Applying ARIMA-SARIMA models for time series analysis on Seasonal and Nonseasonal datasets

KISLAY

Department of Mathematical Sciences, Stevens Institute of Technology, Hoboken, NJ
Project Supervisor: Dr. Hadi Safari Katesari

ABSTRACT

Time series analysis is a technique used to examine data in order to identify patterns and make future predic- tions. In this project, we will conduct time series analyses on two kinds of data: seasonal and non-seasonal. Using R, we will establish a process for analyzing and modeling time series data. The first part of the project involves analyzing and forecasting daily electric production data from a electric production dataset. The second part focuses on the time series of food prices in India, with the goal of analyzing and forecasting monthly prices for specific com- modities in particular regions. Various approaches to time series analysis include autoregressive integrated moving average (ARIMA), seasonal autoregressive integrated moving average (SARIMA), autoregressive moving average (ARMA), moving average (MA), and autoregression (AR). This project's main objective is to provide a comprehensive guide to ARIMA models, examining their combined output and effectiveness in time series modeling and forecasting.

PART A

SEASONAL DATASET: ELECTRIC PRODUCTION

INTRODUCTION

This project consists of predicting the daily production of electricity by using time series and will also predict the future daily production. The data set used here has only 2 columns, once column is date and the other column relates to the consumption percentage. The problem statement is estimating the production of electricity daily for the Dataset. The main purpose of this project is to find the future forecasting of the daily sales of Bakery. The data set used covers a period from January 2016 to December 2017, and includes transaction of bakery items.This can inform future pricing or sales strategies or reveal opportunities for investment.

```
library(data.table)
library(ggplot2)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(tseries)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(TSA)
```

```
## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

DATA DESCRIPTION

Dataset : Dataset is taken from Kaggle.

https://www.kaggle.com/datasets/kandij/electric-production/data

It contains only 2 columns, one column is Date and the other column relates to the consumption percentage.

It shows the consumption of electricity from 1985 till 2018. The goal is to predict electricity consumption for the next years i.e. till 2019.

ANALYSIS AND RESULTS

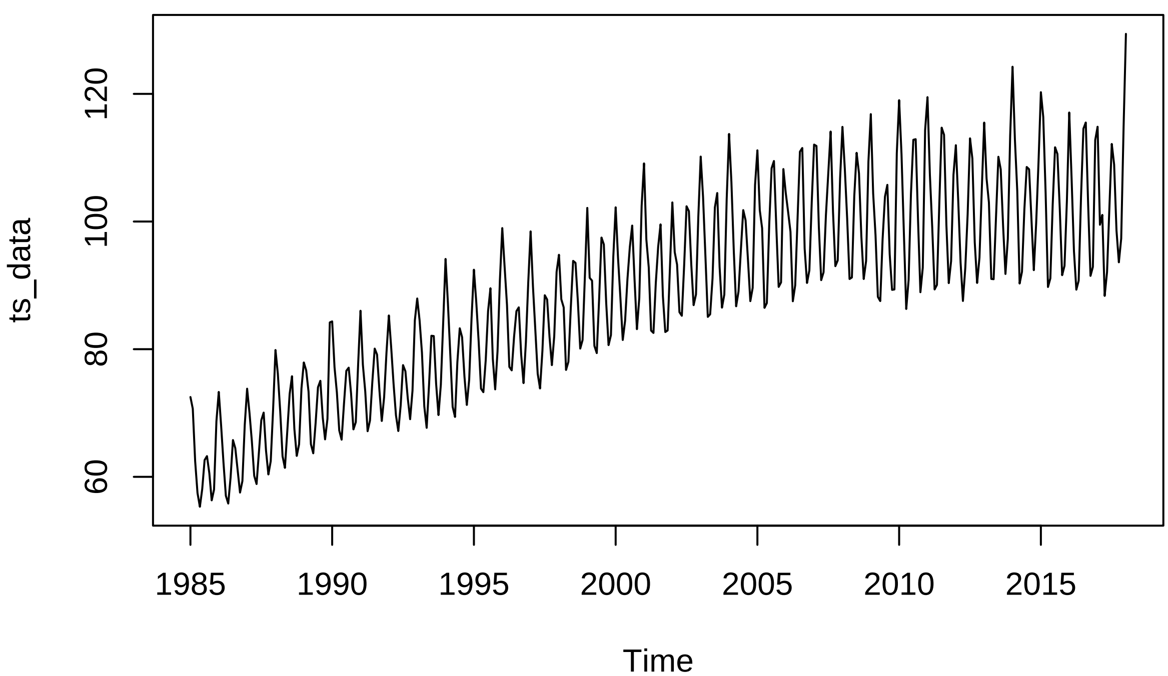
Preprocessing

Extract Date and Month column, convert Date to DateTime format.

```
data <- read.csv("/Users/kislaynandan/Desktop/MA 641/Electric_Production.csv")
df = data
setDT(data)
df$DateTime <- as.POSIXct(paste(df$DATE), format="%Y-%m-%d")
df$Month = month(df$DateTime)

df$Year = year(df$DateTime)
```

```
ts_data <- ts(df$Value, start = min(df$Year), end = max(df$Year), frequency = 12)
plot(ts_data)
```



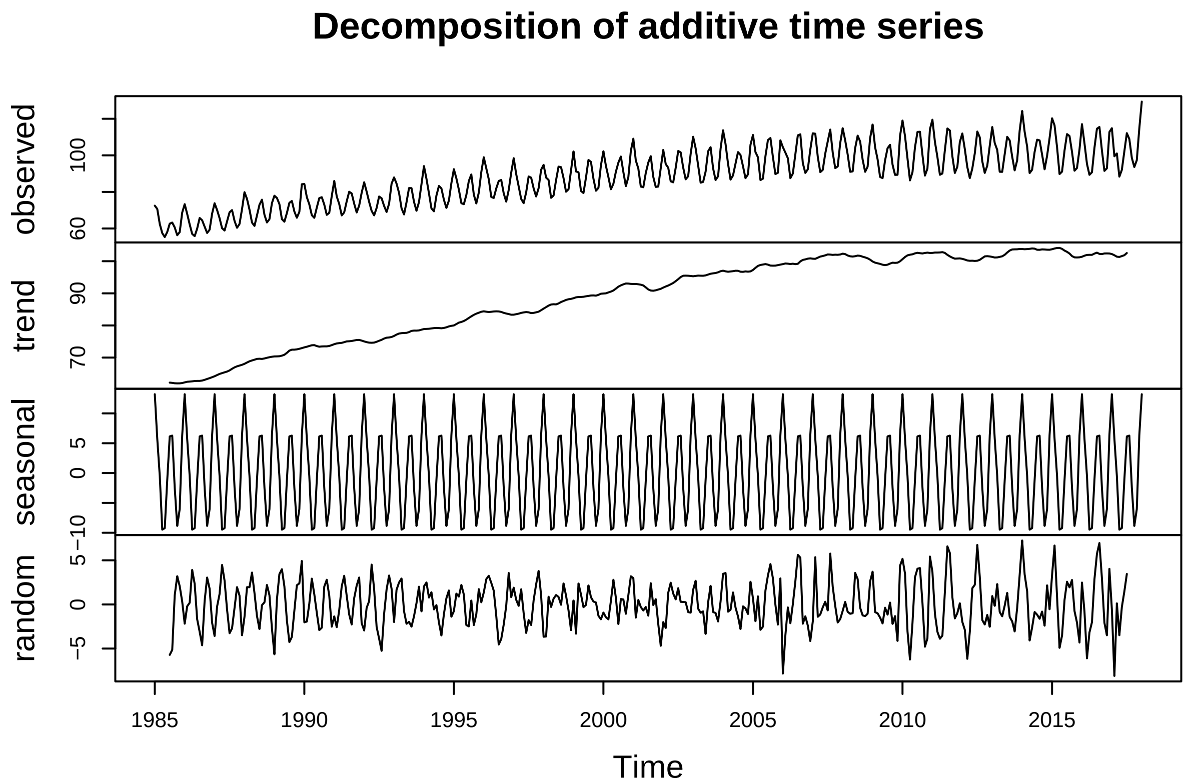
Electricity unit value is converted to time series and plotted.

The plot displays the daily production of electricity over time. It appears to have a time series object with a frequency of 12, which indicates yearly seasonality.

Decomposition

Decomposition is a technique used in time series analysis to break down a time series into its individual components, namely trend, seasonality, and residual (or error). It allows us to understand the underlying patterns and characteristics of the time series, making it easier to analyze and forecast.

```r
decomp_result <- decompose(ts_data)
plot(decomp_result)
```



Stationarity Test

Stationarity is a fundamental concept in time series analysis. A stationary time series is one whose statistical properties, such as mean and variance, remain constant over time. It implies that the series has a consistent behavior, and its patterns are predictable over different time periods.

```r
result <- adf.test(ts_data)
```

```
## Warning in adf.test(ts_data): p-value smaller than printed p-value
```

```r
result
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_data
## Dickey-Fuller = -5.139, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```r
cat("p-value:", result$p.value)
```

