New Delhi Weather Forecasting with Time Series Analysis

BACHELOR OF SCIENCE -PROJECT

MCA 2021-2024

-Debnarayan Bhunia

PROJECT REPORT

FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE IN STATISTICS

:: Project Topic ::

"NEW DELHI WEATHER FORECASTING WITH TIME SERIES ANALYSIS"

Session: 2021-24

Paper: STSDSE-4

:: Submitted by ::

Roll: MC/UG/SEM-VI/24 No.-1851

Reg. No.-2021-0069 of 2021-2022

-:: Certificate ::-

This is to certify that 6th Semester Student with Roll: MC/UG/S-VI/24 No.-1851, have satisfactorily completed the Project Report entitled "NEW DELHI WEATHER FORECASTING WITH TIME-SEIRES" in partial fulfilment of the requirement for the award of degree of Bachelor of Science in Statistics from Department of Statistics,

Midnapore College (Autonomous), Raja Bazar, West Bengal during academic year 2021-2024.

……………………………………….

Respected HOD of Department of Statistics,

Midnapore College (Autonomous)

::[ACKNOWLEDGEMENT]::

I would like to express my special thanks of gratitude to my teacher Rittik Hazra Sir, who gave me the golden opportunity to do this wonderful project of Statistics on **"TIME SERIES ANALYSIS"**, and helped me in completing my project. I have come to know about so many new things both theoretical and practical during completing this project, from him.

I would also like to extend my gratitude to our respected HOD Moumita Roy Ma'am, Professor Bodhoditya Barma Sir and Swikriti Kundu Ma'am, for providing me with all the facility that was required.

Also, I would like to thank my fellow classmates who provided insight and expertise that greatly assisted the project.

::[INDEX]::

Contents	Page No.

::[SUMMARY]::

The project aims to develop a weather forecasting model for New Delhi using time series analysis. The primary objective is to predict key weather parameters such as temperature, humidity, precipitation, wind speed and wind pressure. Accurate weather forecasts can help in various domains including agriculture, disaster management, and daily planning.

Frist, we did Data Collection, where 13 years Historical weather data for New Delhi is sourced from reliable datasets such as the Indian Meteorological Department (IMD) and Kaggle website. The primary variables include daily maximum, minimum and averages were given. We take this set in R-studio. Then for Handling missing values, outlier detection, and correction, data was checked, but no data cleaning was required. Some library function was added for better graphical plots and time-series analysis

Next, breaking down the time series into trend, seasonal, and residual components using methods like STL (Seasonal and Trend decomposition using Loess) was made. Also, Statistical tests for stationarity (e.g., Augmented Dickey-Fuller test) and ACF, PACF were calculated for model.

Then, our Model that is ARIMA (Auto-Regressive Integrated Moving Average) was created for Capturing the linear relationships in the time series and make a long-term forecast. Also, for a Cross-validation techniques such as rolling forecast origin and train-test on the given sample was examined.

In the output Results, we get one year (2024) long-term weather trends and seasonal patterns specific to New Delhi's forecasted value of Average Temperature and find correlations of it with other variables like humidity, precipitation, wind speed and wind pressure.

It has Potential applications in agriculture, urban planning, and disaster preparedness. This project will contribute to more accurate and reliable weather forecasting, aiding in better decision-making and resource management for New Delhi's residents and authorities.

::[Introduction]::

I. About the Project

New-Delhi, the Capital of India, experiences a diverse range of weather conditions throughout the year including hot summers, monsoon rains, and cold winters. The project seeks to leverage advanced weather forecasting techniques and technologies to improve the accuracy of weather prediction and provide valuable information to residents, businesses and policy-markets.

This Project is made on Weather Forecasting of New Delhi for the Year 2024, in the basis of past 13 Years data, with the help of time series in R-Studio. The basic analysis by descriptive statistics, and also the time series analyzed results are given by R.

II. Literature

This particular project topic is also made by different data set of weather of different places. Some of those popular works are done by some Indian and Foreign Data Analysts, which are also provided in websites like Kaggle, Git-hub etc.

III. Work done in different section:

Further in 'Model Motivation' section the project idea, how did it come, is discussed briefly, in 'Methodology' section the steps and procedure are told,

In 'Data Analysis' the procedure and whole weather data is analyzed and collusion are drawn in 'Conclusion' section, and 'Scope', 'Reference' and 'Appendix' are given in the End.

::[Model Motivation]::

I am glad to say that I was highly interested by the Paper Time Series Analysis. It's the just the beginning of semester V, when we have gone through this paper and learnt theoretically different application of Time Series such as Weather Forecasting, Rainfall Forecasting, Stock Price, Automated Stock Tarding, Industry Forecasting, Business Cycle etc. different fields.

Since then, I was very much interested in the Forecast with the Help of Trend. Nowadays 'Trend' is a most common word in our daily life, as well as in Stock markets.

In our daily life, we all check the future Weather forecast, which is very important to know, so that we can stay alert in advance. So, we all are partially dependent on weather forecast for our incoming journey, home errands, and many others things to do.

Therefore, I selected this great Topic "Weather Forecasting by Time Series Analysis".

-: Data Description :-

The dataset includes a variety of meteorological variables recorded for each day from 2010 to 2023, such as temperature, humidity, wind speed, atmospheric pressure, and precipitation of New Delhi. Though in this weather data, all values are imaginary and made for any project work. Still it is collected from a reliable source of 'Kaggle' website.

Here's a quick look at the columns:

- Date, Year, Month of the recorded weather data
- Max Temperature: Maximum temperature (°C) ,
- Avg. Temperature: Average temperature (°C),
- Min Temperature: Minimum temperature (°C),
- Max Dew Point: Maximum dew point (°C),
- Avg. Dew Point: Average dew point (°C),
- Min Dew Point: Minimum dew point (°C),
- Max Humidity: Maximum relative humidity (%),
- Avg. Humidity: Average relative humidity (%),
- Min Humidity: Minimum relative humidity (%),
- Max Wind Speed: Maximum wind speed (km/h),
- Avg. Wind Speed: Average wind speed (km/h),
- Min Wind Speed: Minimum wind speed (km/h)
- Max Pressure: Maximum atmospheric pressure (hPa),
- Avg. Pressure: Average atmospheric pressure (hPa),
- Min Pressure: Minimum atmospheric pressure (hPa)

The data is a Time Series data since variables are measured over different date, month and years. The total data is from 2010 to year 2023. So time series analysis is used on the data to analyze.

-: Methodology :-

The project is made through some important steps below:

- **Graphical Plot:** First plot graphically the huge data by variable 'Daily Avg. Temperature' with Heat Map and boxplot, and visually understand the data.

- **ARIMA (Auto Regressive Integrated Moving Average):** This model is commonly used for time series forecasting. It combines the concepts of autoregression (AR), Integrated(I), and moving average (MA) to capture different aspects of the time series data.

