



PROJECT REPORT

Time Series Analysis DS 809 - **Air Quality**

Rakesh, Shawn , Prem

Contents

- 1. Introduction and Overview.....**
- 2. Regression Models.....**
- 3. Deterministic Time Series Models.....**
 - Estimating an indicator variable model
 - Estimating a Polynomial model
 - Estimating a cyclical model(harmonic)
- 4. Stochastic Time Series Models**
 - Investigating ACF and PACF
- 5. Predictive Performance Comparison**
- 6. Multivariate Time Series Models**
- 7. Conclusion**

1. Introduction and Overview

As a team of students in the DS 809 Time Series Analysis course, we gather and present some information on our research and findings and relative conclusions. In this case, our dataset contains 9358 observations of hourly responses of an Air Quality Multisensory Device.

This case-study is all about analyzing this data set and applying the time series concepts like regression models, deterministic time series, stochastic time series, predictive performance comparison and multivariate time series and obtaining models respectively. We conclude this project by commenting on the findings and discussing their implications.

Overview of Data

In this project, we will be working over the data set which contains 9358 instances of hourly true averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensory Device. The device was located on the field in a significantly polluted area, at road level, within an Italian city. Data were recorded from March 2004 to February 2005 (one year) representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses. Ground Truth hourly averaged concentrations for CO, Non Methanic Hydrocarbons, Benzene, Total Nitrogen Oxides (NO_x) and Nitrogen Dioxide (NO₂) and were provided by a co-located reference certified analyzer. Evidence of cross-sensitivities as well as both concept and sensor drifts are present as described in De Vito et al., Sens. And Act. B, Vol. 129,2,2008 (citation required) eventually affecting the sensor's concentration estimation capabilities. Missing values are tagged with a '-200' value. This dataset can be used exclusively for research purposes. Commercial purposes are fully excluded.

This case-study is all about analyzing this data set.

Attribute Information:

<i>Index</i>	<i>Variable</i>	<i>Column Definition</i>	<i>Derived Column</i>
1	CO(GT)	Hourly avg conc. CO in mg/m³	V1
2	S1(CO)	Hourly avg sensor response of CO targeted	V2
3	C6H6(GT)	Hourly avg benzene conc. in mg/m³	V3
4	S2(NMHC)	Hourly avg sensor response of NMHC targeted	V4

5	<i>NOx(GT)</i>	<i>Hourly avg NOx concentration in ppb</i>	<i>V5</i>
6	<i>S3(NOx)</i>	<i>Hourly avg sensor response of NOx targeted</i>	<i>V6</i>
7	<i>NO2(GT)</i>	<i>Hourly avg NO2 concentration in ppb</i>	<i>V7</i>
8	<i>S4(NO2)</i>	<i>Hourly avg sensor response of NO2 targeted</i>	<i>V8</i>
9	<i>S5(O3)</i>	<i>Hourly avg sensor response of O3 targeted</i>	<i>V9</i>
10	<i>RH</i>	<i>Relative Humidity in %</i>	<i>V10</i>
11	<i>AH</i>	<i>Absolute Humidity</i>	<i>V11</i>
12	<i>T</i>	<i>Temperature in Celsius</i>	<i>T.ts</i>

2. Regression Models :

Multiple Linear Regression:

A multiple linear regression model is a modeling technique that uses variables within the data set as predictors in order to predict a dependent variable. In this analysis, we are attempting to predict the temperature variable that is calculated in celsius. The variables that are used to estimate the temperature variable are assigned a significance level based on how much of an effect it has in predicting the outcome of temperature. In this analysis, all of the variables in the training data were used to begin our understanding of which variables play a significant role within our model. After creating the original model, an updated model was created selecting all of the significant terms in order to increase the model's adjusted R-square. The variables that were selected for the updated model are S1(CO), C6H6(GT), S2(NMHC), NOx(GT), S3(NOx), NO2(GT), S4(NO2), S5(O3), RH, and AH. The final adjusted R-squared for this model was equal to 0.925.

Model:

```
#Model with all significant Variables
mlrfit_all = lm(T.ts~V2+V3+V4+V5+V6+V7+V8+V9+V10+V11)
summary(mlrfit_all)
```

```

Call:
lm(formula = T.ts ~ V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 +
    V11)

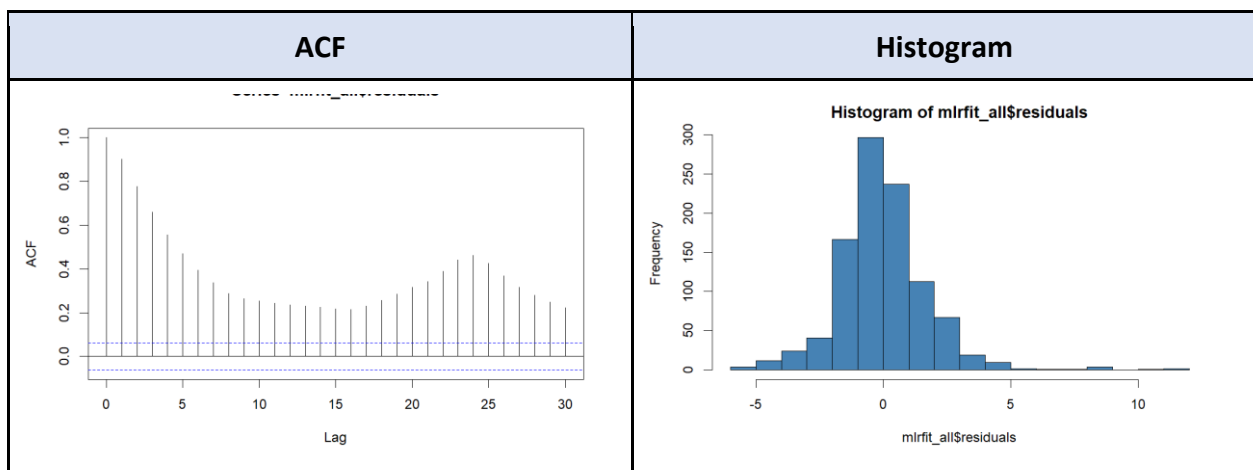
Residuals:
    Min       1Q   Median       3Q      Max
-5.3667 -0.9895 -0.2001  0.8374 11.1802

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.1507710   3.6871368   3.567 0.000379 ***
V2           0.0026052   0.0014539   1.792 0.073455 .
V3          -0.3327601   0.1210759  -2.748 0.006098 **
V4          -0.0086381   0.0030171  -2.863 0.004285 **
V5          -0.0037545   0.0010666  -3.520 0.000451 ***
V6          -0.0065978   0.0014042  -4.699 2.99e-06 ***
V7           0.0054375   0.0018137   2.998 0.002785 **
V8           0.0206498   0.0023232   8.888 < 2e-16 ***
V9          -0.0033172   0.0004175  -7.946 5.22e-15 ***
V10         -0.3245319   0.0050418 -64.369 < 2e-16 ***
V11          9.9906657   0.9377330  10.654 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.803 on 989 degrees of freedom
Multiple R-squared:  0.926,    Adjusted R-squared:  0.9252
F-statistic: 1237 on 10 and 989 DF, p-value: < 2.2e-16

```

After creating the model, the analysis continued with plotting the residuals of the regression model on an ACF plot and a histogram. From the graphs below, the ACF plot visually appears to have a slow decay which would make the residuals non-stationary. Testing to see if the model is white noise or not, the Ljung-box test was performed in order to statistically determine whether or not our residuals are white noise. Based on the results of the p-value being equal to essentially 0, we reject H_0 and accept the H_{α} and claim that the residuals are not white noise. Looking at the histogram of the residuals, they visually appear to follow a normal distribution and it was confirmed by performing the Shapiro-Wilk normality test. Based on the normality test results, we can infer that we accept H_0 and reject H_{α} and claim that the residuals are normally distributed.



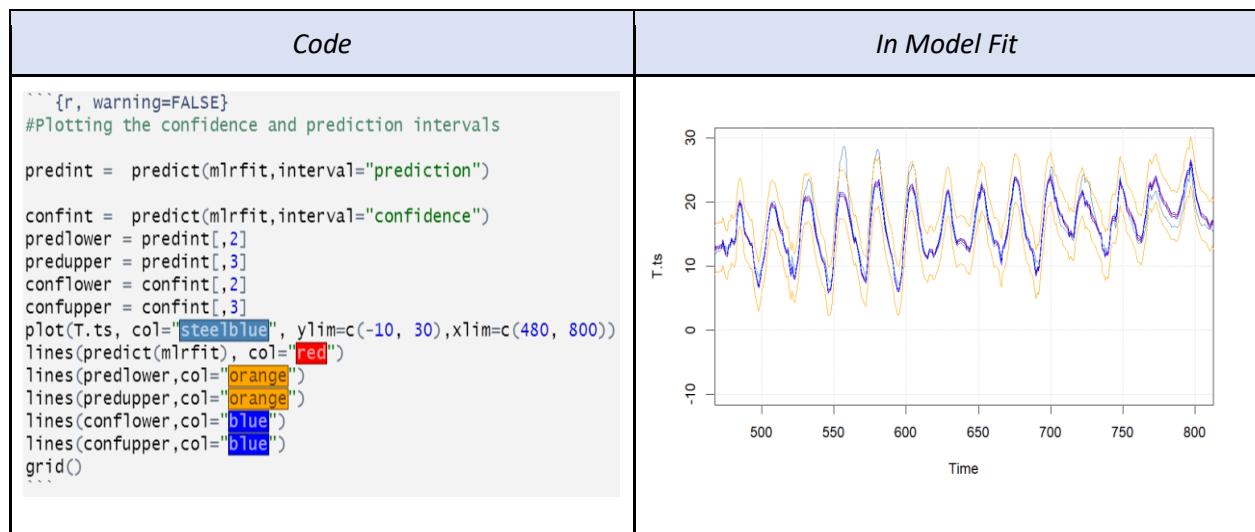
<p>Box-Ljung test</p> <p>data: mlrfit_all\$residuals</p> <p>X-squared = 3509.6, df = 20, p-value < 2.2e-16</p>	<p>Shapiro-Wilk normality test</p> <p>data: mlrfit_all\$residuals</p> <p>W = 0.92148, p-value < 2.2e-16</p>
<p>P-value is <0.05</p> <p>Thus we reject H₀ and accept the H_{alpha} and claim that the residuals are not white noise.</p>	<p>P-value is <0.05</p> <p>Thus we accept H₀ and reject H_{alpha} and claim that the residuals are normally distributed.</p>

Continuing with our analysis, a constant variance test was performed in order to determine if the model's residual had a constant variance or not. By performing the White test, the p-value results approximately zero which allows us to reject H₀ and accept H_{alpha}, we can claim that the residuals of multiple linear regression model are heteroskedastic making our assumptions are violated.

White Test														
<pre>#Constant Variance library(skedastic) white(mlrfit, interactions = TRUE)</pre>														
<p>A tibble: 1 × 5</p> <table> <thead> <tr> <th>statistic <dbl></th><th>p.value <dbl></th><th>parameter <dbl></th><th>method <chr></th><th>alternative <chr></th></tr> </thead> <tbody> <tr> <td>366.4184</td><td>1.47229e-69</td><td>14</td><td>White's Test</td><td>greater</td></tr> </tbody> </table> <p>1 row</p>					statistic <dbl>	p.value <dbl>	parameter <dbl>	method <chr>	alternative <chr>	366.4184	1.47229e-69	14	White's Test	greater
statistic <dbl>	p.value <dbl>	parameter <dbl>	method <chr>	alternative <chr>										
366.4184	1.47229e-69	14	White's Test	greater										
<p>Reject H₀ and accept H_{alpha}, we can claim that the residuals of multiple linear regression model are heteroskedastic, making our assumptions violated.</p>														

Plotted Actual data & Regression Model with confidence intervals:

According to the model plotted along with the actual training data, we can determine that the model appears to fit the real data fairly well and is able to capture the daily trends as well as the overall upward trend. For this model, the confidence intervals are also plotted in order to visualize the range of the predictions for this model.



