Compare the Forecast Accuracy of Time Series Models Using Temperature Data

Zijun Ma – T00711782
Shaomeng Yin – T00708655

Abstract: This project aims to compare the forecast accuracy of multiple time series models when predicting future temperature. After detecting and replacing extreme outliers, three benchmark models and seven advanced models are built on training data, and assumptions and residuals are checked. When forecasting test data, root mean squared error (RMSE) is utilized as the main evaluation metric in comparing the forecast accuracy of different models. The result indicates that The standard regression model outperforms others with the lowest RMSE.

Keywords—time series model, temperature data, accuracy

Ⅰ. Introduction

In predictive analytics, time series forecasting plays a pivotal role in unraveling patterns and trends inherent in sequential data. The goal of this project is to delve into the intricate task of comparing the forecast accuracy of various time series models, using temperature data as the primary dataset. The nature of temperature data lies in its dynamic,multi-factor influenced, and time-dependent characteristics, making it an ideal candidate for evaluating the efficacy of forecasting models.

This project employs multiple time series forecast methods, including 3 benchmark models and 7 advanced models, to forecast future temperatures. RMSE is used as the main metric to evaluate the forecast accuracy of different models.

Ⅱ. Data

The dataset used in the project is “daily climate time series data” from Kaggle, which includes 1575 daily climate records in the city of Delhi from 2013 to 2017, with 5 variables and no missing values. As shown in Table 1, the predictive variables are “humidity”, “wind speed” and “mean pressure”. The outcome is “Mean temperature” [6].

Table 1: Variables used in the analysis

	Variable	Type	Description

1	Date	Continuous	Date of format YYYY-MM-DD
2	Humidity	Continuous	Humidity value for the day (units are grams of water vapor per cubic meter volume of air)
3	Wind speed	Continuous	Wind speed measured in km/ph
4	Mean pressure	Continuous	Pressure reading of weather (measure in atm)
5	Mean temperature	Continuous	Mean temperature averaged out from multiple 3 hours intervals in a day

The analysis starts with Exploratory Data Analysis (EDA), encompassing the detection and handling of missing data and outliers. Then, box plots and scatter plots are applied to visualize each variable. Observations from Figure 1 and 2 suggest potential outliers in wind speed and mean pressure. Using the tsoutliers() function in R, we addressed outliers in these two variables. This function detects possible outliers within time series data and provides details about their positions and potential replacement values. Following the outlier treatment, we re-plotted the scatter plots and box plots. As seen in Figure 3 and 4, the outliers no longer existed. Next, we transformed the data into a “tsibble” table and split it into training and testing datasets.

Ⅲ. Method

A. Three benchmark models

This project initially utilizes three basic approaches: Mean, Naïve, and Drift. Under the Mean method, the predictions for all forthcoming values are set to the average of the historical data. In the case of the Naïve method, all future forecasts are aligned with the value of the latest observation. The Drift method, an extension of the Naïve method that considers the first and last observation, is formulated as follows

$$\hat{y}_{T+h|T} = y_T + h\left(\frac{y_T - y_1}{T-1}\right)$$

B. ARIMA(p,d,q) model

The ARIMA model aims to present the autocorrelations in the data, which combines differencing(d), autoregressive model(AR(p)), and moving average model(MA(q)). The full model is

$$(1 - \phi_1 B - \ldots - \phi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + \ldots + \theta_q B^q)\varepsilon_t$$

The AR(p) model takes lagged observations as inputs, while the MA(q) model takes lagged errors as inputs. appropriate values for p and q can be determined from the ACF and

PACF plot. Assumptions of the ARIMA model are stationarity, no trend or seasonal pattern, and residuals with a pattern resembling white noise [1].

C. Exponentially Weighted Moving Average (EWMA) model

The EWMA model is also called Simple Exponential Smoothing. It is a time series forecasting and smoothing technique that assigns exponentially decreasing weights to past observations. This model is particularly useful for capturing trends and patterns in data while giving more importance to recent observations. The formula is

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \ldots\ldots$$

α is the smoothing parameter ($0 < \alpha < 1$). A larger α gives more weight to recent observations.

D. Standard Regression Model

The form of the model is

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \ldots + \beta_k x_{k,t} + \epsilon_t$$

We are using the mean temperature as a response variable (y_t), and the humidity, wind speed and mean pressure as predictor variables ($x_{k,t}$). The coefficients $\beta_1, \ldots, \beta_k$ measure the effect of each predictor while considering the influence of all remaining predictors within the model. We also assume the error term (ϵ_t) follows the normal distribution with mean 0 and constant variance [2].

The purpose of the standard regression model is to forecast the response variable and get an estimate of the coefficient values of each predictor variable. By using the TSLM() function with a training dataset in R, we get a fitted regression model. After that, we need to check the residual plot of this fitted model to see whether the error term follows a normal distribution and has white noise behavior. The Ljung-Box test is a statistical test used for the presence or absence of autocorrelation of residuals in a model. If the p-value (significance level of the test) is less than the significance level (usually taken as 0.05), the null hypothesis can be rejected, indicating that there is autocorrelation (no white noise behavior) in the residuals[2]. After that, we are using the testing dataset to do the forecast part. To evaluate the forecast performance, we introduce the accuracy() function to check the Root Mean Squared Error (RMSE). The smaller the RMSE value, the better the performance of the forecast.

E. Dynamic Regression Model

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \ldots + \beta_k x_{k,t} + \eta_t,$$

$$\phi(B)(1-B)^d \eta_t = \theta(B)\epsilon_t,$$

$$\epsilon_t \sim IID(0, \sigma^2)$$

We assume the error term of the dynamic regression model is autocorrelated (not white noise) and follows the ARIMA model, and the error term of ARIMA(ϵ_t) follows white noise behavior.

This model can address the autocorrelations seen in the standard regression model because the ARIMA error term in the dynamic regression model captures information that is not explained in the standard regression model. The process of fitting a model, checking the residual plot, doing the forecast, and evaluating the result are similar to the standard regression model part.

F. Neural Network Model

We are using the NNETAR() function under the forecast() package in R. This function uses a neural network-based auto-regression model (NAR) for time series forecasting.The Neural Network Autoregression (NNAR) model captures nonlinear relationships in the data and may be better at forecasting time series with complex patterns or irregular variations. [4]. The model has two parameters, which are p and k. We use the notation NNAR(p, k) to indicate there are p-lagged input and k nodes in the hidden layer.

