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.
Universtiy 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  540758 28.9    1234357   66   621331 33.2
## Vcells 1017933  7.8    8388608   64  1601224 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(easypackages)
libraries("caret","caretEnsemble","caTools","class","cluster","data.tree","devtools","doSNOW","dplyr","e1071","factoextra",
"gbm","FNN","FSelector","ggalt","ggforce","ggfortify","ggplot2","gmodels","klaR","lattice","mlbench","modeest","nnet","neura
lnet","outliers","parallel","psych","purrr","readr","rpart","rpart.plot","spatialEco","stats","tidyr","randomForest","ROSE",
"rsample","ROCR","pROC","glmnet","gridExtra","R6","Epi")
```

Import data into RStudio

```
# input_data <- read_delim("Final_cleaned_data.txt", ",", escape_double = FALSE, 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)
```

Sample data for data exploratory analysis

This sample data is for exploratory data analysis only.

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

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

# 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_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)
```

Sample data for data for plotting

This sample data is for plotting only.

```
# Sample the data
# input_data_100K <- input_data_100K[sample(nrow(input_data), 100000), ]

# Write data to storage
# write.table(input_data_100K, file="sample_df_100K.txt", append = FALSE, sep = "\t", dec = ".", row.names = FALSE, col.name
s = TRUE)

# Load data into data frame
input_data_plot <- read_delim("sample_df_100K.txt", ",", escape_double = FALSE, 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)
```

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] "4349182" "4919379" "11969294" "2219769" ...
##  $ TRANSACTION_TIME  : POSIXct[1:4000000], format: "2020-05-06 09:27:19" "2020-05-26 08:16:53" ...
##  $ TRXN_MONTH        : chr [1:4000000] "5" "5" "12" "3" ...
##  $ CLIENT_ID         : chr [1:4000000] "6249012147" "6249328247" "6249263302" "7116485839" ...
##  $ COUNTRY_NAME      : chr [1:4000000] "United States of America" "United Kingdom of Great Britain & Northern Ireland" "U
nited States of America" "United States of America" ...
##  $ COUNTRY_CODE      : chr [1:4000000] "US" "GB" "US" "US" ...
##  $ CONTINENT_NAME    : chr [1:4000000] "North America" "Europe" "North America" "North America" ...
##  $ CONTINENT_CODE    : chr [1:4000000] "NN" "EU" "NN" "NN" ...
##  $ SWIFT_MSG_TYPE    : chr [1:4000000] "202" "103" "202" "103" ...
##  $ AVG_TRXN_AMT      : num [1:4000000] 12445769 16644115 9503760 29508471 250325 ...
##  $ TRANSACTION_AMOUNT: num [1:4000000] 1140 10704000 7582 364 6936 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   TRANSACTION_ID = col_character(),
##   ..   TRANSACTION_TIME = col_datetime(format = ""),
##   ..   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()
##   .. )
```

Summary statistics of data frame

```
summary(input_data_eda)
```

```
##  TRANSACTION_ID     TRANSACTION_TIME               TRXN_MONTH
##  Length:4000000     Min.   :2020-01-01 00:48:36   Length:4000000
##  Class :character   1st Qu.:2020-03-31 19:10:55   Class :character
##  Mode  :character   Median :2020-07-05 23:45:40   Mode  :character
##                     Mean   :2020-07-05 11:23:41
##                     3rd Qu.:2020-10-06 02:39:14
##                     Max.   :2020-12-31 21:58:20
##   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.   :  203460   Min.   :0.000e+00
##  Class :character   Class :character   1st Qu.: 1849350   1st Qu.:5.440e+03
##  Mode  :character   Mode  :character   Median :10808058   Median :3.558e+04
##                                        Mean   :11563745   Mean   :1.160e+07
##                                        3rd Qu.:17532995   3rd Qu.:3.690e+05
##                                        Max.   :29508471   Max.   :1.695e+10
```

Glimpse of data frame

```
glimpse(input_data_eda)
```

```
## Rows: 4,000,000
## Columns: 11
## $ TRANSACTION_ID     <chr> "4349182", "4919379", "11969294", "2219769", "85...
## $ TRANSACTION_TIME   <dttm> 2020-05-06 09:27:19, 2020-05-26 08:16:53, 2020-...
## $ TRXN_MONTH         <chr> "5", "5", "12", "3", "1", "7", "4", "8", "12", "...
## $ CLIENT_ID          <chr> "6249012147", "6249328247", "6249263302", "71164...
## $ COUNTRY_NAME       <chr> "United States of America", "United Kingdom of G...
## $ COUNTRY_CODE       <chr> "US", "GB", "US", "US", "TH", "SG", "US", "TH", ...
## $ CONTINENT_NAME     <chr> "North America", "Europe", "North America", "Nor...
## $ CONTINENT_CODE     <chr> "NN", "EU", "NN", "NN", "AS", "AS", "NN", "AS", ...
## $ SWIFT_MSG_TYPE     <chr> "202", "103", "202", "103", "202", "103", "202",...
## $ AVG_TRXN_AMT       <dbl> 12445768.9, 16644115.0, 9503760.4, 29508471.2, 2...
## $ TRANSACTION_AMOUNT <dbl> 1139.98, 10704000.00, 7582.00, 363.78, 6935.62, ...
```

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>              <chr>      <chr>     <chr>
## 1 4349182        2020-05-06 09:27:19 5          62490121~ United Stat~
## 2 4919379        2020-05-26 08:16:53 5          62493282~ United King~
## 3 11969294       2020-12-14 10:07:03 12         62492633~ United Stat~
## 4 2219769        2020-03-05 15:20:16 3          71164858~ United Stat~
## 5 852566         2020-01-29 00:49:09 1          62491528~ Thailand-Ki~
## 6 6605370        2020-07-13 10:05:49 7          71173215~ Singapore-R~
## # ... 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>              <chr>      <chr>     <chr>
## 1 6560650        2020-07-10 10:07:13 7          71162836~ United Stat~
## 2 3676899        2020-04-16 11:30:26 4          62492633~ United Stat~
## 3 11499663       2020-11-30 18:41:51 11         71164908~ United Stat~
## 4 10673342       2020-11-06 07:17:52 11         62494104~ Thailand-Ki~
## 5 150403         2020-01-07 09:07:17 1          62492050~ Taiwan
## 6 11149693       2020-11-20 11:41:15 11         71162836~ United King~
## # ... with 6 more variables: COUNTRY_CODE <chr>, CONTINENT_NAME <chr>,
## #   CONTINENT_CODE <chr>, SWIFT_MSG_TYPE <chr>, AVG_TRXN_AMT <dbl>,
## #   TRANSACTION_AMOUNT <dbl>
```