MAPE:

To test the accuracy of the multiple linear regression model, the model was tasked with predicting the next 10 observations. Once we have estimated the next 10 observations, we then calculate the error between the testing data and the new estimated predictions in order to calculate the mean absolute error percentage. After calculating the MAPE value for this model, we can conclude that the model has a 50.18% MAPE value. This MAPE value shows that this model actually did not perform well when it comes to the prediction of the temperature variable.

```

pred_mlr = predict(mlrfit_all, n.ahead = 10)

mape_mlr<- mape(test_data$T, pred_mlr)
mape_mlr

```

```
[1] 0.5018067
```

3. Deterministic Series Models:

Seasonal Model:

A seasonal model was created for our analysis as our data has clear indications of a seasonal pattern that increases and decreases. A seasonal model does not use any of the variables within the dataset but instead uses two created variables, one for the hours of the day and another called time to capture the trend through the passage of time in the dataset. The seasonal model that was created is shown below. The 'day2' variable simply starts the sequence of the hour at the same hour as the first observation within the training data which is at 11pm or 23:00.

```
#Creating Day cycle variable (24hrs)
h <- rep(seq(1:24), 1100)

#Seasonal Model1:
time<-seq(1:length(data$T))

#Starting Time on same hour as Data
day2 <- h[23:1022]
```

```
M1<-lm(data$T~as.factor(day2)+time)
summary(M1)
```

Model 1:

The summary of the seasonal model allows us to infer the significance level of each of the different hours throughout a 24 hour day cycle. Based on the summary below, we can determine that the coefficients that are significant are from 6am through 10 pm as well as at midnight. The following model has an adjusted R-squared of .7774 and a p-value that is essentially zero.


```

call:
lm(formula = data$T ~ as.factor(day2) + time)

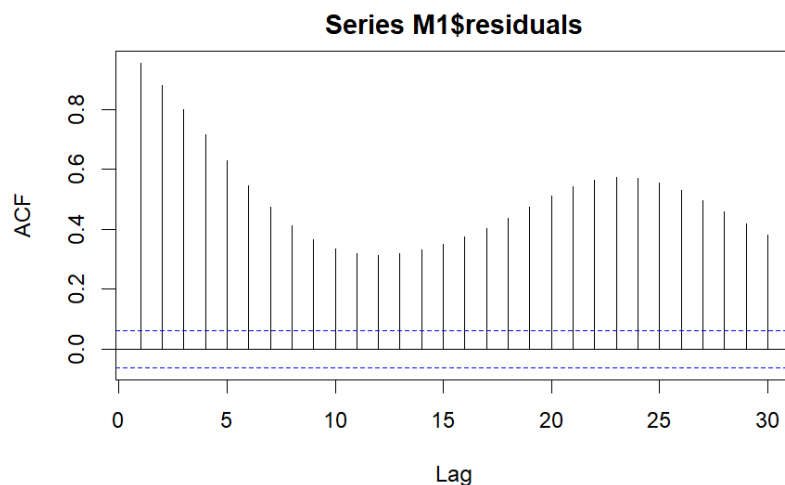
Residuals:
    Min       1Q   Median       3Q      Max
-9.1102 -2.0068  0.3641  2.1566 10.4889

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.7051986  0.5086593   17.114 < 2e-16 ***
as.factor(day2)2  0.4114985  0.6786601    0.606  0.54443
as.factor(day2)3  0.2922429  0.6786604    0.431  0.66684
as.factor(day2)4 -0.3063778  0.6786608   -0.451  0.65177
as.factor(day2)5 -2.1980540  0.6786614   -3.239  0.00124 **
as.factor(day2)6 -3.7367540  0.6786622   -5.506  4.69e-08 ***
as.factor(day2)7 -4.8006524  0.6786631   -7.074  2.88e-12 ***
as.factor(day2)8 -5.5724874  0.6786643   -8.211  6.91e-16 ***
as.factor(day2)9 -6.2838064  0.6786655   -9.259  < 2e-16 ***
as.factor(day2)10 -6.7740937  0.6786670   -9.981  < 2e-16 ***
as.factor(day2)11 -7.1620001  0.6786686  -10.553  < 2e-16 ***
as.factor(day2)12 -7.4933589  0.6786704  -11.041  < 2e-16 ***
as.factor(day2)13 -7.9913843  0.6786724  -11.775  < 2e-16 ***
as.factor(day2)14 -8.4576637  0.6786745  -12.462  < 2e-16 ***
as.factor(day2)15 -8.7020128  0.6827860  -12.745  < 2e-16 ***
as.factor(day2)16 -8.9751931  0.6827865  -13.145  < 2e-16 ***
as.factor(day2)17 -9.0558937  0.6827871  -13.263  < 2e-16 ***
as.factor(day2)18 -9.1154561  0.6827878  -13.350  < 2e-16 ***
as.factor(day2)19 -8.1593681  0.6827888  -11.950  < 2e-16 ***
as.factor(day2)20 -6.5896622  0.6827899   -9.651  < 2e-16 ***
as.factor(day2)21 -4.5073546  0.6827911   -6.601  6.68e-11 ***
as.factor(day2)22 -2.8043154  0.6827926   -4.107  4.34e-05 ***
as.factor(day2)23 -1.3299412  0.6786604   -1.960  0.05032 .
as.factor(day2)24 -0.4230063  0.6786601   -0.623  0.53324
time          0.0166762  0.0003407   48.944 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.11 on 975 degrees of freedom
Multiple R-squared:  0.7828,    Adjusted R-squared:  0.7774
F-statistic: 146.4 on 24 and 975 DF,  p-value: < 2.2e-16

```

Residuals:



Box-Ljung test

```

data: M1$residuals
X-squared = 5730.8, df = 20, p-value < 2.2e-16

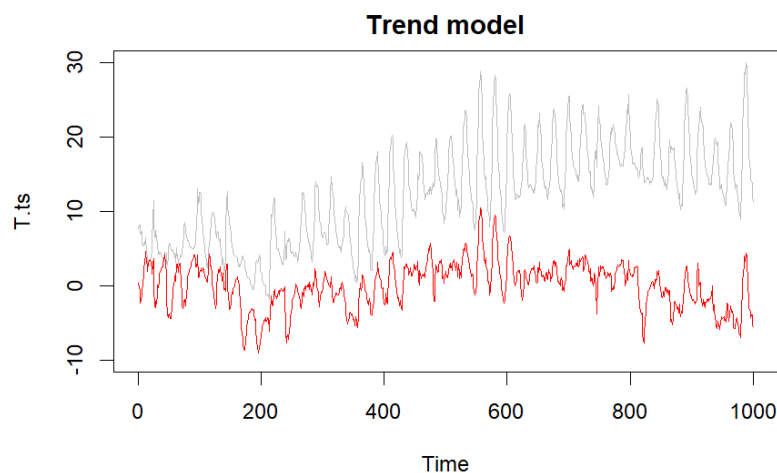
```

Looking at the residuals of the seasonal model, we can visually determine that the acf plot is non-stationary as the lags decay slowly over time. In order to statistically prove that the residuals are not white noise, we performed the box-test in order to determine if the p-value is greater or less than alpha (0.05). The result of the box test has a p-value that is essentially zero which means we reject H_0

and accept the H_α and claim that the residuals are not white noise. And when we plotted the box plot by introducing the seasonality we were able to produce the graph below.

Model Fit:

Looking at the plot below, we can infer that the model does not fit the training data well and is predicting the temperature variable to be much lower than it actually is. Although the day cycle is somewhat captured by this model, it is clear that the overall trend of the model is slightly decreasing over time while the real values from the training data is much higher and appears to be increasing over time.



MAPE:

Based on the calculations below, the trend model has a MAPE value that is equal to 47.89% which is a very high MAPE percentage and confirms that this model is in fact not a good fit for the data provided.

```
mape.trend<- mean(abs(test_data$T - m1_pred)/test_data$T)
mape.trend
```

```
[1] 0.4789454
```

Polynomial Model:

The polynomial model or trend model, is an alternative method of predicting data as it does not use any of the variables within the data set as its parameters. In this analysis, the training data was used for the creation of a simple linear model with 'k' as the number of polynomials used within the model. The model also utilizes a variable called 'time' which is the length of the data at each index to represent the passage of time in order to better capture trends that are occurring over time. Having both 'k' and 'time' as predictors for the model, the goal is to capture both season cycles and determine whether the data is increasing or decreasing over time. For the model used in this analysis, a 'k' of 24 was selected in order to have the best fit. Below we can see the creation of both variables as well as the polynomial model labeled as 'M2'.

Model:

```
time<-seq(1:length(data$T))
k=24

M2<-lm(data$T~poly(time,k))
```

```
Call:
lm(formula = data$T ~ poly(time, k))

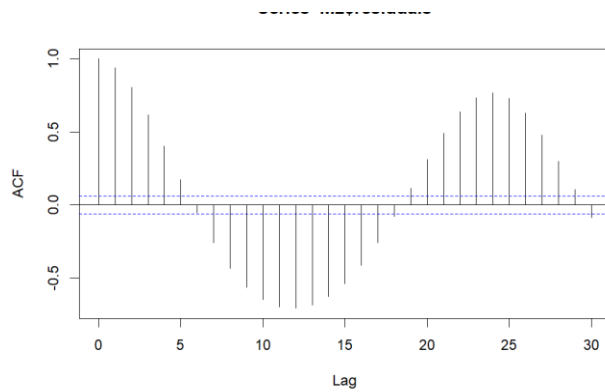
Residuals:
    Min       1Q   Median       3Q      Max
-9.1642 -2.5445 -0.6164  2.3252 11.9281

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  12.0807    0.1238   97.595 < 2e-16 ***
poly(time, k)1  151.3668    3.9144   38.669 < 2e-16 ***
poly(time, k)2  -28.6240    3.9144  -7.312 5.46e-13 ***
poly(time, k)3  -46.1915    3.9144 -11.800 < 2e-16 ***
poly(time, k)4   23.1029    3.9144   5.902 4.95e-09 ***
poly(time, k)5   12.3766    3.9144   3.162 0.001616 **
poly(time, k)6   -9.4525    3.9144  -2.415 0.015926 *
poly(time, k)7    8.9557    3.9144   2.288 0.02358 *
poly(time, k)8    6.0483    3.9144   1.545 0.122638
poly(time, k)9   -3.0321    3.9144  -0.775 0.438760
poly(time, k)10   5.8620    3.9144   1.498 0.134575
poly(time, k)11  -13.1001    3.9144  -3.347 0.000849 ***
poly(time, k)12   14.0130    3.9144   3.580 0.000361 ***
poly(time, k)13    4.7926    3.9144   1.224 0.221113
poly(time, k)14  -21.5053    3.9144  -5.494 5.02e-08 ***
poly(time, k)15   -1.4103    3.9144  -0.360 0.718720
poly(time, k)16  -12.5977    3.9144  -3.218 0.001332 **
poly(time, k)17  -17.4628    3.9144  -4.461 9.10e-06 ***
poly(time, k)18    3.4089    3.9144   0.871 0.384041
poly(time, k)19  -11.3181    3.9144  -2.891 0.003920 **
poly(time, k)20  -10.6181    3.9144  -2.713 0.006794 **
poly(time, k)21   -2.1728    3.9144  -0.555 0.578965
poly(time, k)22    3.1816    3.9144   0.813 0.416537
poly(time, k)23   -6.5489    3.9144  -1.673 0.094640 .
poly(time, k)24  -6.7634    3.9144  -1.728 0.084337 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.914 on 975 degrees of freedom
Multiple R-squared:  0.6559,    Adjusted R-squared:  0.6474
F-statistic: 77.44 on 24 and 975 DF,  p-value: < 2.2e-16
```

The optimal 'k' value was determined based on the high order polynomials still having significance and the adjusted R-squared of the model continued to rise until it peaked at 0.6474. This model was able to achieve the highest R-squared, the model's residuals were then plotted on an acf plot, and it was finally tested against the testing data in order to determine its predictive ability with new incoming data.

