Weather Forecasting

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
df=pd.read_csv('forecast_data.csv')
df.head()
```

	time_epoch	time	temp_c	temp_f	is_day	condition	wind_mph	wind_kph
0	1634236200	2021-10-15 00:00	23.2	73.8	0	{'text': 'Clear', 'icon': '//cdn.weatherapi.co...	4.3	6.8
1	1634239800	2021-10-15 01:00	23.0	73.3	0	{'text': 'Clear', 'icon': '//cdn.weatherapi.co...	4.1	6.6
2	1634243400	2021-10-15 02:00	22.7	72.9	0	{'text': 'Clear', 'icon': '//cdn.weatherapi.co...	4.0	6.4
3	1634247000	2021-10-15 03:00	22.5	72.5	0	{'text': 'Clear', 'icon': '//cdn.weatherapi.co...	3.8	6.1
4	1634250600	2021-10-15 04:00	22.3	72.1	0	{'text': 'Clear', 'icon': '//cdn.weatherapi.co...	3.7	6.0

5 rows × 34 columns

```
df.columns
```

```
Index(['time_epoch', 'time', 'temp_c', 'temp_f', 'is_day', 'condition',
       'wind_mph', 'wind_kph', 'wind_degree', 'wind_dir', 'pressure_mb',
       'pressure_in', 'precip_mm', 'precip_in', 'humidity', 'cloud',
       'feelslike_c', 'feelslike_f', 'windchill_c', 'windchill_f',
       'heatindex_c', 'heatindex_f', 'dewpoint_c', 'dewpoint_f',
       'will_it_rain', 'chance_of_rain', 'will_it_snow', 'chance_of_snow',
       'vis_km', 'vis_miles', 'gust_mph', 'gust_kph', 'state', 'city'],
      dtype='object')
```

```
df.shape
```

```
(29568, 34)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29568 entries, 0 to 29567
Data columns (total 34 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   time_epoch      29568 non-null  int64  
 1   time            29568 non-null  object 
 2   temp_c          29568 non-null  float64
 3   temp_f          29568 non-null  float64
 4   is_day          29568 non-null  int64  
 5   condition       29568 non-null  object 
 6   wind_mph        29568 non-null  float64
 7   wind_kph        29568 non-null  float64
 8   wind_degree     29568 non-null  int64  
 9   wind_dir        29568 non-null  object 
 10  pressure_mb     29568 non-null  float64
 11  pressure_in     29568 non-null  float64
 12  precip_mm       29568 non-null  float64
 13  precip_in       29568 non-null  float64
 14  humidity        29568 non-null  int64  
 15  cloud           29568 non-null  int64  
 16  feelslike_c     29568 non-null  float64
 17  feelslike_f     29568 non-null  float64
 18  windchill_c     29568 non-null  float64
 19  windchill_f     29568 non-null  float64
 20  heatindex_c     29568 non-null  float64
 21  heatindex_f     29568 non-null  float64
```

```
 22  dewpoint_c      29568 non-null  float64
 23  dewpoint_f      29568 non-null  float64
 24  will_it_rain    29568 non-null  int64
 25  chance_of_rain  29568 non-null  int64
 26  will_it_snow    29568 non-null  int64
 27  chance_of_snow  29568 non-null  int64
 28  vis_km          29568 non-null  float64
 29  vis_miles       29568 non-null  float64
 30  gust_mph        29568 non-null  float64
 31  gust_kph        29568 non-null  float64
 32  state           29568 non-null  object
 33  city            29568 non-null  object
dtypes: float64(20), int64(9), object(5)
memory usage: 7.7+ MB
```

```
df.describe()
```

	time_epoch	temp_c	temp_f	is_day	wind_mph	wind_kph	wind_degree	pressure
count	2.956800e+04	29568.000000	29568.000000	29568.000000	29568.000000	29568.000000	29568.000000	29568.000
mean	1.634580e+09	25.152307	77.273793	0.477070	5.077486	8.169619	177.818452	1008.341
std	1.995536e+05	4.797702	8.635461	0.499482	2.847123	4.582337	103.545331	3.438
min	1.634227e+09	1.500000	34.700000	0.000000	0.000000	0.000000	0.000000	999.000
25%	1.634408e+09	22.500000	72.600000	0.000000	3.100000	4.900000	83.000000	1006.000
50%	1.634580e+09	25.100000	77.200000	0.000000	4.500000	7.200000	174.000000	1008.000
75%	1.634753e+09	28.100000	82.600000	1.000000	6.500000	10.400000	273.000000	1011.000
max	1.634958e+09	39.300000	102.700000	1.000000	22.400000	36.000000	359.000000	1021.000

8 rows × 29 columns

Data Cleaning

