Time Series: Investigating Temperature data in Melbourne, Australia during the years 1981-1990.

Ryan Pollard

March 2021

Contents

1 Aim

This project aims to investigate a given piece time series data and investigating methods through which the a statistical model can be built that represents how the data moves through time - eventually forecasting that data for future dates.

2 The Data

A time series data set consisting of daily maximum temperatures (degrees C) in Melbourne will be used in this analysis. The data set covers a period of 1 January 1981 to 31 December 1990 and the data will be split into two datasets, the train dataset will contain the dates 01/01/1981 - 31/01/1988 and and the test dataset which will contain the dates 01/01/1989 to 31/12/1990.

The train data will be used the build the statistical model. The train dataset will be used to compare how accurate our models forecasts into the 2 years after our train data, 1989 and 1990

3 Data Preparation

The data gives daily temperatures values for the dates discussed. However, daily data produces graphs are are noisy and difficult to interpret visually. Monthly data is also easier to work with in regards to data cleaning and model building. For that reason the data is converted to monthly data, taking the average temperature of each month.

There is no missing data in the dataset.

As discussed before, the data is also split into a test and train dataset, allowing for testing of the models' accuracy.

Figure 1 shows the monthly temperature data through the years with a 12 month moving average line.

It shows datra that is seasonal, as expected, in January we have high values for temperature and mid-year we have low values for temperature, summer months and winter respectively in Melbourne, Australia.

The orange bar shows that there are temperature trends across the decade. From 1981 to 1984 there is a drop in average temperature, seeing as a decreasing orange bar up until 1984. It then increases and decreases some more through the years; seemingly there is an underlying temperature trend in this data.

3.1 Decomposition

To understand the data better the data is decomposed. This breaks up the time-series into three underlying variables - trend, seasonality and residue. Using this method it's possible to see how tremperature has changed through the years easily without the effect of seasonality.

The decomposition plot in Figure 2 is broken down into 4 components:

1. **Original Data**: This is just the raw data plotted through time as we saw in Figure 2.
2. **Seasonal**: This shows the pattern of the 12 month seasonality.
3. **Trend**: Shows how temperature changes through time taking out the effect of seasonality.
4. **Remainder**: Taking out the seasonal and trend parts, this gives the remainder and often can be attributed to be noise.

Figure 2 shows the seasonal aspect as we expected, high temperatures in summer and low temperatues in winter and follows a 12-month pattern. We can now see there are trends in the data when we look at the 3rd trend chart, from 1982 to 1987 temperature was on the way down and from 1987 onwards temperature picked back up.

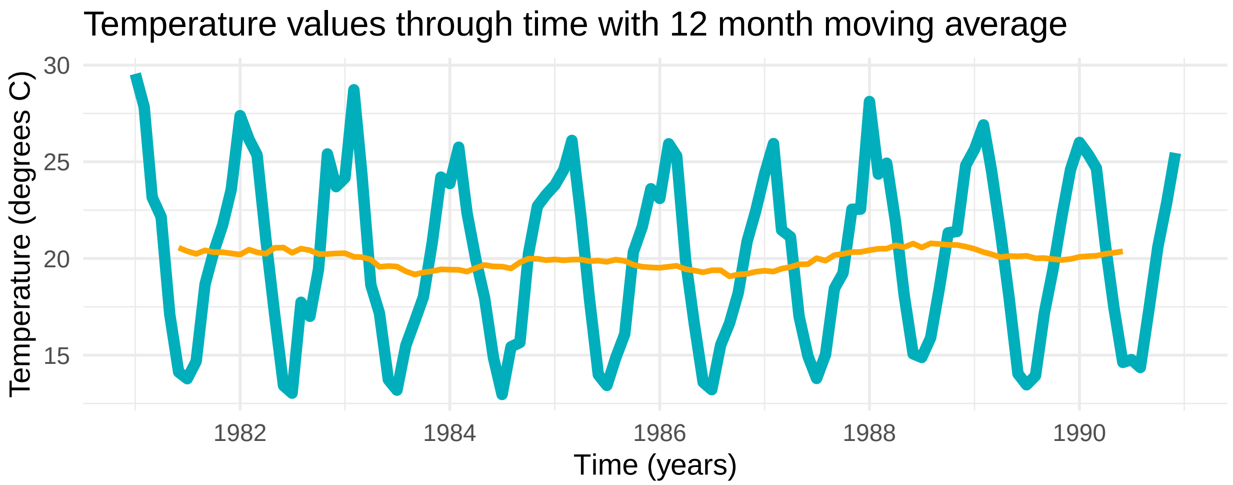


Figure 1: Monthly temperature values through time (blue line) with 12 month monthly average line (orange line)

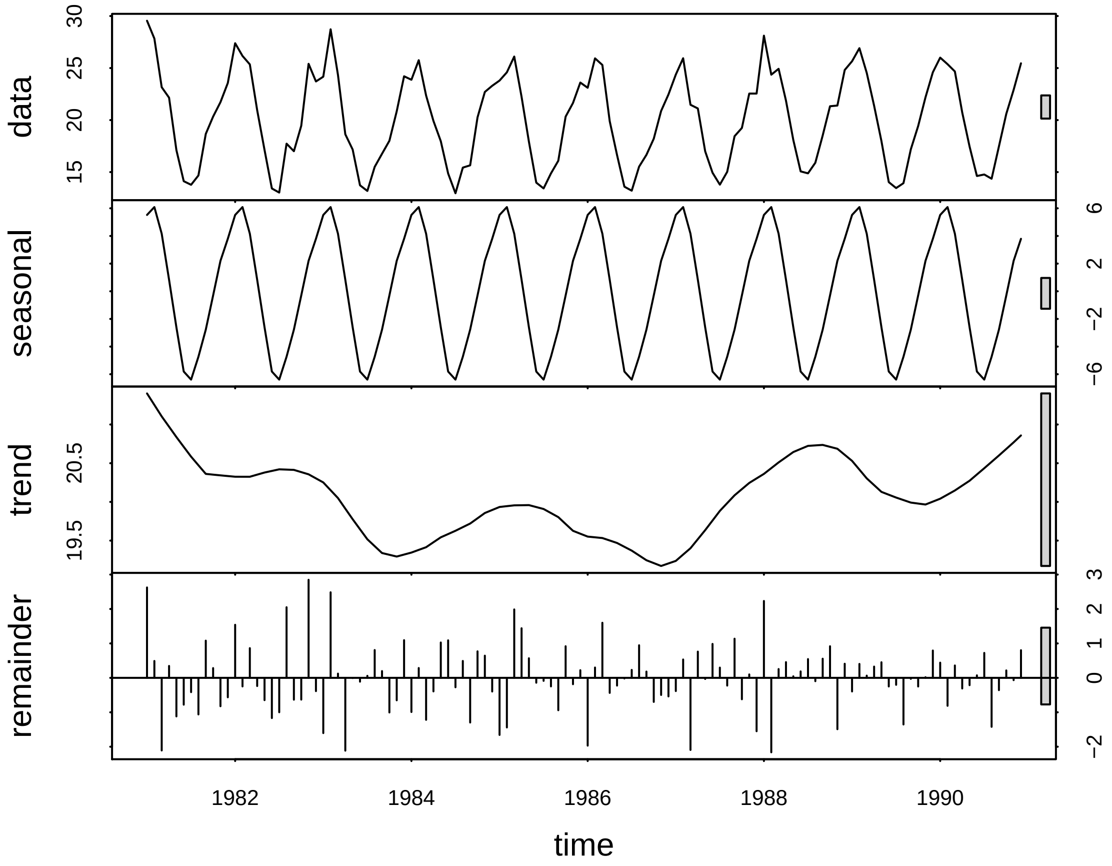


Figure 2: Decomposition of the time-series data

3.2 Anomaly Detection

The presence of anomalies is looked at in order to understand the data better and investigate if there are points in the data that could have a big influence on the model build and thus have an impact on the forecast. For example, with temperature, perhaps there was a year where the temperature was very hot or cold for a particular month but this was a freak occurrence and doesn't represent how normal temperature records behave.

