Final Exam Solutions Fall 2020

Hakan Gogtas

12/5/2020

Question 1

Use the "Final Exam Fall 2020 Question 1.csv" data set. The observations are listed in time order. Variables are;

Website.Delivered (Y): Number of websites completed and delivered to customers during the quarter

Backlog (X_1): Number of website orders in backlog at the close of the quarter

Team (X_2): Team Number 1 to 13

Experience (X_3): Number of months team has been together

Process.Change (X_4): A change in the website development process occurred during the second quarter of 2002: 1 if quarter 2 or 3 of 2002; 0 otherwise

Year (X_5): 2001 or 2002

Quarter (X_6): 1,2,3,or4

Use X_1, X_2, X_3, and X_4 to predict Y. Develop a best subset linear regression model for predicting Y. Justify your choice of model. Assess your model's ability to predict and discuss its use as a tool for management decisions.

Based on all Adjusted R^2, AIC, BIC, and c_p. The best model is below

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.185	0.3651	8.726	5.7e-10
Price	-0.3527	0.1574	-2.241	0.0321
Discount.Price	0.3991	0.05125	7.787	6.995e-09
Promotion	0.118	0.05149	2.292	0.0286

Table 2: Fitting linear model: Market.Share ~ Price + Discount.Price + Promotion

Observations	Residual Std. Error	R^2	Adjusted R^2
36	0.1498	0.7065	0.679

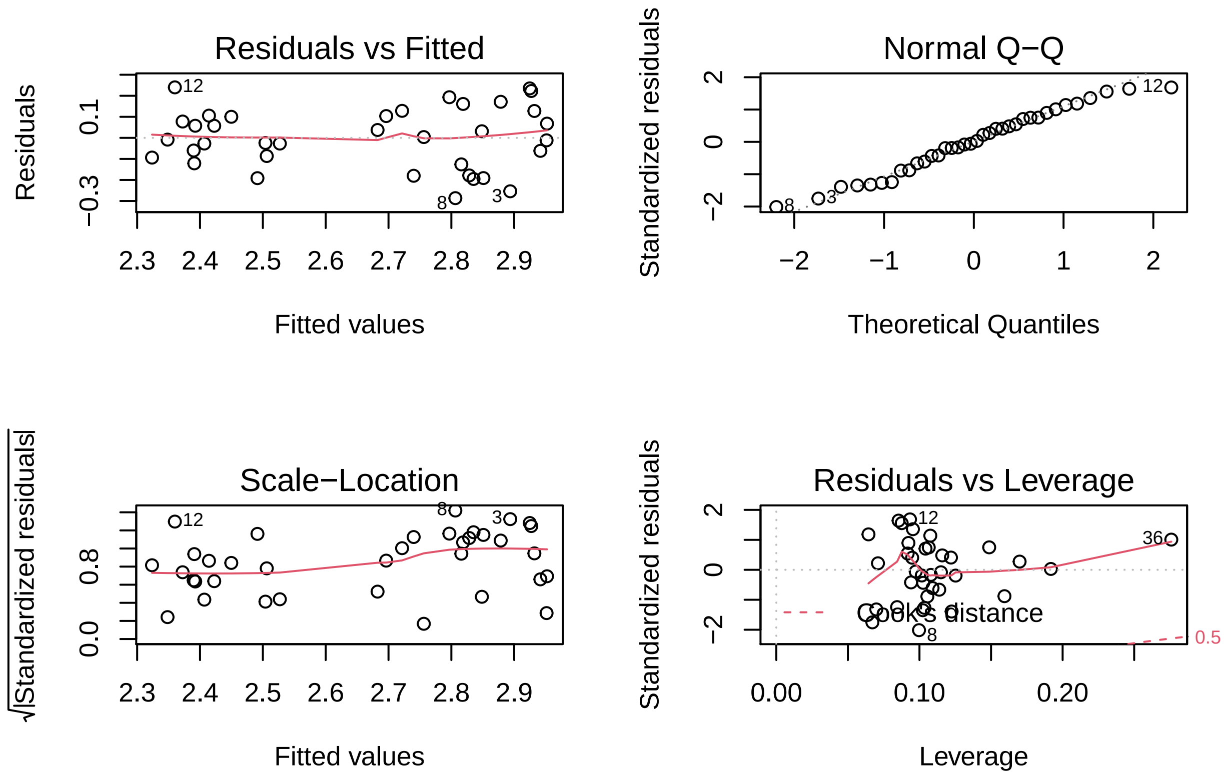
All variables are significant. R^2 is 67%. Lets check for the auto correlation.

Table 3: Durbin-Watson test: `f.m2`

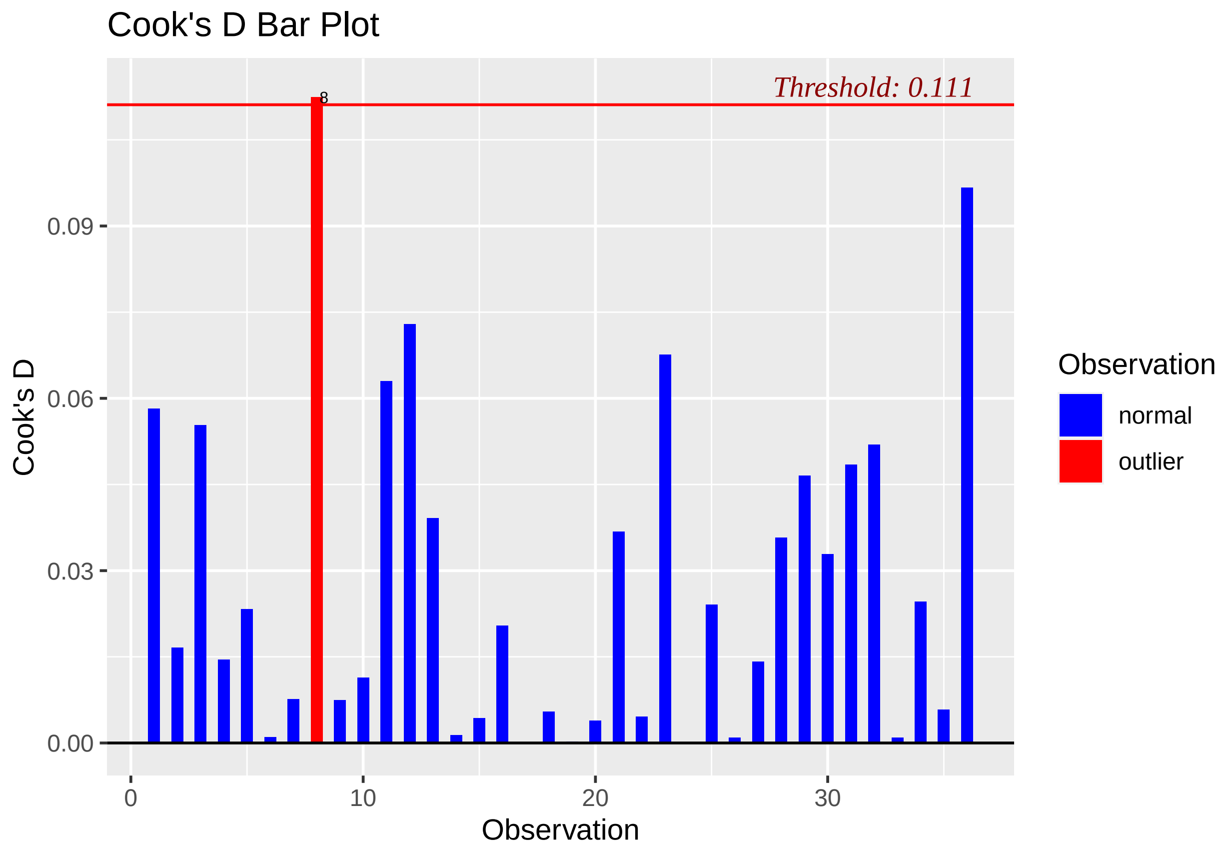
Test statistic	P value	Alternative hypothesis
1.851	0.3484	true autocorrelation is greater than 0

No auto correlation persist in the data.

