## Data Understanding

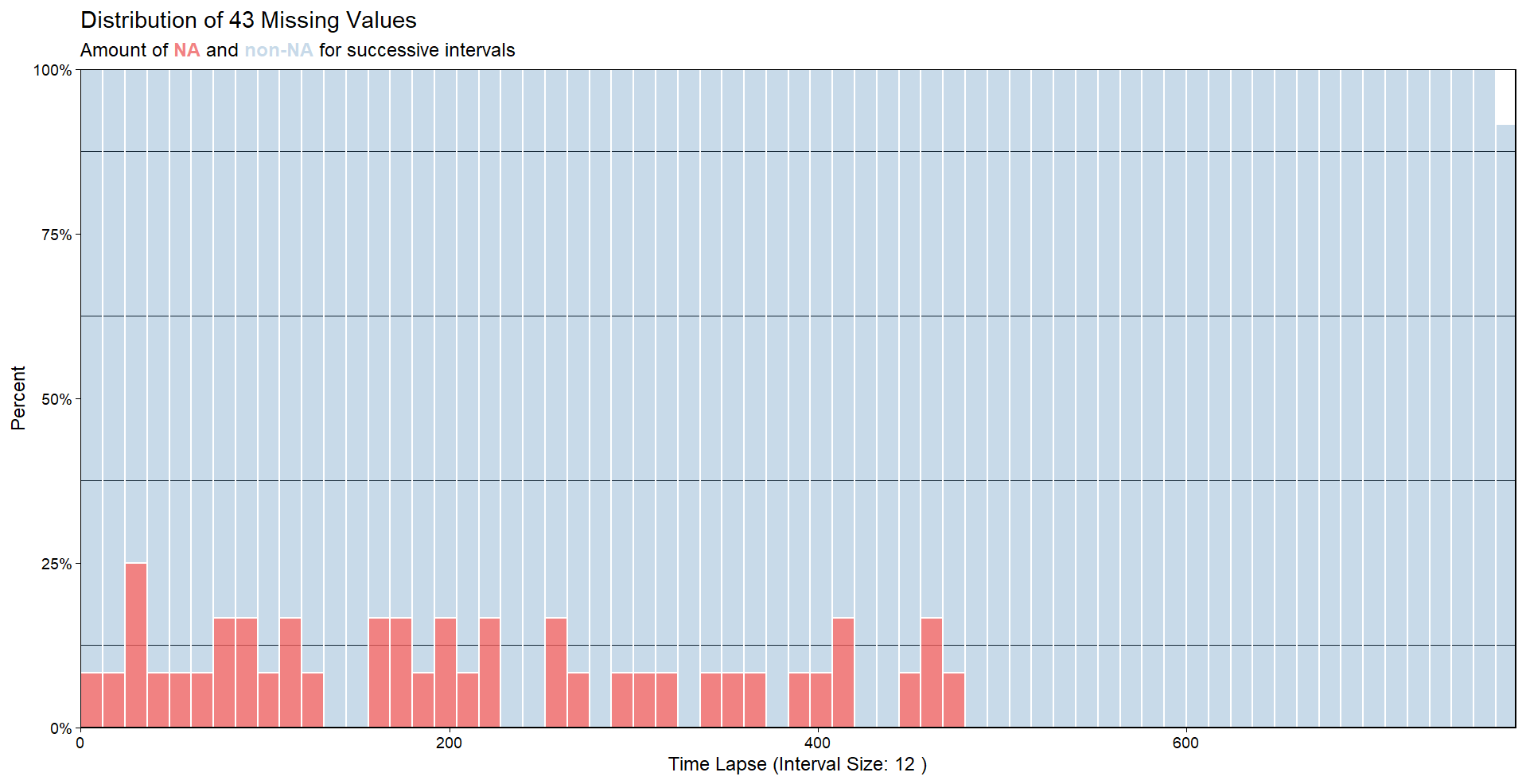
The dataset provided contains a seasonally-adjusted time series which means that the seasonal component has already been removed and the available dataset represents monthly Personal Consumption Expenditures (PCE) values spanning from January 1959 to November 2023.

## Data Cleaning

Initially, we load and read the dataset into R. We first convert the date column to the correct Date format else we won’t be able to perform any temporal analysis.

