HW3

Homayoon Fotros

Question 1

Dimension of matrices:

$$A = \begin{bmatrix} 2 & 1 \\ 3 & 5 \\ 5 & 7 \\ 4 & 8 \end{bmatrix} \quad B = \begin{bmatrix} 6 \\ 9 \\ 3 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 3 & 8 \\ 8 & 6 \\ 5 & 1 \\ 2 & 4 \end{bmatrix}$$

A is 4×2, B is 4×1, C is 4×2.

a. $A + C = \begin{bmatrix} 5 & 9 \\ 11 & 11 \\ 10 & 8 \\ 6 & 12 \end{bmatrix}$

b. $A - C = \begin{bmatrix} -1 & -7 \\ -5 & -1 \\ 0 & 6 \\ 2 & 4 \end{bmatrix}$

c. $B^T A = \begin{bmatrix} 6 & 9 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 3 & 5 \\ 5 & 7 \\ 4 & 8 \end{bmatrix} = \begin{bmatrix} 12+27+15+4 & 6+45+21+8 \end{bmatrix} = \begin{bmatrix} 58 & 80 \end{bmatrix}$

d. $AC^T = \begin{bmatrix} 2 & 1 \\ 3 & 5 \\ 5 & 7 \\ 4 & 8 \end{bmatrix} \begin{bmatrix} 3 & 8 & 5 & 2 \\ 8 & 6 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 6+8 & 16+6 & 10+1 & 4+4 \\ 9+40 & 24+30 & 25+7 & 8+32 \\ 15+56 & 40+42 & 25+7 & 10+28 \\ 12+64 & 32+48 & 20+8 & 8+32 \end{bmatrix} = \begin{bmatrix} 14 & 22 & 11 & 8 \\ 49 & 54 & 20 & 26 \\ 71 & 82 & 32 & 38 \\ 76 & 80 & 28 & 40 \end{bmatrix}$

Question 2

Loan Financing Company, Linear Regression Model.

a,b. The linear regression model is $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ where $\epsilon_i \, ^{iid}N(0, \sigma^2)$.

Using matrix notation, the model is: $\mathbf{Y} = \mathbf{X}\beta + \epsilon$, where $\mathbf{Y}$ is the response vector, β is parameters vector, $\mathbf{X}$ is the matrix of explanatory variables (a.k.a *design matrix*), and ϵ is the error vector.

$$X = \begin{bmatrix} 1 & 4 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 3 \\ 1 & 22 \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \quad Y = \begin{bmatrix} 16 \\ 5 \\ 10 \\ 15 \\ 13 \\ 22 \end{bmatrix}$$

c. Matrix Calculation

```
X <- matrix(c(1,1,1,1,1,1,4,1,2,3,3,22), nrow = 6, ncol = 2)
Y <- matrix(c(16,5,10,15,13,22), nrow = 6, ncol = 1)
```

i. $X^T X$

```
m_1 <- t(X) %*% X

m_1
```

```
##      [,1] [,2]
## [1,]    6   35
## [2,]   35  523
```

ii. $(X^T X)^{-1}$

```
m_2 <- solve(t(X) %*% X)

m_2
```

```
##             [,1]         [,2]
## [1,]  0.27339258 -0.018295870
## [2,] -0.01829587  0.003136435
```

iii. $X^T Y$

```
m_3 <- t(X) %*% Y

m_3
```

```
##      [,1]
## [1,]   81
## [2,]  657
```

iv. $\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} = (X^T X)^{-1} X^T Y$

```
B <- m_2 %*% m_3

B
```

```
##            [,1]
## [1,] 10.1244119
## [2,]  0.5786722
```

v. $\hat{Y}$ (predicted values)

```
Y_hat <- X %*% B

Y_hat
```

```
##          [,1]
## [1,] 12.43910
## [2,] 10.70308
## [3,] 11.28176
## [4,] 11.86043
## [5,] 11.86043
## [6,] 22.85520
```

Question 3

Fitting linear regression models on data sets

```
library(rio)
df_1 <- import('data1.csv')
df_2 <- import('data2.csv')
df_3 <- import('data3.csv')

#Setting up RMSE metric for evaluating models
RMSE <- function(prd, tr) sqrt(mean((prd-tr)^2))

## Splitting data into train and test sets
set.seed(20, sample.kind = 'Rounding')

idx <- sample(100, 50, replace = F)
train_1 <- df_1[sort(idx),]
test_1 <- df_1[-sort(idx),]

## Fitting Linear Regression Model on data1
mdl_1 <- lm(y ~ x, data = train_1)
ft_1 <- fitted(mdl_1)

## RMSE for training set (data-1, linear model)
RMSE_tr1 <- RMSE(ft_1, train_1$y)
RMSE_tr1
```

```
## [1] 5.992493
```

```
## RMSE for test set (data-1, linear model)
prd_1 <- predict(mdl_1, newdata = data.frame(x=test_1$x))

RMSE_tst1 <- RMSE(prd_1, test_1$y)
RMSE_tst1
```

```
## [1] 5.552712
```

```
### Cubic Model
mdl_1cub <- lm(y ~ poly(x,3, raw = T), data = train_1)
ft_1cub <- fitted(mdl_1cub)

## RMSE for training set (data-1, cubic model)
RMSE_tr1cub <- RMSE(ft_1cub, train_1$y)
RMSE_tr1cub
```

```
## [1] 5.830943
```

```
## RMSE for test set (data-1, cubic model)
prd_1cub <- predict(mdl_1cub, newdata = data.frame(x=test_1$x))

RMSE_tst1cub <- RMSE(prd_1cub, test_1$y)
RMSE_tst1cub
```

```
## [1] 6.011775
```

