Ahmedabad University

May – July 2020

Report of the Internship

Name – Khushi Shah & Tirth Bharatbhai Kanani

Course – BS (Hons)

Enrolment No. – AU1920171 & AU1920144

Mentor – Professor Aditya Vaishya

Ahmedabad University | 10 YEARS 2009-2019

Declaration

We, hereby declare that the project report entitled "Report of the Internship" is our work. It has been submitted by us to Ahmedabad University. The report has been completed by *Ms. Khushi Mitesh Shah* and *Mr. Tirth Bharatbhai Kanani*, students of the B.S. (Honors) program at the *School of Arts and Sciences, Ahmedabad University*. The report was completed under the supervision of *Professor Aditya Vaishya*. We further declare that the matter embodied in this project report has not been submitted to any other University or Institution for the award of certificate or degree. This report is our original work and it has not been presented earlier in this manner. This information is pure of academic interest.

Ms. Khushi Mitesh Shah	*Mr. Tirth Bharatbhai Kanani*
B.S. (Honors)	B.S. (Honors)
AU1920171	AU1920144

Certificate

*This is to certify that the project titled **Report of the Internship** is a record of the bonafide work done by **Ms. Khushi Mitesh Shah** (AU1920171) and **Mr. Tirth Bharatbhai Kanani** (AU1920144) under the guidance of **Professor Aditya Vaishya** as a part of the Summer Internship at Ahmedabad University, from May 18 to July 18, 2020.*

Prof. Aditya Vaishya

(Mentor)

Assistant Professor, School of Arts and Sciences
Ahmedabad University

Acknowledgment

Hereby, we would like to thank the *Career Development Centre at Ahmedabad University* for providing us this opportunity to undergo a two-month internship even during these uncertain circumstances. We also extend our thanks to *Prof. Aditya Vaishya*, who has helped us through every step of our internship.

Prof. Vaishya guided us regarding data analysis, the importance of each reading, cleaning, and refining of data good programming practices, and much more.

Once again, we would like to thank each and every one who helped us through this internship

Table of Contents

List of Tables

List of Figures

Abstract

This project is about analysing the variations in the wind over a given geographical location and over a period of time. Variations in wind speed might contain some pattern like waxing and waning wind episodes, calm periods of winds, hurricane or cyclonic activities etc. The meteorological measurements and data consisted of temperature, relative humidity, wind speed, wind direction, and pressure. We analysed the meteorological measurements using computer programming. We analysed the wind speed – wind direction relation, studied the wind speed frequencies, and found out the maxima and minima episodes in wind. Through this we also identified the waxing and waning wind episodes.

For the given meteorological measurements we found out that normally the wind speed is between 4 m/s to 8 m/s. The average wind speed over the period of one year was be 7.4 meter/second. There were a few episodes where the wind speed reached 20-25 m/s. These can be classified/termed as Hurricane winds.

Purpose

Hurricane winds are hazardous for life and property as it can cause wide-spread destruction. Similarly, wind-still episodes can make on very uncomfortable in the hot summer months. The purpose of this study is to identify maxima and minima episodes in wind speed over a given geographical region. Also, a systematic analyses of wind speed and wind direction is crucial for building construction, ventilation etc. Quantification of waxing and waning episodes of wind and a robust statistical analyses of wind speed will help in above mentioned aspects.

Table 1: Definition of Terms

Serial No.	*Term*	*Definition*
1.	*Julian Day*	Julian Day [1] is the count of days passed from the starting of the Julian period. It is the count of the number of days passed since noon on January 1st, 4713.
2.	*Wind Speed*	Wind speed is the fundamental atmospheric quantity that measures the movement of Wind from high pressure to low pressure.
3.	*Local Maxima*	A mathematical term which is used to denote the maximum value of the parameter in a specified range of data.
4.	*Local Minima*	A mathematical term which is used to denote the minimum value of the parameter in a specified range of data.
5.	*Polar Plot*	A polar plot is a graph between the phase angle and magnitude of data. It is a graphical representation of data

Table 2: Methods and Tools Used

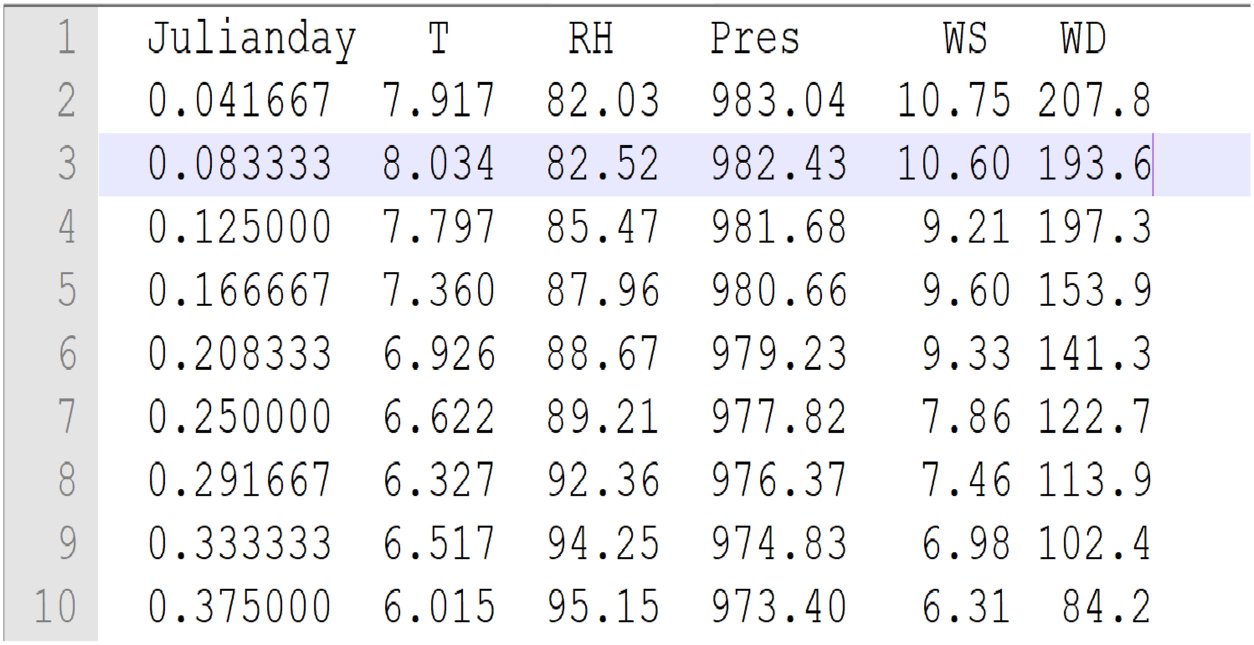
MATLAB: A multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms.

Serial No.	*Inbuilt Function Used*	*Function Definition*
1.	*textscan()*	This function reads formatted data from a text file or string.
2.	*height()*	This function returns the number of table rows.