Fitting an ARIMA model involves selecting the appropriate parameters (p, d, q) for the autoregressive (AR), differencing (I), and moving average (MA) components of the model. Here's a general outline of how you can fit an ARIMA model:

1. **Visualize the Time Series:** Plot the time series data to understand its patterns, trends, and seasonality.

2. **Stationarize the Series:** ARIMA models require the time series to be stationary, meaning its mean, variance, and autocorrelation structure do not change over time. If your series is not stationary, you may need to difference it until it becomes stationary.

3. Identify Parameters: Use autocorrelation function (ACF) and partial autocorrelation function (PACF) plots to identify potential values of q and p. The ACF plot shows the correlation between a series and its lagged values, while the PACF plot shows the correlation between a series and its lagged values, excluding the effects of intervening lags.

4. Fit the Model: Use the identified values of p, d, and q to fit the ARIMA model to the data. This can be done using software R's **forecast** package.

5. Evaluate the Model: Use diagnostic tools like residual plots, ACF plots of residuals, and statistical tests to evaluate the fit of the model. Adjust the model if necessary.

6. Forecast: Once you have a satisfactory model, you can use it to forecast future values of the time series.

➢ Data Read in R:

After the CSV file inputting in R, we got the head function of data:

```
> #fileread
> data<-read.csv(file="C:/Users/debna_d1u9lr5/Desktop/6th sem project/NewD
elhiWeatherInsights(2010-2023).csv",head=TRUE)
> head(data)
```

```
  Date Year Month Max.Temperature Avg.Temperature Min.Temperature
1    1 2010     1              70            55.7              45
2    2 2010     1              59            50.6              46
3    3 2010     1              57            52.8              48
4    4 2010     1              55            51.0              48
5    5 2010     1              63            53.1              46
6    6 2010     1              64            54.4              45
  Max.Dew.Point Avg.Dew.Point Min.Dew.Point Max.Humidity Avg.Humidity
1            52          47.5            41          100         77.5
2            54          49.6            46          100         96.7
3            57          52.1            48          100         97.3
4            50          48.4            48          100         91.9
5            52          47.9            46          100         85.1
6            55          49.7            45          100         85.7
  Min.Humidity Max.Wind.Speed Avg.Wind.Speed Min.Wind.Speed Max.Pressure
1           43              8            4.9              0         29.2
2           77              8            4.8              3         29.3
3           82             12            7.4              3         29.3
4           77             15           11.0              7         29.3
5           55             14            5.7              0         29.2
6           56              8            2.5              0         29.2
  Avg.Pressure Min.Pressure
1         29.2         29.1
2         29.3         29.2
3         29.2         29.2
4         29.2         29.1
5         29.1         29.1
6         29.2         29.1
```

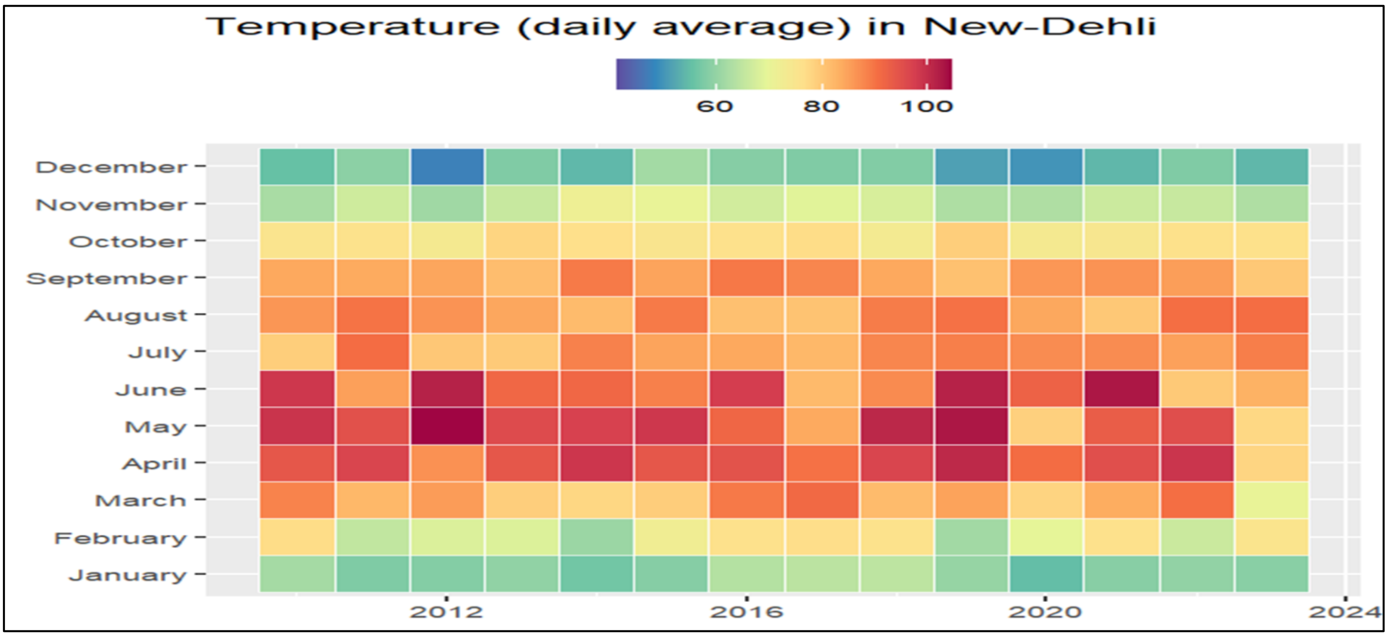
-: Data Analysis :-

1.Data Visualization: The given data set can be visualized by some descriptive statistics plots, for better understanding it graphically:

(i)Heat Map (daily average temperature):

A heatmap of daily average temperature visually represents temperature variations across a geographic area over a specific time period. It uses color gradients to show different temperature ranges, with warmer temperatures depicted in shades of red or orange, and cooler temperatures in shades of blue or green. This graphical representation helps to quickly identify temperature patterns, such as hotspots or cold areas, providing insights into spatial temperature distribution within a region.

