Time series analysis of the GDP per capita of UK

Ding Xu

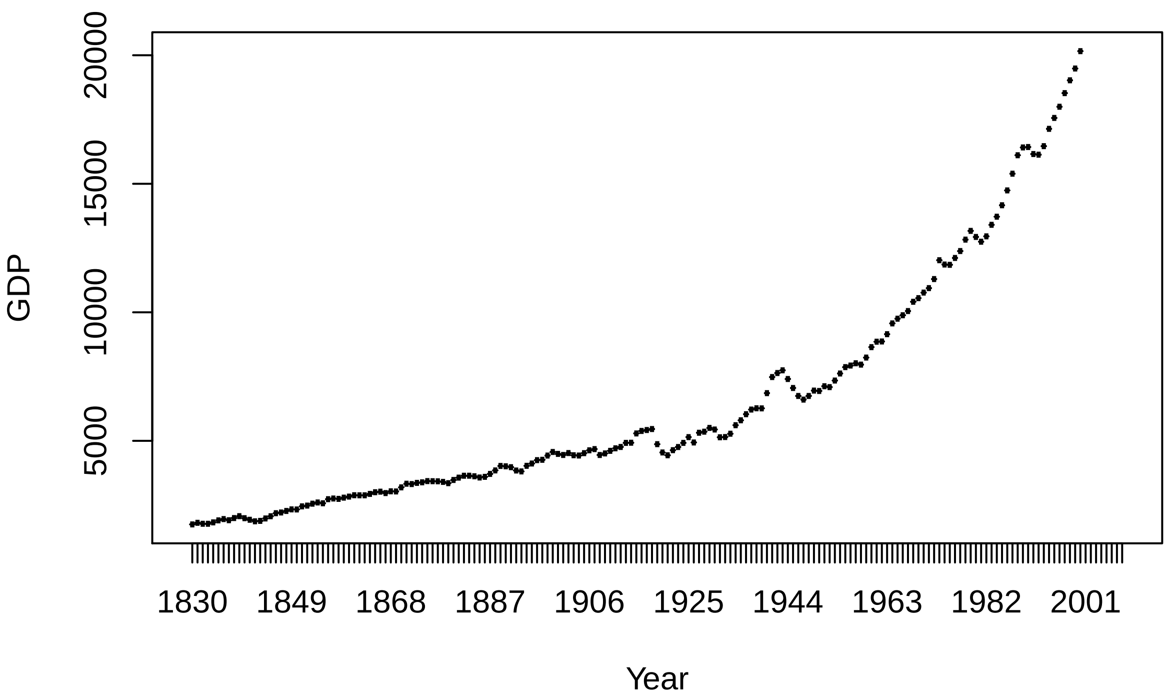
2019/4/20

1.About the data

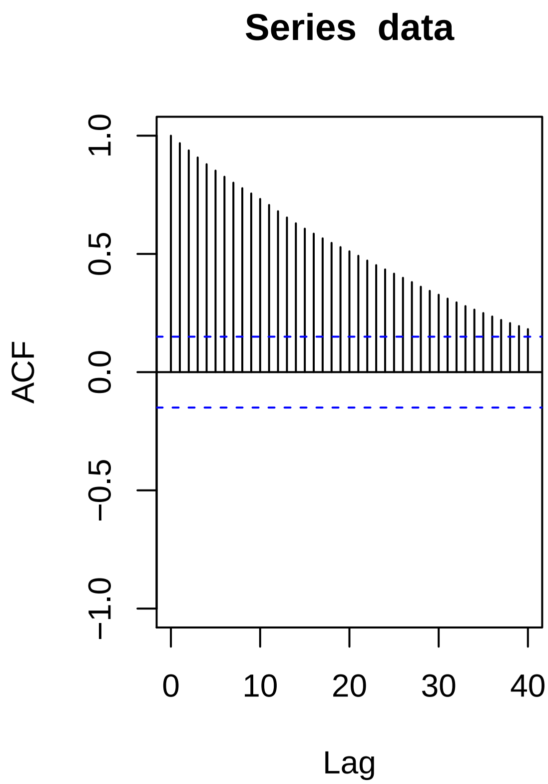
I choose the time series data is GDP per capita of UK from 1830 to 2000. I got the data from "https://datamarket.com/data/set/1cf1/gdp-per-capita-old-version#!ds=1cf1!r2r=5t&display=line". I want to find the increasing trend of the GDP per capita.