3.	*length()*	This function returns the number of array rows.
4.	*table2array()*	This function converts the table to a homogeneous array.
5.	*islocalmax()*	This function checks whether the given value is a local maxima or not. It returns either '0' or '1'.
6.	*islocalmin()*	This function checks whether the given value is a local minima or not. It returns either '0' or '1'.
7.	*datenum()*	This function converts date and time to a serial number.
8.	*datestr()*	This function converts date and time to a string.
9.	*datetime()*	This function converts the array to a date-time format.
10.	*axes()*	This function creates Cartesian axes.
11.	*plot()*	This function plots a 2-D graph across the past dimensions.
12.	*xlabel()*	This function gives the name to the X-axis of the graph.
13.	*ylabel()*	This function gives the name to the Y-axis of the graph.
14.	*title()*	This function gives a title to the graph.
15.	*legend()*	This function creates legend and displays the legend name.
16.	*saveas()*	This function saves a figure in the specified format.
17.	*input()*	This function request user input.
18.	*nanmean()*	This function computes the mean of an array ignoring the NaN values.
19.	*polarplot()*	This function plots specified object parameters on polar coordinates
20.	*mean()*	This function computes the mean or average of the given array.
21.	*median()*	This function returns the median of the given data array.
22.	*std()*	This function returns the standard deviation of the given data array.
23.	*issorted()*	This function returns whether the given data array is sorted in ascending or descending order.

Raw Data

The metrological data in the raw form includes Date, Time, Temperature, Humidity, Pressure, Wind Speed, and Wind Direction. The data has been collected hourly and has 8670 rows and 6 columns. A snip of the same has been given below.



Julianday	T	RH	Pres	WS	WD
0.041667	7.917	82.03	983.04	10.75	207.8
0.083333	8.034	82.52	982.43	10.60	193.6
0.125000	7.797	85.47	981.68	9.21	197.3
0.166667	7.360	87.96	980.66	9.60	153.9
0.208333	6.926	88.67	979.23	9.33	141.3
0.250000	6.622	89.21	977.82	7.86	122.7
0.291667	6.327	92.36	976.37	7.46	113.9
0.333333	6.517	94.25	974.83	6.98	102.4
0.375000	6.015	95.15	973.40	6.31	84.2

Figure 1 Raw Data

Data Analysis

The metrological data that has been collected in the year 2019 has been refined and analysed to form graphs and represent the data in a way that it becomes easy to understand. We had been provided with a series of hourly raw data which included details of Julian day [1] of the year, the temperature, the humidity, pressure, wind speed, and wind direction at that particular day and time of the year. Scanning of the data file, one of the many important tasks, was done using either import data function availability in MATLAB or by using the function 'textscan'. After going through the data and plotting it once, we realized that some part of the data required refining as the data hit very rare readings at a few points. Once the data was cleaned we converted the table to array formats such that working on them individually would become easier. We plotted the first graph of Wind Speed across Julian Day [1]. Again after converting Julian Day [1] to a Date - Time array we plot the Wind Speed again. After doing this we worked upon the Wind Direction data. We observed that even though wind came along different directions, it was possible to categorize wind speed along with wind direction. Therefore, we did the Polar plotting of average Wind Speed along its respective wind direction. Based on what we did in the past two months, we have created the report for the same.

1. Wind Speed - Date

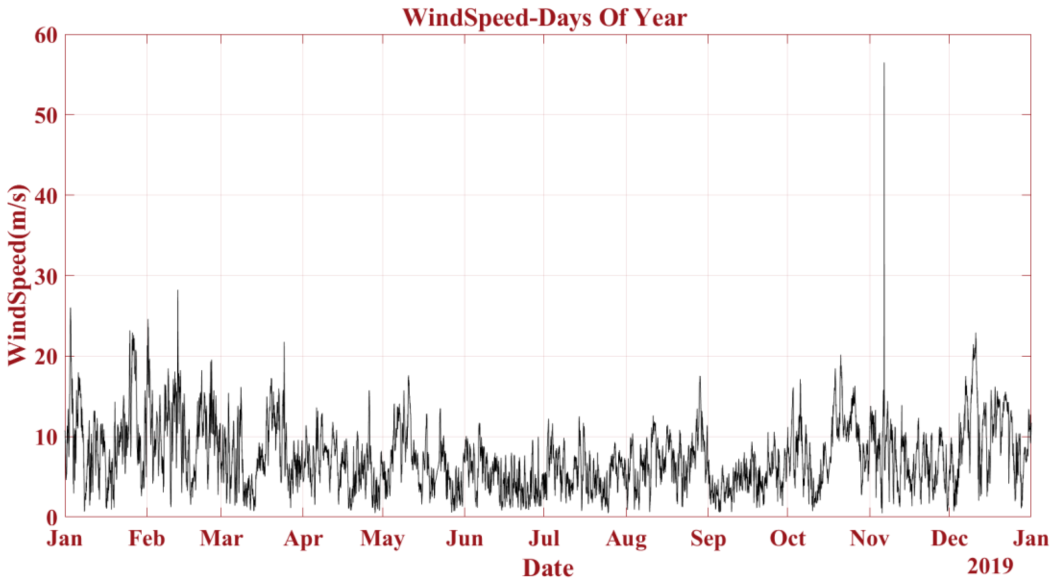


Figure 2 Graph of Wind Speed vs. Date

The above-mentioned graph is of the wind speed recorded for the entire year at a particular station. On the X-axis, we have date from 1st January 2019 to 31st December 2019. On the Y-axis we have wind speed in meter/second

On average, the wind speed is: 7.3898 m/s.

The wind with speed more than 25m/s is termed as Hurricane Winds.

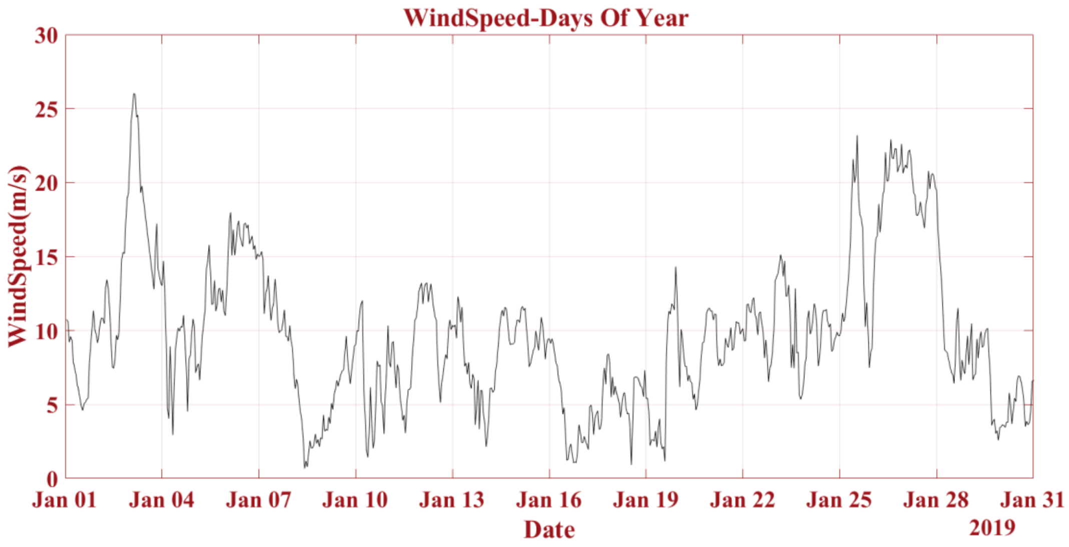


Figure 3 Graph of Wind Speed vs. Date (Enlarged)

The above-mentioned graph in Figure 3 is of the month of January (Winter Season in Northern Hemisphere) and at that time the wind recorded at this particular station generally ranges from 3.00 m/s to 15.00 m/s.

