CSCI E-106: Section 13:

04/25/2019

Contents

(10.09) Reference Brand Preference Problem 6.5.

In a small-scale experimental study of the relation between degree of brand liking (Y) and moisture content (X_1) and sweetness (X_2) of the product, the following results were obtained from the experiment based on a completely randomized design (data are coded).

*Please use datasets titled **CH06PR05.txt** when applicable*

a Obtain the studentized deleted residuals and identify any outlying Y observations. Use the Bonferroni outlier test procedure with $\alpha = .10$. State the decision rule and conclusion.

Student's Solution Below

```
#Load the data
dfCH10PR09 =read.delim(file="CH06PR05.txt", sep="", header = FALSE)
colnames(dfCH10PR09) =c("BrandLiking", "MoistureContent", "Sweetness")

#find our lmfit
lmCH10PR09 =lm(BrandLiking~MoistureContent+Sweetness, dfCH10PR09)

#calculate our deleted residuals using the rstudent function:
# this Returns the Studentized residuals based on rank-based estimation
deletedResiduals = rstudent(lmCH10PR09)
print(deletedResiduals)
```

```
##           1           2           3           4           5           6
## -0.04085498  0.06128781 -1.36059879  1.38602483 -0.36694571 -0.66490618
##           7           8           9          10          11          12
## -0.76716157  0.50461264  0.46506694 -0.60436295  1.82302030  0.97784298
##          13          14          15          16
## -1.13966417 -2.10272640  1.48973208  0.24572878
```

```
#you can round them to the 3rd to get smaller results
round(deletedResiduals,3)
```

```
##      1      2      3      4      5      6      7      8      9     10
## -0.041  0.061 -1.361  1.386 -0.367 -0.665 -0.767  0.505  0.465 -0.604
##     11     12     13     14     15     16
##  1.823  0.978 -1.140 -2.103  1.490  0.246
```

```r
#longer way to do this without the rstudent function
n = length(dfCH10PR09$BrandLiking)

# Number of regression parameters
p = 3

hii =hatvalues(lmCH10PR09)
ei = lmCH10PR09$residuals
SSE =anova(lmCH10PR09)[3,2]

deletedRes = ei*((n-p-1)/(SSE*(1-hii)-ei^2))^.5
round(deletedRes,3)
```

```
##      1      2      3      4      5      6      7      8      9     10
## -0.041  0.061 -1.361  1.386 -0.367 -0.665 -0.767  0.505  0.465 -0.604
##     11     12     13     14     15     16
##  1.823  0.978 -1.140 -2.103  1.490  0.246
```

```r
#degrees of freedom
df = n-p-1
print(df)
```

```
## [1] 12
```

```r
#t- value
t = qt(1-.10/(2*n), df = n-p-1)
print(t)
```

```
## [1] 3.307783
```

```r
# finding our top 3 residuals
head(sort(abs(deletedRes), decreasing = TRUE),3)
```

```
##       14       11       15
## 2.102726 1.823020 1.489732
```

H_0: Index 14 is not a outlier H_a: Index 14 is an outlier Decision Rule: $|t14| <= 3.3$, we conclude H_0. Otherwise, H_a

Conclusion: if $|t\_i| <= 3.308$ conclude no outliers, otherwise conclude that i is an outlier. Since $|t\_14| = 2.103$ we can conclude no outliers

b. Obtain the diagonal elements of the hat matrix, and provide an explanation for the pattern in these elements.

Student's Solution Below

```r
#find diagonal elements
diagonalElements = hatvalues(lmCH10PR09)
print(diagonalElements)
```

```
##      1      2      3      4      5      6      7      8      9     10
## 0.2375 0.2375 0.2375 0.2375 0.1375 0.1375 0.1375 0.1375 0.1375 0.1375
##     11     12     13     14     15     16
## 0.1375 0.1375 0.2375 0.2375 0.2375 0.2375
```

```r
#check if they sum up to 3 (our Number of regression parameters)
sum(diagonalElements)
```

```
## [1] 3
```

Conclusion: Our elements seem to have the same two values showing that we do not have outliers.

c. Are any of the observations outlying with regard to their X values according to the rule of thumb stated in the chapter?

Student's Solution Below

```
#mean leverage value
meanLev = p/n

hii > 2* meanLev
```

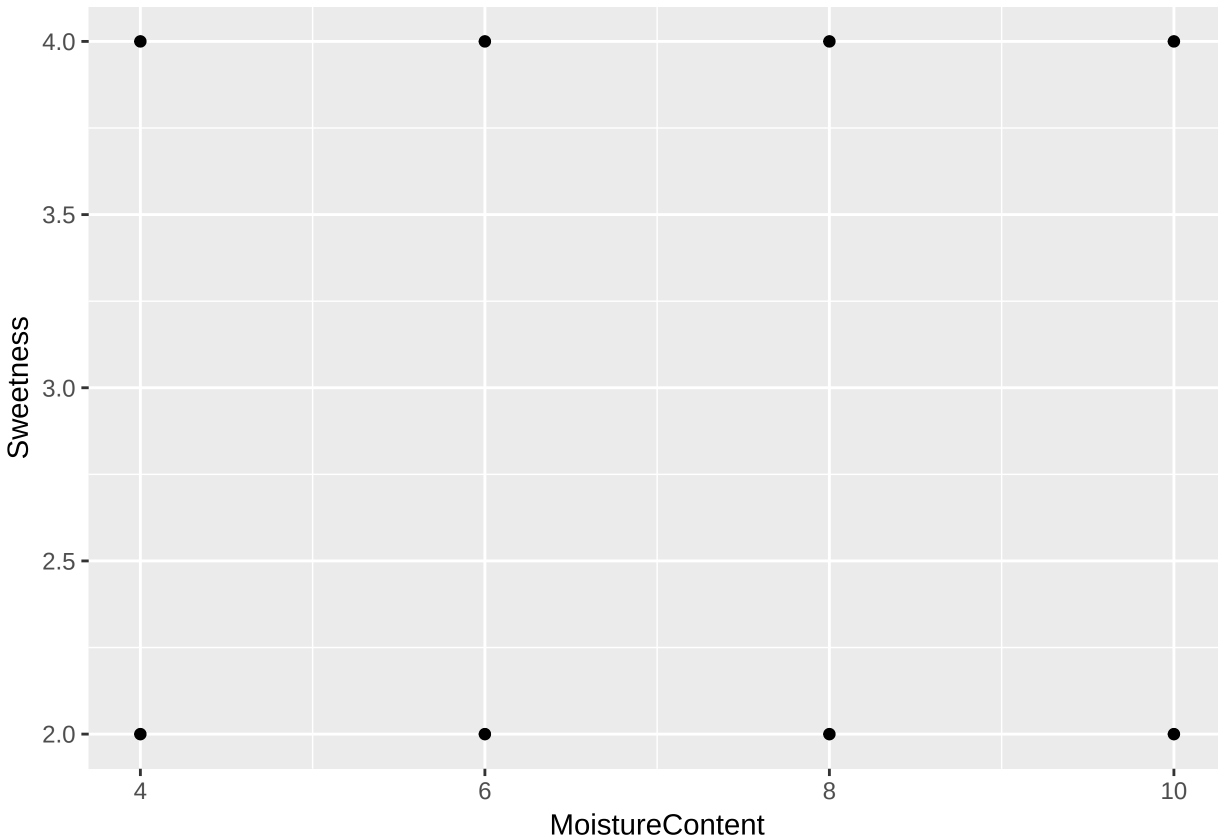
```
##     1     2     3     4     5     6     7     8     9    10    11    12
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    13    14    15    16
## FALSE FALSE FALSE FALSE
```

Conclusion: We see that none of the values are greater than 2 times the leverage so we say that we have no outliers.

d. Management wishes to estimate the mean degree of brand liking for moisture content $X_1 = 10$ and sweetness $X_2 = 3$. Construct a scatter plot of X_2 against X_1 and determine visually whether this prediction involves an extrapolation beyond the range of the data. Also, use (10.29) to determine whether an extrapolation is involved. Do your conclusions from the two methods agree?

Student's Solution Below

```
#scatter plot
ggplot(data=dfCH10PR09,aes(x=MoistureContent, y=Sweetness))+geom_point()
```




```
X = matrix(data=c(rep(1,n), dfCH10PR09$MoistureContent, dfCH10PR09$Sweetness),nrow=n,ncol=p)

X_new  = matrix(data=c(1,10,3),nrow=1,ncol=p)
print(X_new)
```

```
##      [,1] [,2] [,3]
## [1,]    1   10    3
```

```
h_newnew = X_new%*%solve(t(X)%*%X)%*%t(X_new)
print(h_newnew)
```

```
##       [,1]
## [1,] 0.175
```

Conclusion: The values x1 = 10 and x2 = 3 are in the range so we do not have to do any extrapolation. And we see that the leverage point is in line with the existing leverage values.

e. The largest absolute studentized deleted residual is for case 14. Obtain the DFFITS, DFBETAS, and Cook's distance values for this case to assess the influence of this case. What do you conclude?

