Time Series Analysis of Mean Maximum Temperature in Melbourne (1971 – 1990)

Julie Ku

June 9, 2023

Abstract

In this final project, we analyzed the "Mean Maximum Temperature in Melbourne (1971 – 1990)" time series dataset from the Times Series Data Library (tsdl) to explore the relationship between mean maximum temperature in Melbourne and time. We employed various statistical techniques to explore the data, including decomposition, partial/autocorrelation analysis, model identification, trend/seasonality analysis, and SARIMA modeling, to better understand the patterns in the temperature data during the specified period (1971-1990).

Key results revealed a seasonal pattern with a lag of 12 months and to remove seasonality from our data, we performed a Box-Cox transformation to initially stabilize the variance. We then differenced the Box-Cox transformed data at lag 12 to end up with stationary data. Based on these findings, we considered SARIMA models with seasonal differencing for modeling and forecasting the temperature data. From the ACF/PACF plots and AICc values, we identified the SARIMA(0,0,5)(0,1,1) [12] model and SARIMA(5,0,5)(1,1,1)[12] as the best performing predictive model for the data. Further fitting and validation procedures, such as diagnostic checking, helped with identifying the most accurate and robust model: SARIMA(0,0,5)(0,1,1) [12]. Overall, our findings highlight the importance of considering seasonal variations in modeling the "Mean Maximum Temperature in Melbourne" time series data.

Introduction

The objective of this project is to analyze and model the “Mean Maximum Temperature in Melbourne” time series dataset. This dataset contains monthly measurements of the mean maximum temperature measured in degrees Celsius in Melbourne, Australia–consisting of a total of 240 observations from January 1971 to December 1990. We will use the first 228 observations for the training dataset and the remaining 12 observations for the testing dataset. The data is part of the Time Series Data Library (tsdl) created by Rob Hyndman, Professor of Statistics at Monash University, Australia, and obtained from the Australian Bureau of Meteorology to study the relationship between mean maximum temperature and time/seasonality.

Throughout this project, we aimed to develop a forecasting model for the mean maximum temperature in Melbourne using the available historical data. We addressed this problem by utilizing time series analysis techniques to understand the seasonality present in the dataset and build a robust model for forecasting future temperatures in Melbourne. The primary techniques used in this analysis are decomposition, SARIMA (Seasonal Autoregressive Integrated Moving Average) modeling, and diagnostic checking, which incorporates both the seasonal and non-seasonal components of the time series data.

We first perform data visualization to get a better sense of the data we are working with. The plots we obtain from the data visualization process suggest we difference the data at lag 12 to remove seasonality. We then perform model identification with the sample ACF and sample PACF of the stationary data, further allowing us to use SARIMA modeling to fit models. Lastly, we use diagnostic checking tools for our candidate models such as residuals analysis and Box-Pierce/Box-Ljung tests. While not all diagnostic tests were passed, our final model SARIMA(0,0,5)(0,1,1) [12] performed quite well with forecasting the mean maximum temperature in Melbourne on both the original and transformed dataset. The analysis in this paper is made possible by the data provided by the Australian Bureau of Meteorology. The analysis was completed using R software.

Section 1: Data Visualization

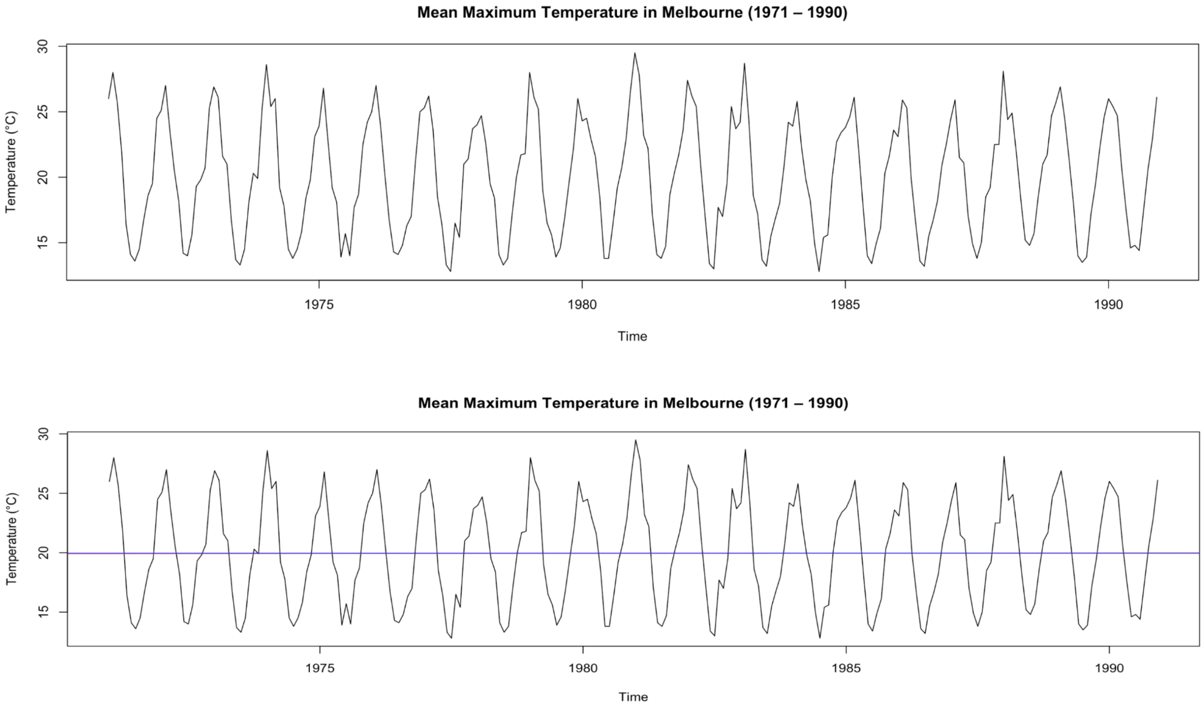


Figure 1: Monthly Mean Maximum Temperature in Melbourne from January 1971 - December 1990 (red line: linear regression fit of the data, blue line: mean of data)

In Figure 1, we can see from the red line, which represents the regression fit of the time series dataset, that there is no significant linear trend between monthly mean maximum temperature in Melbourne and time. There also appears to be a seasonal pattern and slight changes in variance with time. Thus, we can conclude from our data visualizations that the dataset has a seasonal pattern with varying variance and no trend.

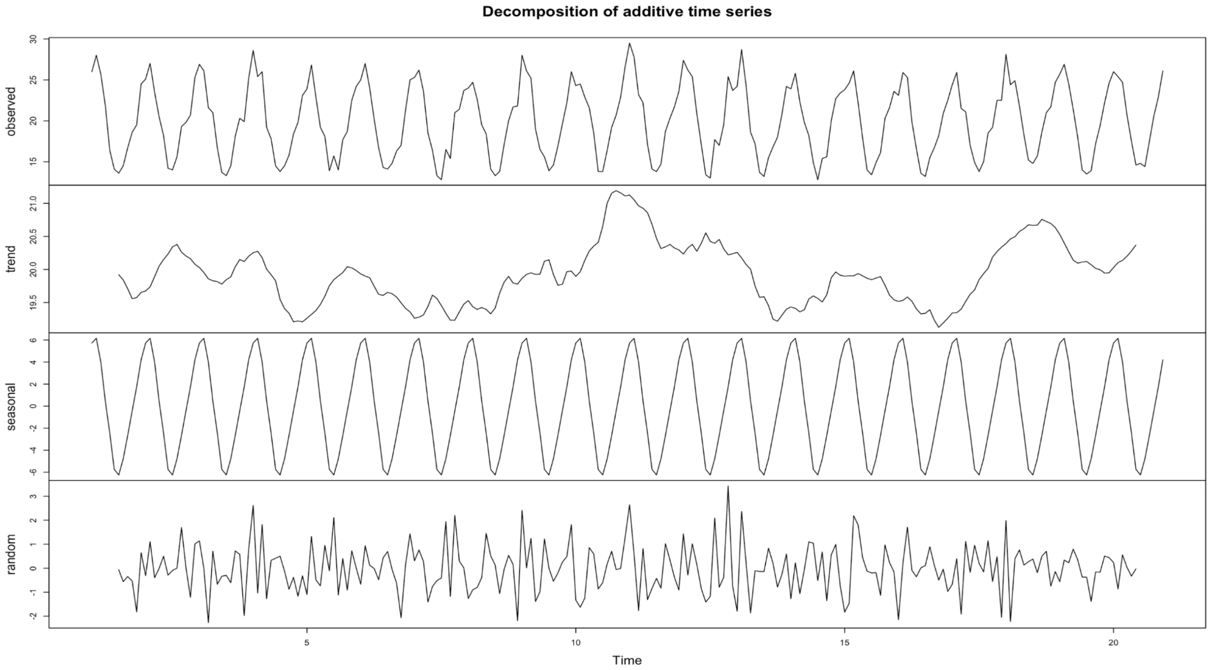


Figure 2: Decomposition of the Time Series in Additive Format (observed, trend , seasonal, & random components)

Furthermore, the decomposition plot of the time series in additive format in Figure 2 clearly shows that our data is seasonal and has no trend with slight changes in variance. Hence, we can conclude that our data will need to undergo either a Box-Cox or Log Transformation in order to stabilize the variance initially and we will need to difference our data to remove seasonality.

