ADAPT Program Day 5 - Time Series

# Loading packages

library(dplyr)

##   
## 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

library(tidyr)  
library(ggplot2)  
library(data.table) #This is to use fread instead of read.csv (faster for big files)

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

library(InformationValue)  
library(tseries)  
library(forecast)

# ARIMA Time Series Modelling

## Loading files

df\_Data <- fread("./data/ModelData\_Quarterly.csv", stringsAsFactors = F)  
  
summary(df\_Data)

## Year Quarter Year and Quarter Throughput   
## Min. :2000 Length:76 Length:76 Min. : 57.0   
## 1st Qu.:2004 Class :character Class :character 1st Qu.: 92.8   
## Median :2009 Mode :character Mode :character Median :129.2   
## Mean :2009 Mean :128.5   
## 3rd Qu.:2014 3rd Qu.:164.9   
## Max. :2018 Max. :196.4   
## QoQ Growth YoY Growth V7 V8   
## Length:76 Length:76 Mode:logical Mode:logical   
## Class :character Class :character NA's:76 NA's:76   
## Mode :character Mode :character   
##   
##   
##   
## V9   
## Mode:logical   
## NA's:76   
##   
##   
##   
##

head(df\_Data)

## Year Quarter Year and Quarter Throughput QoQ Growth YoY Growth V7 V8 V9  
## 1: 2000 Q1 2000-Q1 57.0 NA NA NA  
## 2: 2000 Q2 2000-Q2 61.3 7.54% NA NA NA  
## 3: 2000 Q3 2000-Q3 58.5 -4.57% NA NA NA  
## 4: 2000 Q4 2000-Q4 57.8 -1.20% NA NA NA  
## 5: 2001 Q1 2001-Q1 60.9 5.36% 6.84% NA NA NA  
## 6: 2001 Q2 2001-Q2 64.3 5.58% 4.89% NA NA NA

## Cleaning and Transformations

#Tidying columns  
df\_Data <- df\_Data[,1:6]  
names(df\_Data)[c(3,5:6)] <- c("Year\_and\_Quarter","QoQ\_Growth","YoY\_Growth")  
  
#Converting into factor  
df\_Data$Year <- ordered(df\_Data$Year)  
df\_Data$Quarter <- ordered(df\_Data$Quarter)  
  
#Removing "%" and converting into numeric  
df\_Data$QoQ\_Growth <- as.numeric(sub("%","",df\_Data$QoQ\_Growth))/100  
df\_Data$YoY\_Growth <- as.numeric(sub("%","",df\_Data$YoY\_Growth))/100  
  
summary(df\_Data)

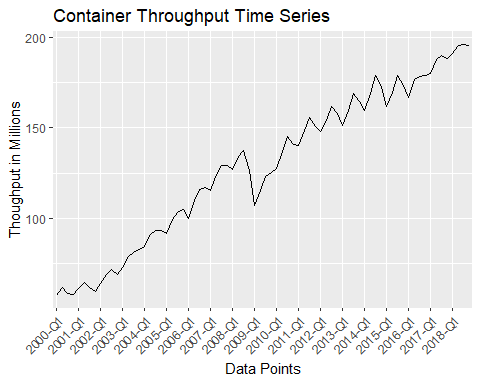
## Year Quarter Year\_and\_Quarter Throughput   
## 2000 : 4 Q1:19 Length:76 Min. : 57.0   
## 2001 : 4 Q2:19 Class :character 1st Qu.: 92.8   
## 2002 : 4 Q3:19 Mode :character Median :129.2   
## 2003 : 4 Q4:19 Mean :128.5   
## 2004 : 4 3rd Qu.:164.9   
## 2005 : 4 Max. :196.4   
## (Other):52   
## QoQ\_Growth YoY\_Growth   
## Min. :-0.15190 Min. :-0.15460   
## 1st Qu.:-0.01800 1st Qu.: 0.03918   
## Median : 0.02050 Median : 0.06410   
## Mean : 0.01767 Mean : 0.07116   
## 3rd Qu.: 0.05590 3rd Qu.: 0.11753   
## Max. : 0.10030 Max. : 0.20700   
## NA's :1 NA's :4

head(df\_Data)