In the linear model, the RMSE values for training and test sets are 5.99 and 5.55 respectively. In the cubic model, the RMSE values for training and test sets are 5.83 and 6.01 respectively. These results show the linear model provided a lower and thus a better fit for this data set (data-1).

Applying the same process on `data-2`.

```
## Splitting data into train and test sets
train_2 <- df_2[sort(idx),]
test_2 <- df_2[-sort(idx),]

## Fitting Linear Regression Model on data2
mdl_2 <- lm(y ~ x, data = train_2)
ft_2 <- fitted(mdl_2)

## RMSE for training set (data-2, linear model)
RMSE_tr2 <- RMSE(ft_2, train_2$y)
RMSE_tr2
```

```
## [1] 5.350207
```

```
## RMSE for test set (data-2, linear model)
prd_2 <- predict(mdl_2, newdata = data.frame(x=test_2$x))

RMSE_tst2 <- RMSE(prd_2, test_2$y)
RMSE_tst2
```

```
## [1] 5.88625
```

```
### Cubic Model
mdl_2cub <- lm(y ~ poly(x,3, raw = T), data = train_2)
ft_2cub <- fitted(mdl_2cub)

## RMSE for training set (data-2, cubic model)
RMSE_tr2cub <- RMSE(ft_2cub, train_2$y)
RMSE_tr2cub
```

```
## [1] 5.134698
```

```
## RMSE for test set (data-2, cubic model)
prd_2cub <- predict(mdl_2cub, newdata = data.frame(x=test_2$x))

RMSE_tst2cub <- RMSE(prd_2cub, test_2$y)
RMSE_tst2cub
```

```
## [1] 5.724945
```

In the linear model, the RMSE values for training and test sets are 5.35 and 5.89 respectively. In the cubic model, the RMSE values for training and test sets are 5.13 and 5.72 respectively. These results show the cubic model provided a lower and thus a better fit for this data set (data-2).

Applying the same process on `data-3`.

```
## Splitting data into train and test sets
train_3 <- df_3[sort(idx),]
test_3 <- df_3[-sort(idx),]


## Fitting Linear Regression Model on data3
mdl_3 <- lm(y ~ x, data = train_3)
ft_3 <- fitted(mdl_3)


## RMSE for training set (data-3, linear model)
```

```
RMSE_tr3 <- RMSE(ft_3, train_3$y)
RMSE_tr3
```

```
## [1] 10.53676
```

```
## RMSE for test set (data-3, linear model)
prd_3 <- predict(mdl_3, newdata = data.frame(x=test_3$x))

RMSE_tst3 <- RMSE(prd_3, test_3$y)
RMSE_tst3
```

```
## [1] 9.073641
```

```
### Cubic Model
mdl_3cub <- lm(y ~ poly(x,3, raw = T), data = train_3)
ft_3cub <- fitted(mdl_3cub)

## RMSE for training set (data-3, cubic model)
RMSE_tr3cub <- RMSE(ft_3cub, train_3$y)
RMSE_tr3cub
```

```
## [1] 5.196431
```

```
## RMSE for test set (data-3, cubic model)
prd_3cub <- predict(mdl_3cub, newdata = data.frame(x=test_3$x))

RMSE_tst3cub <- RMSE(prd_3cub, test_3$y)
RMSE_tst3cub
```

```
## [1] 5.788733
```

In the linear model, the RMSE values for training and test sets are 10.54 and 9.07 respectively. In the cubic model, the RMSE values for training and test sets are 5.20 and 5.79 respectively. These results show the cubic model provided a *significantly* lower and thus a better fit for this data set (data-3).

Table 1: RMSE Values

	Train_Linear	Test_Linear	Train_Cubic	Test_Cubic
data-1	5.99	5.55	5.83	6.01
data-2	5.35	5.89	5.13	5.72
data-3	10.54	9.07	5.20	5.79

Which curve belongs to each data set?