In order to do anomaly detection the time-series is decomposed as in Figure 2 and there is an upper and lower threshold created; the data points which are outside of these thresholds are then identified as anomalies.

Figure 3 shows the time-series data plotted through time and one anomly plot is highlighted in Janurary 1981. This had a particularly high value for January 1981.

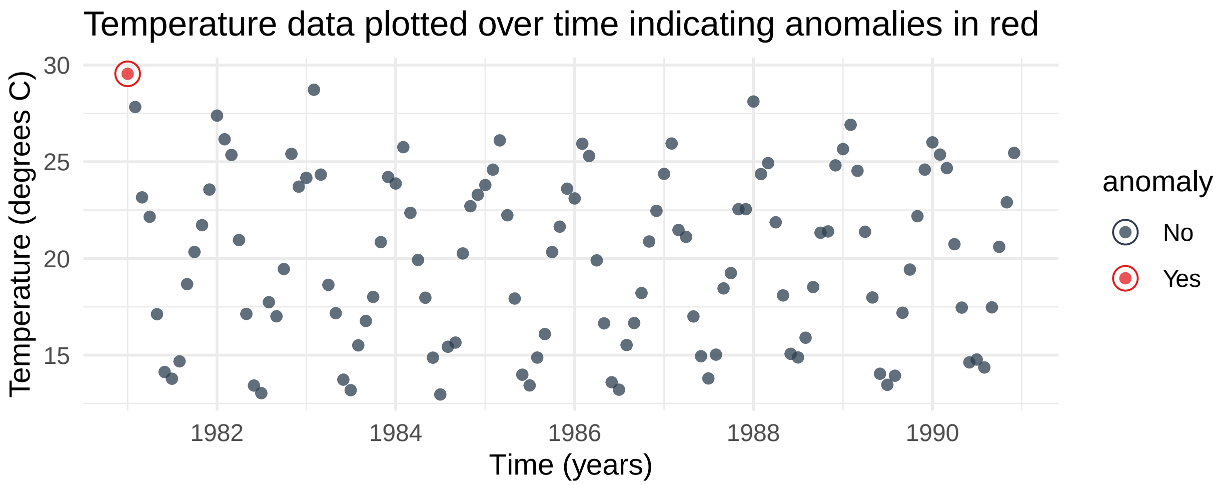


Figure 3: Time-series data plotted through time with anomalies highlighted in red.

Figure 4 shows the time-series data decomposed with the same anomaly highlighted.

Only one anomaly is detected in the data and its value isn't so high that it's a cause for a concern, seemingly it is part of a trend that in 1981 there were higher temperatures in Melbourne.

3.3 Stationarity

It's important for time-series data to be stationary. This means the time series doesn't change over time - it has a constant mean and a constant variance. This is important in order for the statistical model to represent the time-series as best as possible.

First we deal with seasonality. We confirm that there is a 12 month repeating trend to the seasons using a lag plot. If there is no relationship between the data and the data from the past, the lag plot will show a random pattern.

Figure 5 shows a strong positive correlation line for lag 12 indicating that there is a positive correlation between the data and the data 12 months ago - indicating yearly seasonality.

The seasonality is taken out of the time-series by taking away the temperatue values of 12-months prior from the temperature values. After doing this, an Augmented Dickey Fuller test is used to see if the data is stationary.If the data is stationary the p-value will be below 0.05.

The result of the test gives a p-value of 0.2 (2.dp) indicating the deseasonalised data is not stationary.

