```r
knitr::opts_chunk$set(echo = TRUE)

cp<-read.table("CommercialProperties.txt",header=TRUE)

cp
```

```
##         Y X1    X2   X3     X4
## 1  13.500  1  5.02 0.14 123000
## 2  12.000 14  8.19 0.27 104079
## 3  10.500 16  3.00 0.00  39998
## 4  15.000  4 10.70 0.05  57112
## 5  14.000 11  8.97 0.07  60000
## 6  10.500 15  9.45 0.24 101385
## 7  14.000  2  8.00 0.19  31300
## 8  16.500  1  6.62 0.60 248172
## 9  17.500  1  6.20 0.00 215000
## 10 16.500  8 11.78 0.03 251015
## 11 17.000 12 14.62 0.08 291264
## 12 16.500  2 11.55 0.03 207549
## 13 16.000  2  9.63 0.00  82000
## 14 16.500 13 12.99 0.04 359665
## 15 17.225  2 12.01 0.03 265500
## 16 17.000  1 12.01 0.00 299000
## 17 16.000  1  7.99 0.14 189258
## 18 14.625 12 10.33 0.12 366013
## 19 14.500 16 10.67 0.00 349930
## 20 14.500  3  9.45 0.03  85335
## 21 16.500  6 12.65 0.13 235932
## 22 16.500  3 12.08 0.00 130000
## 23 15.000  3 10.52 0.05  40500
## 24 15.000  3  9.47 0.00  40500
## 25 13.000 14 11.62 0.00  45959
## 26 12.500  1  5.00 0.33 120000
## 27 14.000 15  9.89 0.05  81243
## 28 13.750 16 11.13 0.06 153947
## 29 14.000  2  7.96 0.22  97321
## 30 15.000 16 10.73 0.09 276099
## 31 13.750  2  7.95 0.00  90000
## 32 15.625  3  9.10 0.00 184000
## 33 15.625  3 12.05 0.03 184718
## 34 13.000 16  8.43 0.04  96000
## 35 14.000 16 10.60 0.04 106350
## 36 15.250 13 10.55 0.10 135512
## 37 16.250  1  5.50 0.21 180000
## 38 13.000 14  8.53 0.03 315000
## 39 14.500  3  9.04 0.04  42500
## 40 11.500 15  8.20 0.00  30005
## 41 14.250  1  6.13 0.00  60000
## 42 15.500 15  8.32 0.00  73521
## 43 12.000  1  4.00 0.00  50000
## 44 14.250 15 10.10 0.00  50724
## 45 14.000  3  5.25 0.16  31750
## 46 16.500  3 11.62 0.00 168000
```

```
## 47 14.500  4  5.31 0.00  70000
## 48 15.500  1  5.75 0.00  27000
## 49 16.750  4 12.46 0.03 129614
## 50 16.750  4 12.75 0.00 129614
## 51 16.750  2 12.75 0.00 130000
## 52 16.750  2 11.38 0.00 209000
## 53 17.000  1  5.99 0.57 220000
## 54 16.000  2 11.37 0.27  60000
## 55 14.500  3 10.38 0.00 110000
## 56 15.000 15 10.77 0.05 101206
## 57 15.000 17 11.30 0.00 288847
## 58 16.000  1  7.06 0.14 105000
## 59 15.500 14 12.10 0.05 276425
## 60 15.250  2 10.04 0.06  33000
## 61 16.500  1  4.99 0.73 210000
## 62 19.250  0  7.33 0.22 240000
## 63 17.750 18 12.11 0.00 281552
## 64 18.750 16 12.86 0.00 421000
## 65 19.250 13 12.70 0.04 484290
## 66 14.000 20 11.58 0.00 234493
## 67 14.000 18 11.58 0.03 230675
## 68 18.000 16 12.97 0.08 296966
## 69 13.750  1  4.82 0.00  32000
## 70 15.000  2  9.75 0.03  38533
## 71 15.500 16 10.36 0.02 109912
## 72 15.900  1  8.13 0.23 236000
## 73 15.250 15 13.23 0.05 243338
## 74 15.500  4 10.57 0.04 122183
## 75 14.750 20 11.22 0.00 128268
## 76 15.000  3 10.34 0.00  72000
## 77 14.500  3 10.67 0.00  43404
## 78 13.500 18  8.60 0.08  59443
## 79 15.000 15 11.97 0.14 254700
## 80 15.250 11 11.27 0.03 434746
## 81 14.500 14 12.68 0.03 201930
```

```
# Check the class of the dataset, and it should be data frame. Use str function to see
↪  the structure of the dataset
#class(cp)
str(cp)
```