Residuals:



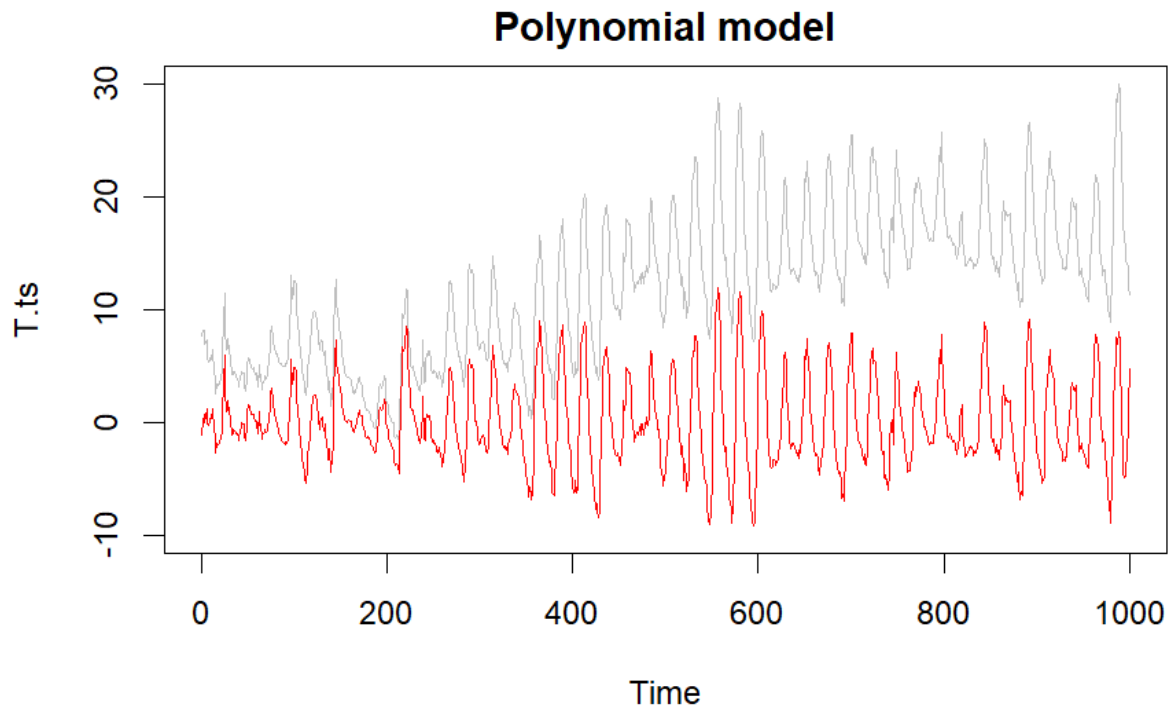
Box-Ljung test

```
data: M2$residuals
X-squared = 5630.9, df = 20, p-value < 2.2e-16
```

Based on the model's acf plot, visually we can infer that the model does decay slowly over time and has a cyclical pattern which is why we can claim the model is non-stationary. After performing the Ljung-Box test, we can infer that p-value is < 0.05 for this model thus we reject H_0 and accept the H_a and claim that the residuals are not white noise.

Model vs. Training Time Series:

By looking at the model plotted against the training data, it is clear that this model does not capture all the elements that can impact the variable temperature. The model does appear to perform well when capturing the impact of the hourly trend but is not able to capture the upward trend that is occurring over the course of the data set.



MAPE:

By taking the mean absolute error percentage of this model against the 10 new observations within the testing data set, we can infer that the model has a MAPE value of 63.92% which proves that this model does not perform well when it comes to the prediction of the independent variable temperature.

```
pred[2,1:10] = predict(M2, data.frame(time=c(test_data$T)))  
mape_poly<- mean(abs(test_data$T - pred[2,1:10])/test_data$T)
```

```
...
```

```
[1] 0.6392381
```

Harmonic Model:

The harmonic model is a type of model that captures seasonality or periodicity in the data by including sine and cosine functions with different frequencies. A harmonic model assumes that the data can be decomposed into different cycles of different frequencies, which are characterized by a set of harmonics. Each harmonic corresponds to a sinusoidal wave of a particular frequency and amplitude, and these harmonics are added together to form the overall seasonal component of the model.

The general form of a harmonic model is:

$$Y_t = \mu + S_t + E_t$$

where Y_t is the observed time series, μ is the overall mean level, S_t is the seasonal component modeled as a sum of harmonics, and E_t is the error term.

The seasonal component S_t is typically modeled as a sum of sine and cosine waves, with the number of harmonics determined by the periodicity of the data. For example, if the data has a yearly seasonality, the harmonic model may include 12 harmonics (one for each month). The amplitude and phase of each harmonic are estimated from the data using methods such as least squares regression.

Firstly, we did detrend the series by using a time variable created as part of model 1. Then using the TSA package we input the detrended residuals to periodogram which will get the estimate of the power spectral density of the time series data. And we captured the spikes from periodogram and performed ordering of those estimated power using `$spec` on residuals.

And based on no. of spikes we picked those powers and produced sine and cosine waves. Below is the code associated with the same.

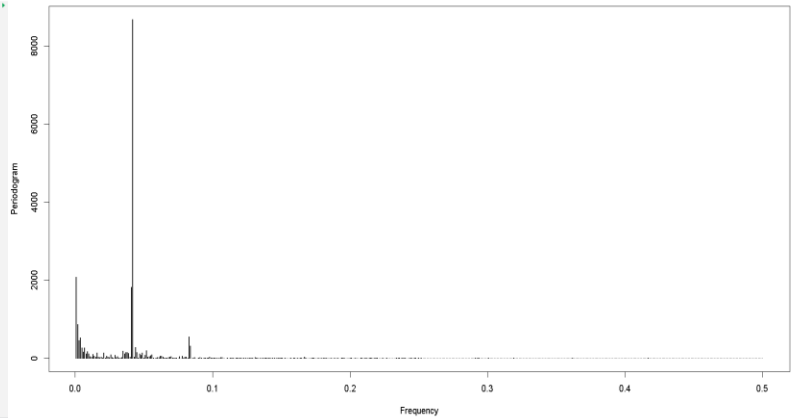
```
## [r]
library(TSA)
#library(M3) #sin/cos
detrnd<-lm(data$T~time)

#periodogram
periodogram(detrnd$residuals)
periodogram(detrnd$residuals)$spec
periodogram(detrnd$residuals)$freq

#ordering the residuals to identify any seasonality and use them in subsequently in sin/cos.
order(periodogram(detrnd$residuals)$spec)
n<-length(data$T)

sin1<-sin(2*pi*time/42/n)
cos1<-cos(2*pi*time/42/n)
sin2<-sin(2*pi*time/1/n)
cos2<-cos(2*pi*time/1/n)
sin3<-sin(2*pi*time/4/n)
cos3<-cos(2*pi*time/4/n)
sin4<-sin(2*pi*time/2/n)
cos4<-cos(2*pi*time/2/n)
sin5<-sin(2*pi*time/3/n)
cos5<-cos(2*pi*time/3/n)
sin6<-sin(2*pi*time/4/n)
cos6<-cos(2*pi*time/4/n)
sin7<-sin(2*pi*time/1/n)
cos7<-cos(2*pi*time/1/n)
sin8<-sin(2*pi*time/8/n)
cos8<-cos(2*pi*time/8/n)
sin9<-sin(2*pi*time/4/n)
cos9<-cos(2*pi*time/4/n)
sin10<-sin(2*pi*time/5/n)
cos10<-cos(2*pi*time/5/n)
sin11<-sin(2*pi*time/7/n)
cos11<-cos(2*pi*time/7/n)
sin12<-sin(2*pi*time/12/n)
cos12<-cos(2*pi*time/12/n)

M3<-lm(data$T~time+sin1+cos1+sin2+cos2+sin3+cos3+sin4+cos4+sin5+cos5+sin6+cos6+sin7+cos7+sin8+cos8+sin9+cos9+sin10+cos10+sin11+cos11+sin12+cos12)
summary(M3)
acf(M3$residuals)
pacf(M3$residuals)
```



Finally, after looking at the summary of the M3 model below we can infer that the Adjusted-R square is higher compared after removing in-significant coefficients. So we went ahead considering the M3 model without taking the in-significant coefficients out.

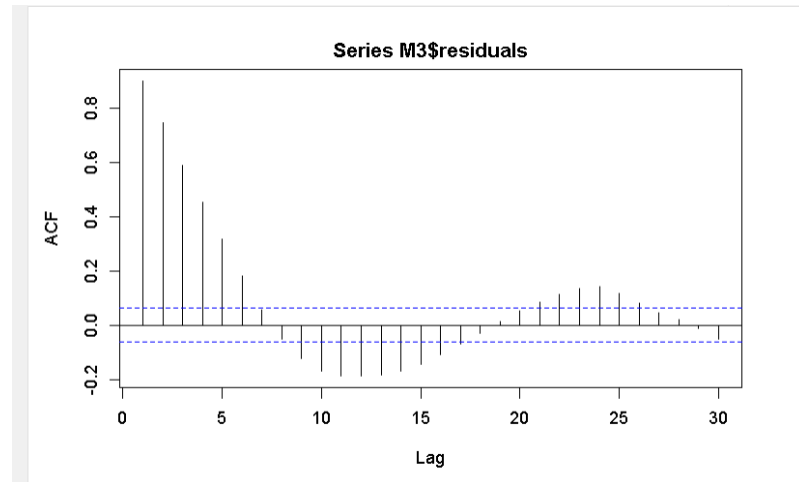
```
Call:
lm(formula = data$T ~ time + sin1 + cos1 + sin2 + cos2 + sin3 +
    cos3 + sin4 + cos4 + sin5 + cos5 + sin6 + cos6 + sin7 + cos7 +
    sin8 + cos8 + sin9 + cos9 + sin10 + cos10 + sin11 + cos11 +
    sin12 + cos12)

Residuals:
    Min       1Q   Median       3Q      Max
-7.4603 -1.2797  0.0459  1.1995  8.5522

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.2510815   0.3779390   13.894 < 2e-16 ***
time          0.0136455   0.0007433   18.358 < 2e-16 ***
sin1         2.8437652   0.0943731   30.133 < 2e-16 ***
cos1        -2.7384485   0.0942097  -29.068 < 2e-16 ***
sin2        -2.2675064   0.2546647   -8.904 < 2e-16 ***
cos2        -1.5183062   0.0942097  -16.116 < 2e-16 ***
sin3        -2.0687965   0.0943814  -21.920 < 2e-16 ***
cos3         0.9540026   0.0942097   10.126 < 2e-16 ***
sin4         0.6112857   0.1512264    4.042 5.71e-05 ***
cos4         0.7687694   0.0942097    8.160 1.03e-15 ***
sin5         0.5189740   0.0942479    5.506 4.68e-08 ***
cos5         1.0578078   0.0942097   11.228 < 2e-16 ***
sin6         0.7510998   0.1112352    6.752 2.50e-11 ***
cos6         0.2961566   0.0942097    3.144 0.00172 **
sin7         0.6450528   0.1228597    5.250 1.86e-07 ***
cos7        -0.0704214   0.0942097   -0.747 0.45494
sin8        -0.4509565   0.0942469   -4.785 1.98e-06 ***
cos8        -0.3771061   0.0942097   -4.003 6.73e-05 ***
sin9         0.4650187   0.0943582    4.928 9.75e-07 ***
cos9        -0.4283862   0.0942097   -4.547 6.12e-06 ***
sin10        0.0144653   0.1054217    0.137 0.89089
cos10       -0.6759953   0.0942097   -7.175 1.43e-12 ***
sin11       -0.1909629   0.1000849   -1.908 0.05668 .
cos11        0.7359093   0.0942097    7.811 1.46e-14 ***
sin12        0.5642889   0.0943146    5.983 3.07e-09 ***
cos12       -0.2292810   0.0942097   -2.434 0.01512 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.107 on 974 degrees of freedom
Multiple R-squared:  0.9004,    Adjusted R-squared:  0.8979
F-statistic: 352.4 on 25 and 974 DF, p-value: < 2.2e-16
```

Residuals - M3



Box-Ljung test

```
data: M3$residuals
X-squared = 2283.4, df = 20, p-value < 2.2e-16
```

Based on the model's acf plot, visually we can infer that the model does decay slowly over time and has a cyclical pattern which is why we can claim the model is non-stationary. After performing the Ljung-Box test, we can infer that p-value is < 0.05 for this model thus we reject H_0 and accept the H_a and claim that the residuals are not white noise.