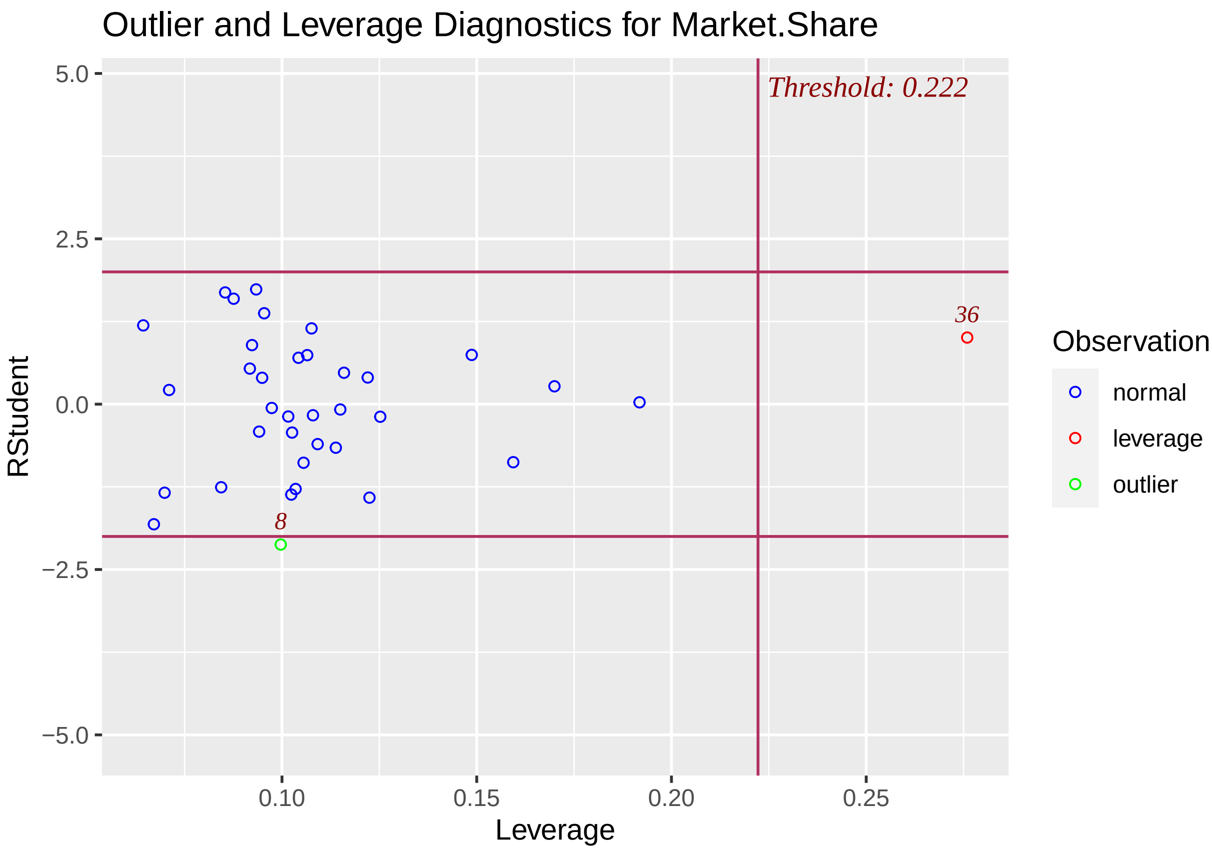
```r
par(mfrow=c(2,2))
plot(f.m2)
```



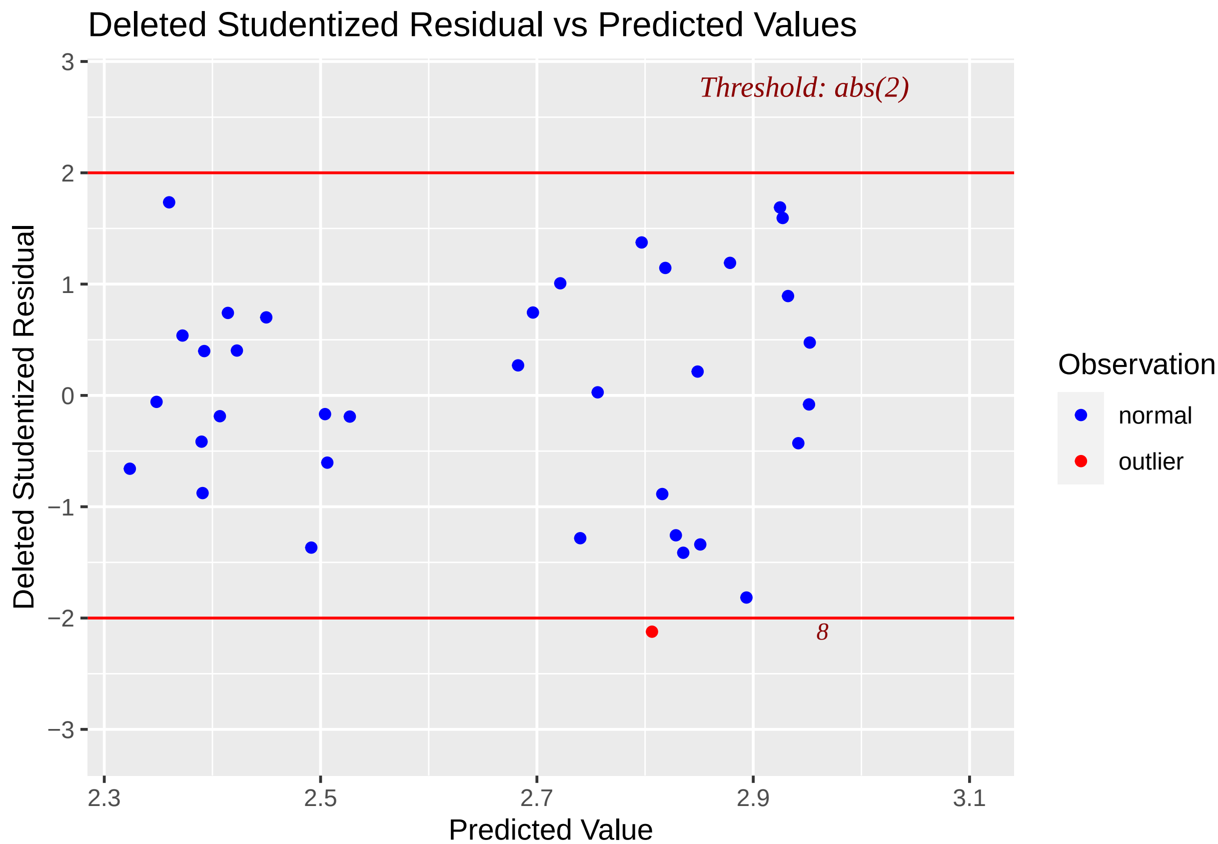

```r
library(olsrr)
ols_plot_cooksd_bar(f.m2)
```




```
ols_plot_resid_lev(f.m2)
```




```
ols_plot_resid_stud_fit(f.m2)
```



From the graphs, all assumptions are met.

Question 2

Use the "Final Exam Fall 2020 Question 2.csv" data set. Residential sales that occurred during the year 2002 were available from a city in the midwest. Data on 522 arms-length transactions include following variables: sales price, finished square feet,

number of bedrooms,

number of bathrooms,

air conditioning (1 if yes; 0 otherwise)

garage size

pool (1 if yes; 0 otherwise),

year built,

quality (3 different qualities),

style (there are 7 different styles, 1 to 7),

lot size,

year built,

a-) Use "set.seed(300)" to create development sample (70% of the data) and hold-out sample (30% of the data).

b-) Use all variables to predict the sales price on the development sample. Do we need to transfer Sales Price? Transform the sales price and refit the model.

c-) Use stepwise (both ways) model selection to select the best model for predicting transformed sales price on the development sample.**Ensure that all variables are significant, use $\alpha = 0.05$. Justify your choice of model. Check the appropriate model assumptions visually from the graphs.**

d-) Use regression Tree to predict the sales price on the development sample. Comment on the model performance.

e-) Use Neuron Network approach to predict the sales price on the development sample.Comment on the model performance.

f-) Score all models on hold-out sample. Compare the SSEs, R^2 and select the best model.

a-)See below.

```
Q2.Dat <- read.csv("/cloud/project/Final Exam Fall 2020 Question 2.csv")

Q2.Dat<-dummy_cols(Q2.Dat, select_columns = 'Style')
Q2.Dat<-dummy_cols(Q2.Dat, select_columns = 'Quality')
set.seed(994)
IND=sample(c(1:522),300)
Q2.Dev<-Q2.Dat[IND,]
Q2.hold<-Q2.Dat[-IND,]
```

b-) Log transformation is needed. Full

```
library(leaps)
library(olsrr)
m.q2<-lm(Sales.price~Finished.square.feet+Number.of.bedrooms+Number.of.bathrooms+Air.conditioning+Garage
```

