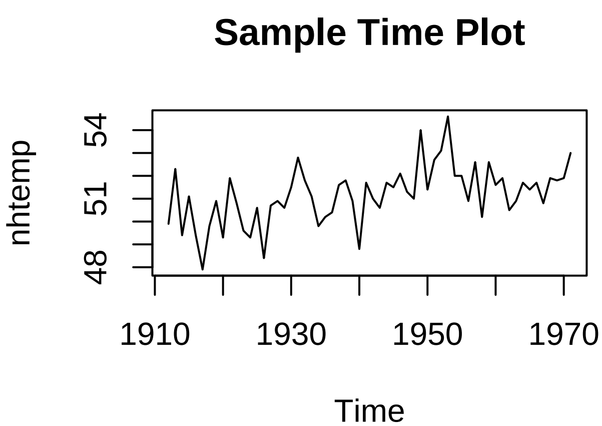
Time series CW2

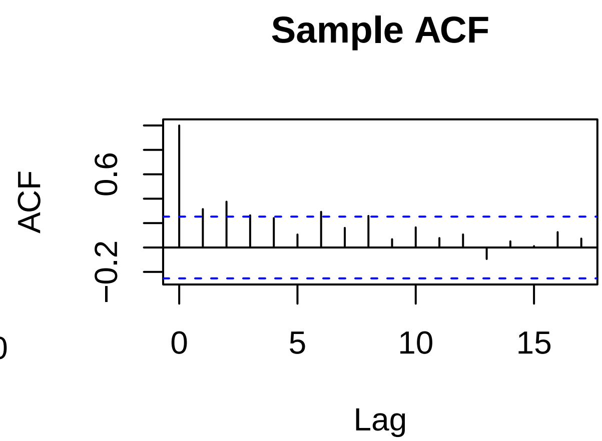
Liangxiao LI,2024-04-10

Q1: nhtemp

Part1: Check Stationarity

First we produce the time plot and ACF plot from the given data:





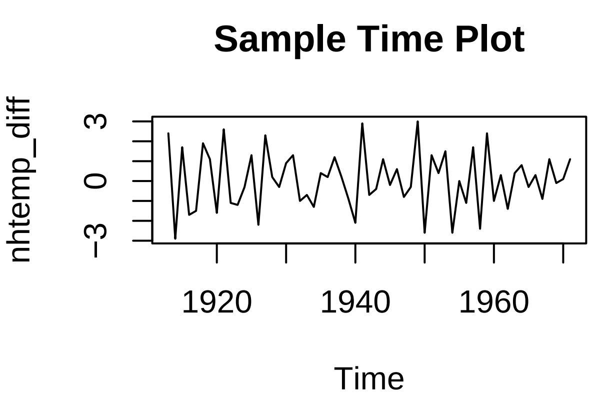
From plots above, we conclude the series is non-stationary and non-seasonal due to following reasons:

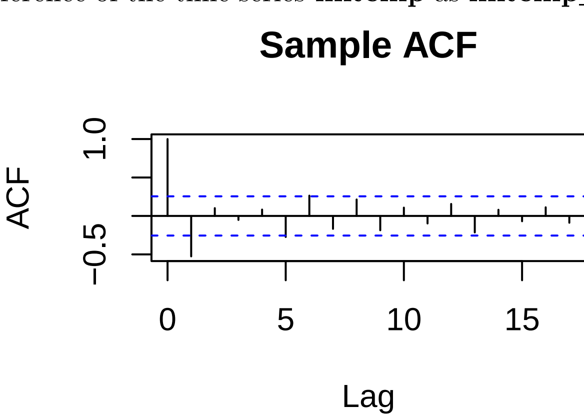
1) Time plot: the mean of the series appears higher between 1940-1970 to the period between 1910-1940.

2) Sample ACF plot: doesn't decline rapidly, therefore it's not stationary.

Part2: Remove non-stationarity through first difference

To remove non-stationarity, we take the first difference of the time series **nhtemp** as **nhtemp_diff**:





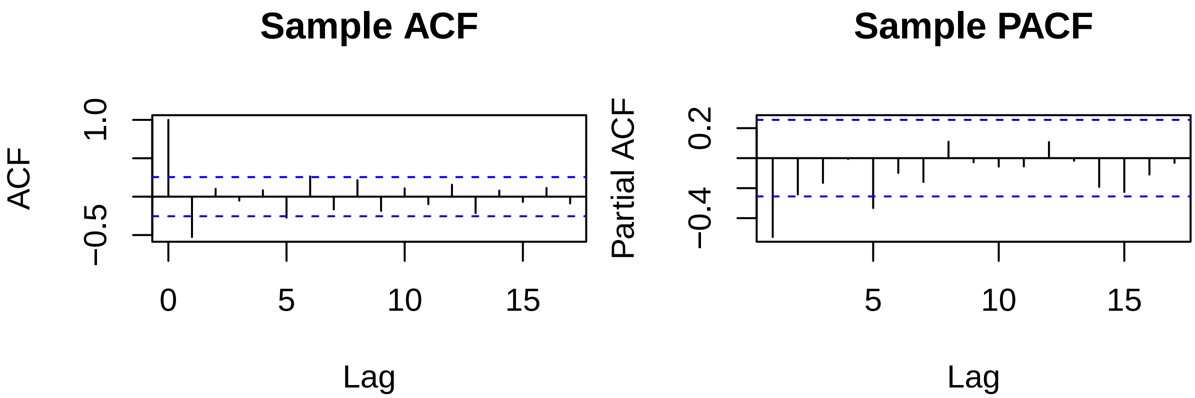
Therefore we conclude the series is (weakly) stationary without seasonality due to following reasons:

1) Time plot: has a mean equal to zero and shows constant variability over time.
2) Sample ACF plot: declines rapidly to zero as the lag increases, cut off after lag 1

In conclusion, we'll explore models with $d = 1$ in the following section.