2. Polar Plot

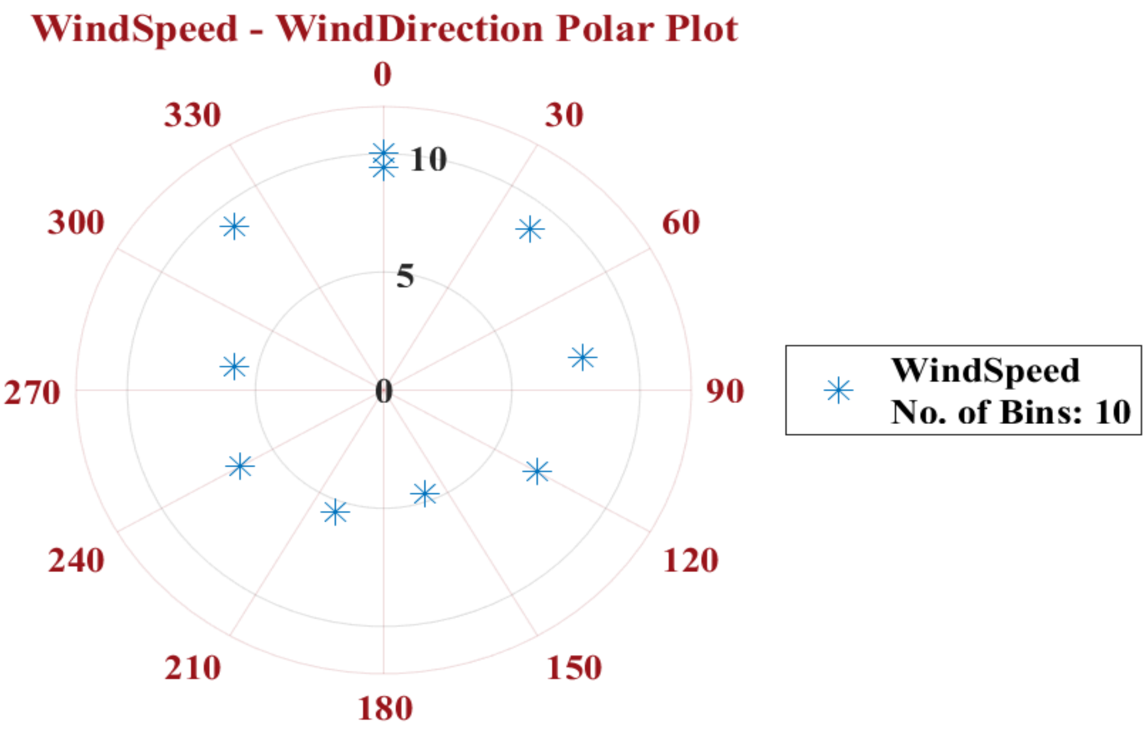


Figure 4 Polar plot of the mean of wind speed regarding its direction

Only observing and plotting wind speed is not sufficient. One may get some interpretation from it but that is neither appropriate nor sufficient.

Analysing the wind direction is also equivalent. One of the best ways to plot both is to use polar as it is done in the above-mentioned figure. We have plotted the mean of Wind Speed regarding Wind Direction.

The number of bins is dynamic. In this figure, the number of bins is 10.

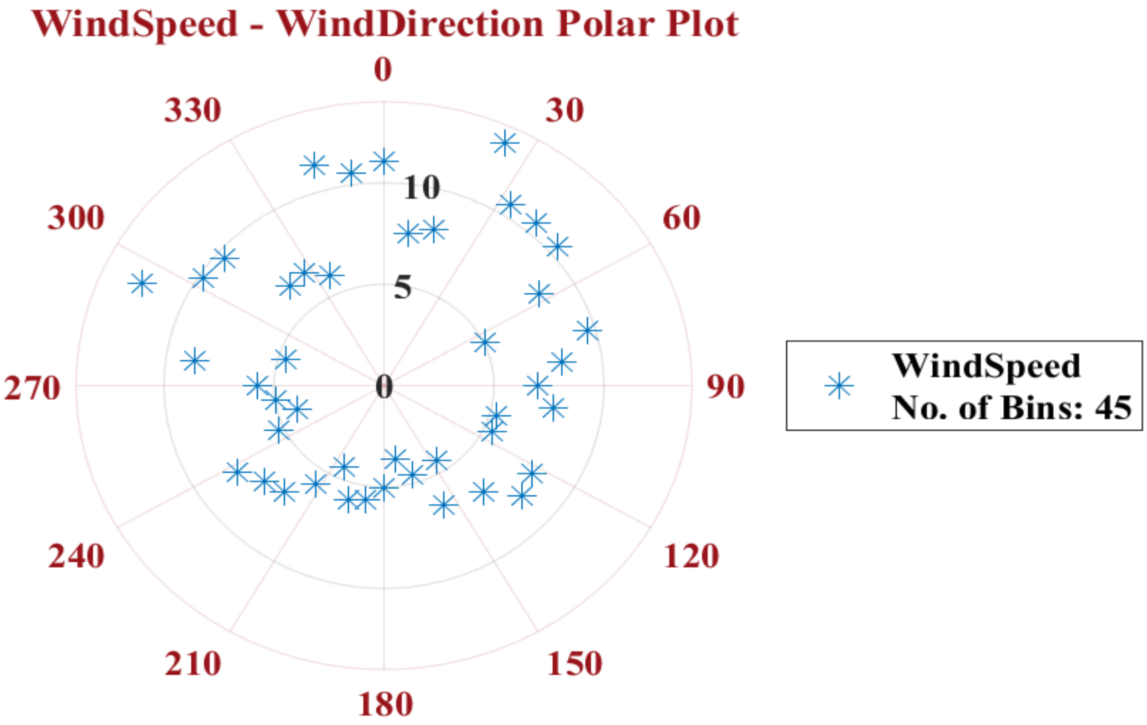


Figure 5 Polar plot of the mean of wind speed with its reference to wind direction with 45 bins

This graph is quite similar to fig. 5 concerning the manner of plotting. Although, here the number of bins is increased to 45. The theta axis ranges from 0° to 360° degrees and the rho axis is set accordingly.