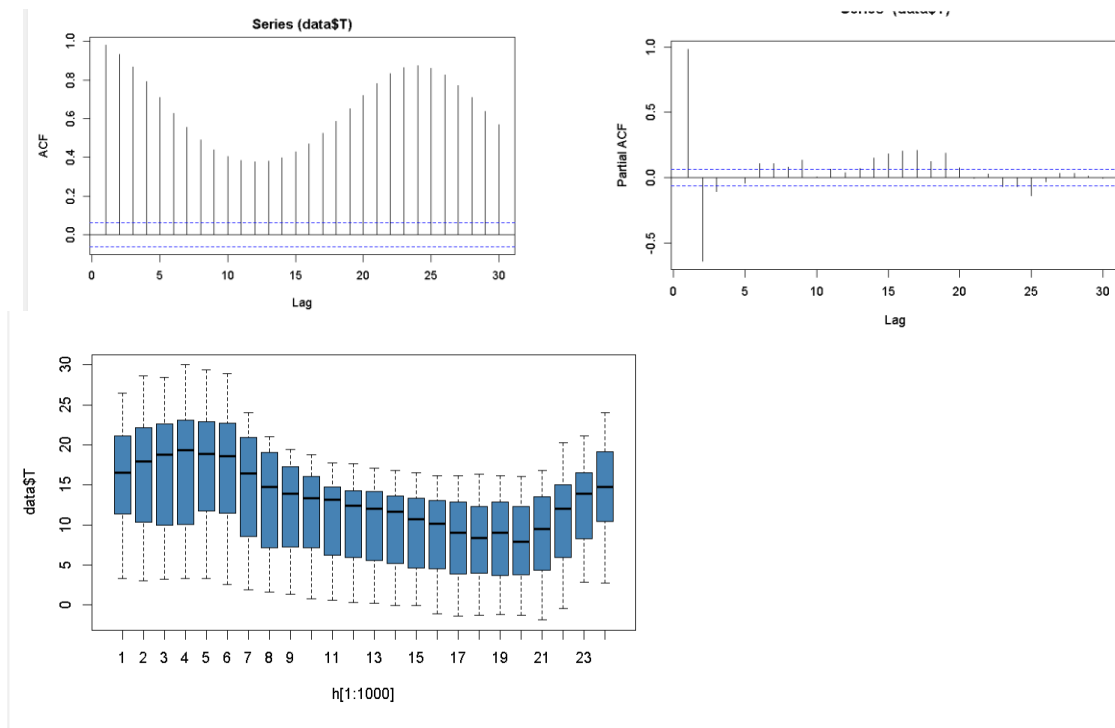
1.

4. Stochastic Time Series Models:

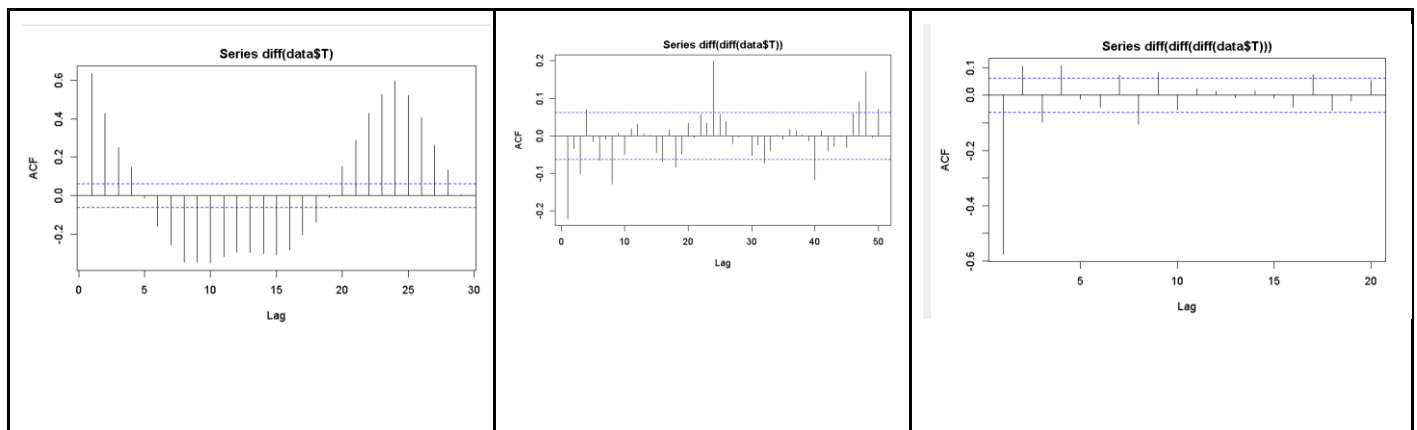
Stochastic time series models used to describe the behaviour of the time series data. These models presume that values of a time series are the result of a stochastic(random) process that is influenced by internal and external factors. As part of this process we have the following types that we discussed in our class and performed the same.

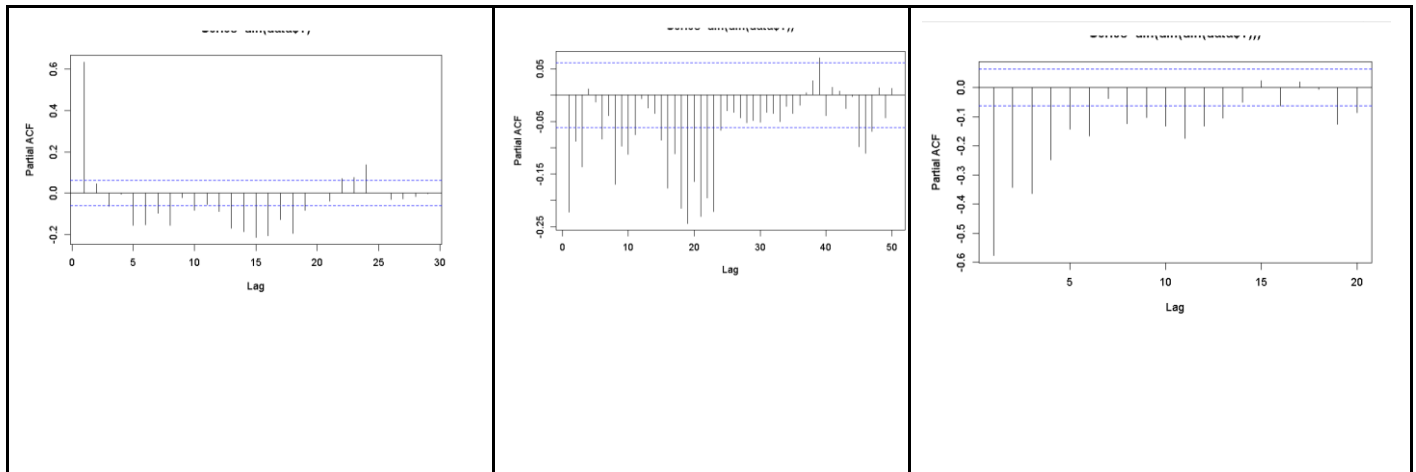
AR, MA, ARIMA, SARIMA and Vector autoregression model.

Below is the time Series plot followed by ACF and PACF.



Based on acf and pacf plots data seems to be non-stationary. To bring some stationarity we performed first order , second order and third order.





The ACF plot is also chopped off and there is no slow decay. As we have higher order AR and MA processes, we considered an to use the ARIMA model (2,1), (1,1) & (2,2). Out of these we got AR(2) and MA(1) with lowest AIC.

```
#ARIMA
ar_ma1 <- arima(x=data$T,order=c(2,1,1) )
ar_ma1
#AIC 2912.9

ar_ma2 <- arima(x=data$T,order=c(1,1,1) )
ar_ma2
#AIC:2915.21

ar_ma3 <- arima(x=data$T,order=c(2,1,2) )
ar_ma3 #2871.61
```

Call:
arima(x = data\$T, order = c(2, 1, 1))

Coefficients:

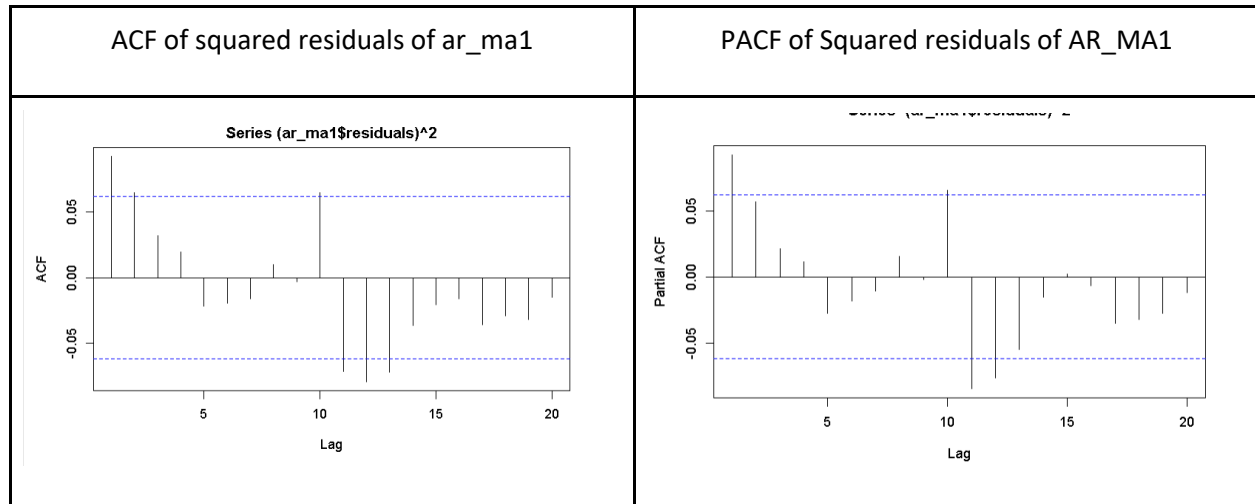
	ar1	ar2	ma1
	0.0124	0.4308	0.5837
s.e.	0.1498	0.0924	0.1534

sigma^2 estimated as 1.074: log likelihood = -1453.45, aic = 2912.9

Box-Ljung test

data: ar_ma1\$residuals
X-squared = 119.59, df = 20, p-value = 3.331e-16

we can infer that p-value is < 0.05 for this model thus we reject H_0 and accept the H_a and claim that the residuals are not white noise. And we plotted the squared residuals of the ARIMA