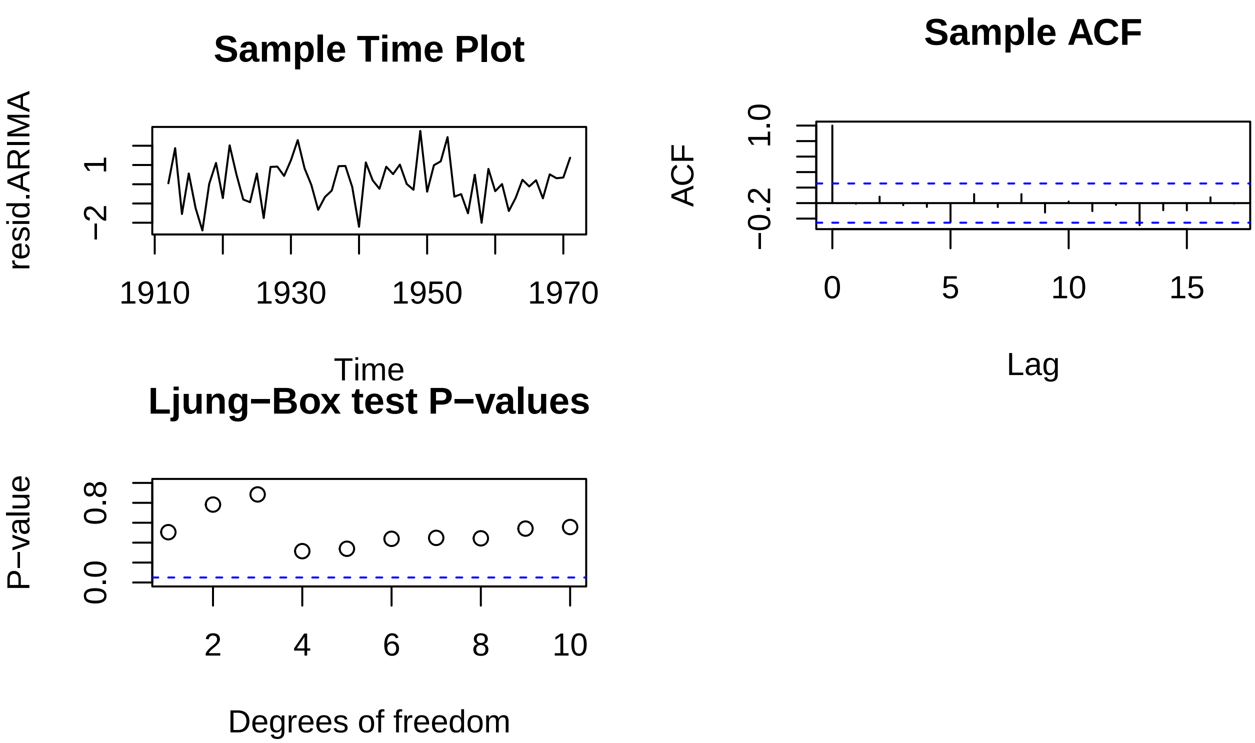
Part3: Model fitting - Parameter analysis

The analysis begins by analyzing the sample ACF and PACF plot for **nhtemp_diff**:



1) Since ACF cut off after lag 1, this suggest we begin by fitting an ARIMA(0,1,1) model
2) Since PACF doesn't cut off, this suggest the time series doesn't contain an AR component.

Part4: Model fitting - ARIMA(0,1,1)



From the plots above, we conclude that ARIMA(0,1,1) is a good fit due to following reasons:

1) Time plot of the model residuals:

The time plot of the residuals looks similar to white noise, with mean zero and constant variance.

2) A plot of the sample ACF of the model residuals:

For all lags > 0, the sample ACF are all close to zero. This suggests that the residuals are independent(uncorrelated).

3) A plot of the first ten P-values for the Ljung-Box test:

All p-values are greater than 0.05(non-significant), this suggests the ARIMA(0,1,1) is a good fit to the data.

Part5: ARIMA(0,1,1) vs. ARIMA(1,1,1)

However, it's still worth checking if adding AR(p) component would be a better fit. Therefore we fit the model again with ARIMA(1,1,1)

```
##
## Call:
## arima(x = nhtemp, order = c(0, 1, 1), method = "ML")
##
## Coefficients:
##           ma1
##       -0.7983
## s.e.   0.0956
##
## sigma^2 estimated as 1.291:  log likelihood = -91.76,  aic = 187.52
```

```
##
## Call:
## arima(x = nhtemp, order = c(1, 1, 1), method = "ML")
##
## Coefficients:
##          ar1      ma1
##       0.0073  -0.8019
## s.e.  0.1802   0.1285
##
## sigma^2 estimated as 1.291:  log likelihood = -91.76,  aic = 189.52
```

From the summary above, we conclude that ARIMA(0,1,1) is better than ARIMA(1,1,1) due to following reasons:

1) AIC for ARIMA(0,1,1) is 187.12 is less than AIC for ARIMA(1,1,1), which is 189.52.

2) Perform hypothesis test: $H_0 : \phi_1 = 0$ vs. $H_1 : \phi_1 \neq 0$. The test statistic $= \frac{0.0073}{0.1802} < 2$, therefore we don't reject the null hypothesis and thus ARIMA(0,1,1) is better than ARIMA(1,1,1) model.

