# Assignment3_TS

Weijie Gao

10/19/2017

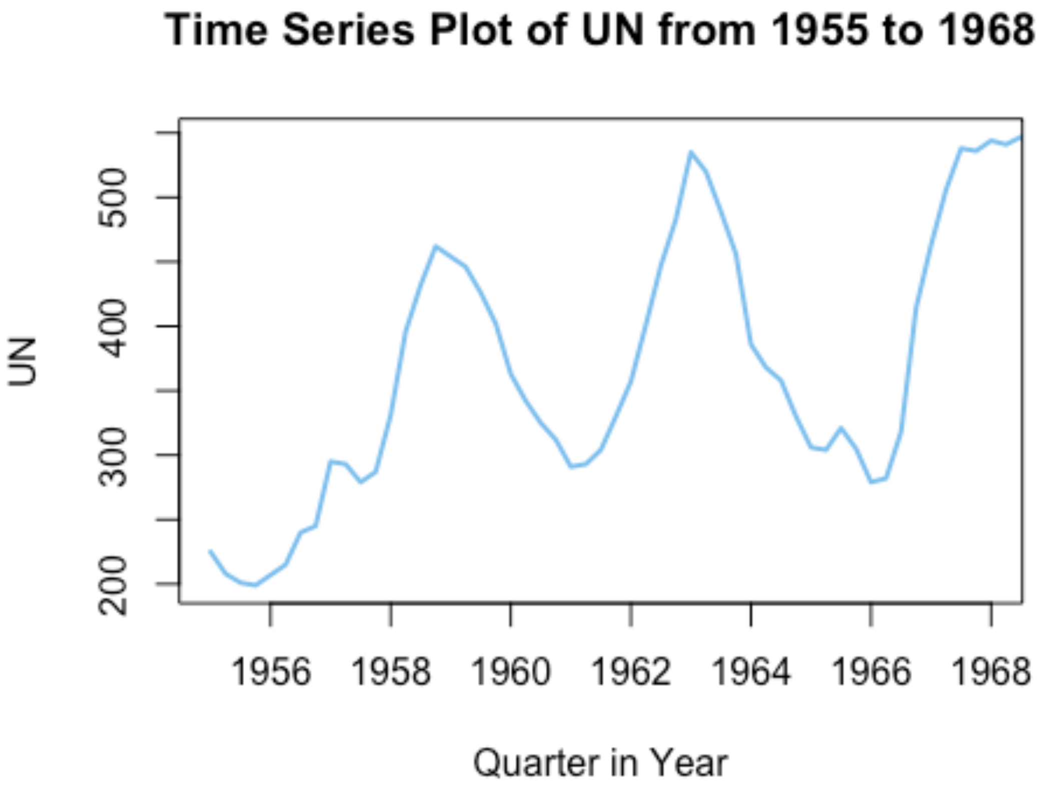
```
library(xlsx)

## Loading required package: rJava

## Loading required package: xlsxjars

library(timeSeries)

## Loading required package: timeDate

library(forecast)
library(tseries)
library(TSA)

## Loading required package: leaps

## Loading required package: locfit

## locfit 1.5-9.1    2013-03-22

## Loading required package: mgcv

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:forecast':
##
##     getResponse

## This is mgcv 1.8-17. For overview type 'help("mgcv-package")'.

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:timeDate':
##
##     kurtosis, skewness

## The following objects are masked from 'package:stats':
##
##     acf, arima
## The following object is masked from 'package:utils':
##
##     tar
```

```
df <- read.xlsx("Unemployment_GDP_UK.xlsx", sheetIndex = 1)
head(df)
```

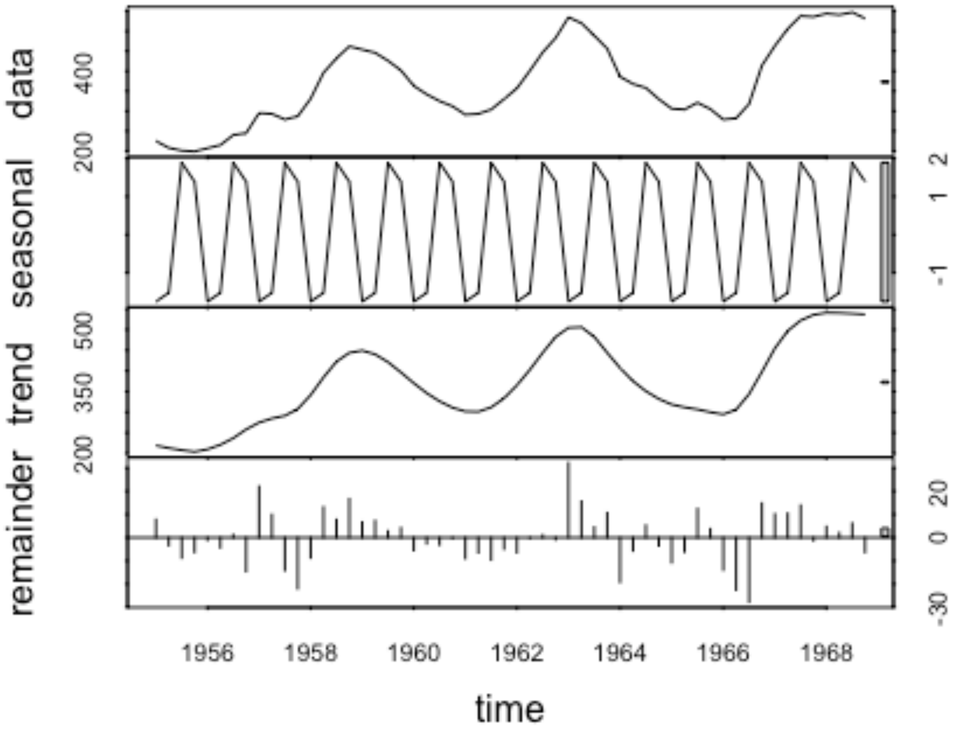
```
##   Year Quarter  UN   GDP
## 1 1955       1 225 81.37
## 2   NA       2 208 82.60
## 3   NA       3 201 82.30
## 4   NA       4 199 83.00
## 5 1956       1 207 82.87
## 6   NA       2 215 83.60
```

## ARIMA Modeling

```
# Use datasets from 1955 to 1968 to build an ARMA or ARIMA models for UN and
GDP
UN <- df[1:56,3]
UN_ts <- ts(UN,start=1955,freq=4)
plot(UN_ts,xlab="Quarter in Year", ylab="UN",lwd=2,
col='skyblue2',lty=1,xlim=c(1955,1968), main= "Time Series Plot of UN from
1955 to 1968")
```



```
plot(stl(UN_ts,s.window="periodic"))
```



From both the time series plot of UN from 1955 to 1968 and the stl decomposition shows that there is an upward trend and there is strong and regular seasonality in this time series, which suggests the time series data is non-stationary and at least we need to take the first difference, so we need ARIMA model.

```
# check stationarity
adf.test(UN, k = 0)

##
##  Augmented Dickey-Fuller Test
##
## data:  UN
## Dickey-Fuller = -1.3298, Lag order = 0, p-value = 0.8448
## alternative hypothesis: stationary
```

The ADF test returns p-value of 0.8448, which is greater than the significance level, meaning we fail to reject the H0 and there is sufficient evidence to suggest this time series is non-stationary.

