PART 1

1. INTRODUCTION

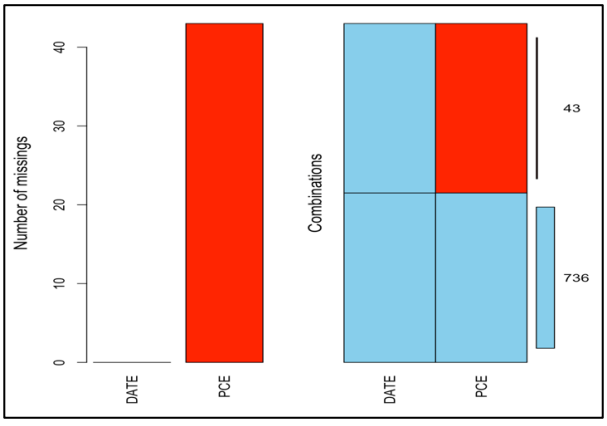
Time series forecasting is a technique to predict future values based on historical data points collected at regular intervals over time. It involves analysing the patterns and trends in time series data to make informed predictions about future values. The purpose of the task is now to predict and compare the predictive ability of the following models using the US seasonally - adjusted personal consumption expenditures (PCE).

- One of the four simple forecasting methods (average, naïve, seasonal naïve or drift)
- An exponential smoothing model.
- An ARIMA model

2. DATA UNDERSTANDING

We are provided with US seasonally adjusted personal consumption expenditures (PCE) data (***PCE.csv***), starting from 01st January 1959 until 01st November 2023, with 779 observations and two variables.

VARIABLE	DESCRIPTION
DATE	Date of the observation
PCE	Personal Consumption Expenditure on that date.

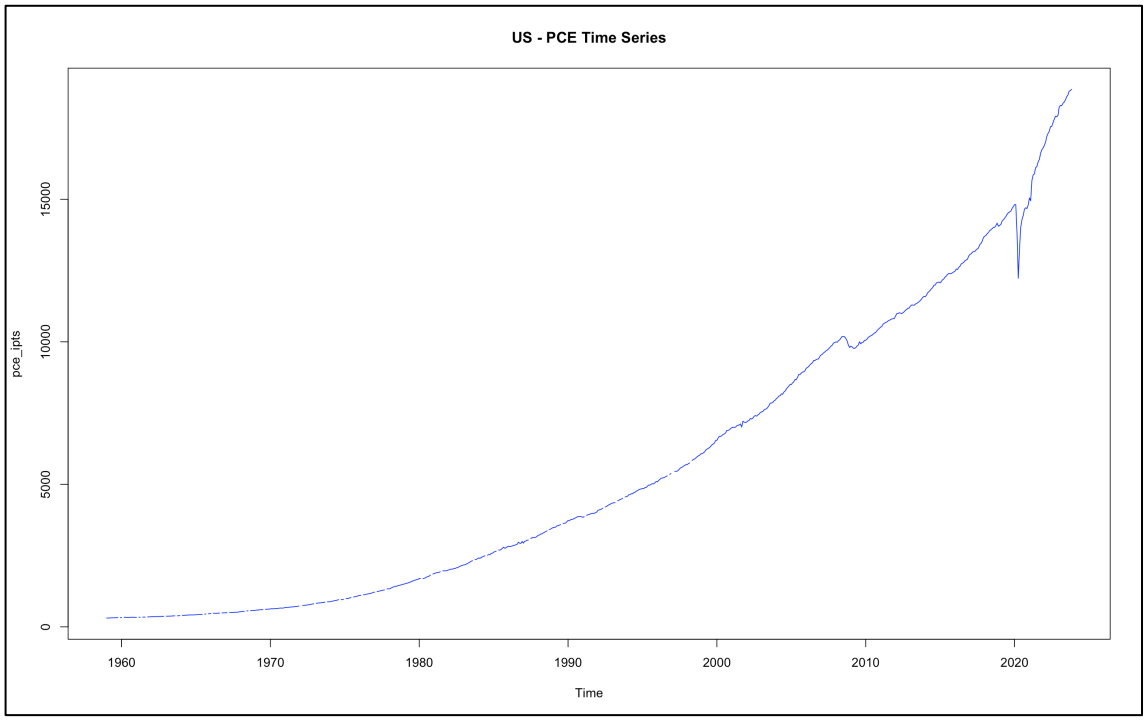


Out of 779, there are 43 missing values, which constitutes **5.51%** of the total data. Hence missing values need to be imputed for further analysis.

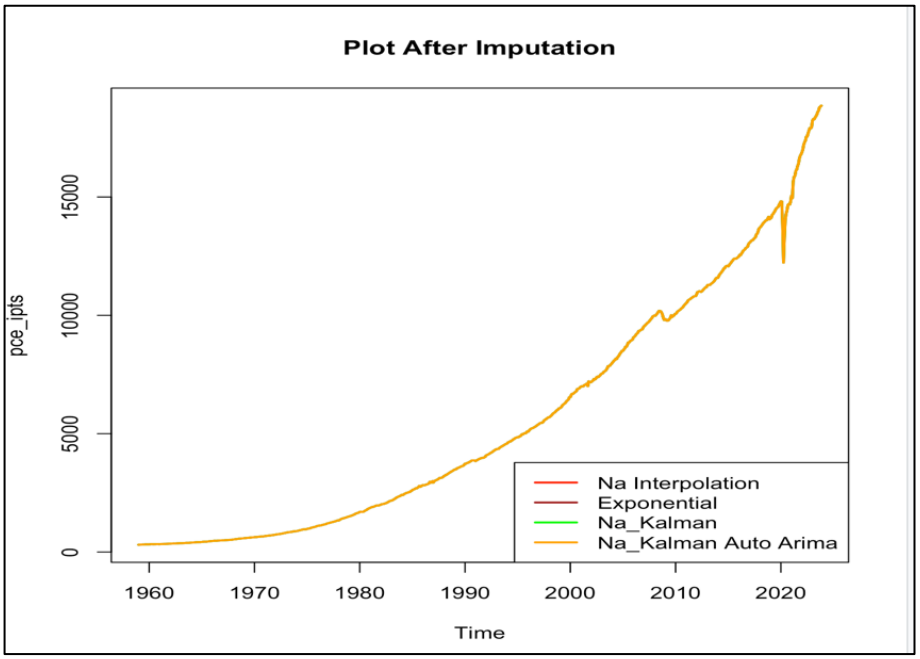
3. Data Preparation

As a primary step, the data imported needs to be converted into a Time Series object as below

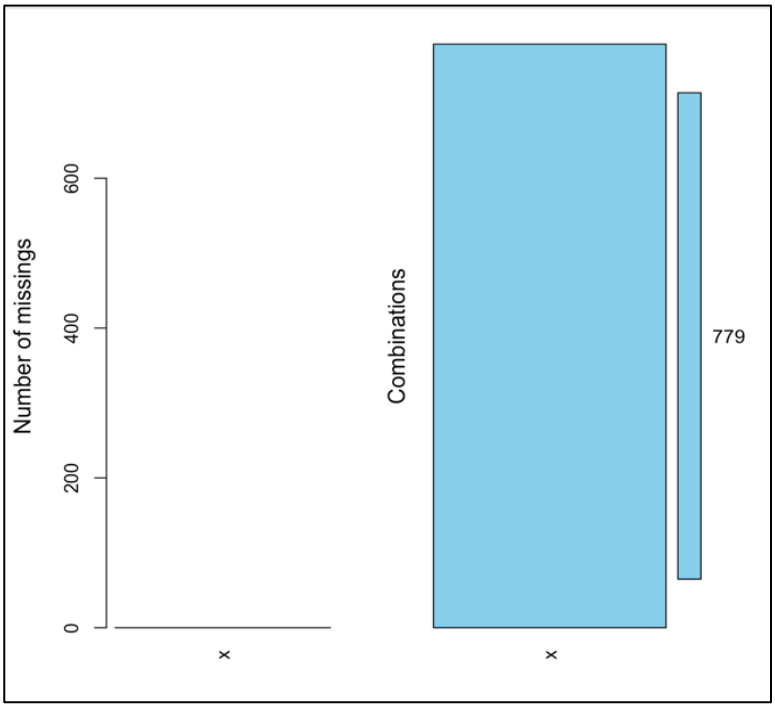
```
# ---------- Chnage to Time series object and Plot -----------
pce_ipts <- ts(pce_data$PCE, start = c(1959, 1), end = c(2023, 11), frequency = 12)
plot(pce_ipts)
```



