Project#3 Data Modeling

Aesha Rathod || Gurjinderpal Singh || Tehsin Shaikh

Contents

PROJECT OVERVIEW

Data modeling is the process of creating a conceptual representation of data structures and relationships. Linear regression is a statistical method used for data modeling and is used to model the relationship between a dependent variable and one or more independent variables. The goal of linear regression is to create a model that fits the data well and can be used to make predictions on new data.

Before building a linear regression model, it's important to understand the data, including the relationships between variables and any patterns that may exist. Once the data has been explored, a linear regression model can be built by selecting the appropriate independent variables, estimating the coefficients using least squares regression, and testing the model assumptions.

IMPORTING DEPENDENCIES

```r
# Import the necessary libraries and packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(readxl)
library(ggplot2)
library(dplyr)
library(stringr)
```

```r
# Load the dataset
library(readxl)
smd <- read_excel("C:/Users/MY PC/Desktop/DAB501_Project#2/Datasets/smd.xlsx")
```

DATA PREPROCESSING

```r
# change column names to make it easy to understand for R
colnames(smd) <-c("Order_Id","Company_Id","Company_Name","Date_ order_placed","DD_Order_Placed_Date","M
print(colnames(smd))
```

```
##  [1] "Order_Id"               "Company_Id"             "Company_Name"
##  [4] "Date_ order_placed"     "DD_Order_Placed_Date"   "MM _Order_Placed_Month"
##  [7] "YYYY_Order_Placed_Year" "Order_Value"            "Converted"
## [10] "Date"                   "Meal_Id"                "Date_of_Meal"
## [13] "YYYY_Meal_Served_Year"  "MM_Meal_Served_Month"   "DD_Meal_Served_Date"
## [16] "Participant"            "Meal_Price"             "Type_of_Meal"
## [19] "Sales_Rep"              "Sales_Rep_Id"
```

```r
# check for missing values in the entire data frame
sum(is.na(smd))
```

```
## [1] 449847
```

```r
# check for missing values in each column of the data frame
colSums(is.na(smd))
```