First, we split our data into a training set and a testing set with a ratio of 228:12. Since variance seems to change with time, we performed the two different transformations noted above. While the plots and histograms of the Box-Cox and Log transformed data seemed no different, we decided to proceed with the Box-Cox transformation as the value of lambda ($\lambda = 0.5050505$) was less than one, indicating that the transformation would help stabilize variance and reduce skewness in the data. Here is the mathematical formula for the Box-Cox transformation:

$$Y_t = \frac{1}{\lambda}(X_t^{\lambda} - 1);$$

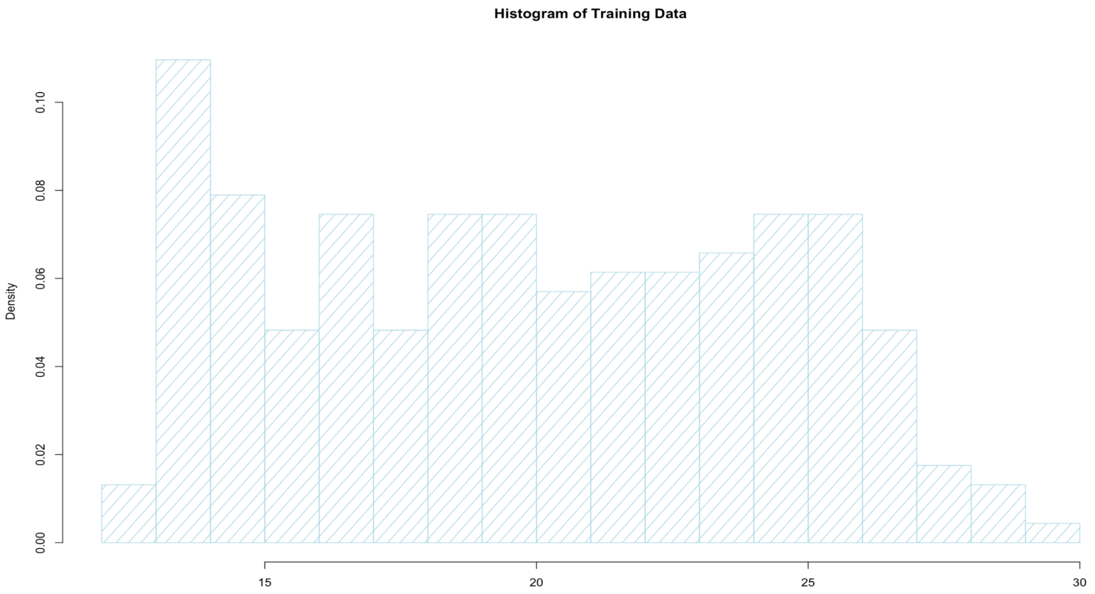


Figure 3: Histogram of Training Data

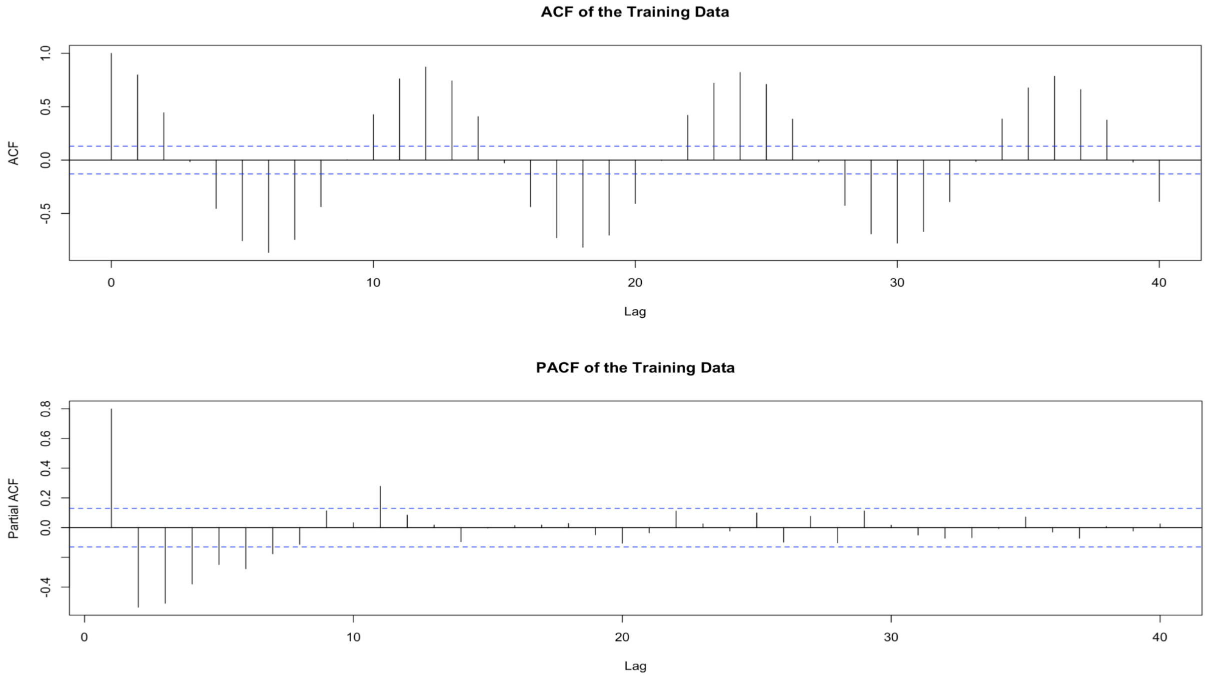


Figure 4: ACF/PACF of Training Data

Section 2: Data Transformation

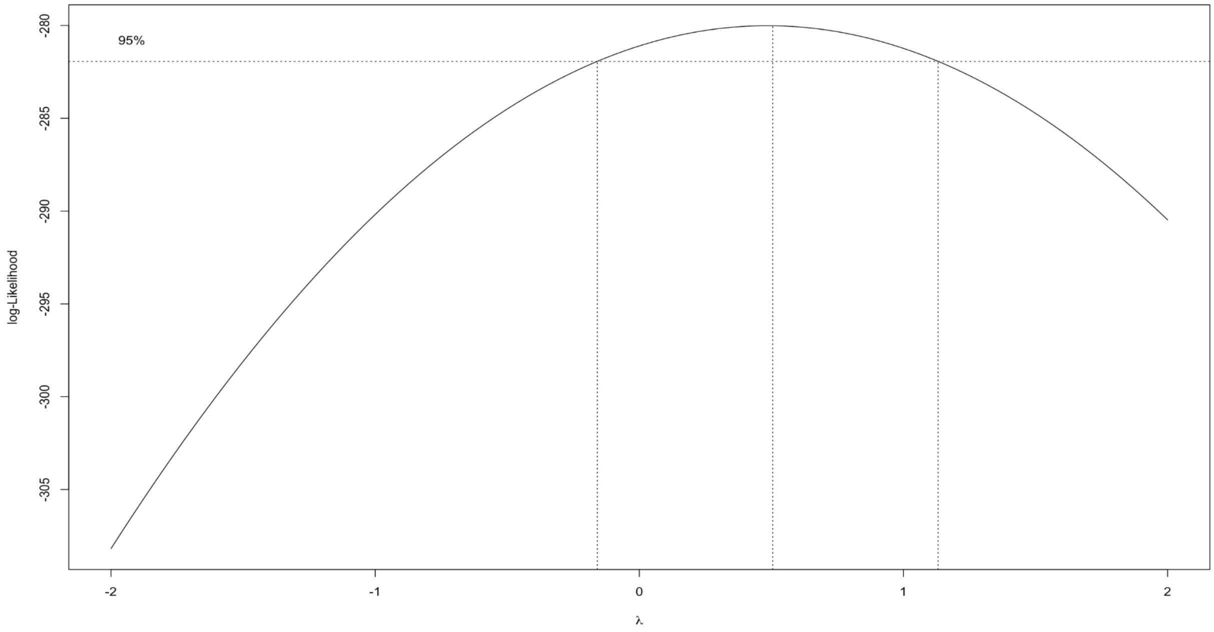


Figure 6: Box-Cox Transformation Plot (λ = 0.5050505)

