Simon Chaisouang

Assignment 2:

Part A

(i)

To construct the appropriate X matrix for this problem, we first take the covariates of the five features (Cement, Water, Flyash, Coarsecomp, and Age\_index) associated with the data to represent each column in the X matrix. Since Age\_index is a categorical covariate, we want to express it numerically to use it in the regression model, so we divide the covariate into two indicator columns for each category in Age\_index. The two indicator columns are represented as Mid\_age and New, with Old being represented as the baseline category. Finally, the intercept column containing all values of 1’s is added to the beginning of the X matrix to serve as a baseline for our model.

We can now express the X matrix as (The full matrix can be found from the R code in Part B):

The X matrix contains 7 columns total that consist of an intercept column, 4 numerical covariates, and 2 indicator columns that represent the categorical data. The least square regression equation can be obtained and expressed as the following:

(ii)

Null Hypothesis (H0): =

Alternative Hypothesis (Ha): **≠** , At least one of the effects is nonzero.

Computing Test Statistics:

The formula above was used to calculate the test statistics which is approximately 106.7167. A was set to to show that we’re testing for the effect of the two categories of index in our model (Mid\_age and New), one being at least nonzero. q is set to 2 since two claims are being tested. The test statistic can then be taken to calculate as shown below:

The distribution of the test statistic under H0 follows an F-distribution with 2 and n-p-1 = 839 degrees of freedom. The p-value was calculated to be significantly less than 0.00001. Since the p-value is less than 0.05, at level 5%, there is enough statistical evidence to prove that at least one of the effects of the categories of age\_index is nonzero. Therefore, the null hypothesis is rejected.

(iii)

ANOVA Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Source of Variation | Sum of Squares (SS) | Degrees of freedom (df) | Mean sum of squares (MS) | R-square | F-statistic |
| Covariates | SSR =  79,333 | p = 6 | MSR = 13,222 | R2 = 0.756 | F = 433.4555 |
| Error | SSE = 25,593 | n – p – 1 = 839 | MSE = 30.5 |  |  |
| Total | SST = 104,926 | n – 1 = 845 |  |  |  |

(iv)

For a new data point with the given information, the mean CSS would be as follows:

The 95% C.I. and 95% P.I. are derived as follows:

(v)

The for the first model, with Coarsecomp, is as follows:

After rebuilding the regression model without Coarsecomp and computing the , the value for model 2 is as follows:

After obtaining the value for both models, we can see the two values barely differ by 0.0001. This tells us that having Coarsecomp as a covariate gives slightly less explanation about y versus having a model without the additional covariate. Since the increases in model 2, we can decide to remove Coarsecomp as a covariate from the model.

Part B

#### Forming the Data

regression\_data = read.table("Data\_Question\_2.txt",header=TRUE)

print(colnames(regression\_data))

y = matrix(regression\_data[ , 2],ncol=1)

n = length(y)

X = matrix(1,n,1)

colnames(X) = "Intercept"

X = cbind(X,regression\_data[, c(1, 4:6)])

X = as.matrix(X)

# Including Category Covariates

Cat\_covariate = factor(regression\_data[ , 3])

level\_names = levels(Cat\_covariate)

print(level\_names)

m = length(level\_names)

Z = matrix(0,n,(m-1))

for (i in 1:(m-1)) Z[ , i] = as.numeric(Cat\_covariate==level\_names[i])

colnames(Z) = level\_names[1:(m-1)]

X = cbind(X, Z)

#### Testing for nonzero effect on CSS from categories of age\_index

##Building Regression Model

p = ncol(X) - 1

## Add a 0 in the regression formula to tell R that you have included the intercept already.

output\_information = lm(y~0+X)

## what is the value of beta\_hat ?

beta\_hat = matrix(output\_information$coefficients,ncol=1)

## what is the value of Y\_hat

y\_hat = X%\*%beta\_hat

## what is the value of e

e = matrix(y-y\_hat,ncol=1)

## what is the value of sigmahat\_sqaure?

sigmahat\_square = sum(e^2)/(n-p-1)

## Suppose q claims together

q = 2

## Define the matrix A from the statement of the problem

A = matrix(0,q,p+1)