Student's Solution Below

```
#obtain DFFITS
dffits_14 = dffits(lmCH10PR09)[14]
print(round(dffits_14,3))
```

```
##     14
## -1.174
```

```r
#obtain DFBETAS
dfbetas_14 = dfbetas(lmCH10PR09)[14,]
print(round(dfbetas_14,3))
```

```
##     (Intercept) MoistureContent       Sweetness
##           0.839          -0.808          -0.602
```

```r
#cooks distance

x = cbind(1,dfCH10PR09$MoistureContent,dfCH10PR09$Sweetness)

h = x%*% solve(t(x)%*%x)%*% t(x)
h_diag = diag(h)
sum_h = sum(h_diag)

MSE = SSE / lmCH10PR09$df.residual

cooksdistance = lmCH10PR09$residuals^2/(sum_h*MSE)*(h_diag/(1-h_diag)^2)
print(cooksdistance[14])
```

```
##        14
## 0.3634123
```

Conclusion: The absolute value of dfits at 14 is bigger than 1 but is a bit close so it could be an influential case. But then when we look at the beta values none of them are near one so we see that this is not an influential case. From all of our reviews it seems that case 14 does seem to be an influential case.

f. Calculate the average absolute percent difference in the fitted values with and without case 14. What does this measure indicate about the influence of case 14?

Student's Solution Below

```r
predWith = fitted(lmCH10PR09)
fitWithout = lm(BrandLiking~MoistureContent+Sweetness, dfCH10PR09[-14,])
predWithout = predict(fitWithout, newdata =dfCH10PR09)

averageAbs = 100*mean(abs(predWith-predWithout)/predWith)
print(averageAbs)
```

```
## [1] 0.677679
```

Conclusion: So we can see here that the difference between the with and without case of 14 would be 68%.

g. Calculate Cook's distance D_i for each case and prepare an index plot. Are any cases influential according to this measure?

Student's Solution Below

```r
Di <- ((ei^2)/(p*MSE))*(hii/(1-hii)^2)
print(Di)
```

```
##            1            2            3            4            5
## 0.0001877130 0.0004223542 0.1803921815 0.1862582123 0.0076655286
##            6            7            8            9           10
## 0.0245466787 0.0322971439 0.0143542862 0.0122308711 0.0204060192
##           11           12           13           14           15
```

```
## 0.1498281704 0.0509831969 0.1318214458 0.3634123447 0.2106609008
##           16
## 0.0067576676
```

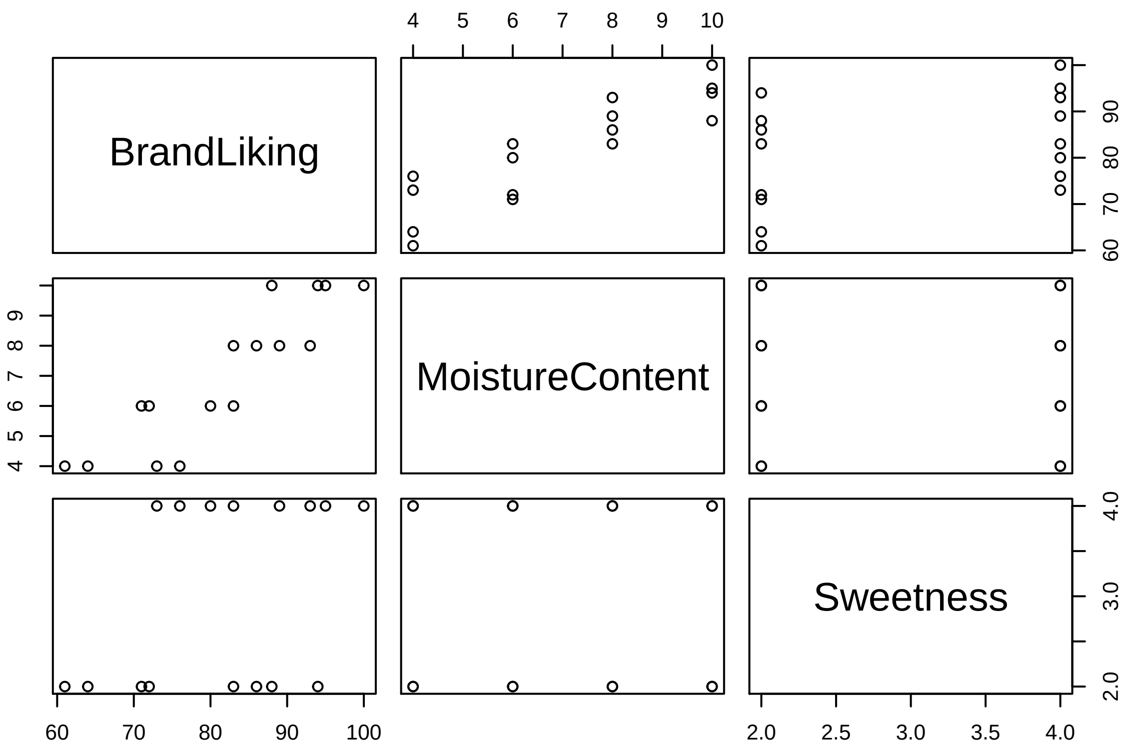
Conclusion: There are no influential points from the graphs.

(10.15) Reference Brand Preference Problem 6.5a.

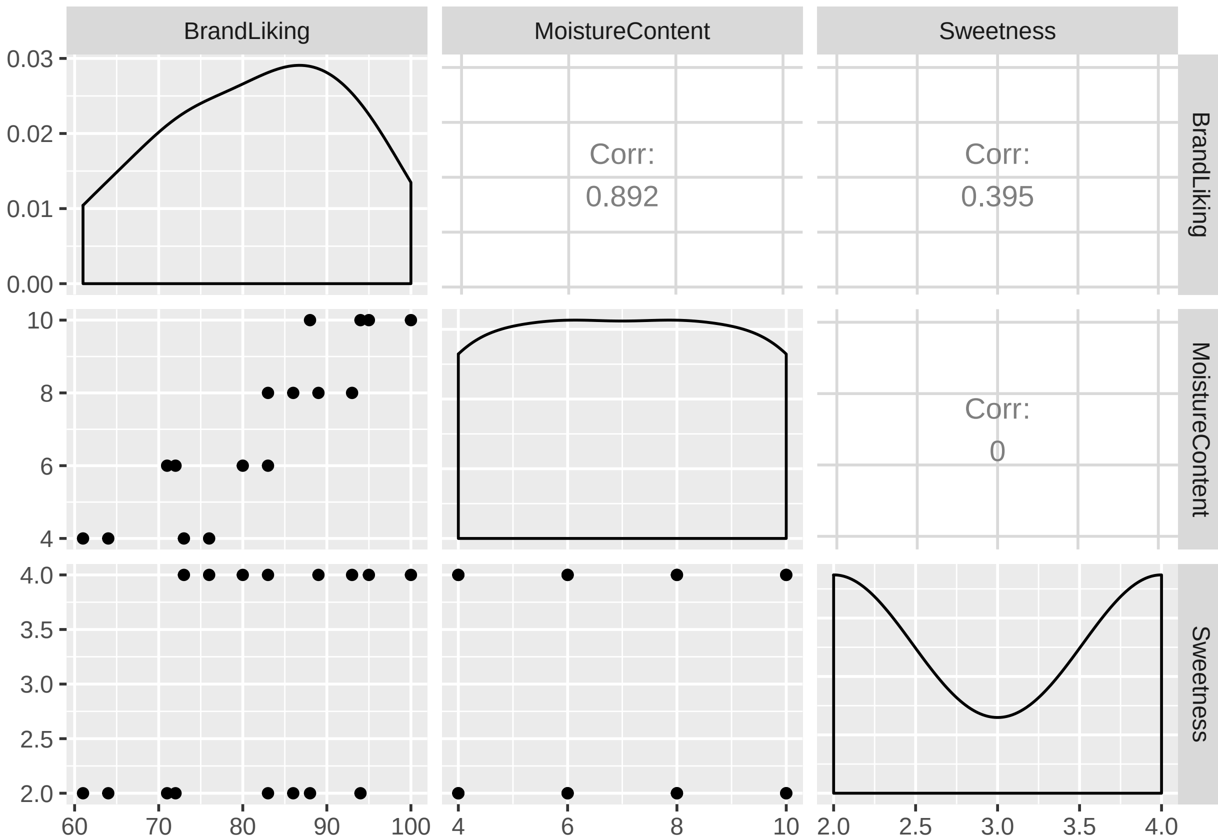
a. What do the scatter plot matrix and the correlation matrix show about pairwise linear associations among the predictor variables?