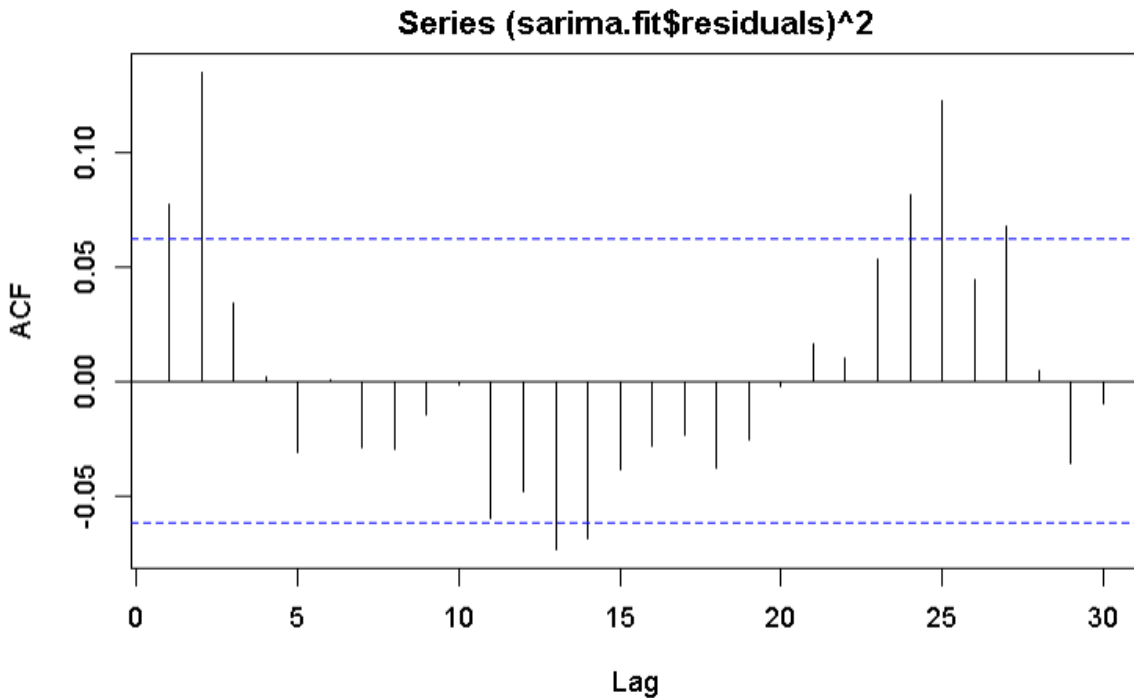
From above we can infer that residuals are not white noise. We can try doing the ARCH/GARCH model to see if the residuals are white noise.

SARIMA Fit: This model will extend ARIMA to include seasonality. And we performed using the order(2,1,2). And we inferred that p-value is < 0.05 for this model thus we reject H_0 and accept the H_a and claim that the residuals are not white noise.

```
Call:
arima(x = data$T, order = c(2, 1, 2), seasonal = list(order = c(1, 0, 1), period = 24))

Coefficients:
      ar1      ar2      ma1      ma2      sar1      sm1
    -0.4451 -0.0336  0.7689  0.2769  0.9835 -0.8376
s.e.   0.1646   0.1107  0.1602  0.1066  0.0059  0.0253

sigma^2 estimated as 0.7541:  log likelihood = -1290.51,  aic = 2593.01
```



Box-Ljung test

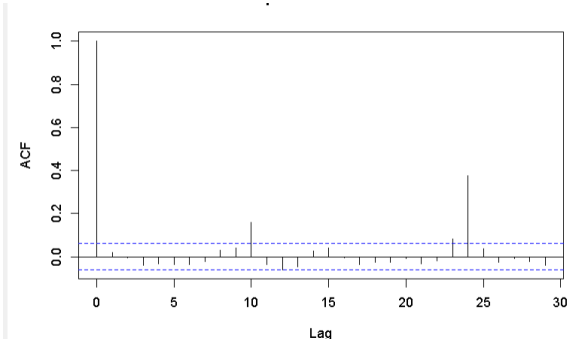
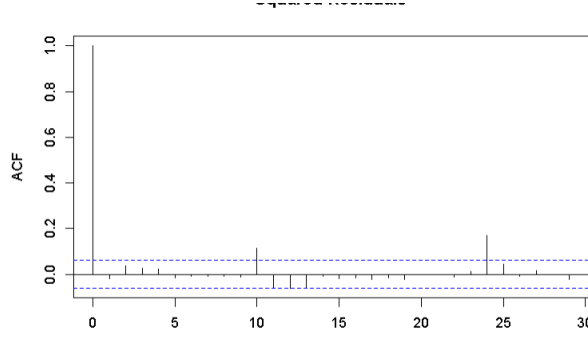
```
data: (sarima.fit$residuals)^2
X-squared = 49.661, df = 20, p-value = 0.0002477
```

AARCH/GARCH Model:

2. The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model is a time series model used to analyze and forecast the volatility of financial returns or other time series. The GARCH model assumes that the conditional variance of the time series is a function of its past values and past values of its own squared error terms. Specifically, the GARCH(p,q) model is defined as:

$$h_t = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \dots + \alpha_q \epsilon_{t-q}^2 + \beta_1 h_{t-1} + \dots + \beta_p h_{t-p},$$

We have applied the AARCH and GARCH model to see if our residuals are white noise. To start with we have applied (2,2) and we observed the residuals aren't white noise and $p\text{-value} < 0.05$. In the case of GARCH model we applied for (2,2) and infer that $p\text{-value} > 0.05$

AARCH Model(2,2)	GARCH Model(2,2)
 <p>ACF plot for AARCH Model(2,2). The x-axis is 'Lag' from 0 to 30, and the y-axis is 'ACF' from 0.0 to 1.0. A significant spike is visible at lag 24, indicating non-white noise residuals.</p>	 <p>ACF plot for GARCH Model(2,2). The x-axis is 'Lag' from 0 to 30, and the y-axis is 'ACF' from 0.0 to 1.0. All spikes are within the confidence interval, indicating white noise residuals.</p>
<p>Box-Pierce test</p> <pre>data: (arch.fit\$residuals/arch.fit@sigma.t)^2 X-squared = 46.894, df = 20, p-value = 0.0006068</pre>	<p>Box-Pierce test</p> <pre>data: (garch.fit\$residuals/garch.fit@sigma.t)^2 X-squared = 28.456, df = 20, p-value = 0.09903</pre>

Finally, we can say that from GARCH model box test that $p\text{-value} > 0.05$ for this model thus we fail to reject H_0 and claim that the residuals are white noise.

Re-estimating the Models:

Reestimating the MLR models using ARMA(1,1,1)

```
Call:
arima(x = mlrfit$residuals, order = c(1, 1, 1))

Coefficients:
      ar1      ma1
    0.1928  0.1512
s.e.  0.0790  0.0777

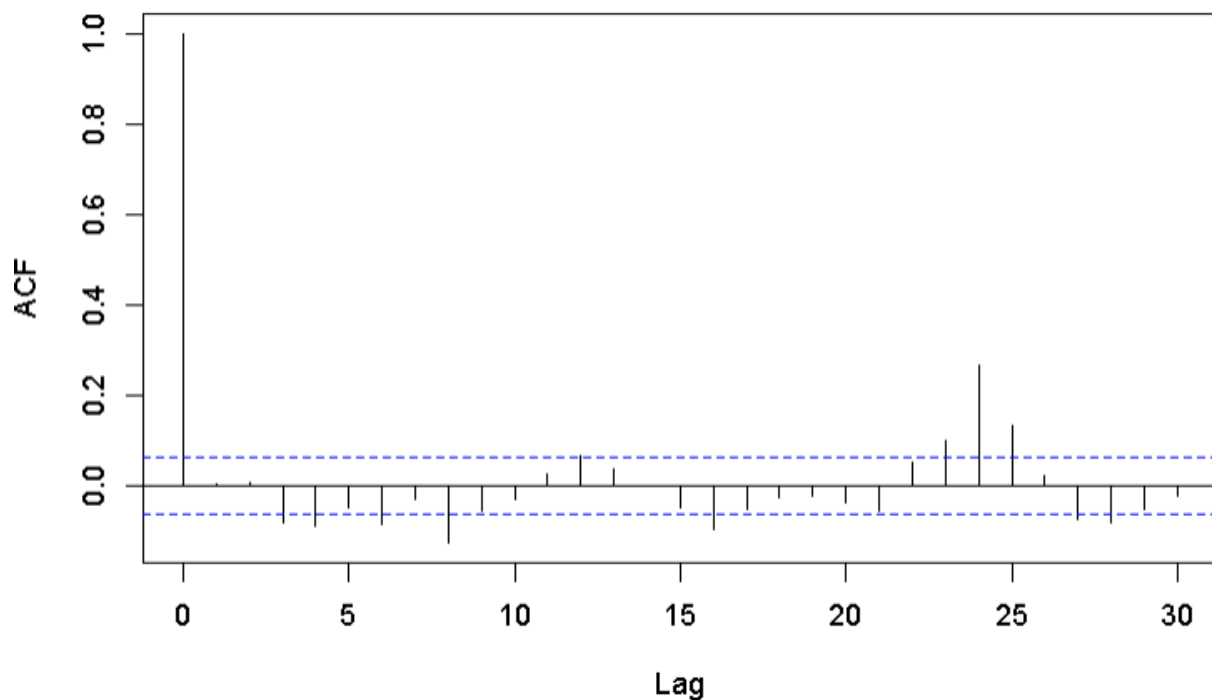
sigma^2 estimated as 0.4128:  log likelihood = -975.62,  aic = 1957.23
```

Box-Ljung test

```
data: rearima_fit$residuals  
X-squared = 69.164, df = 20, p-value = 2.493e-07
```

Based on the results of the p-value being equal to essentially 0, we reject H_0 and accept the H_a and claim that the residuals are not white noise

ACF



Re-estimation of Seasonal Model:

```
Call:  
arima(x = M1$residuals, order = c(2, 1, 2))
```

Coefficients:

	ar1	ar2	ma1	ma2
	-0.3793	0.0198	0.7074	0.2231
s.e.	0.1717	0.1174	0.1693	0.1161

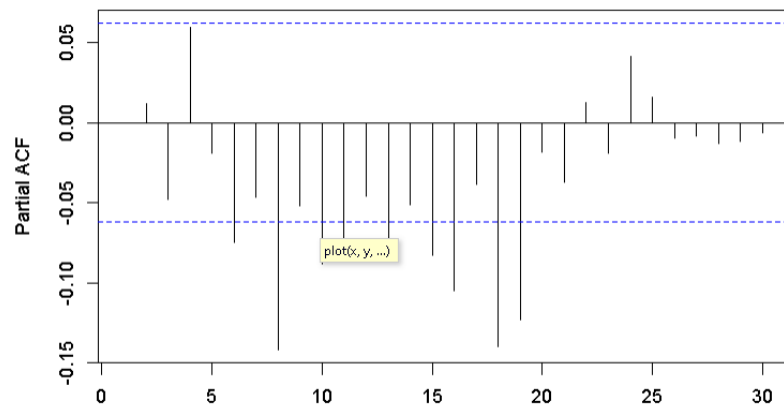
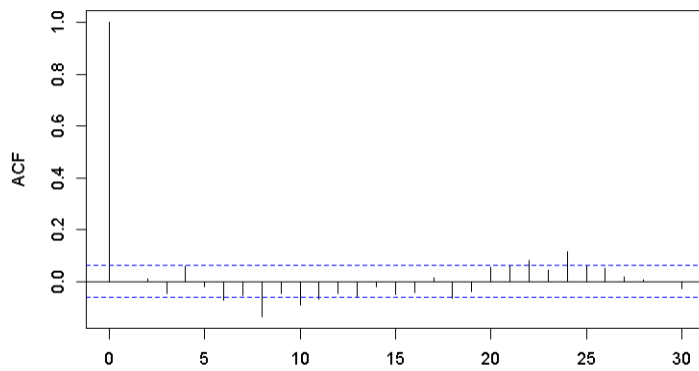
```
sigma^2 estimated as 0.7318: log likelihood = -1261.6, aic = 2533.21
```

```
> |
```

Box-Ljung test

```
data: reesti_m1_fit$residuals  
x-squared = 72.109, df = 20, p-value = 8.211e-08
```

Based on the results of the p-value being equal to essentially 0, we reject H_0 and accept the H_a and claim that the residuals are not white noise



