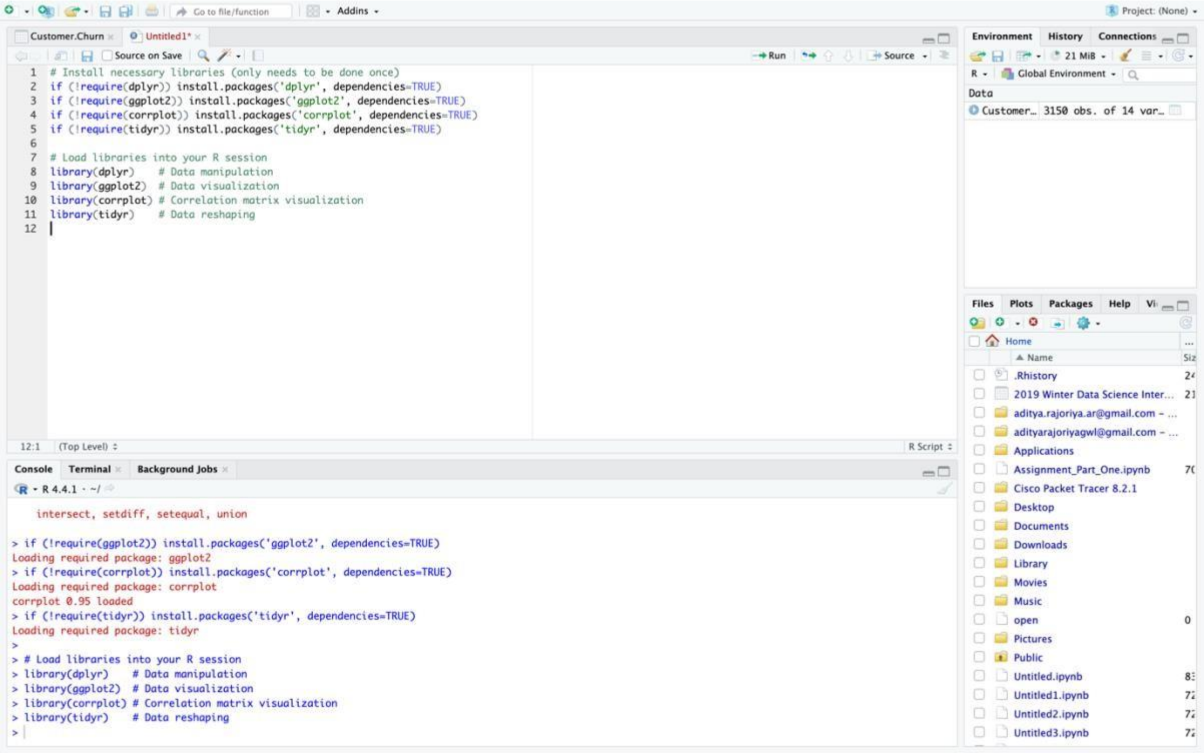
PROGRAMMING FUNDAMENTALS FOR ANALYTICS

R_Assignment

Name: Aditya Shroff (101539669)
Professor: Mohsen Selseleh

Aditya Shroff
101539669

Step 1: Load the dataset and necessary libraries

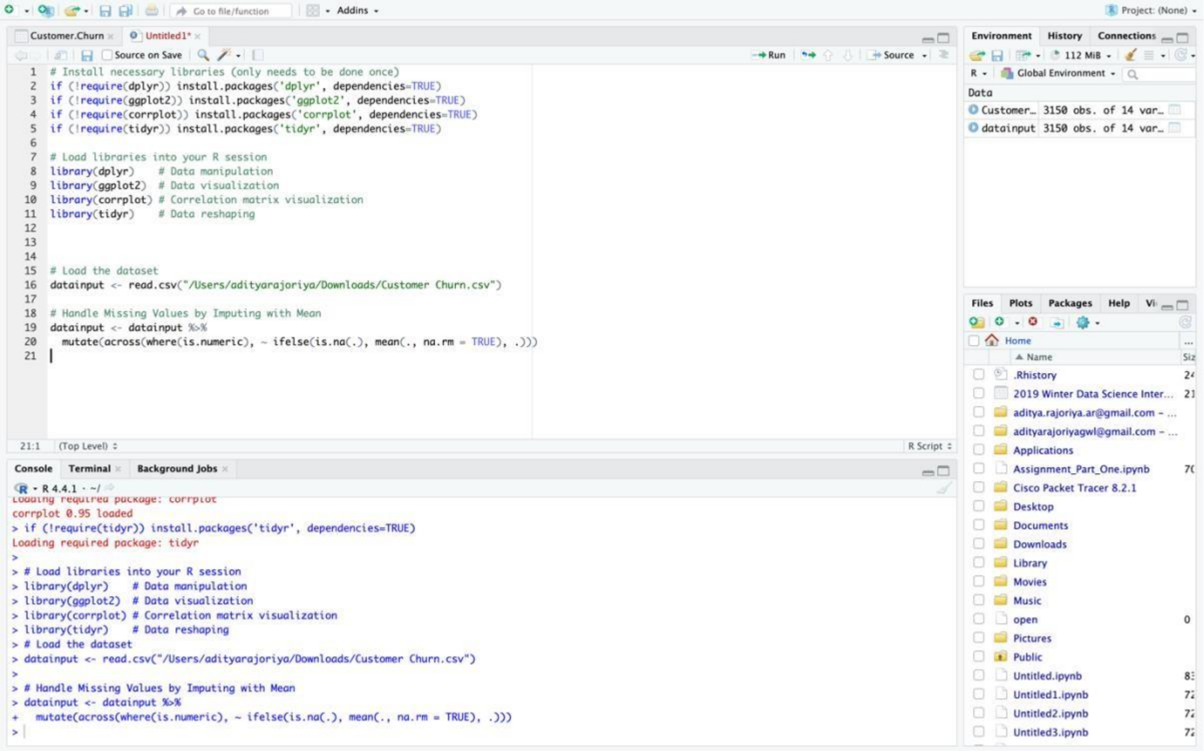


```r
# Install necessary libraries (only needs to be done once)
if (!require(dplyr)) install.packages('dplyr', dependencies=TRUE)
if (!require(ggplot2)) install.packages('ggplot2', dependencies=TRUE)
if (!require(corrplot)) install.packages('corrplot', dependencies=TRUE)
if (!require(tidyr)) install.packages('tidyr', dependencies=TRUE)

# Load libraries into your R session
library(dplyr)    # Data manipulation
library(ggplot2)  # Data visualization
library(corrplot) # Correlation matrix visualization
library(tidyr)    # Data reshaping
```

```
    intersect, setdiff, setequal, union

> if (!require(ggplot2)) install.packages('ggplot2', dependencies=TRUE)
Loading required package: ggplot2
> if (!require(corrplot)) install.packages('corrplot', dependencies=TRUE)
Loading required package: corrplot
corrplot 0.95 loaded
> if (!require(tidyr)) install.packages('tidyr', dependencies=TRUE)
Loading required package: tidyr
>
> # Load libraries into your R session
> library(dplyr)    # Data manipulation
> library(ggplot2)  # Data visualization
> library(corrplot) # Correlation matrix visualization
> library(tidyr)    # Data reshaping
>
```