The high-speed winds generally blow from North-western and North-eastern direction i.e. from 270° to 0° and 0° to 90°. From the south direction, the wind speed generally ranges around 5.00 m/s.

3. Maxima Minima[3]

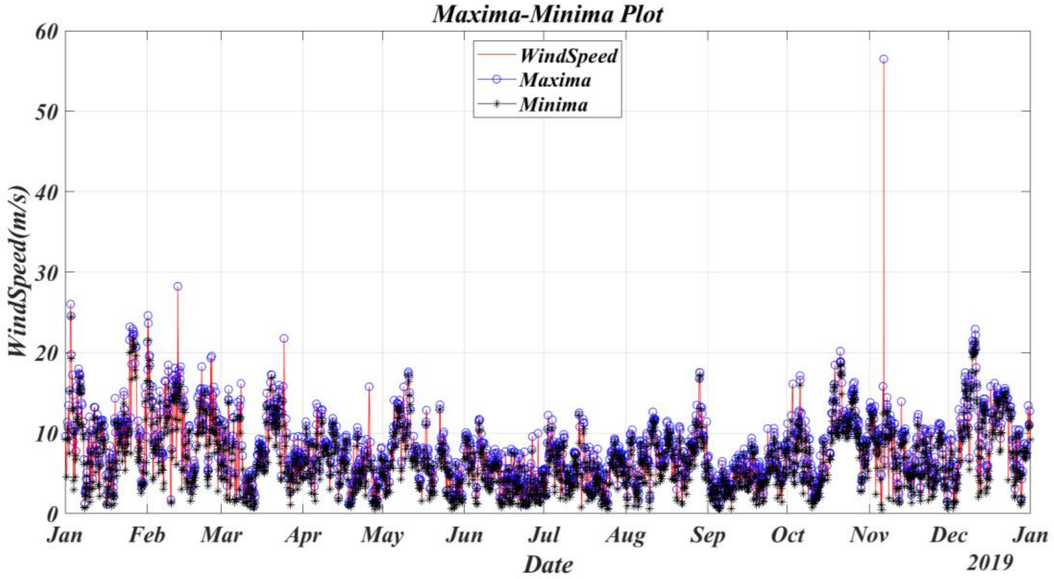


Figure 5 Graph of Maxima's and Minima's of Wind Speed for the entire year

On closely looking at the wind speed data, we get to know much new information such as waxing and waning episodes, maxima-minima[3], and many other things.

In the above graph, we have to plot the maxima and minima.

Definition of Maxima and Minima [3]:

Maxima: A mathematical term that is used to denote the maximum value of the parameter in a specified range of data.

Minima: A mathematical term that is used to denote the minimum value of the parameter in a specified range of data.

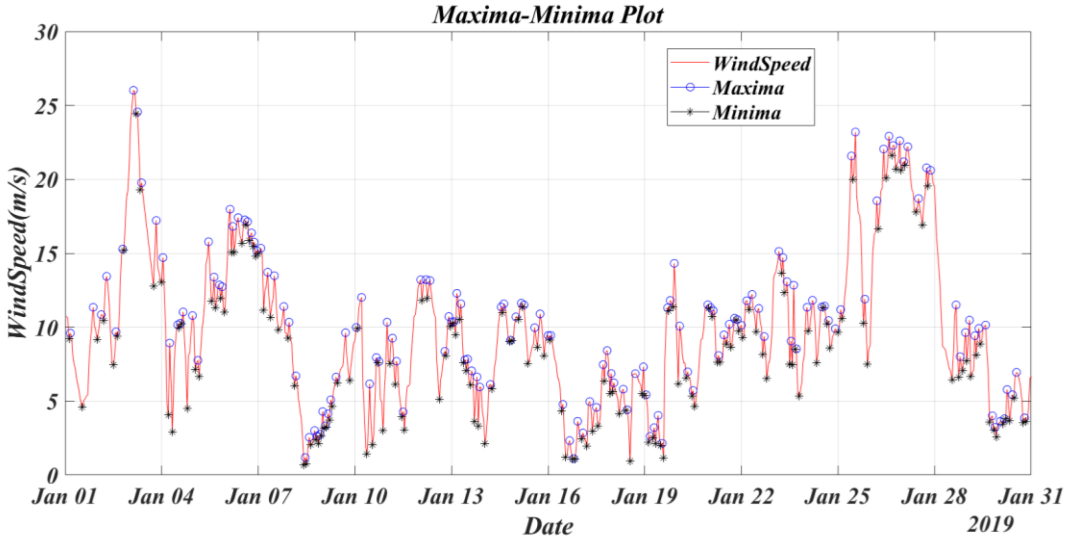


Figure 6 Enlarged graph of Maxima's and Minima's of Wind Speed showing data of one month

When we look at the data of only one month in fig. 7, we can observe the local maxima and local minima very clearly. On the x-axis, we have the date and on the y-axis, we have the wind speed in m/s.

As you can see in the above-mentioned graph, at the starting of the month and the ending of the month, the wind speeds are quite high, generally greater than 20 m/s (>20 m/s) which are also known as hurricane winds. Generally, the winds are created due to variation in temperature and pressure and therefore one would see a lot of fluctuations in the wind speed as the temperature and pressure changes a lot within a day itself.