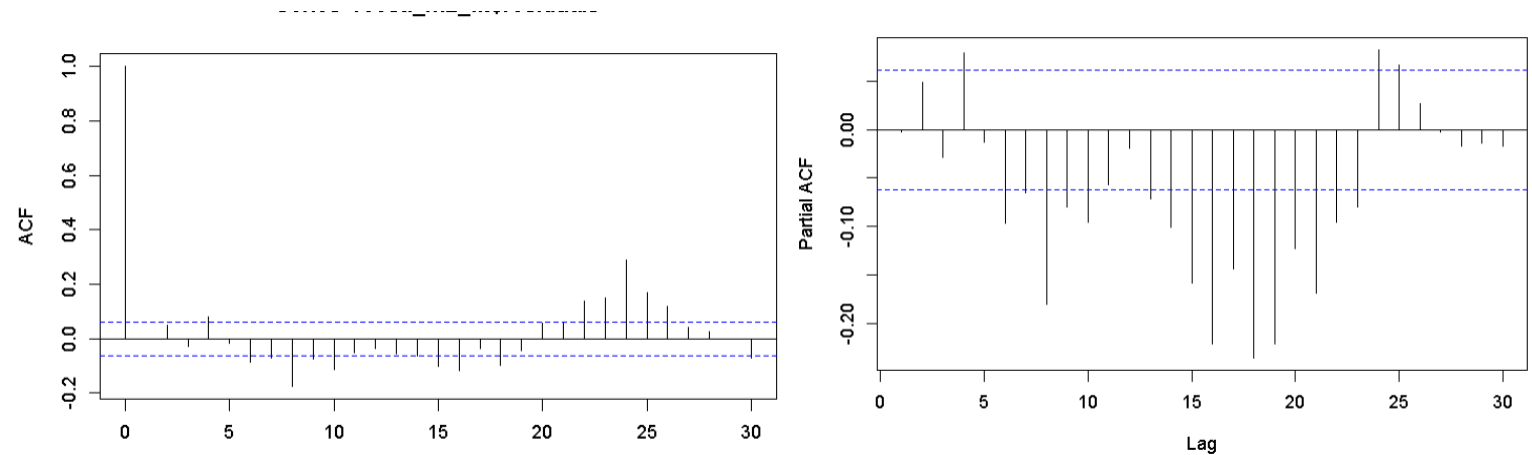
3. Reestimate of M2(Polynomial Model):

```
Call:
arima(x = M2$residuals, order = c(1, 1, 1))
```

Coefficients:

	ar1	ma1
	0.6682	-0.0609
s.e.	0.0350	0.0450

```
sigma^2 estimated as 1.078: log likelihood = -1455.19, aic = 2916.37
```

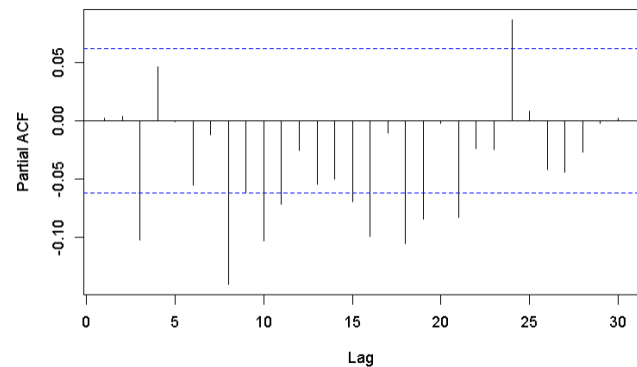
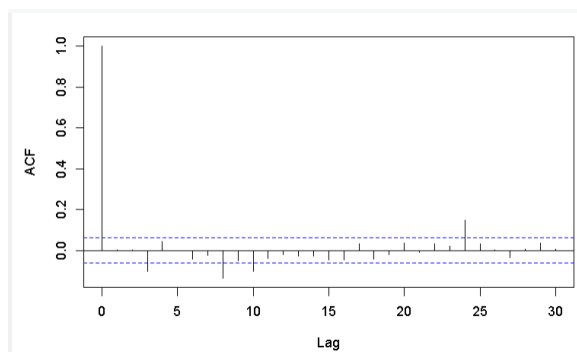


Box-Ljung test

```
ata: reesti_m2_fit$residuals
:-squared = 125.73, df = 20, p-value < 2.2e-16
```

Based on the results of the p-value being equal to essentially 0, we reject H_0 and accept the H_a and claim that the residuals are not white noise

4. Reestimate of M3(Harmonic Model):



```
arima(x = M3$residuals, order = c(1, 1, 1))
```

Coefficients:

	ar1	ma1
	0.1677	0.1183
s.e.	0.0865	0.0846

5. σ^2 estimated as 0.771: log likelihood = -1287.68, aic = 2581.36

Box-Ljung test

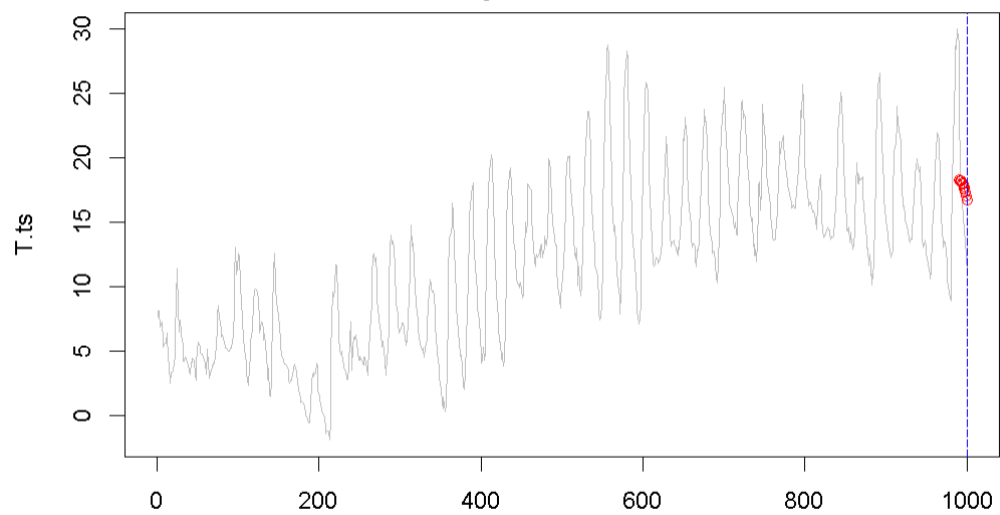
data: reesti_m3_fit\$residuals

X-squared = 59.315, df = 20, p-value = 9.087e-06

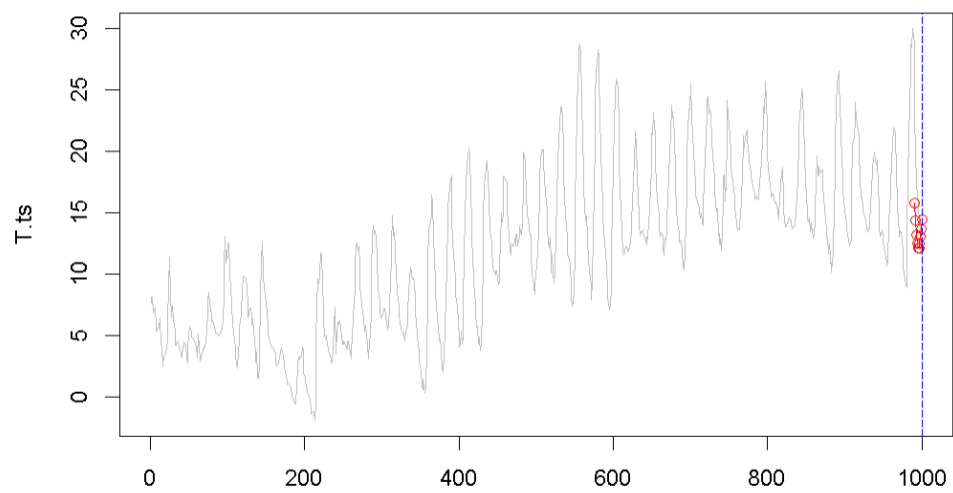
Based on the results of the p-value being equal to essentially 0, we reject H_0 and accept the H_a and claim that the residuals are not white noise

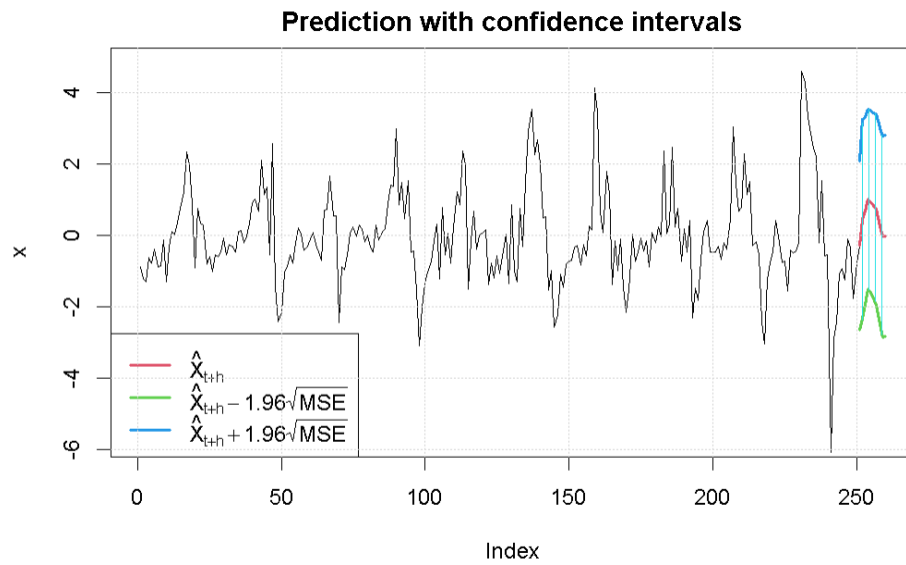
Predictive Comparison:

Polynomial Model

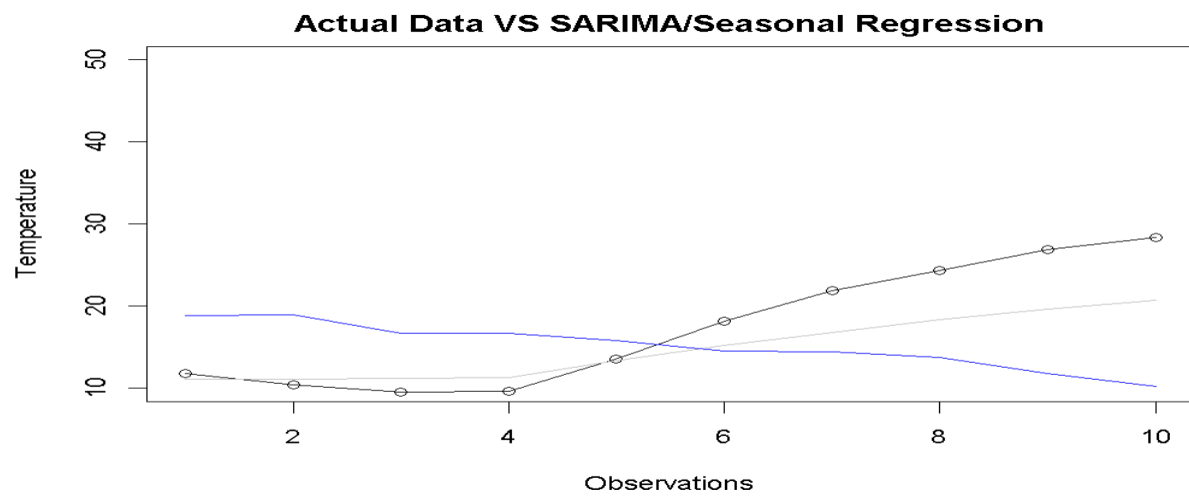


Harmonic Model





6.