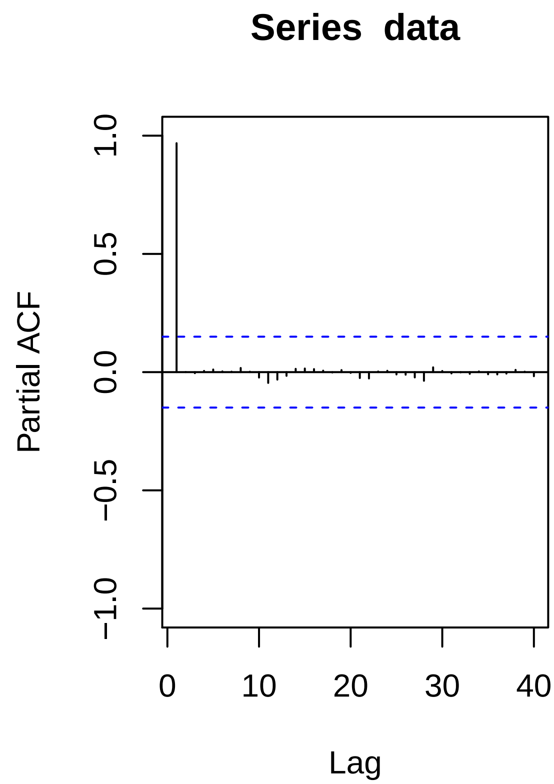
2.Plot the data

```
setwd("/Users/cooper/Documents/Study/STT844/FINAL/")
library(itsmr)
data=read.csv("UKGDP.csv")
test=data[1:178,2]
data=data[1:171,]
colnames(data)=c("Year","GDP")
plot(data)
```




```
data=data[,2]
par(mfrow=c(1,2))
acf(data,40,ylim=c(-1,1))
pacf(data,40,ylim=c(-1,1))
```





From the data plot, i think it is exponential data, so a log transformation is necessary.

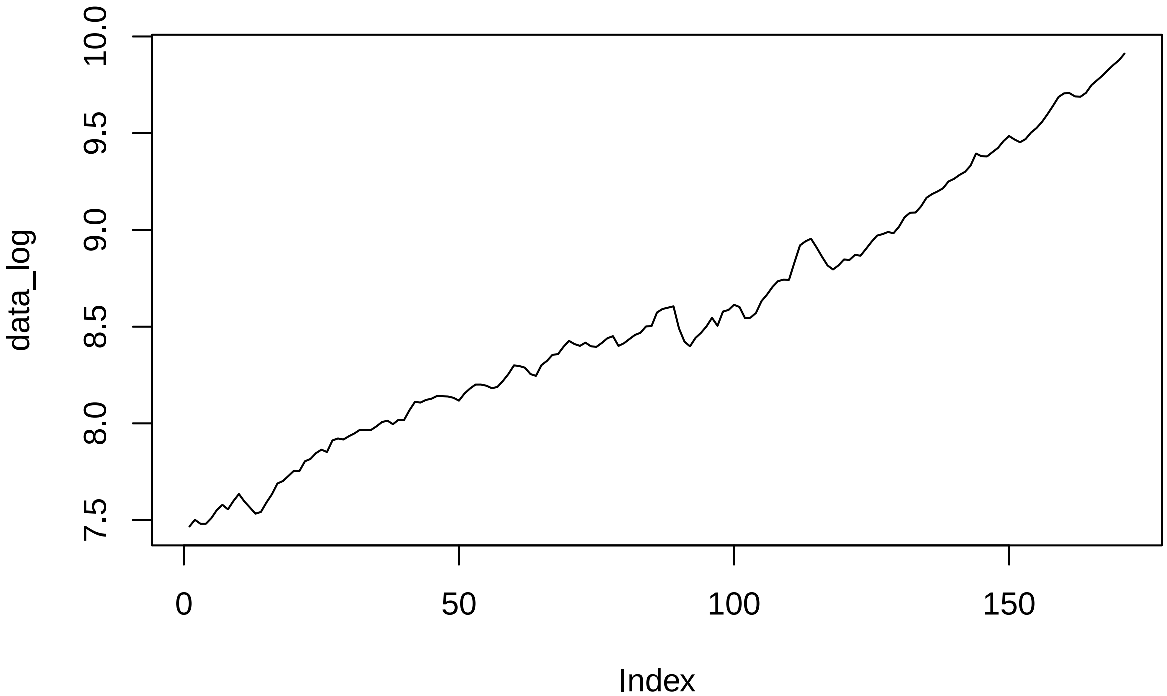
From the ACF plot, i think a differencing transformation is needed.

