HOMEWORK -4

QUESTION 1 :

The 4 models and their co-efficient are given by

```
> coef(lm_trans_1)
(Intercept)          t1          t2
 144.369443    5.462057    2.034549
> coef(lm_trans_2)
(Intercept)           a           d
 144.369443    7.496605    1.713754
> coef(lm_trans_3)
(Intercept)          t2           d
 144.369443    7.496605    5.462057
> coef(lm_trans_4)
(Intercept)          t1          t2           a           d
 144.369443    5.462057    2.034549          NA          NA
```

Type equation here.

The co-efficients of response function of 4 are NA, since the column 4 and 5 corresponding to a and d in the equation are linearly dependent on columns 2 and 3. Since a = t1+t2/2 and d= t1-t2. Thus the matrix of XX' is not invertible, as it does not have full Rank. Hence the values are NA.

The aspects that are same in all the 4 models is that the intercept remains the same. The slopes of the lines keep changing, indicating the parameters keep changing.

The estimates are for t2 are same since they the same models.

QUESTION 2 :

- For the model $\log(fertility) \sim PctUrban$, the fitted equation is given by

$\hat{Y} = log(fertility) = 1.5 - (0.010163)PctUrban$

This is interpreted as the estimated co-efficient of PctUrban is beta1 = -0.010163.

For every unit increase in PctUrban , would result $(e^{0.010163} - 1) * 100 =$

1.011153 % in change in Y.

- For the model $\log(fertility) \sim \log(ppgdp) + lifeExp$ the fitted regression line is

$$\hat{Y} = 3.507 - 0.06544 * log(ppgdp) - (0.02824) lifeExp$$

For 25% increase in ppgdp , solving for for this equation we get

solving we get $\hat{Y} = ((1.25)^{0.06544} - 1) *\ 100$ we get the value of $\hat{Y} = -1.4\%$.

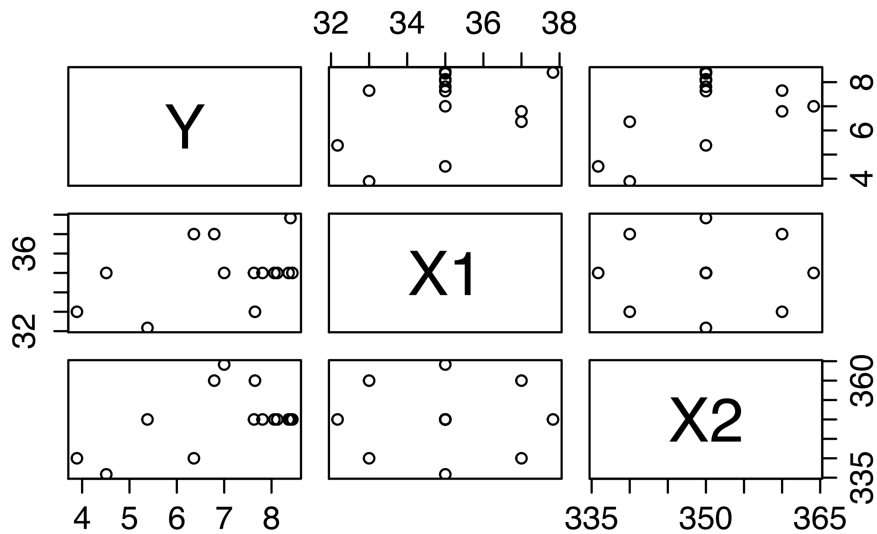
QUESTION 3:

Use cakes from alr4 package. The response variable is Y.

1.(a) Identify the best possible linear regression function using variables X1 and X2 (you can use as many regressors as you like). Show/explain modeling iterations.

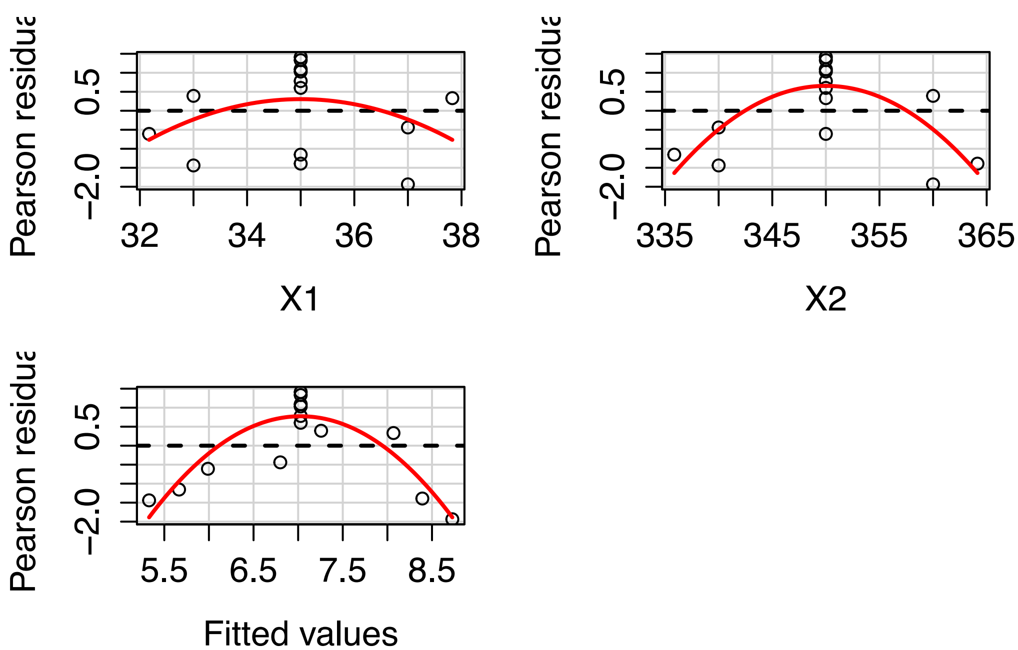
I initially started with the 2 variables X1 and X2 in the model.The correlation matrix is

```
> cor(df_ck)
           Y        X1        X2
Y  1.0000000 0.3882796 0.5091336
X1 0.3882796 1.0000000 0.0000000
X2 0.5091336 0.0000000 1.0000000
```



