TSX

Tan Lin

2023-04-20

```
#Installing Packages and Loading Libraries
install.packages("ggplot2", repos="https://mirror.csclub.uwaterloo.ca/CRAN/")
```

```
## Installing package into 'C:/Users/autot/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\autot\AppData\Local\Temp\RtmpmW4HRD\downloaded_packages
```

```
install.packages("foreacst", repos="https://mirror.csclub.uwaterloo.ca/CRAN/")
```

```
## Installing package into 'C:/Users/autot/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## Warning: package 'foreacst' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.2.3
```

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

Load Data

```
TSX = read.csv("TSX_Data.csv")
```

Data Preparation

Data Types

```
#Converting Text into Dates and Numbers
TSX[,"Date"] = as.Date(TSX[,"Date"])
TSX[,2:7] = lapply(TSX[,2:7],as.numeric)

#Check If Converted
class(TSX[1,1])
```

```
## [1] "Date"
```

```
class(TSX[1,2])
```

```
## [1] "numeric"
```

```
class(TSX[1,3])
```

```
## [1] "numeric"
```

```
class(TSX[1,4])
```

```
## [1] "numeric"
```

```
class(TSX[1,5])
```

```
## [1] "numeric"
```

```
class(TSX[1,6])
```

```
## [1] "numeric"
```

```
class(TSX[1,7])
```

```
## [1] "numeric"
```

NA values

```
# Check for NA Values
apply(sapply(TSX,is.na),2,sum)
```

```
##      Date      Open      High       Low     Close Adj.Close    Volume
##         0       250       250       250       250       250       250
```

```r
# Removing records with NA
TSX = na.omit(TSX)
```

Train and Test Data

```r
# Only Use 20 Years worth of Data
TSX = TSX[TSX[,"Date"] > "2003-01-01",]

# Selecting Dates before 2023 as Training Data
TSX_train = TSX[TSX[,"Date"] < "2023-01-01",]

# Selecting the Dates after 2023 as Testing Data
TSX_test = TSX[TSX[,"Date"] >= "2023-01-01",]
```

Predicting Closing Values

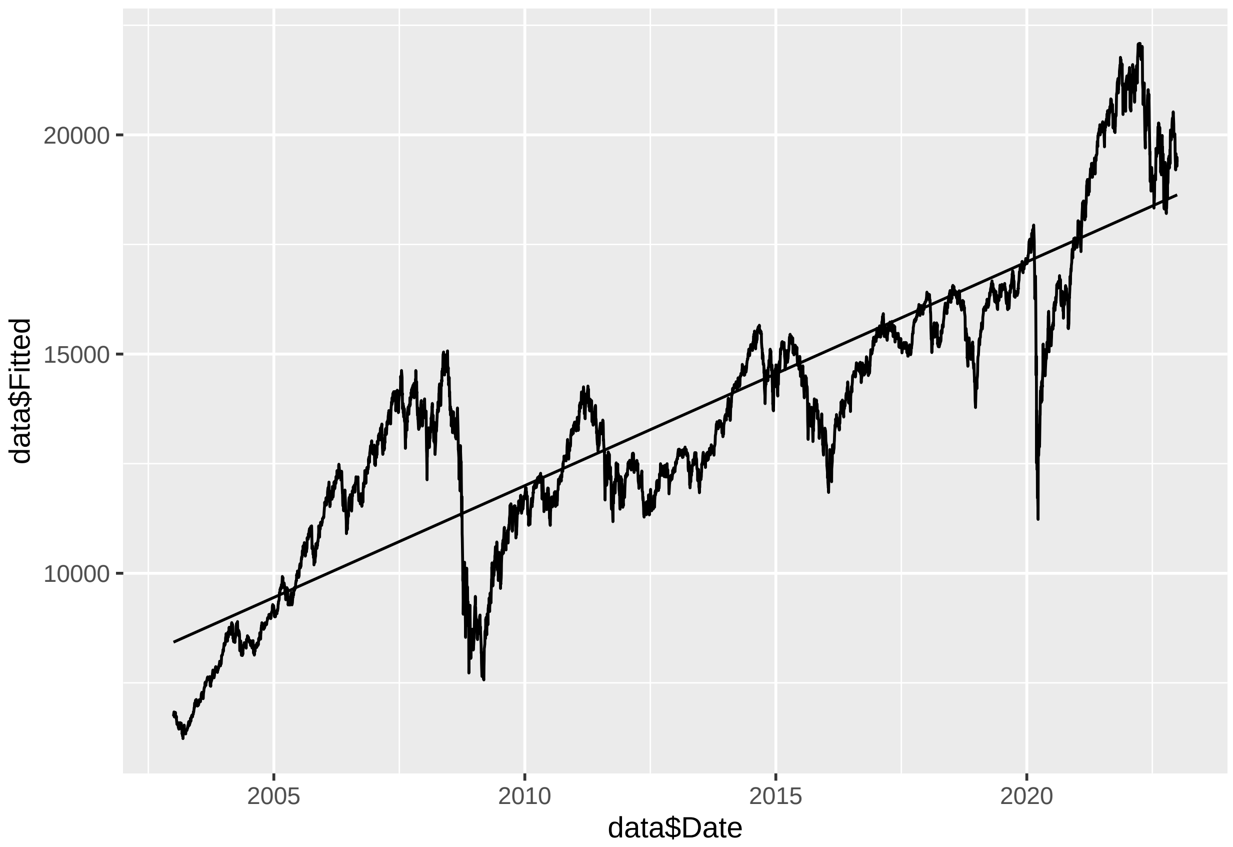
Model 1: Simple Linear Regression

```r
# Creating the Model
model_1 = lm(Close ~ Date, TSX_train)

# Evaluating Model Performance
SSE_1 = sum((fitted(model_1) - TSX_train[,"Close"])^2)
# Used Log of SSE for readability
log(SSE_1,10)
```

```
## [1] 10.07168
```

```r
#Visualizing Results
data = data.frame(TSX_train[,"Date"],TSX_train[,"Close"], fitted(model_1))
colnames(data) = c("Date","Actual", "Fitted")
ggplot(data) + geom_line(aes(x=data$Date, y=data$Fitted)) + geom_line(aes(x=data$Date, y=data$Actual))
```