The model summary table is below: R^2 is 85% and there are seven variables that are not significant. QQ plot indicates that the data needs to be transformed. Box Plot indicates that log transformation is needed.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-2040773	529119	-3.857	0.0001423
Finished.square.feet	100.7	9.231	10.91	2.392e-23
Number.of.bedrooms	2491	4174	0.5967	0.5512
Number.of.bathrooms	3503	5315	0.6591	0.5104
Air.conditioning	1589	10352	0.1535	0.8781
Garage.size	15447	6218	2.484	0.01356
Pool	22555	13051	1.728	0.08504
Year.built	1100	269.1	4.086	5.736e-05
Quality__2	-146345	12178	-12.02	3.884e-27
Quality__3	-156187	16730	-9.336	3.115e-18
Style__2	-27447	11528	-2.381	0.01793
Style__3	-16147	10164	-1.589	0.1133
Style__4	33912	24543	1.382	0.1681
Style__5	-51761	18850	-2.746	0.006421
Style__6	-12653	21531	-0.5877	0.5572
Style__7	-40869	10555	-3.872	0.0001341
Lot.size	1.031	0.2859	3.606	0.0003678
Adjacent.to.highway	-48686	22416	-2.172	0.0307

Table 5: Fitting linear model: Sales.price ~ Finished.square.feet + Number.of.bedrooms + Number.of.bathrooms + Air.conditioning + Garage.size + Pool + Year.built + Quality_2 + Quality_3 + Style_2 + Style_3 + Style_4 + Style_5 + Style_6 + Style_7 + Lot.size + Adjacent.to.highway

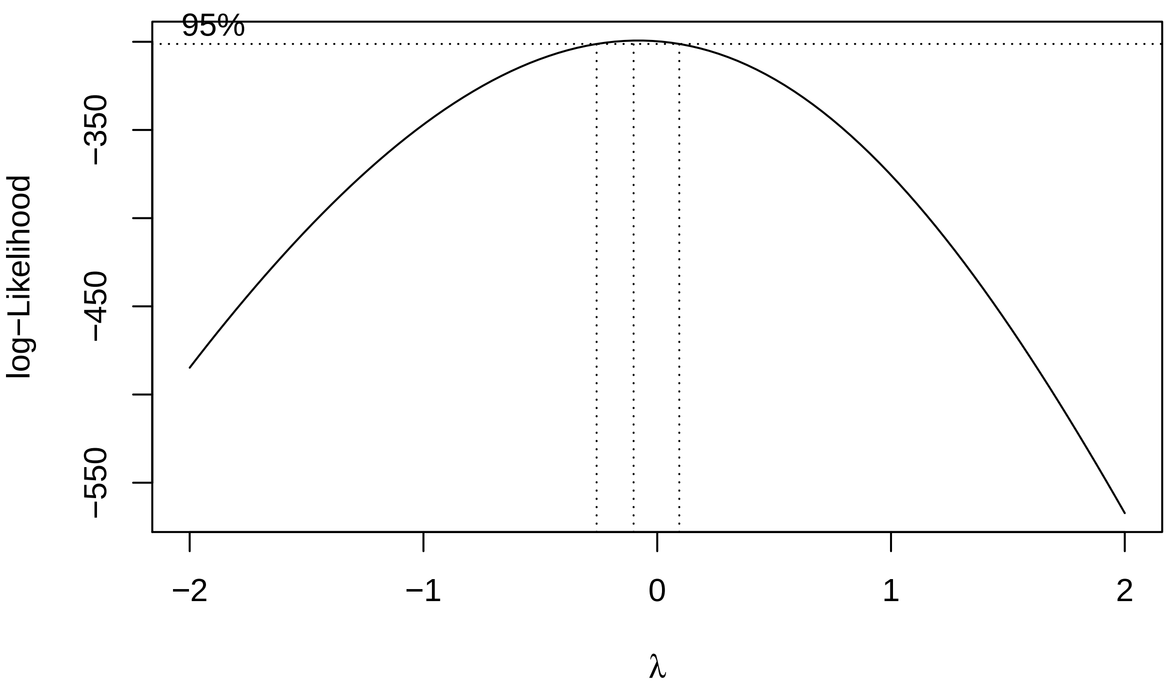
Observations	Residual Std. Error	R^2	Adjusted R^2
300	52807	0.8652	0.8571

```
library(MASS)
```

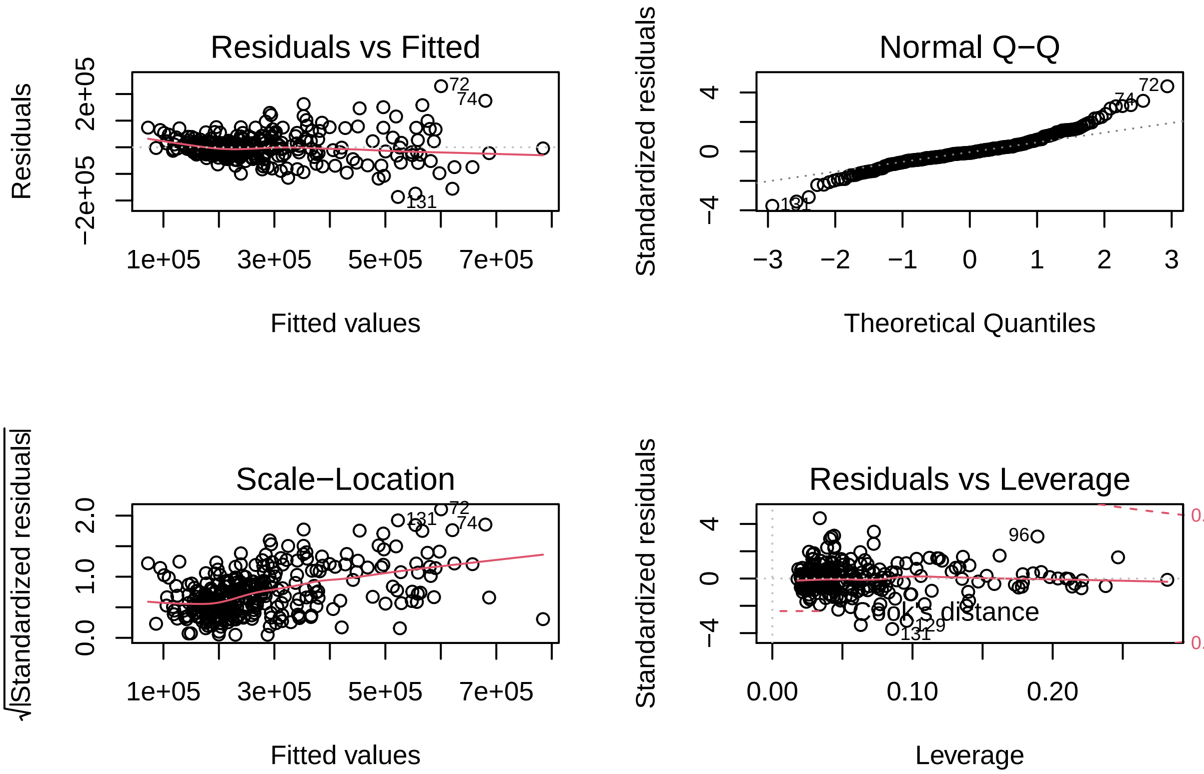
```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:olsrr':
##
##     cement
```

```
boxcox(m.q2,lamda=seq(-2,2,0.1))
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'lamda' will be disregarded
```




```
par(mfrow=c(2,2))
plot(m.q2)
```



c-)

```
m.q21<-lm(log(Sales.price)~Finished.square.feet+Number.of.bedrooms+Number.of.bathrooms+Air.conditioning

k2<-ols_step_both_p(m.q21,prem=0.05,details=FALSE)
k2$model
```

```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Coefficients:
##          (Intercept)  Finished.square.feet            Year.built
##            3.388e+00             2.963e-04             4.265e-03
##             Lot.size               Style_7           Garage.size
##            3.803e-06            -7.417e-02             4.789e-02
##            Quality_3             Quality_2                  Pool
##           -4.418e-01            -3.095e-01             1.182e-01
##              Style_4    Number.of.bedrooms               Style_2
##            1.693e-01             3.625e-02            -7.310e-02
```

```
m.q2.f<-lm(log(Sales.price)~Finished.square.feet+Year.built+Lot.size+Style_7+Garage.size+Quality_3+Quali
```

Based on the stepwise, the best model is below:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.388	1.545	2.192	0.02916
Finished.square.feet	0.0002963	2.588e-05	11.45	2.91e-25
Year.built	0.004265	0.0007818	5.456	1.052e-07

