Temperature Trends Analysis and Forecasting Using Time Series Models

Atiq Ur Rehman

Abstract

This study analyzed two time-series datasets: the monthly maximum temperature of Canberra, Australia, and yearly global temperature anomalies. For the canberra monthly temperature, we used three seasonal auto-regressive integrated moving average (SARIMA) models. These three models are selected based on the auto-correlation function (ACF), partial auto-correlation function (PACF) and Akaike Information criteria (AIC). The diagnostic tests of the three models suggested that they were a good fit for the data considered. From the three models, we needed to select one final model, which was selected based on the hypothesis testing of the coefficients of the model and prediction accuracy of the three models. The final model for the Canberra temperature is SARIMA(1,0,1)(0,1,1)$_{12}$, which provides good forecasting of the Canberra monthly temperature. No seasonality was observed in global temperature anomalies. Therefore, we selected three autoregressive integrated moving average (ARIMA) models based on ACF, PACF, and AIC. Of the three models, we selected one final model based on hypothesis testing of the coefficients and the prediction accuracy of each model. The final ARIMA(2,1,2) model for global temperature anomalies suggests an increase in future global temperatures. The report also emphasized the importance of model evaluation and parsimony in selecting the most appropriate models for temperature forecasting.

Contents

1. Introduction

The main objective of this study is to analyze two time-series datasets using different time-series models and forecast for future temperature trends. The first dataset is the monthly maximum temperature of Canberra, Australia, from September 2008 to April 2023, downloaded from Australian Bureau of Meteorology[1]. There are several weather stations in Canberra; however, we only considered a weather station near Canberra airport. The station number is 070014, with latitude and longitude of-35.3049 and 149.2014, respectively. We can consider other weather stations, but there is not much variation in temperature in Canberra. Therefore, only one can be used for the predictions. The second dataset is the yearly global temperature anomalies from 1850 to 2022 taken from our world in data[2], which is a non-stationary dataset exhibiting a trend. We will see in the upcoming analysis how we can show that these time-series data are stationary or nonstationary.

One question that may arise in your mind is what the temperature anomalies are and why we use it? Temperature anomalies can be defined as a measure of the temperature variation from a reference/base period. They help us understand how temperatures in a given year or period differ from those in the reference period. For clarity, we take an easy example: to calculate a temperature anomaly, we start with a base period, and a span of years may be a decade or two. For example, let us consider the base period from 1990 to 2010 and suppose that the average temperature is 18°C. Now, let us suppose that the mean temperature for 2022 is measured to be 18.5 degrees Celsius. To calculate the anomaly, temperature anomaly = mean temperature for 2022–average temperature for the base period. Temperature Anomaly = 18.5°C-18°C = 0.5°C. In this case, the temperature anomaly for 2022 is +0.5 degrees Celsius. This means that the average temperature for 2022 is 0.5 degrees Celsius warmer than the average temperature during the base period of 1990-2010. Negative temperature anomalies can also occur, indicating cooler temperature. For example, if the average temperature for a particular year was 17.5 degrees Celsius, the Temperature Anomaly would be: temperature anomaly = 17.5°C - 18°C = -0.5°C.

We will explore various time-series models, including SARIMA and ARIMA, to identify the best model that fits each dataset. To ensure that our models adequately capture the underlying patterns and make good forecasts, the analysis involved data pre-processing, model fitting, and diagnostics.

The results of this study will shed light on Canberra temperature trends and long-term climatic changes worldwide. Policymakers, researchers, and other stakeholders who want to comprehend the effects of climate change and make decisions based on the predictions of future temperatures may find this material useful.

2. Data Charactristics

Figure 1(a) shows the monthly maximum temperature (instead of monthly maximum temperature, we will use the term temperature throughout our report) of Canberra, which appears to be a stationary time series upon first glance. The mean of the data is 21.1165. However, we observed that the variance in the data changed slightly over time. To solve this problem, we transformed the data usBox ing–Cox transformation and stabilized the variance. The modified data are shown in Figure 1(b), where it can be observed that the temperature variance is constant across time. Therefore, we work with the transformed data.

We have seen that the temperature data exhibit seasonality, meaning that there is a repeating pattern over a fixed period. In addition, from the ACF and PACF shown in figure 4(a) and 4(b) in the Appendix, we can see that the autocorrelation and partial autocorrelation exhibit a special pattern of seasonality after 12 months. Apart from visual inspection, we can use the augmented Dickey–Fuller test to determine whether the time series is stationary. For Canberra temperature the p-value for the dicky fuller test is 0.8893, which is greater than the significance level. Now we find the the p-value of the dicky fuller test for the seasonal difference of the Canberra temperature data 0.01, we reject the null hypothesis of non-stationarity. To analyze the data, we employed a seasonal autoregressive moving average (SARIMA) model. As the temperature data were already stationary, there was no need to take a difference to remove the trend. Instead, we used seasonal differencing to eliminate seasonality. We can better understand seasonal patterns and predict future temperature values using SARIMA with seasonal differencing.

Figure 1(c) shows the yearly global temperature anomalies, which exhibit a clear upward trend over time.

The nonstationarity of the time series can also be seen in the ACF and PACF of the global temperature in Figures 4(c) and 4(d) in the Appendix. If you use the augmented Dickey Fuller test with the p-value 0.859197, which is greater than the significance level, showing evidence of fever of not rejecting the null hypothesis of non-stationarity. This indicates that the time series was nonstationary, which may be due to long-term climate change. After taking the first difference of the global temperature the p-value of the dicky fuller test is 0.01, which is less than the significance level so rejecting null hypothesis of non-stationarity. To prepare the data for analysis, we took the first difference, as shown in Figure 1(d), which can be seen visually to remove the trend and make the series stationary.

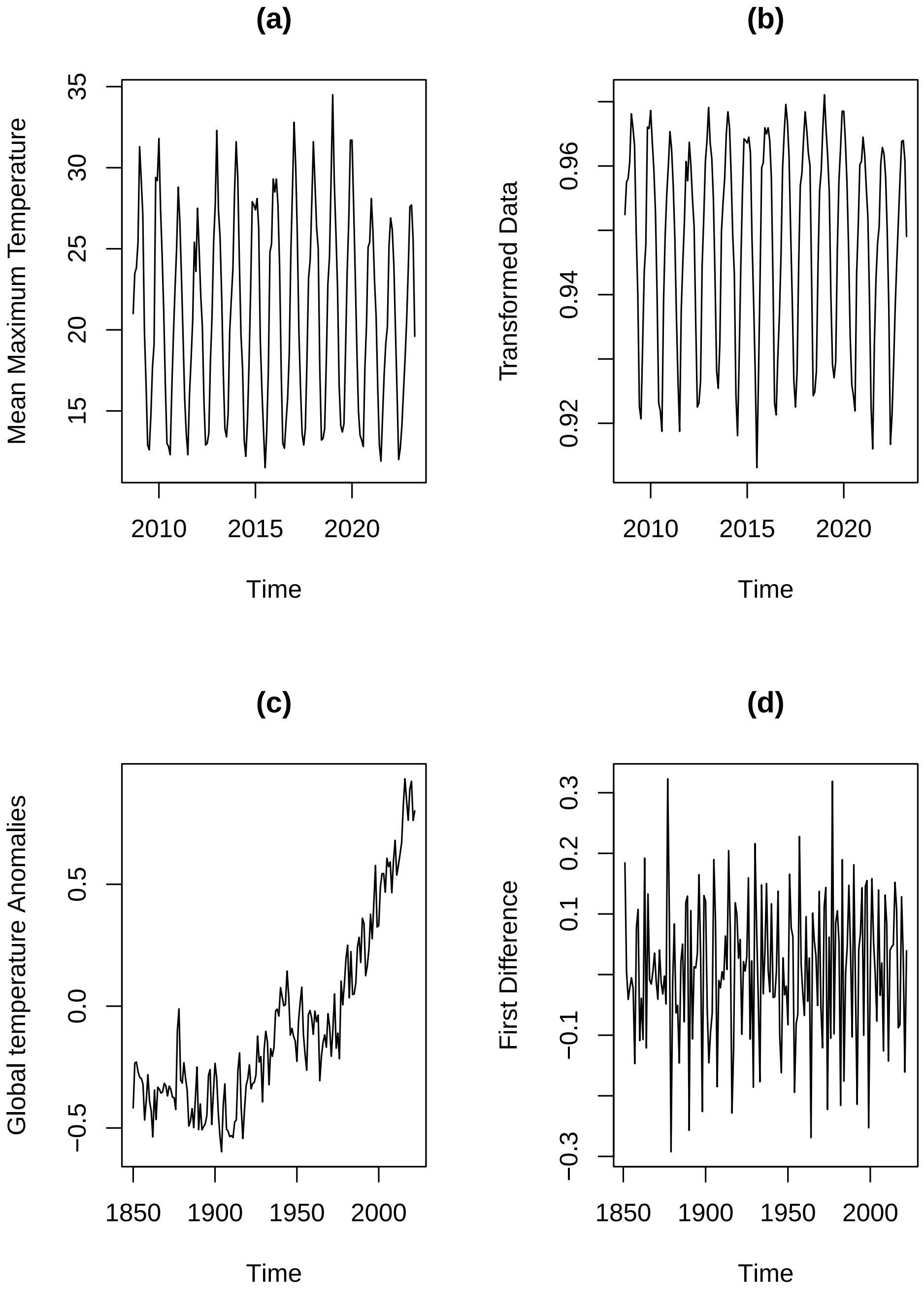

Figure 1: Time series plots illustrating: (a) Monthly temperature in Canberra from September 2008 to April 2023, and (b) the transformed data obtained through the Box-Cox transformation. Additionally, two plots displaying: (c) Original data for global temperature anomalies, and (d) Data after the first-order difference.

