Assignment 3

Hoyu (Ariel) Li

4/26/2020

Load data

```
library(readxl)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.6.2

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

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

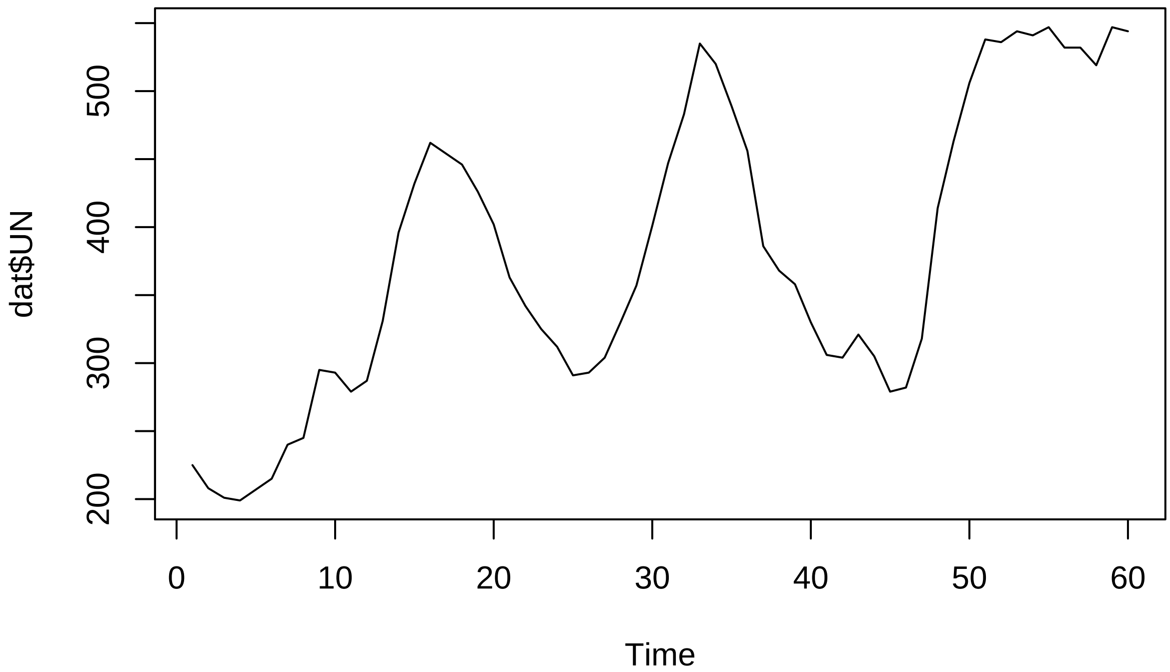
```
library(tseries)
dat <- read_excel('/Users/arielsmac/Desktop/Spring20/TimeSeries/Assignment3/Unemployment_GDP_UK.xlsx')
head(dat)
```

```
## # A tibble: 6 x 4
##    Year Quarter    UN   GDP
##   <dbl>   <dbl> <dbl> <dbl>
## 1  1955       1   225  81.4
## 2    NA       2   208  82.6
## 3    NA       3   201  82.3
## 4    NA       4   199  83
## 5  1956       1   207  82.9
## 6    NA       2   215  83.6
```

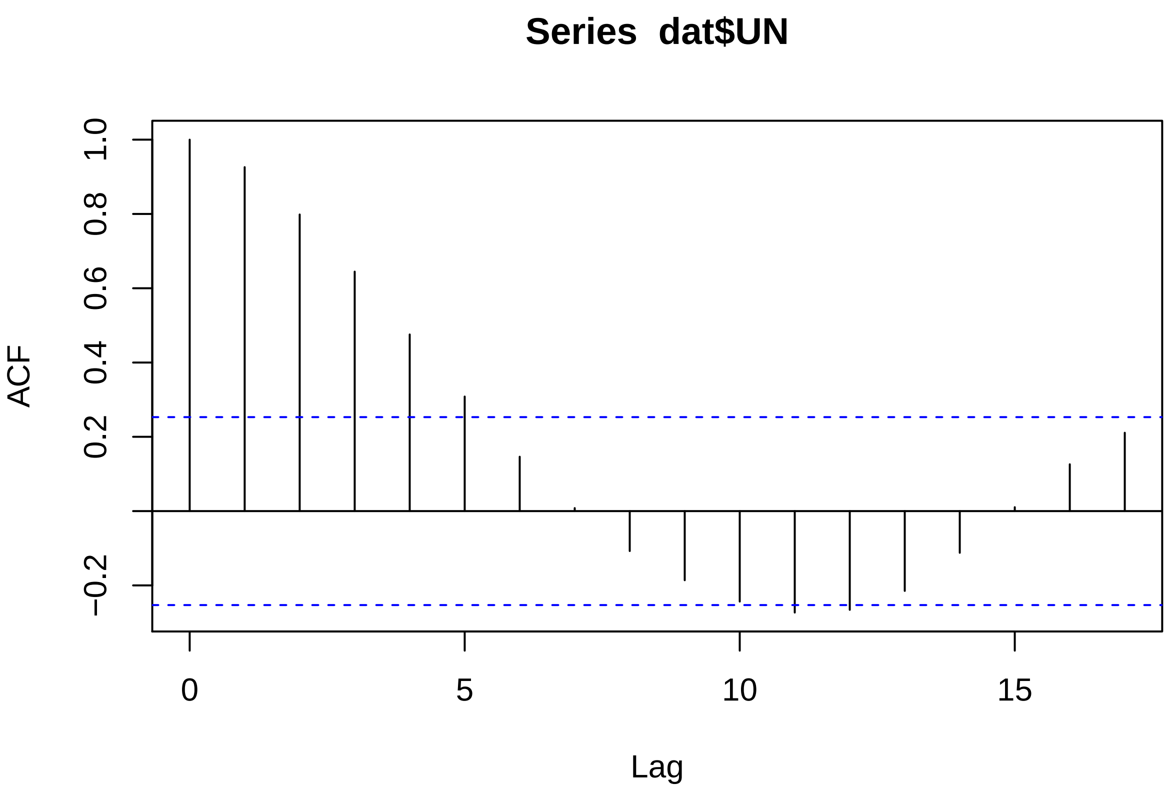
Let's start with data exploration to understand the data we're working with.

Determing Stationarity of UN

