Simon Chaisouang

Assignment 3:

Part A

(i)

In the following tables, we follow a series of procedures for determining if our model contains collinearity from our standardized X matrix containing 6 numerical covariates and 1 categorical covariate. First, we find the condition indices for each covariate and if any of the values are > 100, we declare that collinearity is present in the model. If collinearity is present, we compute the VIF for each covariate to determine which covariate to remove from our model. The corresponding covariate with the highest VIJ > 5 is removed from the model. The previous processes are repeated until collinearity is no longer present in the model.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Eigenvalues | 275 | | 27 | | | 23 | | 12 | | | 9 | | 8 | | | 7 | | 1 | | | 0 | | 0 | |
| Condition Indices | 1 | 10 | | | 12 | | 20 | | | 30 | | 33 | | | 37 | | 529 | | | 5741 | | 4889540 | | |
| Collinearity Present? | Yes | | | | | | | | | | | | | | | | | | | | | | | |
| Covariate-Specific VIFs | Covariate Name | | | Covariate\_01 | | | | | Covariate\_02 | | | | | Covariate\_03 | | | | | Covariate\_05 | | | | | Covariate\_06 |
| VIF | | | 27870 | | | | | 47 | | | | | 5 | | | | | 51 | | | | | 55 |
| Covariate Name | | | Covariate\_07 | | | | | Fall | | | | | Spring | | | | | Summer | | | | |  |
| VIF | | | 27627 | | | | | 128 | | | | | 3 | | | | | 3 | | | | |  |
| Covariate to be removed. | Covariate\_01 | | | | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Eigenvalues | 252 | 27 | | 23 | | 12 | 8 | | 7 | | 3 | 1 | | 0 |
| Condition Indices | 1 | 9 | | 11 | | 19 | 30 | | 34 | | 67 | 483 | | 5951 |
| Collinearity Present? | Yes | | | | | | | | | | | | | |
| Covariate-Specific VIFs | Covariate Name | | Covariate\_02 | | Covariate\_03 | | | Covariate\_05 | | Covariate\_06 | | |  | |
| VIF | | 6 | | 5 | | | 51 | | 55 | | |  | |
| Covariate Name | | Covariate\_07 | | Fall | | | Spring | | Summer | | |  | |
| VIF | | 1 | | 9 | | | 3 | | 3 | | |  | |
| Covariate to be removed. | Covariate\_06 | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Eigenvalues | 228 | 25 | | 23 | | 9 | | 8 | 5 | | 3 | | 1 |
| Condition Indices | 1 | 9 | | 10 | | 27 | | 30 | 42 | | 85 | | 442 |
| Collinearity Present? | Yes | | | | | | | | | | | | |
| Covariate-Specific VIFs | Covariate Name | | Covariate\_02 | | Covariate\_03 | | Covariate\_05 | | |  | |  | |
| VIF | | 5.69 | | 4.81 | | 1.07 | | |  | |  | |
| Covariate Name | | Covariate\_07 | | Fall | | Spring | | | Summer | |  | |
| VIF | | 1.02 | | 3.11 | | 2.63 | | | 3.06 | |  | |
| Covariate to be removed. | Covariate\_02 | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Eigenvalues | 200 | 25 | 23 | 8 | 6 | 5 | 3 |
| Condition Indices | 1 | 8 | 9 | 26 | 33 | 39 | 80 |
| Collinearity Present? | No | | | | | | |

After removing Covariate\_01, Convariate\_02, and Covariate\_06, our model is now free of collinearity/near collinearity since all condition indices are now less than 100.

(ii)

Equation of least square regression line:

(iii)

Null Hypothesis (H0): **βCOVARIATE\_03 = βCOVARIATE\_05 = βCOVARIATE\_07 = βFALL = βSPRING = βSUMMER =0**

Alternative Hypothesis (Ha): **βCOVARIATE\_03 or βCOVARIATE\_05 or βCOVARIATE\_07 or βFALL or βSPRING or βSUMMER ≠ 0**, at least one of the effects is nonzero.

Computing the Test Statistics:

The F-statistic was calculated to be approximately 119.5893. The test statistic for the significance of regression was calculated by taking the mean square due to regression (MSR) divided by the mean squared error (MSE) in R. The test statistics can then be taken to calculate the p-value using the formula below. q in this case is set 6, to test the 6 effects of the regression.

The distribution of the test statistic under H0 follows an F-distribution with 6 and n-p-1=93 degrees of freedom. The p-value was calculated to be significantly less than 0.0001 which is less than 0.01. This means at level 1%, there is enough statistical evidence to prove that at least one of the six covariates of our model must have a significant effect on the response variable y. Therefore, we reject the null hypothesis under significance level 1%.