3) Overall we'd prefer a parsimonious model, thus ARIMA(0,1,1) is better than ARIMA(1,1,1)

Part6: ARIMA(0,1,1) vs. ARIMA(0,1,2)

We check further whether adding an adittional MA(q) component would be a better fit. Therefore we fit the model again with ARIMA(0,1,2)

```
##
## Call:
## arima(x = nhtemp, order = c(0, 1, 2), method = "ML")
##
## Coefficients:
##           ma1      ma2
##       -0.7956  -0.0042
## s.e.   0.1224   0.1221
##
## sigma^2 estimated as 1.291:  log likelihood = -91.76,  aic = 189.52
```

From the summary above, we conclude ARIMA(0,1,1) is better than ARIMA(0,1,2) due to following reasons:

1) AIC for ARIMA(0,1,1) is 187.12 is less than AIC for ARIMA(0,1,2), which is 189.52.

2) Perform hypothesis test: $H_0 : \theta_1 = 0$ vs. $H_1 : \theta_1 \neq 0$. The test statistic $= |\frac{-0.0042}{0.1221}| < 2$, therefore we don't reject the null hypothesis and thus ARIMA(0,1,1) is better than ARIMA(0,1,2) model.

3) Overall we'd prefer a parsimonious model, thus ARIMA(0,1,1) is better than ARIMA(0,1,2)

Part7: Conclusion: ARIMA(0,1,1) best

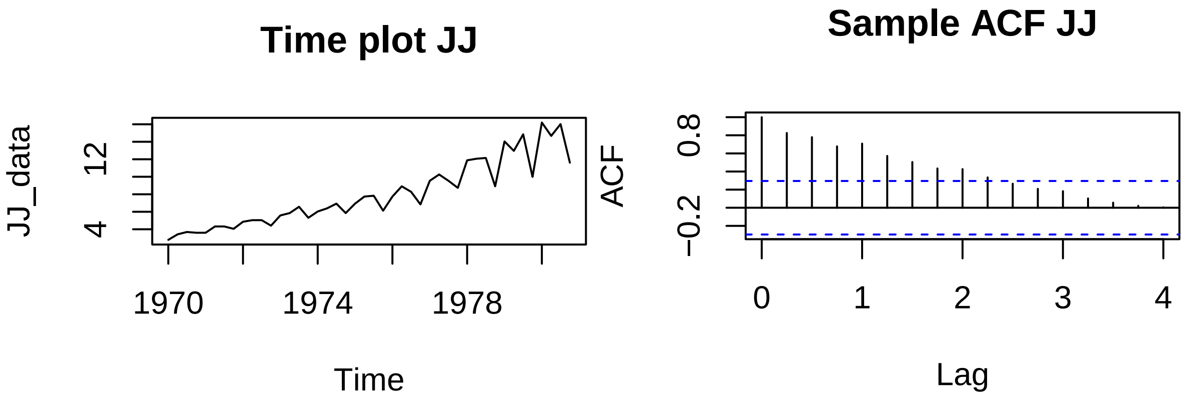
For question 1, the equation for the final fitted model is included below:

$$(1 - B)X_t = (1 - 0.7983B)Z_t$$

Q2: JJ_data

Part1: Check Stationarity

First we produce the time plot and ACF plot from the given data:



From the plots above, we conclude that the series is non-stationary and seasonal due to following reasons:

1) Time plot: both the mean and variance of the series appears to increase overtime, which indicate non-stationarity.

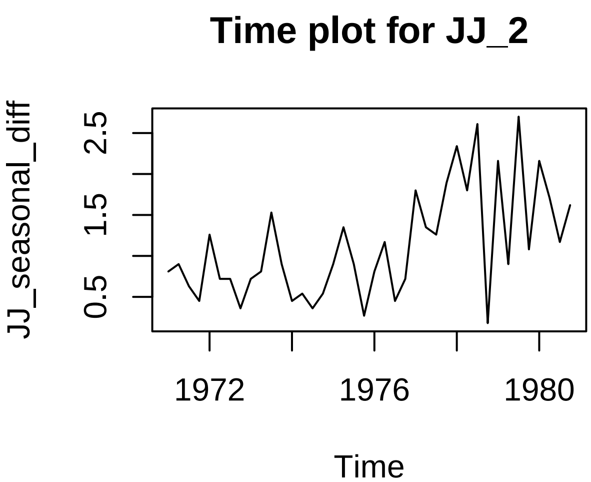
2) Sample ACF plot: doesn't decay rapidly, therefore it's not stationary.

3) Time plot: the data shows seasonality, as the earnings are higher in Qtr 2,3 and lower in Qtr 1,4

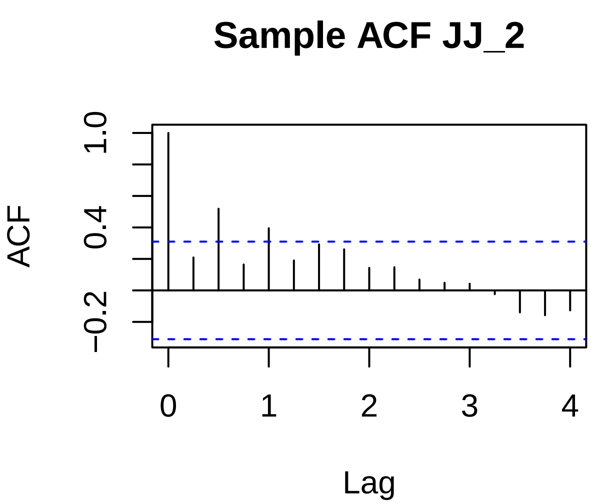
Therefore would need to apply a SARIMA model for JJ_data.

