DSC-680-Z1 Research Practicum Exploratory Data Analysis

Project Description

The research practicum involves on-site experiential learning in a research setting. This setting may be in the private or public sector, it may include such locations as education, governmental, non-governmental, or general research organization. The experience must provide students the opportunity to collect and analyze data, consider ethical implications of research, and draw empirically grounded conclusions.

Purpose: Carry out exploratory data analysis on a set of random sample data extracted for machine learning.
University Name: Utica College
Course Name: DSC-680-Z1 Research Practicum
Student Name: Henry J. Hu
Program Director Name: Dr. McCarthy, Michael
Runtime Environment: RStudio
Programming Language: R
Original Data Frame: 12,705,553 international wires belonging to 139 customers from 3 continents for the entire year of 2020.
Last Update: July 21st, 2021

Clearing R Studio Memory Usage

```
gc()
```

```
##           used (Mb) gc trigger (Mb) max used (Mb)
## Ncells  536234 28.7    1221431 65.3   621331 33.2
## Vcells 1009231  7.7    8388608 64.0  1601285 12.3
```

```
rm(list = ls())
```

Time Counter Start

```
start_time <- Sys.time()
```

Include the knitr package for integration of R code into Markdown

```
knitr::opts_chunk$set(echo = TRUE)
```

All the libraries used in this code

```
library(readr)
library(easypackages)
libraries("caret","caretEnsemble","caTools","class","cluster","data.tree","de
vtools","doSNOW","dplyr","e1071","factoextra","gbm","FNN","FSelector","ggalt"
,"ggforce","ggfortify","ggplot2","gmodels","klaR","lattice","mlbench","modees
t","nnet","neuralnet","outliers","parallel","psych","purrr","readr","rpart","
rpart.plot","spatialEco","stats","tidyr","randomForest","ROSE","rsample","ROC
R","pROC","glmnet","gridExtra","R6","Epi")
```

Import data into RStudio

```
# input_data <- read_delim("Final_cleaned_data.txt", ",", escape_double = FAL
SE, col_types = cols(
#                 TRANSACTION_ID = col_character(),
#                 TRANSACTION_TIME = col_datetime(),
#                 TRXN_MONTH = col_character(),
#                 CLIENT_ID = col_character(),
#                 COUNTRY_NAME = col_character(),
#                 COUNTRY_CODE = col_character(),
#                 CONTINENT_NAME = col_character(),
#                 CONTINENT_CODE = col_character(),
#                 SWIFT_MSG_TYPE = col_character(),
#                 AVG_TRXN_AMT = col_double(),
#                 TRANSACTION_AMOUNT = col_double()
#                 ),
#     trim_ws = TRUE)
```

Data Sampling

```
# Set random seed
set.seed(42)

# Sample the data
# input_data_4M <- input_data[sample(nrow(input_data), 4000000), ]

# Write data to storage
# write.table(input_data_4M, file="sample_df_4M.txt", append = FALSE, sep = "
,", dec = ".", row.names = FALSE, col.names = TRUE)
# write.csv(input_data_4M,"sample_df_4M.txt", row.names = FALSE)

# Load data into data frame
input_data_eda <- read_delim("sample_df_4M.txt", ",", escape_double = FALSE,
col_types = cols(
                 TRANSACTION_ID = col_character(),
                 TRANSACTION_TIME = col_datetime(),
                 TRXN_MONTH = col_integer(),
                 CLIENT_ID = col_character(),
                 COUNTRY_NAME = col_character(),
                 COUNTRY_CODE = col_character(),
                 CONTINENT_NAME = col_character(),
                 CONTINENT_CODE = col_character(),
                 SWIFT_MSG_TYPE = col_character(),
                 AVG_TRXN_AMT = col_double(),
                 TRANSACTION_AMOUNT = col_double()
               ),
    trim_ws = TRUE)
```

Numeric/character field separator

```
num.names <- input_data_eda %>% select_if(is.numeric) %>% colnames()
ch.names <- input_data_eda %>% select_if(is.character) %>% colnames()
```

Descriptive Statistics

These descriptive statistics reveal both the central tendency and dispersion tendency of the sample data for machine learning.

Dimension of data frame

```
dim(input_data_eda)
```

```
## [1] 4000000      11
```