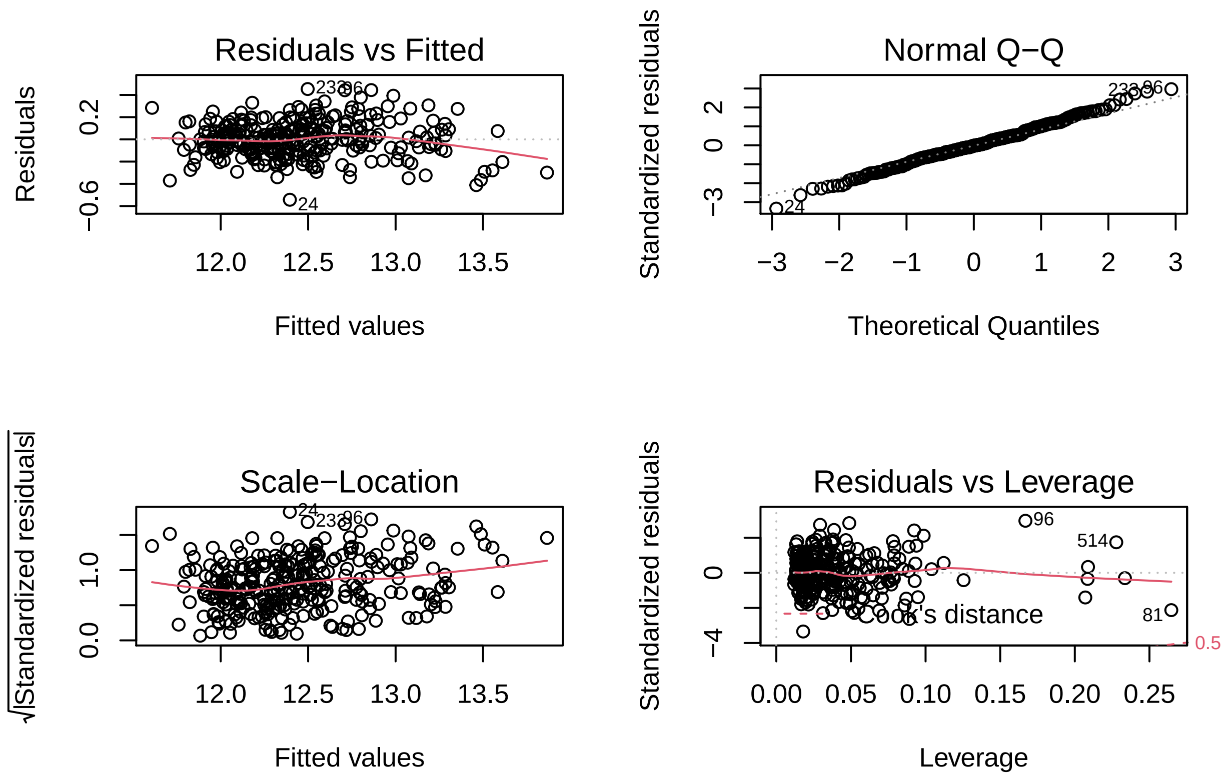
	Estimate	Std. Error	t value	Pr(>|t|)
Lot.size	3.803e-06	8.595e-07	4.425	1.37e-05
Style_7	-0.07417	0.02965	-2.501	0.01294
Garage.size	0.04789	0.01897	2.525	0.01211
Quality_3	-0.4418	0.05039	-8.767	1.628e-16
Quality_2	-0.3095	0.03699	-8.368	2.573e-15
Pool	0.1182	0.04021	2.94	0.003543
Style_4	0.1693	0.07547	2.244	0.0256
Number.of.bedrooms	0.03625	0.01184	3.062	0.00241
Style_2	-0.0731	0.03385	-2.159	0.03164

Table 7: Fitting linear model: log(Sales.price) ~ Finished.square.feet + Year.built + Lot.size + Style_7 + Garage.size + Quality_3 + Quality_2 + Pool + Style_4 + Number.of.bedrooms + Style_2

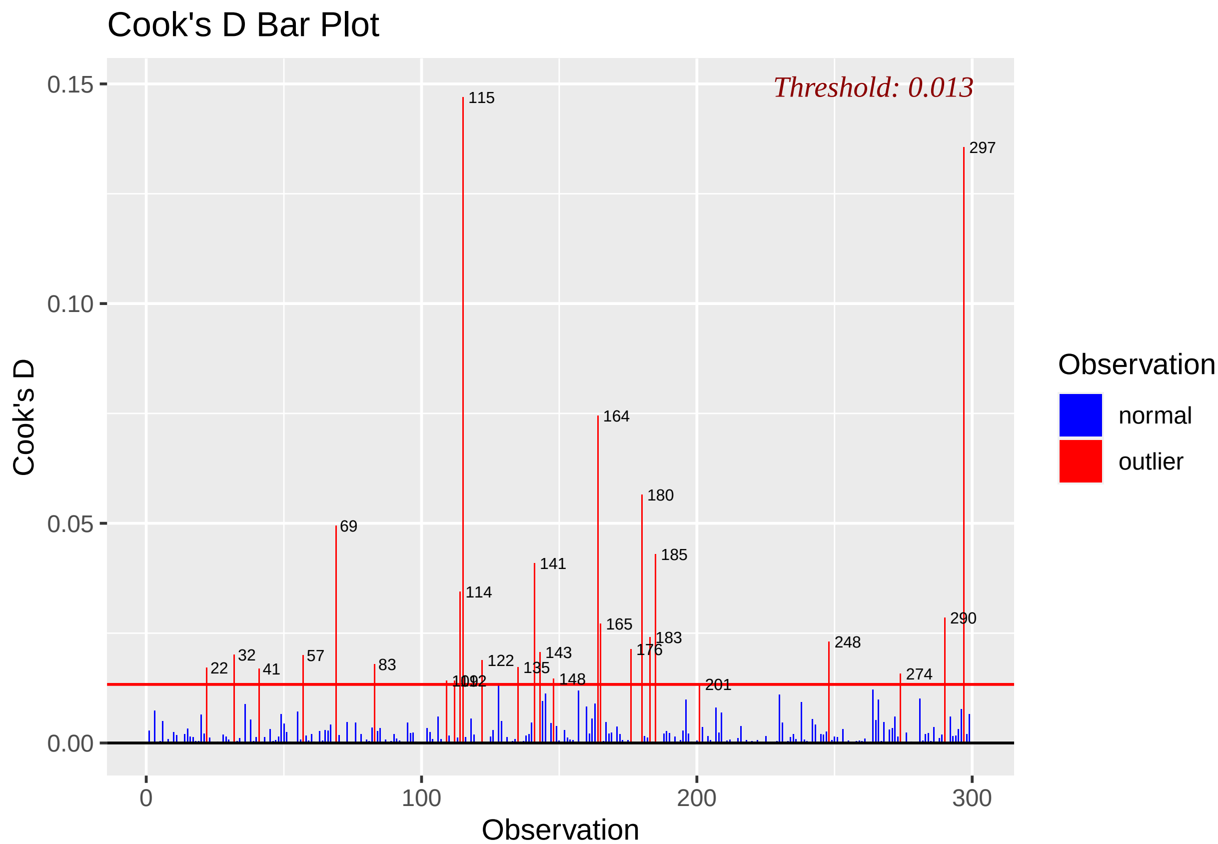
Observations	Residual Std. Error	R^2	Adjusted R^2
300	0.1639	0.8648	0.8597

All variables are significant. R^2 is 86%. QQ plot shows that the data is normal. However, observation 115 is an leverage point based and there are several outliers.