Method.List	MAPE
POLY	0.5856009
HARMONIC	0.5651711
MLR	0.5005007
SARIMA	0.1657316
GARCH	0.3308825

The Mean Absolute Percentage Error on the model we performed. We can state from the comparison that the SARMIA model is having the lowest MAPE rate at 16% and the Polynomial model is estimated high at 58%.

5. Multivariate Series Models:

Varma Model:

A Vector Autoregressive Moving Average (VARMA) model is a stochastic time series model used to analyze the relationship between multiple time series. The VARMA model is an extension of the Autoregressive Moving Average (ARMA) model to multivariate time series.

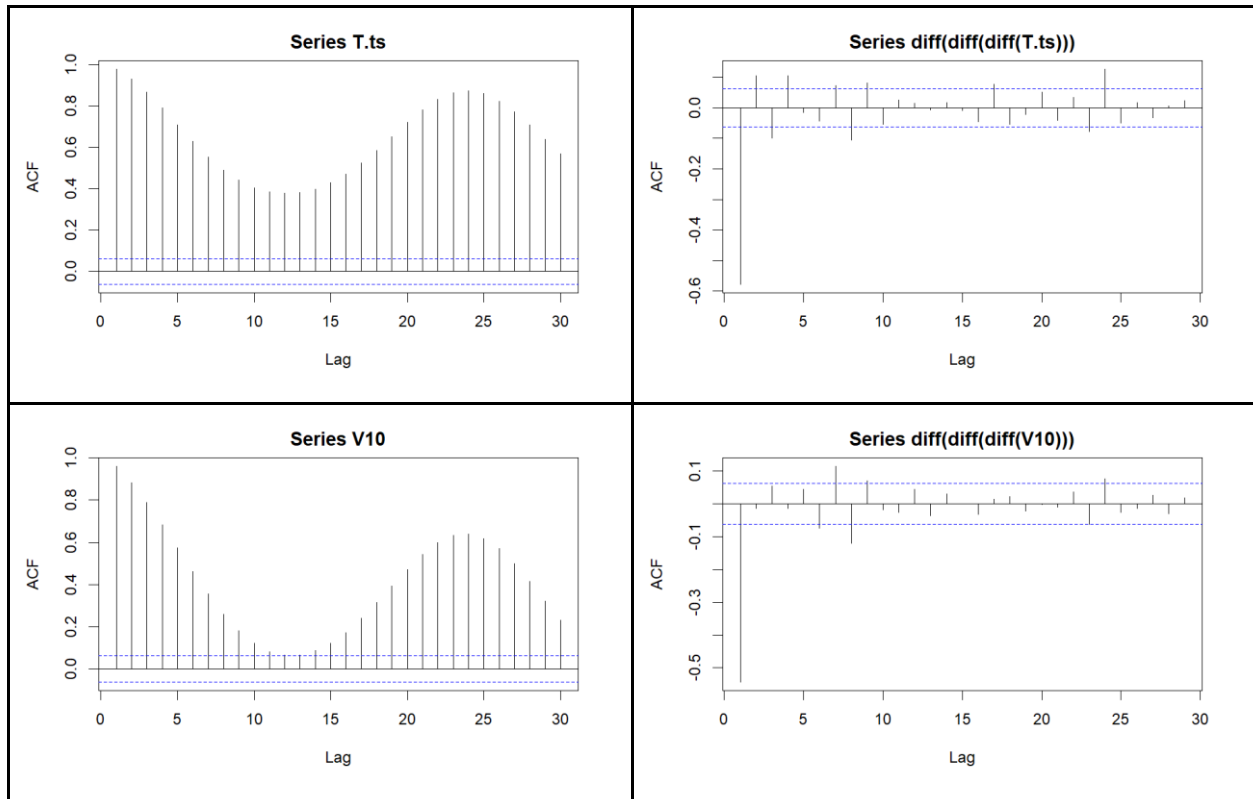
A VARMA(p,q) model consists of p autoregressive (AR) terms and q moving average (MA) terms for each of the k time series being modeled. The model is expressed in matrix notation as:

$$Y_t = \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \dots + \Phi_p Y_{t-p} + \epsilon_t$$

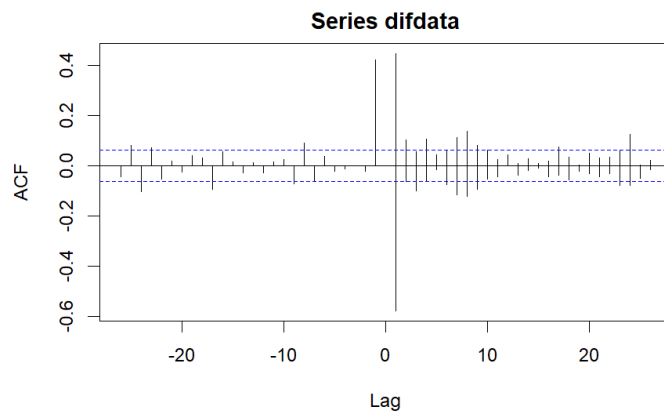
Inducing Stationarity:

In order to work with the VARMA model, both the variables in the model which are Temperature and Relative Humidity need to be stationary. Based on the initial acf plots of both variables, we can determine that they both are stationary as the significance of the lags are slowly decaying over time and does not see a sudden drop off. To induce stationarity, the third difference for both variables was taken and as we can see in the table below they both ended up becoming white noise

ACF	Third Difference ACF
-----	----------------------



CCF: Third difference of both variables



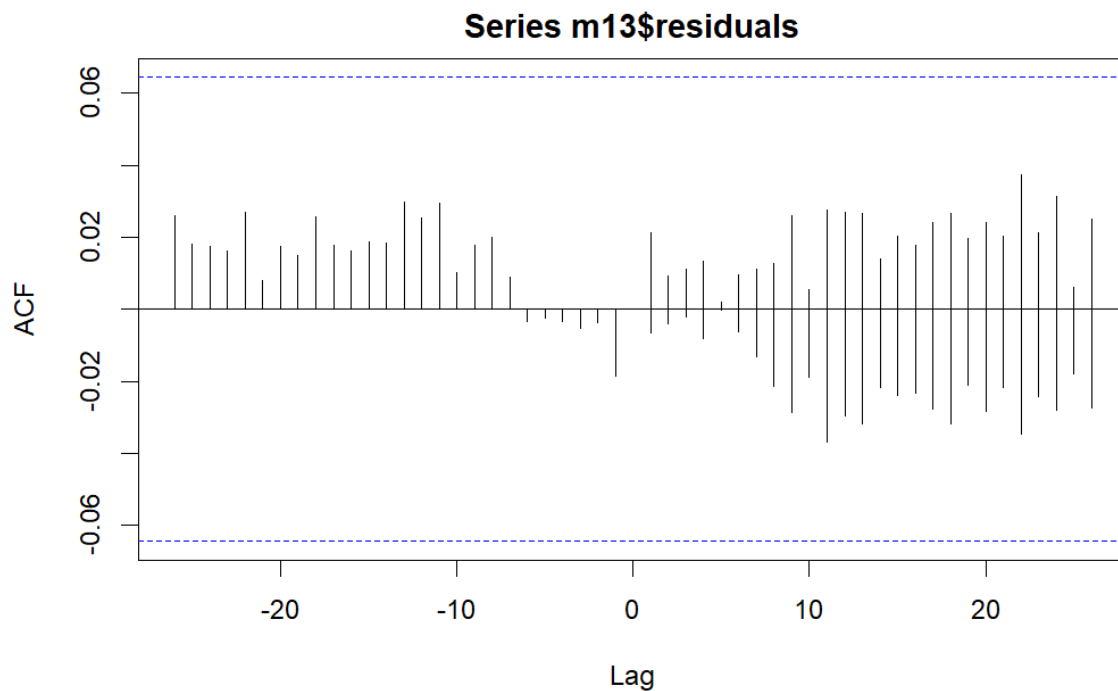
In order to properly determine the appropriate 'P' for this model, the VARorder function was utilized up to $P=100$ to determine at which level the AIC of the model would be at the lowest point. Based on the results from the VARorder, the best P came out to be equal to 71 since it is at that point where the AIC is minimized. The model was also created below using the third difference of both variables referred to as 'difdata' and used a P equal to 71.

```
VARorder(difdata,100)  
m13=VARMA(difdata,p=71,q=0)
```

```
selected order: aic = 71  
selected order: bic = 26  
selected order: hq = 38  
Summary table:
```

P=71

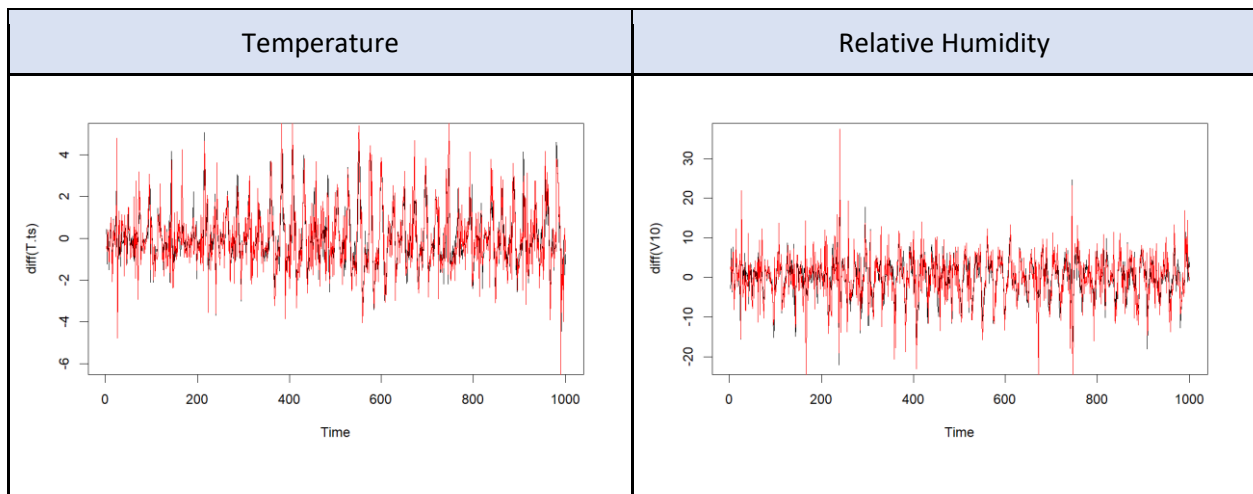
From the varma model that was created, we can see the residuals of the model plotted below. The residuals of the model are indeed white noise as there is no significance at any lags which allows us to interpret that this model was able to capture all aspects of the variables. The model at P equal to 71 was able to achieve an AIC rating of 1.778315 which also shows that this model is a good fit for the given data.



```
aic= 1.778315  
bic= 3.185295
```

Model vs Training Data:

Based on the fitted plots below, we can visually claim that the model was able to accurately predict both of the variables with high accuracy. For both plots, the model's predictions represented by the red line is always very near or the same as the actual training data.



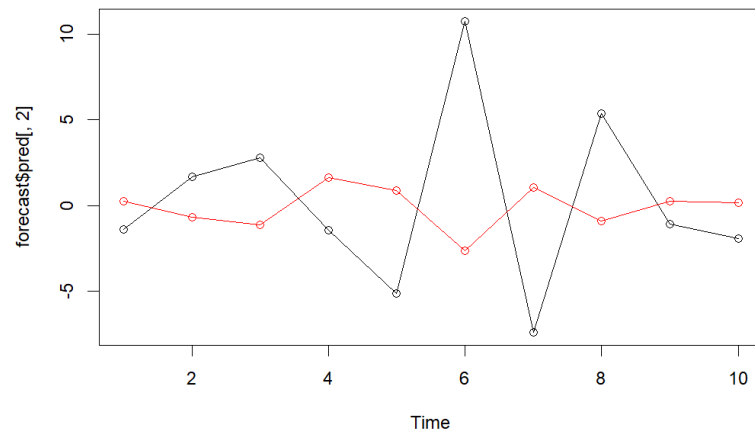
10-Step Ahead Prediction:

For this model, we proceeded by taking the predictions of the next 10 observations for each of the variables and then plotted them on a chart in order to visually see how the model estimates the next 10 periods. The plot below shows the predictions for each hour for the next 10 hours and the variable temperature is represented in black and relative humidity is represented as red.