3. Analyzing and Evaluating the Forecasting Model

3.1 Canberra Monthly Temperature Data

We can use the autocorrelation and partial autocorrelation functions of the transformed data of the Canberra temperature after the seasonal difference, as shown in figure 2(a) and (b), respectively, to determine the ideal order of both the autoregressive (AR) and moving average (MA) parts of the seasonal and non-seasonal terms. Additionally, we have the Akaike information criterion (AIC), which aids in selecting the model with the minimum AIC. By fitting different models and choosing the model with the lowest AIC, we were able to select the best model. We chose only one model based on the AIC criteria. Table 1 displays the three models chosen based on the ACF, PACF, and AIC of the differenced data of the temperature (transformed). The R codes can be run to view the AIC values for several models of interest.

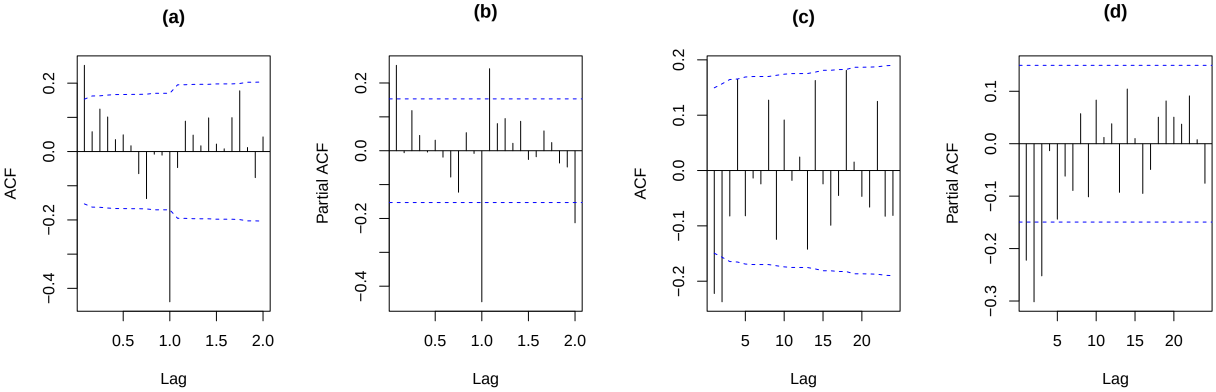

Figure 2: (a) Auto-correlation and (b) partial auto-correlation plots after applying the first seasonal difference to the Canberra temperature monthly data. (c) Auto-correlation and (d) partial auto-correlation plots of the first difference data for global temperature anomalies.

Table 1: Three SARMA models selected from the ACF and APCF plots shown in figure 2 and one from AIC. Note that the d and D shows the seasonal and non-seasonal terms.

Models	s	p	d	q	P	D	Q
SARIMA	12	1	0	1	1	1	1
SARIMA	12	1	0	1	2	1	1
SARIMA(AIC)	12	1	0	1	0	1	1

Thus, after the analysis of the three slecting models, we now need to determine which model is the best to use. We first divide the data to training and validation data, and fit the models. We take 90% of the observations as training data and the last 10% as testing data.

Before fitting the model, it is important to note that we did not include the mean to be estimated in the model. This is because we fitted our models to the training data for the monthly Canberra temperature. This training data is 90% of the observations, which we further reduced, and the SARIMA models that we chose were large. Thus, if we include drift term in the model the “NA” values will be produced on the place of standard errors (as Hessian matrix in not invertible) as well as the on place of p-values in the tables (table for the models shown below). However, we include drift in the last small model shown in Table 4, for which the observations are sufficiently large to estimate standard errors.

Table 2 summarizes the first selected model. We tested the hypothesis for each coefficient in the model and found that the two non-seasonal coefficients (AR and MA) were significant, indicating that they were important for our analysis. We rejected the null hypothesis that these coefficients were equal to zero because the p-values were very small compared to the significance level. However, for the seasonal AR and MA terms, we failed to reject the null hypothesis of the AR term; however, we rejected the null hypothesis of the MA term. This suggests that we should include one AR and MA term for the non-seasonal part and one MA term for the seasonal part in our model.

Table 3 summarizes the model with the two seasonal AR terms. For both seasonal AR terms, the p-values were greater than the significance level; therefore, we failed to reject the null hypothesis that the coefficients were equal to zero. This suggests that it is unnecessary to consider the two lags in the seasonal part.

Finally, we fitted a model without two seasonal AR terms, as fitted in the previous model, and the results are shown in Table 4. We found that it was important to consider one AR and MA term of the non-seasonal part, as well as the MA of the seasonal part, as we rejected the null hypothesis that these coefficients are equal to zero. However, the drift term is not significant; therefore, it is unnecessary to include this term.

In summary, based on statistical inference, we have reached the conclusion of using the model from Table 4, which includes one AR and MA term of the non-seasonal part and one MA term of the seasonal part, except the drift term.

Next, we checked the quality of each model's predictions by examining the residuals. Residuals represent the differences between the predicted and actual temperature values. We can check for correlations in the residuals by examining their auto-correlation and partial auto-correlation functions, which are shown in Figure 5a & b for model in table 2, Figure 5c & d for model in table 3, and 5e & f for model in table 4.

Surprisingly, we found that the residuals for all three models were white noise, meaning that they are white noise or in other words, uncorrelated and independent of each other. We can also check for normality of the residuals using a Q-Q plot, which plots the residuals against a theoretical normal distribution. Our Q-Q plots for all three models show that the residuals are normally distributed, which is depicted in figure 6 a, b, and c for all three models shown in table 2, 3, and 4, respectively. Therefore, we can conclude that all three models are a good fit for the temperature data.

We employed two distinct approaches to measure the root-mean-square error (RMSE). First, we fit the model to the training data and forecast the next 10% of the test observations. We then computed the RMSE for these fitted models. Second, we employ an open window forecasting approach by forecasting one month ahead, adding that observation to the training data, and then again fitting the model on the observations. We repeated this process until we reached the last testing observation and calculated the RMSE at each step.

The purpose of open-window forecasting is to assess the accuracy of the model's predictions. We continued with open-window forecasting until the final observation in the testing data.

The RMSE values for all three models are presented in Table 5 for both the methods. Notably, the model described in Table 4 exhibits the lowest RMSE for both the evaluation methods.

Table 2: SARIMA(1,0,1)(1,1,1)$_{12}$, Note that the 's' means seasonal components

Names	ϕ_1	θ_1	sϕ_1	sθ_1
coeff	0.8126	-0.5660	-0.1440	-1.0000
S.E	0.1287	0.1791	0.0893	0.1208
z test	6.3153	-3.1595	-1.6125	-8.2814
p-value	0.0000	0.0016	0.1069	0.0000

3.2 Global Temperature Anomalies

In the instance of global temperature anomaly data, we investigate the autocorrelation and partial autocorrelation functions of the differenced data, as shown in Figures 2(c) and 2(d), respectively, to establish a suitable order for the ARIMA model. The autocorrelation function indicates the moving average part order, whereas the partial autocorrelation function indicates the autoregressive part order. The order of the moving average

Table 3: SARIMA(1,0,1)(2,1,1)$_{12}$

Names	ϕ_1	θ_1	sϕ_1	sϕ_2	sθ_1
coeff	0.8115	-0.5623	-0.1393	0.0246	-1.0000
S.E	0.1291	0.1805	0.0915	0.0987	0.1136
z test	6.2838	-3.1145	-1.5232	0.2491	-8.8014
p-value	0.0000	0.0018	0.1277	0.8033	0.0000

Table 4: SARIMA(1,0,1)(0,1,1)$_{12}$

Names	ϕ_1	θ_1	sθ_1	drift
coeff	0.7834	-0.5430	-1.0000	0.0000
S.E	0.4291	0.4615	0.0891	0.0003
z test	1.8254	-1.1766	-11.2201	-0.0038
p-value	0.0679	0.2394	0.0000	0.9970

Table 5: Root Mean Square Error of the three models SARMA(1,0,1)(1,1,1)$_{12}$, SARMA(1,0,1)(2,1,1)$_{12}$, and SARMA(1,0,1)(0,1,1)$_{12}$. Note that We denote SARMA(1,0,1)(1,1,1)$_{12}$ RMSE by RMSE1 and Open Window RMSE1.

	rmse
RMSE1	1.6262
RMSE2	1.5993
RMSE3	1.5194
Open Window RMSE1	2.1910
Open Window RMSE2	2.1776
Open Window RMSE3	2.1210

portion should be two, according to Figure 2(c), because the fourth spike barely hits the confidence interval. Figure 2(d) indicates a 3 auto-regressive order.

We chose an ARIMA(2,1,3) model with first differencing based on the ACF and PACF findings, which includes two non-seasonal autoregressive components and three non-seasonal moving average terms. Using the AIC criteria (the results are not displayed here), we discovered two more models with the lowest AIC: ARIMA(2,1,2) and ARIMA(3,1,2). Table 6 displays all three models.

Table 6: Three ARIMA models selected from the ACF and APCF plots shown in figure 2 and two from AIC.

Models	p	d	q
ARIMA	2	2	3
ARIMA (AIC)	1	1	1
ARIMA(AIC)	3	2	2

We carefully chose three models to analyze the global temperature anomaly data by combining the ACF and PACF plots with the AIC criteria. Our goal was to fit these models and evaluate their performance through diagnostic evaluation, specifically by checking the residuals for white noise features, as shown by the ACF and PACF plots, as well as assessing the normality of the residuals.

Analyses of the ACF and PACF plots of the residuals for each model are displayed in Figures 8, 9, and 10 for the ARIMA(2,1,3), ARIMA(3,1,2), and ARIMA(2,1,2) models, respectively. We observed that the residuals of all three models exhibited white noise characteristics. Additionally, we confirmed that the residuals followed a normal distribution, further supporting the appropriateness of these models.

Moving forward, we proceeded to fit all three models; the results are presented in Tables 6, 7, and 8. Upon examining Table 6, which represents the ARIMA(2,1,3) model, we notice that the p-values for the first two MA components exceed the significance level, suggesting that these lags may not contribute significantly to the model. However, considering the slight deviation of the p-value for the first AR coefficient from the significance level, we chose to retain the lag/component.