As observed in the previous step, there is missing data, and we need to impute them to overcome inaccuracies and facilitate proper time series analysis. It also improves forecasting accuracy and visualizations. We utilize **imputeTS** package for imputation. We use four types of imputation methods, to compare and assess the differences that it can induce following the imputation - **na_interpolation()**, **na_kalman()** and the **na_ma()**

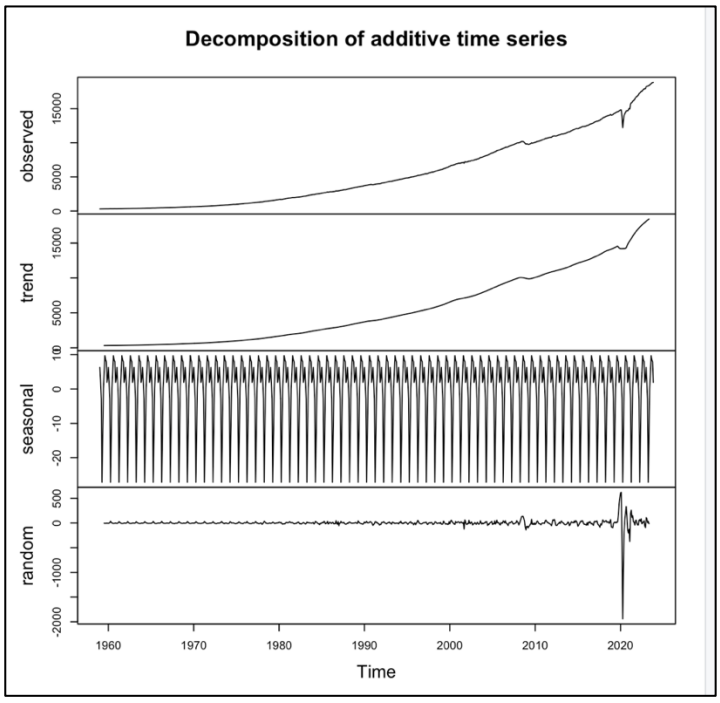


It is observed that, there is no visible difference or change to the time series after imputation. The ***na_kalman*** is most suitable for time series with seasonality and trend. Interpolation takes the mean between the values before missing data and value after and this linear interpolation is the simplest. Therefore, the imputation method selected is interpolation. Now there are no missing values in the imputed data set, as shown below.



DECOMPOSITION

The given time series is seasonally adjusted. But still we explore the possibility of seasonality within the time series. Hence we try to decompose the time series.

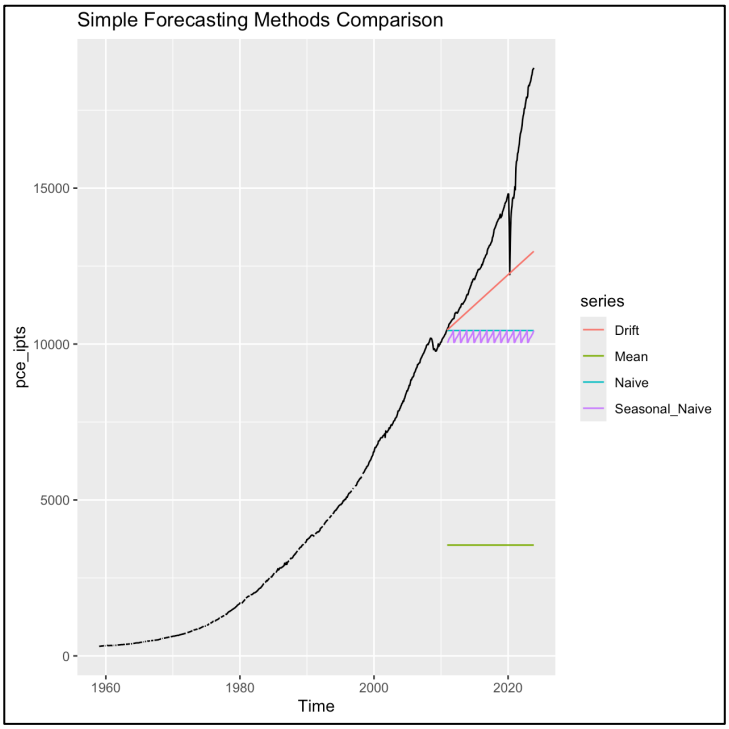


We ignore multiplicative decomposition as the trend is not an exponential growth. But when additive decomposition is performed, it is observed that there is absolutely minimal seasonality and noises in it . For now, it is ignored and taken up for further analysis.

4. TIME SERIES ANALYSIS

4.1 SIMPLE FORECASTING METHODS

In time series analysis, several forecasting methods are used for understanding and modelling sequential data. The *average method*, predicts future values based on the historical average of the series. *Naïve method*, assumes the next observation as the most recent observation, which could be accurate for a stable time series. *Seasonal naïve,* on the other hand, is useful for seasonal data, where the next observation is forecasted to be equal to the same season in the previous year. Finally, the *drift method* is for linear trends for exploring the future values based on the average rate of the change observed in the data.



EVALUATION OF THE FORECASTING METHODS

Evaluation of the accuracy of the forecasted values generated by the models are done using the test dataset. This provides the evaluation of how well the forecast model performs on data that has not been in the training set. This also gives insights on how well the model generalizes to new observations.

NAÏVE METHOD

```
> accuracy(fcnaive,test)
                    ME       RMSE        MAE        MPE       MAPE       MASE      ACF1 Theil's U
Training set   16.28215   29.69766   19.89598  0.5641837  0.6559345  0.09811548 0.1206498        NA
Test set     3204.14551 3976.13173 3204.14551 21.3499291 21.3499291 15.80099418 0.9746013  16.67327
>
```

SEASONAL NAÏVE

```
> accuracy(fcsnaive,test)
                    ME      RMSE       MAE       MPE      MAPE     MASE      ACF1 Theil's U
Training set  194.5976  251.4858  202.7813  6.562192  6.645324  1.00000 0.9761424        NA
Test set     3412.3455 4141.8834 3412.3455 22.931063 22.931063 16.82772 0.9752355  17.50232
>
```

MEAN METHOD

```
> accuracy(fmean,test)
                      ME      RMSE       MAE        MPE      MAPE     MASE      ACF1 Theil's U
Training set -1.430666e-12  3124.649  2658.459 -205.33471 239.61696 13.10998 0.9951973        NA
Test set      1.008371e+04 10354.918 10083.713   73.20916  73.20916 49.72704 0.9746013  49.13873
```

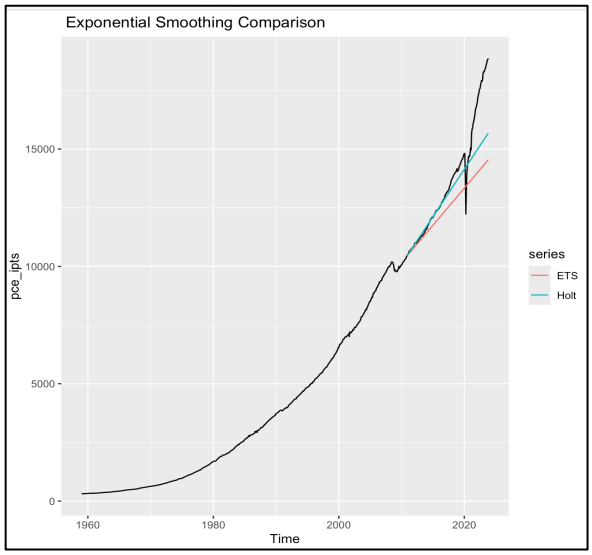
DRIFT METHOD

```
> accuracy(fcdrift,test)
                     ME       RMSE        MAE        MPE      MAPE       MASE      ACF1 Theil's U
Training set 2.321259e-14   24.83631   16.72042 -0.8283488  1.131257 0.08245545 0.1206498        NA
Test set     1.925996e+03 2545.27790 1926.57567 12.5872452 12.591982 9.50075796 0.9702413  10.38423
```

Based on these metrics and the plot above, it appears that the **Drift Method,** performed the best on the test set, as it has the lowest RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) and MPE (Mean Percentage Error) values among the three methods. Seasonal Naïve can be avoided as the series is seasonally adjusted. Therefore, if we prioritize forecast accuracy on unseen data, the drift method would be the preferred choice.