Noticed that only Y and X2 are highly correlated. Now I initially fitted the model,

The residualPlots is like this



Notice that the in the plot of residuals vs fitted there is Heteroscdasticity. I did a ncvTest.

```
> ncvTest(lm_ck_1)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.3363695    Df = 1     p = 0.5619323
```

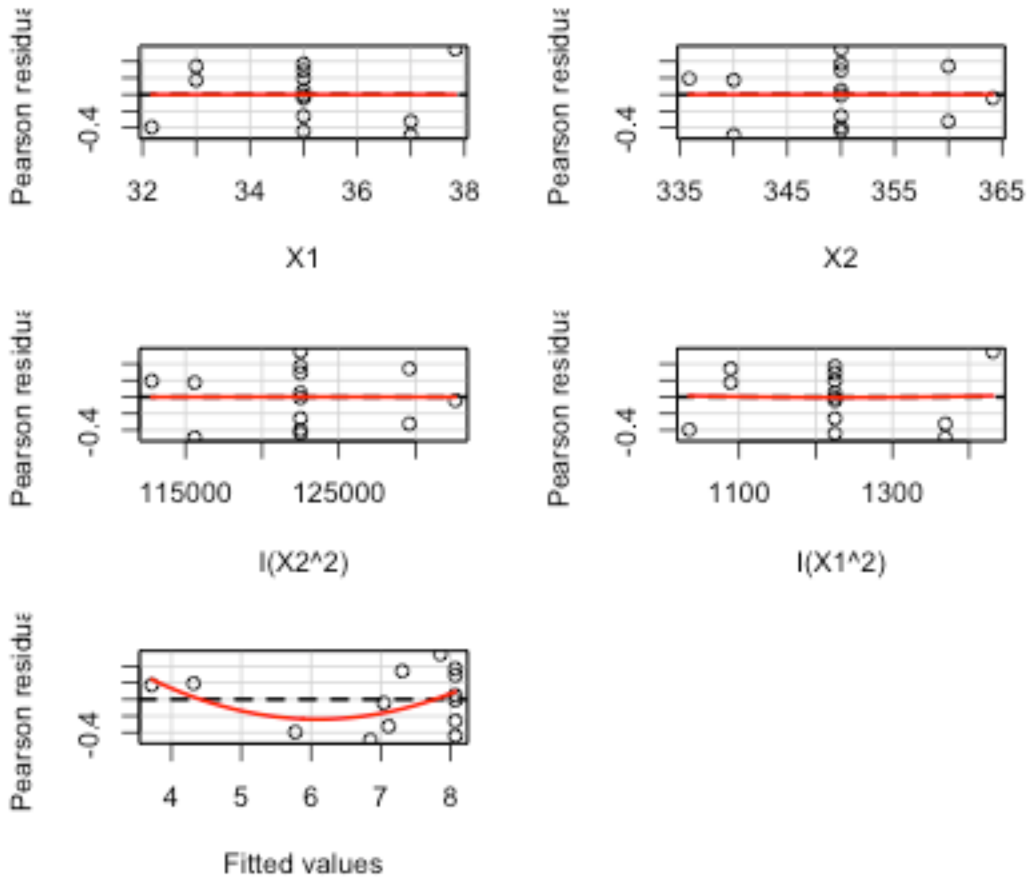
Also did check the normality of the error terms. They are not normal.

```
> shapiro.test(lm_ck_1$residuals)

        Shapiro-Wilk normality test

data:  lm_ck_1$residuals
W = 0.91398, p-value = 0.18
```

Then did a box-cox transformation of the Y variable, and subsequently, it did not improve the heteroscedasticity. However continued to do the Ramsey reset test, and hsowed that the model needs a quadratic term.

```
        RESET test

data:  lm_ck_1
RESET = 41.357, df1 = 1, df2 = 10, p-value = 7.529e-05
```

Finally arrived at the conclusion the model the bptest indicated these results p-value = 0.06484, which is on the border. Plots look like this.



