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 deali
#install.packages("rio")
library(rio)
data <- import_list(path, col_names = TRUE, na = "", which = c(1,3,5)) #Focusing on the Value traded, V
```

```
## New names:
## New names:
## New names:
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
## * `` -> `...7`
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...11`
## * `` -> `...12`
## * `` -> `...13`
## * `` -> `...14`
## * `` -> `...15`
## * `` -> `...16`
## * `` -> `...17`
## * `` -> `...18`
## * `` -> `...19`
## * `` -> `...20`
```

```r
#changing headers
#install.packages("janitor")
#efficient package to expedite the initial data exploration and cleaning (Alternative is to use funct j

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
#install.packages("dplyr")
library(dplyr) #convert NA's to 0's
```

```
## Warning: package 'dplyr' was built under R version 4.1.2

##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
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 nume
}

#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

```r
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

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_ma
}

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_
}


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 <- rb
}

# 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

```
# 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)
```

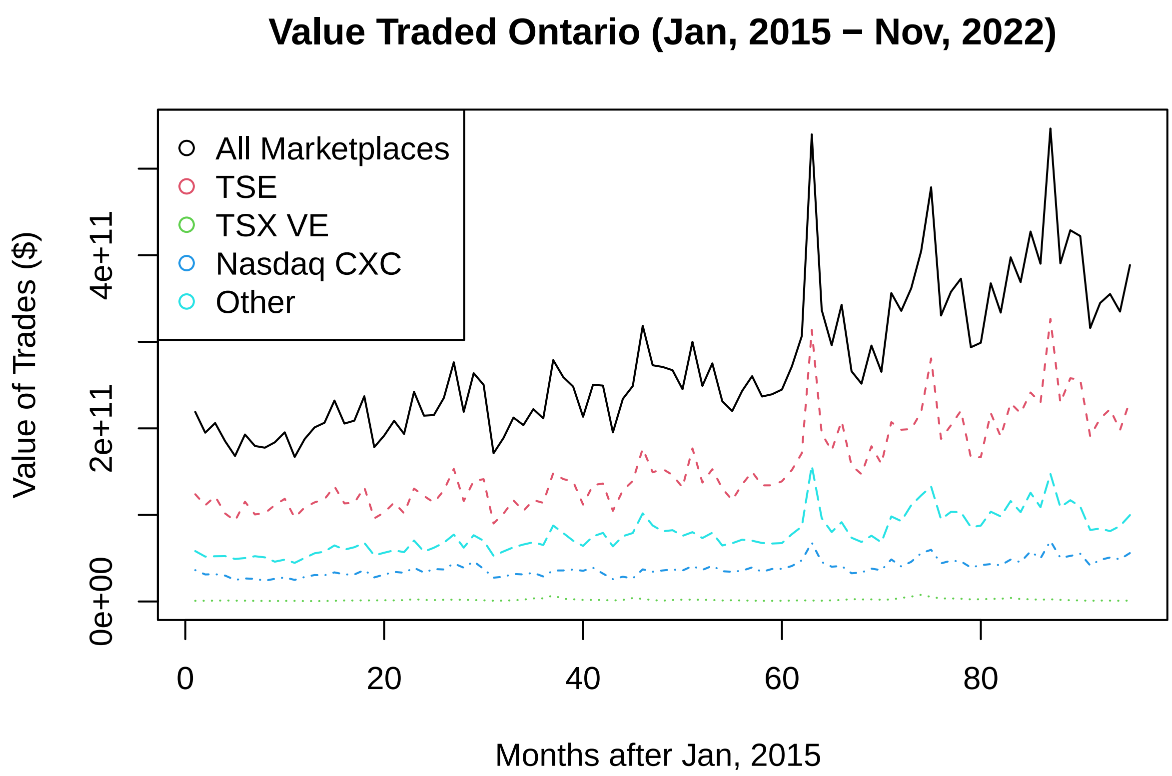
Graphs for Value of Trades, Volume of Trades and number of Trades for All Listings

```
library(readxl)
```

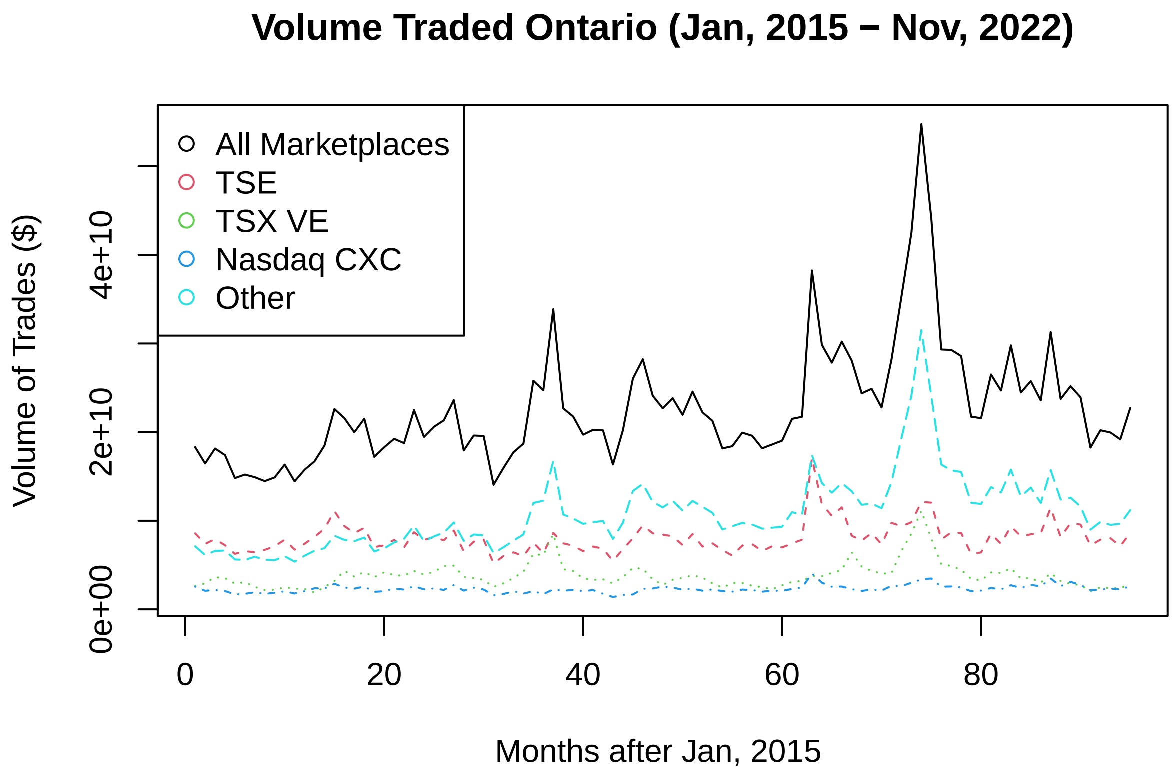
```
## Warning: package 'readxl' was built under R version 4.1.2
```

```
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
```

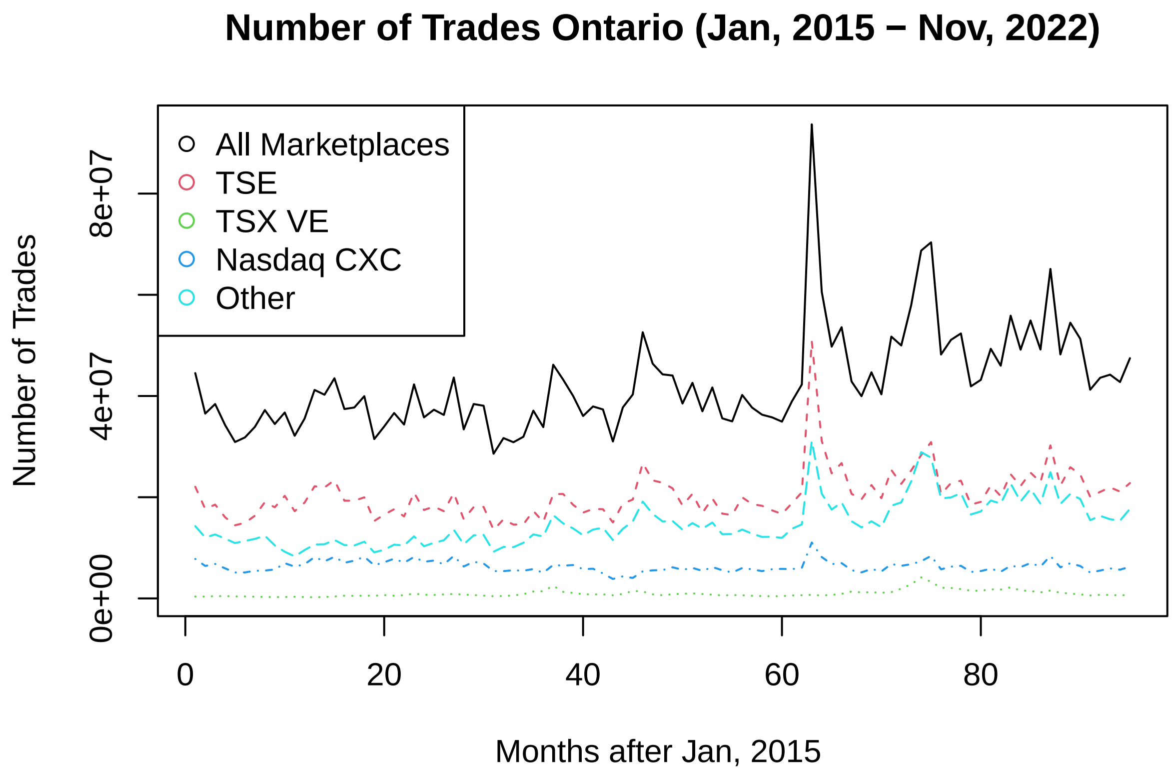
```r
for (i in 1:length(my_data$Month) - 1){
  ifelse(my_data$Month[i] != my_data$Month[i+1],my_data_use <- rbind(my_data_use,my_data[(i+1),]),NA)
}
ts_my_data_use <- data.frame()
ts_my_data_use <- (my_data_use[(3:7)])

matplot(ts_my_data_use, type = c("l"),pch=1,col = 1:5, main="Value Traded Ontario (Jan, 2015 - Nov, 202
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=
```