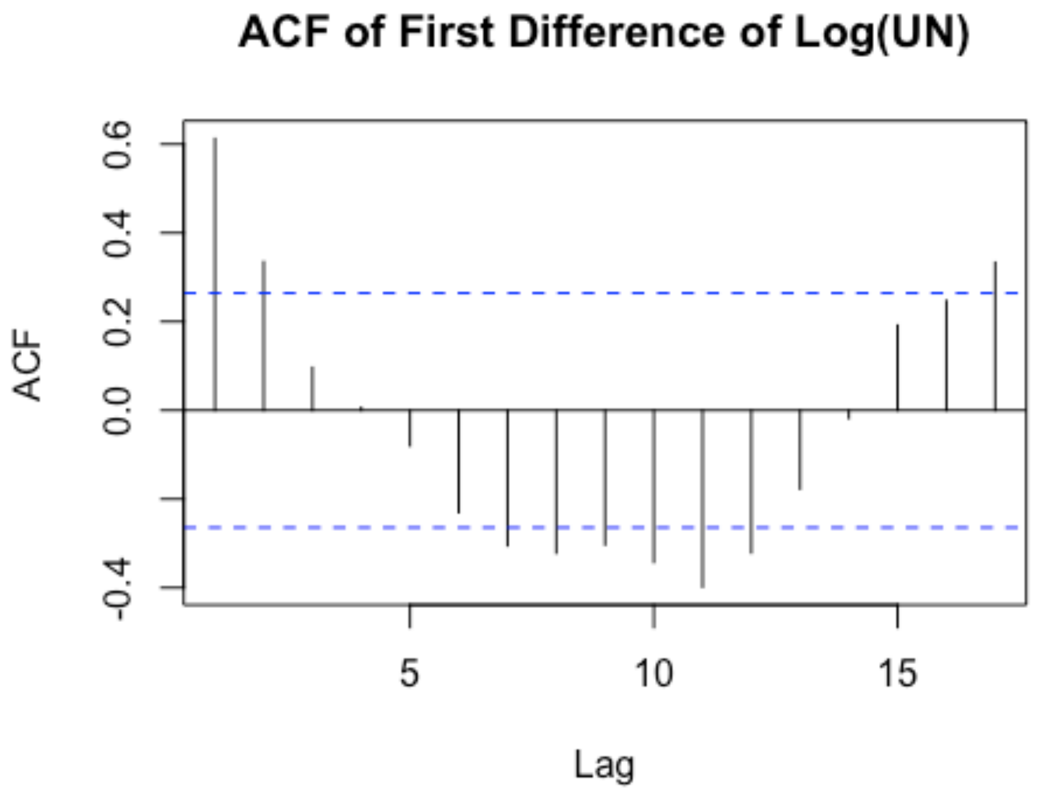
```
# convert to stationarity
UN_diff <- diff(UN)
adf.test(UN_diff, k = 0)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  UN_diff
## Dickey-Fuller = -3.2119, Lag order = 0, p-value = 0.09461
## alternative hypothesis: stationary
```

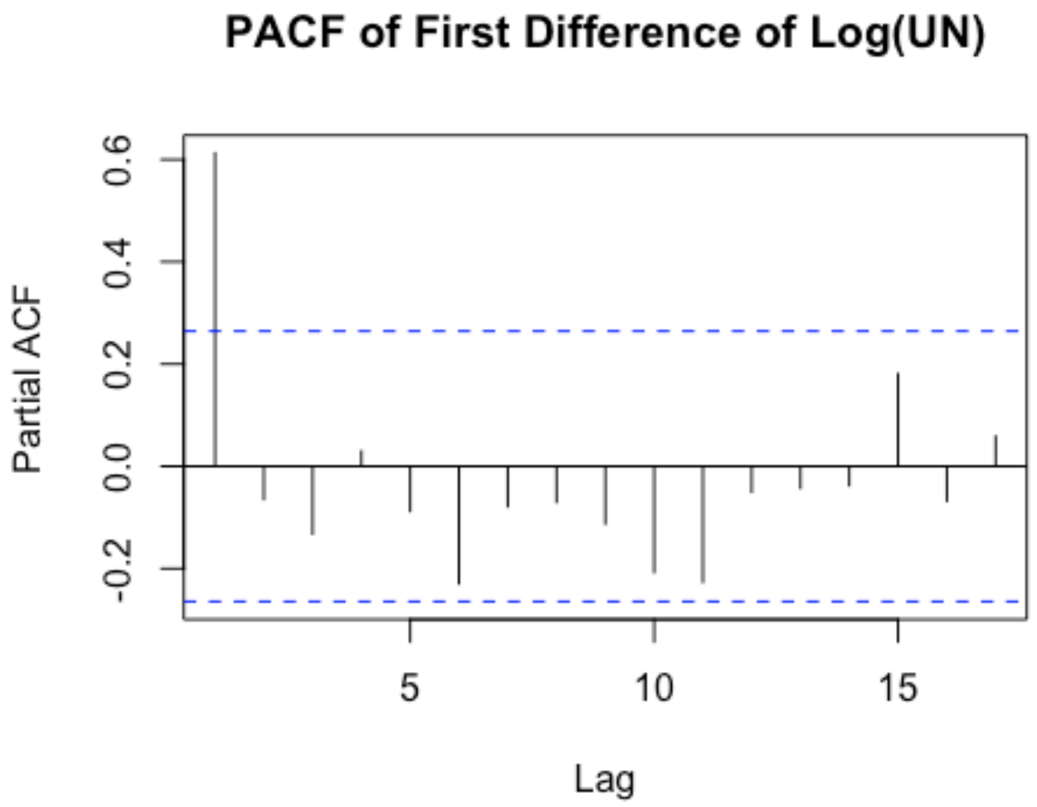
```
# taking first difference only does not convert it into stationary time
series
UN_log_diff <- diff(log(UN))
adf.test(UN_log_diff, k = 0)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  UN_log_diff
## Dickey-Fuller = -3.5499, Lag order = 0, p-value = 0.04548
## alternative hypothesis: stationary
```

```
# check plots
acf(UN_log_diff,main="ACF of First Difference of Log(UN)")
```



```
pacf(UN_log_diff,main="PACF of First Difference of Log(UN)")
```



```r
eacf(UN_log_diff)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x o o o o x x x x x  x  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 o x o o o o o o o o o  o  o  o
## 4 o o x o o o o o o o o  o  o  o
## 5 x o x o o o o o o o o  o  o  o
## 6 o x o o o o o o o o o  o  o  o
## 7 x x o o o o o o o o o  o  o  o
```

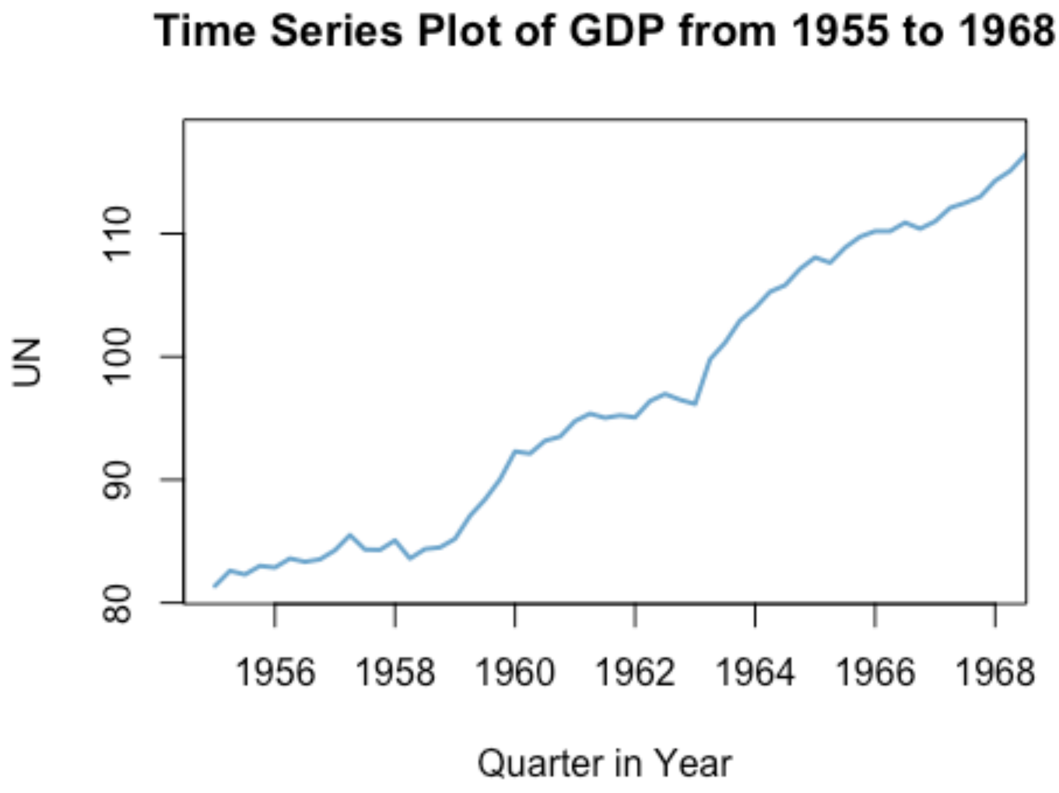
In ACF Plot, there is a significant spike at lag 1 but it's not a clear cut, spikes ocillate. In PACF plot, there is a clear cut at lag 1. So AR(1) model is suggested. Since we need to take the first difference of the data to make it stationary. We need to fit the ARIMA model with p=1, d=1, q=0.

```r
# fit model
UN_log <- log(UN)
fitUN <- Arima(UN_log, order = c(1,1,0), seasonal = FALSE)
summary(fitUN)
```

