Time Series Analysis - ARIMA Models

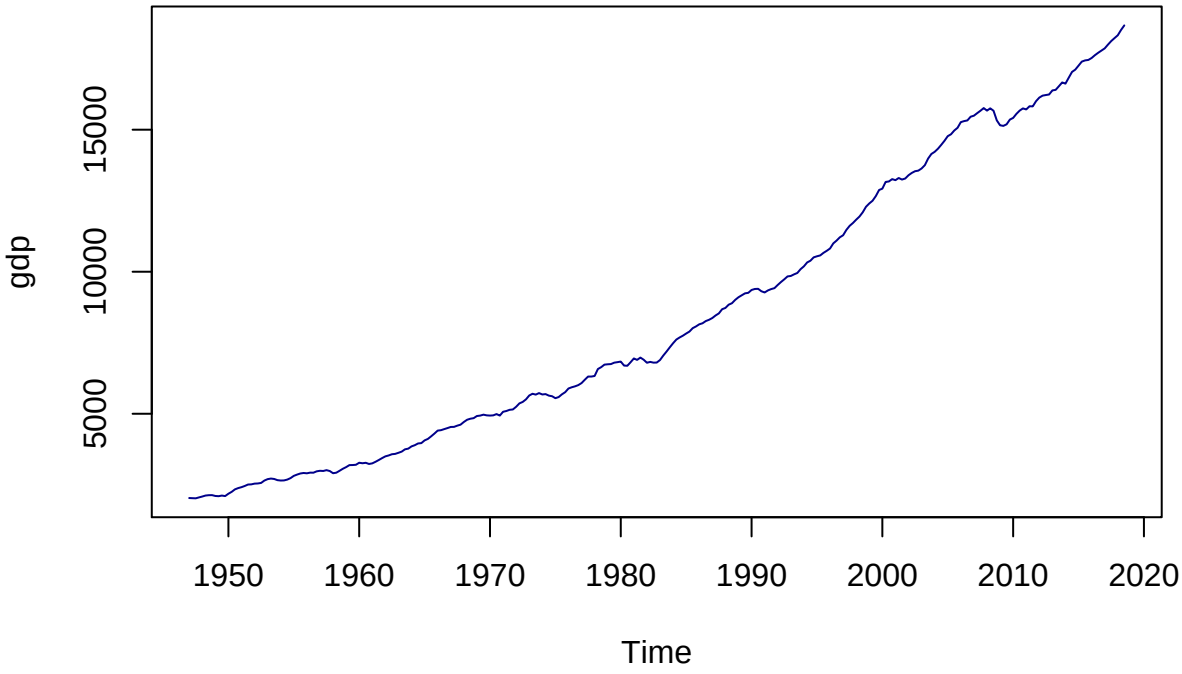
Abdullahi Hussein

03/09/2021

Exploring the data

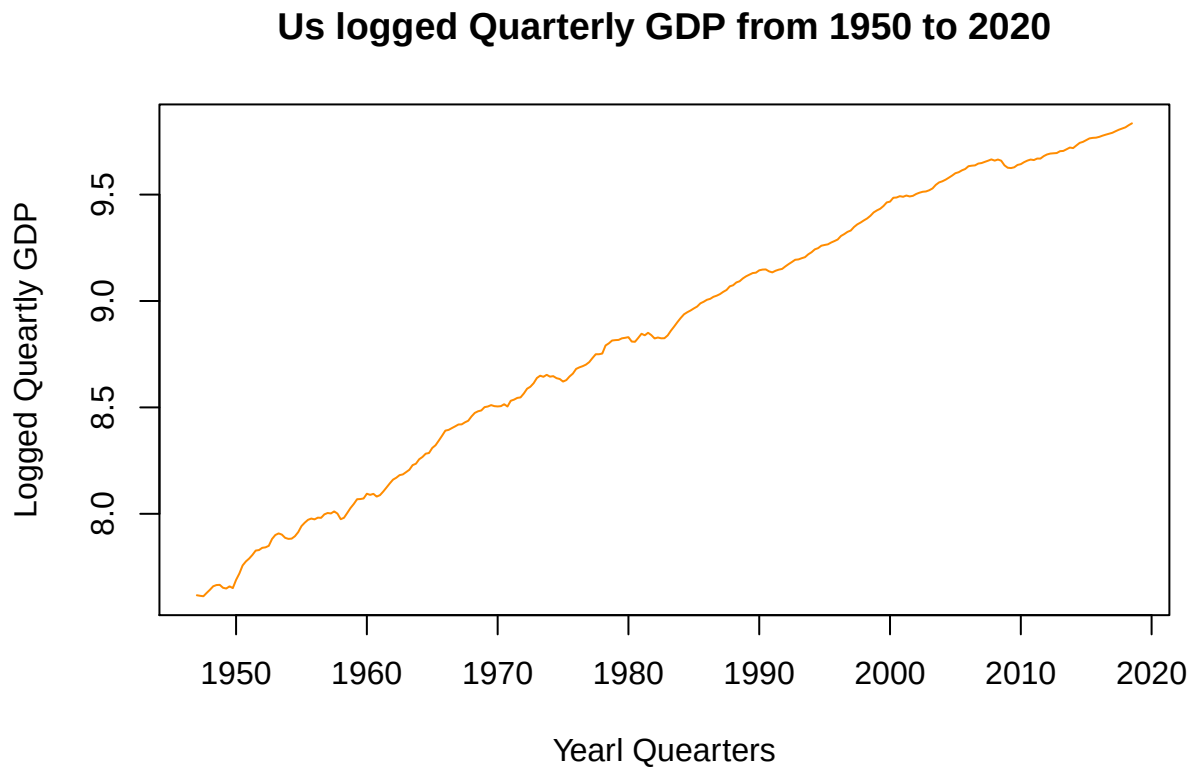
In this article, we are going to use the USA quarterly GDP available in **astsa** library in R. We will start with loading the required libraries and the data.

```
# if library not installed use
# install.packages("astsa")
# loading the library
library(astsa)
#loading the data
data(gdp)
#plotting the data. This is a time series data so we use the appropriate plots
plot.ts(gdp, col = "darkblue")
```



