Homework 3

Adham Suliman

October 13, 2018

```
df <- read.csv('Unemployment_GDP_UK.csv',header=T)
colnames(df) <- c('Year','Quarter','UN','GDP')
```

ARIMA modeling:

Use datasets from 1955 to 1968 to build an ARMA or ARIMA models for UN and GDP. Justify why you chose (ARMA or ARIMA) one over the other. Note there will be 2 models, one for UN and another for GDP. Use the chosen UN and GDP models to forecast the UN and the GDP for 1969. Compare your forecasts with the actual values using error = actual - estimate and plot the errors. Calculate the sum of squared error for each UN and GDP models.

Exploratory Analysis

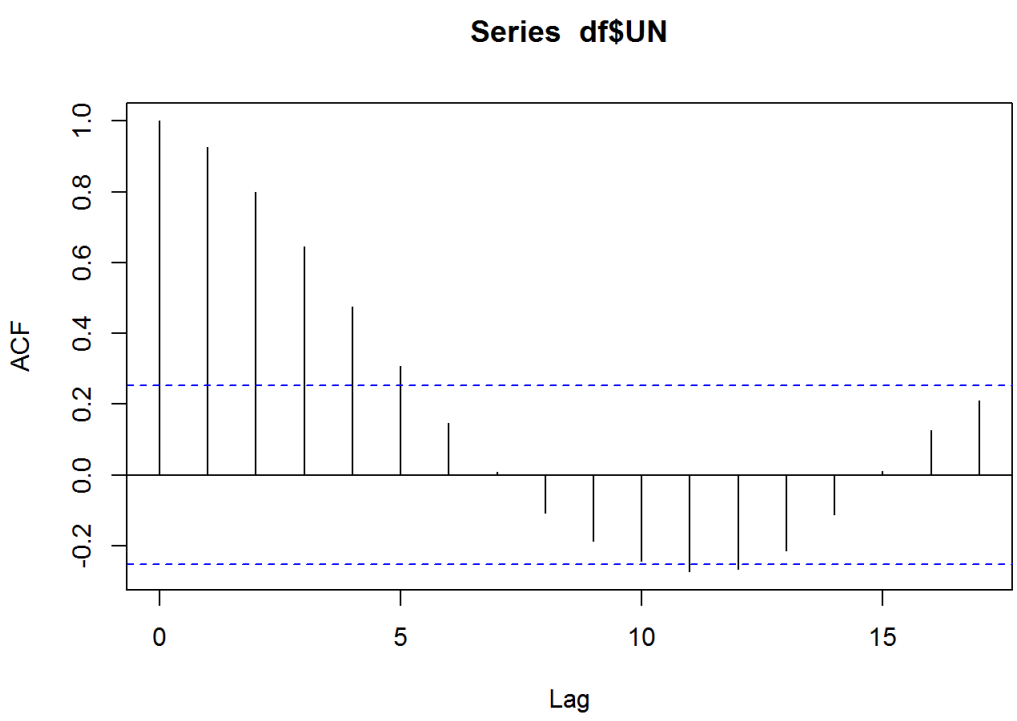
```
adf.test(df$UN)
```

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

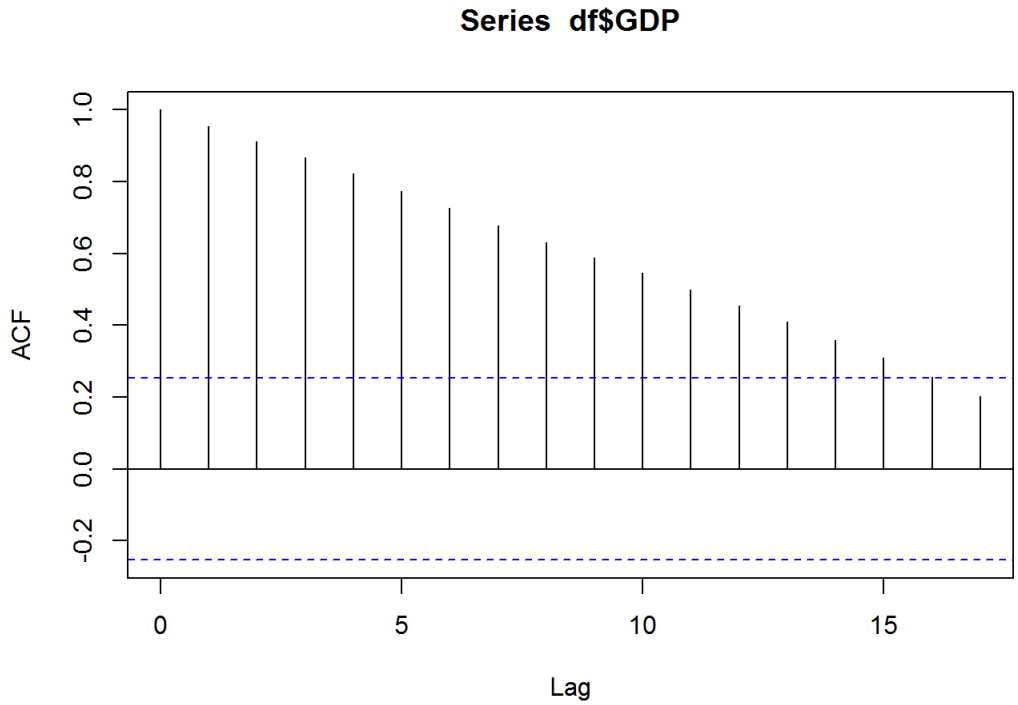
```
adf.test(df$GDP)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  df$GDP
## Dickey-Fuller = -3.1159, Lag order = 3, p-value = 0.1237
## alternative hypothesis: stationary
```

```
acf(df$UN)
```




```
acf(df$GDP)
```



