STAT 504: Linear Regression

Homework 4

Pratima K C

Question 1 (a) (1 point) Perform a logistic regression and report the fitted regression equation.

Answer: The fitted regression equation is given below:

$E[\hat{\eta}] = -4.73931 + 0.06773 \times income + 0.59863 \times age$

$Ey = P\{y = 1\}$

$P\{y = 1\} = \frac{1}{1+e^{-\eta}} = \frac{1}{1+e^{-(4.73931+0.06773 \times income+0.59863 \times age)}}$

The logistic regression summary is given below:

```
##
## Call:
## glm(formula = purchase ~ income + age, family = "binomial", data = car_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6189  -0.8949  -0.5880   0.9653   2.0846
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.73931    2.10195  -2.255   0.0242 *
## income       0.06773    0.02806   2.414   0.0158 *
## age          0.59863    0.39007   1.535   0.1249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 44.987  on 32  degrees of freedom
## Residual deviance: 36.690  on 30  degrees of freedom
## AIC: 42.69
##
## Number of Fisher Scoring iterations: 4
```

Question 1 (b) (2 points) Estimate exp ($\hat{\beta}_{income}$) and exp ($\hat{\beta}_{age}$) and give an interpretation of these estimates.

Answer: The estimate of exp ($\hat{\beta}_{income}$)=1.07007 and exp ($\hat{\beta}_{age}$)=1.8196. The odds of purchasing a car increases by 7.007% for every $1000 increase in the income when age is constant. The odds of purchasing a car increases by 82 (81.9)% for every year incease in the age when income is constant.

```
coef=exp(fit_car$coefficients)
coef
```

```
## (Intercept)      income         age
## 0.008744682 1.070079093 1.819627221
```

Question 1 (c) (1 point) How large is the estimated probability that a family with a yearly household income of 50 000 US $ and whose oldest car is 3 years old will buy a new car?

Answer: The estimated probability that a family with a yearly househod income of $50,000 US and whose oldest car is 3 years old to buy a new car is 0.609.

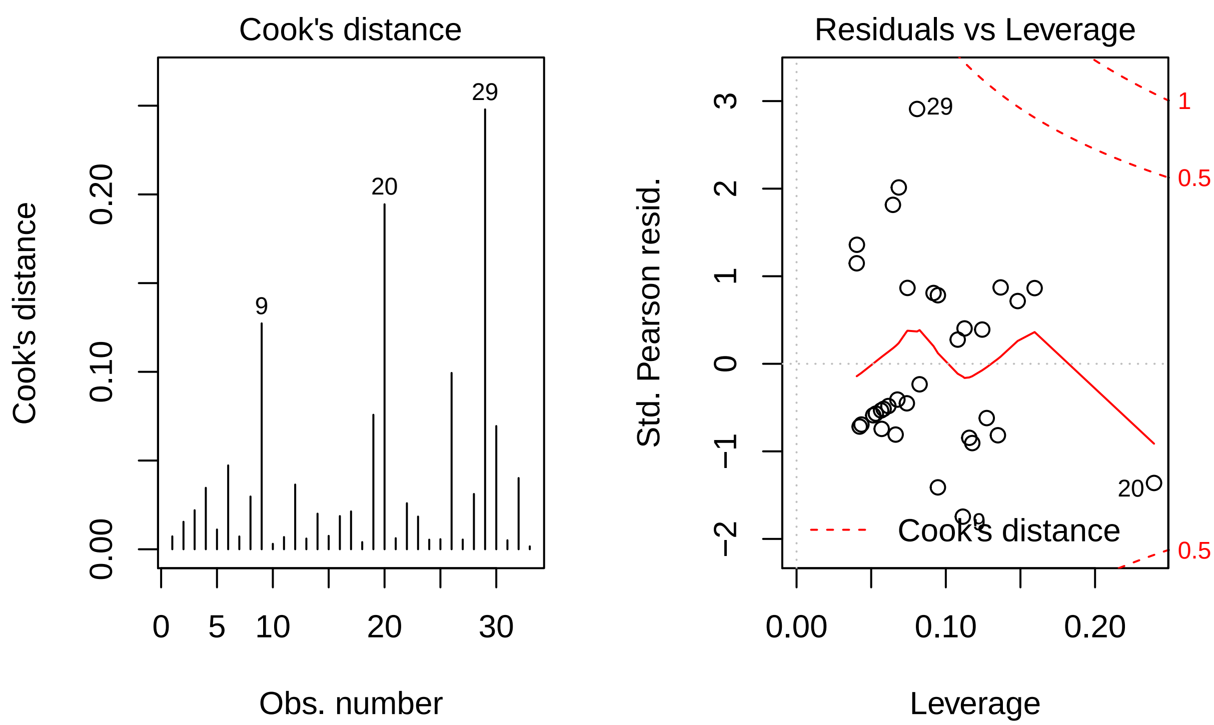
```
new_car=data.frame(income=50, age=3)
predict(fit_car,new_car, type="response")
```

```
##         1
## 0.6090245
```

Question 1 (d) (1 point) Check for the presence of points with a large Cook's distance.