```
## 'data.frame':    81 obs. of  5 variables:
##  $ Y : num  13.5 12 10.5 15 14 10.5 14 16.5 17.5 16.5 ...
##  $ X1: int  1 14 16 4 11 15 2 1 1 8 ...
##  $ X2: num  5.02 8.19 3 10.7 8.97 ...
##  $ X3: num  0.14 0.27 0 0.05 0.07 0.24 0.19 0.6 0 0.03 ...
##  $ X4: int  123000 104079 39998 57112 60000 101385 31300 248172 215000 251015 ...
```

```
attach(cp)
```

```
## The following objects are masked from flow (pos = 4):
##
##     X1, X2, X3, X4
```

```
## The following object is masked from drug (pos = 6):
##
##     Y

## The following objects are masked from cp (pos = 7):
##
##     X1, X2, X3, X4, Y

## The following objects are masked from demand (pos = 8):
##
##     X1, X2, X3, Y

## The following objects are masked from cp (pos = 9):
##
##     X1, X2, X3, X4, Y

## The following objects are masked from demand (pos = 10):
##
##     X1, X2, X3, Y

## The following object is masked from drug (pos = 11):
##
##     Y

## The following object is masked from drug (pos = 14):
##
##     Y

## The following objects are masked from demand (pos = 15):
##
##     X1, X2, X3, Y

## The following objects are masked from demand (pos = 20):
##
##     X1, X2, X3, Y

## The following objects are masked from flow (pos = 21):
##
##     X1, X2, X3, X4

## The following objects are masked from cp (pos = 22):
##
##     X1, X2, X3, X4, Y

## The following objects are masked from cp (pos = 23):
##
##     X1, X2, X3, X4, Y

## The following objects are masked from flow (pos = 24):
##
##     X1, X2, X3, X4
```

```
## The following objects are masked from flow (pos = 25):
##
##     X1, X2, X3, X4
```

```
## The following objects are masked from flow (pos = 26):
##
##     X1, X2, X3, X4
```

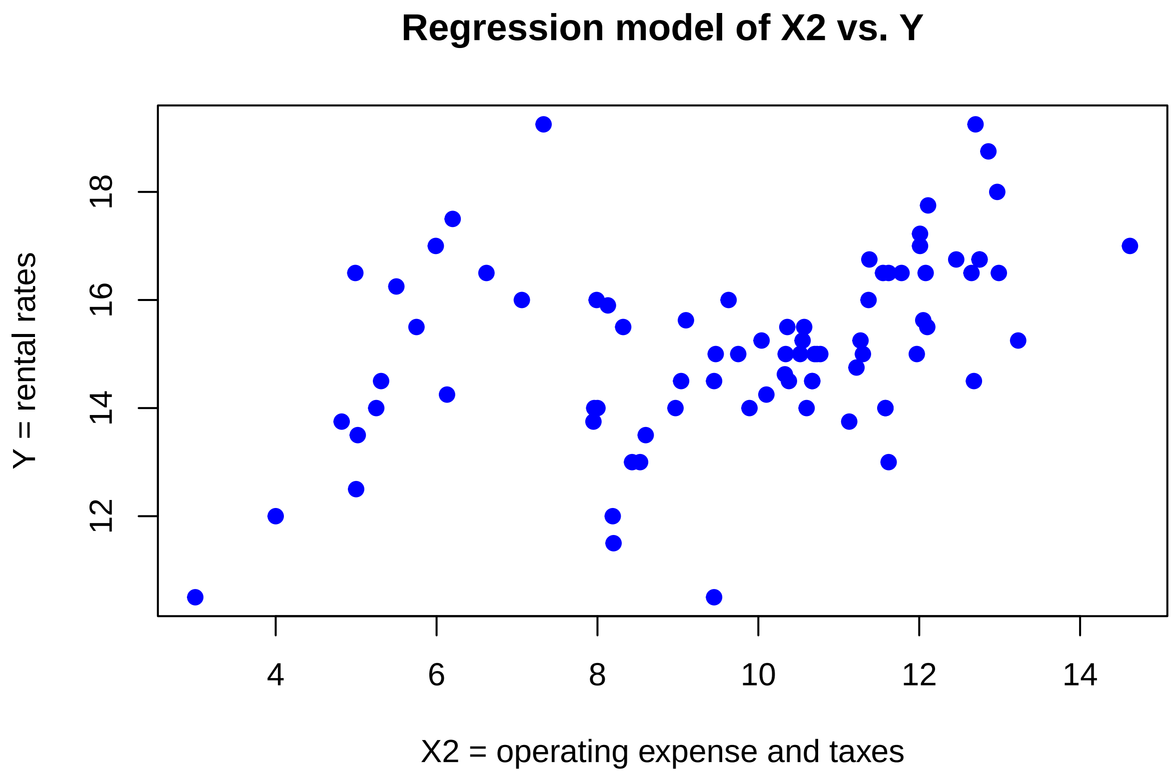
```
# gets dimensions of cp
#dim(cp)

# sets n to number of rows in the data table
n<-dim(cp)[1]

p<-dim(cp)[2]-1
```

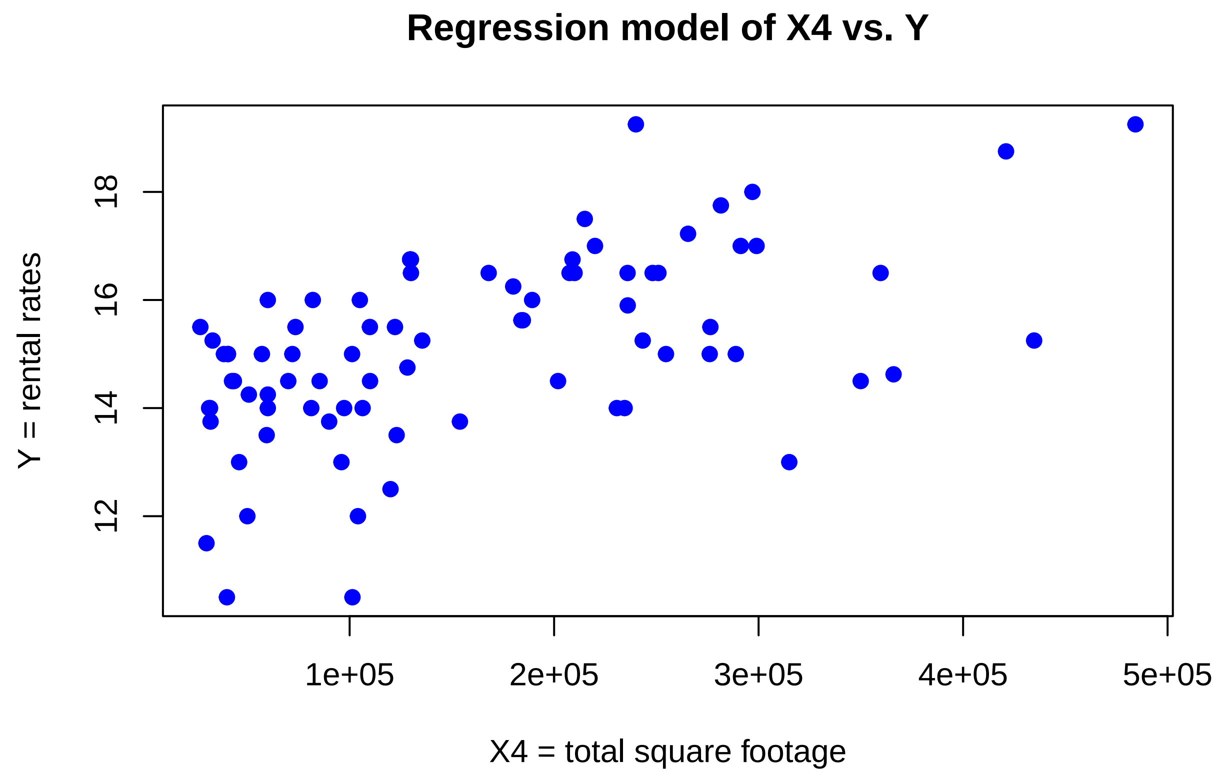
```
# Visualization of correlation of X1 and Y
plot(X1,Y,pch=19,col="blue",xlab="X1 = age of property",ylab="Y = rental
↪  rates",main="Regression model of X1 vs. Y")
```


```