Then i do the transformation one by one.

3.Transformation

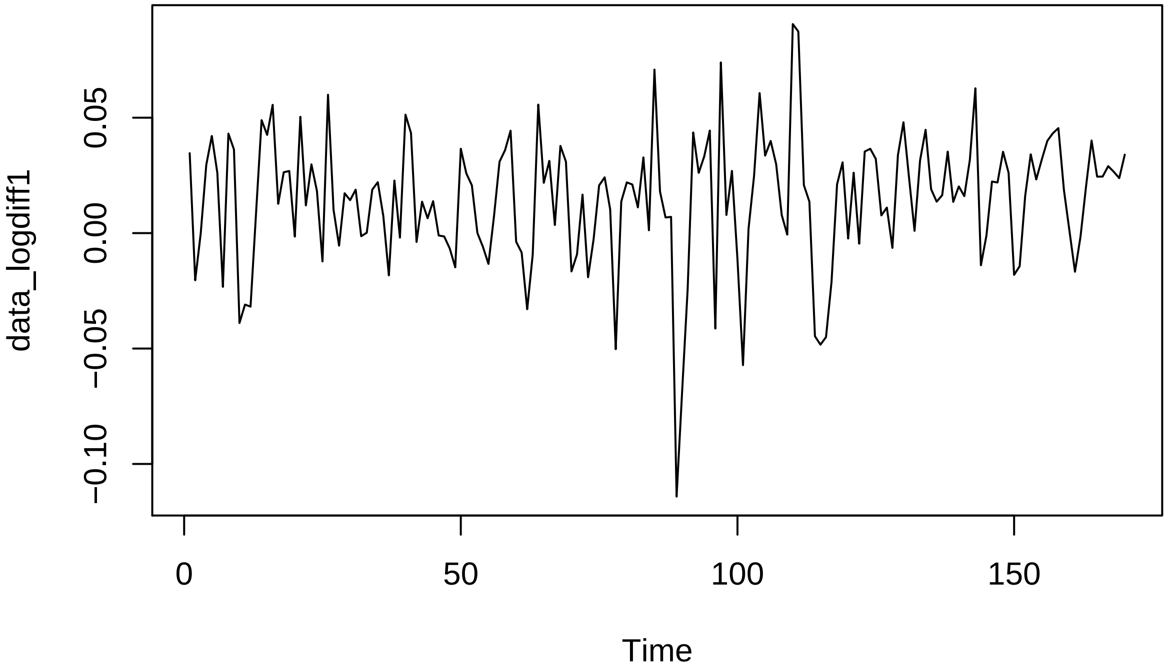
First, i do the log transformation.

```
data_log=log(data)
plot(data_log,type = "l")
```



It is obviously that the data doesn't have exponential feature anymore. But the trend still exists, i do the differencing transformation next.

```
data_logdiff1=ts(diff(data_log,1))
plot(data_logdiff1)
```