Data
Customer… 3150 obs. of 14 var…

2. Load the Dataset and Handle Missing Values



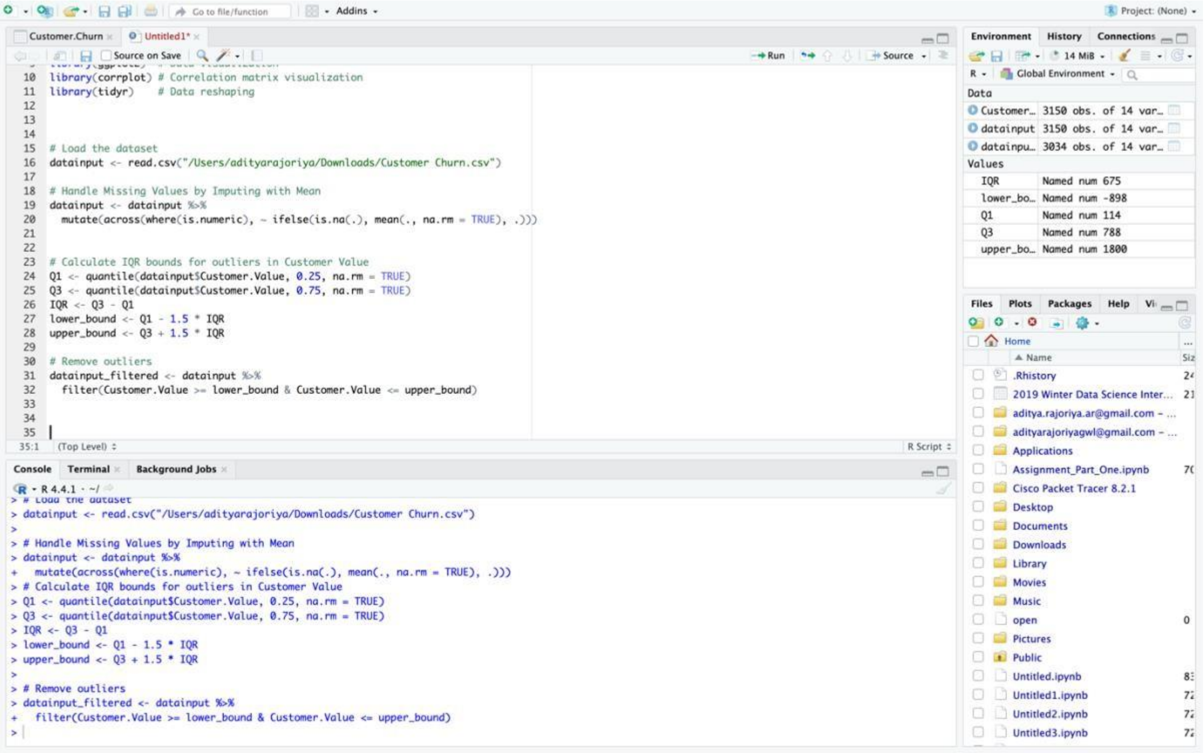
```r
# Install necessary libraries (only needs to be done once)
if (!require(dplyr)) install.packages('dplyr', dependencies=TRUE)
if (!require(ggplot2)) install.packages('ggplot2', dependencies=TRUE)
if (!require(corrplot)) install.packages('corrplot', dependencies=TRUE)
if (!require(tidyr)) install.packages('tidyr', dependencies=TRUE)

# Load libraries into your R session
library(dplyr)    # Data manipulation
library(ggplot2)  # Data visualization
library(corrplot) # Correlation matrix visualization
library(tidyr)    # Data reshaping



# Load the dataset
datainput <- read.csv("/Users/adityarajoriya/Downloads/Customer Churn.csv")

# Handle Missing Values by Imputing with Mean
datainput <- datainput %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))
```

```
Loading required package: corrplot
corrplot 0.95 loaded
> if (!require(tidyr)) install.packages('tidyr', dependencies=TRUE)
Loading required package: tidyr
>
> # Load libraries into your R session
> library(dplyr)    # Data manipulation
> library(ggplot2)  # Data visualization
> library(corrplot) # Correlation matrix visualization
> library(tidyr)    # Data reshaping
> # Load the dataset
> datainput <- read.csv("/Users/adityarajoriya/Downloads/Customer Churn.csv")
>
> # Handle Missing Values by Imputing with Mean
> datainput <- datainput %>%
+   mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))
>
```

Data
Customer… 3150 obs. of 14 var…
datainput 3150 obs. of 14 var…

Aditya Shroff
101539669

3. Calculate IQR and Remove Outliers

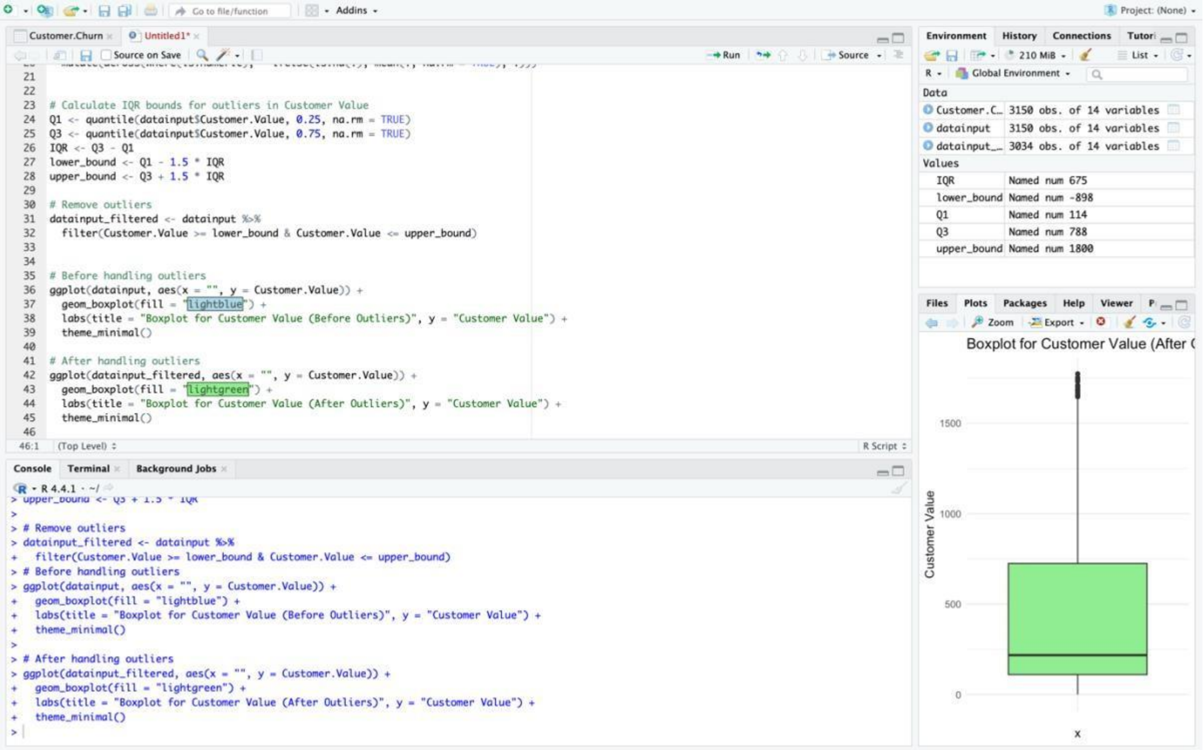


```r
library(corrplot) # Correlation matrix visualization
library(tidyr)    # Data reshaping



# Load the dataset
datainput <- read.csv("/Users/adityarajoriya/Downloads/Customer Churn.csv")

# Handle Missing Values by Imputing with Mean
datainput <- datainput %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))


# Calculate IQR bounds for outliers in Customer Value
Q1 <- quantile(datainput$Customer.Value, 0.25, na.rm = TRUE)
Q3 <- quantile(datainput$Customer.Value, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

# Remove outliers
datainput_filtered <- datainput %>%
  filter(Customer.Value >= lower_bound & Customer.Value <= upper_bound)
```

```
> # Load the dataset
> datainput <- read.csv("/Users/adityarajoriya/Downloads/Customer Churn.csv")
>
> # Handle Missing Values by Imputing with Mean
> datainput <- datainput %>%
+   mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))
> # Calculate IQR bounds for outliers in Customer Value
> Q1 <- quantile(datainput$Customer.Value, 0.25, na.rm = TRUE)
> Q3 <- quantile(datainput$Customer.Value, 0.75, na.rm = TRUE)
> IQR <- Q3 - Q1
> lower_bound <- Q1 - 1.5 * IQR
> upper_bound <- Q3 + 1.5 * IQR
>
> # Remove outliers
> datainput_filtered <- datainput %>%
+   filter(Customer.Value >= lower_bound & Customer.Value <= upper_bound)
> 
```

```
Data
Customer…  3150 obs. of 14 var…
datainput  3150 obs. of 14 var…
datainpu…  3034 obs. of 14 var…
Values
IQR        Named num 675
lower_bo…  Named num -898
Q1         Named num 114
Q3         Named num 788
upper_bo…  Named num 1800
```