#Predictions and Prediction Plots

```
forecast<-VARMAPred(m13,h=10)
```

```
plot.ts(forecast$pred[,2], type="o")  
lines(forecast$pred[,1], col="red", type="o")
```



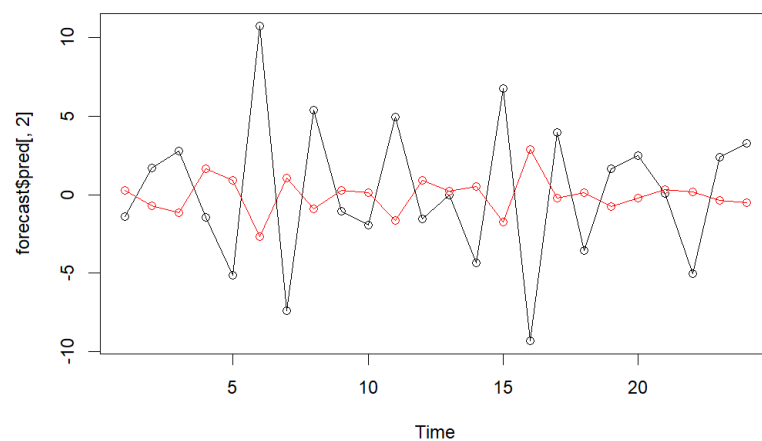
Since the data set we are working with for this analysis is hourly data, we went ahead and plotted the predictions for 1 day in the future which is equivalent to 24 hours. In the plot below, we can see the predictions for each hour for the next day for both temperature represented in black and relative humidity represented in red.

Prediction 24-ahead (1 Day):

#Predictions and Prediction Plots

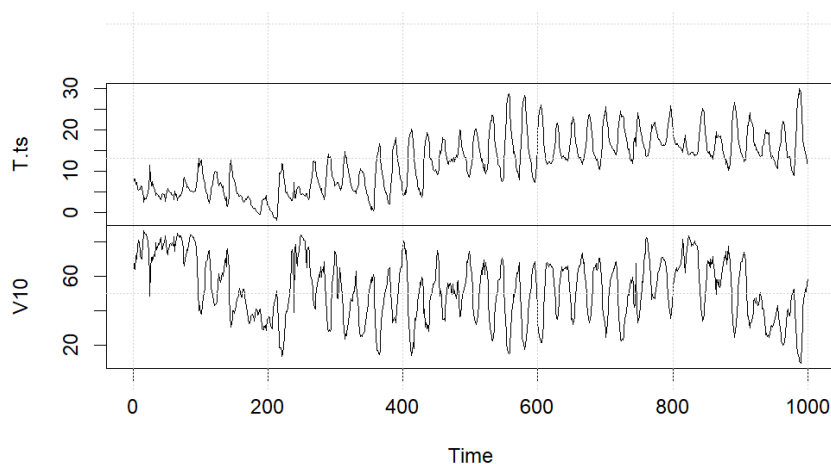
```
forecast<-VARMAPred(m13,h=24)
```

```
plot.ts(forecast$pred[,2], type="o")  
lines(forecast$pred[,1], col="red", type="o")
```



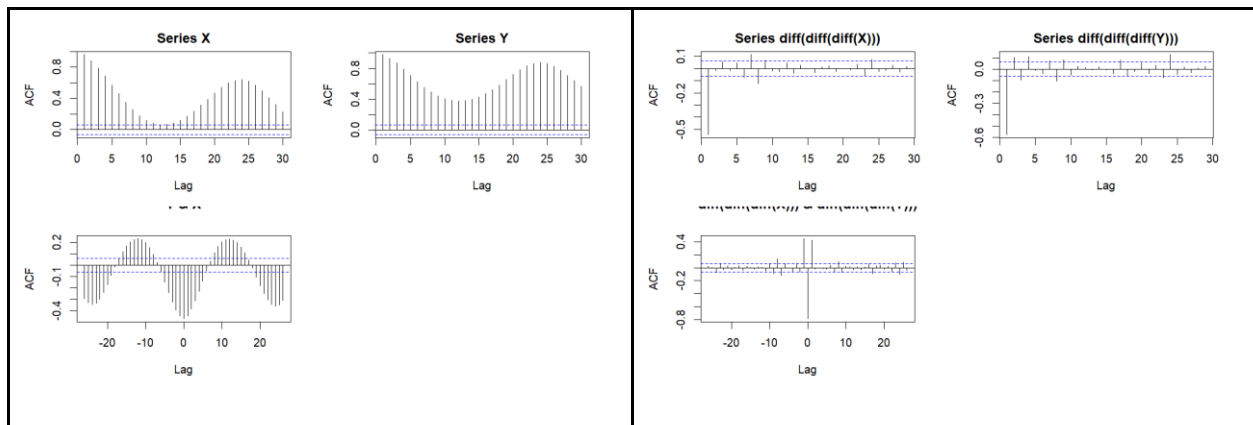
Transfer Function:

The two variables that were used for the transfer function model were 'Temperature' and 'Relative Humidity'. In order to understand if there is any indication of co-movement, both the time series of both variables were plotted in order to visually determine if there are any similar patterns. By looking at the plots below, we can see that the relative humidity and temperature both have similar daily up and down cycles and there also might be co-movement over the course of the data set but the temperature seems to be increasing over time more than the relative humidity.



In order to create the transfer function, both of the variables need to be stationary and more ideally white noise. Based on the initial acf plot, we can visually determine that both of the variables are non-stationary as there is a clear slow decay of the significance spikes over time. In order to induce stationarity, the third difference was the take of both variables which gave us the result of both variables now being stationary. Having both acf plots being stationary, we can now utilize the ccf plot in order to determine the order of our transfer function. The ccf plot has a clear spike at lag 0 which is expected and ignored, then the only other significance is at lag 1 for both series. The final order of the transfer function is (1,0,0) because the significance is at lag 1 then drops off and becomes white noise beyond that point.

ACF & CCF	Third Difference ACF & CCF
-----------	----------------------------



The creation of the model is shown below. The model uses an order of $b=1$, $r=0$, and $s=0$ and does not include the mean of the model as it only appears to worsen the AIC of the model. Based on the coefficients of the model we can claim that ar_1 , $X_{n2}-ar_1$, and $X_{n2}-MA(0)$ since when taking their value and dividing by their standard error, the result is greater than 2. The model achieved an AIC rating of 3288.57.

```
m4<-arimax(Yn2, order=c(1,0,0),xtransf=data.frame(Xn2), transfer=list(c(1,0)), include.mean = TRUE)
m4
```

```
par(mfrow=c(1,2))
acf(m4$residuals[2:length(m4$residuals)], main="")
acf(m4$residuals[2:length(m4$residuals)]^2, main="")
```

Call:

```
arimax(x = Yn2, order = c(1, 0, 0), include.mean = TRUE, xtransf = data.frame(Xn2),
      transfer = list(c(1, 0)))
```

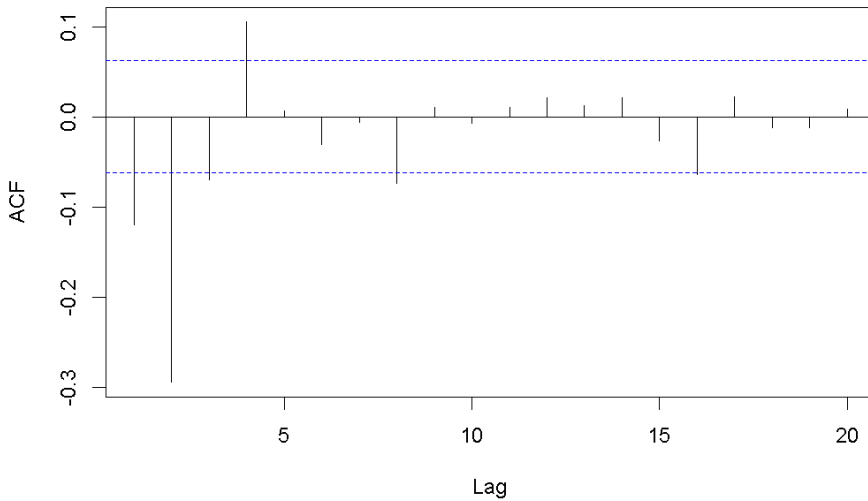
Coefficients:

	ar1	intercept	Xn2-AR1	Xn2-MA0
	-0.4986	0.0457	0.9938	0.1521
s.e.	0.0294	0.0564	0.0068	0.0083

sigma² estimated as 1.577: log likelihood = -1640.28, aic = 3288.57

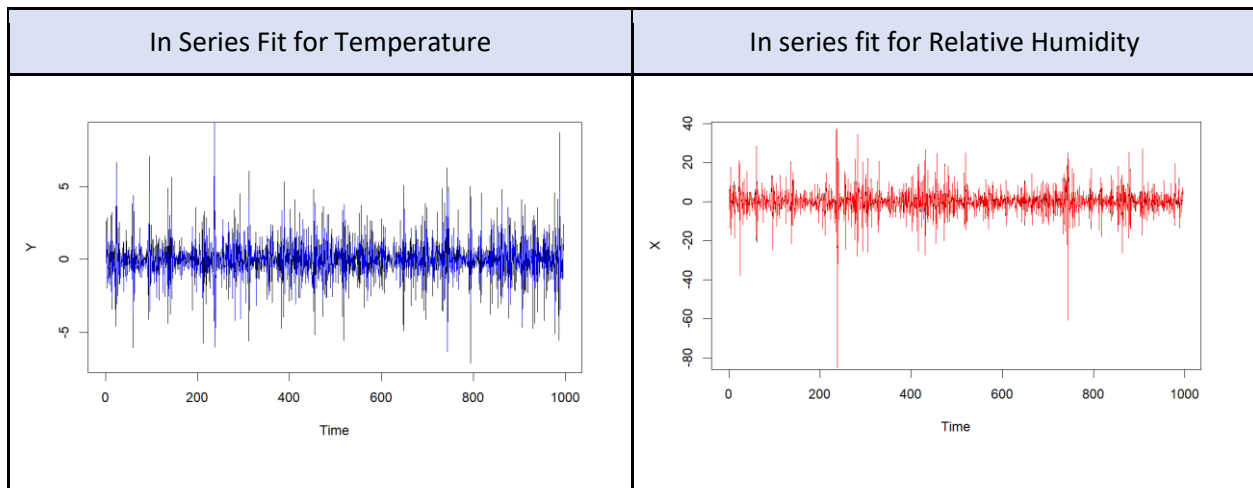
Residuals:

The residuals of this model show that after the initial significant spike at lag 2 and possibly lag 3 but after the initial spikes the rest of the series becomes white noise.



In model fit:

When looking at how well the transfer function performed when plotted against the actual data from the training set. Based on the graph below, the models appear to fit the data fairly well and managed to capture a majority of the large spikes that occur over the course of the time series. Simply based on the model fit against the training data, this model appears to perform well and be able to capture the elements that are influencing our predicted variable



6. Conclusion

- We observed the seasonality within temperature every day is dropping at 7 AM

- From accuracy perspective, SARIMA outstood compared to other models by looking MAPE

Future Scope	Possible Improvements
Integrating model into real world scenarios 7. Agriculture, Weather forecasting	Attempt to integrate more variables 1. Wind Speed, UV index, Weather, Latitude, Ocean Currents
Different Estimation methods for missing values 1. Mean/Median, KNN AI	Testing performance in different settings 1. Other countries, different pollution levels
Improve the Accuracy of the model in forecasting the temperature	Try alternative models not yet tested.