Structure of data frame

```
str(input_data_eda)
```

```
## tibble [4,000,000 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ TRANSACTION_ID    : chr [1:4000000] "3174204" "1237511" "5556094" "2332
371" ...
##  $ TRANSACTION_TIME  : POSIXct[1:4000000], format: "2020-03-31 18:21:17" "
2020-02-07 00:24:34" ...
##  $ TRXN_MONTH        : int [1:4000000] 3 2 6 3 7 12 1 8 9 10 ...
##  $ CLIENT_ID         : chr [1:4000000] "7116490843" "6249255174" "71173963
44" "6249399616" ...
##  $ COUNTRY_NAME      : chr [1:4000000] "United States of America" "India-R
epublic of" "Switzerland-Swiss Confederation" "United States of America" ...
##  $ COUNTRY_CODE      : chr [1:4000000] "US" "IN" "CH" "US" ...
##  $ CONTINENT_NAME    : chr [1:4000000] "North America" "Asia" "Europe" "No
rth America" ...
##  $ CONTINENT_CODE    : chr [1:4000000] "NN" "AS" "EU" "NN" ...
##  $ SWIFT_MSG_TYPE    : chr [1:4000000] "202" "202" "103" "202" ...
##  $ AVG_TRXN_AMT      : num [1:4000000] 39246 26153 124854 39246 29569 ...
##  $ TRANSACTION_AMOUNT: num [1:4000000] 6475 3335 8920000 1784 2446 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   TRANSACTION_ID = col_character(),
##   ..   TRANSACTION_TIME = col_datetime(format = ""),
##   ..   TRXN_MONTH = col_integer(),
##   ..   CLIENT_ID = col_character(),
##   ..   COUNTRY_NAME = col_character(),
##   ..   COUNTRY_CODE = col_character(),
##   ..   CONTINENT_NAME = col_character(),
##   ..   CONTINENT_CODE = col_character(),
##   ..   SWIFT_MSG_TYPE = col_character(),
##   ..   AVG_TRXN_AMT = col_double(),
##   ..   TRANSACTION_AMOUNT = col_double()
##   .. )
```

Summary statistics of data frame

```
summary(input_data_eda)
```

```
##  TRANSACTION_ID     TRANSACTION_TIME                TRXN_MONTH
##  Length:4000000     Min.   :2020-01-01 00:01:48   Min.   : 1.000
##  Class :character   1st Qu.:2020-03-31 19:41:40   1st Qu.: 3.000
##  Mode  :character   Median :2020-07-03 17:02:13   Median : 7.000
##                     Mean   :2020-07-05 11:03:36   Mean   : 6.586
##                     3rd Qu.:2020-10-06 00:01:26   3rd Qu.:10.000
##                     Max.   :2020-12-31 21:58:20   Max.   :12.000
##   CLIENT_ID         COUNTRY_NAME       COUNTRY_CODE       CONTINENT_NAME
##  Length:4000000     Length:4000000     Length:4000000     Length:4000000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  CONTINENT_CODE     SWIFT_MSG_TYPE      AVG_TRXN_AMT    TRANSACTION_AMOUNT
##  Length:4000000     Length:4000000     Min.   : 20551   Min.   :0.000e+00
##  Class :character   Class :character   1st Qu.: 26405   1st Qu.:5.438e+03
##  Mode  :character   Mode  :character   Median : 30873   Median :3.547e+04
##                                        Mean   : 65288   Mean   :1.156e+07
##                                        3rd Qu.:105235   3rd Qu.:3.681e+05
##                                        Max.   :260831   Max.   :1.777e+10
```

Mode of each variable

```
lapply(input_data_eda[,num.names],mfv)
```

```
## $TRXN_MONTH
## [1] 12
##
## $AVG_TRXN_AMT
## [1] 30873.12
##
## $TRANSACTION_AMOUNT
## [1] 1784000
```

Variance of each variable

```
lapply(input_data_eda[,num.names],var)
```

```
## $TRXN_MONTH
## [1] 12.16137
##
## $AVG_TRXN_AMT
## [1] 2778714446
##
## $TRANSACTION_AMOUNT
## [1] 1.90406e+16
```

Standard deviation of each variable

```
lapply(input_data_eda[,num.names],sd)
```

```
## $TRXN_MONTH
## [1] 3.487315
##
## $AVG_TRXN_AMT
## [1] 52713.51
##
## $TRANSACTION_AMOUNT
## [1] 137987698
```

Glimpse of data frame

```
glimpse(input_data_eda)
```

```
## Rows: 4,000,000
## Columns: 11
## $ TRANSACTION_ID     <chr> "3174204", "1237511", "5556094", "2332371", "72
9...
## $ TRANSACTION_TIME   <dttm> 2020-03-31 18:21:17, 2020-02-07 00:24:34, 2020
-...
## $ TRXN_MONTH         <int> 3, 2, 6, 3, 7, 12, 1, 8, 9, 10, 3, 6, 8, 12, 8,
...
## $ CLIENT_ID          <chr> "7116490843", "6249255174", "7117396344", "6249
3...
## $ COUNTRY_NAME       <chr> "United States of America", "India-Republic of"
,...
## $ COUNTRY_CODE       <chr> "US", "IN", "CH", "US", "US", "US", "RU", "US",
...
## $ CONTINENT_NAME     <chr> "North America", "Asia", "Europe", "North Ameri
c...
## $ CONTINENT_CODE     <chr> "NN", "AS", "EU", "NN", "NN", "NN", "EU", "NN",
...
## $ SWIFT_MSG_TYPE     <chr> "202", "202", "103", "202", "202", "103", "202"
,...
## $ AVG_TRXN_AMT       <dbl> 39246.11, 26152.55, 124854.38, 39246.11, 29569.
2...
## $ TRANSACTION_AMOUNT <dbl> 6475.35, 3335.49, 8920000.00, 1784.00, 2446.45,
...
```