4. Visualize Before and After Outlier Removal



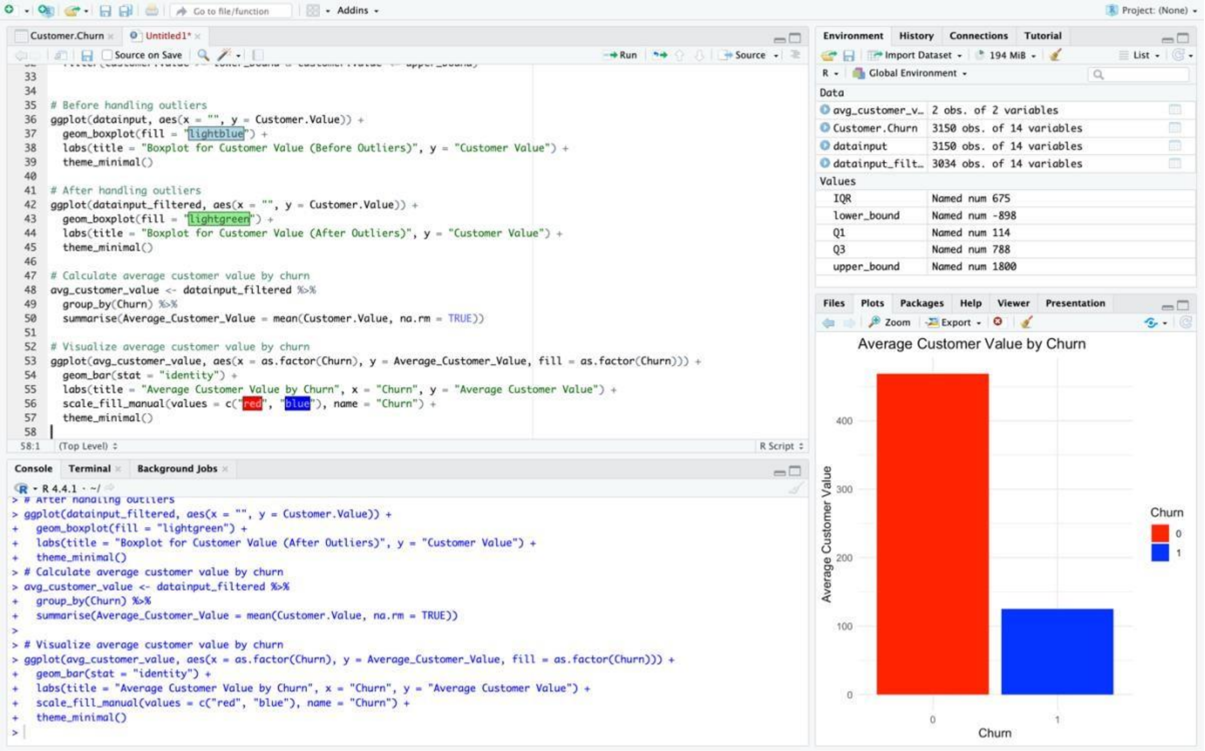
```r


# Calculate IQR bounds for outliers in Customer Value
Q1 <- quantile(datainput$Customer.Value, 0.25, na.rm = TRUE)
Q3 <- quantile(datainput$Customer.Value, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

# Remove outliers
datainput_filtered <- datainput %>%
  filter(Customer.Value >= lower_bound & Customer.Value <= upper_bound)


# Before handling outliers
ggplot(datainput, aes(x = "", y = Customer.Value)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Boxplot for Customer Value (Before Outliers)", y = "Customer Value") +
  theme_minimal()

# After handling outliers
ggplot(datainput_filtered, aes(x = "", y = Customer.Value)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = "Boxplot for Customer Value (After Outliers)", y = "Customer Value") +
  theme_minimal()

```

```
> upper_bound <- Q3 + 1.5 * IQR
>
> # Remove outliers
> datainput_filtered <- datainput %>%
+   filter(Customer.Value >= lower_bound & Customer.Value <= upper_bound)
> # Before handling outliers
> ggplot(datainput, aes(x = "", y = Customer.Value)) +
+   geom_boxplot(fill = "lightblue") +
+   labs(title = "Boxplot for Customer Value (Before Outliers)", y = "Customer Value") +
+   theme_minimal()
>
> # After handling outliers
> ggplot(datainput_filtered, aes(x = "", y = Customer.Value)) +
+   geom_boxplot(fill = "lightgreen") +
+   labs(title = "Boxplot for Customer Value (After Outliers)", y = "Customer Value") +
+   theme_minimal()
> 
```

```
Data
Customer.C…  3150 obs. of 14 variables
datainput    3150 obs. of 14 variables
datainput_…  3034 obs. of 14 variables
Values
IQR          Named num 675
lower_bound  Named num -898
Q1           Named num 114
Q3           Named num 788
upper_bound  Named num 1800
```