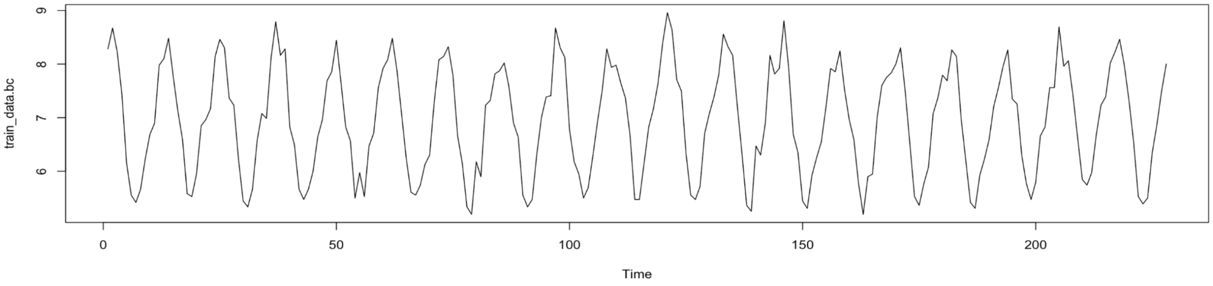


Figure 7: Plot of Box-Cox Transformed Data

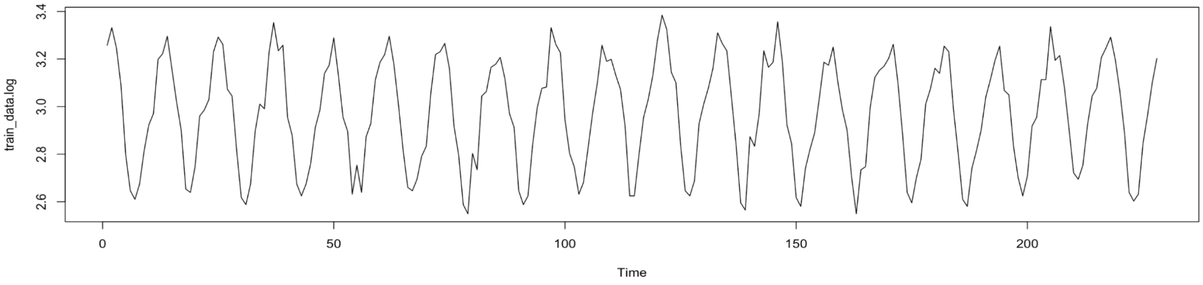


Figure 8: Plot of Log Transformed Data

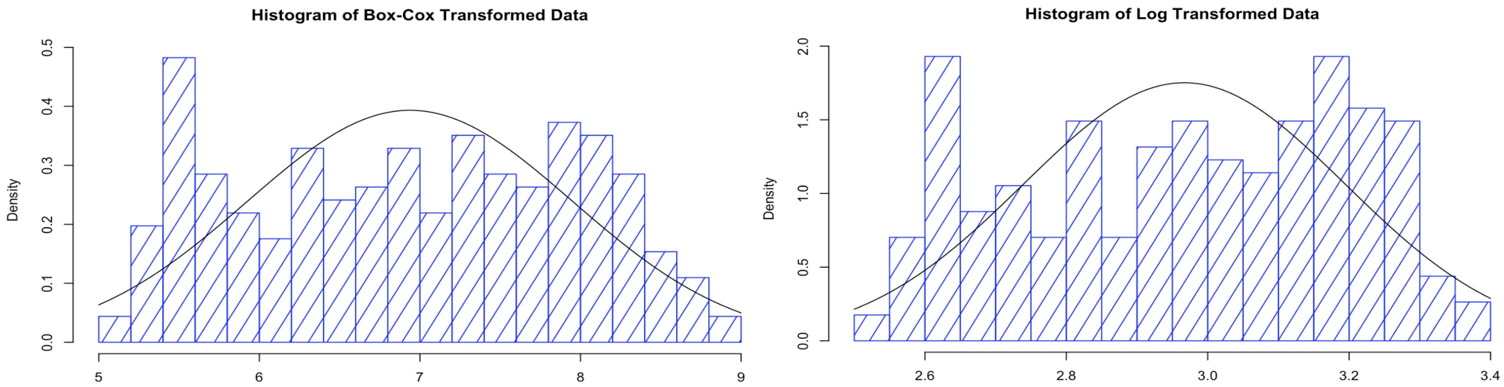


Figure 9: Left: Histogram of Box-Cox Transformed Data, Right: Histogram of Log Transformed Data

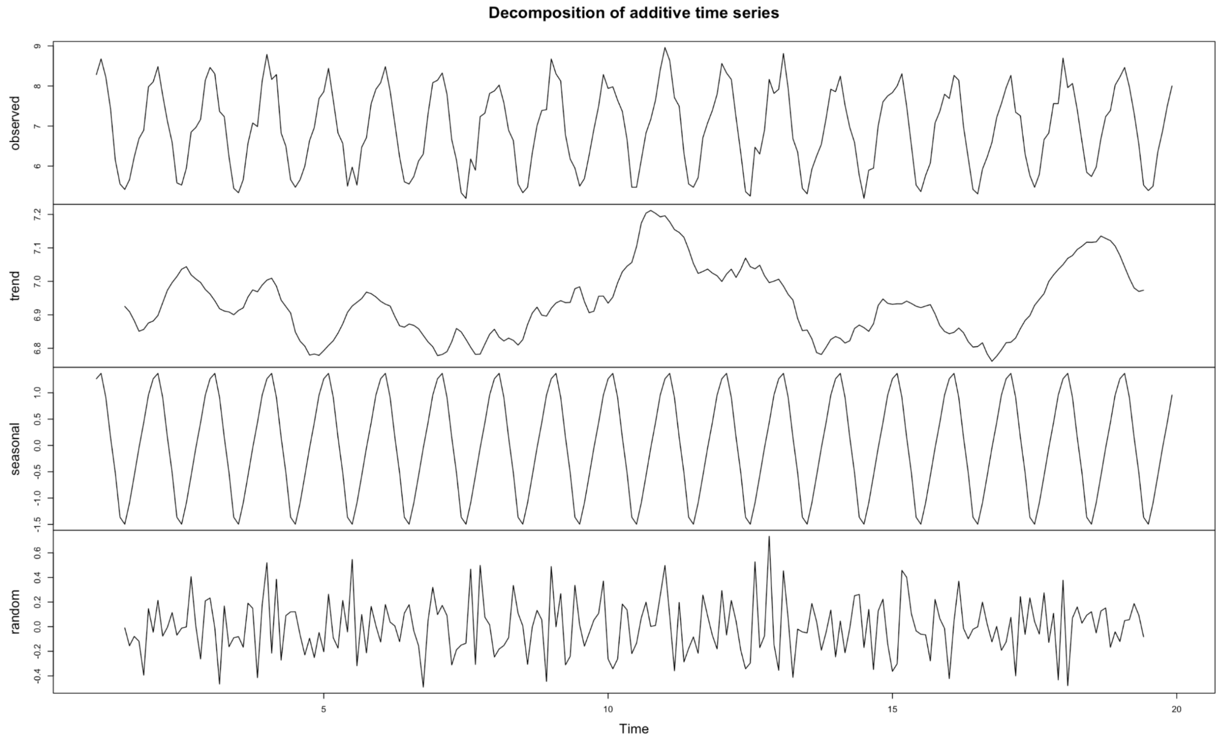


Figure 10: Decomposition of Box-Cox Transformed Data

After transforming the data with a Box-Cox transformation to stabilize variance, we differenced the data at lag 12 to remove seasonality. As shown in Figure 11, we can see that the seasonality is no longer apparent. When comparing the histograms of the original data and transformed data, we can see that the transformation and differencing resulted in a more uniform normal distribution with a bell-shaped curve.

Table 1 shows that the variance of the original training data significantly decreased after undergoing both a Box-Cox transformation and differencing at lag 12. Additionally, the red regression fit line overlaps the mean to show that there is no trend or non-constant variance in the data. Once again, we can conclude that the Box-Cox transformed data differenced at lag 12 is stationary.

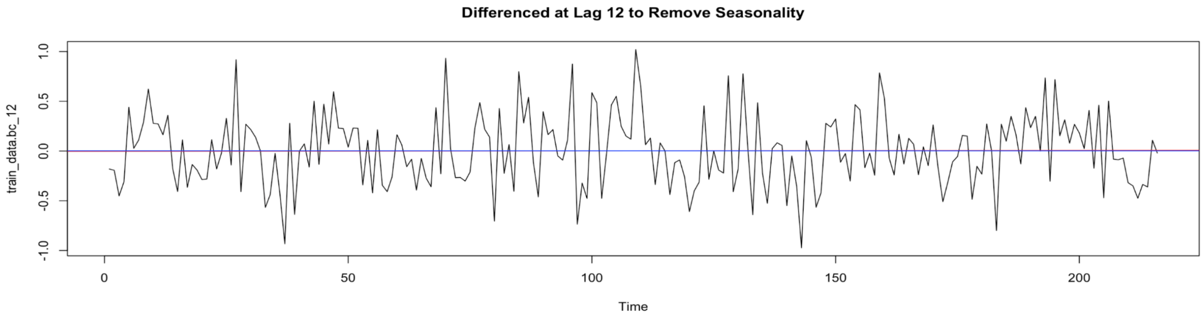


Figure 11: Box-Cox transformed data differenced at lag 12 to remove seasonality (red line: linear regression fit of the data, blue line: mean of data)

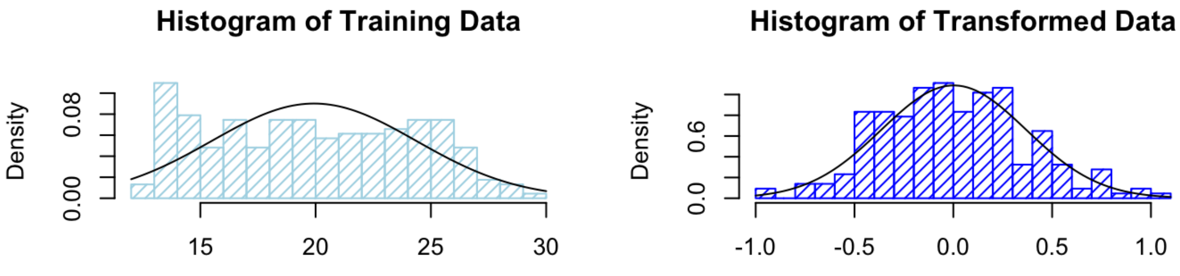


Figure 12: Left: Histogram of Training Data, Right: Histogram of Transformed Data (Box-Cox & Differencing)

Data	**Mean**	**Variance**
Original Training Data	19.95583	19.60256
Box-Cox Transformed Data	6.937977	1.028586
Box-Cox Transformed Data after differencing at lag 12	0.002223132	0.1344798

Table 1: Variances and means of original and transformed data

Section 2: Model Identification

In this section, we perform model identification based on the sample ACF/PACF plots. From the ACF plot in Figure 13, we can see lags 5 , 10, 12, and 22 are significant because the ACF at these lags are outside of the 95% confidence interval. Thus, we should consider q = 5, 10, and Q = 1, 2 for the moving average part of the model. Looking at the plot of the sample PACF, we have significant PACF at lags 5, 7, 8, and 10. This means we should consider p = 5, 7, 8, 10 with P = 1, 2 and maybe 3. It is evident that D =

1, s = 12 since we performed one seasonal differencing at lag 12 and d = 0 since we did not difference further due to nonexistent trends.