# Visualization of correlation of X2 and Y
plot(X2,Y,pch=19,col="blue",xlab="X2 = operating expense and taxes",ylab="Y = rental
↪  rates",main="Regression model of X2 vs. Y")
```




```
# Visualization of correlation of X3 and Y
plot(X3,Y,pch=19,col="blue",xlab="X3 = vacancy rates",ylab="Y = rental
↪  rates",main="Regression model of X3 vs. Y")
```


```r
# Visualization of correlation of X4 and Y
plot(X4,Y,pch=19,col="blue",xlab="X4 = total square footage",ylab="Y = rental
↪  rates",main="Regression model of X4 vs. Y")
```




```
# Observations:
# X1 = property age
# There is almost no correlation between property age and rental rate.

# X2 = operating expense and taxes
# There is a positive correlation between operating expense & taxes and rental rate. It
↪  seems that properties that cost more to maintain are more popular to rent.

# X3 = vacancy rates
# There is almost no correlation between vacancy rate and rental rate.

# X4 = total square footage
# There is a positive correlation between total square footage of a property and rental
↪  rate. It seems that larger properties are more popular to rent.
```

```
m1<-lm(Y~1)
summary(m1)
```

```
##
## Call:
## lm(formula = Y ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6389 -1.1389 -0.1389  1.3611  4.1111
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  15.1389     0.1911   79.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.72 on 80 degrees of freedom
```

```
m2<-lm(Y~X1)
summary(m2)
```

```
##
## Call:
## lm(formula = Y ~ X1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1759 -0.9545  0.1705  0.9157  4.4444
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.64918    0.28978  54.003   <2e-16 ***
## X1          -0.06489    0.02824  -2.298   0.0242 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.675 on 79 degrees of freedom
## Multiple R-squared:  0.06264,    Adjusted R-squared:  0.05078
## F-statistic: 5.279 on 1 and 79 DF,  p-value: 0.02422
```

```
m3<-lm(Y~X1+X2)
summary(m3)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6473 -0.9041 -0.1574  0.4652  4.0687
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.24316    0.60116  20.366  < 2e-16 ***