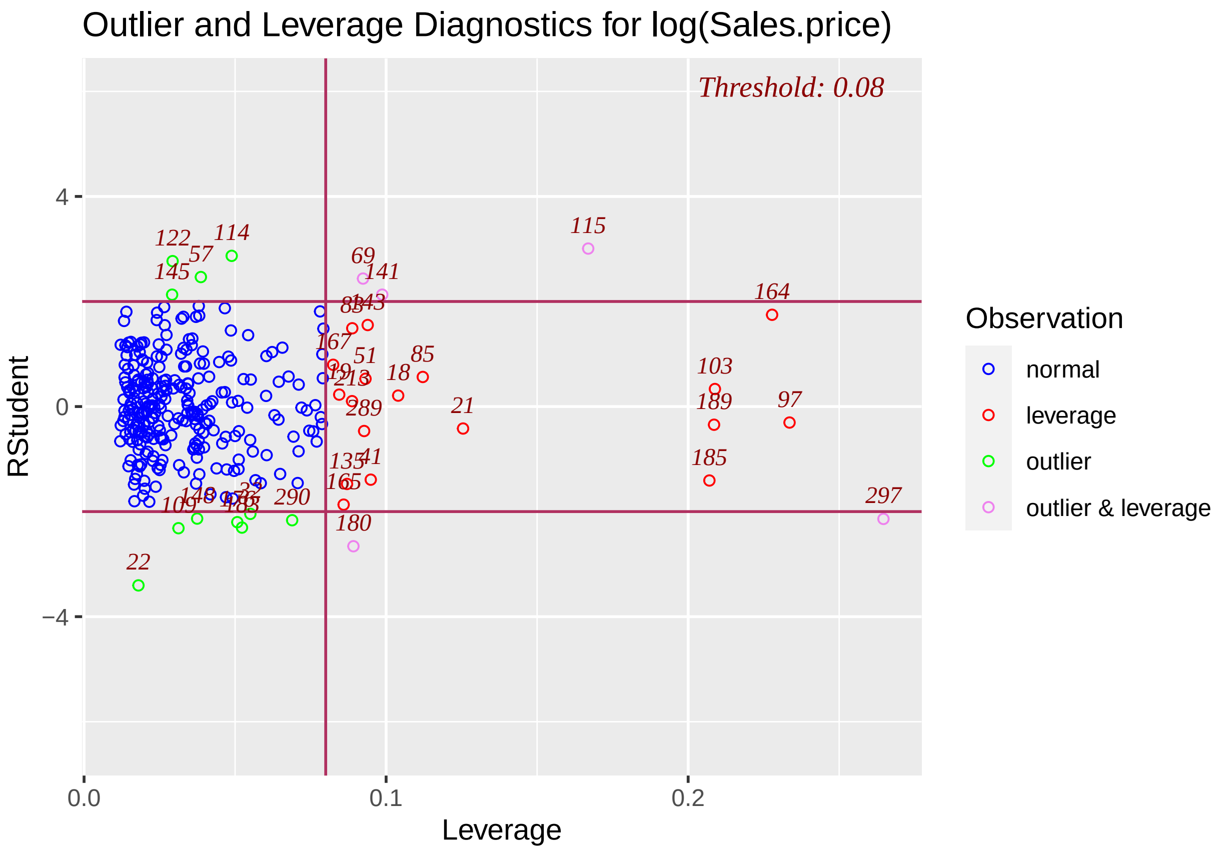
```r
par(mfrow=c(2,2))
plot(m.q2.f)
```



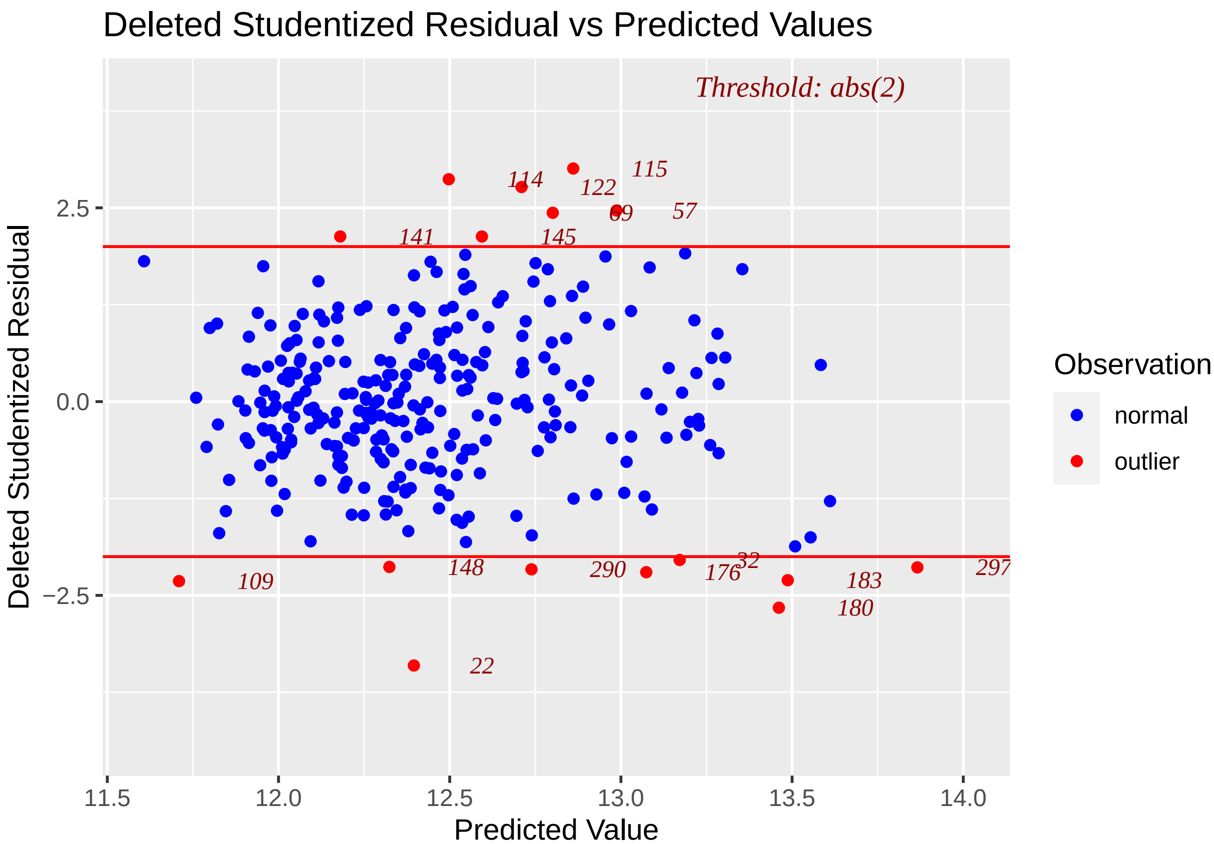

```r
library(olsrr)
ols_plot_cooksd_bar(m.q2.f)
```



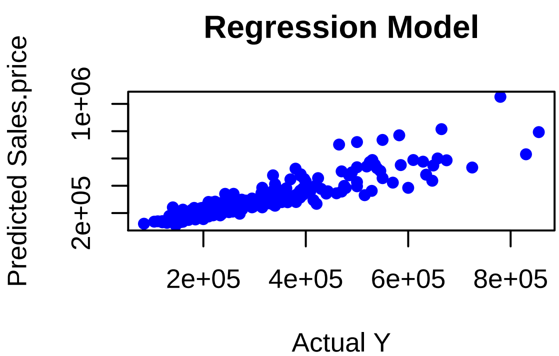

```
ols_plot_resid_lev(m.q2.f)
```




```r
ols_plot_resid_stud_fit(m.q2.f)
```

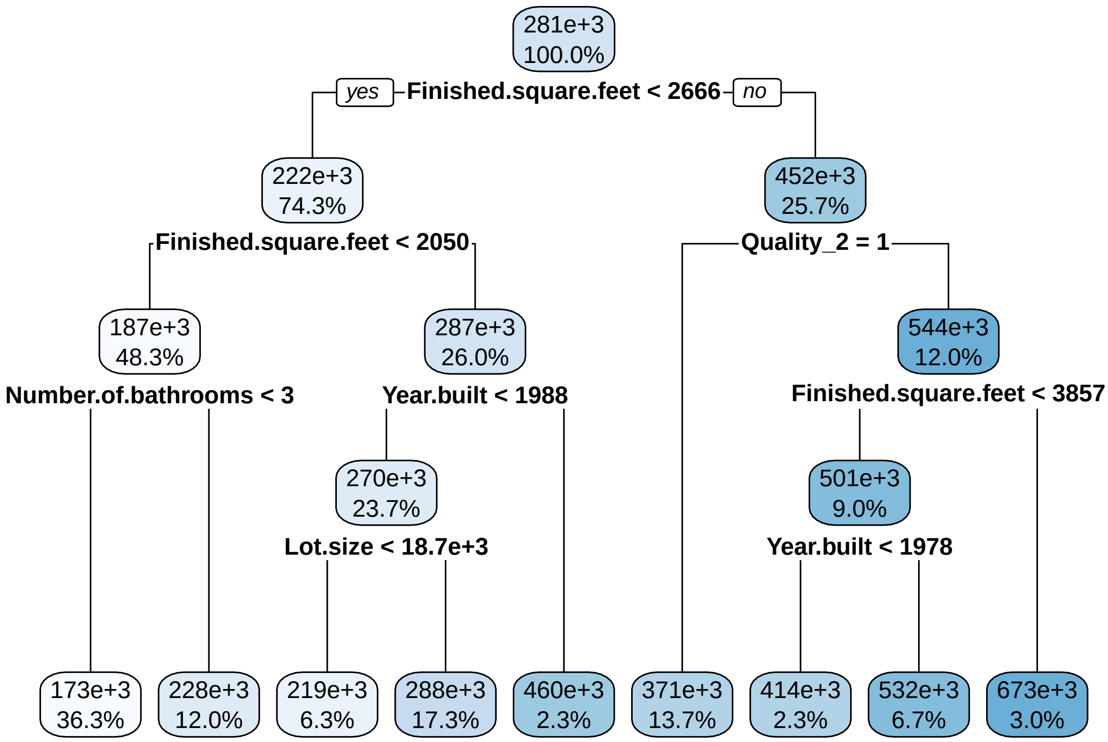



```
plot(Q2.Dev$Sales.price,exp(m.q2.f$fitted.values), col='blue', pch=16, ylab= "Predicted Sales.price", xl
```