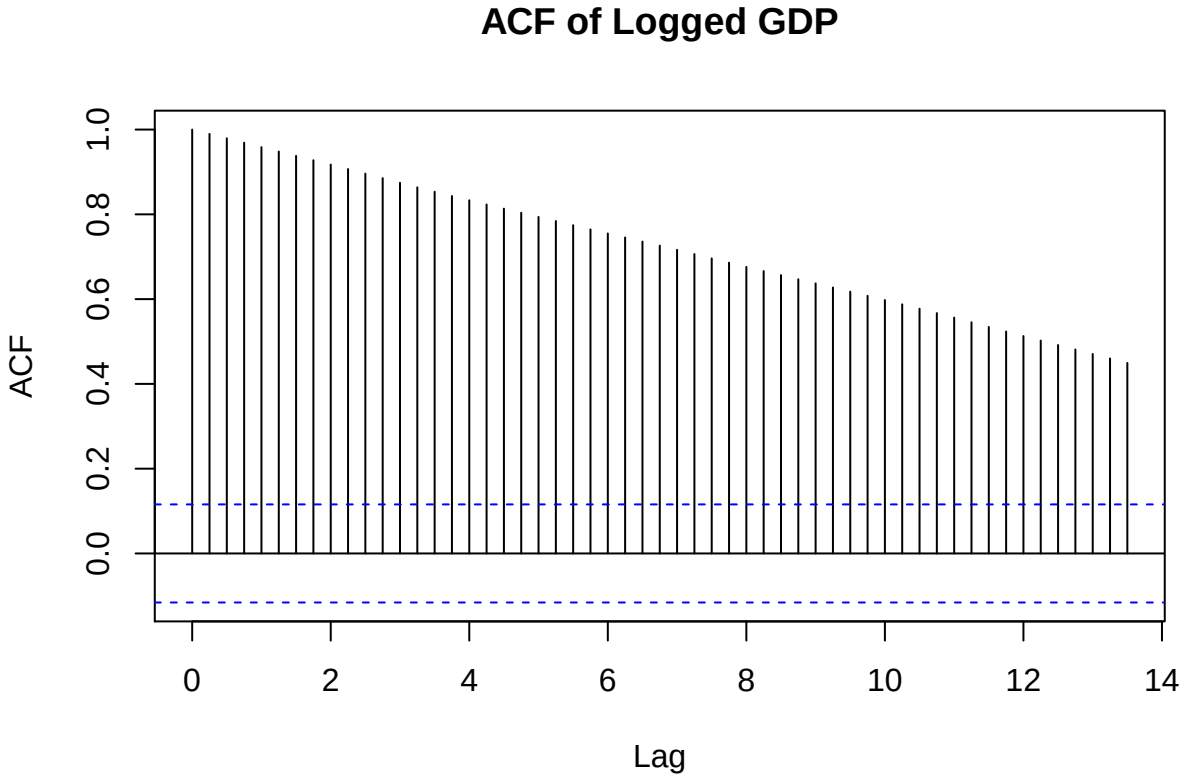
The the above plot we see that Y-axis values are two large. This means, our data is skewed. Since, the data is skewed and all positive, we can use log transformation.

```
# Takaing the log of the data
log_gdp = log(gdp)
# plotting the logged data
plot.ts(log_gdp, xlab = "Yearl Quearters", ylab = "Logged Queartly GDP",
        main = "Us logged Quarterly GDP from 1950 to 2020", col = "darkorange")
```