```
plot.ts(dat$UN)
```




```r
acf(dat$UN)
```




```r
adf.test(dat$UN)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dat$UN
```

```
## Dickey-Fuller = -3.1896, Lag order = 3, p-value = 0.09763
## alternative hypothesis: stationary
```

```r
kpss.test(dat$UN)
```

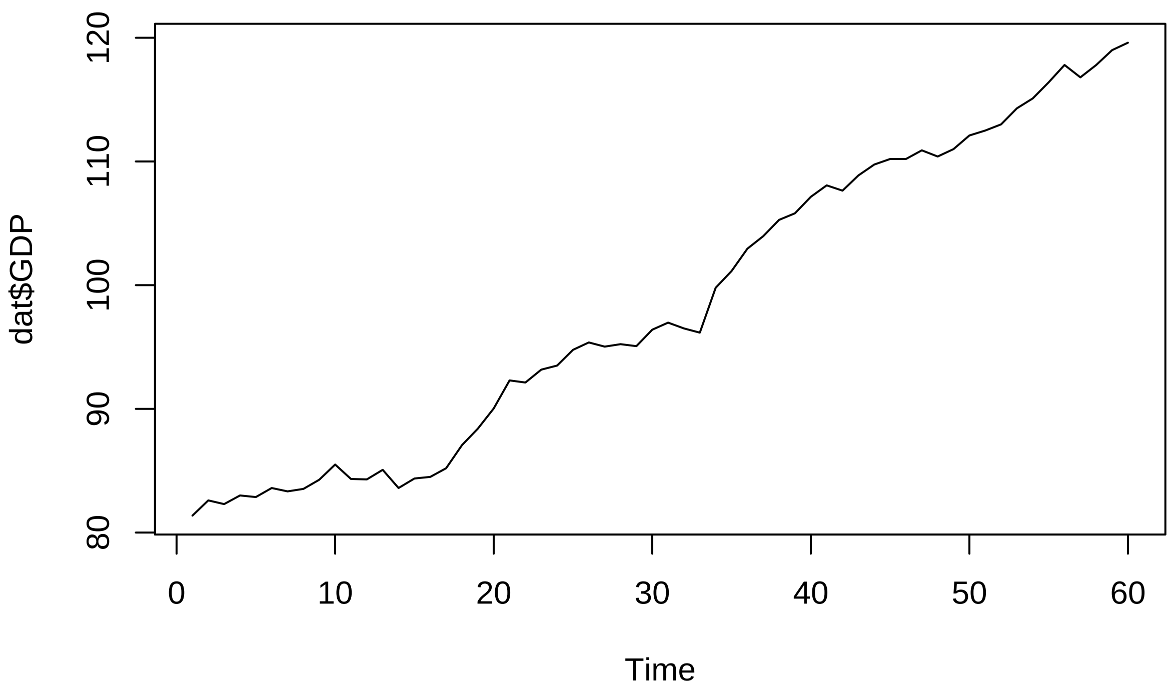
```
## Warning in kpss.test(dat$UN): p-value smaller than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  dat$UN
## KPSS Level = 0.77042, Truncation lag parameter = 3, p-value = 0.01
```

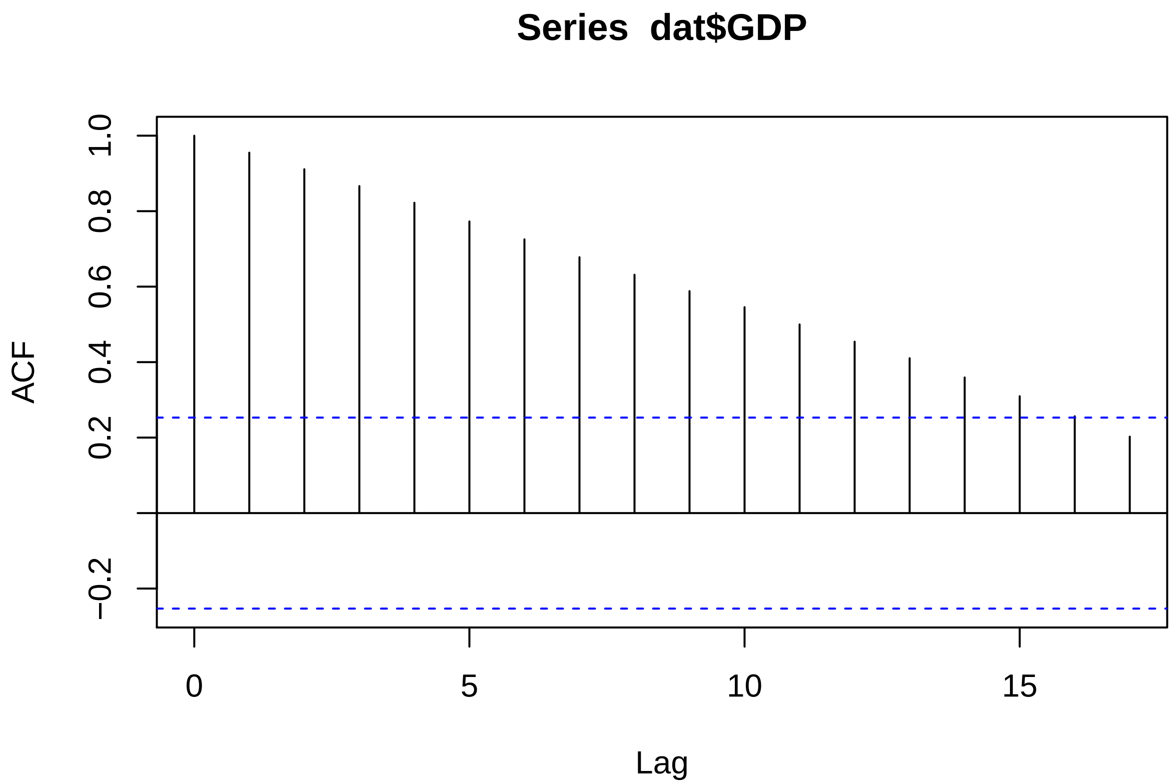
UN does not seem to be stationary in the mean. The ACF does not die down very quickly, which is typical of non-stationary data. The p-value from the ADF test is greater than 0.05, so we fail to reject the null hypothesis of non-stationarity. The p-value from the KPSS test is less than 0.05, thus we reject the null hypothesis of stationarity. The ADF and KPSS tests suggest that the UN series is non-stationary.

Determing Stationarity of GDP

