Finals

Sharad Bajpai

12/1/2019

Installing libraries required in this final assignment

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------------
```

```
## v ggplot2 3.2.0     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

1. You roll five six-sided dice. Write a script in R to calculate the probability of getting between 15 and 20 (inclusive) as the total amount of your roll (ie, the sum when you add up what is showing on all five dice). Exact solutions are preferable but approximate solutions are ok as long as they are precise.

```
set.seed(5)
dice.sum<-function(n.dice, n.sides){ #number of dice 5 and 6 sides of the dice

 dice<-sample(1:n.sides,size =n.dice,replace=T)
 return(sum(dice))
  }
R<-replicate(5,dice.sum (5,6))
R
```

```
## [1] 10 18 19 19 14
```

```
sum((15<= R) & (R<=20))/length(R)
```

```
## [1] 0.6
```

Probability is the number of outcomes / total number of possible outcomes

For R: 10 18 19 19 14.

The probability of getting between 15 and 20 (inclusive) is 0.6

2. Create a simulated dataset of 100 observations, where x is a random normal variable with mean 0 and standard deviation 1, and $y = 0.1 + 2 * x + \epsilon$, where epsilon is also a random normal error with mean 0 and sd 1. (One reminder: remember that in creating simulated data with, say, 100 observations, you need to use rnorm(100) for epsilon, not rnorm(1), to ensure that each observation gets a different error.)

```
set.seed(1000)
n <- 100
x <- rnorm(n, 0, 1) # n: number observations, mean=0, sd=1
e <- rnorm(n, 0, 1) #e: epsilon, mean=0, sd=1
y <- 0.1 + 2*x + e
```

a. Perform a t test for whether the mean of Y equals the mean of X using R.

Null Hypothesis: H0 : The mean of Y is equal to the mean of X: Y = X

Research Hypothesis: H1 : The mean of Y is not equal to the mean of X: Y 6= X

Here, X and Y are not two distinct samples but Y is dependant on X because,

y = 0.1 + 2 * x + e. so we will perform a paired sample t-test, also called the dependent sample t-test

```
t.test(x, y, paired =TRUE)
```

```
##
##  Paired t-test
##
## data:  x and y
## t = -1.6523, df = 99, p-value = 0.1016
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.47715688  0.04354888
## sample estimates:
## mean of the differences
##               -0.216804
```

INTERPRETATION

The t-test is insignificant as the p-value is greater than 0.05, that means we fail to reject null hypothesis which is, the mean of Y is equal to mean of X.

b. Now perform this test by hand using just the first 5 observations. Please write out all your steps in latex.

The following is our dataset of X and Y:

```r
D1<-data.frame(x,y)
head(D1,5)
```

```
##             x           y
## 1 -0.44577826 -2.16096874
## 2 -1.20585657 -2.14869630
## 3  0.04112631  0.06902645
## 4  0.63938841  2.23853210
## 5 -0.78655436 -1.34444325
```

```r
X_2<-c(-0.44577826,-1.20585657,0.04112631,0.63938841,-0.78655436)
Y_2<-c(-2.16096874,-2.14869630 ,0.06902645 ,2.23853210 ,-1.34444325)
X_2sum<-sum(-0.44577826,-1.20585657,0.04112631,0.63938841,-0.78655436)
X_2mean<-X_2sum/5
X_2mean
```

```
## [1] -0.3515349
```

```r
Y_2sum<-sum(-2.16096874,-2.14869630 ,0.06902645 ,2.23853210 ,-1.34444325)
Y_2mean<-Y_2sum/5
Y_2mean
```

```
## [1] -0.6693099
```

In latex: We will use the first 5 observations from x and y. Given: For X: n=5 For Y: n=5 In Latex; Step.1 Calculate the mean for X and Y (First Five Observations)

$Mean_x = \frac{(0.44577826)+(1.20585657)+0.04112631+0.63938841+(0.78655436)}{5} = \frac{-1.756}{5} = -0.3512$

$Mean_y = \frac{(2.160)+(2.148)+(0.0690)+(2.238)+(1.344)}{5} = \frac{-3.345}{5} = \frac{-3.345}{5} = -0.669$

Step.2 Calculate Standard deviation of X and Y (First Five Observations)

$$s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Standard Deviation of X: $s_X = \sqrt{\frac{1}{5-1}(0.44(0.35))2 + (1.20(0.35))2 + (0.04(0.35))2 + (0.63(0.35))2 + (0.78(0.35))^2}$

```r
sd(X_2)
```

```
## [1] 0.718349
```

Standard Deviation of Y:

$$s_Y = \sqrt{\frac{1}{5-1}(-2.16(-0.66))^2 + (2.14(0.66))^2 + (0.069(0.66))^2 + (2.23(0.66))^2 + (1.34(0.66))^2}$$

Step 3: Calculate Standarad errors for X and Y For X:

$$se_X = \frac{s}{\sqrt{n}} = \frac{0.7178337}{\sqrt{5}} = 0.321$$

For Y:

$$se_Y = \sqrt{se_x^2 + se_Y^2} = \sqrt{(0.321)^2 + (0.8327703)^2} = 0.892495$$

Step 4: Calculate Difference in standard error

$$se_diff = \sqrt{(0.321)^2 + (0.8327703)^2}$$

Step 5: Calculate degree of freedom Now we will calculate degrees of freedom, as the n is same but s is different we will use:

$$df = \frac{se_{diff}^4}{\frac{se_X^4}{(n_1-1)} + \frac{se_Y^4}{n_2-1}}$$

$$df = \frac{0.633}{\frac{0.010}{4} + \frac{0.479}{4}} = \frac{0.815}{0.002 + 0.119} = 6.7355 \approx 7$$

Step 6: Calculate Test Statistics

Now we will calculate Test statistics, before that we need: Difference in mean for a paired t-test

$$Mean_X - Mean_Y = Difference\,of\,Mean$$

$$-0.3512 - (-0.699) = 0.3178$$

$$T.S. = \frac{\bar{x}_X - \bar{y}_Y}{se_{diff}} = \frac{0.3178}{0.8924} = 0.3561183$$

Step 7: Calculate the Threshold Value

Now, we will calculate the Threshold values in R:

```
UpperT<-qt(0.975,7)
UpperT
```

```
## [1] 2.364624
```

```
LowerT<-qt(0.025,7)
LowerT
```

```
## [1] -2.364624
```

INTERPRETATION

In our case, the Test statistics does not fall under the Critical region at 95% Confidence interval, so here we are unable to reject the null hypothesis. So, our test is in the favor of our Null Hypothesis i.e. The mean of Y is equal to the mean of X. t-distributions are defined by the DF, which are closely associated with sample size. As the DF increases, the probability density in the tails decreases and the distribution becomes more tightly clustered around the central value. Since we have a small sample, the probability that the sample statistic will be further away from the null hypothesis is greater even when the null hypothesis is true.

c. Using R, test whether the mean of Y is significantly different from 0.

For the T-test in R:

```
t.test(X_2,Y_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  X_2 and Y_2
## t = 0.35602, df = 5.1647, p-value = 0.7359
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.954854  2.590404
## sample estimates:
##  mean of x  mean of y
## -0.3515349 -0.6693099
```

INTERPRETATION Here, as we can see the t.test results match with the manual test results. Also, the p-value is more than 0.05 hence the mean of Y is not significantly equal to 0.

d. Again using the first five obsevations, test by hand whether the mean of Y is different from 0.

Null Hypothesis H0 : $\mu_Y = 0$ Research hypothesis H_a : $\mu_Y \neq 0$ Now we will calculate the Test statistics, df, Threshold/ Confidence intervals Step 1: For Test statistics: Mean of Y $\bar{y} = -0.669$, $\mu_0 = 0$ Standard error of Y $se_Y = 0.832$

$$Test\ Statistics = \frac{\bar{y} - \mu_0}{se_Y} = \frac{-0.669 - 0}{0.832} = -0.804$$

Step 2: For df in one sample t-test:

$$df = n - 1 = 5 - 1 = 4$$

Step 3:For Threshold:

```
Ut<-qt(.975,4)
Ut
```

```
## [1] 2.776445
```

```
Lt<-qt(.025,4)
Lt
```

```
## [1] -2.776445
```

Step 4: For a two-tailed test, p-value:

```
2*(pt(-0.804,4))
```

```
## [1] 0.4664607
```

We can check the values in R:

```
t.test(Y_2, mu = 0, alternative = "two.sided")
```

```
##
##  One Sample t-test
##
## data:  Y_2
## t = -0.80372, df = 4, p-value = 0.4666
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -2.981450  1.642831
## sample estimates:
##  mean of x
## -0.6693099
```

INTERPRETATION

Here, we can see that our sample mean does not fall under the rejection region therefore, we fail to reject Null hypothesis that is $\mu = 0$. The p-value is greater than 0.05 therefore, it is not statistically significant that the mean of Y is different from 0.

e. Assuming the mean and sd of Y that you calculate from the first five observations would not change, what is the minimum total number of observations you would need to be able to conclude that the mean of Y is different from 0 at the p = 0.01 confidence level?

Given, $\bar{y} = -0.6693099$ p=0.01 confidence interval mean, 99% of CI Since, sd, $\bar{y}$, are all same for any sample then: Here, we will create a loop: For minimum n

```
for(n in 1:1000000){
UTh = -0.6693099 + qnorm(0.995) * (1.862131/sqrt(n))
if(UTh < 0){
return(n)
break}
}
n
```

```
## [1] 52
```

INTERPRETATION

Here, we created a function that looped through all the n's but we made it to stop when it reaches the first confidence interval that no longer includes zero (qnorm(0.995),for 99% CI). This is because we know that confidence intervals that contain zero are associated with insignificant effects. This has given us minimum total number of observations (n=52) would be need to be able to conclude that the mean of Y is different from 0 at the p = 0.01 confidence level.

f. Verify (d) (approximately) by increasing the simulated data to the n you calculated in (e) that would be necessary. If the test of Y = 0 is still not significant, explain why. (Go back to using the original 100-observation dataset for g and h.)

Null Hypothesis: $H_0 : \mu_Y = 0$

Research Hypothesis: $H_a : \mu_Y \neq 0$

Given, $\bar{y} = -0.6693099$ $s_Y = 1.862131$ $n = 52$

$$\sqrt{n} = \sqrt{52} = 7.211103$$

$$Test\ Statistics = \frac{\bar{y} - \mu_0}{\frac{s_Y}{\sqrt{n}}}$$

$$Test\ Statistics = \frac{-0.6693099 - 0}{\frac{1.862131}{7.211103}} = -2.591903$$

For df:

$$df = n - 1 = 52 - 1 = 51$$

For Threshold:

```
thU<-qt(0.975,51)
thU
```

```
## [1] 2.007584
```

```
thL<-qt(0.025,51)
thL
```

```
## [1] -2.007584
```

P-value for a two-tailed test:

```
2*(pnorm(-2.591903, lower.tail=T))
```

```
## [1] 0.009544668
```

INTERPRETATION

From the above tests, we can see that the Test statistics falls under the critical region .But, the p-value is less than 0.05 that means there is a statistical significant difference between means. So, we can reject the Null hypothesis and our test is in favour of the Research hypothesis that is, $H_a : \mu_Y \neq 0$

g. Create a categorical (factor) variable c, where c = 1 if x < -1, c = 3 if x > 1, and c = 2 otherwise. Use R to perform an F test for whether the mean of y differs across these three groups.