```r
ts_my_data_use$percentage <- ts_my_data_use$`Value Traded at Nasdaq CXC`/ts_my_data_use$`Value Traded a
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, 20
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=
```



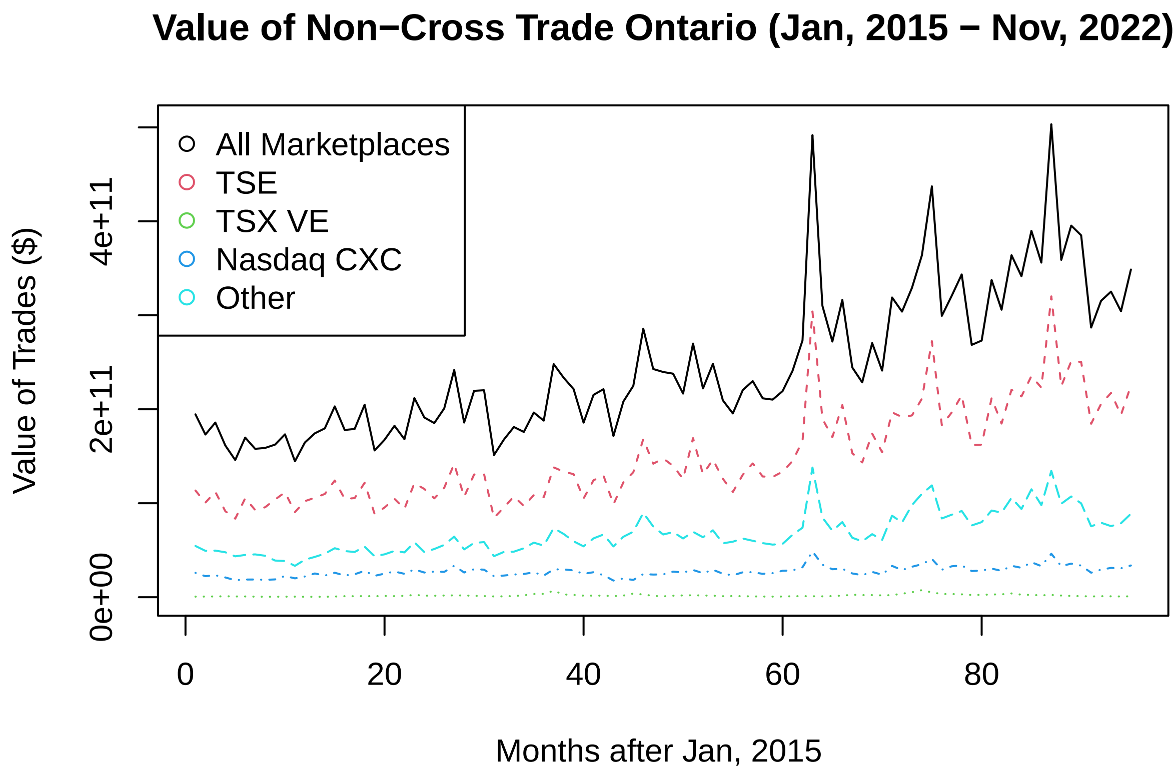

```
ts_my_data_use$percentage <- ts_my_data_use$`Volume Traded at Nasdaq CXC`/ts_my_data_use$`Volume Traded
b <- cbind(ts_my_data_use) #get market share for any variable over time (Volume of All Trade)

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,
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=
```



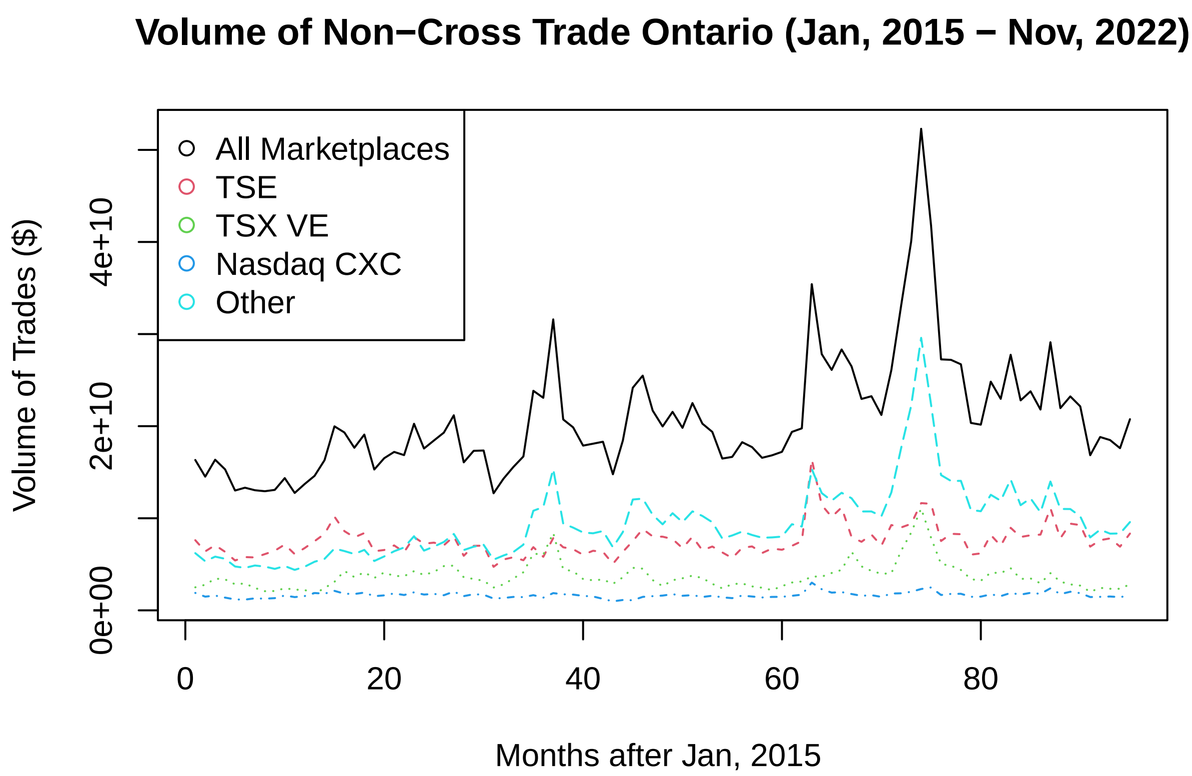

```
ts_my_data_use$percentage <- ts_my_data_use$`# Traded at Nasdaq CXC`/ts_my_data_use$`# Traded at All Tra
c <- cbind(ts_my_data_use) #get market share for any variable over time (Value of All Trade)
```

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

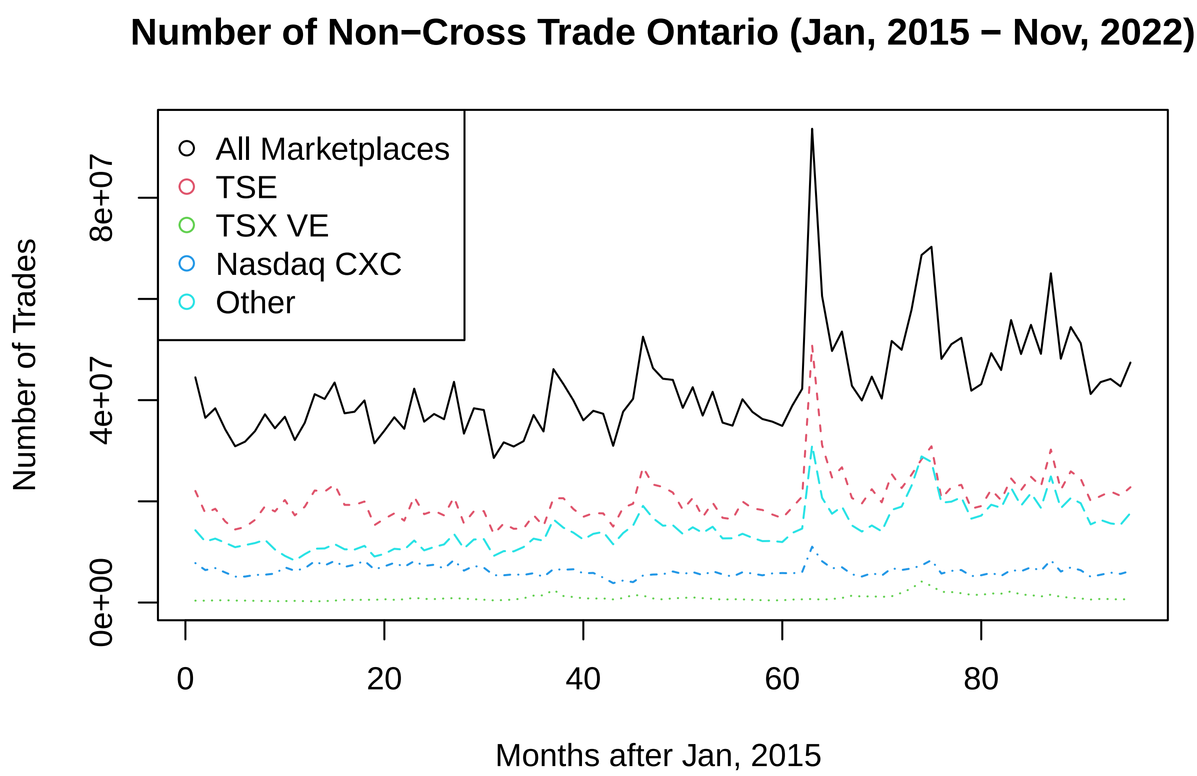
```
library(readxl)
my_data <- read_excel("/Users/lourdescortes/Desktop/Marcos Libros/Job Search/data.xlsx")
my_data_use <- data.frame()
my_data_use2 <- data.frame()
my_data_use <- rbind(my_data_use,my_data[(2),]) #add first row
my_data_use2 <- rbind(my_data_use,my_data[(7),]) #add first row
for (i in 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
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=
```