```
df[["Year","Month",'Day']] = df['time'].str.split('-', n = 4, expand = True)
```

```
df[['Date','Time']]=df['Day'].str.split(' ',n=1,expand=True)
```

Dropping unwanted Columns

```
df.drop(['time','Day','chance_of_snow','will_it_snow','condition'], axis = 1, inplace = True)
```

```
df.head()
```

	time_epoch	temp_c	temp_f	is_day	wind_mph	wind_kph	wind_degree	wind_dir	pressure_mb	pressure_in	...
0	1634236200	23.2	73.8	0	4.3	6.8	41	NE	1006.0	29.69	...
1	1634239800	23.0	73.3	0	4.1	6.6	34	NE	1005.0	29.68	...
2	1634243400	22.7	72.9	0	4.0	6.4	28	NNE	1005.0	29.67	...
3	1634247000	22.5	72.5	0	3.8	6.1	22	NNE	1005.0	29.66	...
4	1634250600	22.3	72.1	0	3.7	6.0	23	NNE	1005.0	29.68	...

5 rows × 34 columns

Checking for Null values

```
df.isna().sum()
```

```
time_epoch        0
temp_c            0
temp_f            0
```

```
is_day            0
wind_mph          0
wind_kph          0
wind_degree       0
wind_dir          0
pressure_mb       0
pressure_in       0
precip_mm         0
precip_in         0
humidity          0
cloud             0
feelslike_c       0
feelslike_f       0
windchill_c       0
windchill_f       0
heatindex_c       0
heatindex_f       0
dewpoint_c        0
dewpoint_f        0
will_it_rain      0
chance_of_rain    0
vis_km            0
vis_miles         0
gust_mph          0
gust_kph          0
state             0
city              0
Year              0
Month             0
Date              0
Time              0
dtype: int64
```

Checking for Duplicates

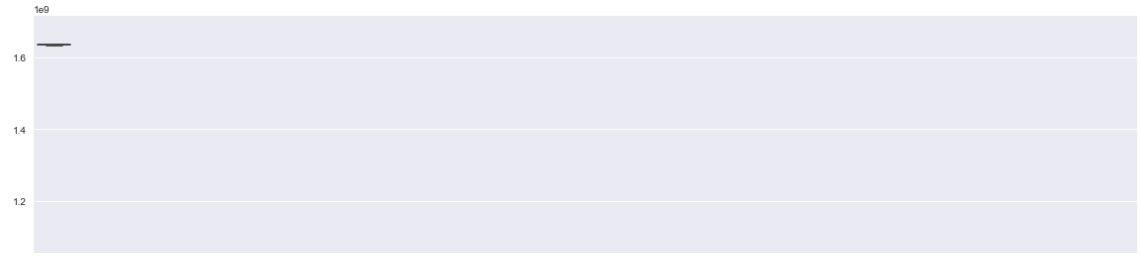
```
df.duplicated().sum()
```