Segegrate and prepare data for plotting

```
input_data_plot$TRANSACTION_AMOUNT=input_data_plot$TRANSACTION_AMOUNT/1000

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

NN_103_df <- input_data_plot[input_data_plot$CONTINENT_CODE =='NN' & input_data_plot$SWIFT_MSG_TYPE=='103',]
NN_103_df$TRXN_MONTH = as.integer(NN_103_df$TRXN_MONTH)
NN_103_df <- NN_103_df[order(NN_103_df$TRXN_MONTH),]
glimpse(NN_103_df)
```

```
## Rows: 14,929
## Columns: 12
## $ TRANSACTION_ID     <chr> "353396", "785734", "967771", "926613", "42307",...
## $ TRANSACTION_TIME   <dttm> 2020-01-13 12:53:36, 2020-01-27 09:02:04, 2020-...
## $ TRXN_MONTH         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ CLIENT_ID          <chr> "6249091671", "7116485839", "7116485839", "71164...
## $ COUNTRY_NAME       <chr> "United States of America", "United States of Am...
## $ COUNTRY_CODE       <chr> "US", "US", "US", "BM", "US", "US", "US", "US", ...
## $ CONTINENT_NAME     <chr> "North America", "North America", "North America...
## $ CONTINENT_CODE     <chr> "NN", "NN", "NN", "NN", "NN", "NN", "NN", "NN", ...
## $ SWIFT_MSG_TYPE     <chr> "103", "103", "103", "103", "103", "103", "103",...
## $ AVG_TRXN_AMT       <dbl> 23712557, 23712557, 23712557, 23712557, 23712557...
## $ TRANSACTION_AMOUNT <dbl> 1.464654e+06, 1.054991e+02, 8.206400e+02, 2.6375...
## $ MONTH_TEXT         <chr> "Jan", "Jan", "Jan", "Jan", "Jan", "Jan", "Jan",...
```

```
dim(NN_103_df)
```

```
## [1] 14929    12
```

```
NN_202_df <- input_data_plot[input_data_plot$CONTINENT_CODE =='NN' & input_data_plot$SWIFT_MSG_TYPE=='202',]
NN_202_df$TRXN_MONTH = as.integer(NN_202_df$TRXN_MONTH)
NN_202_df <- NN_202_df[order(NN_202_df$TRXN_MONTH),]
glimpse(NN_202_df)
```

```
## Rows: 32,884
## Columns: 12
## $ TRANSACTION_ID     <chr> "989201", "112754", "561048", "129326", "509303"...
## $ TRANSACTION_TIME   <dttm> 2020-01-31 14:41:07, 2020-01-06 11:30:29, 2020-...
## $ TRXN_MONTH         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ CLIENT_ID          <chr> "7116490843", "6249263302", "7116490843", "71164...
## $ COUNTRY_NAME       <chr> "United States of America", "United States of Am...
## $ COUNTRY_CODE       <chr> "US", "US", "US", "US", "US", "US", "US", "US", ...
## $ CONTINENT_NAME     <chr> "North America", "North America", "North America...
## $ CONTINENT_CODE     <chr> "NN", "NN", "NN", "NN", "NN", "NN", "NN", "NN", ...
## $ SWIFT_MSG_TYPE     <chr> "202", "202", "202", "202", "202", "202", "202",...
## $ AVG_TRXN_AMT       <dbl> 9597325, 9597325, 9597325, 9597325, 9597325, 959...
## $ TRANSACTION_AMOUNT <dbl> 0.15373, 49.95204, 13.27744, 4.94025, 400.06200,...
## $ MONTH_TEXT         <chr> "Jan", "Jan", "Jan", "Jan", "Jan", "Jan", "Jan",...
```

```
dim(NN_202_df)
```

```
## [1] 32884    12
```

```
EU_103_df <- input_data_plot[input_data_plot$CONTINENT_CODE =='EU' & input_data_plot$SWIFT_MSG_TYPE=='103',]
EU_103_df$TRXN_MONTH = as.integer(EU_103_df$TRXN_MONTH)
EU_103_df <- EU_103_df[order(EU_103_df$TRXN_MONTH),]
glimpse(EU_103_df)
```

```
## Rows: 15,578
## Columns: 12
## $ TRANSACTION_ID     <chr> "924027", "19395", "834920", "628924", "489264",...
## $ TRANSACTION_TIME   <dttm> 2020-01-30 17:07:01, 2020-01-02 10:15:51, 2020-...
## $ TRXN_MONTH         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ CLIENT_ID          <chr> "6249340315", "7117258150", "7116378678", "62492...
## $ COUNTRY_NAME       <chr> "Switzerland-Swiss Confederation", "United Kingd...
## $ COUNTRY_CODE       <chr> "CH", "GB", "PL", "AT", "CH", "BE", "CH", "CH", ...
## $ CONTINENT_NAME     <chr> "Europe", "Europe", "Europe", "Europe", "Europe"...
## $ CONTINENT_CODE     <chr> "EU", "EU", "EU", "EU", "EU", "EU", "EU", "EU", ...
## $ SWIFT_MSG_TYPE     <chr> "103", "103", "103", "103", "103", "103", "103",...
## $ AVG_TRXN_AMT       <dbl> 16936752, 16936752, 16936752, 16936752, 16936752...
## $ TRANSACTION_AMOUNT <dbl> 178399.99643, 94.90825, 1969.67872, 1.01909, 329...
## $ MONTH_TEXT         <chr> "Jan", "Jan", "Jan", "Jan", "Jan", "Jan", "Jan",...
```

```
dim(EU_103_df)
```

```
## [1] 15578    12
```

```
EU_202_df <- input_data_plot[input_data_plot$CONTINENT_CODE =='EU' & input_data_plot$SWIFT_MSG_TYPE=='202',]
EU_202_df$TRXN_MONTH = as.integer(EU_202_df$TRXN_MONTH)
EU_202_df <- EU_202_df[order(EU_202_df$TRXN_MONTH),]
head(EU_202_df)
```

```
## # A tibble: 6 x 12
##   TRANSACTION_ID TRANSACTION_TIME    TRXN_MONTH CLIENT_ID COUNTRY_NAME
##   <chr>          <dttm>                   <int> <chr>     <chr>
## 1 306731         2020-01-10 14:31:11          1 71160051~ United King~
## 2 189192         2020-01-08 07:14:06          1 71165081~ Russian Fed~
## 3 531583         2020-01-17 07:41:42          1 62493403~ Russian Fed~
## 4 937894         2020-01-31 06:02:45          1 62490579~ United King~
## 5 275764         2020-01-10 04:13:11          1 62492099~ Turkey-Repu~
## 6 189175         2020-01-08 07:13:29          1 62493403~ United King~
## # ... with 7 more variables: COUNTRY_CODE <chr>, CONTINENT_NAME <chr>,
## #   CONTINENT_CODE <chr>, SWIFT_MSG_TYPE <chr>, AVG_TRXN_AMT <dbl>,
## #   TRANSACTION_AMOUNT <dbl>, MONTH_TEXT <chr>
```

```
dim(EU_202_df)
```

```
## [1] 7953   12
```

```
AS_103_df <- input_data_plot[input_data_plot$CONTINENT_CODE =='AS' & input_data_plot$SWIFT_MSG_TYPE=='103',]
AS_103_df$TRXN_MONTH = as.integer(AS_103_df$TRXN_MONTH)
AS_103_df <- AS_103_df[order(AS_103_df$TRXN_MONTH),]
glimpse(AS_103_df)
```

```
## Rows: 9,438
## Columns: 12
## $ TRANSACTION_ID     <chr> "182191", "338418", "953572", "230977", "622333"...
## $ TRANSACTION_TIME   <dttm> 2020-01-08 02:17:23, 2020-01-13 09:06:06, 2020-...
## $ TRXN_MONTH         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ CLIENT_ID          <chr> "6249328247", "7116290066", "7116516010", "62493...
## $ COUNTRY_NAME       <chr> "China-People's Republic of", "China-People's Re...
## $ COUNTRY_CODE       <chr> "CN", "CN", "IN", "SG", "AE", "SG", "CN", "KR", ...
## $ CONTINENT_NAME     <chr> "Asia", "Asia", "Asia", "Asia", "Asia", "Asia", ...
## $ CONTINENT_CODE     <chr> "AS", "AS", "AS", "AS", "AS", "AS", "AS", "AS", ...
## $ SWIFT_MSG_TYPE     <chr> "103", "103", "103", "103", "103", "103", "103",...
## $ AVG_TRXN_AMT       <dbl> 10319555, 10319555, 10319555, 10319555, 10319555...
## $ TRANSACTION_AMOUNT <dbl> 17.57240, 89200.00000, 73.46510, 2083.53360, 142...
## $ MONTH_TEXT         <chr> "Jan", "Jan", "Jan", "Jan", "Jan", "Jan", "Jan",...
```

```
dim(AS_103_df)
```

```
## [1] 9438   12
```

```
AS_202_df <- input_data_plot[input_data_plot$CONTINENT_CODE =='AS' & input_data_plot$SWIFT_MSG_TYPE=='202',]
AS_202_df$TRXN_MONTH = as.integer(AS_202_df$TRXN_MONTH)
AS_202_df <- AS_202_df[order(AS_202_df$TRXN_MONTH),]
glimpse(AS_202_df)
```

```
## Rows: 19,218
## Columns: 12
## $ TRANSACTION_ID     <chr> "236101", "555817", "51989", "142674", "49311", ...
## $ TRANSACTION_TIME   <dttm> 2020-01-09 09:02:07, 2020-01-17 14:06:30, 2020-...
## $ TRXN_MONTH         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ CLIENT_ID          <chr> "6249012147", "7116048654", "6249307755", "62493...
## $ COUNTRY_NAME       <chr> "China-People's Republic of", "Singapore-Republi...
## $ COUNTRY_CODE       <chr> "CN", "SG", "CN", "CN", "TW", "VN", "CN", "CN", ...
## $ CONTINENT_NAME     <chr> "Asia", "Asia", "Asia", "Asia", "Asia", "Asia", ...
## $ CONTINENT_CODE     <chr> "AS", "AS", "AS", "AS", "AS", "AS", "AS", "AS", ...
## $ SWIFT_MSG_TYPE     <chr> "202", "202", "202", "202", "202", "202", "202",...
## $ AVG_TRXN_AMT       <dbl> 250325.2, 250325.2, 250325.2, 250325.2, 250325.2...
## $ TRANSACTION_AMOUNT <dbl> 6.57961, 181.25518, 5.63030, 0.81955, 12.00632, ...
## $ MONTH_TEXT         <chr> "Jan", "Jan", "Jan", "Jan", "Jan", "Jan", "Jan",...
```

```
dim(AS_202_df)
```