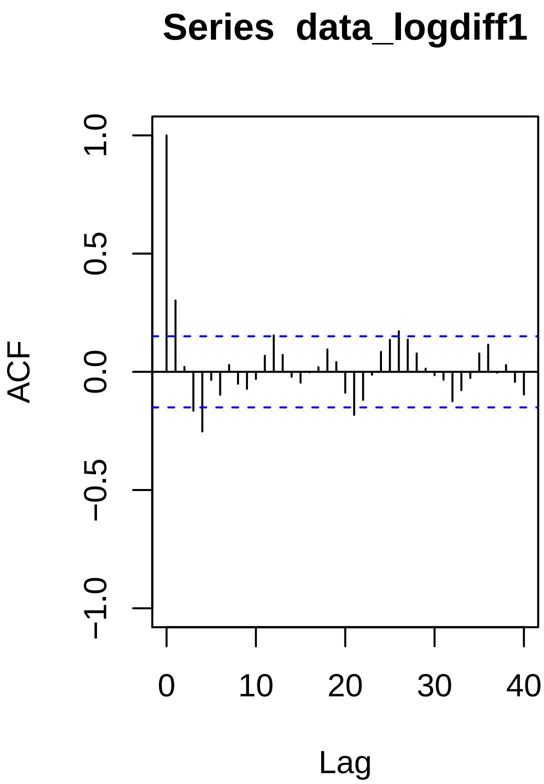


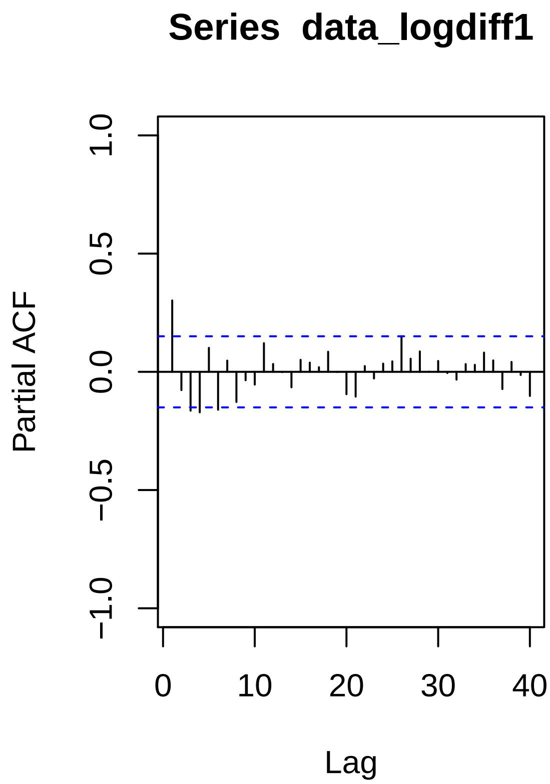
From the data plot above, the trend has eliminated. Then we can consider this transformed data as stationary data and use ARMA model to fit.

4.Fit the data

4.1 Decide the p&q

```r
par(mfrow=c(1,2))
acf(data_logdiff1,40,ylim=c(-1,1))
pacf(data_logdiff1,40,ylim=c(-1,1))
```






```r
autofit(data_logdiff1,p=0:5,q=0:5)
```

```
## $phi
## [1] -0.4475672  0.1999666 -0.1265213 -0.2946879
##
## $theta
## [1] 0.7904837
##
## $sigma2
## [1] 0.000648412
##
## $aicc
## [1] -752.4054
##
## $se.phi
## [1] 0.10822358 0.08480296 0.08088108 0.07433682
##
## $se.theta
## [1] 0.09169904
```

- The ACF plot shows that $q = 4$ as the $ACF = 0$ after lags 4

- The PACF plot shows that $p = 5$ as the $PACF = 0$ after lags 5
- The minimum aicc model is $p = 4$, $q = 1$ with $minAICC = -752.4$

Combining the three results above, we can find the actually the ACF can be considered as 0 after lags 1 and PACF can be considered as 0 after lags 4. So i decide to use $p = 4, \ q = 1$