```
## p-value: 0.01
```

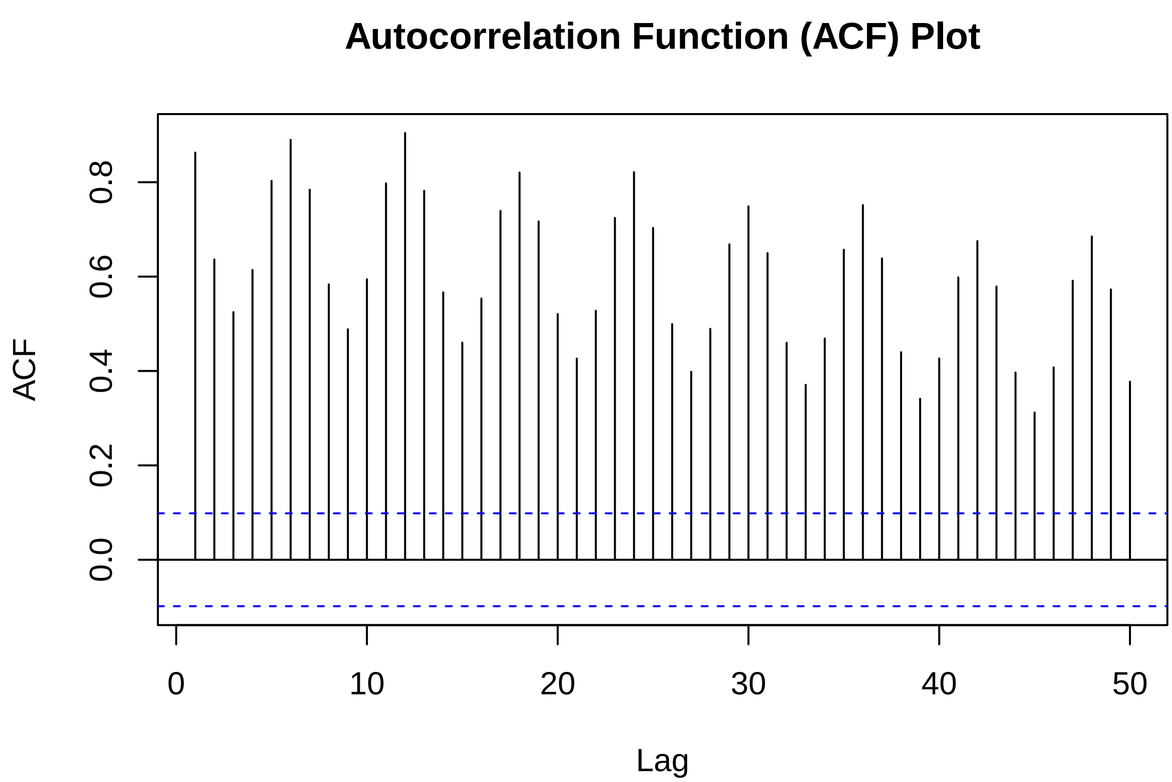
As p value is less than 0.05 the series is Stationary. Since it's stationary,the ARIMA model of order (p,0,q) will be used where 'p' is the order of the AR term and 'q' is the order of the MA.

ACF, PACF & EACF

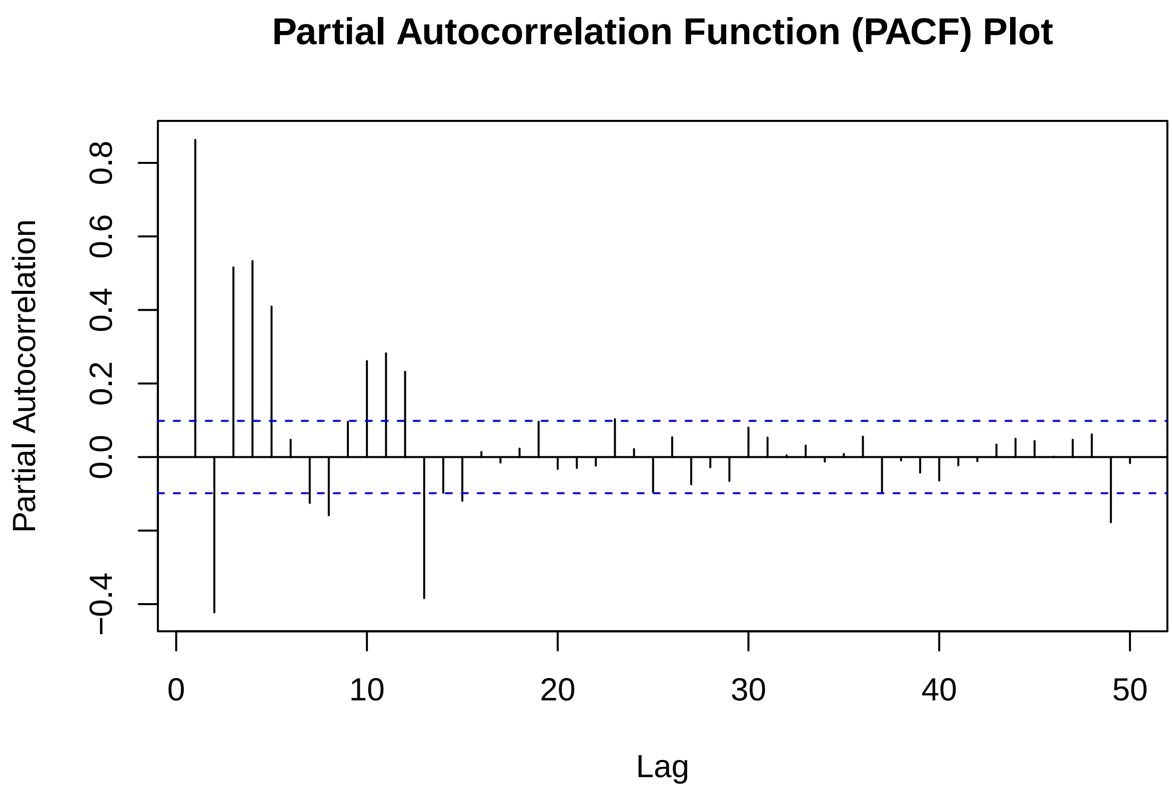
Both the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) were plotted together with EACF. These plots help in identifying the order of the AutoRegressive (AR) or Moving Average (MA) components in ARIMA modeling.The ACF plot displays significant autocorrelation at multiple lags which does not taper off quickly but shows a regular pattern, suggesting seasonality in the data. This regularity and slow decay indicate the need to account for seasonal differences and possibly additional differencing to achieve stationarity if this has not already been addressed.

The PACF plot shows significant partial autocorrelation at the first few lags and cuts off sharply after, which is characteristic of an AR(p) process where 'p' might be around 1 or 2. This suggests that the underlying process could likely be represented with a few AR terms. In EACFThe shift from 'x' to 'o' starts quite early in the rows, which indicates a lower number of AR terms might be sufficient. MA terms: Several 'o's appear right from the first column across different rows, suggesting few MA terms are needed,

```r
acf(df$Value, lag.max = 50,
    main = "Autocorrelation Function (ACF) Plot",
    xlab = "Lag", ylab = "ACF")
```



```r
pacf(df$Value, lag.max = 50,
    main = "Partial Autocorrelation Function (PACF) Plot",
    xlab = "Lag", ylab = "Partial Autocorrelation")
```




```r
eacf(df$Value)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x x x x x x x x x x x  x  x  x
## 2 x x x x x x x x x x x  x  x  x
## 3 x x o x o x o o o o x  x  x  x
## 4 x x x x o x o o o o o  x  x  x
## 5 x x x x o o o o o o o  x  o  o
## 6 x x o x o x o o o o o  x  x  o
## 7 x x o x x o o o o o o  x  x  o
```

Model fitting

SARIMA MODEL FITTING

- Because the series is seasonal, SARIMA (Seasonal ARIMA) will be used instead of ARIMA. From ACF and PACF plot below models are chosen to fit to the data:

- Fit 1: SARIMA(5,0,0)(1,0,0)[12]
- Fit 2: SARIMA(4,0,0)(1,0,0)[12]
- Fit 3: SARIMA(3,0,0)(1,0,0)[12]

Since the data is daily, seasonality is 12.

```
fit <- auto.arima(ts_data)
fit
```

```
## Series: ts_data
## ARIMA(2,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1     sma1
##       0.5503  -0.0683  -0.9477  -0.7635
## s.e.  0.0544   0.0549   0.0193   0.0331
##
## sigma^2 = 5.838:  log likelihood = -888.05
## AIC=1786.11   AICc=1786.27   BIC=1805.86
```

```
sarima_model <- Arima(df$Value, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))

sarima_model
```

```
## Series: df$Value
## ARIMA(2,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1     sma1
##       0.5503  -0.0683  -0.9477  -0.7635
## s.e.  0.0544   0.0549   0.0193   0.0331
##
## sigma^2 = 5.838:  log likelihood = -888.05
## AIC=1786.11   AICc=1786.27   BIC=1805.86
```

```
Arima(df$Value, order = c(5, 0, 0), seasonal = list(order = c(1, 0, 0), period = 12))
```

```
## Series: df$Value
## ARIMA(5,0,0)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ar4     ar5    sar1     mean
##       0.6461  -0.1252  0.2403  -0.0944  0.0688  0.9414  86.7475
## s.e.  0.0541   0.0623  0.0604   0.0611  0.0568  0.0183   6.6373
##
## sigma^2 = 8.452:  log likelihood = -996.96
## AIC=2009.92   AICc=2010.3   BIC=2041.8
```

```
Arima(df$Value, order = c(4, 0, 0), seasonal = list(order = c(1, 0, 0), period = 12))
```

```
## Series: df$Value
## ARIMA(4,0,0)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ar4    sar1     mean
##       0.6369  -0.1072  0.2350  -0.0572  0.9495  86.4348
## s.e.  0.0533   0.0602  0.0602   0.0529  0.0149   6.6934
##
## sigma^2 = 8.428:  log likelihood = -997.71
## AIC=2009.42   AICc=2009.71   BIC=2037.31
```