```
0
```

There are no duplicates in the dataset

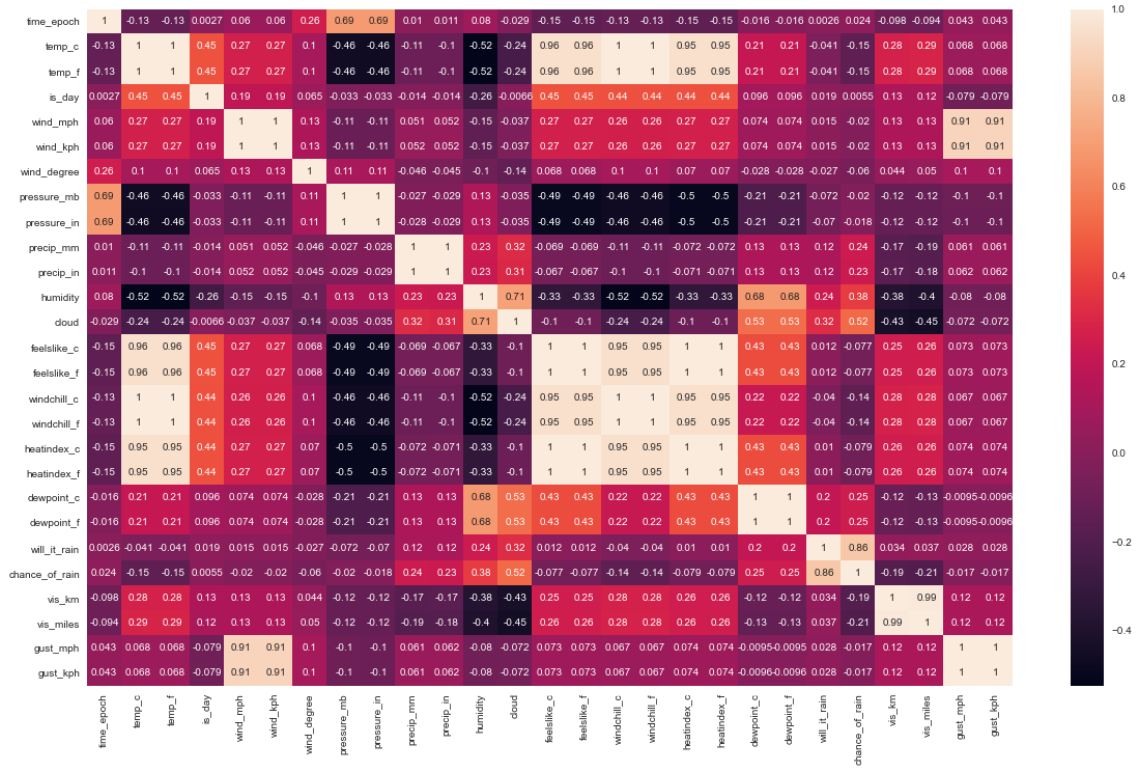
Checking for Outliers

```
plt.figure(figsize=(20,12))
sns.boxplot(data=df)
plt.xticks(rotation=90)
plt.show()
```




```
plt.figure(figsize=(20,12))
sns.heatmap(df.corr(), annot=True)
```

```
<AxesSubplot:>
```




```
df.corr().style.background_gradient()
```

	time_epoch	temp_c	temp_f	is_day	wind_mph	wind_kph	wind_degree	pressure_mb	pressure_in	precip_mm	pre
time_epoch	1.000000	-0.131612	-0.131587	0.002710	0.060042	0.060070	0.259429	0.686185	0.688035	0.010479	0.0
temp_c	-0.131612	1.000000	0.999984	0.448089	0.267645	0.267521	0.102912	-0.459322	-0.460847	-0.106922	-0.1
temp_f	-0.131587	0.999984	1.000000	0.448129	0.267647	0.267522	0.102901	-0.459302	-0.460834	-0.106886	-0.1
is_day	0.002710	0.448089	0.448129	1.000000	0.194623	0.194676	0.065103	-0.032703	-0.033080	-0.014205	-0.(
wind_mph	0.060042	0.267645	0.267647	0.194623	1.000000	0.999929	0.127415	-0.108492	-0.109661	0.051496	0.0
wind_kph	0.060070	0.267521	0.267522	0.194676	0.999929	1.000000	0.127490	-0.108387	-0.109553	0.051545	0.0

```
df_num = df.select_dtypes(["int64","float64"])
df_cat = df.select_dtypes("object")
```

```
df_num
```

	time_epoch	temp_c	temp_f	is_day	wind_mph	wind_kph	wind_degree	pressure_mb	pressure_in	precip_mm
0	1634236200	23.2	73.8	0	4.3	6.8	41	1006.0	29.69	0.0
1	1634239800	23.0	73.3	0	4.1	6.6	34	1005.0	29.68	0.0
2	1634243400	22.7	72.9	0	4.0	6.4	28	1005.0	29.67	0.0
3	1634247000	22.5	72.5	0	3.8	6.1	22	1005.0	29.66	0.0
4	1634250600	22.3	72.1	0	3.7	6.0	23	1005.0	29.68	0.0
...	...	...	...	...	...	...	...	...	...	...
29563	1634909400	25.7	78.3	0	3.0	4.8	334	1010.0	29.84	0.0
29564	1634913000	24.8	76.6	0	3.1	4.9	342	1011.0	29.85	0.0
29565	1634916600	23.9	75.0	0	3.1	5.0	351	1012.0	29.87	0.0
29566	1634920200	23.3	74.0	0	2.8	4.4	237	1012.0	29.87	0.0
29567	1634923800	22.8	73.0	0	2.4	3.8	123	1012.0	29.87	0.0

29568 rows × 27 columns