G. Facebook Prophet model

According to the Mode website, “Producing accurate forecasts historically posed challenges, leading to a shortage of analysts for critical business decisions. To address this, Facebook's Core Data Science team released Prophet, a Python and R forecasting library in 2017. Prophet aims to simplify forecasting for experts and novices alike, delivering superior forecasts with minimal effort and allowing domain knowledge application through intuitive parameters.” [6]

Before employing the `prophet()` function to fit the model, data preprocessing is essential. This model requires two variables: time and the response variable, which, in our case, represents the mean temperature. Thus, we need to create a data frame with two columns: a time column (usually named 'ds' in Prophet) and the corresponding target variable ('y'). Once this data frame is prepared, we proceed to fit the model and conduct the forecasting. Finally, we calculate the accuracy between the forecasted values and the test data to evaluate the model's performance.

GitHub Link: https://github.com/zma233/DASC6510Project

Ⅳ**. Result**

The dataset includes information regarding daily climate records in the city of Delhi from 2013-01-01 to 2017-04-24. The outcome is “Mean temperature”. This analysis builds multiple time series models on the training set (2013-01-01 to 2016-12-31), to forecast daily averaged temperature on the test set (2017-01-01 to 2017-04-24). Figure 5 plots the daily mean temperature across all time. It seems that there is a yearly seasonal trend that needs to be considered in the analysis.

A. Three benchmark models

RMSE scores of Mean, Naïve, and Drift models are shown in Table 2. The mean method performs the best within the group, with the lowest RMSE. Figure 6 shows the forecast result with prediction intervals of the mean method.

Table 2: RMSE scores of three benchmark models

Model	Mean	Naïve	Drift
RMSE	7.38	9.19	8.98

However, Figure 7 indicates that the residuals appear very auto-correlated as many lags exceed the significance threshold. This can also be seen in the residual plot, where there are periods of sustained high and low residuals. The distribution does not appear normally distributed (far from white noise) and is not centered around zero. So, the mean method does not provide a reasonable forecast.

B. ARIMA(p,d,q) model

The seasonal plot suggests there is a yearly seasonal trend, but at the same time, the curve of the seasonal plot is not smooth enough, indicating a significant amount of fluctuations. Also, the assumptions of the ARIMA model require checking stationarity.

In Figure 9, the ACF decreases slowly, and the value of the first lagged is large and positive, thus the data are non-stationary, with strong seasonality and a nonlinear trend. We then compare three differencing methods, including seasonal differencing, first differencing, and double differencing, in Figures 10, 11, and 12. The seasonal differencing and double differencing do not make data stationary, and The ACF and PACF in these two methods do not suggest seasonal patterns. The first difference seems to transfer data into stationary, and from ACF and PACF plots, we built three non-seasonal ARIMA models: ARIMA(20,1,0), ARIMA(0,1,9), and one automatically selected model: ARIMA(2,1,2). Of these models, the best is the ARIMA(2,1,2) model with the lowest AICc value.

Then, we check residuals from the ARIMA(2,1,2) model. In the Ljung-Box test, the P value is 0.03, less than 0.05, indicating the residuals do not pass the test and they are not white noise. Nevertheless, the confidence in rejecting the null hypothesis is not robust, as the p-value is not sufficiently low. In Figure 13, the histogram looks left-skewed.

Figure 14 presents the forecast of the ARIMA(2,1,2) model. The predicted trend is downward, which is contrary to the actual situation observed in the test data.

C. Exponentially Weighted Moving Average (EWMA) model

The smoothing parameter in the fitted EWMA model is 0.78, which means that this model gives more weight to recent observations. Figure 15 shows the result of residuals checking, the histogram is left-skewed. Also, in the Ljung-Box test, the P value is extremely small indicating that residuals are not white noise. The forecast of the EWMA model is shown in Figure 16, and it is not reasonable.

D. Standard Regression Model

By using the TSLM() function in R, we can get the coefficient value for each predictor. The intercept (β_0) is 818.942022, the coefficient of humidity is -0.146824, the coefficient of wind speed is -0.096572, and the coefficient of -0.777462.

From the residual plot of the standard regression model (Figure 17), it's evident that there is clear heteroscedasticity in the residuals. Additionally, the autocorrelation function (ACF) plot displays some significant autocorrelation in the residuals, and the histogram of the residuals shows long tails. All of these indicate that the model's error term does not follow the white noise behavior. The Ljung-Box test result shows the p-value is 0, and the null hypothesis can be rejected, indicating that there is also autocorrelation (no white noise behavior) in the residuals. Following the forecasting process and utilizing the accuracy() function, the Root Mean Squared Error (RMSE) value obtained stands at 2.852384.

Looking at the forecast plot (Figure 18), it's evident that the forecast results closely track the test data, mirroring the same upward trend. The considerably narrow confidence interval further supports the notion that this model performs well in forecasting.

E. Dynamic Regression Model

By using the ARIMA() function in R, we can get the coefficient value for each predictor. The coefficient of humidity is -0.1324, the coefficient of wind speed is -0.0400, the coefficient of -0.2375, and the error term (η_t) follows the ARIMA(2,1,1) model.

From the residual plot of the fitted dynamic regression model (Figure 19), it's evident that there is barely heteroscedasticity in the residuals. Additionally, there are only a few significant autocorrelations within the residuals, while the histogram displays a normal distribution. These observations collectively suggest that the ARIMA errors follow the white noise behavior very closely. Moreover, the Ljung-Box test result shows that the p-value is 0.351, which is larger than 0.05, thus we cannot reject the null hypothesis, affirming that the error term in the ARIMA(2,1,1) model indeed exhibits white noise behavior. Following the forecasting process and utilizing the accuracy() function, we get the RMSE value, which is 3.873715.

The forecast plot (Figure 20) indicates a strong alignment between the forecasted results and the test data in the first half. However, a noticeable deviation emerges between the forecast results and the test data in the latter half, growing more pronounced over time. Additionally, it's worth noting that the confidence interval appears wider when compared to the standard regression model. This wider interval suggests a potential inferiority in the forecast performance of this model when contrasted with the standard regression model.

Also, a combination of the dynamic regression model and the standard regression model is applied, attempting to acquire better accuracy. The RMSE of the combination model is 3.87 which is the same as the dynamic regression model. Figure 21 shows the forecast result of three regression models.

F. NNETAR Model:

The fitted model is NNAR(11, 6), which indicates that it has 11 lagged inputs (y_{t-1}, y_{t-2}, ..., y_{t-11}) and 6 nodes in the hidden layer.