The summary is

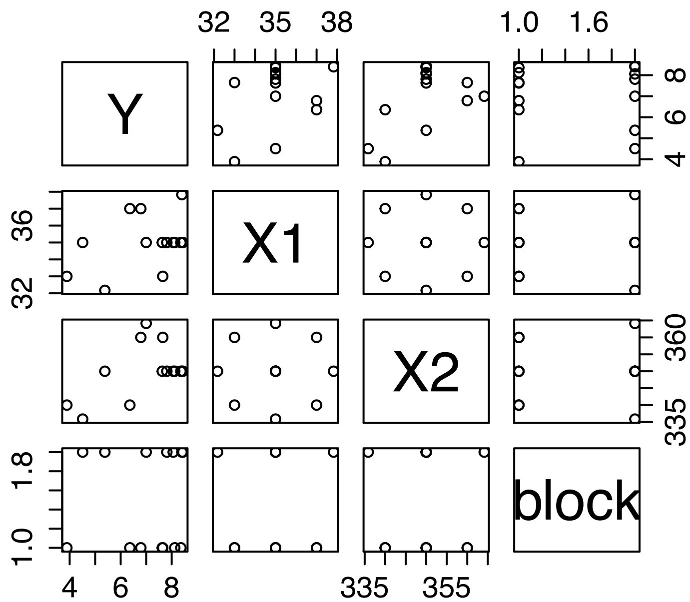
```
> summary(lm_ck_8)

Call:
lm(formula = Y ~ X1 + X2 + I(X2^2) + I(X1^2) + X1:X2, data =
df_ck)

Residuals:
    Min      1Q  Median      3Q     Max
-0.4912 -0.3080  0.0200  0.2658  0.5454

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.204e+03  2.416e+02  -9.125 1.67e-05 ***
X1           2.592e+01  4.659e+00   5.563 0.000533 ***
X2           9.918e+00  1.167e+00   8.502 2.81e-05 ***
I(X2^2)     -1.195e-02  1.578e-03  -7.574 6.46e-05 ***
I(X1^2)     -1.569e-01  3.945e-02  -3.977 0.004079 **
X1:X2       -4.163e-02  1.072e-02  -3.883 0.004654 **
```

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

Residual standard error: 0.4288 on 8 degrees of freedom
Multiple R-squared:  0.9487,  Adjusted R-squared:  0.9167
F-statistic:  29.6 on 5 and 8 DF,  p-value: 5.864e-05
```

2. (b) Repeat (a), but now include the dummy variable block (code the dummy variable 0/1). Show/explain modeling iterations. Is the coefcient of block significant?

I did the almost the same steps as above by including the dummy variable. The summary is

```
> summary(lm_ck_11)

Call:
lm(formula = Y ~ ., data = df_ck_1)

Residuals:
Min      1Q  Median      3Q     Max
-1.8805 -1.0759  0.3622  0.9118  1.3886

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) -39.63043   17.97805  -2.204   0.0521 .
X1            0.36756    0.22963   1.601   0.1405
X2            0.09639    0.04593   2.099   0.0622 .
block1        0.11429    0.69433   0.165   0.8725
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.299 on 10 degrees of freedom
Multiple R-squared:  0.4116,  Adjusted R-squared:  0.235
F-statistic: 2.331 on 3 and 10 DF,  p-value: 0.1359
```



The plot looks like this. The coefficient of block is very high = 0.8725. Adding an interaction term with the dummy variable, caused the difference between R square and R square adjusted to increase R-squared: 0.4135,   Adjusted R-squared: 0.1529 from initial values (R-squared: 0.4116, Adjusted R-squared: 0.235). The Ramsey reset test indicated an higher order term.

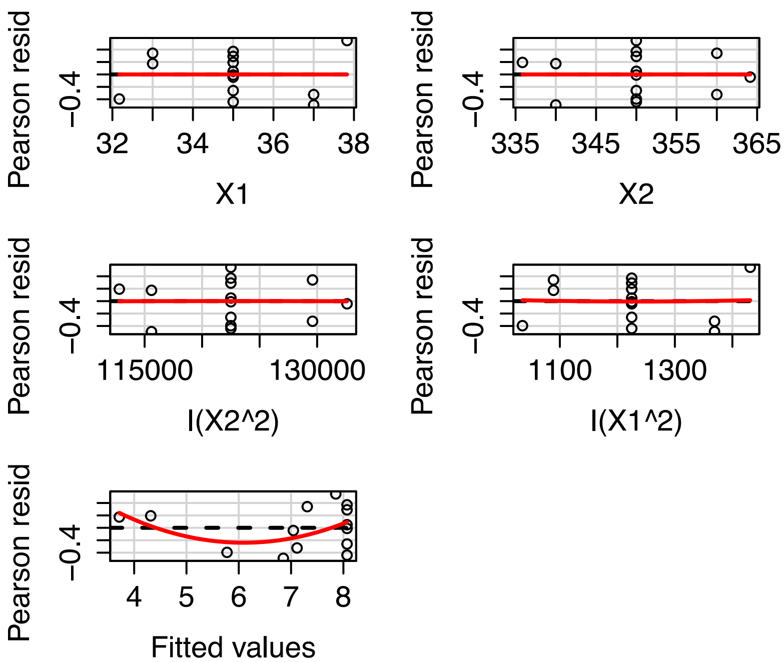
```
> resettest(lm_ck_11, power=2, type="fitted")

     RESET test
data:  lm_ck_11
RESET = 32.604, df1 = 1, df2 = 9, p-value = 0.0002906
```

This prompted me to include the quadratic term.