4.2 Two ways to estimate the parameters

4.2.1 Maximum likelihood

```
M=c("log","trend",1)
e=Resid(data,M)
fit.ml=arma(e,p=4,q=1)
fit.ml
```

```
## $phi
## [1]  0.43616853  0.71407804 -0.19942891 -0.02896586
##
## $theta
## [1] 0.8879376
##
## $sigma2
## [1] 0.0006908794
##
## $aicc
## [1] -743.5021
##
## $se.phi
## [1] 0.09927659 0.12075040 0.09483083 0.08009792
##
## $se.theta
## [1] 0.06407471
```

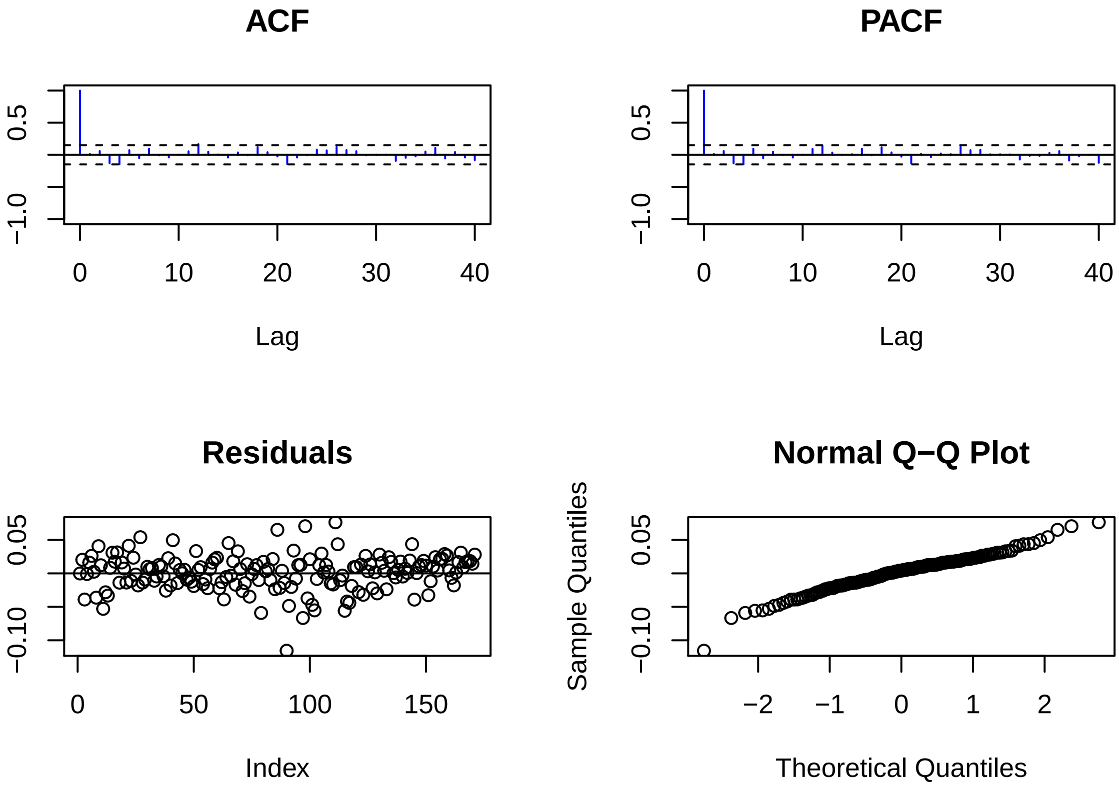
Using the maximum likelihood estimator, we can get that $log(GDP)_t$ fits the $ARIMA(4, 1, 1)$ with

$\hat{\phi}_{ML} = (0.43616853, 0.71407804, -0.19942891, -0.0289658)'$
$\hat{\theta}_{ML} = 0.8879376$
$\hat{\sigma}^2_{ML} = 0.000690879$

Then do the residual test to check if the residual is iid noise.

```
ee.ml=Resid(data,M,fit.ml)
test(ee.ml)
```

```
## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)     20.57    0.4229
## McLeod-Li Q                Q ~ chisq(20)     32.86     0.035 *
## Turning points T  (T-112.7)/5.5 ~ N(0,1)       114    0.8079
## Diff signs S         (S-85)/3.8 ~ N(0,1)        89    0.2907
## Rank P         (P-7267.5)/374.3 ~ N(0,1)      7768    0.1812
```



According to the test above, the residual of maximum likelihood estimate is iid noise.