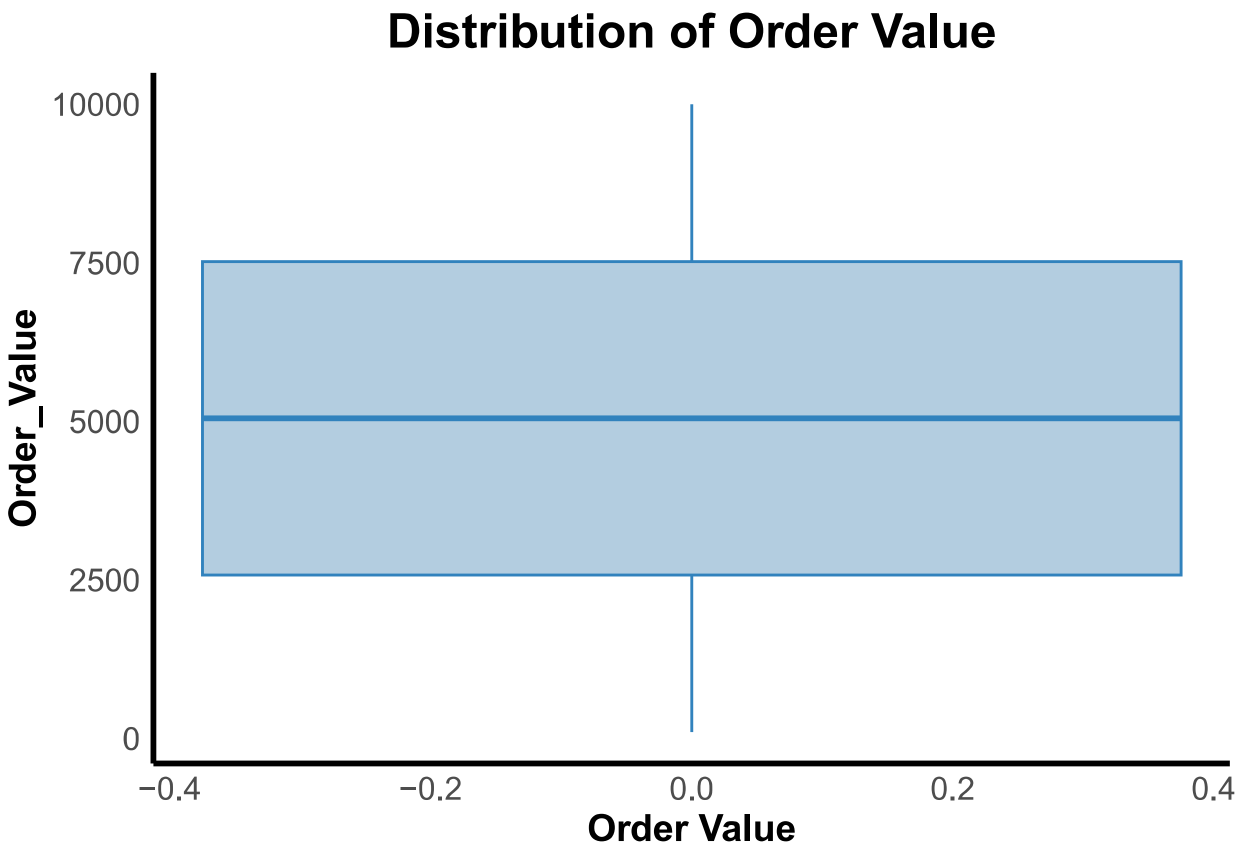
```
##               Order_Id             Company_Id           Company_Name
##                      0                      0                      0
##     Date_ order_placed   DD_Order_Placed_Date MM _Order_Placed_Month
##                      0                      0                      0
## YYYY_Order_Placed_Year            Order_Value              Converted
##                      0                      0                      0
##                   Date                Meal_Id           Date_of_Meal
##                  49983                  49983                  49983
##  YYYY_Meal_Served_Year   MM_Meal_Served_Month    DD_Meal_Served_Date
##                  49983                  49983                  49983
##            Participant             Meal_Price           Type_of_Meal
##                  49983                  49983                  49983
##              Sales_Rep           Sales_Rep_Id
##                      0                      0
```

```r
# Remove missing values from the data frame
smd <-na.omit(smd)
```

```r
# check for missing values in each column of the data frame after clean up
colSums(is.na(smd))
```

```
##               Order_Id             Company_Id           Company_Name 
##                      0                      0                      0 
##     Date_ order_placed   DD_Order_Placed_Date MM _Order_Placed_Month 
##                      0                      0                      0 
## YYYY_Order_Placed_Year            Order_Value              Converted 
##                      0                      0                      0 
##                   Date                Meal_Id           Date_of_Meal 
##                      0                      0                      0 
##  YYYY_Meal_Served_Year   MM_Meal_Served_Month    DD_Meal_Served_Date 
##                      0                      0                      0 
##            Participant             Meal_Price           Type_of_Meal 
##                      0                      0                      0 
##              Sales_Rep           Sales_Rep_Id 
##                      0                      0
```

```r
# Outliers detection and handling
ggplot(data = smd, mapping = aes(y = Order_Value)) +
  geom_boxplot(fill = "#b3cde0", color = "#3182bd") +
  theme_minimal() +
  labs(x = "Order Value", title = "Distribution of Order Value") +
  theme(plot.title = element_text(size = 18, face = "bold", hjust = 0.5),
        axis.text = element_text(size = 12),
        axis.title = element_text(size = 14, face = "bold"),
        axis.line = element_line(linewidth = 1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```



```r
# Calculate the z-score for the Order Value column
z <- abs(scale(smd$Order_Value))

# Identify the index of any z-score greater than 3 (typical threshold for identifying outliers)
outliers <- which(z > 3)

# Print the outliers
smd[outliers,]
```

```
## # A tibble: 0 x 20
## # ... with 20 variables: Order_Id <chr>, Company_Id <chr>, Company_Name <chr>,
## #   Date_ order_placed <chr>, DD_Order_Placed_Date <dbl>,
## #   MM _Order_Placed_Month <dbl>, YYYY_Order_Placed_Year <dbl>,
## #   Order_Value <dbl>, Converted <dbl>, Date <chr>, Meal_Id <chr>,
## #   Date_of_Meal <chr>, YYYY_Meal_Served_Year <dbl>,
## #   MM_Meal_Served_Month <dbl>, DD_Meal_Served_Date <dbl>, Participant <chr>,
## #   Meal_Price <dbl>, Type_of_Meal <chr>, Sales_Rep <chr>, ...
```

If there are no rows in the dataset with a z-score greater than 3, it means that there are no outliers in the dataset according to the z-score method.

MODELING

Question 1

*1. **Identify the explanatory variable:*** “Meal_Price”

Question 2

Identify the response variable: “Order_Value”

Question 3

Create a linear regression model and display the full output of the model.