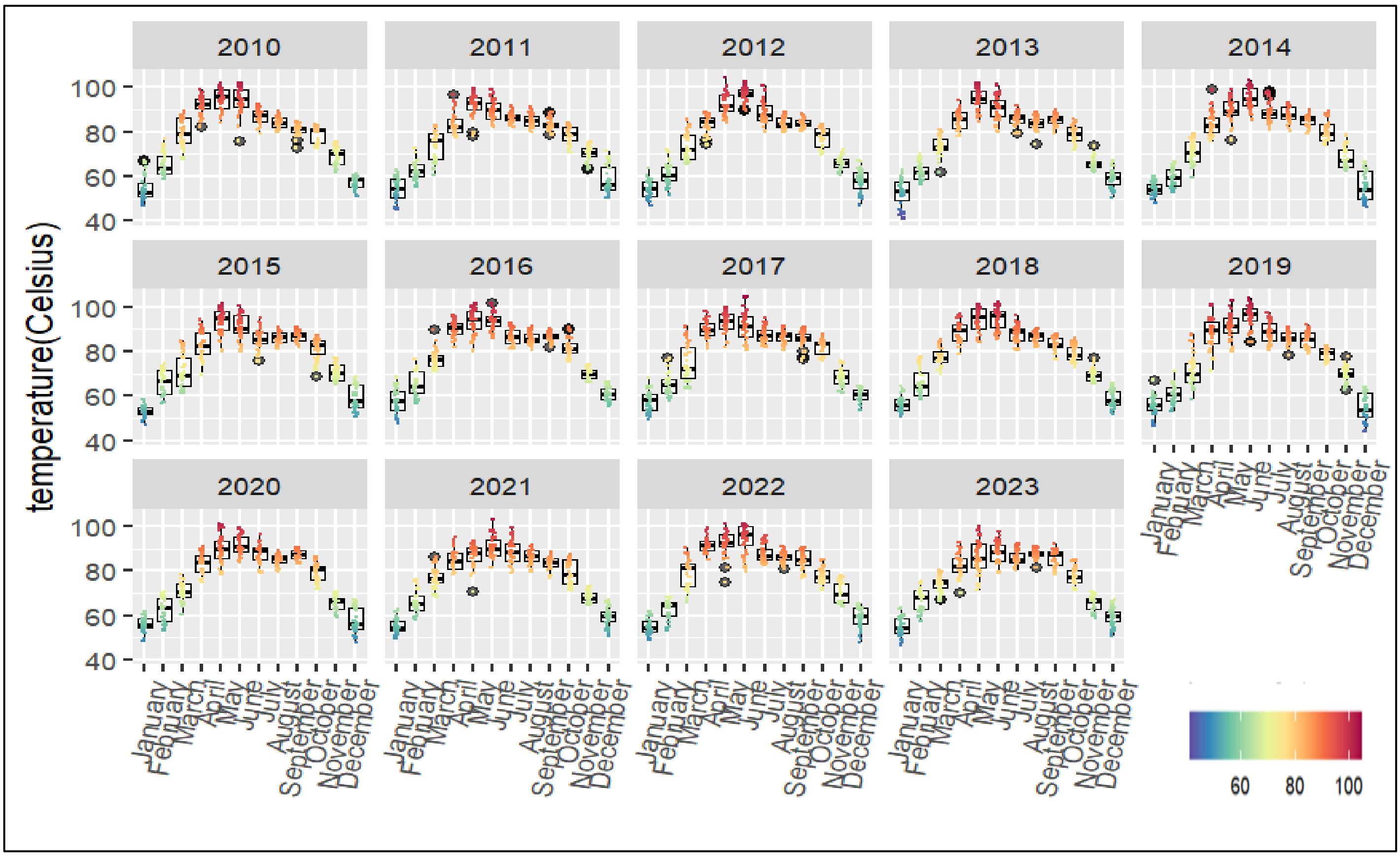
```
> ggplot(data=df1,aes(x=Year,y=ordered_month))+
+   geom_tile(aes(fill=dailyTemp),colour = "white") +
 scale_fill_gradientn(colours=rev(brewer.pal(10,'Spectral')))+
+   theme(legend.title=element_blank(),axis.title.y=element_blank(),
axis.title.x=element_blank(),
legend.position="top") + ggtitle("Temperature (daily average) in New-Dehli")
```



Heat Map of Daily Average Temperature (°C) in New Delhi

(ii)Box Plot (daily average temperature): A boxplot is a graphical summary of the distribution of a dataset. It displays key statistical measures such as the median, quartiles, and potential outliers in a compact format. The plot consists of a box, which represents the interquartile range (IQR) where the middle 50% of the data lies, with a line inside marking the median. Whiskers extend from the box to indicate the range of the data, excluding outliers, which are plotted individually as points beyond the whiskers. Boxplots are useful for comparing distributions across different groups or identifying unusual data points within a dataset.

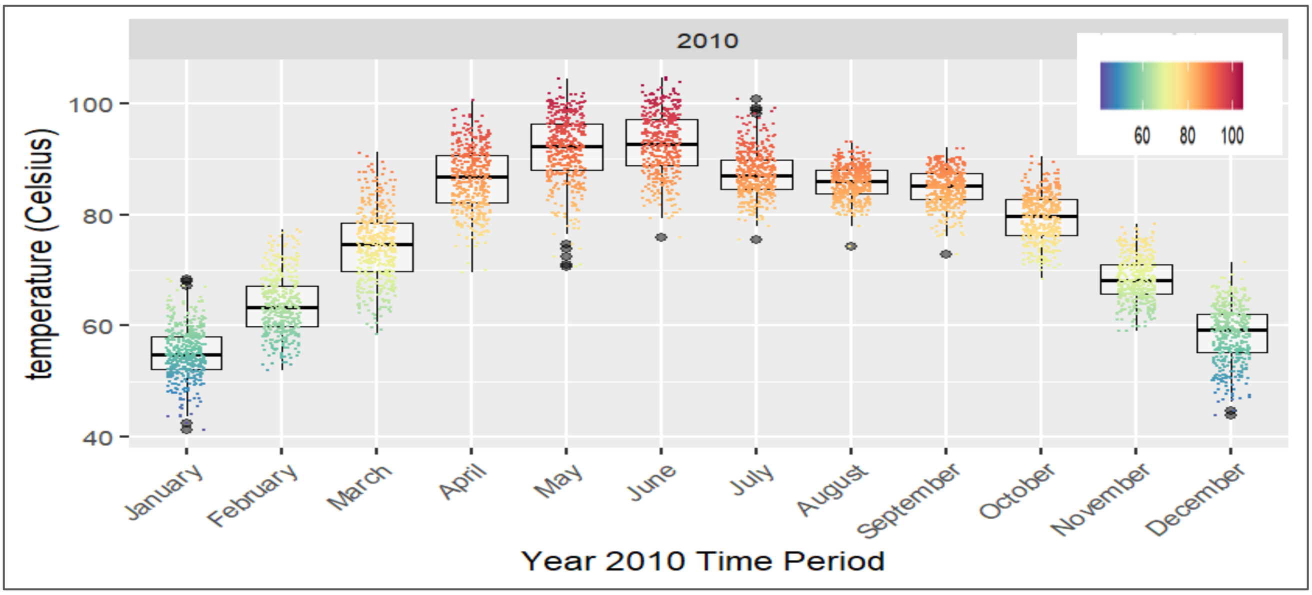
```
> #boxplot
> ggplot(data=df1,aes(x=ordered_month,y=dailyTemp,color=dailyTemp)) +
+   scale_color_gradientn(colours=rev(brewer.pal(10,'Spectral'))) +
+   geom_boxplot(colour='black',size=.4,alpha=.5) +
+   geom_jitter(shape=16,position=position_jitter(0.2),size=.4) +
+   facet_wrap(~factor(Year),ncol=7) +
+   theme(legend.position='none',axis.text.x = element_text(angle=45, hjust=1)) +
+   xlab('') + ylab('temperature (Celsius)')

>
```