4.2 EXPONENTIAL SMOOTHING

This technique used in time series forecasting assigns exponentially decreasing weights to past observations and there several methods to perform exponential smoothing. *Simple Exponential Smoothing (SES)*, calculates weighted average of the past observations with the weight decreasing exponentially as the observations get older. This is suitable for time series data without trend or seasonality. *Holt's Linear Method*, on the other hand, is suitable for data with a trend but no seasonality. It incorporates both linear and trend components to provide more accurate forecasts, *Holt's Winters Method,* incorporates seasonality in addition to trend and level components. It includes three sets of weights, one for level, then for trend and one for the seasonal component. There is Error Trend Seasonality, *ETS* model that models using the train data, captures the time series structures and then generates forecasts based on the best model suggested. In this case it came out to be the ETS(M,A,N) model.



The time series displays a trend but lacks any seasonality after seasonal adjustment. Thus, both simple exponential smoothing designed for no trend and seasonality and the Holt-Winters method, which are designed for seasonal data, are not relevant. Given the absence of seasonality, using Holt's linear method appears suitable and appropriate.

EVALUATION OF THE EXPONENTIAL SMOOTHING METHODS

The assessment of forecast model performance is conducted using the test data. When reviewing the metrics provided below and comparing them with those of the Drift method, it becomes evident that Holt's Linear Method exponential smoothing would be the preferred choice.

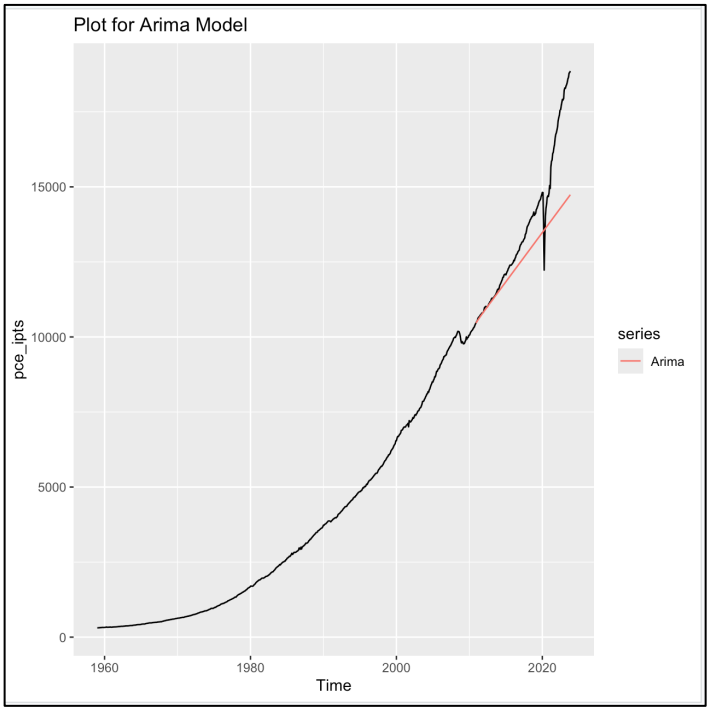
```
[1] "Holt's Linear Exponential Model"
> accuracy(fcholt, test)
                 ME   RMSE   MAE    MPE  MAPE  MASE    ACF1 Theil's U
Training set   0.436   22.5  12.4 0.0249 0.398 0.061 -0.0149        NA
Test set     564.403 1145.7 645.2 3.2546 3.917 3.182  0.9569       4.4
```

```
[1] "ETS Model"
> accuracy(forecast_ets, test)
                   ME   RMSE    MAE    MPE  MAPE   MASE  ACF1 Theil's U
Training set    0.677   22.7   12.3 0.0472 0.394 0.0605 0.101        NA
Test set     1137.644 1697.4 1155.2 7.1864 7.328 5.6967 0.964       6.7
>
```

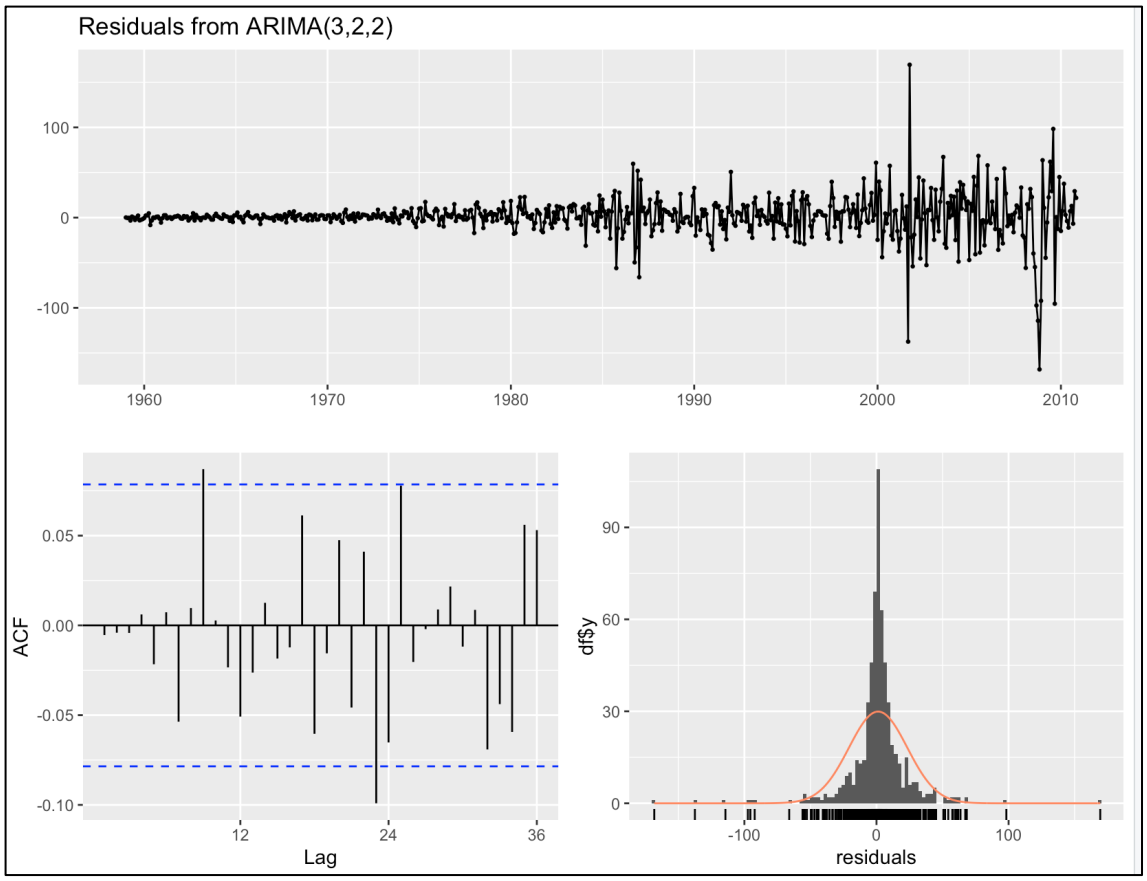
4.3 AUTOREGRESSIVE INTEGRATED MOVING AVERAGE MODEL

ARIMA, is another method used for analysing and forecasting time series data. It has three parts. *Autoregressive (AR),* suggests that the value of the series at a particular time is dependent on its previous values. *Integrated (I),* indicates that the data needs to be differenced to make it stationary, that is mean, variance remains constant over time, *Moving Average (MA),* considers the term errors of the series and predicts the next value based on the weighted average of past error terms. ARIMA models are very flexible and can be used in different forecast models.