Below, An arima model is taken for both df\(UN and df\)GDP with AICs of 540.47 and 153.15 respectively.

```
auto.arima(df$UN)
```

```
## Series: df$UN
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.6442
## s.e.  0.0969
##
## sigma^2 estimated as 524.7:  log likelihood=-268.24
## AIC=540.47   AICc=540.69   BIC=544.63
```

```
auto.arima(df$GDP)
```

```
## Series: df$GDP
## ARIMA(0,1,0) with drift
##
## Coefficients:
##        drift
##       0.6480
## s.e.  0.1115
##
## sigma^2 estimated as 0.7462:  log likelihood=-74.57
## AIC=153.15   AICc=153.36   BIC=157.3
```

Below, An arma model is taken for both df\(UN and df\)GDP with AICs of 560.44 and 158.19 respectively.

```
summary(arma(df$UN))
```

```
##
## Call:
## arma(x = df$UN)
##
## Model:
## ARMA(1,1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -57.2867 -18.7878  -0.9226  14.2943  75.4961
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1         0.94642     0.04341   21.800  < 2e-16 ***
## ma1         0.51237     0.09441    5.427 5.73e-08 ***
## intercept  25.04069    16.99489    1.473    0.141
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 603.6,  Conditional Sum-of-Squares = 35011.73,  AIC = 560.44
```

```
summary(arma(df$GDP))
```

```
##
## Call:
## arma(x = df$GDP)
##
## Model:
## ARMA(1,1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0305 -0.5146  0.1398  0.5464  2.9897
##
## Coefficient(s):
##             Estimate  Std. Error  t value Pr(>|t|)
## ar1        1.0066049   0.0091492  110.021   <2e-16 ***
## ma1       -0.0055129   0.1232925   -0.045    0.964
## intercept -0.0002144   0.9045636    0.000    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 0.7398,  Conditional Sum-of-Squares = 42.91,  AIC = 158.19
```

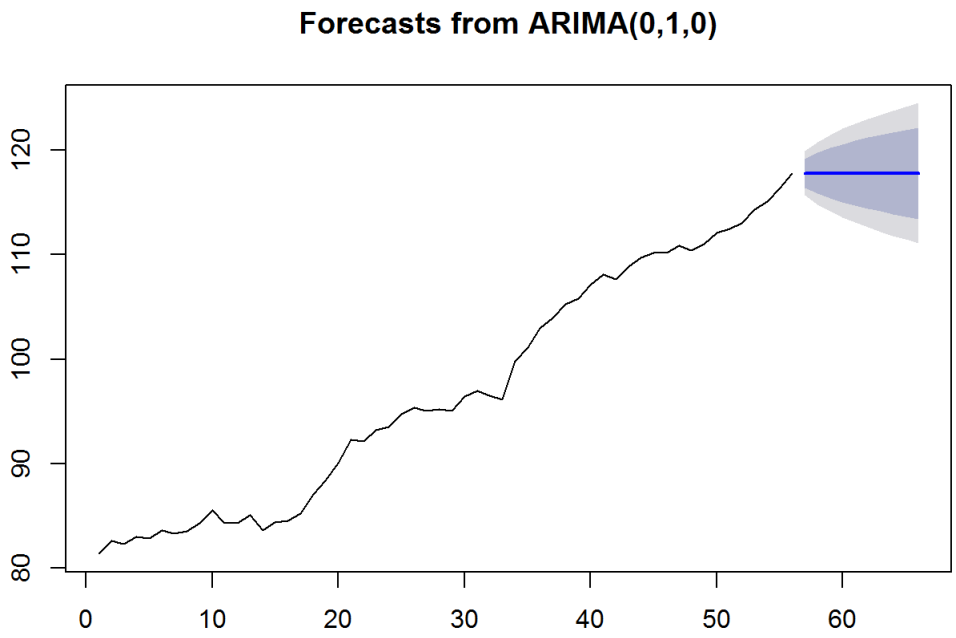
The arima model allows for lower AICs for both GDP and UN.

Forecast for GDP is done below.