Student's Solution Below

```
#load the data
dfCH10PR15 =read.delim(file="CH06PR05.txt", sep="", header = FALSE)
colnames(dfCH10PR15) =c("BrandLiking", "MoistureContent", "Sweetness")

#plot the pairs (theres a few ways to do this)...
pairs(dfCH10PR15)
```




```
#or we could use ggpairs
library(GGally)
ggpairs(dfCH10PR15)
```




```r
#matrix correlation
cor(dfCH10PR15)
```

```
##                 BrandLiking MoistureContent Sweetness
## BrandLiking       1.0000000       0.8923929 0.3945807
## MoistureContent   0.8923929       1.0000000 0.0000000
## Sweetness         0.3945807       0.0000000 1.0000000
```

Conclusion: We can see that there is no correlation between them.

b. Find the two variance inflation factors. Why are they both equal to 1?

Student's Solution Below

```r
library(car)
```

```
## Loading required package: carData
```

```r
#find our lmfit
lmCH10PR15 =lm(BrandLiking~MoistureContent+Sweetness, dfCH10PR09)

#find our two variance inflation factors
vif(lmCH10PR15)
```

```
## MoistureContent       Sweetness
##               1               1
```

(10.27) Refer to the SENIC dataset in Appendix C.1 and Project 9.25.

SENIC DATASET DESCRIPTION: The primary objective of the Study on the Efficacy of Nosocomial Infection Control (SENIC Project) was to determine whether infection surveillance and control programs have reduced the rates of nosocomial (hospital-acquired) infection in United States hospitals. This data set consists of a random sample of 113 hospitals selected from the original 338 hospitals surveyed. Each line of the dataset has an identification number and provides information on 11 variables for a single hospital. The data presented here are for the 1975-76 study period.

*Please use dataset titled **APPENC01.txt** when applicable*

The regression model containing age, routine chest X-ray ratio, and average daily census in first-order terms is to be evaluated in detail based on the model-building data set.

a. Obtain the residuals and plot them separately against $\hat{Y}$, each of the predictor variables in the model, and each of the related cross-product terms. On the basis of these plots, should any modifications of the model be made?

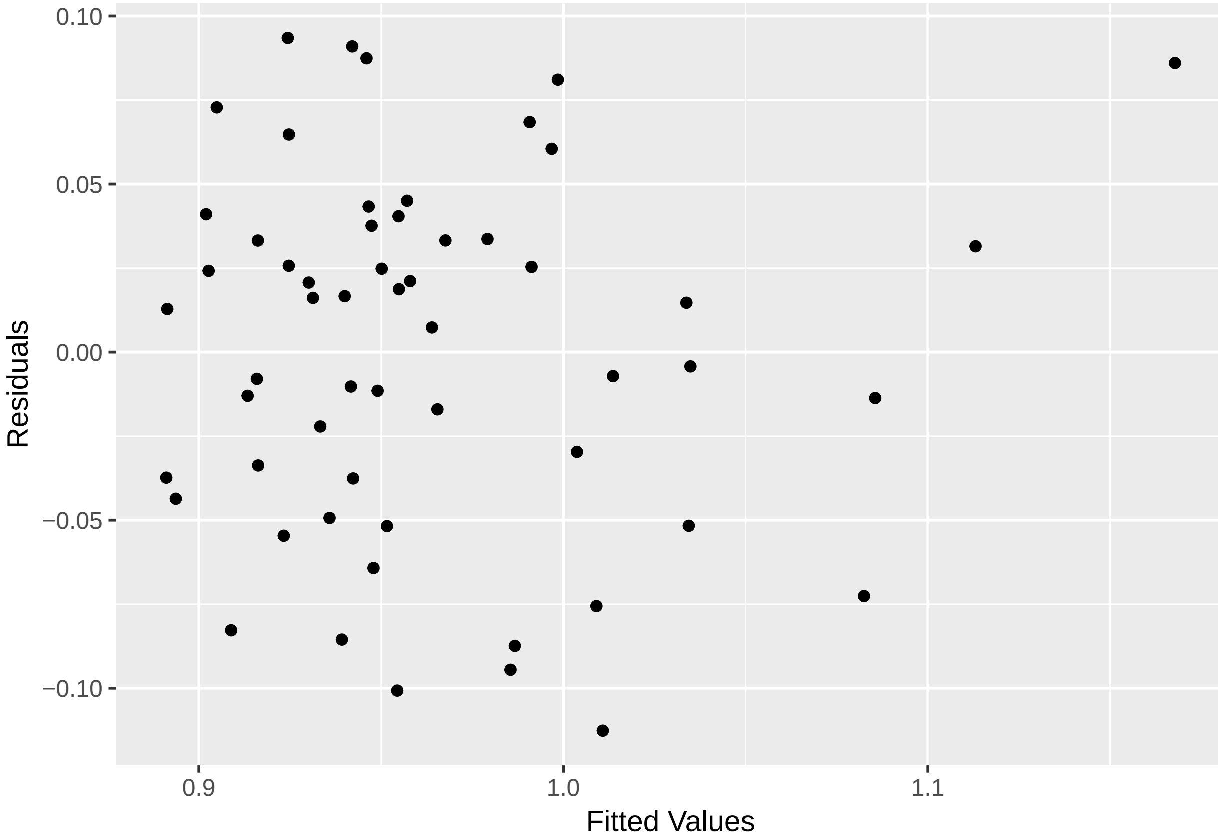
Student's Solution Below

```
#load the dataset and give it values
senic.df = read.table("APPENC01.txt", header=FALSE, col.names = c("id", "los", "age", "infection_risk",

senic.df[ ,c('msa', 'region')] = list(NULL)

#we then see that we need to use cases 57 - 113 only in order to build our model properly

model.df = senic.df[57:113,]
attach(model.df)
model =lm(log10(los)~age+xray+adc, data = model.df)

lm = lm(los~(age+xray+adc)^2,senic.df)


#we can print out our residuals to show them which would satisfy the
#first question to obtain our residuals from the model

ei = model$residuals
print(round(ei,3))
```

```
##      57     58     59     60     61     62     63     64     65     66
## -0.086 -0.064 -0.004  0.068  0.093  0.015 -0.087  0.038 -0.095 -0.030
##      67     68     69     70     71     72     73     74     75     76
##  0.033 -0.076 -0.052 -0.038 -0.055 -0.044  0.021  0.045  0.024 -0.083
##      77     78     79     80     81     82     83     84     85     86
##  0.033 -0.073 -0.017  0.034  0.091 -0.052 -0.034  0.041 -0.008  0.017
##      87     88     89     90     91     92     93     94     95     96
## -0.113  0.025  0.007  0.060  0.016  0.021  0.026 -0.022  0.043 -0.010
##      97     98     99    100    101    102    103    104    105    106
## -0.012  0.081 -0.013 -0.007  0.065  0.040 -0.101  0.031  0.025  0.087
##     107    108    109    110    111    112    113
## -0.037  0.013 -0.014  0.073 -0.049  0.086  0.019
```

