Report of Weather Data

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

Code: x=data["Wind Speed\_km/h"].unique()

Len(x)

Output: 34

Analysis: Here, we have used unique function to find the how many unique values are there in the wind speed.

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

Code: data.Weather.value\_counts().head(5)

output: Mainly Clear 2106

Mostly Cloudy 2069

Cloudy 1728

Clear 1326

Snow 390

Name: Weather, dtype: int64

Analysis: here, we have used value \_count to get the unique values of each object and it counts how many are there. And I used head to get only five of them instead of all.

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

Code: x1=data[data["Wind Speed\_km/h"]==4]

Len(x1)

Output:474

Analysis: here,by this we can get the how many number of times has wind speed is 4km/h. and length (len) to get certain number of times wind speed is 4km/h is repeated.

4. Find out all the Null Values in the data.

Code: data.notnull().sum()

Output: Date/Time 8784

Temp\_C 8784

Dew Point Temp\_C 8784

Rel Hum\_% 8784

Wind Speed\_km/h 8784

Visibility\_km 8784

Press\_kPa 8784

Weather 8784

dtype: int

Analysis: here, we have used notnull function to find how many non-null values present in the overall dataset. And sum() function to sum all values in the dataframe.

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

Code: data.rename(columns={"Weather":"Weather Condition"}.head(5)

output: Date/TimeTemp\_CDew Point Temp\_CRel Hum\_%Wind Speed\_km/hVisibility\_kmPress\_kPa Weather Condition

001-01-2012 00:00-1.3-3.518925.098.67 Clear

101-01-2012 01:007.42.8202424.199.37 Rain

201-01-2012 02:0015.713.4212625.099.84 Cloudy

301-01-2012 03:004.9-2.6271524.1100.94 Mainly Clear

401-01-2012 04:00-13.4-19.730425.0102.32 Mostly Cloudy

Analysis: here, we have used the rename() function to the certain weather column to weather condition. And I have used head(5) to get only first five values

6. What is the mean 'Visibility'?

Code: data.Visibility\_km.mean()

Output: 27.664446721311478

Analysis: here first we go through the data and then go to the visibility column and calculate the mean by using the mean() function.

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

Code: data.Press\_kPa.std()

Output: 0.8440047459486459

Analysis: here we have used std() function to get the standard deviation of pressure. First it goes to the data and then go to the press\_kpa column and find standard deviation by using std () funct-ion.

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

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

Output: 286.2485501985015

Analysis: here,we have used var() function to find the variance of the relative humidity of dataset.

At first it goes thorough the dataset and go to the Rel hum column after that it find the variance.

9. Find all instances when 'Snow' was recorded.

Code: data[data['Weather Condition'].str.contains('Snow')]

| **Date/Time** | **Temp\_C** | **Dew Point Temp\_C** | **Rel Hum\_%** | **Wind Speed\_km/h** | **Visibility\_km** | **Press\_kPa** | **Weather** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **11** | 01-01-2012 11:00 | -6.2 | -9.6 | 37 | 35 | 4.8 | 101.56 | Snow |
| **70** | 03-01-2012 22:00 | -4.0 | -6.6 | 62 | 22 | 16.1 | 100.48 | Snow |
| **73** | 04-01-2012 01:00 | 2.3 | -3.4 | 64 | 35 | 25.0 | 103.43 | Snow |
| **102** | 05-01-2012 06:00 | -7.6 | -9.4 | 70 | 32 | 3.2 | 100.56 | Snow,Fog |
| **105** | 05-01-2012 09:00 | -1.8 | -4.2 | 73 | 15 | 6.4 | 101.28 | Snow |

Output:

Analysis: here, we have used the str.contains () function to find the certain object in the column .at first it goes to the dataset and then it go to the weather column then gives the how many times snow has been recorded.