Creating categorical- factor variable:

```
c<- numeric(length(x))
for (i in seq_along(x)){
if (x[i] < -1) {c[i]<- 1}
else if (x[i]>1) {c[i]<- 3}
else if ({(x[i]>=-1) && (x[i]<=1)}) {c[i]<- 2}
}
c # to check the values of c
```

```
##   [1] 2 1 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 2 3 1 2 2 2 2 1 2 2 3 2 2 2 2 1
##  [36] 2 2 3 2 1 2 3 2 2 2 3 2 2 1 2 2 1 2 2 2 2 1 2 3 2 2 2 3 3 2 2 2 3 2 2
##  [71] 1 2 2 3 2 2 2 3 2 2 3 2 2 3 2 2 2 1 1 3 3 1 2 2 2 2 2 2 2 3
```

```
new_data<- data.frame(y,x,c)
head(new_data)
```

```
##             y           x c
## 1 -2.16096874 -0.44577826 2
## 2 -2.14869630 -1.20585657 1
## 3  0.06902645  0.04112631 2
## 4  2.23853210  0.63938841 2
## 5 -1.34444325 -0.78655436 2
## 6 -0.47800735 -0.38548930 2
```

Here, there are 3 groups now so we will do a F-test. Therefore,

$Null\ Hypothesis\ H_0$: There is no significant difference between mean of y across these groups. That is, $_{c1} = _{c2} = _{c3}$

$Research\ Hypothesis\ H_1 : Meanofyacrossthesegroupsisdifferent.$

```
new_data$c<- as.factor(new_data$c) #converting as factors
```

F-test:

```
aov.ex<- aov(y~c,data=new_data)
summary(aov.ex)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## c            2  295.7   147.8   86.89 <2e-16 ***
## Residuals   97  165.0     1.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

INTERPRETATION:

It can be inferred from this test that the p-value is less than 0.05, this test is statistically significant and that means we can reject Null hypothesis and our test is in favor of the research hypothesis i.e. the mean of y differs across these groups (c=1,2,3)

h. Using the first three observations for each group, calculate the same F test by hand.

First 3 observations of group 1,2,3 :

```
sample_data<-new_data[order(new_data$c,new_data$y),]
by(sample_data,sample_data["c"],head,n=3)#where n=3
```

```
## c: 1
##            y         x c
## 92 -4.903958 -1.920995 1
## 57 -4.479195 -1.790323 1
## 27 -4.301346 -1.783844 1
## ------------------------------------------------------------
## c: 2
##            y          x c
## 39 -2.773969 -0.7416215 2
## 60 -2.700590 -0.8608378 2
## 69 -2.439273 -0.9569277 2
## --------------------------------------------------------
## c: 3
##            y        x c
## 38 0.7313354 1.057601 3
## 30 1.8662525 1.220936 3
## 74 2.0608660 1.388866 3
```

Null Hypothesis: $H_0 : \mu_1 = \mu_2 = \mu_3$: All group means are the same Research Hypothesis: H_1 Atleast one is different

```
m_y1<-c(-7.135465,-5.179625,-4.162452)
mean(m_y1)# mean of y for group 1 where,c=1
```

```
## [1] -5.492514
```

```
m_y2<-c(-3.141609,-3.076083,-2.982907)
mean(m_y2)#mean of y for group 2 where,c=2
```

```
## [1] -3.066866
```

```
m_y3<-c( 0.8939747,2.2208137,2.4733397)
mean(m_y3)#mean of y for group 2 where,c=3
```

```
## [1] 1.862709
```

$\bar{y_1}$=-5.495214 $\bar{y_2}$=-3.066866 $\bar{y_3}$=1.862709

Mean of whole samples pooled together:

```
mean_full_y<-c(-7.135465,-5.179625,-4.162452,
-3.141609,-3.076083,-2.982907,
0.8939747,2.2208137,2.4733397)

mean(mean_full_y)
```

```
## [1] -2.232224
```

Standard deviation of y_1, y_2, y_3:

```
sd(m_y1) #s of y1
```

```
## [1] 1.511002
```

```
sd(m_y2)#s of y2
```

```
## [1] 0.07975144
```

```r
sd(m_y3)#s of y3
```

```
## [1] 0.848397
```

And, $\bar{y} = -1.307234$ $n_1 = 3$ $N = 9$ $G = 3$

Test Statistics Formula:

$$F - Statistics = \frac{Average\ variance\ between\ groups}{Average\ variance\ within\ groups}$$

Step.1 Between Variance and Within Variance $Between\,Variance(BV)$:

$$Between\ groups\ \ = \frac{n_1(\bar{y_1} - \bar{y})^2 + ...n_G(\bar{y_G} - \bar{y})}{df = G - 1}$$

The higher the numerator, the more different they are; but the higher the denominator, the less significant that difference is.

$$(B.V.) = \frac{3(5.49(2.23))^2 + 3(3.06(2.23)^{)}2 + 3(1.86(2.23))^2}{3-1} = 42.0669$$

$Within\,Variance\ (W.V.)$

$$Within\,Variance\ \ = \frac{(n-1)s_1^2 + ... + (n_G - 1)s_G^2}{df = N - G}$$

$$(W.V.) = \frac{(3-1) * (1.51)^2 + (3-1) * 0.079^2 + (3-1) * 0.84^2}{9-3}$$

$$F - Statistics = \frac{BV}{WV} = \frac{42.0669}{0.9973137} = 42.18021$$

Step 2: Degrees of freedom:

$$df_1 = G - 1 = 2$$

$$df_2 = N - G = 6$$

Step 3: Calculating the F threshold

```r
qf(0.95,2,6)
```

```
## [1] 5.143253
```

Step.4: p-values

```r
pf(42.18021, 2, 6, lower.tail=F)
```

```
## [1] 0.0002927649
```

INTERPRETATION

Here we can conclude that our $\mathrm{F} - \mathrm{statistic} > \mathrm{F} - \mathrm{threshold}$: 42.18021> 5.143253 that means our F-statistic falls under the critical region. Also, that our test is in favour of Research hypothesis (atleast one group is different) and we can reject the null hypothesis that is means of all the groups mean are same. Also, our p value is less than 0.05 that helps us to reject the Null Hypothesis.

3. Generate a new 100-observation dataset as before, except now $y = 0.1 + 0.2x + \epsilon$

```r
set.seed(100)
number <- 100
x3 <- rnorm(number, 0, 1) # n: number observations, mean=0, sd=1
e_3 <- rnorm(number, 0, 1) #e: epsilon, mean=0, sd=1
y3 <- (0.1 + 0.2*x3 + e_3)
D3<-data.frame(x3,y3)
head(D3)
```

```
##            x3         y3
## 1 -0.50219235 -0.3333618
## 2  0.13153117  1.4894199
## 3 -0.07891709 -0.3849308
## 4  0.88678481  1.1202326
## 5  0.11697127 -1.3345995
## 6  0.31863009 -0.2365799
```

a. Regress y on x using R, and report the results.

Null Hypothesis: H0 : There is no effect of x on y. That is, $\beta_1 = 0$

Research Hypothesis: H1 : There is an effect of x on y. That is, $\beta_1 \neq 0$

```r
BV1<-lm(y3~x3, data = D3)
summary(BV1)
```

```
##
## Call:
## lm(formula = y3 ~ x3, data = D3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05195 -0.43265 -0.07854  0.48583  1.93858
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.11145    0.07929   1.406    0.163
## x3           0.09463    0.07807   1.212    0.228
##
## Residual standard error: 0.7929 on 98 degrees of freedom
## Multiple R-squared:  0.01477,    Adjusted R-squared:  0.004717
## F-statistic: 1.469 on 1 and 98 DF,  p-value: 0.2284
```

INTERPRETATION

Here, Our p-value >0.05, suggesting our test is NOT statistically significant. And, we fail to reject the Null hypothesis that our coefficient is significantly different from 0, hence our test is in favor of the Null hypothesis.

b. Discuss the coefficient on x and its standard error, and present the 95% CI.

This regression is insignificant with the coefficient $0 = 0.20$, there is a no statistical significant linear dependence of mean of y on x xan be seen.

The standard error for 1 = 0.07 means that the average distance of the data points from the best fitted line is about 7% of y.

The thresholds can be calculated in R:

Here, df = n − k − 1 = 100 − 1 − 1 = 98

```
Ul<-qt(.975,98)
Ul
```

```
## [1] 1.984467
```

INTERPRETATION CI fails to exclude zero, based on which we can again not reject the null hypothesis that the coefficient is significantly different from zero.

c. Use R to calculate the p-value on the coefficient on x from the t value for that coefficient. What does this p-value represent (be very precise in your language here)?

From the Bv test, T est statistics = 1.212 df = n − k − 1 = 100 − 1 − 1 = 98

P-value can be calculated in R:

```
P_val=2*(1-pt(1.212,98))
P_val
```

```
## [1] 0.2284264
```

INTERPRETATION Considering the 95% confidence level with a two-tailed test and alpha(0.05), we got P-value(0.2284264) which is greater than alpha(0.05). Hence, we can conclude that we cannot reject Null and the result for this trial is not statistically significant.

d. Discuss the F-statistic and its p-value, and calculate that p-value from the F statistic using R. What does this test and its p-value indicate?

F-test is another way to measure the overall significance of the model. F statistics is related to the R2, since both are measures of overall model fit.

$$F = \frac{\frac{R^2}{k}}{\frac{1-R^2}{(n-k-1)}}$$

From the BV test, R2 = 0.01477

```
r2<-0.01477
k<-1
n3<-100
```

k : Number of Independant variables k = 1 n = 100 Therefore: F can be:

```
f <- ((r2/k) / ((1-r2)/(n3-k-1)))
f
```

```
## [1] 1.469159
```

For p-value:

```
pf(f,k,(n3-k-1),lower.tail=F)
```

```
## [1] 0.2283926
```

INTERPRETATION As we can see $\text{p} - \text{value} = 0.2283926$, it is greater than 0.05 which suggests that this F-test is not statistically significant and our model does not predicts the dependent variable better than the dependent variable alone.

e. Using the first five observations, calculate by hand the coefficient on x, its standard error, and the adjusted R2. Be sure to show your work.

Our hypothesis: Null Hypothesis: H_0: There is no effect of X on Y such that $\beta_1 = 0$. Research Hypothesis: H_1: There is an effect of X on Y such that $\beta_1 \neq 0$. First 5 observations:

```
D3<-data.frame(x3,y3)
head(D3, 5)
```

```
##            x3         y3
## 1 -0.50219235 -0.3333618
## 2  0.13153117  1.4894199
## 3 -0.07891709 -0.3849308
## 4  0.88678481  1.1202326
## 5  0.11697127 -1.3345995
```

Given, n=5

Step. 1: Mean of x and y (I have named these as x3, y3 for the question 3)

$$\bar{x} : Mean\, of\, x_3 = \frac{(-0.50) + (0.13) + (-0.07) + (0.88) + (0.11)}{5} = \frac{0.55}{5} = 0.11$$

Can also be checked in R:

```
mx<-c(-0.50219235,0.13153117,-0.07891709,0.88678481,0.1169712)
mean(mx)
```

```
## [1] 0.1108355
```

$$\bar{y} : Mean\, of\, y_3 = \frac{(-0.33) + (1.48) + (-0.38) + (1.12) + (-1.33)}{5} = \frac{0.56}{5} = 0.111$$

In R:

```
my<-c(-0.3333618,1.4894199,-0.3849308 ,1.1202326,-1.3345995)
mean(my)
```

```
## [1] 0.1113521
```

```
De<-data.frame(mx,my)#Create a dataframe for question 3.e
```

Step. 2: Covariance of x and y

$$Cov_{(x,y)} = \frac{1}{n-1}\sum(x_i-\bar{x})(\bar{y}_i-\bar{y})$$

$$Cov_{x,y} = \frac{1}{5-1}(-0.50-0.11)(-0.33-0.111)+(0.13-0.11)(1.48-0.111)+(-0.07-0.11)(-0.38-0.111)+(0.88-0.11)(1.12-0.11$$

$$Cov_{x,y} = \frac{1}{4}+(0.027)+(0.088)+(0.77)+(0) = \frac{1.145}{4} = 0.286 \approx 0.29$$

Can also be checke in R:

```
cov_x3_y3<-cov(mx,my)
cov_x3_y3
```

```
## [1] 0.2923203
```

Step.3: Standard deviation of x and y s_x : Standard deviation of x

$$s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

$$s_x = \sqrt{(\frac{1}{5-1})(-0.50-0.11)^2+(0.13-0.11)^2+(-0.07-0.11)^2+(0.88-0.11)^2+(0.11-0.11)^2}$$

$$s_x = \sqrt{\frac{1}{4}*0.9978} = 0.499 \approx 0.50$$

In R:

```
sx<- sd(mx)
sx
```

```
## [1] 0.5035803
```

s_y: Standard deviation of y

$$s_y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i-\bar{y})^2}$$

$$s_y = \sqrt{(\frac{1}{5-1})(-0.33-0.111)^2+(-0.38-0.111)^2+(1.12-0.111)^2+(-1.33-0.111)^2+(-1.33-0.111)^2}$$

$$s_y = \sqrt{\frac{1}{4}*5.404} = 1.167$$

In R:

```
sy<-sd(my)
sy
```

```
## [1] 1.16745
```

Step 4: Calculating coefficients For β_1:

$$r = \frac{Cov_{x,y}}{s_x * s_y} = \beta_1 \frac{s_x}{s_y}$$

$$r = \frac{0.29}{0.50 * 1.16} = \beta_1 \frac{0.50}{1.16} = \frac{0.29}{0.58}$$

$$\beta_1 = 1.15$$

For β_0

$$\beta = \bar{y} - \beta_1 \bar{x}$$

$$\beta_0 = (0.111 - 1.16 * 0.11) = -0.016$$

Step 5: Calculate the predicted $\hat{y_i}$ for each x_i For each x_i, $\hat{y}$ is:

$$\hat{y} = \beta_0 + \beta_1 x_1$$

$$\hat{y_1} = -0.016 + 1.15 * (-0.50) = -0.591$$

$$\hat{y_2} = -0.016 + 1.15 * (0.13) = 0.1335$$

$$\hat{y_3} = -0.016 + 1.15 * (-0.07) = -0.0965$$

$$\hat{y_4} = -0.016 + 1.15 * (0.11) = 0.1105$$

Step 6: Calculate the standard error of $se_{\hat{y}}$

$$se_{\hat{y}} = \sqrt{\frac{\sum (y_i - \hat{y_i})^2}{n-2}}$$

$$se_{\hat{y}} = \sqrt{\frac{(-0.33 - (-0.591))^2 + (1.48 - 0.1335)^2 + (-0.38 - (-0.0965))^2 + (1.12 - (-0.0965))^2 + (1.33 - 0.1105)^2}{5-2}}$$

$$se_{\hat{y}} = \sqrt{\frac{4.928608}{3}} = \sqrt{1.642869} = 1.281745$$

Now, we will calculate standard error of β_1 that is se_{β_1}

$$se_{\beta_1} = se_{\hat{y}} \frac{1}{\sum (x_i - \bar{x})^2}$$

$$se_{\beta_1} = 1.281745 * \frac{1}{(-0.50 - 0.11)^2 + (0.13 - 0.11)^2 + (-0.07 - 0.11)^2 + (0.88 - 0.11)^2 + (0.11 - 0.11)^2}$$

$$se_{\beta_1} = 1.281745 * \frac{1}{0.9978} = 1.287403$$

Step 7: Calculate SEE and TSS

$$R^2 = \frac{TSS - SEE}{TSS}$$

$$TSS = \sum_i (y_i - \bar{y})^2$$

$$TSS = (0.330.111)^2 + (1.480.111)^2 + (0.380.111)^2 + (1.120.111)^2 + (1.330.111)^2 = 5.40428$$

Now, we will calculate SSE: Sum of Squared errors:

$$SSE = \sum_i (y_i - \hat{y})^2$$

$$SSE = (0.33(0.591))^2 + (1.480.1335)^2 + (0.38(0.0965))^2 + (1.12(0.0965))^2 + (1.330.1105)^2 = 4.928608$$

$$R^2 = \frac{5.40428 - 4.928608}{5.40428} = 0.08801765$$

Step 8: Calculate Adjusted R^2

$df_t = n1 = 51 = 4$ $df_e = nk1 = 511 = 3$

$$Adjusted\ R^2 = \frac{\frac{TSS}{df_t} - \frac{SEE}{df_e}}{\frac{TSS}{df_t}}$$

$$Adjusted\ R^2 = \frac{\frac{5.40428}{4} - \frac{4.928608}{3}}{\frac{5.40428}{4}} = -0.2159764$$

INTERPRETATION Theoretically, we can add dozens of variables to our model and, since each one would at worst do nothing to improve $\hat{y}$ and might accidentally improve it, if we added enough variables we could get a very high R^2 entirely by chance.

This is over fitting, and is an especially severe problem for complex models with many variables. To compensate for overfitting, the adjusted R^2 was developed, which basically penalizes the R^2 for each additional independent variable.

In our model we only had one variable, so Adjusted R^2 has no meaning. Test-statistics for first 5 observationsof x and y.

$$Test - statistics\ for\ first\ 5\ observations\ of\ x\ and\ y$$

$$Test - statistics = \frac{\beta_1}{se_{\beta_1}} = \frac{1.15}{1.28} = 0.898 \approx 0.98$$

For Threshold:

```
Ule<-qt(.975,3)
Ule
```

```
## [1] 3.182446
```

```
Lle<-qt(0.025,3)
Lle
```

```
## [1] -3.182446
```

So, Threshold > Test Statistics : 3.18 > 0.98 We can conclude that the effect of X on Y is NOT Statistically significant We can also confirm this by: For p-value:

```
2*(1-pt(0.98,3)) #n-k-1=5-1-1=3, where k is the number of variables
```

```
## [1] 0.3993551
```

This can be crossed-checked in R:

```
BVe<-lm(my~mx, data = De)
summary(BVe)
```

```
##
## Call:
## lm(formula = my ~ mx, data = De)
##
## Residuals:
##       1       2       3       4       5
##  0.2619  1.3542 -0.2776  0.1144 -1.4530
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.01641    0.53867  -0.030    0.978
## mx           1.15271    1.16129   0.993    0.394
##
## Residual standard error: 1.17 on 3 degrees of freedom
## Multiple R-squared:  0.2472, Adjusted R-squared:  -0.003691
## F-statistic: 0.9853 on 1 and 3 DF,  p-value: 0.3941
```

INTERPRETATION We can interpret from the above Bivariate test and manual calculations, that the p-value is greater than 0.05, suggesting our test is NOT statistically significant. And, we fail to reject the Null hypothesis, that is there is no effect of X on Y and our coefficient is significantly different from 0, hence our test is in favor of the Null hypothesis.

4. Now generate y = 0.1 + 0.2 ∗ x − 0.5 ∗ x^2 + E with 100 observations.

```
set.seed(150)
number <- 100
x4 <- rnorm(number, 0, 1) # n: number observations, mean=0, sd=1
e_4 <- rnorm(number, 0, 1) #e: epsilon, mean=0, sd=1
y4 <- (0.1+0.2*(x4)-0.5*(x4)^2+e_4)
D4<-data.frame(x4,y4)
head(D4)
```

```
##            x4         y4
## 1 -1.63230970 -1.7717820
## 2 -0.06299626  1.2100664
## 3 -0.70544686 -1.0954035
## 4 -0.31417818 -0.4726095
## 5 -0.26694627  0.6031266
## 6  0.15315947  1.1102629
```

a. Regress y on x and x^2 and report the results. If x or x^2 are not statistically significant, suggest why.

```
mod_quad <- lm(y4 ~ x4 + I(x4^2), data=D4) # y4 is y, x4 is x and x4^2 is X^2

summary(mod_quad)
```

```
##
## Call:
## lm(formula = y4 ~ x4 + I(x4^2), data = D4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7032 -0.6851 -0.1432  0.5657  2.4964
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.15487    0.11262   1.375    0.172
## x4           0.41599    0.08685   4.790    6e-06 ***
## I(x4^2)     -0.51077    0.06450  -7.918    4e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.894 on 97 degrees of freedom
## Multiple R-squared:  0.4766, Adjusted R-squared:  0.4658
## F-statistic: 44.16 on 2 and 97 DF,  p-value: 2.315e-14
```

```r
mod_quad <- lm(y4 ~ x4 + I(x4^2), data=D4)

vars <- c("y","x", "$x^2$")

library(stargazer)
```

```r
stargazer(mod_quad,no.space=TRUE, header=FALSE, dep.var.labels=vars[1],
          covariate.labels=vars[-1],
          omit.stat=c("LL", "ser", "f"),
          p.auto=FALSE, star.char = c("*", "**", "***"),
          star.cutoffs = c(0.05, 0.01, 0.001),
          notes=c("*p<0.05; **p<0.01; ***p<0.001"),
          notes.append=F, type="latex")
```

Table 1:

	Dependent variable:
	y
x	0.416***
	(0.087)
$x^2$	−0.511***
	(0.065)
Constant	0.155
	(0.113)
Observations	100
R^2	0.477
Adjusted R^2	0.466
Note:	*p<0.05; **p<0.01; ***p<0.001

INTERPRETATION We can see that x (I have named it as: x4) and $x^2$ (named as $x42$) both have p-value less than 0.05 that means they have a significant effect on y (named y4). Also,x has a $\beta = 0.41599$, that means it has a positive effect on y and with a unit increment on x, y will will be increased by 0.41599. On the other hand, $x^2$ has a $\beta = -0.51077$, that means it has a negative effect on y and with a unit increment in $x^2$, y will be decreased by -0.51077. X haS significant positive impact at 95% Confidence interval, the R^2 = 0.4766 suggesting we have reduced 47.66% of error in variation Y to other independent variables (x & $x^2$). Also the Adjusted $R^2 = 0.4658$ has penalised for the addtional variables, suggesting the model will turn out to provide an accurate predictive model.

b. Based on the known coefficients that we used to create y, what is the effect on y of increasing x by 1 unit from 1 to 2?

$$y = 0.1 + 0.2 * x - 0.5 * x^2 + \epsilon$$

Now if we increase x by 1 unit from 1 to 2 then:

$$y = \beta_0 + \beta_1 * x + \beta_2 * x^2$$

So,

$$y = \beta_0 + \beta_1 * 1 + \beta_2 * 1^2$$

And,

$$y = \beta_0 + \beta_1 * 2 + \beta_2 * 2^2$$

From our equation we know,

$$\beta_0 = 0.1, \beta_1 = 0.2, \beta_@ = -0.5$$

Change:

$$y_2 - y_1 = (\beta_0 + \beta_1 * 2 + \beta_2 * 2^2) - (\beta_0 + \beta_1 * 1 + \beta_2 * 1^2)$$
$$y_2 - y_1 = (0.1 + 0.2 * 2 + (-0.5) * 2^2) - (0.1 + 0.2 * 1 + (-0.5) * 1^2)$$

INTERPRETATION

The quadratic term is statistically significant in the model. There is a change in y of -1.3 with a 1-unit increase in the x. That means, y will decrease by -1.3 if x is increased from 1 to 2.

c. Based on the coefficients estimated from 4(a), what is the effect on y of changing x from -0.5 to -0.7?

$_x = 0.41599$ $^2_x = 0.51077$ We can calculate the difference in R:

```
y1 <- mod_quad$coefficients[2]*(-0.5) + mod_quad$coefficients[3]*(-0.5)^2
y2 <- mod_quad$coefficients[2]*(-0.7) + mod_quad$coefficients[3]*(-0.7)^2
y2 - y1
```

```
##         x4
## -0.205783
```

INTERPRETATION Based on the coefficients estimated from 4(a), on changing x from -0.5 to -0.7, There is a change in y of -0.205783 with a 1-unit increase in the x.

5. Now generate x2 as a random normal variable with a mean of -1 and a sd of 1. Create a new dataset where y = 0.1 + 0.2 ∗ x − 0.5 ∗ x ∗xˆ2+E

n: number observations, mean=0, standard deviation=1, for x n: number observations, mean=-1, standard deviation=1, for x2 e: epsilon, mean=0, standard deviation=1 For the new dataset:

```
set.seed(250)
number <- 100
x5 <-rnorm(number,0,1)
x2_5<-rnorm(number,-1,1)
e_5 <- rnorm(number, 0, 1)
y5 <- (0.1+0.2*(x5)-0.5*(x5)*(x2_5)+e_5)
Data5 <- data.frame(y5,x5,x2_5)
head(Data5)
```

```
##           y5           x5       x2_5
## 1 -0.9487406 -0.626779843 -0.9645268
## 2  0.1434355 -0.957793042 -0.8542161
## 3  1.0729715  0.841433324 -1.3592294
## 4  2.0259765  0.937809627 -2.3011931
## 5  0.3154915  0.000663045 -0.4970152
## 6 -1.1079792 -0.366967423 -1.1100259
```

a. Based on the known coefficients, what is the effect of increasing x2 from 0 to 1 with x held at its mean?

Mean of x:

```r
meanx5 <- mean(Data5$x5)
meanx5
```

```
## [1] 0.02747904
```

$\bar{x} = 0.02747904$

$y = 0.1 + 0.2 * x - 0.5 * x * x_2$

When $x_2 = 0$,

$y_1 = 0.1 + 0.2 * 0.027 - 0.5 * 0.027 * 0 = 0.104$

$y_1 = 0.104$

When $x_2 = 1$,

$y_2 = 0.1 + 0.2 * 0.02 - 0.5 * 0.027 * 1 = 0.0905$

Difference $(y_2 - y_1)$

$y_2 - y_1 = 0.0905 - 0.104 = -0.0135$

The effect of x_2 when x is held at mean, and x_2 is changes from zero to one is -0.0135

b. Regress y on x, x2, and their interaction. Based on the regression-estimated coefficients, what is the effect on y of shifting x from -0.5 to -0.7 with x2 held at 1?