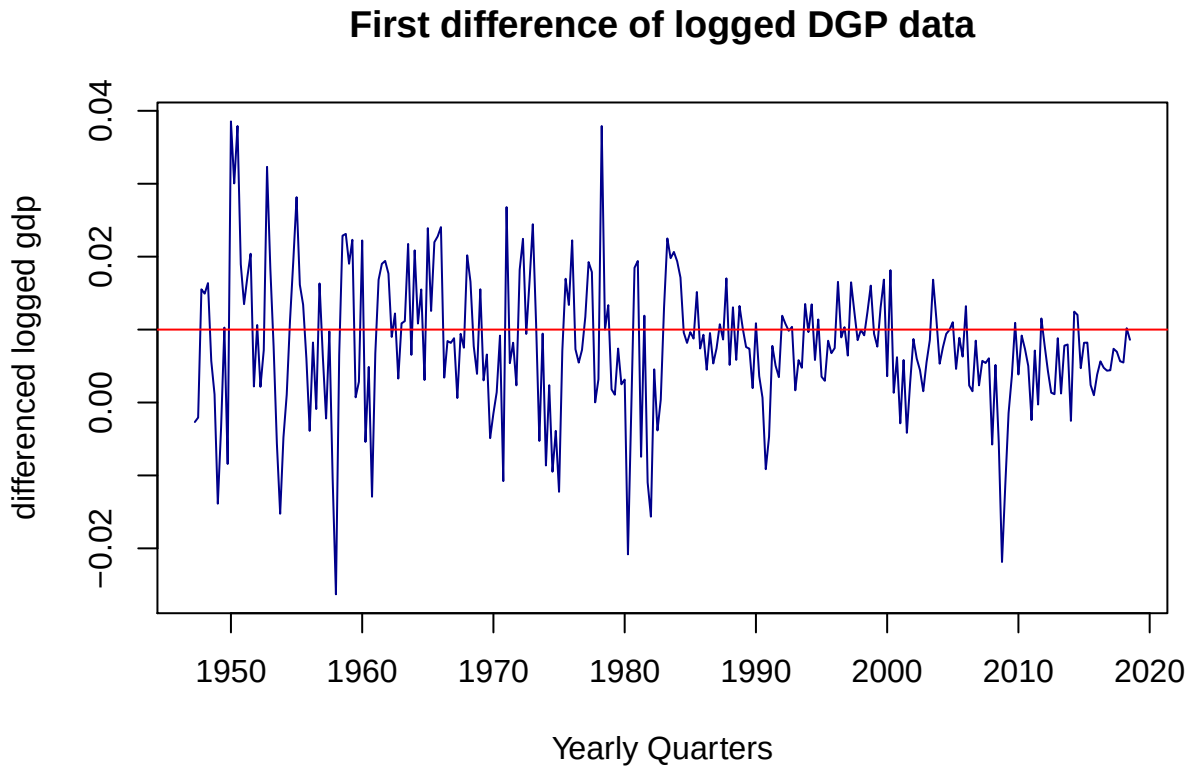
We can clearly see from the above plot, that the data has an upward trend. Therefore, the data is not stationary. To confirm this, we will plot the Auto-correlation function of the data.

```
# plotting the ACF of logged gdp data
acf(log_gdp, lag.max = 54, main = "ACF of Logged GDP")
```



The sample ACF, $\hat{\rho}(h)$ does to decay to zero fast enough as h, which is the time lag increases. This suggest the use of differencing transformation to remove non-stationarity of the data.

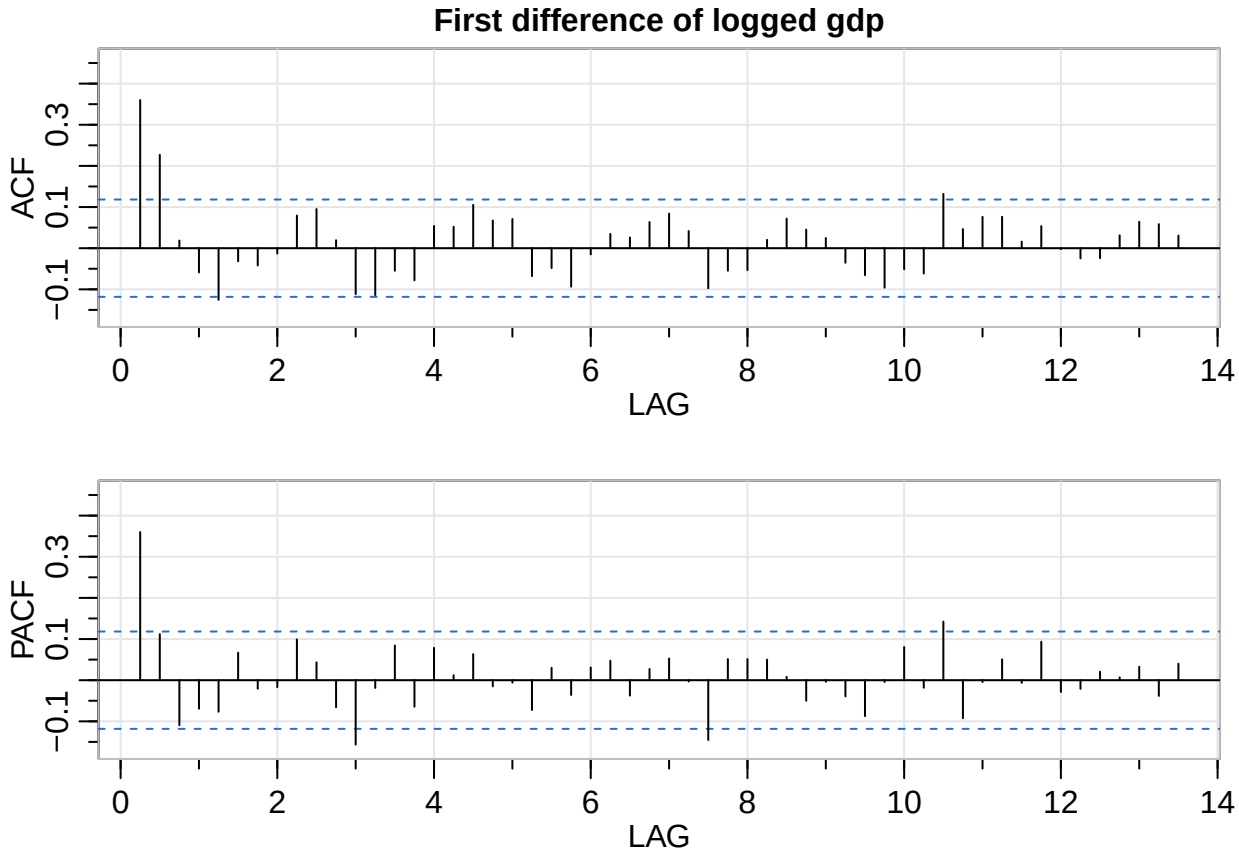
```
# Taking the first difference of the logged data and plotting it.
plot(diff(log_gdp), xlab = "Yearly Quarters", ylab = "differenced logged gdp",
     main = "First difference of logged DGP data", col = "darkblue")
# Adding a line
abline(h = 0.01, col = "red")
```