```r
plot.ts(dat$GDP)
```




```r
acf(dat$GDP)
```




```r
adf.test(dat$UN)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dat$UN
## Dickey-Fuller = -3.1896, Lag order = 3, p-value = 0.09763
## alternative hypothesis: stationary
```

```r
kpss.test(dat$UN)
```

```
## Warning in kpss.test(dat$UN): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  dat$UN
## KPSS Level = 0.77042, Truncation lag parameter = 3, p-value = 0.01
```

GDP is not stationary in the mean. The ACF dies down rather slowly, which is typical of non-stationary data. The p-value from the ADF test is greater than 0.05, so we fail to reject the null hypothesis of non-stationarity. The p-value from the KPSS test is less than 0.05, thus we reject the null hypothesis of stationarity. The ADF and KPSS tests suggest that the GDP series is non-stationary.

ARIMA Modeling

Split into Train and Test Data

Splitting the dataset so that the train set contains data from 1955 to 1968 and the test set contains data for 1969.

```
# First, convert to time series data
df <- ts(dat,start = c(1955,1), frequency = 4)
# Split time series data into train and test sets
train <- window(df, end=c(1968,4))
test <- window(df, start=c(1969,1), end=c(1969,4))
head(train)
```

```
##         Year Quarter  UN   GDP
## 1955 Q1 1955       1 225 81.37
## 1955 Q2   NA       2 208 82.60
## 1955 Q3   NA       3 201 82.30
## 1955 Q4   NA       4 199 83.00
## 1956 Q1 1956       1 207 82.87
## 1956 Q2   NA       2 215 83.60
```

```
test
```

```
##         Year Quarter  UN   GDP
## 1969 Q1 1969       1 532 116.8
## 1969 Q2   NA       2 519 117.8
## 1969 Q3   NA       3 547 119.0
## 1969 Q4   NA       4 544 119.6
```

```
# Separate train and test sets of UN and GDP for ease of reference
UN.train <- train[,3]
UN.test <- test[,3]
GDP.train <- train[,4]
GDP.test <- test[,4]
```

UN ARIMA Model

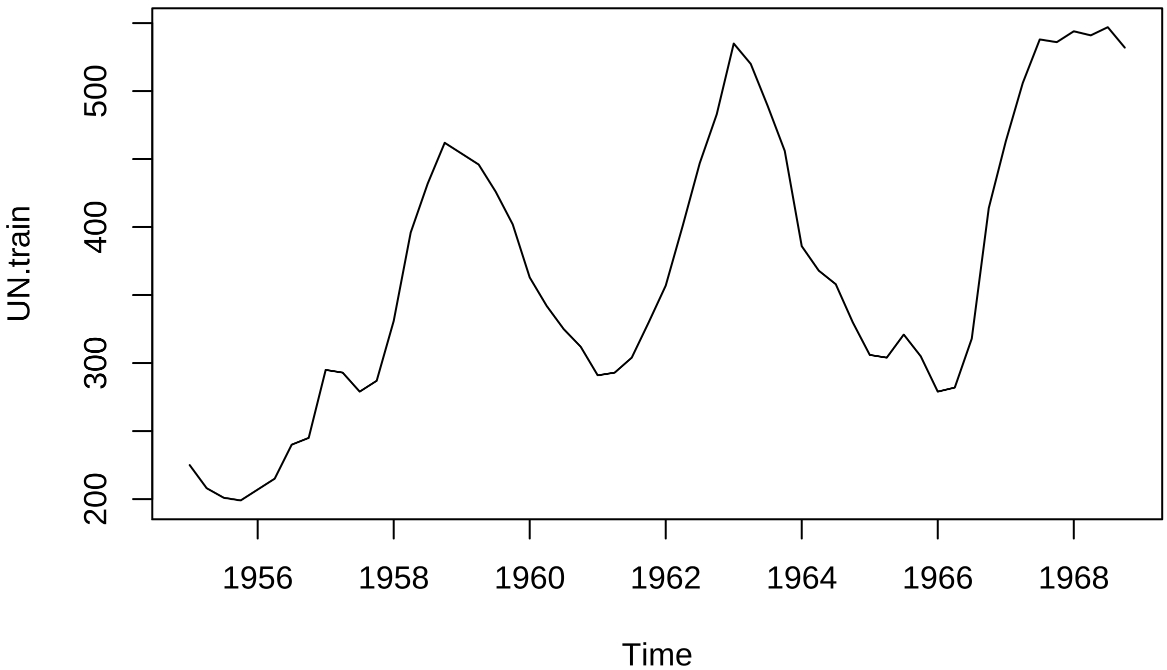
1) Use datasets from 1955 to 1968 to build an ARMA or ARIMA model for UN.