4.2.2 Preliminary estimation: Hannan-Rissanen algorithms

```
fit.hannan=hannan(e,p=4,q=1)
fit.hannan
```

```
## $phi
## [1]  1.4993688 -0.5377175 -0.1461222  0.1738734
##
## $theta
## [1] -0.1838148
##
## $sigma2
## [1] 0.0006907752
##
## $aicc
## [1] -742.6393
##
## $se.phi
## [1] 0.31470960 0.37638251 0.15461240 0.08311773
##
## $se.theta
## [1] 0.3344455
```

Using the Hannan-Rissanen algorithms, we can get that $log(GDP)_t$ fits the $ARIMA(4,1,1)$ with

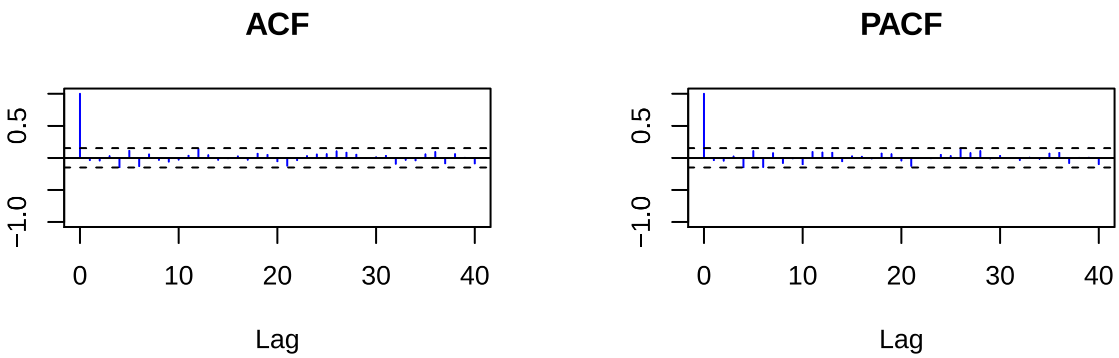
$\hat{\phi}_{HR} = (1.4993688, -0.5377175, -0.1461222, 0.1738734)'$
$\hat{\theta}_{HR} = -0.1838148$

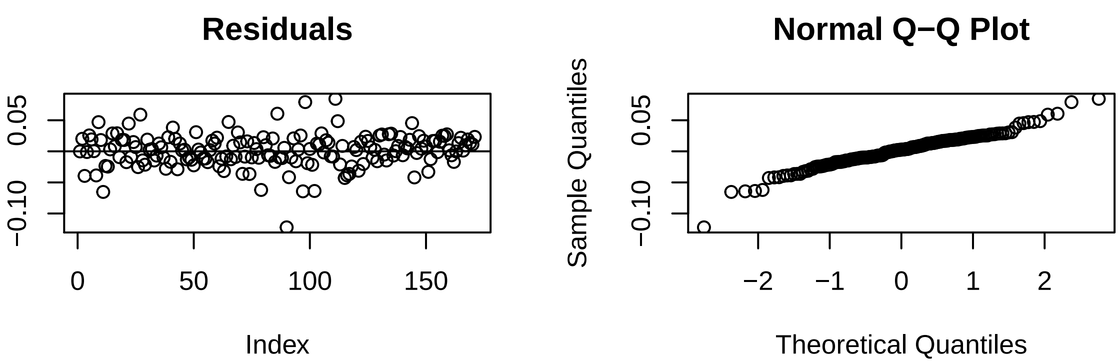
$\hat{\sigma}^2_{HR} = 0.0006907752$

Then do the residual test to check if the residual is iid noise.

```
ee.hannan=Resid(data,M,fit.hannan)
test(ee.hannan)
```

```
## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)     17.55    0.6167
## McLeod-Li Q                Q ~ chisq(20)     21.95    0.3432
## Turning points T  (T-112.7)/5.5 ~ N(0,1)       114    0.8079
## Diff signs S         (S-85)/3.8 ~ N(0,1)        83    0.5973
## Rank P          (P-7267.5)/374.3 ~ N(0,1)      7809     0.148
```