Answer: The point with larger Cook's distance are 9, 20, & 29. However, the Cook's distance of these point are less than 0.5. Therefore, these points may not necessaryly be the outlier. The Cook's distance plot is given below:

```
## [1] 0.3148805
```



Question 1 (e) (1 point) Is the predictor age significant at the 5% level?

Answer: The p-value of coefficient of age is 0.1249 that is larger than 0.05. Therefore, predictor age is not significant at the 5% level.

Question 1 (f) (1 point) Is there a non-negligible interaction between income and age?

Answer: The interaction between income and age is negligible because the p-value of the coefficient of interaction term (income:age) is 0.276, larger than 0.05 that mean it is not significant at 5% level. Also, based on the anova test the p-value=0.2569 is very high that is not significant at 0.05 level. So we fail to reject the null hypothesis and we choose the smaller model without interaction term (purchase ~ income + age).

```
##
## Call:
## glm(formula = purchase ~ income * age, family = "binomial", data = car_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6096  -0.8222  -0.5334   0.8731   1.9924
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.372993   2.862477  -0.829    0.407
## income       0.001326   0.064770   0.020    0.984
## age         -0.303860   0.890512  -0.341    0.733
## income:age   0.028860   0.026493   1.089    0.276
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 44.987  on 32  degrees of freedom
## Residual deviance: 35.404  on 29  degrees of freedom
## AIC: 43.404
##
## Number of Fisher Scoring iterations: 4
```

```
## Analysis of Deviance Table
##
## Model 1: purchase ~ income + age
## Model 2: purchase ~ income * age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        30     36.690
## 2        29     35.404  1   1.2855   0.2569
```

Question 2 (a) (1 point) In order to fit a binomial logistic regression model construct a response matrix with two columns containing the number of people with and without hypertension, respectively.

Answer: The code is given below:

```
no.yes <- c("No", "Yes")
smoking <- gl(2,1,7, no.yes)
obesity <- gl(2,2,7, no.yes)
snoring <- gl(2,4,7, no.yes)
n.total <- c(60, 17, 8, 187, 85, 51, 23)
n.hyper <- c(5, 2, 1, 35, 13, 15, 8)

data=data.frame(cbind(smoking, obesity, snoring))
data
```

```
##   smoking obesity snoring
```

```
## 1       1       1       1
## 2       2       1       1
## 3       1       2       1
## 4       2       2       1
## 5       1       1       2
## 6       2       1       2
## 7       1       2       2
```

```
hyper_matrix=cbind(hyper=n.hyper, non_hyper=n.total-n.hyper)
hyper_matrix
```

```
##      hyper non_hyper
## [1,]     5        55
## [2,]     2        15
## [3,]     1         7
## [4,]    35       152
## [5,]    13        72
## [6,]    15        36
## [7,]     8        15
```

Question 2 (b) (1 point) Fit a binomial regression model to the data. Assess the goodness-of-fit via the chi-square test for the residual deviance.

Answer: The p-value of the chi-square test is 0.006649 which is small and is significant at 0.05 level. That means the larger model (hyper ~ smoking + obesity + snoring) is a good fit.

```
fit_hyper=glm(formula = hyper_matrix ~ data$smoking+data$obesity+data$snoring, family = "binomial")
summary(fit_hyper)
```

```
##
## Call:
## glm(formula = hyper_matrix ~ data$smoking + data$obesity + data$snoring,
##     family = "binomial")
##
## Deviance Residuals:
##        1         2         3         4         5         6         7
##  0.50780   0.10458   0.02847  -0.21903  -0.63361   0.32485   0.51753
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.9773     1.1351  -4.385 1.16e-05 ***
## data$smoking   0.5488     0.3132   1.752  0.07976 .
## data$obesity   0.6668     0.3455   1.930  0.05360 .
## data$snoring   1.1184     0.3656   3.059  0.00222 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13.3181  on 6  degrees of freedom
## Residual deviance:  1.0924  on 3  degrees of freedom
## AIC: 34.011
##
## Number of Fisher Scoring iterations: 4
```

```r
fit_hyper_empty=glm(formula= hyper_matrix ~ 1, family = "binomial")
summary(fit_hyper_empty)
```

```
##
## Call:
## glm(formula = hyper_matrix ~ 1, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1952  -0.7394  -0.4469   1.0037   1.9201
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.4942     0.1245     -12   <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: 13.318  on 6  degrees of freedom
## Residual deviance: 13.318  on 6  degrees of freedom
## AIC: 40.237
##
## Number of Fisher Scoring iterations: 4
```

```r
anova(fit_hyper,fit_hyper_empty, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: hyper_matrix ~ data$smoking + data$obesity + data$snoring
## Model 2: hyper_matrix ~ 1
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         3     1.0924
## 2         6    13.3181 -3  -12.226 0.006649 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 2 (c) (2 points) Which variables in the model are significant at the 5% level? Use the likelihood-ratio test to obtain the answer. Hint: drop1 function in R.

Answer: Based on likelihood-ratio test the snoring predictor has p-value=0.00132 that is significant at 0.05 level. However, smoking predictor has p-value=0.07788 & obesity predictor has p-value= 0.05169. Both of these predictor have larger p-value than 0.05. Therefore, these two variables are not significant at 0.05 level.