From the plot of the differenced data, we can see the trend element of the data has been removed, and the data seems stationary.

We can plot the ACF and the PACF (Partial Auto-correlation function) of the differenced data to decide what models need to be fitted.

```
# plots of ACF and PACF of differenced data
P_acf = acf2(diff(log_gdp), max.lag = 54, main = "First difference of logged gdp")
```



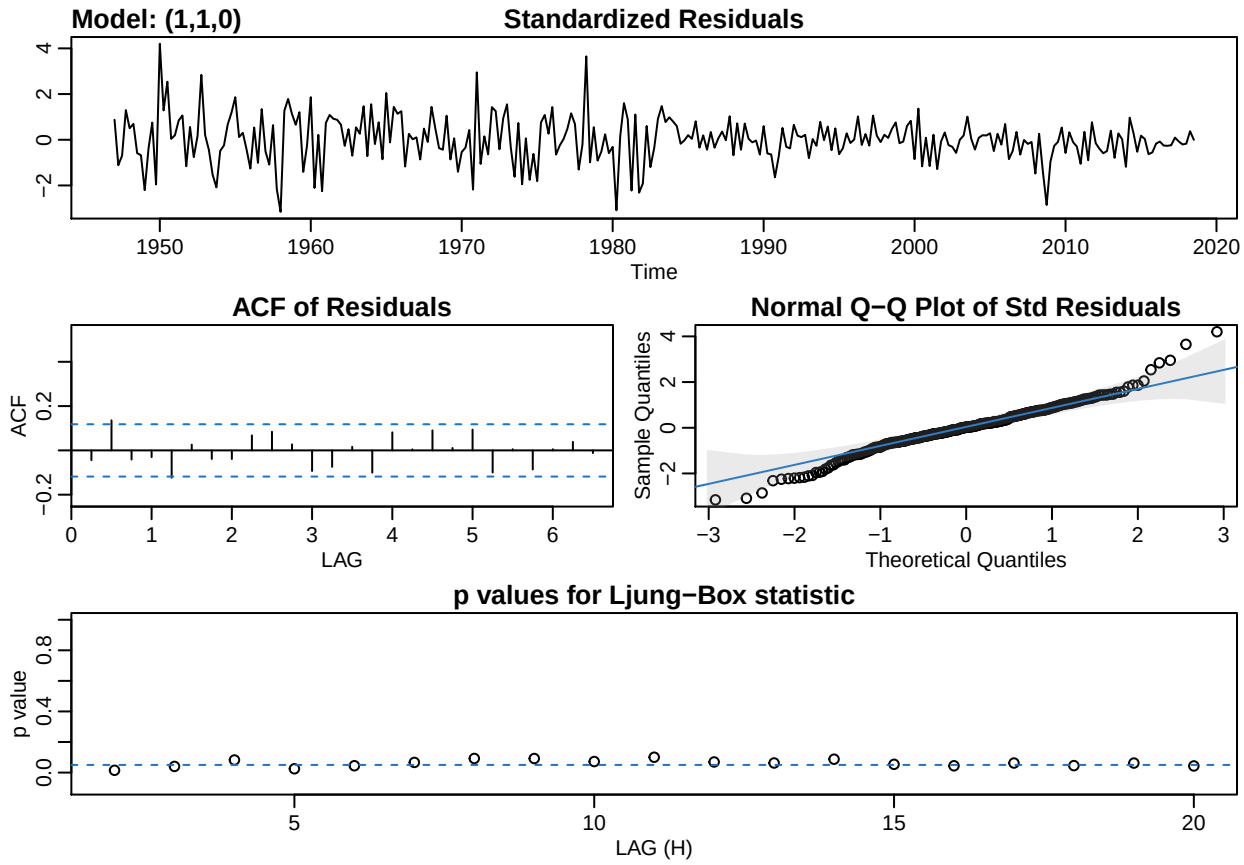
From the two plots above, it seems that PACF cuts of at lag 1 and ACF tails off at lag1. This suggests ARIMA(1, 1, 0) model for the logged data. We can also argue that ACF cuts of at lag 2 and PACF tails off at lag 2. This suggest ARIMA(0, 1, 2) for the logged data.