Aditya Shroff
101539669

5. Group Data and Visualize Average Customer Value by Churn

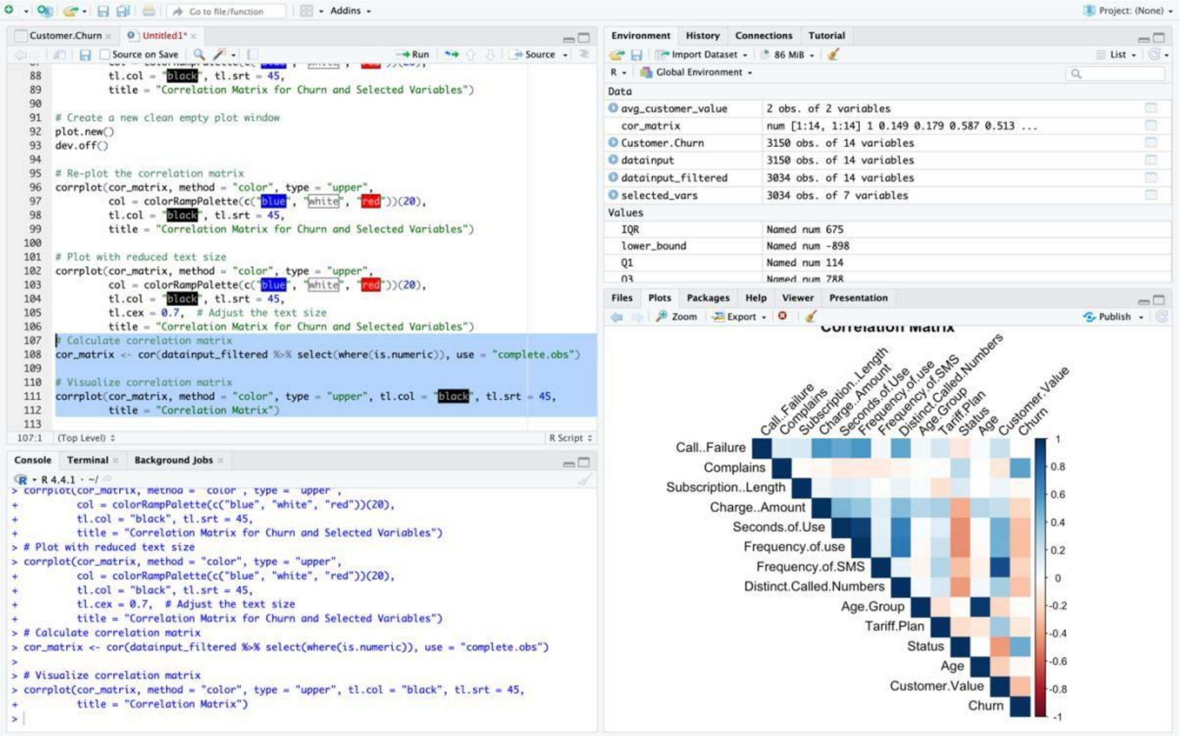


```r
# Before handling outliers
ggplot(datainput, aes(x = "", y = Customer.Value)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Boxplot for Customer Value (Before Outliers)", y = "Customer Value") +
  theme_minimal()

# After handling outliers
ggplot(datainput_filtered, aes(x = "", y = Customer.Value)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = "Boxplot for Customer Value (After Outliers)", y = "Customer Value") +
  theme_minimal()

# Calculate average customer value by churn
avg_customer_value <- datainput_filtered %>%
  group_by(Churn) %>%
  summarise(Average_Customer_Value = mean(Customer.Value, na.rm = TRUE))

# Visualize average customer value by churn
ggplot(avg_customer_value, aes(x = as.factor(Churn), y = Average_Customer_Value, fill = as.factor(Churn))) +
  geom_bar(stat = "identity") +
  labs(title = "Average Customer Value by Churn", x = "Churn", y = "Average Customer Value") +
  scale_fill_manual(values = c("red", "blue"), name = "Churn") +
  theme_minimal()
```

```
> # After handling outliers
> ggplot(datainput_filtered, aes(x = "", y = Customer.Value)) +
+   geom_boxplot(fill = "lightgreen") +
+   labs(title = "Boxplot for Customer Value (After Outliers)", y = "Customer Value") +
+   theme_minimal()
> # Calculate average customer value by churn
> avg_customer_value <- datainput_filtered %>%
+   group_by(Churn) %>%
+   summarise(Average_Customer_Value = mean(Customer.Value, na.rm = TRUE))
>
> # Visualize average customer value by churn
> ggplot(avg_customer_value, aes(x = as.factor(Churn), y = Average_Customer_Value, fill = as.factor(Churn))) +
+   geom_bar(stat = "identity") +
+   labs(title = "Average Customer Value by Churn", x = "Churn", y = "Average Customer Value") +
+   scale_fill_manual(values = c("red", "blue"), name = "Churn") +
+   theme_minimal()
>
```

```
Data
avg_customer_v…   2 obs. of 2 variables
Customer.Churn    3150 obs. of 14 variables
datainput         3150 obs. of 14 variables
datainput_filt…   3034 obs. of 14 variables
Values
IQR               Named num 675
lower_bound       Named num -898
Q1                Named num 114
Q3                Named num 788
upper_bound       Named num 1800
```

6. Create and Visualize Correlation Matrix



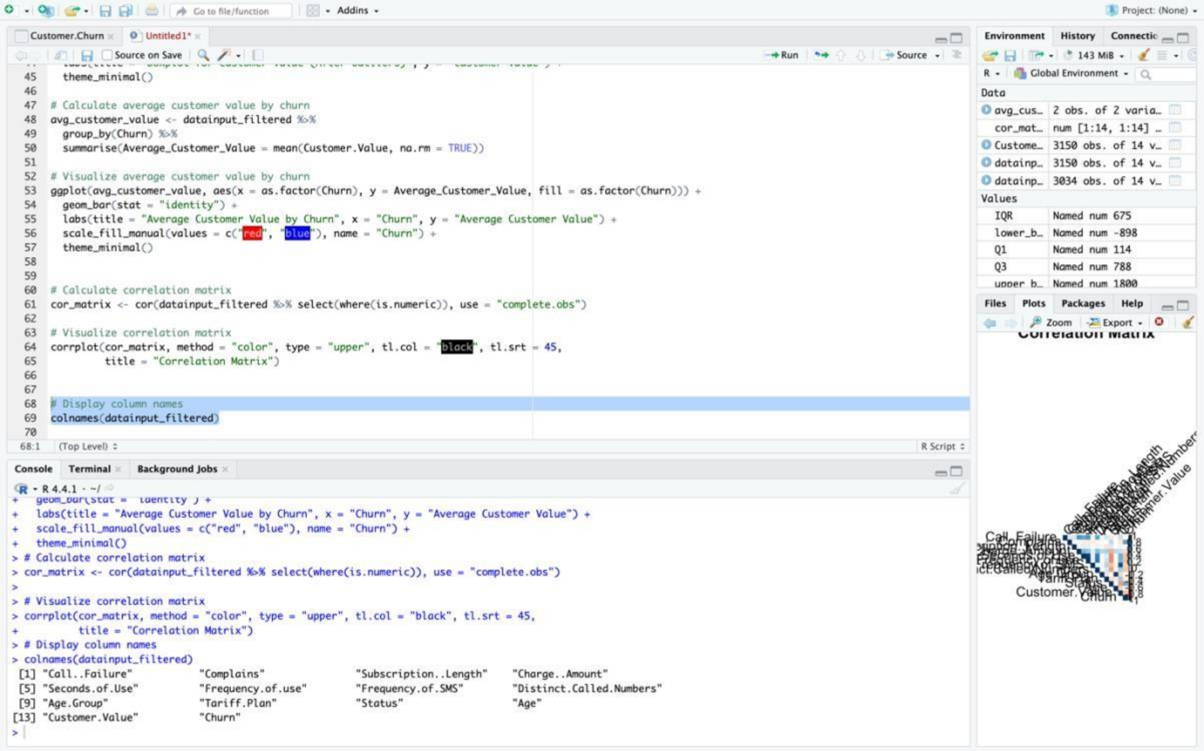
```r
          tl.col = "black", tl.srt = 45,
          title = "Correlation Matrix for Churn and Selected Variables")

# Create a new clean empty plot window
plot.new()
dev.off()

# Re-plot the correlation matrix
corrplot(cor_matrix, method = "color", type = "upper",
          col = colorRampPalette(c("blue", "white", "red"))(20),
          tl.col = "black", tl.srt = 45,
          title = "Correlation Matrix for Churn and Selected Variables")

# Plot with reduced text size
corrplot(cor_matrix, method = "color", type = "upper",
          col = colorRampPalette(c("blue", "white", "red"))(20),
          tl.col = "black", tl.srt = 45,
          tl.cex = 0.7,  # Adjust the text size
          title = "Correlation Matrix for Churn and Selected Variables")
# Calculate correlation matrix
cor_matrix <- cor(datainput_filtered %>% select(where(is.numeric)), use = "complete.obs")

# Visualize correlation matrix
corrplot(cor_matrix, method = "color", type = "upper", tl.col = "black", tl.srt = 45,
          title = "Correlation Matrix")
```

```
> corrplot(cor_matrix, method = "color", type = "upper",
+          col = colorRampPalette(c("blue", "white", "red"))(20),
+          tl.col = "black", tl.srt = 45,
+          title = "Correlation Matrix for Churn and Selected Variables")
> # Plot with reduced text size
> corrplot(cor_matrix, method = "color", type = "upper",
+          col = colorRampPalette(c("blue", "white", "red"))(20),
+          tl.col = "black", tl.srt = 45,
+          tl.cex = 0.7,  # Adjust the text size
+          title = "Correlation Matrix for Churn and Selected Variables")
> # Calculate correlation matrix
> cor_matrix <- cor(datainput_filtered %>% select(where(is.numeric)), use = "complete.obs")
>
> # Visualize correlation matrix
> corrplot(cor_matrix, method = "color", type = "upper", tl.col = "black", tl.srt = 45,
+          title = "Correlation Matrix")
>
```

```
Data
avg_customer_value   2 obs. of 2 variables
cor_matrix           num [1:14, 1:14] 1 0.149 0.179 0.587 0.513 ...
Customer.Churn       3150 obs. of 14 variables
datainput            3150 obs. of 14 variables
datainput_filtered   3034 obs. of 14 variables
selected_vars        3034 obs. of 7 variables
Values
IQR                  Named num 675
lower_bound          Named num -898
Q1                   Named num 114
Q3                   Named num 788
```