```
## Single term deletions
##
## Model:
## hyper_matrix ~ data$smoking + data$obesity + data$snoring
##              Df Deviance    AIC    LRT Pr(>Chi)
## <none>            1.0924 34.011
## data$smoking  1   4.2010 35.120 3.1086  0.07788 .
## data$obesity  1   4.8781 35.797 3.7857  0.05169 .
```

```
## data$snoring  1  11.4062 42.325 10.3138  0.00132 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 2(d) (2 points) Find a suitable sub-model compared to the model above using likelihood-ratio tests and backward elimination based on p-values. What is the model that you would choose?

Answer: Based on the summary of the full model, first we droped the smoking from the model because it has the larger p-value (0.07976) that is larger than 0.05 level and larger than obesity and snoring. Then created a new model excluding smoking such as (hyper ~ obesity + snoring). Then we used likelihood-ratio to test if any of these variable are insignificant. Based on the output of likelihood-ratio both the predictor variables have significant p-value (obesity:p-value= 0.013904 & snoring:p-value= 0.004421) at 0.05 level in the model. Therefore, we choose the the model. It is given below:

hyper ~ obesity + snoring

The summary of chosen model with likelihood-ratio test is given below:

```r
fit_hyper_drop_smoking=glm(formula = hyper_matrix ~ data$obesity+data$snoring, family = "binomial")
summary(fit_hyper_drop_smoking)
```

```
##
## Call:
## glm(formula = hyper_matrix ~ data$obesity + data$snoring, family = "binomial")
##
## Deviance Residuals:
##        1         2         3         4         5         6         7
## -0.28404   0.32506  -0.44798   0.13068  -1.21440   1.52066  -0.09844
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.9496     0.8764  -4.507 6.59e-06 ***
## data$obesity   0.7745     0.3225   2.401   0.0163 *
## data$snoring   0.9075     0.3240   2.801   0.0051 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13.318  on 6  degrees of freedom
## Residual deviance:  4.201  on 4  degrees of freedom
## AIC: 35.12
##
## Number of Fisher Scoring iterations: 4
```

```r
drop1(fit_hyper_drop_smoking, test="Chisq")
```

```
## Single term deletions
##
## Model:
## hyper_matrix ~ data$obesity + data$snoring
##              Df Deviance    AIC    LRT Pr(>Chi)
## <none>             4.201 35.120
```

```
## data$obesity  1   10.251 39.170 6.0503 0.013904 *
## data$snoring  1   12.303 41.222 8.1021 0.004421 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 2 (e) (1 point) Compare the observed and fitted proportions for hypertension using the model you found in d). Additionally, compare the fitted and observed counts of hypertension in each group. Note that the fitted count is not always a whole number.

Answer: The fitted and observed proportions for hypertension seems to be fairly close to each other. However, for observation no. 5 the fitted proportion is little higher than observed. And for observaton no. 6 the fitted proportion is less than the observed proportions.

Similarly, the fitted counts were very similar to actual count. I rounded the fitted count because they were a whole number. Based on the rounded fitted count, the fitted counts are same in observation no. 2, 3, 7. The observation no. 1 & 5 has higher fitted count and observation no. 4 & 6 has less fitted cound than actual count.

Overall we could say that the fitted values are a good estimates of actual values. The code & table is given below:

```r
#fitted(fit_hyper_drop_smoking)
fitted=predict(fit_hyper_drop_smoking, data, type = "response")

#for the fitted proportions
cbind(fitted, acutal_prop=n.hyper/n.total)
```

```
##      fitted acutal_prop
## 1 0.0938417  0.08333333
## 2 0.0938417  0.11764706
## 3 0.1834574  0.12500000
## 4 0.1834574  0.18716578
## 5 0.2042220  0.15294118
## 6 0.2042220  0.29411765
## 7 0.3576440  0.34782609
```

```r
#for the fitted counts
cbind(fitted_count=fitted*n.total, fitted_round=round(fitted*n.total), actual_count=n.hyper)
```