d-)Please see below,

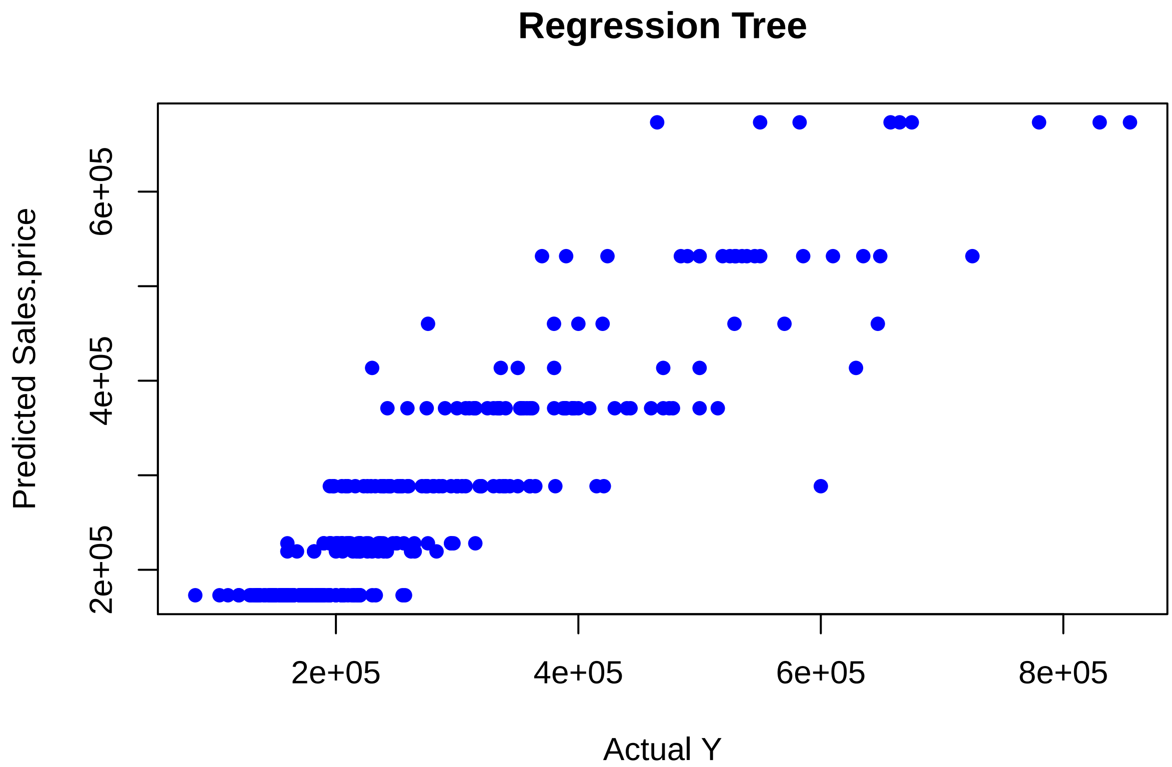
```
library(rpart)
q2.tr<-rpart(Sales.price~Finished.square.feet+Number.of.bedrooms+Number.of.bathrooms+Air.conditioning+Ga
library(rpart.plot)
par(mfrow=c(1,1))
rpart.plot(q2.tr,digits = 3)
```




```
SSE.Tree.Dev<-sum((predict(q2.tr)-Q2.Dev$Sales.price)^2)
SSE.Tree.Dev
```

```
## [1] 1.064733e+12
```

```
p.rpart<-predict(q2.tr,Q2.Dev)
plot(Q2.Dev$Sales.price,p.rpart, col='blue', pch=16, ylab= "Predicted Sales.price", xlab= "Actual Y",ma
```



e-) Please see below,

```
install.packages("neuralnet")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
library(neuralnet)
normalize <- function(x) {return((x - min(x)) / (max(x) - min(x)))}
scaled.Q2.Dat <- as.data.frame(lapply(Q2.Dat, normalize))
scaled.Q2.Dev<- scaled.Q2.Dat[IND,]
scaled.Q2.Hold<- scaled.Q2.Dat[-IND,]

NN = neuralnet(Sales.price~Finished.square.feet+Number.of.bedrooms+Number.of.bathrooms+Air.conditioning

plot(NN)
predict_testNN= compute(NN, scaled.Q2.Dev[,-c(1,13,20)])
#we need to transform it back to orginal scale
predict_testNN1 = (predict_testNN$net.result* (max(Q2.Dat$Sales.price) -min(Q2.Dat$Sales.price))) + min
plot(Q2.Dev$Sales.price, predict_testNN1, col='blue', pch=16, ylab= "Predicted Sales.price", xlab= "Act
```

f-) see below

```
library(glmnet)
```

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

```
## Loaded glmnet 4.0-2
```

```
x <- model.matrix(Sales.price~., Q2.Dev)[,-c(1,9,10,13,20)]
xnew<-model.matrix(Sales.price~., Q2.hold)[,-c(1,9,10,13,20)]
y <- Q2.Dev$Sales.price
```

```r
EnetMod <- glmnet(x,y, alpha=0.5, nlambda=100,lambda.min.ratio=0.0001)
CvElasticnetMod <- cv.glmnet(x, y,alpha=0.5,nlambda=100,lambda.min.ratio=0.0001)
best.lambda.enet <- CvElasticnetMod$lambda.min
coef(CvElasticnetMod, s = "lambda.min")
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                                  1
## (Intercept)          -2.051060e+06
## Finished.square.feet  1.009280e+02
## Number.of.bedrooms    1.029699e+03
## Number.of.bathrooms   4.153118e+03
## Air.conditioning      1.632205e+03
## Garage.size           1.639802e+04
## Pool                  2.017147e+04
## Year.built            1.099804e+03
## Lot.size              1.003212e+00
## Adjacent.to.highway  -4.062501e+04
## Style_2              -2.187622e+04
## Style_3              -1.176956e+04
## Style_4               3.005483e+04
## Style_5              -4.388727e+04
## Style_6              -6.619667e+03
## Style_7              -3.467349e+04
## Quality_2            -1.379790e+05
## Quality_3            -1.445683e+05
```

g-) Elastic Net is the best model. it has lowest SSE and highest R^2.

```r
#Measuring performance with the SSE
SSE <- function(actual, predicted) {sum((actual - predicted)^2)}

#Measuring performance with the RSquare
R2 <- function(actual, predicted) {sum((actual - predicted)^2)/((length(actual)-1)*var(actual))}

#Regression
reg.predict<-exp(predict(m.q2.f,Q2.hold))
#Tree
tree.predict<-predict(q2.tr,Q2.hold)
#NN
nn.predict<-compute(NN, scaled.Q2.Hold)
nn.predict1 = (nn.predict$net.result* (max(Q2.Dat$Sales.price) -min(Q2.Dat$Sales.price))) + min(Q2.Dat$S
#Elastic Net
enet.predict <- predict(CvElasticnetMod , s = best.lambda.enet, newx = xnew)

#SSEs

cbind(REG=SSE(Q2.hold$Sales.price,reg.predict),
Tree=SSE(Q2.hold$Sales.price,tree.predict),
NN=SSE(Q2.hold$Sales.price,nn.predict1),ENET=SSE(Q2.hold$Sales.price,enet.predict))
```

```
##                REG         Tree           NN         ENET
## [1,] 1.439502e+12 1.217865e+12 1.050905e+12 911323267016
```

```r
#R Squares
cbind(Reg=1-R2(Q2.hold$Sales,reg.predict),Tree=1-R2(Q2.hold$Sales,tree.predict),NN=1-R2(Q2.hold$Sales,nn
```

```
##             Reg      Tree        NN      ENET
## [1,] 0.6464859 0.7009156 0.7419179 0.7761965
```

Question 3

Refer to the data in Question 2. Create a binary response variable Y, called high quality, by letting Y=1 if quality variable equals to 1 otherwise 0.

a-) Fit a model to predict Y, ensure that all variables are significant by using the backward elimination to decide which predictor variables can be dropped from the regression model. Use $\alpha = 0.05$.

b-) Conduct the Hosmer-Lemeshow goodness of fit test for the appropriateness of the logistic regression function by forming five groups. State the alternatives, decision rule, and conclusion.

c-) What is the estimated probability that houses below have good quality? Calculate the 95% confidence interval. Variables names are shorten to fit all data neatly below

a-) Built in function performed the backward elimination. However, there are variables on the model that are not significant. We will eliminate them one at a time, based on the highest p value.