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

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

Output:

| **Date/Time** | **Temp\_C** | **Dew Point Temp\_C** | **Rel Hum\_%** | **Wind Speed\_km/h** | **Visibility\_km** | **Press\_kPa** | **Weather Condition** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **2** | 01-01-2012 02:00 | 15.7 | 13.4 | 21 | 26 | 25.0 | 99.84 | Cloudy |
| **73** | 04-01-2012 01:00 | 2.3 | -3.4 | 64 | 35 | 25.0 | 103.43 | Snow |
| **126** | 06-01-2012 06:00 | 10.0 | 5.4 | 77 | 39 | 25.0 | 101.30 | Cloudy |
| **158** | 07-01-2012 14:00 | 1.9 | -2.1 | 87 | 26 | 25.0 | 100.87 | Rain,Snow Grains |
| **184** | 08-01-2012 16:00 | 14.2 | 9.2 | 35 | 44 | 25.0 | 99.49 | Mostly Cloudy |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **8707** | 9/27/2012 5:00 | -1.0 | -6.0 | 70 | 33 | 25.0 | 98.56 | Mostly Cloudy |
| **8714** | 9/28/2012 10:00 | 2.6 | 0.3 | 72 | 26 | 25.0 | 101.60 | Rain |
| **8738** | 9/29/2012 10:00 | 22.8 | 12.3 | 80 | 28 | 25.0 | 101.60 | Mostly Cloudy |
| **8745** | 9/29/2012 17:00 | -10.3 | -12.9 | 82 | 28 | 25.0 | 102.16 | Cloudy |
| **8776** | 9/30/2012 23:00 | 19.2 | 13.2 | 93 | 43 | 25.0 | 101.60 | Mainly Clear |

308 rows × 8 columns

Analysis: here, we have used And (&) operator to get the 'Wind Speed is above 24' and 'Visibility is 25’, at first it goes through the dataset and check how many times are there the wind speed is above 24 and it also checks the visibility is 25 then it will execute the output with that condition.

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

Code: data.groupby('Weather').mean().head(15)

output:

| **Temp\_C** | **Dew Point Temp\_C** | **Rel Hum\_%** | **Wind Speed\_km/h** | **Visibility\_km** | **Press\_kPa** |
| --- | --- | --- | --- | --- | --- |
| **Weather** |  |  |  |  |  |  |
| **Clear** | 6.825716 | 0.089367 | 67.127451 | 10.557315 | 30.153243 | 101.084495 |
| **Cloudy** | 7.970544 | 2.375810 | 67.349537 | 16.127315 | 26.625752 | 101.056852 |
| **Drizzle** | 7.353659 | 5.504878 | 69.048780 | 16.097561 | 17.931707 | 101.099268 |
| **Drizzle,Fog** | 8.067500 | 7.033750 | 70.062500 | 11.862500 | 5.257500 | 100.820750 |
| **Drizzle,Ice Pellets,Fog** | 0.400000 | -0.700000 | 52.000000 | 20.000000 | 4.000000 | 99.440000 |
| **Drizzle,Snow** | 1.050000 | 0.150000 | 44.000000 | 14.000000 | 10.500000 | 100.490000 |
| **Drizzle,Snow,Fog** | 0.693333 | 0.120000 | 69.800000 | 15.533333 | 5.513333 | 100.971333 |
| **Fog** | 4.303333 | 3.159333 | 66.466667 | 7.946667 | 6.248000 | 101.149400 |
| **Freezing Drizzle** | -5.657143 | -8.000000 | 68.857143 | 16.571429 | 9.200000 | 101.070000 |
| **Freezing Drizzle,Fog** | -2.533333 | -4.183333 | 64.000000 | 17.000000 | 5.266667 | 100.851667 |
| **Freezing Drizzle,Haze** | -5.433333 | -8.000000 | 63.333333 | 10.333333 | 2.666667 | 101.136667 |
| **Freezing Drizzle,Snow** | -5.109091 | -7.072727 | 62.454545 | 16.272727 | 5.872727 | 100.380909 |
| **Freezing Fog** | -7.575000 | -9.250000 | 68.000000 | 4.750000 | 0.650000 | 101.222500 |
| **Freezing Rain** | -3.885714 | -6.078571 | 60.785714 | 19.214286 | 8.242857 | 101.500714 |
| **Freezing Rain,Fog** | -2.225000 | -3.750000 | 52.750000 | 15.500000 | 7.550000 | 100.267500 |

Analysis: here, we have used two function that are groupby() and mean() .groupby is used to go through the entire column and mean function is used to find the mean of the objects in the column.