Fitting ARIMA models

Our initial exploration suggested two different ARIMA models. we are going to fit both of them and decide which one explains the data lag the best.

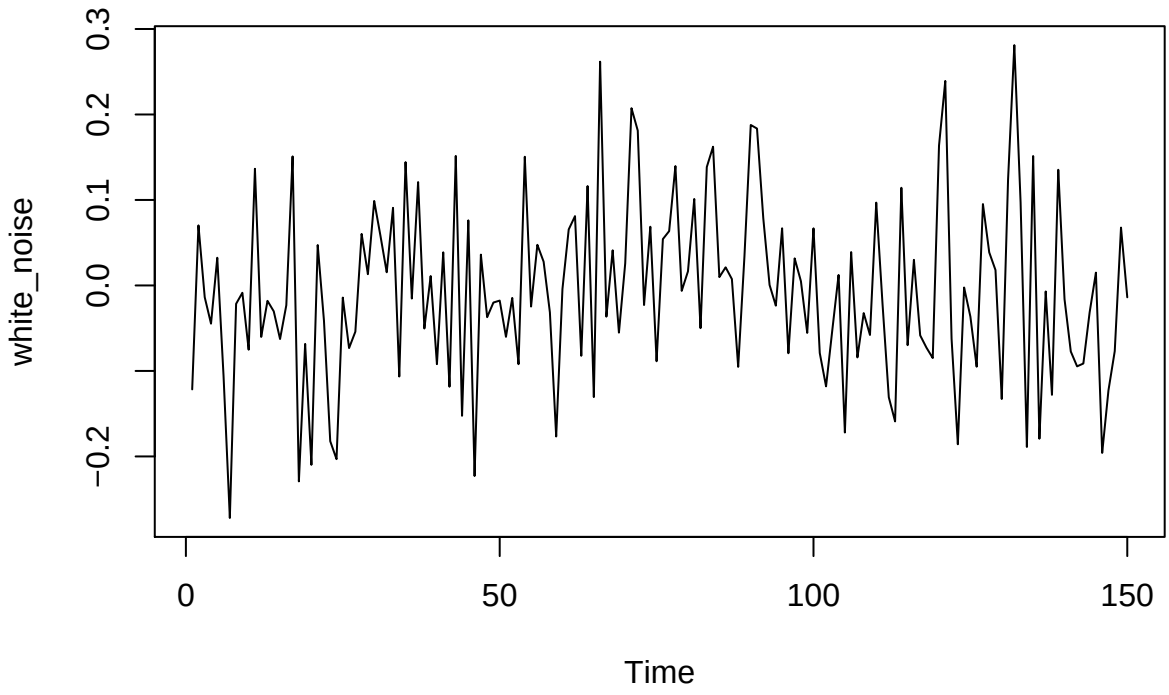
```
# Fitting ARIMA(1,1, 0). Note, this is an Autogressive model with lag 1.
AR1 = sarima(log_gdp, 1, 1, 0,)
```

```
## initial  value -4.673186
## iter   2 value -4.742918
## iter   3 value -4.742921
## iter   4 value -4.742923
## iter   5 value -4.742925
## iter   6 value -4.742925
## iter   6 value -4.742925
## final  value -4.742925
## converged
## initial  value -4.742227
## iter   2 value -4.742232
## iter   3 value -4.742244
## iter   3 value -4.742244
## iter   3 value -4.742244
## final  value -4.742244
## converged
```



We can see that **Standardized Residual** plot looks like the following white noise plot. This means, there is no indication of non constant variance.

```
white_noise = rnorm(150,0, 0.1)
plot.ts(white_noise)
```



ACF residuals are within the blue lines. This suggest, there is no serial correlation of the residuals at different lags. We can see almost all of ACF residuals fall within $2/(\sqrt{n})$ where n is th lenght of our dataset. This is given by 0.1180563

From Normal Q-Q plot of the std Residuals, we see that most of the residuals are normally distributed except may be few outliers at the two ends.

However, We can see that most of the p-value for **Ljung-Box statistic** are within a significance level of 5%. Therefore, we will not reject the hypothesis that the residuals have some serial correlation (if most of the p-vluess are on or below the blue line, there is indication of autocorrelation).

```
#coefficents and their p-values
AR1$ttable
```