```
ts_my_data_use$percentage <- ts_my_data_use$`Value Traded at Other Marketplaces`/ts_my_data_use$`Value `
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, 20
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=
```



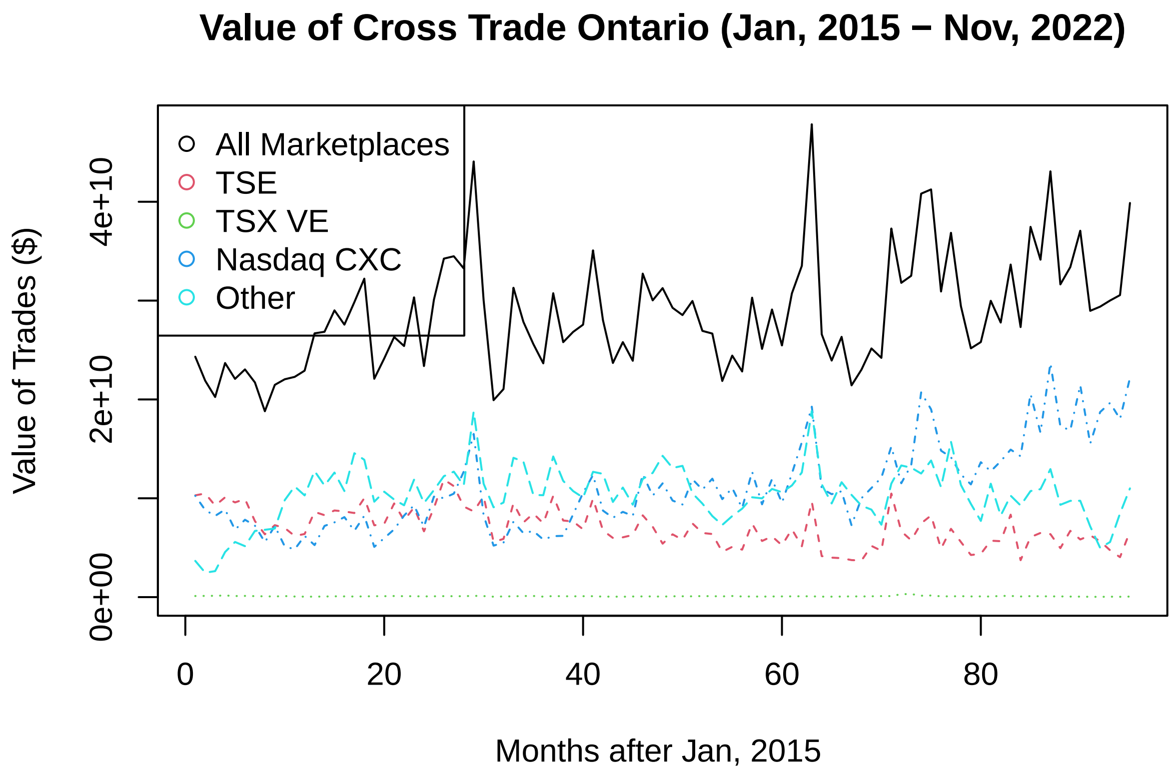

```
ts_my_data_use$percentage <- ts_my_data_use$`Volume Traded at Other Marketplaces`/ts_my_data_use$`Volum
non_b <- cbind(ts_my_data_use) #get market share for any variable over time (Volume of Non-Cross Trade)

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, 20
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=
```



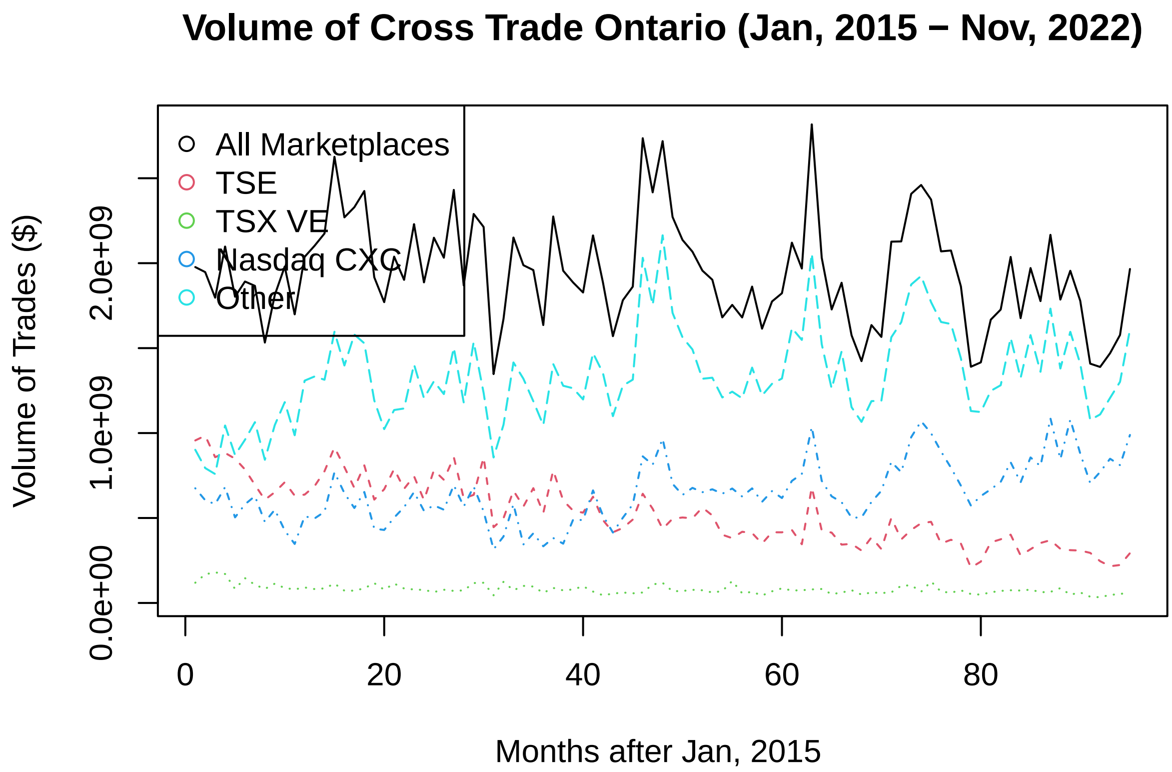