4. Consecutive Points

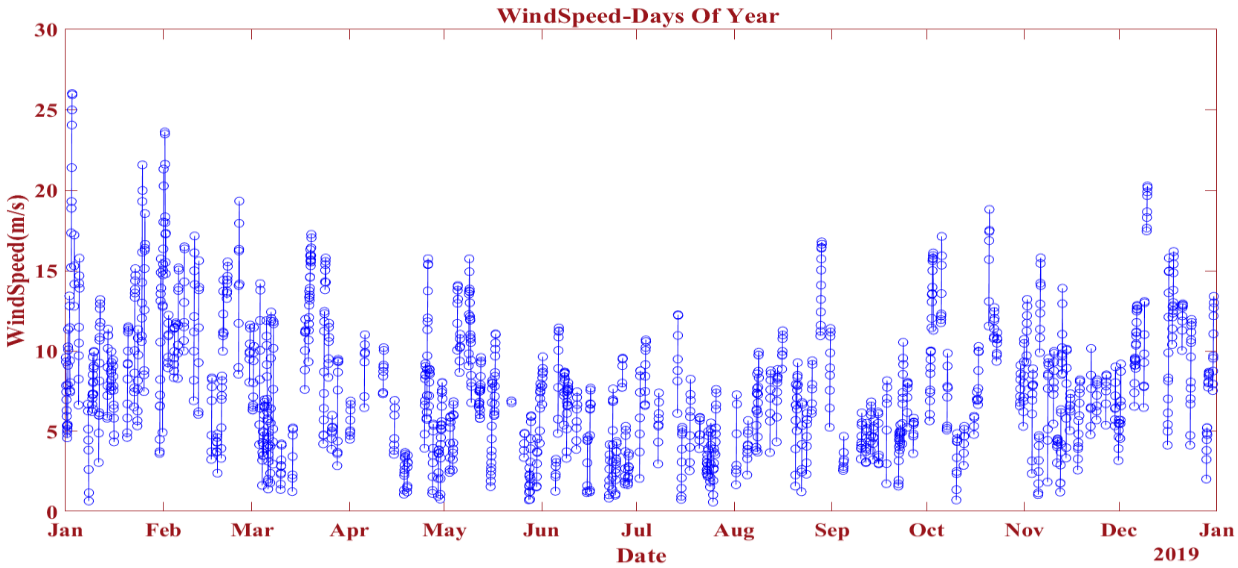


Figure 7 Consecutive Points with DCP as 7

In this graph, we plot the points that are consecutive in either ascending or descending order, to show the overall increase/ decrease in Wind speed over the time. While plotting this graph we have a dynamic input for the number of consecutive to check. In this graph, the dynamic input is 7, which means that all the points that are in either ascending or descending order for at least 7 point consecutive points are plotted in this graph.

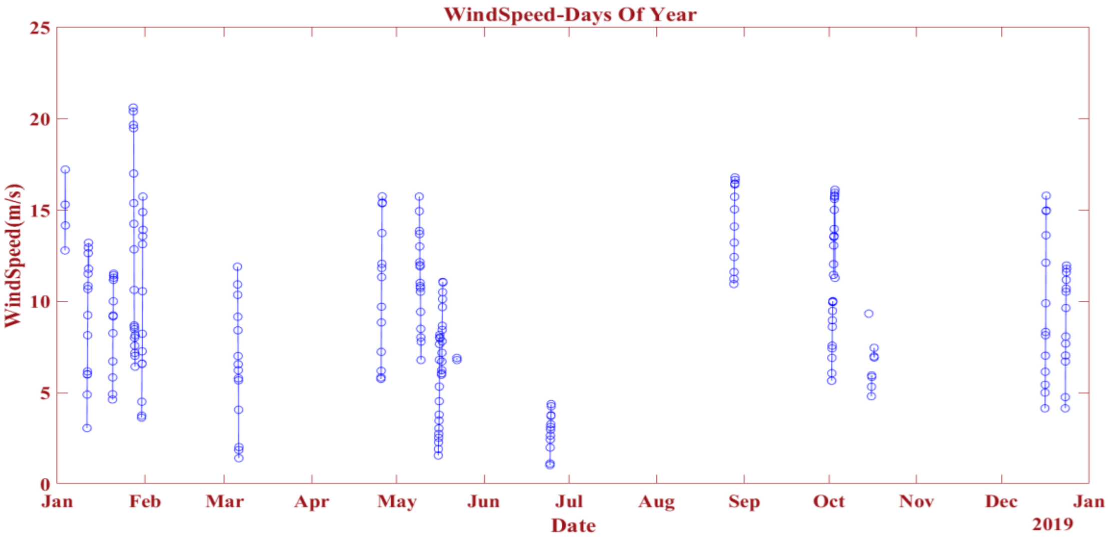


Figure 8 Consecutive Points with DCP as 12

In this graph, the dynamic input is 12. Thus all the points out of the complete data array, where there are either 12 or more than 12 sorted consecutive points are plotted on this graph. In the above-mentioned graph, the plotted thing is the waxing or waning episodes of the wind speed over a particular year. The waxing or the waning episodes are the periods or the slots where either the wind speed is continuously increasing or wind speed is continuously decreasing for a certain amount of hours. Analysing that thing plays a very crucial role in the study of weather conditions.

5. Wind Speed – Julian Day [1]

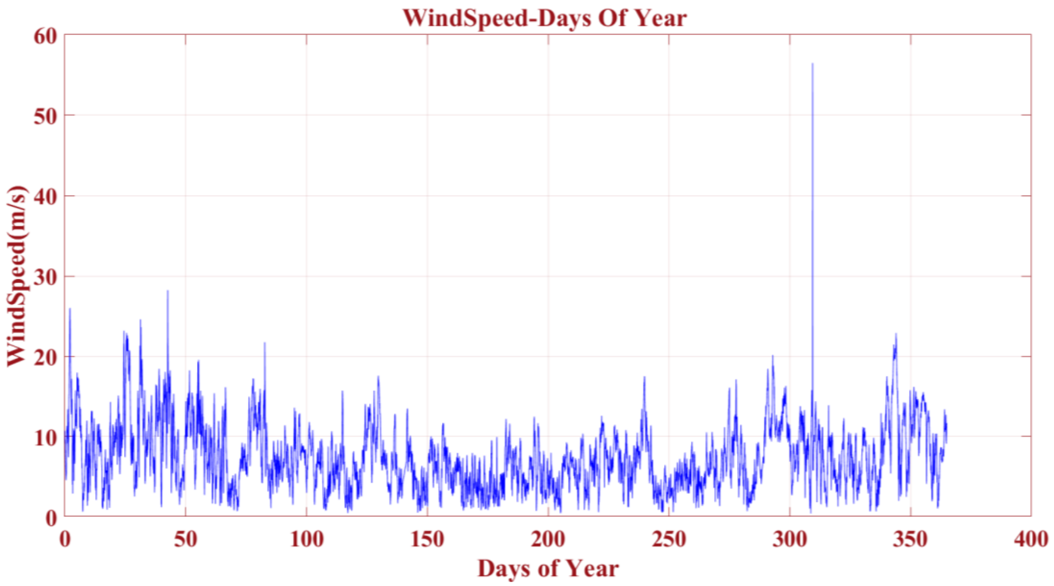


Figure 9 Graph of Wind Speed vs. Julian Day for the entire year

The above-mentioned graph is on wind speed vs. Julian day [1].

Julian Day [1]: Julian Day [1] is the count of days passed from the starting of the Julian period. It is the count of the number of days passed since noon on January 1st, 4713.

On the x-axis we have Julian day [1] and, on the y-axis, we have wind speed in meter/second.