Head of data frame

```
head(input_data_eda)
```

```
## # A tibble: 6 x 11
##   TRANSACTION_ID TRANSACTION_TIME    TRXN_MONTH CLIENT_ID COUNTRY_NAME
##   <chr>          <dttm>                   <int> <chr>     <chr>
## 1 3174204        2020-03-31 18:21:17          3 71164908~ United Stat~
## 2 1237511        2020-02-07 00:24:34          2 62492551~ India-Repub~
## 3 5556094        2020-06-11 13:46:22          6 71173963~ Switzerland~
## 4 2332371        2020-03-10 05:15:07          3 62493996~ United Stat~
## 5 7295929        2020-07-31 17:23:36          7 71164908~ United Stat~
## 6 11840677       2020-12-09 13:52:20         12 71164858~ United Stat~
## # ... with 6 more variables: COUNTRY_CODE <chr>, CONTINENT_NAME <chr>,
## #   CONTINENT_CODE <chr>, SWIFT_MSG_TYPE <chr>, AVG_TRXN_AMT <dbl>,
## #   TRANSACTION_AMOUNT <dbl>
```

Tail of data frame

```
tail(input_data_eda)
```

```
## # A tibble: 6 x 11
##   TRANSACTION_ID TRANSACTION_TIME    TRXN_MONTH CLIENT_ID COUNTRY_NAME
##   <chr>          <dttm>                   <int> <chr>     <chr>
## 1 10019468       2020-10-20 14:40:45         10 71164858~ United Stat~
## 2 2156063        2020-03-04 10:30:43          3 71163786~ Turkey-Repu~
## 3 6491516        2020-07-08 14:32:18          7 71164858~ United Stat~
## 4 1429804        2020-02-13 10:06:57          2 71162836~ Cayman Isla~
## 5 11226381       2020-11-24 06:54:13         11 62493552~ Hong Kong-S~
## 6 4252038        2020-05-01 19:40:36          5 71164908~ United Stat~
## # ... with 6 more variables: COUNTRY_CODE <chr>, CONTINENT_NAME <chr>,
## #   CONTINENT_CODE <chr>, SWIFT_MSG_TYPE <chr>, AVG_TRXN_AMT <dbl>,
## #   TRANSACTION_AMOUNT <dbl>
```

Correlation plot

```
oldw <- getOption("warn")
options(warn = -1)
pairs.panels(input_data_eda[,num.names],gap=0,bg=c("green","red","yellow","blue","pink","purple"),pch=21)
```


```
options(warn = oldw)
```

Segregate and prepare data for bar plots

```
input_data_eda$AVG_TRXN_AMT=input_data_eda$AVG_TRXN_AMT/1000

input_data_eda <- input_data_eda %>%
  mutate(
          MONTH_TEXT =
           case_when(
                      TRXN_MONTH == 1 ~ "Jan",
                      TRXN_MONTH == 2 ~ "Feb",
                      TRXN_MONTH == 3 ~ "Mar",
                      TRXN_MONTH == 4 ~ "Apr",
                      TRXN_MONTH == 5 ~ "May",
                      TRXN_MONTH == 6 ~ "Jun",
                      TRXN_MONTH == 7 ~ "Jul",
                      TRXN_MONTH == 8 ~ "Aug",
                      TRXN_MONTH == 9 ~ "Sep",
                      TRXN_MONTH == 10 ~ "Oct",
                      TRXN_MONTH == 11 ~ "Nov",
                      TRXN_MONTH == 12 ~ "Dec"
                    )
         )

NN_103_df <- input_data_eda[input_data_eda$CONTINENT_CODE   =='NN' & input_da
ta_eda$SWIFT_MSG_TYPE=='103',]
NN_103_df <- NN_103_df[,c(3,10,12)]
NN_103_df = NN_103_df %>% distinct()
NN_103_df <- NN_103_df[order(NN_103_df$TRXN_MONTH),]
glimpse(NN_103_df)

## Rows: 12
## Columns: 3
## $ TRXN_MONTH   <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
## $ AVG_TRXN_AMT <dbl> 127.9748, 137.2108, 260.8305, 156.2500, 131.9741, 130
....
## $ MONTH_TEXT   <chr> "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug
"...

dim(NN_103_df)

## [1] 12  3

NN_202_df <- input_data_eda[input_data_eda$CONTINENT_CODE   =='NN' & input_da
ta_eda$SWIFT_MSG_TYPE=='202',]
NN_202_df <- NN_202_df[,c(3,10,12)]
NN_202_df = NN_202_df %>% distinct()
NN_202_df <- NN_202_df[order(NN_202_df$TRXN_MONTH),]
glimpse(NN_202_df)

## Rows: 12
## Columns: 3
```

```
## $ TRXN_MONTH   <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
## $ AVG_TRXN_AMT <dbl> 29.51093, 24.95118, 39.24611, 32.63196, 28.07124, 31.
8...
## $ MONTH_TEXT   <chr> "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug
"...
```

```
dim(NN_202_df)
```

```
## [1] 12  3
```

```
EU_103_df <- input_data_eda[input_data_eda$CONTINENT_CODE   =='EU' & input_da
ta_eda$SWIFT_MSG_TYPE=='103',]
EU_103_df <- EU_103_df[,c(3,10,12)]
EU_103_df = EU_103_df %>% distinct()
EU_103_df <- EU_103_df[order(EU_103_df$TRXN_MONTH),]
glimpse(EU_103_df)
```

```
## Rows: 12
## Columns: 3
## $ TRXN_MONTH   <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
## $ AVG_TRXN_AMT <dbl> 132.44920, 140.76402, 211.85000, 155.32919, 121.98100
,...
## $ MONTH_TEXT   <chr> "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug
"...
```

```
dim(EU_103_df)
```

```
## [1] 12  3
```

```
EU_202_df <- input_data_eda[input_data_eda$CONTINENT_CODE   =='EU' & input_da
ta_eda$SWIFT_MSG_TYPE=='202',]
EU_202_df <- EU_202_df[,c(3,10,12)]
EU_202_df = EU_202_df %>% distinct()
EU_202_df <- EU_202_df[order(EU_202_df$TRXN_MONTH),]
glimpse(EU_202_df)
```

```
## Rows: 12
## Columns: 3
## $ TRXN_MONTH   <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
## $ AVG_TRXN_AMT <dbl> 24.61920, 22.97792, 24.90286, 23.14130, 22.19758, 21.
7...
## $ MONTH_TEXT   <chr> "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug
"...
```

```
dim(EU_202_df)
```

```
## [1] 12  3
```

```
AS_103_df <- input_data_eda[input_data_eda$CONTINENT_CODE   =='AS' & input_da
ta_eda$SWIFT_MSG_TYPE=='103',]
AS_103_df <- AS_103_df[,c(3,10,12)]
AS_103_df = AS_103_df %>% distinct()
```

```r
AS_103_df <- AS_103_df[order(AS_103_df$TRXN_MONTH),]
glimpse(AS_103_df)
```

```
## Rows: 12
## Columns: 3
## $ TRXN_MONTH   <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
## $ AVG_TRXN_AMT <dbl> 57.26533, 73.94609, 112.92542, 92.02471, 88.45564, 89
....
## $ MONTH_TEXT   <chr> "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug
"...
```

```r
dim(AS_103_df)
```

```
## [1] 12  3
```

```r
AS_202_df <- input_data_eda[input_data_eda$CONTINENT_CODE   =='AS' & input_da
ta_eda$SWIFT_MSG_TYPE=='202',]
AS_202_df <- AS_202_df[,c(3,10,12)]
AS_202_df = AS_202_df %>% distinct()
AS_202_df <- AS_202_df[order(AS_202_df$TRXN_MONTH),]
glimpse(AS_202_df)
```

```
## Rows: 12
## Columns: 3
## $ TRXN_MONTH   <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
## $ AVG_TRXN_AMT <dbl> 29.67334, 26.15255, 23.30047, 25.11515, 25.55450, 24.
5...
## $ MONTH_TEXT   <chr> "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug
"...
```