## X1          -0.12559    0.02528  -4.967 3.92e-06 ***
## X2           0.40084    0.06492   6.175 2.79e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.382 on 78 degrees of freedom
## Multiple R-squared:  0.3704, Adjusted R-squared:  0.3543
## F-statistic: 22.94 on 2 and 78 DF,  p-value: 1.457e-08
```

```r
m4<-lm(Y~X1+X2+X3)
summary(m4)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1001 -0.7584 -0.0338  0.4415  3.8629
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.54066    0.67303  17.147  < 2e-16 ***
## X1          -0.11901    0.02491  -4.778 8.31e-06 ***
## X2           0.44611    0.06689   6.669 3.50e-09 ***
## X3           2.62039    1.22289   2.143   0.0353 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.351 on 77 degrees of freedom
## Multiple R-squared:  0.4058, Adjusted R-squared:  0.3827
## F-statistic: 17.53 on 3 and 77 DF,  p-value: 9.057e-09
```

```r
m5<-lm(Y~X1+X2+X3+X4)
summary(m5)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1872 -0.5911 -0.0910  0.5579  2.9441
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.220e+01  5.780e-01  21.110  < 2e-16 ***
## X1          -1.420e-01  2.134e-02  -6.655 3.89e-09 ***
## X2           2.820e-01  6.317e-02   4.464 2.75e-05 ***
## X3           6.193e-01  1.087e+00   0.570     0.57
## X4           7.924e-06  1.385e-06   5.722 1.98e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.137 on 76 degrees of freedom
## Multiple R-squared:  0.5847, Adjusted R-squared:  0.5629
## F-statistic: 26.76 on 4 and 76 DF,  p-value: 7.272e-14
```

```r
#m5$coef are the beta values (beta0,beta1,...,beta4)
```

```r
# Comparing R^2 values of the above models
summary(m1)[8]
```

```
## $adj.r.squared
## [1] 0
```

```r
summary(m2)[8]
```

```
## $r.squared
## [1] 0.06264236
```

```r
summary(m3)[8]
```

```
## $r.squared
## [1] 0.3703984
```

```r
summary(m4)[8]
```

```
## $r.squared
## [1] 0.4058292
```

```r
summary(m5)[8]
```

```
## $r.squared
## [1] 0.5847496
```

```r
# The multiple correlation coefficient is highest at model 5 (R^2 = 0.585), decreases at
↪  model 4, and keeps decreasing until it reaches 0 at model 1.
```

```r