Below is the auto Arima model, **ARIMA(3,2,2)** , the model selected by *auto.arima()* function based on the information criteria like AIC (Akaike Information Criterion) , where the original time series data was differenced twice to make it stationary, three autoregressive terms or lag observations and two moving average terms, included two lagged forecast errors.



Below is the assessment on the residuals for the Auto Arima model.



Ljung Box test , tests the presence of the serial correlation in the residuals. The p-value less than 0.05 suggests significance of correlation in the residuals, that is rejecting the null hypothesis, indicating the model would be inadequate. In our case, p-value is 0.085 and that is greater than 0.05, we fail to reject the null hypothesis, indicating there is no significant autocorrelation in the residuals.

```
        Ljung-Box test

data:  Residuals from ARIMA(3,2,2)
Q* = 27.895, df = 19, p-value = 0.08546

Model df: 5.   Total lags used: 24
```

4.3.1 SENSITIVITY ANALYSIS ON ARIMA MODELS

To improve the model's fit to the data, AR and MA orders were adjusted and their accuracies were evaluated for each of the new models.

AUTO ARIMA (3,2,2)

```
> accuracy(fcauto_arima,test)
                     ME       RMSE        MAE        MPE      MAPE       MASE         ACF1 Theil's U
Training set    1.256645   22.10412   12.36151 0.06713522 0.4029679 0.06095981 -0.005380479        NA
Test set     1021.242570 1593.06514 1042.73242 6.36201334 6.5351073 5.14215375  0.963691956  6.240727
```

ARIMA (3,2,3)

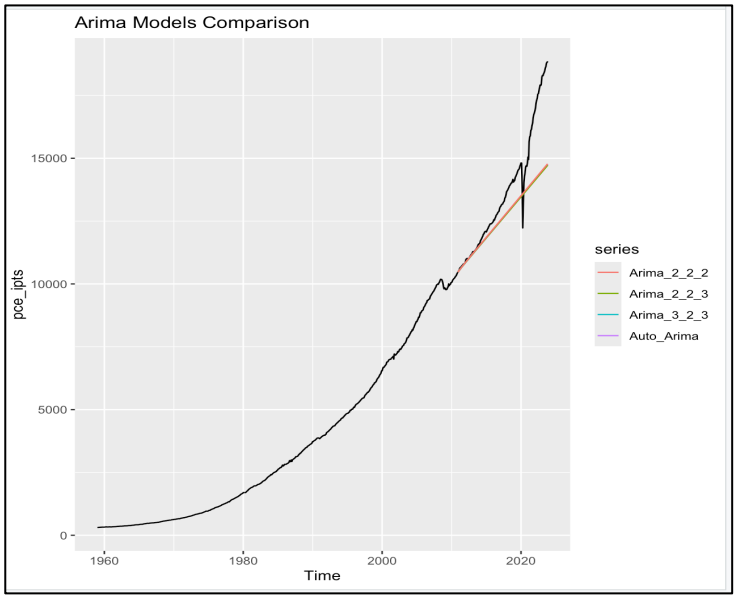
```
> accuracy(fc_arima_stand,test)
                     ME       RMSE        MAE        MPE      MAPE       MASE         ACF1 Theil's U
Training set    1.257592   22.10508   12.36756 0.06723445 0.4029932 0.06098967 -0.004348677        NA
Test set     1021.706233 1593.49076 1043.17479 6.36527288 6.5381924 5.14433527  0.963695384  6.242561
```

ARIMA (2,2,3)

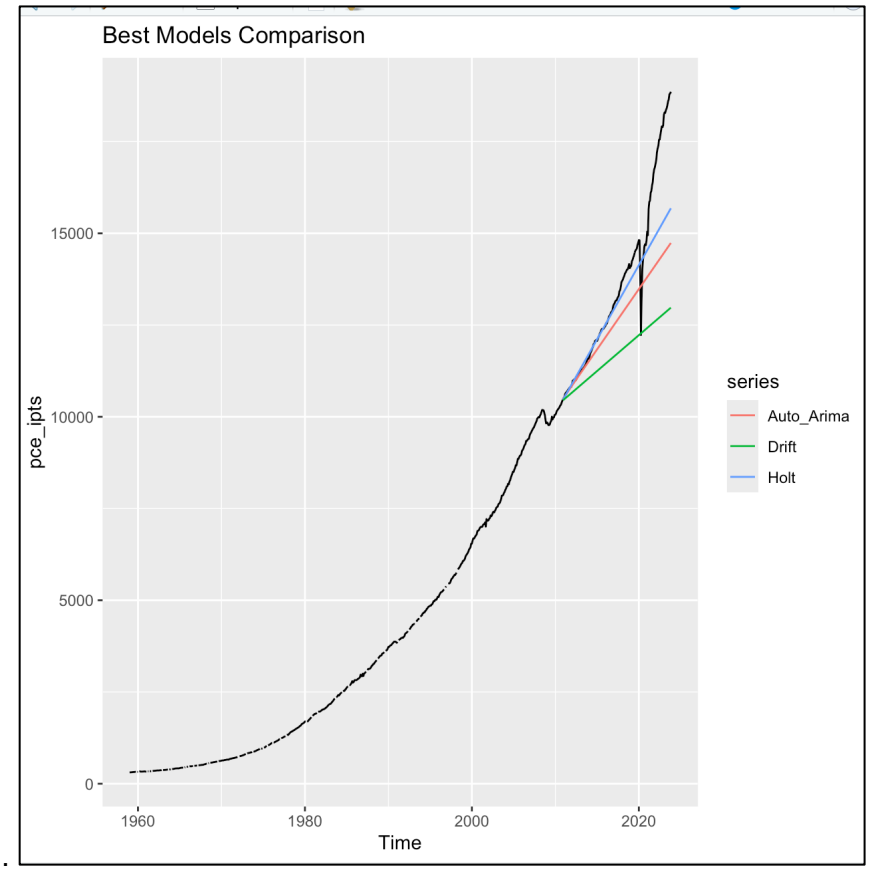
```
> accuracy(fc_arima_stand1,test)
                     ME       RMSE        MAE        MPE      MAPE       MASE         ACF1 Theil's U
Training set    1.242534   22.10563   12.35201 0.06692875 0.4030765 0.06091299 -0.004971711        NA
Test set     1029.789063 1601.21840 1050.94494 6.42142154 6.5918474 5.18265319  0.963765315  6.275768
```

ARIMA (2,2,2)

```
> accuracy(fc_arima_stand2,test)
                     ME       RMSE        MAE        MPE     MAPE       MASE        ACF1 Theil's U
Training set   1.283512   22.12531   12.34349 0.06713263 0.402185 0.06087095 -0.01185922        NA
Test set     998.272569 1571.46171 1020.86956 6.20197538 6.384153 5.03433877  0.96349488  6.147851
```



The plot above shows that ARIMA models are quite similar to each other, with little difference between them. However, in terms of accuracy, exponential smoothing, particularly Holt's Linear Method, outperforms ARIMA, which, in turn, is better than the drift model. A clearer difference is observed when comparing drift, Holt, and ARIMA models in the plot below.