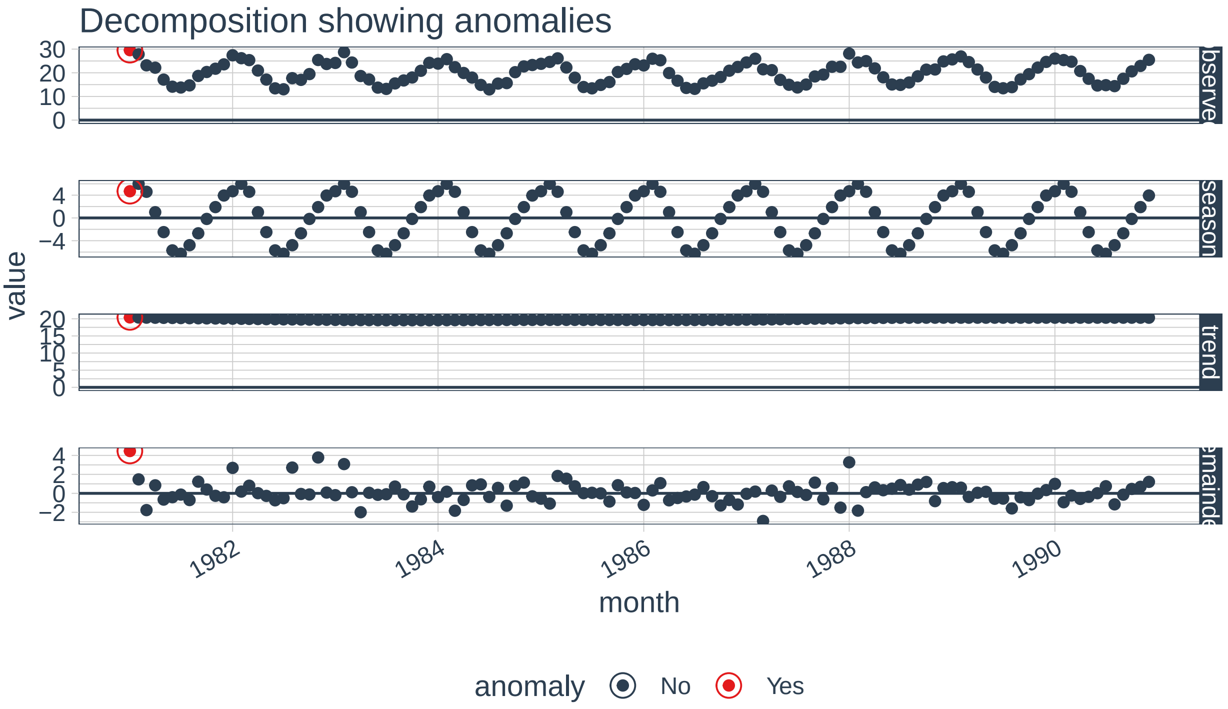


Figure 4: Decomposition of time-series data showing the anomalies

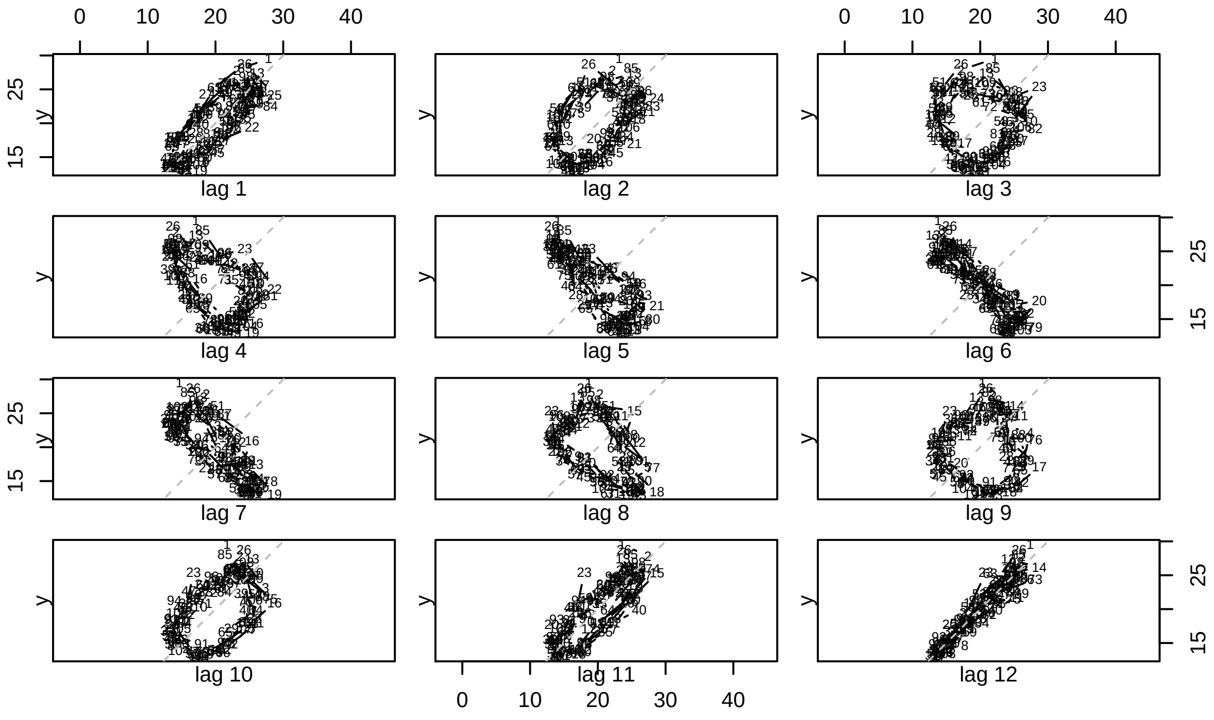


Figure 5: Lag plot of time-series data showing existence of seasonality as seen with strong poisitive correlation line in lag = 12.

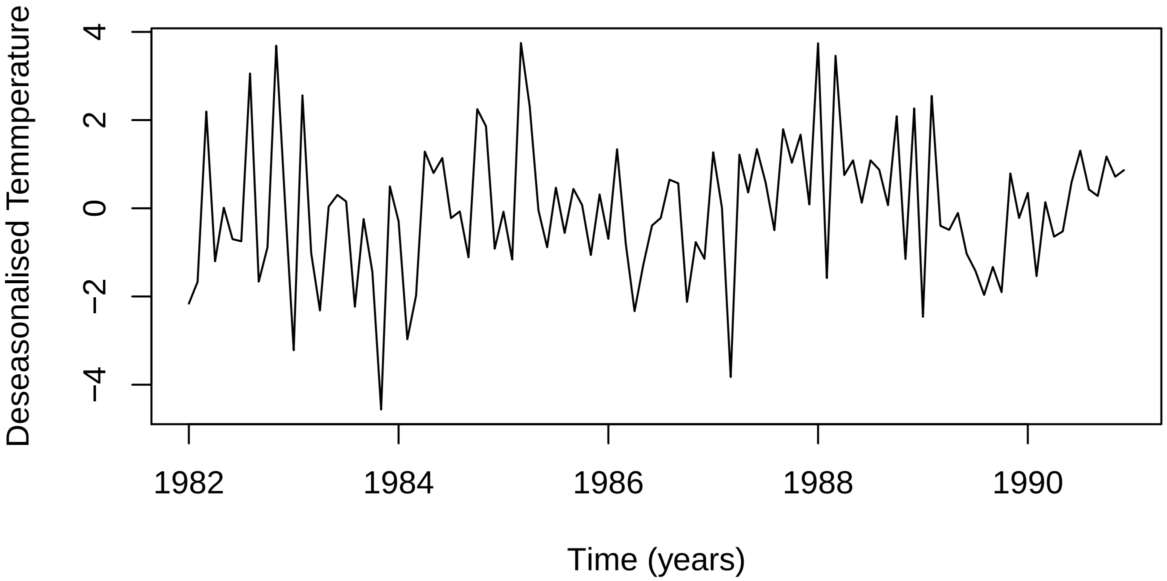


Figure 6: Deseasonalised time series

Figure 6 shows the deasonalised data. It does not look completely random as there are up and down trends seen through the years and doesn't stay around temperature value 0 very often.

This trend is taken out by differencing the data after taking away the seasonality - taking away the the previous month value from each of the temperature values.

The ADF test now gives a p-value of 0.01 and Figure 7 shows the deasonalised and differenced data, showing a white noise pattern and a costant mean and variable - this data is now ready to do time-series analysis.

This type of differencing is called first differencing. If this didn't work we would continue to take the 2nd, 3rd, 4th etc difference until the data is stationary.

3.4 Which model is used - the Box Jenkins method.

Now the data is ready to begin a model build it's necessary to investigate what type of model best suits the data. To do this we use the Box Jenkins method.

Looking at the ACF and the PACF of the differenced and deseasonalised data will determine what type of model to be used.

For an AR model the PACF must cut off after lag p with not many lags deemed significant. The ACF must exponentially decay. From Figure 9 this does not happen - the PACF shows many significant lags. For an MA model the ACF must cut off after lag q with not many significant values and the PACF must show an exponential decay. From Figure 8 both of these criteria are not met. For an ARIMA model, the PACF and ACF must tail off and complex behaviors of the lags are allowed . This does happen in both our PACF and ACF.

The PACF and ACF both decay which is necessary to build a model.