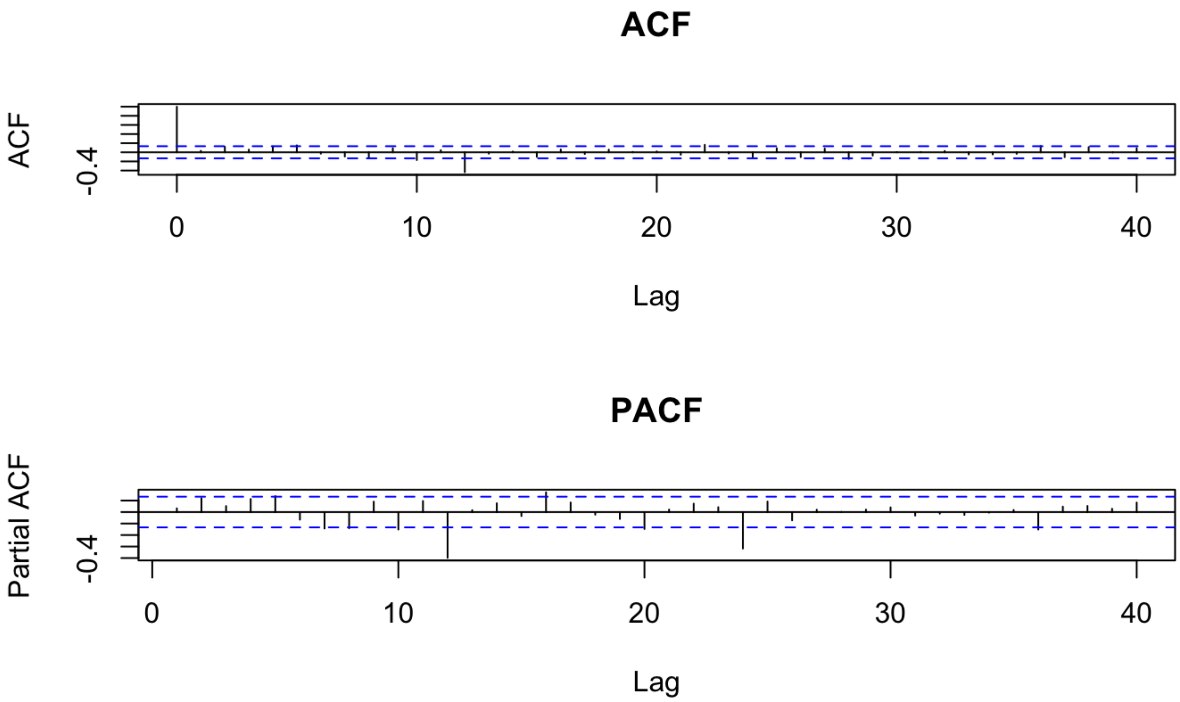


Figure 13: ACF/PACF of Data

Thus we will try these candidate models:
SAR: s=12, d=0, D=1, p=5, P=1
SMA: s=12, d=0, D=1, q=5, Q=1
SARIMA: s=12, d=0, D=1, p=q=5, P=Q=1

- **SARIMA**(5,0,5)(1,1,0)[12]
- **SARIMA**(0,0,5)(0,1,1)[12]
- **SARIMA**(5,0,5)(1,1,1)[12]

Section 2: Model Fitting

Model	**AICc**
SARIMA(5,0,5)(1,1,0)[12]	131.7257
SARIMA(0,0,5)(0,1,1)[12]	69.06345
SARIMA(5,0,5)(1,1,1)[12]	78.54304

Table 2: Fitted models and their AICC

After fitting different models from Table 2, we see that SARIMA(0,0,5)(0,1,1)[12] and SARIMA(5,0,5)(1,1,1)[12] models highlighted in blue and orange have the two lowest AICc values. With this in mind, we will move onto fixing the parameters for these two models: SARIMA(0,0,5)(0,1,1)[12] with ma1, ma2, ma3, ma4 fixed to 0 and SARIMA(5,0,5)(1,1,1)[12] with ma1, ma2, ma3, ma4, ma5, ar1, ar2, ar3, ar4, ar5, and sar1 fixed to 0. It is important to note that fixing the parameters resulted in a lower AIC value than before, indicating that fixing the AR components of the models might be a better approach to fitting the best model.

Model	**AIC**
1: SARIMA(0,0,5)(0,1,1)[12] with ma1-ma4 fixed to 0	67.9
2: SARIMA(5,0,5)(1,1,1)[12] with ma1-ma5, ar1-ar5, sar1 fixed to 0	73.69

Table 3: 2 best performing models with fixed parameters and their AIC

Section 3.1: Diagnostic Checking for Model 1

Now, let's move onto diagnostic checking for Model 1 with fixed parameters to see if the residuals follow a white noise distribution. We first check normality assumptions. Looking at Figure 14, the residuals seem to follow a normal distribution from the histogram and q-q plot. There is no seasonality or trend in the plot of the residuals. From Figure 15, the ACF/PACF of the residuals seems to be mostly all within the confidence interval. Thus, we can conclude that the residuals of Model 1 resemble white noise.