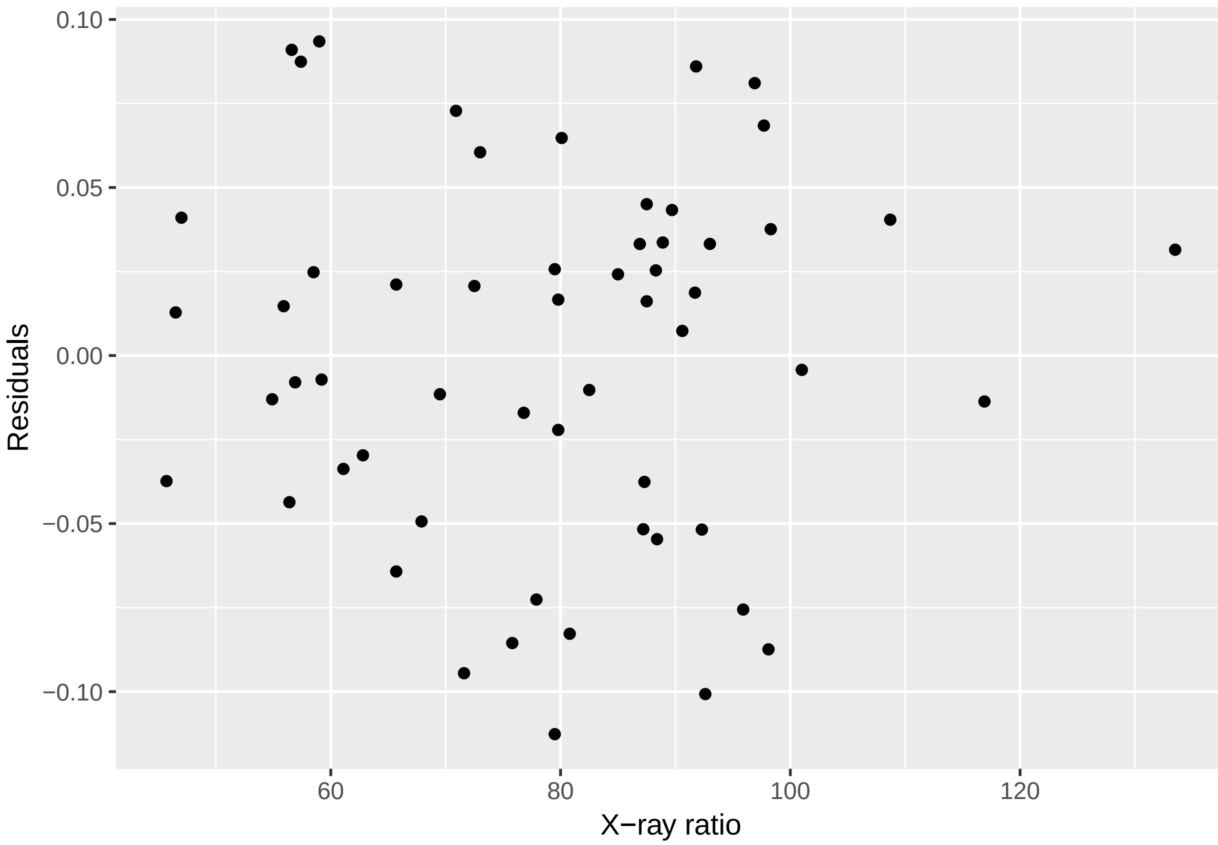
```
#now lets plot the points seperately against y hat
ggplot(model,aes(x=model$fitted.values, y=model$residuals))+
  geom_point()+ geom_smooth(method=lm)+xlab("Fitted Values")+ ylab("Residuals")
```

```
## Warning: Computation failed in `stat_smooth()`:
## 'what' must be a function or character string
```



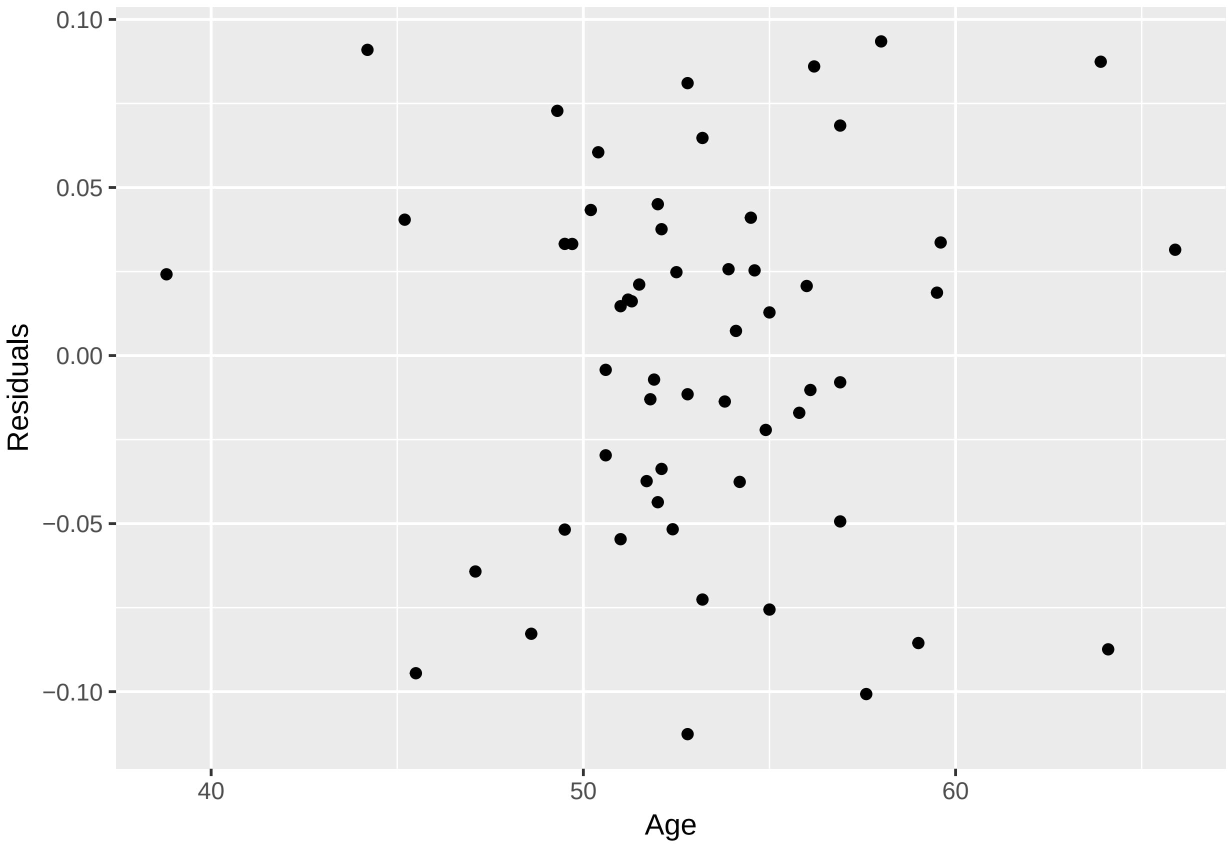

```r
ggplot(model,aes(x=xray, y=model$residuals))+ geom_point()+
  geom_smooth(method=lm)+xlab("X-ray ratio")+ ylab("Residuals")
```

```
## Warning: Computation failed in `stat_smooth()`:
## 'what' must be a function or character string
```



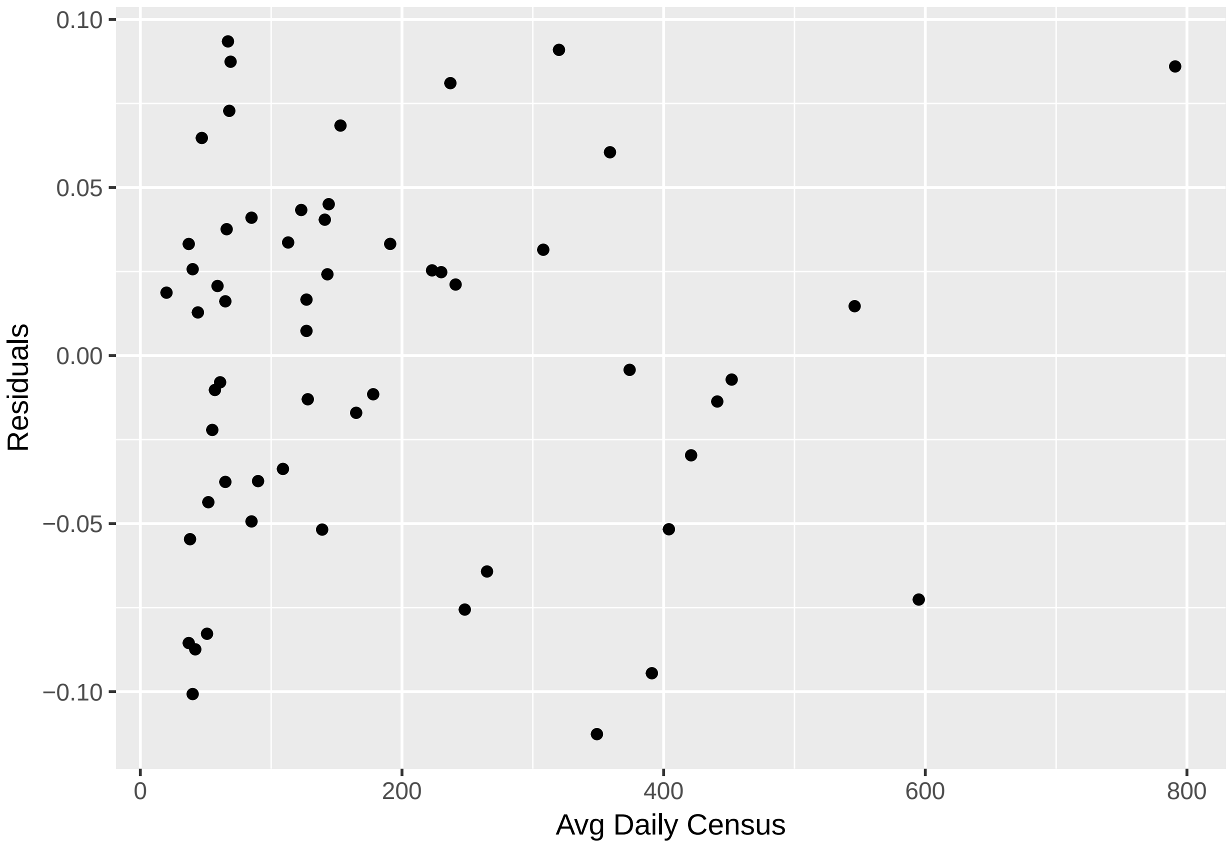

```r
ggplot(model,aes(x=age, y=model$residuals))+ geom_point()+
  geom_smooth(method=lm)+xlab("Age")+ ylab("Residuals")
```

```
## Warning: Computation failed in `stat_smooth()`:
## 'what' must be a function or character string
```




```
ggplot(model,aes(x=adc, y=model$residuals))+ geom_point()+
  geom_smooth(method=lm)+xlab("Avg Daily Census")+ ylab("Residuals")
```

```
## Warning: Computation failed in `stat_smooth()`:
## 'what' must be a function or character string
```