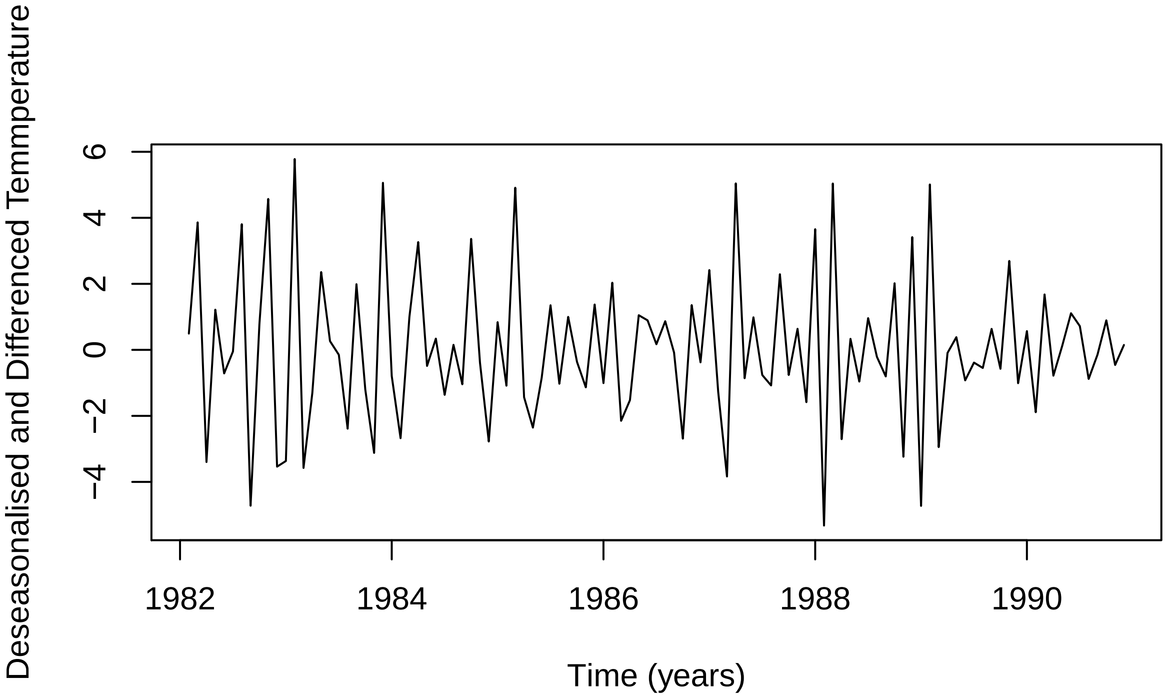


Figure 7: Deseasonalised and differenced time-series showing a pattern similar to white noise

A Seasonal ARIMA model is chosen on this basis, and the fact we do not have native stationary data. However, choosing the order of the ARIMA is difficult using the ACF and PACF plots.

There are 4 significant values for the ACF, and the PACF has 8 significant separate points - but 3 of these points at the start seem to be related to the first point so we consider the PACF to have 5 significant points For this reason we may choose p and q to be 5 and 4.

Selection of the best model is not an exact science and in this project different models will be tried and tested against eachother - and parsimonious models will be tended towards.

Since the ACF shows a big spike in the first lag and a decay after, perhaps q = 1 would be a good selection. A similar argument can be made for the PACF and perhaps p = 1 would be a good selection. Spikes at after year 1 are also seen in both plots so a SARIMA(1,1,1)(1,1,1)12 will be chosen.

4 Analysis

For the built models the train dataset is used. The model is used to forecast and compare against the train dataset and accuracy measures are compared to determine what is the best model to forecast.

First a base-line model is built - a simplistic model from which the other models can be compared. This is a seasonal naive model which effectively takes the temperature value from 12 months ago and adds on an error value as a means of prediction and model fit.

4.1 Seasonal Naive Model

Figure 10 shows model diagnostics for the seasonal naive method. Th top plot shows the residuals and a good model shows a white-noise process here. It does look quite random but perhaps some trends can be