In Table 7, which represents the ARIMA(3,1,2) model, we observe that the p-value for the second AR component exceeds the significance level, providing evidence to refrain from rejecting the null hypothesis. However, all the other p-values were below the significance level, indicating their significance in the model.

Table 8 depicts the results of the ARIMA(2,1,2) model, where the p-values for the first AR component and the second MA component surpass the significance level, suggesting a lack of evidence to reject the null hypothesis that these coefficients are equal to zero. It important to note that the drift term in all three models are not significant. Therefore, it suggests that the drift term in all three models is not important to consider.

Now, we check the assumptions of each model, that is, whether the residuals are white noise and are normally distributed. Figure 7 shows the ACF and PACF of the residuals of all three ARIMA models fitted for global temperature anomalies. The first two subfigures, a & b, are the ACF and PACF for the model shown in Table 7. Sub-figures c and d are for the model shown in Table 8 and the last two sub-figures are for 9. It can be seen from these figures that the residuals for all three models are white noise. The residuals are also normally distributed, and the Q-Q plots for each model are shown in figure 8.

Considering the results of the hypothesis testing across all three models, we are unable to definitively determine the optimal model from the given options. Further analysis and evaluation may be required to make a final decision regarding the most suitable model.

Another option is to divide the time series into training and testing data and fit all three models above and check the root mean square error (RMSE). There are two methods: as we described earlier one is direct method (we call it direct method) and another one is open window method. The RMSE of all three models for both methods are shown in Table 10. The RMSE of both methods of the model with two AR and two MA models was less than that of the other two methods. In addition, we must follow the principle of parsimony (selecting a model with fewer parameters). Thus, we selected ARIMA(2,1,2) as the final model.

Table 7: ARIMA(2,1,3)

Names	ϕ_1	ϕ_2	θ_1	θ_2	θ_3	drift
coeff	-0.3459	-0.6849	-0.0865	0.2946	-0.4530	0.0056
S.E	0.1578	0.2316	0.1617	0.2687	0.1197	0.0031
z test	-2.1923	-2.9569	-0.5352	1.0963	-3.7835	1.7810
p-value	0.0284	0.0031	0.5925	0.2730	0.0002	0.0749

Table 8: ARIMA(3,1,2)

Names	ϕ_1	ϕ_2	ϕ_3	θ_1	θ_2	drift
coeff	-0.6734	-0.0140	-0.2752	0.2844	-0.5464	0.0056
S.E	0.1400	0.1833	0.0941	0.1355	0.1299	0.0031
z test	-4.8092	-0.0764	-2.9235	2.0991	-4.2063	1.7927
p-value	0.0000	0.9391	0.0035	0.0358	0.0000	0.0730

Table 9: ARIMA(2,1,2)

Names	ϕ_1	ϕ_2	θ_1	θ_2	drift
coeff	0.4434	-0.2539	-0.8594	0.1535	0.0056
S.E	0.4708	0.1742	0.4823	0.3737	0.0031
z test	0.9417	-1.4580	-1.7820	0.4109	1.8202
p-value	0.3463	0.1448	0.0747	0.6811	0.0687

Table 10: Root Mean Square Error of the three models ARIMA(2,1,3), ARIMA(3,1,2), and ARIMA(2,1,2).

	rmse
RMSE1	1.6262
RMSE2	1.5993
RMSE3	1.5194
Open Window RMSE1	0.2264
Open Window RMSE2	0.2199
Open Window RMSE3	0.2038

4. Prediction

We have selected two model for the two dataset based on three methods, that is, Hypothesis testing of the model coefficients, RMSE of the direct method, and RMSE based on the open window method. The models coefficients estimation and their standard errors are shown in the following tables. Table 11 shows the results of the final model of the Canberra monthly temperature, and Table 12 shows the results for global temperature anomalies. The forecasting results of both these methods are plotted in Figure 3 where 3(a) shows the forecast for the next 10 months of the Canberra temperature. For Canberra temperature, this model can be used to predict future monthly temperatures. However, in the case of global temperature anomalies, the forecasting for the ten years is shown in 3(b), which is increasing as we expect future global temperature to increase due to greenhouse gases. We have included the drift term in the model as without drift the forecasting incorrect. However, hypothesis, testing shows that drift is insignificant, which I consider one limitation of the ARIMA model for global temperature anomalies.

Table 11: SARIMA$(1,0,1)(0,1,1)_{12}$

	Names	ϕ_1	θ_1	sθ_1
ar1	coeff	0.8480	-0.5888	-1.0000
	S.E	0.1139	0.1793	0.0958

Table 12: ARIMA(2,1,2)

	Names	ϕ_1	ϕ_2	θ_1	θ_2
ar1	coeff	0.3717	-0.2767	-0.7470	0.1390
	S.E	0.3746	0.1454	0.3851	0.2731

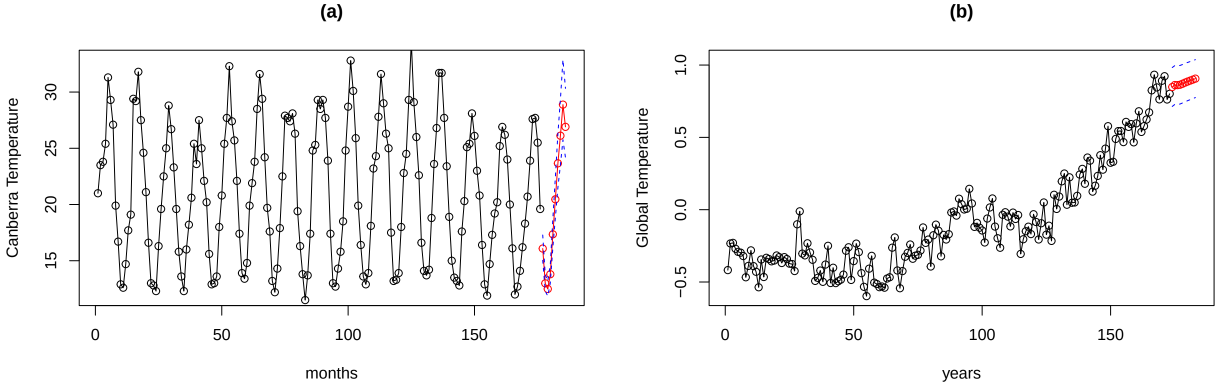

Figure 3: Forcasting of the temperature of the final model of (a): Canberra monthly temperature model shown in table 5 and (b) Global temperature anomolies model shown in 8.

5. Conclusion

In this study, we analyzed two time-series datasets: the monthly maximum temperature of Canberra, Australia, and yearly global temperature anomalies. For Canberra data, we identified a suitable SARMA model that effectively captured seasonal patterns and provided accurate forecasts. The final model incorporated non-seasonal autoregressive and moving average terms as well as a seasonal moving average term. The diagnostics confirmed the model's goodness-of-fit with white noise residuals and normality. This model can offer valuable insights into Canberra temperature trends and assist in decision-making regarding climate change.

For the global temperature anomalies, we considered three candidate ARIMA models: ARIMA(2,1,3), ARIMA(3,1,2), and ARIMA(2,1,2). In three models, based on the RMSE, we selected the final model. The final model gave good forecasting of increasing global temperature. However, the only limitation of the final model is that we should include drift term to make good forecast, but based on the hypothesis testing the drift term is not necessary to include, that is, insignificant.

In summary, our study showed the effectiveness of a SARMA model for analyzing and forecasting Canberra temperature data and the ARIMA model for global temperature anomalies with promising results. These insights provide valuable information for policymakers, researchers, and stakeholders interested in understanding temperature trends and climate change impacts, thus facilitating informed decision-making in the future.

APPENDIX

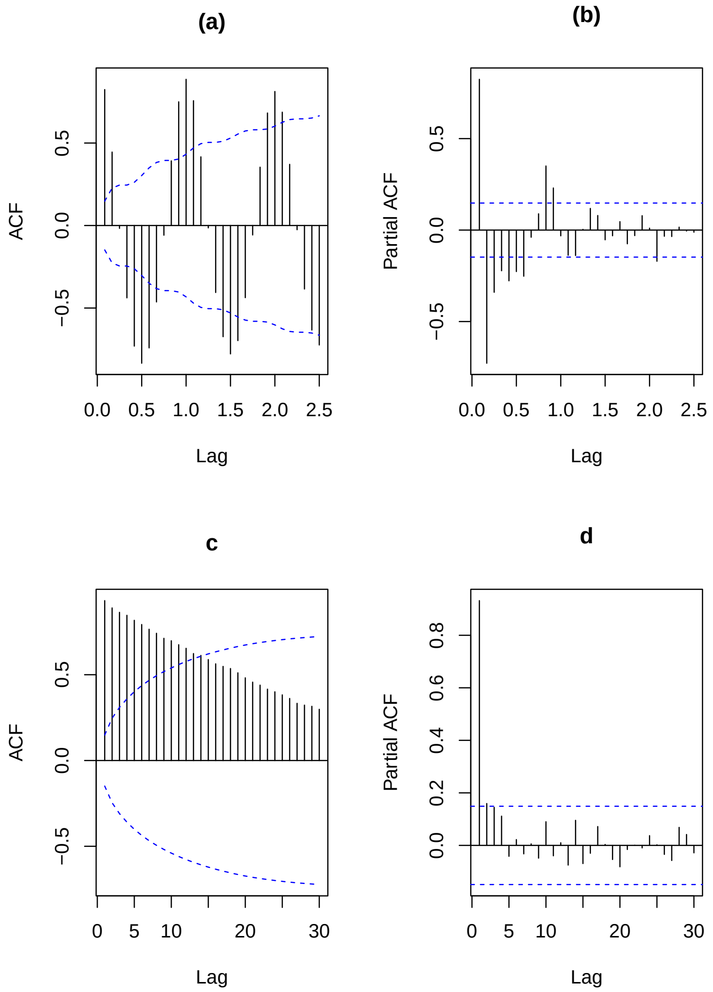

Figure 4: (a) ACF & (b) PACF plots for the canberra temperature. (c) ACF & (d) PACF for the global temperature anomalies