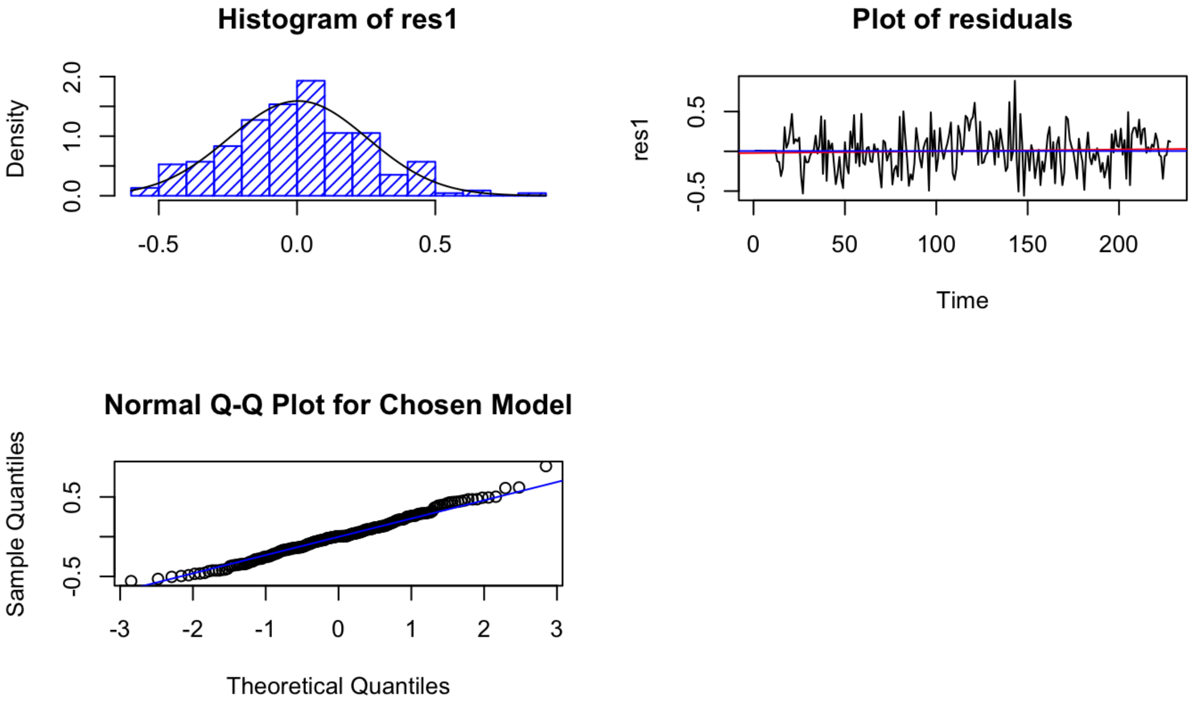


Figure 14: Diagnostic checking plots for normality of residuals of Model 1

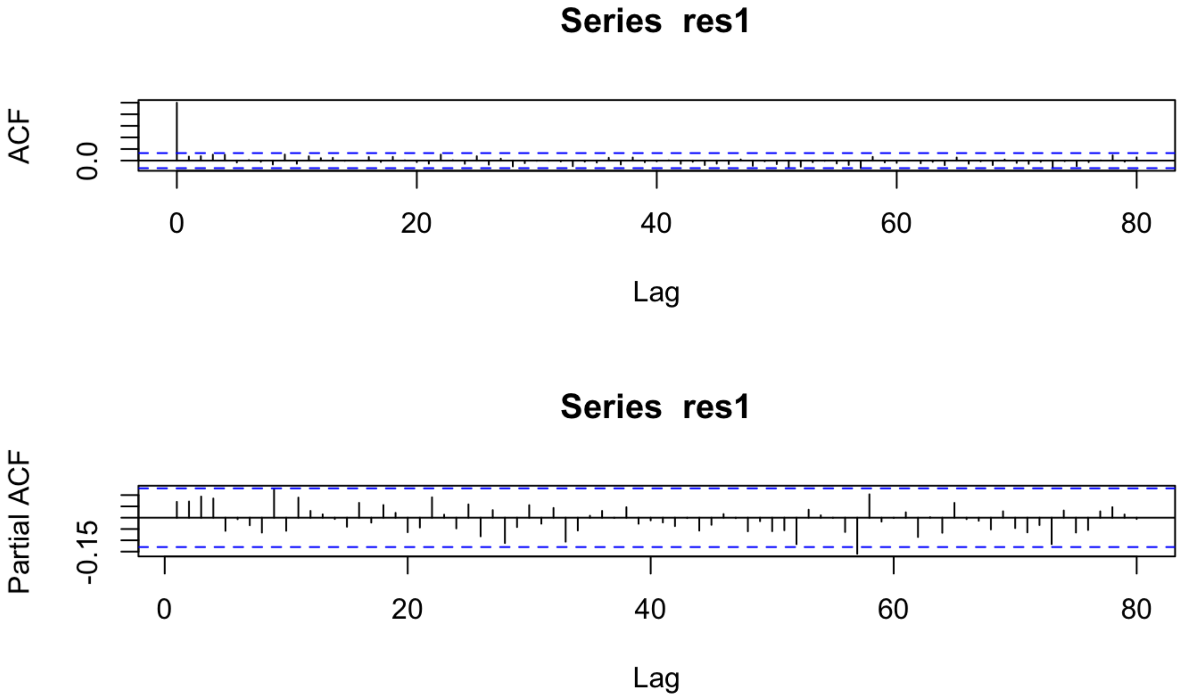


Figure 15: ACF/PACF of residuals of Model 1

As part of diagnostic checking, we also check for several independence assumptions in Table 4. Model 1 passes the majority of the four tests (3), indicating that this is a potential final model.

Test	**Statistics**	**P-value**	**Result**
Shapiro-Wilk	0.99145	0.2039	pass
Box-Pierce	16.009	0.592	pass
Ljung-Box	16.715	0.5428	pass
McLeod-Li	40.041	0.004936	fail

Table 4: Testing results for Model 1

Section 3.2: Diagnostic Checking for Model 2

Upon trying to determine a new candidate model, fitting any model with q = 10 and p = 7, 8, 10 does not produce significant results. A combination of these parameters either made the seasonal moving average part of the model not invertible or resulted in NaNs in the coefficients or 0 in its confidence interval. After countless model fitting, I determined that the models with p = q = 5 parameters performed best with fruitful results, such as the lowest AICc values amongst all models.

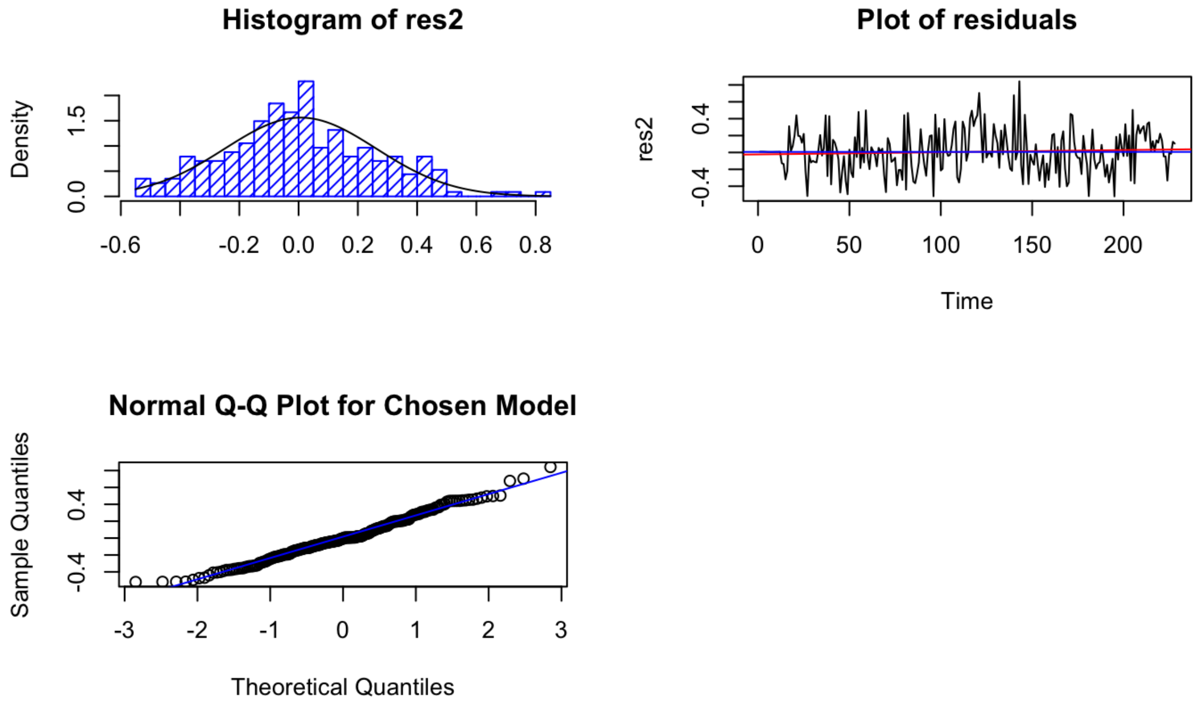


Figure 16: Diagnostic checking plots for normality of residuals of Model 2

We can immediately see in Figure 16, that Model 2 is outperformed by Model 1 in the tests for normality as the histogram of the residuals has a slight right-skewed distribution. The Q-Q plot and plot of residuals seem relatively similar to Model 1.

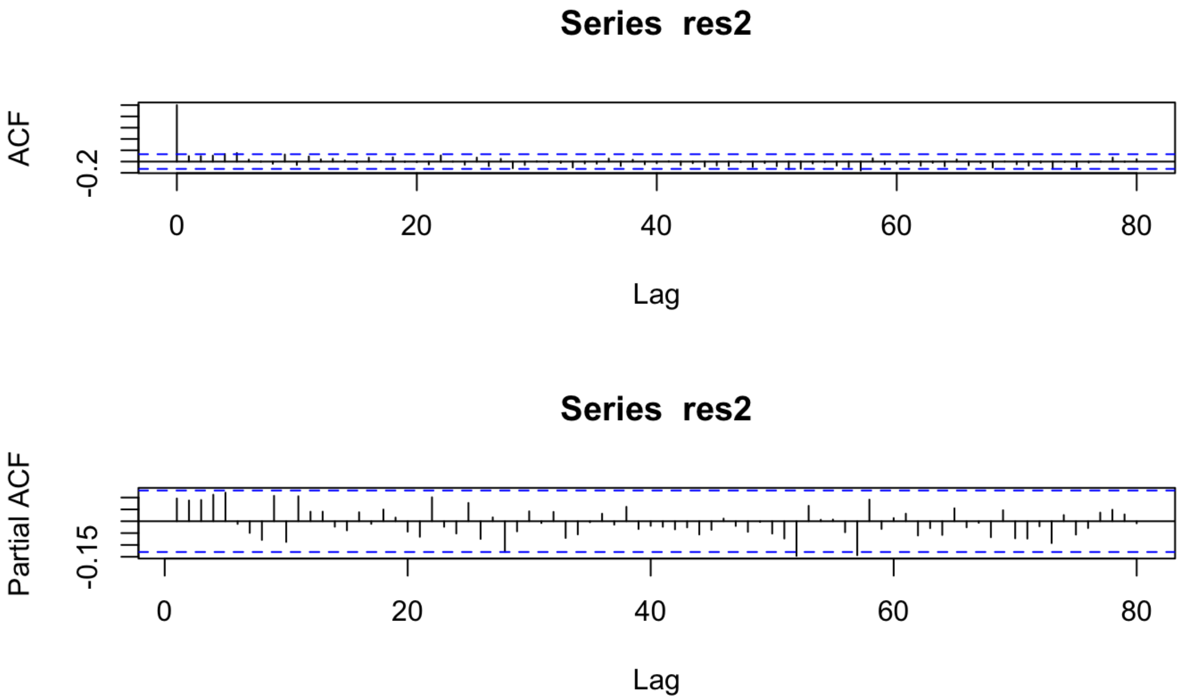


Figure 17: ACF/PACF of residuals of Model 2

As part of diagnostic checking, we also check for several independence assumptions in Table 5. Model 2 passes only half of the four tests, which is less than Model 1's passes.