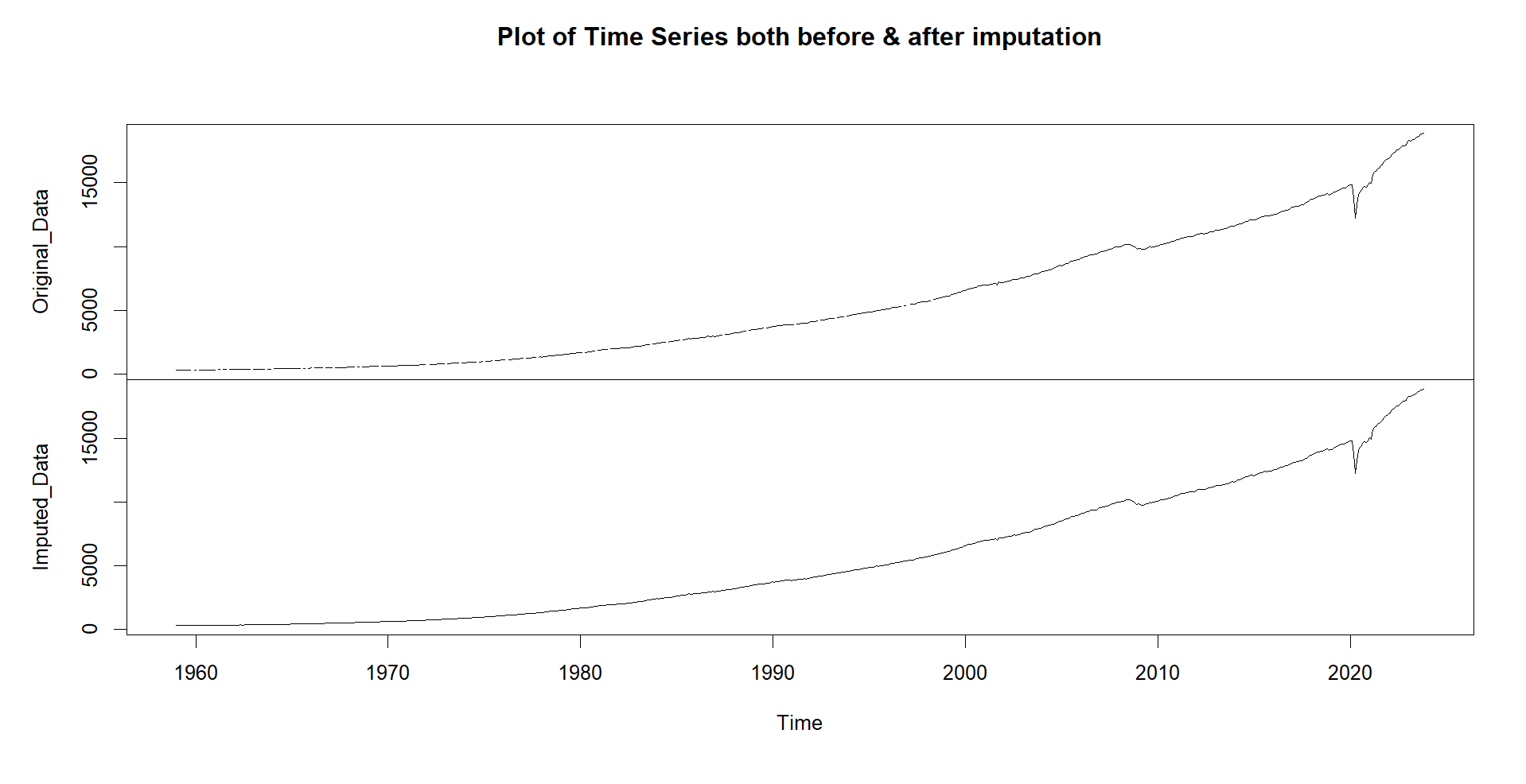
We then transform the dataset into a time series object by specifying the start of the dataset as January-1959, the end as November-2023 and frequency as 12 denoting monthly data.

We notice that the data has 43 missing values which we can see from Figure\_1.