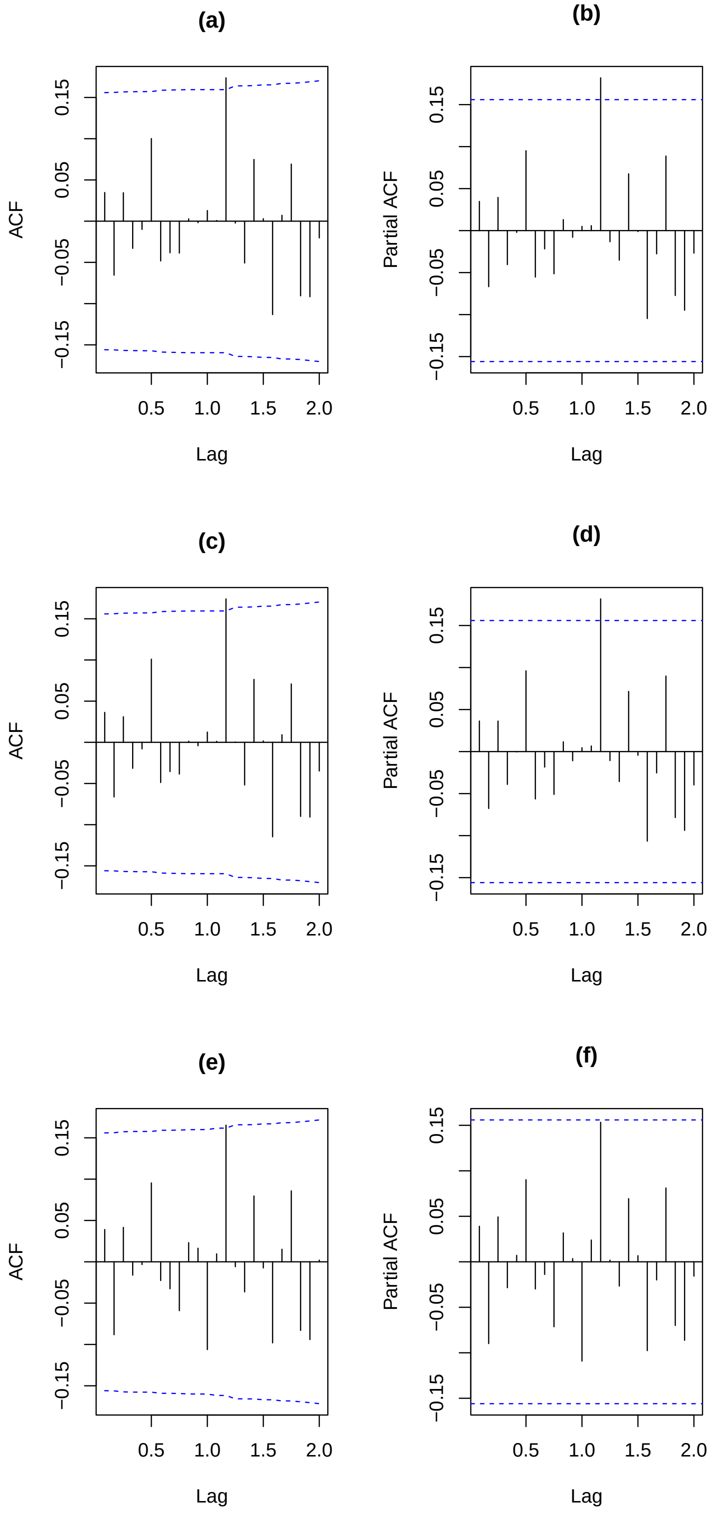

Figure 5: ACF and PACF plots showcasing the residuals of seasonal ARMA models for the monthly temperature in Canberra. Specifically: (a) & (b) ACF and PACF plots of the model shown in Table 3. (c) & (d) ACF and PACF plots of the model shown in Table 4. (e) & (f) ACF and PACF plots of the model shown in Table 5.

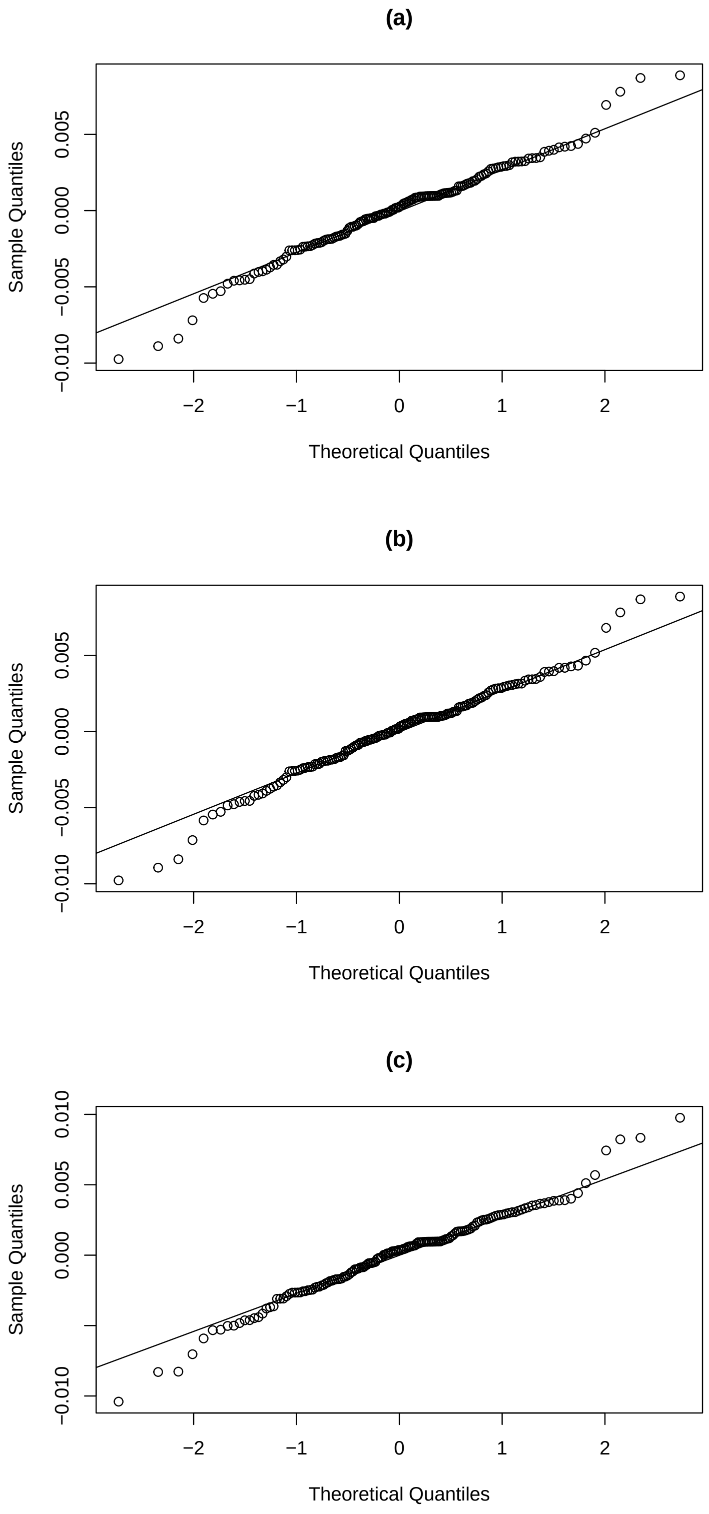

Figure 6: The normal Q-Q plots for each seasonal ARMA model of the monthly temperature in Canberra are as follows: (a) Q-Q plot for the model shown in Table 3. (b) Q-Q plot for the model shown in Table 4. (c) Q-Q plot for the model shown in Table 5.