```
model <- lm(Order_Value ~ Meal_Price, data = smd)
summary(model)
```

```
##
## Call:
## lm(formula = Order_Value ~ Meal_Price, data = smd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3270.7  -314.1   -95.0   309.4  3911.3
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 246.22846    5.64910   43.59   <2e-16 ***
## Meal_Price   11.89778    0.01211  982.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 634.3 on 50015 degrees of freedom
## Multiple R-squared:  0.9507, Adjusted R-squared:  0.9507
## F-statistic: 9.645e+05 on 1 and 50015 DF,  p-value: < 2.2e-16
```

Based on the results of the linear regression model, we can conclude that changes in meal price do affect the total order value. The coefficient of the Meal_Price variable is positive (11.89778), which means that for every one unit increase in meal price, the order value is predicted to increase by 11.89778 units on average. The p-value for this coefficient is very small (<2.2e-16), which indicates that it is statistically significant and unlikely to have occurred by chance. The high R-squared value (0.9507) also suggests that the model fits the data well and can explain a large proportion of the variability in the total order value.

Question 4

Using the variables noted in #1 and #2 above and the results of #3, write the equation for your model.

The equation for the linear regression model is:

Order_Value = b0 + b1Meal_Price

Where b0 is the intercept, and b1 is the coefficient for the independent variable Meal_Price.

Question 4

Explain what the intercept means in the context of the data.

In the context of the data, the intercept (0) represents the expected value of Order_Value when the Meal_Price is zero. However, it is important to note that in this case, a Meal_Price of zero is not a meaningful or realistic value. In practical terms, the intercept represents the fixed costs or other factors that contribute to the Order_Value, independent of the Meal_Price. For example, it could represent the cost of labor, rent, utilities, or other overhead expenses associated with preparing and delivering meals.