The forecast plot (Figure 22) reveals a notably wide confidence interval. Moreover, the forecasted values gradually diverge from the test data, indicating a decreasing alignment over time. These aspects collectively suggest that the model's forecast performance might not be particularly strong.

By checking the residual plot of the fitted model (Figure 23), we can see there is barely heteroscedasticity in the residuals. The model also has few significant autocorrelations in the residuals, and the histogram of the residuals shows a normal distribution. It indicates that the model error follows the white noise behavior. After doing the forecast and using the accuracy() function, we get the RMSE value, which is 6.33.

G. Facebook Prophet Model:

The forecast plot (Figure 24) shows a great match with the seasonal pattern, and the confidence interval is quite narrow, suggesting a reliable forecast. Upon computing the accuracy between forecasted and test values, which yielded 3.171376, it further confirms the forecast's effectiveness.

The plot (Figure 25)displays the forecast segmented into trend, weekly seasonality, and yearly seasonality components. From the second and last plots, it's evident that this model can detect seasonal patterns that are challenging for an ARIMA model to capture. This further highlights the practicality and effectiveness of this model.

Table 3: RMSE scores of seven advanced models

Model	ARIMA	EWMA	Standard Regression	Dynamic Regression	Combinated Regression	NNAR	Prophet
RMSE	12.25	9.29	2.85	3.87	3.87	6.18	3.17

Ⅴ. Conclusion

Upon comparing the Root Mean Squared Error (RMSE) values across different models, it becomes evident that the Standard Regression Model has the lowest RMSE at 2.85. Therefore, based on this assessment, we can confidently assert that this particular model demonstrates the most accurate forecasting performance.

Reference

[1] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, UK, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.

[2] Hyndman RJ, Athanasopoulos G. Forecasting: Principles and practice (3rd ed) [Internet]. [cited 2023 Dec 3]. Available from: https://otexts.com/fpp3/

[3] Hyndman RJ, Athanasopoulos G. Forecasting: Principles and practice (3rd ed) [Internet]. [cited 2023 Dec 3]. Available from: https://otexts.com/fpp3/regression.html

[4] Hyndman RJ, Athanasopoulos G. Forecasting: Principles and practice (3rd ed) [Internet]. [cited 2023 Dec 3]. Available from: https://otexts.com/fpp3/nnetar.html

[5] Forecasting in R with prophet: Reports - mode [Internet]. 2018 [cited 2023 Dec 3]. Available from: https://mode.com/example-gallery/forecasting_prophet_r_cookbook

[6] Sumanthvrao. Daily climate time series data [Internet]. 2019 [cited 2023 Dec 3]. Available from: https://www.kaggle.com/datasets/sumanthvrao/daily-climate-time-series-data/

Appendix.

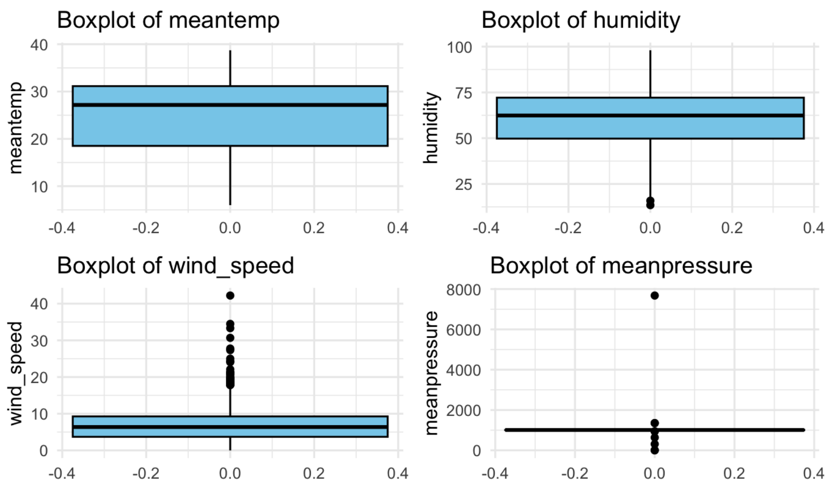

Figure 1: Boxplot of each variable.

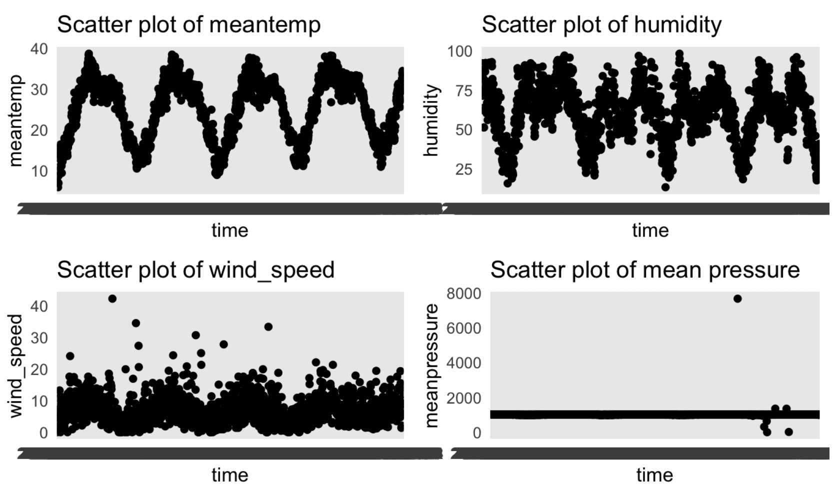

Figure 2: Scatter plot of each variable.

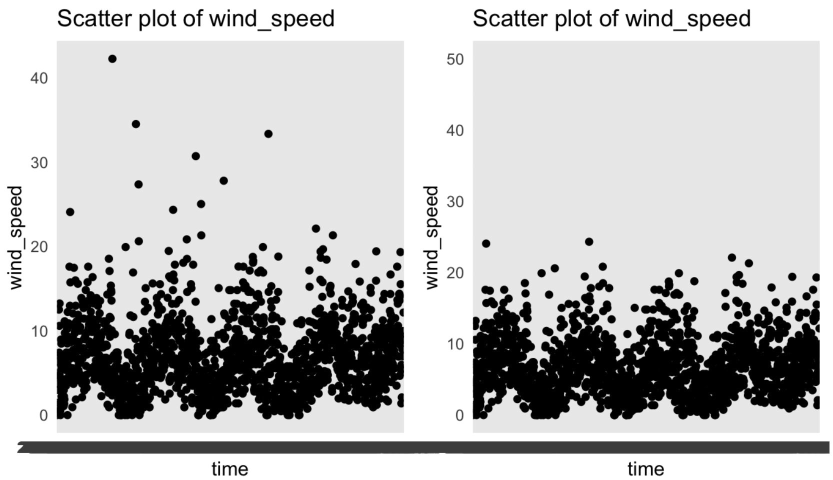

Figure 3: Comparison scatter plot of wind speed (Left: with outliers; Right: without outliers).