5. Time Series Prediction

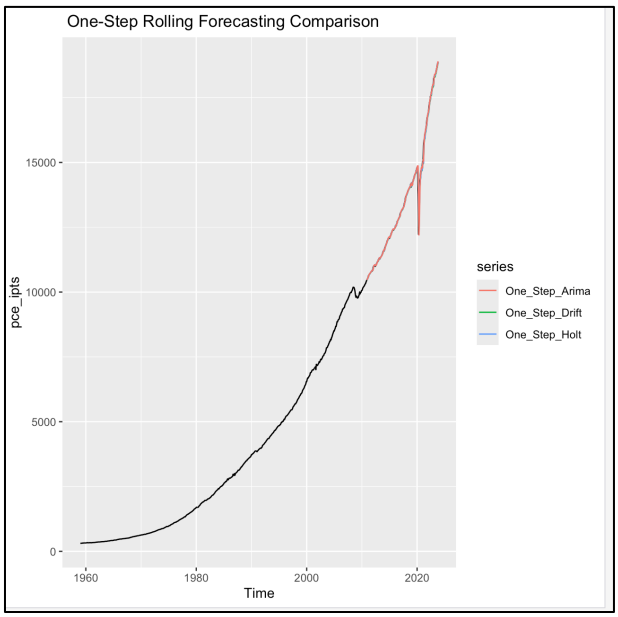
In this stage, we forecast the outcome for October 2024 by employing our preferred model, exponential smoothing utilizing Holt's Linear Method. The *holt* function, predicts from the last observation of the time series, November 2023 to 11 future steps and obtains value for the month of October 2024.

```
[1] "Estimation of the PCE expenditures for October 2024"
> predict_oct_24$mean[11]
[1] 19566.92
>
```

6. ONE STEP ROLLING FORECASTING WITHOUT RE-ESTIMATION

The method of generating forecasts for time series where the model estimates a single set of training data and then one-step forecasts are computed on the remaining data sets (Namin, et,al, 2018).The model generates a forecast for the next time step based on the observed value at the current time step, without considering any future observations. The *rolling* aspect refers to the fact that the model is applied sequentially, with the forecasting window moving forward one step at a time, that is the model predicts just one step ahead, and then updating the model with the actual result before predicting the next step.

When we perform one step rolling forecasting for the preferred methods, it is observed that they show very minimal/ no difference between each other when it comes to prediction. Among them Holt's linear model has slightly better predictions compared to drift and arima models.




```
[1] "Drift Model"
> accuracy(One_Step_Drift,test)
           ME RMSE  MAE   MPE  MAPE  ACF1 Theil's U
Test set 30.2  201 75.7 0.188 0.541 0.183     0.976
```

```
[1] "Holt's Linear Exponential Model"
> accuracy(One_Step_Holt,test)
          ME RMSE  MAE   MPE  MAPE  ACF1 Theil's U
Test set 17  200 69.4 0.101 0.501 0.174     0.974
```

```
[1] "Arima Model"
> accuracy(One_Step_Arima,test)
           ME RMSE  MAE    MPE  MAPE  ACF1 Theil's U
Test set 8.84  221 73.9 0.0496 0.536 0.262      1.09
```

PART 2

1. INTRODUCTION

The purpose of the task is to analyse customer online reviews for hotels and their corresponding ratings using topic modelling and text analytics and discuss the top three factors influencing customer satisfaction and dissatisfaction. Topic modelling is used to abstract the topics and themes present in the collection of documents and text analytics will be employed to analyse unstructured data to derive meaningful insights.

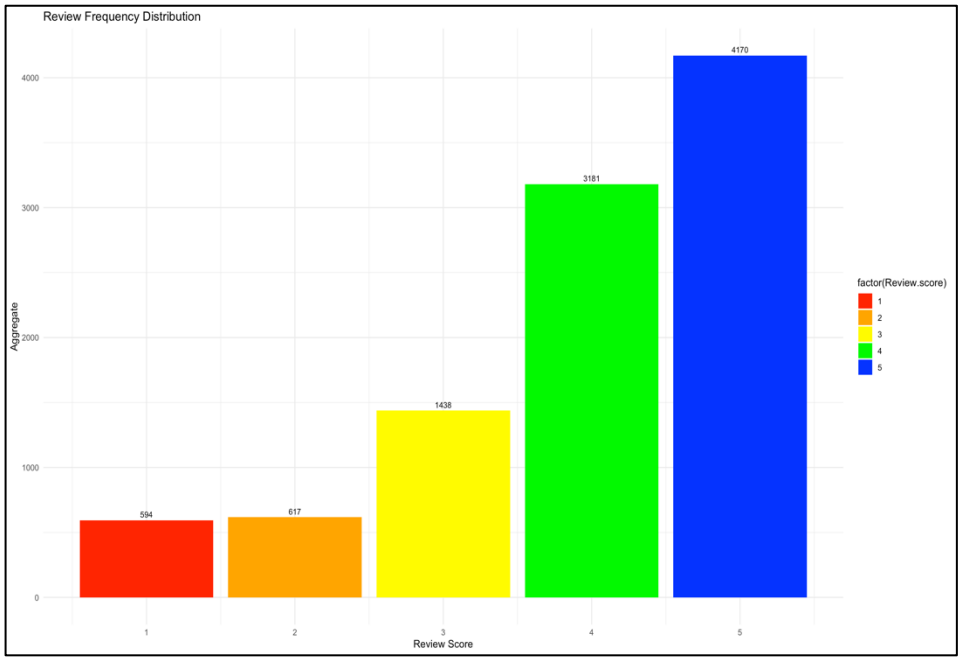
2. DATA UNDERSTANDING

We are provided with hotel data (***HotelsData.csv***), which contains two variables and 10,000 observations.

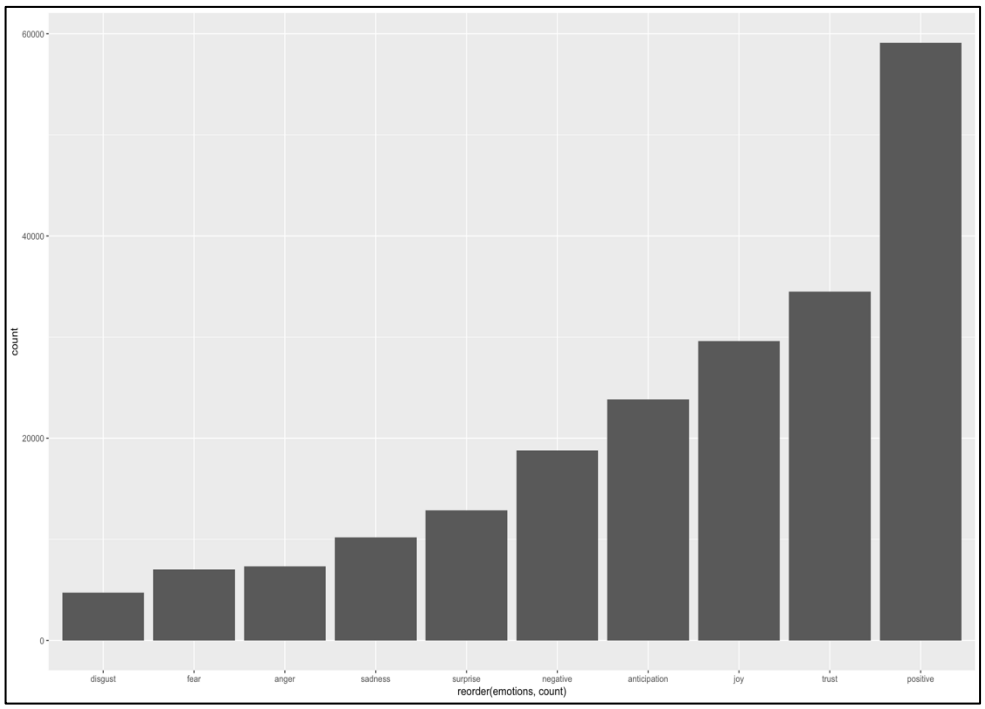
VARIABLE NAME	TYPE	DESCRIPTION
Review.Score	int	Hotel review ratings corresponding to Likert Scales (1/low to 5/high satisfaction)
Text.1	chr	Online reviews in text.

```
> str(hotel_data_raw)
'data.frame':   10000 obs. of  2 variables:
 $ Review.score: int  4 3 4 4 2 5 5 4 5 4 ...
 $ Text.1      : chr  "I last stayed at this hotel some time ago, and am pleased to note that the standards of
accommodation remain as"| __truncated__ "Booking in was a nightmare. I had a reservation but the staff had no
rooms available and after lots of running "| __truncated__ "questo è il 5 anno di fila che vado in questo hote
l in quanto vicino a dove faccio dei corsi . ques'anno di buo"| __truncated__ "The best hotel! Very clean, goo
d profetional staff" ...
```

Likert scales are widely used across different sectors to measure the satisfaction levels of consumers/users/stakeholders on a scale of 1 to 5. To understand the composition of rating distribution across the data set, we can visualise it as below plot. Out of 10,000, 4170 were 5-rated (marked blue), 3181 were 4-rated (marked green), 1438 as neutral – 3 rated (marked yellow) and 617 and 594 as 2 and 1 rated respectively (marked orange and red)



Sentiment analysis on the text was performed to understand its overall sentiment/emotion distribution. Sentiment Analysis is a methodology to analyse the sentiments of the texts expressed by others. We implement using the ***syuzhet*** package, leveraging the ***get_nrc_sentiment*** function.