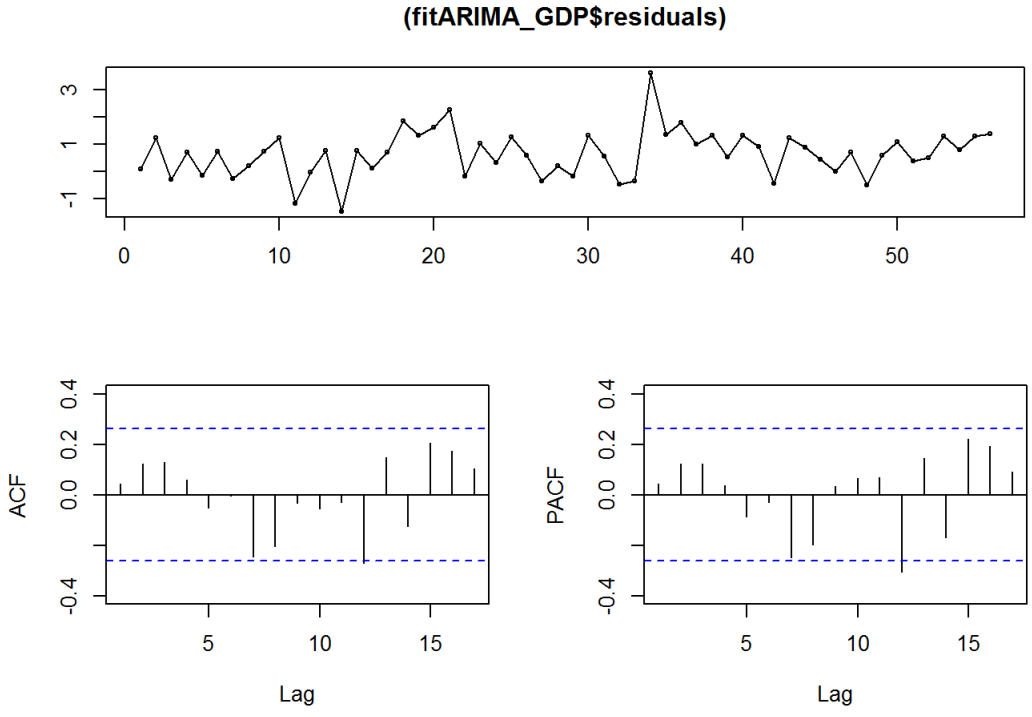
```
df1 <- subset(df, Year!=1969)
df2 <- subset(df, Year==1969)
fitARIMA_GDP <- arima(df1$GDP, order=c(0,1,0))
summary(fitARIMA_GDP)
```

```
##
## Call:
## arima(x = df1$GDP, order = c(0, 1, 0))
##
##
## sigma^2 estimated as 1.168:  log likelihood = -82.31,  aic = 166.63
##
## 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
```

```
plot(forecast(fitARIMA_GDP))
```




```
tsdisplay((fitARIMA_GDP$residuals))
```