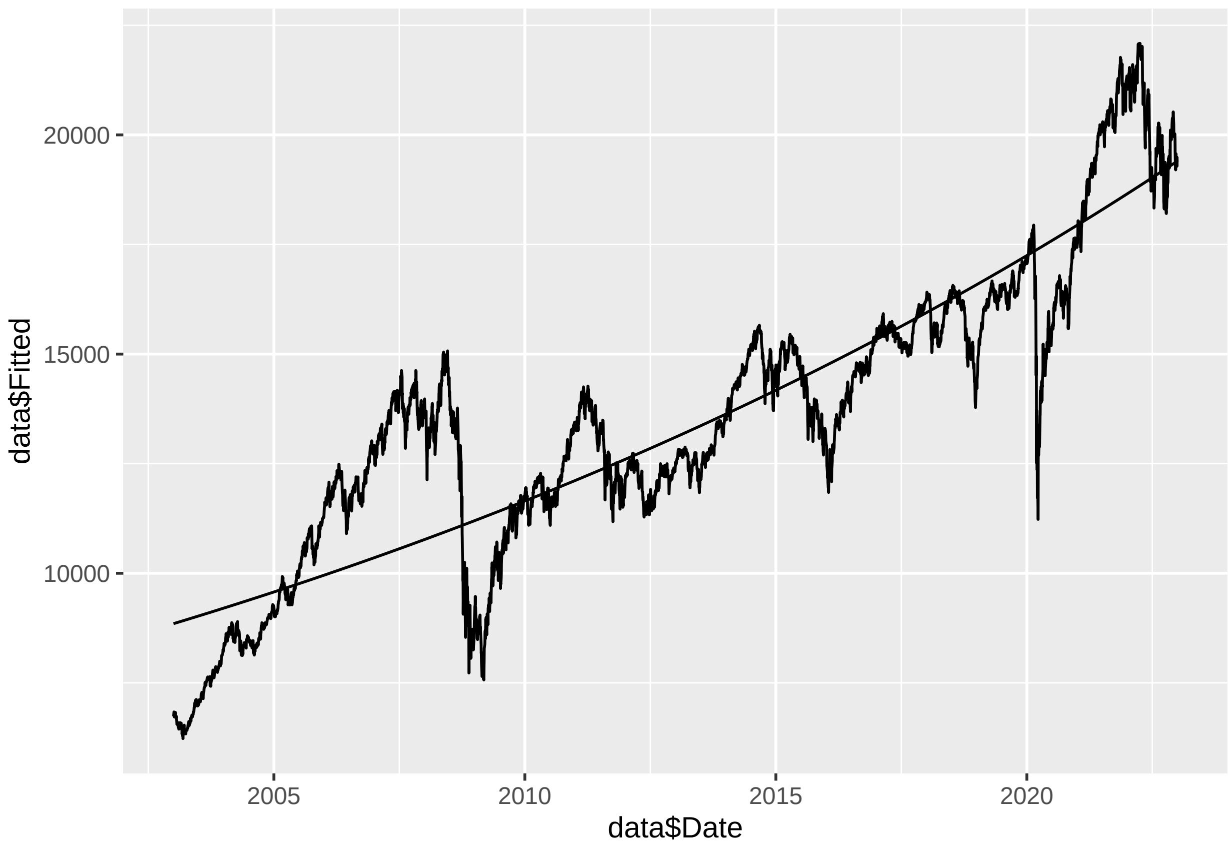
## Model 2: Logarithmic Regression

```
# Creating the Model
model_2 = lm(log(Close) ~ Date, TSX_train)

# Evaluating Model Performance
SSE_2 = sum((TSX_train[,"Close"] - exp(fitted(model_2)))^2)
log(SSE_2,10)
```

```
## [1] 10.05307
```

```
# Visualizing Results
data = data.frame(TSX_train[,"Date"],TSX_train[,"Close"],exp(fitted(model_2)))
colnames(data) = c("Date","Actual", "Fitted")
ggplot(data) + geom_line(aes(x=data$Date, y=data$Fitted)) + geom_line(aes(x=data$Date, y=data$Actual))
```