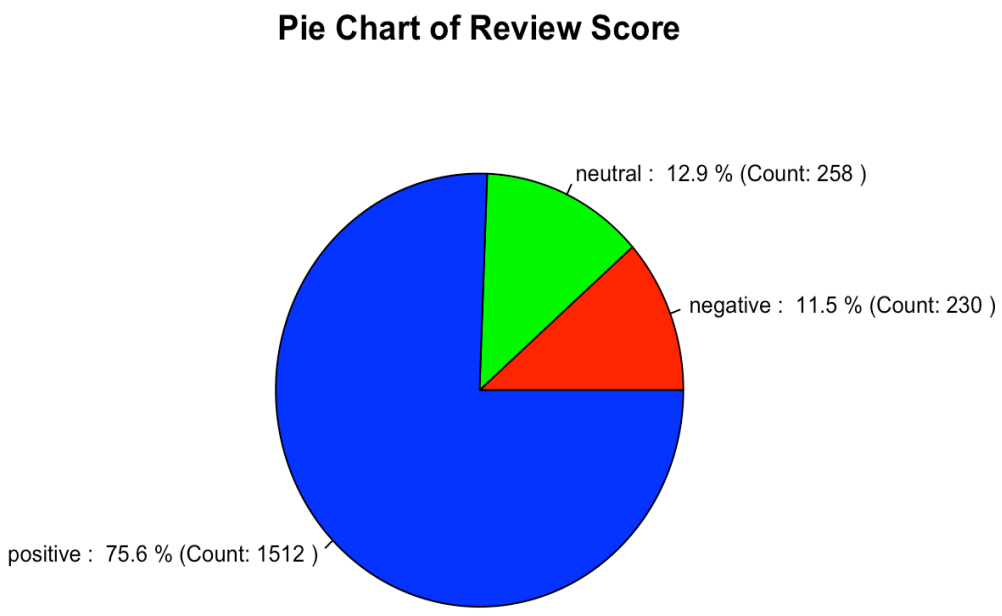


As shown in the above plot, it is observed that there are mostly positive sentiments forming the major texts, along with other emotions such as trust, joy, and anticipation. From the review plot, XXXX also we could see the dissatisfaction percentage is much lesser (12.11% - rated 1 and 2).

```
hotel_full_data <- cbind(hotel_data_raw, detected_languages = textcat(hotel_data_raw$Text.1))
hotel_data_english_indices <- hotel_full_data$detected_languages == "english"
hotel_english_data <- hotel_data_raw[hotel_data_english_indices, ]
```

After thorough analysis, it was observed that the **Text** variable contains review descriptions in languages other than English. To interpret the reviews more effectively, we utilize the **textcat** library to append a language column to the dataset. Subsequently, we extract a subset of the raw data comprising only English text, reducing the data sample to be 7690.

Now, we randomly select 2000 samples from the new data set, after setting a defined seed value. As shown below, proportion of review Scores of 5 and 4 constitutes more in the sample compared to negative reviews (1 and 2)



3. Data Preparation

This step involves preparing the data for further analysis. Next, we append an additional column to the new dataset, labelled **sentiment**. In this column, it is categorized as ***positive*** if its review score is either 4 or 5, ***negative*** if the score is 1 or 2, and any other score is labelled as ***neutral***.

```
#Add new column with negative and positive based on the review score
test <- test %>%
  mutate(sentiment = case_when(
    Review.score %in% c(4, 5) ~ "positive",
    Review.score %in% c(1, 2) ~ "negative",
    Review.score == 3 ~ "neutral"
  ))
```

Subsequently, we divide this dataset into two subsets: ***positive_reviews*** and ***negative_reviews***, based on the values in the newly created *sentiment* column. 1512 positive and 230 negative reviews in these separate data sets.

```
# Positive Reviews dataset
positive_reviews <- test %>%
  filter(sentiment == "positive")

# Negative Reviews dataset
negative_reviews <- test %>%
  filter(sentiment == "negative")
```

UTF-8 Encoding

UTF-8 encoding is done and a common practise, that encodes all the languages, special characters, emojis that is not handled by *tm* packages, thereby preserving the original characters and maintains data integrity.

Corpus Creation

This is the process of assembling a collection of text documents into a format that is suitable for analysis in Natural Language Processing (NLP) tasks. The format is later used for text analysis and topic modelling.

Document Term Matrix

DTM, is a structured way of representing textual data in a tabular format. Each row represents a document or text sample, each column represents a unique word found in

the corpus of documents and cell values in the matrix represent the frequency of occurrence of each term in the document. Creation of DTM involves several steps. *Tokenization*- splitting each document into individual words. *Cleaning*- removing stop words, punctuations, numbers, converting all texts to lower cases. *Lemmatization* – to normalise the words and reduce words to their base form, lemma. For example, cooked, cooking, cook would yield "cook". By reducing the words to lemma, it improves the accuracy of text analysis. *Final step of DTM* - counting the frequency of each term in each document and constructing the matrix.

FREQUENCY TABLE

This table returns information on how each word are used in the whole text. Most frequently used words in either of the data sets are **"hotel", "room", "stay".**
The top 10 frequent terms in the positive and negative data sets are as shown below.

Positive

```
> frequency[1:10]
   hotel     room    staff   london     good     stay breakfast    great
    2422     1903     1177     1030      945      891      826      819
location   stayed
     728      621
```

Negative

```
> frequency_neg[1:10]
    room    hotel      one    staff     stay   night breakfast     good
     569      455      148      138      138      131      126      119
     bed    rooms
     114      114
```

WORD CLOUD

This is a visual representation of text data, where size of each word indicates the frequency or importance within the text. Words are displayed in various fonts and sizes and most frequent words appear larger and prominent.