```r
#y~(x)+(x_2)+(x)*(x_2) :y on x, x2, and their interaction

MV5 <- lm(y5 ~ x5 + x2_5 + x5*x2_5, data=Data5)
summary(MV5)
```

```
##
## Call:
## lm(formula = y5 ~ x5 + x2_5 + x5 * x2_5, data = Data5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.22124 -0.55737 -0.08463  0.54210  2.57060
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.08148    0.12893   0.632    0.529
## x5          -0.05462    0.13910  -0.393    0.695    
## x2_5         0.04315    0.08899   0.485    0.629    
## x5:x2_5     -0.51815    0.09084  -5.704 1.29e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9314 on 96 degrees of freedom
## Multiple R-squared:  0.3852, Adjusted R-squared:  0.3659 
## F-statistic: 20.05 on 3 and 96 DF,  p-value: 3.603e-10
```

From the Multivariate regression, we observe the interaction term is significant, and the $\beta =$ -0.51815, which indicates that with one unit increase in x and x2, the combined effect on y will be negative, with a coefficient of -0.51815.

$y = 0.1 + 0.2 * x - 0.5 * x * x2$ When x = -0.5,

```
q5_y1<-MV5$coefficients[2]*(-0.5)+MV5$coefficients[3]*(1)+
MV5$coefficients[4]*(1*(-0.5))
q5_y1
```

```
##        x5 
## 0.3295298
```

```
#When x = -0.7
q5_y2<-MV5$coefficients[2]*(-0.7)+MV5$coefficients[3]*(1)+
MV5$coefficients[4]*(1*(-0.7))
q5_y2
```

```
##       x5 
## 0.444083
```

```
#Subtracting (y2-y1)
(q5_y2)-(q5_y1)
```

```
##        x5 
## 0.1145532
```

Looking at our results, there is an increase of 0.1145532 in y with a one unit increase in the x, holding the x2 constant at 1.

c. Regress the current y on x alone. Using the R2 from this regression and the R2 from 5(b), perform by hand an F test of the complete model (5b) against the reduced, bivariate model. What does this test tell you?

Testing the nested models so, H_0 : The reduced model is statistically better than the complete model H_1 : The reduced model is not better than the complete model.

```
RMV5c<-lm(y5 ~ x5, data=Data5) #y5=y, x5=x
summary(RMV5c)
```

```
##
## Call:
## lm(formula = y5 ~ x5, data = Data5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.44628 -0.62867 -0.00538  0.59500  2.65527
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02338    0.10695  -0.219    0.827
## x5           0.51132    0.11288   4.530 1.67e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.069 on 98 degrees of freedom
## Multiple R-squared:  0.1731, Adjusted R-squared:  0.1647
## F-statistic: 20.52 on 1 and 98 DF,  p-value: 1.666e-05
```

P-value is less than 0.05 and so x has a significant effect. The coefficient is 0.511 displaying that with one unit increase in x, y will increase 0.511. The R^2 value suggests that we have reduced the error by 17.31% and explained all the error in variation y explained by x.

Reduced Bivariate regression model: $R_r{}^2 = 0.1731$ Complete regression model: $R_c{}^2 = 0.3852$ F-test: $F = \frac{(R_c{}^2 - R_r{}^2)/df1}{(1-R_c{}^2)/df2}$ In this, df1 = number of additional variables in the complete model = 2 df2 = n-k-1 = for the complete model,k is the total number of independent variables in the complete model = 100-1-1 = 98

Substituting values in our F-test formula: $F = \frac{(0.3852-0.1731)/2}{(1-0.3852)/98} = \frac{0.2121/2}{0.00627} = 16.52$

Threshold:

```r
qf(0.95,2,98)
```

```
## [1] 3.089203
```

So, F-statistic > Threshold value (16.52> 3.089) P-value

```r
pf(16.52,2,98,lower.tail=F)
```

```
## [1] 6.565796e-07
```

As we can see from the results, P-value is less than 0.05 indicating that the test is statistically significant.

Verifying using R:

```r
Full <- lm(y5 ~ x5 +x2_5 + x5*x2_5, data=Data5)
Reduced <- lm(y5 ~ x5, data=Data5)
anova(Reduced, Full)
```

```
## Analysis of Variance Table
##
## Model 1: y5 ~ x5
## Model 2: y5 ~ x5 + x2_5 + x5 * x2_5
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)    
## 1     98 112.005                                  
## 2     96  83.284  2    28.721 16.553 6.662e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the result, we can see that the value of f-test that we calculated by hand is 16.55 and the p-value is less than 0.05 and after running nested model in R we observe that x, x2 and interaction terms $x * x2$ both belong in the regression. Fstatistic compares the sum of squared error from reduced model $({R_r}^2)$ against the sum of squared error from the reduced model $({R_s}^2)$ relative to the additional number of parameters we are adding to the model. So we can see that f statistic falls under the critical threshold value and the p-value is also less than 0.05, which suggests test is statistically significant. This means we can reject the null hypothesis and our test results in favour of the research hypothesis stating the complete model is better than the Restricted model.

6. Generate a new variable y2 using the data from (5) which is 1 if y > 0 and 0 otherwise.

Using Data fron question 5:

```
set.seed(250)
number <- 100
x5<-rnorm(number,0,1)# n: number observations, mean=0, sd=1, this is x
x2_5<-rnorm(number,-1,1) # n: number observations, mean=-1, sd=1, this is x2
e_5 <- rnorm(number, 0, 1) #e: epsilon, mean=0, sd=1
y5 <- (0.1+0.2*(x5)-0.5*(x5)*(x2_5)+e_5) #y5 for question 5

D5 <- data.frame(y5,x5,x2_5)
head(D5)
```

```
##           y5           x5       x2_5
## 1 -0.9487406 -0.626779843 -0.9645268
## 2  0.1434355 -0.957793042 -0.8542161
## 3  1.0729715  0.841433324 -1.3592294
## 4  2.0259765  0.937809627 -2.3011931
## 5  0.3154915  0.000663045 -0.4970152
## 6 -1.1079792 -0.366967423 -1.1100259
```

```
y2_6<- numeric(length(y5)) #this is y2 for question 6
for (i in seq_along(y5)){
if (y5[i] > 0) {y2_6[i]<- 1}
else if (y5[i]<0) {y2_6[i]<- 0}
}
head(y2_6) #to check value of y2 if y>0->1 and y<0->0
```

```
## [1] 0 1 1 1 1 0
```

a. Perform a logistic regression of y2 on x, x2, and their interaction, and interpret the results.

```
lr <- glm(y2_6 ~ x5 + x2_5 + x5*x2_5,
family="binomial"(link="logit"), D5)
summary(lr)
```

```
## 
## Call:
## glm(formula = y2_6 ~ x5 + x2_5 + x5 * x2_5, family = binomial(link = "logit"), 
##     data = D5)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.6485  -1.1119   0.3328   1.0798   2.0640  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)   
## (Intercept)  0.125602   0.303186   0.414  0.67867   
## x5          -0.323435   0.360894  -0.896  0.37014   
## x2_5        -0.003907   0.247183  -0.016  0.98739   
## x5:x2_5     -1.108885   0.350131  -3.167  0.00154 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 138.47  on 99  degrees of freedom
## Residual deviance: 115.50  on 96  degrees of freedom
## AIC: 123.5
## 
## Number of Fisher Scoring iterations: 5
```

```r
vars <- c("y2","x", "x2","x*x2")
library(stargazer)
stargazer(lr,no.space=TRUE, header=FALSE, dep.var.labels=vars[1],
covariate.labels=vars[-1], omit.stat=c("LL", "ser", "f"),
p.auto=FALSE, star.char = c("*", "**", "***"),
star.cutoffs = c(0.05, 0.01, 0.001),
notes=c("*p<0.05; **p<0.01; ***p<0.001"),
notes.append=F, type="text")
```

```
## 
## ===============================================
##                        Dependent variable:     
##                   -----------------------------
##                                y2              
## -----------------------------------------------
## x                            -0.323            
##                              (0.361)           
## x2                           -0.004            
##                              (0.247)           
## x*x2                        -1.109**           
##                              (0.350)           
## Constant                      0.126            
##                              (0.303)           
## -----------------------------------------------
## Observations                   100             
## Akaike Inf. Crit.            123.505           
## ===============================================
## Note:             *p<0.05; **p<0.01; ***p<0.001
```

```
exp(lr$coef)
```

```
## (Intercept)          x5        x2_5     x5:x2_5
##   1.1338310   0.7236588   0.9961010   0.3299265
```

INTERPRETATION After running the logistic regression on our binary dependent variable y2, we can infer that, the interation term x $*$ x2 term is has a statistically significant effect on y2. It has a negative coefficient that means, with one unit increase in x $*$ x2 the binary variable will decrease by 1.109 because, $\beta = -1.109$. After exponentiating the coefficients, we can get their effect on the odds: here we can see, interaction term x $* x^2$ increases the odds of y by about 33%

b. What is the effect of increasing x2 from 0 to 1 with x held at its mean on the probability that y2 is 1? Average of x:

```
mean(x5)
```

```
## [1] 0.02747904
```

According to our equation:

$$P_1(y=1) = \frac{e^{\beta_0+\beta_1 x+\beta_2 x_2+\beta_x * x_2}}{1+e^{\beta_0+\beta_1 x+\beta_2 x_2+\beta_x * x_2}}$$

From 6.b $\beta_0 = 0.125602$ $\beta_1 = -0.323435$ $\beta_2 = -0.003907$ $\beta_x x_2 = -1.108885$ When $x_2 = 0$ and mean of x = 0.02: Substituting values:

$$P_1(y=1) = \frac{e^{0.12+(-0.32)*(0.02)+(-0.003)*(0)+(-1.10)*(0.02*0)}}{1+e^{0.12+(-0.32)*(0.02)+(-0.003)*(0)+(-1.10)*(0.02*0)}}$$

$$P_2(y=1) = \frac{e^{0.1136}}{1+e^{0.1136}}$$

Calculate in R for Exponential:

```
exp(0.1136)
```

```
## [1] 1.120304
```

$$P_2(y=1) = \frac{1.120304}{1+1.120304} = 0.5283695$$

When $x_2 = 1$ and mean of x=0.02:

$$P_1(y=1) = \frac{e^{(0.12+(-0.32)*(0.02)+(-0.003)*(1)+(-1.10)*(0.02*1))}}{1+e^{(0.12+(-0.32)*(0.02)+(-0.003)*(1)+(-1.10)*(0.02*1))}}$$

$$P_2(y=1) = \frac{e^{0.0886}}{1+e^{0.0886}}$$

```
exp(0.0886)
```

```
## [1] 1.092644
```

$$P_2(y=1) = \frac{1.092644}{1+1.092644} = 0.5221356$$

Change from 0 to 1 is P2-P1:

$$P2 - P1 = 0.5221356 - 0.5283695 = -0.0062339$$

INTERPRETATION When x2 is increased from 0 to 1 with x held at its mean on the probability that y2 is 1, then the change is -0.0062339.

7. Generate a dataset with 300 observations and three variables: f, x1, and x2. f should be a factor with three levels, where level 1 corresponds to observations 1-100, level 2 to 101-200, and level 3 to 201-300. (Eg, f can be "a" for the first 100 observations, "b" for the second 100, and "c" for the third 100.) Create x1 such that the first 100 observations have a mean of 1 and sd of 2; the second 100 have a mean of 0 and sd of 1; and the third 100 have a mean of 1 and sd of 0.5. Create x2 such that the first 100 observations have a mean of 1 and sd of 2; the second 100 have a mean of 1 and sd of 1; and the third 100 have a mean of 0 and sd of 0.5. (Hint: It is probably easiest to create three 100-observation datasets first, and then stack them with rbind(). And make sure to convert f to a factor before proceeding.)