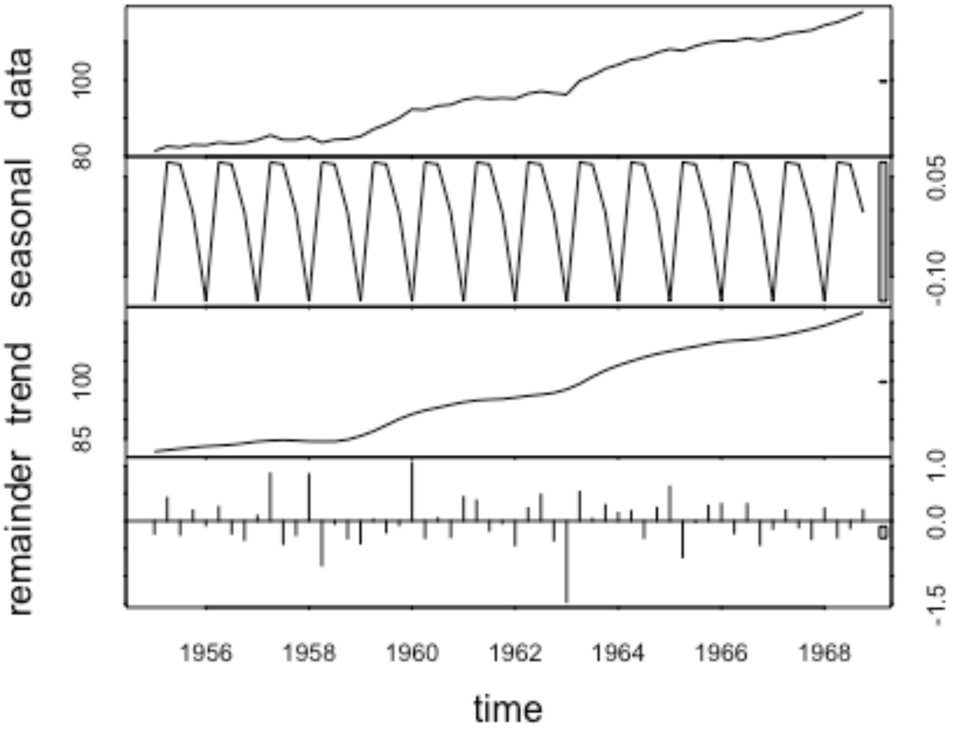
```
## Series: UN_log
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.6308
## s.e.  0.1027
##
## sigma^2 estimated as 0.004255:  log likelihood=72.36
## AIC=-140.71   AICc=-140.48   BIC=-136.7
##
## Training set error measures:
##                       ME       RMSE        MAE       MPE      MAPE
## Training set 0.005770515 0.06405255 0.04982217 0.1039695 0.8534669
##                   MASE       ACF1
## Training set 0.7554639 0.02918212
```

## GDP:

```
GDP_55_68 <- df[1:56,4]
GDP_55_68_ts <- ts(GDP_55_68,start=1955,freq=4)
plot(GDP_55_68_ts,ylab="UN", xlab="Quarter in Year", lwd=2,
col='skyblue3',lty=1,xlim=c(1955,1968), main= "Time Series Plot of GDP from
1955 to 1968")
```



```
plot(stl(GDP_55_68_ts,s.window="periodic"))
```

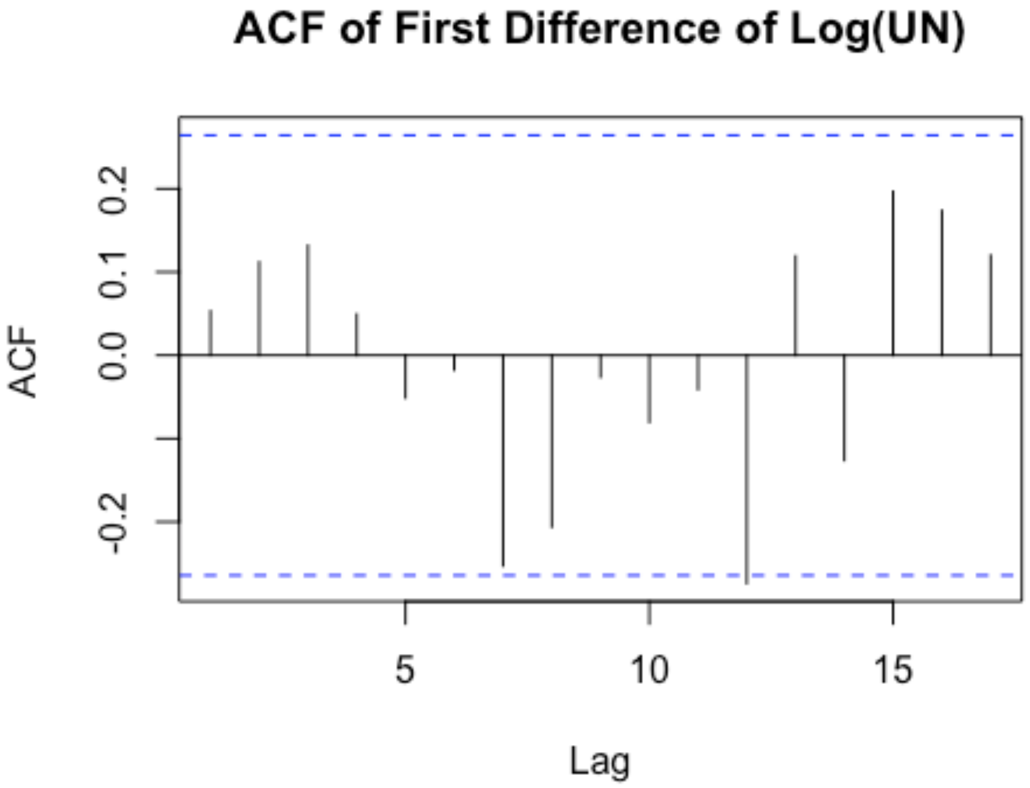


From time series plot of GDP from 1955 to 1968, we can see an upward trend and seasonality. The stl decomposition shows that there is an upward trend and there is strong and regular seasonality in this time series. These implies that the time series data is non-stationary and at least we need to take the first different, so we need ARIMA model. So it's ARIMA(1,1,0).

