Problem_set_1_solutions_2020

February 25, 2020

1 Problem Set 1 - Solutions

4. Forecasting Portugal GDP annual growth rate

First, lets download the data using quantmod:

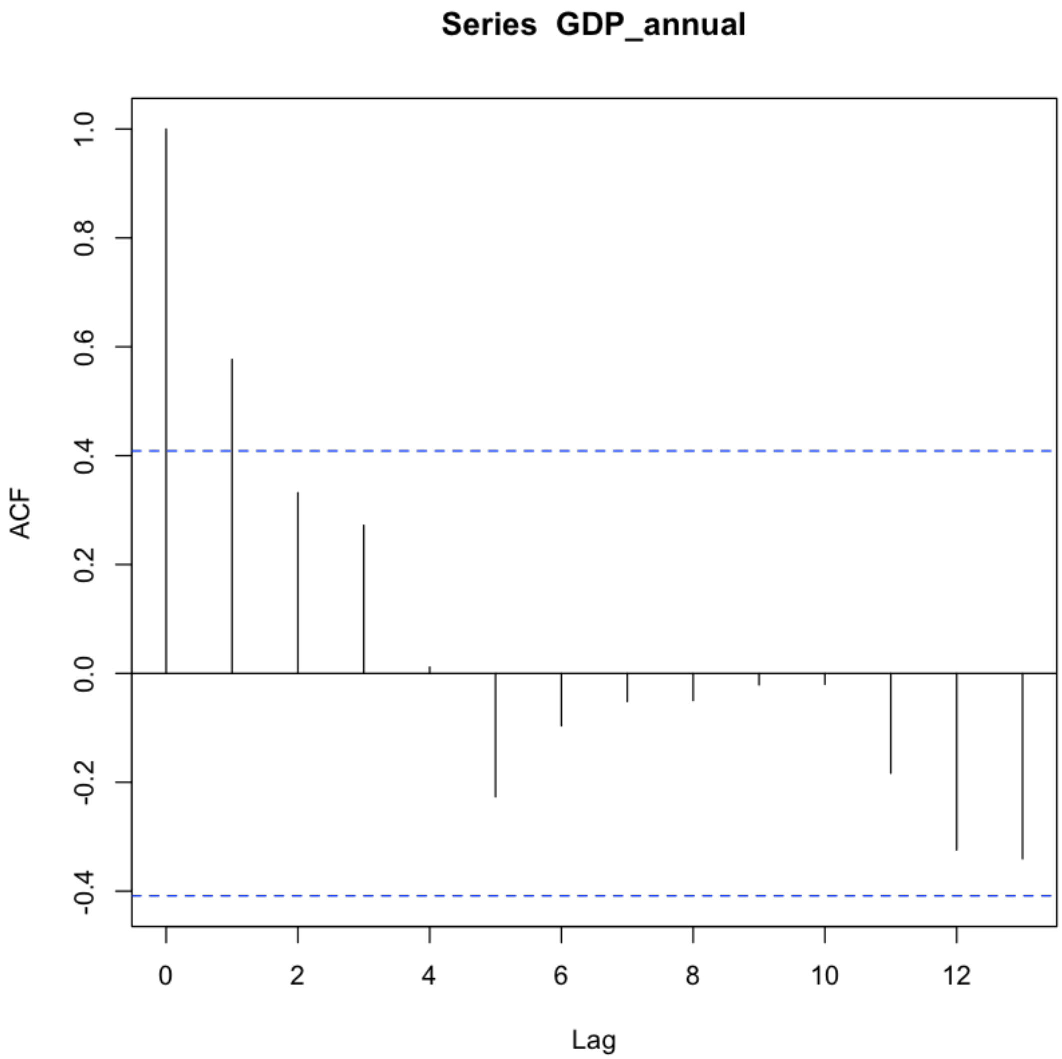
[5]:
```
library(quantmod)
getSymbols("NAEXKP01PTA657S", src="FRED")
names(NAEXKP01PTA657S) = "GDP_annual"
GDP_annual = NAEXKP01PTA657S
```

'NAEXKP01PTA657S'

To check whether the series is stationary or not, lets first plot it and then plot its autocorrelation function

[4]:
```
plot(GDP_annual)
```

[5]:
```
acf(GDP_annual)
```



We can conclude that the series looks stationary as its first and second moments seem contant over the sample period.