```
UN.arima <- auto.arima(UN.train)
summary(UN.arima)
```

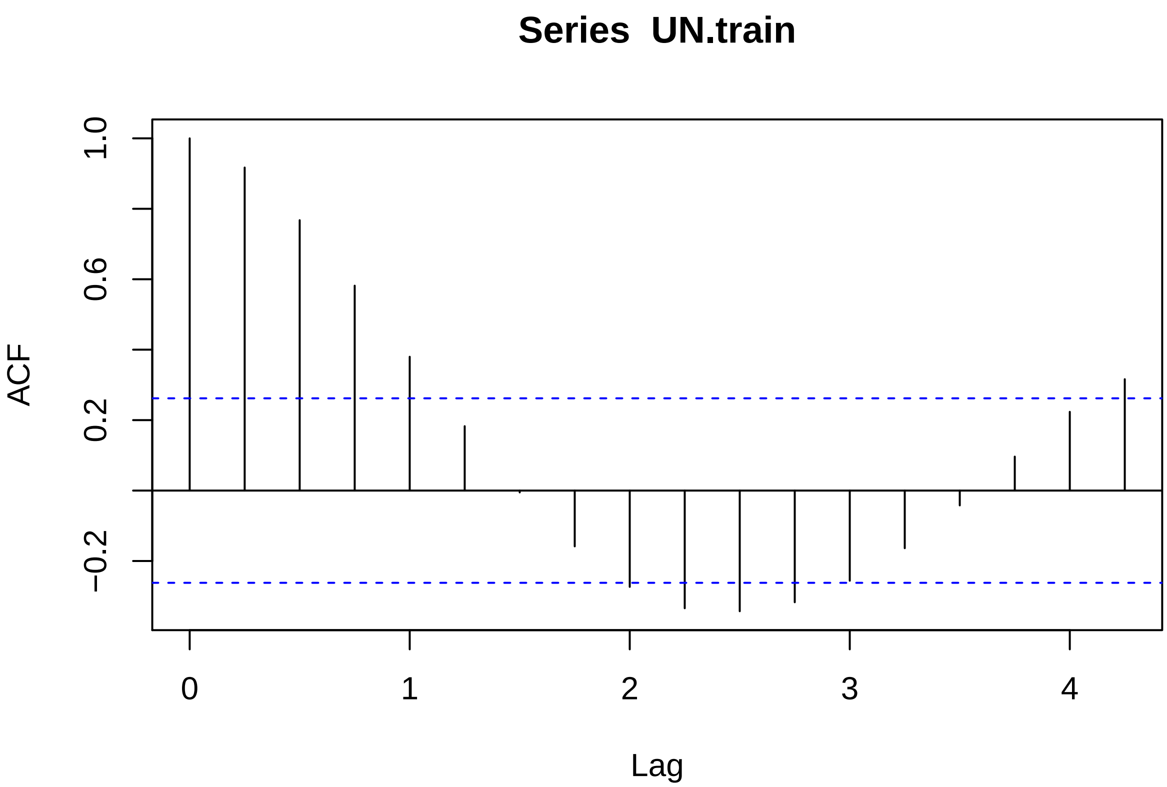
```
## Series: UN.train
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.6666
## s.e.  0.0977
##
## sigma^2 estimated as 525.1:  log likelihood=-250.08
## AIC=504.15   AICc=504.39   BIC=508.17
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 1.730281 22.50123 17.50413 0.6750518 4.903581 0.2142186
##                    ACF1
## Training set 0.04532975
```

2) Justify why you chose (ARMA or ARIMA) one over the other.

```
plot(UN.train) # mean not stationary
```




```
acf(UN.train) # dies down slowly, suggesting non-stationarity
```




```
adf.test(UN.train) # p-value > 0.05, fail to reject H0 of non-stationarity
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  UN.train
```

```
## Dickey-Fuller = -3.3336, Lag order = 3, p-value = 0.07538
## alternative hypothesis: stationary
```