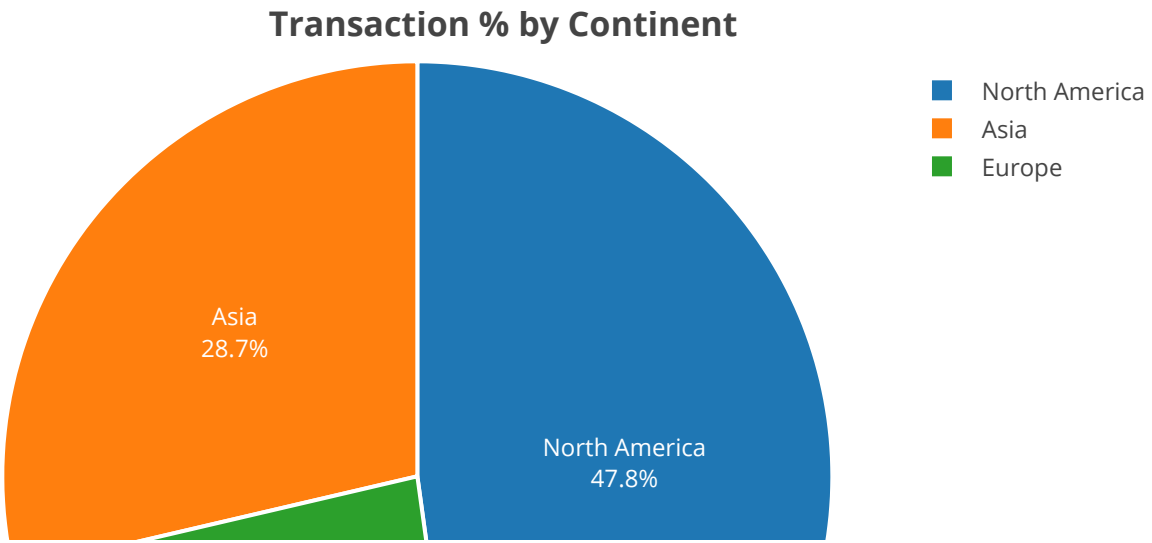
```
## [1] 19218    12
```

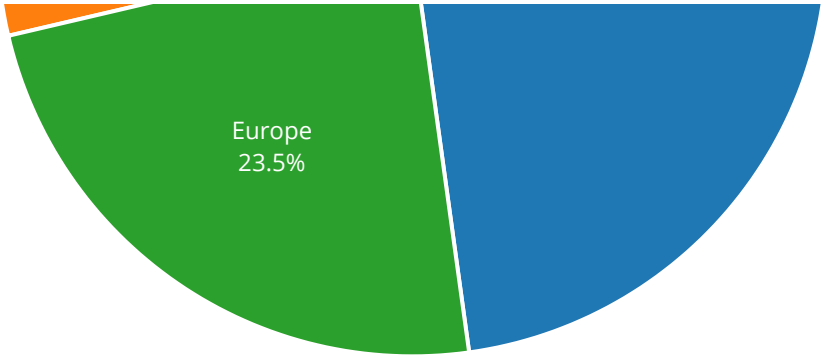
Pie Chart

North America has the most number of wire tranfers.

```
library(plotly)

input_data_plot$pie_count = 1

input_data_plot$CONTINENT_NAME <-  factor(input_data_plot$CONTINENT_NAME, levels=unique(input_data_plot$CONTINENT_NAME))

plot_ly(input_data_plot,
        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 = FALSE),
       yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
```

```
## 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.
```