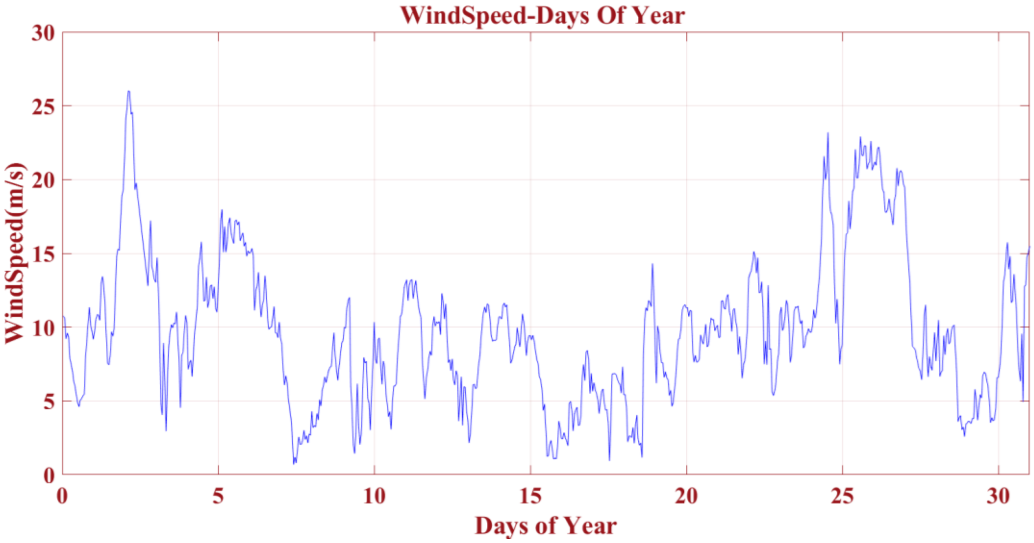


Figure 10 Graph of Wind Speed vs. Julian Day [1] for a particular time

The above graph is the enlarged version of fig. 11. It shows the wind speed variance over the period of 30-35 days. On the x-axis we have Julian Day [1] and on the y-axis, we have wind speed in meter/second (m/s). The above-mentioned graph is similar to figure 3, where the only difference is seen on the x-axis. On average the wind speed is between 3.00 m/s to 15.00 m/s with some fluctuations at the beginning and the end of the graph.

After refining the data and plotting all these graphs, we come to a result of the following statistics:

- Average Wind Speed throughout the year is 7.39 m/s.
- The median of Wind Speed throughout the year is 6.81 m/s.
- Standard Deviation of Wind Speed throughout the year is 4.11 m/s

Month	***Average Wind Speed (m/s)***	***Median of Wind Speed (m/s)***	***Standard Deviation of Wind Speed (m/s)***
January	9.67	9.26	4.93
February	10.80	10.93	4.43
March	7.50	6.70	3.88
April	6.37	6.55	2.88
May	6.84	6.40	3.57

June	4.87	4.58	2.37
July	5.55	5.42	2.39
August	7.27	7.28	2.92
September	4.69	4.55	1.97
October	8.88	8.99	4.06
November	7.25	7.09	3.63
December	9.01	9.31	5.30

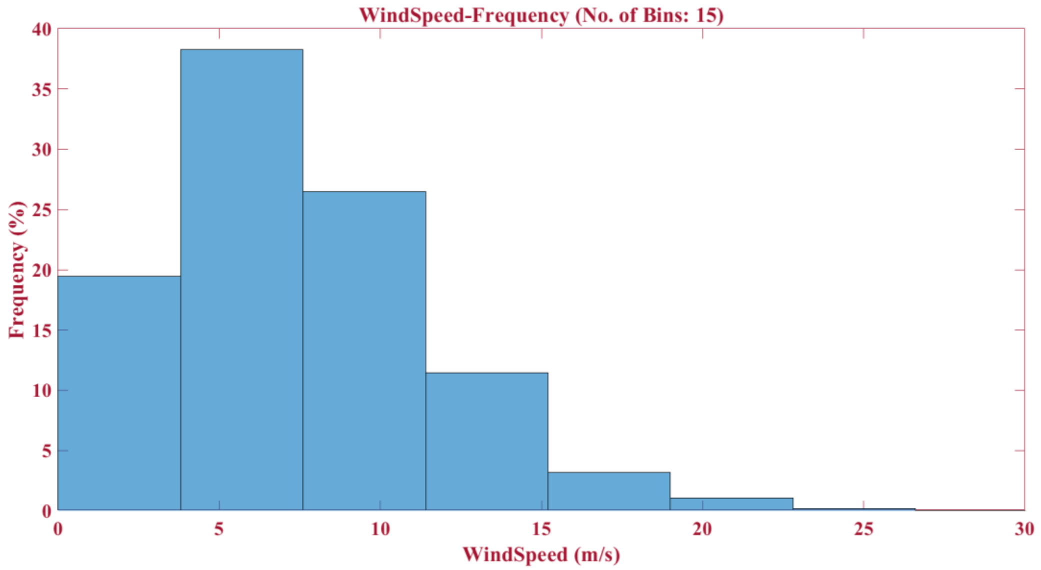


Figure 11 Histogram showing the frequency of wind speed

As you can see, the highest occurring wind speed is from around 4.00 m/s to 7.50 m/s with the occurrence of around 38% followed by 7.50 to 12.00 m/s which occurs for roughly of 26% to 27%. The histogram gives an overall idea of how much percent of wind speed ranges over a period of the year, showing the highest and the least occurring wind speeds.

Appendix

1. References

1. *Wikipedia. (n.d.).* Julian Day. *Retrieved from Wikipedia:* https://en.wikipedia.org/wiki/Julian_day

2. *MATLAB . (n.d.).* MATLAB Documentation . *Retrieved from Mathworks.in :* https://in.mathworks.com/help/matlab/

3. *MATLAB . (n.d.).* MATLAB Community. *Retrieved from bolgs.mathworks.com:* https://blogs.mathworks.com/community

4. *Wikipedia. (n.d.).* Maxima minima. *Retrieved from Wikipedia.com:* https://en.wikipedia.org/wiki/Maxima_and_minima

2. Programs:

1.) Wind Speed – Date

Code

```
%% Program to plot graph between Time and Wind Speed
%% Clearing the previous variables and Command Window
clear;
clc;
%% Importing data
filename ='..//Raw Data//Raw Met Data.dat'; %File path
startRow = 2;
formatSpec = '%10f%7f%7f%8f%7f%f%[^\n\r]'; %datatyape specifiers
f1 = fopen(filename,'r');
%% Scanning of data
dataArray = textscan(f1, formatSpec, 'Delimiter', '', 'WhiteSpace',...
'TextType', 'string', 'EmptyValue', NaN, 'HeaderLines' ,...
startRow-1, 'ReturnOnError', false, 'EndOfLine', '\r\n');

RawMetData1 = table(dataArray{1:end-1}, 'VariableNames', {'Julianday',...
'TRHPr','esWSW','D','WindSpeed','WindDirection'});
%% Cleaning of data
```

```
toClear = RawMetData1.WindSpeed > 998;
RawMetData1(toClear,:) ={0};
%% Converting table to array
WS(:,1)=table2array(RawMetData1(:,5)); %WindSpeed data to be plotted on
Y axis
DT(:,1)=table2array(RawMetData1(:,1));  %Julian Day data to be plotted on
X axis
%% Converting Julian Day to Date-Time type
D(:,1)=datenum(2018,12,31,23,59,59)+DT(:,1);
D=datetime(datestr(datevec(D)));
%% Plotting the graph
figure1 = figure('WindowState','maximized'); %Figure properties
axes1 = axes('Parent',figure1); %Axis properties
hold(axes1,'on');
plot(D(WS~=0),WS(WS~=0),'black-') %plotting the graph
ylabel('WindSpeed(m/s)','FontWeight','bold','FontName','Times New
Roman','FontSize',18); %Y-axis title and properties
xlabel('Date','FontWeight','bold','FontName','Times New
Roman','FontSize',18,...
'Color',[0.6 0.08 0.1]); %X-axis title and properties
title('WindSpeed-Days Of Year ','FontName','Times New
Roman','FontSize',18,... %Graph title and properties
'Color',[0.6 0.08 0.1]);
box(axes1,'on');
grid on;
set(axes1,'FontName','Times New
Roman','FontSize',18,'FontWeight','bold','XColor',... %Axis properties
[0.6 0.08 0.1],'YColor',...
[0.6 0.08 0.1],'ZColor',...
[0.6 0.08 0.1]);
%% Clearing unrequired data
clear i j toDelete figure1 axes1;
clear startRow row formatSpec ;
%% Saving the graph
saveas(gcf,'..//Results//WindSpeedTime.png') %File path
```