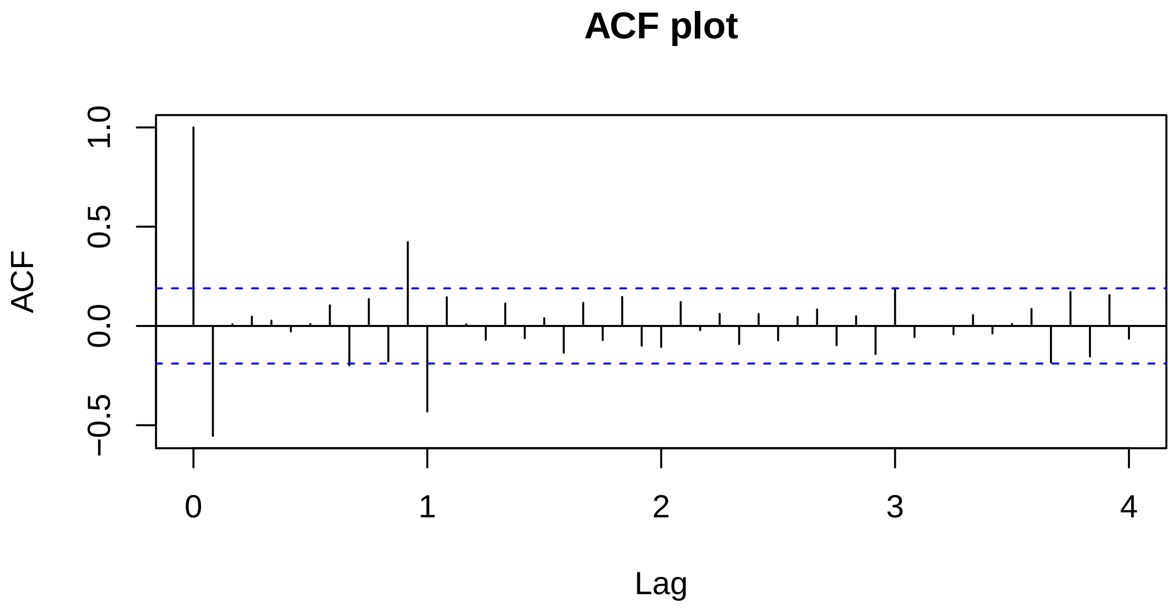


Figure 8: ACF plot

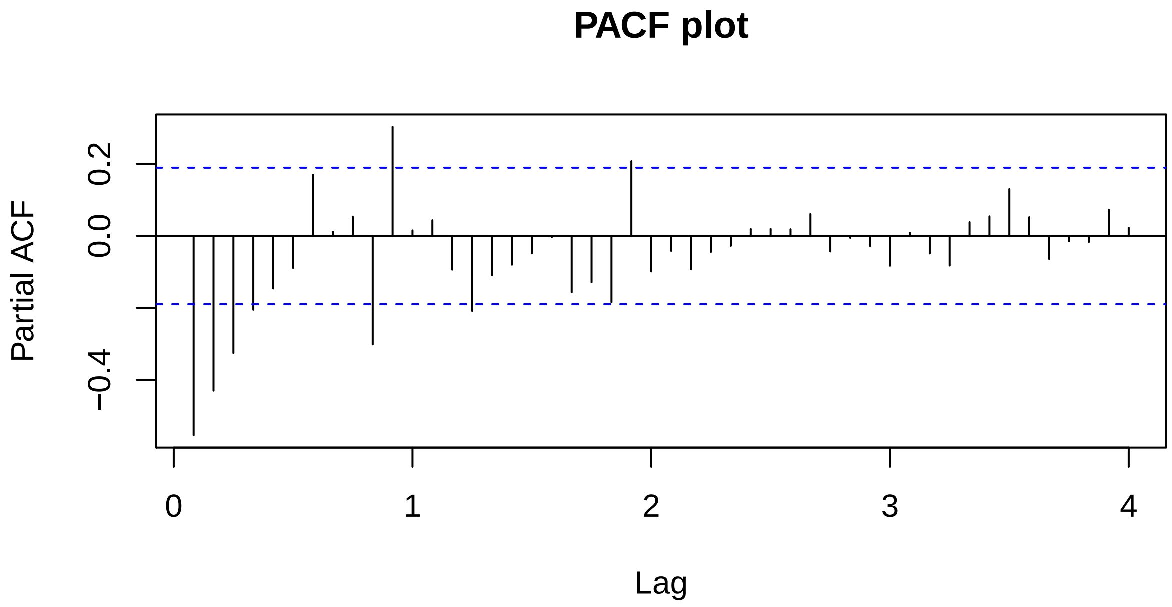


Figure 9: PACF plot

spotted from this plot.

The ACF shows some significant plots which suggests there is a better model out there that can explain some of the trends. The residuals look normally distributed.

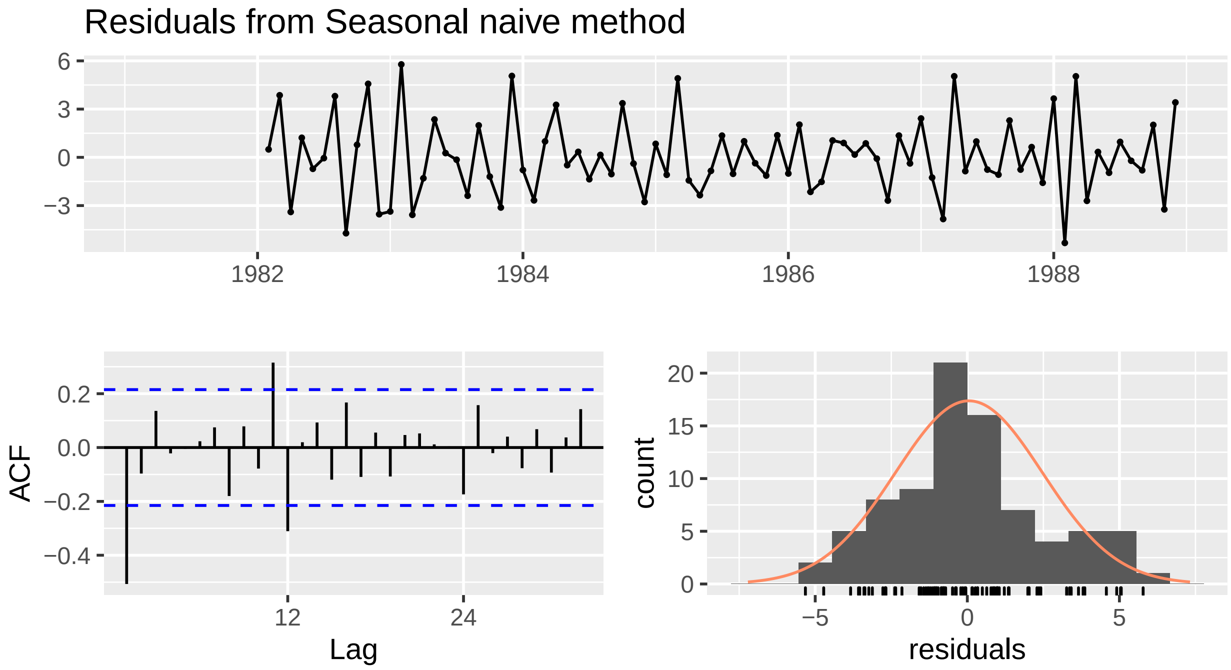


Figure 10: Seasonal Naive Model Diagnostics

4.2 Exponential Smoothing Model

An exponential smoothing model is built on the train data. This model assigns exponentially decreasing weights to past data as the data gets older. This allows it to pick up on underlying trends of the deseasonalised data. The output for this can be seen in the appendices.

Figure 11 shows model diagnostics. This model looks better as we have no signfiicant ACF spikes and the residuals look random in the top plot.

4.3 SARIMA Model

To begin s simple ARIMA will be used model with the order derived from the PACF and ACF plots. A SARIMA(1,1,1)(1,1,1)12 model will be used which ensures a simple model but represents the ACF and PACF plots.

In figure 11 the residuals demonstrate perhaps random behaviour, it is not as clear as the exponential smoothing model. No spikes are significant in the ACF plot and the residuals are normally distributed - all indicating a good model fit.

4.4 Auto SARIMA Model

R has a function which tests many different orders of the SARIMA model and using model diagnostics compares these models to find the best one. Since it was deemed difficult to choose the best SARIMA using

