R Codes for Chapter-6
Multiple Regression in R

The function **"lm"** can be used to perform multiple linear regression in R and much of the syntax is the same as that used for fitting simple linear regression models. To perform multiple linear regression with **p** explanatory variables use the command:

lm(response ~ explanatory_1 + explanatory_2 + … + explanatory_p)

Here the terms response and explanatory_i in the function should be replaced by the names of the response and explanatory variables, respectively, used in the analysis.

Ex. Problem 6.9, "Grocery Retailer.": Y=Retailer, X_1 = Cases, X_2 = Costs, and X_3 = Holiday.

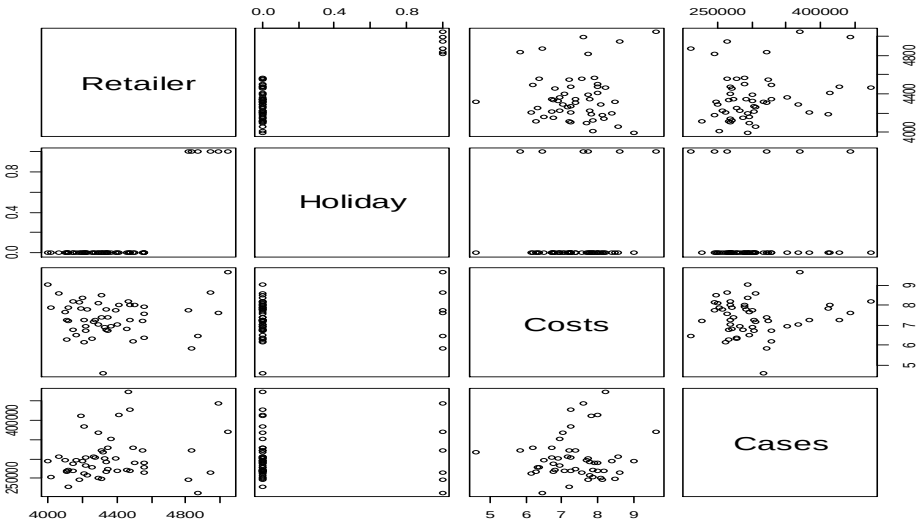
The following program reads in the data.

```
> data<-read.table("http://www.stat.ufl.edu/~rrandles/sta4210/Rclassnotes/data/textdatasets/KutnerData/Chapter%20%206%20Data%20Sets/CH06PR09.txt", header=FALSE , sep="")

> data
     V1     V2   V3 V4
1  4264 305657 7.17  0
2  4496 328476 6.20  0
3  4317 317164 4.61  0
4  4292 366745 7.02  0
.
.
.
> Cases=data[,2]
> Costs=data[,3]
> Holiday=data[,4]
> Retailer=data[,1]
> data=data.frame(Retailer,Holiday,Costs,Cases)
```

Before fitting our regression model we want to investigate how the variables are related to one another. We can do this graphically by constructing scatter plots of all pair-wise combinations of variables in the data frame. This can be done by typing:

```
> plot(data)
```



We also can calculate correlations for all the possible pairs using the command "cor( )" as follows.

```
> cor(data)
          Retailer    Holiday      Costs      Cases
Retailer 1.0000000 0.20766494 0.06002960 0.81057940
Holiday  0.2076649 1.00000000 0.08489639 0.04565698
Costs    0.0600296 0.08489639 1.00000000 0.11337076
Cases    0.8105794 0.04565698 0.11337076 1.00000000
```

Here we can see that there is no a significant interaction among predictor variables.

Fitting the Model

To fit a multiple linear regression model, use the command:

```
> MLR=lm(Retailer~Cases+Costs+Holiday)
> MLR

Call:
lm(formula = Retailer ~ Cases + Costs + Holiday)

Coefficients:
(Intercept)        Cases        Costs      Holiday
  4.150e+03    6.236e+02   -1.317e+01    7.871e-04
```

This output indicates that the fitted model is given by $\hat{Y} = 4150 + 0.000787X_1 - 13.17X_2 + 623.6X_3$.