```
ts_my_data_use$percentage <- ts_my_data_use$`# Traded at Other Marketplaces`/ts_my_data_use$`# Traded a
non_c <- cbind(ts_my_data_use) #get market share for any variable over time (Number of Non-Cross Trade)
```

Graphs for Value of Trades, Volume of Trades and number of Trades for Cross Trade

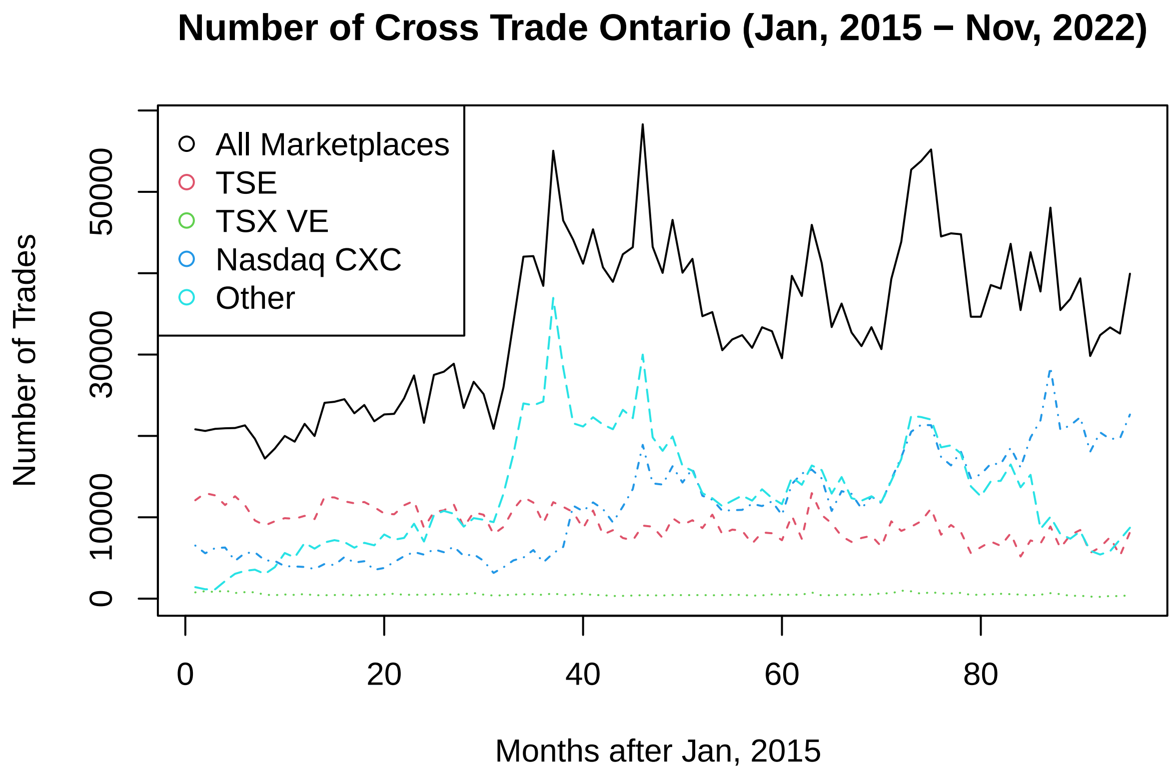
```
my_data <- read_excel("/Users/lourdescortes/Desktop/Marcos Libros/Job Search/data.xlsx")
my_data_use <- data.frame()
my_data_use2 <- data.frame()
my_data_use <- rbind(my_data_use,my_data[(2),]) #add first row
my_data_use2 <- rbind(my_data_use,my_data[(7),]) #add first row
for (i in 1:length(my_data$Month)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




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

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 -
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=
```




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

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 -
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=
```




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

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.


#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.

#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

#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, 20
#xlab="Months after Jan, 2015 ", ylab="Number of Trades") #plot
#legend("topleft", legend = c("All Marketplaces", "TSE"), col=1:2, pch=1) # optional legend
```

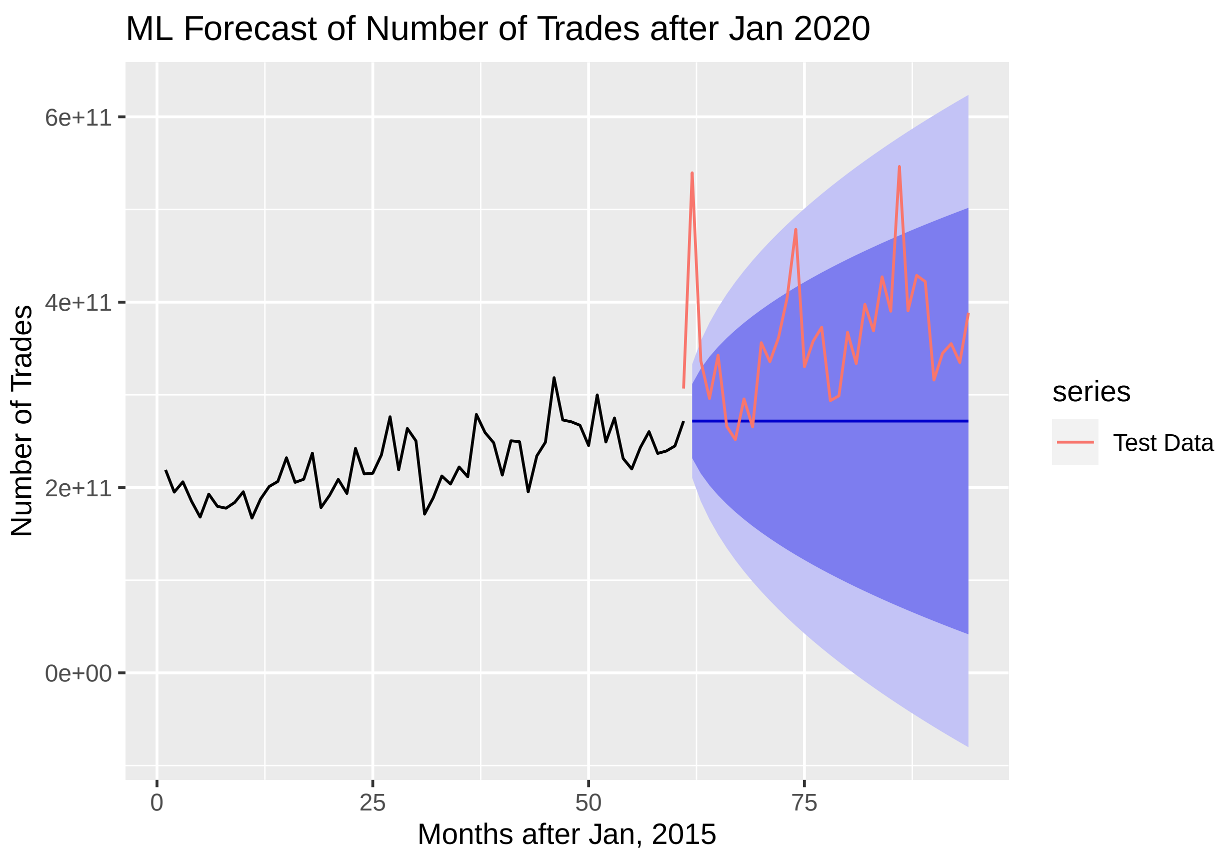
Modelling (ML and ARIMA) (Framework used for all variables!)

```
#Naive Method
# Number of period we want to forecast
library(forecast)
```

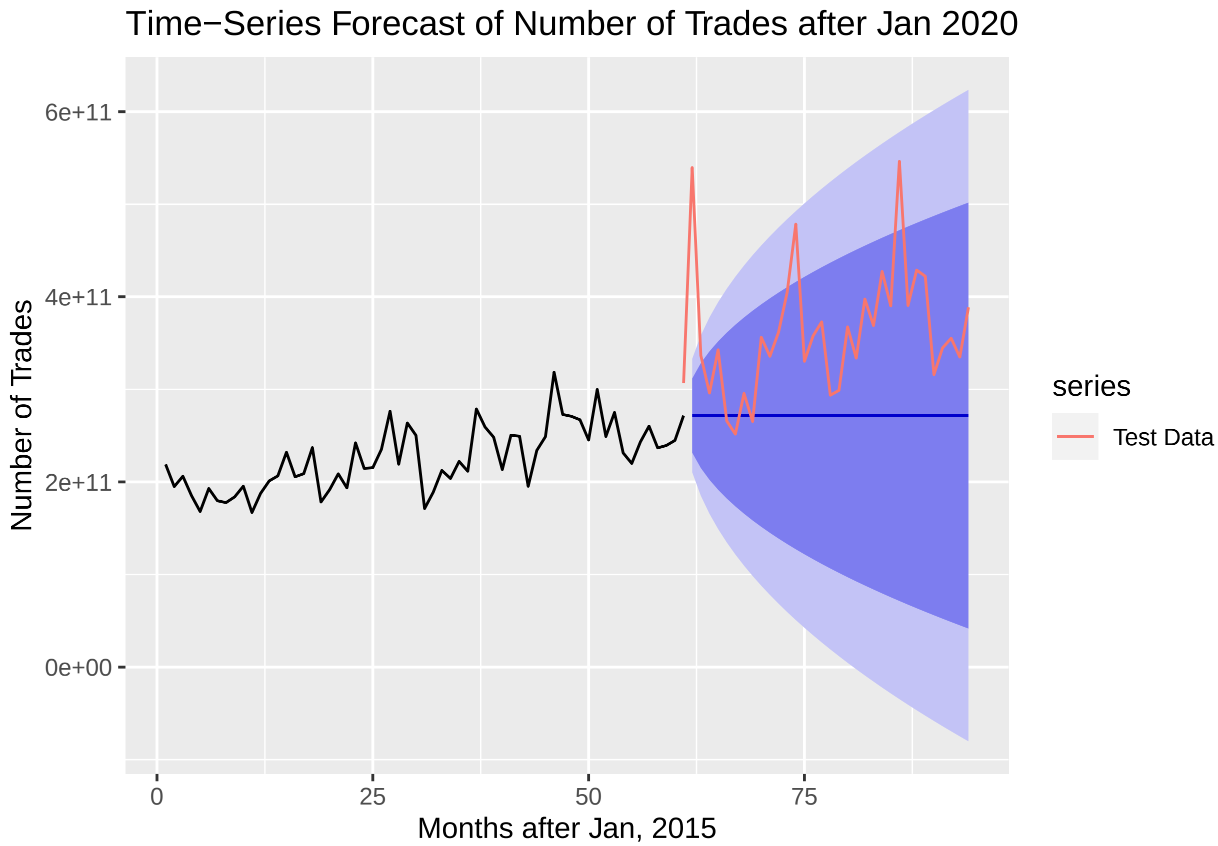
```
## Warning: package 'forecast' was built under R version 4.1.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