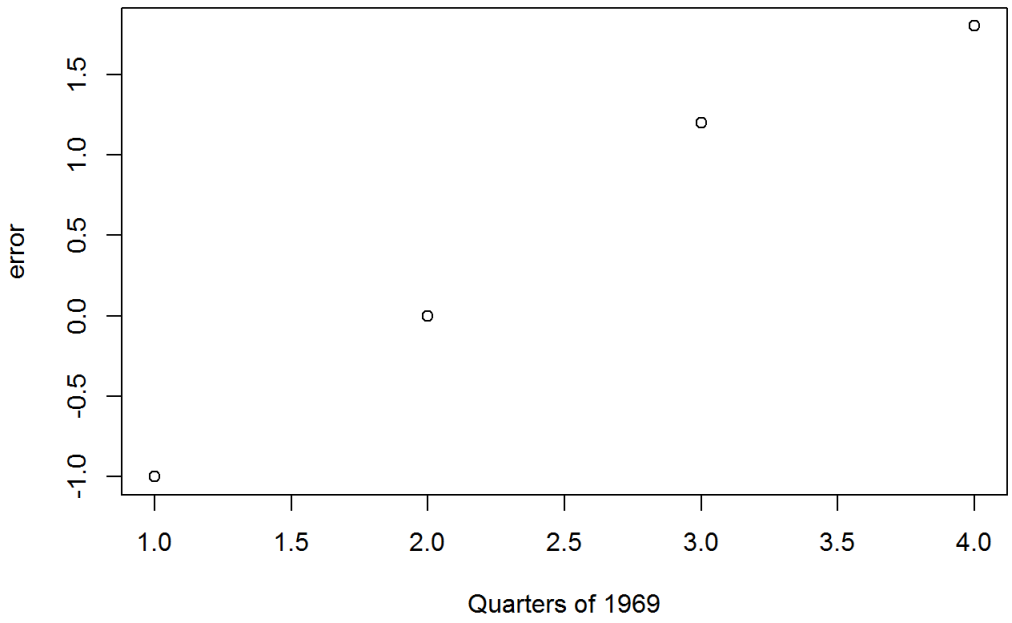


Compare your forecasts with actual values using error=actual - estimate and plot

```
predict(fitARIMA_GDP,n.ahead = 4)
```

```
## $pred
## Time Series:
## Start = 57
## End = 60
## Frequency = 1
## [1] 117.8 117.8 117.8 117.8
##
## $se
## Time Series:
## Start = 57
## End = 60
## Frequency = 1
## [1] 1.080793 1.528473 1.871989 2.161587
```

```
df4 <- as.data.frame(cbind("GDP"=df2$GDP,"pred"=c(117.8, 117.8, 117.8, 117.8)))
df4$error <- (df4[,1]-df4[,2])
plot(df4$error,
     xlab ="Quarters of 1969",
     ylab = "error")
```




```
df4$sum_of_squares <- (df4[,1]-df4[,2])^2
library('dplyr')
```

```
##
## Attaching package: 'dplyr'
```

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

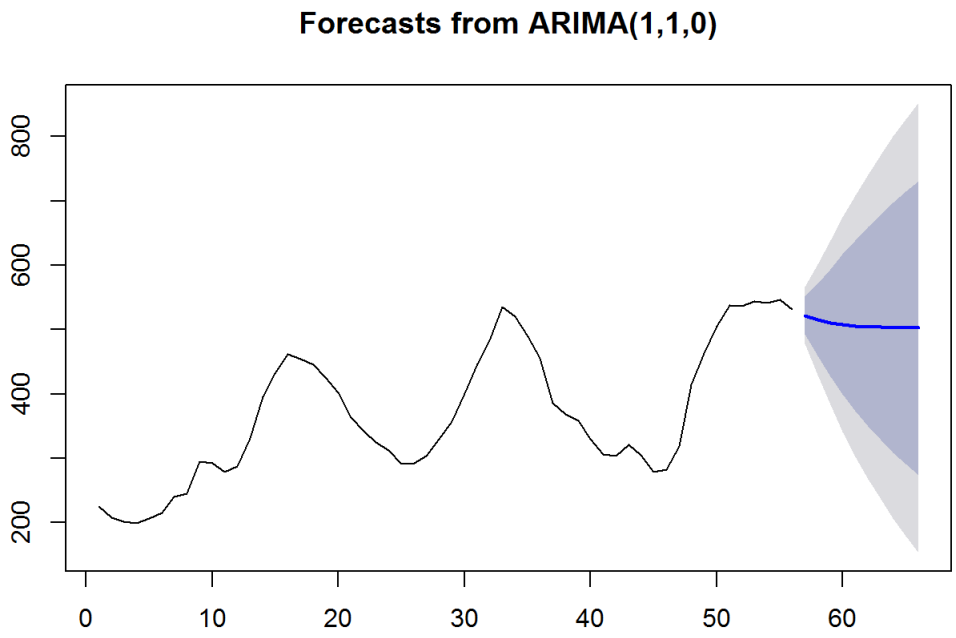
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df4%>%
  summarize(sum(sum_of_squares))
```

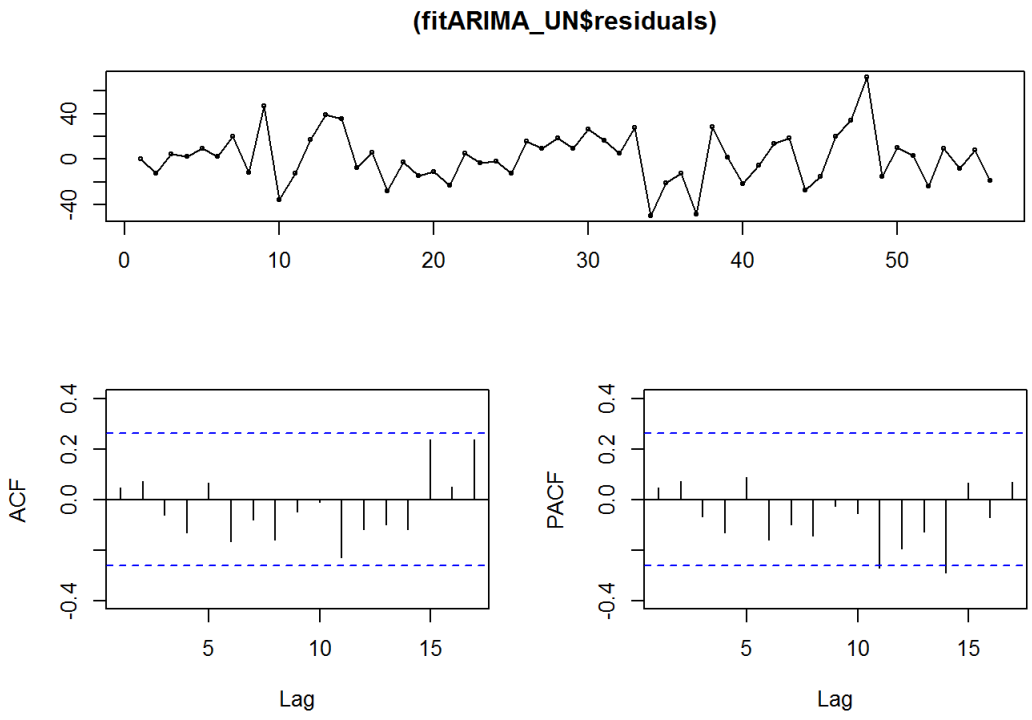
```
##   sum(sum_of_squares)
## 1                5.68
```

Forecast for UN is done below.