## Year Quarter Year\_and\_Quarter Throughput QoQ\_Growth YoY\_Growth  
## 1: 2000 Q1 2000-Q1 57.0 NA NA  
## 2: 2000 Q2 2000-Q2 61.3 0.0754 NA  
## 3: 2000 Q3 2000-Q3 58.5 -0.0457 NA  
## 4: 2000 Q4 2000-Q4 57.8 -0.0120 NA  
## 5: 2001 Q1 2001-Q1 60.9 0.0536 0.0684  
## 6: 2001 Q2 2001-Q2 64.3 0.0558 0.0489

## Plotting

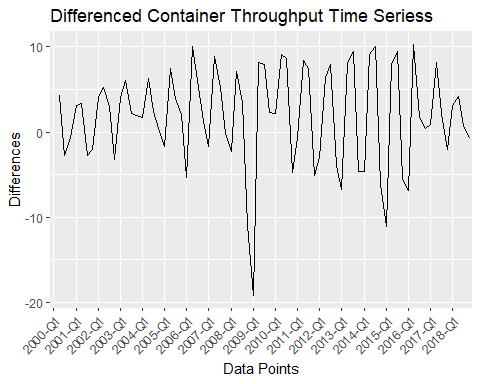
ticks <- df\_Data$Year\_and\_Quarter  
for (i in 1:length(ticks)) {  
 if(i %% 4 != 1) {ticks[i] <- NA}  
}  
  
g <- ggplot(df\_Data,aes(Year\_and\_Quarter))  
g + geom\_line(aes(y=Throughput, group=1)) +  
 labs(title="Container Throughput Time Series",  
 x="Data Points",  
 y="Thoughput in Millions") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_x\_discrete(breaks=ticks)



## Checking stationary of Mean & Variance

df\_Data$Diff <- c(NA,diff(df\_Data$Throughput))  
  
  
g <- ggplot(df\_Data,aes(Year\_and\_Quarter))  
g + geom\_line(aes(y=Diff, group=1)) +  
 labs(title="Differenced Container Throughput Time Seriess",  
 x="Data Points",  
 y="Differences") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_x\_discrete(breaks=ticks)

## Warning: Removed 1 rows containing missing values (geom\_path).



## Auto ARIMA Model

fit\_model <- auto.arima(df\_Data$Throughput, approximation = F, trace = T, seasonal = T)

##   
## ARIMA(2,1,2) with drift : 421.4401  
## ARIMA(0,1,0) with drift : 480.3688  
## ARIMA(1,1,0) with drift : 482.3755  
## ARIMA(0,1,1) with drift : 477.1023  
## ARIMA(0,1,0) : 485.538  
## ARIMA(1,1,2) with drift : Inf  
## ARIMA(2,1,1) with drift : 440.2481  
## ARIMA(3,1,2) with drift : 423.7  
## ARIMA(2,1,3) with drift : 423.8664  
## ARIMA(1,1,1) with drift : 476.5008  
## ARIMA(1,1,3) with drift : 447.0265  
## ARIMA(3,1,1) with drift : 440.879  
## ARIMA(3,1,3) with drift : 426.3383  
## ARIMA(2,1,2) : 437.4177  
##   
## Best model: ARIMA(2,1,2) with drift

summary(fit\_model)

## Series: df\_Data$Throughput   
## ARIMA(2,1,2) with drift   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 drift  
## -0.0274 -0.9434 0.2027 0.6288 1.8846  
## s.e. 0.0453 0.0374 0.1380 0.0921 0.3901  
##   
## sigma^2 estimated as 14.08: log likelihood=-204.1  
## AIC=420.2 AICc=421.44 BIC=434.11  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.0008456349 3.601286 2.560877 -0.06828713 2.187546 0.511085  
## ACF1  
## Training set -0.006042813

str(fit\_model)