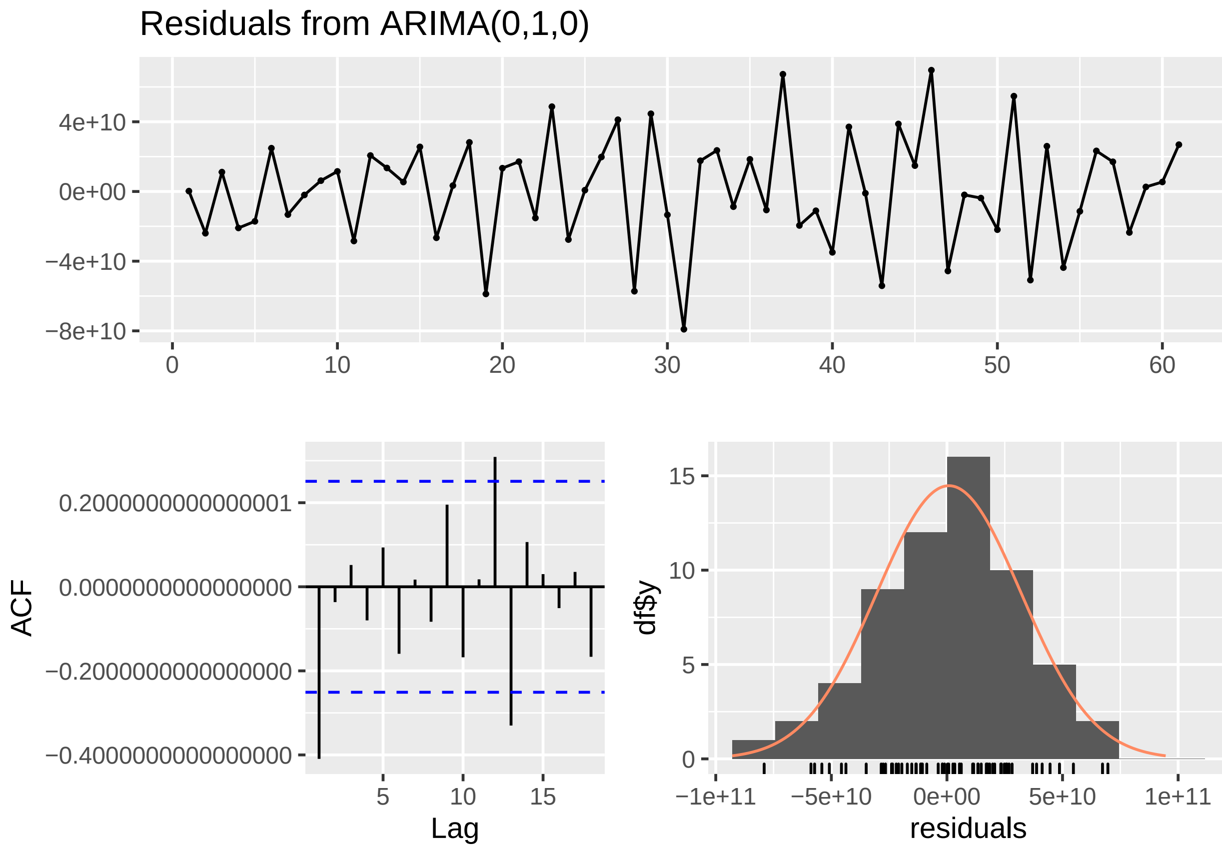
```
n <- 33

# 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 Numbe
```




```r
# 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
  autolayer(ts(test, start= length(train)), series="Test Data")
```




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

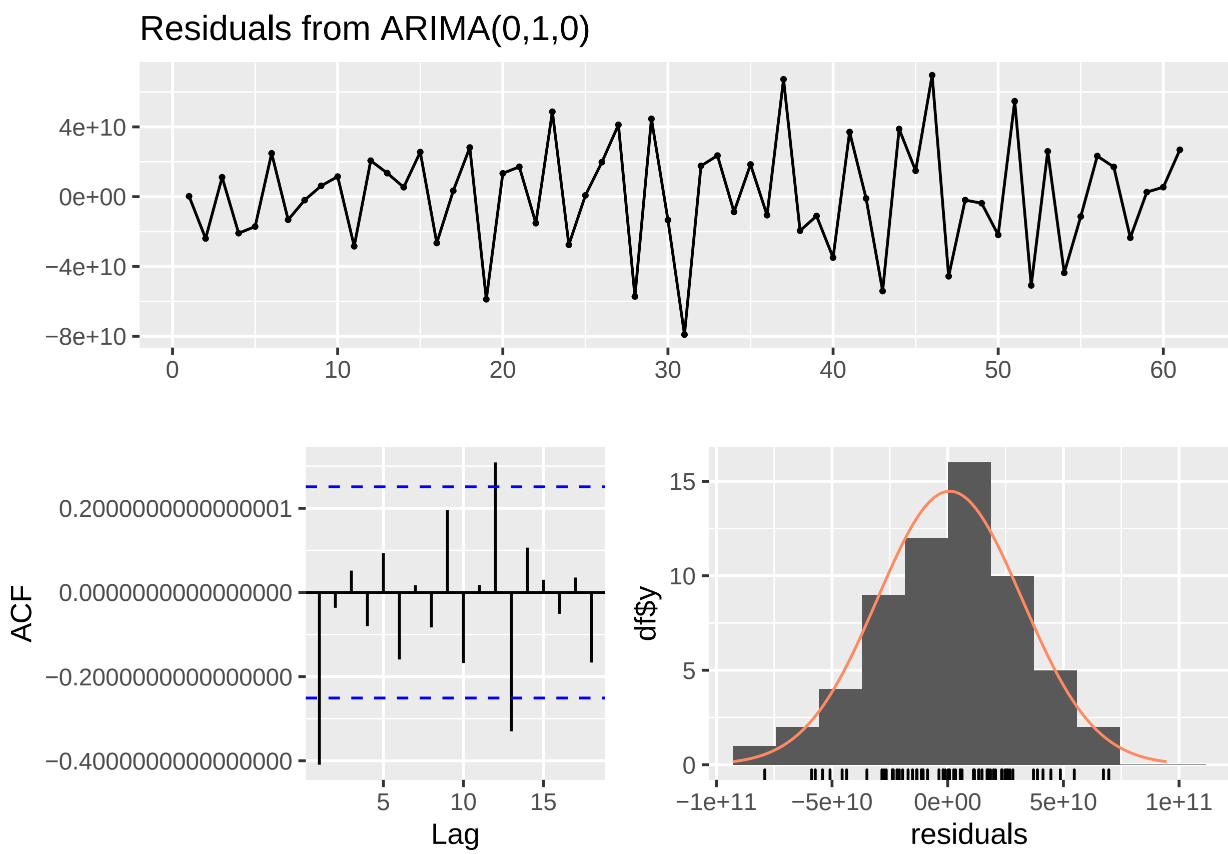



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0)
## Q* = 19.267429590833, df = 10, p-value = 0.03699449120552
##
## Model df: 0.   Total lags used: 10
```

```
accuracy(model)
```

```
##                               ME              RMSE               MAE
## Training set 867864454.9538559 31002478050.77285 24683530336.82609
##                                MPE             MAPE               MASE
## Training set -0.5600536775653298 10.96753355952061 0.9837496279118348
##                               ACF1
## Training set -0.4093512373791702
```

```r
checkresiduals(model_arima)
```



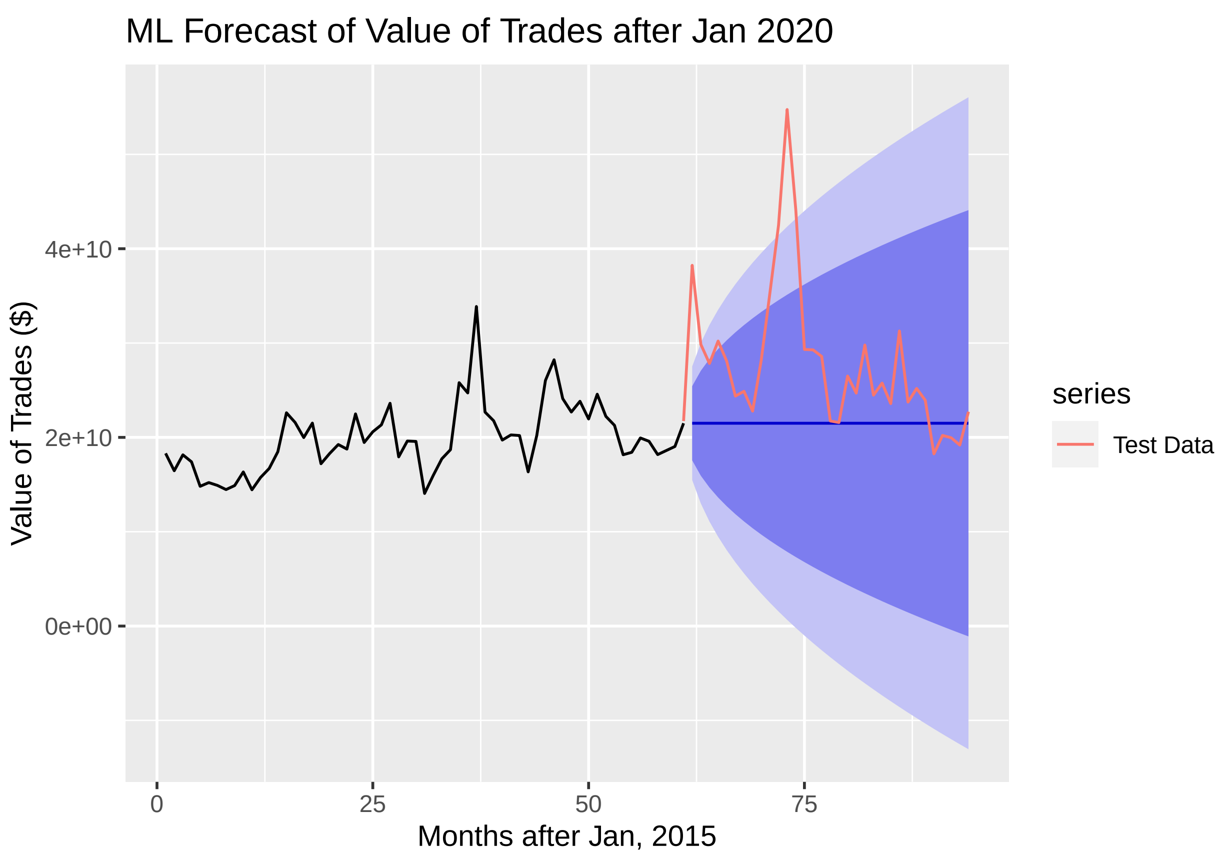

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0)
## Q* = 19.267429590833, df = 10, p-value = 0.03699449120552
##
## Model df: 0.   Total lags used: 10
```

```r
accuracy(model_arima)
```