```
df_cat
```

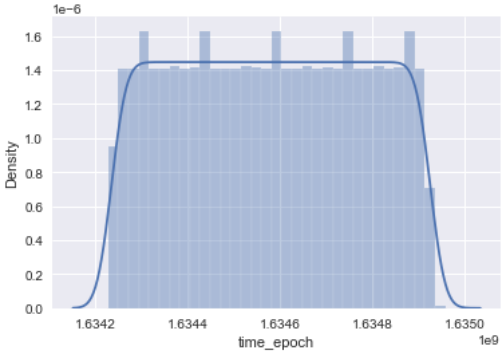
	wind_dir	state	city	Year	Month	Date	Time
0	NE	Andhra Pradesh	Amaravati	2021	10	15	00:00
1	NE	Andhra Pradesh	Amaravati	2021	10	15	01:00
2	NNE	Andhra Pradesh	Amaravati	2021	10	15	02:00
3	NNE	Andhra Pradesh	Amaravati	2021	10	15	03:00
4	NNE	Andhra Pradesh	Amaravati	2021	10	15	04:00
...	...	...	...	...	...	...	...
29563	NNW	West Bengal	Durgapur	2021	10	22	19:00
29564	NNW	West Bengal	Durgapur	2021	10	22	20:00
29565	N	West Bengal	Durgapur	2021	10	22	21:00
29566	WSW	West Bengal	Durgapur	2021	10	22	22:00
29567	ESE	West Bengal	Durgapur	2021	10	22	23:00

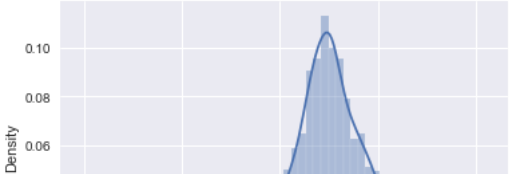
29568 rows × 7 columns

Skewness

```
from scipy.stats import skew
for col in df_num:
    print(col)
    print(skew(df_num[col]))
    plt.figure()
    sns.distplot(df_num[col])
```

```
time_epoch
3.175869101175923e-05
temp_c
-0.3732350532008425
temp_f
-0.3732736912356756
is_day
0.0918173840705094
wind_mph
1.2854757652672613
wind_kph
1.2857984833134077
wind_degree
0.044109079235098256
pressure_mb
0.018272741386440163
pressure_in
0.016012249597430347
precip_mm
15.459624434131015
precip_in
15.204741238214728
humidity
-0.8660316101695934
cloud
0.21478995150923844
feelslike_c
-0.38565380795654086
feelslike_f
-0.38560130202689746
windchill_c
-0.467966164412389
windchill_f
-0.46809216323036684
heatindex_c
-0.3622367376918974
heatindex_f
-0.36211956097620884
dewpoint_c
-1.4228260237210473
dewpoint_f
-1.4228636014673481
will_it_rain
2.8668422413037247
chance_of_rain
2.0078870025899476
vis_km
-3.214396102575789
vis_miles
-2.9061022138012143
gust_mph
0.9620650249283084
gust_kph
0.9616097951777816
```