# Null hypothesis (H0): beta_1 = beta_2 = beta_3 = beta_4
# Alternative hypothesis (H1): beta_j != 0, for at least one value of j

# F = MSM/MSE = (SSM/DFM) / (SSE/DFE)

anova(m1,m5)
```

```
## Analysis of Variance Table
##
## Model 1: Y ~ 1
## Model 2: Y ~ X1 + X2 + X3 + X4
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     80 236.558
## 2     76  98.231  4    138.33 26.756 7.272e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# calculating ss.diff
anova(m1,m5)[2]
```

```
##       RSS
## 1 236.558
## 2  98.231
```

```r
ss.res.null <- 236.558
ss.res.full <- 92.231

ss.diff <- ss.res.null - ss.res.full


# calculating F.stat
anova(m1,m5)[1]
```

```
##   Res.Df
## 1     80
## 2     76
```

```r
ms.res <- ss.res.full / 76
#ms.diff <- ss.diff / (80-76)
ms.diff <- ss.diff / 4

F.stat <- ms.diff / ms.res
F.stat
```

```
## [1] 29.73201
```

```r
# Finding the 95% CI for test statistic
# F-table value at alpha = 0.05, DFM = 4, DFE = 76
qf(0.95, 4, 76)
```

```
## [1] 2.492049
```

```r
# Conclusion:
# Since F.stat = 29.73201 is not in the range of qf(0.95, 4, 76) = [0, 2.492049], then
↪  the null hypothesis is rejected.
```

```r
# Assuming that alpha = 0.05, consider the p value of X2.
summary(m5)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.1872 -0.5911 -0.0910  0.5579  2.9441
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.220e+01  5.780e-01  21.110  < 2e-16 ***
## X1          -1.420e-01  2.134e-02  -6.655 3.89e-09 ***
## X2           2.820e-01  6.317e-02   4.464 2.75e-05 ***
## X3           6.193e-01  1.087e+00   0.570     0.57
## X4           7.924e-06  1.385e-06   5.722 1.98e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.137 on 76 degrees of freedom
## Multiple R-squared:  0.5847, Adjusted R-squared:  0.5629
## F-statistic: 26.76 on 4 and 76 DF,  p-value: 7.272e-14
```

```
# The p value of X2 is 2.75e-05. Since this value is greatly less than the significance
↪  level (alpha = 0.05), then this means that this predictor (X2) has a very
↪  statistically significant relationship with the t-value in model 5. This means that
↪  X2 cannot be dropped from model 5.
