```
---
output:
  pdf_document: default
  html_document: default
---
Openning File and Initial Cleaning of Data
```{r}
path <- "/Users/lourdescortes/Downloads/MarketplaceStats2015Present_en_202211.xlsx"
#Using package rio enable us to easily obtain all sheets of the excel file under a list
#stringsAsFactors to convert str into doubles doesn't seem to work, but as all the numbers
we are dealing with have 16>= digits, we can use as.numeric(x) and be confident our output
would be precise.
#install.packages("rio")
library(rio)
data <- import_list(path, col_names = TRUE, na = "", which = c(1,3,5)) #Focusing on the
Value traded, Volume and Number of Trades worksheets, hence, we eliminate undesired
worksheets

#changing headers
#install.packages("janitor")
#efficient package to expedite the initial data exploration and cleaning (Alternative is
to use funct janitor::row_to_names(file, row_num))

for (i in 1:length(data)){
  data[[names(data)[i]]] <- janitor::row_to_names(data[[names(data)[i]]], 1)
}

#Now we take a look at the df in isolation, normally we will use funct summary() but as we
are dealing with strings types we will take another approach.
#install.packages("dplyr")
library(dplyr) #convert NA's to 0's
options(digits=16) #precision of as.numeric(x)


for (i in 1:length(data)){
  data[[names(data)[i]]][is.na(data[[names(data)[i]]])] <- 0 #convert NA's to 0's
  data[[names(data)[i]]][, 3:20] <- sapply(data[[names(data)[i]]][, 3:20], as.numeric)
#making str numeric
}

#adding id to tables to help us merge them later
for (i in 1:length(data)){
  data[[names(data)[i]]] <- data[[names(data)[i]]] %>% mutate(id = row_number())
}
#merge data
merged_data <- merge(data$`Value Traded`,data$`Volume Traded`, by="id", all=TRUE)
merged_data <- merge(merged_data,data$`Number of Trades`, by="id", all=TRUE)
#delete repeated cols
merged_data <- merged_data[,-c(22,23,42,43)]
```
Opening File and Cleaning
```
path <- "/Users/lourdescortes/Downloads/Value by Market - Hoja 1.csv"
value_by_market <- read.csv(path)
#install.packages("janitor") #efficienct package to expedite the initial data exploration
and cleaning (Alternative is to use funct janitor::row_to_names(file, row_num))

value_by_market <- janitor::row_to_names(value_by_market, 1)
value_by_market  <- value_by_market[,-1]

value_by_market_use <- data.frame()
value_by_market_use <- rbind(value_by_market_use,value_by_market[(1),]) #add first row
for (i in 1:length(value_by_market$Month) - 1){
  ifelse(value_by_market$Month[i] != value_by_market$Month[i+1],value_by_market_use <-
```
```

```
rbind(value_by_market_use,value_by_market[(i+1),]),NA)
}

path <- "/Users/lourdescortes/Downloads/VOLUME TRADED BY MARKETPLACE - Hoja 1.csv"

volume_by_market <- read.csv(path)
volume_by_market  <- volume_by_market[,-1]

volume_by_market <- janitor::row_to_names(volume_by_market, 1)


volume_by_market_use <- data.frame()
volume_by_market_use <- rbind(volume_by_market_use,volume_by_market[(1),]) #add first row
for (i in 1:length(volume_by_market$Month) - 1){
  ifelse(volume_by_market$Month[i] != volume_by_market$Month[i+1],volume_by_market_use <-
rbind(volume_by_market_use,volume_by_market[(i+1),]),NA)
}


path <- "/Users/lourdescortes/Downloads/NUMBER OF TRADES BY MARKETPLACE - Hoja 1.csv"

num_trades_by_market <- read.csv(path)
num_trades_by_market <- janitor::row_to_names(num_trades_by_market, 2)
num_trades_by_market  <- num_trades_by_market[,-1]

num_trades_by_market_use <- data.frame()
num_trades_by_market_use <- rbind(num_trades_by_market_use,num_trades_by_market[(1),])
#add first row
for (i in 1:length(num_trades_by_market$Month) - 1){
  ifelse(num_trades_by_market$Month[i] !=
num_trades_by_market$Month[i+1],num_trades_by_market_use <-
rbind(num_trades_by_market_use,num_trades_by_market[(i+1),]),NA)
}

# Export created CSVs
write.csv(num_trades_by_market_use,file='/Users/lourdescortes/Downloads/num_trades.csv',
row.names=FALSE)
write.csv(volume_by_market_use,file='/Users/lourdescortes/Downloads/vol_trades.csv',
row.names=FALSE)
write.csv(value_by_market_use,file='/Users/lourdescortes/Downloads/val_trades.csv',
row.names=FALSE)
```