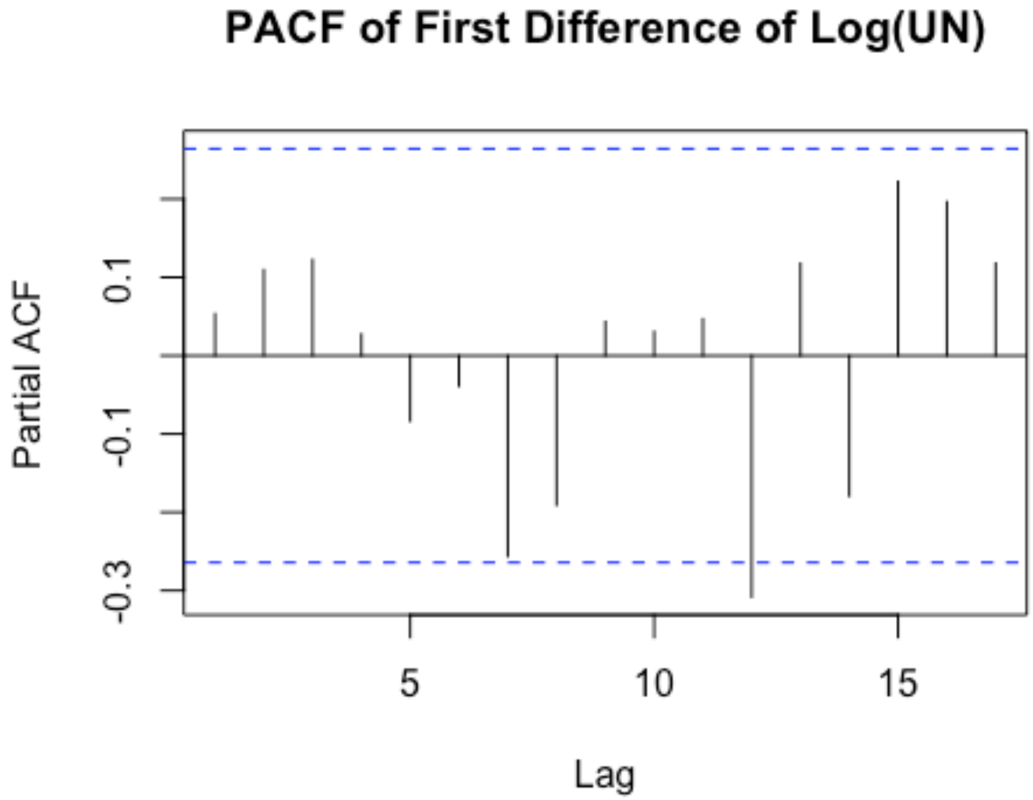
```
# convert to stationary
GDP_55_68_diff <- diff(GDP_55_68)
adf.test(GDP_55_68_diff, k = 0)

## Warning in adf.test(GDP_55_68_diff, k = 0): p-value smaller than printed
p-
## value

##
##  Augmented Dickey-Fuller Test
##
## data:  GDP_55_68_diff
## Dickey-Fuller = -7.0623, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

After taking the first difference of data, the ADF test shows p-value smaller than 0.05, meaning we reject the H0 and can concludes the time series is now stationary.

```
# check plots
acf(GDP_55_68_diff,main="ACF of First Difference of Log(UN)")
```



```
pacf(GDP_55_68_diff,main="PACF of First Difference of Log(UN)")
```



```
eacf(GDP_55_68_diff)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o o o o o o o o o o o  o  o  o
## 1 x o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 o x o o o o o o o o o  o  o  o
## 4 x x o o o o o o o o o  o  o  o
## 5 x o o x o o o o o o o  o  o  o
## 6 o o o x o o o o o o o  o  o  o
## 7 x x o o o o o o o o o  o  o  o
```

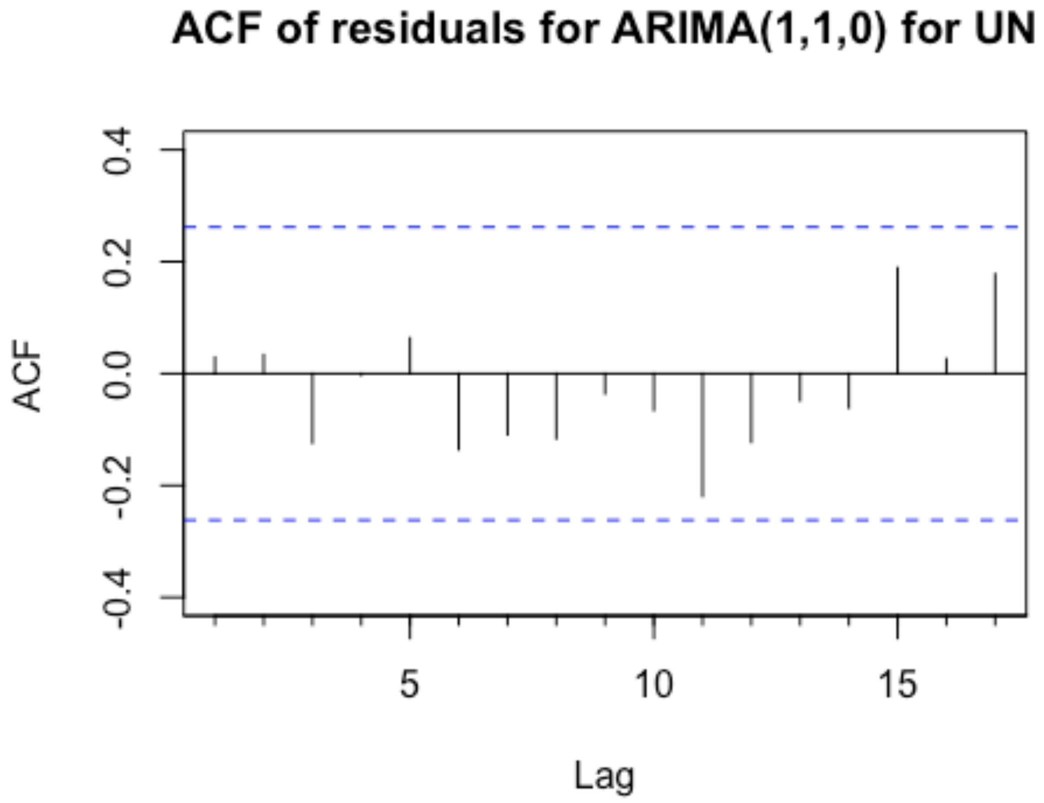
Both ACF and PACF plot of GDP first difference shows there is no clear pattern and no spikes where has a clear cut-off, and from EACF plot we could see both p and q could equal to 0. This suggests that GDP is a random walk process with d = 1, p = q = 0. So it's ARIMA (0,1,0).

```
# fit model
fitGDP <- Arima(GDP_55_68, order = c(0,1,0))
summary(fitGDP)