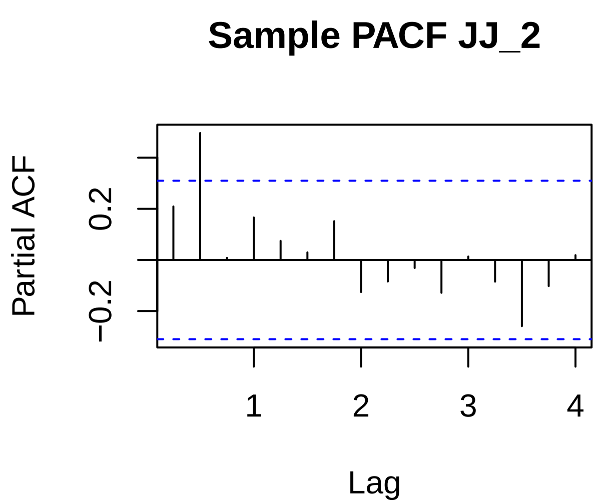
Part2: Apply Seasonal difference on JJ_1

According to the data description, JJ is a time series of the quarterly earnings between years, so the seasonal difference lag should be set to $h = 4$. There fore if JJ_1 denotes our original time series, we define the lag 4 difference time series JJ_2 as $JJ_2 = \nabla_4 JJ_1 = (1 - B^4)JJ_1$



From the time plot, it seems that the seasonality has been removed in JJ_2.

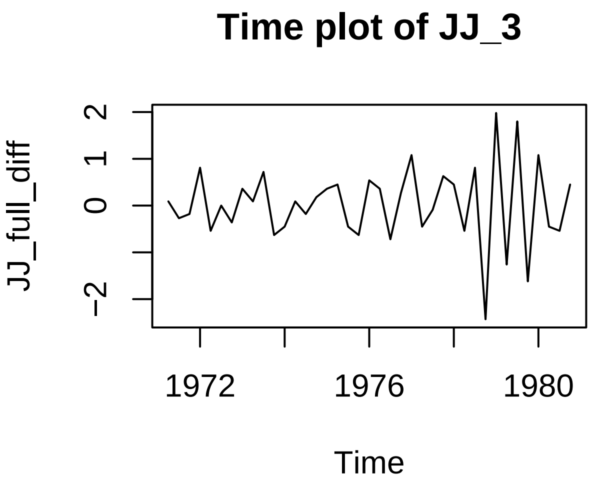


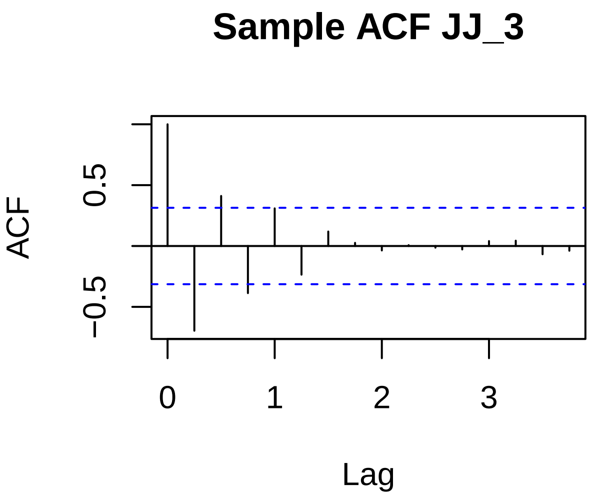


However, according to the sample ACF and sample PACF for the seasonally differnecd data, it suggest non-stationarity, because the ACF decays slowly.

Part3: Apply First difference on JJ_2

Therefore we'll take the first difference of JJ_2 and obtain $JJ_3 = \nabla^1 JJ_2 = (1-B)JJ_2$





Now JJ_3 appear to be stationary without seasonality.

However, the Time plot for JJ_3 shows a trend of non-constant variance, as the final part of the time series has greater variance compared with earlier part. Therefore we applied transformation to tackle with this problem.

Part4: Apply Transformation on JJ_1 to tackle non-constant variance

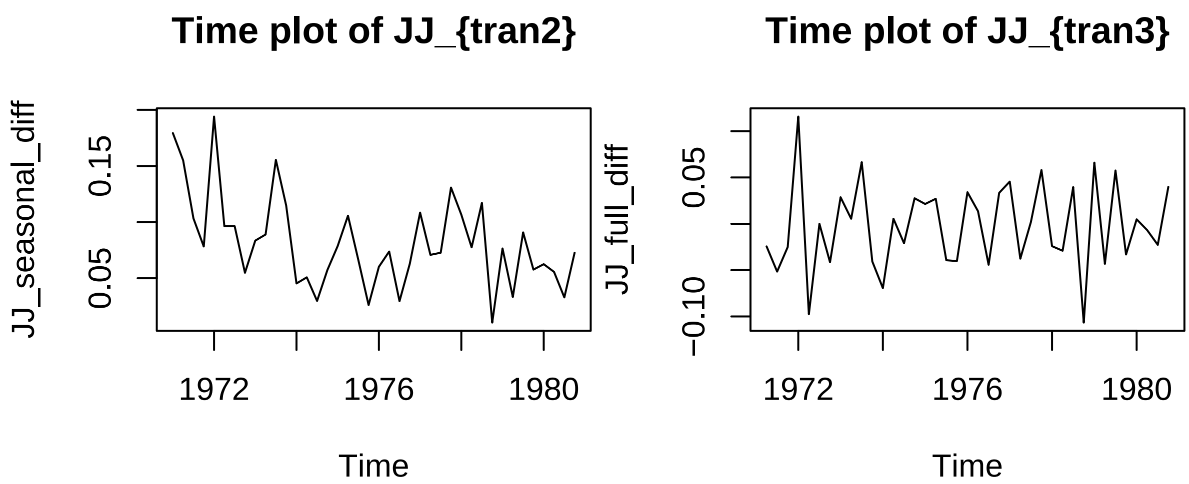
To identify

Here I used a window width of n=15 for calculations here. We see that the right hand plot of s_k^2 against $\hat{\mu}_k^2$ appears to be more linear than that of s_k^2 against $\hat{\mu}_k$, so we might consider a log transformation for this set of data to account for non-constant variance.