```
> lm_ck_16_2 <- lm(Y~X1+X2+block+I(X2^2)+I(X1^2)+X2:X1:block,
data=df_ck_1)
> summary(lm_ck_16_2)

Call:
lm(formula = Y ~ X1 + X2 + block + I(X2^2) + I(X1^2) +
X2:X1:block,
    data = df_ck_1)

Residuals:
     Min       1Q   Median       3Q      Max
-0.38286 -0.22952  0.00748  0.25131  0.34714

Coefficients:
```

```
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -2.196e+03  2.163e+02 -10.156 5.30e-05 ***
X1            2.580e+01  4.170e+00   6.187 0.000821 ***
X2            9.907e+00  1.044e+00   9.488 7.81e-05 ***
block1       -8.029e+00  4.253e+00  -1.888 0.107952    
I(X2^2)      -1.195e-02  1.412e-03  -8.462 0.000149 ***
I(X1^2)      -1.569e-01  3.531e-02  -4.443 0.004361 ** 
X1:X2:block0 -4.163e-02  9.594e-03  -4.338 0.004884 ** 
X1:X2:block1 -4.096e-02  9.601e-03  -4.266 0.005284 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3838 on 6 degrees of freedom
Multiple R-squared:  0.9692,  Adjusted R-squared:  0.9332 
F-statistic: 26.96 on 7 and 6 DF,  p-value: 0.0003986

> residualPlots(lm_ck_16_2)
           Test stat Pr(>|t|)
X1            -1.057    0.339
X2             0.222    0.833
block             NA       NA
I(X2^2)       -2.029    0.098
I(X1^2)        2.040    0.097
Tukey test     2.056    0.040
>
```

However the coefficient of the dummy variable is still high hence discarding it and getting the model.



This is the final model.

QUESTION 4:

The overall F-test is performed to test the regression relation between response variable Y and the set of predictor variables.

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_{p-1} = 0$$

$$H_a: not\ all\ \beta_k (k = 1 \ldots . p-1)\ equal\ zero$$

we use the test statistic $F^* = \frac{MSR}{MSE}$

The decision rule to control Type I error at α is:

$$If\ F^* \leq F(1-\alpha; p-1, n-p)\ conclude\ H_0$$

$$If\ \ F^* > F(1-\alpha; p-1, n-p)\ conclude\ H_a$$

In our case the number of predictor variable is p-1 = 6

The regression function is given by

$$\hat{Y} = -4.902e + 05 + 6.368e - 01x_1 + 7.690e + 00\, x_2 + 5.850e + 00\, x_3 + 4.562e + 01\, x_4 + -1.945e - 02\, x_5 + -2.087e + 04\, x_6$$

The F-statistic = 374.6 on 6 an d 44 DF and p-value less than significance level 0.05, conclude alternate hypothesis that there is a regression relation between predictors and response variable. The co-efficient /parameters are statistically significant from zero. Following is the R result.

```
> summary(lm_fuel)

Call:
lm(formula = Y ~ ., data = df_fuel)

Residuals:
     Min       1Q   Median       3Q      Max 
-1480910  -158802    19267   174208  1090089 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.902e+05  8.199e+05  -0.598 0.552983    
X1           6.368e-01  1.452e-01   4.386 7.09e-05 ***
X2           7.690e+00  1.632e+01   0.471 0.639793    
X3           5.850e+00  1.621e+00   3.608 0.000784 ***
X4           4.562e+01  3.565e+01   1.280 0.207337    
X5          -1.945e-02  1.245e-01  -0.156 0.876586    
X6          -2.087e+04  1.324e+04  -1.576 0.122235    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 398400 on 44 degrees of freedom
Multiple R-squared:  0.9808,	Adjusted R-squared:  0.9782 
F-statistic: 374.6 on 6 and 44 DF,  p-value: < 2.2e-16
```

```
> anova(lm_fuel)
Analysis of Variance Table

Response: Y
          Df     Sum Sq    Mean Sq   F value     Pr(>F)    
X1         1 3.5301e+14 3.5301e+14 2223.9334 < 2.2e-16 ***
X2         1 6.7563e+11 6.7563e+11    4.2563 0.0450357 *  
X3         1 2.1698e+12 2.1698e+12   13.6694 0.0006012 ***
X4         1 5.2927e+11 5.2927e+11    3.3343 0.0746401 .  
X5         1 2.1795e+10 2.1795e+10    0.1373 0.7127507    
X6         1 3.9416e+11 3.9416e+11    2.4832 0.1222348    
Residuals 44 6.9843e+12 1.5873e+11                        
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

QUESTION 5: Fit the model and compute the hypothesis tests.