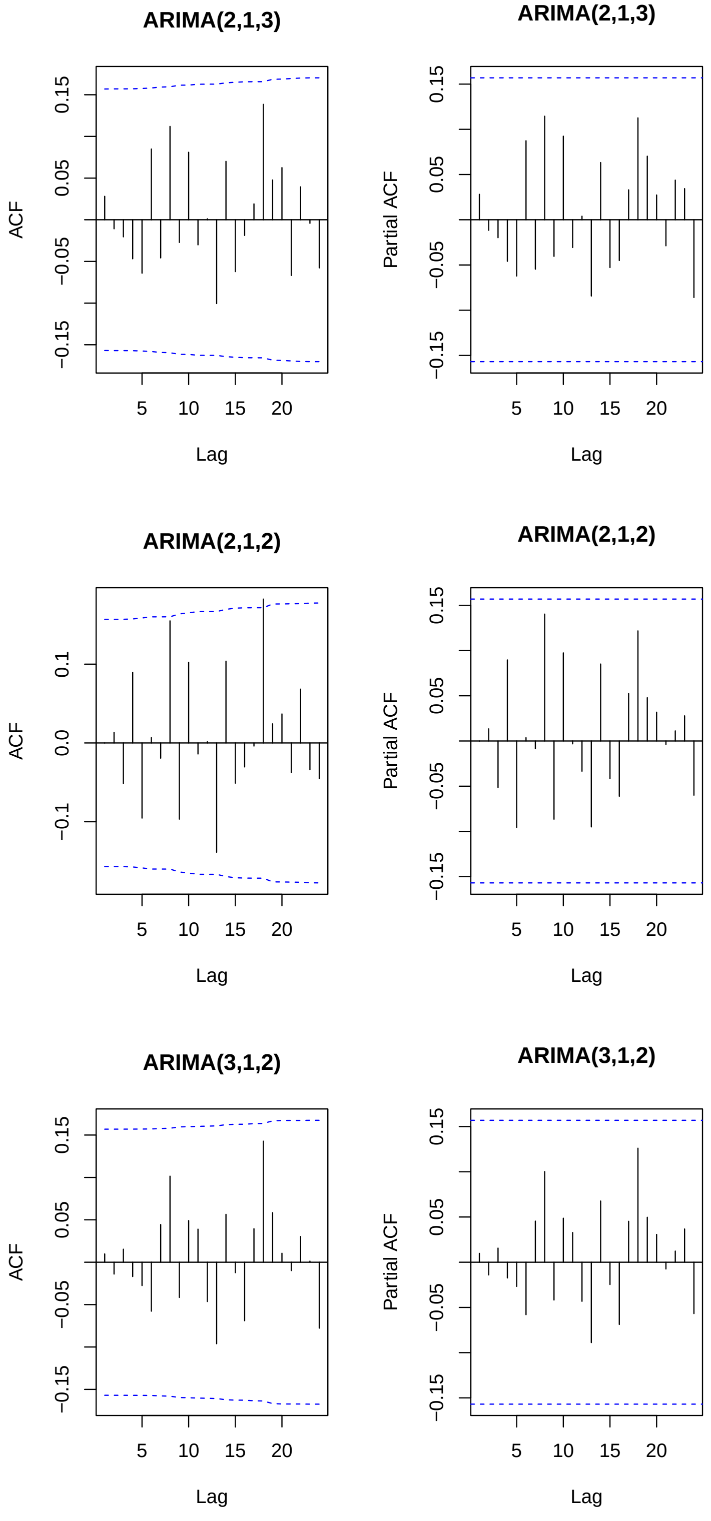

Figure 7: ACF and PACF plots showcasing the residuals of ARIMA models for the Global temperature anomalies. Specifically: (a) & (b) ACF and PACF plots of the model shown in Table 6. (c) & (d) ACF and PACF plots of the model shown in Table 7. (e) & (f) ACF and PACF plots of the model shown in Table 8.