Conclusion: We say that our variances look to be ok and do not seem to show any certain pattern.

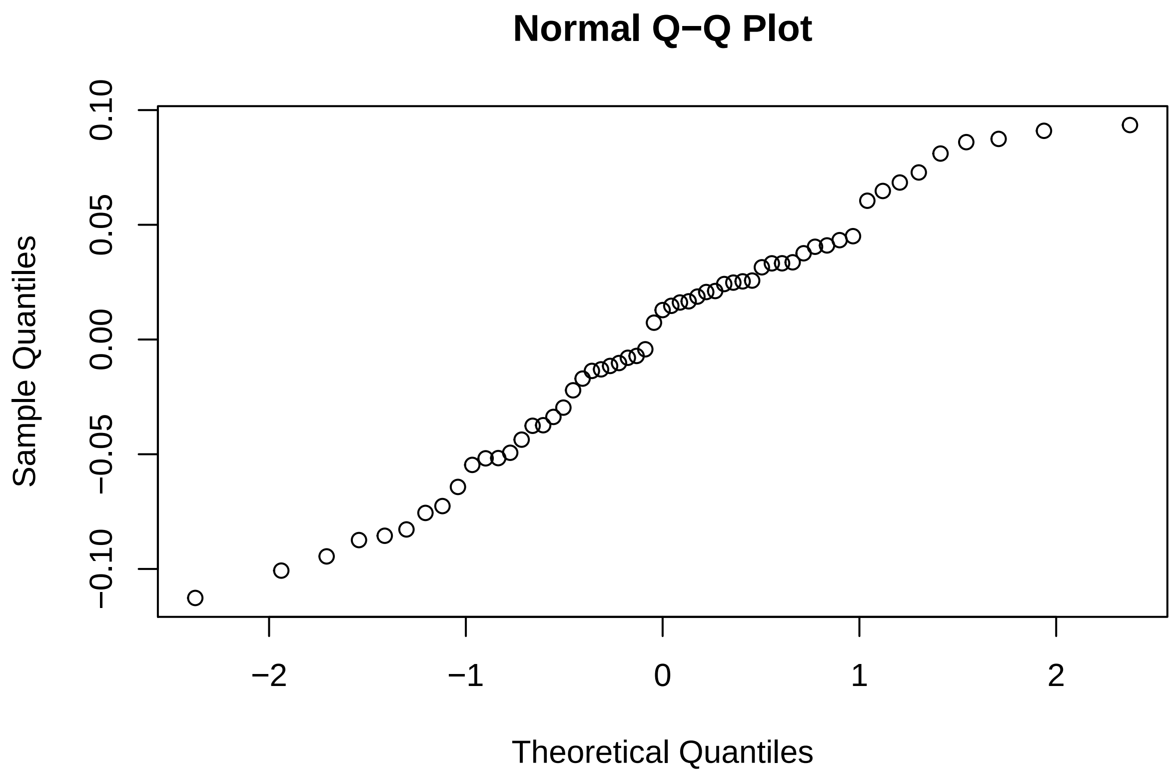
b. Prepare a normal probability plot of the residuals. Also obtain the coefficient of correlation between the ordered residuals and their expected values under normality. Test the reasonableness of the normality assumption, using Table B.6 and $\alpha = .05$. What do you conclude?

Student's Solution Below

```
#first we want to show our expected values
MSE =anova(model)$"Mean Sq"[4]
k = rank(ei)
n = length(age)
expV =sqrt(MSE)*qnorm((k-.375)/(n+0.25))
print(round(expV,3))
```

```
##      57     58     59     60     61     62     63     64     65     66
## -0.077 -0.057 -0.005  0.066  0.127  0.002 -0.084  0.039 -0.093 -0.028
##      67     68     69     70     71     72     73     74     75     76
##  0.033 -0.066 -0.046 -0.036 -0.053 -0.039  0.015  0.053  0.017 -0.071
##      77     78     79     80     81     82     83     84     85     86
##  0.031 -0.061 -0.022  0.036  0.105 -0.049 -0.031  0.046 -0.010  0.007
##      87     88     89     90     91     92     93     94     95     96
## -0.127  0.022 -0.002  0.057  0.005  0.012  0.025 -0.025  0.049 -0.012
##      97     98     99    100    101    102    103    104    105    106
## -0.015  0.077 -0.017 -0.007  0.061  0.043 -0.105  0.028  0.020  0.093
##     107    108    109    110    111    112    113
## -0.033  0.000 -0.020  0.071 -0.043  0.084  0.010
```

```r
#normal probability plot
qqnorm(ei)
```



Conclusion:

H0: normal Ha: not normal We will conclude H0 in this case since our r = .990 based on table b.6 our r critical is .98 so our errors are normally distributed.

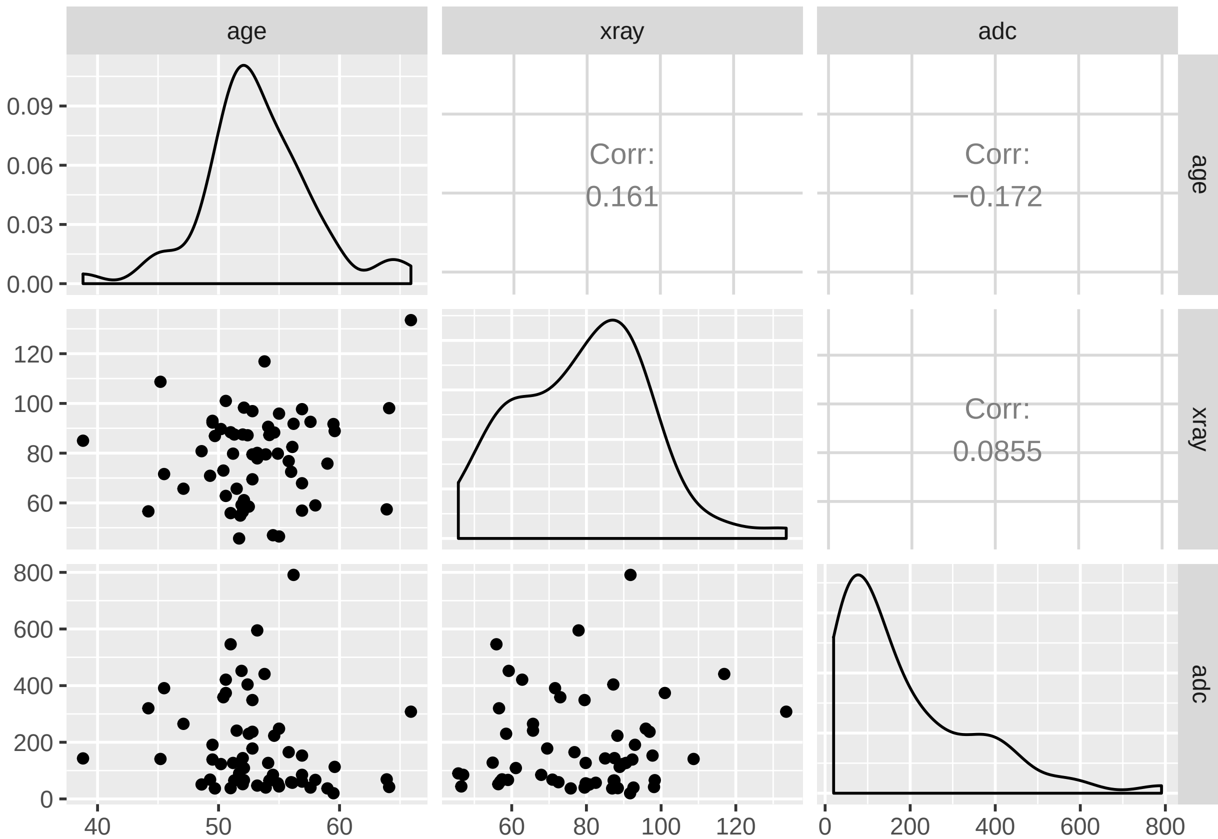
c. Obtain the scatter plot matrix, the correlation matrix of the X variables, and the variance inflation factors. Are there any indications that serious multicollinearity problems are present? Explain.