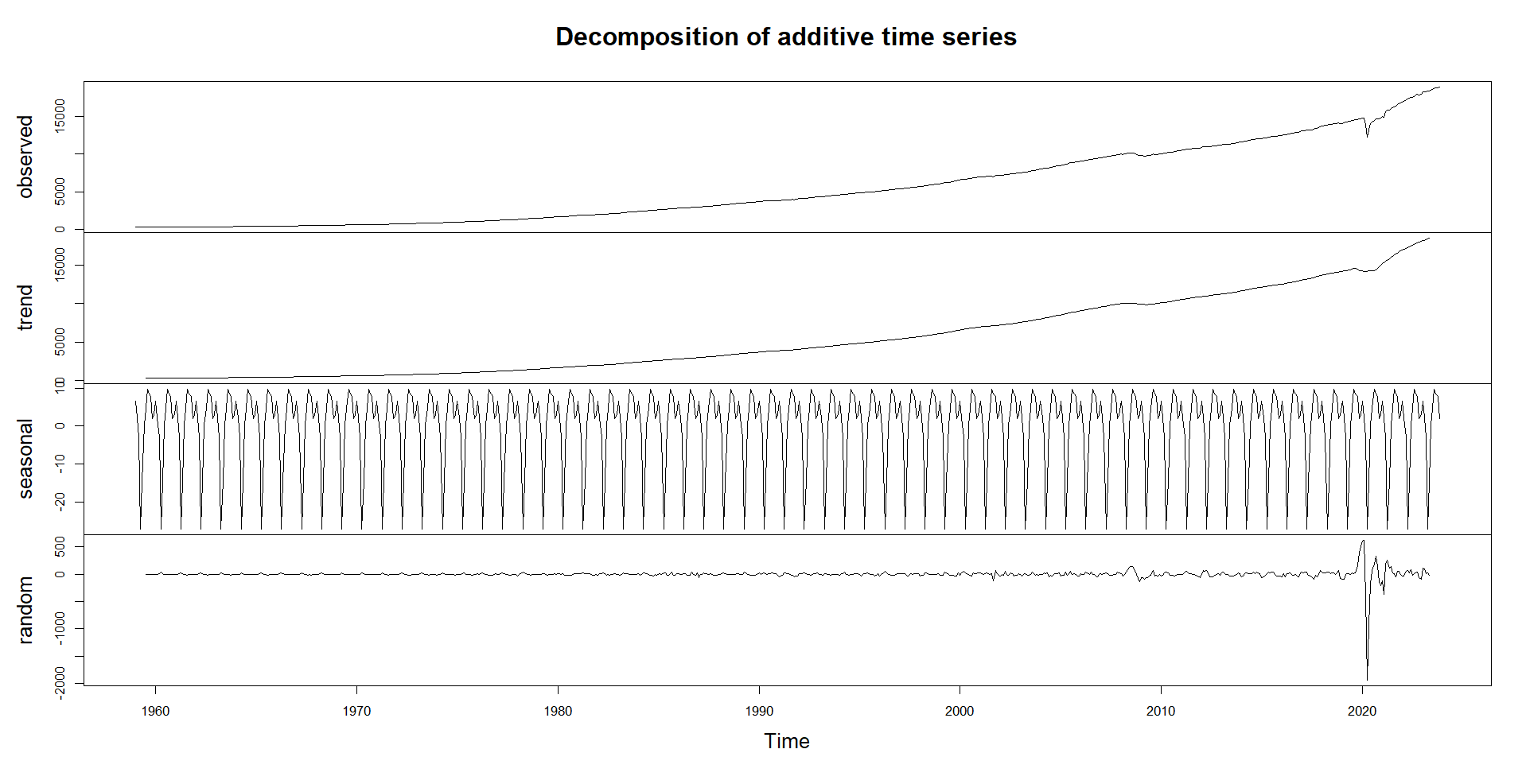
Figure\_: Missing values

We impute the missing data using the Kalman filter method. We then plot the imputed dataset to ensure successful imputation. The plot to visualize the imputation impact on the time series data is shown in Figure\_2.



Figure\_: Imputed time series

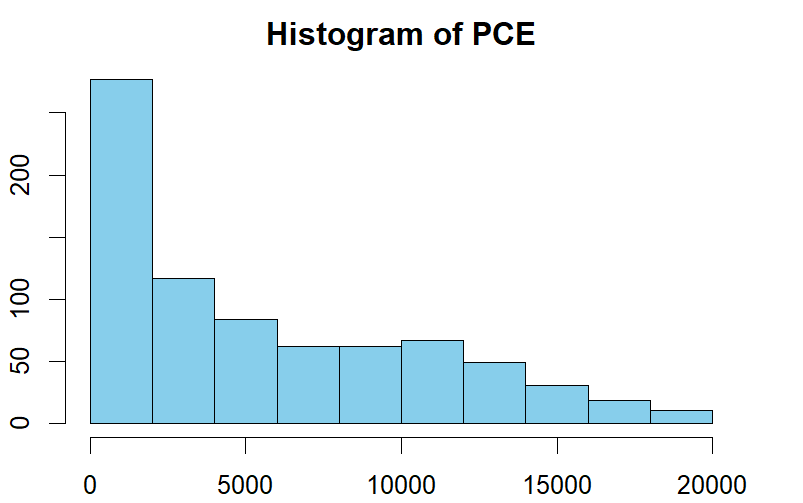
In order to understand the data pattern, we need to decompose the imputed time series into its constituent components: trend, seasonal, and random. As the variance of the time series doesn’t change over time, hence we decompose using the additive model as we can see in Figure\_3.



Figure\_: Additive decomposition

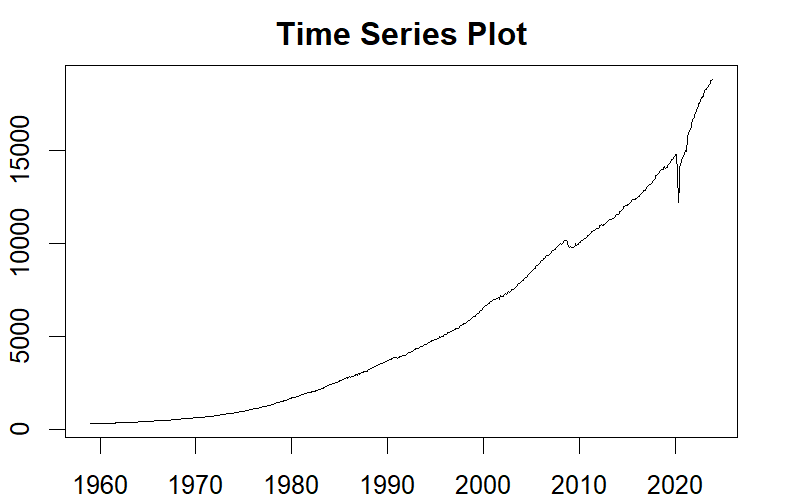
## Data Exploration

The histogram of the data in Figure\_4 indicates that the data is right skewed with more values towards the lower end of the range.



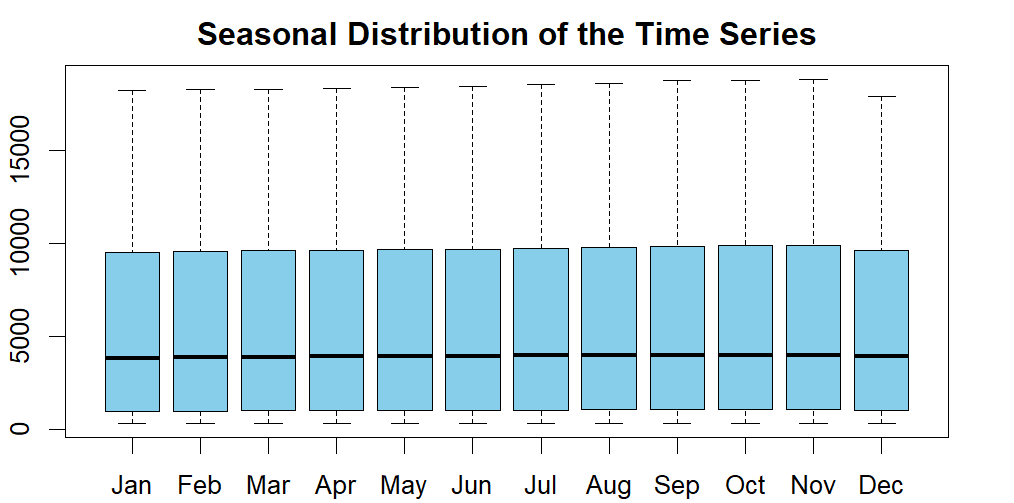
Figure\_: PCE distribution

By looking at the time series plot in Figure\_5, we can see that there is a linear upward trend over time which means that the PCE values have been steadily increasing over time.