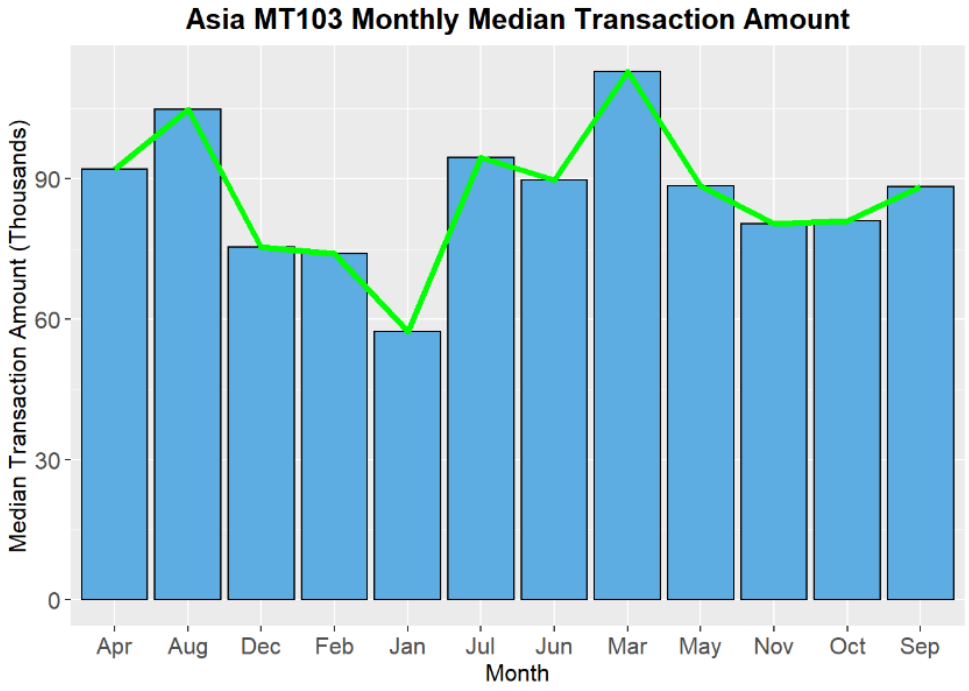
```r
dim(AS_202_df)
```

```
## [1] 12  3
```