```
df1 <- subset(df, Year!=1969)
fitARIMA_UN <- arima(df1$UN, order=c(1,1,0))
plot(forecast(fitARIMA_UN))
```




```
tsdisplay((fitARIMA_UN$residuals))
```

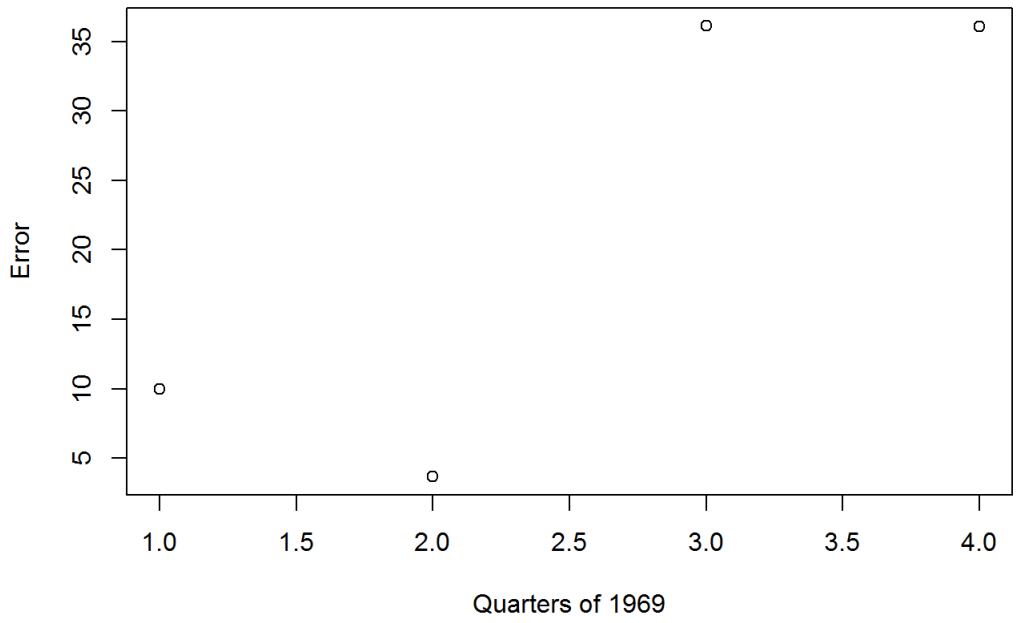


Compare your forecasts with actual values using error=actual - estimate and plot

```
predict(fitARIMA_UN,n.ahead = 4)
```

```
## $pred
## Time Series:
## Start = 57
## End = 60
## Frequency = 1
## [1] 522.0004 515.3343 510.8904 507.9280
##
## $se
## Time Series:
## Start = 57
## End = 60
## Frequency = 1
## [1] 22.70484 44.12973 65.15219 85.04261
```

```
df4 <- as.data.frame(cbind("GDP"=df2$UN,"pred"=c(522.0004, 515.3343, 510.8904, 507.9280)))
df4$error <- (df4[,1]-df4[,2])
plot(df4$error,
     xlab ="Quarters of 1969",
     ylab = "Error")
```