(a) **H0 :β5 =0**

$$F^* = \frac{MSR\left(X_1X_2 \dots X_{k-1}X_{k+1} \dots X_{p-1}\right)}{MSE}$$

$$\mathrm{F} * = \frac{SSR(X_5|X_1X_2X_3X_4)}{n-5} \div \frac{SSE(X_1X_2X_3X_4\ X_5)}{n-6}$$

F* = 15.07962, and f-value is qf (0.95, 1, 8) = 5.317655. F* > f-value , hence conclude alternate hypothesis, and thus β5is statistically significant from zero. Also the p-value = 0.00465338 = pf(q=15.07962, df1=1, df2=8,lower.tail=FALSE) is less than alpha level. Hence conclude that beta5 is not equal to 0.

(b) H0 :β2 =0

$$\mathrm{F} * = \frac{SSR(X_3|X_1X_2X_5X_4)}{n-5} \div \frac{SSE(X_1X_2X_3X_4X_5)}{n-6}$$

F_stat_beta_2 =15.81669
F-value = qf(0.95,1,8) = 5.317655
p_value = pf(15.81669, 1, 8, lower.tail = FALSE) = 0.004078661

F* = 15.81669, and f-value is qf(0.95,1,8) = 5.317655. F* > f-value, hence conclude alternate hypothesis, and thus β2 is statistically significant from zero. Also the p-value = 0.004078661 is less than alpha level =0.05. Hence conclude that beta2 is not equal to 0.

(c) **H0 :β1 =β2 =β5 =0**

$$\mathrm{F}* = \frac{SSR(X_1|\,X_2X_2^2) + SSR(X_1^2|X_1X_2X_2^2) + SSR(X_1\,X_2\,|X_1X_1^2X_2X_2^2)}{n-5} \div \frac{SSE(X_1\,X_2\,X_1X_1^2X_2X_2^2)}{n-6}$$

F_stat =18.13758
F-value = qf(0.95,1,8) = 5.317655
p_value = pf(15.81669, 1, 8, lower.tail = FALSE) = 0.002765803
F* = 18.13758, and f-value is qf(0.95,1,8) = 5.317655. F* > f-value, **hence conclude alternate hypothesis**, and thus β1, β2 ,β5 are statistically significant from zero. Also the p-value = 0.002765803 is less than alpha level =0.05. Hence conclude that β1, β2, β5 are not equal to 0.

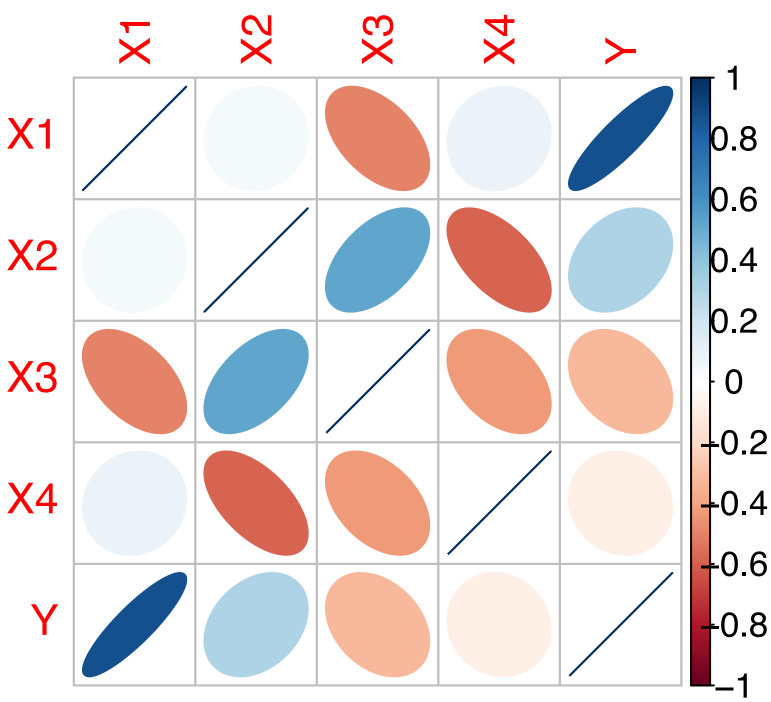
Here we check for the hypothesis. We get the results from the model as

```
> anova(lm_cake_1)
Analysis of Variance Table

Response: Y
          Df Sum Sq Mean Sq F value    Pr(>F)    
X2         1 7.4332  7.4332  40.432 0.0002186 ***
I(X2^2)    1 9.7685  9.7685  53.135 8.475e-05 ***
X1         1 4.3232  4.3232  23.515 0.0012730 ** 
I(X1^2)    1 2.9077  2.9077  15.816 0.0040789 ** 
X2:X1      1 2.7722  2.7722  15.079 0.0046537 ** 
Residuals  8 1.4707  0.1838                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
###β1 =β2 =β5 =0
#F_stat_beta125 = SSR(X1|X3X4) + SSR(X2|X1X3X4) +
SSR(X5|X1X2X3X4)/3/SSE(X1X2X3X4X%)/df_total
lm_cake_1 <- lm(Y~X2+I(X2^2)+X1+I(X1^2)+X1:X2, data = df_cake)
summary(lm_cake_1)
anova(lm_cake_1)
F_stat_beta235 <- ((4.3232 + 2.9077+ 2.7722)/3)/ (1.4707/8)
#18.13758
qf(0.95,1,8) # 5.317655
p_value <- pf(18.13758, 1, 8, lower.tail = FALSE) #0.002765803
#reject null hypothesis.
```