Bar Plot

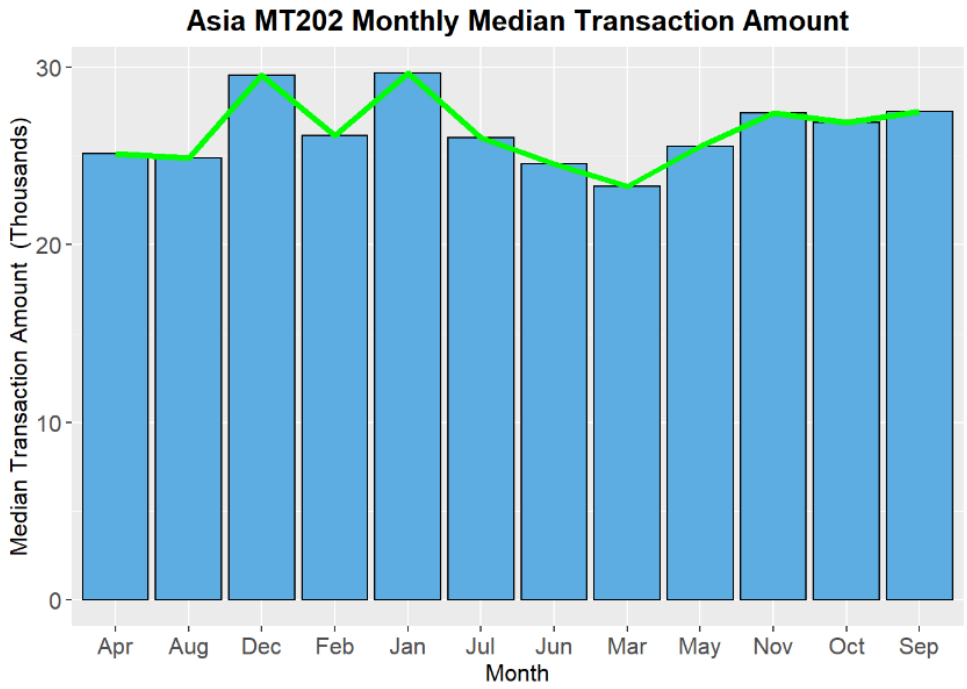
These bar plots reveal the monthly median transaction amounts for each cohort of continent, SWIFT message type and month.

```
library(ggplot2)

options(repr.plot.width = 15, repr.plot.height = 10)

AS_103_df$MONTH_TEXT <- factor(AS_103_df$MONTH_TEXT)
ggplot(AS_103_df) +
geom_bar( aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), stat = "identity", fill='#5D
ADE2', color="#000000") +
geom_line(aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), size = 1.5, color="green", g
roup = 1) +
ggtitle("Asia MT103 Monthly Median Transaction Amount") +
xlab("Month") +
ylab("Median Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=12),
      axis.title = element_text(size=12),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"))
```

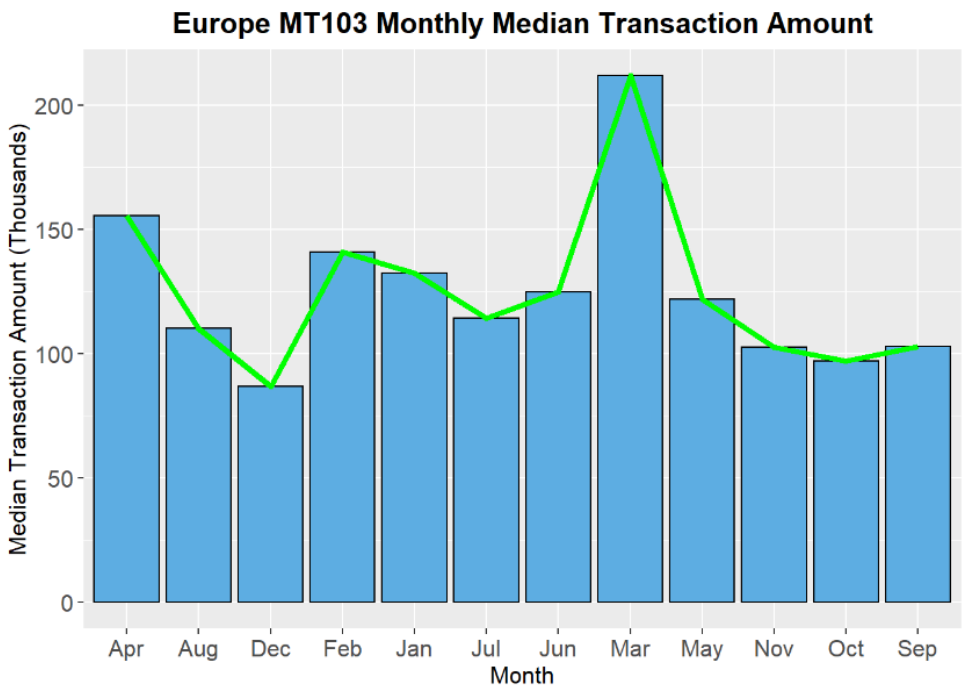



```
AS_202_df$MONTH_TEXT <- factor(AS_202_df$MONTH_TEXT)
ggplot(AS_202_df) +
geom_bar( aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), stat = "identity", fill='#5D
ADE2', color="#000000") +
geom_line(aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), size = 1.5, color="green", g
roup = 1) +
ggtitle("Asia MT202 Monthly Median Transaction Amount") +
```