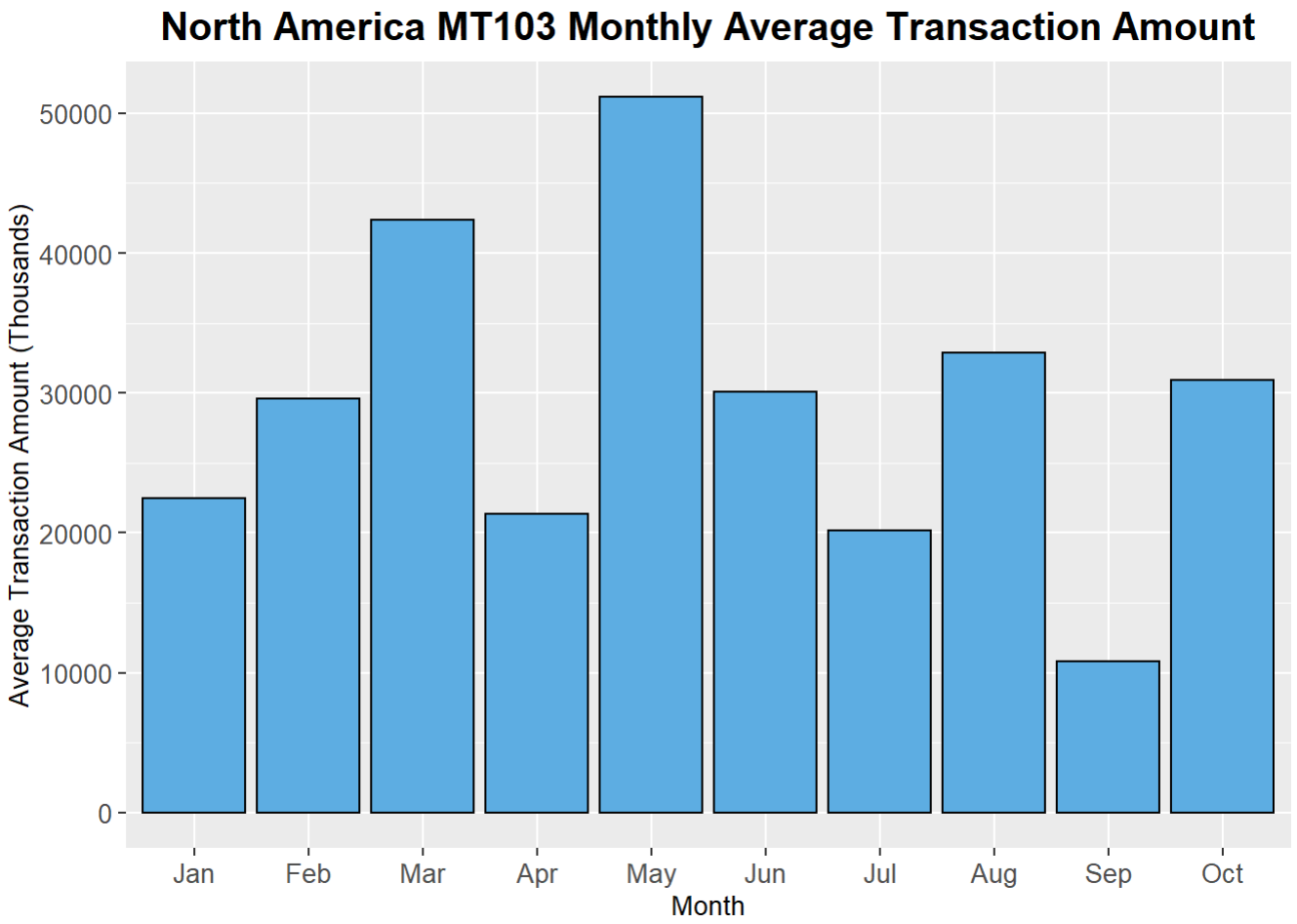




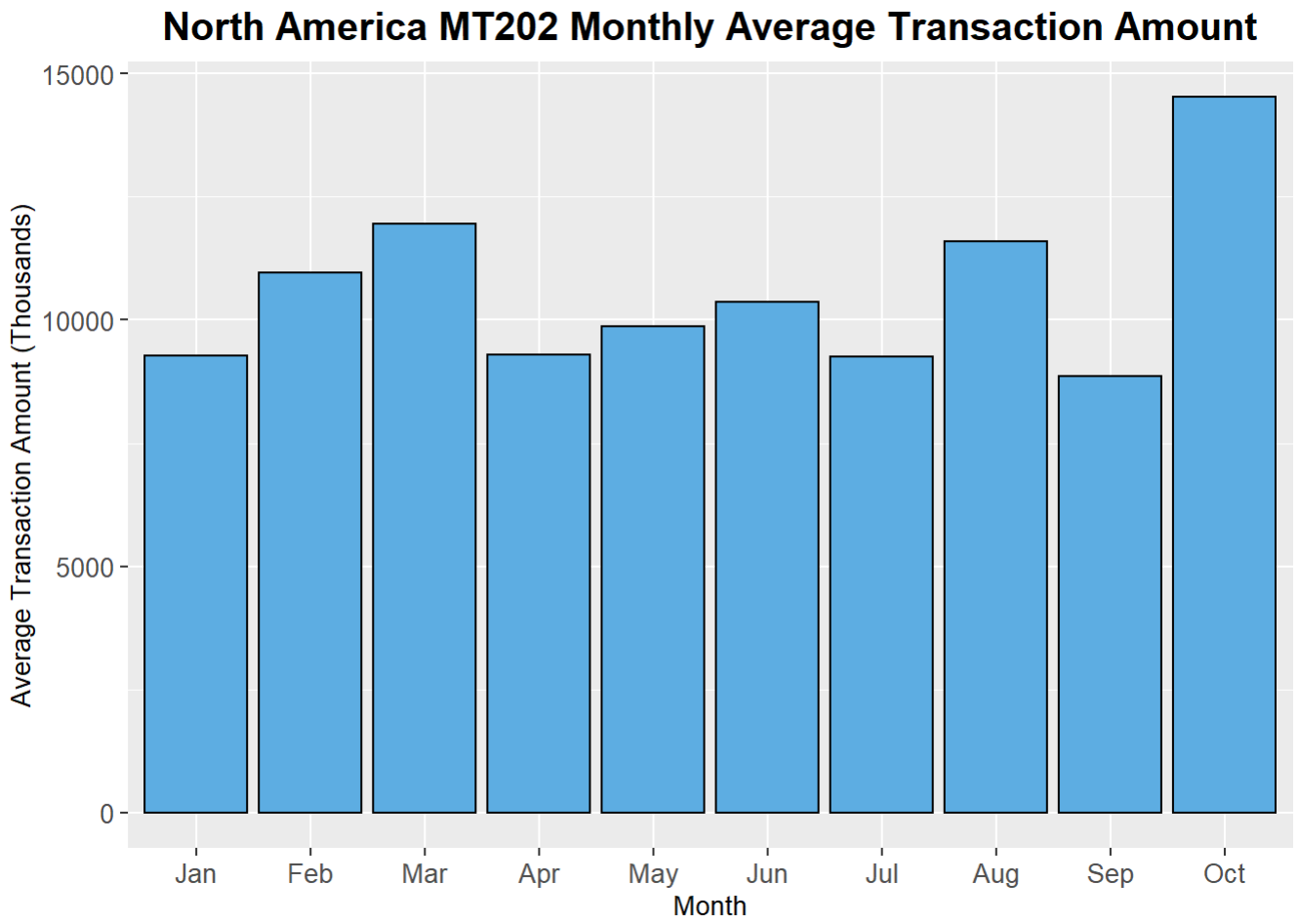
Bar Plot

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

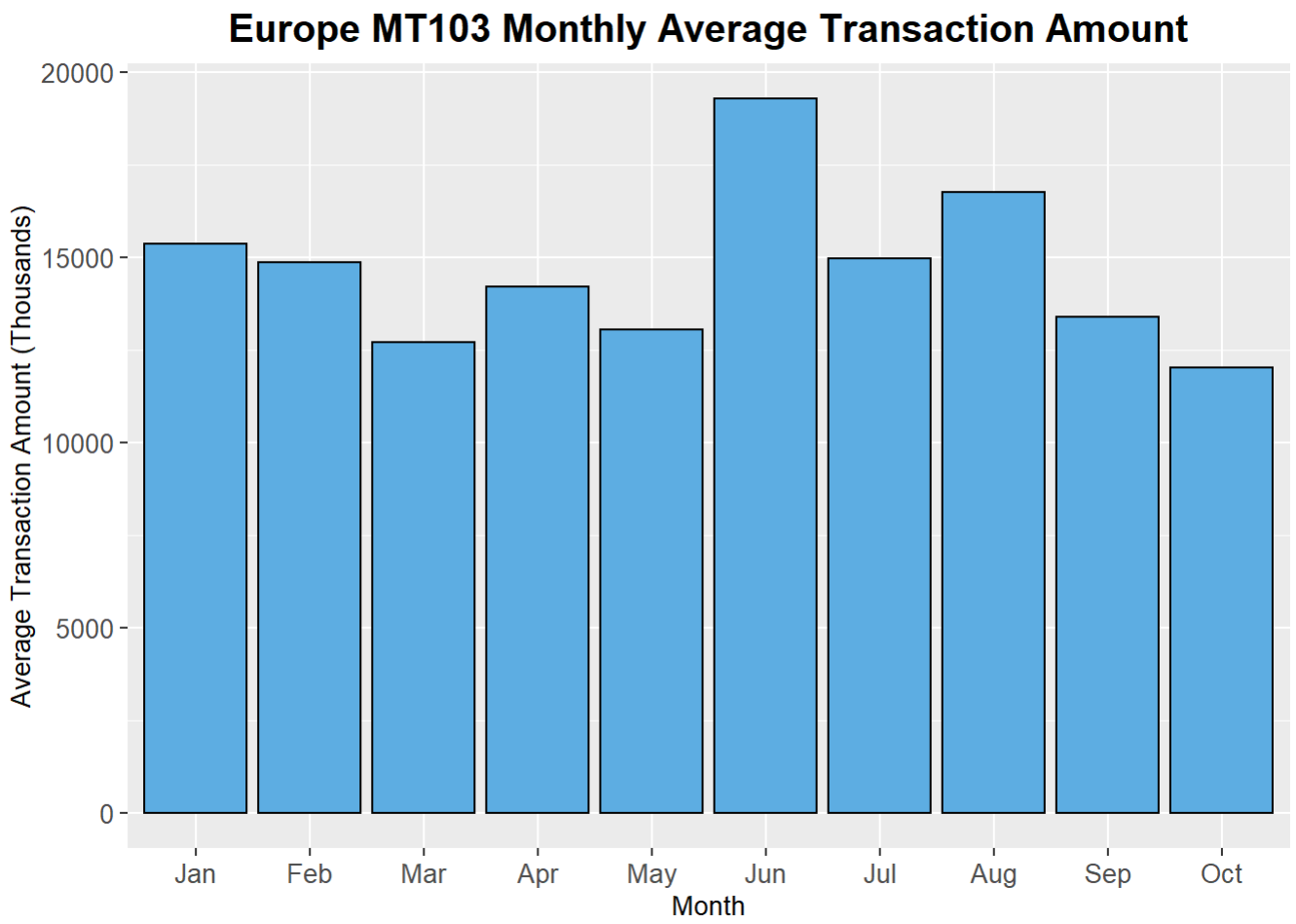
```
library(ggplot2)

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

NN_103_df$MONTH_TEXT <-  factor(NN_103_df$MONTH_TEXT, levels=unique(NN_103_df$MONTH_TEXT))
ggplot(NN_103_df, aes(x = MONTH_TEXT, y = TRANSACTION_AMOUNT)) +
geom_bar(stat = "summary", fun = "mean", fill='#5DADE2', color="#000000") +
ggtitle("North America MT103 Monthly Average Transaction Amount") +
xlab("Month") +
ylab("Average Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=10),
      axis.title = element_text(size=10),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"),
      legend.key.size = unit(1,"line"))
```