The cubic model best fits with the data-3, while the RMSE values for the linear model on this data set is considerably larger than the same metric for other data sets. Therefore, we can state that data-3 is likely generated by a cubic relationship (e.g., black curve). The linear model's fit for data-1's test set performs significantly better than the cubic model. Therefore, data-1 is likely generated by a linear relationship (e.g., orange line). The remaining data set, data-2, yields close values of test RMSE when fitted with linear and cubic models. Therefore, it was likely generated by a relationship like the green curve.

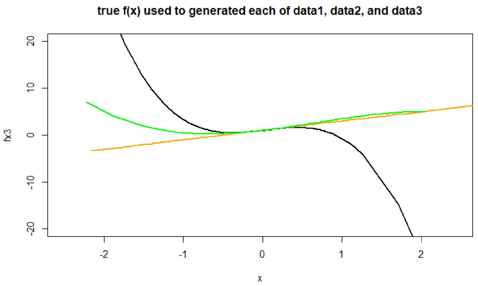


Figure 1: Data Curves

Question 4

Multicollinearity and Simulation

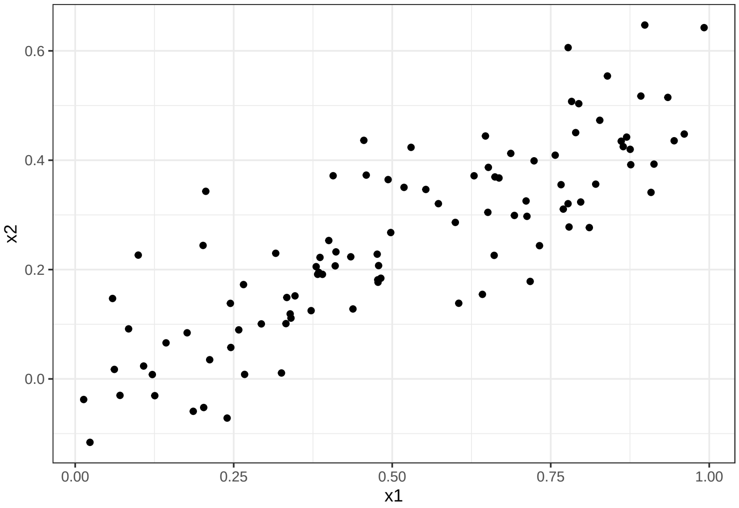
a. Generating a linear model

```
set.seed (1, sample.kind = 'Rounding')

x1 <- runif(100)
x2 <- 0.5*x1 + rnorm(100)/10
y <- 2 + 2*x1 + 0.3*x2 + rnorm(100)
```

The linear model is $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$ where the intercept is $\beta_0 = 2$ and coefficients are $\beta_1 = 2, \beta_2 = 0.3$.

b. The correlation between x_1 and x_2 is `0.84`.



c. Fitting Linear Regression using x_1 and x_2

```
q4_mdl <- lm(y ~ x1 + x2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	2.13	0.232	9.188	0.000
x1	1.44	0.721	1.996	0.049
x2	1.01	1.134	0.891	0.375

The estimated coefficients are $\hat{\beta}_1 = 1.44$ and $\hat{\beta}_2 = 1.01$. The estimated coefficients differ from those specified as true values. The t-stat value for β_1 estimate is large (p.value $= 0.0487$) and statistically significant at 0.05 level. Therefore, we can reject the null hypothesis ($H_0 : \beta_1 = 0$). The t-stat value for β_2 is small (p.value $=$ 0.3754) and not statistically significant. Therefore, we cannot reject the null hypothesis ($H_0 : \beta_2 = 0$).

d. Fitting Linear Regression using only x_1

```
q4d_mdl <- lm(y ~ x1)
```

term	estimate	std.error	statistic	p.value
(Intercept)	2.112	0.231	9.155	0
x1	1.976	0.396	4.986	0

The estimated coefficients is $\hat{\beta_1} = 1.976$, which is almost identical to the true value. The t-stat value for β_1 estimate is large (p.value ~ 0) and statistically significant at 0.01 level. Therefore, we can reject the null hypothesis ($H_0 : \beta_1 = 0$).

e. Fitting Linear Regression using only x_2

```r
q4e_mdl <- lm(y ~ x2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	2.39	0.195	12.261	0
x2	2.90	0.633	4.580	0

The estimated coefficients is $\hat{\beta_2} = 2.900$, which is very different from its true value. The t-stat value for β_2 estimate is large (p.value ~ 0) and statistically significant at 0.01 level. Therefore, we can reject the null hypothesis ($H_0 : \beta_2 = 0$).

f. Are the above results (parts c to e) contradictory?

The above results may seem at first contradictory, but it actually illustrates the issue of multicollinearity in regression models where the predicting features are highly correlated. The reason that using either x_1 or x_2 in the regression model yields significant estimates is because these two variables are highly correlated. Using both of them in a regression model will make one of them statistically significant as one of them (e.g., x_1) can absorb the other variable's effect on the outcome. In other words, if we 'partial out' x_1 from x_2, nothing significant remains for the latter to affect the outcome.