Final Project – Seryl Einhorn

Part I

Tableau Visualisation

A graph and chart with numbers

Description automatically generated with medium confidence

[Public Tableau – Final](https://public.tableau.com/views/Final_17343466393700/Dashboard1?:language=en-US&:sid=&:redirect=auth&:display_count=n&:origin=viz_share_link)

After adding a column to compare data from one car company to another, I created an overview at the bottom of the dataset. This overview summarizes each company's statistics, including the count of cars, average acceleration, average displacement, average MPG, and average weight. One of the most striking differences between companies is the average weight. For instance, Volkswagen has the lightest average at 1,845 lbs, while Hi has the heaviest average at 4,732 lbs.

Next, a line chart illustrates the relationship between weight and the number of cylinders over the years. Initially, cars were significantly heavier, with an average of 7 cylinders. Over time, cars became progressively lighter, except for a brief period between 1978 and 1979 when they temporarily increased in weight before dropping to one of their lightest averages. The chart also shows a clear correlation: the more cylinders a car has, the heavier it tends to be.

Lastly, a bar chart compares the average horsepower and MPG for each company. The height of the bars represents horsepower, while the darkness of the bars indicates higher MPG. This visualization reveals an inverse relationship between horsepower and fuel efficiency: higher horsepower generally corresponds to lower MPG, while lower horsepower is associated with greater fuel economy.

**Simple Linear Regression**

> auto\_300 <- `auto.mpg(1)`[1:300, ] *# in order to use only the first 300 samples*

> sim\_lin\_model <- lm(mpg ~ weight, data = auto\_300) *# created a simple linear regression model using mpg and weight*

> summary(sim\_lin\_model) *# viewed the summary of the model*

A screenshot of a computer

Description automatically generated

1) R2 = 77.41% of the variability in mpg is explained by the variable weight

2) Adjusted R2 = 0.7733, slightly lower than R2. This is expected since there's only one predictor, and adjusted R2 penalizes for unnecessary predictors.

3) Linear Regression Equation = mpg = (40.3879−0.0062524)weight. The negative slope reflects an inverse relationship: as vehicle weight increases, fuel efficiency (measured in mpg) decreases.

**Multiple Linear Regression**

> auto\_300$horsepower <- as.numeric(as.character(auto\_300$horsepower)) *#changed the variable from a character to a numeric*

> multiple\_model <- lm(mpg ~ weight + horsepower + acceleration, data = auto\_300) *# created a simple linear regression model comparing mpg to weight, horsepower and acceleration.*

> summary(multiple\_model) *# viewed the summary of the model*

A screenshot of a computer

Description automatically generated

1) R2 = 78.21% of the variability in mpg is explained by the predictors in the model.

2) Adjusted R2 = 0.7799, slightly lower than R2. This reflects that adding the variable acceleration, which is not significant, slightly reduces the model's explanatory power.

3) Linear Regression Equation =   
mpg = (40.7751) – (0.0051403)weight – (0.0298549)horsepower – (0.0345989)acceleration. **Intercept (40.7751) =** This is the predicted mpg when all independent variables (weight, horsepower, and acceleration) are zero. While this scenario may not be realistic, it is a necessary part of the regression equation.  
**Weight (-0.0051403) =** For each one-unit increase in weight, mpg decreases by 0.0051403, assuming other variables remain constant. This variable is highly significant (p < 2e−16)  
**Horsepower (-0.0298549) =** For each one-unit increase in horsepower, mpg decreases by 0.0298549, assuming other variables remain constant. This variable is statistically significant (p=0.0106).  
**Acceleration (-0.0345989)=**For each one-unit increase in acceleration, mpg decreases by 0.0345989, assuming other variables remain constant. This variable is **not statistically significant** (p=0.7282), meaning it does not have a meaningful impact on mpg in the presence of the other variables.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model Type** | **Predictors** | **Multiple R-squared** | **Adjusted R-squared** | **Regression Equation** |
| **Simple Linear** | **weight** | **0.72** | **0.71** | **mpg = 40.3879 - 0.006 \* weight** |
| **Multiple Linear** | **weight, horsepower, acceleration** | **0.78** | **0.77** | **mpg = 40.7751 - 0.0051 \* weight - 0.0299 \* horsepower - 0.0346 \* acceleration** |