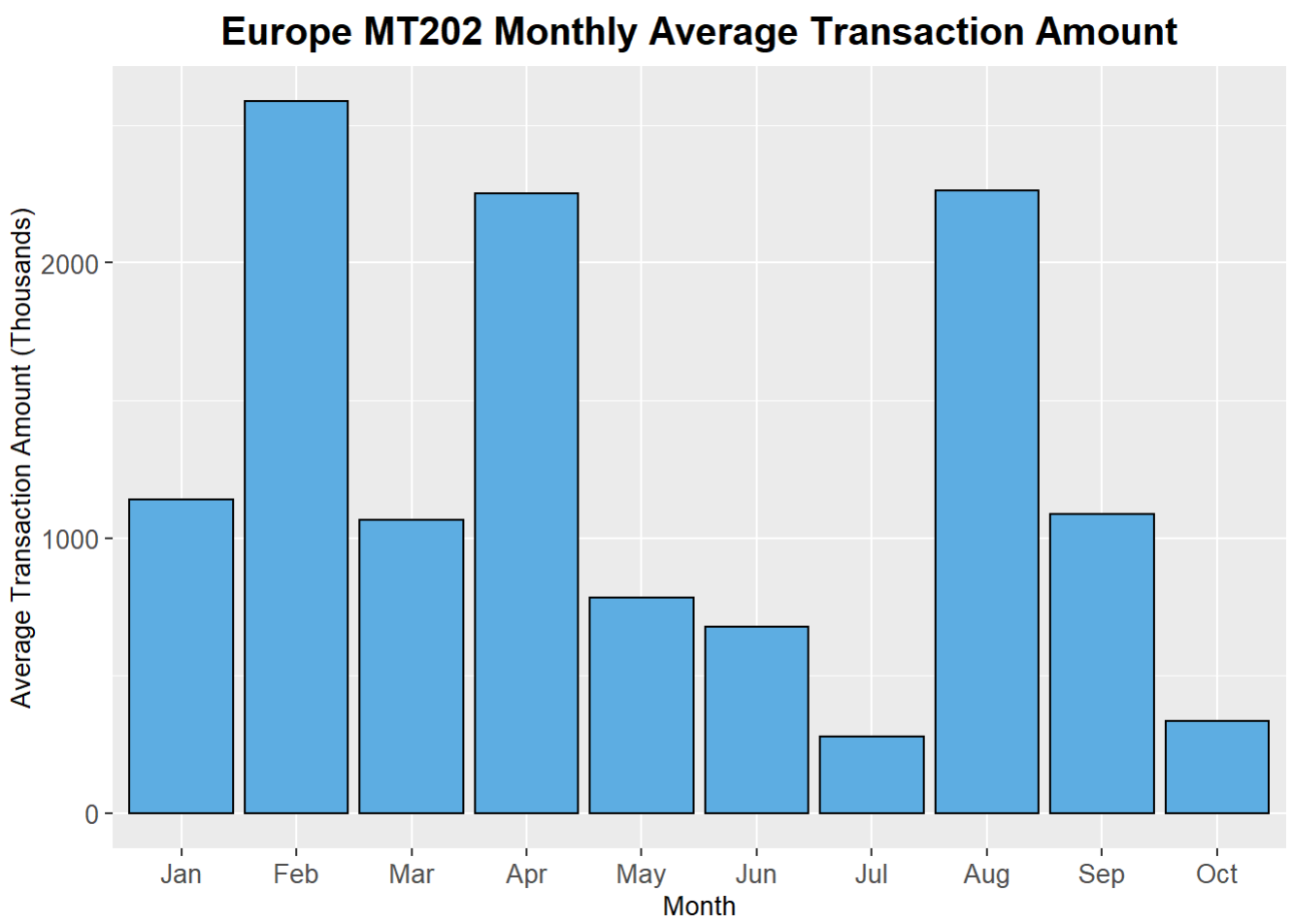

```
NN_202_df$MONTH_TEXT <-  factor(NN_202_df$MONTH_TEXT, levels=unique(NN_202_df$MONTH_TEXT))
ggplot(NN_202_df, aes(x = MONTH_TEXT, y = TRANSACTION_AMOUNT)) +
geom_bar(stat = "summary", fun = "mean", fill='#5DADE2', color="#000000") +
ggtitle("North America MT202 Monthly Average Transaction Amount") +
xlab("Month") +
ylab("Average Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=10),
      axis.title = element_text(size=10),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"),
      legend.key.size = unit(1,"line"))
```