Question 6

Is the intercept a useful/meaningful value in the context of our data? If yes, explain. If not,explain what purpose it serves.

In a linear regression model, the intercept represents the value of the response variable (Order_Value) when the explanatory variable (Meal_Price) is zero. However, in the context of our data, the intercept value of 246.22846 does not make sense because the Meal_Price variable cannot be zero in our dataset.

Therefore, the intercept does not have a meaningful interpretation in the context of our data. However, the intercept serves the purpose of providing the starting point for the regression line, which is important for estimating the coefficients and making predictions based on the model. Additionally, the intercept is used in calculating the residual sum of squares (RSS) and the residual standard error (RSE) of the model.

Question 7

Explain what the slope means in the context of the data.

The slope in the context of the data represents the change in the mean Order_Value for each one-unit increase in Meal_Price. In other words, it indicates the rate of change in the Order_Value for each additional unit increase in the Meal_Price.

In this case, the slope coefficient is 11.89778, which means that for each one dollar increase in the Meal_Price, the Order_Value is expected to increase by $11.90, on average. This implies that there is a positive and significant relationship between the Meal_Price and the Order_Value. As the Meal_Price increases, the Order_Value also increases, indicating that customers are willing to pay more for their meals.

MODEL DIAGNOSTICS

Question 1

Create two new data columns based on your best model: predicted values for your response variable and the corresponding residuals.

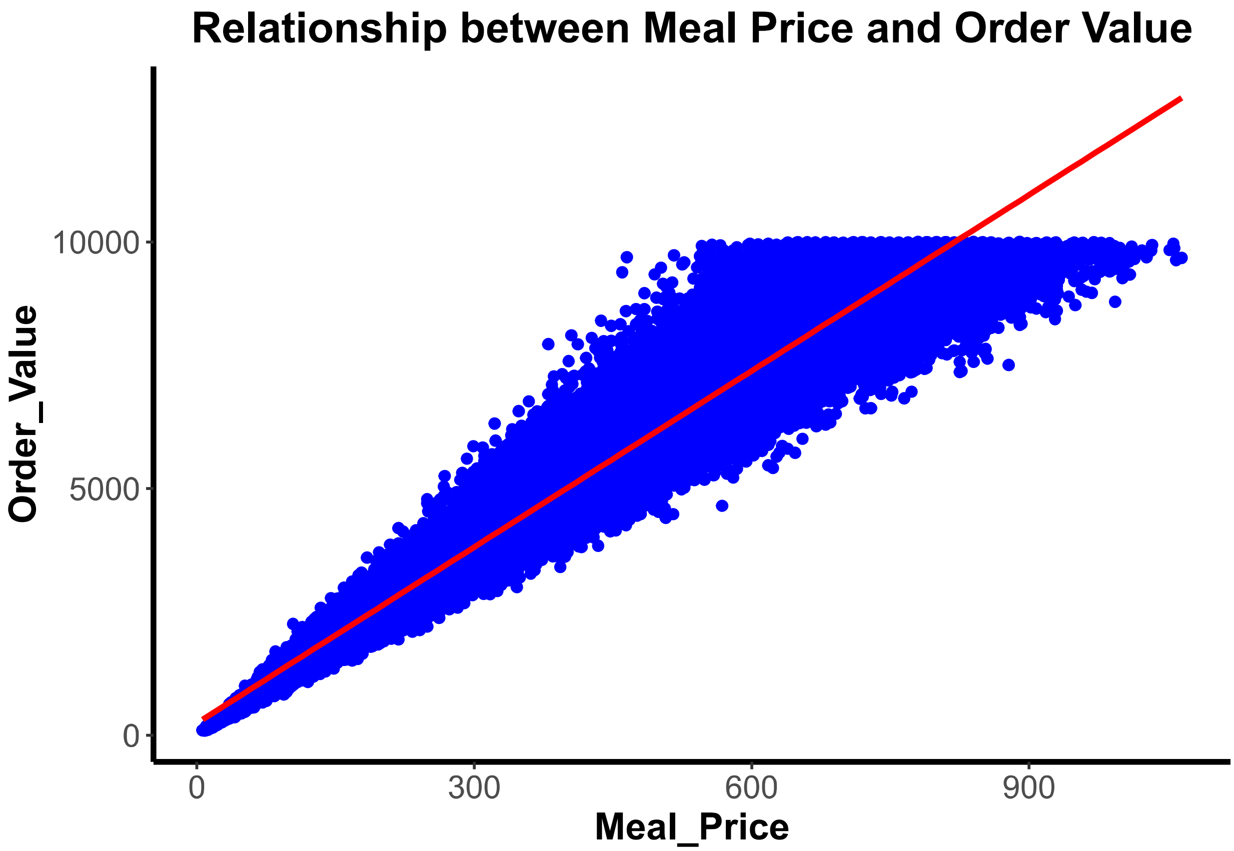
```r
# calculate predicted values
smd$predicted_value <- predict(model, newdata = smd)
view(smd)
# calculate residuals
smd$residuals <- smd$Order_Value - smd$predicted_value
```

Question 2

Create a plot to check the assumption of linearity. State whether or not this condition is met and explain your reasoning