```
kpss.test(UN.train) # p-value < 0.05, reject H0 of starionarity
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  UN.train
## KPSS Level = 0.61451, Truncation lag parameter = 3, p-value =
## 0.02132
```

I chose to use the ARIMA model for UN series data since UN is non-stationary. Stationarity is a requirement of the ARMA model. The ARIMA model, on the other hand, can handle non-stationary data by first differencing to achieve stationarity before applying the ARMA(p,q) model. Our ARIMA(1,1,0) model suggests 1 difference is needed.

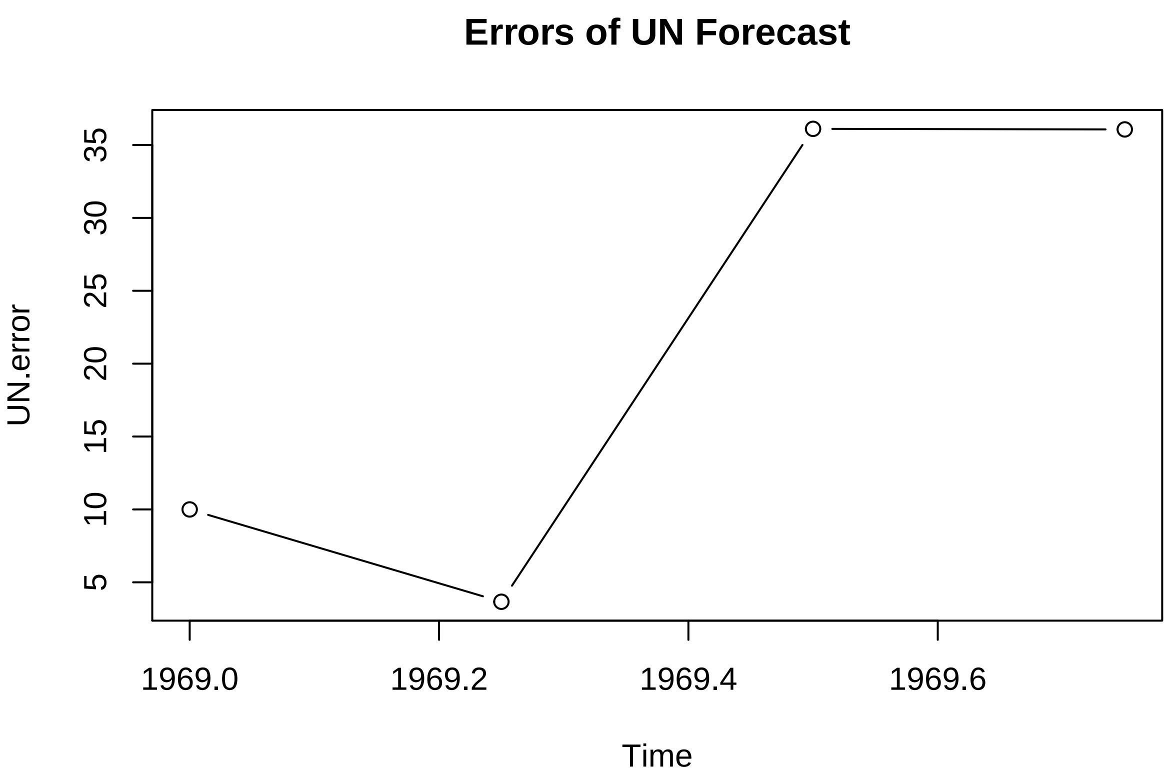
3) Use the chosen UN model to forecast the UN for 1969.

```
UN.fc <- forecast(UN.arima,h=4)
UN.fc
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1969 Q1       522.0004 492.6348 551.3661 477.0896 566.9113
## 1969 Q2       515.3343 458.2585 572.4101 428.0444 602.6243
## 1969 Q3       510.8904 426.6249 595.1560 382.0174 639.7635
## 1969 Q4       507.9280 397.9369 617.9191 339.7111 676.1448
```

4) Compare your forecasts with the actual values using error = actual - estimate and plot the errors.

```
UN.error <- UN.test - UN.fc$mean
plot(UN.error, type = "b", main = "Errors of UN Forecast")
```



5) Calculate the sum of squared error for the UN model.

```r
sum(UN.error^2)
```

```
## [1] 2718.519
```

GDP ARIMA Model

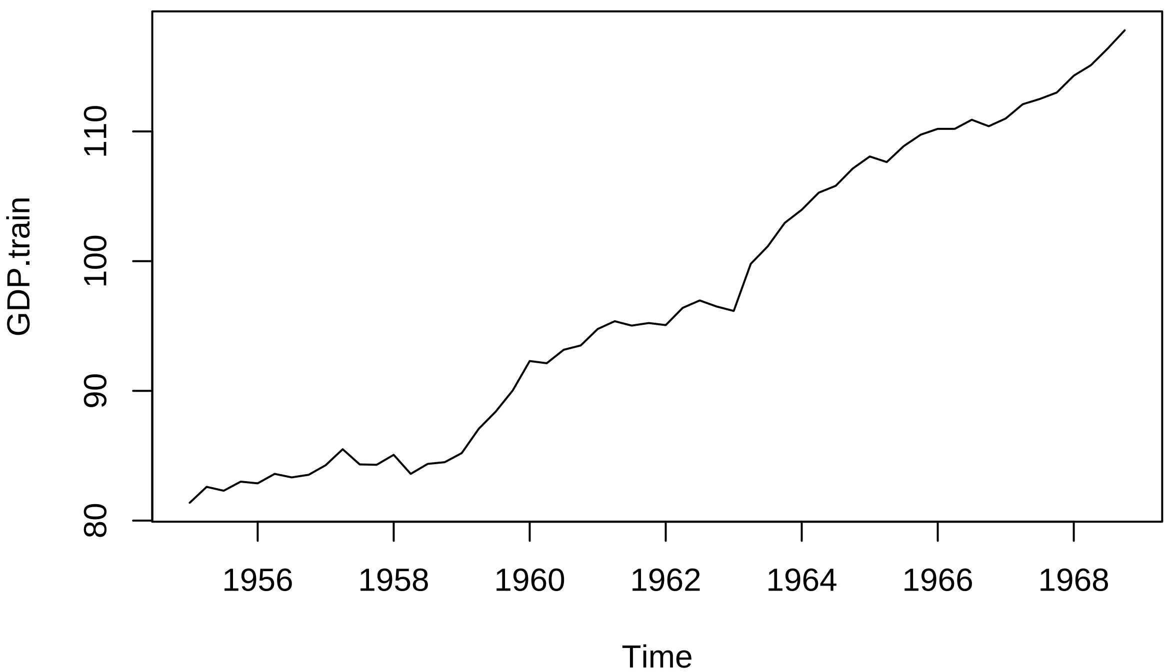
1) Use datasets from 1955 to 1968 to build an ARMA or ARIMA model for GDP.