```r
Arima(df$Value, order = c(3, 0, 0), seasonal = list(order = c(1, 0, 0), period = 12))
```

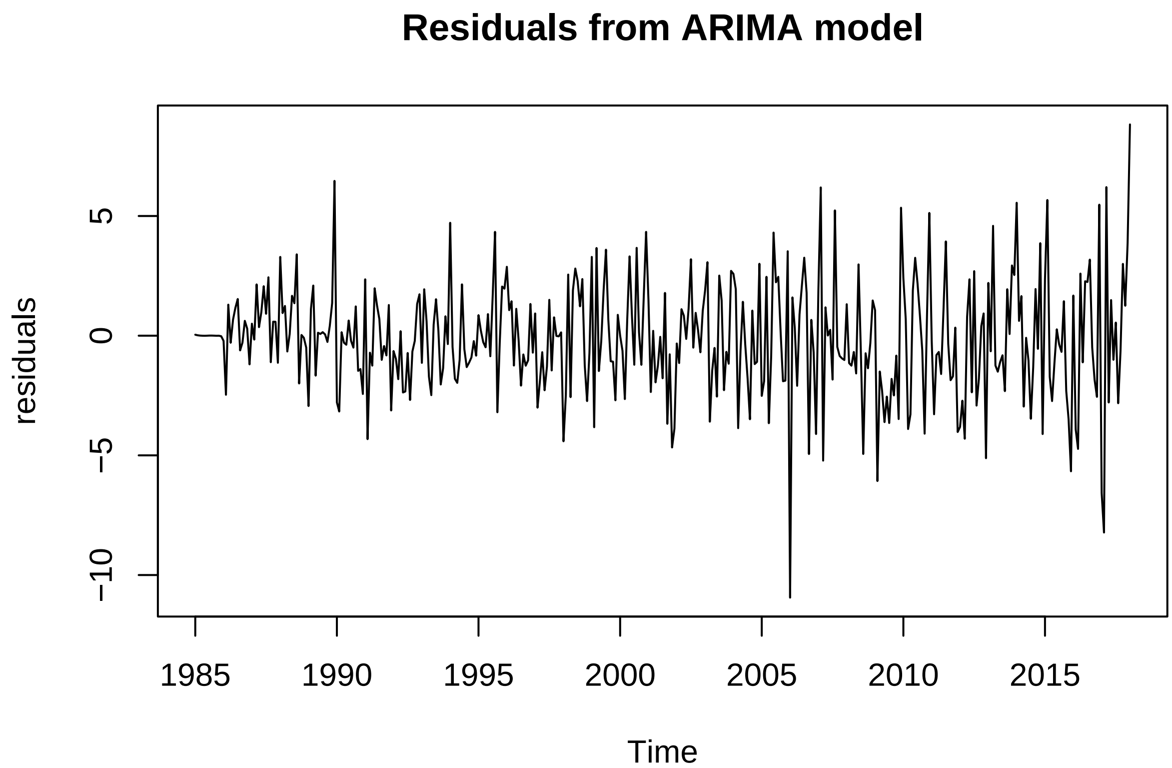
```
## Series: df$Value
## ARIMA(3,0,0)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3    sar1     mean
##       0.6293  -0.1021  0.1997  0.9454  86.7786
## s.e.  0.0532   0.0602  0.0506  0.0153   6.7813
##
## sigma^2 = 8.449:  log likelihood = -998.29
## AIC=2008.58   AICc=2008.8   BIC=2032.49
```

Several Seasonal ARIMA (SARIMA) models are considered given the clear seasonal pattern observed in the data. The model selection is based on the criteria of the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Models with different combinations of AR, MA, and seasonal components were evaluated.Comparing the fits based on AIC and BIC values: Auto arima has the least AIC and BIC values.Hence best model is the one suggested by the autoarima function with SARIMA Model (2,1,1)(0,1,1)[12].

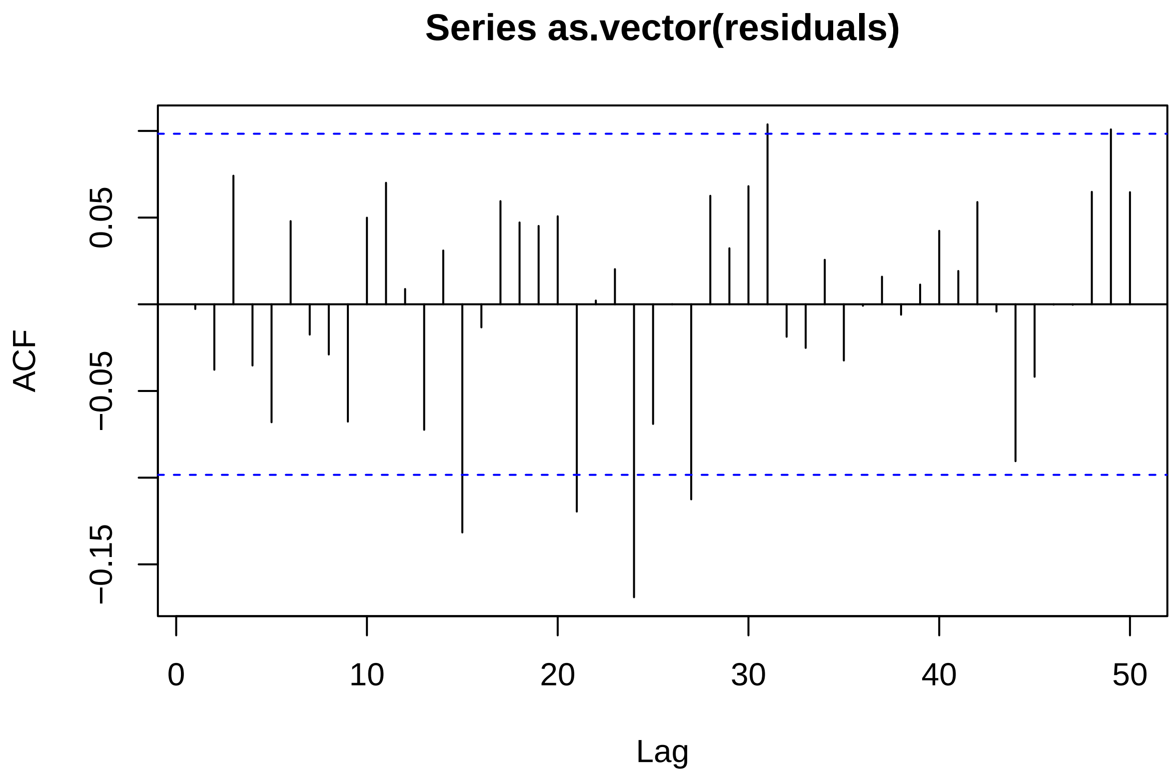
Residual Analysis

The residuals of the best-fitting model were analyzed to check the adequacy of the model fit. The residuals appeared to be white noise, as indicated by their ACF, and were approximately normally distributed based on the Shapiro-Wilk test and Q-Q plots.

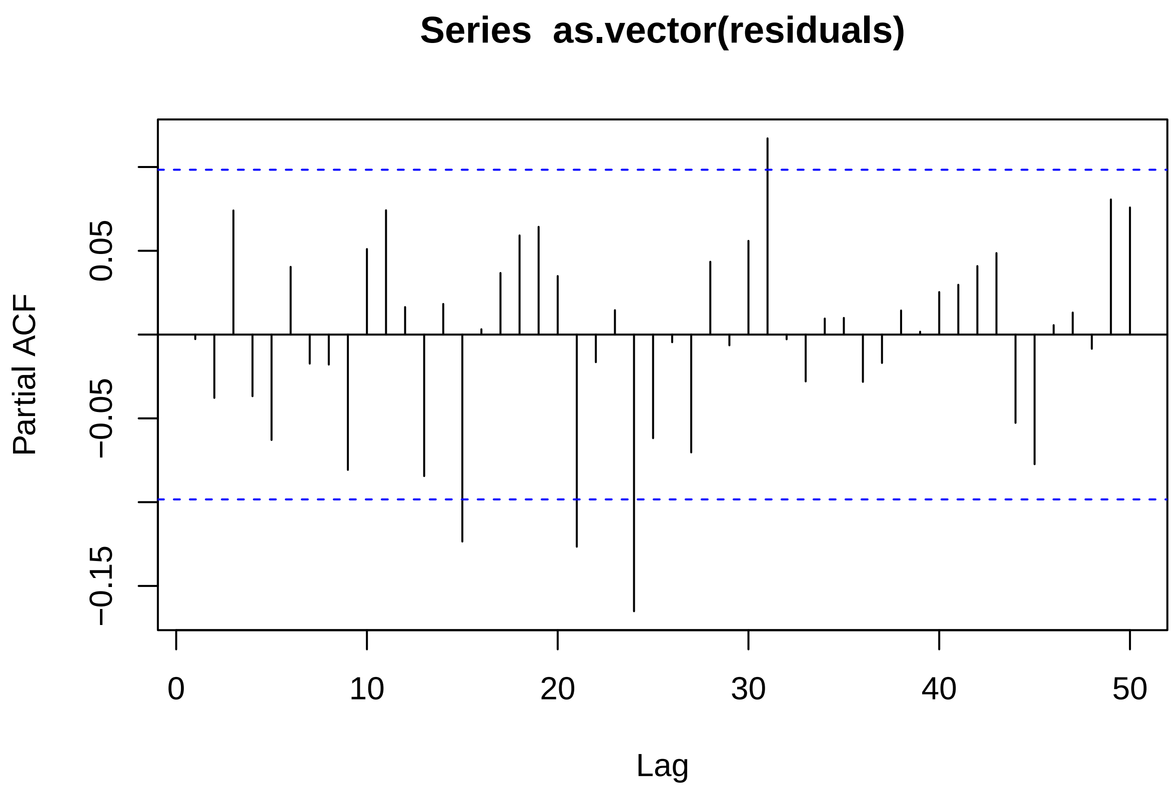
```r
residuals <- residuals(fit)
plot(residuals, main="Residuals from ARIMA model")
```



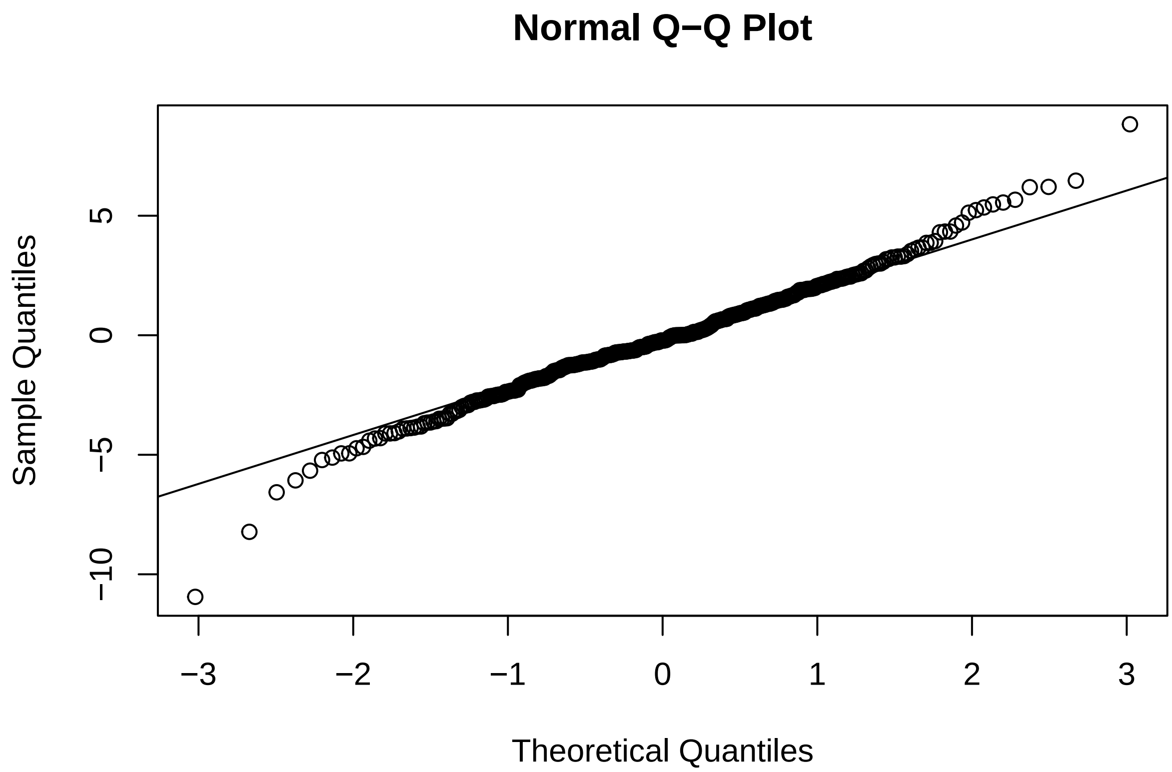