2.) Polar Plot

<u>Code:</u>

```
%% Program to draw a polar plot
%% Cleaning the previous data and Command Window
clear;
clc;
%% Importing data
```

```
filename='..//Raw Data//Raw Met Data.dat'; %File path
% filename = '/Users/tirth_8205/Desktop/Ahmedabad University/Internship
[May - July,2020]/CDC-Internship/Raw Data/Raw Met Data.dat';
startRow = 2;
formatSpec = '%10f%7f%7f%8f%7f%f%[^\n\r]'; %data type format Specifier
f1 = fopen(filename,'r');
dataArray = textscan(f1, formatSpec, 'Delimiter', '', 'WhiteSpace', '',
'TextType', 'string', 'EmptyValue', NaN, 'HeaderLines' ,startRow-1,
'ReturnOnError', false, 'EndOfLine', '\r\n');
RawMetData1 = table(dataArray{1:end-1}, 'VariableNames',
{'Julianday','TRHPr','esWSW','D','VarName5','VarName6'});
%% Cleaning of data
toDelete = RawMetData1.VarName6>=360;
RawMetData1(toDelete,:) = {0};
%% Dynamic input for number of bins
nbins=(input("Please enter the number of Bins you want e.g 10:"));%to take
the value from user, like how many beans they want
binwidth=360/nbins;
%% Converting table to array
row=height(RawMetData1); %height function returns the length of the table
X1(:,1)=table2array(RawMetData1(:,5)); %WindSpeed data
Y1(:,1)=table2array(RawMetData1(:,1));  %Wind Direction data
%% Creating bins and segragating data
%for binwidth=1:500
e=binwidth;%intializing end point as the higher limit for creating bins, for
eg, if user wants 10 bins, then a=360/10=36, therefore the bins will be 0-36,
36-72 and so on
s=0;%intializing start point as the lower limit for creating bins
count=1; %will be used as counter vairiable
c=1; %will be used as counter variable
while (e<=360)
for i=1:row
if ((Y1(i,1)>=s)&&(Y1(i,1)<e)) %condition to check whether the given value
falls under the condition or not

M(c,count)=(X1(i,1)); %stores the values. FOR BETTER
UNDERSTANDING, PLEASE OPEN THE SECTION FROM THE
WORKSPACE
c=c+1;%increases the counter variable by 1
end
end
count=count+1;%increases the counter variable by 1
s=e;%changes the value after completion of the every loop, here for instance,
initially, it was 0, which will now change to 36, based on the above eg.
e=e+binwidth;%increases the uper limit, from 36 to 72, 72 to 108 and so on,
based on the above eg.
```

```
c=1;%after completion of every loop it should start with 1, therefore
assigning the value again.
end
row1=length(M);
add1=0;
t=0;
for j=1:nbins
M(M==0)=NaN;
u=nanmean(M);
end
polarplot(u,'*','MarkerSize',10);%plots the graph of average values
%% Polarplot properties
pax=gca;
pax.ThetaLim=[0 360];
pax.ThetaAxisUnits='degrees';
pax.ThetaDir='clockwise';
pax.ThetaZeroLocation='top';
pax.ThetaColor=[0.6 0.08 0.1];
pax.Title.String='WindSpeed - WindDirection Polar Plot';
pax.Title.Color=[0.6 0.08 0.1];
legendTitle=sprintf("WindSpeed\nNo. of Bins: %d",nbins);
legend({legendTitle},'FontSize',13)
pax.FontSize=13;
grid on;
pax.FontName='Times';
pax.FontWeight='bold';
%% Cleaning unrequired data
clear toDelete row i formatSpec startRow f1 j e M N k row row1 ...
s t count u c binwidth add1;
%% Saving the graph
saveas(gcf,'..//Results//WindSpeed-WindDirection PolarPlot(45 bins).png')
```

3.) *Maxima Minima*

Code:

```
%% Finding the local Maxima-Minima of WindSpeed and Plotting the graph
%% Cleaning previous variables and Command Window
clear all;
clc;
%% Importing the data
filename = '..//Raw Data//Raw Met Data.dat';   %selection of file
startRow = 2;
formatSpec = '%10f%7f%7f%8f%7f%f%[^\n\r]'; %datatype format specifiers
f1 = fopen(filename,'r');
```

```
%% Scanning of data
dataArray = textscan(f1, formatSpec, 'Delimiter', '', 'WhiteSpace', '',...
'TextType', 'string', 'EmptyValue', NaN, 'HeaderLines' ,startRow-1,...
'ReturnOnError', false, 'EndOfLine', '\r\n');
fclose(f1);
RawMetData1= table(dataArray{1:end-1}, 'VariableNames', {'JulianDay',...
'TRHPr','esWSW','D','WindSpeed','WindDirection'});
%% Cleaning of data
toDelete = RawMetData1.WindSpeed > 998;
RawMetData1(toDelete,:) = [];
%% Converting table to array
WS(:,1)=table2array(RawMetData1(:,5)); %WindSpeed data to be plotted on
Y axis
DT(:,1)=table2array(RawMetData1(:,1));  %Julian Day data to be plotted on
X axis
%% Finding Maxima Minima
amax=islocalmax(WS); %returns whether X1 is local minima or not
amin=islocalmin(WS); %returns whether X1 is local minima or not
Maxima=WS;
Maxima(~amax,1)=NaN;
Minima=WS;
Minima(~amin,1)=NaN;
%% Julian day to date conversion
JD=NaN(length(DT),1);
JD=datetime(datestr(datenum(2018,12,31,23,59,59)+DT(:,1)));
%% Plotting Maxima Minima
figure1 = figure;
axes1 = axes('Parent',figure1); %Figure properties
hold(axes1,'on');
plot(JD,WS,'DisplayName','WindSpeed','Color',[1 0 0]); %plots WindSpeed
hold on;
plot(JD,Maxima,'DisplayName','Maxima','Marker','o',... %plots Maxima
points
'Color',[0 0 1]);
hold on;
plot(JD,Minima,'DisplayName','Minima','Marker','*','Color',[0 0 0]); %plots
the Minima points
hold off;
%% Plot Properties
ylabel('WindSpeed(m/s)','FontSize',18,'FontName','Times New Roman'); %Y-
axis title and properties
xlabel('Date','FontSize',18,'FontName','Times New Roman'); %X-axis title
and properties
title('Maxima-Minima Plot','FontSize',18,'FontName','Times New Roman');
%Graph title and properties
box(axes1,'on');
```

```
grid on;
set(axes1,'FontAngle','italic','FontName','Times New Roman','FontSize',18,...
'FontWeight','bold','XGrid','on','YGrid','on');
legend1 = legend(axes1,'show'); %Legend properties
set(legend1,'Location','best');
%% Cleaning unrequired data
clear i toDelete figure1;
clear startRow row formatSpec;
%% Saving the image
saveas(gcf,'..//Results//MaximaMinima.png') %File path
```