```
##   fitted_count fitted_round actual_count
## 1     5.630502            6            5
## 2     1.595309            2            2
## 3     1.467659            1            1
## 4    34.306530           34           35
## 5    17.358868           17           13
## 6    10.415321           10           15
## 7     8.225811            8            8
```

Question 3(a) (2 points) Load the data. Apply a log transformation on the response upo3 and remove the outlier (observation number 92).

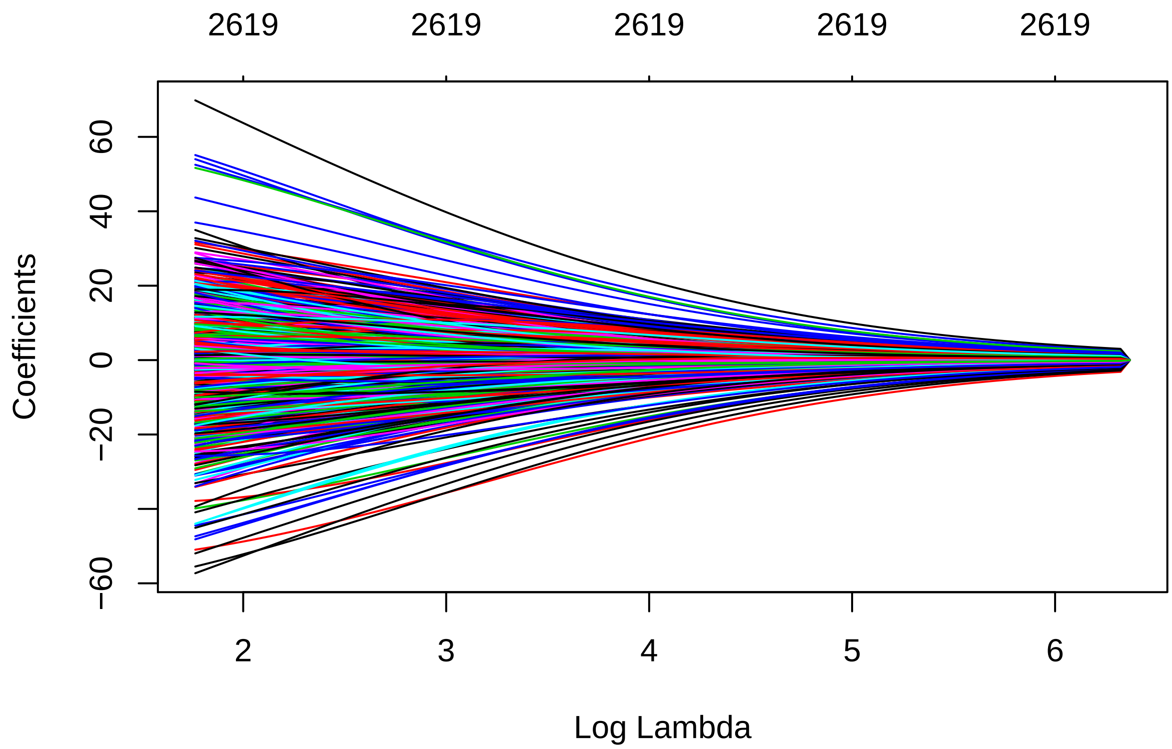
Answer: In the ridge regression the coefficient doesnot become zero even when the log lambda is greater than 6. It also looks like all the far out coefficients in either direction are moving towards zero in same rate but never became exact zero.

In the lasso regression all coefficients shrink to zero when log lambda is around -2. Similar, in the elastic net regression all coefficients shrink to zero when log lambda is around -1. In both lasso & elastic net regression different coefficients far out have different rates to shrink to zero. However, the elastic net regression have larger lambda term that rule out the predictors with larger penality than lasso regression.

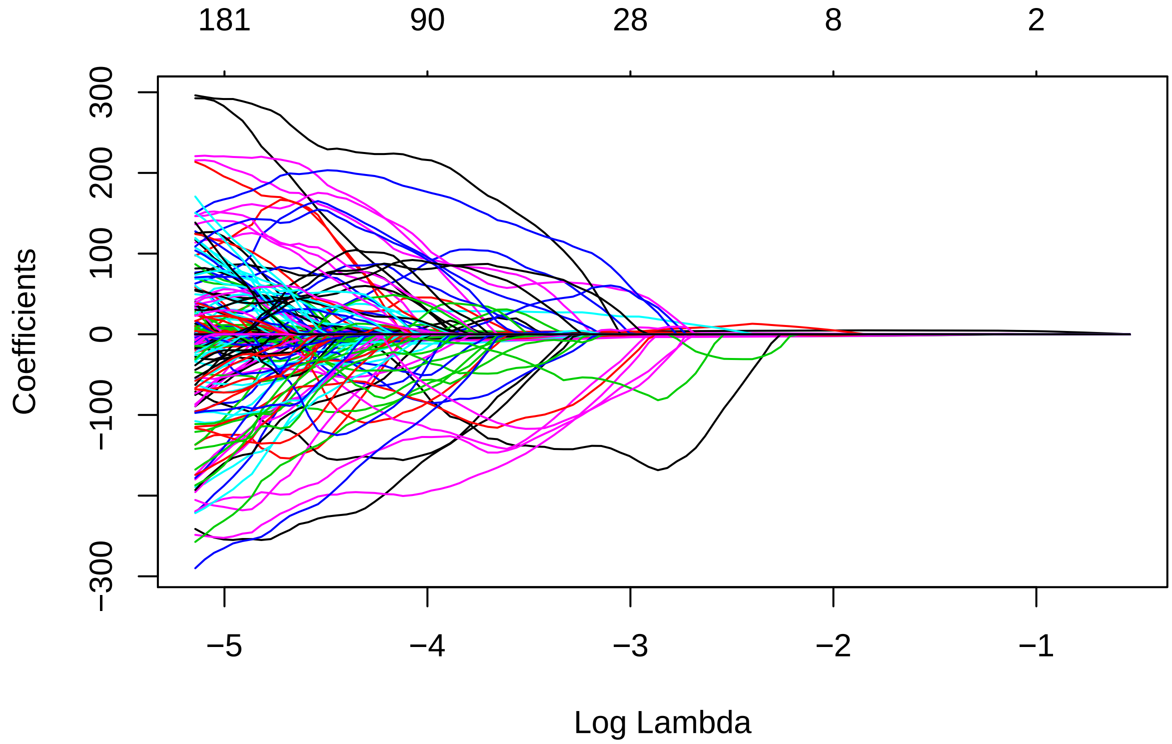
The code is given below:

```
library(gss)
data(ozone, package= "gss")

logupo3=log(ozone$upo3)
ozone$logupo3=logupo3
d.ozone.e=ozone[-92,-1]

# generate a design matrix
require(sfsmisc)
ff <- wrapFormula(logupo3~., data=d.ozone.e, wrapString="poly(*,degree=3)")
ff <- update(ff, logupo3 ~ .^3)
mm <- model.matrix(ff, data=d.ozone.e)

# penalized regression
require(glmnet)
ridge <- glmnet(mm, d.ozone.e$logupo3, alpha=0)
lasso <- glmnet(mm, d.ozone.e$logupo3, alpha=1)
elnet <- glmnet(mm, d.ozone.e$logupo3, alpha=.5)
```

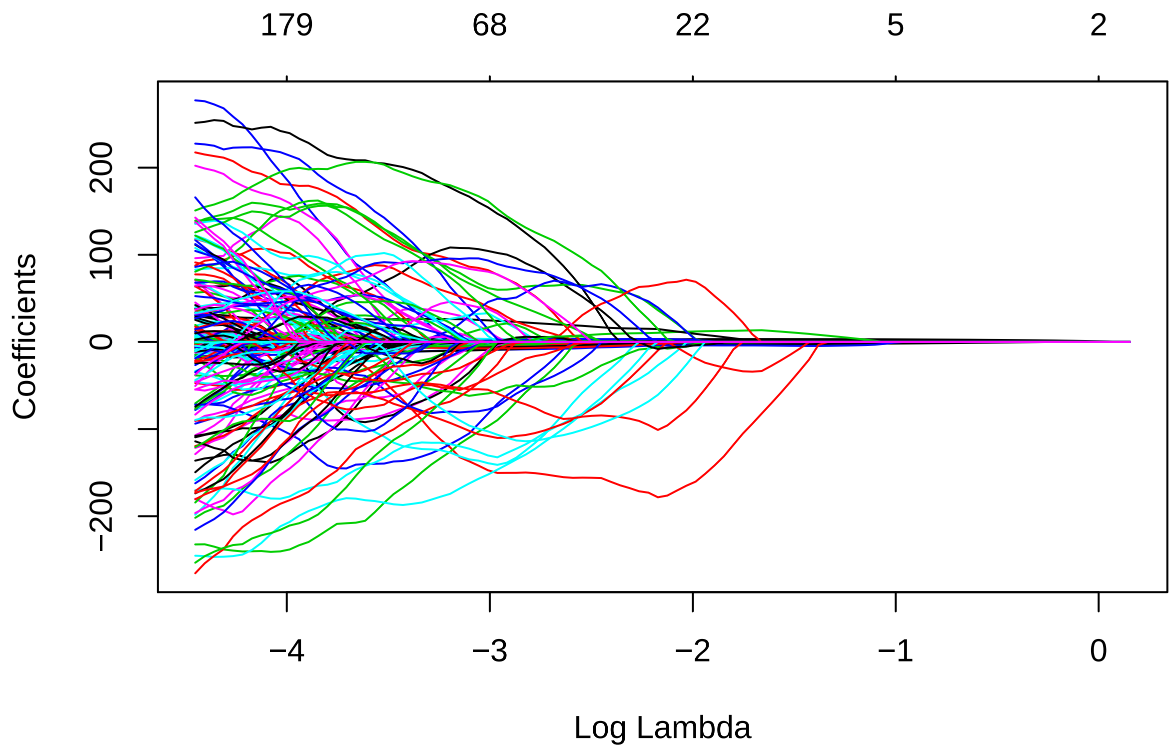
```
plot(ridge, xvar="lambda")
```