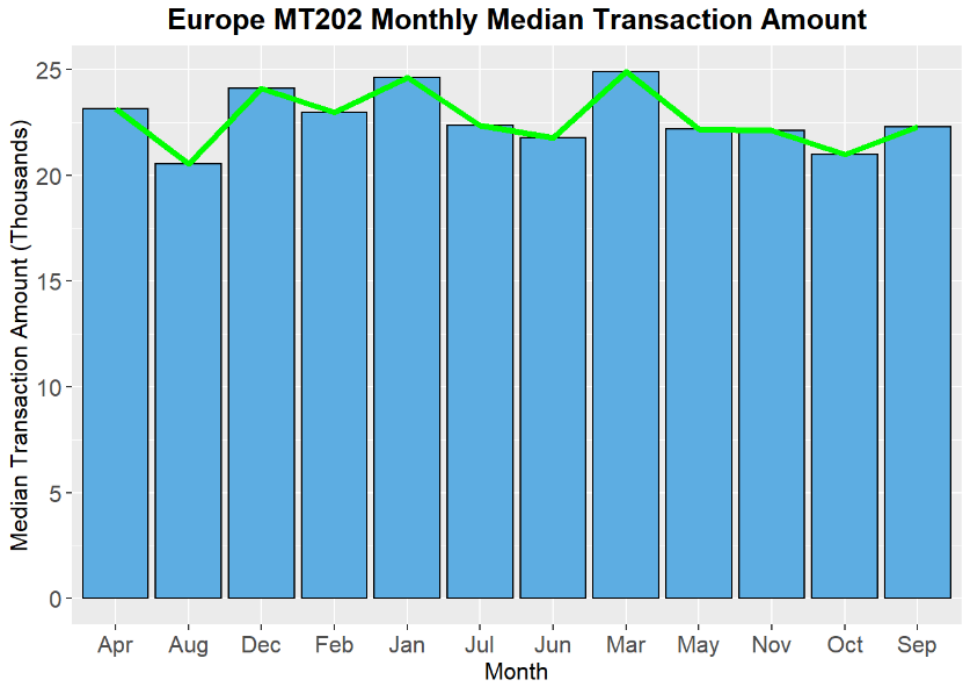
```
xlab("Month") +
ylab("Median Transaction Amount  (Thousands)") +
theme(axis.text=element_text(size=12),
      axis.title = element_text(size=12),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"))
```



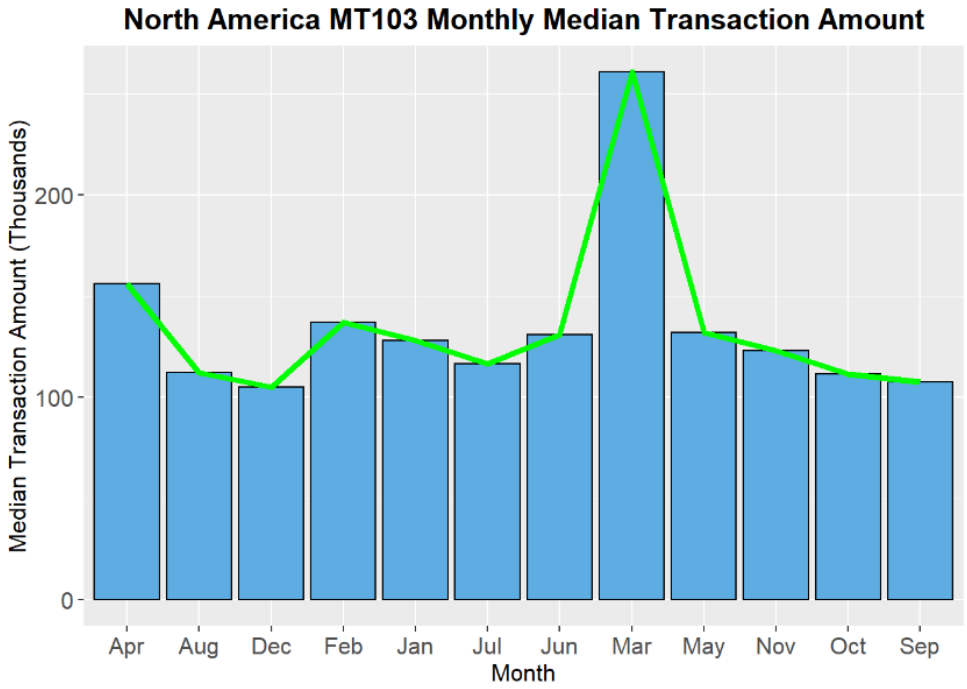

```
EU_103_df$MONTH_TEXT <- factor(EU_103_df$MONTH_TEXT)
ggplot(EU_103_df) +
geom_bar( aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), stat = "identity", fill='#5D
ADE2', color="#000000") +
geom_line(aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), size = 1.5, color="green", g
roup = 1) +
ggtitle("Europe MT103 Monthly Median Transaction Amount") +
xlab("Month") +
ylab("Median Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=12),
      axis.title = element_text(size=12),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"))
```