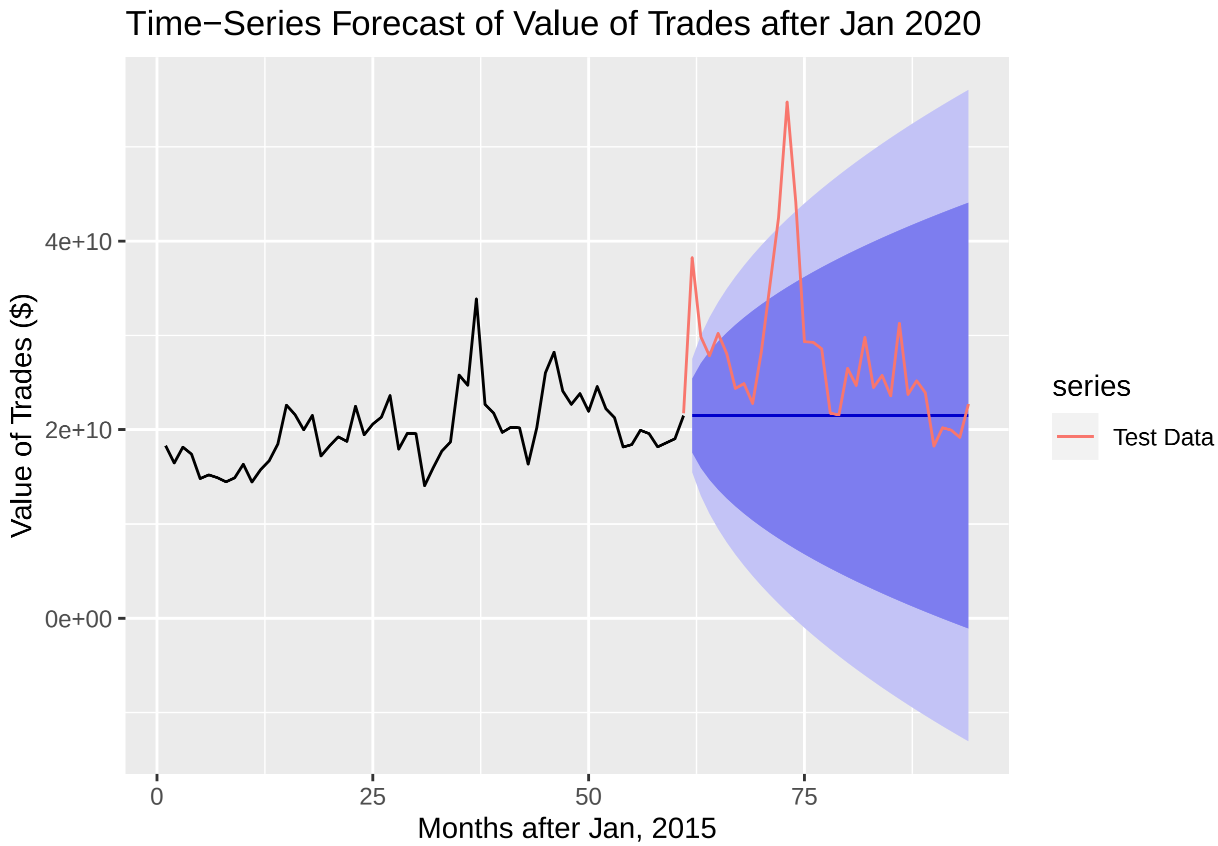
```
##                              ME              RMSE                MAE
## Training set 867864454.9538559 31002478050.77285 24683530336.82609
##                               MPE              MAPE                MASE
## Training set -0.5600536775653298 10.96753355952061 0.9837496279118348
##                              ACF1
## Training set -0.4093512373791702
```

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

# Splitting the data
```

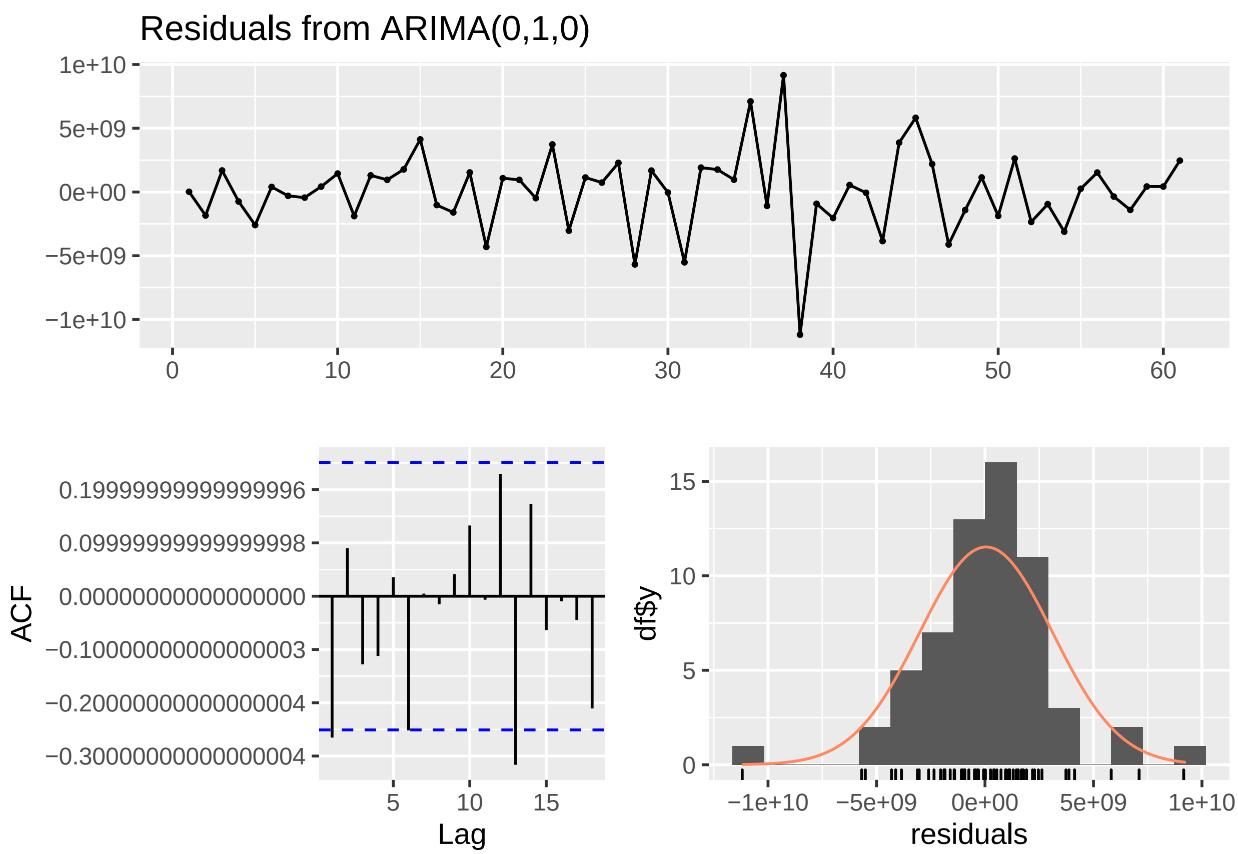
```
train <- ts_val_trades[1:61]
test <- ts_val_trades[62:95]

