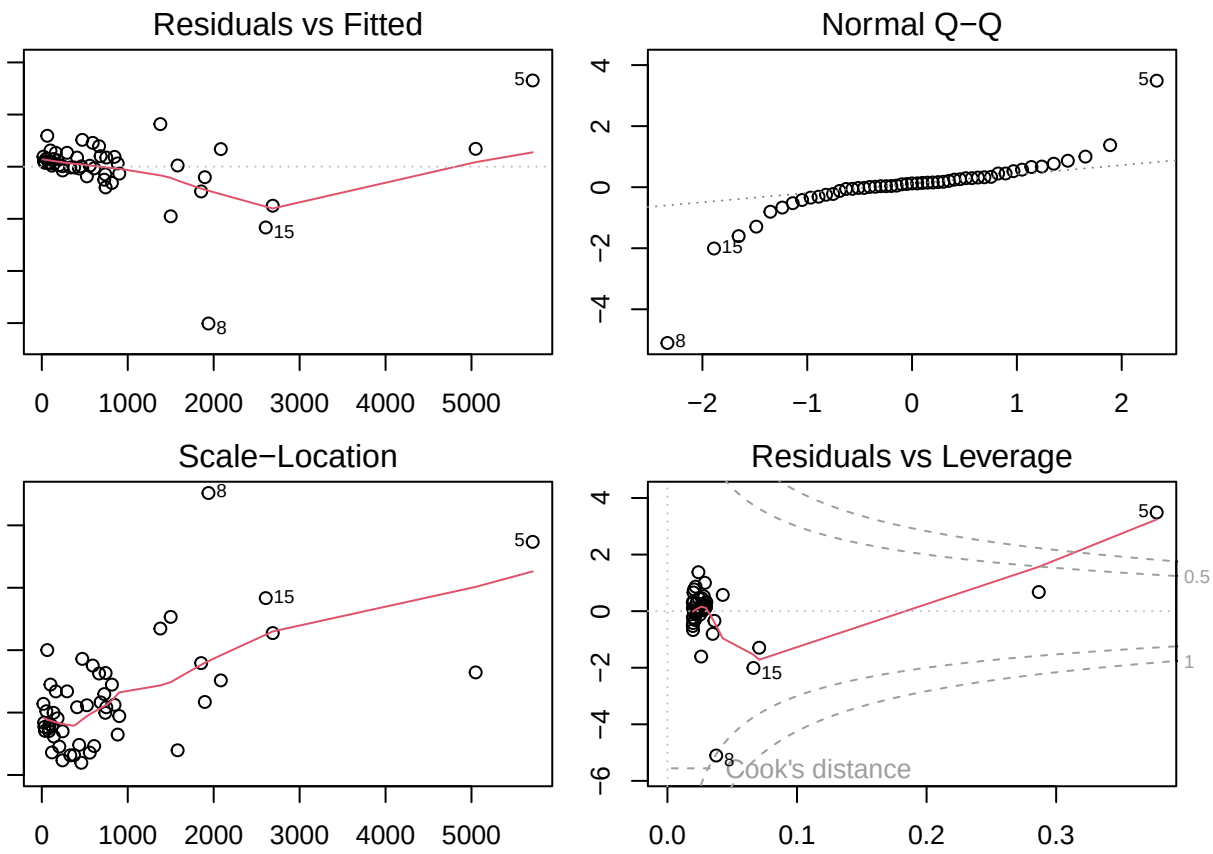

```
par(mfrow=c(2,2), mar=c(2,1,2,2))
library(MASS)
rlm_fit1 <- lm(data$expend~data$bad, data=data)
rlm_fit2 <- lm(data$expend~data$crime, data=data)
rlm_fit3 <- lm(data$expend~data$lawyers, data=data)
rlm_fit4 <- lm(data$expend~data$employ, data=data)
rlm_fit5 <- lm(data$expend~data$pop, data=data)
plot(rlm_fit1)
```