## Fill in the rows as appropriate

A[1,] = c(0,0,0,0,0,1,0)

A[2,] = c(0,0,0,0,0,0,1)

## and keep doing this until you have filled in q rows

## Now compute the test statistic

Dispersion = A%\*%solve(t(X)%\*%X)%\*%t(A)

test\_stat = t(A%\*%beta\_hat)%\*%solve(Dispersion)%\*%(A%\*%beta\_hat)/(q\*sigmahat\_square)

## Read the level of the test alpha from the statement of the question

alpha = 0.05

## Compute the p-value

p\_value = 1 - pf(test\_stat, df1 = q, df2 = n-p-1)

#### Constructing ANOVA

SSE = t(e)%\*%e

SSR = t(y\_hat)%\*%y\_hat - n\*(mean(y)^2.0)

SST = SSR + SSE

MSR = SSR/p

MSE = sigmahat\_square

Fstatistic = MSR/MSE

R\_square = SSR/SST

R\_square\_adjusted = ((n-1)\*R\_square - p )/(n-p-1)

#### Prediction for a new data point

x0 = matrix(c(1, 0.640, 0.541, 0.010, 0.404, 0, 0),ncol=1)

mean\_y0\_hat = t(x0)%\*%beta\_hat

## interval predictions

## (ii) confidence interval (1-alpha)

alpha = 0.05

variance\_compute01 = sigmahat\_square\* ( t(x0)%\*%solve(t(X)%\*%X)%\*%x0 )

## lower and upper boundaries

conf\_int\_L = mean\_y0\_hat -

qt(alpha/2,df=n-p-1,lower.tail=F)\*sqrt(variance\_compute01)

conf\_int\_U = mean\_y0\_hat +

qt(alpha/2,df=n-p-1,lower.tail=F)\*sqrt(variance\_compute01)

## (iii) predictive interval (1-alpha)

alpha = 0.05

variance\_compute02 = sigmahat\_square\* ( 1 + t(x0)%\*%solve(t(X)%\*%X)%\*%x0 )

## lower and upper boundaries

pred\_int\_L = mean\_y0\_hat -

qt(alpha/2,df=n-p-1,lower.tail=F)\*sqrt(variance\_compute02)

pred\_int\_U = mean\_y0\_hat +

qt(alpha/2,df=n-p-1,lower.tail=F)\*sqrt(variance\_compute02)

#### Exploring R\_a^2 with/without Coarsecomp

## R^2 adjusted for models with Coarsecomp

R\_square\_adjusted = ((n-1)\*R\_square - p )/(n-p-1)

##Computing R\_a^2 adjusted without Coarsecomp

y = matrix(regression\_data[ , 2],ncol=1)

n = length(y)

X = matrix(1,n,1)

colnames(X) = "Intercept"

X = cbind(X,regression\_data[, c(1, 4:5)])

X = as.matrix(X)

# Including Category Covariates

Cat\_covariate = factor(regression\_data[ , 3])

level\_names = levels(Cat\_covariate)

print(level\_names)

m = length(level\_names)

Z = matrix(0,n,(m-1))

for (i in 1:(m-1)) Z[ , i] = as.numeric(Cat\_covariate==level\_names[i])

colnames(Z) = level\_names[1:(m-1)]

X = cbind(X, Z)

## Rebuilding Regression Model

p = ncol(X) - 1

## Add a 0 in the regression formula to tell R that you have included the intercept already.

output\_information = lm(y~0+X)

## what is the value of beta\_hat ?

beta\_hat = matrix(output\_information$coefficients,ncol=1)

## what is the value of Y\_hat

y\_hat = X%\*%beta\_hat

## what is the value of e

e = matrix(y-y\_hat,ncol=1)

## what is the value of sigmahat\_sqaure?

sigmahat\_square = sum(e^2)/(n-p-1)

SSE = t(e)%\*%e

SSR = t(y\_hat)%\*%y\_hat - n\*(mean(y)^2.0)

SST = SSR + SSE

R\_square = SSR/SST

R\_square\_adjusted = ((n-1)\*R\_square - p )/(n-p-1)