```r
GDP.arima <- auto.arima(GDP.train)
summary(GDP.arima)
```

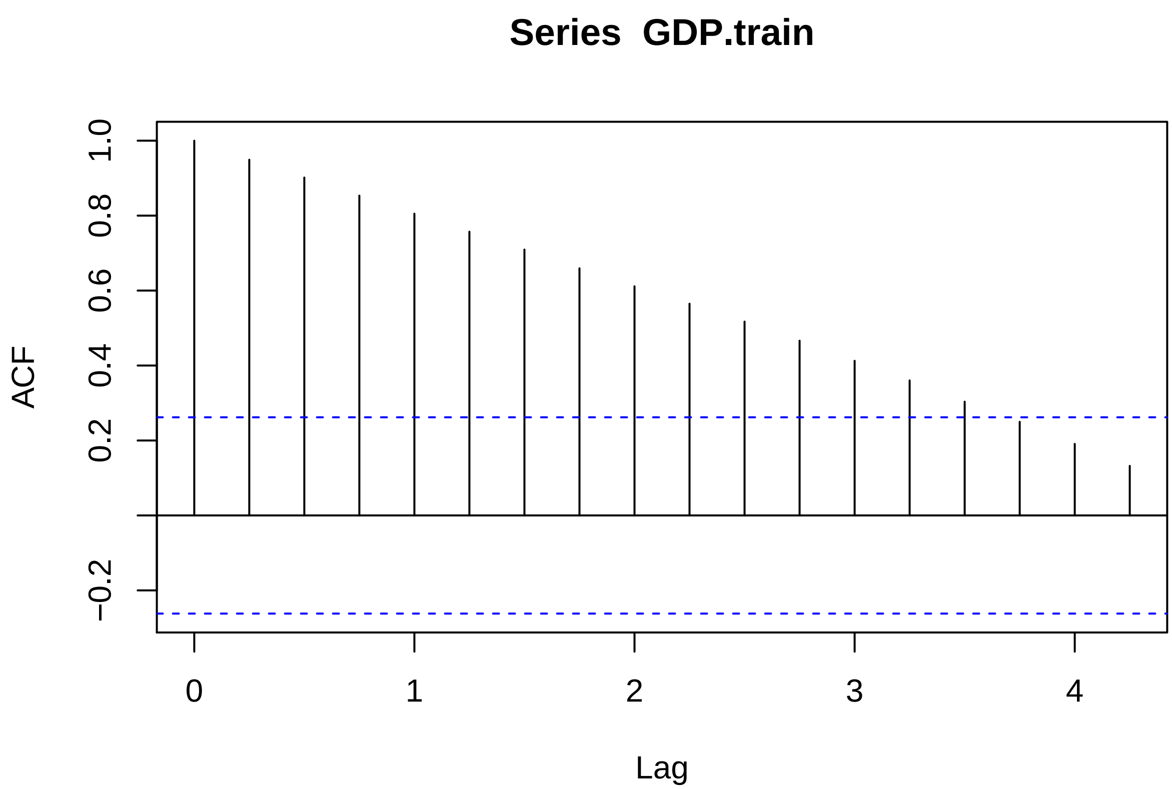
```
## Series: GDP.train
## ARIMA(0,1,0) with drift
##
## Coefficients:
##        drift
##       0.6624
## s.e.  0.1152
##
## sigma^2 estimated as 0.743:  log likelihood=-69.36
## AIC=142.73   AICc=142.96   BIC=146.74
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE      MAPE
## Training set 0.001441207 0.8464517 0.6381945 -0.02097366 0.6730658
##                   MASE       ACF1
## Training set 0.2402527 0.05442436
```

2) Justify why you chose (ARMA or ARIMA) one over the other for GDP.

```r
plot(GDP.train) # mean not stationary
```




```r
acf(GDP.train) # dies down slowly, suggesting non-stationarity
```




```r
adf.test(GDP.train) # p-value > 0.05, fail to reject H0 of non-stationarity
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  GDP.train
## Dickey-Fuller = -2.9551, Lag order = 3, p-value = 0.1895
## alternative hypothesis: stationary
```