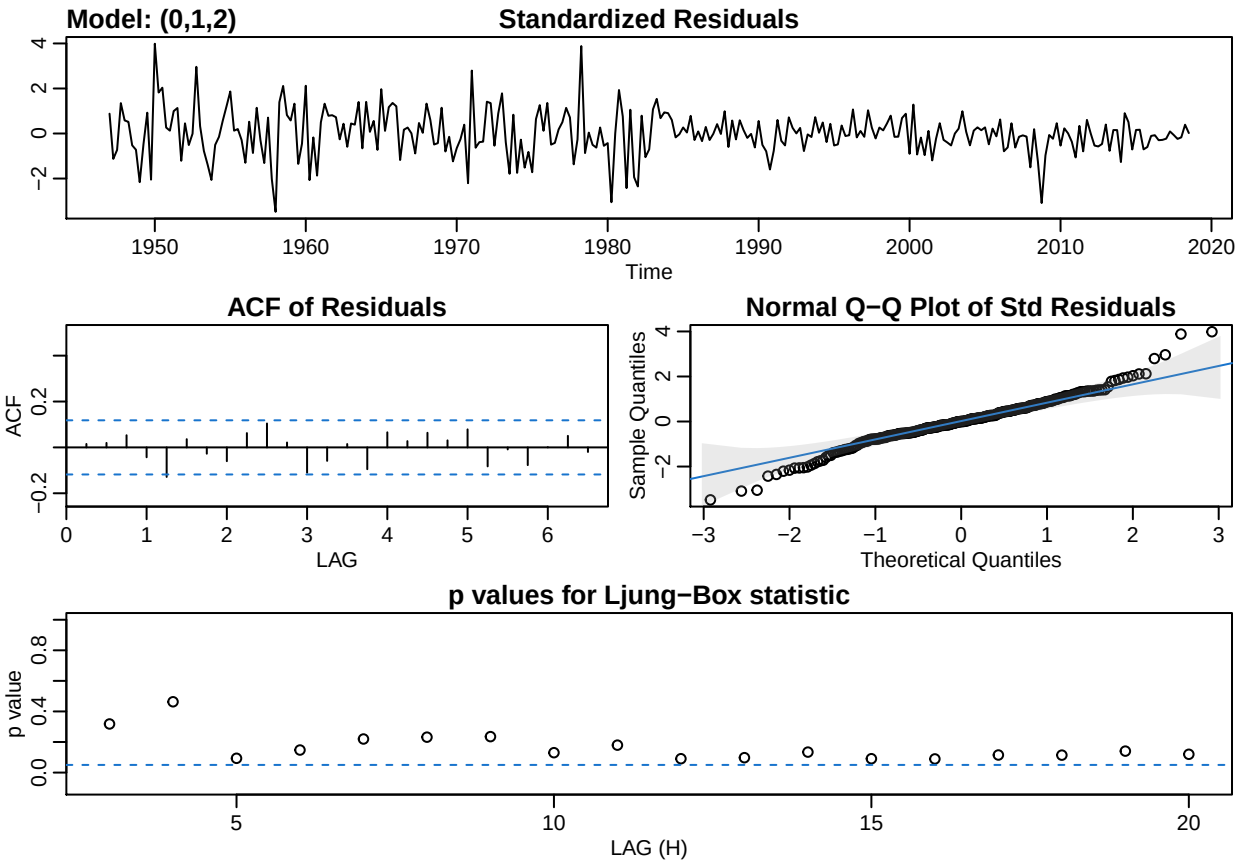
```
##          Estimate     SE t.value p.value
## ar1        0.3603 0.0551  6.5365       0
## constant   0.0077 0.0008  9.5915       0
```

We can see that the p-value fo the coefficiens and the constant term are both significant.

Next, We start fitting ARIMA(0, 1, 2) model to the logged data and go through the same diagnosis as the model above.

```
# Fitting ARIMA (0, 1, 2) model to the logged data
MA2 = sarima(log_gdp, 0, 1, 2)
```

```
## initial  value -4.672758
## iter   2 value -4.749239
## iter   3 value -4.750696
## iter   4 value -4.750723
## iter   5 value -4.750724
## iter   6 value -4.750725
## iter   7 value -4.750725
## iter   7 value -4.750725
## iter   7 value -4.750725
## final  value -4.750725
## converged
## initial  value -4.751076
## iter   2 value -4.751078
## iter   3 value -4.751079
## iter   4 value -4.751079
## iter   5 value -4.751079
## iter   5 value -4.751079
## iter   5 value -4.751079
## final  value -4.751079
## converged
```



The standardized Residuals plot, the ACF of Residuals does not show show any indication of serial correlation. The normal Q-Q Plot of Std Residuals suggest most of the residuals are Normally distributed except for few outliers at the end of the two ends. The LJung-Box statistics plot p-values are all above the reasonable significant level for most lags, unlike the ARIM(1, 1, 0) above.

```
MA2$ttable
```

```
##          Estimate     SE t.value p.value
## ma1        0.3070 0.0579  5.2987       0
## ma2        0.2258 0.0547  4.1270       0
## constant   0.0077 0.0008  9.8631       0
```

We can see that the coefficients and the constant term of this model are all significant

Using Ljung-Box statistics, we decided to proceed with ARIMA(0, 1, 2) for the prediction. For further Proof of the superiority of ARIMA(0, 1, 2), we can take a look at the AIC, AICc, BIC of each model.

```
# AIC, AICc, BIC of ARIMA(1,1,0) and ARIMA(0, 1, 2)
entries = data.frame("ARIMA(1,1,0)" = c(AR1$AIC, AR1$AICc, AR1$BIC),
                     "ARIMA(0,1,2)" = c(MA2$AIC, MA2$AICc, MA2$BIC),
                     row.names = c("AIC", "AICc","BIC"))
knitr::kable(entries)
```