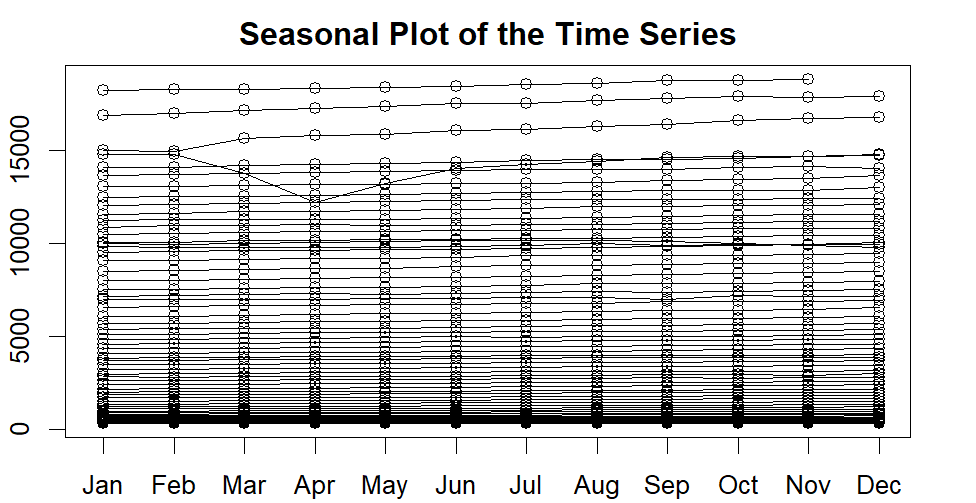


Figure\_: Time series plot

We also check the seasonal distribution of the data in Figure\_6 and Figure\_7 to see that the PCE value distribution remains the same for different months across all years showing no seasonality.

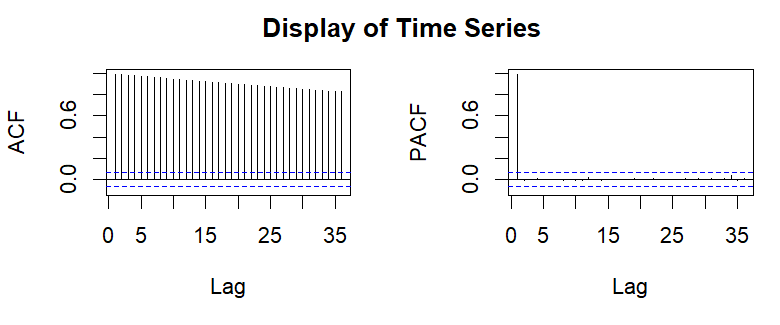


Figure\_: Seasonal distribution



Figure\_: Seasonal plot

The time series autocorrelation plot in Figure\_8 gives us further insight into the data. We notice that the autocorrelation slowly decreases as the lag increases, however there is significant autocorrelation at all lags which means that the data might not be stationary .



Figure\_: Autocorrelation plot

We apply both Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests and Figure\_9 shows that the original data is not stationary.

A computer screen shot of a program

Description automatically generated

Figure\_: Non-Stationarity

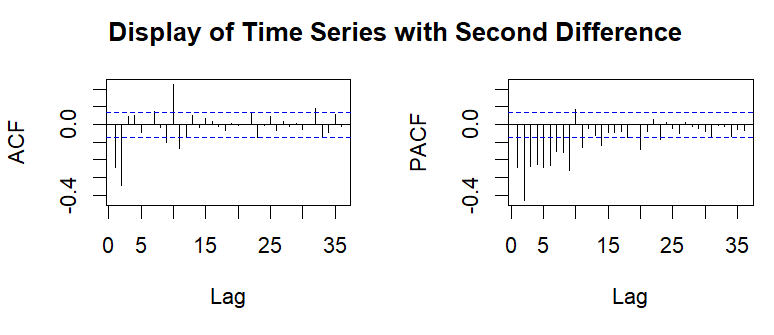
However, the data becomes stationary after taking the second difference which is depicted in Figure\_10. This essentially means that the degree of differencing (d) is 2.