## Series: GDP_55_68
## ARIMA(0,1,0)
```

```
##
## sigma^2 estimated as 1.168:  log likelihood=-82.31
## AIC=166.63   AICc=166.71   BIC=168.64
##
## Training set error measures:
##                       ME     RMSE       MAE       MPE      MAPE      MASE
## Training set 0.6519887 1.071155 0.8584173 0.6562516 0.8859078 0.9838081
##                     ACF1
## Training set 0.04448263
```

As discussed above, both UN and GDP data are non-stationary time series data so that first difference is needed to make them stationary for analysis. That's why in both cases, ARIMA model is needed. We have also tested the results through auto.arima(). To see how ARIMA models performs, we will perform some residuals analysis.

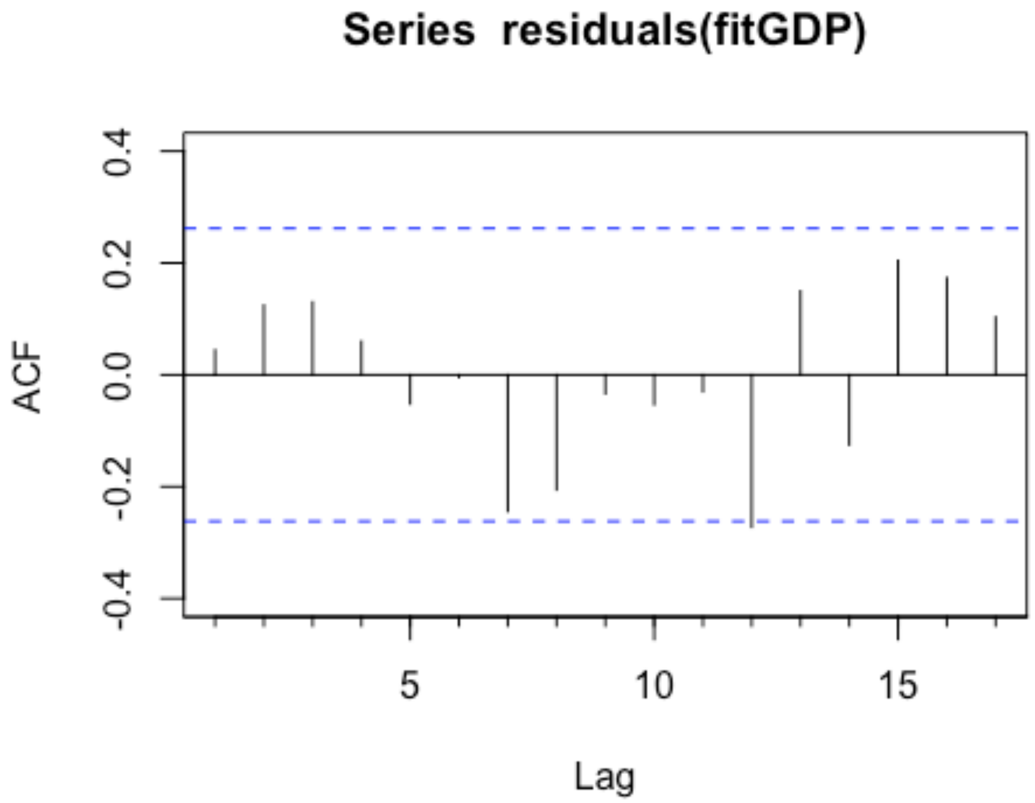
```
# residuals for UN
Acf(residuals(fitUN),main="ACF of residuals for ARIMA(1,1,0) for UN")
```



```
Box.test(residuals(fitUN), type="Ljung")
```

```
##
##  Box-Ljung test
##
```

```
## data:  residuals(fitUN)
## X-squared = 0.050291, df = 1, p-value = 0.8226

# residuals for GDP
Acf(residuals(fitGDP))
```



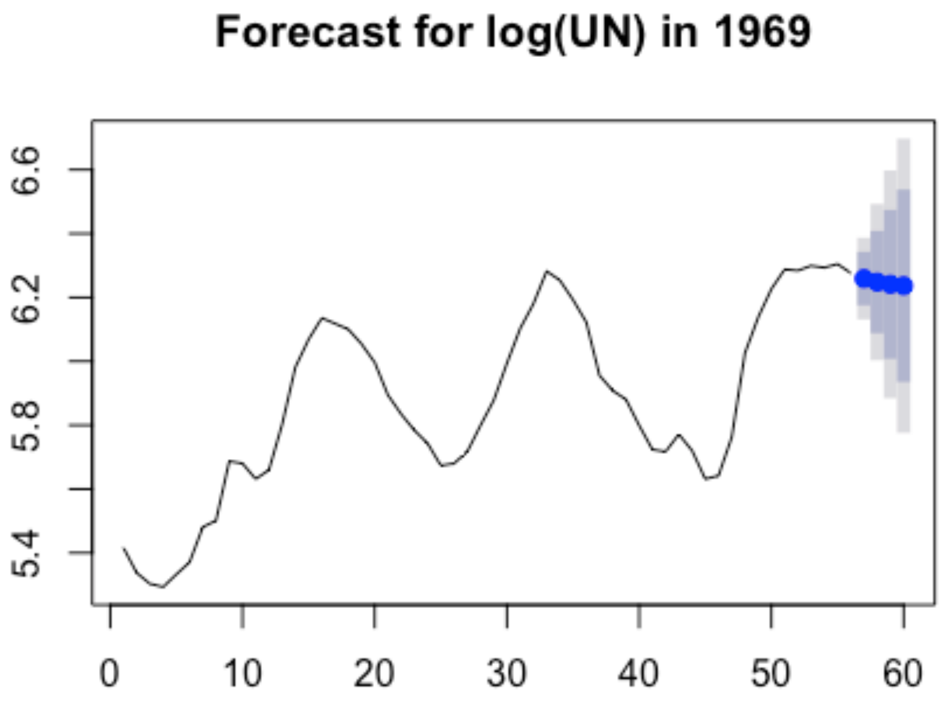
```
Box.test(residuals(fitGDP), type="Ljung")

##
##  Box-Ljung test
##
## data:  residuals(fitGDP)
## X-squared = 0.11685, df = 1, p-value = 0.7325
```

UN: From ACF of residuals plot, we can see all spikes are within the boundary, suggesting residuals are not correlated. From Ljung Box test, p-value is greater than 0.05, suggesting we fail to reject the null hypothesis and there is sufficient evidence to suggest residuals are independent, like white noise. So our model for UN fits well.

GDP: Similarly,the ACF plot shows residuals are not correlated. Ljung Box test has a large p-value, suggesting residuals are independent, like white noise. So our model for GDP fits well too.

### Use the chosen UN and GDP models to forecast the UN and the GDP for 1969.

```
# UN: ARIMA(1,1,0)
UN_69_log <- forecast(fitUN,4)
UN_69 <- exp(UN_69_log$mean)
UN_69

## Time Series:
## Start = 57
## End = 60
## Frequency = 1
## [1] 522.7496 516.9970 513.4005 511.1446

plot(UN_69_log,main = "Forecast for log(UN) in 1969")
```

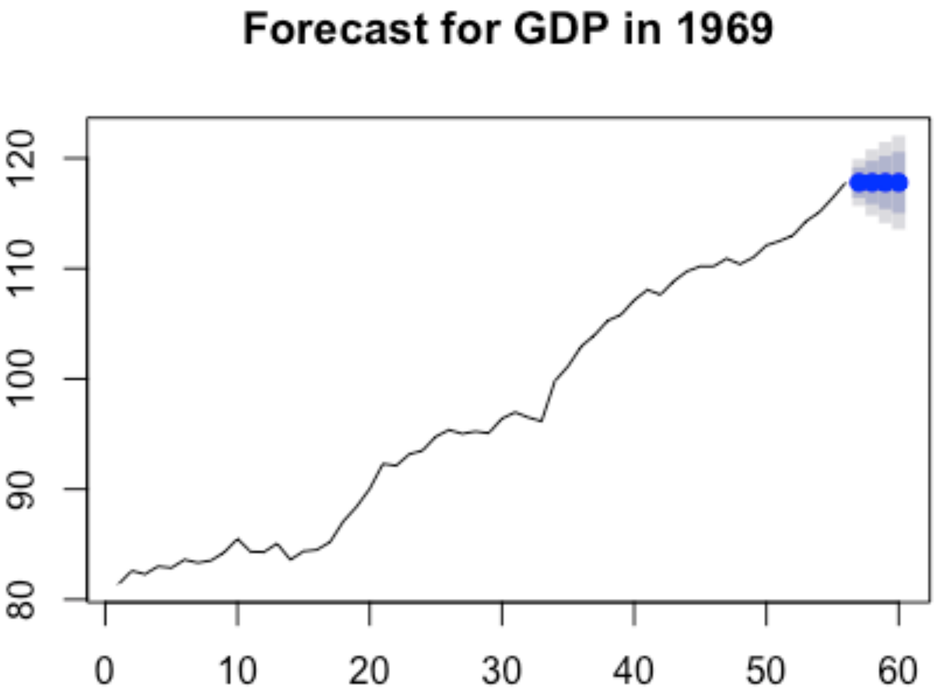


```
# GDP: ARIMA(0,1,0)
GDP_69 <- forecast(fitGDP,4)
GDP_69

##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 57          117.8 116.4148 119.1852 115.6816 119.9184
## 58          117.8 115.8411 119.7589 114.8041 120.7959
## 59          117.8 115.4008 120.1992 114.1308 121.4692
## 60          117.8 115.0297 120.5703 113.5631 122.0369
```