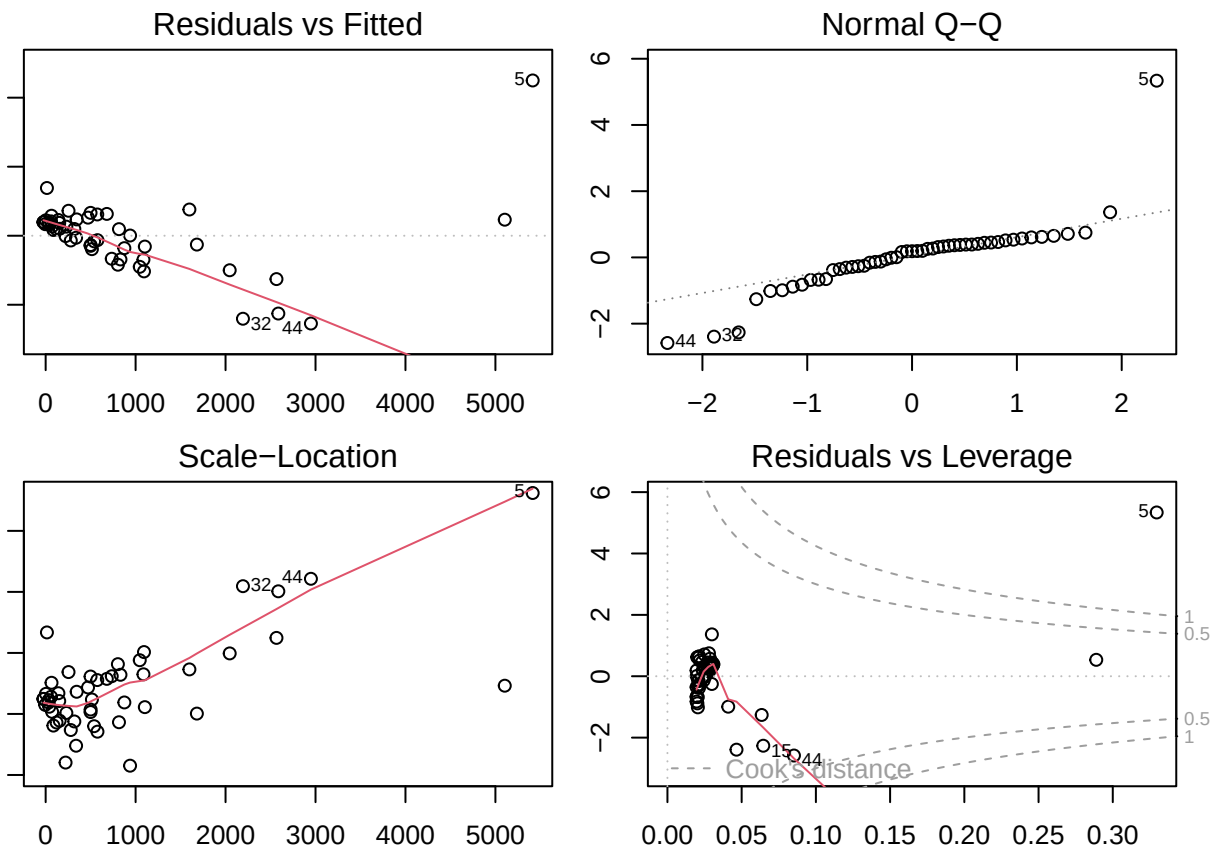

```
plot(rlm_fit2)
```



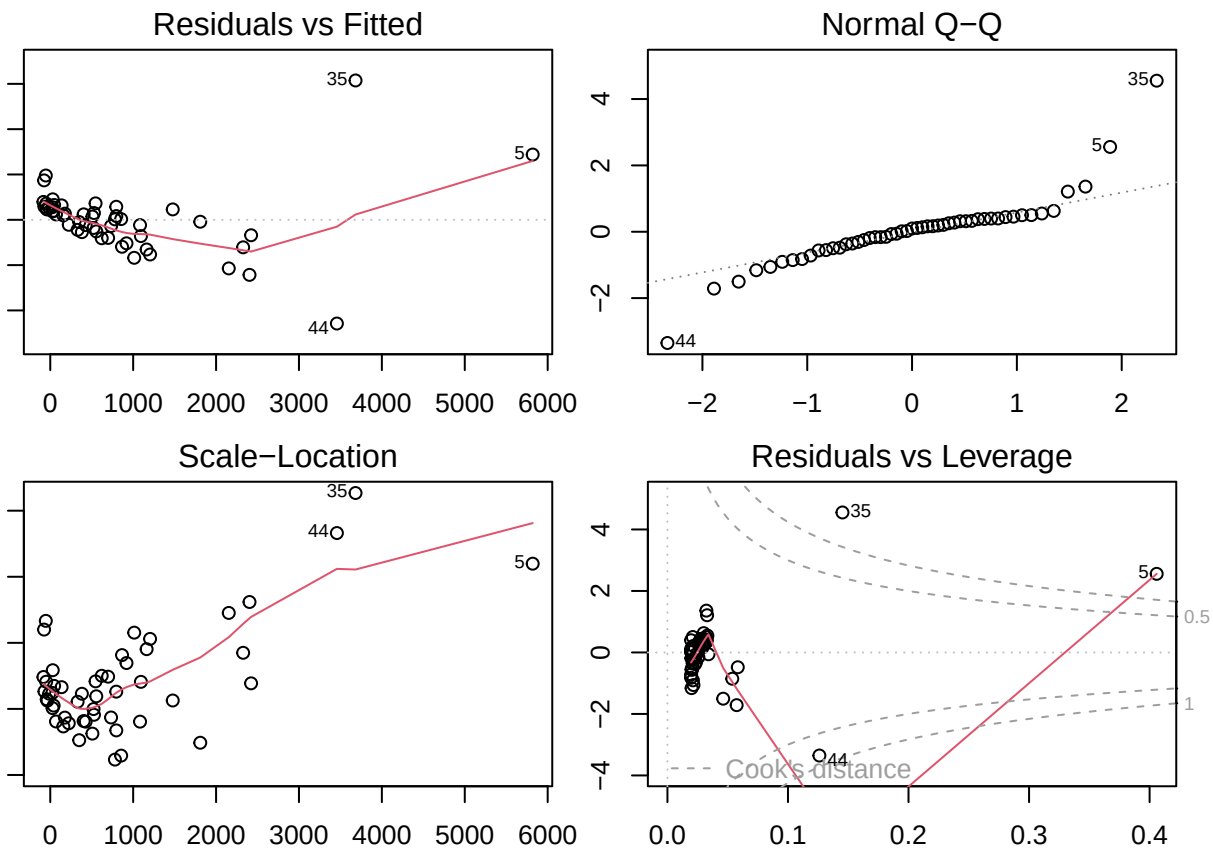

```
plot(rlm_fit3)
```




```
plot(rlm_fit4)
```

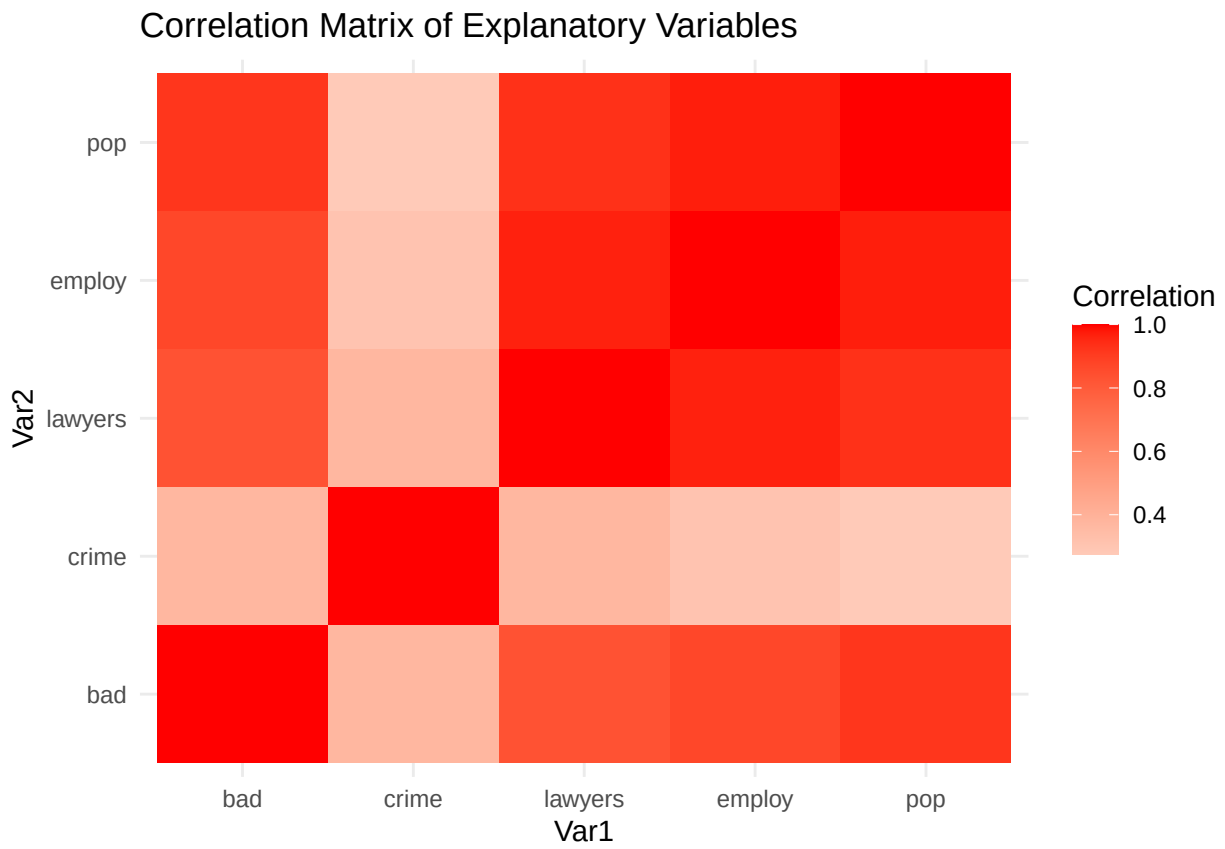



```
plot(rlm_fit5)
```



By looking at the correlation matrix, it can be seen that there are some multicollinearity problems, since the variable "bad" is highly correlated with other independent variables.