```
acf(as.vector(residuals), lag.max = 50)
```



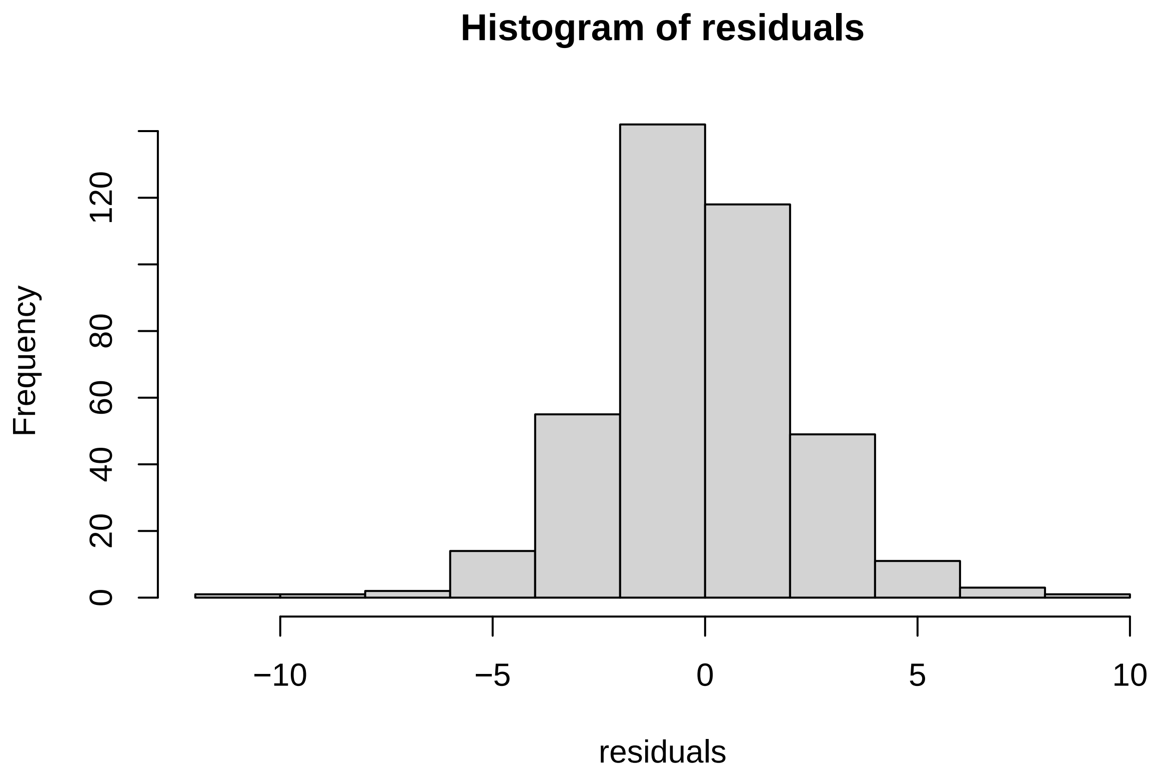

```
pacf(as.vector(residuals), lag.max = 50)
```




```
qqnorm(residuals)
qqline(residuals)
```




```
hist(residuals)
```




```
shapiro.test(residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98648, p-value = 0.0009324
```

```
Box.test(residuals,lag=10, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 9.4504, df = 10, p-value = 0.49
```

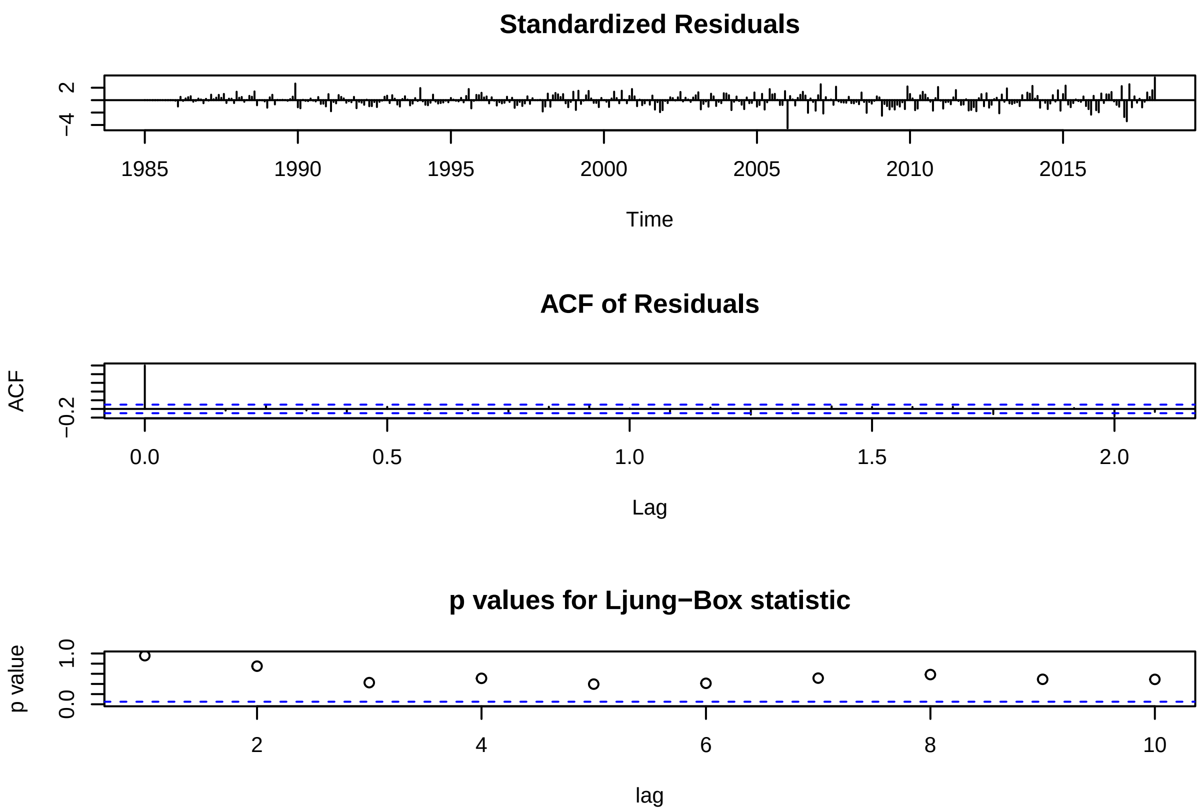
The ACF plot of the residuals shows that most autocorrelations are within the confidence bounds (the blue dotted lines), which is a good indication that the residuals are white noise.

The plot shows most points lie close to the reference line, suggesting that the residuals are approximately normally distributed.

The histogram shows a relatively bell-shaped curve, but it is not perfectly symmetric, and there appears to be a slight skew to the right.

With a p-value of 0.49, which is above the alpha level of 0.05, we fail to reject the null hypothesis that the residuals are independently distributed, meaning there is no autocorrelation.This further supports the hypothesis that the residuals are random (i.e., no autocorrelation present), indicating a good fit of the model to the data.

```
tsdiag(fit)
```



The analysis of the residuals from the fitted time series model suggests that the model is adequate. The absence of patterns in the residuals and the confirmation of white noise behavior through the ACF plot and Ljung-Box test results indicate that the model captures the underlying process well, with no need for additional complexity in the model structure.

Prediction

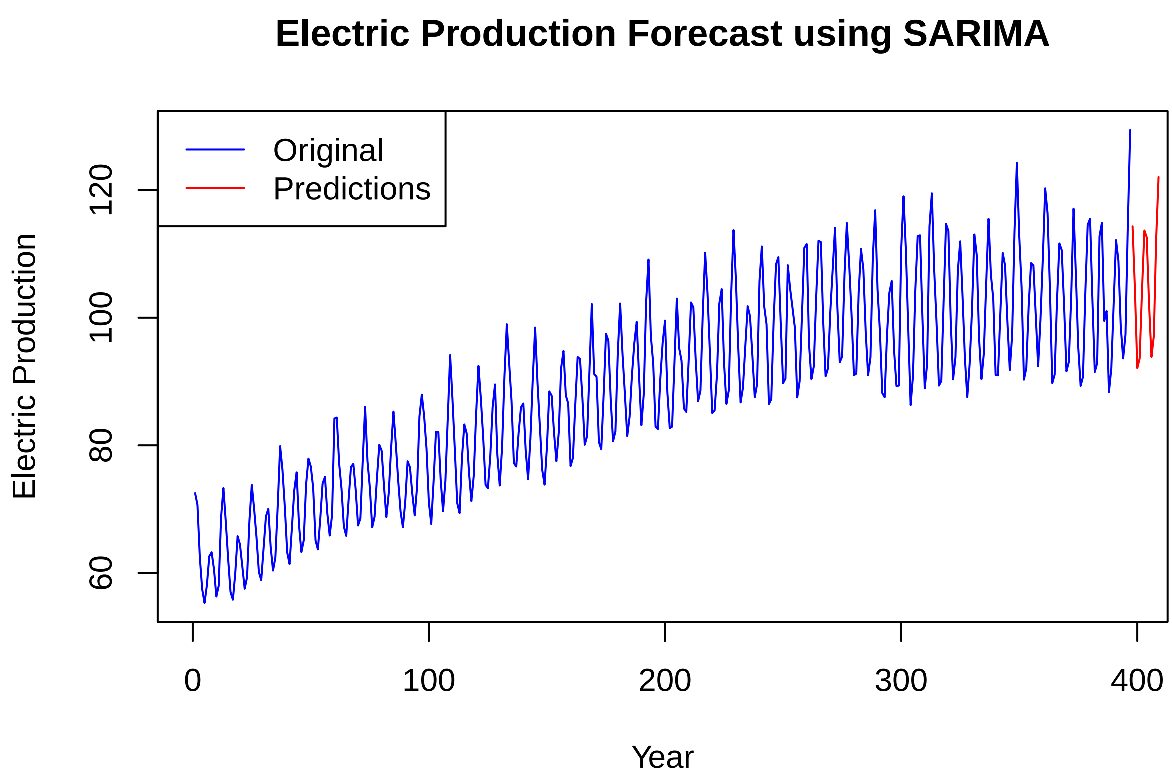
Forecasts were generated using the best-fitting SARIMA model. The point forecasts along with the confidence intervals were plotted, which provided insights into expected future values of electric production. The forecasts are crucial for planning and decision making in energy management.

```
predictions <- forecast(sarima_model, h = 12)
predictions
```

```
##     Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 398      114.31111 111.21470 117.40753 109.57555 119.04667
## 399      104.45857 100.84328 108.07386  98.92946 109.98768
```

```
## 400       92.09910  88.35393  95.84426  86.37136  97.82683
## 401       93.63140  89.84270  97.42011  87.83708  99.42572
## 402      104.31821 100.50733 108.12909  98.48997 110.14645
## 403      113.65996 109.83314 117.48678 107.80734 119.51258
## 404      112.58325 108.74256 116.42394 106.70943 118.45708
## 405      101.93541  98.08160 105.78922  96.04152 107.82930
## 406       93.85642  89.98978  97.72305  87.94291  99.76992
## 407       97.12217  93.24285 101.00150  91.18925 103.05509
## 408      112.41629 108.52434 116.30823 106.46407 118.36850
## 409      122.04284 118.13833 125.94735 116.07140 128.01428
```

```
plot(df$Value, type = "l", col = "blue", xlab = "Year", ylab = "Electric Production", main = "Electric I
lines(predictions$mean, col = "red")
legend("topleft", legend = c("Original", "Predictions"), col = c("blue", "red"), lty = c(1, 1))
```



Conclusions

The analysis successfully modeled the electric production data using SARIMA models, taking into account both non-stationarity and seasonality. The model provided satisfactory forecasts with reasonable confidence intervals, indicating robustness in the predictive capability. Future work could explore more complex models or external variables that could potentially improve the forecast accuracy.

PART B

NON SEASONAL DATASET: NYC Weather

INTRODUCTION

This project aims to perform a comprehensive time series analysis on weather data collected from New York City's Central Park in 2016. Time series analysis is a crucial statistical method used to analyze a sequence of data points collected over time intervals. Such analysis can reveal underlying patterns, trends, and seasonal variations, which are vital for forecasting and making informed decisions in meteorology, urban planning, and resource management.

DATA DESCRIPTION

Dataset : Dataset is taken from Kaggle.

https://www.kaggle.com/datasets/mathijs/weather-data-in-new-york-city-2016

The dataset comprises weather measurements from Central Park, NYC, for the year 2016. These observations have been collected daily, providing a granular view of the city's weather dynamics.