Model 3: Moving Average

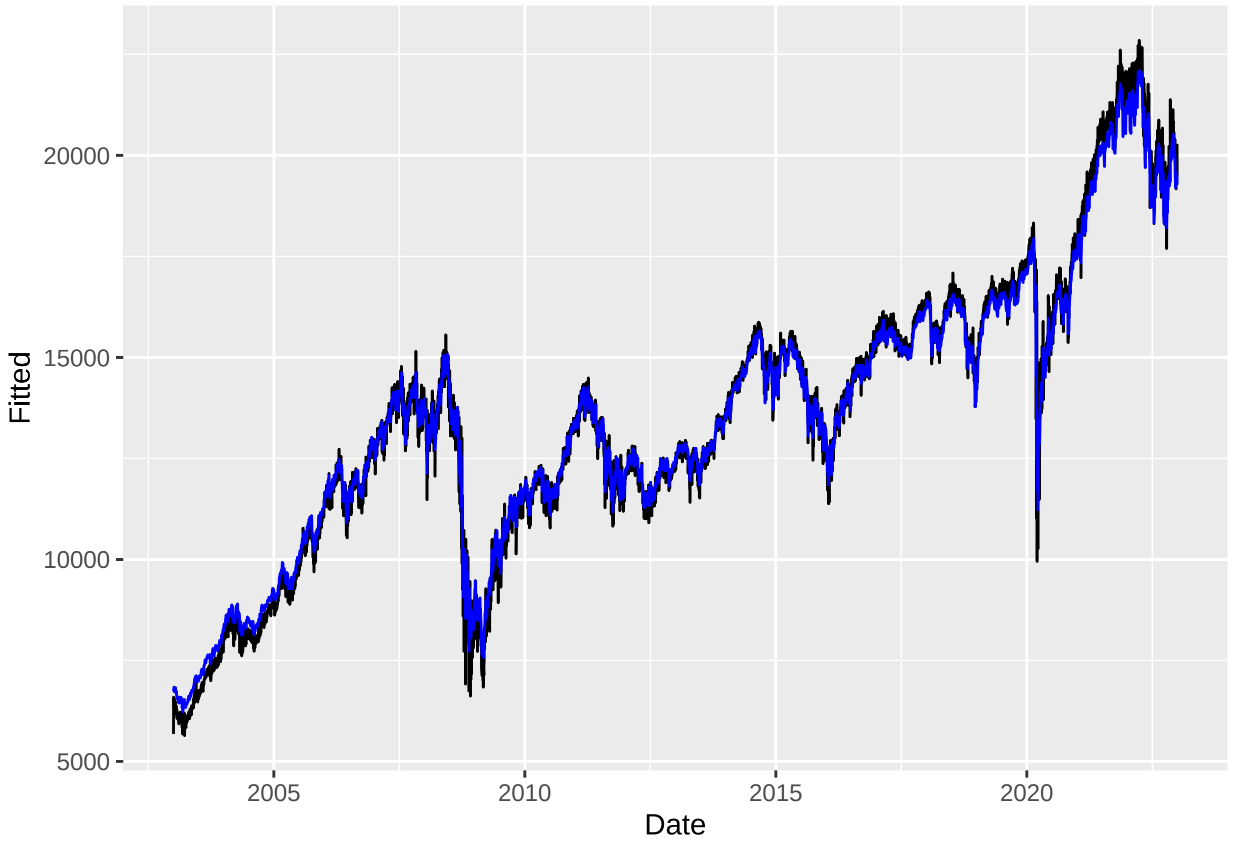
```r
# Converting Close into a Time Series
closing = as.ts(TSX_train[,"Close"])

# Creating the Model
model_3 = arima(closing, order = c(0,0,7))
# Evaluating Model Performance
SSE_3 = sum(model_3$residuals^2)
# Used Log of SSE for readability
log(SSE_3,10)
```

```
## [1] 8.644308
```

```r
#Visualizing Results
data = data.frame(TSX_train[,"Date"],TSX_train[,"Close"],TSX_train[,"Close"] + model_3$residuals)
colnames(data) = c("Date","Actual", "Fitted")
ggplot(data) + geom_line(aes(x=Date, y=Fitted)) + geom_line(aes(x=Date, y=Actual),color='blue')
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```