According to the test above, the residual of hannan-rissanen algorithms is iid noise.

5. Prediction

Using the GDP per capita of UK from 2001 to 2007 to compare with the forecasting data from maximum likelihood estimate and hannan-rissanen algorithms

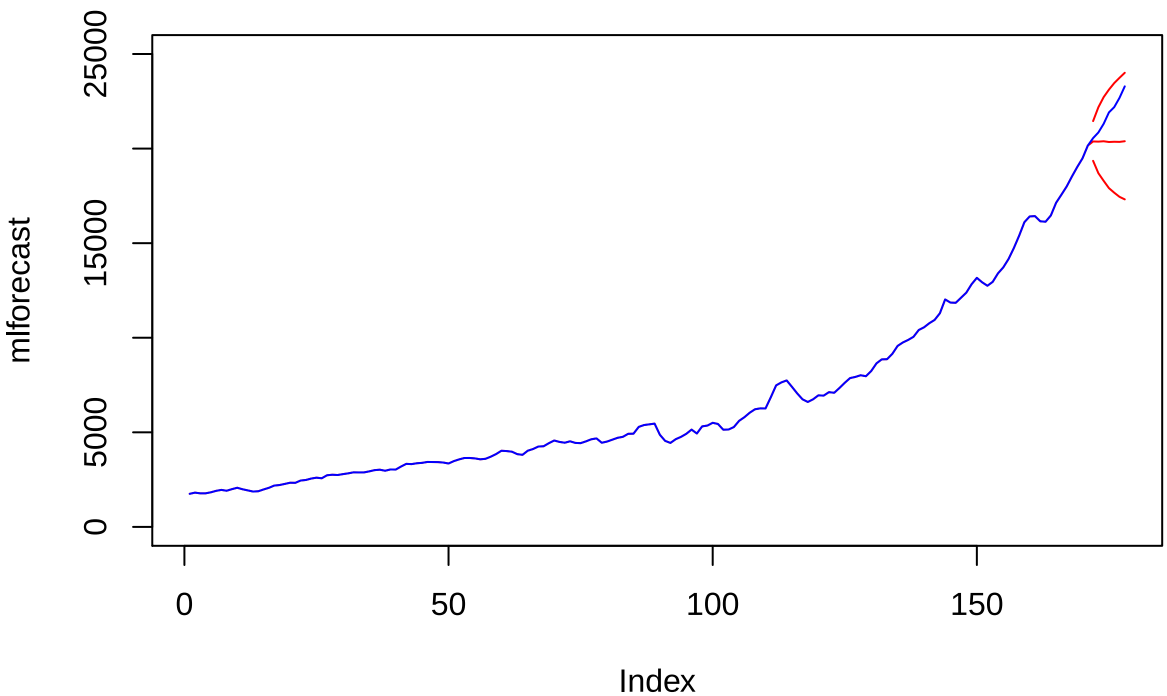
5.1 Maximum likelihood

```
forecast(data,M,fit.ml,h=7,opt=1)
```

```
##   Step      Prediction    Lower Bound    Upper Bound
##      1        20376.58        19353.4       21453.85
```

```
##      2        20367.66        18698.91        22185.33
##      3        20388.27        18295.06        22720.97
##      4        20347.73        17909.89        23117.39
##      5        20360.71        17668.72        23462.85
##      6        20354.09        17449.99        23741.51
##      7         20388.5        17314.25        24008.61
```

```
ml=forecast(data,M,fit.ml,h=7,opt=0)
mlforecast=c(data,ml$pred)
plot(mlforecast,type = "l",col="red",ylim = c(0,25000))
points(172:178,ml$l,col="red",type="l")
points(172:178,ml$u,col="red",type="l")
lines(test,col="blue")
```