Word Cloud for Positive Reviews

Word Cloud for Negative Reviews

4. Topic Modelling

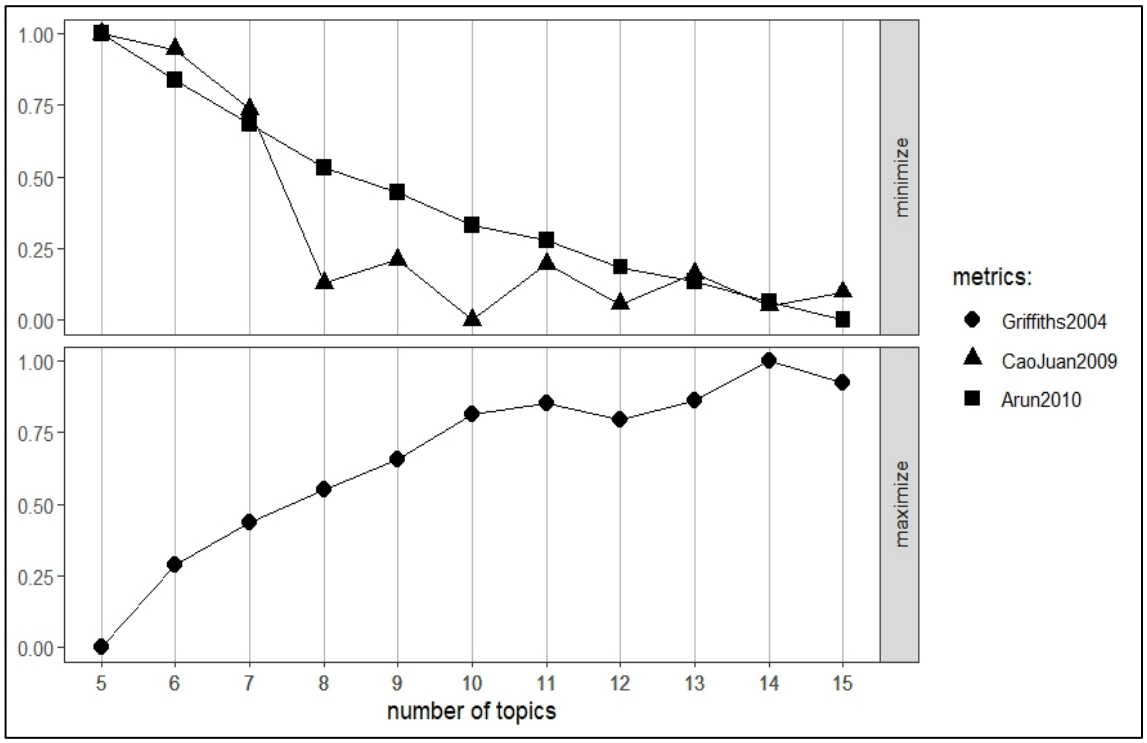
Topic modelling predicts the likelihood of a word being associated with a specific topic based on how often the words appear together in the documents (corpus). LDA (Latent Dirichlet Allocation) shall be employed, to perform topic modelling. **LDA** function, in topic modelling package will be used.

Number of topics

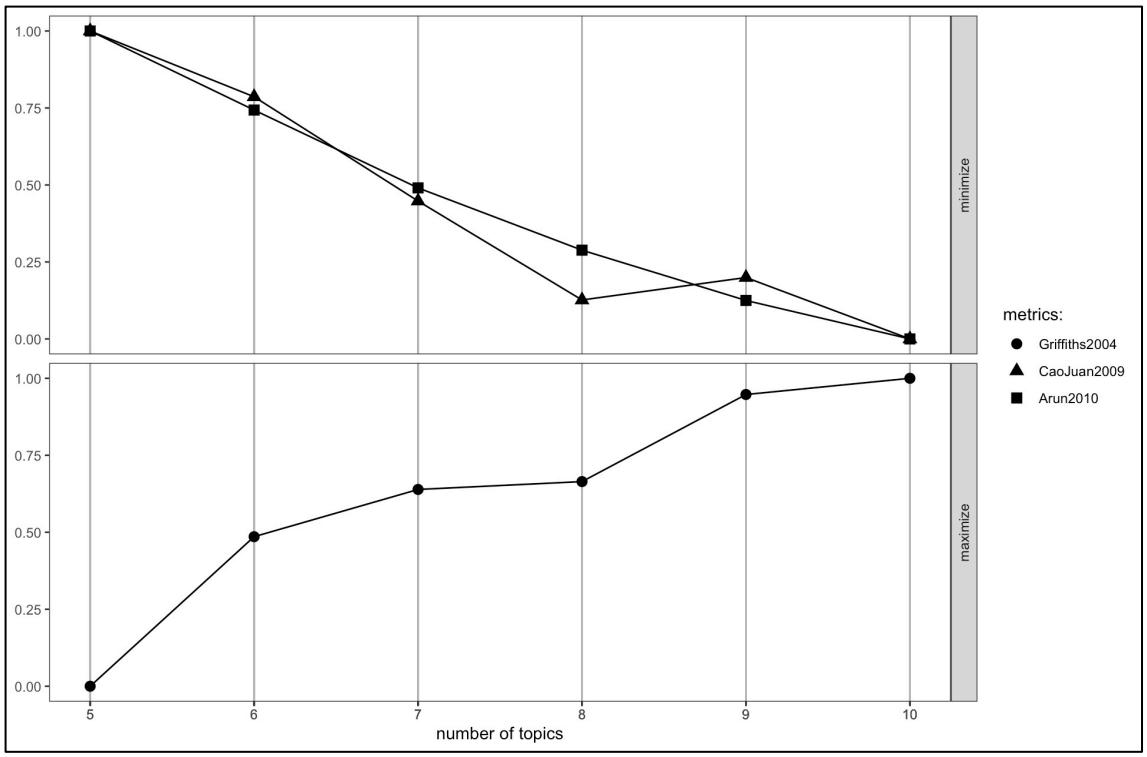
It is essential to get the number of topics (k) that LDA would create. **ldatuning,** helps us to decide a number of topics. We want to minimize the selected measures, *Arun2010* and *CaoJuan2009* and maximize the *Griffiths2004*, there is *Deveaud2014*, which is not considered for now. We choose among a range of k possible topics, and an optimal

number based on these criteria is selected. For both, we choose sequences from 5 to 15, due to their sample size and to get more specific topics.

For positive, k = 14 is chosen



For negative, k = 10 is chosen



LATENT DIRICHLET ALLOCATION

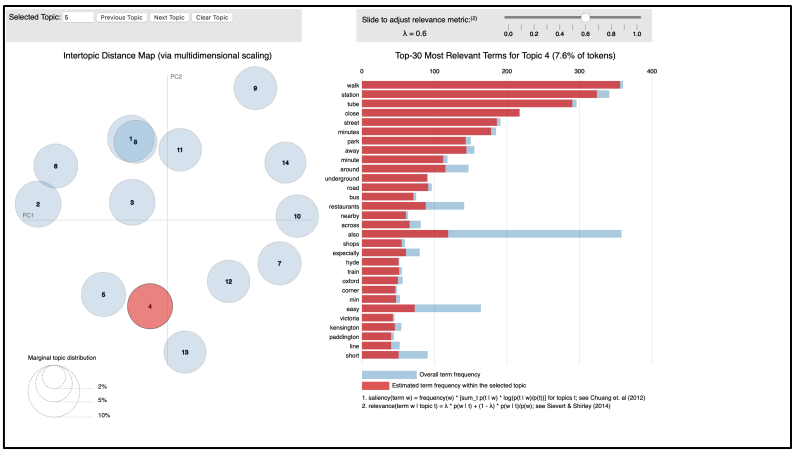
LDA, is a model used for topic modelling in NLP. This model discovers topics within a collection of documents by analysing the distribution of words across topics and distribution of topics across documents. In R, a function, LDA in topic modelling is called. A variable iter, specifies the iterations to find out the right model choice. A distribution over terms for a topic, phi and theta is the probability distribution over topics for a document. Using the command *terms(),* we could choose the top 10 terms of all the topics. LDA is done separately for positive and negative data sets. Numeric indices are assigned to *reviews* data to facilitate data manipulation. Subsequently, they are merged with LDA topic assignments based on a common index. The merged data frame is then sorted by index for organized analysis. Additionally, topic probabilities are extracted from the LDA output to understand the distribution of topics across documents. Finally, the vocabulary, representing terms used in the LDA model, is extracted from the topic-term distribution matrix.

A JSON object, using ***createJSON****,* containing LDA modelling results, is created, and the ***serVis*** function is used to generate visualization based on the JSON object. This visualization provides insights into the distribution of topics across documents and the relationship between terms and topics in the corpus.