Student's Solution Below

```r
#show our variance inflation factors
round(vif(model),3)
```

```
##   age  xray   adc
## 1.065 1.041 1.045
```

```r
#scatter plot matrix
ourxvars =data.frame(age, xray, adc)

ggpairs(ourxvars)
```



Conclusion: We see no multicollinearity issues as our VIF values are all close to 1.

d. Obtain the studentized deleted residuals and prepare a dot plot of these residuals. Are any outliers present? Use the Bonferroni outlier test procedure with $\alpha = .01$. State the decision rule and conclusion.

Student's Solution Below

```
#first lets find our hii
one =rep(1,length(los))
x_matrix =cbind(one, age, xray, adc)
x_matrix_trans =t(x_matrix)
h_matrix = x_matrix%*%(solve(x_matrix_trans%*%x_matrix))%*%x_matrix_trans
hii =diag(h_matrix)
print(round(hii,3))
```

```
##  [1] 0.055 0.055 0.069 0.045 0.069 0.142 0.129 0.056 0.084 0.073 0.044
## [12] 0.037 0.051 0.032 0.048 0.055 0.031 0.026 0.213 0.055 0.055 0.133
## [23] 0.024 0.054 0.106 0.047 0.038 0.082 0.066 0.024 0.035 0.024 0.028
## [34] 0.043 0.039 0.035 0.031 0.030 0.040 0.033 0.023 0.037 0.051 0.093
## [45] 0.031 0.149 0.052 0.288 0.043 0.157 0.082 0.090 0.131 0.044 0.042
## [56] 0.288 0.067
```

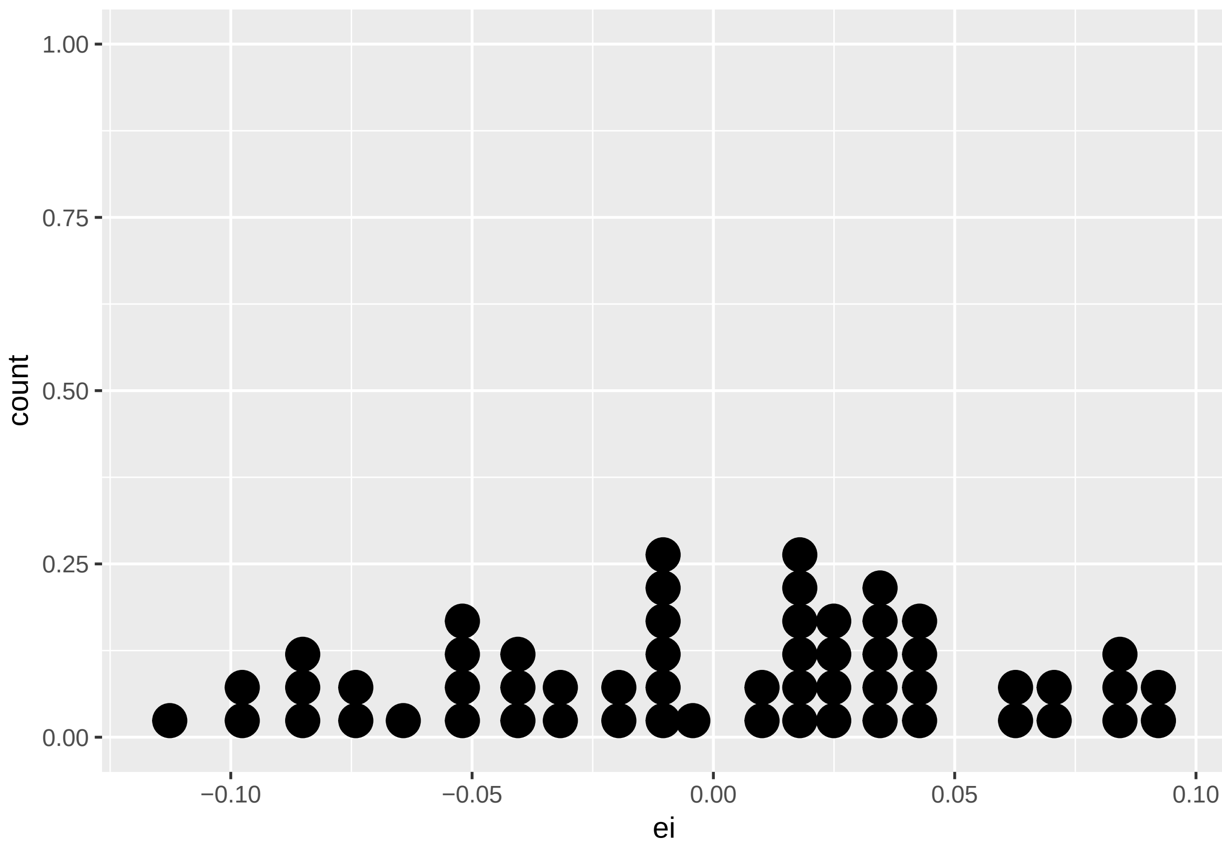
```
#now we need to find the student deleted residuals

model.anova = anova(model)
ei = model$residuals
n = length(los)
p = 4
```

```
sse = model.anova$"Sum Sq"[4]

ti = ei*sqrt(((n-p-1)/(sse*(1-hii)-ei^2)))
print(round(ti,3))
```

```
##     57     58     59     60     61     62     63     64     65     66
## -1.617 -1.201 -0.079  1.274  1.789  0.284 -1.726  0.697 -1.826 -0.554
##     67     68     69     70     71     72     73     74     75     76
##  0.611 -1.406 -0.959 -0.688 -1.014 -0.810  0.385  0.823  0.489 -1.561
##     77     78     79     80     81     82     83     84     85     86
##  0.614 -1.425 -0.309  0.622  1.776 -0.959 -0.619  0.772 -0.148  0.302
##     87     88     89     90     91     92     93     94     95     96
## -2.145  0.461  0.133  1.122  0.295  0.378  0.469 -0.403  0.797 -0.187
##     97     98     99    100    101    102    103    104    105    106
## -0.209  1.513 -0.239 -0.135  1.195  0.790 -1.918  0.672  0.456  1.756
##    107    108    109    110    111    112    113
## -0.702  0.241 -0.263  1.358 -0.911  1.889  0.348
```

```
#prepare our dotplot
ggplot(model,aes(x = ei))+ geom_dotplot()
```

```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#now use the Bonferroni outlier test procedure with $\alpha =.01$

t_crit = qt(1-.01/(2*n),n-p-1)
print(round(t_crit,3))
```

```
## [1] 4.042
```

Conclusion: If our ti <= t_crit then we say no outliers. If our ti > t_crit then we say there are outliers.

We can see that all of our abs(ti) <= 4.04 so we can conclude that there are no outliers.

e. Obtain the diagonal elements of the hat matrix. Using the rule of thumb in the text, identify any outlying X observations.