```r
kpss.test(GDP.train) # p-value < 0.05, reject H0 of starionarity
```

```
## Warning in kpss.test(GDP.train): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  GDP.train
## KPSS Level = 1.4844, Truncation lag parameter = 3, p-value = 0.01
```

GDP data is also non-stationary. Since the ARIMA model handles non-stationary data by first differencing the data to achieve stationarity, I chose the ARIMA model over ARMA.

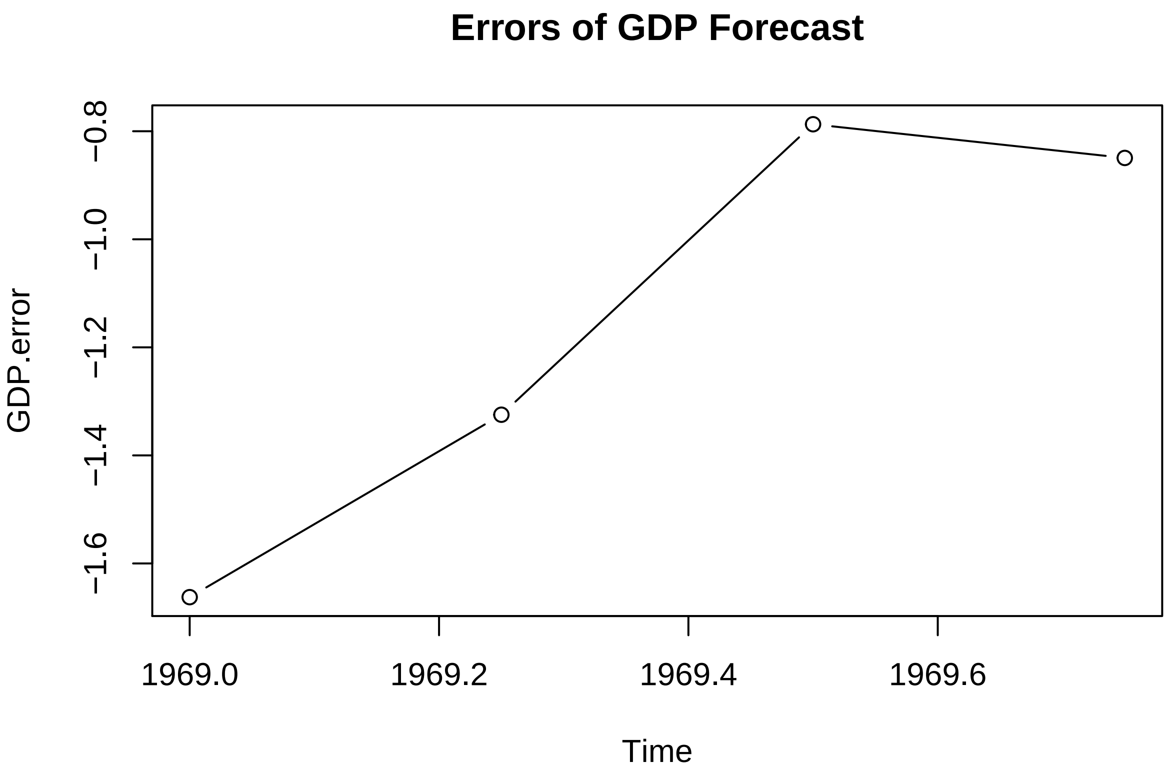
3) Use the chosen GDP model to forecast the GDP for 1969.

```r
GDP.fc <- forecast(GDP.arima,h=4)
GDP.fc
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1969 Q1       118.4624 117.3577 119.5670 116.7729 120.1518
## 1969 Q2       119.1247 117.5625 120.6870 116.7355 121.5140
## 1969 Q3       119.7871 117.8737 121.7004 116.8609 122.7133
## 1969 Q4       120.4495 118.2401 122.6588 117.0705 123.8284
```

4) Compare your forecasts for GDP with the actual values using error = actual - estimate and plot the errors.

```
GDP.error <- GDP.test - GDP.fc$mean
plot(GDP.error, type = "b", main = "Errors of GDP Forecast")
```



5) Calculate the sum of squared error for the GDP model.

```
sum(GDP.error^2)
```

```
## [1] 5.85944
```

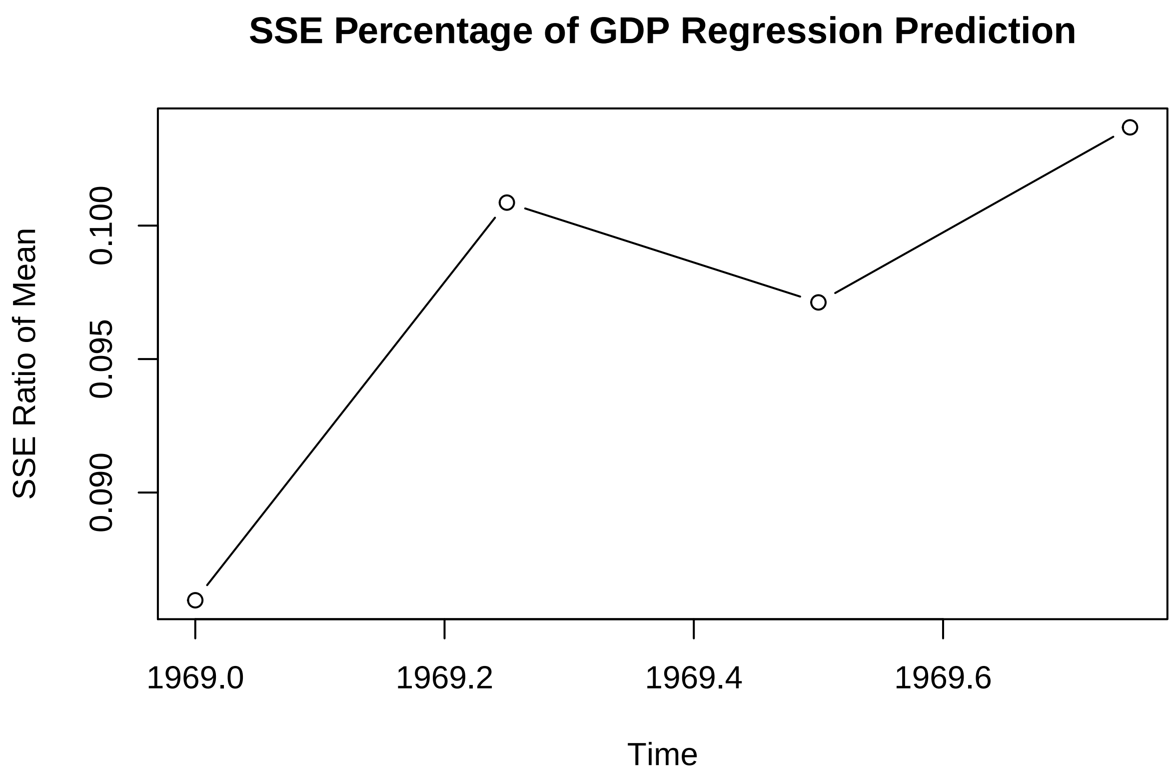
Regression

1) Build a regression model that uses UN as the independent variable and GDP as the dependent variable - use data from 1955 to 1968 to build the model. Forecast for 1969 and plot the errors as a percentage of the mean. Also calculate the sum of squared(error) as a percentage of the mean.