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::compute() masks neuralnet::compute()
## x tidyr::expand()  masks Matrix::expand()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x tidyr::pack()    masks Matrix::pack()
## x dplyr::select()  masks MASS::select()
## x tidyr::unpack()  masks Matrix::unpack()
```

```r
Q3.Dat <- data.frame(read.csv("/cloud/project/Final Exam Fall 2020 Question 2.csv"))
Q3.Dat$Y=I(Q3.Dat$Quality==1)*1
Q3.Dat<-dummy_cols(Q3.Dat, select_columns='Style')

f.q3<-glm(Y~Sales.price+Finished.square.feet+Number.of.bedrooms+Number.of.bathrooms+Air.conditioning+Ga
```

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

```r
t0<-step(f.q3,direction="backward",trace=0)
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```r
summary(t0)
```

```
##
## Call:
## glm(formula = Y ~ Sales.price + Finished.square.feet + Number.of.bedrooms +
##     Number.of.bathrooms + Air.conditioning + Garage.size + Style_6 +
##     Style_7 + Lot.size, family = binomial, data = Q3.Dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3004  -0.1069  -0.0396   0.0000   2.5137
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.999e+01  1.428e+03  -0.021   0.9832
## Sales.price           2.090e-05  3.820e-06   5.471 4.47e-08 ***
## Finished.square.feet  1.020e-03  6.665e-04   1.531   0.1259
## Number.of.bedrooms   -5.660e-01  2.862e-01  -1.977   0.0480 *
## Number.of.bathrooms   6.286e-01  3.327e-01   1.889   0.0589 .
## Air.conditioning      1.661e+01  1.428e+03   0.012   0.9907
## Garage.size           9.067e-01  4.684e-01   1.936   0.0529 .
## Style_6              -1.958e+01  3.162e+03  -0.006   0.9951
## Style_7              -9.438e-01  6.748e-01  -1.399   0.1619
## Lot.size             -3.473e-05  2.223e-05  -1.562   0.1182
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 403.92  on 521  degrees of freedom
## Residual deviance: 104.74  on 512  degrees of freedom
## AIC: 124.74
##
```

```
## Number of Fisher Scoring iterations: 19
```

```r
t1<-glm(formula = Y ~ Sales.price + Finished.square.feet + Number.of.bedrooms +
    Number.of.bathrooms + Air.conditioning + Garage.size + Style_6 +
    Style_7 + Lot.size, family = binomial, data = Q3.Dat)
```

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

```r
summary(t1)
```

```
##
## Call:
## glm(formula = Y ~ Sales.price + Finished.square.feet + Number.of.bedrooms +
##     Number.of.bathrooms + Air.conditioning + Garage.size + Style_6 +
##     Style_7 + Lot.size, family = binomial, data = Q3.Dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3004  -0.1069  -0.0396   0.0000   2.5137
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.999e+01  1.428e+03  -0.021   0.9832
## Sales.price           2.090e-05  3.820e-06   5.471 4.47e-08 ***
## Finished.square.feet  1.020e-03  6.665e-04   1.531   0.1259
## Number.of.bedrooms   -5.660e-01  2.862e-01  -1.977   0.0480 *
## Number.of.bathrooms   6.286e-01  3.327e-01   1.889   0.0589 .
## Air.conditioning      1.661e+01  1.428e+03   0.012   0.9907
## Garage.size           9.067e-01  4.684e-01   1.936   0.0529 .
## Style_6              -1.958e+01  3.162e+03  -0.006   0.9951
## Style_7              -9.438e-01  6.748e-01  -1.399   0.1619
## Lot.size             -3.473e-05  2.223e-05  -1.562   0.1182
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 403.92  on 521  degrees of freedom
## Residual deviance: 104.74  on 512  degrees of freedom
## AIC: 124.74
##
## Number of Fisher Scoring iterations: 19
```

```r
#dropping Style_6 pvalue is 0.9951
t2<-glm(formula = Y ~ Sales.price + Finished.square.feet + Number.of.bedrooms +
    Number.of.bathrooms + Air.conditioning + Garage.size  +
    Style_7 + Lot.size, family = binomial, data = Q3.Dat)
summary(t2)
```

```
##
## Call:
## glm(formula = Y ~ Sales.price + Finished.square.feet + Number.of.bedrooms +
##     Number.of.bathrooms + Air.conditioning + Garage.size + Style_7 +
##     Lot.size, family = binomial, data = Q3.Dat)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -3.2739  -0.1094  -0.0441   0.0000   2.6072
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.972e+01  1.447e+03  -0.021   0.9836
## Sales.price           2.011e-05  3.525e-06   5.706 1.16e-08 ***
## Finished.square.feet  1.004e-03  6.519e-04   1.541   0.1234
## Number.of.bedrooms   -5.816e-01  2.859e-01  -2.034   0.0420 *
## Number.of.bathrooms   4.555e-01  3.271e-01   1.393   0.1637
## Air.conditioning      1.664e+01  1.447e+03   0.011   0.9908
## Garage.size           1.018e+00  4.621e-01   2.203   0.0276 *
## Style_7              -5.009e-01  6.321e-01  -0.792   0.4281
## Lot.size             -2.941e-05  2.112e-05  -1.392   0.1639
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 403.92  on 521  degrees of freedom
## Residual deviance: 111.04  on 513  degrees of freedom
## AIC: 129.04
##
## Number of Fisher Scoring iterations: 19
```

```r
#dropping Style_7 pvalue is 0.4281
t3<-glm(formula = Y ~ Sales.price + Finished.square.feet + Number.of.bedrooms +
    Number.of.bathrooms + Air.conditioning + Garage.size  +
     Lot.size, family = binomial, data = Q3.Dat)
summary(t3)
```

```
##
## Call:
## glm(formula = Y ~ Sales.price + Finished.square.feet + Number.of.bedrooms +
##     Number.of.bathrooms + Air.conditioning + Garage.size + Lot.size,
##     family = binomial, data = Q3.Dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3214  -0.1082  -0.0433   0.0000   2.6703
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.951e+01  1.446e+03  -0.020   0.9837
## Sales.price           2.058e-05  3.457e-06   5.953 2.63e-09 ***
## Finished.square.feet  7.982e-04  5.911e-04   1.350   0.1769
## Number.of.bedrooms   -5.754e-01  2.858e-01  -2.013   0.0441 *
## Number.of.bathrooms   4.156e-01  3.188e-01   1.304   0.1923
## Air.conditioning      1.678e+01  1.446e+03   0.012   0.9907
## Garage.size           9.569e-01  4.544e-01   2.106   0.0352 *
## Lot.size             -2.675e-05  2.055e-05  -1.302   0.1930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 403.92  on 521  degrees of freedom
## Residual deviance: 111.67  on 514  degrees of freedom
## AIC: 127.67
##
## Number of Fisher Scoring iterations: 19
```

```
#dropping Air.conditioning  pvalue is 0.9907
t4<-glm(formula = Y ~ Sales.price + Finished.square.feet + Number.of.bedrooms +
    Number.of.bathrooms + Garage.size  +
     Lot.size, family = binomial, data = Q3.Dat)
summary(t4)
```

```
##
## Call:
## glm(formula = Y ~ Sales.price + Finished.square.feet + Number.of.bedrooms +
##     Number.of.bathrooms + Garage.size + Lot.size, family = binomial,
##     data = Q3.Dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2746  -0.1100  -0.0456  -0.0248   2.7013
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.289e+01  1.970e+00  -6.543 6.05e-11 ***
## Sales.price           2.088e-05  3.347e-06   6.238 4.43e-10 ***
## Finished.square.feet  6.517e-04  5.542e-04   1.176   0.2396
## Number.of.bedrooms   -4.882e-01  2.769e-01  -1.763   0.0779 .
## Number.of.bathrooms   3.827e-01  3.056e-01   1.252   0.2104
## Garage.size           1.046e+00  4.450e-01   2.351   0.0187 *
## Lot.size             -3.009e-05  1.992e-05  -1.511   0.1309
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 403.92  on 521  degrees of freedom
## Residual deviance: 116.23  on 515  degrees of freedom
## AIC: 130.23
##
## Number of Fisher Scoring iterations: 8
```

```
#dropping inished.square.feet pvalue is  0.2396
t5<-glm(formula = Y ~ Sales.price + Number.of.bedrooms +
     Number.of.bathrooms + Garage.size  +
     Lot.size, family = binomial, data = Q3.Dat)
summary(t5)
```