```
GDP_69$mean # point forecast
```

```
## Time Series:
## Start = 57
## End = 60
## Frequency = 1
## [1] 117.8 117.8 117.8 117.8
```

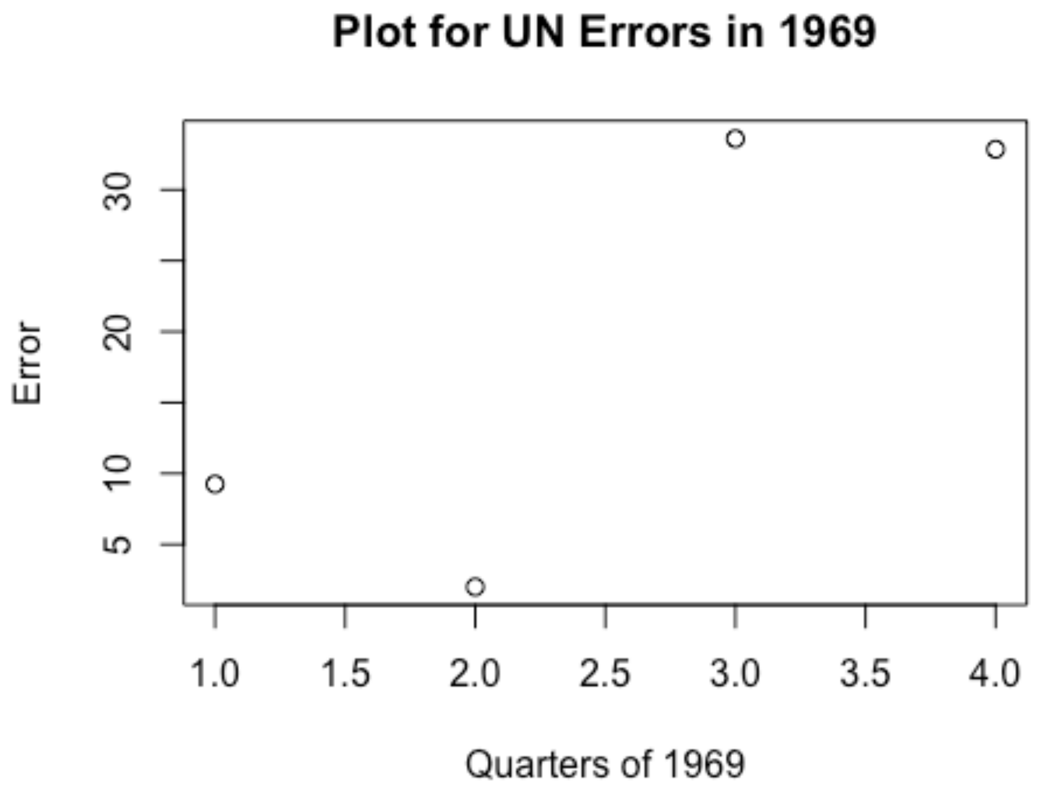
```
# GDP does not need log transformation, can directly interpret in graphs
plot(GDP_69, main = "Forecast for GDP in 1969")
```



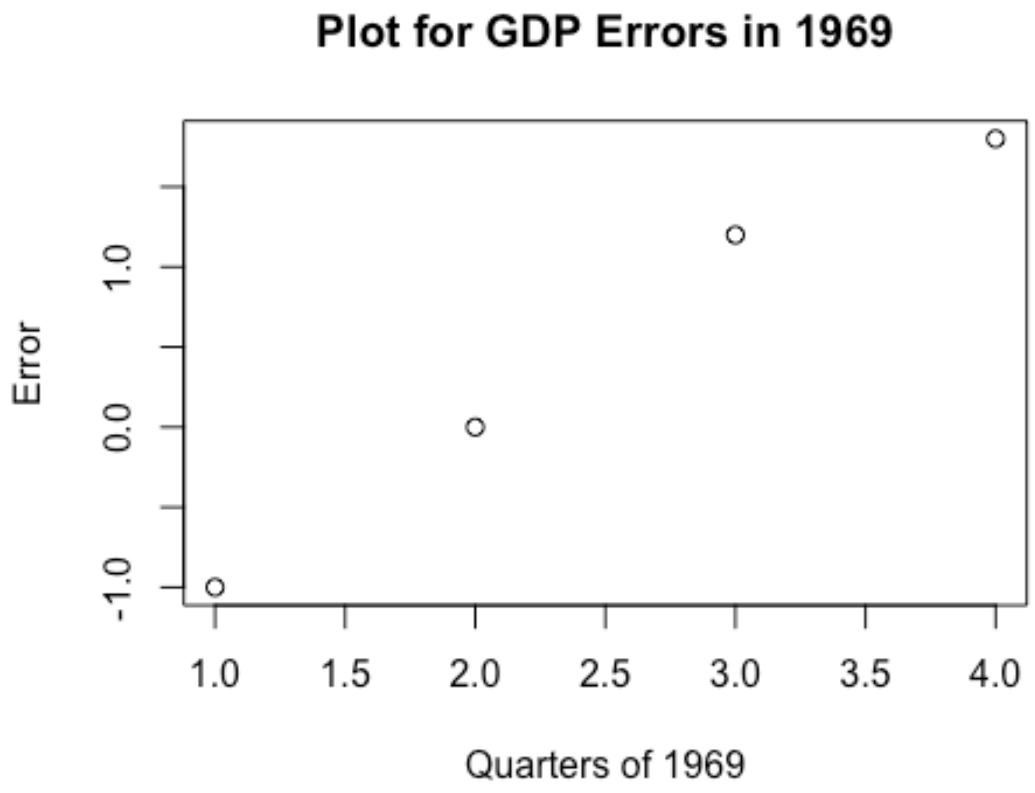
```
# UN:
UN_error <- df[57:60,3] - UN_69
UN_error
```

```
## Time Series:
## Start = 57
## End = 60
## Frequency = 1
## [1]  9.250378  2.003025 33.599451 32.855362
```

```
plot(UN_error, type = "p", x = 1:4, xlab = "Quarters of 1969", ylab =
"Error", main = "Plot for UN Errors in 1969")
```



```
# GDP:
GDP_error <- df[57:60,4] - GDP_69$mean
GDP_error

## Time Series:
## Start = 57
## End = 60
## Frequency = 1
## [1] -1.0  0.0  1.2  1.8

plot(GDP_error, type = "p", x = 1:4, xlab = "Quarters of 1969", ylab =
"Error", main = "Plot for GDP Errors in 1969")
```



```
# UN: SSE
(UN_sse <- sum((df[57:60,3] - UN_69)^2))

## [1] 2297.979

# GDP: SSE
(GDP_sse <- sum((df[57:60,4] - GDP_69$mean)^2))

## [1] 5.68
```

- The sum of squared error for UN model is 2297.979. The sum of squared error for GPA model is 5.68.

```
# build model
lm1 <- lm(GDP~UN, data = df[1:56, 3:4])
summary(lm1)

##
## Call:
## lm(formula = GDP ~ UN, data = df[1:56, 3:4])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.024  -7.146  -1.932   8.024  18.411
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 75.42255    4.87117  15.483  < 2e-16 ***
## UN           0.05866    0.01271   4.616 2.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.84 on 54 degrees of freedom
## Multiple R-squared:  0.283,  Adjusted R-squared:  0.2697 
## F-statistic: 21.31 on 1 and 54 DF,  p-value: 2.453e-05
```

```
(GDP_lm_69_with_ci <- predict(lm1, newdata = data.frame(UN=df[57:60,3]),
interval="confidence"))
```