```
df["precip_mm"]=np.sqrt(df["precip_mm"])
```

```
skew(df['precip_mm'])
```

```
3.1280984931678013
```

```
df["precip_in"]=np.sqrt(df["precip_in"])
```

```
skew(df['precip_in'])
```

```
3.2294794265785254
```

Data Exploration

Pandas Profiling

```
from pandas_profiling import ProfileReport
```

```
profile = ProfileReport(df, title="Pandas Profiling Report")
```

```
profile
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

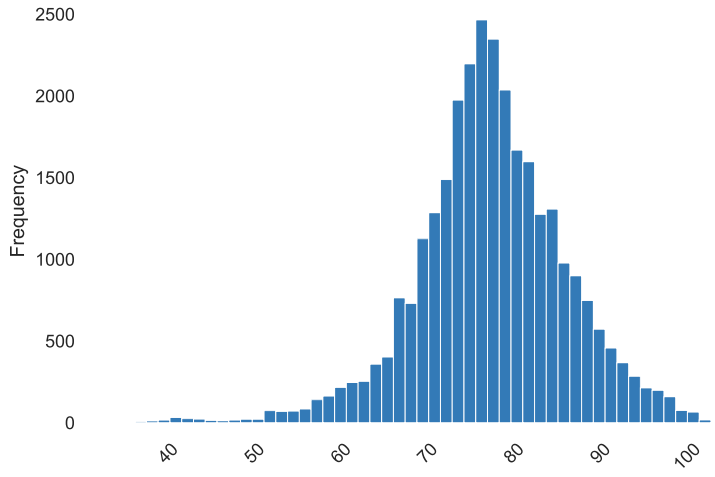
(time)

Quantile statistics

Minimum	34.7
5-th percentile	63.1
Q1	72.6
median	77.2
Q3	82.6
95-th percentile	91.5
Maximum	102.7
Range	68
Interquartile range (IQR)	10

Descriptive statistics

Standard deviation	8.635460579
Coefficient of variation (CV)	0.1117514786
Kurtosis	1.424014913
Mean	77.27379261
Median Absolute Deviation (MAD)	5
Skewness	-0.3732926288
Sum	2284831.5
Variance	74.57117942
Monotonicity	Not monotonic



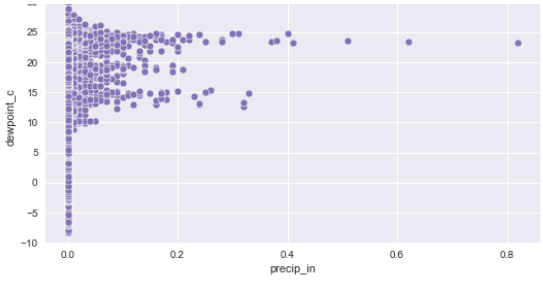
AutoViz

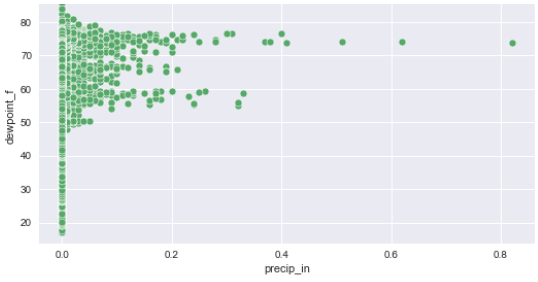
```
from autoviz.AutoViz_Class import AutoViz_Class
```

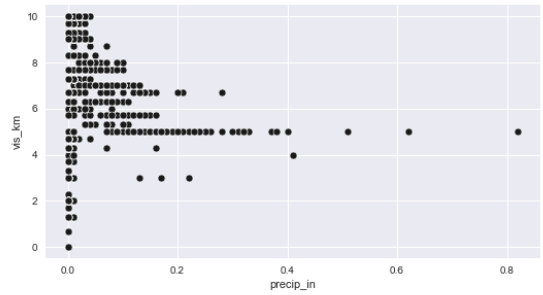
```
AV = AutoViz_Class()
```

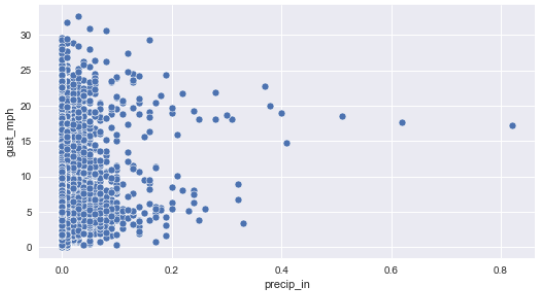
```
     Imported AutoViz_Class version: 0.0.84. Call using:
         AV = AutoViz_Class()
         AV.AutoViz(filename, sep=',', depVar='', dfte=None, header=0, verbose=0,
                                 lowess=False,chart_format='svg',max_rows_analyzed=150000,max_cols_analyzed=30)
     Note: verbose=0 or 1 generates charts and displays them in your local Jupyter notebook.
           verbose=2 does not show plot but creates them and saves them in AutoViz_Plots directory in your local machine.
```