CSV imported from Python to aid in data cleaning process

```{r}
# Import edited CSVs from Python
path <- '/Users/lourdescortes/Desktop/1.csv'
num_trades <- read.csv(path)
num_trades  <- num_trades[,-1]

path <- '/Users/lourdescortes/Desktop/2.csv'
vol_trades <- read.csv(path)
vol_trades  <- vol_trades[,-1]

path <- '/Users/lourdescortes/Desktop/3.csv'
val_trades <- read.csv(path)
val_trades  <- val_trades[,-1]

# save a numeric vector containing 95 monthly observations
# from Jan 2015 to Nov 2022 as a time series object
ts_num_trades <- ts(num_trades[2], start=c(2015, 1), end=c(2022, 11), frequency=12)
ts_num_trades_tsx <- ts(num_trades[3], start=c(2015, 1), end=c(2022, 11), frequency=12)

ts_vol_trades <- ts(vol_trades[2], start=c(2015, 1), end=c(2022, 11), frequency=12)
ts_vol_trades_tsx <- ts(vol_trades[3], start=c(2015, 1), end=c(2022, 11), frequency=12)
```

```
ts_val_trades <- ts(val_trades[2], start=c(2015, 1), end=c(2022, 11), frequency=12)
ts_val_trades_tsx <- ts(val_trades[3], start=c(2015, 1), end=c(2022, 11), frequency=12)

```

Graphs for Value of Trades, Volume of Trades and number of Trades for All Listings
```{r}
library(readxl)
path <- "/Users/lourdescortes/Desktop/Marcos Libros/Job Search/data.xlsx"
my_data <- read_excel("/Users/lourdescortes/Desktop/Marcos Libros/Job Search/data.xlsx")
my_data_use <- data.frame()
my_data_use <- rbind(my_data_use,my_data[(1),]) #add first row

for (i in 1:length(my_data$Month) - 1){
  ifelse(my_data$Month[i] != my_data$Month[i+1],my_data_use <-
rbind(my_data_use,my_data[(i+1),]),NA)
}
ts_my_data_use <- data.frame()
ts_my_data_use <- (my_data_use[(3:7)])

matplot(ts_my_data_use, type = c("l"),pch=1,col = 1:5, main="Value Traded Ontario (Jan,
2015 - Nov, 2022)",
xlab="Months after Jan, 2015 ", ylab="Value of Trades ($)") #plot
legend("topleft", legend = c("All Marketplaces", "TSE", "TSX VE", "Nasdaq CXC", "Other"),
col=1:5, pch=1)

ts_my_data_use$percentage <- ts_my_data_use$`Value Traded at Nasdaq
CXC`/ts_my_data_use$`Value Traded at All Marketplaces`*100
a <- cbind(ts_my_data_use) #get market share for any variable over time (Value of All
Trade)

ts_my_data_use <- data.frame()
ts_my_data_use <- (my_data_use[(22:26)])

matplot(ts_my_data_use, type = c("l"),pch=1,col = 1:5, main="Volume Traded Ontario (Jan,
2015 - Nov, 2022)",
xlab="Months after Jan, 2015 ", ylab="Volume of Trades ($)") #plot
legend("topleft", legend = c("All Marketplaces", "TSE", "TSX VE", "Nasdaq CXC", "Other"),
col=1:5, pch=1)

ts_my_data_use$percentage <- ts_my_data_use$`Volume Traded at Nasdaq
CXC`/ts_my_data_use$`Volume Traded at All Marketplaces`*100
b <- cbind(ts_my_data_use) #get market share for any variable over time (Volume of All
Trade)

ts_my_data_use <- data.frame()
ts_my_data_use <- (my_data_use[(42:46)])

matplot(ts_my_data_use, type = c("l"),pch=1,col = 1:5, main="Number of Trades Ontario
(Jan, 2015 - Nov, 2022)",
xlab="Months after Jan, 2015 ", ylab="Number of Trades")
legend("topleft", legend = c("All Marketplaces", "TSE", "TSX VE", "Nasdaq CXC", "Other"),
col=1:5, pch=1) # optional legend

ts_my_data_use$percentage <- ts_my_data_use$`# Traded at Nasdaq CXC`/ts_my_data_use$`#
Traded at All Traded Marketplaces`*100
c <- cbind(ts_my_data_use) #get market share for any variable over time (Value of All
Trade)