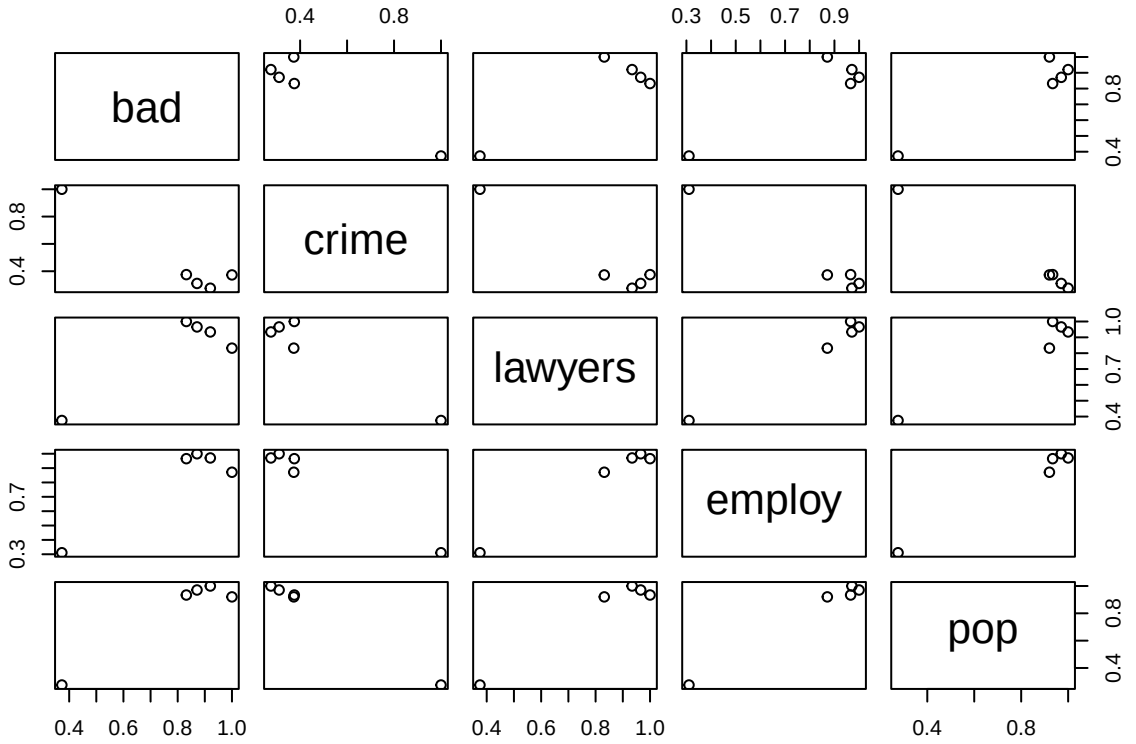
```
library("reshape2")
library('ggplot2')
# calculate correlation matrix
cor_matrix <- cor(data[, c("bad", "crime", "lawyers", "employ", "pop")])

# plot correlation matrix
ggplot(data = reshape2::melt(cor_matrix)) +
  geom_tile(aes(x = Var1, y = Var2, fill = value)) +
  scale_fill_gradient2(low = "blue", high = "red", midpoint = 0,
                       name = "Correlation") +
  theme_minimal() +
  labs(title = "Correlation Matrix of Explanatory Variables")
```




```
round(cor_matrix,2)
```

```
##           bad crime lawyers employ  pop
## bad     1.00  0.37    0.83   0.87 0.92
## crime   0.37  1.00    0.38   0.31 0.28
## lawyers 0.83  0.38    1.00   0.97 0.93
## employ  0.87  0.31    0.97   1.00 0.97
## pop     0.92  0.28    0.93   0.97 1.00
```

```
pairs(cor_matrix)
```



B)

The setp-up method selects as best model: $\hat{e} = \beta_0 + \beta_1 \cdot bad + \beta_2 \cdot lawyers + \beta_3 \cdot employ + \beta_4 \cdot pop$ where all coefficients are significant with at least 5% level.

```
library(MASS)

# fit full model
full_model <- lm(expend ~ bad + crime + lawyers + employ + pop, data=data)

# step-up method to find best model
full_model.step <- stepAIC(full_model, direction="both")
```

```
## Start:  AIC=558
## expend ~ bad + crime + lawyers + employ + pop
##
##           Df Sum of Sq     RSS AIC
## - crime    1     67546 2357262 558
## <none>                 2289716 558
## - pop      1    249704 2539420 562
## - bad      1    265249 2554964 562
## - lawyers  1    424835 2714551 565
## - employ   1    482202 2771918 566
##
## Step:  AIC=558
## expend ~ bad + lawyers + employ + pop
```

```
##
##           Df Sum of Sq     RSS AIC
## <none>                  2357262 558
## + crime    1     67546 2289716 558
## - pop      1    190369 2547631 560
## - bad      1    200346 2557608 560
## - employ   1    476538 2833800 565
## - lawyers  1    625997 2983259 568
```

```
summary(full_model.step)
```

```
##
## Call:
## lm(formula = expend ~ bad + lawyers + employ + pop, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -635.6  -80.2   18.8  114.5  809.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.46e+02   4.54e+01   -3.22   0.0023 **
## bad         -2.24e+00   1.13e+00   -1.98   0.0540 .
## lawyers      2.65e-02   7.57e-03    3.50   0.0011 **
## employ       2.28e-02   7.49e-03    3.05   0.0038 **
## pop          6.37e-02   3.30e-02    1.93   0.0601 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 226 on 46 degrees of freedom
## Multiple R-squared:  0.967,  Adjusted R-squared:  0.964
## F-statistic:  333 on 4 and 46 DF,  p-value: <2e-16
```

C) pending question: can you improve the interval?