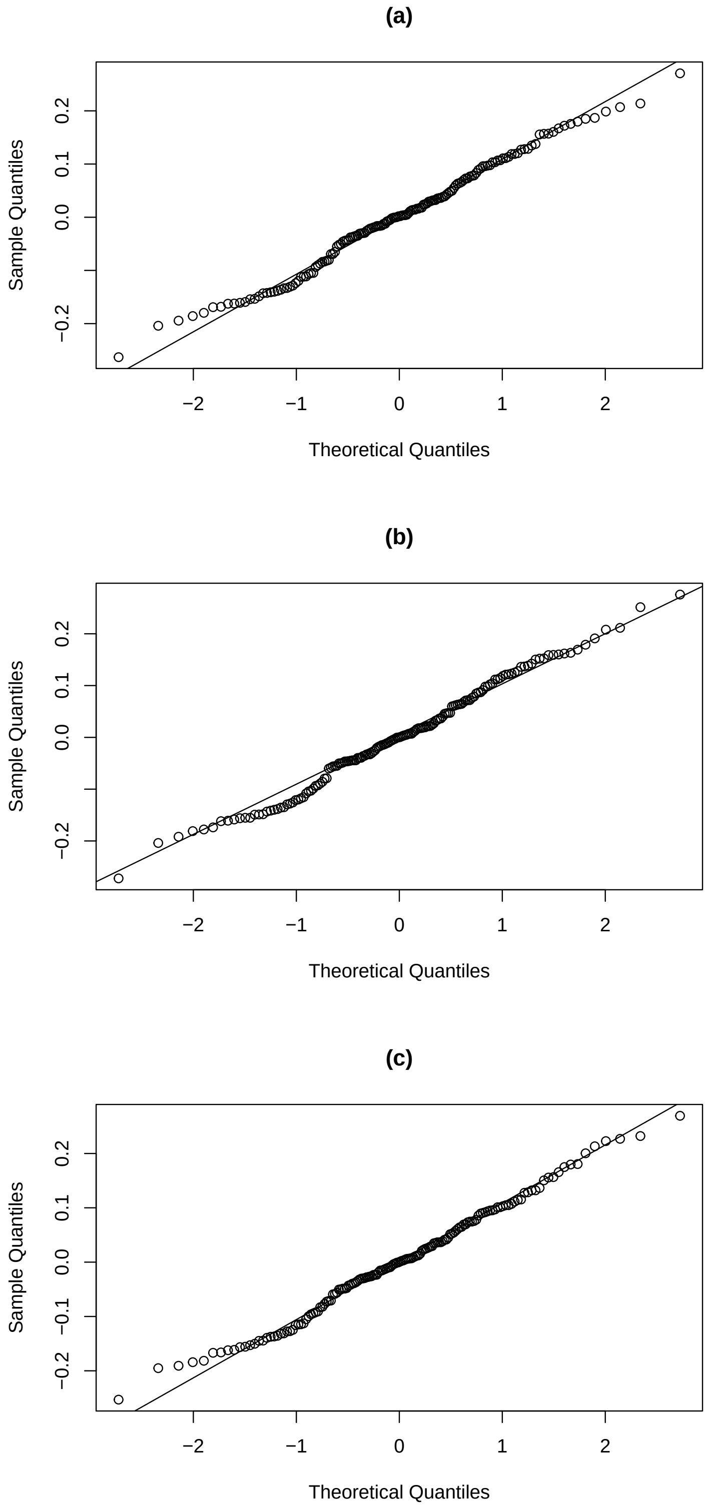

Figure 8: The normal Q-Q plots for each ARIMA model of the Global temperature anomallies are as follows: (a) Q-Q plot for the model shown in Table 6. (b) Q-Q plot for the model shown in Table 7. (c) Q-Q plot for the model shown in Table 8.

```r
library(tidyverse)
library(lubridate)
library(TSA)
library(forecast)
library(MASS)
library(progress)
library(knitr)
library(kableExtra)
library(tseries)
stationary <- read.csv("Canb_monthly_temp.csv", header = T)
non_stationary <- read.csv("temperature-anomaly.csv", header = T)



canb_data <- stationary %>%
  mutate(monthly = make_date(year = Year, month = Month),
         Max.temp = Mean.maximum.temperature...C.) %>%
  dplyr::select(-c("Product.code", "Bureau.of.Meteorology.station.number",
                   "Year", "Month",
                   "Quality", "Mean.maximum.temperature...C.")) %>%
  {ts(.$Max.temp, start=c(2008,9), frequency=12)}

Glob_temp <- non_stationary %>%
  filter(Entity == "Global") %>%
  dplyr::select(-c("Entity", "Code",
                   "Upper.bound..95..confidence.interval..of.the.annual.temperature.anomaly",
                   "Lower.bound..95..confidence.interval..of.the.annual.temperature.anomaly")) %>%
  mutate(Temp = Global.average.temperature.anomaly.relative.to.1961.1990) %>%
  dplyr::select(-c("Global.average.temperature.anomaly.relative.to.1961.1990")) %>%
  {ts(.$Temp, start = 1850, frequency = 1)}

## Seems like the variance is changing
## I use Box Cox transformation

BC_lambda <- BoxCox.lambda(canb_data)
tranf_data <- BoxCox(canb_data, BC_lambda)



par(mfrow = c(1, 2))
plot(canb_data, main = "(a)", ylab = "Mean Maximum Temperature")
# lines(ma6, col = "red", lwd = 2)
# lines(ma12, col = "blue", lwd = 2)

plot(tranf_data, main = "(b)", ylab = "Transformed Data")

first_diff <- diff(Glob_temp)
par(mfrow = c(1, 2))
plot(Glob_temp, main = "(c)", ylab = "Global temperature Anomalies")
plot(first_diff, main = "(d)", ylab = "First Difference")

# acf(canb_data)
# pacf(canb_data)
```

```r
## Taking first difference to remove seasonality

first_diff <- diff(tranf_data, 12)

par(mfrow = c(1, 2))
acf(first_diff, 24, ci.type = "ma", main = "(a)")
pacf(first_diff, 24, main = "(b)")

par(mfrow = c(1, 2))
acf(diff(Glob_temp), 24, ci.type = "ma", main = "(c)")
pacf(diff(Glob_temp), 24, main = "(d)")

arima_orders <- expand.grid(p = 0:3, q = 0:2)

pb <- NULL
pb <- progress_bar$new(total = nrow(arima_orders))
# Fit the ARIMA models and calculate their AIC
models <- lapply(1:nrow(arima_orders), function(x) {
  fit <- Arima(tranf_data,
               order = c(arima_orders[x, "p"], 1,
                         arima_orders[x, "q"]))

  pb$tick() # update progress bar
  Sys.sleep(1 /nrow(arima_orders))
  AIC(fit)
})

# Combine the models and their AIC values into a data frame
results <- data.frame(arima_order = arima_orders, AIC = unlist(models))

# Print the results in order of increasing AIC
#results[results$AIC, ]
# Define the different SARIMA models you want to try
sarima_orders <- expand.grid(p = 0:2, q = 0:1, P = 0:2, Q = 0:1)

pb <- NULL
pb <- progress_bar$new(total = nrow(sarima_orders))
# Fit the ARIMA models and calculate their AIC
models <- lapply(1:nrow(sarima_orders), function(x) {
  fit <- Arima(tranf_data,
               order = c(sarima_orders[x, "p"], 0,
                         sarima_orders[x, "q"]),
               seasonal = list(order = c(sarima_orders[x, "P"],
                                         1,
                                         sarima_orders[x, "Q"]),
                               period = 12))

  pb$tick() # update progress bar
  Sys.sleep(1 /nrow(sarima_orders))
  AIC(fit)
})

# Combine the models and their AIC values into a data frame
```

```
results <- data.frame(sarima_order = sarima_orders, AIC = unlist(models))

# Print the results in order of increasing AIC
# results[results$AIC, ]

## Making table of the model selected from the ACF and PACF

df <- data.frame(Models = c("SARIMA", "SARIMA", "SARIMA(AIC)"),
                 s = c(rep(12,3)),
                 p = c(1,1,1),
                 d = c(0,0,0),
                 q = c(1,1,1),
                 P = c(1,2,0),
                 D = c(1,1,1),
                 Q = c(1,1,1))

kable(df,
      caption = "Three SARMA models selected from the
      ACF and APCF plots shown in figure 2 and one from AIC.
      Note that the d and D shows the seasonal and non-seasonal terms.  \\label{tab:Tab1}") %>%
  kable_styling(latex_options = "hold_position")
n_train <- round(length(canb_data) * 0.9)
n_test <- length(canb_data) - n_train

canb_train <- ts(canb_data[1:n_train], start=c(2008,9), frequency=12)
canb_test <- ts(canb_data[n_train+1:n_test], start=c(2008,11), frequency=12)

sarma1 <- Arima(canb_train,
                  order = c(1, 0, 1),
                  seasonal = list(order = c(1, 1, 1),
                                  period = 12),
                lambda = BC_lambda)

sarma2 <- Arima(canb_train,
                  order = c(1, 0, 1),
                  seasonal = list(order = c(2, 1, 1),
                                  period = 12),
                lambda = BC_lambda)

sarma3 <- Arima(canb_train,
                  order = c(1, 0, 1),
                  seasonal = list(order = c(0, 1, 1),
                                  period = 12),
                lambda = BC_lambda,
                include.constant = TRUE)


######################### Table for sarma 1
df_sarma1 <- data.frame(Names = c("coeff", "S.E", "z test", "p-value"),
                 phi1 = c(sarma1$coef[1],
                          sqrt(sarma1$var.coef[1,1]),
                          sarma1$coef[1]/sqrt(sarma1$var.coef[1,1]),
                          2 * pnorm(abs(sarma1$coef[1]/sqrt(sarma1$var.coef[1,1])),
                                    lower.tail = FALSE)),
                 theta1 = c(sarma1$coef[2],
                          sqrt(sarma1$var.coef[2,2]),
                          sarma1$coef[2]/sqrt(sarma1$var.coef[2,2]),
                          2 * pnorm(abs(sarma1$coef[2]/sqrt(sarma1$var.coef[2,2])),
                                    lower.tail = FALSE)),
                 sphi1 = c(sarma1$coef[3],
                          sqrt(sarma1$var.coef[3,3]),
                          sarma1$coef[3]/sqrt(sarma1$var.coef[3,3]),
                          2 * pnorm(abs(sarma1$coef[3]/sqrt(sarma1$var.coef[3,3])),
                                    lower.tail = FALSE)),
                 stheta1 = c(sarma1$coef[4],
                          sqrt(sarma1$var.coef[4,4]),
                          sarma1$coef[4]/sqrt(sarma1$var.coef[4,4]),
                          2 * pnorm(abs(sarma1$coef[4]/sqrt(sarma1$var.coef[4,4])),
                                    lower.tail = FALSE)))
names(df_sarma1)[2] <- "$\\phi_{1}$"
names(df_sarma1)[3] <- "$\\theta_{1}$"
names(df_sarma1)[4] <- "s$\\phi_{1}$"
names(df_sarma1)[5] <- "s$\\theta_{1}$"

kable(df_sarma1,
      digits = 4,
      caption = "SARIMA(1,0,1)(1,1,1)$_{12}$, Note that the 's' means seasonal components \\label{tab:Ta
      format = "latex",
      booktabs = T,
      escape = FALSE) %>%
  kable_styling(latex_options = "hold_position")

######################## table for sarma 2
df_sarma2 <- data.frame(Names = c("coeff", "S.E", "z test", "p-value"),
                 phi1 = c(sarma2$coef[1],
                          sqrt(sarma2$var.coef[1,1]),
                          sarma2$coef[1]/sqrt(sarma2$var.coef[1,1]),
                          2 * pnorm(abs(sarma2$coef[1]/sqrt(sarma2$var.coef[1,1])),
                                    lower.tail = FALSE)),
                 theta1 = c(sarma2$coef[2],
                          sqrt(sarma2$var.coef[2,2]),
                          sarma2$coef[2]/sqrt(sarma2$var.coef[2,2]),
                          2 * pnorm(abs(sarma2$coef[2]/sqrt(sarma2$var.coef[2,2])),
                                    lower.tail = FALSE)),
                 sphi1 = c(sarma2$coef[3],
                          sqrt(sarma2$var.coef[3,3]),
                          sarma2$coef[3]/sqrt(sarma2$var.coef[3,3]),
                          2 * pnorm(abs(sarma2$coef[3]/sqrt(sarma2$var.coef[3,3])),
                                    lower.tail = FALSE)),
                 sphi2 = c(sarma2$coef[4],
                          sqrt(sarma2$var.coef[4,4]),
                          sarma2$coef[4]/sqrt(sarma2$var.coef[4,4]),
                          2 * pnorm(abs(sarma2$coef[4]/sqrt(sarma2$var.coef[4,4])),
                                    lower.tail = FALSE)),
                 stheta1 = c(sarma2$coef[5],
                          sqrt(sarma2$var.coef[5,5]),
                          sarma2$coef[5]/sqrt(sarma2$var.coef[5,5]),
                          2 * pnorm(abs(sarma2$coef[5]/sqrt(sarma2$var.coef[5,5])),
                                    lower.tail = FALSE)))
names(df_sarma2)[2] <- "$\\phi_{1}$"
names(df_sarma2)[3] <- "$\\theta_{1}$"
names(df_sarma2)[4] <- "s$\\phi_{1}$"
names(df_sarma2)[5] <- "s$\\phi_{2}$"
names(df_sarma2)[6] <- "s$\\theta_{1}$"

kable(df_sarma2,
      digits = 4,
      caption = "SARIMA(1,0,1)(2,1,1)$_{12}$ \\label{tab:Tab3}",
      format = "latex",
      booktabs = T,
      escape = FALSE) %>%
  kable_styling(latex_options = "hold_position")


####################### table for 3
df_sarma3 <- data.frame(Names = c("coeff", "S.E", "z test", "p-value"),
                 phi1 = c(sarma3$coef[1],
                          sqrt(sarma3$var.coef[1,1]),
                          sarma3$coef[1]/sqrt(sarma3$var.coef[1,1]),
                          2 * pnorm(abs(sarma3$coef[1]/sqrt(sarma3$var.coef[1,1])),
                                    lower.tail = FALSE)),
                 theta1 = c(sarma3$coef[2],
                          sqrt(sarma3$var.coef[2,2]),