```
ggplot(data = smd, aes(x = Meal_Price, y = Order_Value)) +
     geom_point(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
    labs(title = "Relationship between Meal Price and Order Value") +
    theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
        axis.text = element_text(size = 12),
        axis.title = element_text(size = 14, face = "bold"),
        axis.line = element_line(linewidth = 1),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_rect(fill = "white"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



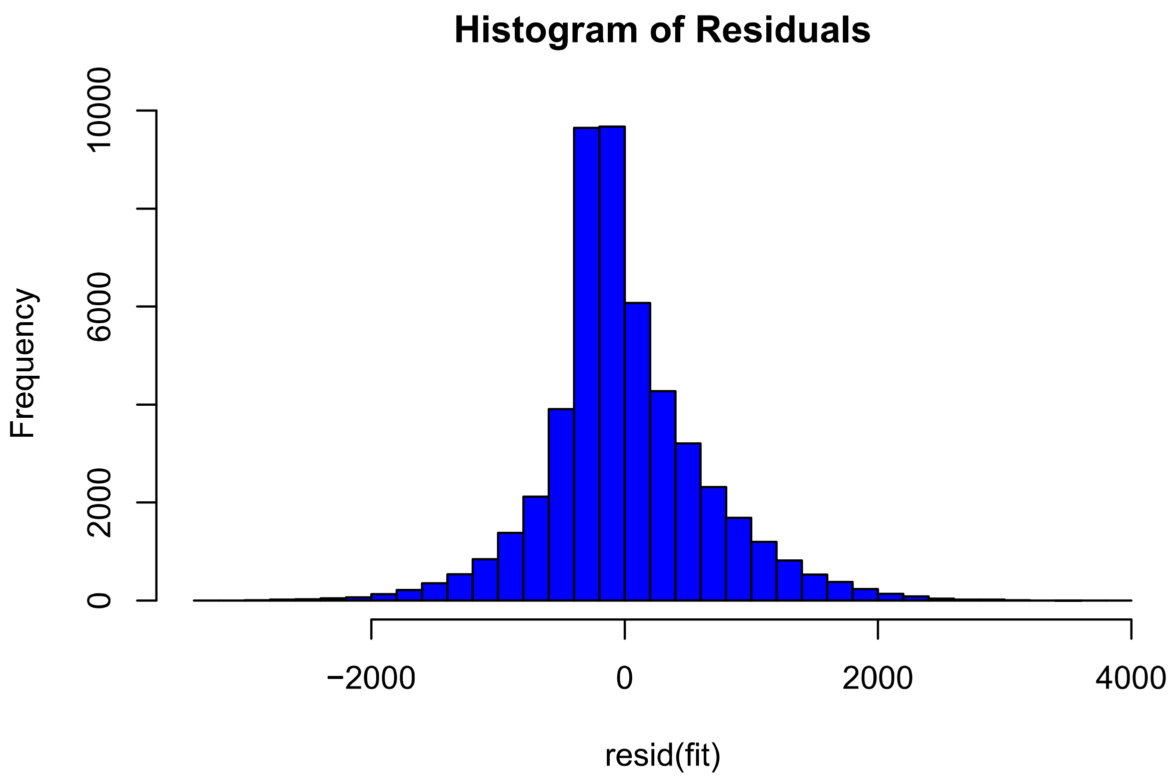
Based on the scatter plot with a regression line, we can see that there appears to be a linear relationship between the Meal Price and Order Value variables in the dataset. The regression line is also a good fit to the data points, indicating that the assumption of linearity is met. Therefore, we can use a linear regression model to make predictions about Order Value based on Meal Price.

Question 3

Create a plot to check the assumption of nearly normal residuals. State whether or not this condition is met and explain your reasoning.

```r
# Fit the linear regression model
fit <- lm(Order_Value ~ Meal_Price, data = smd)

# Create a histogram of the residuals
hist(resid(fit), breaks = 30, col = "Blue", main = "Histogram of Residuals")
```