```
##
## Call:
## glm(formula = Y ~ Sales.price + Number.of.bedrooms + Number.of.bathrooms +
##     Garage.size + Lot.size, family = binomial, data = Q3.Dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -3.2262  -0.1225  -0.0502  -0.0255   2.6691
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.245e+01  1.879e+00  -6.624 3.50e-11 ***
## Sales.price          2.243e-05  3.169e-06   7.079 1.45e-12 ***
## Number.of.bedrooms  -3.390e-01  2.418e-01  -1.402   0.1610
## Number.of.bathrooms  4.706e-01  2.893e-01   1.627   0.1038
## Garage.size          1.114e+00  4.329e-01   2.572   0.0101 *
## Lot.size            -3.811e-05  1.862e-05  -2.047   0.0407 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 403.92  on 521  degrees of freedom
## Residual deviance: 117.63  on 516  degrees of freedom
## AIC: 129.63
##
## Number of Fisher Scoring iterations: 8
```

```r
#dropping Number.of.bedrooms  pvalue is 0.16
t6<-glm(formula = Y ~ Sales.price +
     Number.of.bathrooms + Garage.size  +
     Lot.size, family = binomial, data = Q3.Dat)
summary(t6)
```

```
##
## Call:
## glm(formula = Y ~ Sales.price + Number.of.bathrooms + Garage.size +
##     Lot.size, family = binomial, data = Q3.Dat)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.14618  -0.12996  -0.05383  -0.02552   2.62225
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.295e+01  1.831e+00  -7.072 1.52e-12 ***
## Sales.price          2.176e-05  3.001e-06   7.252 4.11e-13 ***
## Number.of.bathrooms  2.865e-01  2.589e-01   1.107  0.26841
## Garage.size          1.136e+00  4.305e-01   2.638  0.00835 **
## Lot.size            -3.764e-05  1.830e-05  -2.057  0.03969 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 403.92  on 521  degrees of freedom
## Residual deviance: 119.63  on 517  degrees of freedom
## AIC: 129.63
##
## Number of Fisher Scoring iterations: 8
```

```r
#dropping Number.of.bathrooms  pvalue is 0.26
t7<-glm(formula = Y ~ Sales.price + Garage.size+Lot.size, family = binomial, data = Q3.Dat)
summary(t7)
```

```
##
## Call:
## glm(formula = Y ~ Sales.price + Garage.size + Lot.size, family = binomial,
##     data = Q3.Dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1660  -0.1395  -0.0606  -0.0312   2.5725
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.213e+01  1.581e+00  -7.672  1.7e-14 ***
## Sales.price  2.252e-05  2.906e-06   7.748  9.3e-15 ***
## Garage.size  1.123e+00  4.245e-01   2.646  0.00813 **
## Lot.size    -3.998e-05  1.807e-05  -2.212  0.02694 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 403.92  on 521  degrees of freedom
## Residual deviance: 120.79  on 518  degrees of freedom
## AIC: 128.79
##
## Number of Fisher Scoring iterations: 8
```

b-) The fit is good.

```r
library(ResourceSelection)
```

```
## ResourceSelection 0.3-5   2019-07-22
```

```r
hoslem.test(t7$y,fitted(t7),g=5)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  t7$y, fitted(t7)
## X-squared = 0.68924, df = 3, p-value = 0.8757
```

c-) Please see below

```r
test.dat<-data.frame(matrix(c(559000,2791,3,4,1,3,0,1992,1,30595,0
,535000,3381,5,4,1,3,0,1988,7,23172,0
,525000,3459,5,4,1,2,0,1978,5,35351,0),byrow=T,nrow=3,ncol=11))
dimnames(test.dat)[[2]]<-c("Sales.price","Finished.square.feet","Number.of.bedrooms","Number.of.bathroo
predict(t7,test.dat,type="response")
```

```
##         1         2         3
## 0.9310401 0.9136546 0.6279717
```

Question 4

Use ships data sets in the MASS package. Copy and paste and following code "library(MASS);data(ships,package ="MASS")".

Data contains the number of wave damage incidents and aggregate months of service for different types of ships broken down by year of construction and period of operation.

a-) All variables and model are significant.

```
library(MASS);data(ships,package = "MASS")
f.m4<-glm(incidents~.,data=ships,family=poisson)
summary(f.m4)
```

```
##
## Call:
## glm(formula = incidents ~ ., family = poisson, data = ships)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.1013  -1.9648  -0.5380   0.9899   4.6212
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.706e+00  1.221e+00  -4.673 2.96e-06 ***
## typeB        8.135e-01  2.023e-01   4.021 5.79e-05 ***
## typeC       -1.205e+00  3.275e-01  -3.679 0.000234 ***
## typeD       -8.595e-01  2.875e-01  -2.989 0.002795 **
## typeE       -2.226e-01  2.348e-01  -0.948 0.343173
## year         4.519e-02  1.341e-02   3.370 0.000752 ***
## period       6.055e-02  8.945e-03   6.768 1.30e-11 ***
## service      5.970e-05  7.016e-06   8.509  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 730.25  on 39  degrees of freedom
## Residual deviance: 174.00  on 32  degrees of freedom
## AIC: 287.86
##
## Number of Fisher Scoring iterations: 6
```

```
drop1(f.m4,test="Chi")
```

```
## Single term deletions
##
## Model:
## incidents ~ type + year + period + service
##         Df Deviance    AIC    LRT  Pr(>Chi)
## <none>       174.00 287.86
## type     4   250.24 356.11 76.248 1.085e-15 ***
## year     1   185.75 297.62 11.755 0.0006067 ***
## period   1   225.85 337.72 51.854 5.979e-13 ***
## service  1   250.31 362.18 76.314 < 2.2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(f.m4,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: incidents
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                       39     730.25
## type     4   454.60        35     275.65 < 2.2e-16 ***
## year     1     8.11        34     267.54    0.0044 **
## period   1    17.23        33     250.31 3.313e-05 ***
## service  1    76.31        32     174.00 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

b-)

```r
test.data<-data.frame(type="B",year=60,period=60,service=44882)
predict(f.m4,test.data,type="response")
```

```
##        1
## 62.24901
```

Question 5

Refer the question 2-c and your final model . There are outliers in the model. Build a robust regression model (use the same variables) and compare your regression model and outputs with the model you built in question 2c. (10 Points)

The results do look so much different, infact regression model is performing slightly better.

```r
library(MASS)
rr.huber <- rlm(log(Sales.price)~Finished.square.feet+Year.built+Lot.size+Style_7+Garage.size+Quality_3
summary(rr.huber)
```

```
##
## Call: rlm(formula = log(Sales.price) ~ Finished.square.feet + Year.built +
##     Lot.size + Style_7 + Garage.size + Quality_3 + Quality_2 +
##     Pool + Style_4 + Number.of.bedrooms + Style_2, data = Q2.Dev)
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.532510 -0.096264 -0.004888  0.089295  0.529939
##
## Coefficients:
##                      Value   Std. Error t value
## (Intercept)           2.8937  1.5626     1.8518
## Finished.square.feet  0.0003  0.0000    12.4636
## Year.built            0.0045  0.0008     5.6573
## Lot.size              0.0000  0.0000     3.3199
```

```
## Style_7              -0.0916  0.0300    -3.0536
## Garage.size           0.0489  0.0192     2.5518
## Quality_3            -0.3911  0.0509    -7.6758
## Quality_2            -0.2849  0.0374    -7.6187
## Pool                  0.1169  0.0407     2.8762
## Style_4               0.1547  0.0763     2.0275
## Number.of.bedrooms    0.0406  0.0120     3.3931
## Style_2              -0.0680  0.0342    -1.9871
##
## Residual standard error: 0.1373 on 288 degrees of freedom
```

```
cbind(rr.huber$coefficients,m.q2.f$coefficients)
```

```
##                             [,1]          [,2]
## (Intercept)         2.893692e+00  3.388140e+00
## Finished.square.feet  3.261144e-04  2.962757e-04
## Year.built           4.471772e-03  4.265129e-03
## Lot.size             2.884953e-06  3.803152e-06
## Style_7             -9.155457e-02 -7.416572e-02
## Garage.size          4.893267e-02  4.788747e-02
## Quality_3           -3.910760e-01 -4.417603e-01
## Quality_2           -2.849109e-01 -3.094909e-01
## Pool                 1.169353e-01  1.182377e-01
## Style_4              1.547080e-01  1.693390e-01
## Number.of.bedrooms   4.062443e-02  3.625362e-02
## Style_2             -6.800607e-02 -7.309603e-02
```

```
reg.predict<-exp(predict(m.q2.f,Q2.hold))
rob.predict<-exp(predict(rr.huber,Q2.hold))


#SSEs

cbind(REG=SSE(Q2.hold$Sales.price,reg.predict),
Robust=SSE(Q2.hold$Sales.price,rob.predict))
```

```
##               REG       Robust
## [1,] 1.439502e+12 1.555987e+12
```

```
#R Squares
cbind(Reg=1-R2(Q2.hold$Sales,reg.predict),Robust=1-R2(Q2.hold$Sales,rob.predict))
```

```
##            Reg    Robust
## [1,] 0.6464859 0.6178795
```