```
library(data.table)
library(forecast)
library(tseries)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(TSA)
```

ANALYSIS AND RESULTS

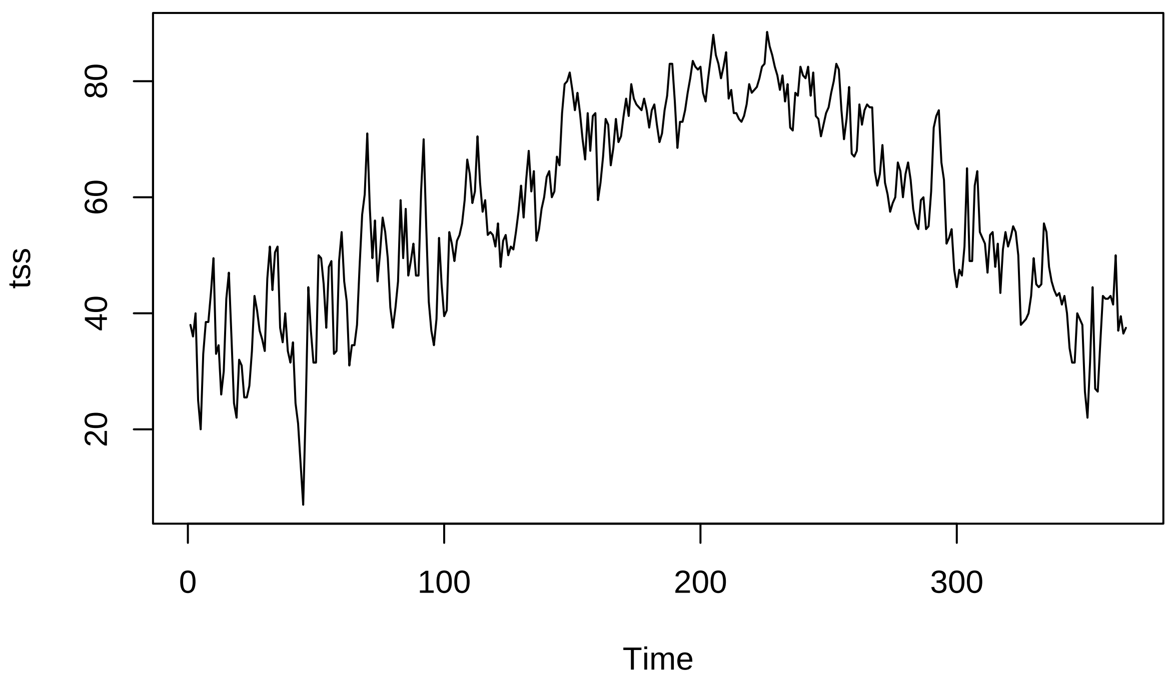
Preprocessing

Extract Date and Month column, convert Date to DateTime format.

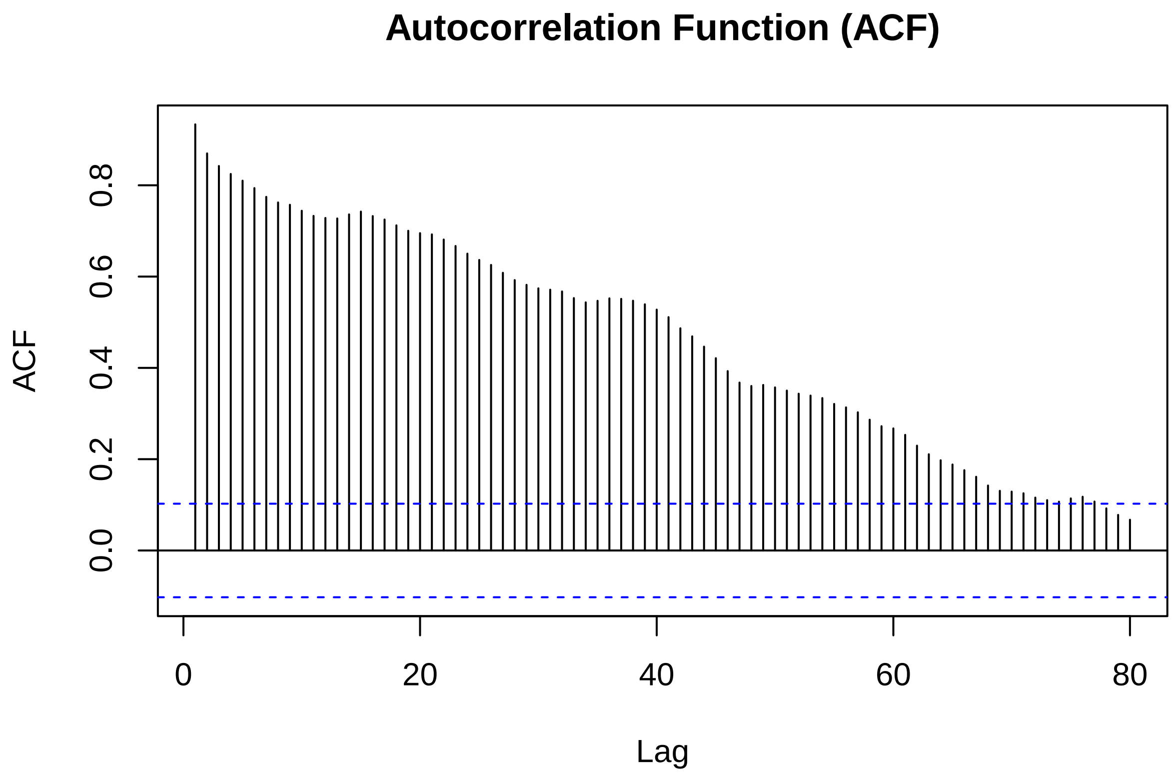
Average Temperature is converted to time series and plotted.

The plot displays the daily average temperature over time. It appears to have a time series object with a frequency of 12, which indicates yearly seasonality.

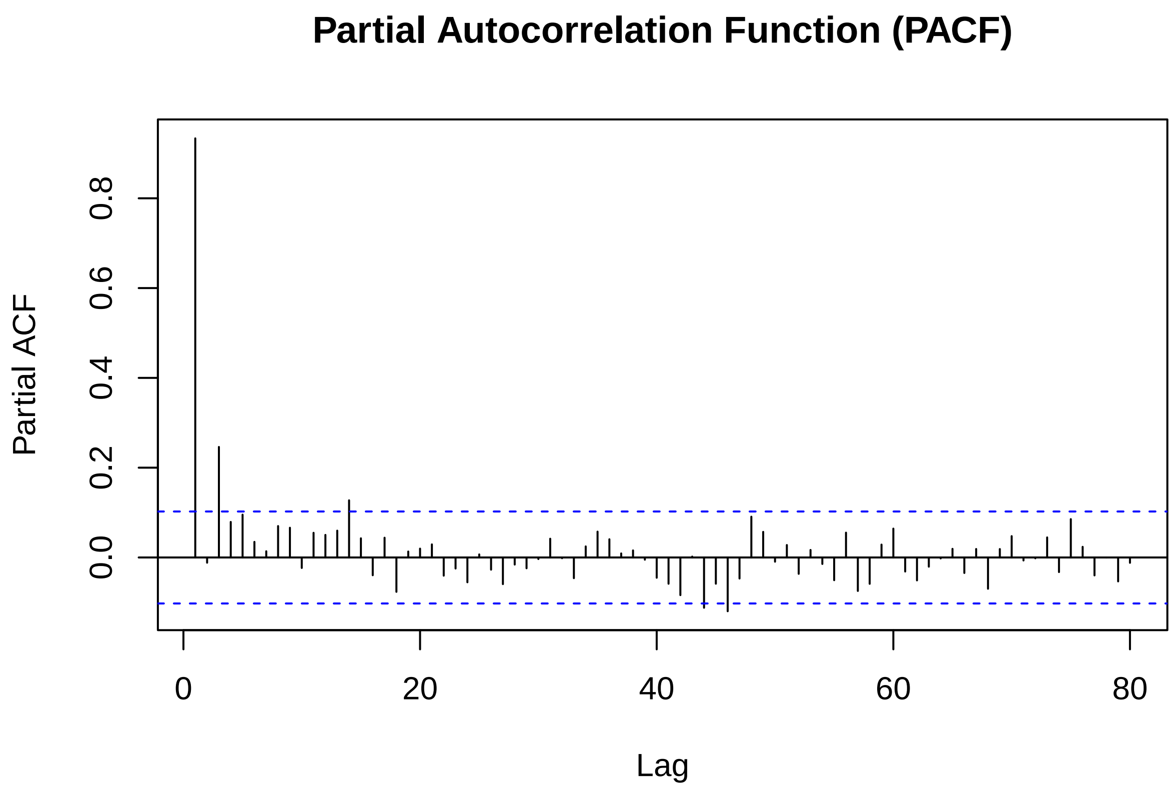
```
plot(tss)
```




```
acf(tss, main = "Autocorrelation Function (ACF)", lag.max = 80)
```




```
pacf(tss, main = "Partial Autocorrelation Function (PACF)",lag.max = 80)
```



The above ACF is decaying/decreasing, very slowly, and remains well above the significance range (dotted blue lines). The slow decay in the ACF indicates that the data may have a trend or some form of integrated behavior, requiring differencing to achieve stationarity. The PACF plot suggests that an autoregressive model might be appropriate for this data, with the order of the AR model potentially being indicated by the last significant lag.

Stationarity Test

```
g = as.numeric(tss)
adf.test(g)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  g
## Dickey-Fuller = -1.5185, Lag order = 7, p-value = 0.7802
## alternative hypothesis: stationary
```

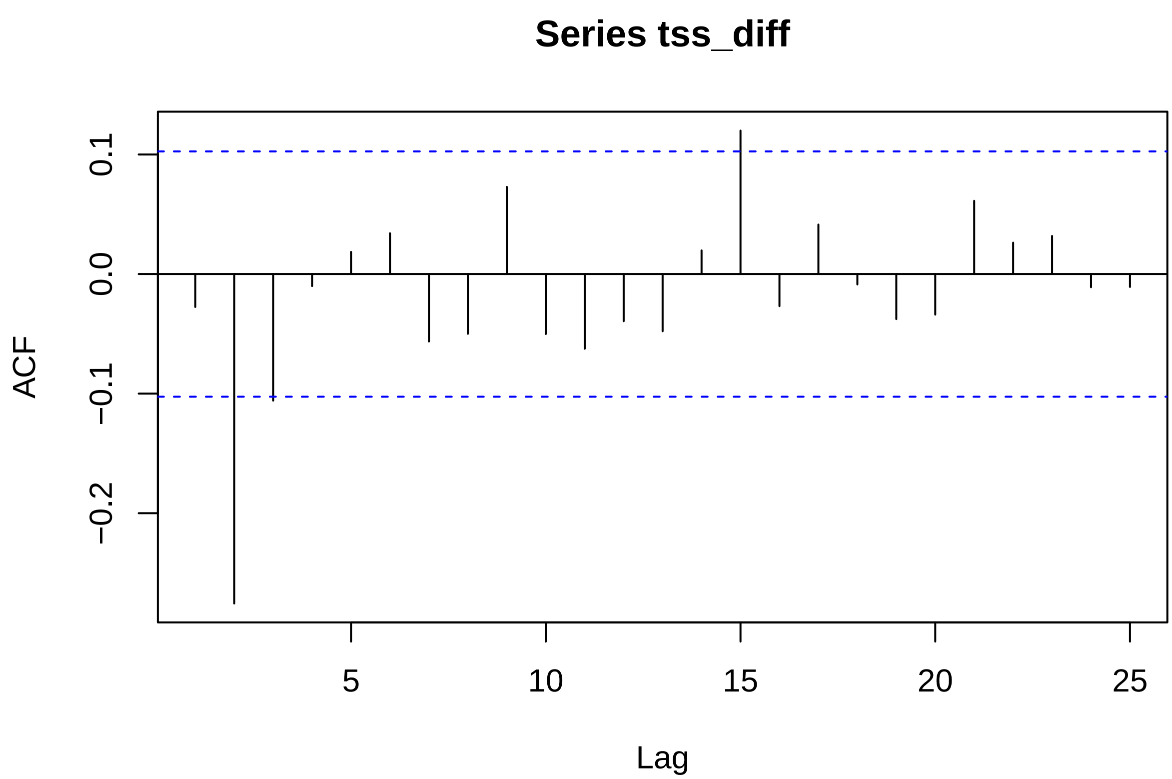
The initial ADF test result shows a p-value of 0.7802, which is significantly greater than 0.05 (common threshold for statistical significance). This high p-value indicates that we fail to reject the null hypothesis of the presence of a unit root, confirming that the original time series is non-stationary.

