**Q. 1) Find all the unique 'Wind Speed' values in the data.**

Code: data['Wind Speed\_km/h']**.**unique()

Output: array([ 9, 24, 26, 15, 4, 0, 19, 17, 11, 22, 35, 13, 20, 6, 7, 30, 32,

41, 39, 28, 44, 33, 37, 52, 46, 2, 50, 48, 57, 63, 43, 83, 70, 54],

dtype=int64)

Analysis part: The code data['Wind Speed\_km/h'] will access the wind speed\_km/hr column and then after selecting the column we have used unique( )function method using pandas . This method will return an array containing all unique values in the selected column.

**Q. 2) Find the number of times when the 'Weather is exactly Clear'.**

Code: weather\_count = (data['Weather'] == 'Clear').sum()

weather\_count

Output: 1326

Analysis part: data['Weather'] will access the weather column in the data set. It

selects the entire 'Weather' column, which contains weather condition for each entry in the dataset.

--(data['Weather'] == 'Clear') where this is a comparision operator it will cheques whether the weather column is having clear if it is there then it will

take.

--After that sum() method we used this will calculate the sum of the values in

the series if there then it will count the values which corresponds to the number of times 'Weather' is exactly 'Clear' in the dataset.

**Q. 3) Find the number of times when the 'Wind Speed was exactly 4 km/h'.**

Code: wind\_speed= (data['Wind Speed\_km/h'] == 4).sum()

wind\_speed

Output: 474

Analysis part: In the above code data[Wind Speed\_km/h’] will access the weather column in the data set. It selects the entire Wind column.

-- (data['Wind Speed\_km/h'] == 4) where this is a comparision operator it will cheque whether the wind speed is having the value of ‘4’ if it is true then it will take.

--After that sum() method will be used to calculate the number of times when the wind speed is exactly 4km/h.

**Q. 4) Find out all the Null Values in the data.**

Code: null\_values = data.isnull()

null\_column = null\_values.sum()

null\_column

total\_count = null\_column.sum()

total\_count

Output: 0

Analysis part: null\_values = data.isnull() in this whether each cell contains a Boolean value indicating whether the corresponding cell in the data is null or missing where true is nothing but we are having null value and false is not having a null value.

-- null\_column = null\_values.sum() this will tell the sum of true values in each column which are null

-- total\_count = null\_column.sum() this will tell the total count of missing values by adding the values in null\_column .so, it gives the full count where are the null values are present.

**Q. 5) Rename the column name 'Weather' of the dataframe to 'Weather Condition'.**

Code: data.rename (columns={‘Weather’: ‘Weather Condition’},inplcae=True

Data.head()

Analysis part: It renames the 'Weather' column to 'Weather Condition' for all rows in the DataFrame. Where the inplace= True parameter will tell you the modification you applied to the original data.

**Q. 6) What is the mean 'Visibility' ?**

Code: data**.**Visibility\_km**.**mean()

Output: 27.664446721311478

Analysis part: The above code will give us the single value where the visibility\_km column will give us the average of the total values. Where mean is the measure of central tendecny which will give us insights about the data set

**Q. 7) What is the Standard Deviation of 'Pressure' in this data?**

Code: data**.**Press\_kPa**.**std()

Output: 0.8440047459486483

Analysis part: The above code will give us the single value where the pressure column will give us the standard deviation. Where std dev is the measure of central tendency which will give us the insights about the data set.

**Q. 8) What is the Variance of 'Relative Humidity' in this data ?**

Code: data['Rel Hum\_%']**.**var()

Output: 286.248550198502

Analysis part: The above code will give the single value where the relative humidity column will give the variance of it. Where variance is the measure of dispersion/variability which will help us to give insights about the data set

**Q. 9) Find all instances when 'Snow' was recorded.**

Code: snow=data[data[‘Weather Condition’]==’Snow’]

snow

Analysis part: The [‘Weather Condition’] will access the weather column and by using comparision operator it will checks whether there is snow if it is true then it will represent wherever the snow was recorded.

**Q. 10) Find all instances when 'Wind Speed is above 24' and 'Visibility is 25'.**

Code: wind\_speed = data[(data['Wind Speed\_km/h'] > 24) & (data['Visibility\_km'] == 25)]

wind\_speed

Analysis part: In the above we are checking in the wind speed column where the wind speed is above 24 and where the visibility column is 25 by using arithmetic operator > and comparision operator == these will get us the all instances of above condition.

**Q. 11) What is the Mean value of each column against each 'Weather Condition ?**

Code: daat2= data.groupby('Weather Condition').mean()

daat2

Analysis part: The above code will groups into the unique values in the weather condition column it will create separate groups and then we are doing mean for that groups which will give us the average of all the values for each unique group .

**Q. 12) What is the Minimum & Maximum value of each column against each 'Weather Condition ?**

Code: max\_values=data.groupby('Weather Condition').max()

max\_values

min\_values=data.groupby('Weather Condition').min()

min\_values

Analysis part: The above code will groups into the unique values in the weather condition column it will create separate groups and then we are doing minimum and maximum for that groups which will give us the minimum and maximum of all the values for each unique group .

**Q. 13) Show all the Records where Weather Condition is Fog.**

Code: fogrecords = data[data['Weather Condition'] == 'Fog']

fogrecords

Analysis part: In the above code we are accessing the weather column first then we are checking using comparision whether fog is there if there then it is true if not then it is false and it will represent the true values.

**Q. 14) Find all instances when 'Weather is Clear' or 'Visibility is above 40'.**

Code: clear\_visibility = data[(data['Weather Condition'] == 'Clear') | (data['Visibility\_km'] > 40)]

clear\_visibility

Analysis part: In this code we are accessing the weather column in that column we are checking which are ‘clear’ or where in the visibility column which are >40 any one condition has to be satistied and we write this code using arithematic and comparision operator.

**Q. 15) Find all instances when :**

**A. 'Weather is Clear' and 'Relative Humidity is greater than 50'**

**or**

**B. 'Visibility is above 40'**

Code: data2 = data[((data['Weather Condition'] == 'Clear') & (data['Rel Hum\_%'] > 50)) | (data['Visibility\_km'] > 40)]

data2

Analysis part: In the above code we are accessing weather column first in that we are checking whether weather equal to clear by using comparision operator and also relative hum column which are > 50 or visibility column >40 the first condition to be satisfied the two and three condition any one has to be satisfied then we will get instances based on the conditions.