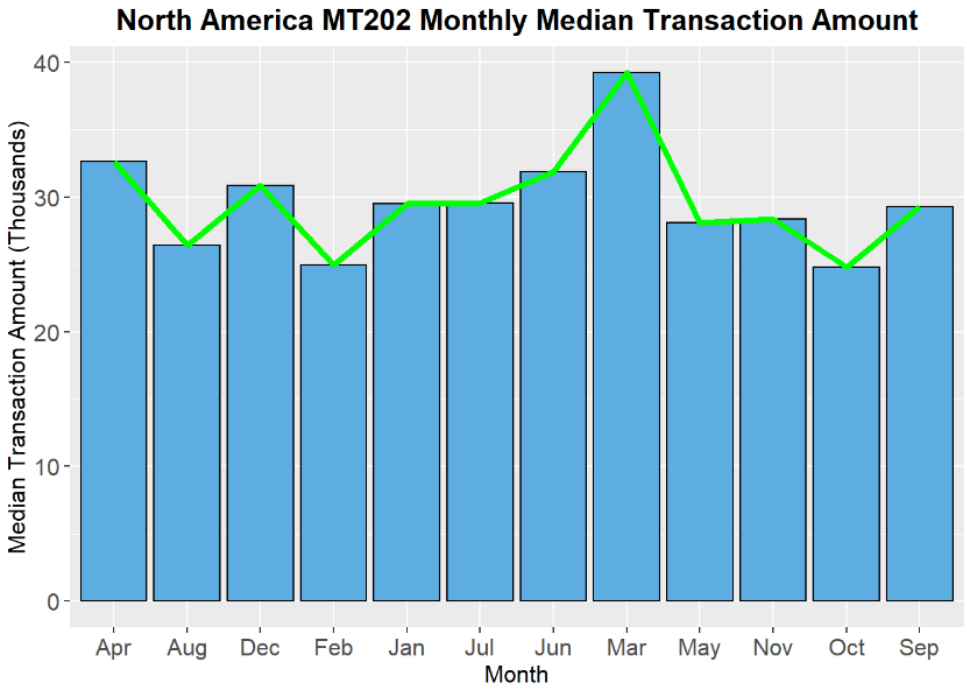

```
EU_202_df$MONTH_TEXT <- factor(EU_202_df$MONTH_TEXT)
ggplot(EU_202_df) +
geom_bar( aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), stat = "identity", fill='#5D
ADE2', color="#000000") +
geom_line(aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), size = 1.5, color="green", g
roup = 1) +
ggtitle("Europe MT202 Monthly Median Transaction Amount") +
xlab("Month") +
ylab("Median Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=12),
      axis.title = element_text(size=12),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"))
```




```
NN_103_df$MONTH_TEXT <- as.factor(NN_103_df$MONTH_TEXT)
ggplot(NN_103_df) +
geom_bar( aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), stat = "identity", fill='#5D
ADE2', color="#000000") +
geom_line(aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), size = 1.5, color="green", g
roup = 1) +
ggtitle("North America MT103 Monthly Median Transaction Amount") +
xlab("Month") +
ylab("Median Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=12),
      axis.title = element_text(size=12),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"))
```

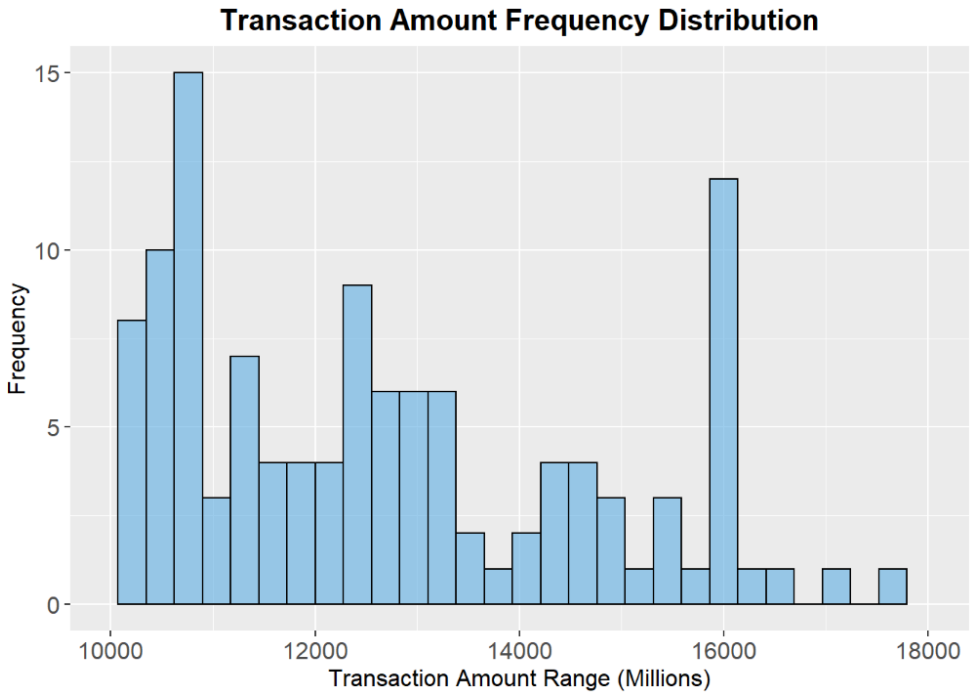



```
NN_202_df$MONTH_TEXT <- factor(NN_202_df$MONTH_TEXT)
ggplot(NN_202_df) +
geom_bar( aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), stat = "identity", fill='#5D
ADE2', color="#000000") +
geom_line(aes(x = MONTH_TEXT, y = AVG_TRXN_AMT), size = 1.5, color="green", g
roup = 1) +
ggtitle("North America MT202 Monthly Median Transaction Amount") +
xlab("Month") +
ylab("Median Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=12),
      axis.title = element_text(size=12),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"))
```