```
df4$sum_of_squares <- (df4[,1]-df4[,2])^2
library('dplyr')
df4%>%
  summarize(sum(sum_of_squares))
```

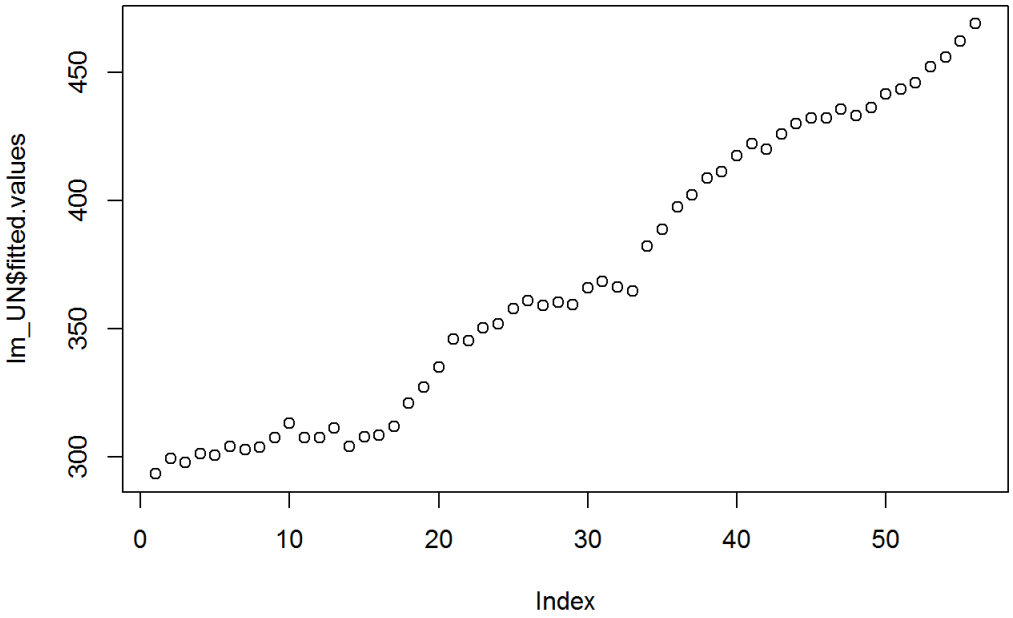
```
##   sum(sum_of_squares)
## 1            2718.522
```

Regression - build regression models that use:

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. 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. 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 ?

Just put pedicted against true, no reason to bring in the before data

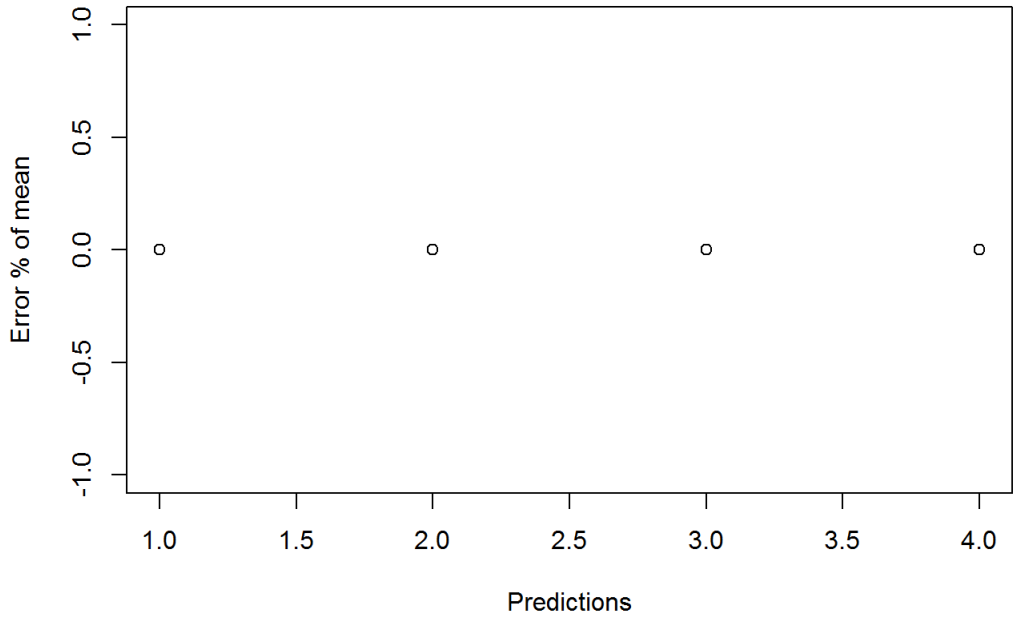
```
lm_UN <- lm(UN~GDP,df1)
lm_UN_pred <- predict(lm_UN,newdata=df2)
df_UN <- df
plot(lm_UN$fitted.values)
```