A computer screen shot of a program

Description automatically generated

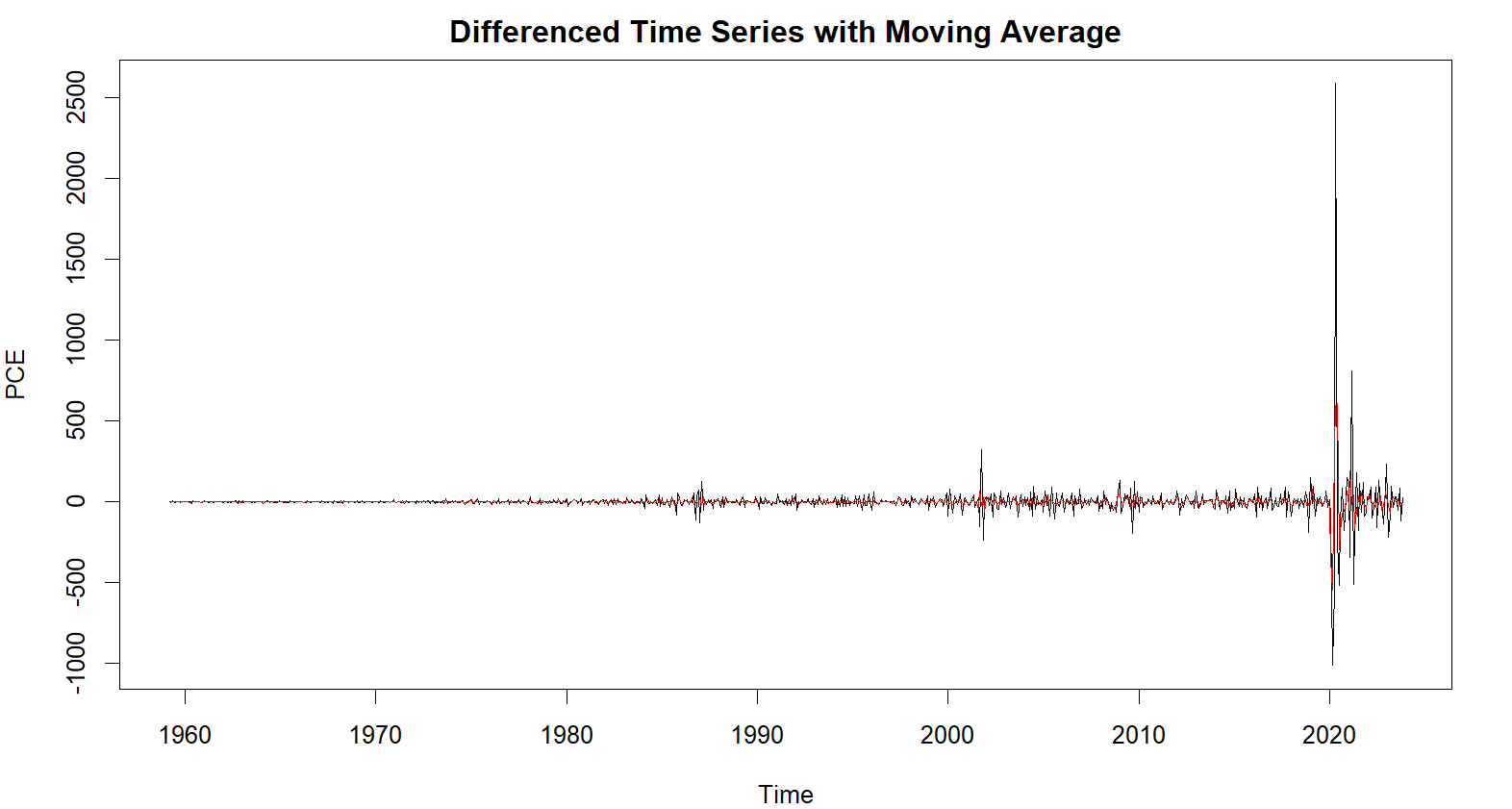
Figure\_: Stationarity

In Figure\_11, we look at the autocorrelation plot of the differenced time series. We can see from the ACF/PACF plots that there are moving average and autoregressive components to the time series. Additionally, there are some lags that lie outside the confidence interval in both plots. By visually inspecting the PACF plot, we can consider the order of autoregressive model (p) as 2. Similarly, by looking at the ACF plot, we can consider the order of moving-average model (q) as 2.



Figure\_: Autocorrelation plot – differenced series

To reduce noise and highlight the trends properly, we smoothen the data and plot the same to showcase the smoother version of the data as seen in Figure\_12.



Figure\_: Smoothening

## Modelling

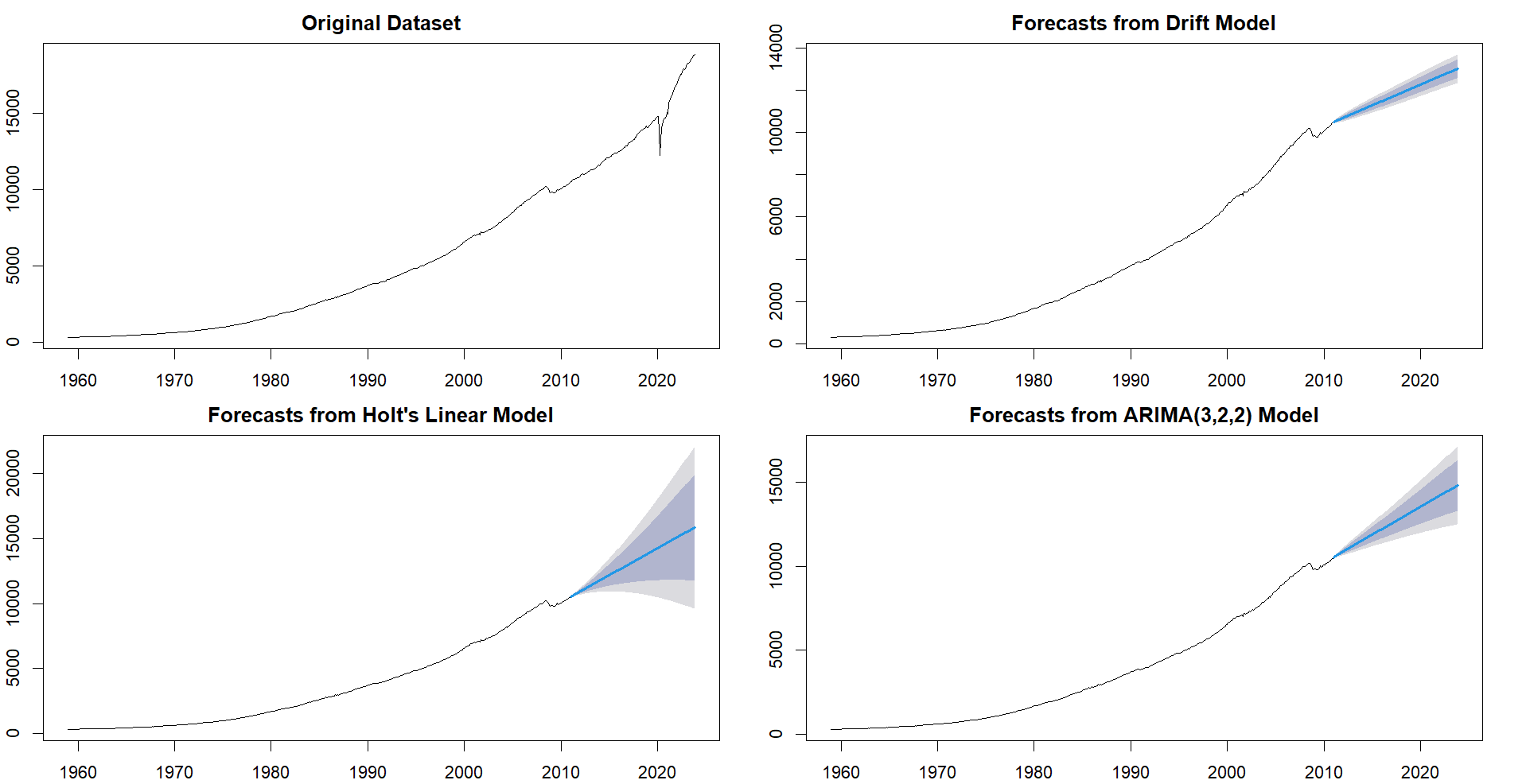
We split the data into 2 parts and allocate around 80% of the data for training and remaining 20% for testing. The training set contains data till end of 2010, whereas the test set contains data from January 2011 onwards. The test data is reserved for evaluating the performance of the trained models, providing an unbiased assessment of their predictive capabilities on new observations.