Loading Libraries:

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(GPArotation)
```

Creating Dataset:

```
#Data 1 :
set.seed(10)
X1<-rnorm(100,1,2)
set.seed(15)
X2<-rnorm(100,1,2)
f<-rep("A",100)
Dat1<-data.frame(X1,X2,f)

#Data 2 :
set.seed(20)
X1<-rnorm(100,0,1)
set.seed(25)
X2<-rnorm(100,1,1)
f<-rep("B",100)
Dat2<-data.frame(X1,X2,f)

#Data 3 :
```

```r
set.seed(30)
X1<-rnorm(100,1,0.5)
set.seed(35)
X2<-rnorm(100,0,0.5)
f<-rep("C",100)
Dat3<-data.frame(X1,X2,f)

#Forming one dataset,
Data7<-rbind.data.frame(Dat1,Dat2,Dat3)
Data7<-as.data.frame(Data7)
Data7$X1<-as.numeric(Data7$X1)
Data7$X2<-as.numeric(Data7$X2)
Data7$f<-as.factor(Data7$f)
summary(Data7)
```

```
##        X1                 X2           f
##  Min.   :-3.3706   Min.   :-3.8750   A:100
##  1st Qu.:-0.1940   1st Qu.:-0.1486   B:100
##  Median : 0.6331   Median : 0.3633   C:100
##  Mean   : 0.5647   Mean   : 0.7061
##  3rd Qu.: 1.2860   3rd Qu.: 1.3068
##  Max.   : 5.4410   Max.   : 5.9704
```

```r
#Forming a subset
Sub7<-Data7[,1:2]
summary(Sub7)
```

```
##        X1                 X2
##  Min.   :-3.3706   Min.   :-3.8750
##  1st Qu.:-0.1940   1st Qu.:-0.1486
##  Median : 0.6331   Median : 0.3633
##  Mean   : 0.5647   Mean   : 0.7061
##  3rd Qu.: 1.2860   3rd Qu.: 1.3068
##  Max.   : 5.4410   Max.   : 5.9704
```

a. Using the k-means algorithm, peform a cluster analysis of these data using a k of 3 (use only x1 and x2 in your calculations; use f only to verify your results). Comparing your clusters with f, how many datapoints are correctly classified into the correct cluster? How similar are the centroids from your analysis to the true centers?

1: K-means = cluster analysis with 3 centers:

```r
kout <- kmeans(Sub7,centers=3,nstart=25)
print(kout)
```

```
## K-means clustering with 3 clusters of sizes 40, 78, 182
##
## Cluster means:
##          X1         X2
## 1  1.419343 3.04824013
## 2 -1.024659 1.13612433
## 3  1.058037 0.00699535
##
## Clustering vector:
##   [1] 3 1 2 1 1 3 2 1 2 3 1 3 3 3 1 3 2 3 3 3 1 2 2 2 2 2 2 2 1 1 2 3 1 1 2
##  [36] 2 3 2 3 3 1 2 1 3 2 1 3 1 2 1 1 1 3 1 1 3 2 1 1 3 2 3 2 3 3 3 3 2 3 2
##  [71] 3 2 3 3 2 3 2 1 2 1 3 1 3 3 3 3 3 1 2 3 3 3 1 1 3 2 2 1 3 3 3 2 3 2 3
## [106] 3 2 2 2 2 3 3 2 3 2 2 3 2 1 3 2 2 3 1 3 2 2 2 2 2 3 3 3 3 2 2 2 3 3 3
## [141] 2 2 1 2 2 2 3 3 3 2 3 2 3 3 1 3 3 2 2 3 3 2 2 2 1 2 3 2 3 2 1 2 3 2 1
## [176] 3 2 2 2 3 2 3 3 3 2 2 2 3 3 3 1 2 3 2 3 1 1 3 2 3 3 3 3 3 1 3 3 3 3 3
## [211] 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [246] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [281] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##
## Within cluster sum of squares by cluster:
## [1] 134.5467 157.9887 219.2510
##  (between_SS / total_SS =  53.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

So we can see K-means clustering with 3 clusters of sizes 40, 78, 182 in the result. We observe that cluster centers for three groups across the two variables. (X1, X2) We also see the cluster assignment for each observation

```
CR <-kout$cluster
table(CR,Data7$f)
```

```
##
## CR   A  B  C
##   1 29 10  1
##   2 30 47  1
##   3 41 43 98
```

In this table of confusion, we interpret that cluster 1,2,3 corresponds to A,B,C respectively. Total number of accurately classified instances are 41+43+1=85 Total number of inaccuratey classified instances are 30+1+29+10=70 Accuracy = 85/(85+70)=0.5483871 So that shows that our model has 54.8387% accuracy.

Making groups of f to A, B and C:

```
#Group A
print("GroupA")
```

```
## [1] "GroupA"
```

```
table(kout$cluster[1:100])
```

```
##
##  1  2  3
## 29 30 41
```

```
#Group B
print("GroupB")
```

```
## [1] "GroupB"
```

```
table(kout$cluster[101:200])
```

```
##
##  1  2  3
## 10 47 43
```

```
#Group C
print("GroupC")
```

```
## [1] "GroupC"
```

```
table(kout$cluster[201:300])
```

```
##
##  1  2  3
##  1  1 98
```

2: Calculating the real centers using f, for group a, b and c:

```
dfA <- Data7[Data7$f=="A",]
str(dfA)
```

```
## 'data.frame':    100 obs. of  3 variables:
##  $ X1: num  1.037 0.631 -1.743 -0.198 1.589 ...
##  $ X2: num  1.518 4.662 0.321 2.794 1.976 ...
##  $ f : Factor w/ 3 levels "A","B","C": 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(dfA)
```

```
##        X1                 X2            f
##  Min.   :-3.3706   Min.   :-3.87497   A:100
##  1st Qu.:-0.6582   1st Qu.:-0.05167   B:  0
##  Median : 0.6134   Median : 1.02749   C:  0
##  Mean   : 0.7269   Mean   : 1.20454
##  3rd Qu.: 2.1867   3rd Qu.: 2.73111
##  Max.   : 5.4410   Max.   : 5.97036
```

3: Calculating Mean:

```
meanX1A <- mean(dfA$X1)
meanX1A
```

```
## [1] 0.7269021
```

```r
meanX2A <- mean(dfA$X2)
meanX2A
```

```
## [1] 1.204542
```

The true center of group A is (X1,X2) is 0.7269021 and 1.204542

#Group B

```r
dfB <- Data7[Data7$f=="B",]
str(dfB)
```

```
## 'data.frame':    100 obs. of  3 variables:
##  $ X1: num  1.163 -0.586 1.785 -1.333 -0.447 ...
##  $ X2: num  0.7882 -0.0416 -0.1533 1.3215 -0.5001 ...
##  $ f : Factor w/ 3 levels "A","B","C": 2 2 2 2 2 2 2 2 2 2 ...
```

```r
summary(dfB)
```

```
##        X1                  X2            f      
##  Min.   :-2.889718   Min.   :-1.34491   A:  0  
##  1st Qu.:-0.600497   1st Qu.:-0.03316   B:100  
##  Median :-0.024861   Median : 0.85801   C:  0  
##  Mean   : 0.004908   Mean   : 0.81784          
##  3rd Qu.: 0.768380   3rd Qu.: 1.35716          
##  Max.   : 2.208443   Max.   : 3.36776          
```

Calculating Mean:

```r
meanX1B <- mean(dfB$X1)
meanX1B
```

```
## [1] 0.004908104
```

```r
meanX2B <- mean(dfB$X2)
meanX2B
```

```
## [1] 0.8178423
```

The true center of group B is (X1,X2) is 0.004908104 and 0.8178423

#Group C

```r
dfC <- Data7[Data7$f=="C",]
str(dfC)
```

```
## 'data.frame':    100 obs. of  3 variables:
##  $ X1: num  0.356 0.826 0.739 1.637 1.912 ...
##  $ X2: num  0.5326 0.0664 -0.017 -0.0225 1.6689 ...
##  $ f : Factor w/ 3 levels "A","B","C": 3 3 3 3 3 3 3 3 3 3 ...
```

```
summary(dfC)
```

```
##        X1                  X2            f      
##  Min.   :-0.4672   Min.   :-1.02338   A:  0  
##  1st Qu.: 0.6272   1st Qu.:-0.25221   B:  0  
##  Median : 0.9639   Median : 0.08279   C:100  
##  Mean   : 0.9623   Mean   : 0.09582          
##  3rd Qu.: 1.2832   3rd Qu.: 0.39625          
##  Max.   : 2.2992   Max.   : 1.66892          
```

Calculating Mean:

```
meanX1C <- mean(dfC$X1)
meanX1C
```

```
## [1] 0.9623201
```

```
meanX2C <- mean(dfC$X2)
meanX2C
```

```
## [1] 0.09582055
```

The true center of group C is (X1,X2) is 0.9623201 and 0.09582055

4: Creating two matrices for the true centers from f and the cluster centers,

```
fcentx1 <- c(meanX1A , meanX1B, meanX1C)
fcentx2 <- c(meanX2A, meanX2B, meanX2C)
Fcent <- as.matrix(cbind(fcentx1, fcentx2))
Fcent
```

```
##          fcentx1    fcentx2
## [1,] 0.726902113 1.20454176
## [2,] 0.004908104 0.81784227
## [3,] 0.962320063 0.09582055
```

We have two columns with true centers and cluster centers.

```
kout$centers
```

```
##          X1         X2
## 1  1.419343 3.04824013
## 2 -1.024659 1.13612433
## 3  1.058037 0.00699535
```

5: Calculating the distance between the true centers and the cluster centers

```
Center <- Fcent - kout$centers
Center
```

```
##           fcentx1    fcentx2
## [1,] -0.69244137 -1.8436984
## [2,]  1.02956697 -0.3182821
## [3,] -0.09571664  0.0888252
```

#Calculating the euclidean distance between true and cluster center 1

```
Center1_dist <- sqrt(Center[1,1]^2 + Center[1,2]^2)
Center1_dist
```

```
##  fcentx1
## 1.969441
```

```
Center2_dist <- sqrt(Center[2,1]^2 + Center[2,2]^2)
Center2_dist
```

```
##  fcentx1
## 1.077642
```

```
Center3_dist <- sqrt(Center[3,1]^2 + Center[3,2]^2)
Center3_dist
```

```
##   fcentx1
## 0.1305817
```

Looking at the results from the confusion matrices and the euclidean distances between true and cluster centers, the cluster which had the shortest distance between true and cluster centers (cluster 1) is also the one with the most accurate classifications, and the cluster with the longest distance between true and cluster centers (cluster 3) is the one with the least accurate classifications.

b. Perform a factor analysis of this data using your preferred function. Using the scree plot, how many factors do you think you should include? Speculate about how these results relate to those you got with the cluster analysis.

Using data that only has X1 and X2:

```
Sub7 <- Data7[,1:2]
str(Sub7)
```

```
## 'data.frame':    300 obs. of  2 variables:
##  $ X1: num  1.037 0.631 -1.743 -0.198 1.589 ...
##  $ X2: num  1.518 4.662 0.321 2.794 1.976 ...
```

```
summary(Sub7)
```

```
##        X1                 X2
##  Min.   :-3.3706   Min.   :-3.8750
##  1st Qu.:-0.1940   1st Qu.:-0.1486
##  Median : 0.6331   Median : 0.3633
##  Mean   : 0.5647   Mean   : 0.7061
##  3rd Qu.: 1.2860   3rd Qu.: 1.3068
##  Max.   : 5.4410   Max.   : 5.9704
```

```
ScaledSubset7 <- scale(Sub7)
summary(ScaledSubset7)
```

```
##        X1                  X2
##  Min.   :-2.96888   Min.   :-3.2898
##  1st Qu.:-0.57243   1st Qu.:-0.6138
##  Median : 0.05162   Median :-0.2462
##  Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.54412   3rd Qu.: 0.4314
##  Max.   : 3.67882   Max.   : 3.7804
```

#Direct eigen of cov for determination of correct number of factors which should be extracted

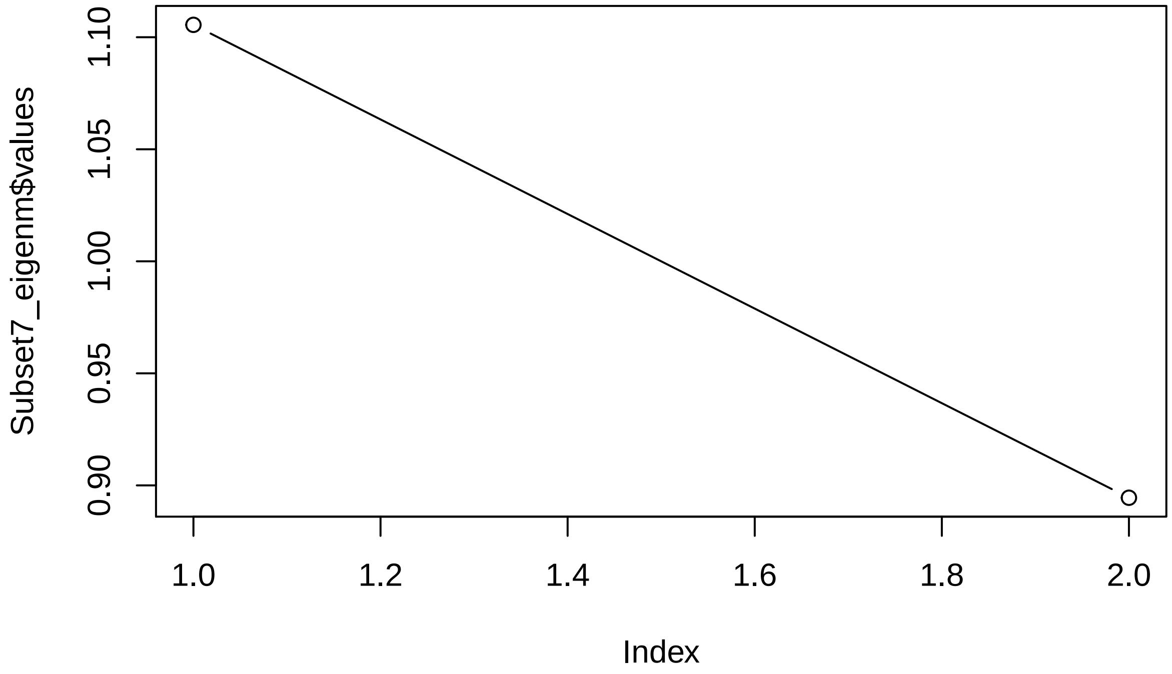
```
Subset7_cov <- cov(ScaledSubset7)
Subset7_eigenm <- eigen(Subset7_cov)
str(Subset7_eigenm)
```

```
## List of 2
##  $ values : num [1:2] 1.106 0.894
##  $ vectors: num [1:2, 1:2] -0.707 0.707 -0.707 -0.707
##  - attr(*, "class")= chr "eigen"
```

From these results we can see that only one variable has an eigen value greater than one indicating it increasingly contributes to variance explained in data.