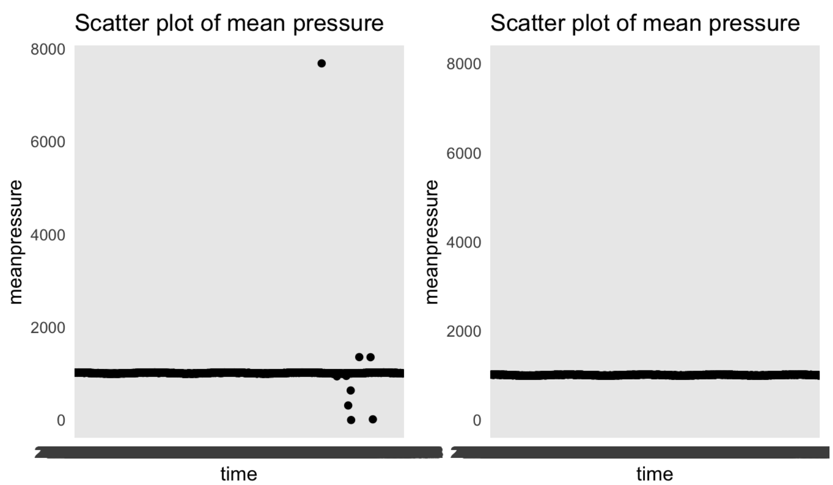

Figure 4: Comparison scatter plot of mean pressure (Left: with outliers; Right: without outliers).

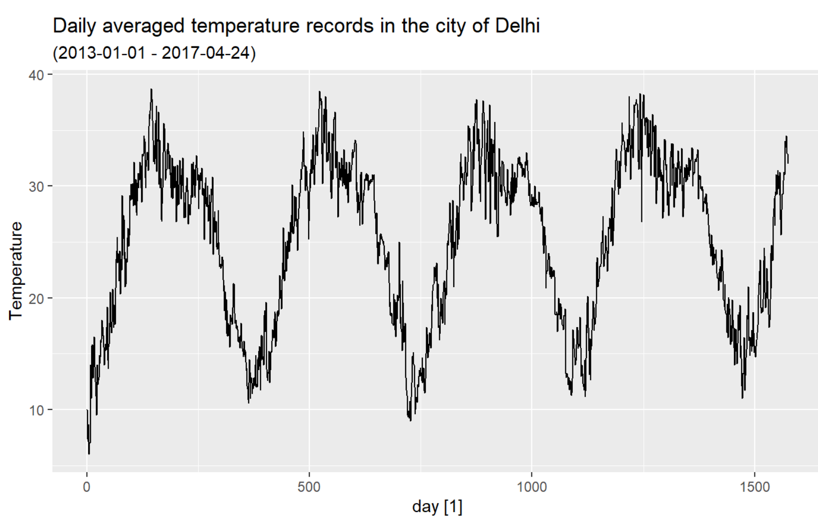

Figure 5: Daily averaged temperature records in the city of Delhi, 2013-01-01 to 2017-04-24.

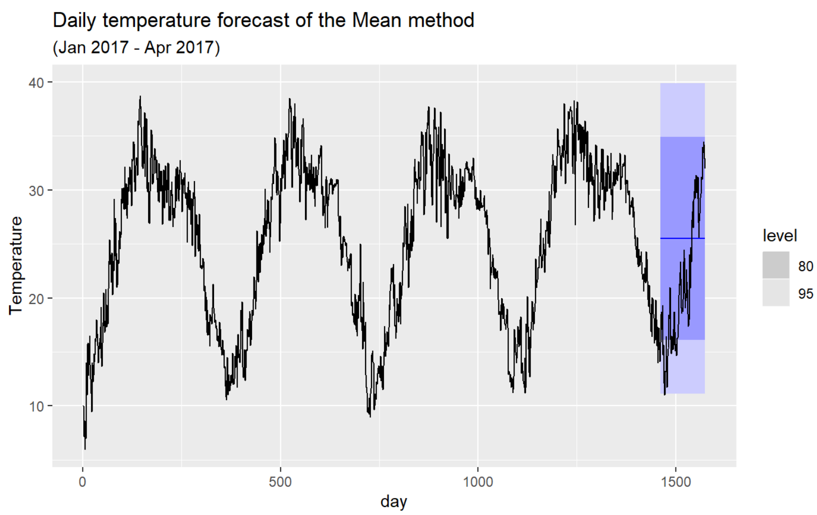

Figure 6: Daily temperature forecast of the Mean method, Jan 2017 - Apr 2017.

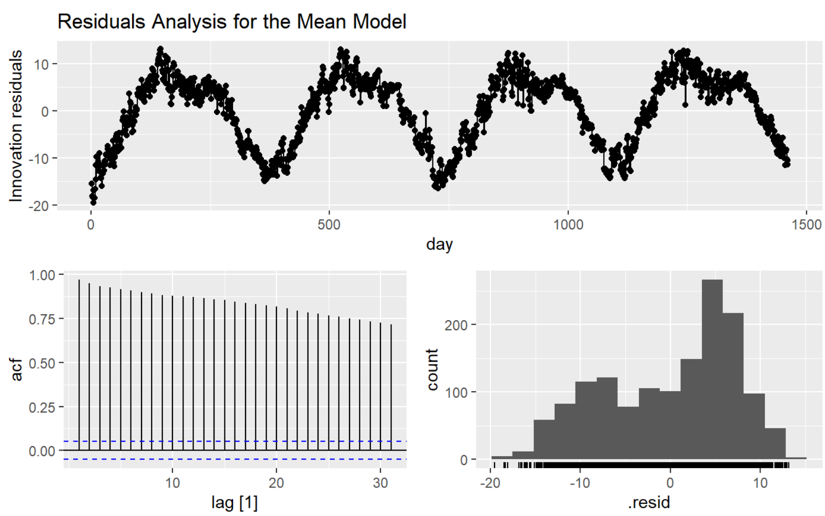

Figure 7: Residuals Analysis for the Mean Model.