## List of 19  
## $ coef : Named num [1:5] -0.0274 -0.9434 0.2027 0.6288 1.8846  
## ..- attr(\*, "names")= chr [1:5] "ar1" "ar2" "ma1" "ma2" ...  
## $ sigma2 : num 14.1  
## $ var.coef : num [1:5, 1:5] 0.002049 -0.000176 -0.00385 -0.000651 -0.000213 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:5] "ar1" "ar2" "ma1" "ma2" ...  
## .. ..$ : chr [1:5] "ar1" "ar2" "ma1" "ma2" ...  
## $ mask : logi [1:5] TRUE TRUE TRUE TRUE TRUE  
## $ loglik : num -204  
## $ aic : num 420  
## $ arma : int [1:7] 2 2 0 0 1 1 0  
## $ residuals: Time-Series [1:76] from 1 to 76: 0.0551 1.6305 -3.2646 -0.4682 -1.6226 ...  
## $ call : language auto.arima(y = df\_Data$Throughput, seasonal = T, trace = T, approximation = F, x = list(x = c(57, 61.3, 58.5| \_\_truncated\_\_ ...  
## $ series : chr "df\_Data$Throughput"  
## $ code : int 0  
## $ n.cond : int 0  
## $ nobs : int 75  
## $ model :List of 10  
## ..$ phi : num [1:2] -0.0274 -0.9434  
## ..$ theta: num [1:2] 0.203 0.629  
## ..$ Delta: num 1  
## ..$ Z : num [1:4] 1 0 0 1  
## ..$ a : num [1:4] -2.685 0.931 1.029 55.058  
## ..$ P : num [1:4, 1:4] 0.00 0.00 0.00 1.30e-17 -2.78e-17 ...  
## ..$ T : num [1:4, 1:4] -0.0274 -0.9434 0 1 1 ...  
## ..$ V : num [1:4, 1:4] 1 0.203 0.629 0 0.203 ...  
## ..$ h : num 0  
## ..$ Pn : num [1:4, 1:4] 1.00 2.03e-01 6.29e-01 -2.38e-17 2.03e-01 ...  
## $ xreg : int [1:76, 1] 1 2 3 4 5 6 7 8 9 10 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : NULL  
## .. ..$ : chr "drift"  
## $ bic : num 434  
## $ aicc : num 421  
## $ x : Time-Series [1:76] from 1 to 76: 57 61.3 58.5 57.8 60.9 64.3 61.5 59.5 63.6 68.8 ...  
## $ fitted : Time-Series [1:76] from 1 to 76: 56.9 59.7 61.8 58.3 62.5 ...  
## - attr(\*, "class")= chr [1:3] "ARIMA" "forecast\_ARIMA" "Arima"

## Comparing Actual vs Prediction

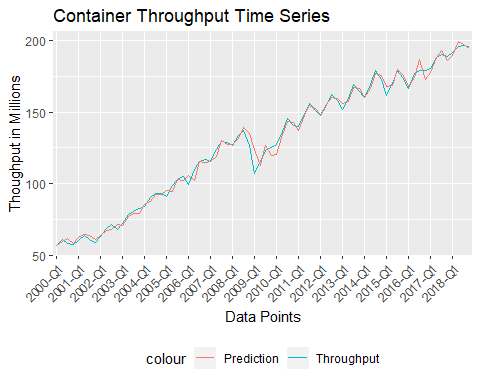
###Creating a new DF for the comparison

df\_Fit <- df\_Data[,3:4]  
df\_Fit <- cbind(df\_Fit,Prediction=as.numeric(fit\_model$fitted))  
df\_Fit$Error <- df\_Fit$Throughput - df\_Fit$Prediction  
  
head(df\_Fit)