$$JJ_{tran1} = log(JJ_1)$$

Part5: Remove non-stationarity and seasonality from JJ_tran1

Then we carry on the same process to remove the non-stationarity and seasonality. We first difference JJ_{tran1} with a seasonal difference lag $h = 4$ and gain $JJ_{tran2} = \nabla_4(JJ_{tran1}) = (1-B^4)(JJ_{tran1})$, then we take the first difference on JJ_{tran2} and obtain $JJ_{tran3} = \nabla^1 JJ_{tran2} = (1-B)JJ_{tran2}$. Below is the time plot for JJ_{tran2} and JJ_{tran3}

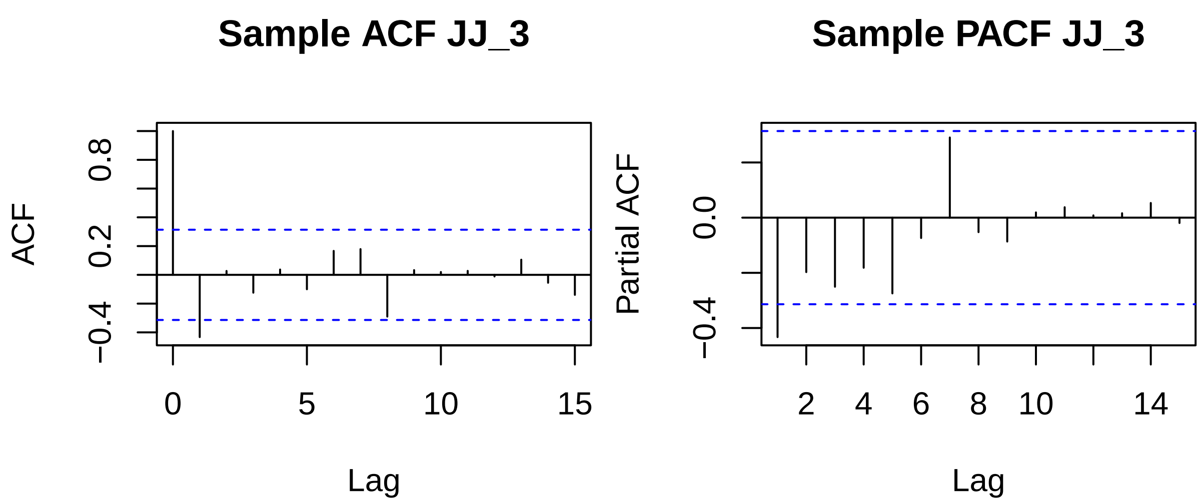


From the time plot, it seems both non-stationarity and seasonality has been removed from JJ_{tran1}.

Part6: SARIMA Parameter analysis for JJ_3 and JJ_{tran3}

Now we start our fitting attempt with SARIMA(p,1,q)x(P,1,Q)[4].

Part 6.1: Non-transformed data JJ_3



The best model should be **SARIMA(1,1,1) x (0,1,0)[4]** due to following reasons:

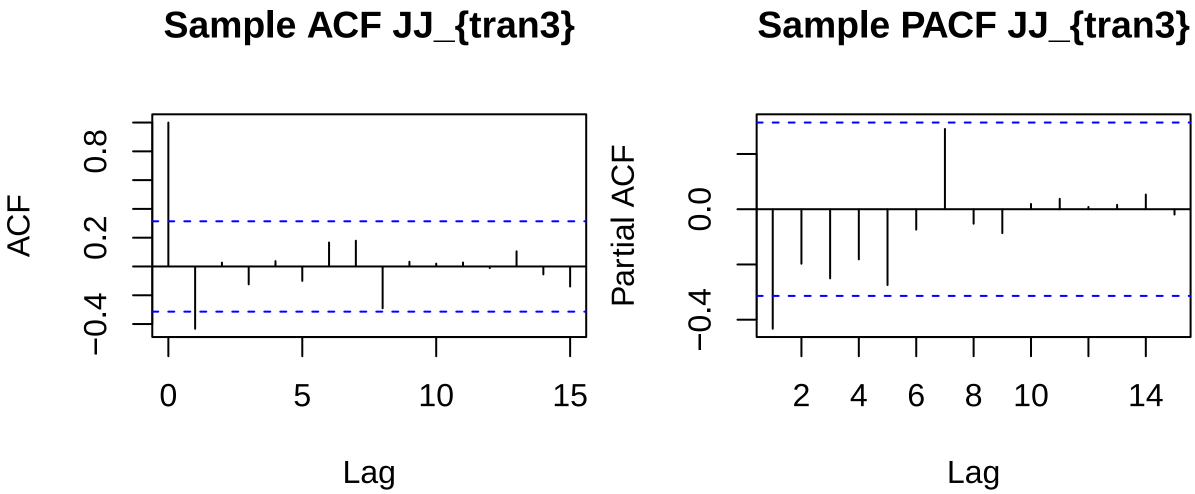
Seasonal components: (P,Q)

1) P: Check PACF at lag = 4,8,12 … PACF cut off already at lag = 4, therefore we choose P = 0.
2) Q: Check ACF at lag = 4,8,12 … ACF cut off already at lag = 4, therefore we choose Q = 0

Non Seasonal components : (p,q)

3) p: PACF cut off after lag = 1, therefore we choose p = 1
4) q: ACF cut off after lag = 1, therefore we choose q = 1.