# 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 Val
```




```
# 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 Foreca
  autolayer(ts(test, start= length(train)), series="Test Data")
```




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

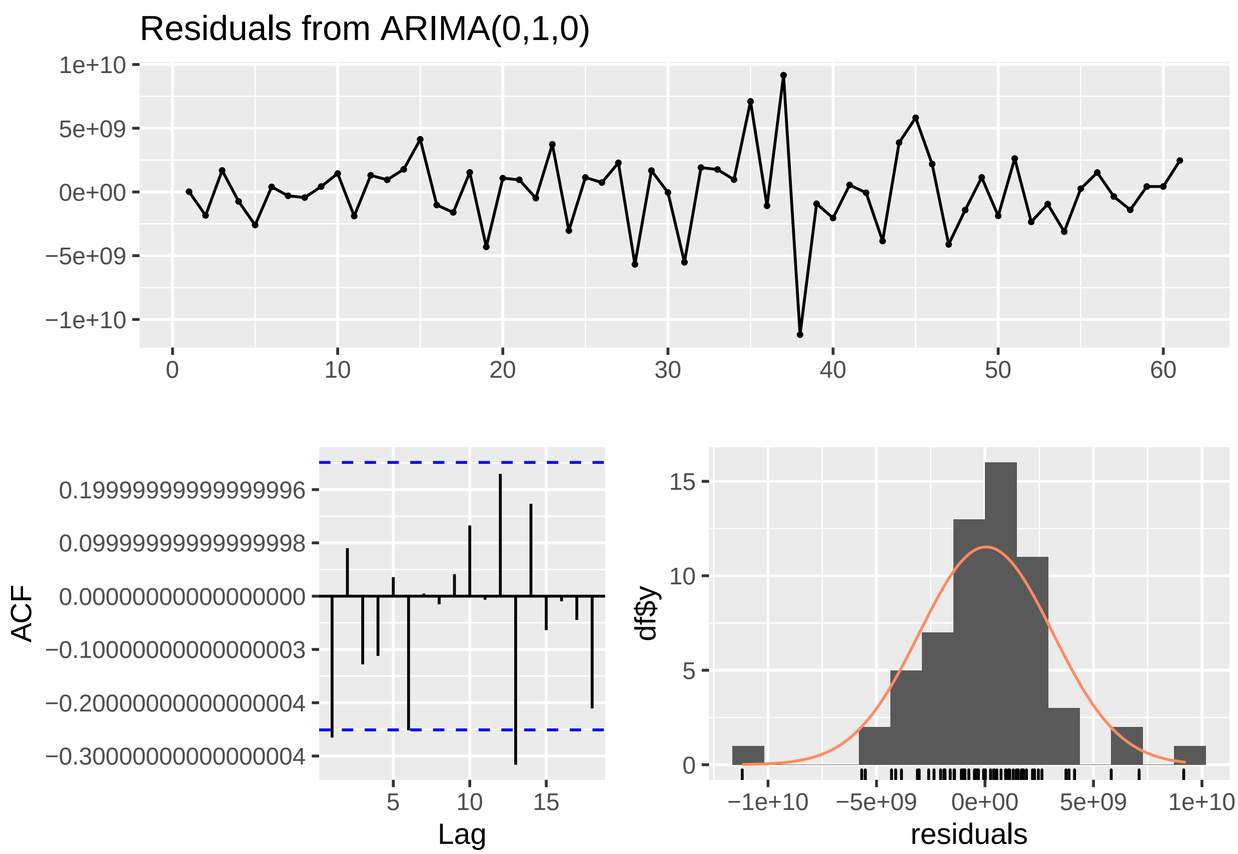



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0)
## Q* = 12.9604352059, df = 10, p-value = 0.2258923512494
##
## Model df: 0.   Total lags used: 10
```

```
accuracy(model)
```

```
##                             ME              RMSE               MAE
## Training set 52611058.0073972 3043745321.697508 2159446458.040444
##                             MPE             MAPE                MASE
## Training set -0.6986630231759118 10.58298484869522 0.9837432807102547
##                             ACF1
## Training set -0.2653537021834715
```

```
checkresiduals(model_arima)
```



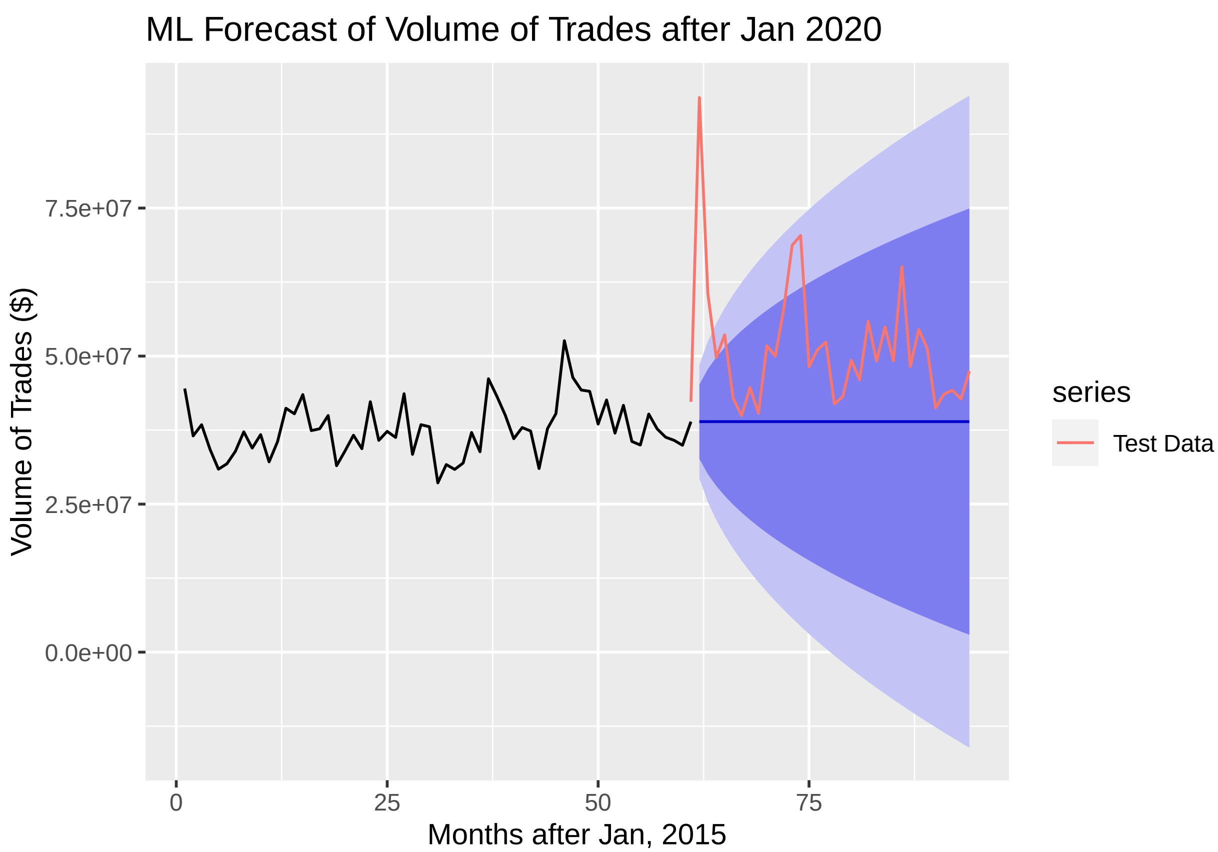

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0)
## Q* = 12.9604352059, df = 10, p-value = 0.2258923512494
##
## Model df: 0.   Total lags used: 10
```

```
accuracy(model_arima)
```

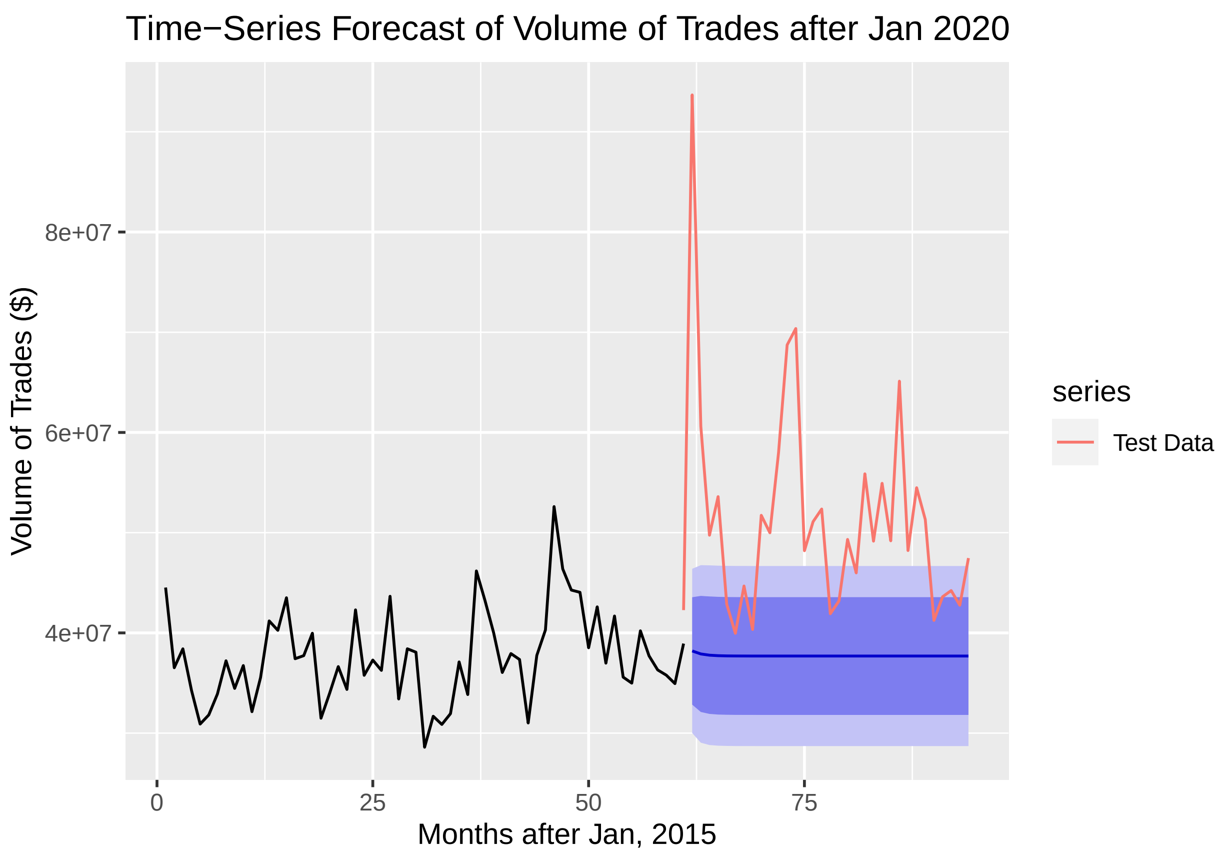
```
##                             ME              RMSE                 MAE
## Training set 52611058.0073972 3043745321.697508 2159446458.040444
##                            MPE              MAPE               MASE
## Training set -0.6986630231759118 10.58298484869522 0.9837432807102547
##                           ACF1
## Training set -0.2653537021834715
```

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

# Splitting the data
```