```r
tss_diff = diff(tss)
# Stationarity after differencing
h = as.numeric(tss_diff)
adf.test(h)
```

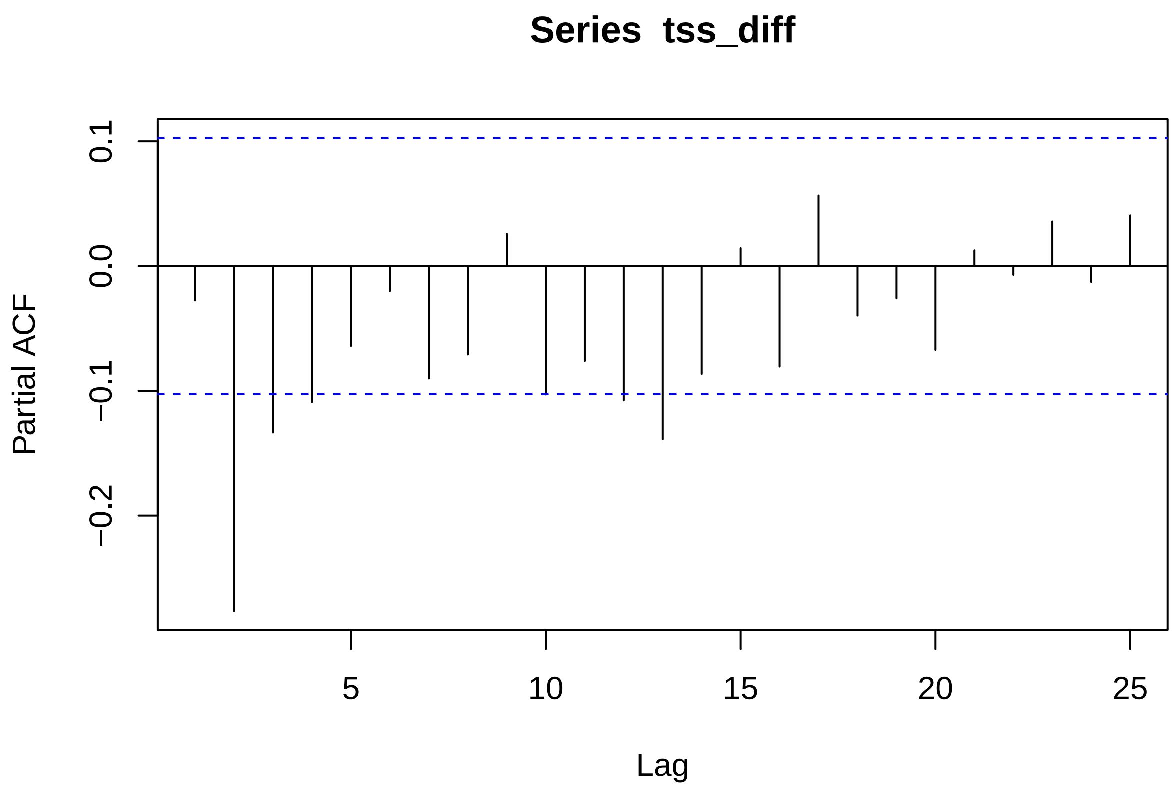
```
## Warning in adf.test(h): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  h
## Dickey-Fuller = -9.2043, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```r
acf(tss_diff)
```




```r
pacf(tss_diff)
```




```
eacf(tss_diff)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o x x o o o o o o o o  o  o  o
## 1 o x o o o o o o o o o  o  o  o
## 2 x x o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 x x x o x o o o o o o  o  o  o
## 5 x o x x o o o o o o o  o  o  o
## 6 x x x o o o o o o o o  o  o  o
## 7 x x x x o o o o o o o  o  o  o
```

Following the non-stationary result, the time series was differenced once. Differencing is a common technique used to transform a non-stationary time series into a stationary one by removing trends and cycles.

ACF plot displays significant autocorrelations up to lag 1 and then becomes insignificant, it suggests an MA(2) model, so q=2

PACF plot shows significant partial autocorrelations at the first two lags and then cuts off, it suggests an AR(3) model, so p could be 3.

The EACF table helps identify an appropriate ARMA model by showing where zeros ('o') dominate after a mix of significant ('x') and non-significant ('o') correlations. In this case, rows 3 and coulmn 2 quickly turn to 'o' across most MA terms, which suggests limited autoregressive or moving average components are needed.

Model fitting

As the series is non-seasonal, the ARIMA model will be used. From ACF and PACF & EACF plots we chose below models to fit the data:

- Model 1: ARIMA(3,1,2)
- Model 2: ARIMA(2,1,2)
- Model 3: ARIMA(1,1,1)
- Model 4: ARIMA(0,1,1)

```
arimafit <- auto.arima(tss)
arimafit
```

```
## Series: tss
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.4199  -0.5435  -0.2880
## s.e.  0.0957   0.0964   0.0669
##
## sigma^2 = 32.26:  log likelihood = -1150.7
## AIC=2309.4   AICc=2309.51   BIC=2325
```

```
Arima(tss, order = c(3, 1, 2))
```

```
## Series: tss
## ARIMA(3,1,2)
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2
##       0.8679  -0.3260  0.1226  -0.9952  0.0916
## s.e.  0.4978   0.3643  0.1010   0.5012  0.4379
##
## sigma^2 = 32.38:  log likelihood = -1150.3
## AIC=2312.61   AICc=2312.84   BIC=2336.01
```

```
Arima(tss, order = c(2, 1, 2))
```

```
## Series: tss
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1     ar2      ma1      ma2
##       0.3719  0.0402  -0.4973  -0.3330
## s.e.  0.2056  0.1548   0.1976   0.1821
##
## sigma^2 = 32.35:  log likelihood = -1150.66
## AIC=2311.33   AICc=2311.5   BIC=2330.83
```

```r
Arima(tss, order = c(1, 1, 2))
```

```
## Series: tss
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.4199  -0.5435  -0.2880
## s.e.  0.0957   0.0964   0.0669
##
## sigma^2 = 32.26:  log likelihood = -1150.7
## AIC=2309.4   AICc=2309.51   BIC=2325
```