```

```
# By performing H0 : beta_1 = beta_2 = 0 on model 5, the X1 and X2 predictors are now
↪  removed.

model.Q1e<-lm(Y~X3+X4)

summary(model.Q1e)
```

```
##
## Call:
## lm(formula = Y ~ X3 + X4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1886 -0.7879  0.3140  0.9820  3.4021
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.376e+01  3.027e-01  45.469  < 2e-16 ***
## X3          3.007e-01  1.226e+00   0.245    0.807
## X4          8.407e-06  1.512e-06   5.561 3.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.47 on 78 degrees of freedom
## Multiple R-squared:  0.2871, Adjusted R-squared:  0.2688
## F-statistic:  15.7 on 2 and 78 DF,  p-value: 1.859e-06
```

```
# From the p-value of X3 being 0.807, it is easy to conclude at alpha = 0.05 that X3 is
↪  very statistically insignificant. The predictor X3 very well can be dropped.

# For the p-value of the entire model, it is now 1.859e-06 from the original p-value of
↪  model 5 at 7.272e-14. The p-value is still sufficiently lower than the significance
↪  level (alpha = 0.05), so the null hypothesis can still be rejected.
```

```r
# 95% CI for beta_0, beta_1, beta_2, beta_3, and beta_4

X<-matrix(0,n,p+1)
X<-cbind(rep(1,n),X1,X2,X3,X4)
#X

XTXinv<-solve(t(X)%*%X)
#XTXinv

sigmasqhat<-deviance(m5)/(n-(p+1))
alpha<-0.05

c(m5$coef[1]-qt(1-alpha/2,n-(p+1))*sqrt(sigmasqhat*XTXinv[1,1]),
  m5$coef[1]+qt(1-alpha/2,n-(p+1))*sqrt(sigmasqhat*XTXinv[1,1]))
```

```
## (Intercept) (Intercept)
##    11.04949    13.35169
```

```r
c(m5$coef[2]-qt(1-alpha/2,n-(p+1))*sqrt(sigmasqhat*XTXinv[2,2]),
  m5$coef[2]+qt(1-alpha/2,n-(p+1))*sqrt(sigmasqhat*XTXinv[2,2]))
```

```
##          X1          X1
## -0.18454113 -0.09952615
```

```r
c(m5$coef[3]-qt(1-alpha/2,n-(p+1))*sqrt(sigmasqhat*XTXinv[3,3]),
  m5$coef[3]+qt(1-alpha/2,n-(p+1))*sqrt(sigmasqhat*XTXinv[3,3]))
```

```
##        X2        X2
## 0.1561979 0.4078352
```

```r
c(m5$coef[4]-qt(1-alpha/2,n-(p+1))*sqrt(sigmasqhat*XTXinv[4,4]),
  m5$coef[4]+qt(1-alpha/2,n-(p+1))*sqrt(sigmasqhat*XTXinv[4,4]))
```

```
##        X3        X3
## -1.545232  2.783919
```

```r
c(m5$coef[5]-qt(1-alpha/2,n-(p+1))*sqrt(sigmasqhat*XTXinv[5,5]),
  m5$coef[5]+qt(1-alpha/2,n-(p+1))*sqrt(sigmasqhat*XTXinv[5,5]))
```

```
##           X4           X4
## 5.166283e-06 1.068232e-05
```

```r
#built-in CI function
#confint(m5)
```

```r
# The 95% Bonferroni Confidence interval for all beta

t<-qt(1-0.025/(p+1),n-(p+1))


c(m5$coef[1]-t*sqrt(sigmasqhat*XTXinv[1,1]),