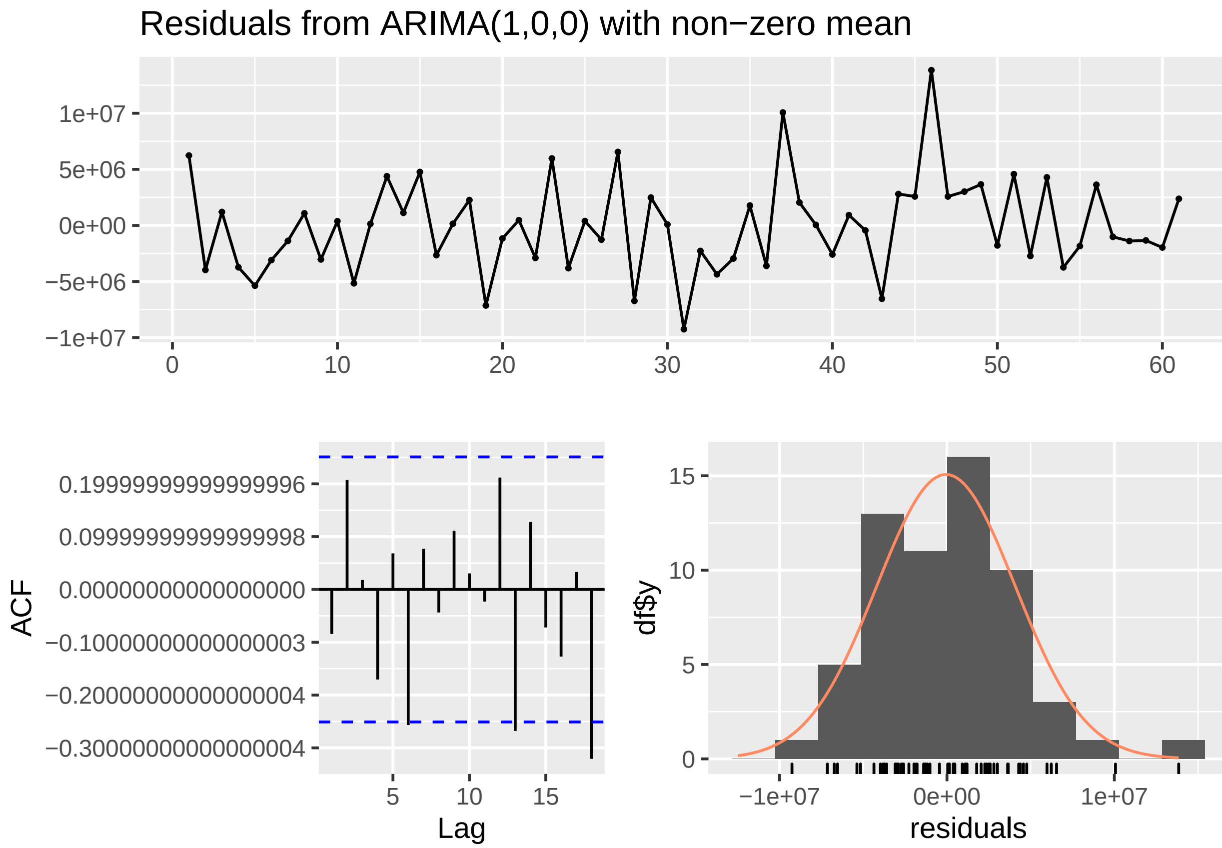
```r
train <- ts_vol_trades[1:61]
test <- ts_vol_trades[62:95]


# 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 Vo
```




```r
# 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 Fore
  autolayer(ts(test, start= length(train)), series="Test Data")
```




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

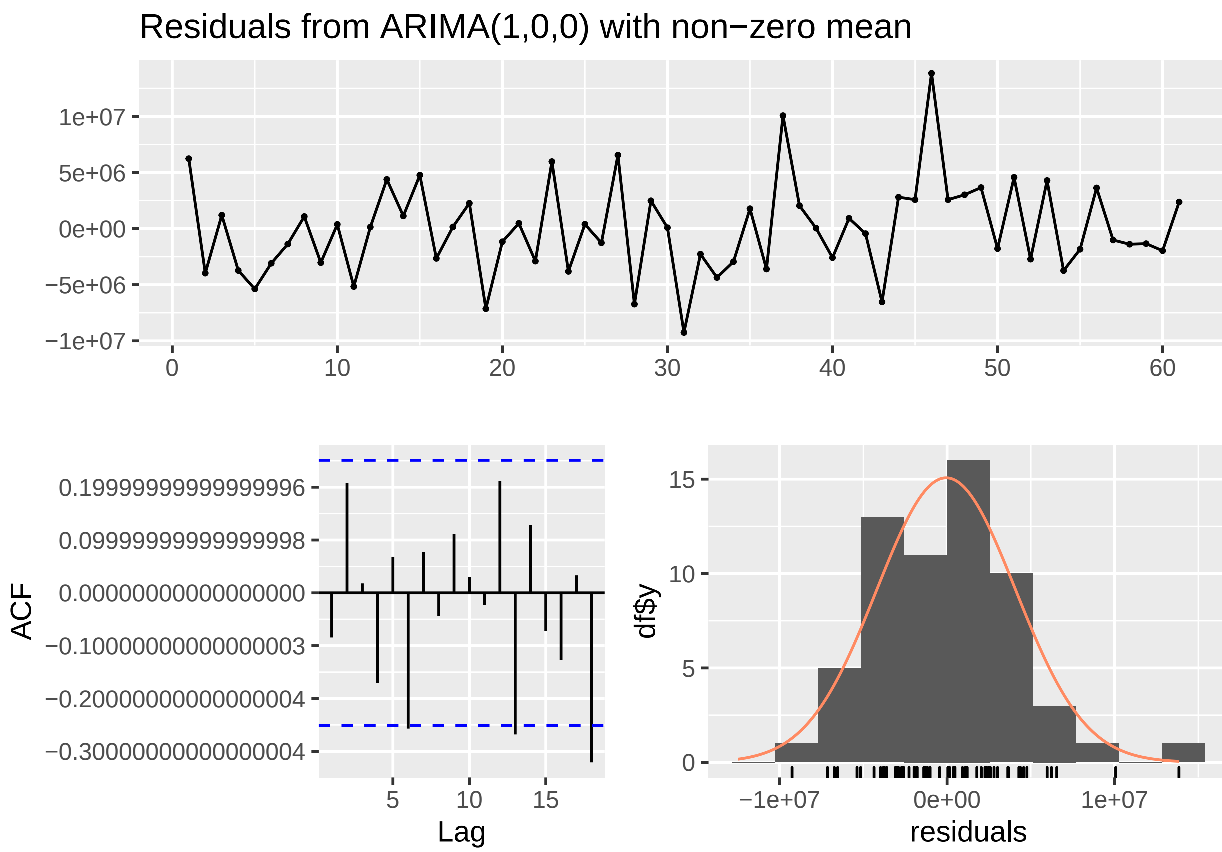



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 11.725788584091, df = 9, p-value = 0.2292131508306
##
## Model df: 1.   Total lags used: 10
```

```
accuracy(model)
```

```
##                                ME              RMSE               MAE
## Training set -55892.33645907887 4110967.654912055 3198088.353583062
##                               MPE              MAPE              MASE
## Training set -1.306366768030053 8.545188522488607 0.8110082461215385
##                                ACF1
## Training set -0.08442026667577432
```

```
checkresiduals(model_arima)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 11.725788584091, df = 9, p-value = 0.2292131508306
##
## Model df: 1.   Total lags used: 10
```

```
accuracy(model_arima)
```

```
##                               ME              RMSE               MAE
## Training set -55892.33645907887 4110967.654912055 3198088.353583062
##                              MPE              MAPE              MASE
## Training set -1.306366768030053 8.545188522488607 0.8110082461215385
##                               ACF1
## Training set -0.08442026667577432
```