Student's Solution Below

```r
#Obtain the diagonal elements of the hat matrix which weve already done
print(round(hii,3))
```

```
##  [1] 0.055 0.055 0.069 0.045 0.069 0.142 0.129 0.056 0.084 0.073 0.044
## [12] 0.037 0.051 0.032 0.048 0.055 0.031 0.026 0.213 0.055 0.055 0.133
## [23] 0.024 0.054 0.106 0.047 0.038 0.082 0.066 0.024 0.035 0.024 0.028
## [34] 0.043 0.039 0.035 0.031 0.030 0.040 0.033 0.023 0.037 0.051 0.093
## [45] 0.031 0.149 0.052 0.288 0.043 0.157 0.082 0.090 0.131 0.044 0.042
## [56] 0.288 0.067
```

```r
#use rule of thumb to identify any outlying x observations
find_any_cases = which(hii> (2*p/n))
find_any_cases = find_any_cases+56
print(find_any_cases)
```

```
## [1]  62  75 102 104 106 112
```

```r
#so we see that the following are cases that could have outlying observations as we can see in part f
```

f. Cases $62, 75, 106$, and 112 are moderately outlying with respect to their X values, and case 87 is reasonably far outlying with respect to its Y value. Obtain DFFITS, DFBETAS, and Cook's distance values for these cases to assess their influence. What do you conclude?

Student's Solution Below

```r
#obtain DFFITS

dffits = dffits(lm)[c(62,75,87,106,112)]
print(dffits)
```

```
##            62            75            87           106           112
##  0.3024343198  0.0008426962 -0.2337295766  0.9070194440  1.6461992492
```

```r
#obtain DFBETAS
df_betas = dfbetas(lm)[c(62,75,87,106,112),]
print(df_betas>2/sqrt(n))
```

```
##     (Intercept)   age  xray   adc age:xray age:adc xray:adc
## 62        FALSE FALSE FALSE FALSE    FALSE   FALSE    FALSE
## 75        FALSE FALSE FALSE FALSE    FALSE   FALSE    FALSE
## 87        FALSE FALSE FALSE FALSE    FALSE   FALSE    FALSE
## 106       FALSE  TRUE  TRUE FALSE    FALSE   FALSE    FALSE
## 112       FALSE FALSE FALSE FALSE    FALSE    TRUE     TRUE
```

```r
#cooks distance

cooks_dist = cooks.distance(lm)[c(62,75,87,106,112)]
```

```r
f = pf(cooks_dist,p,n-p)
print(cooks_dist)
```

```
##           62           75           87          106          112
## 1.312777e-02 1.024143e-07 7.673756e-03 1.148007e-01 3.615651e-01
```

(9.33) Case Study. Reference to Real estate sales Case Study 9.31.

The regression model identified in Case Study 9.31 is to be validated by means of the validation data set consisting of those cases not selected for the model building data set.

9.31. *Residential sales that occurred during the year 2002 were available from a city in the Midwest. Data on 522 arms-length transactions include sales price, style, finished square feet, number of bedrooms, pool, lot size, year built, air conditioning, and whether or not the lot is adjacent to a highway. The city tax assessor was interested in predicting sales price based on the demographic variable information given above. Select a random sample of 300 observations to use in the model-building data set. Develop a best subset model for predicting sales price. Justify your choice of model. Assess your model's ability to predict and discuss its use as a tool for predicting sales price.*

Data Set C.7. Real Estate Sales. Page 1353 *The city tax assessor was interested in predicting residential home sales prices in a Midwestern city as a function of various characteristics of the home and surrounding property. Data on 522 arms-length transactions were obtained for home sales during the year 2002. Each line of the data set has an identification number and provides information on 12 other variables.*

```r
df_933 = read.table(file='APPENC07.txt', sep='', header=FALSE,
                    col.names=c('id','salesPrice','sqFt','nBeds','nBaths',
                    'ac','garageSize','pool','year','quality',
                    'style','lotSize','hwy'))
```

a. Fit the regression model identified in Case Study 9.31 to the validation data set. Compare the estimated regression coefficients and their estimated standard errors with those obtained in Case Study 9.31. Also compare the error mean square and coefficients of multiple determination. Does the model fitted to the validation data set yield similar estimates as the model fitted to the model-building data set?