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

For min()

Code: data.groupby('Weather Condition').min().head(10)

Output:

| **Date/Time** | **Temp\_C** | **Dew Point Temp\_C** | **Rel Hum\_%** | **Wind Speed\_km/h** | **Visibility\_km** | **Press\_kPa** |
| --- | --- | --- | --- | --- | --- | --- |
| **Weather** |  |  |  |  |  |  |  |
| **Clear** | 01-01-2012 00:00 | -23.3 | -28.5 | 18 | 0 | 11.3 | 97.75 |
| **Cloudy** | 01-01-2012 02:00 | -21.4 | -26.8 | 20 | 0 | 11.3 | 97.52 |
| **Drizzle** | 01-06-2012 08:00 | 1.1 | -0.2 | 37 | 0 | 6.4 | 98.29 |
| **Drizzle,Fog** | 01-04-2012 01:00 | 0.0 | -1.6 | 38 | 0 | 1.0 | 98.32 |
| **Drizzle,Ice Pellets,Fog** | 7/24/2012 5:00 | 0.4 | -0.7 | 52 | 20 | 4.0 | 99.44 |
| **Drizzle,Snow** | 05-02-2012 09:00 | 0.9 | 0.1 | 39 | 9 | 9.7 | 100.27 |
| **Drizzle,Snow,Fog** | 03-11-2012 20:00 | 0.3 | -0.1 | 46 | 7 | 2.4 | 99.26 |
| **Fog** | 01-01-2012 13:00 | -16.0 | -17.2 | 21 | 0 | 0.2 | 97.97 |
| **Freezing Drizzle** | 04-01-2012 03:00 | -9.0 | -12.2 | 43 | 6 | 4.8 | 99.75 |
| **Freezing Drizzle,Fog** | 10/15/2012 4:00 | -6.4 | -9.0 | 31 | 6 | 3.6 | 98.81 |

Analysis: here,we have used groupby for the go through the entire column and min() function used to find the minimum value of the weather. Head is used get only some values of the dataset

For max()

Code: data.groupby('Weather Condition').max().head(10)

Output:

| **Date/Time** | **Temp\_C** | **Dew Point Temp\_C** | **Rel Hum\_%** | **Wind Speed\_km/h** | **Visibility\_km** | **Press\_kPa** |
| --- | --- | --- | --- | --- | --- | --- |
| **Weather** |  |  |  |  |  |  |  |
| **Clear** | 9/30/2012 7:00 | 32.8 | 20.4 | 100 | 33 | 48.3 | 103.63 |
| **Cloudy** | 9/30/2012 8:00 | 30.5 | 22.6 | 100 | 54 | 48.3 | 103.52 |
| **Drizzle** | 9/15/2012 22:00 | 18.8 | 17.7 | 97 | 30 | 25.0 | 103.58 |
| **Drizzle,Fog** | 9/19/2012 15:00 | 19.9 | 19.1 | 98 | 28 | 9.7 | 103.56 |
| **Drizzle,Ice Pellets,Fog** | 7/24/2012 5:00 | 0.4 | -0.7 | 52 | 20 | 4.0 | 99.44 |
| **Drizzle,Snow** | 3/17/2012 1:00 | 1.2 | 0.2 | 49 | 19 | 11.3 | 100.71 |
| **Drizzle,Snow,Fog** | 9/21/2012 12:00 | 1.1 | 0.6 | 94 | 32 | 9.7 | 102.47 |
| **Fog** | 9/30/2012 19:00 | 20.8 | 19.6 | 99 | 22 | 9.7 | 103.22 |
| **Freezing Drizzle** | 8/21/2012 5:00 | -2.3 | -3.3 | 89 | 26 | 12.9 | 101.78 |
| **Freezing Drizzle,Fog** | 7/26/2012 6:00 | -0.3 | -2.3 | 80 | 33 | 8.0 | 103.01 |

Analysis: here,we have used groupby for the go through the entire column and max() function used to find the maximum value of the weather. Head is used get only some values of the dataset.