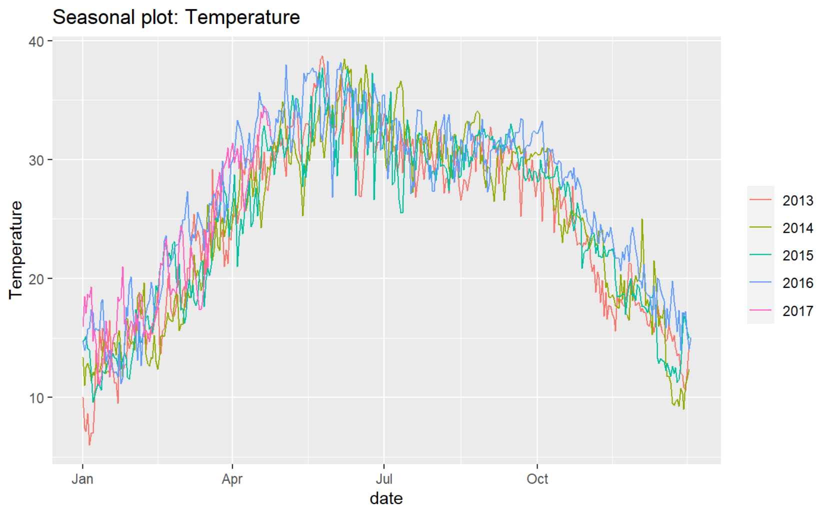

Figure 8: Seasonal plot of daily averaged temperature in Delhi.

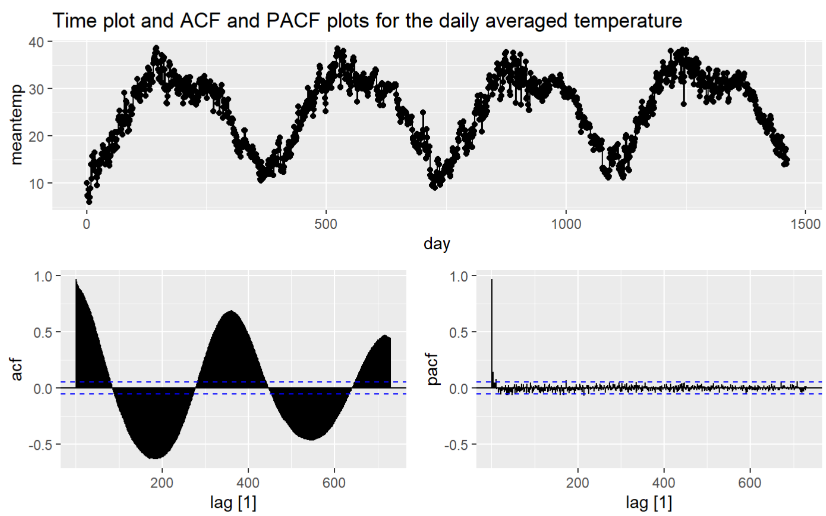

Figure 9: Time plot and ACF and PACF plots for the daily averaged temperature.

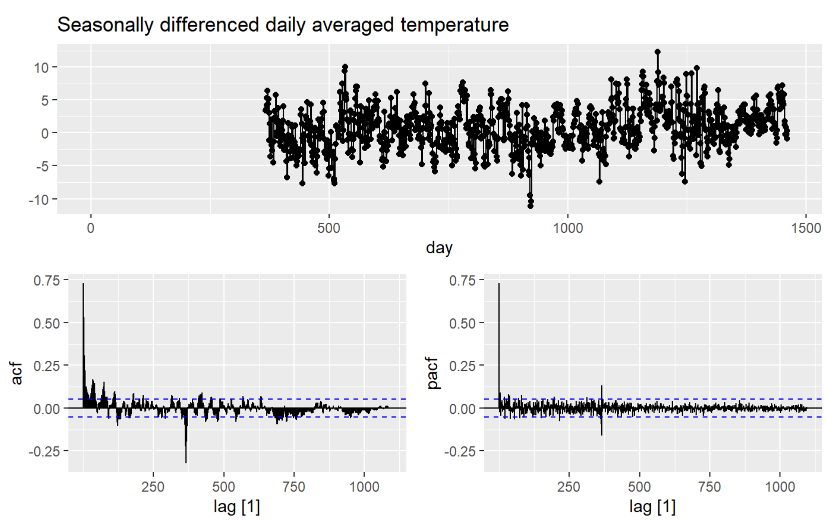

Figure 10: Seasonally differenced daily averaged temperature.

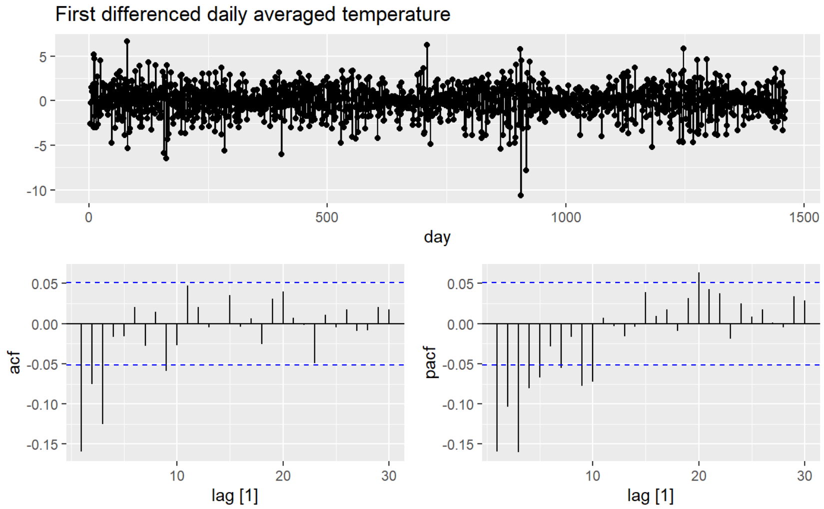

Figure 11: First differenced daily averaged temperature.