                          sarma3$coef[2]/sqrt(sarma3$var.coef[2,2]),
                          2 * pnorm(abs(sarma3$coef[2]/sqrt(sarma3$var.coef[2,2])),
                                    lower.tail = FALSE)),
                 stheta1 = c(sarma3$coef[3],
                          sqrt(sarma3$var.coef[3,3]),
                          sarma3$coef[3]/sqrt(sarma3$var.coef[3,3]),
                          2 * pnorm(abs(sarma3$coef[3]/sqrt(sarma3$var.coef[3,3])),
                                    lower.tail = FALSE)),
                 drift = c(sarma3$coef[4],
                          sqrt(sarma3$var.coef[4,4]),
                          sarma3$coef[4]/sqrt(sarma3$var.coef[4,4]),
                          2 * pnorm(abs(sarma3$coef[4]/sqrt(sarma3$var.coef[4,4])),
                                    lower.tail = FALSE)))
names(df_sarma3)[2] <- "$\\phi_{1}$"
names(df_sarma3)[3] <- "$\\theta_{1}$"
names(df_sarma3)[4] <- "s$\\theta_{1}$"
names(df_sarma3)[5] <- "drift"

kable(df_sarma3,
      digits = 4,
      caption = "SARIMA(1,0,1)(0,1,1)$_{12}$ \\label{tab:Tab4}",
      format = "latex",
      booktabs = T,
      escape = FALSE) %>%
  kable_styling(latex_options = "hold_position")
pred1 <- forecast(sarma1, 18)
```

```r
pred2 <- forecast(sarma2, 18)
pred3 <- forecast(sarma3, 18)

rmse1 <- sqrt(mean(as.vector(pred1$mean) - as.vector(canb_test))^2)
rmse2 <- sqrt(mean(as.vector(pred2$mean) - as.vector(canb_test))^2)
rmse3 <- sqrt(mean(as.vector(pred3$mean) - as.vector(canb_test))^2)


op_wind_train1 <- canb_train
op_wind_train2 <- canb_train
op_wind_train3 <- canb_train
num_iter <- 18

# Create a vector to store the forecasts
pred1 <- rep(NA, num_iter)
pred2 <- rep(NA, num_iter)
pred3 <- rep(NA, num_iter)

# Loop through the iterations
for (i in 1:num_iter) {

# Fit the model to the training data
op_wind_sarma1 <- Arima(op_wind_train1,
                  order = c(1, 0, 1),
                  seasonal = list(order = c(1, 1, 1),
                                  period = 12),
                lambda = BC_lambda)

op_wind_sarma2 <- Arima(op_wind_train2,
                  order = c(1, 0, 1),
                  seasonal = list(order = c(2, 1, 1),
                                  period = 12),
                lambda = BC_lambda)

op_wind_sarma3 <- Arima(op_wind_train3,
                  order = c(1, 0, 1),
                  seasonal = list(order = c(0, 1, 1),
                                  period = 12),
                lambda = BC_lambda)

# Forecast one value ahead
fc1 <- forecast(op_wind_sarma1, 1)
fc2 <- forecast(op_wind_sarma2, 1)
fc3 <- forecast(op_wind_sarma3, 1)

# Store the forecasted value
pred1[i] <- fc1$mean
pred2[i] <- fc2$mean
pred3[i] <- fc3$mean

# Add the forecasted value to the training data
op_wind_train1 <- c(op_wind_train1, pred1[i])
op_wind_train2 <- c(op_wind_train2, pred2[i])
```

```r
op_wind_train3 <- c(op_wind_train3, pred3[i])
}

rmse <- c(sqrt(mean((as.vector(pred1) - as.vector(canb_test))^2)),
          sqrt(mean((as.vector(pred2) - as.vector(canb_test))^2)),
          sqrt(mean((as.vector(pred3) - as.vector(canb_test))^2)))



df <- data.frame(rmse = c(RMSE1 = rmse1,
                          RMSE2 = rmse2,
                          RMSE3 = rmse3,
                          "Open Window RMSE1" = rmse[1],
                          "Open Window RMSE2" = rmse[2],
                          "Open Window RMSE3" = rmse[3]))

kable(df,
      digits = 4,
      caption = "Root Mean Square Error of the three models SARMA(1,0,1)(1,1,1)$_{12}$,
      SARMA(1,0,1)(2,1,1)$_{12}$, and SARMA(1,0,1)(0,1,1)$_{12}$. Note that We denote SARMA(1,0,1)(1,1,
  kable_styling(latex_options = "hold_position")

## Making table of the model selected from the ACF and PACF

df <- data.frame(Models = c("ARIMA", "ARIMA (AIC)", "ARIMA(AIC)"),
                 p = c(2,1,3),
                 d = c(2,1,2),
                 q = c(3,1,2))

kable(df,
      caption = "Three ARIMA models selected from the
      ACF and APCF plots shown in figure 2 and two from AIC. \\label{tab:Tab6}") %>%
  kable_styling(latex_options = "hold_position")

n_train <- round(length(Glob_temp) * 0.9)
n_test <- length(Glob_temp) - n_train

train <- ts(Glob_temp[1:n_train], start = 1850, frequency = 1)
test <- ts(Glob_temp[n_train+1:n_test], start = 2006, end = 2022, frequency = 1)

arima2.3 <- Arima(train, order = c(2, 1, 3), include.constant = TRUE)

arima3.2 <- Arima(train, order = c(3, 1, 2), include.constant = TRUE)

arima2.2 <- Arima(train, order = c(2, 1, 2), include.constant = TRUE)


######################### Table for arima2.3
df_arima2.3 <- data.frame(Names = c("coeff", "S.E", "z test", "p-value"),
                 phi1 = c(arima2.3$coef[1],
                          sqrt(arima2.3$var.coef[1,1]),
                          arima2.3$coef[1]/sqrt(arima2.3$var.coef[1,1]),
                          2 * pnorm(abs(arima2.3$coef[1]/sqrt(arima2.3$var.coef[1,1])),
                                    lower.tail = FALSE)),
                 phi2 = c(arima2.3$coef[2],
                          sqrt(arima2.3$var.coef[2,2]),
                          arima2.3$coef[2]/sqrt(arima2.3$var.coef[2,2]),
                          2 * pnorm(abs(arima2.3$coef[2]/sqrt(arima2.3$var.coef[2,2])),
                                    lower.tail = FALSE)),
                 theta1 = c(arima2.3$coef[3],
                          sqrt(arima2.3$var.coef[3,3]),
                          arima2.3$coef[3]/sqrt(arima2.3$var.coef[3,3]),
                          2 * pnorm(abs(arima2.3$coef[3]/sqrt(arima2.3$var.coef[3,3])),
                                    lower.tail = FALSE)),
                 theta2 = c(arima2.3$coef[4],
                          sqrt(arima2.3$var.coef[4,4]),
                          arima2.3$coef[4]/sqrt(arima2.3$var.coef[4,4]),
                          2 * pnorm(abs(arima2.3$coef[4]/sqrt(arima2.3$var.coef[4,4])),
                                    lower.tail = FALSE)),
                 theta3 = c(arima2.3$coef[5],
                          sqrt(arima2.3$var.coef[5,5]),
                          arima2.3$coef[5]/sqrt(arima2.3$var.coef[5,5]),
                          2 * pnorm(abs(arima2.3$coef[5]/sqrt(arima2.3$var.coef[5,5])),
                                    lower.tail = FALSE)),
                 drift = c(arima2.3$coef[6],
                          sqrt(arima2.3$var.coef[6,6]),
                          arima2.3$coef[6]/sqrt(arima2.3$var.coef[6,6]),
                          2 * pnorm(abs(arima2.3$coef[6]/sqrt(arima2.3$var.coef[6,6])),
                                    lower.tail = FALSE)))
names(df_arima2.3)[2] <- "$\\phi_{1}$"
names(df_arima2.3)[3] <- "$\\phi_{2}$"
names(df_arima2.3)[4] <- "$\\theta_{1}$"
names(df_arima2.3)[5] <- "$\\theta_{2}$"
names(df_arima2.3)[6] <- "$\\theta_{3}$"
names(df_arima2.3)[7] <- "drift"

kable(df_arima2.3,
      digits = 4,
      caption = "ARIMA(2,1,3) \\label{tab:Tab7}",
      format = "latex",
      booktabs = T,
      escape = FALSE) %>%
  kable_styling(latex_options = "hold_position")

######################## table for arima3.2

df_arima3.2 <- data.frame(Names = c("coeff", "S.E", "z test", "p-value"),
                 phi1 = c(arima3.2$coef[1],
                          sqrt(arima3.2$var.coef[1,1]),
                          arima3.2$coef[1]/sqrt(arima3.2$var.coef[1,1]),
                          2 * pnorm(abs(arima3.2$coef[1]/sqrt(arima3.2$var.coef[1,1])),
                                    lower.tail = FALSE)),
                 phi2 = c(arima3.2$coef[2],
                          sqrt(arima3.2$var.coef[2,2]),
                          arima3.2$coef[2]/sqrt(arima3.2$var.coef[2,2]),
                          2 * pnorm(abs(arima3.2$coef[2]/sqrt(arima3.2$var.coef[2,2])),
                                    lower.tail = FALSE)),
                 phi3 = c(arima3.2$coef[3],
                          sqrt(arima3.2$var.coef[3,3]),
                          arima3.2$coef[3]/sqrt(arima3.2$var.coef[3,3]),
                          2 * pnorm(abs(arima3.2$coef[3]/sqrt(arima3.2$var.coef[3,3])),
                                    lower.tail = FALSE)),
                 theta2 = c(arima3.2$coef[4],
                          sqrt(arima3.2$var.coef[4,4]),
                          arima3.2$coef[4]/sqrt(arima3.2$var.coef[4,4]),
                          2 * pnorm(abs(arima3.2$coef[4]/sqrt(arima3.2$var.coef[4,4])),
                                    lower.tail = FALSE)),
                 theta3 = c(arima3.2$coef[5],
                          sqrt(arima3.2$var.coef[5,5]),
                          arima3.2$coef[5]/sqrt(arima3.2$var.coef[5,5]),
                          2 * pnorm(abs(arima3.2$coef[5]/sqrt(arima3.2$var.coef[5,5])),
                                    lower.tail = FALSE)),
                 drift = c(arima3.2$coef[6],
                          sqrt(arima3.2$var.coef[6,6]),
                          arima3.2$coef[6]/sqrt(arima3.2$var.coef[6,6]),
                          2 * pnorm(abs(arima3.2$coef[6]/sqrt(arima3.2$var.coef[6,6])),
                                    lower.tail = FALSE)))

names(df_arima3.2)[2] <- "$\\phi_{1}$"
names(df_arima3.2)[3] <- "$\\phi_{2}$"
names(df_arima3.2)[4] <- "$\\phi_{3}$"
names(df_arima3.2)[5] <- "$\\theta_{1}$"
names(df_arima3.2)[6] <- "$\\theta_{2}$"
names(df_arima3.2)[7] <- "drift"

kable(df_arima3.2,
      digits = 4,
      caption = "ARIMA(3,1,2) \\label{tab:Tab8}",
      format = "latex",
      booktabs = T,
      escape = FALSE) %>%
  kable_styling(latex_options = "hold_position")

####################### table for arima2.2

df_arima2.2 <- data.frame(Names = c("coeff", "S.E", "z test", "p-value"),
                 phi1 = c(arima2.2$coef[1],
                          sqrt(arima2.2$var.coef[1,1]),
                          arima2.2$coef[1]/sqrt(arima2.2$var.coef[1,1]),