```r
model <- Arima(tss, order = c(1, 1, 2))
model
```

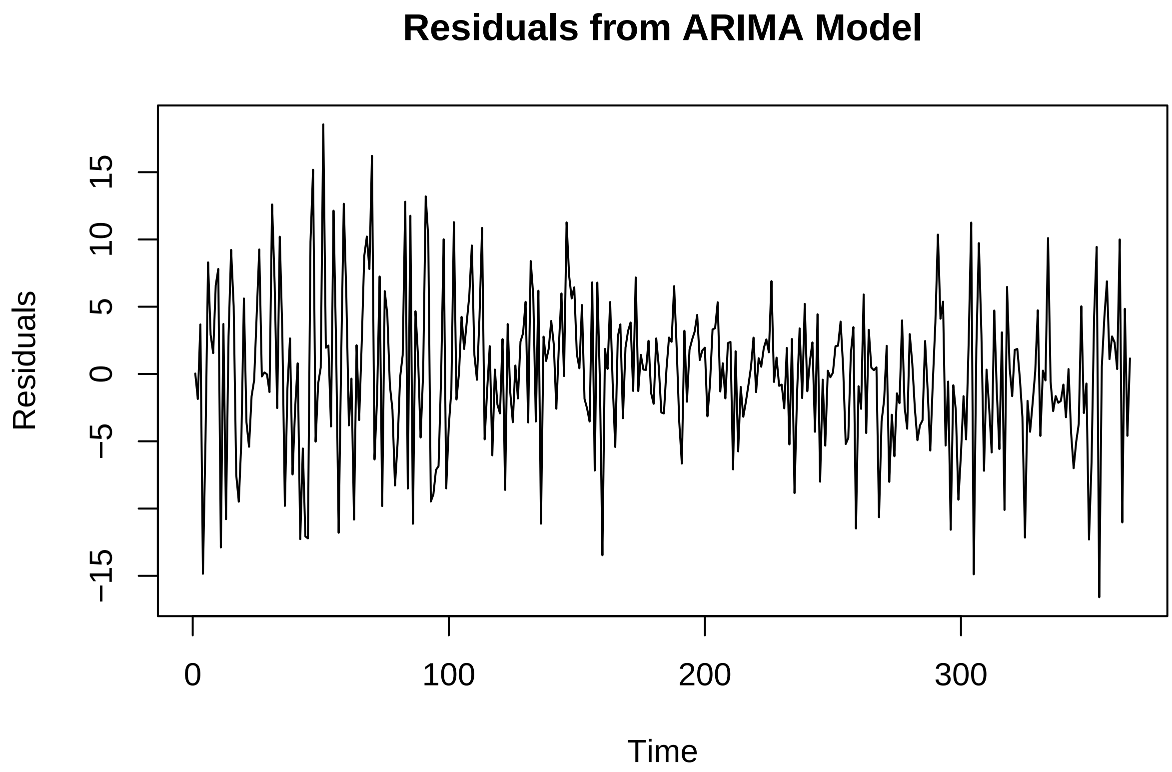
```
## Series: tss
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.4199  -0.5435  -0.2880
## s.e.  0.0957   0.0964   0.0669
##
## sigma^2 = 32.26:  log likelihood = -1150.7
## AIC=2309.4   AICc=2309.51   BIC=2325
```

Several ARIMA (ARIMA) models are considered given the clear seasonal pattern observed inthe data. The model selection is based on the criteria of the Akaike Information Criterion (AIC) andthe Bayesian Information Criterion (BIC). Models with different combinations of AR and MA were evaluated.Comparing the fits based on AIC and BIC values: Auto arima has the least AIC and BIC values. Hence best model is the one suggested by the autoarima function with SARIMA Model(1,1,2).

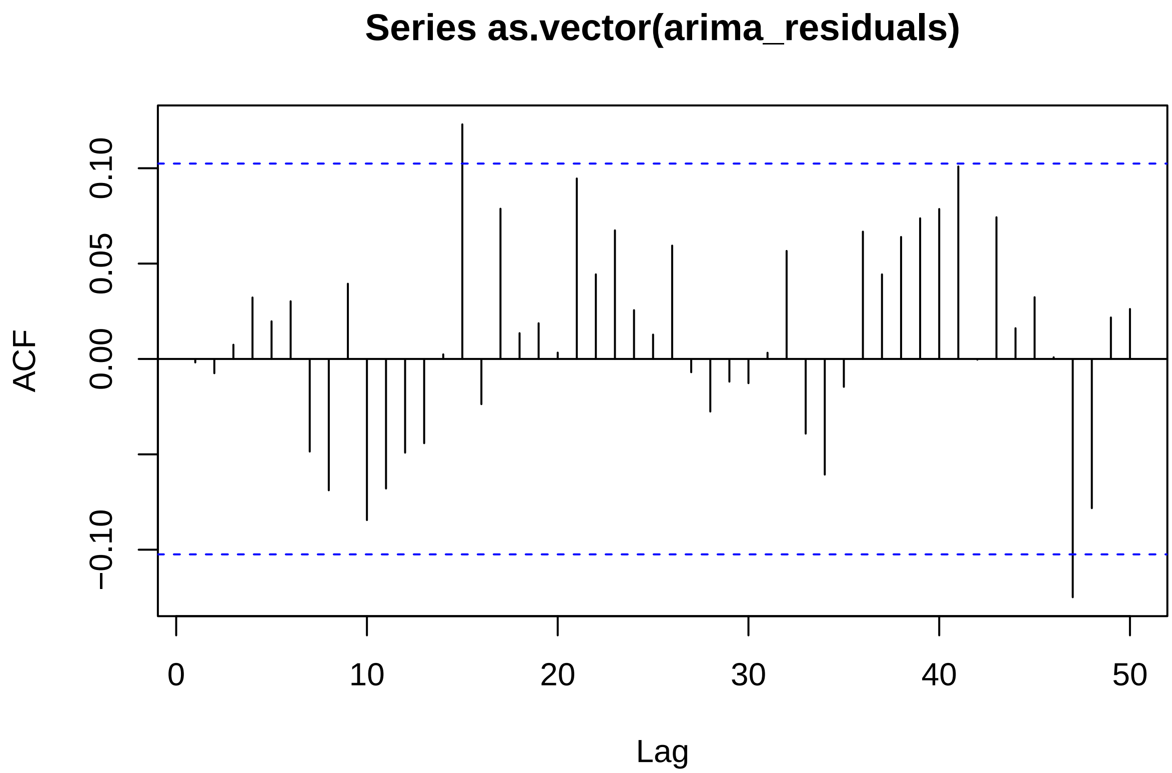
Residual Analysis

The residuals of the best-fitting model were analyzed to check the adequacy of the model fit. The residualsappeared to be white noise, as indicated by their ACF, and were approximately normally distributed basedon the Shapiro-Wilk test and Q-Q plots.

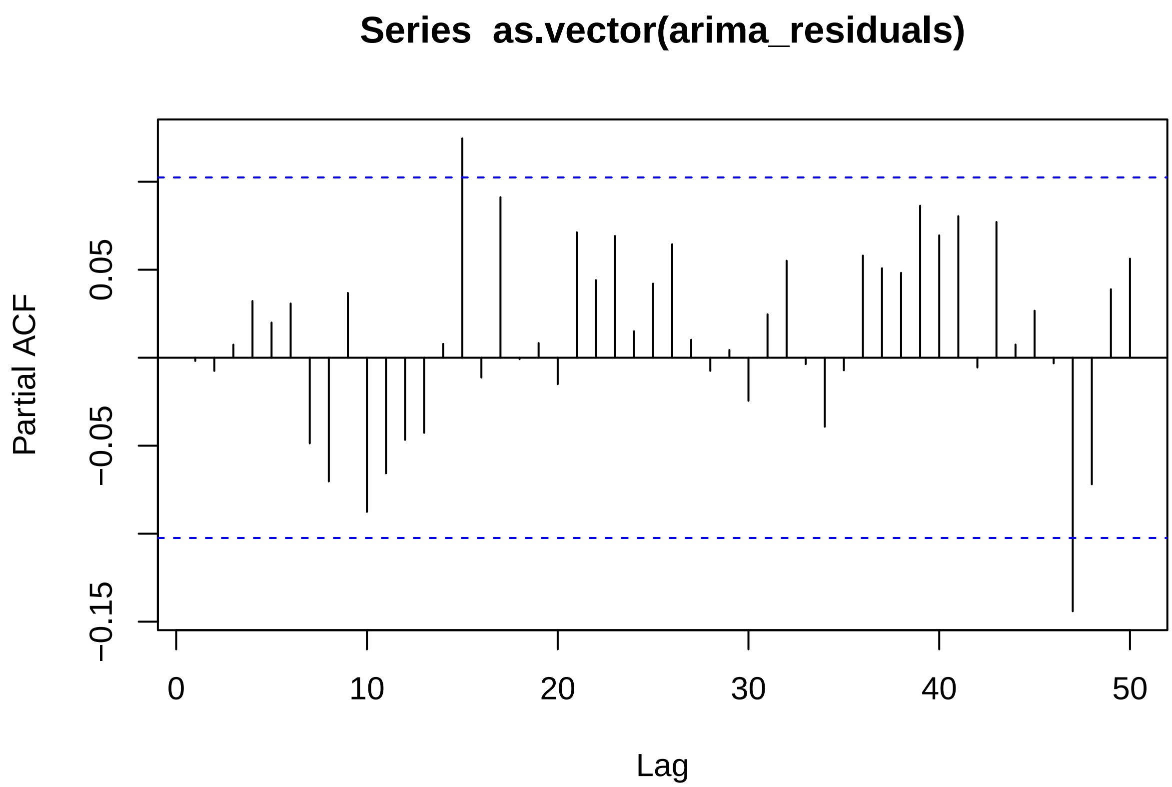
```r
arima_residuals <- residuals(model)
plot(arima_residuals, main = "Residuals from ARIMA Model", ylab = "Residuals")
```



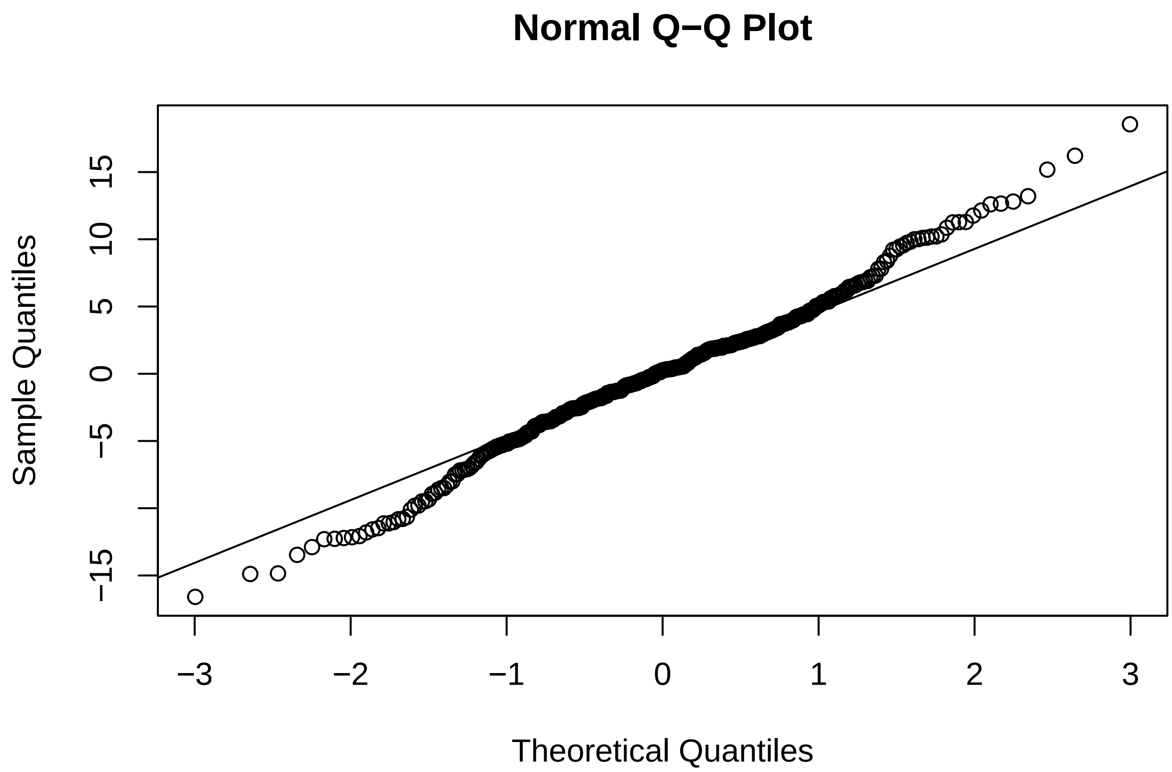

```
acf(as.vector(arima_residuals), lag.max = 50)
```



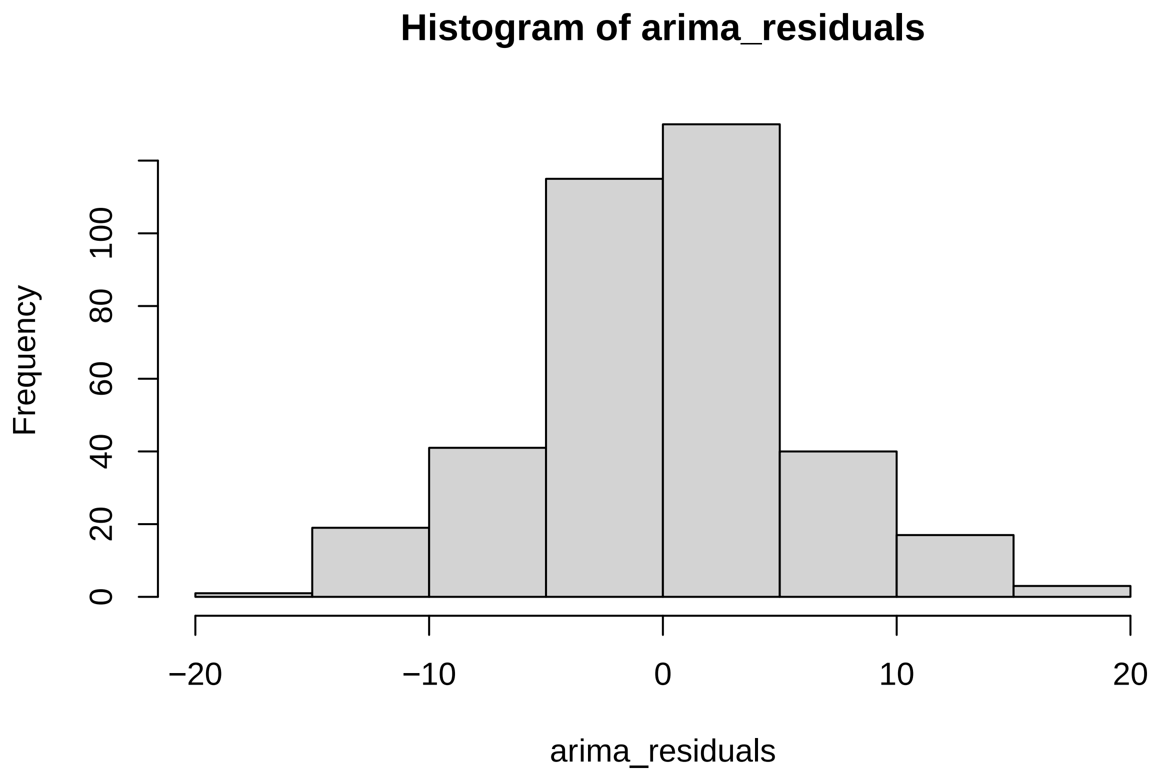