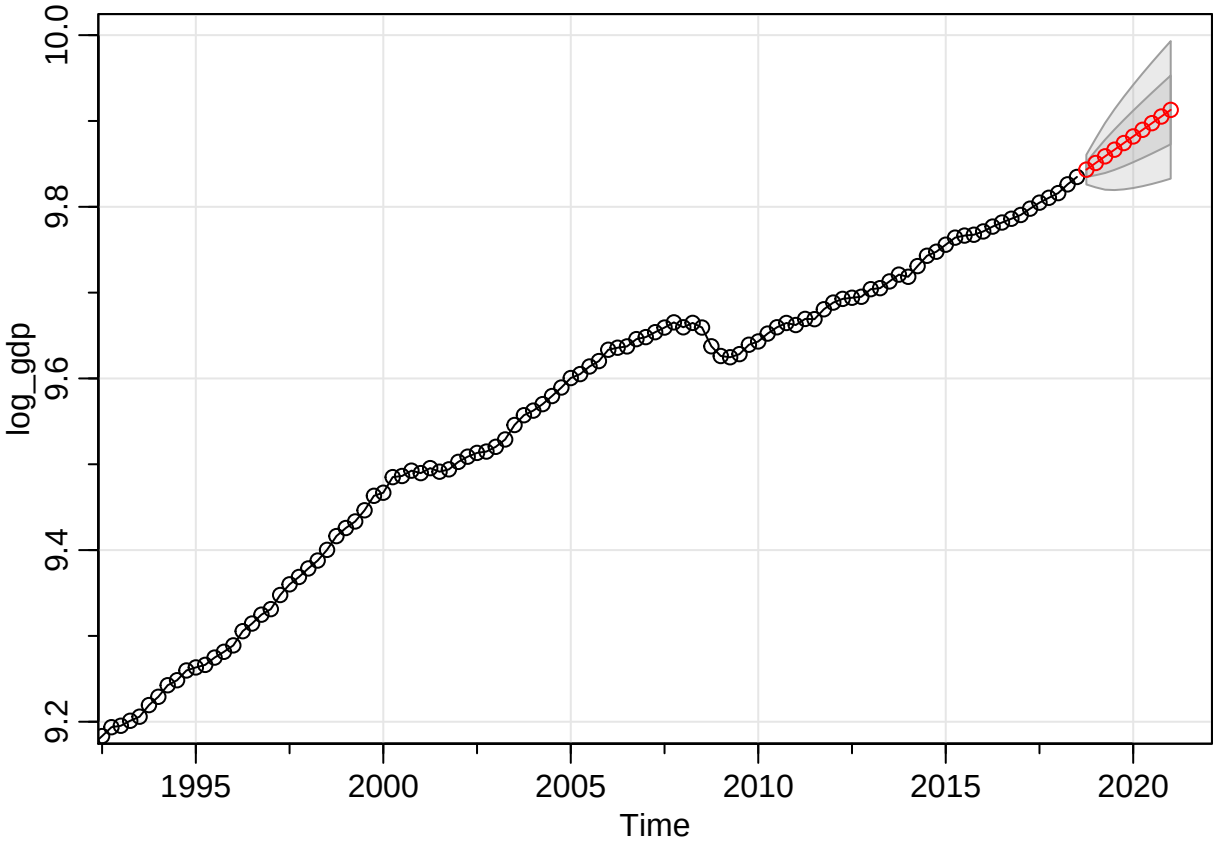
	ARIMA.1.1.0.	ARIMA.0.1.2.
AIC	-6.625631	-6.636309
AICc	-6.625483	-6.636011
BIC	-6.587281	-6.585176

We can that the AIC, and AICc of ARIMA(0, 1, 2) are less than that of ARIMA(1,1,0) although the difference is not that much. Therefore, we are going to use ARIMA(0, 1, 2) for predictions.

Predictions

We will try and forecast the next 10 quarters of the logged GDP data using ARIMA(0, 1, 2) model

```
#predicting next 10 quarters of the logged GDP data
pred = sarima.for(log_gdp, 10, 0, 1, 2)
```




```
#95% confidence prediction interval and taking the anti-log
#Upper 95% CI for the prediction and taking anti-log
upper = exp(pred$pred + qnorm(0.975)*pred$se)
#Lower 95 CI for the prediction and taking anti-log
lower = exp(pred$pred - qnorm(0.975)*pred$se)
#Putting the CI interval into a table
pred_data = data.frame(Predicted = exp(pred$pred), Lower_95 = lower,
                       Upper_95 = upper)

knitr::kable(pred_data)
```

Predicted	Lower_95	Upper_95
18831.76	18515.55	19153.37
18978.65	18457.07	19514.98
19125.97	18411.30	19868.39
19274.44	18406.29	20183.53
19424.05	18423.34	20479.11
19574.83	18454.82	20762.81