```
df_UN$UN[57:60] <-lm_UN_pred
summary(lm(UN~GDP, df1))
```

```
##
## Call:
## lm(formula = UN ~ GDP, data = df1)
##
## 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
```

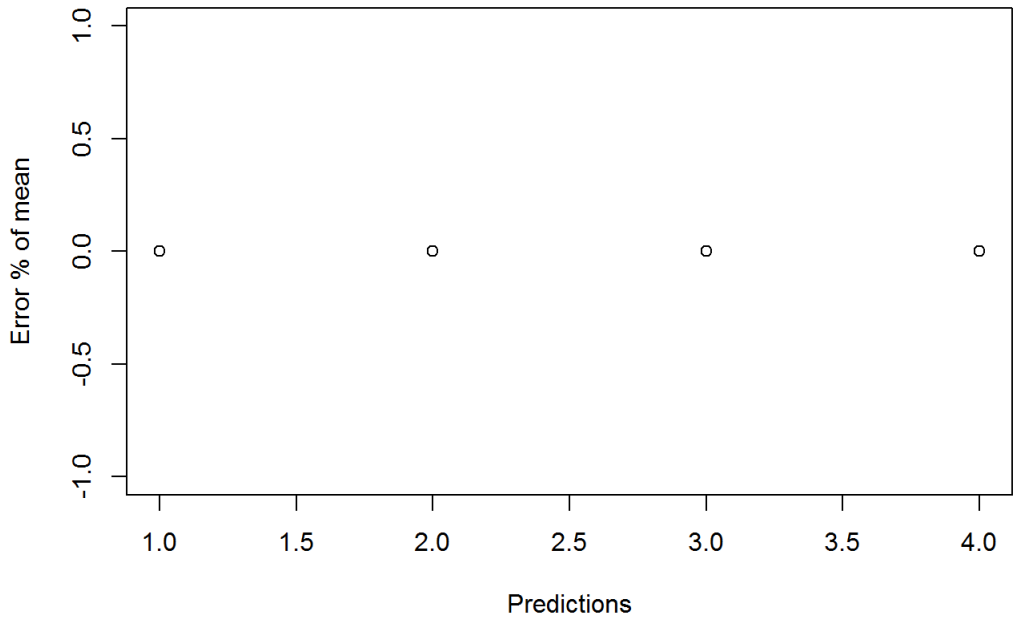
```
plot((df_UN$UN[57:60] -lm_UN_pred)/df_UN$UN[57:60],
  xlab = "Predictions",
  ylab = "Error % of mean")
```



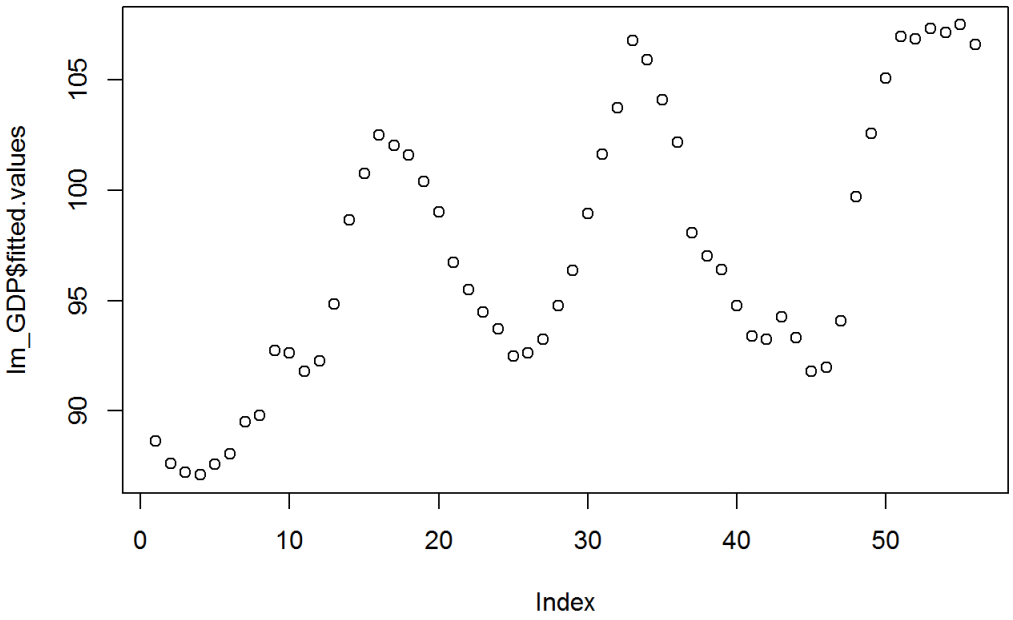

```
sum(((df_UN$UN[57:60] -lm_UN_pred)^2)/df_UN$UN[57:60])
```

```
## [1] 0
```

```
plot(((df_UN$UN[57:60] -lm_UN_pred)^2)/df_UN$UN[57:60],
  xlab = "Predictions",
  ylab = "Error % of mean")
```



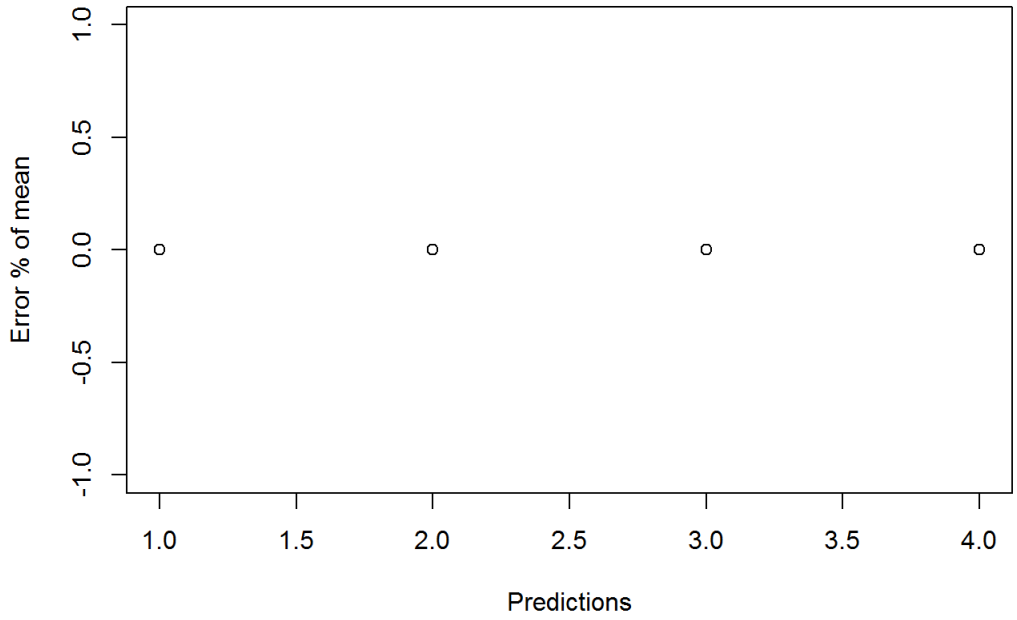