The interval is: $ (-192.8264, 805.6644)$

```
# create new data frame with hypothetical values
new_data <- data.frame(bad=50, crime=5000, lawyers=5000, employ=5000, pop=5000)

# predict expend using selected model
pred <- predict(full_model.step, newdata=new_data, interval="prediction", level=0.95)


pred
```

```
##   fit  lwr upr
## 1 306 -193 806
```

D)

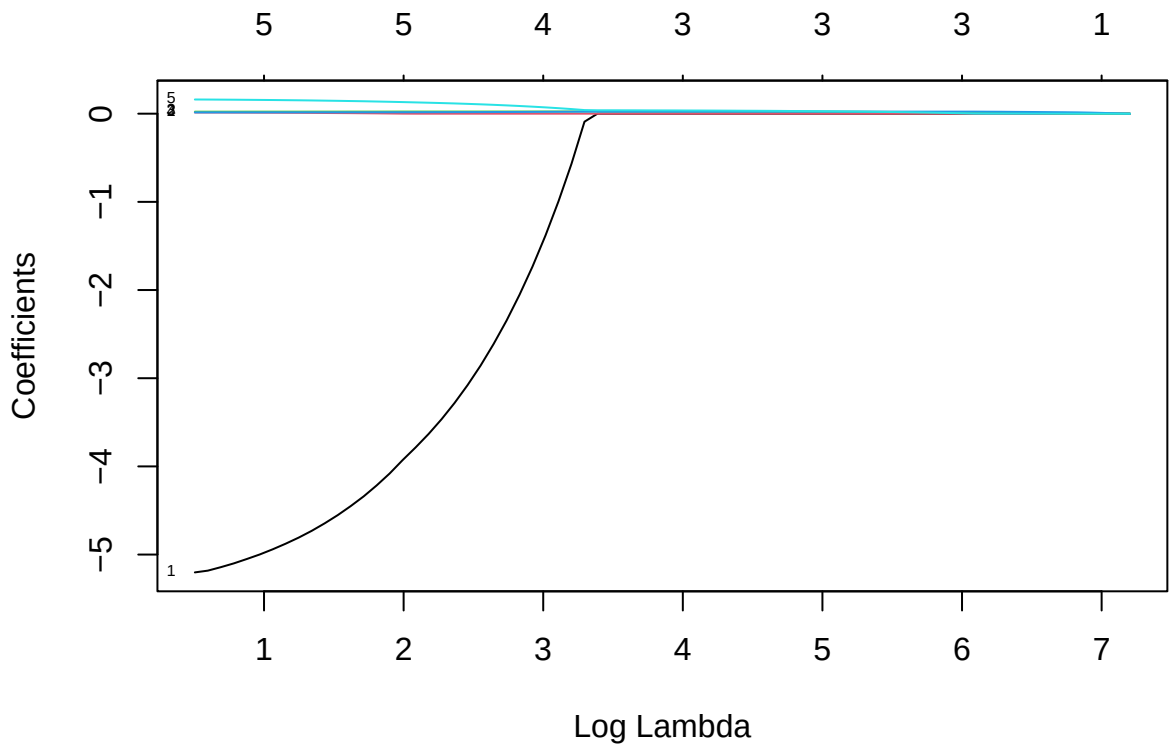
Comparing the lasso model with the step-up model, the lasso model set the variables "bad" and "crime" to zero, which means that those variables are not important. As a result, we end up with a much simpler model.

```r
set.seed(73) #sheldon prime !
library(glmnet)
```

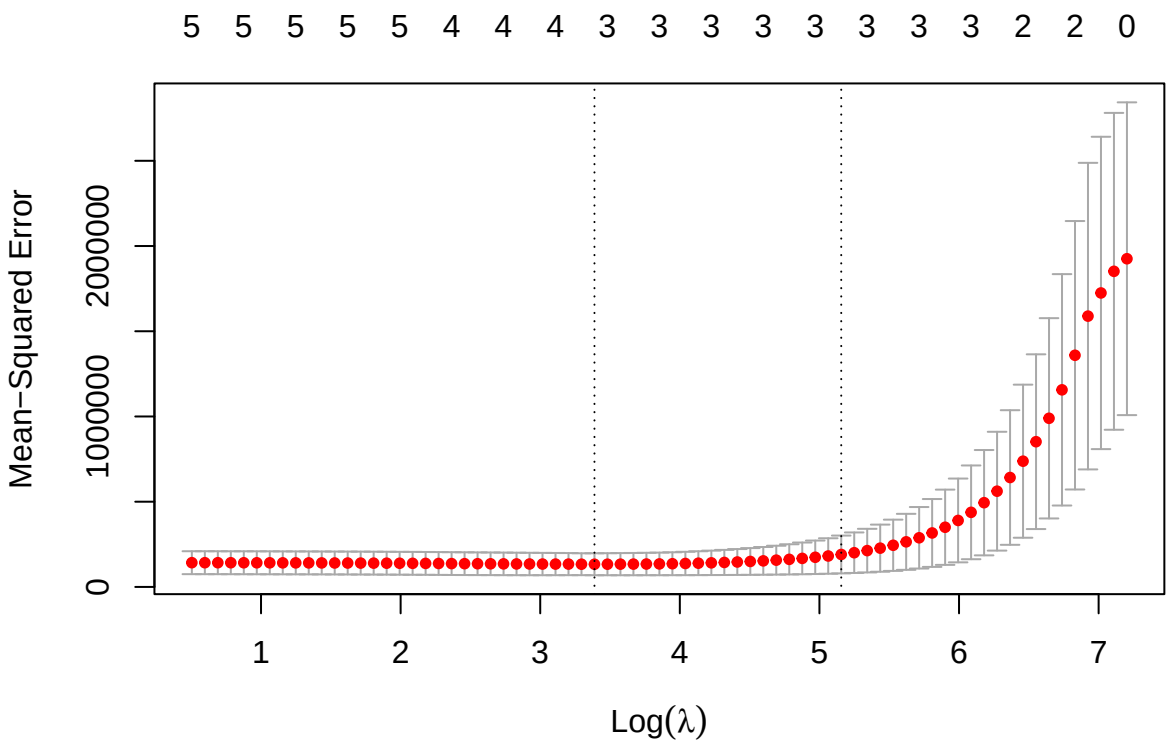
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-6
```

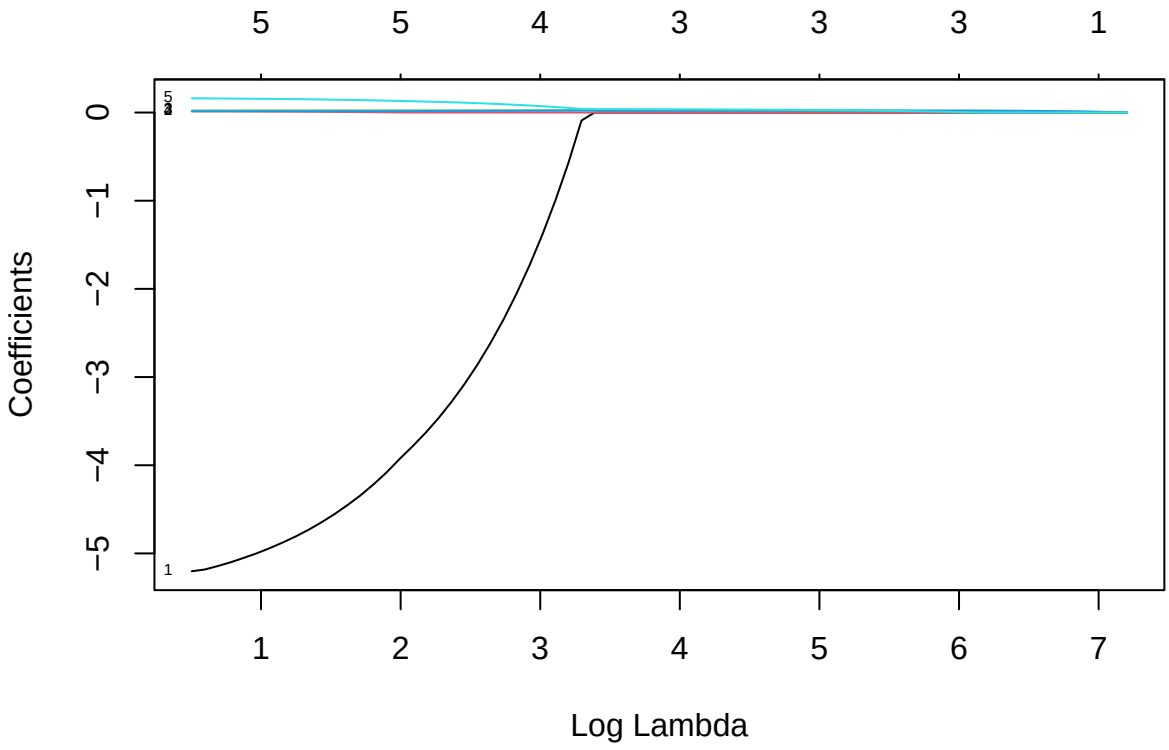
```r
x <- as.matrix(data[, c("bad", "crime", "lawyers", "employ", "pop")])
y <- data$expend
train=sample(1:nrow(x),0.67*nrow(x)) # train by using 2/3 of the data
x.train=x[train,]; y.train=y[train] # data to train
x.test=x[-train,]; y.test=y[-train] # data to test the prediction quality