Aditya Shroff
101539669

7. Display Column Names

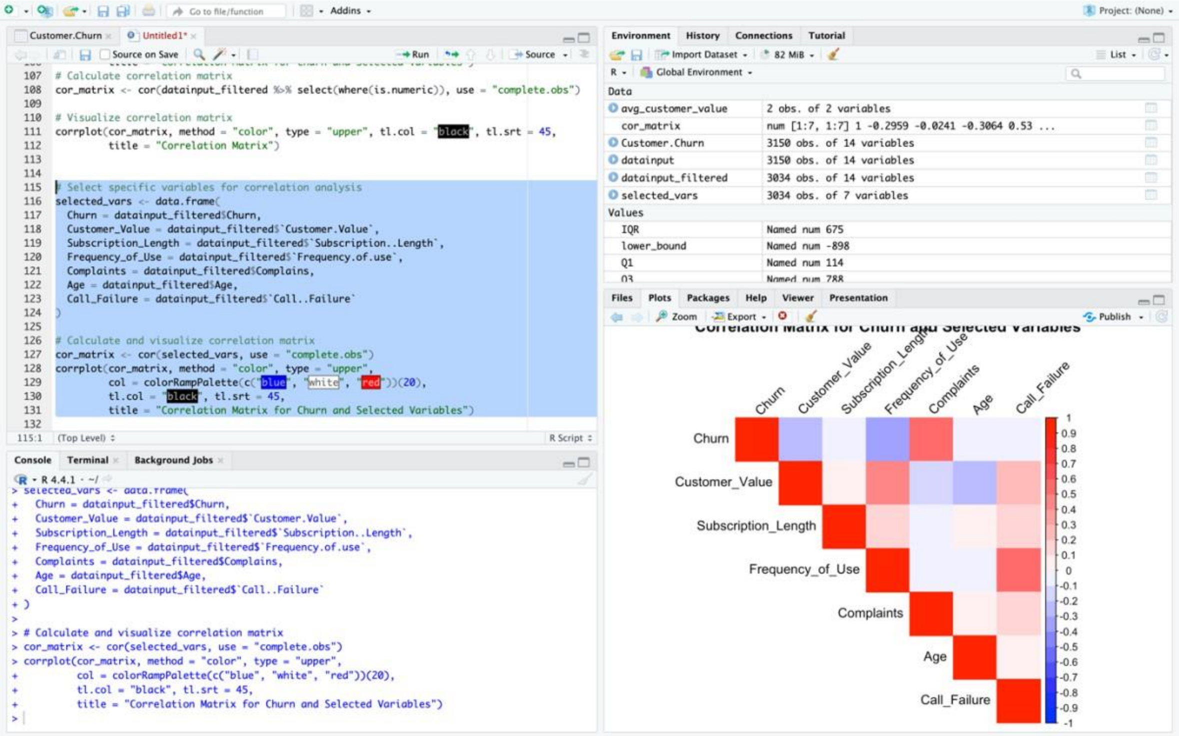


```r
  theme_minimal()

# Calculate average customer value by churn
avg_customer_value <- datainput_filtered %>%
  group_by(Churn) %>%
  summarise(Average_Customer_Value = mean(Customer.Value, na.rm = TRUE))

# Visualize average customer value by churn
ggplot(avg_customer_value, aes(x = as.factor(Churn), y = Average_Customer_Value, fill = as.factor(Churn))) +
  geom_bar(stat = "identity") +
  labs(title = "Average Customer Value by Churn", x = "Churn", y = "Average Customer Value") +
  scale_fill_manual(values = c("red", "blue"), name = "Churn") +
  theme_minimal()


# Calculate correlation matrix
cor_matrix <- cor(datainput_filtered %>% select(where(is.numeric)), use = "complete.obs")

# Visualize correlation matrix
corrplot(cor_matrix, method = "color", type = "upper", tl.col = "black", tl.srt = 45,
         title = "Correlation Matrix")


# Display column names
colnames(datainput_filtered)
```

```
+   geom_bar(stat = "identity") +
+   labs(title = "Average Customer Value by Churn", x = "Churn", y = "Average Customer Value") +
+   scale_fill_manual(values = c("red", "blue"), name = "Churn") +
+   theme_minimal()
> # Calculate correlation matrix
> cor_matrix <- cor(datainput_filtered %>% select(where(is.numeric)), use = "complete.obs")
>
> # Visualize correlation matrix
> corrplot(cor_matrix, method = "color", type = "upper", tl.col = "black", tl.srt = 45,
+          title = "Correlation Matrix")
> # Display column names
> colnames(datainput_filtered)
 [1] "Call..Failure"           "Complains"               "Subscription..Length"    "Charge..Amount"
 [5] "Seconds.of.Use"          "Frequency.of.use"        "Frequency.of.SMS"        "Distinct.Called.Numbers"
 [9] "Age.Group"               "Tariff.Plan"             "Status"                  "Age"
[13] "Customer.Value"          "Churn"
>
```