Part 6.2: Transformed data JJ_{tran3}



The best model should be **SARIMA(0,1,1) x (0,1,0)**[4] due to following reasons:

Seasonal components: (P,Q)

1) P: Check PACF at lag = 4,8,12 … PACF cut off already at lag = 4, therefore we choose P = 0.

2) Q: Check ACF at lag = 4,8,12 … ACF cut off already at lag = 4, therefore we choose Q = 0

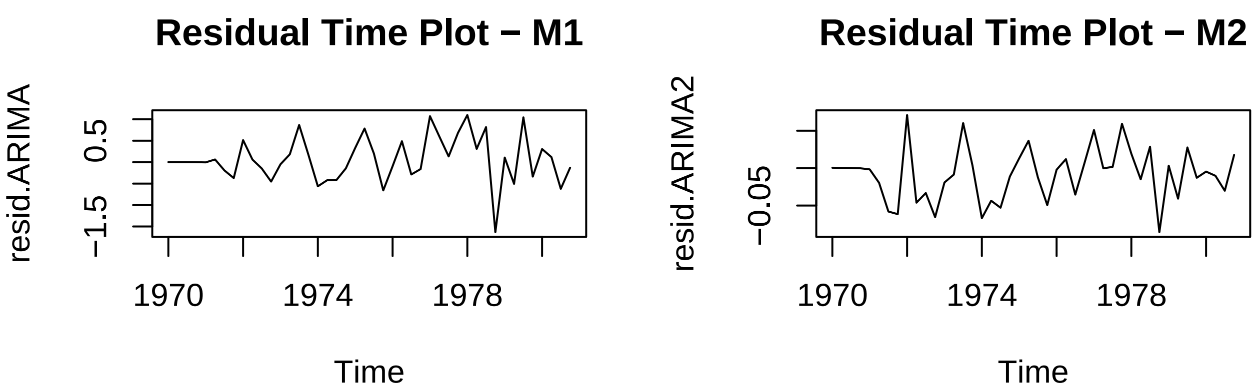
Non Seasonal components : (p,q)

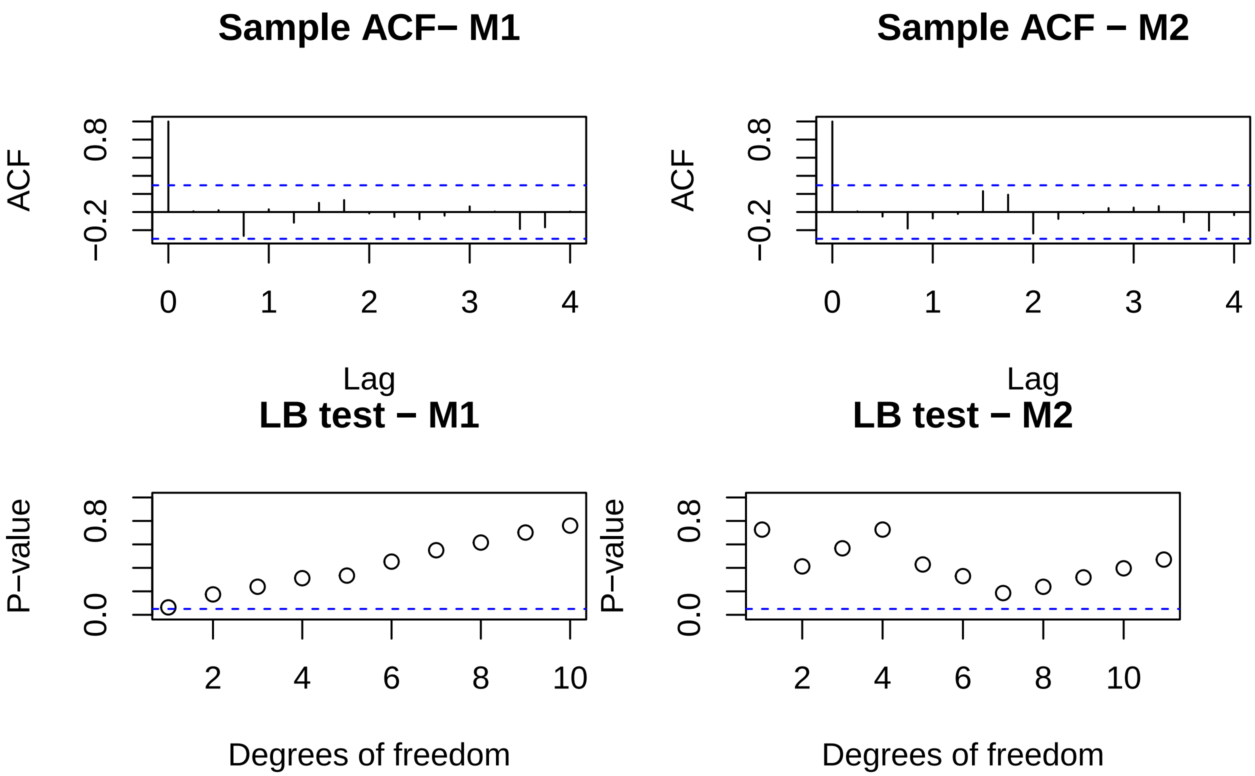
3) p: PACF cut off after lag = 0, therefore we choose p = 0

4) q: ACF cut off after lag = 1, therefore we choose q = 1.