Solution Below

```r
# Prep from 9.31

# Feature Engineering
age = 2002 - df_933$year
style1 = as.numeric(df_933$style == 7)
uniform = runif(nrow(df_933))
df_933_sorted = cbind(df_933, age, style1, uniform)
df_933_sorted = as.data.frame(df_933_sorted[order(uniform),])

# Partition Train and Test sets
trainSample = as.data.frame(df_933_sorted[1:300,])
valSample = as.data.frame(df_933_sorted[301:522,])

# To find the best model, basically fit the model and iteratively delete the insignificant variables

# Recall the factor variables: garage size, quality, style

summary(lm(log(salesPrice) ~ sqFt + nBeds + nBaths + ac + factor(garageSize) + pool + age + factor(quali
```

```
##
## Call:
## lm(formula = log(salesPrice) ~ sqFt + nBeds + nBaths + ac + factor(garageSize) +
##     pool + age + factor(quality) + style1 + lotSize + hwy, data = trainSample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52049 -0.10579 -0.00739  0.10699  0.52594
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.183e+01  1.135e-01 104.228  < 2e-16 ***
## sqFt                 2.777e-04  2.870e-05   9.676  < 2e-16 ***
## nBeds               -1.255e-02  1.256e-02  -0.999  0.31846
## nBaths               5.243e-02  1.696e-02   3.091  0.00219 **
## ac                   3.674e-02  3.167e-02   1.160  0.24700
## factor(garageSize)1  4.109e-02  8.446e-02   0.486  0.62703
## factor(garageSize)2  7.655e-02  8.123e-02   0.942  0.34677
## factor(garageSize)3  2.182e-01  8.756e-02   2.491  0.01329 *
## factor(garageSize)4  2.501e-02  1.492e-01   0.168  0.86696
## factor(garageSize)7  2.391e-02  2.006e-01   0.119  0.90521
## pool                 4.419e-02  3.809e-02   1.160  0.24689
## age                 -3.667e-03  7.812e-04  -4.694 4.17e-06 ***
## factor(quality)2    -2.199e-01  3.959e-02  -5.554 6.43e-08 ***
## factor(quality)3    -3.312e-01  5.568e-02  -5.949 7.94e-09 ***
## style1              -6.240e-02  3.197e-02  -1.952  0.05189 .
## lotSize              5.243e-06  8.724e-07   6.010 5.71e-09 ***
## hwy                 -1.420e-01  6.245e-02  -2.274  0.02372 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1708 on 283 degrees of freedom
## Multiple R-squared:  0.8639, Adjusted R-squared:  0.8563
## F-statistic: 112.3 on 16 and 283 DF,  p-value: < 2.2e-16
```

```r
summary(lm(log(salesPrice) ~ sqFt + nBaths + ac + factor(garageSize) + pool + age + factor(quality) + s
```

```
##
## Call:
## lm(formula = log(salesPrice) ~ sqFt + nBaths + ac + factor(garageSize) +
##     pool + age + factor(quality) + style1 + lotSize + hwy, data = trainSample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54910 -0.10282 -0.00634  0.10769  0.50579
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.183e+01  1.134e-01 104.397  < 2e-16 ***
## sqFt                 2.695e-04  2.750e-05   9.800  < 2e-16 ***
## nBaths               4.772e-02  1.630e-02   2.929  0.00368 **
## ac                   3.417e-02  3.157e-02   1.082  0.27997
## factor(garageSize)1  3.073e-02  8.382e-02   0.367  0.71418
## factor(garageSize)2  6.795e-02  8.077e-02   0.841  0.40090
## factor(garageSize)3  2.107e-01  8.724e-02   2.415  0.01635 *
```

```
## factor(garageSize)4  2.007e-02  1.491e-01   0.135  0.89302
## factor(garageSize)7  3.310e-02  2.004e-01   0.165  0.86892
## pool                 4.446e-02  3.809e-02   1.167  0.24404
## age                 -3.673e-03  7.812e-04  -4.702 4.03e-06 ***
## factor(quality)2    -2.268e-01  3.899e-02  -5.817 1.62e-08 ***
## factor(quality)3    -3.397e-01  5.503e-02  -6.172 2.31e-09 ***
## style1              -6.109e-02  3.194e-02  -1.913  0.05680 .
## lotSize              5.200e-06  8.714e-07   5.967 7.17e-09 ***
## hwy                 -1.444e-01  6.240e-02  -2.314  0.02139 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1708 on 284 degrees of freedom
## Multiple R-squared:  0.8635, Adjusted R-squared:  0.8563
## F-statistic: 119.7 on 15 and 284 DF,  p-value: < 2.2e-16
```

```
summary(lm(log(salesPrice) ~ sqFt + nBaths + ac + pool + age + factor(quality) + style1 + lotSize+ hwy,
```

```
##
## Call:
## lm(formula = log(salesPrice) ~ sqFt + nBaths + ac + pool + age +
##     factor(quality) + style1 + lotSize + hwy, data = trainSample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56512 -0.10969 -0.01351  0.11222  0.50999
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.197e+01  8.971e-02 133.448  < 2e-16 ***
## sqFt              2.794e-04  2.752e-05  10.151  < 2e-16 ***
## nBaths            4.339e-02  1.663e-02   2.610  0.00953 **
## ac                4.276e-02  3.196e-02   1.338  0.18194
## pool              3.932e-02  3.923e-02   1.002  0.31702
## age              -4.367e-03  7.863e-04  -5.554 6.33e-08 ***
## factor(quality)2 -2.829e-01  3.804e-02  -7.437 1.18e-12 ***
## factor(quality)3 -4.105e-01  5.471e-02  -7.503 7.76e-13 ***
## style1           -4.556e-02  3.243e-02  -1.405  0.16119
## lotSize           5.645e-06  8.840e-07   6.386 6.78e-10 ***
## hwy              -1.506e-01  6.423e-02  -2.345  0.01969 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1766 on 289 degrees of freedom
## Multiple R-squared:  0.8515, Adjusted R-squared:  0.8463
## F-statistic: 165.7 on 10 and 289 DF,  p-value: < 2.2e-16
```

```
summary(lm(log(salesPrice) ~ sqFt + nBaths + ac + age + factor(quality) + style1 + lotSize+ hwy, data=t
```

```
##
## Call:
## lm(formula = log(salesPrice) ~ sqFt + nBaths + ac + age + factor(quality) +
##     style1 + lotSize + hwy, data = trainSample)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.52934 -0.11514 -0.01392  0.11084  0.50866
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.197e+01  8.971e-02 133.450  < 2e-16 ***
## sqFt                2.787e-04  2.751e-05  10.130  < 2e-16 ***
## nBaths              4.555e-02  1.649e-02   2.763  0.00609 **
## ac                  4.374e-02  3.194e-02   1.369  0.17196
## age                -4.333e-03  7.856e-04  -5.516 7.68e-08 ***
## factor(quality)2 -2.846e-01  3.800e-02  -7.489 8.38e-13 ***
## factor(quality)3 -4.128e-01  5.466e-02  -7.552 5.61e-13 ***
## style1             -4.679e-02  3.241e-02  -1.444  0.14988
## lotSize             5.586e-06  8.821e-07   6.333 9.12e-10 ***
## hwy                -1.528e-01  6.420e-02  -2.380  0.01795 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1766 on 290 degrees of freedom
## Multiple R-squared:  0.851,  Adjusted R-squared:  0.8463
## F-statistic:   184 on 9 and 290 DF,  p-value: < 2.2e-16
```

```
trainModel = lm(log(salesPrice) ~ sqFt + nBaths + ac + age + factor(quality) + style1 + lotSize+ hwy, da
```

```
# Preliminary Analysis
#summary(df_933)
#lapply(df_933, mode)
#lapply(df_933, class)


testModel = lm(log(salesPrice) ~ sqFt + nBaths + ac + age + factor(quality) + style1 + lotSize+ hwy, dat

summary(trainModel)
```

```
##
## Call:
## lm(formula = log(salesPrice) ~ sqFt + nBaths + ac + age + factor(quality) +
##     style1 + lotSize + hwy, data = trainSample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52934 -0.11514 -0.01392  0.11084  0.50866
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.197e+01  8.971e-02 133.450  < 2e-16 ***
## sqFt                2.787e-04  2.751e-05  10.130  < 2e-16 ***
## nBaths              4.555e-02  1.649e-02   2.763  0.00609 **
## ac                  4.374e-02  3.194e-02   1.369  0.17196
## age                -4.333e-03  7.856e-04  -5.516 7.68e-08 ***
## factor(quality)2 -2.846e-01  3.800e-02  -7.489 8.38e-13 ***
## factor(quality)3 -4.128e-01  5.466e-02  -7.552 5.61e-13 ***
## style1             -4.679e-02  3.241e-02  -1.444  0.14988
## lotSize             5.586e-06  8.821e-07   6.333 9.12e-10 ***
## hwy                -1.528e-01  6.420e-02  -2.380  0.01795 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1766 on 290 degrees of freedom
## Multiple R-squared:  0.851,  Adjusted R-squared:  0.8463
## F-statistic:   184 on 9 and 290 DF,  p-value: < 2.2e-16
```

```r
summary(valSample)
```

```
##        id          salesPrice          sqFt          nBeds
##  Min.   :  2.0   Min.   : 84000   Min.   : 980   Min.   :0.000
##  1st Qu.:154.2   1st Qu.:183480   1st Qu.:1694   1st Qu.:3.000
##  Median :266.5   Median :231500   Median :2092   Median :3.000
##  Mean   :260.5   Mean   :269237   Mean   :2243   Mean   :3.509
##  3rd Qu.:380.5   3rd Qu.:314625   3rd Qu.:2602   3rd Qu.:4.000
##  Max.   :522.0   Max.   :920000   Max.   :4973   Max.   :6.000
##      nBaths            ac           garageSize         pool
##  Min.   :0.000   Min.   :0.0000   Min.   :0.000   Min.   :0.00000
##  1st Qu.:2.000   1st Qu.:1.0000   1st Qu.:2.000   1st Qu.:0.00000
##  Median :3.000   Median :1.0000   Median :2.000   Median :0.00000
##  Mean   :2.662   Mean   :0.8604   Mean   :2.081   Mean   :0.05856
##  3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:0.00000
##  Max.   :7.000   Max.   :1.0000   Max.   :5.000   Max.   :1.00000
##       year         quality          style          lotSize
##  Min.   :1885   Min.   :1.000   Min.   :1.000   Min.   : 6746
##  1st Qu.:1957   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:18188
##  Median :1966   Median :2.000   Median :3.000   Median :22079
##  Mean   :1967   Mean   :2.212   Mean   :3.523   Mean   :24027
##  3rd Qu.:1980   3rd Qu.:3.000   3rd Qu.:7.000   3rd Qu.:26692
##  Max.   :1997   Max.   :3.000   Max.   :7.000   Max.   :86248
##       hwy                age            style1          uniform
##  Min.   :0.00000   Min.   :  5.0   Min.   :0.0000   Min.   :0.6009
##  1st Qu.:0.00000   1st Qu.: 22.0   1st Qu.:0.0000   1st Qu.:0.6986
##  Median :0.00000   Median : 36.0   Median :0.0000   Median :0.8021
##  Mean   :0.01351   Mean   : 34.7   Mean   :0.3018   Mean   :0.7996
##  3rd Qu.:0.00000   3rd Qu.: 45.0   3rd Qu.:1.0000   3rd Qu.:0.8944
##  Max.   :1.00000   Max.   :117.0   Max.   :1.0000   Max.   :0.9983
```

ANALYSIS

R^2 value for the training model was slightly higher and we see that the R^2 value for the validation model dropped. We can further analyze the variables in the model summaries. Comparing the variables, we see some notable differences. Specifically: number of baths, AC, style1 (our dummy variable), highway.

b. Calculate the mean squared prediction error (9.20) and compare it to MSE obtained from the model-building data set. Is there evidence of a substantial bias problem in MSE here?

Solution Below

```r
anova(trainModel)
```

```
## Analysis of Variance Table
##
## Response: log(salesPrice)
##                   Df Sum Sq Mean Sq   F value    Pr(>F)
## sqFt               1 43.391  43.391 1391.6554 < 2.2e-16 ***
```

```
## nBaths             1  1.973   1.973   63.2801 4.052e-14 ***
## ac                 1  0.526   0.526   16.8628 5.231e-05 ***
## age                1  1.675   1.675   53.7339 2.297e-12 ***
## factor(quality)    2  2.507   1.254   40.2106 3.862e-16 ***
## style1             1  0.202   0.202    6.4696   0.01149 *
## lotSize            1  1.172   1.172   37.5772 2.864e-09 ***
## hwy                1  0.177   0.177    5.6658   0.01795 *
## Residuals        290  9.042   0.031
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
(MSE = anova(trainModel)[9,3])
```

```
## [1] 0.03117945
```

```r
# See MSE is ~0.033
MSE = paste0(signif(MSE, digits=4))

predsTest = predict(trainModel, valSample)

(MSPR = sum((log(valSample$salesPrice) - predsTest)^2)/(nrow(valSample)))
```

```
## [1] 0.03193739
```

```r
MSPR = paste0(signif(MSPR, digits=4))
```

ANALYSIS

The MSE obtainined from the model-building set is 0.03118 and the mean squared prediction error is 0.03194. We see that the two values are fairly similar, with variations between the two being small. There is no evidence of a substantial bias problem in the MSE.