```
plot(lasso, xvar="lambda")
```




```r
plot(elnet, xvar="lambda")
```



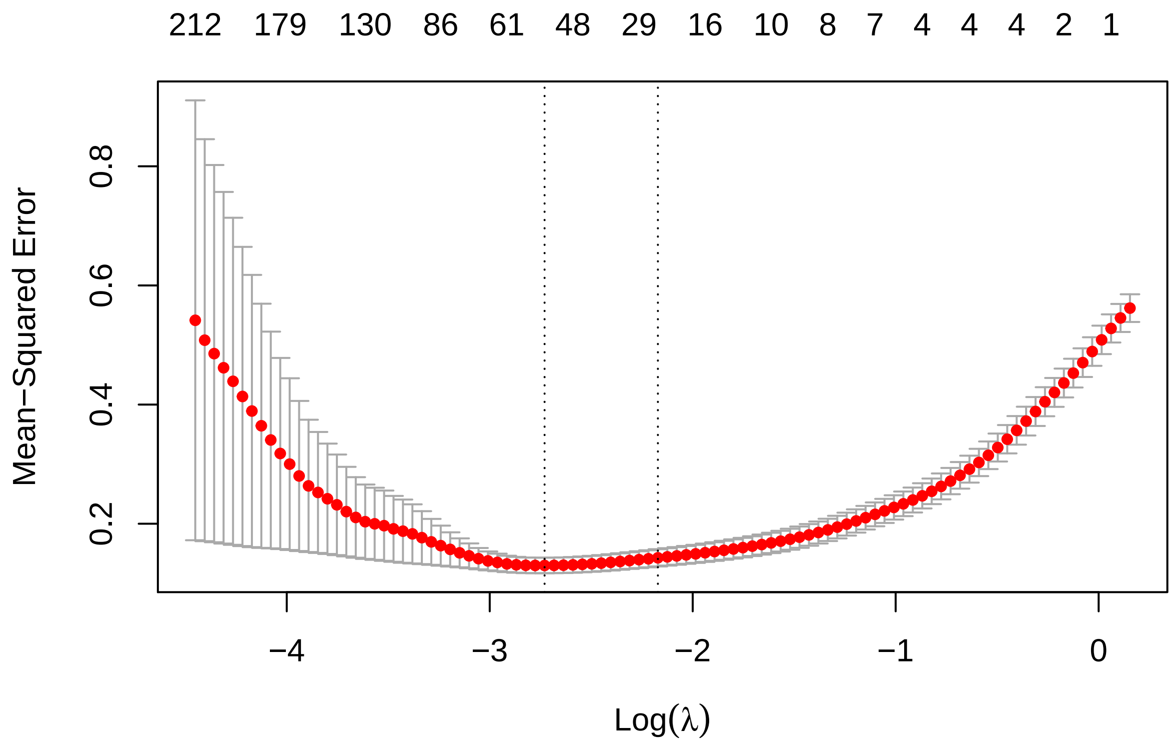
Question 3 (b) (2 points) Select an optimal tuning parameter λ with an elastic net penalty $\alpha =$ 0.5 via 10-fold cross validation. Find an optimal λ according to the "1-std error rule" from a plot that shows the mean squared error as a function of log(λ).

Answer: The optial lambda as 1-std error rule, is 0.1140. The plot is given below:

```
set.seed(1)
cv.eln <- cv.glmnet(mm,d.ozone.e$logupo3,alpha=0.5, nfolds=10)
cv.eln$lambda.1se
```

```
## [1] 0.1140085
```

```
plot(cv.eln)
```



Question 4 (a) (2 points) First fit an OLS with all variables and perform a residual analysis. Hint: Check whether all varialbes are encoded properly (see as.factor).

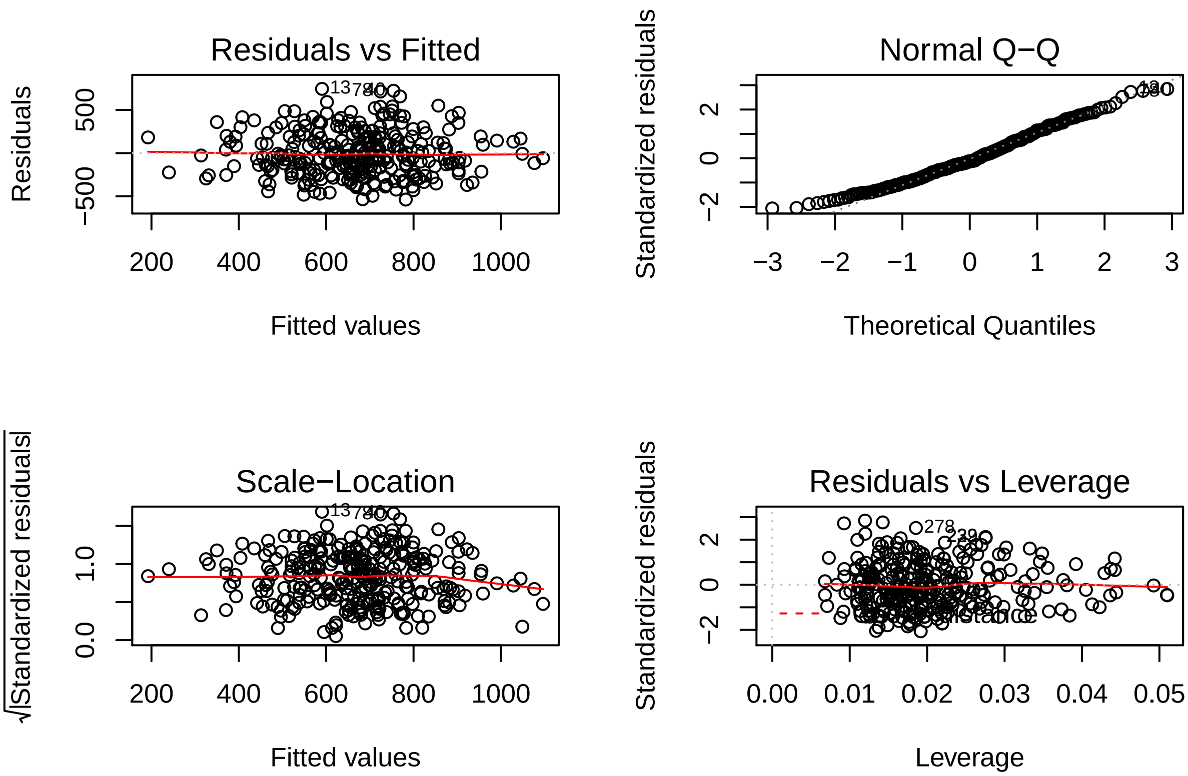
Answer: Based on the QQ plot normality assumption is not violated. The TA plot shows that the mean of residual is zero since there is no curvetur in mean line therefore this assumption is not violated. However, the variance is little tapering towards both ends (lower and higher fitted values) but over all there is a constant variance. Therefore, it seems to hold the constant variance assumption too. The summary in graphs are given below:

```
load(file="CustomerWinBack.rda")
#str(cwb)
cwb$gender=as.factor(cwb$gender)
fit1=lm(formula=duration~., data = cwb)
summary(fit1)
```

```
##
## Call:
## lm(formula = duration ~ ., data = cwb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -538.14 -196.27  -34.79  182.57  744.55
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 836.15414  101.51985   8.236 6.21e-15 ***
```

```
## offer        -11.91067    3.70155  -3.218  0.00144 **
## lapse          1.09216    0.38843   2.812  0.00527 **
## price         -8.32047    1.03278  -8.056 2.08e-14 ***
## gender1      113.45371   31.73589   3.575  0.00041 ***
## age            0.06461    1.08476   0.060  0.95255
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 263.3 on 289 degrees of freedom
## Multiple R-squared:   0.24,  Adjusted R-squared:  0.2269
## F-statistic: 18.25 on 5 and 289 DF,  p-value: 9.598e-16
```

```r
par(mfrow=c(2,2))
plot(fit1)
```



Question 4 (b) (1 point) Choose a model using stepwise model selection (forward-backward) starting from the model given in part a) and the AIC criterion. What predictors are included in the optimal model according to the above selection?

Answer: Based on the stepwise model selection (forward-backward) and AIC criterion the optimal model has AIC 3292.27 value and includes predictors such as offer, lapse, price and genter. The optimal model is given below:

$duration \sim offer + lapse + price + gender$

```r
# AIC forward-backward (both) stepwise variable selection:
scp <- list(lower = ~ 1, upper = ~ offer+lapse+price+gender+age, data=cwb)
fit.aic <- step(fit1, scope = scp, direction = "both", k =2)
```

```
## Start:  AIC=3294.27
## duration ~ offer + lapse + price + gender + age
##
##          Df Sum of Sq      RSS    AIC
## - age     1       246 20041387 3292.3
## <none>                20041141 3294.3
## - lapse   1    548224 20589366 3300.2
## - offer   1    718009 20759150 3302.6
## - gender  1    886259 20927400 3305.0
## - price   1   4500960 24542101 3352.0
##
## Step:  AIC=3292.27
## duration ~ offer + lapse + price + gender
##
##          Df Sum of Sq      RSS    AIC
## <none>                20041387 3292.3
## + age     1       246 20041141 3294.3
## - lapse   1    552534 20593922 3298.3
## - offer   1    733612 20774999 3300.9
## - gender  1    888951 20930338 3303.1
## - price   1   4503240 24544627 3350.1
```

```r
summary(fit.aic)
```

```
##
## Call:
## lm(formula = duration ~ offer + lapse + price + gender, data = cwb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -539.09 -196.33  -33.47  182.37  745.29
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 838.6231    92.5116   9.065  < 2e-16 ***
## offer       -11.8743     3.6445  -3.258 0.001255 **
## lapse         1.0896     0.3853   2.828 0.005017 **
## price        -8.3215     1.0309  -8.072 1.85e-14 ***
## gender1     113.3137    31.5943   3.587 0.000393 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 262.9 on 290 degrees of freedom
## Multiple R-squared:   0.24,  Adjusted R-squared:  0.2295
## F-statistic:  22.9 on 4 and 290 DF,  p-value: < 2.2e-16
```

Question 4 (c) (1 point) Choose a model using stepwise model selection (forward-backward) starting from the model given in part a) and the BIC criterion. What predictors are included in the optimal model according to the above selection?

Answer: Based on the stepwise model selection (forward-backward) and BIC criterion the optimal model has BIC 3310.7 value and includes predictors offer, lapse, price and genter. The optimal model is given below:

$duration \sim offer + lapse + price + gender$

```r
# BIC forward-backward (both) stepwise variable selection:
scp <- list(lower = ~ 1, upper = ~ offer+lapse+price+gender+age, data=cwb)
fit.bic <- step(fit1, scope = scp, direction = "both", k = log(295))
```

```
## Start:  AIC=3316.39
## duration ~ offer + lapse + price + gender + age
##
##          Df Sum of Sq      RSS    AIC
## - age     1       246 20041387 3310.7
## <none>                20041141 3316.4
## - lapse   1    548224 20589366 3318.7
## - offer   1    718009 20759150 3321.1
## - gender  1    886259 20927400 3323.5
## - price   1   4500960 24542101 3370.5
##
## Step:  AIC=3310.7
## duration ~ offer + lapse + price + gender
##
##          Df Sum of Sq      RSS    AIC
## <none>                20041387 3310.7
## - lapse   1    552534 20593922 3313.0
## - offer   1    733612 20774999 3315.6
## + age     1       246 20041141 3316.4
## - gender  1    888951 20930338 3317.8
## - price   1   4503240 24544627 3364.8
```