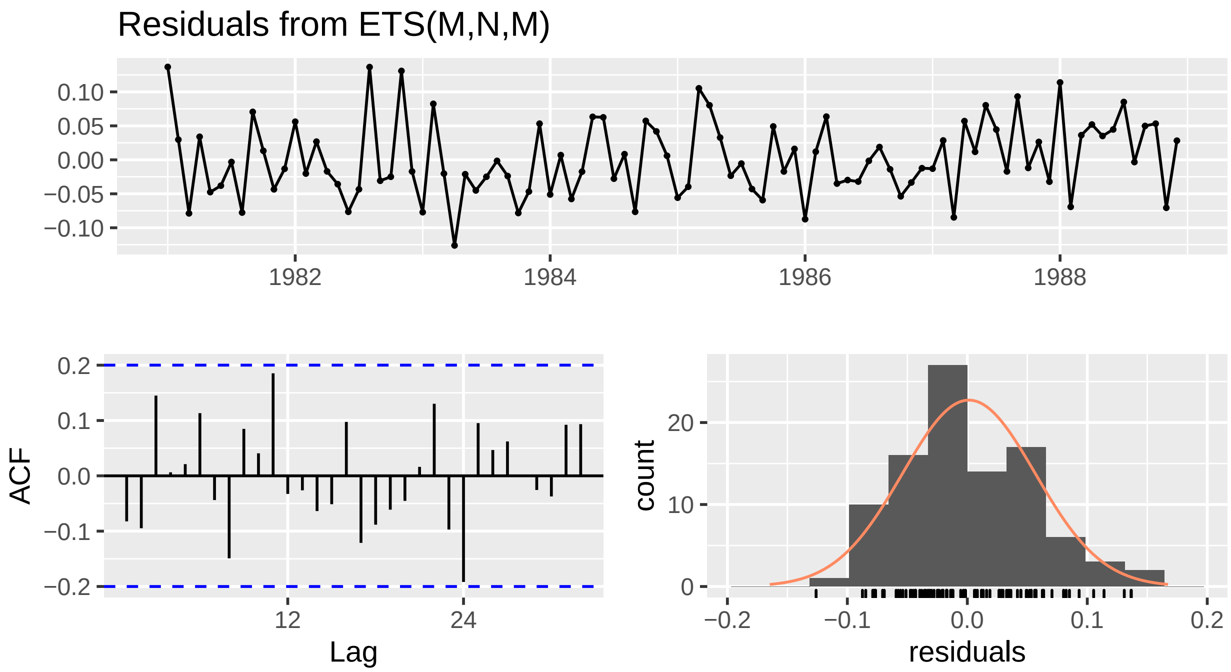


Figure 11: Exponential Smoothing Model Diagnostics

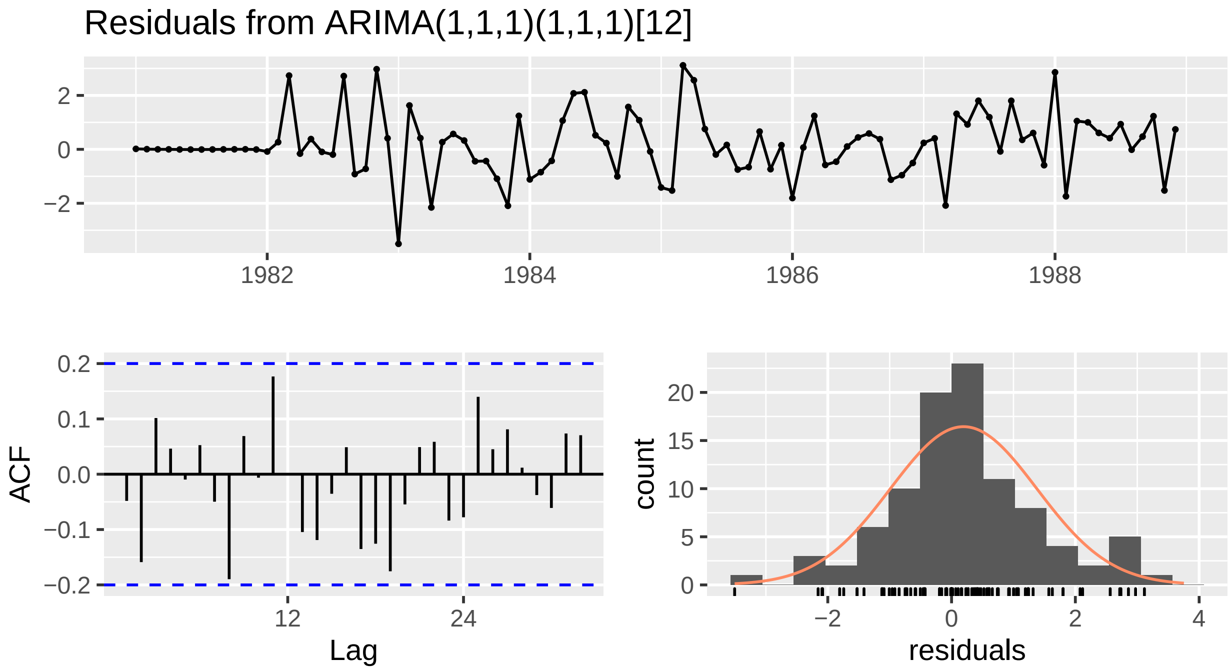


Figure 12: SARIMA(1,1,1)(1,1,1)12 Model Diagnostics

the Box-Jenkins method using the ACF and PACF, using an algorithm that selected the best SARIMA model could be very useful here and therefore this function is ran.

After running this function, R claims that the SARIMA(2,1,1)(2,1,0)12 model is the best model.

In figure 13 the residuals demonstrate perhaps random. No spikes are significant in the ACF plot and the residuals are normally distributed - indiciating a good model fit.

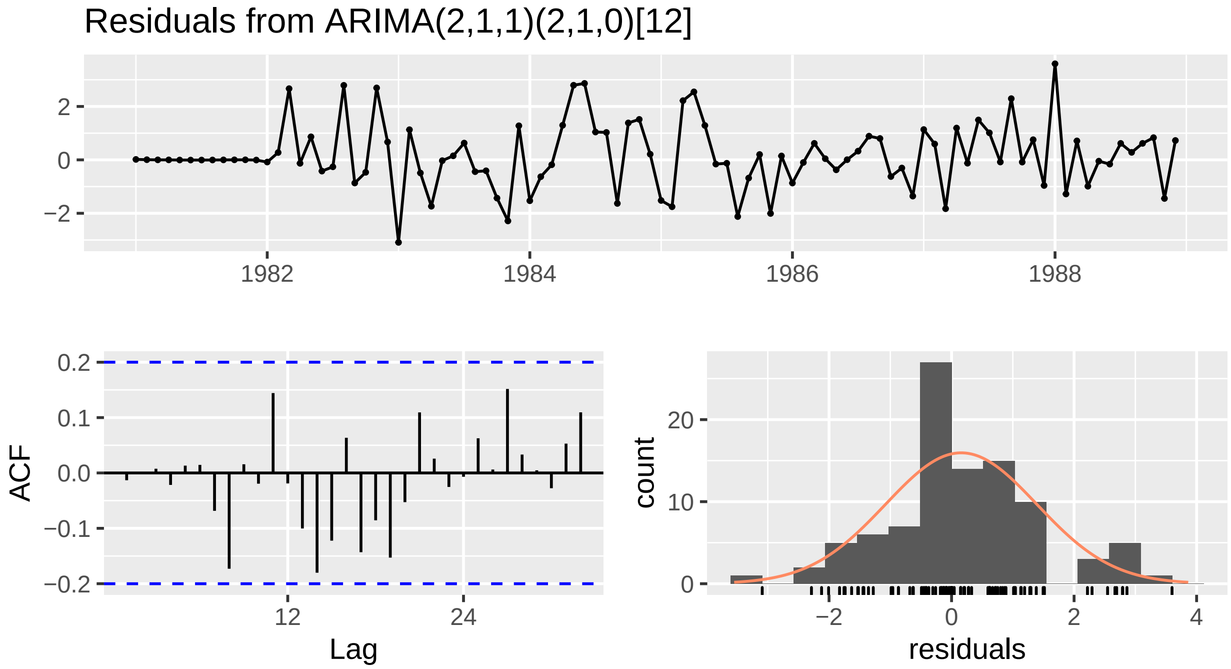


Figure 13: SARIMA(2,1,1)(2,1,0)12 Model Diagnostics

4.5 Forecast

The models are used to forecast 2 years into the future and compared with the test data. From this we can get various metrics which represent the accuracy of that prediction compares to the real test data. These values are given in Figure 14. Red colours are associated with poor performance and dark green are the best performance.

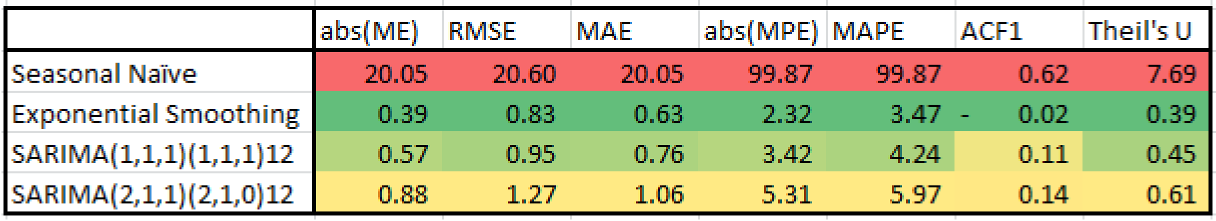
Focussing on MAPE value, this is the Mean absolute percentage error. It takes the test values and the fitted values and calculates the absolute difference between the two as a percentage of the test value and calculates the mean. The lower the value the better.