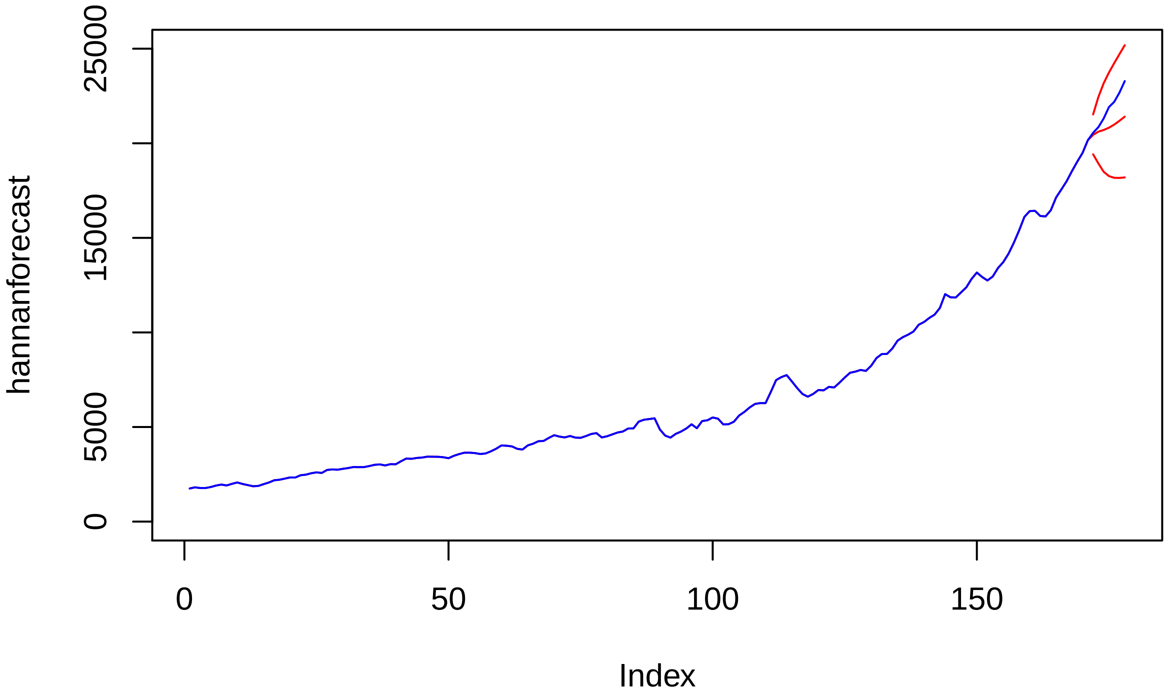
5.2 Preliminary estimation: Hannan-Rissanen algorithms

```
forecast(data,M,fit.hannan,h=7,opt=1)
```

```
##   Step     Prediction     Lower Bound     Upper Bound
##      1        20443.86        19417.39         21524.6
##      2        20614.41        18932.21        22446.07
##      3        20701.53        18494.48        23171.96
##      4        20823.11        18265.19        23739.26
##      5        20987.75        18173.78        24237.44
##      6        21190.72        18168.65        24715.48
##      7        21408.19        18195.88         25187.6
```

```
hannan=forecast(data,M,fit.hannan,h=7,opt=0)
hannanforecast=c(data,hannan$pred)
plot(hannanforecast,type = "l",col="red",ylim = c(0,25000))
points(172:178,hannan$l,col="red",type="l")
```

```r
points(172:178,hannan$u,col="red",type="l")
lines(test,col="blue")
```



6.Findings

Comparing the forecasts of both methods, the Hannan-Rissanen estimation is much better than the maximum likelihood estimation. But neither of them is forecasting very well with the further estimation, and the confidence interval of both estimation prove that.