```r
summary(fit.bic)
```

```
##
## Call:
## lm(formula = duration ~ offer + lapse + price + gender, data = cwb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -539.09 -196.33  -33.47  182.37  745.29
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 838.6231    92.5116   9.065  < 2e-16 ***
## offer       -11.8743     3.6445  -3.258 0.001255 **
## lapse         1.0896     0.3853   2.828 0.005017 **
## price        -8.3215     1.0309  -8.072 1.85e-14 ***
## gender1     113.3137    31.5943   3.587 0.000393 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 262.9 on 290 degrees of freedom
## Multiple R-squared:   0.24,  Adjusted R-squared:  0.2295
## F-statistic:  22.9 on 4 and 290 DF,  p-value: < 2.2e-16
```

Question 4 (d) (2 points) What is the optimal lambda (see lambda.1se in R)? What predictors are included in this model? What is the fitted ridge equation?

Answer: The optimal lambda is 428.7102. The predictors included in the model are offer, lapse, price, genger1, and age. The fitted ridge equation is given below:

$E[du\hat{ration}] = 725.3438498 - 3.7541471 \times offer + 0.4083362 \times laspe - 3.4432545 \times price + 40.8052383 \times gender - 0.1931633 \times age$

```
library(glmnet)
## Lasso does not work with factor variables
set.seed(1)
xx <- model.matrix(duration~ 0+., cwb)[,-4]
yy <- cwb$duration
cv.rigdge=cv.glmnet(xx, yy, alpha=0)
optimal_lamba_R=cv.rigdge$lambda.1se
optimal_lamba_R
```

```
## [1] 428.7102
```

```
fit.rigdge=glmnet(xx,yy, alpha = 0, lambda = optimal_lamba_R)
coef(fit.rigdge)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 725.3438498
## offer        -3.7541471
## lapse         0.4083362
## price        -3.4432545
## gender1      40.8052383
## age          -0.1931633
```

Question 4 e (2 points) Fit a lasso regression with optimized λ. What is the optimal lambda (see lambda.1se in R)? What predictors are included in this model? What is the fitted lasso equation?

Answer: The optimal lambda is 49.29126. The predictors included in the model is price. The fitted ridge equation is given below:

$E[du\hat{ration}] = 675.433450 - 5.094912 \times price$

```
library(glmnet)
## Lasso does not work with factor variables
set.seed(1)
xx <- model.matrix(duration~ 0+., cwb)[,-4]
yy <- cwb$duration
cv.lasso=cv.glmnet(xx, yy, alpha=1)
optimal_lamba=cv.lasso$lambda.1se
optimal_lamba
```

```
## [1] 49.29126
```

```r
fit.lasso=glmnet(xx,yy, alpha = 1, lambda = optimal_lamba)
coef(fit.lasso)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept) 675.433450
## offer         .
## lapse         .
## price        -5.094912
## gender1       .
## age           .
```

Question 4 (f) (2 points) Finally, use a 5-fold cross validation to compare the predictive performance of all of the models in this task. What are the best and worst performing models?

Answer: The AIC & BIC models have the smallest mean squared prediciton error. Therefore, these are the best performing models. However, lasso regresion has the higher mean squared prediction error value so it is the worst performing model.

```r
## cross validation preparation
pre.ols <- c()
pre.aic <- c()
pre.bic <- c()
pre.rr <- c()
pre.las <- c()
folds <- 5
sb <- round(seq(0,nrow(cwb),length=(folds+1)))

## cross validation Loop
for (i in 1:folds){
  ## define training and test datasets
  test <- (sb[((folds+1)-i)]+1):(sb[((folds+2)-i)])
  train <- (1:nrow(cwb))[-test]

  ## fit models
  fit.ols <- lm(duration ~ ., data=cwb[train,])
  fit.aic <- lm(duration ~ offer+lapse+price+gender, data=cwb[train,])
  fit.bic <- lm(duration ~ offer+lapse+price+gender, data=cwb[train,])
  xx <- model.matrix(duration~0+., cwb[train,])[,-4]
  yy <- cwb$duration[train]
  fit.rr <- glmnet(xx,yy, lambda = cv.rigdge$lambda.1se, alpha =0)
  fit.las <- glmnet(xx,yy, lambda = cv.lasso$lambda.1se, alpha = 1)

  ## create predictions
  pre.ols[test] <- predict(fit.ols, newdata=cwb[test,])
  pre.aic[test] <- predict(fit.aic, newdata=cwb[test,])
  pre.bic[test] <- predict(fit.bic, newdata=cwb[test,])
  pre.rr[test] <- model.matrix(duration~., cwb[test,])%*%as.numeric(coef(fit.rr))
  pre.las[test] <- model.matrix(duration~., cwb[test,])%*%as.numeric(coef(fit.las))

}
## Finally, compute the mean squared prediction error:
mean((cwb$duration-pre.ols)^2)
```

```
## [1] 70795.21
```

```r
mean((cwb$duration-pre.aic)^2)
```

```
## [1] 70216.84
```

```r
mean((cwb$duration-pre.bic)^2)
```

```
## [1] 70216.84
```

```r
mean((cwb$duration-pre.rr)^2)
```

```
## [1] 78001.73
```

```r
mean((cwb$duration-pre.las)^2)
```

```
## [1] 78256.12
```