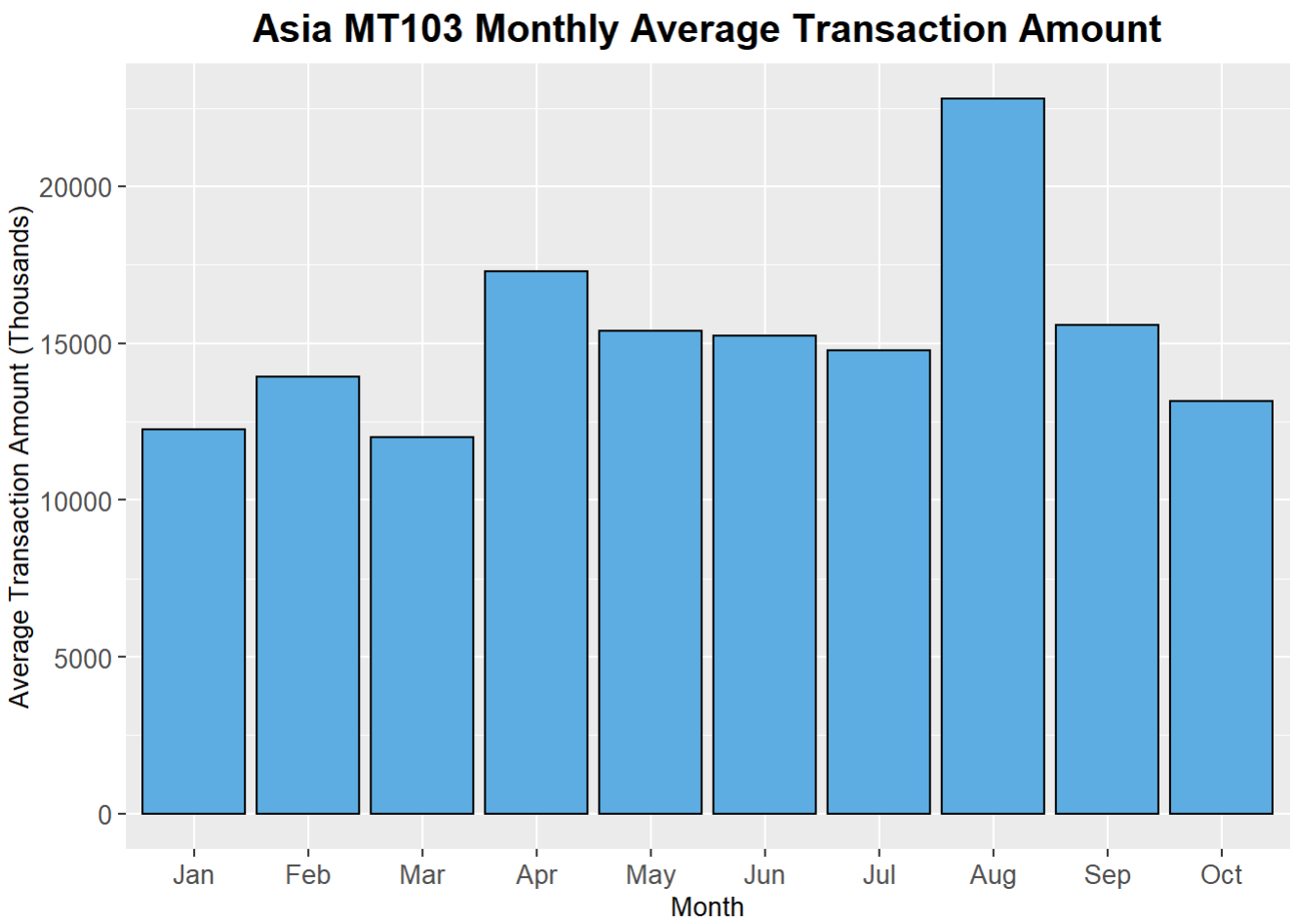

```
EU_103_df$MONTH_TEXT <-  factor(EU_103_df$MONTH_TEXT, levels=unique(EU_103_df$MONTH_TEXT))
ggplot(EU_103_df, aes(x = MONTH_TEXT, y = TRANSACTION_AMOUNT)) +
geom_bar(stat = "summary", fun = "mean", fill='#5DADE2', color="#000000") +
ggtitle("Europe MT103 Monthly Average Transaction Amount") +
xlab("Month") +
ylab("Average Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=10),
      axis.title = element_text(size=10),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"),
      legend.key.size = unit(1,"line"))
```



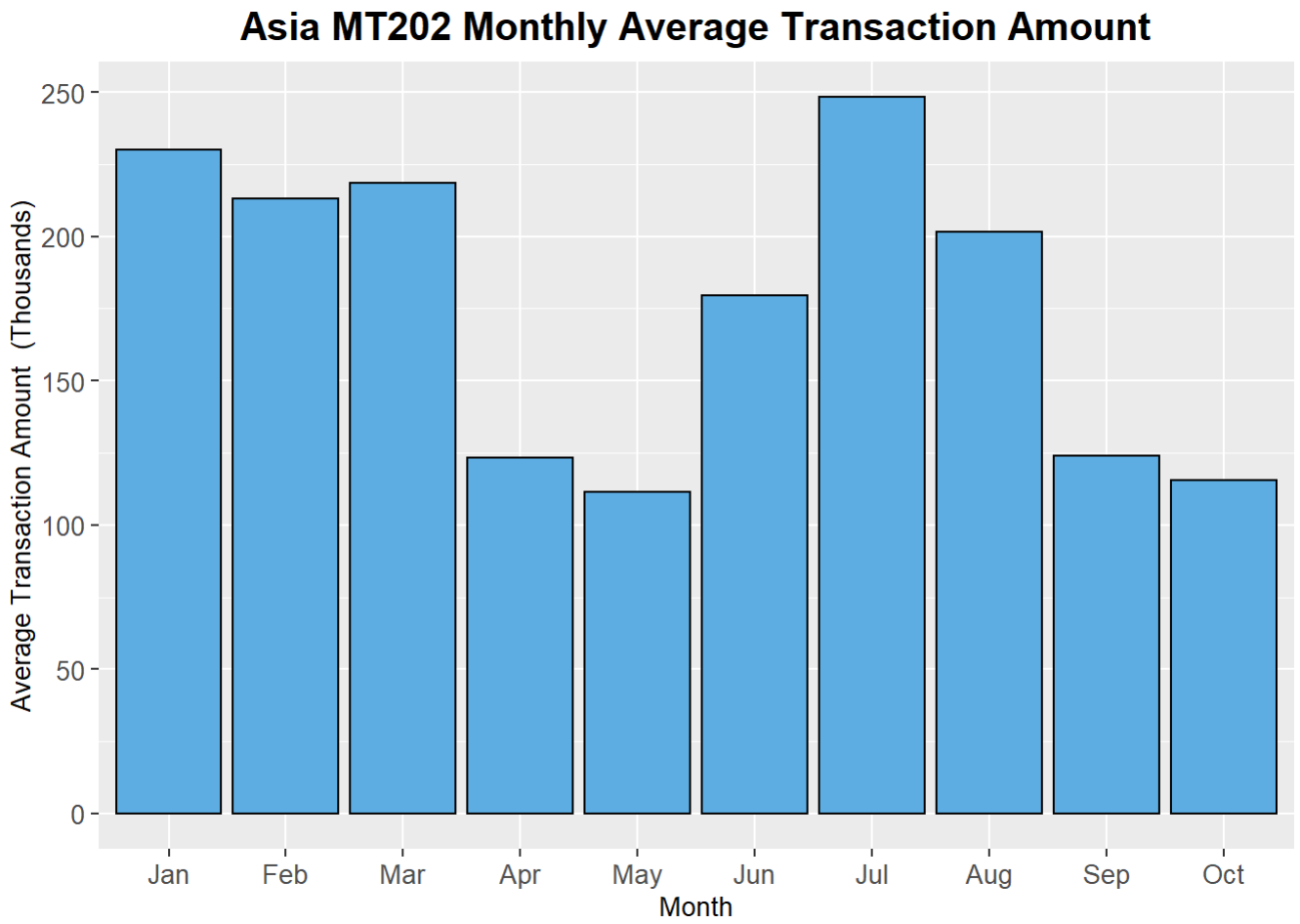

```
EU_202_df$MONTH_TEXT <-  factor(EU_202_df$MONTH_TEXT, levels=unique(EU_202_df$MONTH_TEXT))
ggplot(EU_202_df, aes(x = MONTH_TEXT, y = TRANSACTION_AMOUNT)) +
geom_bar(stat = "summary", fun = "mean", fill='#5DADE2', color="#000000") +
ggtitle("Europe MT202 Monthly Average Transaction Amount") +
xlab("Month") +
ylab("Average Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=10),
      axis.title = element_text(size=10),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"),
      legend.key.size = unit(1,"line"))
```




```
AS_103_df$MONTH_TEXT <-  factor(AS_103_df$MONTH_TEXT, levels=unique(AS_103_df$MONTH_TEXT))
ggplot(AS_103_df, aes(x = MONTH_TEXT, y = TRANSACTION_AMOUNT)) +
geom_bar(stat = "summary", fun = "mean", fill='#5DADE2', color="#000000") +
ggtitle("Asia MT103 Monthly Average Transaction Amount") +
xlab("Month") +
ylab("Average Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=10),
      axis.title = element_text(size=10),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"),
      legend.key.size = unit(1,"line"))
```