## Evaluation

Once the training and testing dataset is prepared, we now apply different forecasting models on the training dataset and use that model to predict future values for the test dataset and then calculate the model’s accuracy by comparing the predicated values to the real values in the testing dataset. The models considered are:

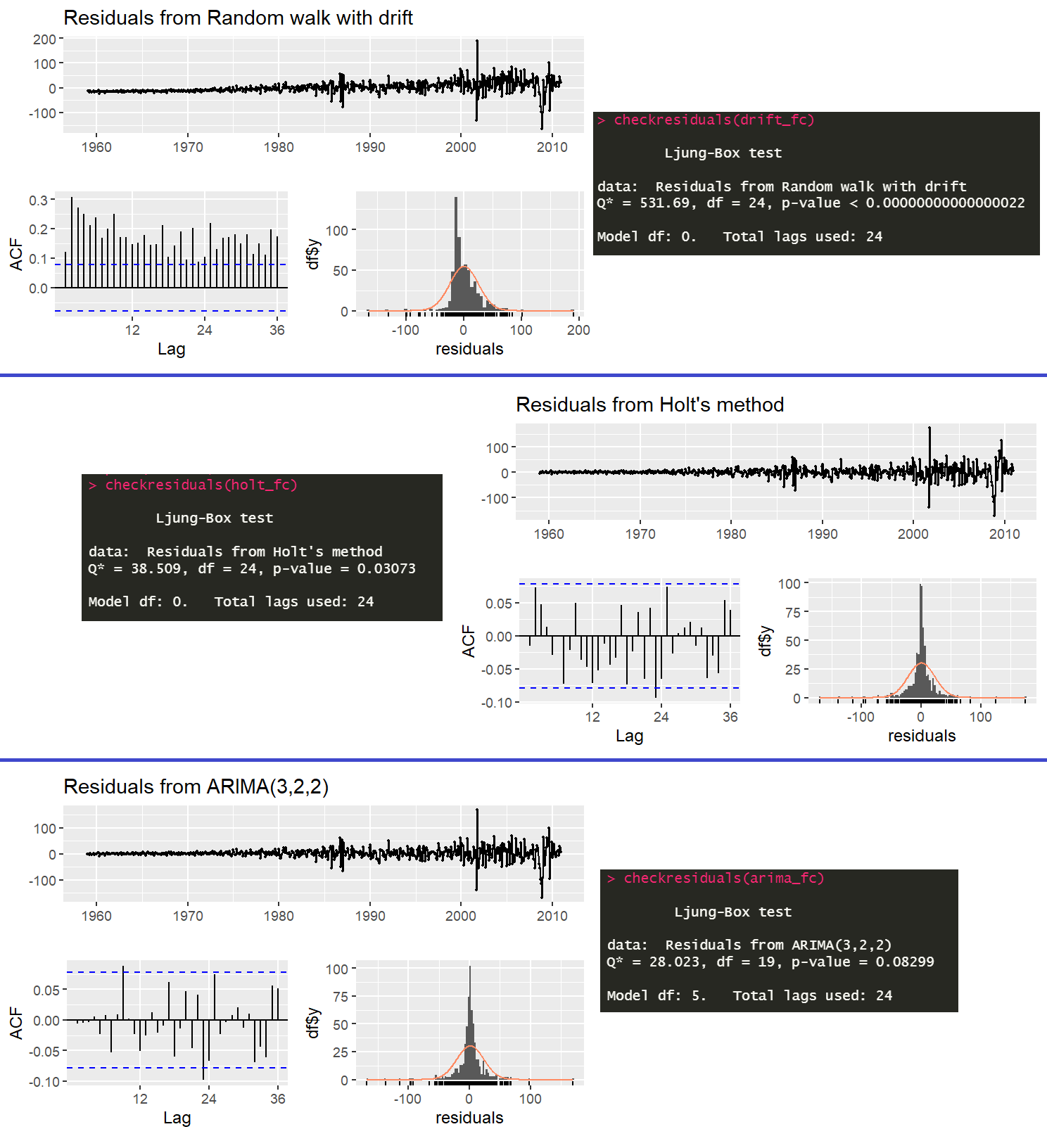
* **Drift:** Among the simple forecasting techniques (Average, Naïve, Seasonal Naïve, and Drift), we selected the drift method as there is an upward linear trend in the data but no seasonality since the time series is already seasonally-adjusted. The forecast generated in this method is by assuming that the next value will be equal to the last observed value along with a drift constant added to it which is the average amount of change over time seen in the historical data.
* **Holt:** Amont the exponential smoothing models (Simple exponential smoothing, Holt, Additive Holt-Winters, Multiplicative Holt-Winters), Holt’s linear method was selected as the data shows an additive trend component and no seasonal component. Holt's Linear Model incorporates both level and trend components to capture linear trends in the data and provides forecast by accounting both the current level and the trend’s rate of change.
* For the **Autoregressive integrated moving average (ARIMA)** Model, we fit the model using the trend parameters (p,d,q) as (2,2,2) which we got while exploring the time series data earlier. This essentially means that the model has autoregressive and moving average components of the second order while taking the second difference of the original time series. The selected ARIMA model is then used to generate forecasts for the test period. As we used the ACF and PACF plots to understand the components of the model before applying, this model might not be the best model when it comes to deciding the parameters. Using the ‘auto.arima()’ function, we now re-generate the best model. By looking at the summary, we notice that the best model has parameters (3,2,2) which means that this is essentially a model with only a trend component.