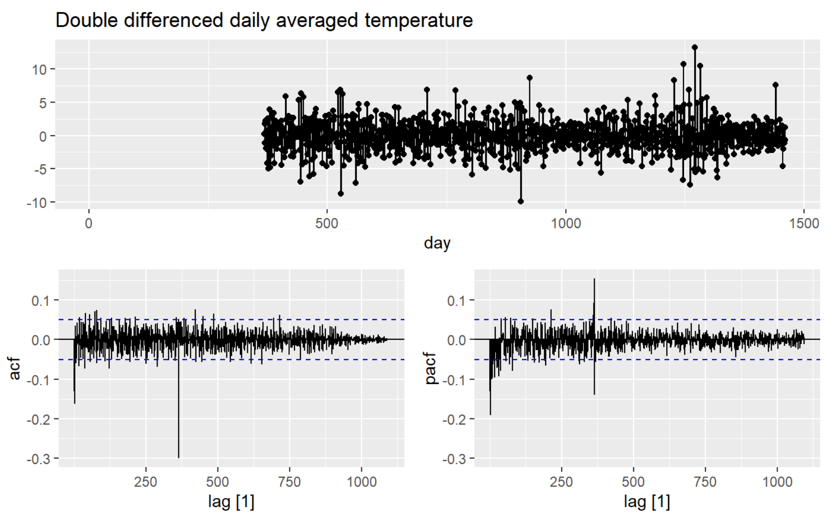

Figure 12: Double differenced daily averaged temperature.

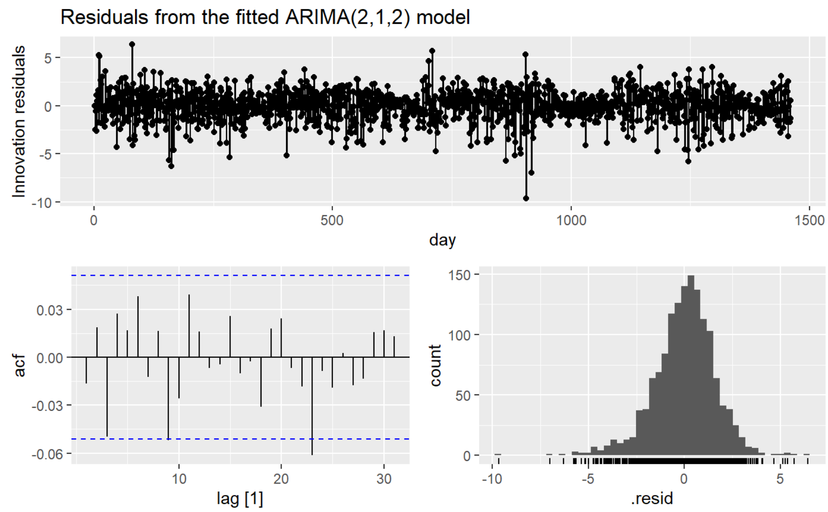

Figure 13: Residuals from the fitted ARIMA(2,1,2) model.

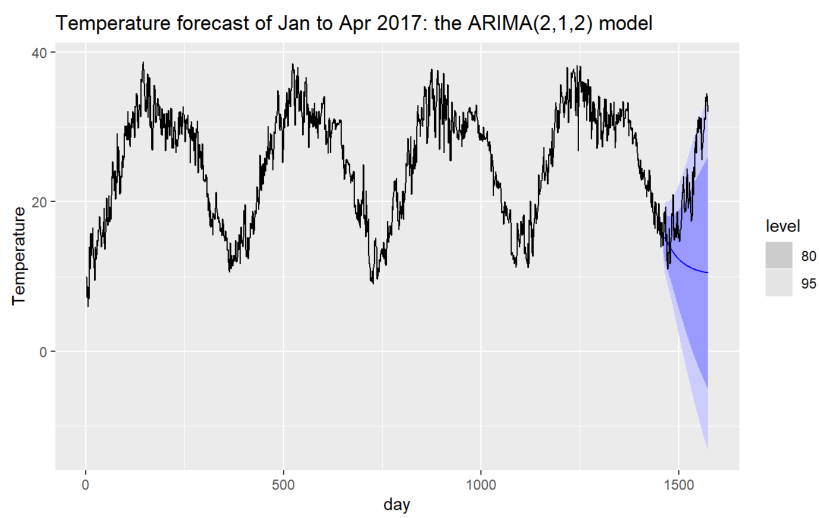

Figure 14: Temperature forecast of Jan to Apr 2017: the ARIMA(2,1,2) model.

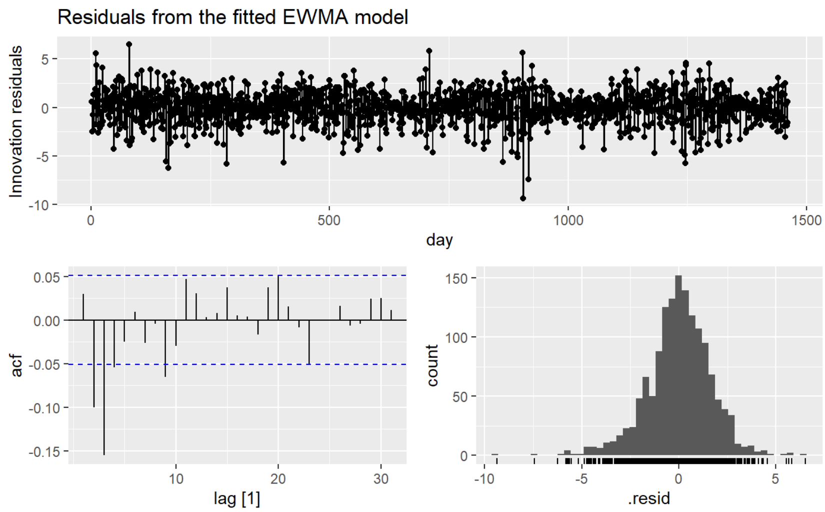

Figure 15: Residuals from the fitted EWMA model.

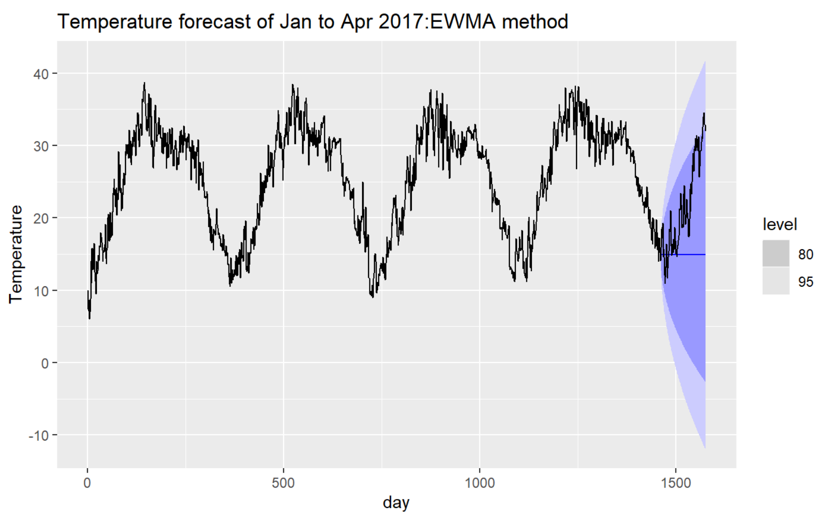

Figure 16: Temperature forecast of Jan to Apr 2017: EWMA model.