Is the data normally distributed?

```
library(ggplot2)

input_data_eda$TRANSACTION_AMOUNT <- input_data_eda$TRANSACTION_AMOUNT/100000
0

# hist(input_data_eda$TRANSACTION_AMOUNT, main = "Transaction Amount Frequenc
y Distribution", xlab="Transaction Amount Range (Thousands)")

options(repr.plot.width = 15, repr.plot.height = 10)
ggplot(data = input_data_eda, aes(x=TRANSACTION_AMOUNT)) + geom_histogram(alp
ha=0.6, fill='#5DADE2', color="#000000") + xlim(10000,18000) + ylim(0,15) +
ggtitle("Transaction Amount Frequency Distribution") +
xlab("Transaction Amount Range (Millions)") +
ylab("Frequency") +
theme(axis.text=element_text(size=12),
      axis.title = element_text(size=12),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"))

## Warning: Removed 3999878 rows containing non-finite values (stat_bin).

## Warning: Removed 2 rows containing missing values (geom_bar).
```

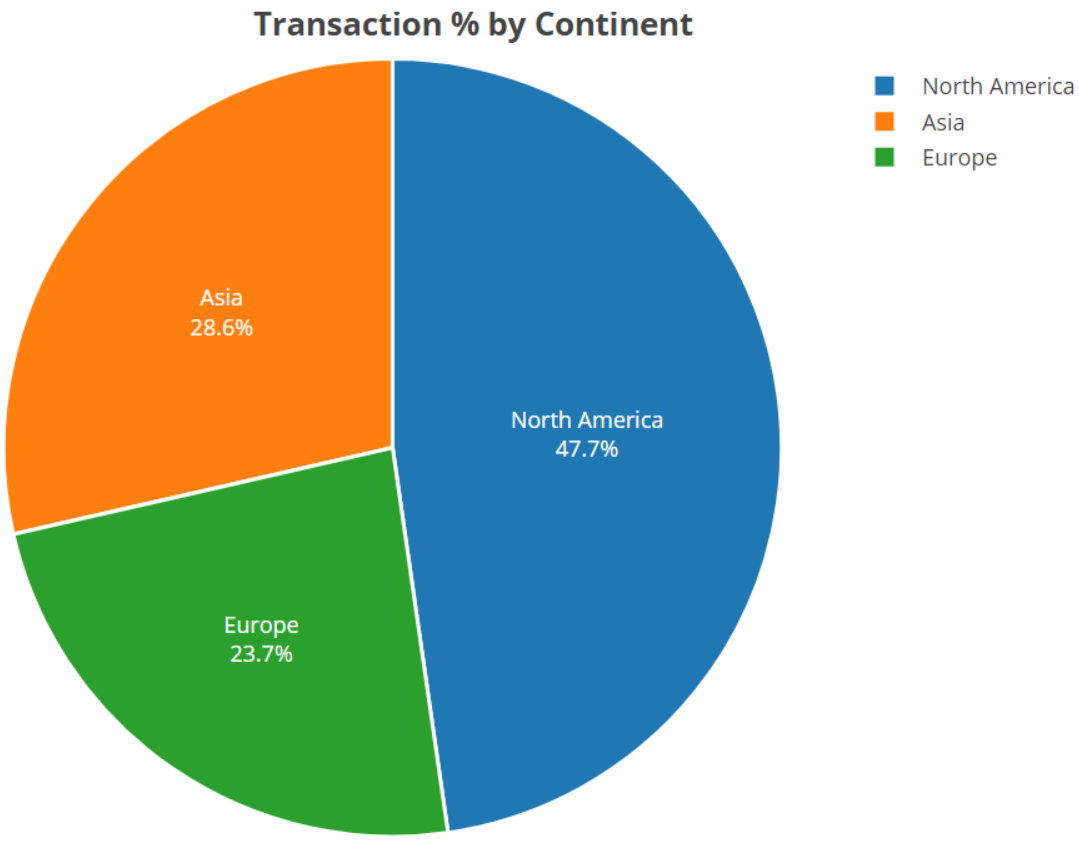

Transaction Amount Frequency Distribution
Frequency
15
10
5
0
10000
12000
14000
16000
18000
Transaction Amount Range (Millions)

Pie Chart

North America has the most number of wire tranfers.

```
  library(plotly)

  input_data_eda$pie_count = 1

  input_data_eda$CONTINENT_NAME <-  as.factor(input_data_eda$CONTINENT_NAME)

  plot_ly(input_data_eda,
          labels = ~CONTINENT_NAME,
          values = ~pie_count,
          type = 'pie',
          textposition = 'inside',
          textinfo = 'label+percent',
          insidetextfont = list(color = '#FFFFFF'),
          marker = list(colors = colors,line = list(color = '#FFFFFF', width
= 2)),
          showlegend = TRUE)  %>%
  layout(title='<b>Transaction % by Continent</b>',
         xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FA
LSE),
         yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FA
LSE))
```

```
## Warning: `arrange_()` is deprecated as of dplyr 0.7.0.
## Please use `arrange()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```



Process Runtime

```
end_time <- Sys.time()
end_time - start_time

## Time difference of 1.385403 mins
```