```
lm_GDP <- lm(GDP~UN,df1)
lm_GDP_pred <- predict(lm_GDP,newdata=df2)
df_GDP <- df
plot(lm_GDP$fitted.values)
```




```
df_GDP$GDP[57:60] <-lm_GDP_pred
summary(lm(GDP~UN,df_GDP))
```

```
##
## Call:
## lm(formula = GDP ~ UN, data = df_GDP)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.0242  -6.4761  -0.7599   7.2314  18.4109
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 75.42255    4.47497  16.854  < 2e-16 ***
## UN           0.05866    0.01132   5.182  2.9e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.495 on 58 degrees of freedom
## Multiple R-squared:  0.3165, Adjusted R-squared:  0.3047
## F-statistic: 26.85 on 1 and 58 DF,  p-value: 2.896e-06
```

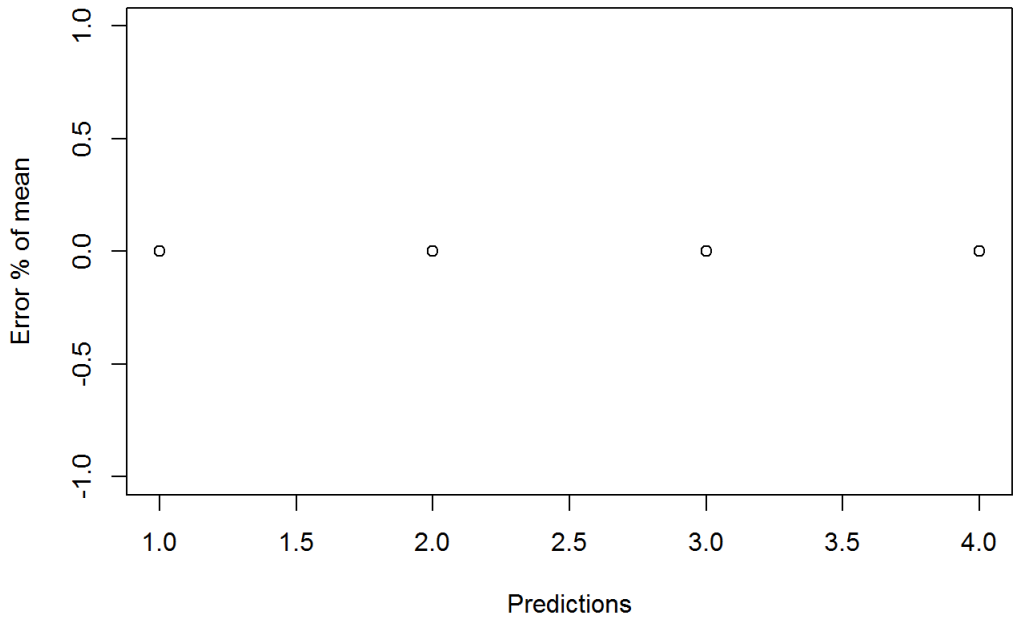
```
plot((df_GDP$GDP[57:60] -lm_GDP_pred)/df_GDP$GDP[57:60],
  xlab = "Predictions",
  ylab = "Error % of mean")
```




```
sum(((df_GDP$GDP[57:60] -lm_GDP_pred)^2)/df_GDP$GDP[57:60])
```

```
## [1] 0
```

```
plot(((df_GDP$GDP[57:60] -lm_GDP_pred)^2)/df_GDP$GDP[57:60],
  xlab = "Predictions",
  ylab = "Error % of mean")
```



Since the sum of square error divided by actuals for the GDP model are smaller than the UN model, UN should be the independent variable, while GDP the de dependent to produce the most accurate model. This is possible because the linear model for lm_GDP does a better job at predicting future values.

```
sum(((df_UN$UN[57:60] -lm_UN_pred)^2)/df_UN$UN[57:60])
```

```
## [1] 0
```

```
sum(((df_GDP$GDP[57:60] -lm_GDP_pred)^2)/df_GDP$GDP[57:60])
```

```
## [1] 0
```