In order to identify an ARMA model, we need to compute the information criteria for all possible specifications and choose the one that minimizes that criteria. Here, you can compute the Akaike information criteria to all combinations via a loop or use the `auto.arima()` function from the forecast library:

[9]:
```
library(forecast)
arma1 = auto.arima(GDP_annual, ic = "aic")
summary(arma1)
```

```
Series: GDP_annual
ARIMA(1,0,0) with non-zero mean
```

```
Coefficients:
         ar1    mean
      0.5815  1.5514
s.e.  0.1651  0.8657

sigma^2 estimated as 3.653:  log likelihood=-46.69
AIC=99.39   AICc=100.65   BIC=102.79

Training set error measures:
                     ME     RMSE      MAE      MPE     MAPE      MASE
Training set -0.0652033 1.826219 1.494489 15.42618 98.02314 0.8789918
                   ACF1
Training set -0.0160883
```

Hence, the model that minimizes the AIC is an ARMA(1,0). That is, a model with only an autoregressive component. In this case the identification and model estimation is done all in one step.

Also, since it is a simple AR(1), we can simply check the coefficient on the ar1 component to check for stability. Since, in this case the homogeneous solutions is given by: $y_t^h = A a_1 ^t $. The solution is stable iff $|a_1| < 1$. In this case, $a_1 = 0.5815$ and so the ARMA(1,0) is stable.

Now, we will show how to use a loop to get a table with AIC for different model specifications:

[18]:
```
INFO = matrix(1,6,6) # Create matrix for storing AIC results



# Save AIC of each model estimated, note i is the lag on AR component and j is␣
↪the lag on the MA component
for (i in 0:5){
    for (j in 0:5){
    INFO[i+1,j+1] = arima(GDP_annual, order = c(i,0,j))$aic
    }
}
```

[21]:
```
INFO # remember that first row is the LAG 0 on AR component
```

A matrix: 6 × 6 of type dbl

106.88282	101.1851	102.4924	102.2526	101.2937	103.1562
99.38727	101.3013	103.1819	102.9473	103.0645	104.0114
101.33754	103.3088	104.9263	102.6937	105.0346	105.0242
102.71758	104.3322	104.1788	106.0303	107.0331	106.3402
103.02052	104.3580	104.0448	105.9807	106.5687	108.4797
103.44346	104.9340	106.9126	108.8148	108.3502	110.3231

What is the model that minimizes the AIC?