Part B

## First, read the dataset as before

regression\_data = read.table("Data\_Question\_3.txt",header = TRUE)

## Read the names of variables in the data

colnames(regression\_data)

## how many observations

n = nrow(regression\_data)

X = matrix(1,n,1) ## start with the intercept column

colnames(X) = "Intercept"

X = cbind(X,regression\_data[, c(1:3, 6:8)])

X = as.matrix(X)

## Adding Categorical covariate

Cat\_covariate = factor(regression\_data[ , c(5)])

level\_names = levels(Cat\_covariate)

m = length(level\_names)

## So, we need to create (m-1) columns

## Let us exclude last category

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)

print(colnames(X))

## Standardizing Covariates

X\_standard = X

## Standardize starting from 2nd column

for (j in 2:ncol(X\_standard)) X\_standard[,j] =

(X[,j]-min(X[,j]))/(max(X[,j])-min(X[,j]))

eigenvalues\_total = eigen(t(X\_standard)%\*%X\_standard)$values

condition\_indices = max(eigenvalues\_total)/eigenvalues\_total

## Compute VIF for all X\_standard columns except the intercept

for (i in 2:ncol(X\_standard))

{

reg\_output = lm(X\_standard[,i]~0+X\_standard[,-c(i)])

e\_vector = matrix(reg\_output$residuals,ncol=1)

SSE = sum(e\_vector^2.0)

SST = sum(X\_standard[,i]^2.0) - n \*(mean(X\_standard[,i])^2.0)

R2 = (SST-SSE)/SST

print(1/(1-R2))

}

## If largest VIF is > 5, what is its position ?

X\_standard = X\_standard[ , -c(1 + 1)] ## adding 1

## is necessary because the intercept is in the first column, so start counting after that.

print(colnames(X\_standard)) ## which covariates remained in the X\_standard matrix ?

## Recomputing condition indicies and VIFs (2nd time)

eigenvalues\_total = eigen(t(X\_standard)%\*%X\_standard)$values

condition\_indices = max(eigenvalues\_total)/eigenvalues\_total

## Compute VIF for all X\_standard columns except the intercept

for (i in 2:ncol(X\_standard))

{

reg\_output = lm(X\_standard[,i]~0+X\_standard[,-c(i)])

e\_vector = matrix(reg\_output$residuals,ncol=1)

SSE = sum(e\_vector^2.0)

SST = sum(X\_standard[,i]^2.0) - n \*(mean(X\_standard[,i])^2.0)

R2 = (SST-SSE)/SST

print(1/(1-R2))

}

## If largest VIF is > 5, what is its position ?

X\_standard = X\_standard[ , -c(4 + 1)] ## adding 1

## is necessary because the intercept is in the first column, so start counting after that.

print(colnames(X\_standard)) ## which covariates remained in the X\_standard matrix ?

## Recomputing condition indicies and VIFs (3rd time)

eigenvalues\_total = eigen(t(X\_standard)%\*%X\_standard)$values

condition\_indices = max(eigenvalues\_total)/eigenvalues\_total

## Compute VIF for all X\_standard columns except the intercept

for (i in 2:ncol(X\_standard))

{

reg\_output = lm(X\_standard[,i]~0+X\_standard[,-c(i)])

e\_vector = matrix(reg\_output$residuals,ncol=1)

SSE = sum(e\_vector^2.0)

SST = sum(X\_standard[,i]^2.0) - n \*(mean(X\_standard[,i])^2.0)

R2 = (SST-SSE)/SST

print(1/(1-R2))

}

## If largest VIF is > 5, what is its position ?

X\_standard = X\_standard[ , -c(1 + 1)] ## adding 1

## is necessary because the intercept is in the first column, so start counting after that.

print(colnames(X\_standard)) ## which covariates remained in the X\_standard matrix ?

## Recomputing condition indicies and VIFs (4th time)

eigenvalues\_total = eigen(t(X\_standard)%\*%X\_standard)$values

condition\_indices = max(eigenvalues\_total)/eigenvalues\_total

## Final Model

print(colnames(X\_standard)) ## which covariates stay in the final collinearity free matrix ?

## Now, we can go back to using original values of covariates for the columns

## that remained after collinearity removal

X = X[ ,colnames(X\_standard)]

print(colnames(X))

## Now, we can determine what p should be

## do not determine the value of p before completing the collinearity

## analysis, as removal of any column from X matrix will change the value of p

p = ncol(X) - 1

## construct the response vector

y = matrix(regression\_data[ , 4],ncol=1)

## Obtain coefficients for Least Square Regression Line

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

## Test for significance of linear regression

## 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

MSR = SSR/p

MSE = sigmahat\_square

Fstatistic = MSR/MSE

p\_value = 1 - pf(Fstatistic, df1=(6) , df2=(n-p-1))