13. Show all the Records where Weather Condition is Fog.

Code: data[data["Weather Condition"]=='Fog'].head(5)

Output: Date/TimeTemp\_CDewPoinTemp\_CRelHum\_%WindSpeed\_km/hVisibility\_kmPress\_kPaWeather

13 01-01-2012 13:009.57.840136.4100.90 Fog

53 03-01-2012 05:00-3.6-4.35779.7101.32 Fog

136 06-01-2012 16:0014.813.580199.7100.86 Fog

197 09-01-2012 05:002.10.743118.0101.44 Fog

278 12-01-2012 14:001.20.670136.4103.22 Fog

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

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

Output:

| **Date/Time** | **Temp\_C** | **Dew Point Temp\_C** | **Rel Hum\_%** | **Wind Speed\_km/h** | **Visibility\_km** | **Press\_kPa** | **Weather Condition** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 01-01-2012 00:00 | -1.3 | -3.5 | 18 | 9 | 25.0 | 98.67 | Clear |
| **9** | 01-01-2012 09:00 | 20.0 | 3.8 | 35 | 17 | 48.3 | 100.11 | Clear |
| **16** | 01-01-2012 16:00 | 23.8 | 17.6 | 42 | 9 | 25.0 | 100.52 | Clear |
| **17** | 01-01-2012 17:00 | -6.8 | -9.8 | 42 | 20 | 48.3 | 100.76 | Mainly Clear |
| **18** | 01-01-2012 18:00 | 2.3 | -2.4 | 42 | 6 | 48.3 | 101.05 | Cloudy |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **8774** | 9/30/2012 21:00 | 23.0 | 14.7 | 92 | 13 | 48.3 | 101.93 | Mostly Cloudy |
| **8777** | 9/30/2012 3:00 | 9.3 | 5.8 | 95 | 9 | 48.3 | 101.25 | Mainly Clear |
| **8779** | 9/30/2012 5:00 | 1.4 | -3.7 | 97 | 22 | 48.3 | 100.16 | Cloudy |
| **8780** | 9/30/2012 6:00 | -4.6 | -9.5 | 98 | 11 | 48.3 | 101.46 | Mostly Cloudy |
| **8781** | 9/30/2012 7:00 | 1.5 | -6.3 | 99 | 30 | 24.1 | 101.48 | Clear |

3027 rows × 8 columns

Analysis: here we have use Or (|) function between 'Weather is Clear' or 'Visibility is above 40'. Here it will give the out either weather is clear or visibility above 40. Some times it can consider both. Find all instances when :

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

or

B. 'Visibility is above 40'

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

Output:

| **Date/Time** | **Temp\_C** | **Dew Point Temp\_C** | **Rel Hum\_%** | **Wind Speed\_km/h** | **Visibility\_km** | **Press\_kPa** | **Weather Condition** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **9** | 01-01-2012 09:00 | 20.0 | 3.8 | 35 | 17 | 48.3 | 100.11 | Clear |
| **17** | 01-01-2012 17:00 | -6.8 | -9.8 | 42 | 20 | 48.3 | 100.76 | Mainly Clear |
| **18** | 01-01-2012 18:00 | 2.3 | -2.4 | 42 | 6 | 48.3 | 101.05 | Cloudy |
| **19** | 01-01-2012 19:00 | -12.7 | -17.2 | 43 | 17 | 48.3 | 101.16 | Clear |
| **23** | 01-01-2012 23:00 | 29.5 | 16.8 | 45 | 4 | 48.3 | 101.07 | Mainly Clear |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **8774** | 9/30/2012 21:00 | 23.0 | 14.7 | 92 | 13 | 48.3 | 101.93 | Mostly Cloudy |
| **8777** | 9/30/2012 3:00 | 9.3 | 5.8 | 95 | 9 | 48.3 | 101.25 | Mainly Clear |
| **8779** | 9/30/2012 5:00 | 1.4 | -3.7 | 97 | 22 | 48.3 | 100.16 | Cloudy |
| **8780** | 9/30/2012 6:00 | -4.6 | -9.5 | 98 | 11 | 48.3 | 101.46 | Mostly Cloudy |
| **8781** | 9/30/2012 7:00 | 1.5 | -6.3 | 99 | 30 | 24.1 | 101.48 | Clear |

2864 rows × 8 columns

Analysis: here ,we have used the both And (&) and Or(|) between 'Weather is Clear', 'Relative Humidity is greater than 50' , 'Visibility is above 40'. Here it will consider the and(&) operator which is used and some times or(|) operator is consider or maybe not.