Part7: Model Diagonostic JJ_1 vs. JJ_{tran1}

To make expression concise, I'll use '$M1$' to refer the SARIMA(1,1,1)x(0,1,0)[4] model for JJ_1 and '$M2$' to refer the SARIMA(0,1,1)x(0,1,0)[4] model for JJ_{tran1}





From the plots above, we conclude that both M1 and M2 are good fits, while **M2 is slightly better** due to following reasons:

1) Time plot of the model residuals:

The time plot of the residuals for M2 looks to be white noise, with mean zero and constant variance. (However for M1 the **variance isn't constant**)

2) A plot of the sample ACF of the model residuals

For all lags > 0, the sample ACF are all close to zero except at lag $= 1$. This suggests that the residuals are almost independent(uncorrelated).

3) A plot of the first ten P-values for the Ljung-Box test

All p-values for M2 are greater than 0.05(non-significant), however for M1 the first p-value is significant)

Part8: SARIMA(0,1,1)x(0,1,0)[4] vs. SARIMA(1,1,1)x(0,1,0)[4] for M2

However, it's still worth checking if adding AR(p) component would be a better fit. Therefore we fit the model again with SARIMA(1,1,1)x(0,1,0)[4] for JJ_{tran1}

```
##
## Call:
## arima(x = JJ_data_transformed, order = c(0, 1, 1), seasonal = list(order = c(0,
##     1, 0), period = 4), method = "ML")
##
## Coefficients:
##          ma1
```

```
##       -0.7326
## s.e.   0.1219
##
## sigma^2 estimated as 0.001485:  log likelihood = 71.27,  aic = -138.54
```

```
##
## Call:
## arima(x = JJ_data_transformed, order = c(1, 1, 1), seasonal = list(order = c(0,
##     1, 0), period = 4), method = "ML")
##
## Coefficients:
##          ar1      ma1
##       0.0837  -0.7693
## s.e.  0.2131   0.1438
##
## sigma^2 estimated as 0.001479:  log likelihood = 71.35,  aic = -136.7
```

From the summary above, we conclude that SARIMA(0,1,1)x(0,1,0)[4] is better than SARIMA(1,1,1)x(0,1,0)[4] due to following reasons:

1) AIC for SARIMA(0,1,1)x(0,1,0)[4] is -138.54, this is less than AIC for SARIMA(1,1,1)x(0,1,0)[4], which is -136.7.

2) Overall we'd prefer a parsimonious model, thus SARIMA(0,1,1)x(0,1,0)[4] is better than SARIMA(1,1,1)x(0,1,0)[4]

Part9: Conclusion

Let X_t denote the original JJ_data

For the **non-transformed** data, the best model is SARIMA(1,1,1)x(0,1,0)[4], and the equation is:

$$(1+0.3465B)(1-B)(1-B^4)X_t = (1-0.6308B)Z_t$$

For the **boxcox transformed** data, the best model is SARIMA(0,1,1)x(0,1,0)[4], and the equation is:

$$(1-B)(1-B^4)boxcox(X_t) = (1-0.7325B)Z_t$$

here $boxcox(\cdot)$ denotes the transformation performed on the original JJ_data.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)

library(forecast)
LB_test<-function(resid,max.k,p,q){
  lb_result<-list()
  df<-list()
```

```
  p_value<-list()
  for(i in (p+q+1):max.k){
    lb_result[[i]]<-Box.test(resid,lag=i,type=c("Ljung-Box"),fitdf=(p+q))
    df[[i]]<-lb_result[[i]]$parameter
    p_value[[i]]<-lb_result[[i]]$p.value
  }
  df<-as.vector(unlist(df))
  p_value<-as.vector(unlist(p_value))
  test_output<-data.frame(df,p_value)
  names(test_output)<-c("deg_freedom","LB_p_value")
  return(test_output)
}
load("nhtemp.rda")
# Time Series Plot
ts.plot(nhtemp, main="Sample Time Plot")

# ACF Plot
acf(nhtemp, main="Sample ACF")

# PACF Plot
#pacf(nhtemp, main="Sample PACF")
nhtemp_diff<-diff(nhtemp)
ts.plot(nhtemp_diff, main="Sample Time Plot")
acf(nhtemp_diff, main="Sample ACF")
#pacf(nhtemp_diff)
acf(nhtemp_diff, main="Sample ACF")
pacf(nhtemp_diff, main = "Sample PACF")
ARIMA<-arima(nhtemp,order=c(0,1,1),method="ML")
ARIMA
resid.ARIMA<-residuals(ARIMA)
ts.plot(resid.ARIMA, main = "Sample Time Plot")
acf(resid.ARIMA, main = "Sample ACF")
ARIMA.LB<-LB_test(resid.ARIMA,max.k=11,p=0,q=1)
#To produce a plot of the P-values against the degrees of freedom and
#add a blue dashed line at 0.05, we run the commands
plot(ARIMA.LB$deg_freedom,ARIMA.LB$LB_p_value,xlab="Degrees of freedom",ylab="P-value",main="L
abline(h=0.05,col="blue",lty=2)
ARIMA
ARIMA<-arima(nhtemp,order=c(1,1,1),method="ML")
ARIMA
ARIMA<-arima(nhtemp,order=c(0,1,2),method="ML")
ARIMA
load("JJ_data.rda")
#JJ_data_ts <- ts(JJ_data, start=c(1970, 1))
LB_test_SARIMA<-function(resid,max.k,p,q,P,Q){
 lb_result<-list()
 df<-list()
```