```
Graphs for Value of Trades, Volume of Trades and number of Trades for Non-Cross Trade

```{r}
library(readxl)
```

```
my_data <- read_excel("/Users/lourdescortes/Desktop/Marcos Libros/Job Search/data.xlsx")
my_data_use <- data.frame()
my_data_use2 <- data.frame()
my_data_use <- rbind(my_data_use,my_data[(2),]) #add first row
my_data_use2 <- rbind(my_data_use,my_data[(7),]) #add first row
for (i in 1:length(my_data$Month) - 1){
  ifelse(my_data$Month[i] != my_data$Month[i+1],my_data_use <-
rbind(my_data_use,my_data[(i+2),]),NA)
  ifelse(my_data$Month[i] != my_data$Month[i+1],my_data_use2 <-
rbind(my_data_use2,my_data[(i+7),]),NA)
}
my_data_use2 <- my_data_use2[-1,]

ts_my_data_use <- data.frame()
ts_my_data_use <- (my_data_use[(3:7)])

matplot(ts_my_data_use, type = c("l"),pch=1,col = 1:5, main="Value of Non-Cross Trade
Ontario (Jan, 2015 - Nov, 2022)",
xlab="Months after Jan, 2015 ", ylab="Value of Trades ($)") #plot
legend("topleft", legend = c("All Marketplaces", "TSE", "TSX VE", "Nasdaq CXC", "Other"),
col=1:5, pch=1)

ts_my_data_use$percentage <- ts_my_data_use$`Value Traded at Other
Marketplaces`/ts_my_data_use$`Value Traded at All Marketplaces`*100
non_a <- cbind(ts_my_data_use) #get market share for any variable over time (Value of Non-
Cross Trade)

ts_my_data_use <- data.frame()
ts_my_data_use <- (my_data_use[(22:26)])

matplot(ts_my_data_use, type = c("l"),pch=1,col = 1:5, main="Volume of Non-Cross Trade
Ontario (Jan, 2015 - Nov, 2022)",
xlab="Months after Jan, 2015 ", ylab="Volume of Trades ($)") #plot
legend("topleft", legend = c("All Marketplaces", "TSE", "TSX VE", "Nasdaq CXC", "Other"),
col=1:5, pch=1)

ts_my_data_use$percentage <- ts_my_data_use$`Volume Traded at Other
Marketplaces`/ts_my_data_use$`Volume Traded at All Marketplaces`*100
non_b <- cbind(ts_my_data_use) #get market share for any variable over time (Volume of
Non-Cross Trade)

ts_my_data_use <- data.frame()
ts_my_data_use <- (my_data_use[(42:46)])

matplot(ts_my_data_use, type = c("l"),pch=1,col = 1:5, main="Number of Non-Cross Trade
Ontario (Jan, 2015 - Nov, 2022)",
xlab="Months after Jan, 2015 ", ylab="Number of Trades")
legend("topleft", legend = c("All Marketplaces", "TSE", "TSX VE", "Nasdaq CXC", "Other"),
col=1:5, pch=1) # optional legend

ts_my_data_use$percentage <- ts_my_data_use$`# Traded at Other
Marketplaces`/ts_my_data_use$`# Traded at All Traded Marketplaces`*100
non_c <- cbind(ts_my_data_use) #get market share for any variable over time (Number of
Non-Cross Trade)
```
Graphs for Value of Trades, Volume of Trades and number of Trades for Cross Trade
```{r}
my_data <- read_excel("/Users/lourdescortes/Desktop/Marcos Libros/Job Search/data.xlsx")
my_data_use <- data.frame()
my_data_use2 <- data.frame()
my_data_use <- rbind(my_data_use,my_data[(2),]) #add first row
my_data_use2 <- rbind(my_data_use,my_data[(7),]) #add first row
for (i in 1:length(my_data$Month) - 1){
  ifelse(my_data$Month[i] != my_data$Month[i+1],my_data_use <-
rbind(my_data_use,my_data[(i+2),]),NA)
```

```
  ifelse(my_data$Month[i] != my_data$Month[i+1],my_data_use2 <-
rbind(my_data_use2,my_data[(i+7),]),NA)
}
my_data_use2 <- my_data_use2[-1,]

ts_my_data_use <- data.frame()
ts_my_data_use <- (my_data_use2[(3:7)])

matplot(ts_my_data_use, type = c("l"),pch=1,col = 1:5, main="Value of Cross Trade Ontario
(Jan, 2015 - Nov, 2022)",
xlab="Months after Jan, 2015 ", ylab="Value of Trades ($)") #plot
legend("topleft", legend = c("All Marketplaces", "TSE", "TSX VE", "Nasdaq CXC", "Other"),
col=1:5, pch=1)

ts_my_data_use$percentage <- ts_my_data_use$`Value Traded at Other
Marketplaces`/ts_my_data_use$`Value Traded at All Marketplaces`*100
int_a <- cbind(ts_my_data_use) #get market share for any variable over time (Value of
Cross Trade Ontario)

ts_my_data_use <- data.frame()
ts_my_data_use <- (my_data_use2[(22:26)])

matplot(ts_my_data_use, type = c("l"),pch=1,col = 1:5, main="Volume of Cross Trade Ontario
(Jan, 2015 - Nov, 2022)",
xlab="Months after Jan, 2015 ", ylab="Volume of Trades ($)") #plot
legend("topleft", legend = c("All Marketplaces", "TSE", "TSX VE", "Nasdaq CXC", "Other"),
col=1:5, pch=1)

ts_my_data_use$percentage <- ts_my_data_use$`Volume Traded at Other
Marketplaces`/ts_my_data_use$`Volume Traded at All Marketplaces`*100
int_b <- cbind(ts_my_data_use) #get market share for any variable over time (Volume of
Cross Trade Ontario)

ts_my_data_use <- data.frame()
ts_my_data_use <- (my_data_use2[(42:46)])

matplot(ts_my_data_use, type = c("l"),pch=1,col = 1:5, main="Number of Cross Trade Ontario
(Jan, 2015 - Nov, 2022)",
xlab="Months after Jan, 2015 ", ylab="Number of Trades")
legend("topleft", legend = c("All Marketplaces", "TSE", "TSX VE", "Nasdaq CXC", "Other"),
col=1:5, pch=1) # optional legend

ts_my_data_use$percentage <- ts_my_data_use$`# Traded at Other
Marketplaces`/ts_my_data_use$`# Traded at All Traded Marketplaces`*100
int_c <- cbind(ts_my_data_use) #get market share for any variable over time (Number of
Cross Trade Ontario)
```