Box Plot of Daily Average Temperature (°C) of New Delhi

(iii)Box Plot(monthly): Since the daily average temperature gives very hazardous visuals throughout all the years, so we make a single year boxplot monthly. Let us take year 2010. The difference here (compared to above) is that the mean is taken over the data within a month, not within a day.

```
#ggpllotforoneyear
> year1<-c("2010")
> df1$Year1<-year1[df1$Year]
> ggplot(data=df1,aes(x=ordered_month,y=dailyTemp,color=dailyTemp)) +
+   scale_color_gradientn(colours=rev(brewer.pal(10,'Spectral'))) +
+   geom_boxplot(colour='black',size=.4,alpha=.5) +
+   geom_jitter(shape=20,position=position_jitter(0.2),size=.4) +
+   facet_wrap(~factor(year1)) +
+   theme(legend.position='none',axis.text.x = element_text(angle=45, hjust=1)) +
+   xlab('Year 2010 Time Period ') + ylab('temperature (Celsius)')
```



Box Plot on Monthly Avg. Temperature((°C) of New Delhi

Conclusion: Both the Heatmap and Boxplot shows the same seasonality i.e. Temperatures are higher in the summer (although it seems the max. temperature is reached around May/June).

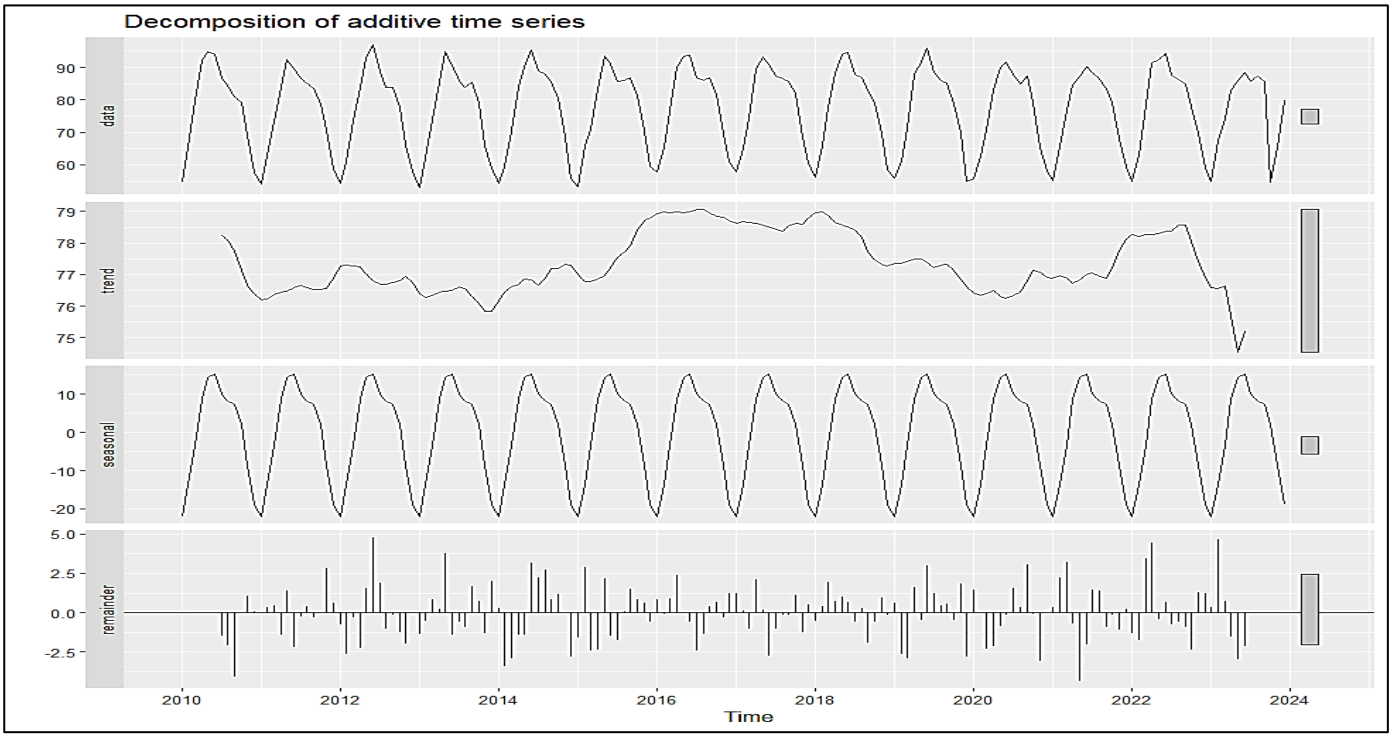
We can't say only by eye observation a potential increase of this behaviour vs. Year; 13 years data is not enough to notice a global increase 'warming' in New Delhi.

2.Time Series Analysis:

I. Time Series Decomposition: Time series decomposition is a statistical method used to break down a time series dataset into its underlying components: trend, seasonal variation, and residual (or noise). Our given time series data for Monthly Average Temperature data is decomposed below:

1. **Trend**: The long-term movement or directionality of the data over time. It represents the overall pattern of growth or decline in the series.
2. **Seasonal Variation**: The periodic fluctuations in the data that occur at fixed intervals (e.g., daily, weekly, monthly). Seasonality reflects the regular and predictable patterns tied to calendar or seasonal factors.
3. **Residual (or Noise)**: The random variation or irregular component left after removing the trend and seasonal effects. It captures unpredictable factors affecting the time series that are not accounted for by trend or seasonality.

```
#timeseries
>library(forecast)
> df2<-as.data.frame(data %>% select(Year,Month,Avg.Temperature) %>% group_by(Year
,Month) %>% summarise(monthlyTemp = mean(Avg.Temperature)))
`summarise()` has grouped output by 'Year'. You can override using the `.groups`
argument.
> myts<-ts(df2$monthlyTemp,start=c(2010,1),end=c(2023,12),frequency=12)
> autoplot(decompose(myts))
```