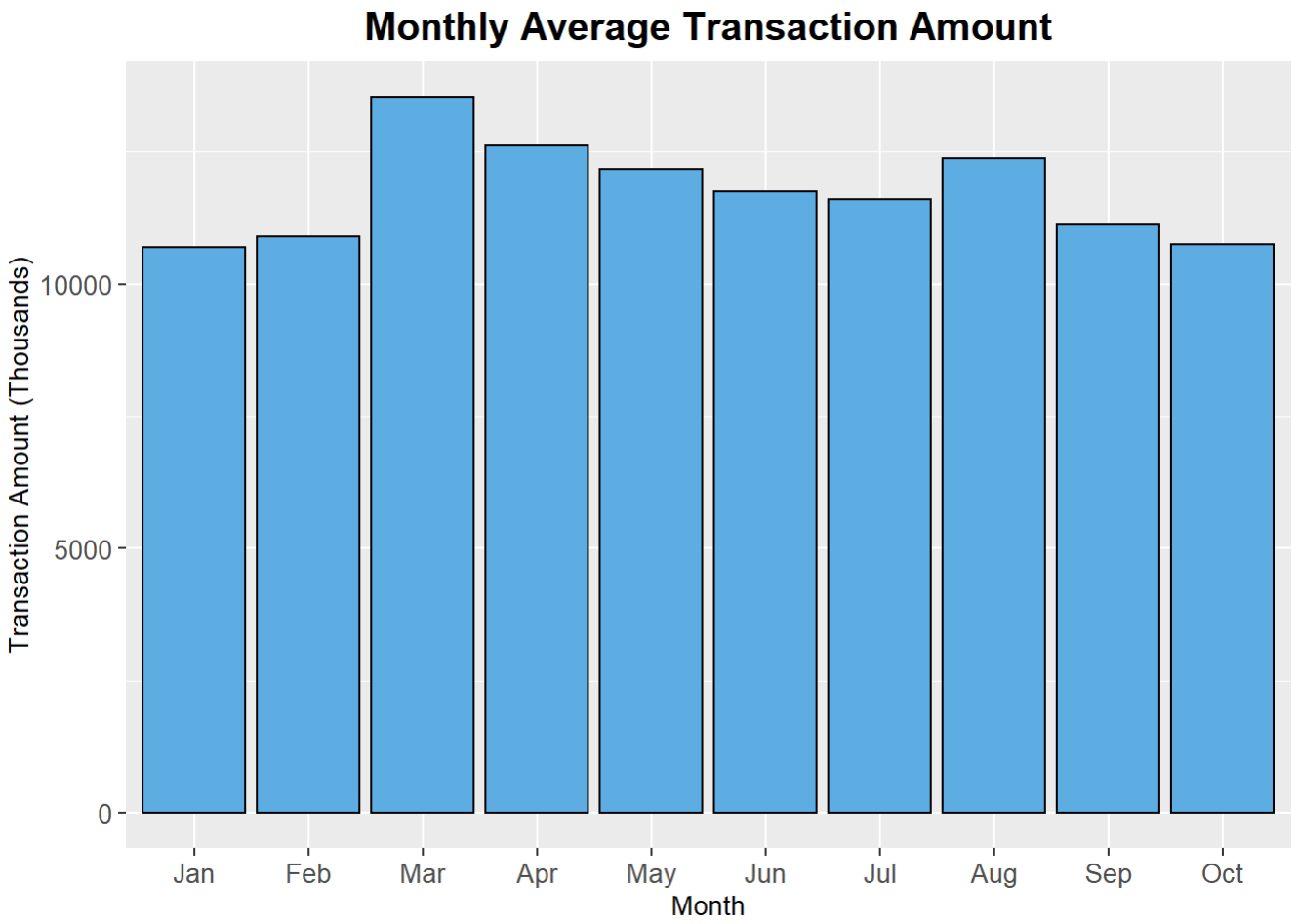



```
AS_202_df$MONTH_TEXT <-  factor(AS_202_df$MONTH_TEXT, levels=unique(AS_202_df$MONTH_TEXT))
ggplot(AS_202_df, aes(x = MONTH_TEXT, y = TRANSACTION_AMOUNT)) +
geom_bar(stat = "summary", fun = "mean", fill='#5DADE2', color="#000000") +
ggtitle("Asia MT202 Monthly Average Transaction Amount") +
xlab("Month") +
ylab("Average Transaction Amount  (Thousands)") +
theme(axis.text=element_text(size=10),
      axis.title = element_text(size=10),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"),
      legend.key.size = unit(1,"line"))
```



Is the data normally distributed?

```
library(ggplot2)

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

input_data_eda$TRXN_MONTH = as.integer(input_data_eda$TRXN_MONTH)
input_data_eda <- input_data_eda[order(input_data_eda$TRXN_MONTH),]
input_data_eda$MONTH_TEXT <-  factor(input_data_eda$MONTH_TEXT, levels=unique(input_data_eda$MONTH_TEXT))
input_data_eda$TRANSACTION_AMOUNT=input_data_eda$TRANSACTION_AMOUNT/1000

options(repr.plot.width = 15, repr.plot.height = 10)
ggplot(input_data_eda, aes(x = MONTH_TEXT, y = TRANSACTION_AMOUNT)) +
geom_bar(stat = "summary", fun = "mean", fill='#5DADE2', color="#000000") +
ggtitle("Monthly Average Transaction Amount") +
xlab("Month") +
ylab("Transaction Amount (Thousands)") +
theme(axis.text=element_text(size=10),
      axis.title = element_text(size=10),
      plot.title = element_text(hjust = 0.5, size=15,face="bold"),
      legend.key.size = unit(1,"line"))
```



Process Runtime

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

```
## Time difference of 42.80462 secs
```