lasso.mod=glmnet(x.train,y.train,alpha=1)
cv.lasso=cv.glmnet(x.train,y.train,alpha=1,type.measure='mse')
plot(lasso.mod,label=T,xvar="lambda") #have a look at the lasso path
```




```r
plot(cv.lasso) # the best lambda by cross-validation
```




```
plot(cv.lasso$glmnet.fit,xvar="lambda",label=T)
```




```
lambda.min=cv.lasso$lambda.min; lambda.1se=cv.lasso$lambda.1se
coef(lasso.mod,s=cv.lasso$lambda.min) #beta's for the best lambda
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                     s1
## (Intercept) -109.9850
## bad            .
## crime          .
## lawyers        0.0259
## employ         0.0230
## pop            0.0385
```

```
y.pred=predict(lasso.mod,s=lambda.min,newx=x.test) #predict for test
mse.lasso=mean((y.test-y.pred)^2) #mse for the predicted test rows
```

Excersice 3

A)

```
# install.packages("rms",dependencies = TRUE)
#install.packages("Hmisc")
#
library(ggplot2);
library(Hmisc);
```

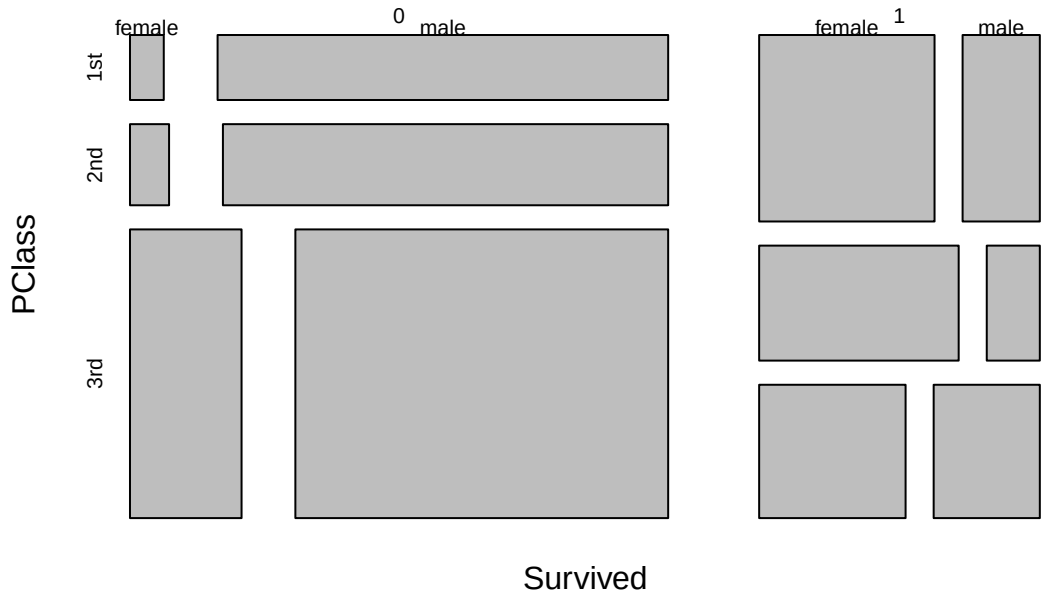
```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(rms);
```

```
## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##     backsolve
```

```
library(rmsb);
```

```
titanic_df <- read.table("titanic.txt", header=TRUE)
plot(xtabs(~Survived + PClass + Sex, titanic_df))
```

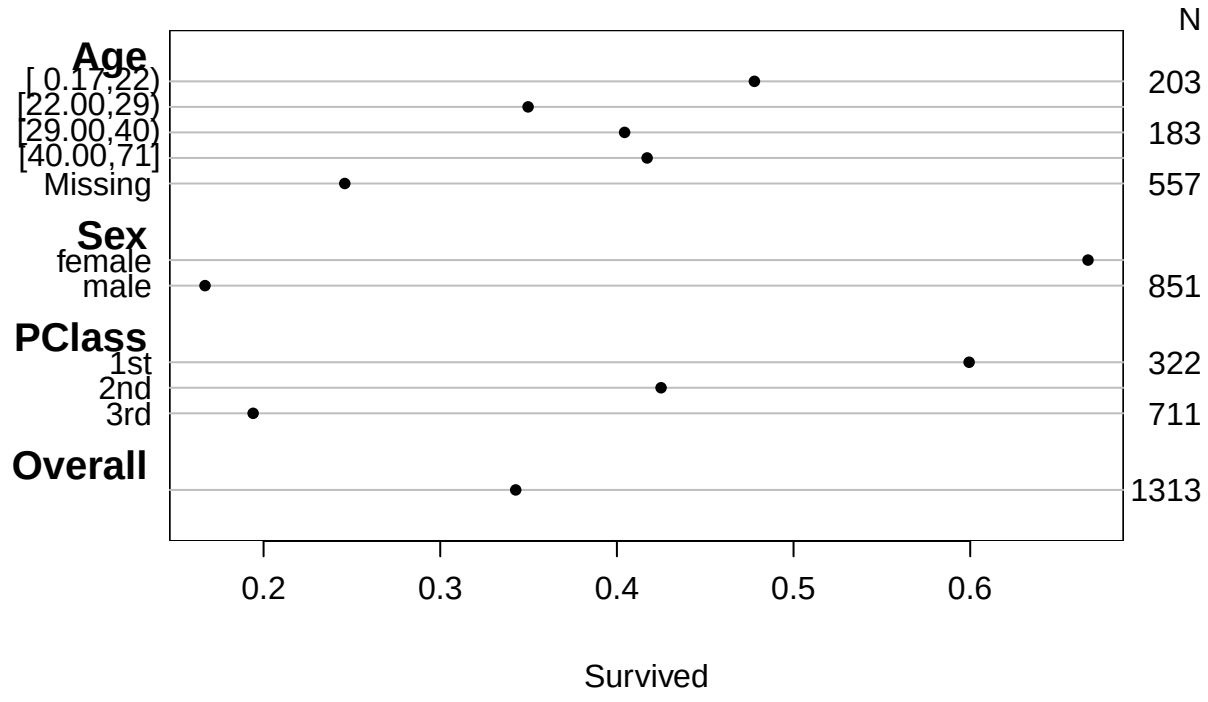