```
##        fit      lwr      upr
## 1 106.6305 101.7135 111.5476
## 2 105.8679 101.2271 110.5088
## 3 107.5105 102.2668 112.7541
## 4 107.3345 102.1568 112.5122
```

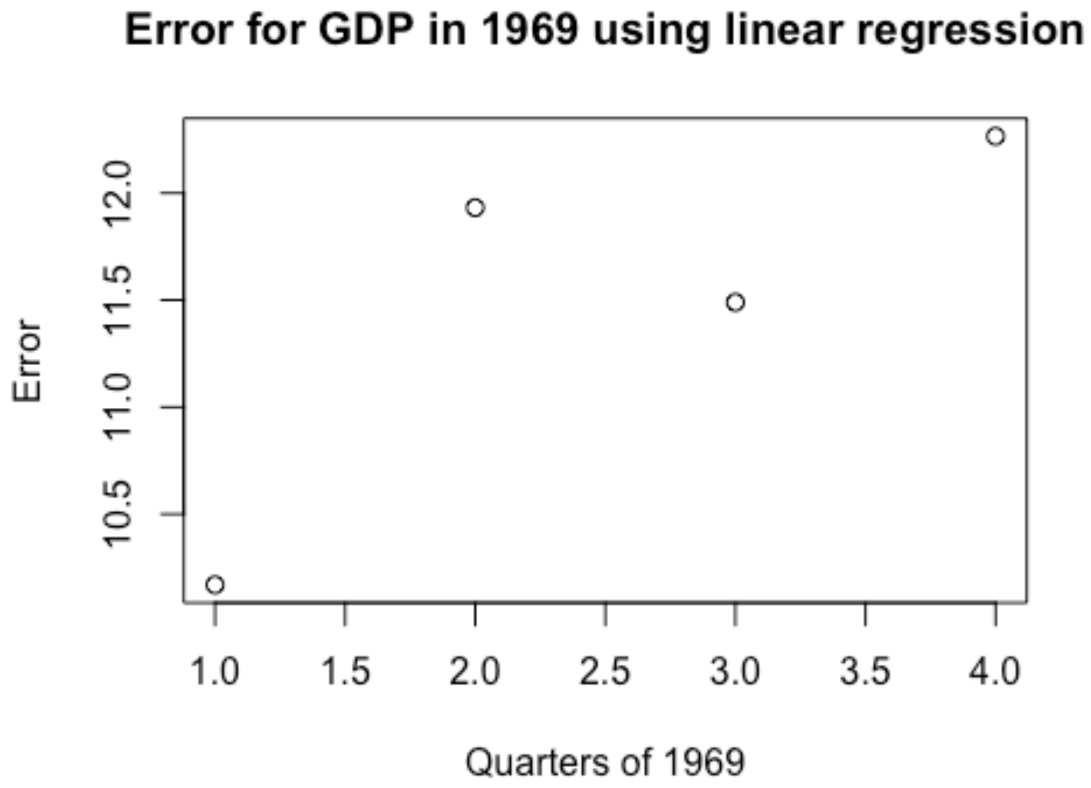
```
# point forecast only
(GDP_lm_69 <- predict(lm1, newdata = data.frame(UN=df[57:60,3])))
```

```
##        1        2        3        4 
## 106.6305 105.8679 107.5105 107.3345
```

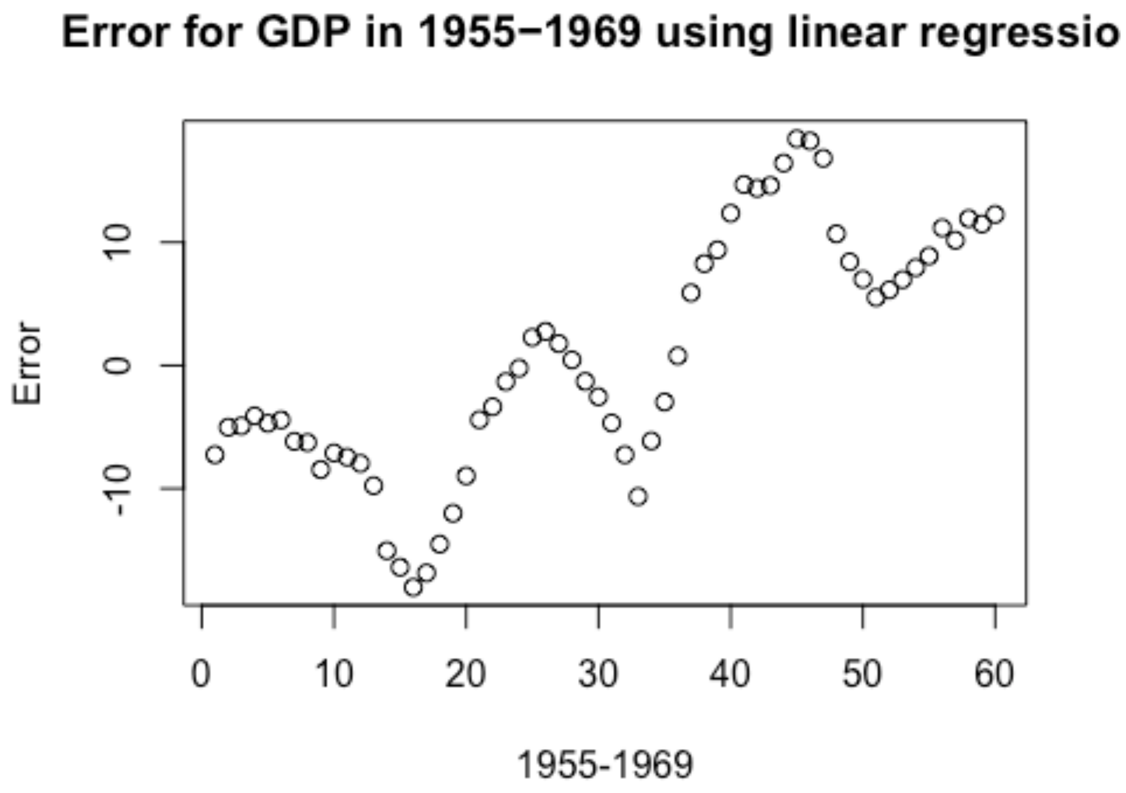
```
(GDP_lm_69_error <- df[57:60,4] - GDP_lm_69)
```

```
##        1        2        3        4 
## 10.16947 11.93207 11.48954 12.26553
```

```
plot(GDP_lm_69_error, type = "p", x = 1:4, xlab = "Quarters of 1969", ylab =
"Error", main = "Error for GDP in 1969 using linear regression")
```



```
# sum of squared error
(GDP_lm_69_sse <- sum((df[57:60,4] - GDP_lm_69)^2))

## [1] 528.2449

GDP_lm_error <- df$GDP - predict(lm1, newdata = data.frame(UN = df$UN))
# plot error
plot(GDP_lm_error, type = "p", xlab = "1955-1969", ylab = "Error", main =
"Error for GDP in 1955–1969 using linear regression")
```



```
(GDP_lm_sse <- sum((df$GDP - predict(lm1, newdata = data.frame(UN =
df$UN)))^2))

## [1] 5756.887

# build model
lm2 <- lm(UN~GDP, data = df[1:56, 3:4])
summary(lm2)

##
## Call:
## lm(formula = UN ~ GDP, data = df[1:56, 3:4])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -153.40  -67.99  -17.84   84.76  170.32
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -99.147    102.132  -0.971    0.336
## GDP            4.824      1.045   4.616 2.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 89.23 on 54 degrees of freedom
## Multiple R-squared:  0.283,  Adjusted R-squared:  0.2697
## F-statistic: 21.31 on 1 and 54 DF,  p-value: 2.453e-05
```

```
(UN_lm_69_with_ci <- predict(lm2, newdata = data.frame(GDP=df[57:60,4]),
interval="confidence"))
```