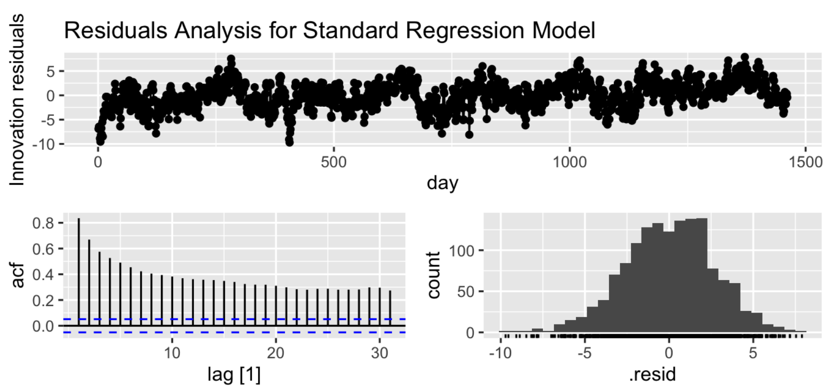

Figure 17: Residuals from the fitted Standard regression model.

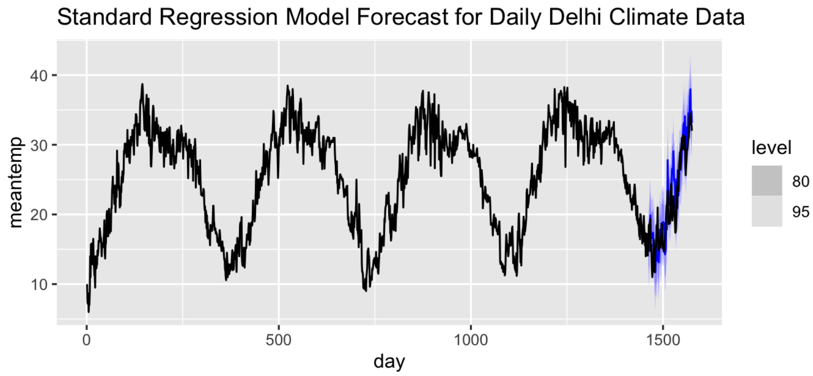

Figure 18: Temperature forecast of Jan to Apr 2017: Standard Regression Model.

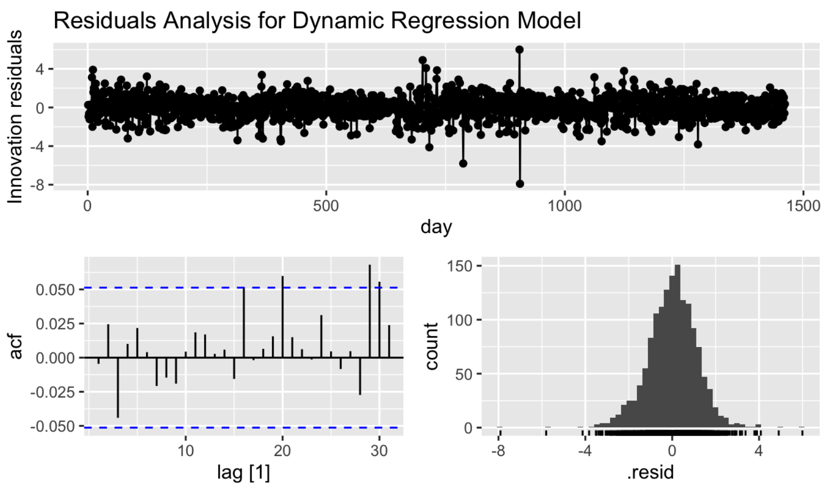

Figure 19: Residuals from the fitted Standard regression model.

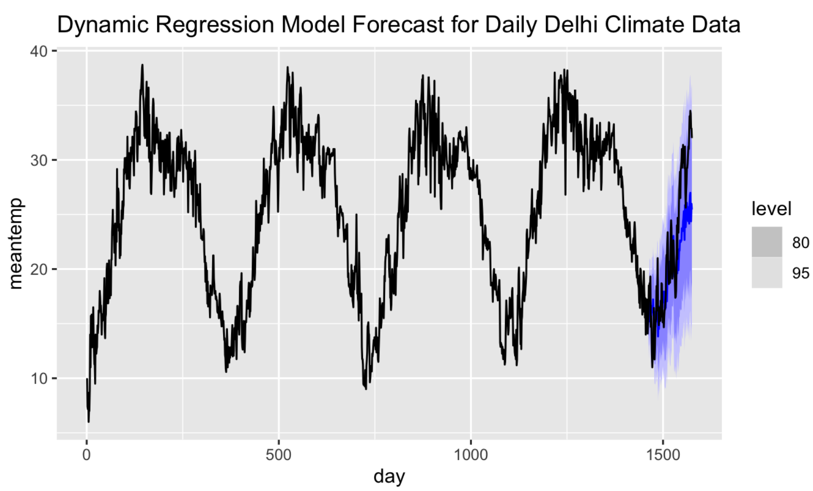

Figure 20: Temperature forecast of Jan to Apr 2017: Dynamic Regression Model.