```
# options(prType='html')
v <- c('PClass','Survived','Age','Sex')
titanic <- titanic_df[, v]
describe(titanic)
```

```
## titanic
##
##  4  Variables      1313  Observations
## ---------------------------------------------------------------------------
## PClass
##        n  missing distinct
##     1313        0        3
##
## Value        1st   2nd   3rd
## Frequency    322   280   711
## Proportion 0.245 0.213 0.542
## ---------------------------------------------------------------------------
## Survived
##        n  missing distinct     Info      Sum     Mean      Gmd
##     1313        0        2    0.676      450   0.3427   0.4509
##
## ---------------------------------------------------------------------------
## Age
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      756      557       75    0.999     30.4    15.95        6       16
##      .25      .50      .75      .90      .95
```

```
##       21       28       39       50       57
##
## lowest :  0.17  0.33  0.80  0.83  0.92, highest: 65.00 67.00 69.00 70.00 71.00
## ---------------------------------------------------------------------------
## Sex
##        n  missing distinct
##     1313        0        2
##
## Value      female   male
## Frequency     462    851
## Proportion  0.352  0.648
## ---------------------------------------------------------------------------
```

```
# # spar(ps=4,rt=3)spar
dd <- datadist(titanic_df)
# describe distributions of variables to rms
options(datadist='dd')
s <- summary(Survived ~ Age + Sex + PClass , data=titanic_df)
plot(s, main='', subtitles=FALSE)
```



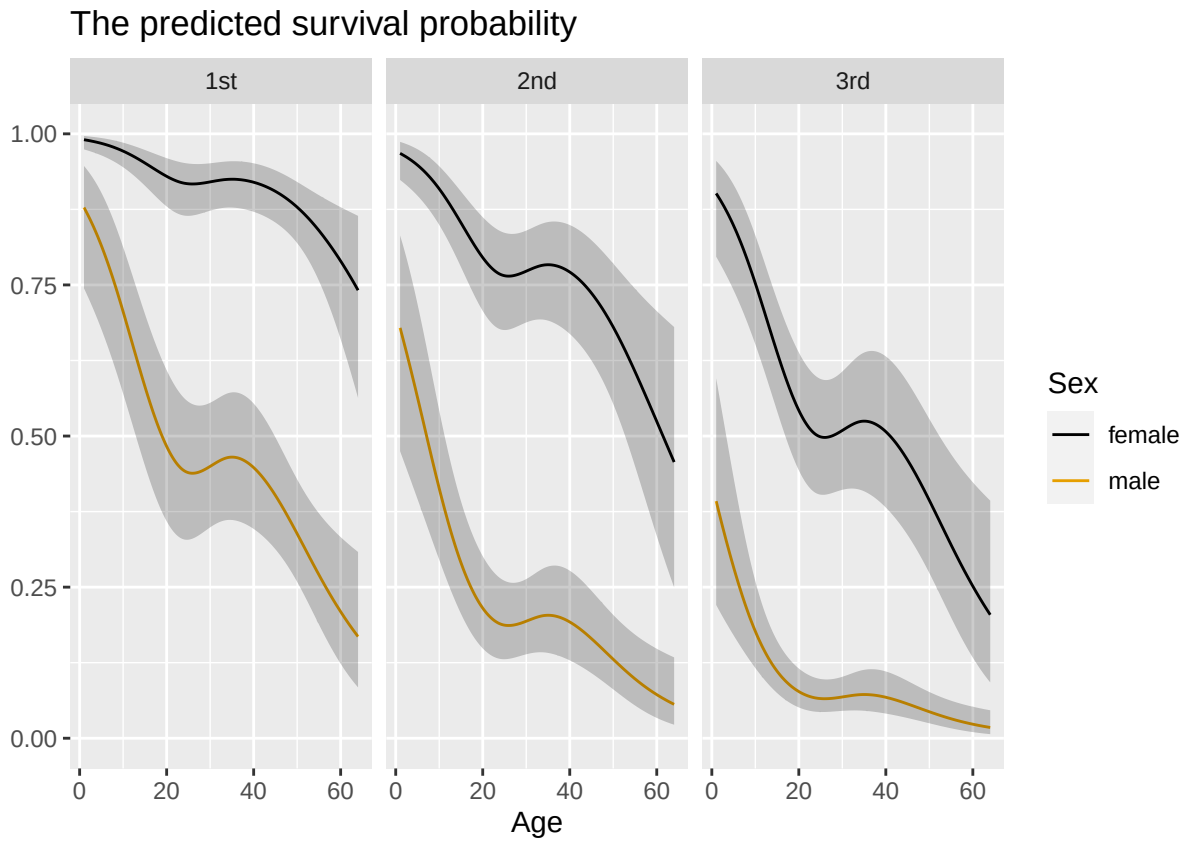

```
model  <- glm(Survived ~ PClass + Age + Sex, data = titanic_df, family = binomial())
summary(model)
```

```
##
## Call:
```

```
## glm(formula = Survived ~ PClass + Age + Sex, family = binomial(),
##     data = titanic_df)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.723  -0.707  -0.392   0.649   2.529
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.75966    0.39757    9.46  < 2e-16 ***
## PClass2nd   -1.29196    0.26008   -4.97  6.8e-07 ***
## PClass3rd   -2.52142    0.27666   -9.11  < 2e-16 ***
## Age         -0.03918    0.00762   -5.14  2.7e-07 ***
## Sexmale     -2.63136    0.20151  -13.06  < 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: 1025.57  on 755  degrees of freedom
## Residual deviance:  695.14  on 751  degrees of freedom
##   (557 observations deleted due to missingness)
## AIC: 705.1
##
## Number of Fisher Scoring iterations: 5
```

we can exponentiate their coefficients to get the odds ratios for survival. For example, the odds ratio for PClass2nd is exp(-1.29196) = 0.274, which suggests that passengers in second-class were 0.274 times as likely to survive as passengers in first-class. Similarly, the odds ratio for Age is exp(-0.03918) = 0.962, which means that for each one-unit increase in age, the odds of survival decrease by a factor of 0.962. The odds ratio for Sexmale is exp(-2.63136) = 0.072, which suggests that males were 0.072 times as likely to survive as females.