```
##        fit      lwr      upr
## 1 464.2380 416.4971 511.9789
## 2 469.0615 419.4961 518.6269
## 3 474.8497 423.0679 526.6315
## 4 477.7438 424.8438 530.6438
```

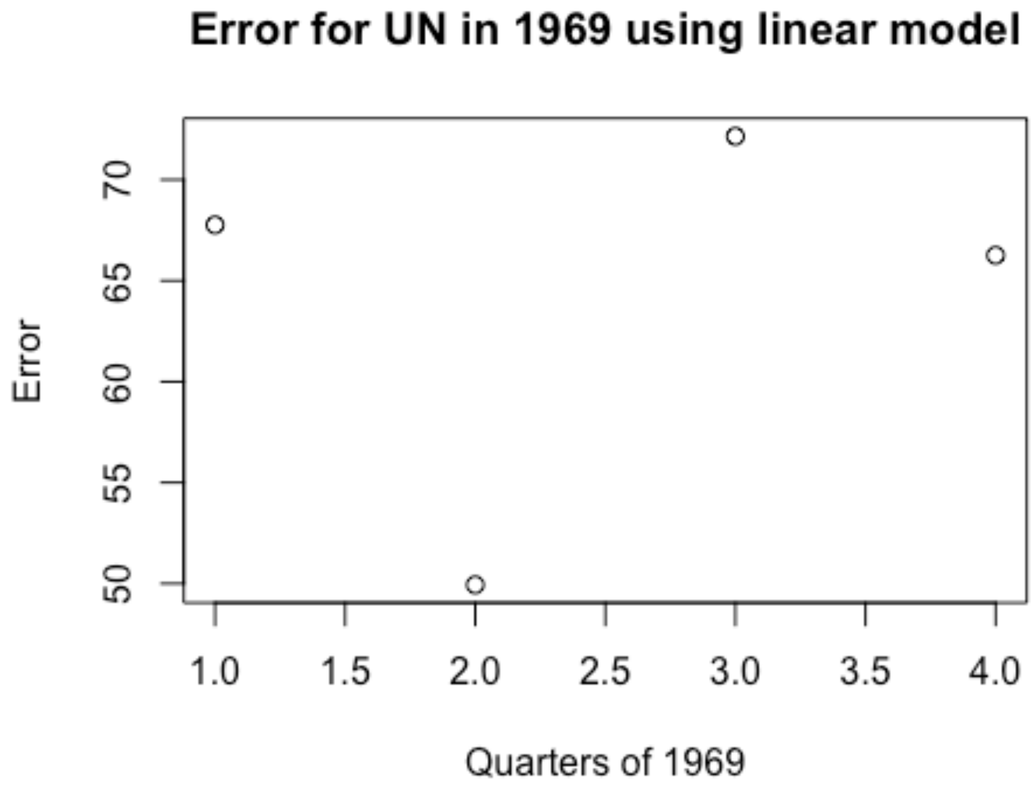
```
# point forecast only
(UN_lm_69 <- predict(lm2, newdata = data.frame(GDP=df[57:60,4])))
```

```
##        1        2        3        4
## 464.2380 469.0615 474.8497 477.7438
```

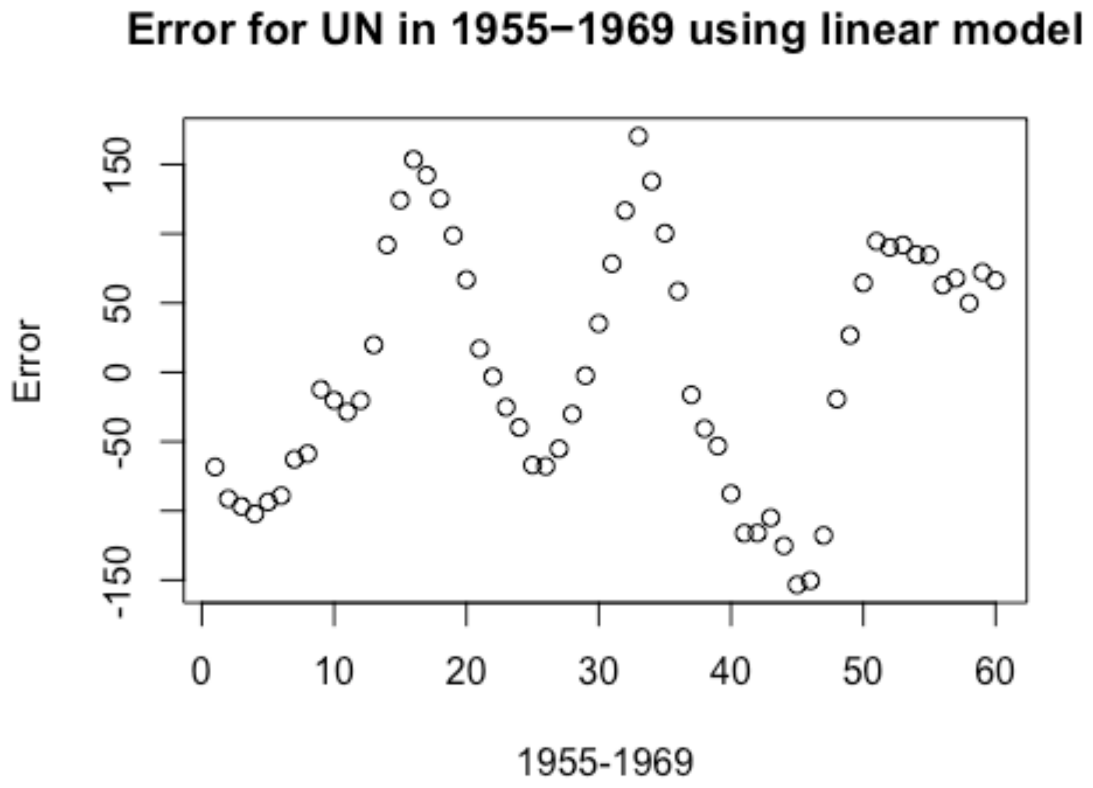
```
(UN_lm_69_error <- df[57:60,3] - UN_lm_69)
```

```
##        1        2        3        4
## 67.76199 49.93849 72.15029 66.25619
```

```
plot(UN_lm_69_error, type = "p", x = 1:4, xlab = "Quarters of 1969", ylab =
"Error", main = "Error for UN in 1969 using linear model")
```



```
# sum of squared error
(UN_lm_69_sse <- sum((df[57:60,3] - UN_lm_69)^2))

## [1] 16681.09

UN_lm_error <- df$UN - predict(lm2, newdata = data.frame(GDP = df$GDP))
# plot error
plot(UN_lm_error, type = "p", xlab = "1955-1969", ylab = "Error", main =
"Error for UN in 1955–1969 using linear model")
```



```
# sum of squared error
(UN_lm_sse <- sum((df$UN - predict(lm2, newdata = data.frame(GDP =
df$GDP)))^2))

## [1] 446610.6

cbind(Forecast_GDP_SSE = GDP_lm_69_sse, Forecast_UN_SSE = UN_lm_69_sse)

##      Forecast_GDP_SSE Forecast_UN_SSE
## [1,]         528.2449        16681.09

summary(lm1) # use UN to predict GDP

##
## Call:
## lm(formula = GDP ~ UN, data = df[1:56, 3:4])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.024  -7.146  -1.932   8.024  18.411
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 75.42255    4.87117  15.483  < 2e-16 ***
## UN           0.05866    0.01271   4.616 2.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.84 on 54 degrees of freedom
## Multiple R-squared:  0.283,  Adjusted R-squared:  0.2697
## F-statistic: 21.31 on 1 and 54 DF,  p-value: 2.453e-05
```

```r
summary(lm2) # use GDP to predict UN
```

```
##
## Call:
## lm(formula = UN ~ GDP, data = df[1:56, 3:4])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -153.40  -67.99  -17.84   84.76  170.32
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -99.147    102.132  -0.971    0.336
## GDP            4.824      1.045   4.616 2.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 89.23 on 54 degrees of freedom
## Multiple R-squared:  0.283,  Adjusted R-squared:  0.2697
## F-statistic: 21.31 on 1 and 54 DF,  p-value: 2.453e-05
```

Based on sum of squared errors, the first model performs better since it has a way smaller value of SSE. And based on summary tables, both models have a low R-squared of around 26%, which indicates both models did a poor job in explain the variation in the response variable. In first model, both intercept and predictor are significant while in second model, only predictor is significant. Based on those, it is likely that the better method is to have UN as the independent variable and GDP as the dependent variable.