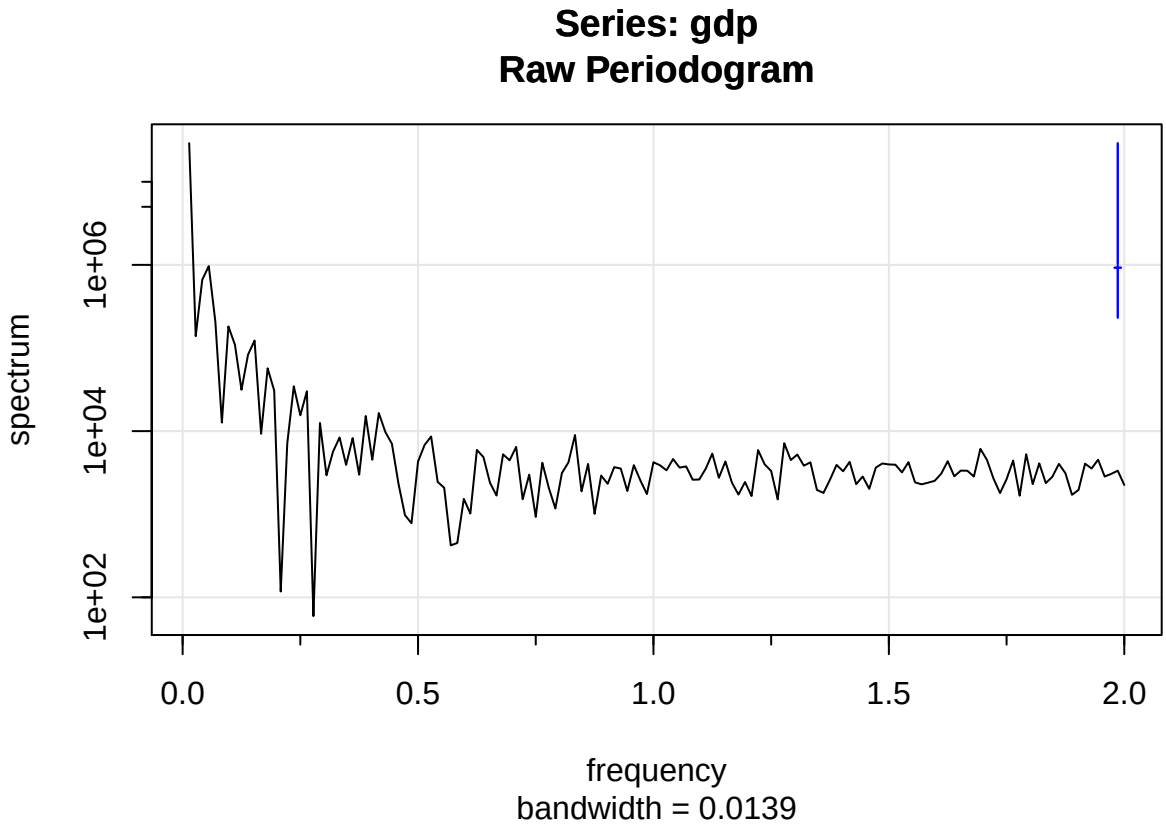
Predicted	Lower_95	Upper_95
19726.77	18496.73	21038.62
19879.90	18546.68	21308.95
20034.21	18603.12	21575.39
20189.73	18664.96	21839.05

We can see that our intervals are not that wide, which might mean our model might have done good job of fitting the data well. But, we can see towards the end as we move to values far away from the observed data, the prediction interval gets wider. This is expected for any prediction. Our model is likely to do better job of predicting values closer to the observed data.

Spectral Analysis:

For the rest of this document, we will perform spectral analysis to identify the three most dominant periods. In most of the time series data, cyclic dynamics are the rule rather than the exception. Spectral analysis enables us to estimate the periodicity of time series data. What spectral analysis is in a non-technical explanation is the decomposition of a time series into underlying sin and cosine functions of different frequencies. The aim to determine those frequencies that seem particularly strong.

```
library(MASS)
# spectral analysis for GDP data for USA
gdp.per = mvspec(gdp, log = "yes")
```




```
# Frequencies
P = gdp.per$details[order(gdp.per$details[, 3], decreasing = TRUE), ]
# the three most dominant Frequencies
P[1:3,]
```

```
##      frequency period   spectrum
## [1,]    0.0139     72 29110984.0
## [2,]    0.0556     18   966116.2
## [3,]    0.0417     24   662663.6
```

```r
#95% confidence intervals  for the three dominant frequencies
gdp_u1 = 2*P[1, 3]/qchisq(0.025, 2)
gdp_l1 = 2*P[1, 3]/qchisq(0.975, 2)
gdp_u2 = 2*P[2, 3]/qchisq(0.025, 2)
gdp_l2 = 2*P[2, 3]/qchisq(0.975, 2)
gdp_u3 = 2*P[3, 3]/qchisq(0.025, 2)
gdp_l3 = 2*P[3, 3]/qchisq(0.975, 2)
# creating tables for nice formatting.
knitr::kable(data.frame(series = rep("gdp", 3), Dominant.Freq = P[1:3, 1],
                        Spec = P[1:3,3], Lower = c(gdp_l1, gdp_l2, gdp_l3),
                        Upper = c(gdp_u1, gdp_u2, gdp_u3)))
```

series	Dominant.Freq	Spec	Lower	Upper
gdp	0.0139	29110984.0	7891552.0	1149822450
gdp	0.0556	966116.2	261899.6	38159552
gdp	0.0417	662663.6	179638.2	26173815

We see that the confidence intervals are extremely wide for all three cases, therefore, we can't establish the significance of the peaks.