**Best Model**

> auto\_301 <- `auto.mpg(1)`[301:398, ] *# created a dataset with the alst 98 samples.*

> best\_model <- lm(mpg ~ weight + horsepower, data = auto\_300) *# created a mulitple linear model with out acceleration as it wasn’t impactful.*

> summary(best\_model) *# viewed the summary to confirm it's the best model*

**A screenshot of a computer

Description automatically generated**

> auto\_301$horsepower <- as.numeric(as.character(auto\_301$horsepower)) *#changed the variable from a character to a numeric*

> predicted\_mpg <- predict(best\_model, newdata = auto\_301) *# Predict mpg for test data*

> auto\_301$predicted\_mpg <- predicted\_mpg *# Add predictions and residuals to the test dataset*

> auto\_301$residuals <- auto\_301$mpg - auto\_301$predicted\_mpg

> plot(auto\_301$predicted\_mpg, auto\_301$residuals, *# Residual plot* main = "Residual Plot",  
 xlab = "Predicted MPG",  
 ylab = "Residuals",  
 pch = 19, col = "blue")  
>abline(h = 0, col = "red", lwd = 2) *# Add a horizontal line at y = 0*

**A graph with blue dots

Description automatically generated**

> hist(auto\_301$residuals, # Histogram of residuals  
 main = "Histogram of Residuals",  
 xlab = "Residuals",  
 col = "lightblue",  
 border = "black")

**A graph of a person with a bar graph

Description automatically generated with medium confidence**

Part II

**Question One  
What factors influence customer satisfaction?**

> clean\_data <- na.omit(Call\_Center) *# removed the NA’s in the data as many lines did ot have a CSAT*

> clean\_data$Call.Duration.In.Minutes <- as.numeric(clean\_data$Call.Duration.In.Minutes) *#changed to numeric*

> clean\_data$Csat.Score <- as.numeric(clean\_data$Csat.Score) *#changed to numeric*

> clean\_data$Response.Time <- factor(clean\_data$Response.Time,

+ levels = c("Below SLA", "Within SLA", "Above SLA"),

+ labels = c(-1, 0, 1)) *#mapped categories to numeric value*

> clean\_data$Response.Time <- as.numeric(as.character(clean\_data$Response.Time)) *# changed to numeric*

> cor(clean\_data$Response.Time, clean\_data$Csat.Score, use = "complete.obs") *# checked the correlation between Response Time and Csat* 

*#The correlation is approximately 0.0035, which is very close to zero. This suggests that there is virtually no linear relationship between Response Time and Csat Score in the dataset.*

> cor(clean\_data$Call.Duration.In.Minutes, clean\_data$Csat.Score, use = "complete.obs*") # checked the correlation between Call.Duration and Csat*



*#The correlation is approximately -0.00997, which is also very close to zero, indicating a very weak negative linear relationship between Call Duration and Csat Score.*

> anova\_channel <- aov(Csat.Score ~ Channel + Reason + Sentiment, data = clean\_data) *#anova for categorical variables*

> summary(anova\_channel)

A screenshot of a computer

Description automatically generated

*#Reason has a p-value much greater than 0.05 and therefore is non-significant. Channel and Sentiment have extremely small p-vlaues and are therefore significant predictors of Csat Score.*

> call\_model<- lm(Csat.Score ~ Channel + Sentiment, data = clean\_data) *#created a linear regression model using the significant predictors.*

> summary(call\_model)

A screenshot of a computer

Description automatically generated

*#The Adjusted R2 says that 80.64% of the variance in the Csat is explained by Channel and Sentiment, indicating that the model is well-fitted and the predictors are relevent.*

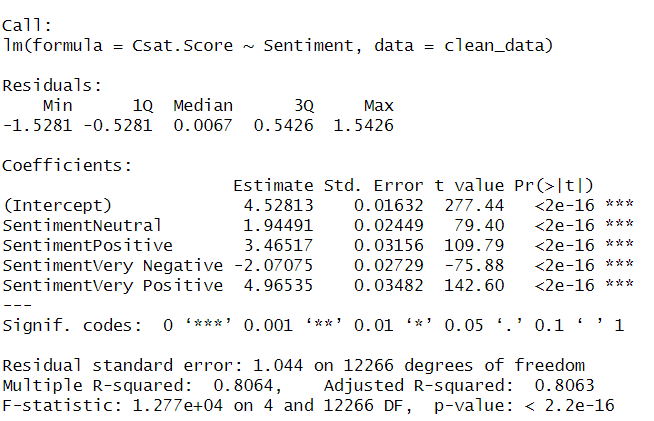
*# The intercept, 4.55, is the predicted Csat Score when both Channel and Sentiment are at the baseline levels.*