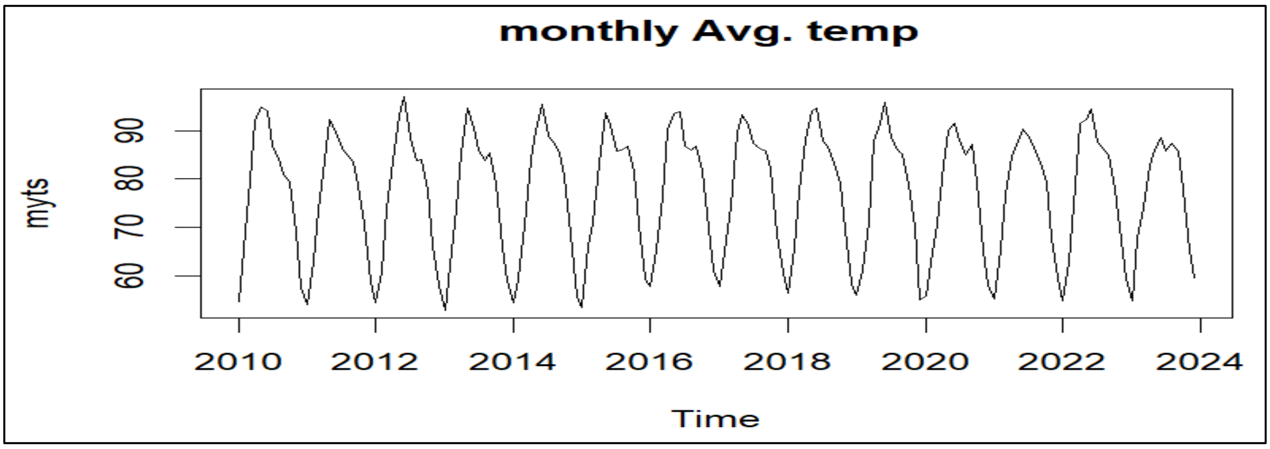


Decomposition of Additive Time Series Model of Monthly Avg. Temperature (°C)

II. Seasonal Time Series Data Extraction:

Now from the decomposed time series data we took only Seasonally decomposed data for further use of models and forecasting.

```
> #seasonaldecomposition
> plot(myts,main="monthly Avg. temp ")
```



III. Stationarity check with Augmented Dickey- Fuller Test

The Augmented Dickey-Fuller (ADF) test is a statistical test used to determine if a time series dataset is stationary or non-stationary. Stationarity refers to a series having constant statistical properties over time, including a stable mean and variance.

The "augmented" version of the test includes additional lagged terms in the regression equation to account for possible autocorrelation and improve the accuracy of the test. The ADF test is commonly used in econometrics and time series analysis to assess the stationarity of data before applying statistical models like ARIMA (Auto-Regressive Integrated Moving Average) for forecasting.

Unit Root: The test examines whether a unit root is present in the data. A unit root indicates non-stationarity, implying that the series exhibits trends or cycles that can lead to unreliable statistical conclusions.

Null Hypothesis: The null hypothesis of the ADF test is that the series has a unit root, meaning it is non-stationary. Rejection of the null hypothesis suggests the series is stationary.

Test Statistic: The ADF test calculates a test statistic that compares the amount of autocorrelation in the series to that expected under the null hypothesis of a unit root. The significance of the test statistic is compared against critical values from statistical tables to determine if the null hypothesis should be rejected.

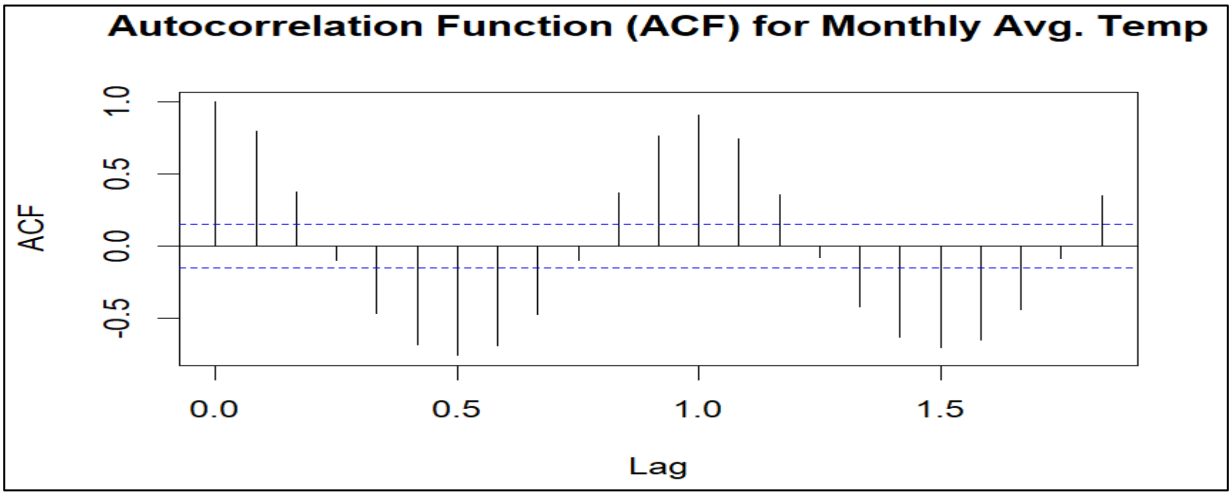
```
> #sationaritycheck
> install.packages(
"tseries")

> install.packages(
"tseries")
> adf_results<-adf.
test(myts)
```

```
> adf_results

        Augmented Dickey-Fuller Test

data:  myts
Dickey-Fuller = -13.461, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```

Conclusion: Hence, the data is stationary since p-value=0.01<0.05, we use ARIMA model for forecasting and for that calculate ACF and PACF.

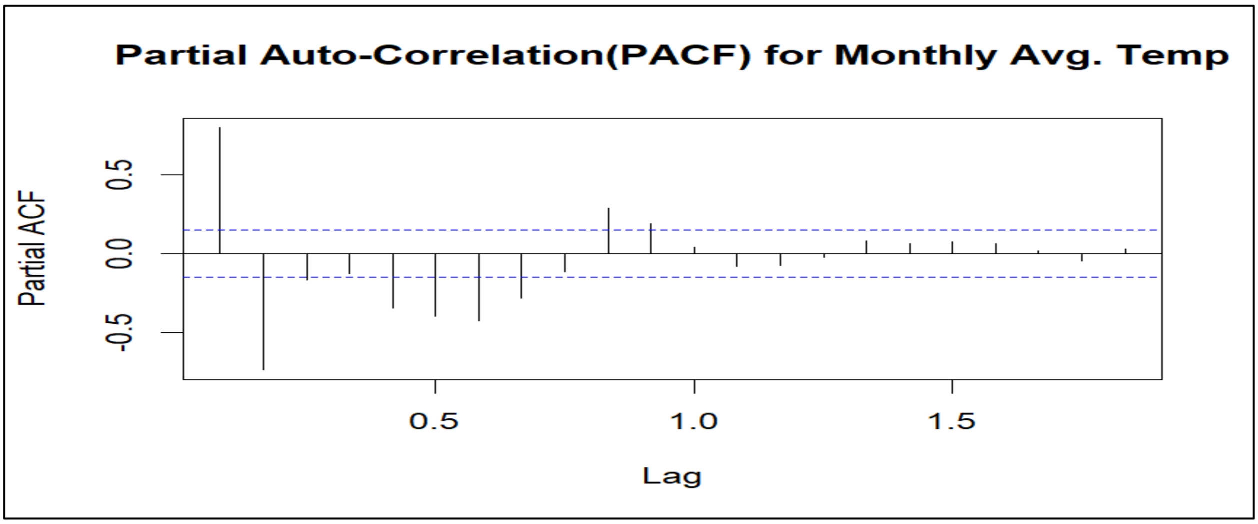
III. ACF Check: The ACF measures the correlation between a series and its lagged values. It helps identify the order of the Moving Average (MA) component (q) in an ARIMA model. Significant autocorrelations at higher lags suggest a non-zero MA order.