We can access the results of each test by typing:

```
> summary(MLR)

Call:
lm(formula = Retailer ~ Cases + Costs + Holiday)

Residuals:
    Min      1Q  Median      3Q     Max
-264.05 -110.73  -22.52   79.29  295.75

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.150e+03  1.956e+02  21.220  < 2e-16 ***
Cases        6.236e+02  6.264e+01   9.954 2.94e-13 ***
Costs       -1.317e+01  2.309e+01  -0.570   0.5712
Holiday      7.871e-04  3.646e-04   2.159   0.0359 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 143.3 on 48 degrees of freedom
Multiple R-squared:  0.6883,   Adjusted R-squared:  0.6689
F-statistic: 35.34 on 3 and 48 DF,  p-value: 3.316e-12
```

Confidence and Prediction Intervals:

Confidence intervals for model parameters:

```
> confint(MLR)  # 95% (default significance level) confidence intervals for
                  beeta0 and beet1
                    2.5 %       97.5 %
(Intercept)  3.756677e+03 4.543098e+03
Cases        4.976064e+02 7.495025e+02
Costs       -5.959506e+01 3.326302e+01
Holiday      5.409544e-05 1.520065e-03
```

Confidence intervals for model parameters with other significance levels:

```
> confint(MLR,level=0.99) # 99% confidence intervals for beeta0 and beet1
                     0.5 %       99.5 %
(Intercept)  3.625341e+03 4.674434e+03
Cases        4.555387e+02 7.915703e+02
Costs       -7.510275e+01 4.877072e+01
Holiday     -1.907278e-04 1.764889e-03
```

Confidence interval for mean response:

```
> newx=data.frame(Cases=300000,Costs=6,Holiday=0) # Creating a data frame
                                                    with the new value of predictors.

> predict.lm(MLR,newx, interval="confidence", level=0.95) #95% Confidence
                                                          interval at the new X level.
        fit       lwr       upr
1 187070415 149285995 224854835
```

Prediction interval for a new observation:

```
> newpx=data.frame(Cases=310000,Costs=7,Holiday=1) # Creating a data frame
                                                    with the new value of predictors.

> predict.lm(MLR,newpx, interval="prediction", level=0.99) #99% Confidence
                                                          interval at the new X level.
        fit       lwr       upr
1 193305947 141221055 245390838
```

Note:

- Residual plots: Same as those for Simple Linear Regression Model.
- Simultaneous Inference: Same as those in Simple Linear Regression In R.

The General Linear Regression Model

The command "glm(response ~ explanatory_1 + explanatory_2 + … + explanatory_p-1,family=gaussian)" can be used to fit the general linear regression model after all the corresponding transformations. Here "family=gaussin" is used to fit the model with normal (Gaussian) errors.

```
> GLRM=glm(Retailer~Cases+Costs+Holiday, family=gaussian)
> GLRM

Call:  glm(formula = Retailer ~ Cases + Costs + Holiday, family = gaussian)

Coefficients:
(Intercept)        Cases        Costs      Holiday
  4.150e+03    6.236e+02   -1.317e+01    7.871e-04

Degrees of Freedom: 51 Total (i.e. Null);  48 Residual
Null Deviance:     3162000
Residual Deviance: 985500     AIC: 669.8


> summary(GLRM)

Call:
glm(formula = Retailer ~ Cases + Costs + Holiday, family = gaussian)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-264.05  -110.73   -22.52    79.29   295.75

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.150e+03  1.956e+02  21.220  < 2e-16 ***
Cases        6.236e+02  6.264e+01   9.954 2.94e-13 ***
Costs       -1.317e+01  2.309e+01  -0.570   0.5712
Holiday      7.871e-04  3.646e-04   2.159   0.0359 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 20531.87)

    Null deviance: 3162136  on 51  degrees of freedom
Residual deviance:  985530  on 48  degrees of freedom
AIC: 669.75
```