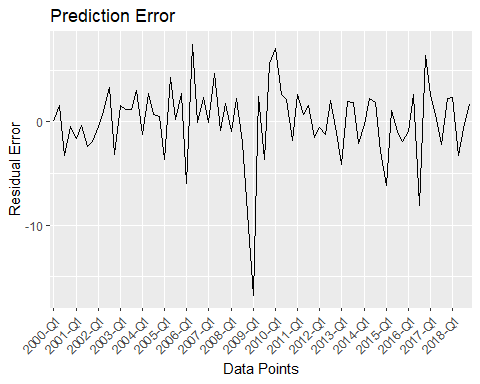
## Year\_and\_Quarter Throughput Prediction Error  
## 1: 2000-Q1 57.0 56.94488 0.05511537  
## 2: 2000-Q2 61.3 59.66953 1.63047321  
## 3: 2000-Q3 58.5 61.76456 -3.26455857  
## 4: 2000-Q4 57.8 58.26821 -0.46820816  
## 5: 2001-Q1 60.9 62.52261 -1.62261478  
## 6: 2001-Q2 64.3 64.61387 -0.31386574

### Some graphics

g1 <- ggplot(df\_Fit,aes(Year\_and\_Quarter)) +  
 geom\_line(aes(y=Throughput, group=1, col="Throughput")) +  
 geom\_line(aes(y=Prediction, group=1, col="Prediction")) +  
 labs(title="Container Throughput Time Series",  
 x="Data Points",  
 y="Thoughput in Millions") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1),  
 legend.position="bottom") +  
 scale\_x\_discrete(breaks=ticks)  
  
g2 <- ggplot(df\_Fit,aes(Year\_and\_Quarter)) +  
 geom\_line(aes(y=Error, group=1)) +  
 labs(title="Prediction Error",  
 x="Data Points",  
 y="Residual Error") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_x\_discrete(breaks=ticks)  
  
g1



g2



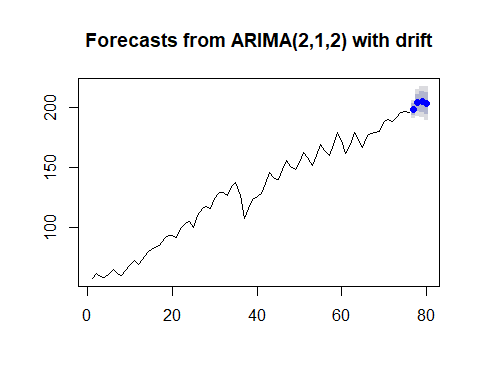
## Forecasting 4 next values

df\_Forecast <- data.frame(Year\_and\_Quarter=paste0("2019-Q",1:4),stringsAsFactors=F)  
df\_Forecast <- cbind(df\_Forecast,forecast(fit\_model, h=4))  
df\_Forecast[,3:4] <- NULL  
names(df\_Forecast)[2] <- "Forecast"  
  
df\_Forecast

## Year\_and\_Quarter Forecast Lo 95 Hi 95  
## 77 2019-Q1 198.4892 191.1345 205.8438  
## 78 2019-Q2 203.9078 192.5582 215.2573  
## 79 2019-Q3 204.7478 191.7695 217.7261  
## 80 2019-Q4 203.3269 189.3669 217.2868

## A quick plot

plot(forecast(fit\_model, h=4))



## Combining all the information in the same dataset

newRows <- cbind(df\_Forecast$Year\_and\_Quarter,NA,df\_Forecast$Forecast,NA)  
df\_Fit <- rbind(df\_Fit, newRows, use.names=F)  
  
tail(df\_Fit,8)

## Year\_and\_Quarter Throughput Prediction Error  
## 1: 2018-Q1 191.4 189.088106969408 2.3118930305925  
## 2: 2018-Q2 195.6 198.827535869469 -3.22753586946908  
## 3: 2018-Q3 196.4 197.07394887147 -0.673948871470458  
## 4: 2018-Q4 195.6 193.963776042507 1.63622395749275  
## 5: 2019-Q1 <NA> 198.489162391852 <NA>  
## 6: 2019-Q2 <NA> 203.907763407398 <NA>  
## 7: 2019-Q3 <NA> 204.747827327742 <NA>  
## 8: 2019-Q4 <NA> 203.326854085391 <NA>