```
#acf&pacf_check
> acf(myts, main = "Autocorrelation Function (ACF) for Monthly Avg. Temp")
> pacf(myts,main="Partial Auto-Correlation(PACF) for Monthly Avg. Temp")
```



conclusion: Here the significant spike is at lag 1 in ACF plot, this suggests MA (1) model i.e. q=1.

IV. PACF check: The PACF measures the correlation between a series and its lagged values, removing the effect of intermediate lags. It helps identify the order of the Auto-Regressive (AR) component (p) in an ARIMA model. Significant partial autocorrelations at higher lags suggest a non-zero AR order.

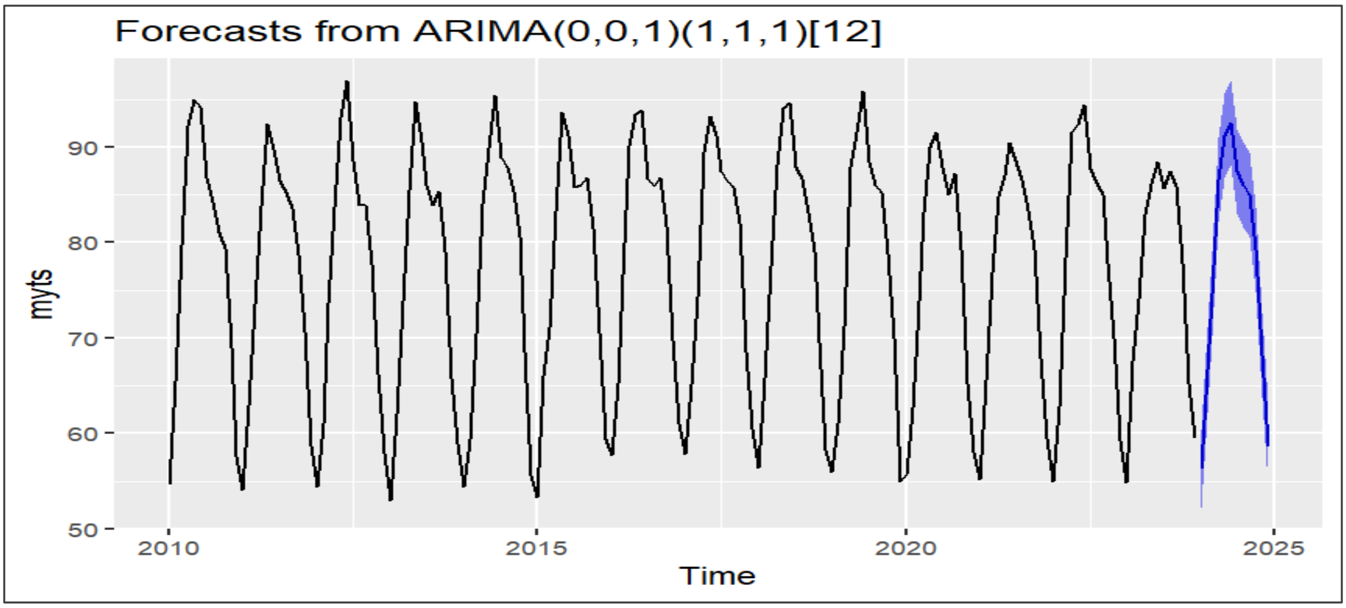


conclusion: Here the significant spike is at lag 1 in PACF plot, this suggests AR (1) model i.e. p=1.

V. ARIMA (1,1,1) Fitting:

By the above procedure mentioned in 'Methodology' we fit the ARIMA (1,1,1) model to the time series data and get the result below with a forecasted Blue Line shaded area for year 2024.Which shows a possible pattern curve, what the temperature of the year Jan,2024 to Dec,2024 would possibly occur.

```
> #arima
> d.arima <-auto.arima(myts)
d.forecast <-forecast(d.arima,level=c(95),h=12)
autoplot(d.forecast)
```



Auto-Arima prediction Year 2024 over 12 months

Then get the summary of forecast in R with the command: `> summary(d.forecast)`

VI. summary(forecast):

```
Forecast method: ARIMA(0,0,1)(1,1,1)[12]

Model Information:
Series: myts
ARIMA(0,0,1)(1,1,1)[12]

Coefficients:
         ma1     sar1     sma1
      0.4433  -0.0435  -0.8567
s.e.  0.0714   0.1095   0.1102

sigma^2 = 4.274:  log likelihood = -341.55
AIC=691.09   AICc=691.36   BIC=703.29

Error measures:
                       ME     RMSE      MAE         MPE     MAPE      MASE
Training set -0.02607379 1.972834 1.557079 -0.06046625 2.066159 0.7406791
                   ACF1
Training set 0.03837686
```

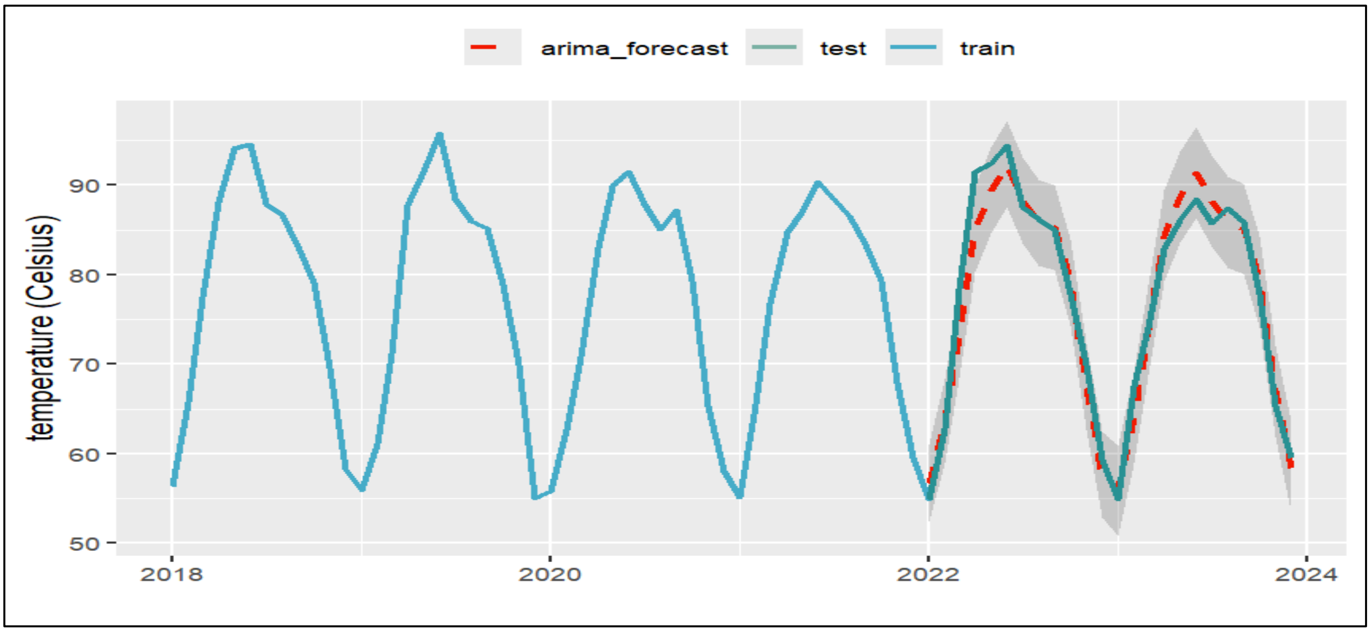
```
Forecasts:
     Point        Forecast    Lo 95    Hi 95
Jan 2024          56.27784 52.21825 60.33744
Feb 2024          63.97855 59.53840 68.41869
Mar 2024          74.65307 70.21293 79.09322
Apr 2024          86.54915 82.10900 90.98929
May 2024          91.19727 86.75712 95.63742
Jun 2024          92.54463 88.10448 96.98477
Jul 2024          87.43887 82.99873 91.87902
Aug 2024          86.03982 81.59968 90.47997
Sep 2024          85.03061 80.59047 89.47076
Oct 2024          79.28753 74.84738 83.72767
Nov 2024          68.24838 63.80823 72.68853
Dec 2024          58.60252 54.16273 63.04231
```

VI. Test the Accuracy of forecasting of Model for Year Interval 2023-24: Because we have data over 13 years, we can split them into a train/test sample:

Train sample: year 2010 to 2021 and Test sample: year 2022-2023

Run ARIMA prediction over train sample and compare with (true) test data