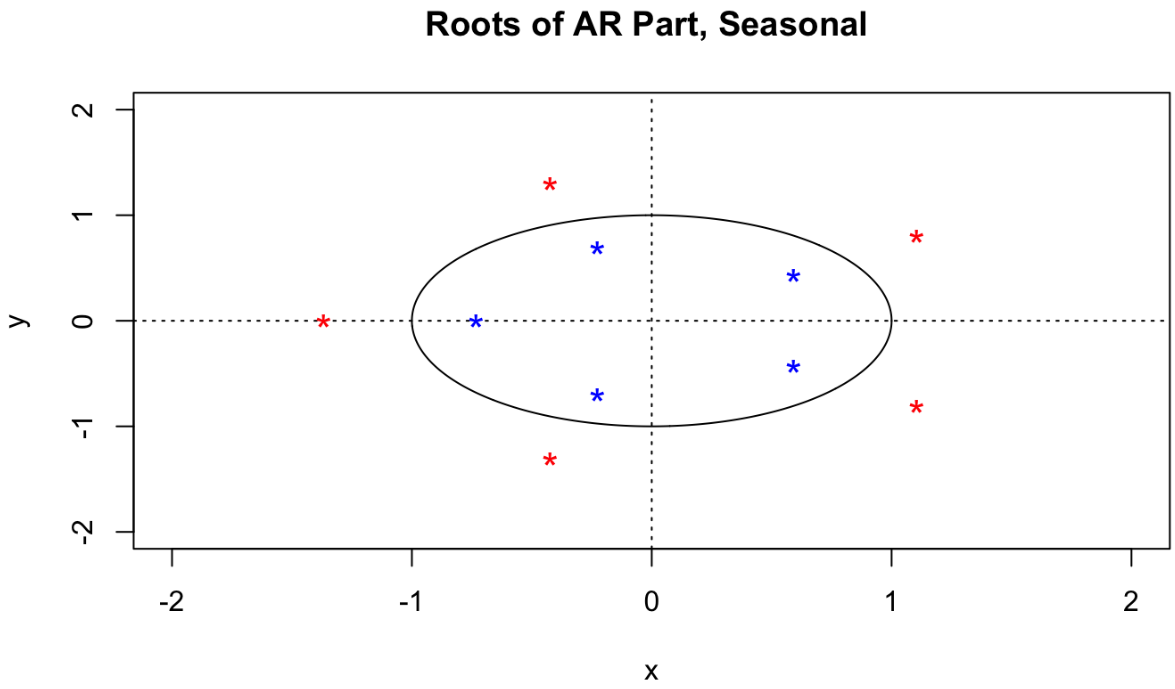
Test	Statistics	P-value	Result
Shapiro-Wilk	0.98783	0.04978	fail/borderline pass
Box-Pierce	26.31	0.1218	pass
Ljung-Box	27.354	0.09668	pass
McLeod-Li	37.251	0.01092	fail

Table 5: Testing results for Model 2

From the diagnostic checking sections 3.1 and 3.2, we can conclude that Model 1 (SARIMA(0,0,5)(0,1,1)[12]) is the better of the two models and proceed onto checking for invertibility and stationarity of our final model.

Section 4: Invertibility & Stationarity Check

Finally, we want to determine if our final model is invertible and stationary. In the seasonal moving average part, there is only one parameter, which has an estimated coefficient of 0.2084, so it is invertible. This is also supported by the roots of polynomials plotted in red, which are located outside the unit circle, indicating invertibility. Checking the roots of the SMA1 polynomial in Figure 18, we can see that the roots lie directly on the unit circle and are unit roots, meaning the model is non-stationary.



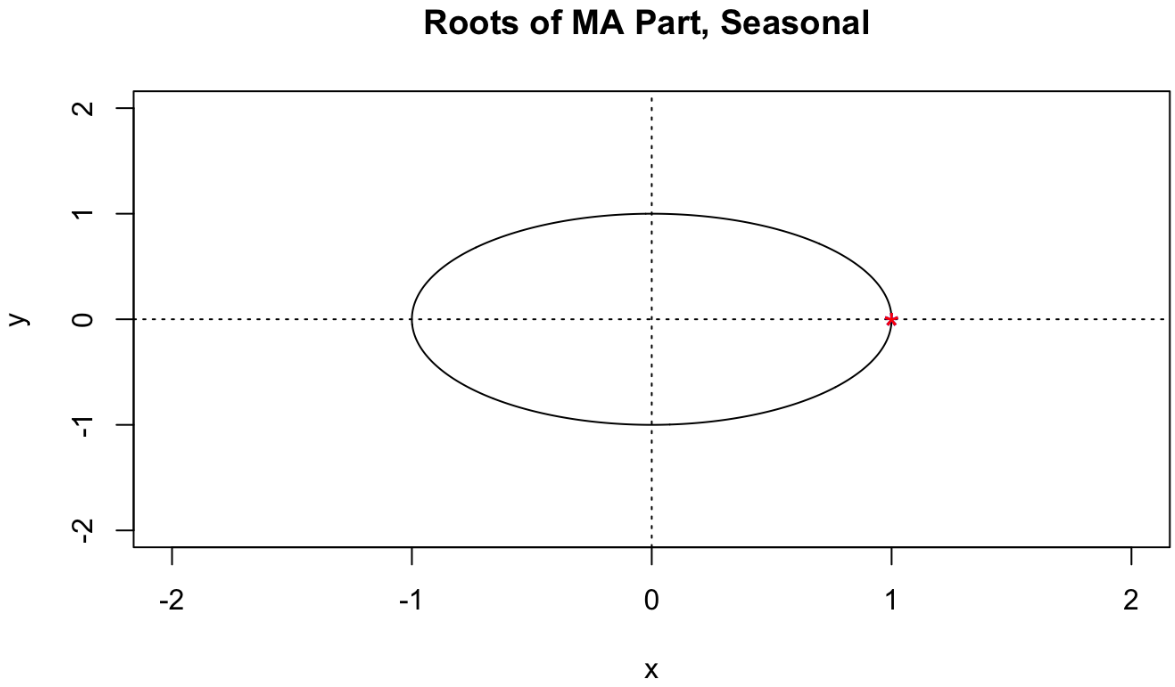


Figure 18: Roots of polynomials represented by red stars

Section 5: Final Model

Here is the final model: SARIMA(0,0,5)(0,1,1)[12] in algebraic form:

$$\nabla_{12}(U_t) = (1 + 0.2084_{(0.0726)}B^5)(1 - 1_{(0.1386)}B^{12})Z_t$$

Section 6: Forecasting

For forecasting, we predict the mean maximum temperature in Melbourne, Australia based on our final model and compare the predictions with the true values. To start off with, I produced a graph with 12 forecasts on the transformed data.

We can see that in both graphs of Figure 19, the 12 forecasts and confidence intervals are almost identical, indicating the need to further assess the accuracy of these forecasts.