The exponential smoothing model comes out as the best performing model for this metric, but also all the other metrics. The output for this model is given in the Appendix. This model be used for forecasting. It has a sigma value of 0.06 (2 d.p) meaning the model misses the actual temperature value by 0.06 degrees C on average - meaning it is very accurate.

Surprisingly, our more parsimonious SARIMA model outperforms the automatically selected SARIMA model on the forecast data.

Using the exponential smoothing model to forecast in the future, it's possible to see how the model predicts the future values, that is from 1989 to the end of 1990, compared to real data of these years.

Figure 15 shows the time series data (black line) but the black line, from 1989 onwards, is the forecasted data from the exponential smoothing model. The blue areas the 80% and 95% confidence intervals of these



	abs(ME)	RMSE	MAE	abs(MPE)	MAPE	ACF1	Theil's U
Seasonal Naïve	20.05	20.60	20.05	99.87	99.87	0.62	7.69
Exponential Smoothing	0.39	0.83	0.63	2.32	3.47	- 0.02	0.39
SARIMA(1,1,1)(1,1,1)12	0.57	0.95	0.76	3.42	4.24	0.11	0.45
SARIMA(2,1,1)(2,1,0)12	0.88	1.27	1.06	5.31	5.97	0.14	0.61

Figure 14: Table of performance accuracy metrics of the models compared to the test data

forecasts. The red line is real test data of 1989-1991. The red line of real values follows the black line forecasted line relatively closely, with some slight differences.

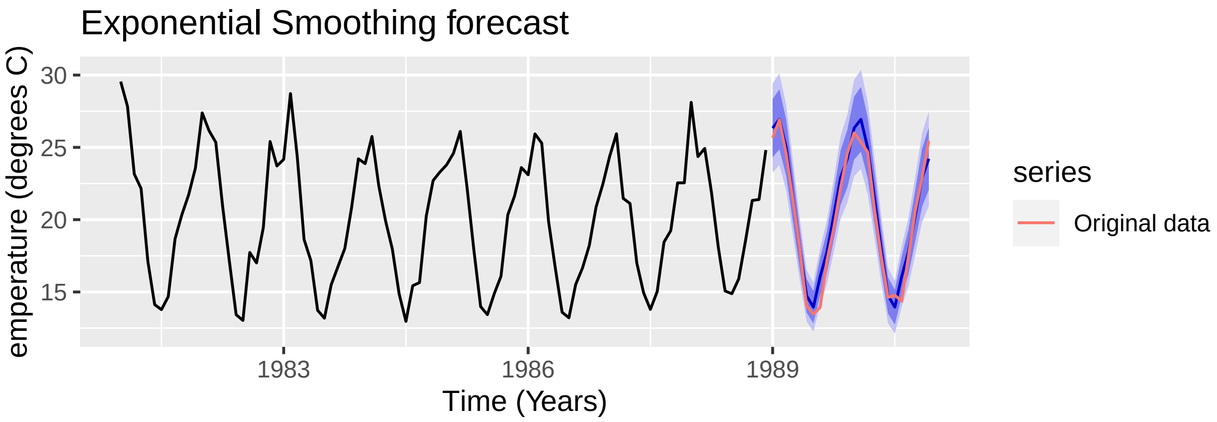


Figure 15: Exponential Smooting Model Forecast. From 1989 the black line represents forecasted data, the red line represents real data.

For comparison sake, Figure 16 shows how the forecast for the Seasonal Naive model (this shows differenced data of order 1). The red line follows the black line, but not as well as the exponential but certainly the seasonal pattern is captured - so the forecast of the seasonal naive model visually looks quite good.

This poses the question of, does our selected model capture the trend? Seemingly, the seasonal nature of a time series is relatively easy to forecast since it's something we know happens each year, but underlying trends are not.

Previously it was seen that a time series could be broken up into different parts - one part being the trend. This can be used on the forecasted data to see how the trend compares with the trend of the original time series.

Figure 17 shows the forecasted trend data for the exponential smoothing model - namely the red line from 1988 (although the test data is from 1989 onwards, trends are calculates using data before this date). The black line is the real data trend.

The red line forecasted trend shows a dip, just like the real black line data shows, which is a good sign but doesn't predict the magnitude of the dip.

The exponential smoothing method is seemingly picking up slightly on the cyclical nature of temperature fluctuations that happen on a 2-4 year basis. The temperature seems to go up a certain amount but comes back down again. This is what is observed from the data given, at least.

From the metrics the second best model is the SARIMA(1,1,1)(1,1,1)12 and for comparison sake the forecasted trend of this model is plotted in Figure 18. It shows that the ARIMA model assumes the trend will continue as a straight line and doesn't give a dip in temperatures like the exponential does. There is some slight dip there, but nowhere near the magnitude of the dip seen in the exponential model.

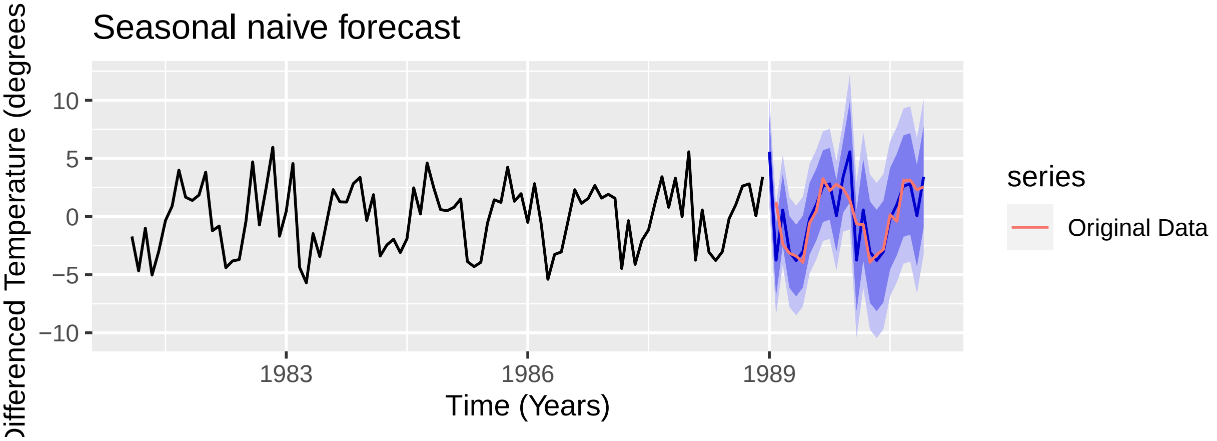


Figure 16: Seasonal Naive Model Forecast. From 1989 the black line represents forecasted data, the red line represents real data.