```
> #select train/test sample
> train <- df3 %>% dplyr::filter(Year<2024)
> test <- df3 %>% dplyr::filter(Year>=2022)
> train_ts <- ts(train$monthlyTemp,start=c(2010,1), end=c(2021,12), frequency=12)
> test_ts <- ts(test$monthlyTemp,start=c(2022,1), end=c(2023,12), frequency=24)
```



ARIMA Forecasting of Year 2022 to 2023 by Train and Testing Sample

Forecast Results: From the above test we see Auto-Arima(red) tends to underestimate the true data (green). Hence, we can say our ARIMA model almost accurate forecast for Temperature of year 2022 and 2023. Therefore, our previous ARIMA (1,1,1) model also predicted most possible and accurate temperatures for the whole year 2024.

VI. Correlation with Other variables

Though there are some other remaining variables like Wind Speed, Wind pressure, Dew Points and Humidity. If we can find similarly the forecasted values for these weather variables for Year 2024 with the help of ARIMA model.

Then we can correlate these variables with each other like (Temperature, Humidity), (Temperature, Wind Pressure), (Wind Speed, Wind Pressure),

(Dew Points, Humidity) etc. Thus, we can get total forecasted possible Weather data for the whole Year 2024.

-: Conclusion :-

In New Delhi Weather historical data analysis, we get some basic key conclusions like below:

- In 'Data Visualisation' Both the Heatmap and Boxplot shows the same seasonality i.e. Temperatures are higher in the summer (although it seems the max. temperature is reached around May/June). We can't say only by eye observation a potential increase of this behaviour vs. Year; 13 years data is not enough to notice a global increase 'warming' in New Delhi.
- After 'time series decomposition' we get the seasonal and trend of data, there we check the data stationarity by ADF test and got it Stationary.
- After fitting ARIMA (1,1,1) model we forecast the monthly average temperature of the Year 2024 and got the forecasted values.
- To check the ARIMA model accuracy we train it with data of year 2010 to 2021 and then test forecast of year 2022 to 2023. There we see Auto-Arima(red) tends to underestimate the true data (green). Hence, we can say our ARIMA model almost accurate forecast for Temperature of year 2022 and 2023.
- Therefore, our previous ARIMA (1,1,1) model also predicted most possible and accurate temperatures for the whole year 2024.
- If we also use the forecast model after stationarity checking of the other remainder variable like Humidity, Wind Speed, Wind Pressure and Dew points, we also got some forecasted values for the year 2024. And we could find some correlation between these variables.

-: Scope :-

In essence, time series analysis expands the scope of weather prediction by providing robust methodologies to analyse historical data, identify trends and patterns, and forecast future weather conditions with increasing accuracy and reliability. These forecasts support a wide range of applications, from daily weather updates to strategic planning for climate change adaptation and mitigation efforts.

- **Seasonal and Annual Variations**: Time series analysis helps in understanding seasonal variations in weather patterns, such as temperature fluctuations across different seasons or annual cycles in rainfall.