```
f <- lrm(Survived ~ Sex + PClass + rcs(Age,4), data=titanic_df)
p <- Predict(f, Age, Sex, PClass, fun=plogis)
plot <- ggplot(p)
plot + ggtitle("The predicted survival probability ")
```



3B) The model with interactions between Age and PClass and between Age and Sex does not seem to improve the fit substantially as compared to the simpler model with main effects of PClass, Age, and Sex.

Firstly, the coefficients for Age and its interactions in the more complex model are not statistically significant, indicating that the effect of Age on survival does not vary significantly across different PClass or Sex groups.

Secondly, the inclusion of interaction terms increases the complexity of the model without much improvement in AIC, indicating that the simpler model is more parsimonious and hence preferable.

Therefore, we can choose the simpler model with main effects of PClass, Age, and Sex as given in A) as the resulting model.

```
# Fit a logistic regression model with interactions
model3 <- glm(Survived ~ PClass * Sex * Age, data = titanic_df, family = "binomial")
summary(model3)
```

```
##
## Call:
## glm(formula = Survived ~ PClass * Sex * Age, family = "binomial",
##     data = titanic_df)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.574  -0.641  -0.381   0.461   2.893
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
```

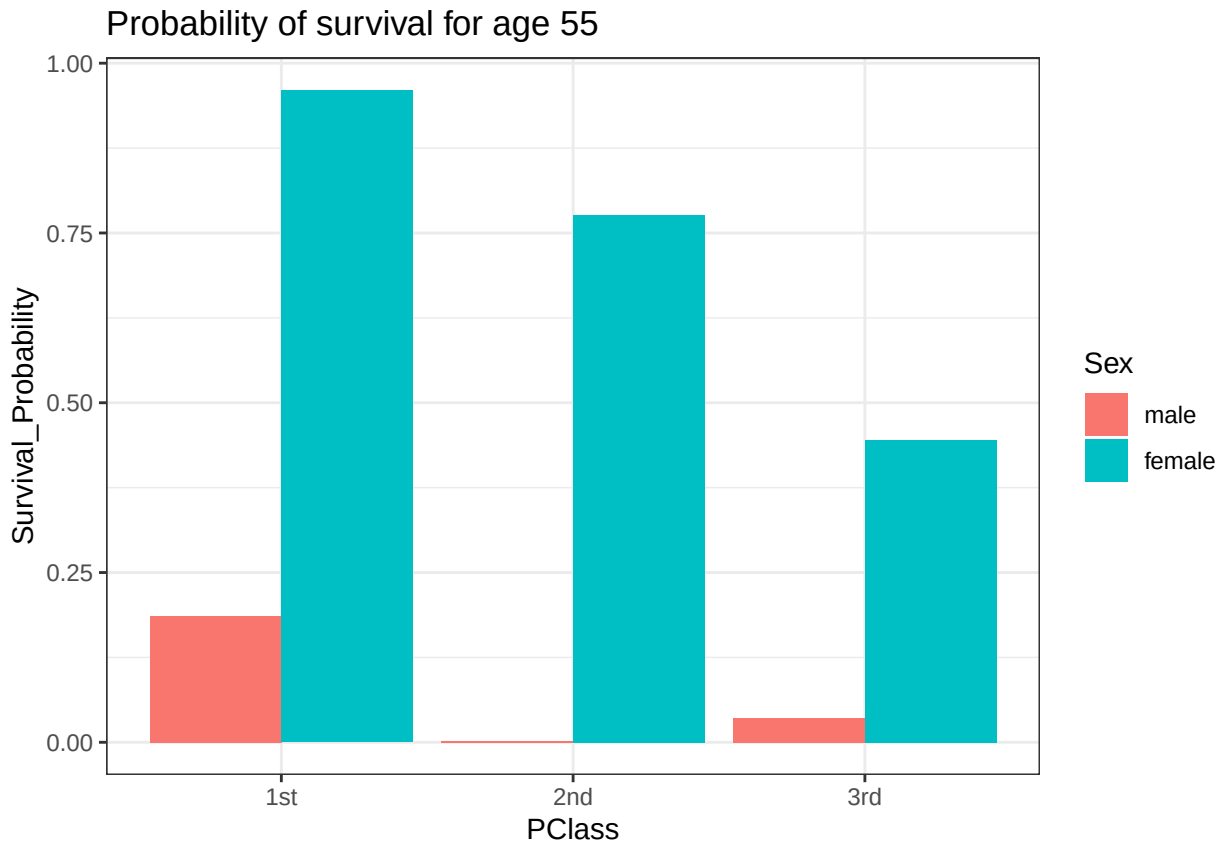
```
## (Intercept)             2.5105     1.1856    2.12    0.034 *
## PClass2nd               0.3693     1.4985    0.25    0.805
## PClass3rd              -2.6915     1.2595   -2.14    0.033 *
## Sexmale                -1.0055     1.3338   -0.75    0.451
## Age                     0.0122     0.0309    0.39    0.694
## PClass2nd:Sexmale      -0.1458     1.7713   -0.08    0.934
## PClass3rd:Sexmale       0.6815     1.4873    0.46    0.647
## PClass2nd:Age          -0.0419     0.0415   -1.01    0.313
## PClass3rd:Age          -0.0128     0.0351   -0.37    0.714
## Sexmale:Age            -0.0664     0.0344   -1.93    0.054 .
## PClass2nd:Sexmale:Age  -0.0478     0.0543   -0.88    0.379
## PClass3rd:Sexmale:Age   0.0164     0.0432    0.38    0.703
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1025.57  on 755  degrees of freedom
## Residual deviance:  639.64  on 744  degrees of freedom
##   (557 observations deleted due to missingness)
## AIC: 663.6
##
## Number of Fisher Scoring iterations: 6
```

```
# Create a new dataset with all possible combinations of PClass, Sex, and Age
newdata <- expand.grid(PClass = c("1st", "2nd", "3rd"),
                       Sex = c("male", "female"),
                       Age = 55)
# Add a column with predicted survival probabilities
newdata$Survival_Probability <- predict(model3, newdata, type = "response")
head(newdata)
```

```
##   PClass    Sex Age Survival_Probability
## 1    1st   male  55              0.18566
## 2    2nd   male  55              0.00206
## 3    3rd   male  55              0.03589
## 4    1st female  55              0.96005
## 5    2nd female  55              0.77673
## 6    3rd female  55              0.44549
```

The table provides the survival probabilities for six different combinations of PClass, Sex, and Age, based on the model used to analyze the Titanic dataset. according to the table, a 55-year-old male passenger in 1st class had a survival probability of 0.18566, while a 55-year-old female passenger in 1st class had a much higher survival probability of 0.96005. Similarly, a 55-year-old male passenger in 2nd class had a very low survival probability of 0.00206, while a 55-year-old female passenger in 2nd class had a much higher survival probability of 0.77673.