Data	
avg_cus…	2 obs. of 2 varia…
cor_mat…	num [1:14, 1:14] …
Custome…	3150 obs. of 14 v…
datainp…	3150 obs. of 14 v…
datainp…	3034 obs. of 14 v…
Values	
IQR	Named num 675
lower_b…	Named num -898
Q1	Named num 114
Q3	Named num 788
upper_b…	Named num 1800

8. Select Specific Variables and Create Another Correlation Matrix



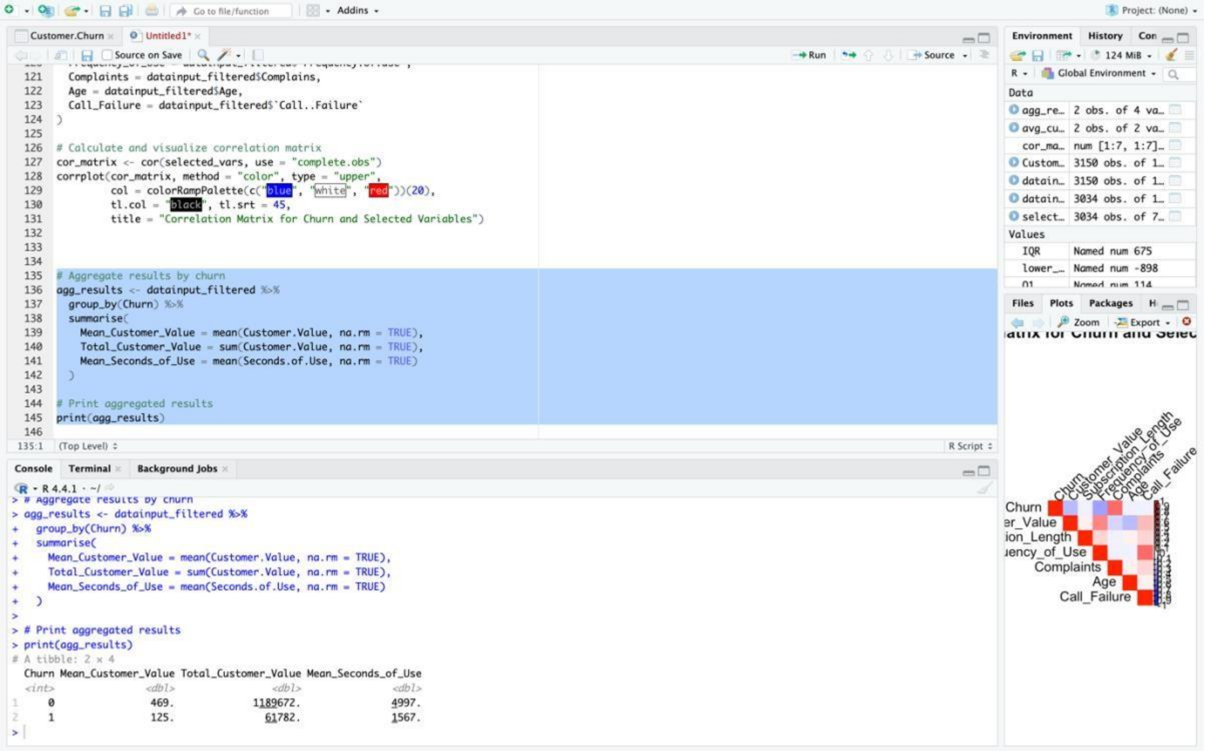
```r
# Calculate correlation matrix
cor_matrix <- cor(datainput_filtered %>% select(where(is.numeric)), use = "complete.obs")

# Visualize correlation matrix
corrplot(cor_matrix, method = "color", type = "upper", tl.col = "black", tl.srt = 45,
         title = "Correlation Matrix")


# Select specific variables for correlation analysis
selected_vars <- data.frame(
  Churn = datainput_filtered$Churn,
  Customer_Value = datainput_filtered$`Customer.Value`,
  Subscription_Length = datainput_filtered$`Subscription..Length`,
  Frequency_of_Use = datainput_filtered$`Frequency.of.use`,
  Complaints = datainput_filtered$Complains,
  Age = datainput_filtered$Age,
  Call_Failure = datainput_filtered$`Call..Failure`
)

# Calculate and visualize correlation matrix
cor_matrix <- cor(selected_vars, use = "complete.obs")
corrplot(cor_matrix, method = "color", type = "upper",
         col = colorRampPalette(c("blue", "white", "red"))(20),
         tl.col = "black", tl.srt = 45,
         title = "Correlation Matrix for Churn and Selected Variables")
```

```
> selected_vars <- data.frame(
+   Churn = datainput_filtered$Churn,
+   Customer_Value = datainput_filtered$`Customer.Value`,
+   Subscription_Length = datainput_filtered$`Subscription..Length`,
+   Frequency_of_Use = datainput_filtered$`Frequency.of.use`,
+   Complaints = datainput_filtered$Complains,
+   Age = datainput_filtered$Age,
+   Call_Failure = datainput_filtered$`Call..Failure`
+ )
>
> # Calculate and visualize correlation matrix
> cor_matrix <- cor(selected_vars, use = "complete.obs")
> corrplot(cor_matrix, method = "color", type = "upper",
+          col = colorRampPalette(c("blue", "white", "red"))(20),
+          tl.col = "black", tl.srt = 45,
+          title = "Correlation Matrix for Churn and Selected Variables")
>
```

Data	
avg_customer_value	2 obs. of 2 variables
cor_matrix	num [1:7, 1:7] 1 -0.2959 -0.0241 -0.3064 0.53 ...
Customer.Churn	3150 obs. of 14 variables
datainput	3150 obs. of 14 variables
datainput_filtered	3034 obs. of 14 variables
selected_vars	3034 obs. of 7 variables
Values	
IQR	Named num 675
lower_bound	Named num -898
Q1	Named num 114
Q3	Named num 788