#Scree plot from our manual eigenvector method:

```
plot(Subset7_eigenm$values,type="b")
```



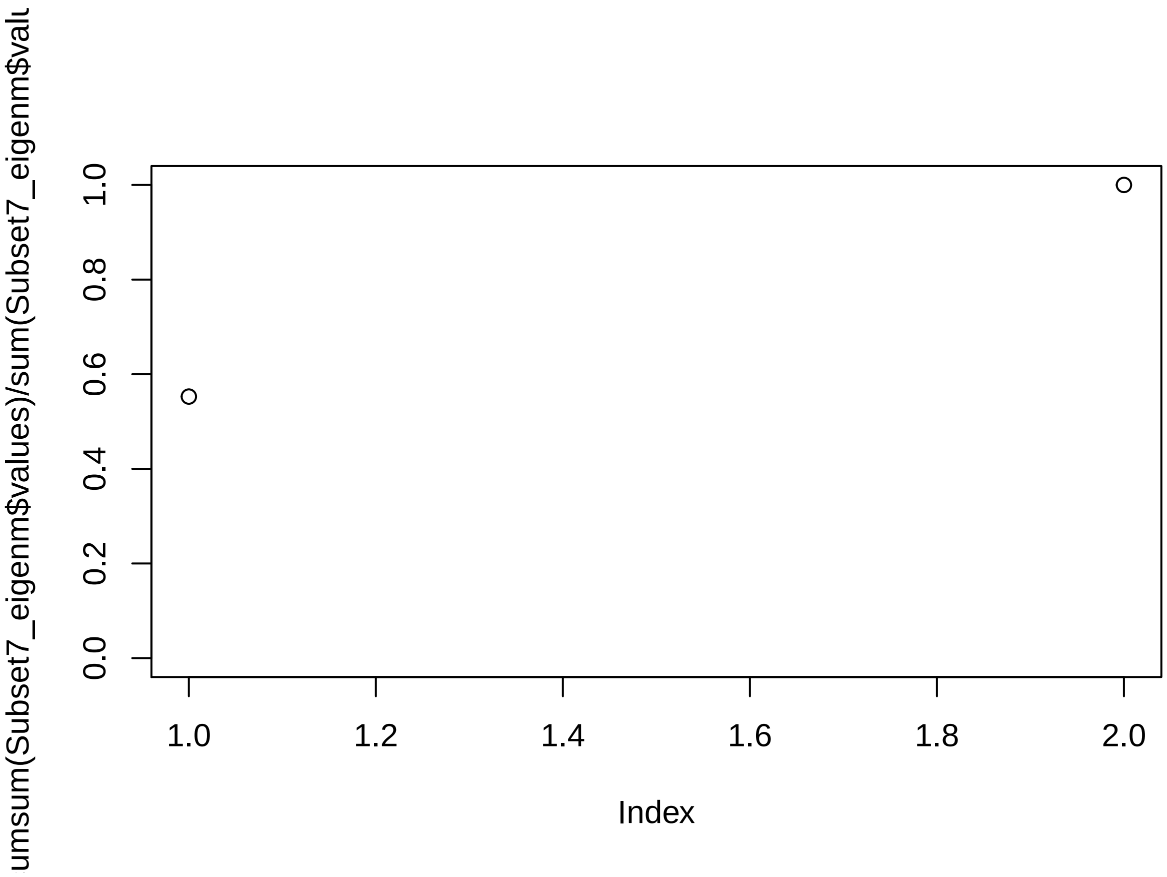
To choose number of factors with just two variables does'nt gives us an 'elbow' to help us determine which factors should be selected. But what we can infer from this plot is that factor 1 has a eigen value greater than one which could represent the data in a great way. Basically the point being we should not choose number of factors with only two variables.

Factor analysis with the 1 factor that we chose: #Calculating the first factor only.

```
Q7b_fact <- fa(ScaledSubset7,nfactors=1)
```

Cumulative explained variance of the first factor,

```
plot(cumsum(Subset7_eigenm$values)/sum(Subset7_eigenm$values),ylim=c(0,1))
```



We observe from the results that about 60% of total variance in the data is explained by first factor along with that when we add the second factor we explain the variance of the whole sample. We cannot have more factors than variables, more than 2 in our case.

Extracting the cumulative variance of the variables within each factor,

```
Q7b_fact$loadings
```

```
##
## Loadings:
##    MR1
## X1 -0.325
## X2  0.325
##
##                   MR1
## SS loadings    0.211
## Proportion Var 0.106
```

We observe that X2 is the variable that loads stronger over to the first factor.

8. Generate a dataset of 200 observations, this time with 90 independent variables, each of mean 0 and sd 1. Create y such that: $y = 2x1 + ... + 2x30 - x31 - ... - x60 + 0 * x61 + ... + 0 * x90 + E$ where E is a random normal variable with mean 0 and sd 10. (Ie, the first 30 x's have a coefficient of 2; the next 30 have a coefficient of -1; and the last 30 have a coefficient of 0.)

Creating Dataset:

```
library(glmnet)
```

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

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
## Loaded glmnet 3.0-1
```

```
library(mvtnorm)
coefs <- rep(c(2, -1, 0), each=30)
mu <- rnorm(200, 0, 10)
m <- rep(0, 90) # mean of independent variables
sig <- diag(90) # cov of indep variables
x <- rmvnorm(200, mean=m, sigma=sig) # generates 200 observations from multivariate normal
y <- x%*%coefs + mu
view(x)
str(y)
```

```
##  num [1:200, 1] 1.19 -4.87 17.56 9.85 -13.68 ...
```

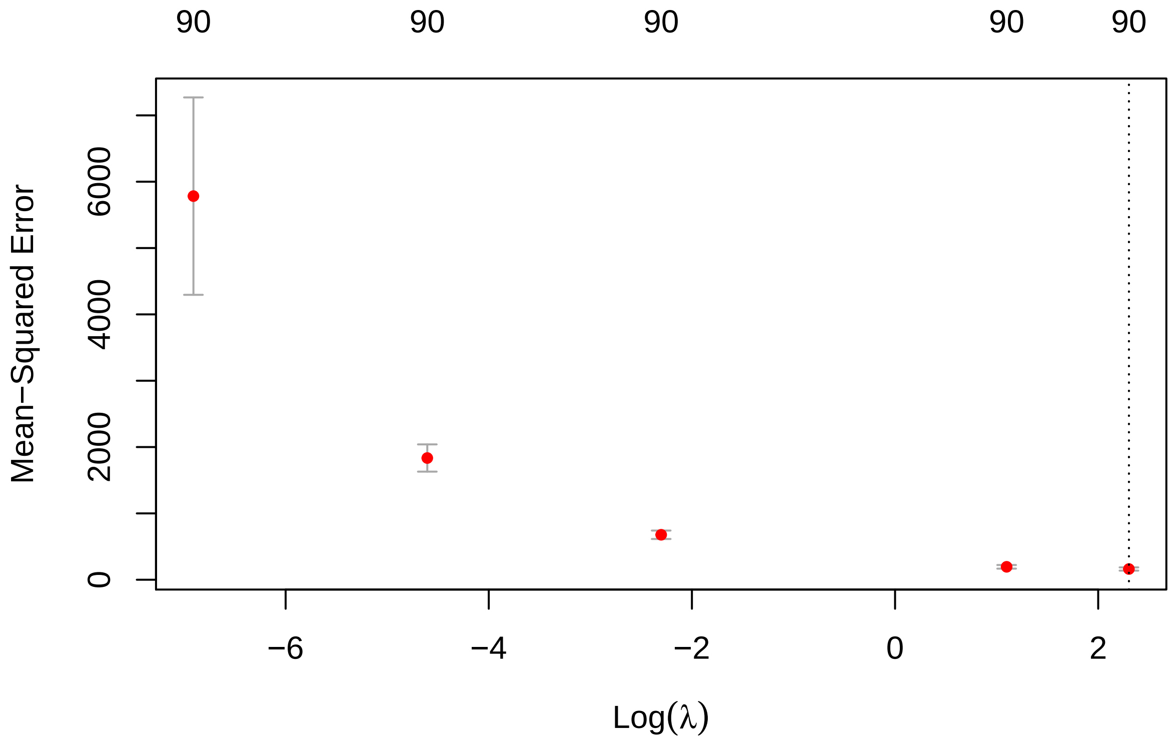
```
FirstHundred <- x[1:100,] #Choosing first 100 observation of x
y_100 <- y[1:100,] #Choosing first 100 observation of y

y2_100<- y[101:200,] #Selecting 2nd 100 overvations of x
x2_100<- x[101:200,] #Selecting 2nd 100 observations of y
```

a. Perform an elastic net regression of y on all the x variables using just the first 100 observations. Use 10-fold cross-validation to find the best value of and approximately the best value of .

```
alphas=c(0,0.5,1)
lambdas=c(0.001,0.01,0.1,3,10)
```

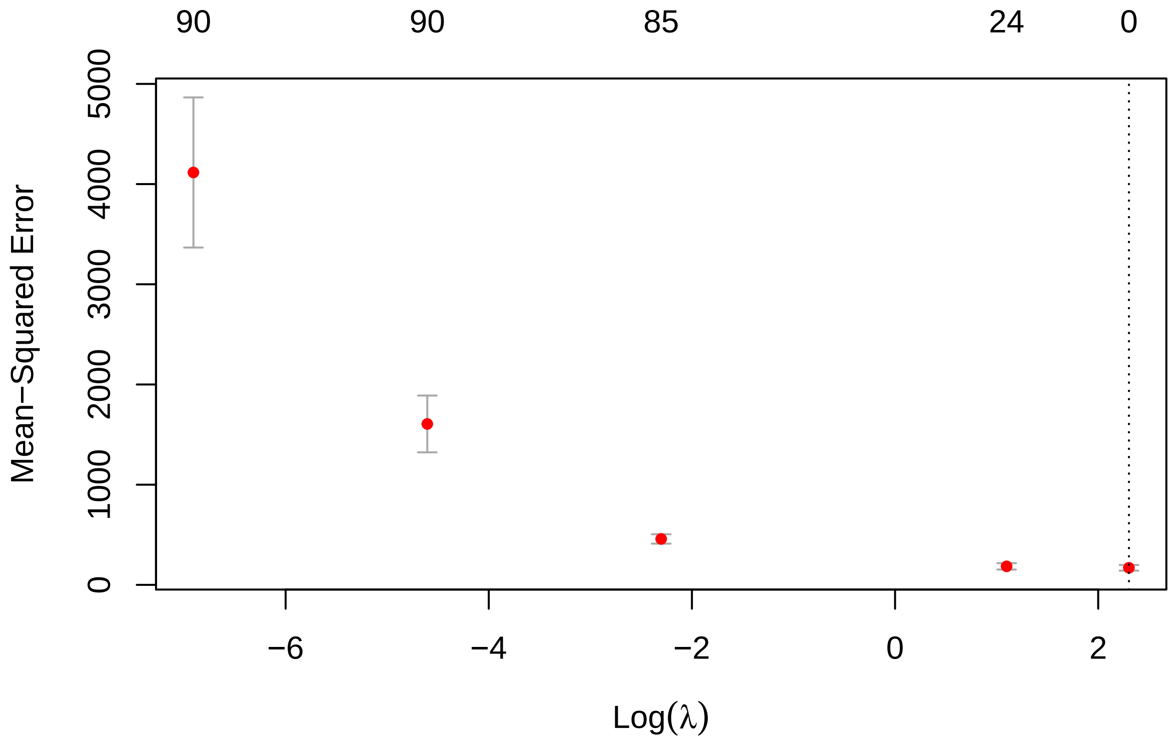
```
models=list()
for(i in 1:length(alphas)){
  models<-c(models,list(cv.glmnet(FirstHundred,y_100,alpha=alphas[i],lambda = lambdas)))
}
```