[28]:
```
which(INFO == min(INFO), arr.ind = TRUE)
```

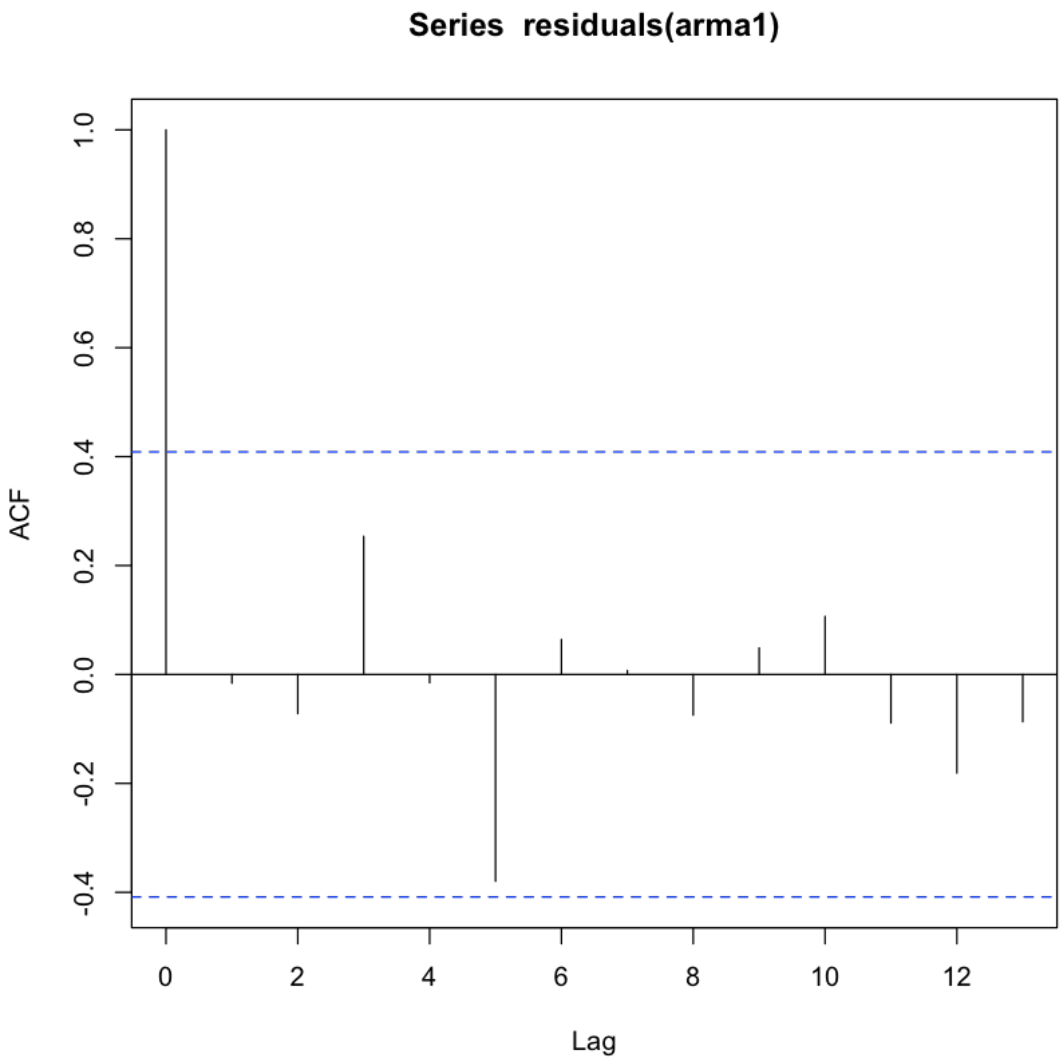
A matrix: 1 × 2 of type int

row	col
2	1

That corresponds to Lag 1 for AR component and Lag 0 for MA component.

Next we check if the model is adquate. We need to check the residuals.

[10]:
```
acf(residuals(arma1))
```



[11]:
```
Box.test(residuals(arma1),type="Ljung",lag=12)
```

```
Box-Ljung test

data:  residuals(arma1)
X-squared = 9.6602, df = 12, p-value = 0.6457
```

We do not reject no serial correlation and as a consequence the model is deemed adequate. Lets use the model to make forecasts!

[13]:
```
forecast(auto.arima(GDP_annual, ic = "aic"), h=2)
```

```
   Point Forecast      Lo 80    Hi 80     Lo 95    Hi 95
24       2.068844 -0.3804637 4.518151 -1.677049 5.814736
25       1.852284 -0.9810175 4.685586 -2.480877 6.185445
```

The point forecasts are reasonalby close to the ones published by the Bank of Portugal. If we round the forcasts to one decimal place we have:

- 2019: 2.1 Vs. 2.0
- 2020: 1.9 Vs. 1.7

However, there is a high degree of uncertainty as the forecast bands are quite far apart. This is somewhat expected as we have a small number of observations.