```
p<- ggplot(newdata, aes(x = PClass, y = Survival_Probability, fill = Sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_bw()
p + ggtitle("Probability of survival for age 55")
```



3C)

We could use Logistic Regression to model the probability of a certain passenger surviving or not. To evaluate the model, we could use R^2 or Accuracy. To implement the model, we would need to clean the dataset, handling missing values, encoding the categorical variables, and normalizing the numerical variables.

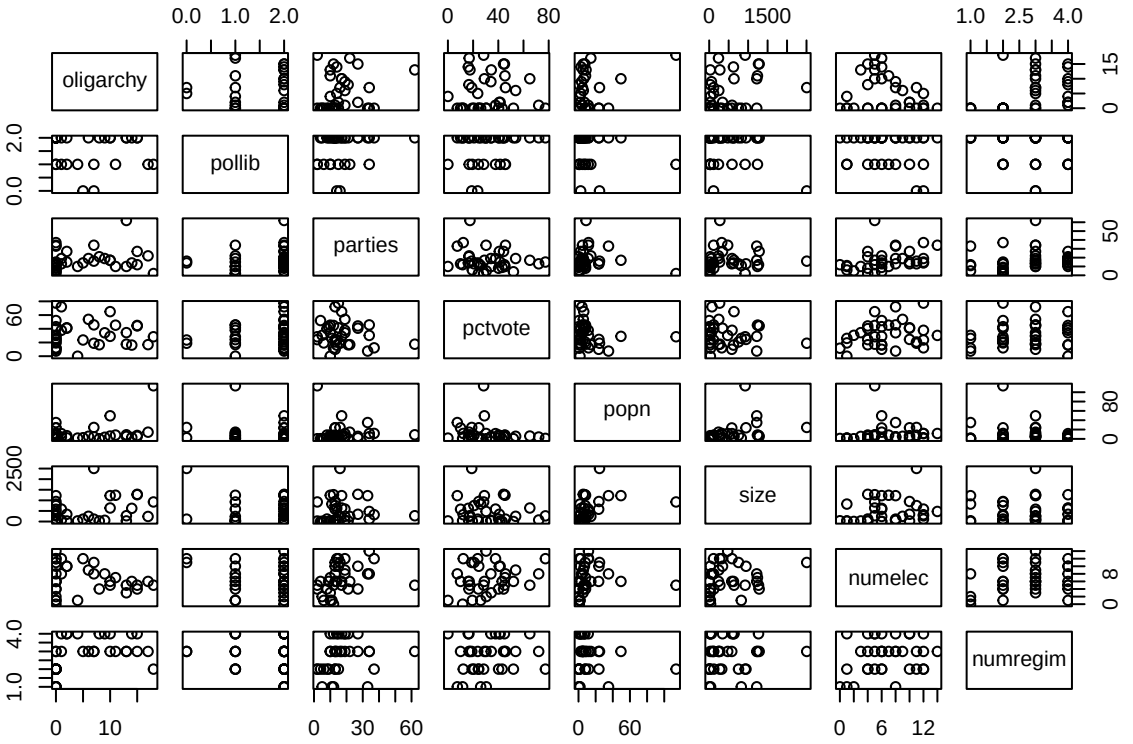
Excersice 4

A)

We check for correlation between all pairs of variables.The plot shows that there is no correlation.

We perform Poisson regression and find that `oligarchy`, `pollib` and `parties` have a significant effect on`miltcoup`, because their p-values are <0.05.

```
data = read.table(file = "coups.txt", header = TRUE)
pairs(data[,-1])
```




```
glmcoups = glm(miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size + numelec + numregim, fa
summary(glmcoups)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn + size + numelec + numregim, family = poisson, data = data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.344  -0.954  -0.259   0.391   1.695
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.510269   0.905330   -0.56   0.5730
## oligarchy    0.073081   0.034596    2.11   0.0346 *
## pollib      -0.712978   0.272563   -2.62   0.0089 **
## parties      0.030774   0.011187    2.75   0.0059 **
## pctvote      0.013872   0.009753    1.42   0.1549
## popn         0.009343   0.006595    1.42   0.1566
## size        -0.000190   0.000248   -0.76   0.4445
## numelec     -0.016078   0.065484   -0.25   0.8060
## numregim     0.191735   0.229289    0.84   0.4030
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 28.668  on 27  degrees of freedom
## AIC: 111.5
##
## Number of Fisher Scoring iterations: 6
```

B)

We will use the step-down approach to reduce the number of explanatory variables. This means we keep the variables that have the most significant effect. Analyzing the `summary` in a), we iterate through and remove the variables with the highest p-values. We end up with `oligarchy`, `pollib` and `parties`. Comparing the results to a), the step down approach model is .....

```
glmcoups2 = glm(miltcoup~oligarchy+pollib+parties, family = poisson, data = data)
summary(glmcoups2)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties, family = poisson,
##     data = data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.358  -1.042  -0.286   0.628   1.752
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.25138    0.37269    0.67    0.500
## oligarchy    0.09262    0.02178    4.25  2.1e-05 ***
## pollib      -0.57410    0.20438   -2.81    0.005 **
## parties      0.02206    0.00896    2.46    0.014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 32.856  on 32  degrees of freedom
## AIC: 105.7
##
## Number of Fisher Scoring iterations: 5
```

C) #Using the model from b), predict the number of coups for a hypothetical country for all the three levels of political liberalization and the (overall) averages of all the other (numerical) characteristics. Comment on your findings.

The predicted average of coups per country increases as the policitical liberalization decreases.???

```
avg1 =0.25138+0.09262*mean(data$oligarchy)-0.57410*0+0.02206*mean(data$parties)
avg2 =0.25138+0.09262*mean(data$oligarchy)-0.57410*1+0.02206*mean(data$parties)
avg3 =0.25138+0.09262*mean(data$oligarchy)-0.57410*2+0.02206*mean(data$parties)
```

```r
avg =c(exp(avg1), exp(avg2), exp(avg3))
avg1; avg2; avg3; avg
```

```
## [1] 1.11
```

```
## [1] 0.538
```

```
## [1] -0.0363
```

```
## [1] 3.040 1.712 0.964
```