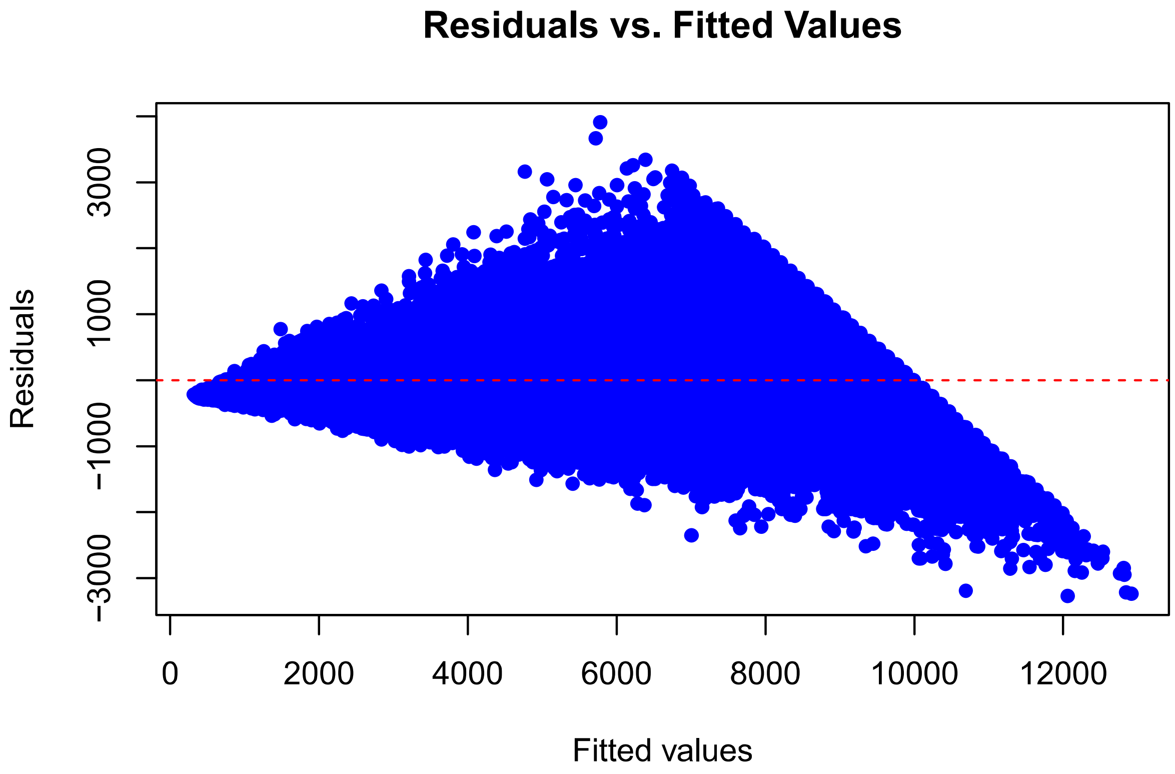


In the histogram, we want to see a roughly normal distribution of the residuals. In this case, the histogram does show some skewness to the right, but it's not severe enough to be a major concern.Therefore, we can conclude that the assumption of nearly normal residuals is approximately met.

Question 4

Create a plot to check the assumption of constant variability. State whether or not this condition is met and explain your reasoning.

```r
fit <- lm(Order_Value ~ Meal_Price, data = smd)
plot(fit$fitted.values, resid(fit), col = "blue", pch = 16,
     xlab = "Fitted values", ylab = "Residuals", main = "Residuals vs. Fitted Values")
abline(h = 0, col = "red", lty = 2)
```



In the given plot, we can see that the residuals are randomly scattered around the horizontal line at zero, which is the line of no residual error. There is no clear pattern of the residuals becoming larger or smaller as the fitted values increase or decrease. The spread of the residuals appears to be fairly consistent across all levels of the fitted values, indicating that the assumption of constant variability is likely met.

Therefore, based on this plot, we can assume that the condition of constant variability is met, and we can proceed with other assumptions of the linear regression model.

CONCLUSION

Based on the results of the "Model Diagnostics" section above, what can you conclude about your model?

Based on the results of the "Model Diagnostics" section, we can conclude that the linear regression model is a good fit for the data and satisfies the assumptions of linearity, nearly normal residuals, and constant variability. Therefore, the model can be used to make predictions about Order Value based on Meal Price with a reasonable degree of accuracy.

REFERENCES

a. Datacamp learnings
b. In-Class materials
c. The Constant Variance Assumption (https://www.statology.org/constant-variance-assumption)

Final Project

DAB501

24 April 2023

Student Information

Name (print): Tehsin Shaikh ID: 0831234

Signature: Shaikh

Academic Integrity

I, Tehsin Shaikh, hereby state that I have not communicated with or gained information in any way from any person or resource that would violate the College's academic integrity policies, and that all work is my own.

Student Information

Name (print): Gurjinderpal Singh ID: 0821129

Signature: Gurjinderpal Singh

Academic Integrity

I, Gurjinderpal Singh, hereby state that I have not communicated with or gained information in any way from any person or resource that would violate the College's academic integrity policies, and that all work is my own.

Student Information

Name (print): Aesha Rathod ID: 0813512

Signature: Aesha

Academic Integrity

I, Aesha Rathod, hereby state that I have not communicated with or gained information in any way from any person or resource that would violate the College's academic integrity policies, and that all work is my own.