```
pacf(as.vector(arima_residuals), lag.max = 50)
```




```
qqnorm(arima_residuals)
qqline(arima_residuals)
```




```
hist(arima_residuals)
```




```r
print(shapiro.test(arima_residuals))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  arima_residuals
## W = 0.99101, p-value = 0.02506
```

```r
ljung_box_test <- Box.test(arima_residuals, lag = 10, type = "Ljung-Box")
ljung_box_test
```

```
##
##  Box-Ljung test
##
## data:  arima_residuals
## X-squared = 6.8724, df = 10, p-value = 0.7374
```

The ACF plot shows that most of the autocorrelations fall within the confidence intervals (blue dashed lines), with only a few lags marginally exceeding these limits. This is a generally good sign, indicating minimal autocorrelation within the residuals.

The PACF plot similarly demonstrates that residuals have almost no significant partial autocorrelations, with all values lying well within the confidence bounds.

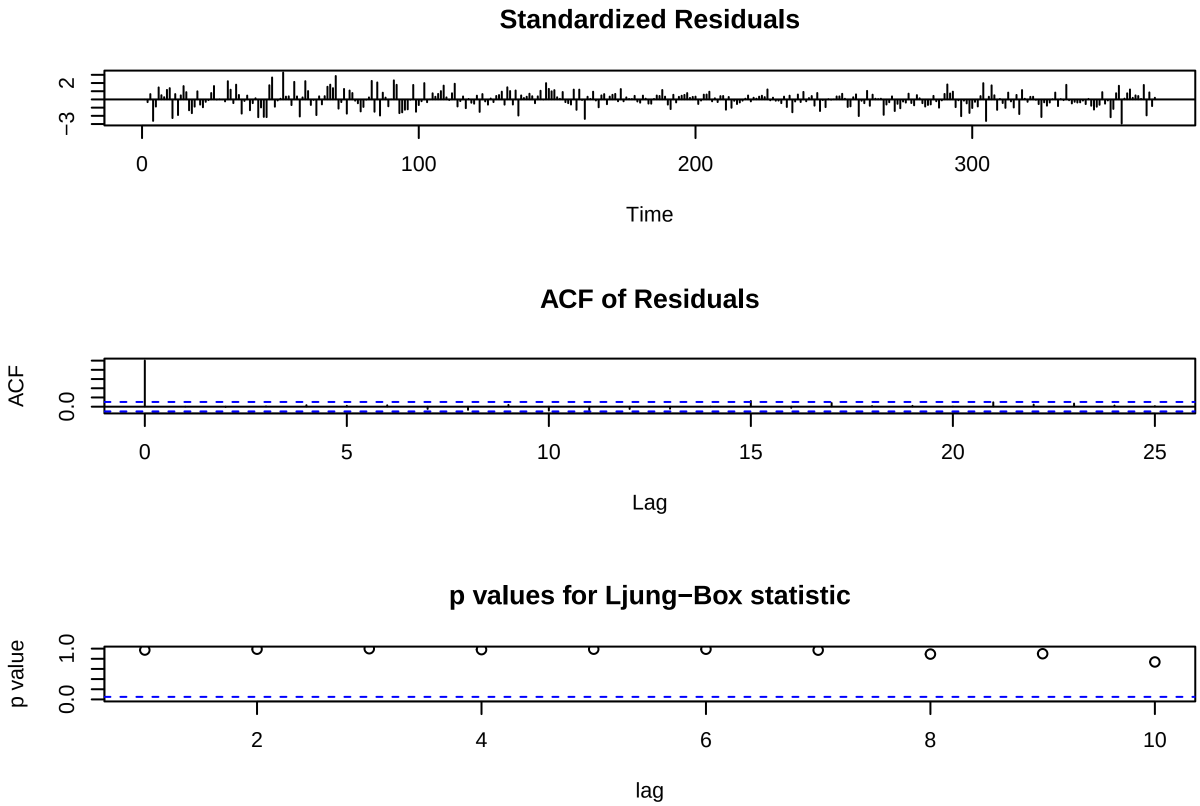
The Q-Q plot points largely align with the theoretical straight line, except for slight deviations in the tails. This indicates that the residuals are nearly normally distributed.

The histogram shows that the residuals are mostly symmetrically distributed about zero but not perfectly bell-shaped.

With a p-value of 0.7374, there is no evidence to reject the null hypothesis of no autocorrelation among the residuals.

ARIMA model seems to fit the data reasonably well, as indicated by the lack of autocorrelation in the residuals and their approximate normal distribution

```
tsdiag(model)
```



The analysis of the residuals from the fitted time series model suggests that the model is adequate. The absence of patterns in the residuals and the confirmation of white noise behavior through the ACF plot and Ljung-Box test results indicate that the model captures the underlying process well, with no need for additional complexity in the model structure

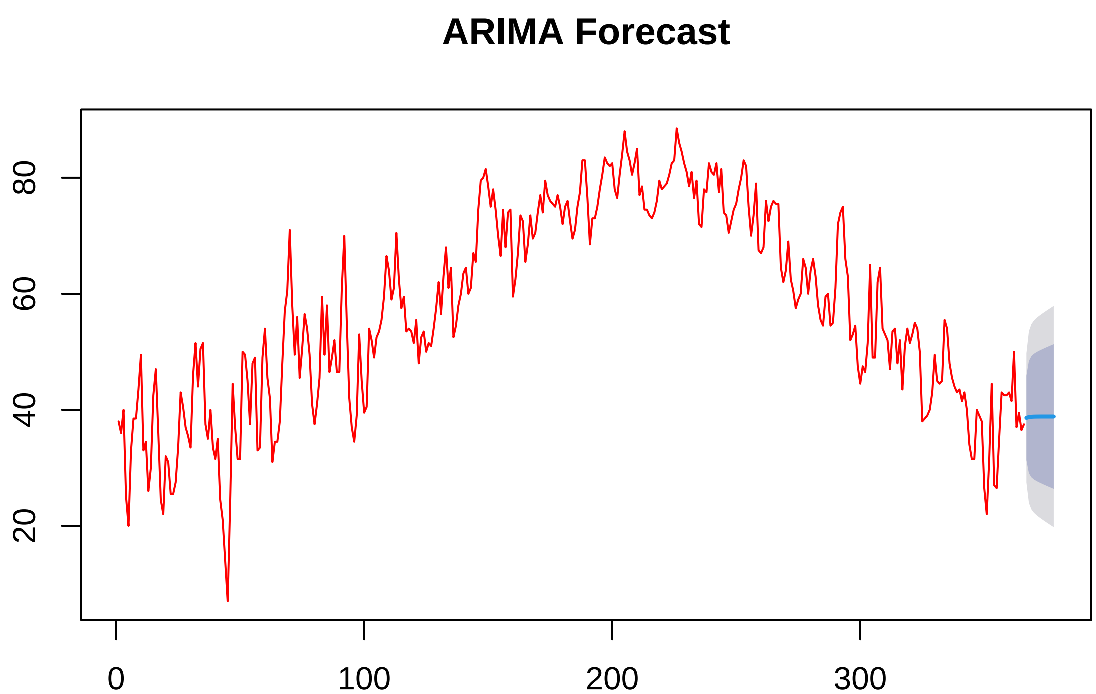
Prediction

Forecasts were generated using the best-fitting ARIMA model. The point forecasts along with the confidence intervals were plotted, which provided insights into expected future average temperatures. The forecast provides valuable information to the public, helping individuals plan activities and personal energy usage.

```
forecast_best_model <- forecast(model, h = 12)
forecast_best_model
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 367       38.61261 31.33317 45.89205 27.47967 49.74556
## 368       38.74589 29.06627 48.42550 23.94220 53.54958
## 369       38.80185 28.36396 49.23974 22.83848 54.76523
## 370       38.82535 28.00095 49.64975 22.27086 55.37985
## 371       38.83522 27.74132 49.92912 21.86857 55.80188
## 372       38.83937 27.52015 50.15858 21.52812 56.15061
## 373       38.84111 27.31571 50.36650 21.21453 56.46768
## 374       38.84184 27.11982 50.56386 20.91456 56.76912
## 375       38.84214 26.92917 50.75512 20.62283 57.06146
## 376       38.84227 26.74236 50.94218 20.33706 57.34749
## 377       38.84233 26.55873 51.12592 20.05619 57.62846
## 378       38.84235 26.37795 51.30675 19.77969 57.90501
```

```
plot(forecast_best_model, col="red", main = "ARIMA Forecast")
```



Conclusion

The ARIMA model's forecast of declining average temperatures in the upcoming periods suggests the need for adaptive strategies across various sectors in New York City. While the model effectively captures the historical temperature patterns and provides a detailed forecast, it is crucial to consider the inherent uncertainties in such predictions. These forecasts should be updated regularly with new data to refine predictions and adjust plans accordingly.

REFERENCES

1. Safari-Katesari, H., Samadi, S. Y., & Zaroudi, S. (2020). Modelling count data via copulas. Statistics, 54(6), 1329-1355.

2. Safari-Katesari, H., & Zaroudi, S. (2020). Count copula regression model using generalized beta distribution of the second kind. Statistics, 21, 1-12.

3. Safari-Katesari, H., & Zaroudi, S. (2021). Analysing the impact of dependency on conditional survival functions using copulas. Statistics in Transition New Series, 22(1).

4. Safari Katesari, H., (2021) Bayesian dynamic factor analysis and copula-based models for mixed data, PhD dissertation, Southern Illinois University Carbondale.

5. Zaroudi, S., Faridrohani, M. R., Behzadi, M. H., & Safari-Katesari, H. (2022). Copula-based Modeling for IBNR Claim Loss Reserving. arXiv preprint arXiv:2203.12750.

LINKS

1. https://www.kaggle.com/datasets/kandij/electric-production/data

2. https://www.kaggle.com/datasets/mathijs/weather-data-in-new-york-city-2016

3. https://www.kaggle.com/code/sujithmandala/how-to-time-series-forecasting