```r
 p_value<-list()
  for(i in (p+q+P+Q+1):max.k){
   lb_result[[i]]<-Box.test(resid,lag=i,type=c("Ljung-Box"),fitdf=(p+q+P+Q))
   df[[i]]<-lb_result[[i]]$parameter
   p_value[[i]]<-lb_result[[i]]$p.value
  }
 df<-as.vector(unlist(df))
 p_value<-as.vector(unlist(p_value))
 test_output<-data.frame(df,p_value)
 names(test_output)<-c("deg_freedom","LB_p_value")
 return(test_output)
 }
ts.plot(JJ_data, main = "Time plot JJ")
acf(JJ_data,main = "Sample ACF JJ")
#pacf(JJ_data)
JJ_seasonal_diff <- diff(JJ_data,lag=4)
ts.plot(JJ_seasonal_diff,main ="Time plot for JJ_2")
acf(JJ_seasonal_diff,main = "Sample ACF JJ_2")
pacf(JJ_seasonal_diff, main = "Sample PACF JJ_2")
JJ_full_diff <- diff(JJ_seasonal_diff)
ts.plot(JJ_full_diff,main = "Time plot of JJ_3")
acf(JJ_full_diff, main = "Sample ACF JJ_3")
#pacf(JJ_full_diff, main = "Sample PACF JJ_3")
# Assuming JJ_data is your time series data vector

# Define the window size
n <- 15

# Initialize vectors to store the running mean and variance
running_mean <- vector("numeric", length(JJ_data)-n+1)
running_variance <- vector("numeric", length(JJ_data)-n+1)

# Calculate running mean and variance using a for loop
for (i in 1:((length(JJ_data))-n+1)) {
  # Determine the start and end of the current window
  window_start <- i
  window_end <- i+n-1
  print(i)
  # Slice the window from the data
  window <- JJ_data[window_start:window_end]
  running_mean[i] <- mean(window)
  running_variance[i] <- var(window)
}

plot(running_mean, running_variance,
     main="Scatter Plot of Running Mean vs Running Variance",
     xlab="Running Mean",
```

```
     ylab="Running Variance",
     pch=19, col="blue") # 'pch=19' specifies the point type, 'col' specifies the color

plot(running_mean^2, running_variance,
     main="Scatter Plot of Running Mean^2 vs Running Variance",
     xlab="Running Mean",
     ylab="Running Variance",
     pch=19, col="blue") # 'pch=19' specifies the point type, 'col' specifies the color

# Estimate the optimal lambda for the Box-Cox transformation
lambda <- BoxCox.lambda(JJ_data)
# Apply the Box-Cox transformation
JJ_data_transformed <- BoxCox(JJ_data, lambda)
ts.plot(JJ_data_transformed, main ="Time plot of JJ_{tran1}")
acf(JJ_data_transformed, main = "sample ACF of JJ_{tran1}")
#pacf(JJ_data)
JJ_seasonal_diff <- diff(JJ_data_transformed,lag=4)
ts.plot(JJ_seasonal_diff,main ="Time plot of JJ_{tran2}")
JJ_full_diff <- diff(JJ_seasonal_diff)
ts.plot(JJ_full_diff,main = "Time plot of JJ_{tran3}")
JJ_data_ts <- ts(JJ_full_diff, start=c(1970, 1))
acf(JJ_data_ts, main = "Sample ACF JJ_3")
pacf(JJ_data_ts, main = "Sample PACF JJ_3")
JJ_data_ts <- ts(JJ_full_diff, start=c(1970, 1))
acf(JJ_data_ts, main = "Sample ACF JJ_{tran3}")
pacf(JJ_data_ts, main = "Sample PACF JJ_{tran3}")
fit <- auto.arima(JJ_data_transformed)
summary(fit)
ARIMA<-arima(JJ_data,order=c(1,1,1),seasonal=list(order=c(0,1,0),period=4),method="ML")
ARIMA2<-arima(JJ_data_transformed,order=c(0,1,1),seasonal=list(order=c(0,1,0),period=4),method=

resid.ARIMA<-residuals(ARIMA)
resid.ARIMA2<-residuals(ARIMA2)

ts.plot(resid.ARIMA, main = "Residual Time Plot - M1")
ts.plot(resid.ARIMA2, main = "Residual Time Plot - M2")


acf(resid.ARIMA, main = "Sample ACF- M1")
acf(resid.ARIMA2, main = "Sample ACF - M2")


ARIMA.LB<-LB_test_SARIMA(resid.ARIMA,max.k=12,p=1,q=1,P=0,Q=0)
#To produce a plot of the P-values against the degrees of freedom and
#add a blue dashed line at 0.05, we run the commands
plot(ARIMA.LB$deg_freedom,ARIMA.LB$LB_p_value,xlab="Degrees of freedom",ylab="P-value",main="LB
abline(h=0.05,col="blue",lty=2)
```

```r
ARIMA.LB2<-LB_test_SARIMA(resid.ARIMA2,max.k=12,p=0,q=1,P=0,Q=0)
#To produce a plot of the P-values against the degrees of freedom and
#add a blue dashed line at 0.05, we run the commands
plot(ARIMA.LB2$deg_freedom,ARIMA.LB2$LB_p_value,xlab="Degrees of freedom",ylab="P-value",main=
abline(h=0.05,col="blue",lty=2)
```

```r
ARIMA2
ARIMA2<-arima(JJ_data_transformed,order=c(1,1,1),seasonal=list(order=c(0,1,0),period=4),method=
ARIMA2
```