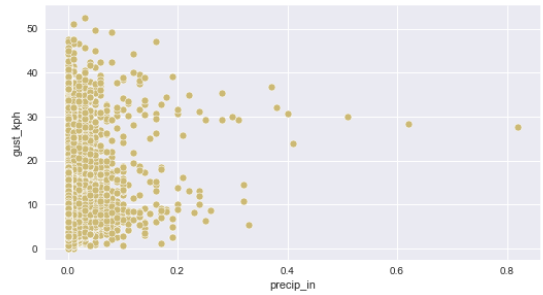
```
df = AV.AutoViz('forecast_data.csv')
```

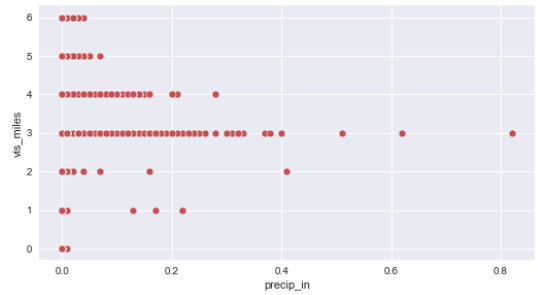

dewpoint_c
precip_in

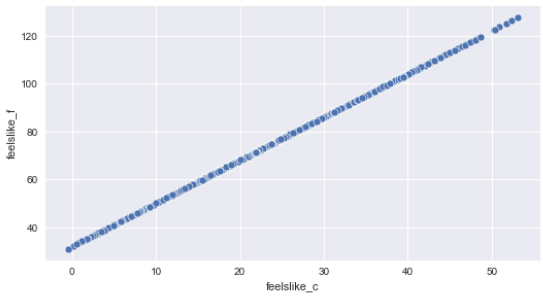

dewpoint_f
precip_in

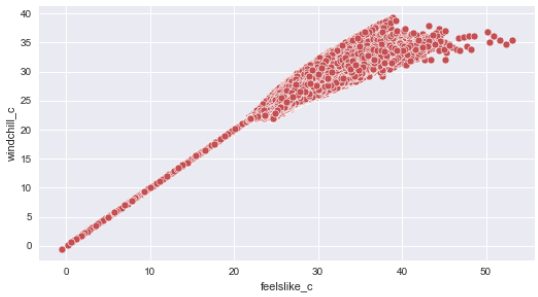

vis_km
precip_in

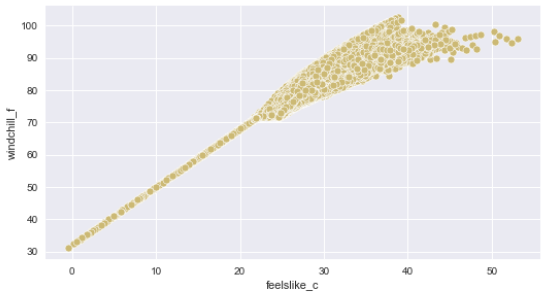

gust_mph
precip_in

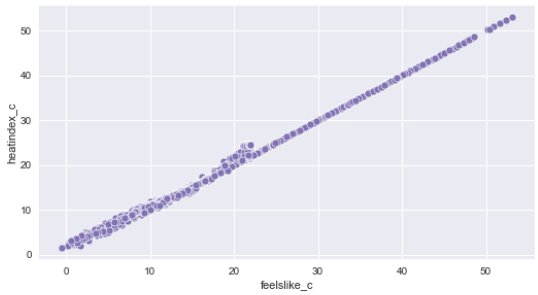

gust_kph
precip_in

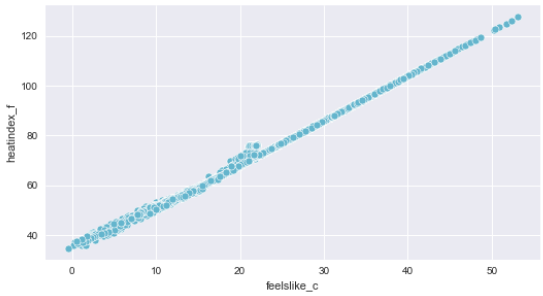

vis_miles
precip_in