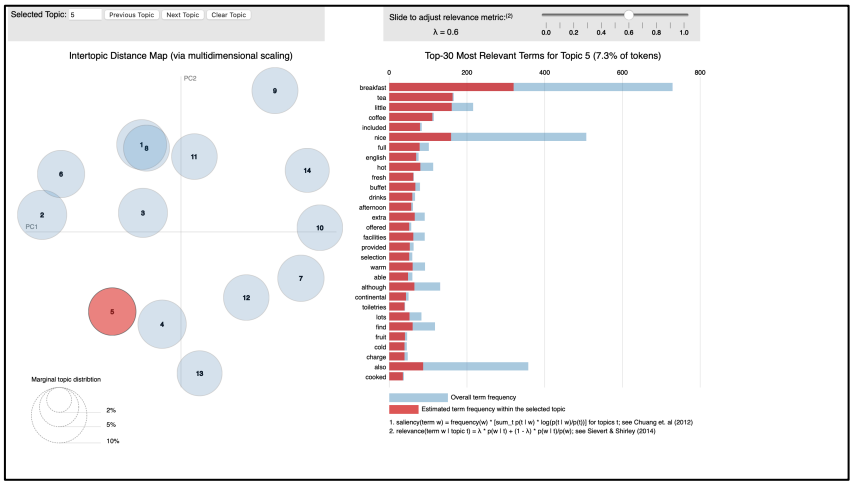
Labelled Topics for Positive Reviews

```
> topic_labels
 [1] "Great Comfort"            "Accomodation Inconvenience"  "Cuisine"                  "Accessibility to Local Hubs"
 [5] "Food"                     "Experience"                  "Convenient Location"      "Amazing Service"
 [9] "Cleanliness"              "Affordability"               "Amazing Amenities"        "Poor Customer Service"
[13] "Lodging Issues"           "Dining Excellence"
```

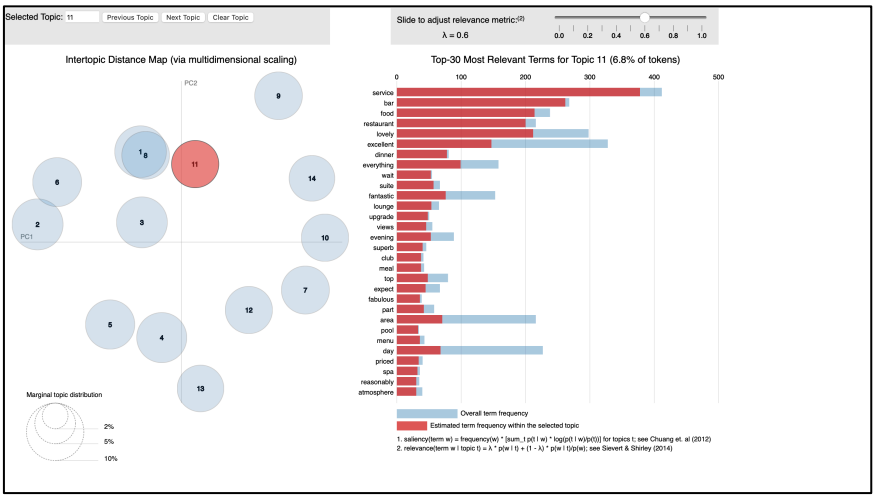
TOP THREE FACTORS FOR CUSTOMER SATISFACTION



Topic 4: Accessibility to Local Hubs. Visitors enjoyed the stay as they were assessable to nearby shops, underground and other landmarks.



Topic 5: Food. Guests enjoyed buffet, drinks, continental breakfast.

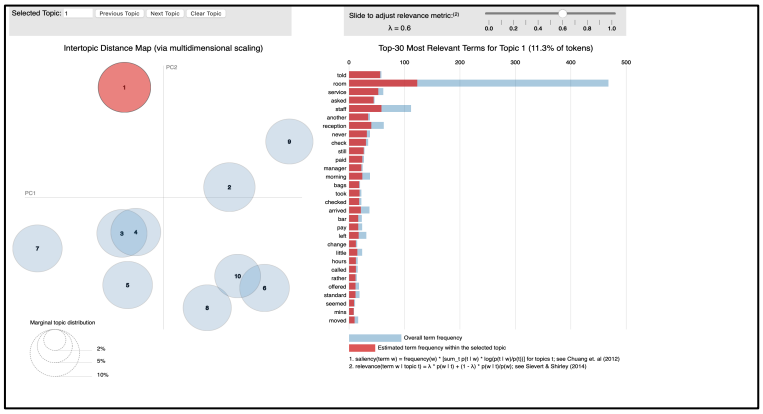


Topic 11: Amazing Amenities. Visitors enjoyed bar service, lounge, spa, dining, pool with excellent evening views.

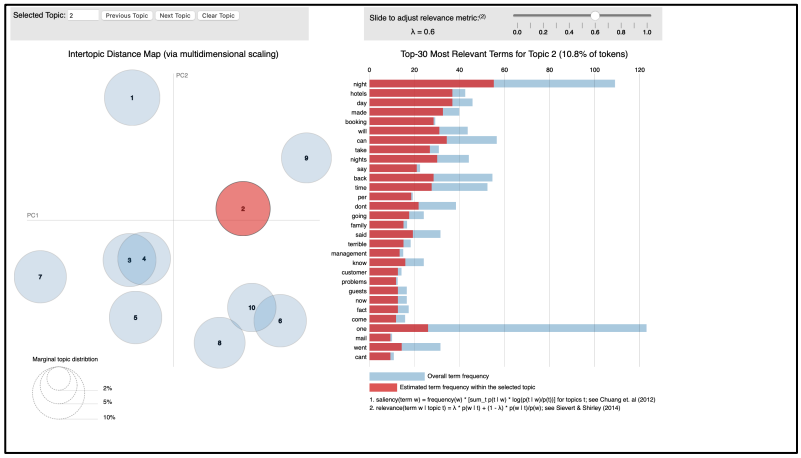
Labelled Topics for Negative Reviews

```
> topic_labels_neg
 [1] "Poor Room Service"           "Terrible Customer Management" "Pleasant Stay"
 [4] "Accessibility to Local Hubs" "Bad Rooms"                    "Substandard Rooms"
 [7] "Noisy"                       "Ventilation"                  "Not Worthy"
[10] "General Problem"
```

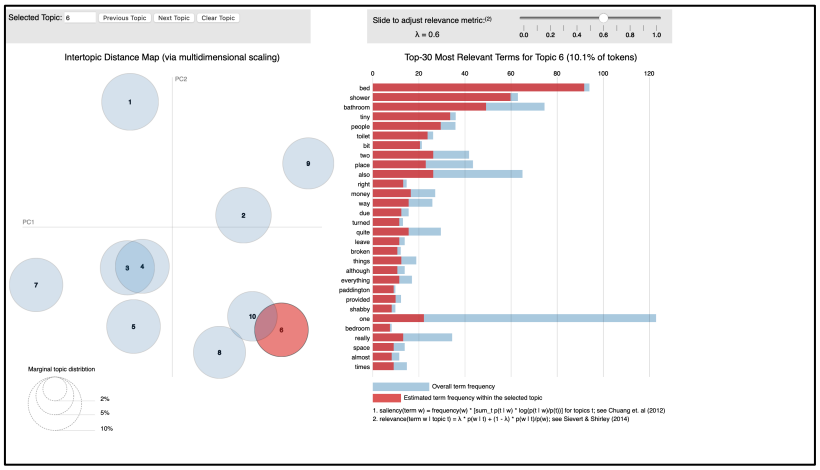
TOP THREE FACTORS FOR CUSTOMER DISSATISFACTION



Topic 1: Poor Room Service – Guests complained room service was below par, and they were not addressed well.



Topic 2: Terrible customer management – guests complained about the customer service and not understanding their problems.



Topic 6: Substandard Rooms. Guests complained about the complaints in the bed and bathrooms.

5. LIMITATIONS

- The division of positive and negative reviews was based solely on review scores led to a discrepancy where text content contradicted the assigned review for few records. This suggests that review scores might not be reliable for data segregation, to an extent.
- Positive topics had negative topics as well and vice versa.
- Meanwhile, the neutral dataset was available for text analysis, although sentiment analysis revealed a blend of emotions within it, hence could not assign to either of the reviews.
- With a larger dataset, the efficiency and accuracy of topic modelling could have been enhanced.
- Additionally, other potential limitations include issues related to data quality, such as incomplete or biased information. Cultural differences can impact the review scores and the text.

REFERENCES

Siami-Namini, S., Tavakoli, N. and Namin, A.S., 2018, December. A comparison of ARIMA and LSTM in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)* (pp. 1394-1401). IEEE.