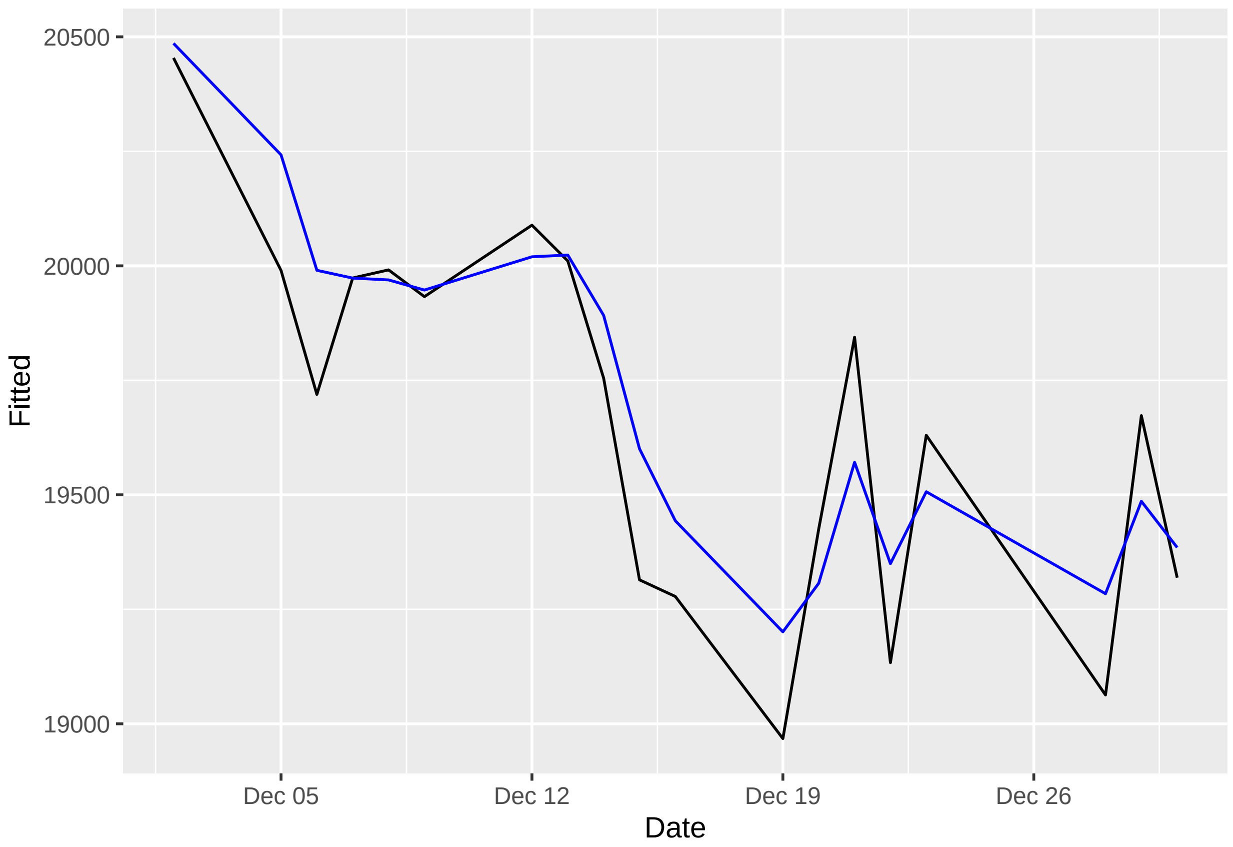
Model 4: ARMA

```
# Converting Close into a Time Series
closing = as.ts(TSX_train[,"Close"])

# Creating the Model
model_4 = arima(closing, order = c(3,0,7))
# Evaluating Model Performance
SSE_4 = sum(model_4$residuals^2)
# Used Log of SSE for readability
log(SSE_4,10)
```

```
## [1] 7.963171
```

```
#Visualizing Results
data = data.frame(TSX_train[,"Date"],TSX_train[,"Close"],TSX_train[,"Close"] + model_4$residuals)
colnames(data) = c("Date","Actual", "Fitted")
ggplot(data[data$Date > "2022-12-01",]) + geom_line(aes(x=Date, y=Fitted)) + geom_line(aes(x=Date, y=Ac
```


20500
20000
19500
19000
Fitted
Dec 05
Dec 12
Dec 19
Dec 26
Date