                          2 * pnorm(abs(arima2.2$coef[1]/sqrt(arima2.2$var.coef[1,1])),
                                    lower.tail = FALSE)),
                 phi2 = c(arima2.2$coef[2],
                          sqrt(arima2.2$var.coef[2,2]),
                          arima2.2$coef[2]/sqrt(arima2.2$var.coef[2,2]),
                          2 * pnorm(abs(arima2.2$coef[2]/sqrt(arima2.2$var.coef[2,2])),
                                    lower.tail = FALSE)),
                 theta2 = c(arima2.2$coef[3],
                          sqrt(arima2.2$var.coef[3,3]),
                          arima2.2$coef[3]/sqrt(arima2.2$var.coef[3,3]),
                          2 * pnorm(abs(arima2.2$coef[3]/sqrt(arima2.2$var.coef[3,3])),
                                    lower.tail = FALSE)),
                 theta3 = c(arima2.2$coef[4],
                          sqrt(arima2.2$var.coef[4,4]),
                          arima2.2$coef[4]/sqrt(arima2.2$var.coef[4,4]),
                          2 * pnorm(abs(arima2.2$coef[4]/sqrt(arima2.2$var.coef[4,4])),
                                    lower.tail = FALSE)),
                 drift = c(arima2.2$coef[5],
                          sqrt(arima2.2$var.coef[5,5]),
                          arima2.2$coef[5]/sqrt(arima2.2$var.coef[5,5]),
                          2 * pnorm(abs(arima2.2$coef[5]/sqrt(arima2.2$var.coef[5,5])),
                                    lower.tail = FALSE)))
names(df_arima2.2)[2] <- "$\\phi_{1}$"
names(df_arima2.2)[3] <- "$\\phi_{2}$"
names(df_arima2.2)[4] <- "$\\theta_{1}$"
names(df_arima2.2)[5] <- "$\\theta_{2}$"
names(df_arima2.2)[6] <- "drift"

kable(df_arima2.2,
      digits = 4,
      caption = "ARIMA(2,1,2) \\label{tab:Tab9}",
      format = "latex",
      booktabs = T,
      escape = FALSE) %>%
  kable_styling(latex_options = "hold_position")


pred2.3 <- forecast(arima2.3, 17)
pred3.2 <- forecast(arima3.2, 17)
pred2.2 <- forecast(arima2.2, 17)

rmse2.3 <- sqrt(mean(as.vector(pred2.3$mean) - as.vector(test))^2)
rmse3.2 <- sqrt(mean(as.vector(pred3.2$mean) - as.vector(test))^2)
rmse2.2 <- sqrt(mean(as.vector(pred2.2$mean) - as.vector(test))^2)


op_wind_train1 <- train
op_wind_train2 <- train
op_wind_train3 <- train
num_iter <- 17

# Create a vector to store the forecasts
pred1 <- rep(NA, num_iter)
pred2 <- rep(NA, num_iter)
pred3 <- rep(NA, num_iter)

# Loop through the iterations
for (i in 1:num_iter) {

# Fit the model to the training data
op_wind_arima2.3 <- Arima(op_wind_train1, order = c(2, 1, 3))

op_wind_arima3.2 <- Arima(op_wind_train2,
                  order = c(3, 1, 2))

op_wind_arima2.2 <- Arima(op_wind_train3,
                        order = c(2, 1, 2))

# Forecast one value ahead
fc1 <- forecast(op_wind_arima2.3, 1)
fc2 <- forecast(op_wind_arima3.2, 1)
fc3 <- forecast(op_wind_arima2.2, 1)

# Store the forecasted value
pred1[i] <- fc1$mean
pred2[i] <- fc2$mean
pred3[i] <- fc3$mean

# Add the forecasted value to the training data
op_wind_train1 <- c(op_wind_train1, pred1[i])
op_wind_train2 <- c(op_wind_train2, pred2[i])
op_wind_train3 <- c(op_wind_train3, pred3[i])
}

rmse <- c(sqrt(mean((as.vector(pred1) - as.vector(test))^2)),
          sqrt(mean((as.vector(pred2) - as.vector(test))^2)),
          sqrt(mean((as.vector(pred3) - as.vector(test))^2)))



df <- data.frame(rmse = c(RMSE1 = rmse1,
                          RMSE2 = rmse2,
                          RMSE3 = rmse3,
                          "Open Window RMSE1" = rmse[1],
                          "Open Window RMSE2" = rmse[2],
                          "Open Window RMSE3" = rmse[3]))


kable(df,
      digits = 4,
      caption = "Root Mean Square Error of the three models ARIMA(2,1,3),
      ARIMA(3,1,2), and ARIMA(2,1,2). \\label{tab:Tab10}") %>%
  kable_styling(latex_options = "hold_position")

sarma_final <- Arima(canb_data,
                  order = c(1, 0, 1),
                  seasonal = list(order = c(0, 1, 1),
                                  period = 12),
                lambda = BC_lambda)

df_sarma_final <- data.frame(Names = c("coeff", "S.E"),
                 phi1 = c(sarma_final$coef[1],
                          sqrt(sarma_final$var.coef[1,1])),
                 theta1 = c(sarma_final$coef[2],
                          sqrt(sarma_final$var.coef[2,2])),
                 stheta1 = c(sarma_final$coef[3],
                          sqrt(sarma_final$var.coef[3,3])))

names(df_sarma_final)[2] <- "$\\phi_{1}$"
names(df_sarma_final)[3] <- "$\\theta_{1}$"
names(df_sarma_final)[4] <- "s$\\theta_{1}$"

kable(df_sarma_final,
      digits = 4,
      caption = "SARIMA(1,0,1)(0,1,1)$_{12}$ \\label{tab:Tab11}",
      format = "latex",
      booktabs = T,
      escape = FALSE) %>%
  kable_styling(latex_options = "hold_position")

arima_final <- Arima(Glob_temp, order = c(2, 1, 2))

df_arima_final <- data.frame(Names = c("coeff", "S.E"),
                 phi1 = c(arima_final$coef[1],
                          sqrt(arima_final$var.coef[1,1])),
                 phi2 = c(arima_final$coef[2],
                          sqrt(arima_final$var.coef[2,2])),
                 theta2 = c(arima_final$coef[3],
                          sqrt(arima_final$var.coef[3,3])),
                 theta3 = c(arima_final$coef[4],
                          sqrt(arima_final$var.coef[4,4])))

names(df_arima_final)[2] <- "$\\phi_{1}$"
names(df_arima_final)[3] <- "$\\phi_{2}$"
names(df_arima_final)[4] <- "$\\theta_{1}$"
names(df_arima_final)[5] <- "$\\theta_{2}$"

kable(df_arima_final,
      digits = 4,
      caption = "ARIMA(2,1,2) \\label{tab:Tab12}",
      format = "latex",
      booktabs = T,
      escape = FALSE) %>%
  kable_styling(latex_options = "hold_position")


op_wind <- canb_data
num_iter <- 10

# Create a vector to store the forecasts
pred <- rep(NA, num_iter)
lowr_int <- rep(NA, num_iter)
upper_int <- rep(NA, num_iter)

# Loop through the iterations
for (i in 1:num_iter) {

# Fit the model to the training data
```

```
op_wind_sarma <- Arima(op_wind,
                  order = c(1, 0, 1),
                  seasonal = list(order = c(0, 1, 1),
                                  period = 12),
                lambda = BC_lambda,
                include.constant = TRUE)

# Forecast one value ahead
fc <- forecast(op_wind_sarma, 1)

# Store the forecasted value
pred[i] <- fc$mean
lowr_int[i] <- fc$lower
upper_int[i] <- fc$upper


# Add the forecasted value to the training data
op_wind <- c(op_wind, pred[i])
}

# For Global Temperature (GT)

op_wind_GT <- Glob_temp

# Create a vector to store the forecasts
pred_GT <- rep(NA, num_iter)
lowr_int_GT <- rep(NA, num_iter)
upper_int_GT <- rep(NA, num_iter)

# Loop through the iterations
for (i in 1:num_iter) {

# Fit the model to the training data

op_wind_arima <- Arima(op_wind_GT,
                  order = c(2, 1, 2),
                  include.constant = TRUE)

# Forecast one value ahead
fc_GT <- forecast(op_wind_arima, 1)

# Store the forecasted value
pred_GT[i] <- fc_GT$mean
lowr_int_GT[i] <- fc_GT$lower
upper_int_GT[i] <- fc_GT$upper


# Add the forecasted value to the training data
op_wind_GT <- c(op_wind_GT, pred_GT[i])
}


months <- seq(1,length(canb_data), 1)
```

```r
fc_months <- months[176]+1:length(pred)
plot(months, canb_data, type = 'o',
     xlim = c(months[1],
              fc_months[10]),
     ylim = c(min(lowr_int),
              max(upper_int)),
     main = "(a)",
     ylab = "Canberra Temperature")
lines(fc_months, pred, col = "red", type = 'o')
lines(fc_months, upper_int, col = "blue", lty = "dashed")
lines(fc_months, lowr_int, col = "blue", lty = "dashed")

years <- seq(1,length(Glob_temp), 1)
fc_years <- years[173]+1:length(pred_GT)
plot(years, Glob_temp, type = 'o',
     xlim = c(years[1],
              fc_years[10]),
     ylim = c(min(Glob_temp),
              max(upper_int_GT)),
     main = "(b)",
     ylab = "Global Temperature")
lines(fc_years, pred_GT, col = "red", type = 'o')
lines(fc_years, upper_int_GT, col = "blue", lty = "dashed")
lines(fc_years, lowr_int_GT, col = "blue", lty = "dashed")


par(mfrow = c(1, 2))
acf(tranf_data, 30, ci.type = "ma", main = "(a)")
pacf(tranf_data, 30, main = "(b)")

par(mfrow = c(1, 2))
acf(Glob_temp, 30, ci.type = "ma", main = "c")
pacf(Glob_temp, 30, main = "d")

par(mfrow = c(1, 2))
acf(sarma1$residuals, 24, ci.type = "ma", main = "(a)")
pacf(sarma1$residuals, 24, main= "(b)")

par(mfrow = c(1, 2))
acf(sarma2$residuals, 24, ci.type = "ma", main = "(c)")
pacf(sarma2$residuals, 24, main = "(d)")

par(mfrow = c(1, 2))
acf(sarma3$residuals, 24, ci.type = "ma", main = "(e)")
pacf(sarma3$residuals, 24, main = "(f)")


# plot(qqnorm(sarma1$residuals));qqline(sarma1$residuals)
#
# plot(qqnorm(sarma2$residuals));qqline(sarma2$residuals)


par(mfrow = c(1, 1))
```

```
qqnorm(sarma1$residuals, main = "(a)")
qqline(sarma1$residuals)

par(mfrow = c(1, 1))
qqnorm(sarma2$residuals, main = "(b)")
qqline(sarma2$residuals)

par(mfrow = c(1, 1))
qqnorm(sarma3$residuals, main = "(c)")
qqline(sarma3$residuals)

par(mfrow = c(1, 2))
acf(arima2.3$residuals, 24, ci.type = "ma", main = "ARIMA(2,1,3)")
pacf(arima2.3$residuals, 24, main = "ARIMA(2,1,3)")

par(mfrow = c(1, 2))
acf(arima2.2$residuals, 24, ci.type = "ma", main = "ARIMA(2,1,2)")
pacf(arima2.2$residuals, 24, main = "ARIMA(2,1,2)")

par(mfrow = c(1, 2))
acf(arima3.2$residuals, 24, ci.type = "ma", main = "ARIMA(3,1,2)")
pacf(arima3.2$residuals, 24, main = "ARIMA(3,1,2)")


par(mfrow = c(1, 1))
qqnorm(arima2.3$residuals, main = "(a)")
qqline(arima2.3$residuals)

par(mfrow = c(1, 1))
qqnorm(arima2.2$residuals, main = "(b)")
qqline(arima2.2$residuals)

par(mfrow = c(1, 1))
qqnorm(arima3.2$residuals, main = "(c)")
qqline(arima3.2$residuals)
```

REFERENCE

1. Australian Bureau of Meteorology. Climate Data Online. (Accessed 6 May 2023).

2. Roser, M. Our world in data.