Aditya Shroff
101539669

9. Aggregate Data and Calculate Summary Statistics

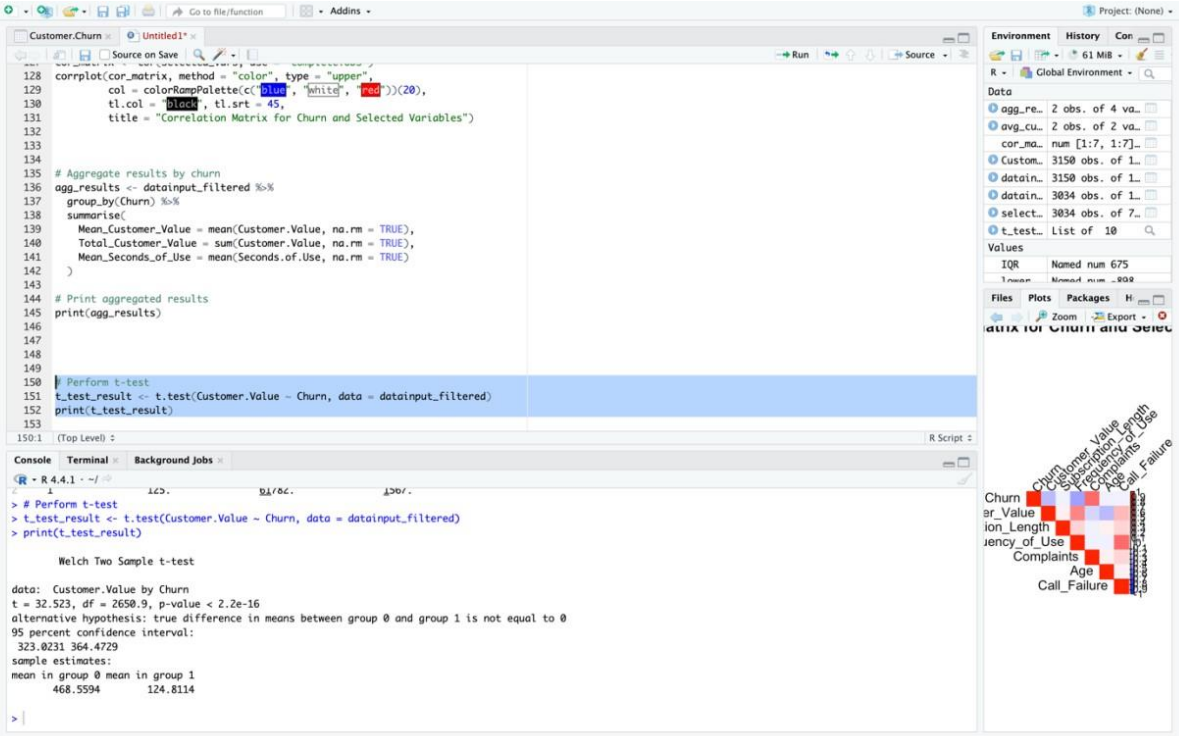


```
  Complaints = datainput_filtered$Complains,
  Age = datainput_filtered$Age,
  Call_Failure = datainput_filtered$`Call..Failure`
)

# Calculate and visualize correlation matrix
cor_matrix <- cor(selected_vars, use = "complete.obs")
corrplot(cor_matrix, method = "color", type = "upper",
         col = colorRampPalette(c("blue", "white", "red"))(20),
         tl.col = "black", tl.srt = 45,
         title = "Correlation Matrix for Churn and Selected Variables")



# Aggregate results by churn
agg_results <- datainput_filtered %>%
  group_by(Churn) %>%
  summarise(
    Mean_Customer_Value = mean(Customer.Value, na.rm = TRUE),
    Total_Customer_Value = sum(Customer.Value, na.rm = TRUE),
    Mean_Seconds_of_Use = mean(Seconds.of.Use, na.rm = TRUE)
  )

# Print aggregated results
print(agg_results)
```

```
> # Aggregate results by churn
> agg_results <- datainput_filtered %>%
+   group_by(Churn) %>%
+   summarise(
+     Mean_Customer_Value = mean(Customer.Value, na.rm = TRUE),
+     Total_Customer_Value = sum(Customer.Value, na.rm = TRUE),
+     Mean_Seconds_of_Use = mean(Seconds.of.Use, na.rm = TRUE)
+   )
>
> # Print aggregated results
> print(agg_results)
# A tibble: 2 × 4
  Churn Mean_Customer_Value Total_Customer_Value Mean_Seconds_of_Use
  <int>               <dbl>                <dbl>               <dbl>
1     0                469.             1189672.               4997.
2     1                125.               61782.               1567.
>
```

```
Data
agg_re…   2 obs. of 4 va…
avg_cu…   2 obs. of 2 va…
cor_ma…   num [1:7, 1:7]…
Custom…   3150 obs. of 1…
datain…   3150 obs. of 1…
datain…   3034 obs. of 1…
select…   3034 obs. of 7…
Values
IQR       Named num 675
lower_…   Named num -898
Q1        Named num 114
```

10. Perform T-Test



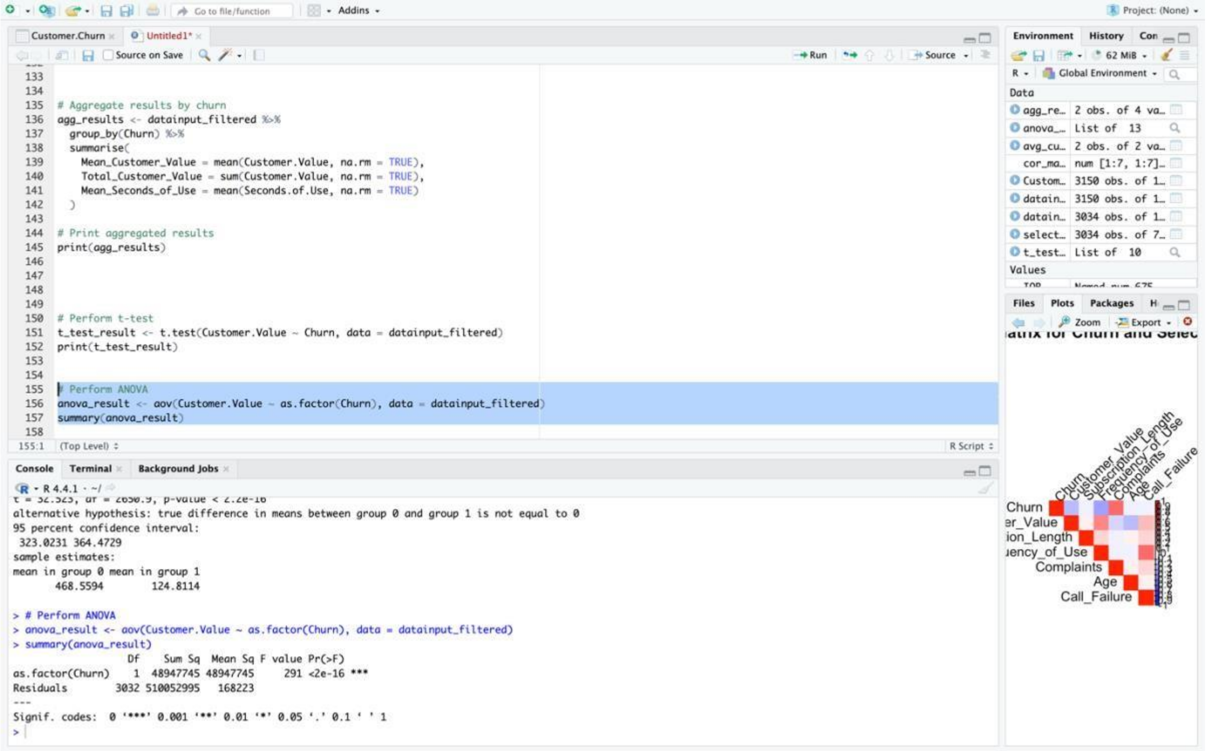
```
corrplot(cor_matrix, method = "color", type = "upper",
         col = colorRampPalette(c("blue", "white", "red"))(20),
         tl.col = "black", tl.srt = 45,
         title = "Correlation Matrix for Churn and Selected Variables")



# Aggregate results by churn
agg_results <- datainput_filtered %>%
  group_by(Churn) %>%
  summarise(
    Mean_Customer_Value = mean(Customer.Value, na.rm = TRUE),
    Total_Customer_Value = sum(Customer.Value, na.rm = TRUE),
    Mean_Seconds_of_Use = mean(Seconds.of.Use, na.rm = TRUE)
  )

# Print aggregated results
print(agg_results)




# Perform t-test
t_test_result <- t.test(Customer.Value ~ Churn, data = datainput_filtered)
print(t_test_result)
```

```
> # Perform t-test
> t_test_result <- t.test(Customer.Value ~ Churn, data = datainput_filtered)
> print(t_test_result)

	Welch Two Sample t-test

data:  Customer.Value by Churn
t = 32.523, df = 2650.9, p-value < 2.2e-16
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
95 percent confidence interval:
 323.0231 364.4729
sample estimates:
mean in group 0 mean in group 1 
       468.5594        124.8114 

>
```

```
Data
agg_re…   2 obs. of 4 va…
avg_cu…   2 obs. of 2 va…
cor_ma…   num [1:7, 1:7]…
Custom…   3150 obs. of 1…
datain…   3150 obs. of 1…
datain…   3034 obs. of 1…
select…   3034 obs. of 7…
t_test…   List of  10
Values
IQR       Named num 675
lower_…   Named num -898
```