Miscellaneous
```{}
#plots for presentation
# plot series PLOT THEM TOGETHER CHANGE USE GGPLOT
merged_data$Month.x = vol_trades$Month
#val_trades$TSX <- merged_data$`TSX Venture Exchange.x`
#val_trades$Nasdaq.CXC <- merged_data$`Nasdaq CXC.x`
#val_trades$Other = rowSums(merged_data[,c("CSE.x", "Liquidnet.x", "MATCH Now.x",
"Omega.x", "Instinet.x", "Alpha.x", "Instinet.x", "TMX Select.x","Nasdaq CX2.x", "Lynx.x",
"NEO-N.x", "NEO-L.x", "Nasdaq CXD.x", "NEO-D.x", "CSE2.x")])


#vol_trades$TSX <- merged_data$`TSX Venture Exchange.y`
#vol_trades$Nasdaq.CXC <- merged_data$`Nasdaq CXC.y`
#val_trades$Other = rowSums(merged_data[,c("CSE.y", "Liquidnet.y", "MATCH Now.y",
"Omega.y", "Instinet.y", "Alpha.y", "Instinet.y", "TMX Select.y","Nasdaq CX2.y", "Lynx.y",
"NEO-N.y", "NEO-L.y", "Nasdaq CXD.y", "NEO-D.y", "CSE2.y")])
```

```
#num_trades$TSX <- merged_data$`TSX Venture Exchange`
#num_trades$Nasdaq.CXC <- merged_data$`Nasdaq CXC`
#num_trades$Other = rowSums(merged_data[,c("CSE", "Liquidnet", "MATCH Now", "Omega",
"Instinet", "Alpha", "Instinet", "TMX Select","Nasdaq CX2", "Lynx", "NEO-N", "NEO-L",
"Nasdaq CXD", "NEO-D", "CSE2")])

#matplot(val_trades, type = c("l"),pch=1,col = 2:3, main="Value Traded Ontario (Jan, 2015
- Nov, 2022)",
#xlab="Months after Jan, 2015 ", ylab="Value Traded ($)") #plot
#legend("topleft", legend = c("All Marketplaces", "TSE"), col=2:3, pch=1) # optional
legend

#matplot(vol_trades, type = c("l"),pch=1,col = 4:5, main="Volume Traded Ontario (Jan, 2015
- Nov, 2022)",
#xlab="Months after Jan, 2015 ", ylab="Volume Traded ($)") #plot
#legend("topleft", legend = c("All Marketplaces", "TSE"), col=4:5, pch=1) # optional
legend

#matplot(num_trades, type = c("l"),pch=1,col = 1:2, main="Number of Trades Ontario (Jan,
2015 - Nov, 2022)",
#xlab="Months after Jan, 2015 ", ylab="Number of Trades") #plot
#legend("topleft", legend = c("All Marketplaces", "TSE"), col=1:2, pch=1) # optional
legend