```
plot(models[[1]])
```



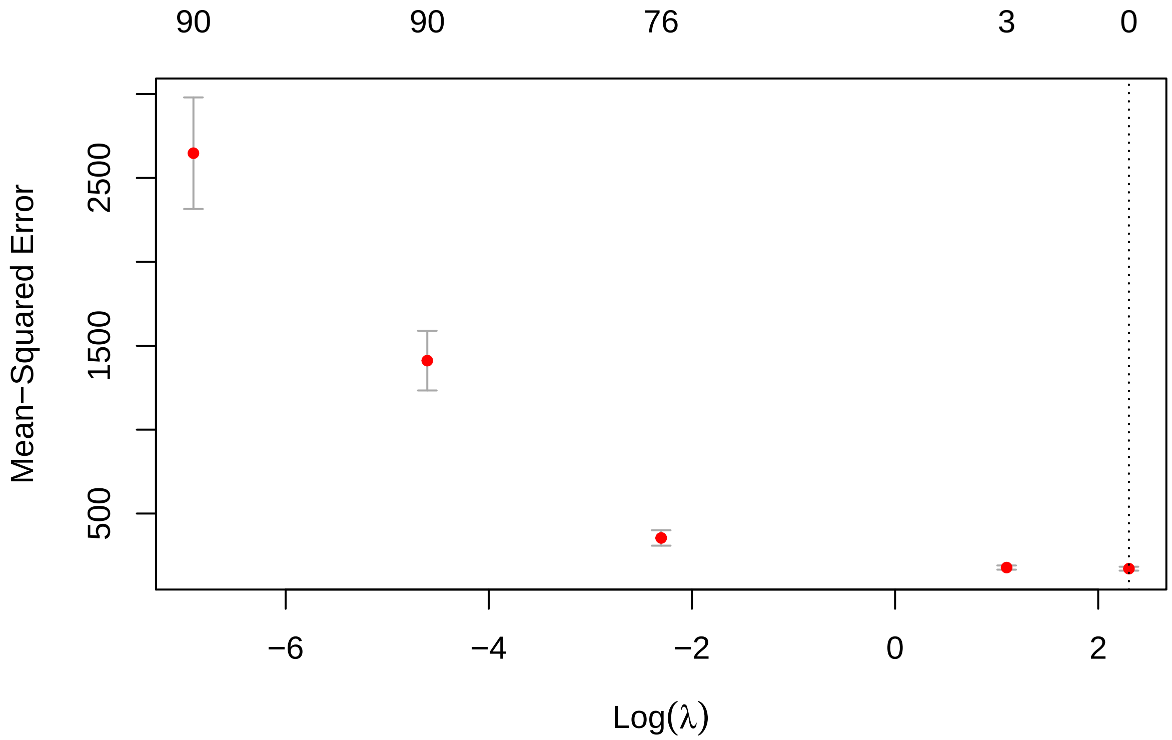
This is a ridge model where = 0. It has the lambda that gets the best MSE as 90.

```r
plot(models[[2]])
```



This is a elastic model and Log of (λ) is on the x-axis whereas Mean-Squared Error is on the y-axis. It has the lambda that gets the best MSE at 24, and the dotted line on the right is the MSE that is 1 standard error larger which is somewhat 0.

```
plot(models[[3]])
```



This is a pure lasso model with = 1 It has the lambda that gets the best MSE as 6. The dotted line on the right is the MSE that is 1 standard error larger which is approximately 0.

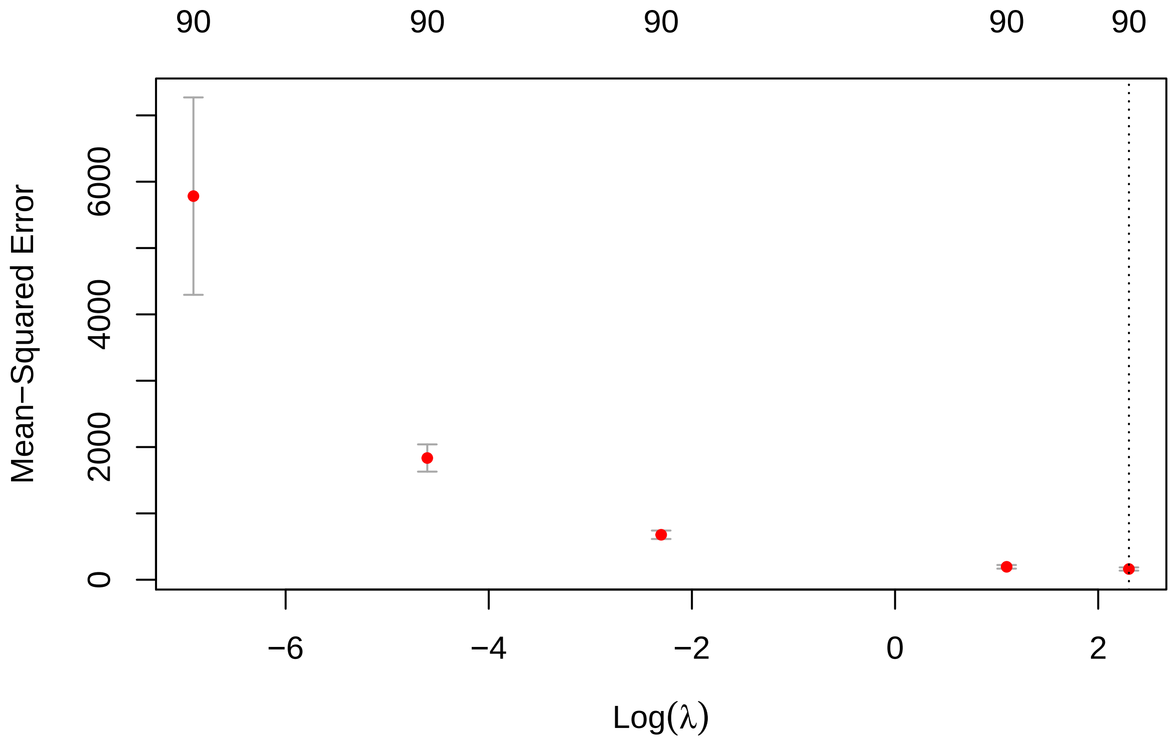
```
model_cvms<-sapply(models,function(x) return(x$cvm))
ind<-which.min(model_cvms)
best_model<-models[[ind]]
print(best_model)
```

```
##
## Call:  cv.glmnet(x = FirstHundred, y = y_100, lambda = lambdas, alpha = alphas[i])
##
## Measure: Mean-Squared Error
##
##     Lambda Measure    SE Nonzero
## min     10   161.7 24.55      90
## 1se     10   161.7 24.55      90
```

```
print(ind)
```

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

```
plot(best_model)
```



```
A<- c("Ridge", "Elastic_net","Lasso")
print(paste0("Best performing model is:", A[ind]))
```

```
## [1] "Best performing model is:Ridge"
```

Ridge was the best performing model, where alpha=0. The Best value of lambda is 10 and the SME is 90. I chose to use a few but spreaded lambda values instead of a range of values which is why the plot looks less filled.

b. How accurate are your coefficients from (a)? Summarize your results any way you like, but please don't give us the raw coefficients from 90 variables.

```
coeffs<-predict(best_model,type="coefficients",s=best_model$lambda.min)
coeffs
```

```
## 91 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  0.76921182
## V1           0.95279007
## V2           0.59553773
## V3           0.80365289
## V4           0.29399427
## V5           0.25705705
## V6          -0.05266660
## V7           0.76050112
## V8           0.83083134
## V9           1.09850971
## V10          1.25613326
## V11          0.84743444
## V12         -0.10789623
## V13          0.75696476
## V14          0.63429535
## V15          1.07259686
## V16         -0.35951418
## V17          1.51375055
## V18          0.45757116
## V19          1.42701730
## V20          0.68766171
## V21          1.14690027
## V22          0.89871372
## V23          0.19843160
## V24          0.31064767
## V25          0.26066044
## V26          0.43980266
## V27          0.55097285
## V28          0.61436655
## V29          0.21982513
## V30          1.19009853
## V31          0.21517783
## V32          0.03043592
## V33         -1.11046970
## V34          0.01173449
## V35          0.07632496
## V36         -0.86785963
## V37         -0.35148076
## V38         -0.15969476
## V39         -0.12362814
## V40         -0.80741865
## V41          0.79576070
## V42          0.55230116
## V43         -0.03556707
## V44         -1.04683546
## V45         -0.01155978
## V46         -0.79520384
## V47          0.20360783
## V48         -0.58646376
## V49         -0.76199382
## V50         -1.29525049
## V51          0.04698590
## V52         -0.16065540
## V53         -0.78922273
## V54          0.47981663
## V55          0.52832523
## V56         -0.07521056
## V57         -0.27197720
## V58         -0.13703929
## V59         -0.13933767
## V60         -0.32312575
## V61         -0.15297994
## V62         -0.96971006
## V63         -0.70804541
## V64         -0.31312498
## V65          0.99945093
## V66         -0.72132228
## V67         -0.34811735
## V68          0.10599922
## V69         -0.86029916
## V70          0.02448724
## V71         -0.99080111
## V72         -0.26852261
## V73          0.31262553
## V74         -0.65543355
## V75         -0.69503422
## V76          0.17042720
## V77          0.35901978
## V78          0.33555918
## V79         -0.07250452
## V80         -1.00405944
## V81          0.34528972
## V82          0.43614717
## V83         -0.65032701
## V84         -0.03244327
## V85         -0.47930634
## V86         -0.06223187
## V87         -0.41684561
## V88          0.32296018
## V89          0.58028336
## V90          0.61815102
```

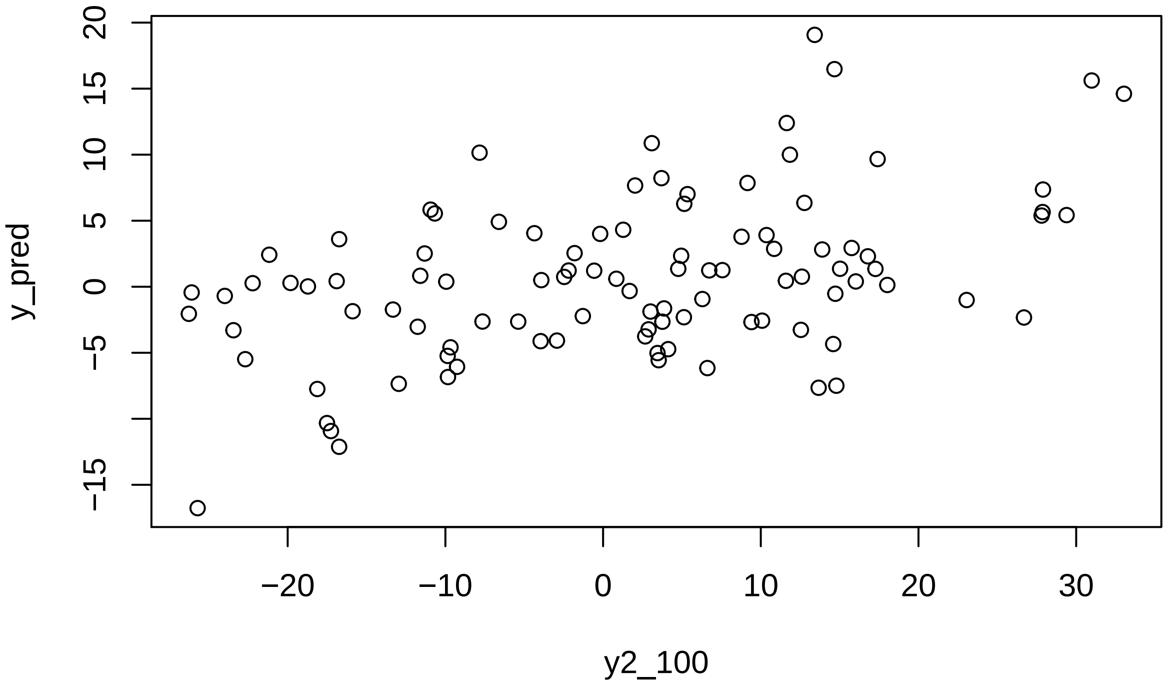
Inference: Since the best model is the ridge model, it wont shrink the last 30 values to zero, thus we wont get the desired value. With respect to top 60, many values are even close to 2 or -1 in their respective sections. Thus the accuracy of the coefficients is poor.

c. Using the results from (b), predict y for the second 100 observations. How accurate is your prediction?