The forecasts generated by all these models are presented below in Figure\_13.



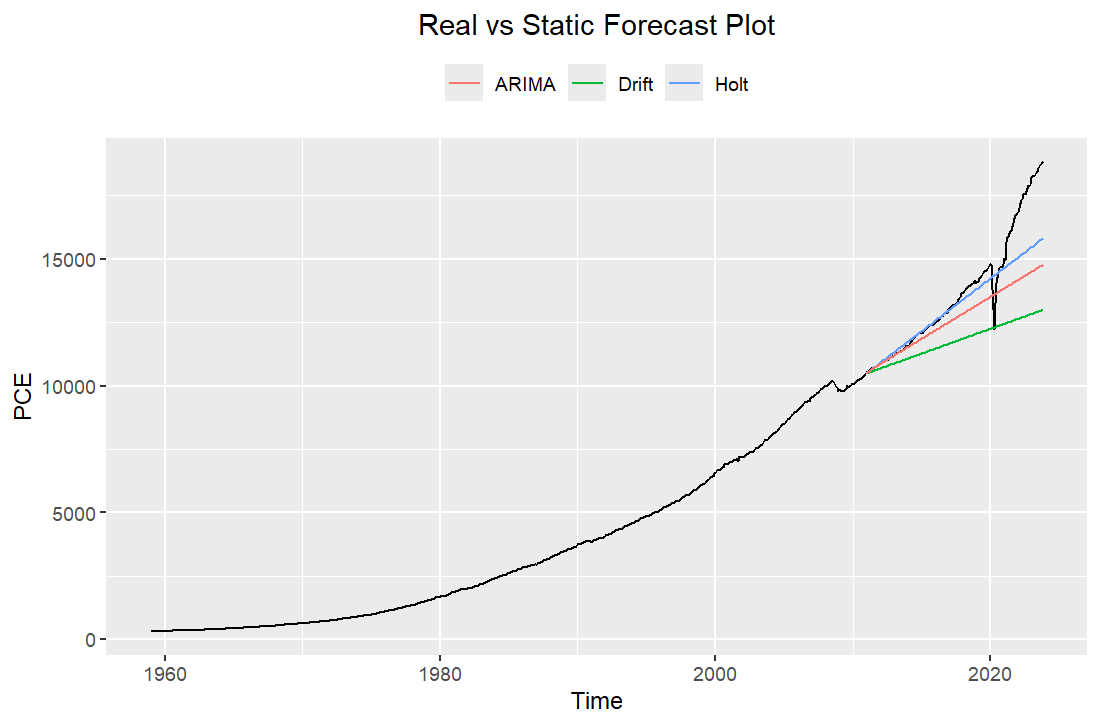
Figure\_: Model forecasts

As seen in Figure\_14, we can say from the Ljung-Box test that Drift and Holt showed significant correlation in the residuals as p-value was lower than 0.05, whereas ARIMA showed no significant correlation in the residuals.



Figure\_: Model residuals

To compare the performance of the three models, a plot is created to visualize the original PCE data alongside the forecasts generated by each model. This allows for a visual comparison of the models' predictive accuracy as shown in Figure\_15.



Figure\_: Real vs Static forecast

To analyse and compare each model, we look at accuracy metrics which are robust, widely-popular, and easy to interpret for a seasonally-adjusted time series, such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) for each model. These metrics are summarized and presented in Figure\_16 below where we notice that the RMSE, MAE, and MAPE values for the Holt’s Linear model are the lowest suggesting that it is the best model as it contributes to the lowest error.