```

Modelling (ML and ARIMA) (Framework used for all variables!)
```{r}
#Naive Method
# Number of period we want to forecast
library(forecast)
n <- 33

# Splitting the data
train <- ts_num_trades[1:61]
test <- ts_num_trades[62:95]

# Forecast the data
model <- naive(train, h=n)

# Plot the result
autoplot(model, xlab = "Months after Jan, 2015", ylab = "Number of Trades", main= "ML
Forecast of Number of Trades after Jan 2020") + autolayer(ts(test, start=length(train)),
series = "Test Data")


# Create the Model Arima
model_arima <- auto.arima(train, seasonal=FALSE)

# Forecast n periods of the data
model <- forecast(model_arima, h=n)

# Plot the result
autoplot(model, xlab = "Months after Jan, 2015", ylab = "Number of Trades", main= "Time-
Series Forecast of Number of Trades after Jan 2020")+
  autolayer(ts(test, start= length(train)), series="Test Data")

#Check Residuals of Models and Accurracy
checkresiduals(model)
accuracy(model)
checkresiduals(model_arima)
accuracy(model_arima)
```

```

```{r}
#Naive Method
# Number of period we want to forecast
n <- 33

# Splitting the data
train <- ts_val_trades[1:61]
test <- ts_val_trades[62:95]

# Forecast the data
model <- naive(train, h=n)

# Plot the result
autoplot(model, xlab = "Months after Jan, 2015", ylab = "Value of Trades ($)", main= "ML
Forecast of Value of Trades after Jan 2020") + autolayer(ts(test, start=length(train)),
series = "Test Data")


# Create the Model Arima
model_arima <- auto.arima(train, seasonal=FALSE)

# Forecast n periods of the data
model <- forecast(model_arima, h=n)

# Plot the result
autoplot(model, xlab = "Months after Jan, 2015", ylab = "Value of Trades ($)", main=
"Time-Series Forecast of Value of Trades after Jan 2020")+
  autolayer(ts(test, start= length(train)), series="Test Data")

#Check Residuals of Models and Accurracy
checkresiduals(model)
accuracy(model)
checkresiduals(model_arima)
accuracy(model_arima)

```

```{r}
#Naive Method
# Number of period we want to forecast
n <- 33

# Splitting the data
train <- ts_vol_trades[1:61]
test <- ts_vol_trades[62:95]


# Forecast the data
model <- naive(train, h=n)

# Plot the result
autoplot(model, xlab = "Months after Jan, 2015", ylab = "Volume of Trades ($)", main= "ML
Forecast of Volume of Trades after Jan 2020") + autolayer(ts(test, start=length(train)),
series = "Test Data")


# Create the Model Arima
model_arima <- auto.arima(train, seasonal=FALSE)

# Forecast n periods of the data
model <- forecast(model_arima, h=n)

# Plot the result
```
```

```
autoplot(model, xlab = "Months after Jan, 2015", ylab = "Volume of Trades ($)", main=
"Time-Series Forecast of Volume of Trades after Jan 2020")+
  autolayer(ts(test, start= length(train)), series="Test Data")

#Check Residuals of Models and Accurracy
checkresiduals(model)
accuracy(model)
checkresiduals(model_arima)
accuracy(model_arima)
```