QUESTION 6:

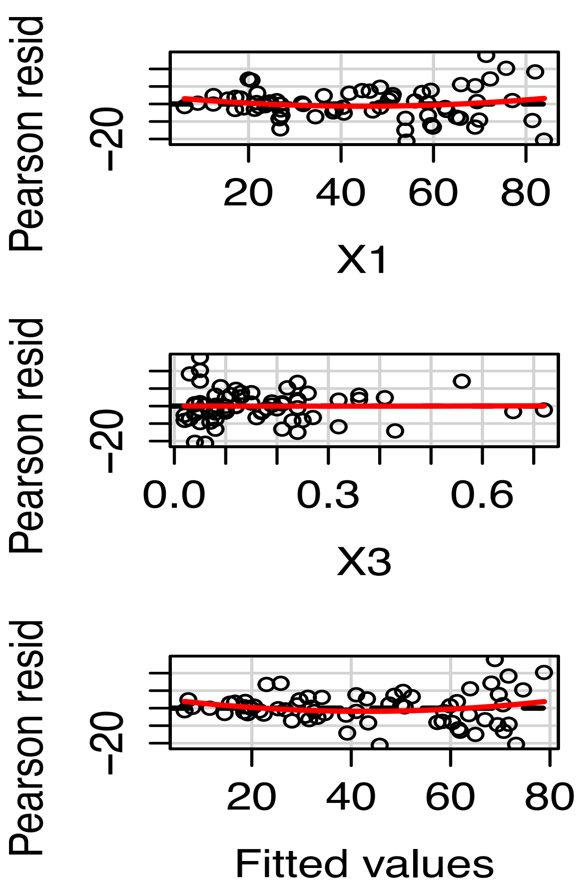
Initial analysis on the data conducted is as follows. Observed the variables are not highly correlated.



Initial model was checked for the p-values of the co-efficient. Noticed the p-vlaue of x3 and x4 is very high

```
Call:
lm(formula = Y ~ ., data = landrent)

Residuals:
     Min       1Q   Median       3Q      Max 
-21.2287  -4.8686  -0.0287   4.7547  27.7666 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -2.8282     4.6749  -0.605 0.547399    
X1            0.8833     0.0690  12.801  < 2e-16 ***
X2            0.4318     0.1080   3.999 0.000172 ***
X3          -11.3804    11.8937  -0.957 0.342359    
X4           -1.0117     2.8490  -0.355 0.723706    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.311 on 62 degrees of freedom
Multiple R-squared:  0.8404,	Adjusted R-squared:  0.8301
```

. The residual plots are as follows. The error terms appear to be random and no



Dropping X4 from the model. The new model evaluated had the results as follows. **mod1 <- lm(Y~X1+X2+X3, data=landrent).** The summary is as follows.

```
Call:
lm(formula = Y ~ X1 + X2 + X3, data = landrent)


Residuals:
     Min       1Q   Median       3Q      Max 
-21.4213  -5.0208  -0.0848   4.7825  28.3391 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -3.70912    3.93489  -0.943    0.349    
X1            0.88212    0.06845  12.888  < 2e-16 ***
X2            0.44890    0.09590   4.681 1.56e-05 ***
X3          -10.90999   11.73739  -0.930    0.356    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 9.246 on 63 degrees of freedom
Multiple R-squared:   0.84,   Adjusted R-squared:  0.8324 
F-statistic: 110.3 on 3 and 63 DF,  p-value: < 2.2e-16
```

Here there is an improvement in R square and R square adjusted. The ncvTest gave a non significant p-value. Hence no heteroscedasticity. Even the residual plots show similar results.

```
> ncvTest(mod1)
Non-constant Variance Score Test 
Variance formula: ~ fitted.values 
Chisquare = 20.21597    Df = 1     p = 6.917313e-06
```

Then I added the interaction term and noticed the R sqaure go up,

```
mod2 <- lm(Y~X1*X2*X3,data=landrent)
```

but the p-values of the coefficents of X3 terms indicated that they were non significant. Also did a Ramsey reset test, which showed the necessity of quadratic term

```
> resettest(mod1, power=2, type="regressor")
RESET test
data:  mod1
RESET = 3.108, df1 = 3, df2 = 60, p-value = 0.03299
```

After this tested the models with X2^2 and X1^2, noticed with X2^2 the R-square term improved significantly than with X1^2