  m5$coef[1]+t*sqrt(sigmasqhat*XTXinv[1,1]))
```

```
## (Intercept) (Intercept)
##    10.67358    13.72759
```

```r
c(m5$coef[2]-t*sqrt(sigmasqhat*XTXinv[2,2]),
  m5$coef[2]+t*sqrt(sigmasqhat*XTXinv[2,2]))
```

```
##          X1          X1
## -0.1984225 -0.0856448
```

```r
c(m5$coef[3]-t*sqrt(sigmasqhat*XTXinv[3,3]),
  m5$coef[3]+t*sqrt(sigmasqhat*XTXinv[3,3]))
```

```
##        X2        X2
## 0.1151102 0.4489228
```

```r
c(m5$coef[4]-t*sqrt(sigmasqhat*XTXinv[4,4]),
  m5$coef[4]+t*sqrt(sigmasqhat*XTXinv[4,4]))
```

```
##        X3        X3
## -2.252101  3.490788
```

```r
c(m5$coef[5]-t*sqrt(sigmasqhat*XTXinv[5,5]),
  m5$coef[5]+t*sqrt(sigmasqhat*XTXinv[5,5]))
```

```
##           X4           X4
## 4.265617e-06 1.158299e-05
```

```r
# The 95% Scheffe Confidence interval for all beta

sf<-sqrt((p+1)*qf(1-0.05,p+1,n-(p+1)))

c(m5$coef[1]-sf*sqrt(sigmasqhat*XTXinv[1,1]),
  m5$coef[1]+sf*sqrt(sigmasqhat*XTXinv[1,1]))
```

```
## (Intercept) (Intercept)
##    10.22582    14.17535
```

```r
c(m5$coef[2]-sf*sqrt(sigmasqhat*XTXinv[2,2]),
  m5$coef[2]+sf*sqrt(sigmasqhat*XTXinv[2,2]))
```

```
##          X1          X1
## -0.21495731 -0.06910998
```

```r
c(m5$coef[3]-sf*sqrt(sigmasqhat*XTXinv[3,3]),
  m5$coef[3]+sf*sqrt(sigmasqhat*XTXinv[3,3]))
```

```
##         X2         X2
## 0.06616854 0.49786452
```

```r
c(m5$coef[4]-sf*sqrt(sigmasqhat*XTXinv[4,4]),
  m5$coef[4]+sf*sqrt(sigmasqhat*XTXinv[4,4]))
```

```
##        X3        X3
## -3.094091  4.332778
```

```r
c(m5$coef[5]-sf*sqrt(sigmasqhat*XTXinv[5,5]),
  m5$coef[5]+sf*sqrt(sigmasqhat*XTXinv[5,5]))
```

```
##           X4           X4
## 3.192786e-06 1.265582e-05
```

```r
# beta_0 is obviously different from zero
# beta_1 is different from zero for all of the 3 CI tests
# beta_2 is different from zero for all of the 3 CI tests
# beta_4 is different from zero for all of the 3 CI tests

# beta_3 is obviously not different from zero and can potentially have zero correlation
```

```r
# X1 = 11, X2 = 8.97, X3 = 0.07, X4 = 60000


# create new data frame with new input
new_data = data.frame(X1=11,X2=8.97,X3=0.07,X4=60000)

predict(m5, new_data, interval="confidence", level = 0.95)
```

```
##        fit      lwr      upr
## 1 13.68672 13.28764 14.08579
```

```r
# The 95% confidence interval is [13.28764, 14.08579].
```

```r
predict(m5, new_data, interval="predict", level = 0.95)
```

```
##        fit      lwr      upr
## 1 13.68672 11.38751 15.98592
```

```
# The 95% prediction interval is [11.38751, 15.98592.
```