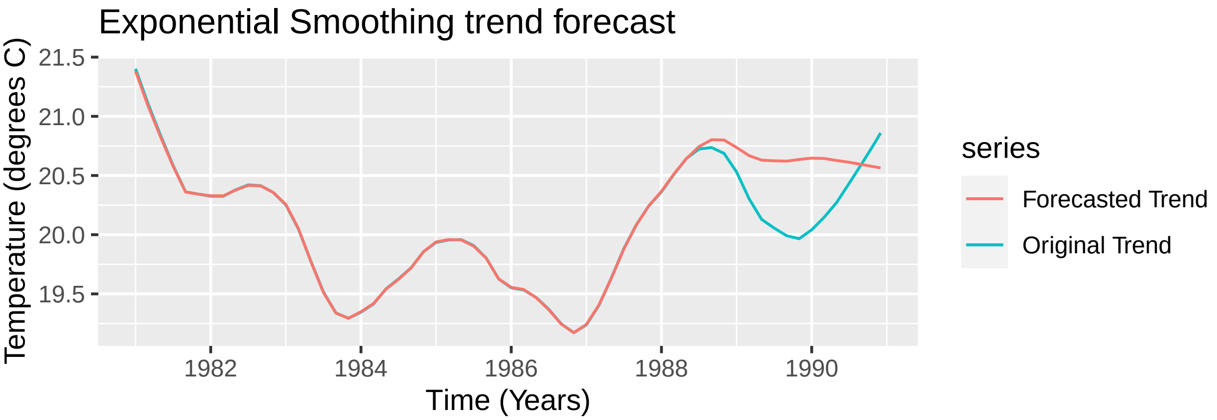


Figure 17: Exponential Smoothing Model Forecast of underlying trend, compared with real data trend. The red line starting from mid-1988 is the forecasted trend of the model.

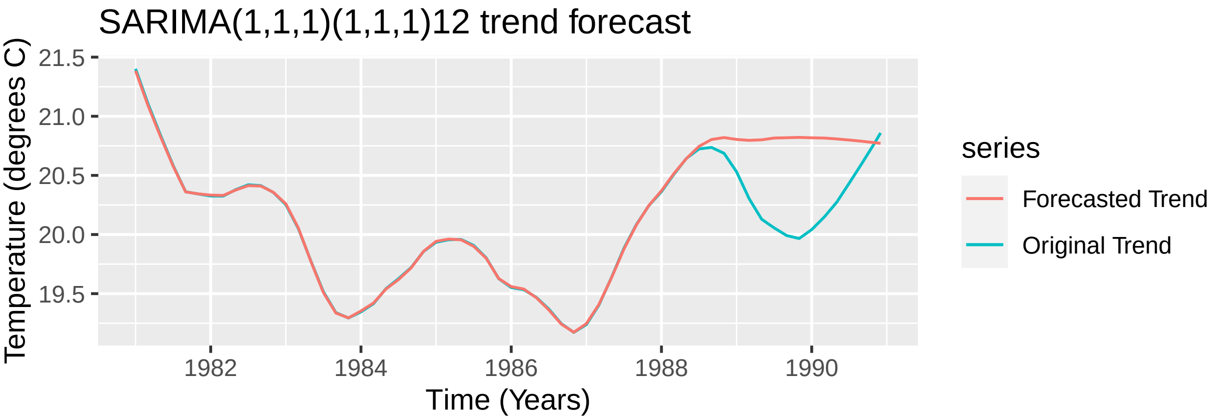


Figure 18: SARIMA(1,1,1)(1,1,1)12 Model Forecast of underlying trend, compared with real data trend. The red line starting from mid-1988 is the forecasted trend of the model.

5 Conclusion

Selecting an model that fits a time-series data isn't an exact science and choosing different models and testing and comparing their accuracy is the best approach to take. The autoarima function, which selected the best ARIMA model didn't fare as well when used on the test data as an ARIMA chosen using the Box-Jenkins method. Perhaps this was due to overfitting of the data and the fact that the model only had 7 years to model with (1981-1988). If this analysis were to be carried out again, more data would be required to get more accurate models. With the data we have, the Exponential Smoothing Model produces the best results and even detects a dip in the temperature trend which the other models fail to.

6 Appendix

6.1 Exponential Smoothing Model output

```
## ETS(M,N,M)
##
## Call:
##  ets(y = ytrain)
##
##   Smoothing parameters:
##     alpha = 0.1204
##     gamma = 1e-04
##
##   Initial states:
##     l = 20.3657
##     s = 1.1736 1.1101 0.9827 0.8612 0.7785 0.676
##            0.7148 0.8697 1.0412 1.2101 1.3057 1.2764
##
##   sigma:  0.0596
##
##      AIC     AICc      BIC
## 481.9197 487.9197 520.3849
##
## Training set error measures:
##                      ME     RMSE       MAE        MPE     MAPE      MASE
## Training set 0.02594933 1.164287 0.9222917 -0.1670888 4.498579 0.7237759
##                    ACF1
## Training set -0.1473697
##                      ME     RMSE       MAE        MPE     MAPE      MASE
## Training set 0.02594933 1.164287 0.9222917 -0.1670888 4.498579 0.7237759
##                    ACF1
## Training set -0.1473697
```

7 Citations

- Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

- Yihui Xie (2020). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.30.

- Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963

- Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595

- Davis Vaughan and Matt Dancho (2020). tibbletime: Time Aware Tibbles. R package version 0.1.6. https://CRAN.R-project.org/package=tibbletime

- Matt Dancho and Davis Vaughan (2020). anomalize: Tidy Anomaly Detection. R package version 0.2.2. https://CRAN.R-project.org/package=anomalize

- Matt Dancho and Davis Vaughan (2021). timetk: A Tool Kit for Working with Time Series in R. R package version 2.6.1. https://CRAN.R-project.org/package=timetk

- Hyndman R, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2021). *forecast: Forecasting functions for time series and linear models*. R package version 8.14, <URL: https://pkg.robjhyndman.com/forecast/>. Hyndman RJ, Khandakar Y (2008). "Automatic time series forecasting: the forecast package for R." *Journal of Statistical Software*, *26*(3), 1-22. <URL: https://www.jstatsoft.org/article/view/v027i03>.

- Adrian Trapletti and Kurt Hornik (2020). tseries: Time Series Analysis and Computational Finance. R package version 0.10-48.

- Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2020). dplyr: A Grammar of Data Manipulation. R package version 1.0.2. https://CRAN.R-project.org/package=dplyr

- Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL http://www.jstatsoft.org/v40/i03/.

- Rami Krispin (2020). TSstudio: Functions for Time Series Analysis and Forecasting. R package version 0.1.6. https://CRAN.R-project.org/package=TSstudio

- Achim Zeileis and Gabor Grothendieck (2005). zoo: S3 Infrastructure for Regular and Irregular Time Series. Journal of Statistical Software, 14(6), 1-27. doi:10.18637/jss.v014.i06

- H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

- Achim Zeileis, Torsten Hothorn (2002). Diagnostic Checking in Regression Relationships. R News 2(3), 7-10. URL https://CRAN.R-project.org/doc/Rnews/