Aditya Shroff
101539669

11. Perform ANOVA

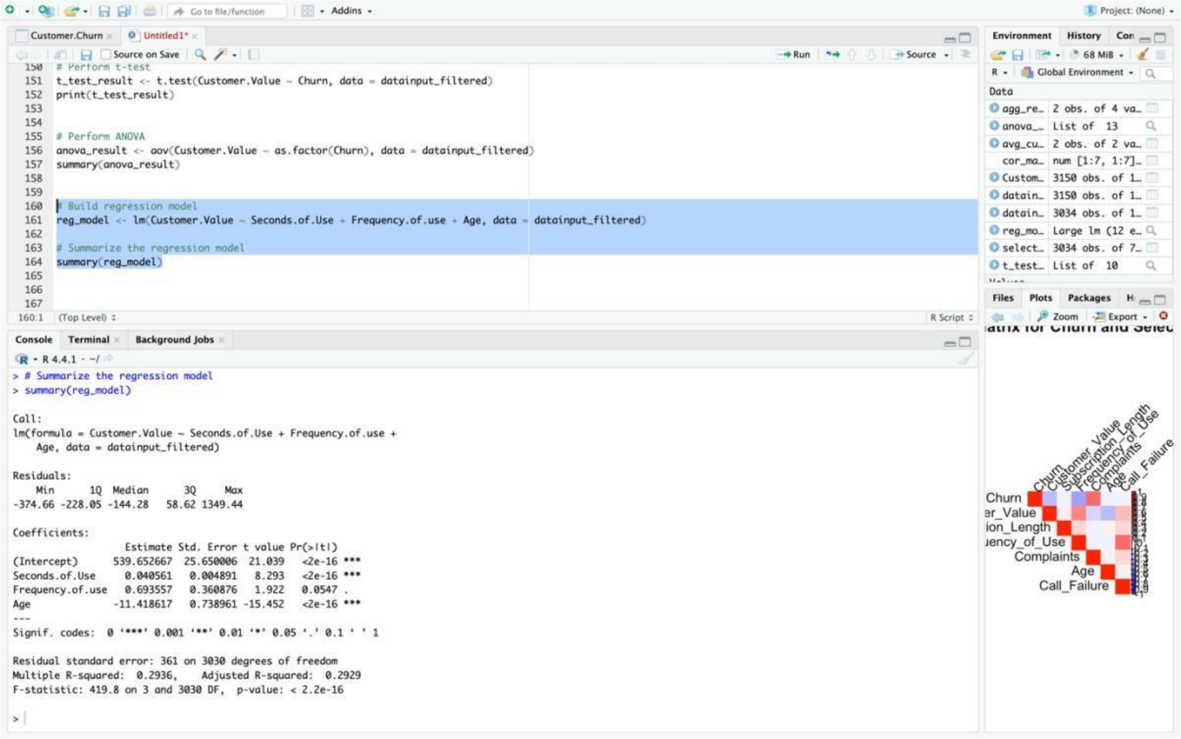


```
# Aggregate results by churn
agg_results <- datainput_filtered %>%
  group_by(Churn) %>%
  summarise(
    Mean_Customer_Value = mean(Customer.Value, na.rm = TRUE),
    Total_Customer_Value = sum(Customer.Value, na.rm = TRUE),
    Mean_Seconds_of_Use = mean(Seconds.of.Use, na.rm = TRUE)
  )

# Print aggregated results
print(agg_results)




# Perform t-test
t_test_result <- t.test(Customer.Value ~ Churn, data = datainput_filtered)
print(t_test_result)


# Perform ANOVA
anova_result <- aov(Customer.Value ~ as.factor(Churn), data = datainput_filtered)
summary(anova_result)
```

```
Data
agg_re…   2 obs. of 4 va…
anova_…   List of 13
avg_cu…   2 obs. of 2 va…
cor_ma…   num [1:7, 1:7]…
Custom…   3150 obs. of 1…
datain…   3150 obs. of 1…
datain…   3034 obs. of 1…
select…   3034 obs. of 7…
t_test…   List of 10
Values
```

```
t = 32.523, df = 2650.9, p-value < 2.2e-16
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
95 percent confidence interval:
 323.0231 364.4729
sample estimates:
mean in group 0 mean in group 1
       468.5594        124.8114

> # Perform ANOVA
> anova_result <- aov(Customer.Value ~ as.factor(Churn), data = datainput_filtered)
> summary(anova_result)
                   Df    Sum Sq  Mean Sq F value Pr(>F)
as.factor(Churn)    1  48947745 48947745     291 <2e-16 ***
Residuals        3032 510052995   168223
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

12. Build and Summarize Regression Model



```
# Perform t-test
t_test_result <- t.test(Customer.Value ~ Churn, data = datainput_filtered)
print(t_test_result)


# Perform ANOVA
anova_result <- aov(Customer.Value ~ as.factor(Churn), data = datainput_filtered)
summary(anova_result)


# Build regression model
reg_model <- lm(Customer.Value ~ Seconds.of.Use + Frequency.of.use + Age, data = datainput_filtered)

# Summarize the regression model
summary(reg_model)
```

```
Data
agg_re…   2 obs. of 4 va…
anova_…   List of 13
avg_cu…   2 obs. of 2 va…
cor_ma…   num [1:7, 1:7]…
Custom…   3150 obs. of 1…
datain…   3150 obs. of 1…
datain…   3034 obs. of 1…
reg_mo…   Large lm (12 e…
select…   3034 obs. of 7…
t_test…   List of 10
```

```
> # Summarize the regression model
> summary(reg_model)

Call:
lm(formula = Customer.Value ~ Seconds.of.Use + Frequency.of.use + 
    Age, data = datainput_filtered)

Residuals:
    Min      1Q  Median      3Q     Max 
-374.66 -228.05 -144.28   58.62 1349.44 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)      539.652667  25.650006  21.039   <2e-16 ***
Seconds.of.Use     0.040561   0.004891   8.293   <2e-16 ***
Frequency.of.use   0.693557   0.360876   1.922   0.0547 .  
Age              -11.418617   0.738961 -15.452   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 361 on 3030 degrees of freedom
Multiple R-squared:  0.2936,	Adjusted R-squared:  0.2929 
F-statistic: 419.8 on 3 and 3030 DF,  p-value: < 2.2e-16

>
```