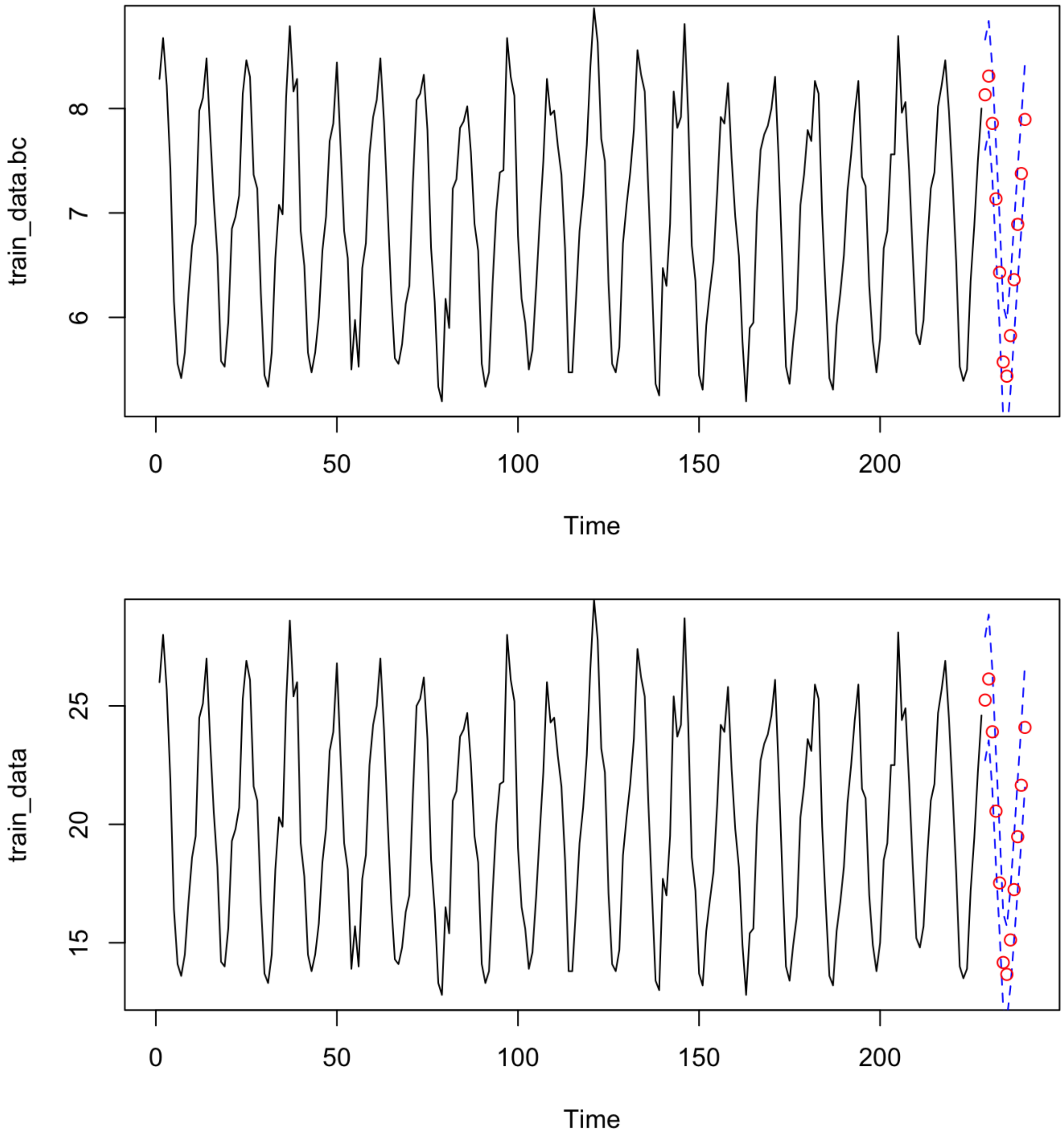
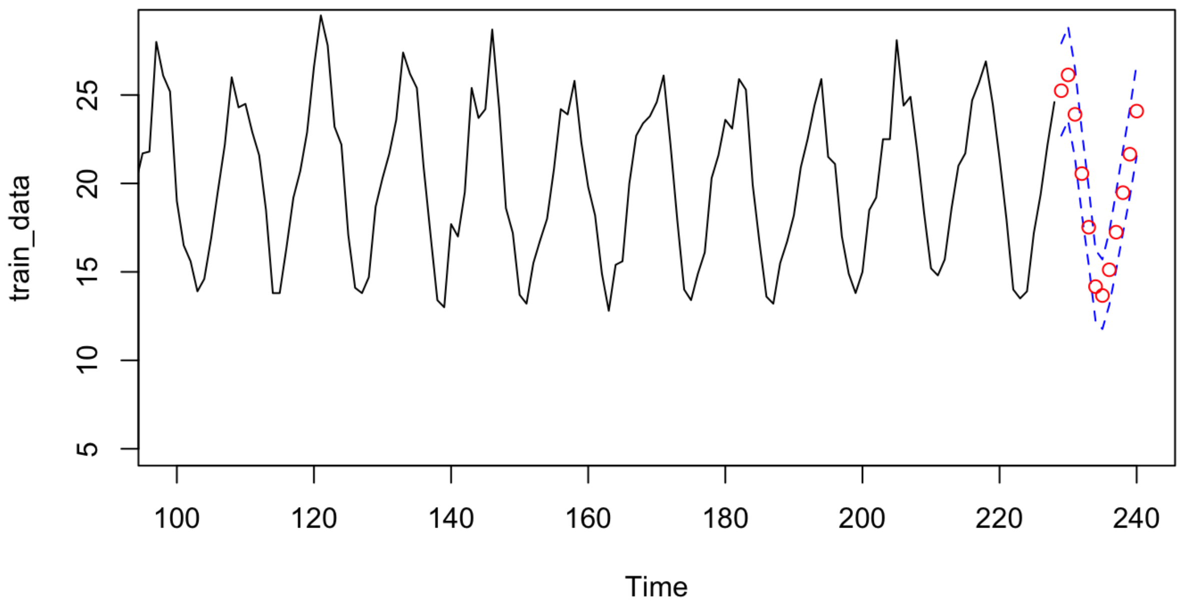
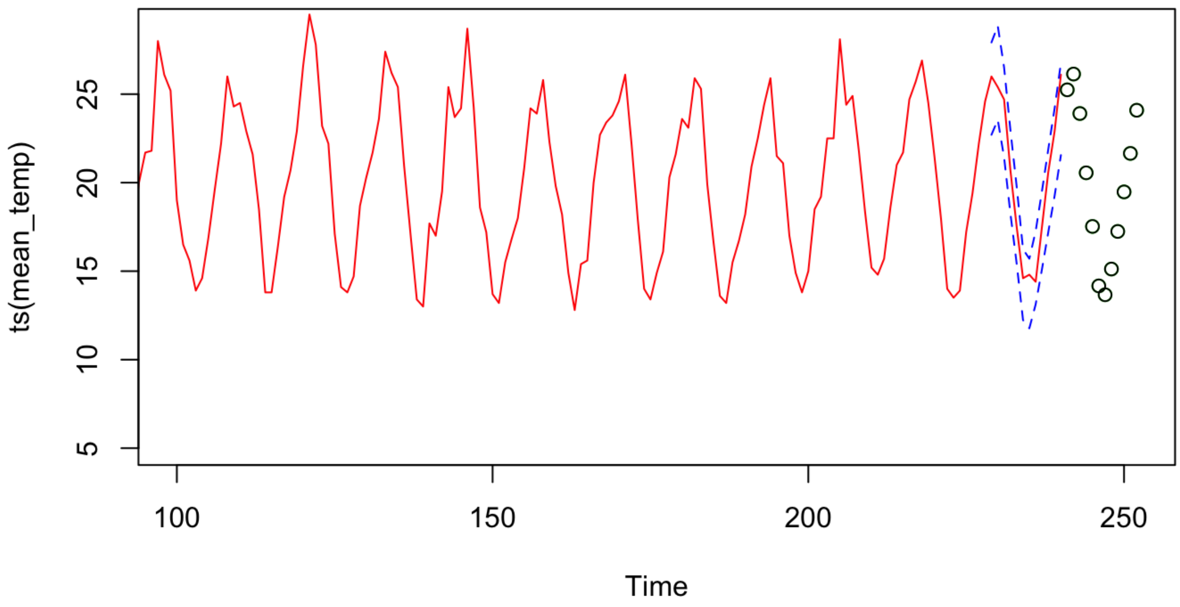


Figure 19: Top: forecasting on transformed dataset, Bottom: forecasting on original dataset

Again, the forecasted points are all within the confidence interval in the zoomed in graph starting from entry 100, indicating that the model performed quite well in forecasting.





Section 7: Conclusion

In this project, we used a dataset of monthly mean maximum temperature in Melbourne, Australia obtained from the Australian Bureau of Meteorology and also part of the Time Series Data Library (tsdl) to investigate the relationship between temperature and time. To address our goal of developing a predictive model for the mean maximum temperature in Melbourne using the "Mean Maximum Temperature in Melbourne (1971 – 1990)" dataset, we applied different time series analysis and diagnostic techniques, specifically SARIMA modeling, Box-Jenkins methodology, and checking residuals for normality. By doing so, we aimed to capture the seasonal components of the temperature data and provide reliable forecasts. Our analysis suggested that SARIMA modeling was the most suitable for this dataset, resulting in our final model: SARIMA(0,0,5)(0,1,1)[12] with an algebraic form:

$$\nabla_{12}(U_t) = (1 + 0.2084_{(0.0726)}B^5)(1 - 1_{(0.1386)}B^{12})Z_t$$

We successfully identified a clear seasonal pattern in data, allowing us to apply a Box-Cox transformation as well as differencing the data at lag 12. By passing 3 of the 4 diagnostic checks, our final model performed satisfactorily, capturing the important patterns and dependencies in the data.

In conclusion, our goals of developing a forecasting model and utilizing time series techniques to understand the temperature patterns in Melbourne were achieved.

Section 8: References

Yang, F. (n.d.). tsdl: Time Series Data Library (Version 2023.03.1+446) [R package]. Retrieved from https://github.com/FinYang/tsdl

Section 9: Appendix

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

# load packages
library(tsdl)
library(MASS)
library(forecast)
library(tidyverse)
library(ggplot2)
library(ggfortify)
library(MuMIn)
library(forecast)
```

```r
# create time series object
mean_temp <- tsdl[[90]]

# length of the data / number of observations
length(mean_temp)

# subject of the data
attr(mean_temp, "subject")

# source of the data
attr(mean_temp, "source")

# description of the data
attr(mean_temp, "description")

summary(mean_temp)
str(mean_temp)
mean_temp
```

```r
# plot of original series
#par(mfrow=c(2, 1))
ts.plot(mean_temp, main = "Mean Maximum Temperature in Melbourne (1971 – 1990)", ylab =
"Temperature (°C)")
mean(mean_temp)
var(mean_temp)
```

```
# plot original time series
#par(mfrow=c(2, 1))
ts.plot(mean_temp, main = "Mean Maximum Temperature in Melbourne (1971 – 1990)", ylab =
"Temperature (°C)")

# add trend to data plot
fit <- lm(mean_temp ~ time(mean_temp))
abline(fit, col = "red")

# add mean (constant) to data plot
mean(mean_temp)
abline(h=mean(mean_temp), col="blue")

# produce decomposition
y <- ts(as.ts(mean_temp), frequency = 12)
decomp <- decompose(y)
plot(decomp)
```

Immediate Observations: - no linear trend - seasonality - stationary - -constant variance and mean

```
# training dataset
train_data <- mean_temp[c(1:228)]

# testing dataset
test_data <- mean_temp[c(229:240)]
```

```
# plot training data
plot.ts(train_data)
fit <- lm(train_data ~ as.numeric(1:length(train_data)))
abline(fit, col="red") # added trend to data plot
abline(h=mean(train_data), col="blue") # added mean (constant) to data plot
```

```
# histogram to confirm non-stationarity of original data
hist(train_data, density = 20, breaks = 20, col= "light blue", xlab="", main="Histogram
of Training Data", prob = TRUE)

# acf plot of original data
#par(mfrow=c(2, 1))
acf(train_data,lag.max=40, main="ACF of the Training Data")
pacf(train_data,lag.max=40, main="PACF of the Training Data")
```

```
# BOX-COX TRANSFORMATION

# plot the graph
bcTransform <- boxcox(train_data~ as.numeric(1:length(train_data)))

# find value of parameter lambda
bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]

# apply box-cox transformation
train_data.bc = (1/lambda)*(train_data^lambda-1)

# plot
#par(mfrow=c(2, 1))
plot.ts(train_data.bc)
```

```
# LOG TRANSFORMATION

# apply log transformation
train_data.log <- log(train_data)

# plot data
#par(mfrow=c(2, 1))
plot.ts(train_data.log)

# compare variances
var(train_data.bc) # box-cox
mean(train_data.bc)

var(train_data.log) # log
mean(train_data.log)
```

```
# histogram of box-cox transformed data
#par(mfrow=c(2, 2))
hist(train_data.bc, density = 20, breaks = 20, col= "blue", xlab="", main="Histogram of
Box-Cox Transformed Data", prob = TRUE)
m <- mean(train_data.bc)
std <- sqrt(var(train_data.bc))
curve(dnorm(x, m, std), add=TRUE )