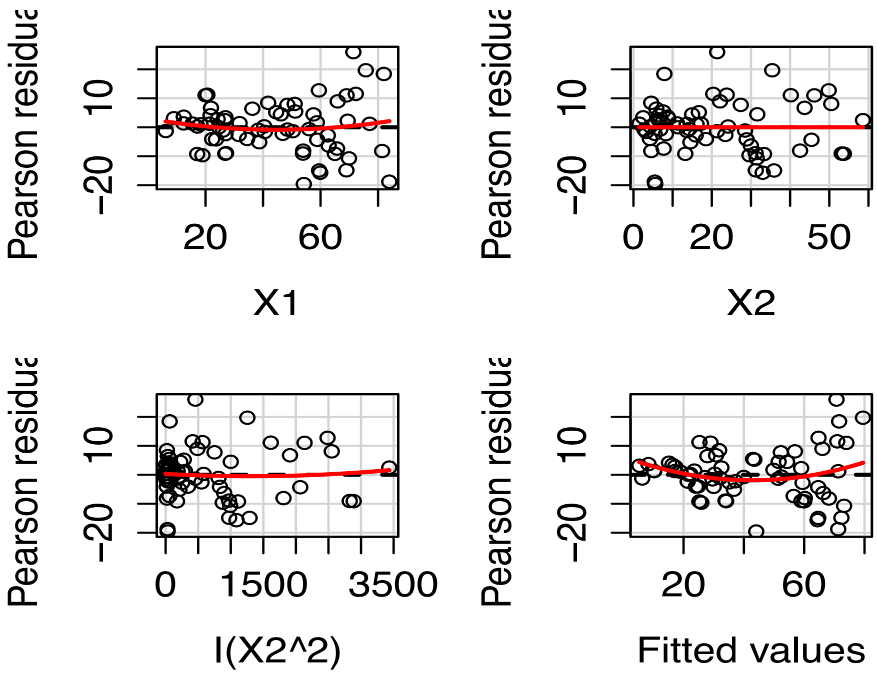
```
mod4 <- lm(Y~X1+X2+X3+I(X1^2),data=landrent)
(Multiple R-squared:  0.8438, Adjusted R-squared:  0.8337).

mod5 <- lm(Y~X1+X2+X3+I(X2^2),data=landrent)
Multiple R-squared:  0.8588,  Adjusted R-squared:  0.8496
```

Also dropping the X3 term caused a better residual plot and improved R-square and R-square adjusted.
Hence dropped X3 from the model. And came up with final model.

```
     mod6 <- lm(Y~X1+X2+I(X2^2),data=landrent)
>
summary(mod6)

Call:
lm(formula = Y ~ X1 + X2 + I(X2^2), data = landrent)

Residuals:
    Min      1Q  Median      3Q     Max 
-19.599  -4.736   0.261   4.457  25.951 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -10.794514   3.472330  -3.109 0.002820 ** 
X1            0.918023   0.051983  17.660  < 2e-16 ***
X2            0.990835   0.261847   3.784 0.000346 ***
I(X2^2)      -0.011423   0.004808  -2.376 0.020576 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.918 on 63 degrees of freedom
Multiple R-squared:  0.8512,  Adjusted R-squared:  0.8441 
F-statistic: 120.1 on 3 and 63 DF,  p-value: < 2.2e-16

> ncvTest(mod6)
Non-constant Variance Score Test 
Variance formula: ~ fitted.values
```

```
Chisquare = 22.96436    Df = 1     p = 1.650329e-06
```

The residual plots showed



QUESTION 7:The stepwise function does both forward and backward run through the system. The basic criteria to check is the p-value followed by R-square. Initially we assume all the numeric columns from the dataset in the model. After the model is created, the condition for backward stepwise is if the individual variable p-value is greater than alpha level. This indicates the coefficients are statistically insignificant, and are close to zero. Hence the dropping the variable one at a time. The main stopping criteria is that the p-value should be less than alpha = 0.05. After couple of iterations, the code performs forward stepwise for the removed items, adding one at a time. And then re-evaluate the model, based on these circumstances. Once all the forward are done, the results are compared, and the best model with minimum

p-value predictors are chosen, and final model is constructed. Here is output.

```
> stepwise(Rateprof)
Column Removed =  raterInterest
Column Removed =  sdRaterInterest
Column Removed =  sdQuality
Column Removed =  sdClarity
Column Removed =  sdHelpfulness
Column Removed =  numCourses
Column Removed =  numYears
Column Removed =  easiness

 Final list of columns retianed: numRaters quality helpfulness clarity sdEasiness
>
```

QUESTION 8:

The following code performs the functionality of the residualPlots function output with the results here.

R code:

```
#################################################################
#question 8

SetColumnNames = function(numOfCols){

  nameVector <- c()
  nameVector <- append(nameVector, 'Y')
  for(i in 1:numOfCols){
    nameVector <- append(nameVector, paste('X', i, sep=''))
  }
  return(nameVector)
}
msan601residualTest= function(dataSet) {
  df_frame <- dataSet
  num <- ncol(df_frame)-1
  names(df_frame) <- SetColumnNames(num)
  colIndex <- c(1:num)
  pow <- c(1.2, 1.5, 1.7)
  cat('','power','Test stat','Pr(>|t|)','\n')
  for(i in 1:num){
    for(j in 1:3){
      formula <- paste ('Y~.+','I(X',colIndex[i],'^',pow[j],')',
sep='')
```

```
      lm_fit <- lm(formula, data=df_frame)
      str_1 <- paste('X',i, sep='')
      p_val <- coef(summary(lm_fit))[,"Pr(>|t|)"][4]
      t_val <- coef(summary(lm_fit))[,"t value"][4]
      cat(str_1, " ", pow[j], "  " , t_val, "  ", p_val, "\n")
    }
  }
}
msan601residualTest(landrent)
#################################################################
```