- **Extreme Event Prediction**: Identifying and predicting extreme weather events, such as hurricanes, heatwaves, or heavy rainfall, is crucial for disaster preparedness and risk management. Time series models can detect patterns indicative of such events based on historical data.
- **Short-Term Forecasting**: Time series analysis allows for the prediction of weather conditions in the near future, typically from hours to a few days ahead. Techniques like ARIMA models or exponential smoothing can capture short-term trends and patterns in weather data.
- **Data-Driven Decision Making**: Weather forecasts derived from time series analysis support decision-making in various sectors, including agriculture, transportation, energy, and emergency response. Accurate predictions enable better planning and resource allocation.

Though it has further applications in satellite imagery and remote sensing data, exploring hybrid models combining statistical and machine learning approaches,

-: Reference :-

- Data Source:
- https://www.kaggle.com/datasets/ashx010/weather-patterns-and-trends

- Code help Website:
- ‘Kaggle’ ,‘R-pubs’, ‘Git-hub’, ‘R-tutorial’, etc.

- Advance knowledge and facts Help:
  ChatGPT
- Books:

1.‘Fundamentals of Applied Statistics’-S.C. Gupta, V.K. Kapoor

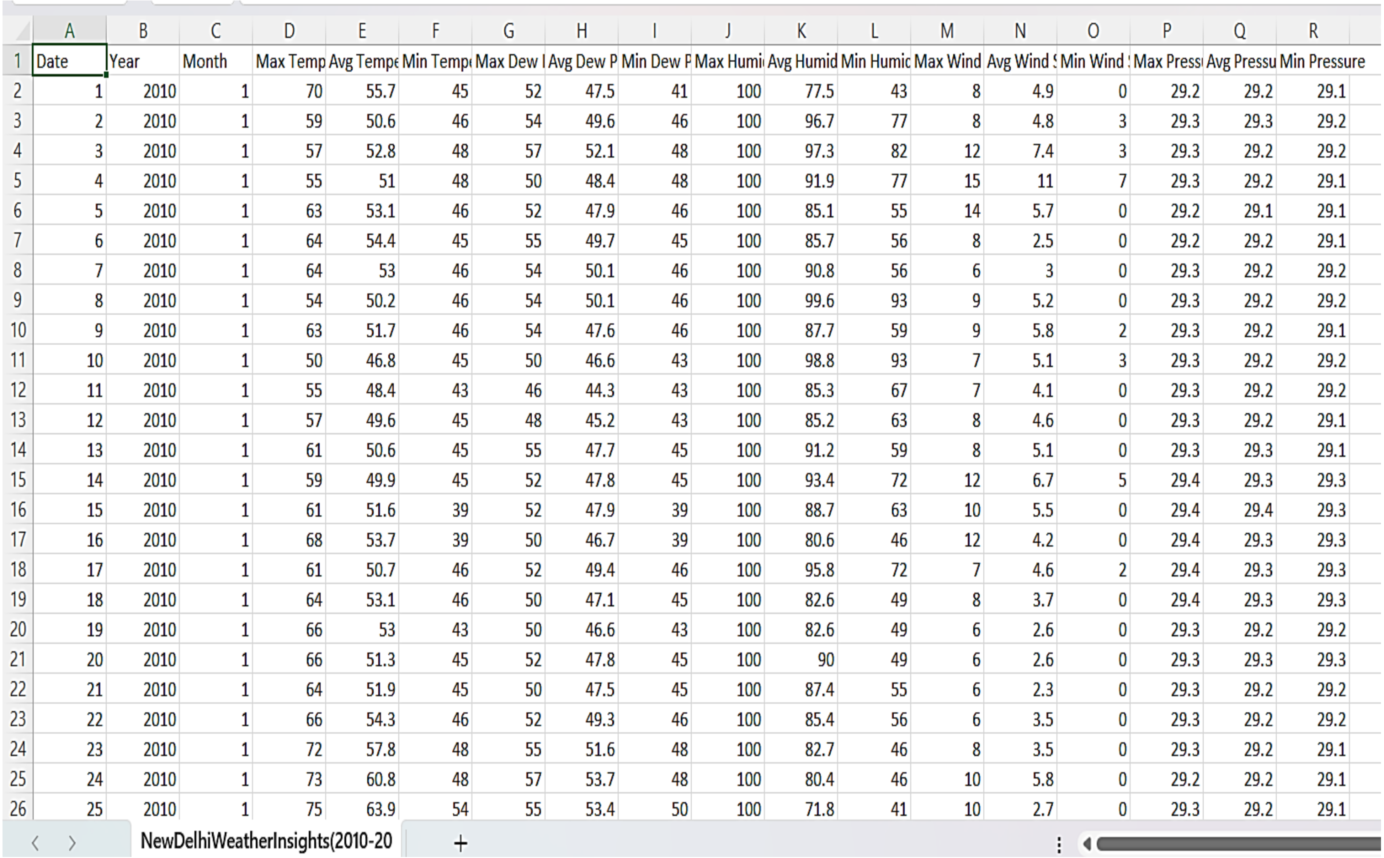
2.‘An Insights into Statistics’- Sarkhel, Dutta

- Image:

‘Google.com’,

-: Appendix :-

*New Delhi Weather Data Sample Screenshot:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Date	Year	Month	Max Temp	Avg Tempe	Min Tempe	Max Dew I	Avg Dew P	Min Dew P	Max Humi	Avg Humid	Min Humic	Max Wind	Avg Wind S	Min Wind S	Max Press	Avg Pressu	Min Pressure
2	1	2010	1	70	55.7	45	52	47.5	41	100	77.5	43	8	4.9	0	29.2	29.2	29.1
3	2	2010	1	59	50.6	46	54	49.6	46	100	96.7	77	8	4.8	3	29.3	29.3	29.2
4	3	2010	1	57	52.8	48	57	52.1	48	100	97.3	82	12	7.4	3	29.3	29.2	29.2
5	4	2010	1	55	51	48	50	48.4	48	100	91.9	77	15	11	7	29.3	29.2	29.1
6	5	2010	1	63	53.1	46	52	47.9	46	100	85.1	55	14	5.7	0	29.2	29.1	29.1
7	6	2010	1	64	54.4	45	55	49.7	45	100	85.7	56	8	2.5	0	29.2	29.2	29.1
8	7	2010	1	64	53	46	54	50.1	46	100	90.8	56	6	3	0	29.3	29.2	29.2
9	8	2010	1	54	50.2	46	54	50.1	46	100	99.6	93	9	5.2	0	29.3	29.2	29.2
10	9	2010	1	63	51.7	46	54	47.6	46	100	87.7	59	9	5.8	2	29.3	29.2	29.1
11	10	2010	1	50	46.8	45	50	46.6	43	100	98.8	93	7	5.1	3	29.3	29.2	29.2
12	11	2010	1	55	48.4	43	46	44.3	43	100	85.3	67	7	4.1	0	29.3	29.2	29.2
13	12	2010	1	57	49.6	45	48	45.2	43	100	85.2	63	8	4.6	0	29.3	29.2	29.1
14	13	2010	1	61	50.6	45	55	47.7	45	100	91.2	59	8	5.1	0	29.3	29.3	29.1
15	14	2010	1	59	49.9	45	52	47.8	45	100	93.4	72	12	6.7	5	29.4	29.3	29.3
16	15	2010	1	61	51.6	39	52	47.9	39	100	88.7	63	10	5.5	0	29.4	29.4	29.3
17	16	2010	1	68	53.7	39	50	46.7	39	100	80.6	46	12	4.2	0	29.4	29.3	29.3
18	17	2010	1	61	50.7	46	52	49.4	46	100	95.8	72	7	4.6	2	29.4	29.3	29.3
19	18	2010	1	64	53.1	46	50	47.1	45	100	82.6	49	8	3.7	0	29.4	29.3	29.3
20	19	2010	1	66	53	43	50	46.6	43	100	82.6	49	6	2.6	0	29.3	29.2	29.2
21	20	2010	1	66	51.3	45	52	47.8	45	100	90	49	6	2.6	0	29.3	29.3	29.3
22	21	2010	1	64	51.9	45	50	47.5	45	100	87.4	55	6	2.3	0	29.3	29.2	29.2
23	22	2010	1	66	54.3	46	52	49.3	46	100	85.4	56	6	3.5	0	29.3	29.2	29.2
24	23	2010	1	72	57.8	48	55	51.6	48	100	82.7	46	8	3.5	0	29.3	29.2	29.1
25	24	2010	1	73	60.8	48	57	53.7	48	100	80.4	46	10	5.8	0	29.2	29.2	29.1
26	25	2010	1	75	63.9	54	55	53.4	50	100	71.8	41	10	2.7	0	29.3	29.2	29.1

NewDelhiWeatherInsights(2010-20

*Code in R:

[Project code]

*Data Set:

[My data Set]

END