# histogram of log transformed data
#par(mfrow=c(2, 2))
hist(train_data.log, density = 20, breaks = 20, col= "blue", xlab="", main="Histogram of
Log Transformed Data", prob = TRUE)
m <- mean(train_data.log)
std <- sqrt(var(train_data.log))
curve(dnorm(x, m, std), add=TRUE )
```

- variance more stable after transformations
- choose Box-Cox Transformation

```
# decomposition of Box-Cox transformed data
x <- ts(as.ts(train_data.bc), frequency = 12)
decomp <- decompose(x)
plot(decomp)
```

```
# DIFFERENING

# differencing box-cox transformed data at lag 12
train_data.bc_12 <- diff(train_data.bc, lag = 12)

# plot data
#par(mfrow=c(2, 1))
plot.ts(train_data.bc_12, main="Differenced at Lag 12 to Remove Seasonality")
var(train_data.bc_12)

# add trend to data plot
fit <- lm(train_data.bc_12 ~ as.numeric(1:length(train_data.bc_12))); abline(fit, col="r
ed")
mean(train_data.bc_12)
abline(h=mean(train_data.bc_12), col="blue")
```

- seasonality no longer apparent
- significantly lower variance
- no trend
- no need to difference any further

```
# sample acf/pacf of original data
#par(mfrow=c(2, 2))
acf(train_data, lag.max = 40, main = "ACF")
pacf(train_data, lag.max = 40, main = "PACF")

# sample acf/pacf of box-cox transformed data
acf(train_data.bc, lag.max = 40, main = "ACF")
pacf(train_data.bc, lag.max = 40, main = "PACF")

# sample acf/pacf of box-cox transformed & differenced at lag 12 data
#par(mfrow=c(2, 1))
acf(train_data.bc_12, lag.max = 40, main = "ACF")
pacf(train_data.bc_12, lag.max = 40, main = "PACF")
```

- p = 5, 7, 8, 10;
- d = 0
- q = 5 or 10 (ACF slightly outside C.I.)
- P = 1 or 2 or 3?
- Q = 1 (ACF spike at lag 12) or 2 (ACF spike at lag 22)
- D = 1
- s = 12

```
# histogram of original data with normal curve
#par(mfrow=c(2, 2))

hist(train_data, density = 20, breaks = 20, col= "light blue", xlab="", main="Histogram of Training Data", prob = TRUE)
m <- mean(train_data)
std <- sqrt(var(train_data))
curve(dnorm(x, m, std), add=TRUE )

# histogram of box-cox transformed & differenced (lag 12) data with normal curve
hist(train_data.bc_12, density=20,breaks=20, col="blue", xlab="", main="Histogram of Transformed Data", prob=TRUE)
m <- mean(train_data.bc_12)
std <- sqrt(var(train_data.bc_12))
curve(dnorm(x, m, std), add=TRUE )
```

- histogram looks symmetric and almost Gaussian

List of Candidate Models:

SAR: s=12, d=0, D=1, p=5, P=1 SMA: s=12, d=0, D=1, q=5, Q=1 SARIMA: s=12, d=0, D=1, p=q=5, P=Q=1

- SARIMA(5,0,5)(1,1,0)[12]
- SARIMA(0,0,5)(0,1,1)[12]
- SARIMA(5,0,5)(1,1,1)[12]

```
# Fitting Models
# SAR
fit0 <- arima(train_data.bc, order = c(5, 0, 0), seasonal = list(order = c(1, 1, 0), period = 12), method = "ML")
fit0

# SMA
fit1 <- arima(train_data.bc, order = c(0, 0, 5), seasonal = list(order = c(0, 1, 1), period = 12), method = "ML")
fit1

# SARIMA
fit2 <- arima(train_data.bc, order = c(5, 0, 5), seasonal = list(order = c(1, 1, 1), period = 12), method = "ML")
fit2

AICc(fit0)
AICc(fit1) # lowest AICc
AICc(fit2)
```

```
# MODEL 1: SMA
model1 <- arima(train_data.bc, order = c(0, 0, 5), seasonal = list(order = c(0, 1, 1), period = 12), method = "ML", fixed = c(0,0,0,0,NA,NA))

model1
```

$∇*{12}(U_t) = (1 + 0.2084*{(0.0726)}B^5)(1 - 1_{(0.1386)}B^{12})Z_t $

```
# MODEL 2: SARIMA(5,0,5)(1,1,1)[12]
model2 <- arima(train_data.bc, order = c(5, 0, 5), seasonal = list(order = c(1, 1, 1), period = 12), method = "ML", fixed = c(0,0,0,0,0,0,0,0,0,0,0,NA))

model2
```

```
# MODEL 1 Diagnostic Tests
model1 <- arima(train_data.bc, order = c(0, 0, 5), seasonal = list(order = c(0, 1, 1), period = 12), method = "ML", fixed = c(0,0,0,0,NA,NA))

res1 <- residuals(model1)

# diagnostic check
#par(mfrow=c(2, 2))
hist(res1, density=20, breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res1)
std <- sqrt(var(res1))
curve(dnorm(x,m,std), add=TRUE )

plot.ts(res1, main = "Plot of residuals")
res_fit1 <- lm(res1 ~ as.numeric(1:length(res1)))
abline(res_fit1, col="red")
abline(h=mean(res1), col="blue")

qqnorm(res1,main= "Normal Q-Q Plot for Chosen Model")
qqline(res1,col="blue")

#par(mfrow=c(2, 1))
acf(res1, lag.max=80)
pacf(res1, lag.max=80)

#par(mfrow=c(1, 1))
shapiro.test(res1)
Box.test(res1, lag = 20, type = c("Box-Pierce"), fitdf = 2)
Box.test(res1, lag = 20, type = c("Ljung-Box"), fitdf = 2)
Box.test(res1^2, lag = 20, type = c("Ljung-Box"), fitdf = 0)
acf(res1^2, lag.max=40)
ar(res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
mean(res1)
```

```
# MODEL 2 Diagnostic Tests
model2 <- arima(train_data.bc, order = c(5, 0, 5), seasonal = list(order = c(1, 1, 1), period = 12), method = "ML", fixed = c(0,0,0,0,0,0,0,0,0,0,0,NA))

res2 <- residuals(model2)

# diagnostic check
#par(mfrow=c(2, 2))
hist(res2, density=20, breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res2)
std <- sqrt(var(res2))
curve(dnorm(x,m,std), add=TRUE )

plot.ts(res2, main = "Plot of residuals")
res_fit2 <- lm(res2 ~ as.numeric(1:length(res2)))
abline(res_fit2, col="red")
abline(h=mean(res2), col="blue")

qqnorm(res2,main= "Normal Q-Q Plot for Chosen Model")
qqline(res2,col="blue")

#par(mfrow=c(2, 1))
acf(res2, lag.max=80)
pacf(res2, lag.max=80)

#par(mfrow=c(1, 1))
shapiro.test(res2)
Box.test(res2, lag = 20, type = c("Box-Pierce"), fitdf = 1)
Box.test(res2, lag = 20, type = c("Ljung-Box"), fitdf = 1)
Box.test(res2^2, lag = 20, type = c("Ljung-Box"), fitdf = 0)
acf(res2^2, lag.max=40)
ar(res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
mean(res2)
```

```
# check invertibility/stationarity of final model (MODEL 1)

final_model <- arima(train_data.bc, order = c(0, 0, 5), seasonal = list(order = c(0, 1, 1), period = 12), method = "ML", fixed = c(0,0,0,0,NA,NA))
final_model

source("plot.roots.R.txt")

plot.roots(NULL,polyroot(c(1, 0, 0, 0, 0, 0.2084)), main="Roots of AR Part, Seasonal ")
# SMA model is invertible b/c roots outside unit circle

plot.roots(NULL,polyroot(c(1, -1.0000)), main="Roots of MA Part, Seasonal ")
# SMA model is nonstationary b/c root lies on the unit circle; unit root; does NOT pass
# forecasting using final model
final_model <- arima(train_data.bc, order = c(0, 0, 5), seasonal = list(order = c(0, 1, 1), period = 12), method = "ML", fixed = c(0,0,0,0,NA,NA))

forecast(final_model)

# graph with 12 forecasts on transformed data
pred.tr <- predict(final_model, n.ahead = 12)
U.tr = pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(train_data.bc, xlim=c(1,length(train_data.bc)+12), ylim = c(min(train_data.bc),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(train_data.bc)+1):(length(train_data.bc)+12), pred.tr$pred, col="red")
```

```
# graph with forecasts on original data
pred <- exp(log(lambda*pred.tr$pred+1)/lambda)
U.tr <- exp(log(lambda*U.tr+1)/lambda)
L.tr <- exp(log(lambda*L.tr+1)/lambda) # inverse box-cox transformation
ts.plot(train_data, xlim=c(1,length(train_data)+12), ylim = c(min(train_data), max(U.tr))) # change limits; same scale
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(train_data)+1):(length(train_data)+12), pred, col="red")
```

```
# zoom the graph starting from entry 100
ts.plot(train_data, xlim = c(100,length(train_data)+12), ylim = c(5,max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(train_data)+1):(length(train_data)+12), pred, col="red")
```

```
# plot zoomed forecasts and true values (in mean_temp)
ts.plot(ts(mean_temp), xlim = c(100,length(mean_temp)+12), ylim = c(5,max(U.tr)), col="red")
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(mean_temp)+1):(length(mean_temp)+12), pred, col="green")
points((length(mean_temp)+1):(length(mean_temp)+12), pred, col="black")
```