4.) Consecutive Points

<u>Code:</u>

```
%% Program to find Consecutive points in order to find the flow of data
WindSpeed
%% Cleaning history and command Window
clear;
clc;
%% Importing Data file
filename ='..//Raw Data//Raw Met Data.dat'; %File path
startRow = 2;
formatSpec = '%10f%7f%7f%8f%7f%f%[^\n\r]'; %datatype specifiers
f1 = fopen(filename,'r');
dataArray = textscan(f1, formatSpec, 'Delimiter', '', 'WhiteSpace',...
'', 'TextType', 'string', 'EmptyValue', NaN, 'HeaderLines' ,...
startRow-1, 'ReturnOnError', false, 'EndOfLine', '\r\n');
RawMetData1 = table(dataArray{1:end-1}, 'VariableNames', {'Julianday',...
'TRHPr','esWSW','D','WindSpeed','WindDirection'});
%% Cleaning of data
toClear = RawMetData1.WindSpeed > 998;
RawMetData1(toClear,:) ={0};
%% Converting table to array
row=height(RawMetData1);
WS(:,1)=table2array(RawMetData1(:,5)); %WindSpeed data to be plotted on Y
axis
DT(:,1)=table2array(RawMetData1(:,1));  %Julian Day data to be plotted on X
axis
%% Consecutive Points Algorithm
DCP=input('Enter Consecutive Points: '); %Dynamic Input for the minimum
number of consecutive points
i=1; % constant declared for loop
P1((1:length(WS)),1)=0; %pre-defined array
for j=1:row
```

```
if ((issorted(WS(i:j),'ascend')))%issorted function returns whether the data is sorted or not
continue;
else
if((j-i+1)>=DCP)
P1((i:j),1)=(WS((i:j),1));
i=j+1;
else
i=i+1;
end
end
end
P1(P1==0)=NaN; % Replaces all the zeroes by NaN
i=1;
for j=1:row
if (issorted(WS(i:j),'descend')) %issorted function returns whether the data is sorted or not
continue;
else
if((j-i+1)>=DCP)
P2((i:j),1)=(WS((i:j),1));
i=j+1;
else
i=i+1;
end
end
end
P2(P2==0)=NaN; % Replaces all the zeroes by NaN
%% Conversion of Julian day to date
D(:,1)=datenum(2018,12,31,23,00,01)+DT(:,1);
D=datetime(datestr(datevec(D)));
%% Plotting the graph
figure1 = figure('WindowState','maximized');
axes1 = axes('Parent',figure1);
hold(axes1,'on');
plot(D,P1,P2,'bo-',)
ylabel('WindSpeed(m/s)','FontWeight','bold','FontName','Times'); %Y-axis title and properties
xlabel('Date','FontWeight','bold','FontName','Times',... %X-axis title and properties
'Color',[0.6 0.08 0.1]);
title('WindSpeed-Days Of Year ',... %Graph title and properties
'Color',[0.6 0.08 0.1]);
box(axes1,'on');
set(axes1,'FontName','Times','FontSize',15,'FontWeight','bold','XColor',...
[0.6 0.08 0.1],'YColor',...
[0.6 0.08 0.1],'ZColor',...
[0.6 0.08 0.1]);
%% Clearing unrequeired data
clear i j toDelete figure1 axes1 startRow formatSpec ;
%% Saving the plot image
%saveas(gcf,'..//Results//ConsecutivePoints.png')
```

5.) Wind Speed – Julian Day

Code:

```
%% Program to plot graph between JulianDay and Wind Speed
%% Clearing the previous variables and Command Window
clear;
clc;
%% Import the data
filename ='..//Raw Data//Raw Met Data.dat'; %File path
startRow = 2;
formatSpec = '%10f%7f%7f%8f%7f%f%[^\n\r]'; %datatype specifiers
f1 = fopen(filename,'r');
%% Scanning of data
dataArray = textscan(f1, formatSpec, 'Delimiter', '', 'WhiteSpace',...
'', 'TextType', 'string', 'EmptyValue', NaN, 'HeaderLines' ,...
startRow-1, 'ReturnOnError', false, 'EndOfLine', '\r\n');
RawMetData1 = table(dataArray{1:end-1}, 'VariableNames', {'Julianday',...
'TRHPr','esWSW','D','WindSpeed','WindDirection'});%% Cleaning of data
toClear = RawMetData1.WindSpeed > 998;
RawMetData1(toClear,:) ={0};
%% Converting table to array
WS(:,1)=table2array(RawMetData1(:,5)); %WindSpeed data to be plotted on
Y axis
DT(:,1)=table2array(RawMetData1(:,1));  %Julian Day data to be plotted on
X axis
%% Plotting the graph
figure1 = figure('WindowState','maximized'); %Figure properties
axes1 = axes('Parent',figure1); %Axis properties
hold(axes1,'on');
plot(DT(WS~=0),WS(WS~=0),'b-') % Plotting the data
ylabel('WindSpeed(m/s)','FontWeight','bold','FontName','Times New
Roman','FontSize',18); %Y-axis title and properties
xlabel('Days of Year','FontWeight','bold','FontName','Times New
Roman','FontSize',18,... %X-axis title and properties
'Color',[0.6 0.08 0.1]);
title('WindSpeed-Days Of Year ','FontName','Times New
Roman','FontSize',18,... %Graph title and properties
'Color',[0.6 0.08 0.1]);
box(axes1,'on');
grid on;
set(axes1,'FontName','Times New
Roman','FontSize',18,'FontWeight','bold','XColor',... %Axis properties
[0.6 0.08 0.1],'YColor',...
[0.6 0.08 0.1],'ZColor',...
[0.6 0.08 0.1]);
%% Clearing unrequired data
clear i j toDelete figure1 axes1;
clear startRow row formatSpec ;
%% Saving the graph
saveas(gcf,'..//Results//WindSpeedJulianDay.png'); %File path
```