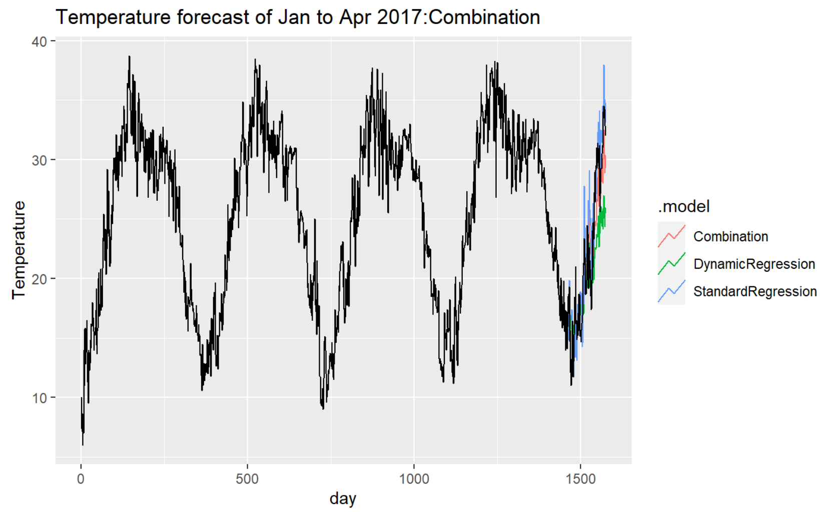

Figure 21: Temperature forecast of Jan to Apr 2017: Combination Model.

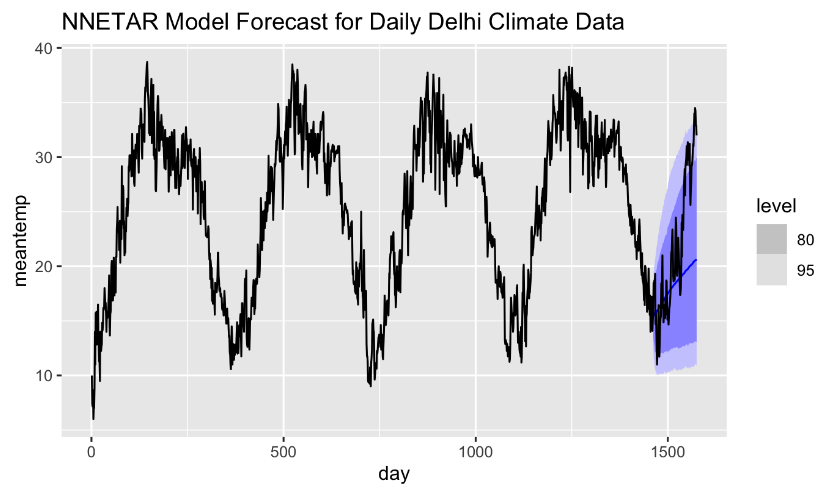

Figure 22: Temperature forecast of Jan to Apr 2017: NNETAR Model.

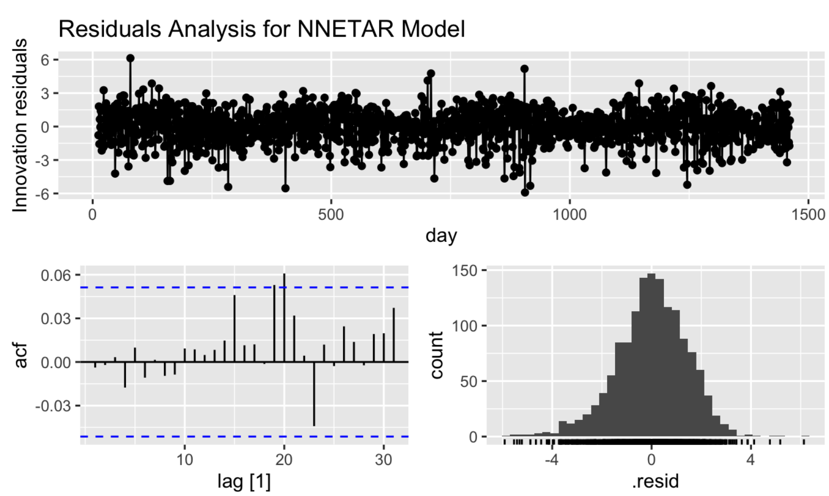

Figure 23: Residuals from the fitted Standard regression model.

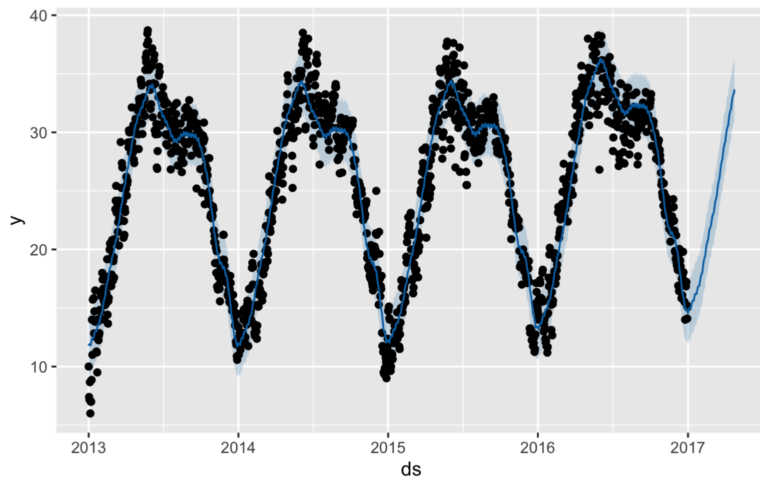

Figure 24: Temperature forecast of Jan to Apr 2017: Facebook Prophet Model.

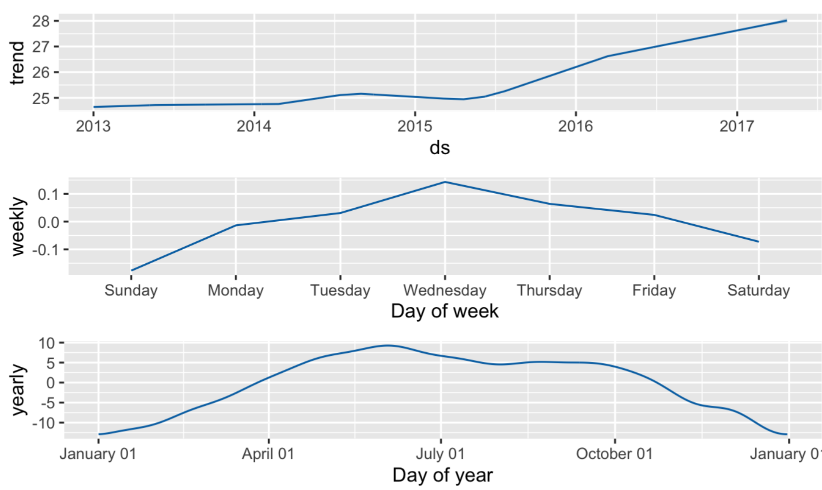

Figure 25: forecast trending, weekly seasonality, and yearly seasonality