R-results:

```
     power Test stat Pr(>|t|)
X1   1.2    0.149939    0.8813074
X1   1.5    0.1624437    0.8714934
X1   1.7    0.1730131    0.8632139
X2   1.2    0.75962    0.4504075
X2   1.5    0.7369639    0.4639694
X2   1.7    0.7198797    0.4743484
X3   1.2    0.495366    0.622093
X3   1.5    0.495366    0.622093
X3   1.7    0.495366    0.622093
X4   1.2    -0.7251012    0.4711624
X4   1.5    -0.6628539    0.5099212
X4   1.7    -0.6172701    0.539354
```

```
################################################################
```

QUESTION 9:

The following code performs the functionality of the computing the partial coefficient of determination for an input. function output with the results here.

R code:

```
#################################################################
#question 9

msan601partials = function(dataSet){

  df_ld <- data.frame(dataSet)
  num_x_cols <- ncol(df_ld) - 1
  names(df_ld) <- SetColumnNames(num_x_cols)
  #Type 1 Print the single values
```

```
for( i in 1:num_x_cols){
  formula = paste('Y~','X',i, sep='')
  fit <- lm(formula, data=df_ld)
  summary(fit)$r.squared
  str = paste('X',i,' = ',summary(fit)$r.squared,sep='')
  print(str)
}
#iterate through all the possible combinations
for(k in 1:num_x_cols){
  num_sel_cols <- k
  comb = combinations(num_x_cols, num_sel_cols)
  n_comb_rows <- nrow(comb)
  comb_vec<- NULL
  #create a vector of combination values
  for(h in 1:nrow(comb)){
    st <- paste(comb[h,], collapse = '#')
    comb_vec <- append(comb_vec , st)
  }
  #This iterates through all the possible columns selected
  for(i in 1:num_x_cols){
    for(j in 1:n_comb_rows){
      found = FALSE
        vec <- strsplit(comb_vec[j], "#")
        vec <- unlist(vec)
        for(elem in vec){
          if(i == elem){
            found = TRUE
            break;
          }
        }
      #Get the vector of variables for the formula
      if(!found){
        col_vec <- vector('character')
        col1 <- paste('X', i, sep='')
        col_vec[1] <- col1
        vec1 <- strsplit(comb_vec[j], "#")
        vec1 <- unlist(vec)
        ik<-2
        for(ele in vec1){
          col_vec[ik] <- paste('X', ele, sep='')
          ik <- ik+1
        }
        #Construct the formula
        str_1 = 'Y~'
        str_2 <- paste(col_vec, collapse='+')
        formula_str <- paste(str_1, str_2, sep='')
```

```
          #Calculate the partial coefficient of determination
          fit <- lm(formula_str, data=df_ld)
          numer = Anova(fit)$'Sum Sq'[1]
          denom = Anova(fit)$'Sum Sq'[length(Anova(fit)$'Sum
Sq')]
          val = numer/denom
          str_comb_vec <- paste(vec1, collapse='')
          str <- paste('X', i, '|', str_comb_vec ,' = ',
val,sep='')
          print(str)
        }
      }
    }
  }
}
msan601partials(landrent)
#################################################################
```

R-results:

```
> msan601partials(landrent)
[1] "X1 = 0.00237352346437981"
[1] "X2 = 0.249822713247114"
[1] "X3 = 0.00791442270644653"
[1] "X4 = 0.766977048038946"
[1] "X1|2 = 0.213811649373991"
[1] "X1|3 = 0.0157157989107288"
[1] "X1|4 = 0.303347035974544"
[1] "X2|1 = 0.614192618633298"
[1] "X2|3 = 0.35666015416507"
[1] "X2|4 = 0.290127518772281"
[1] "X3|1 = 0.0213886753531955"
[1] "X3|2 = 0.025854681078748"
[1] "X3|4 = 0.136931037504051"
[1] "X4|1 = 4.57993751370786"
[1] "X4|2 = 3.15334349450512"
[1] "X4|3 = 3.84043685479394"
[1] "X1|23 = 0.185799447770621"
[1] "X1|24 = 0.0627366058403004"
[1] "X1|34 = 0.15576379538737"
[1] "X2|13 = 0.583835619517361"
[1] "X2|14 = 0.0519575389804326"
[1] "X2|34 = 0.161906907385185"
[1] "X3|12 = 0.00218012814711426"
[1] "X3|14 = 0.00819175149069459"
[1] "X3|24 = 0.0239360113437986"
[1] "X4|12 = 2.63640452002227"
```

```
[1] "X4|13 = 4.50784154054617"
[1] "X4|23 = 3.14557544059969"
[1] "X1|234 = 0.0420014134054361"
[1] "X2|134 = 0.047539855957434"
[1] "X3|124 = 0.00395786236546969"
[1] "X4|123 = 2.6428550178574"
```

##