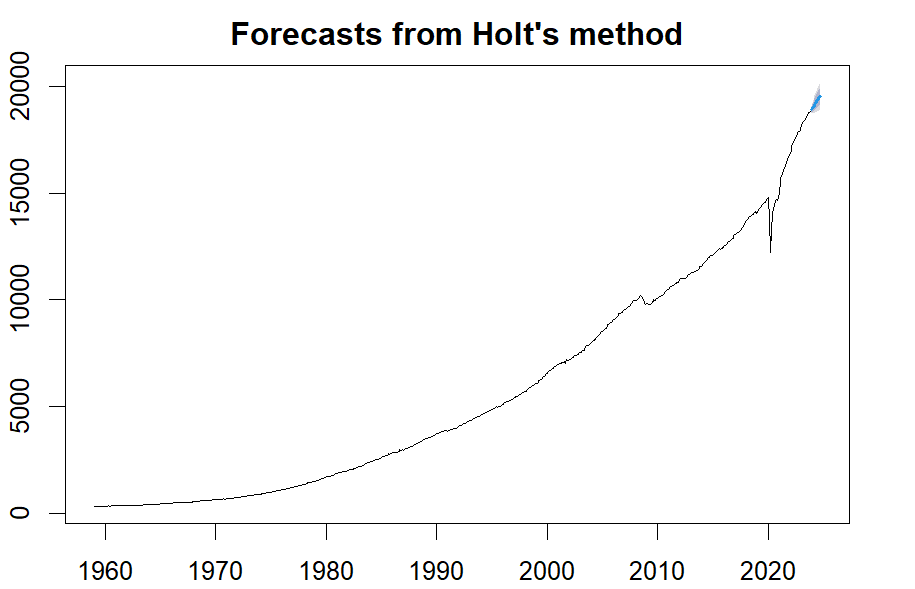
A blue and black numbers

Description automatically generated with medium confidence

Figure\_: Static model measures

## Forecasting

In order to forecast the PCE values for October 2024, we apply the Holt’s Linear model to the complete dataset. We use the refitted Holt model to forecast PCE values for the next 11 months till October 2024 which can be seen in Figure\_17 that visualizes the predicted trend and variability over the forecast horizon. The PCE forecast for October 2024 comes to 19566.92 as shown in Figure\_18.



Figure\_: Holt forecast

A screenshot of a computer screen

Description automatically generated

Figure\_: Oct-2024 forecast

## Rolling Forecast

Rolling Forecast is a way of optimizing our forecast by considering the values till date. The process involves forecasting for smaller horizons and including the actual values till date and include that in the model to make better future forecasts.

We performed a one-step rolling forecast on each model maintaining the parameters we got during our model training phase. This meant training the models using a rolling window of past data, refitting them periodically, and evaluating their performance based on the test dataset.

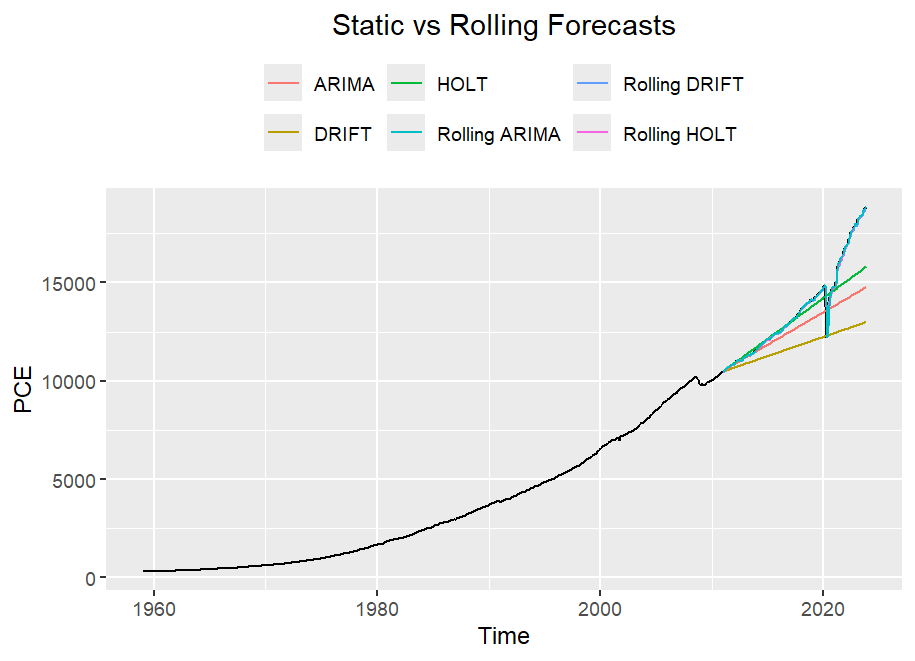
Finally, we compare the original forecasts with the rolling forecast models. We can clearly see that from Figure\_19 that rolling forecasts helped reduce the errors drastically and hence they are deemed superior to static forecasts. For instance, Rolling Holt reduced the RMSE of the Holt model from 1085.0608 to 200.7651.

A table with numbers and a few digits

Description automatically generated with medium confidence

Figure\_: Static and Rolling forecast measures

Among these models, the Rolling Holt model demonstrates the best performance based on all metrics. These values indicate that the Rolling Holt model has the smallest overall prediction errors compared to the other models. Specifically, it outperforms the other rolling forecasting methods such as Rolling Drift and Rolling ARIMA, as well as the static models including ARIMA, Drift, and Holt. Therefore, based on the provided evaluation metrics, the Rolling Holt model emerges as the most effective choice for forecasting in this context. The comparative forecast plot for all static and rolling forecast models are shown below in Figure\_20.



Figure\_: Static vs Rolling forecasts