*#when channel is broken down to its 3 categories, they all have a p-value greater than 0.05, showing that actually they don’t have significant effect, after accounting for Sentiment.*

*#In Sentiment, Neutral, Positive, and Very Positive, all have positive coeficiants with very small p-values, inidcating a positive effect on Csat Score. While Very Negative has a negative coefficiant and a small p-value, indicating a negative effect on Csat Score.*

> best\_call\_model<- lm(Csat.Score ~ Sentiment, data = clean\_data) *#created a regression model with the updated list of significant predictors, just Sentiment.*

> summary(best\_call\_model)



**Conclusion: It seems that customer sentiment is what dirrectly effects Csat, so to rase the average Csat, you must imporve on customer sentiment.**

**Question Two  
What are the busiest times for the call center (by day and month)?**

> clean\_data <- Call\_Center *# reset to undo any changes previously made*

> head(clean\_data$Call.Timestamp) *#looked at TimeStamp to look at the data*



> clean\_data$Call.Timestamp <- as.Date(clean\_data$Call.Timestamp, format="%m/%d/%y") *#changed to date format, without time, as they were all 0:00.*

> clean\_data$Day <- weekdays(clean\_data$Call.Timestamp) *#extracted day of week and saved as variable*

> clean\_data$Month <- format(clean\_data$Call.Timestamp, "%B") *#extracted month and saved as variable*

> daily\_calls <- table(clean\_data$Day) *#count calls by day*

> daily\_calls *#print the table*

A black text on a white background

Description automatically generated

> barplot(daily\_calls, main="Calls by Day", xlab="Day of Week", ylab="Number of Calls", col="lightgreen") *#boxplot to visualize*

A graph of green rectangular objects

Description automatically generated with medium confidence

> tallest\_day <- which.max(daily\_calls) *#day with max calls which will be the tallest bar*

> shortest\_day <- which.min(daily\_calls) *#day with min calls which will be the shortest bar*

> barplot(daily\_calls, main="Calls by Day", xlab="Day of Week", ylab="Number of Calls", col=ifelse(1:length(daily\_calls) == tallest\_day, "blue", ifelse(1:length(daily\_calls) == shortest\_day, "red", "lightyellow")), ylim=c(0, max(daily\_calls) + 5))

A graph of blue and red rectangles with white text

Description automatically generated

> monthly\_calls <- table(clean\_data$Month) *#count calls by month*

> monthly\_calls *#print the table*

A close-up of a number

Description automatically generated

**Conclusion: Since we only have data for October, you can focus on the daily analysis to determine the busiest and least busy days. With Friday being the busiest and Sunday the least busy, you can make staffing decisions accordingly. For example, you might want to allocate more staff on Fridays to handle the higher call volume and reduce staff on Sundays when call volume is lower.**

**Question Three  
How does call duration vary by reason or channel?**

> clean\_data <- Call\_Center *# reset to undo any changes previously made*

> summary\_data <- aggregate(Call.Duration.In.Minutes ~ Channel, data = clean\_data, FUN = mean) *#get average call time per channel*

> print(summary\_data) *#print the table*

A screen shot of a computer

Description automatically generated

> barplot(summary\_data$Call.Duration.In.Minutes, names.arg = summary\_data$Channel, col = "skyblue", main = "Average Call Duration by Channel", xlab = "Channel", ylab = "Average Call Duration (minutes)")

A graph of a number of blue rectangular objects

Description automatically generated

**Conclusion: The average call duration is the same across all four channels (call, chatbot, email, and web). This suggests that the efficiency of customer service is similar regardless of the channel. However, this doesn't necessarily mean all channels are equally effective. For example, a chatbot might take the same amount of time as a call but could be less effective in resolving complex issues. Similarly, email and web interactions are asynchronous, which might affect customer expectations and satisfaction, even if the duration is similar. Therefore, effectiveness depends not just on time, but also on how well each channel meets customer needs and expectations.**