```
# The train and test data are the same as in the previous section
GDP.mod <- lm(GDP ~ UN, data = train)
GDP.pred <- predict(GDP.mod, test)
GDP.reg.error <- GDP.test - GDP.pred
GDP.reg.error
```

```
##          Qtr1     Qtr2     Qtr3     Qtr4
## 1969 10.16947 11.93207 11.48954 12.26553
```

```
# Calculate error as a percentage of the mean
GDP.error.pt <- GDP.reg.error/mean(GDP.test)
plot(GDP.error.pt, type = "b", main = "SSE Percentage of GDP Regression Prediction",
     ylab = "SSE Ratio of Mean")
```




```
# Calculate the sum of squared error as a percentage of the mean
SSE.GDP.pt <- sum(GDP.reg.error^2)/mean(GDP.test)
SSE.GDP.pt
```

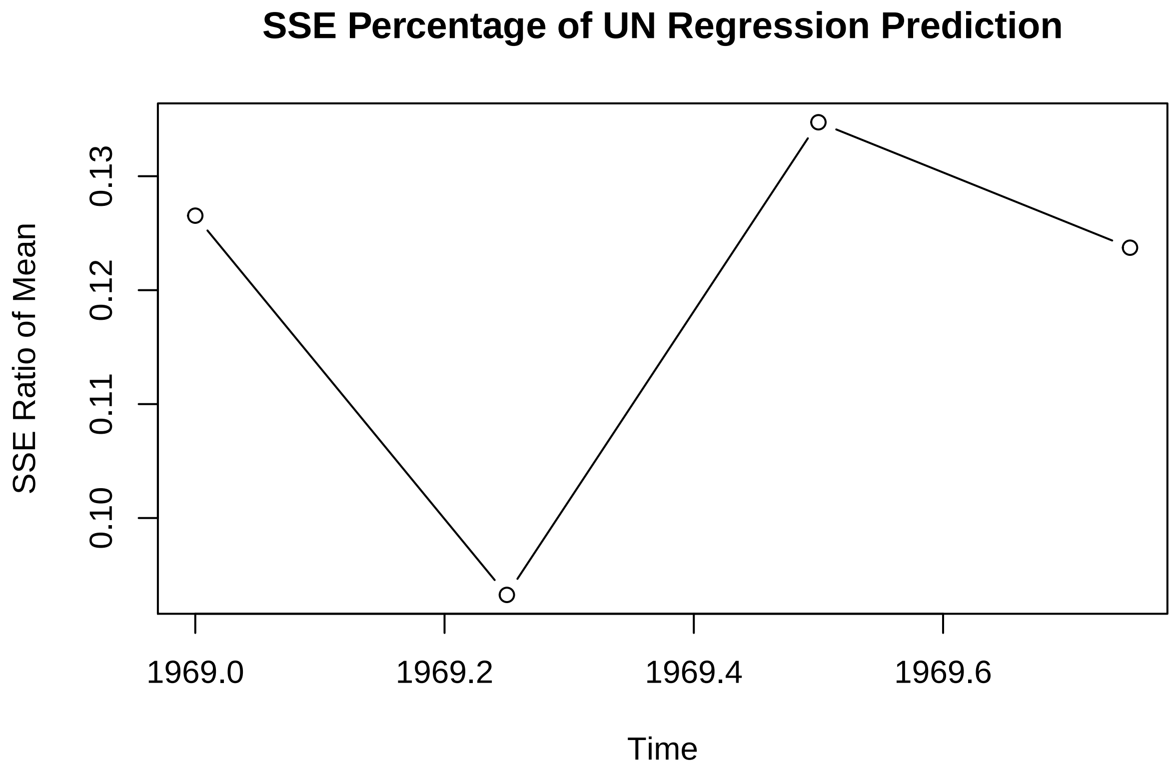
```
## [1] 4.465299
```

2) Build a regression model that uses GDP as the independent variable and UN as the dependent variable - use data from 1955 to 1968 to build the model. Forecast for 1969 and plot the errors as a percentage of the mean. Also calculate the sum of squared (error) as a percentage of the mean of the actual values.

```
UN.mod <- lm(UN ~ GDP, data = train)
UN.pred <- predict(UN.mod, test)
UN.reg.error <- UN.test - UN.pred
UN.reg.error
```

```
##          Qtr1     Qtr2     Qtr3     Qtr4
## 1969 67.76199 49.93849 72.15029 66.25619
```

```
# Calculate error as a percentage of the mean
UN.error.pt <- UN.reg.error/mean(UN.test)
plot(UN.error.pt, type = "b", main = "SSE Percentage of UN Regression Prediction",
     ylab = "SSE Ratio of Mean")
```




```r
# Calculate the sum of squared error as a percentage of the mean
SSE.UN.pt <- sum(UN.reg.error^2)/mean(UN.test)
SSE.UN.pt
```

```
## [1] 31.15049
```

3) Compare the 2 models using the sum of squared error as a percentage of the mean of the actual values - any reason to believe which should be the independent and the dependent variable?

The errors for the 2 regression models will have different orders of magnitude, making it difficult to compare without normalization. To normalize for comparison, the sum of the squared errors for each model were taken and divided by the mean, so we can obtain a percentage that can now be compared across the 2 models.

Comparing the 2 models, SSE as a percentage of the mean of UN (31.15) is much greater than SSE as a percentage of the mean of GDP (4.47). This suggest that the model in which GDP is the dependent variable has more accurate predictions and thus GDP should be the dependent variable and UN the independent variable.