```
y_pred<-predict(best_model,newx=x2_100,s=best_model$lambda.min)
y_pred
```

```
##                  1
##   [1,]   7.00928124
##   [2,]  16.47854377
##   [3,]  -6.83954730
##   [4,]  -2.05208412
##   [5,]   8.22216123
##   [6,]  -3.23015651
##   [7,]   9.99892426
##   [8,]  -2.31020243
##   [9,]   2.30121311
##  [10,]  10.87201159
##  [11,]   4.00322699
##  [12,]   3.78484135
##  [13,]  -3.76073445
##  [14,]   1.35026329
##  [15,]  15.61752797
##  [16,]  -1.64293704
##  [17,]   2.42249765
##  [18,]   7.66431773
##  [19,]   2.93326499
##  [20,]  -0.93330199
##  [21,]   0.13344968
##  [22,]   3.60107605
##  [23,]  -2.33042557
##  [24,]   1.25645409
##  [25,]   0.44860916
##  [26,]  -1.85502893
##  [27,]   0.42698270
##  [28,] -12.12259173
##  [29,]   9.66812272
##  [30,]   0.60270785
##  [31,]  -4.72837073
##  [32,]  -6.15970779
##  [33,]   0.28693141
##  [34,]   0.82965178
##  [35,]  -0.32562550
##  [36,]  -7.35176449
##  [37,]  -1.88051648
##  [38,]   0.50103190
##  [39,]  -2.56860805
##  [40,]  -2.63558927
##  [41,]   2.54365542
##  [42,]   2.34443247
##  [43,]   2.86868509
##  [44,]   0.76057669
##  [45,]  -7.49522723
##  [46,]   6.27756934
##  [47,]   7.85766954
##  [48,]  -0.69731937
##  [49,]  -0.43539009
##  [50,]   3.90931243
##  [51,]   2.51734840
##  [52,] -10.32678404
##  [53,]  -5.02512165
##  [54,]   6.34969252
##  [55,]  19.07200729
##  [56,]   5.54854641
##  [57,]   5.65413030
##  [58,]   5.42466001
##  [59,]  -7.74422696
##  [60,]   5.83842998
##  [61,]  -7.65220131
##  [62,]  12.39705919
##  [63,]   5.38981981
##  [64,]  -2.67663302
##  [65,]  -6.06473011
##  [66,]  -4.33968159
##  [67,]  14.61133451
##  [68,]   4.91412792
##  [69,]  -2.64148884
##  [70,]   0.74418777
##  [71,]  -2.64950564
##  [72,]  -3.26529297
##  [73,]  -4.12338025
##  [74,]  -5.48790684
##  [75,]   1.22643220
##  [76,] -16.76499092
##  [77,]  -2.22465398
##  [78,]   0.39963014
##  [79,]   2.82133726
##  [80,]  -0.53197101
##  [81,]  -3.03112097
##  [82,]  -1.72814796
##  [83,] -10.92905962
##  [84,]   1.23818497
##  [85,]  -5.23808372
##  [86,]   0.38453309
##  [87,]   0.26800589
##  [88,]   4.05223745
##  [89,]  -3.29364576
##  [90,]   1.21474055
##  [91,]   4.31346902
##  [92,]  10.15017830
##  [93,]  -4.07900789
##  [94,]   1.36061040
##  [95,]   1.35568792
##  [96,]  -5.56436207
##  [97,]   0.02221117
##  [98,]   7.36072486
##  [99,]  -1.00782577
## [100,]  -4.58922363
```

```
plot(y2_100,y_pred)
```



```
mse<-mean((y2_100-y_pred)^2)
mse
```

```
## [1] 164.8764
```

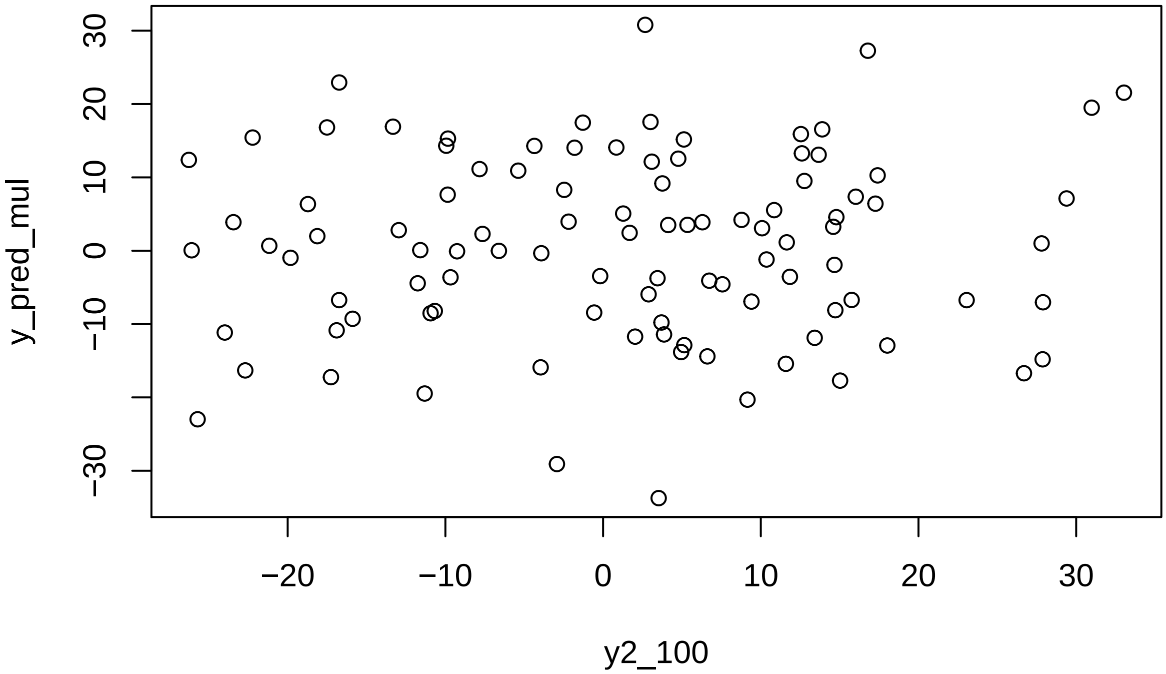
The MSE that we obtained is 178.11 and we saw in the previous question that our coefficients are not accurate. Thus, the model prediction is not very accurate.

d. Attempt to compare the predictive accuracy here to the accuracy of a prediction made using regular multiple regression. Explain your results, including if the regular regression failed for any reason.

```
mod<-lm(y_100 ~ FirstHundred)
mod
```

```
##
## Call:
## lm(formula = y_100 ~ FirstHundred)
##
## Coefficients:
##    (Intercept)   FirstHundred1   FirstHundred2   FirstHundred3
##        -4.9464         -2.1438         -2.7084          0.7227
##  FirstHundred4   FirstHundred5   FirstHundred6   FirstHundred7
##        -3.4666         -1.6323         -4.5912         -3.5088
##  FirstHundred8   FirstHundred9  FirstHundred10  FirstHundred11
##         4.0441          5.1868          4.2922          5.8906
## FirstHundred12  FirstHundred13  FirstHundred14  FirstHundred15
##         0.2553          5.2668          2.6216          4.1026
## FirstHundred16  FirstHundred17  FirstHundred18  FirstHundred19
##        -1.4529          4.4528          3.8748          9.1182
## FirstHundred20  FirstHundred21  FirstHundred22  FirstHundred23
##         4.6622          6.5963          4.4715         -6.0567
## FirstHundred24  FirstHundred25  FirstHundred26  FirstHundred27
##        -0.2737         -2.8513         -7.8261         -3.0777
## FirstHundred28  FirstHundred29  FirstHundred30  FirstHundred31
##         2.1500          1.3327          4.3683         -1.6621
## FirstHundred32  FirstHundred33  FirstHundred34  FirstHundred35
##         5.1473        -11.2334         -1.7906         -2.8891
## FirstHundred36  FirstHundred37  FirstHundred38  FirstHundred39
##         2.9808         -7.0012          2.5156         -0.4616
## FirstHundred40  FirstHundred41  FirstHundred42  FirstHundred43
##        -1.4100         -7.3051         -0.2628          1.9964
## FirstHundred44  FirstHundred45  FirstHundred46  FirstHundred47
##        -0.5204          1.3905         -2.1438         -5.1481
## FirstHundred48  FirstHundred49  FirstHundred50  FirstHundred51
##         1.8077          3.5337         -4.1970         -3.1639
## FirstHundred52  FirstHundred53  FirstHundred54  FirstHundred55
##         3.7644         -3.5212         -2.6646         -0.6540
## FirstHundred56  FirstHundred57  FirstHundred58  FirstHundred59
##        -6.3626          1.8388          0.7231          2.0217
## FirstHundred60  FirstHundred61  FirstHundred62  FirstHundred63
##        -8.7800         -0.1405         -2.0700         -3.4436
## FirstHundred64  FirstHundred65  FirstHundred66  FirstHundred67
##        -2.2794          5.1531          1.1906         -2.2116
## FirstHundred68  FirstHundred69  FirstHundred70  FirstHundred71
##        -3.0672         -0.7041         -1.5874         -3.1112
## FirstHundred72  FirstHundred73  FirstHundred74  FirstHundred75
##         0.4233          1.3899          3.3452          0.2022
## FirstHundred76  FirstHundred77  FirstHundred78  FirstHundred79
##        11.2151          6.2486         -0.5127          5.4300
## FirstHundred80  FirstHundred81  FirstHundred82  FirstHundred83
##        -1.2785          6.6610          2.8084         -2.8531
## FirstHundred84  FirstHundred85  FirstHundred86  FirstHundred87
##         1.1666          2.3886          4.0053          1.4534
## FirstHundred88  FirstHundred89  FirstHundred90
##         1.6248         -5.3222          6.0636
```

```
y_pred_mul<-predict(mod,newx=x2_100)
plot(y2_100,y_pred_mul)
```



```
mse_ml<-mean((y2_100-y_pred_mul)^2)
mse_ml
```

```
## [1] 345.1626
```

MSE of the multiple regression model is 498.065 which is way greater than the regularized model. Thus, looking at the mean squared values, regular regression performed better.

9. As in problem 6, use the data from 8 to generate a new y2 that is 1 if y > 0 and 0 otherwise.

a. Using the same process as in 8, estimate an SVM model of y2 on all the x variables for the first 100 variables. Use 10-fold cross-validation to select the best kernel.

```
w<-function(x){
   if(x>0)
   return(1)
   else
   return(0)
  }
y2<-sapply(y,w)
y2<-as.factor(y2)
dat<-as.data.frame(x)
dat['y2']<-y2
costs=c(0.001,0.01,0.1,1,10,100)
```

```r
library(e1071)
svm_lin<-tune(svm,y2~.,data=dat[1:100,],ranges = list(costs),kernel='linear')
svm_rad<-tune(svm,y2~.,data=dat[1:100,],ranges = list(costs),kernel='radial')
summary(svm_lin)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##   Var1
##  0.001
##
## - best performance: 0.37
##
## - Detailed performance results:
##    Var1 error dispersion
## 1 1e-03  0.37   0.105935
## 2 1e-02  0.37   0.105935
## 3 1e-01  0.37   0.105935
## 4 1e+00  0.37   0.105935
## 5 1e+01  0.37   0.105935
## 6 1e+02  0.37   0.105935
```

```r
summary(svm_rad)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##   Var1
##  0.001
##
## - best performance: 0.36
##
## - Detailed performance results:
##    Var1 error dispersion
## 1 1e-03  0.36  0.1349897
## 2 1e-02  0.36  0.1349897
## 3 1e-01  0.36  0.1349897
## 4 1e+00  0.36  0.1349897
## 5 1e+01  0.36  0.1349897
## 6 1e+02  0.36  0.1349897
```

As we can see in the above code's result, linear kernel has performed better with lower dispersion and error.

b. Using the results from (a), predict y2 for the second 100 observations, and report your accuracy

```
y_hat_svm<-predict(svm_lin$best.model,newdata=dat[101:200,])
table(predicted=y_hat_svm,actual=dat[101:200,'y2'])
```

```
##          actual
## predicted  0  1
##         0 27 22
##         1 16 35
```

```
#Accuracy:
accuracy<-sum(y_hat_svm==dat[101:200,'y2'])/length(y_hat_svm)
accuracy
```

```
## [1] 0.62
```

The code result includes the predicted y2 values and the confusion matrix. The accuracy I detected was over 0.50