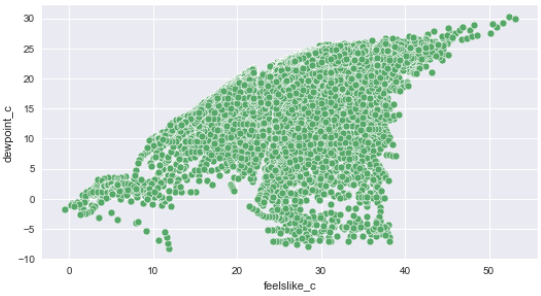

feelslike_f
feelslike_c


windchill_c
feelslike_c


windchill_f
feelslike_c


heatindex_c
feelslike_c

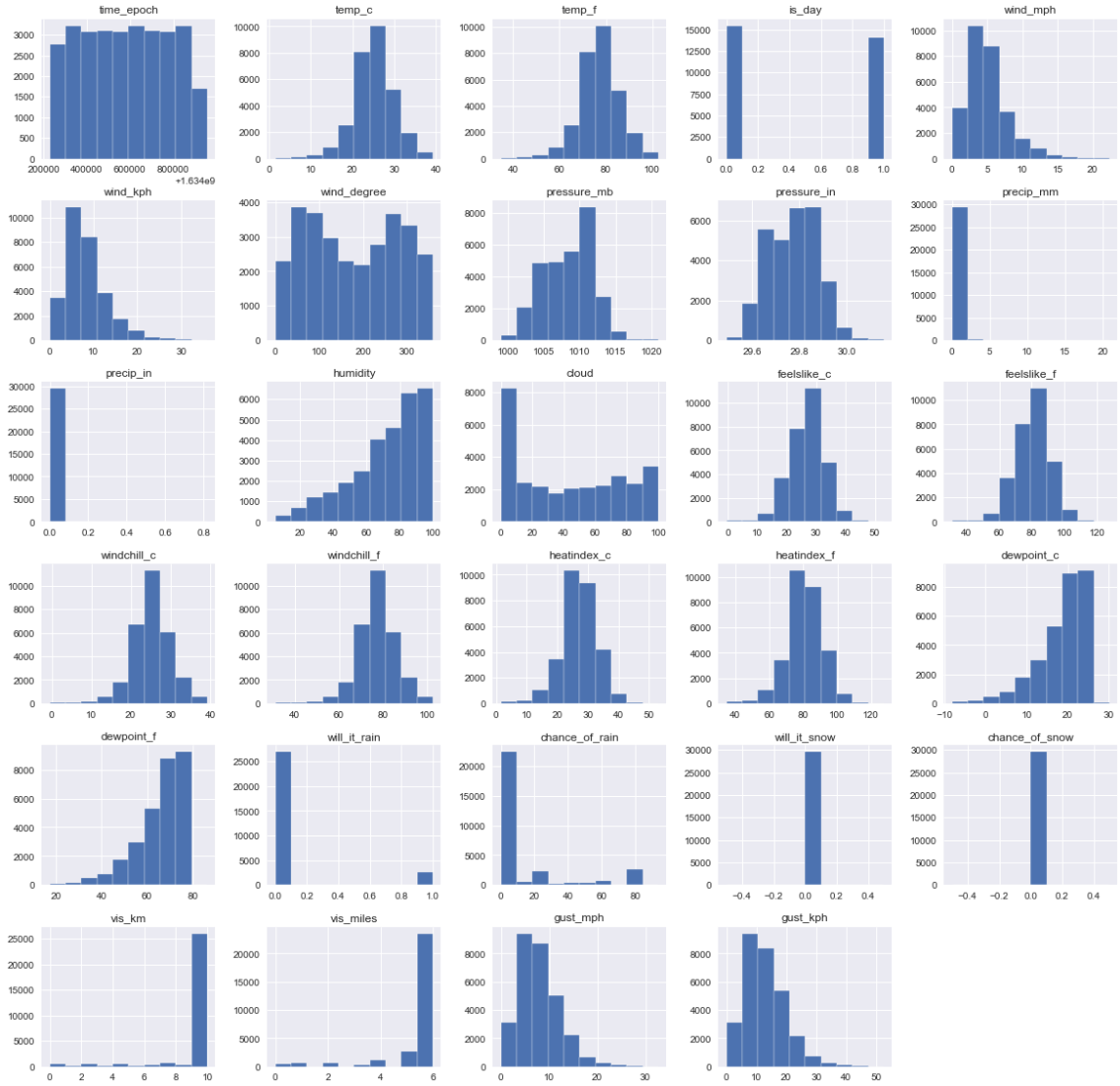

heatindex_f
feelslike_c


dewpoint_c
feelslike_c

EDA

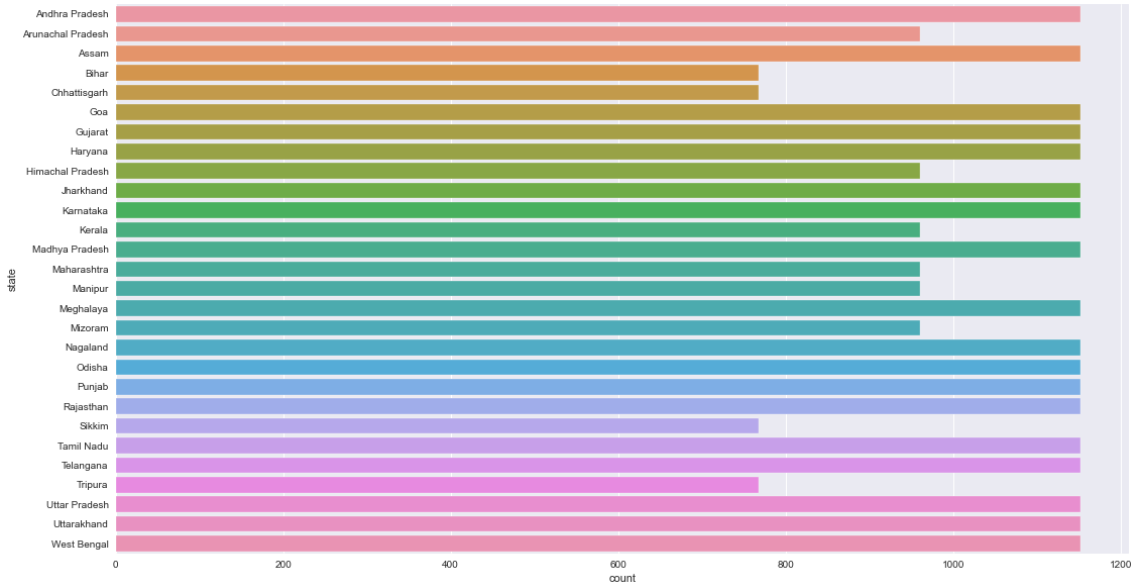
Univeriate Analysis

```
df.hist(figsize=(20,20))
plt.show()
```




```
plt.figure(figsize=(18,10))
sns.countplot(data=df, y='state')
```

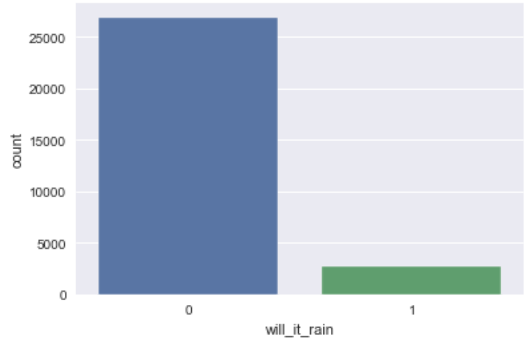
```
<AxesSubplot:xlabel='count', ylabel='state'>
```



we see that, number of predections done for state Sikkam Tripura Bihar Chattisgarh

```
sns.countplot(data=df, x='will_it_rain')
```

```
<AxesSubplot:xlabel='will_it_rain', ylabel='count'>
```



Unsupported Cell Type. Double-Click to inspect/edit the content.

Biveriate Analysis

```
figure = plt.figure(figsize=(15,8))
sns.lineplot(x="windchill_c",y="chance_of_rain",data=df)
plt.show()
```

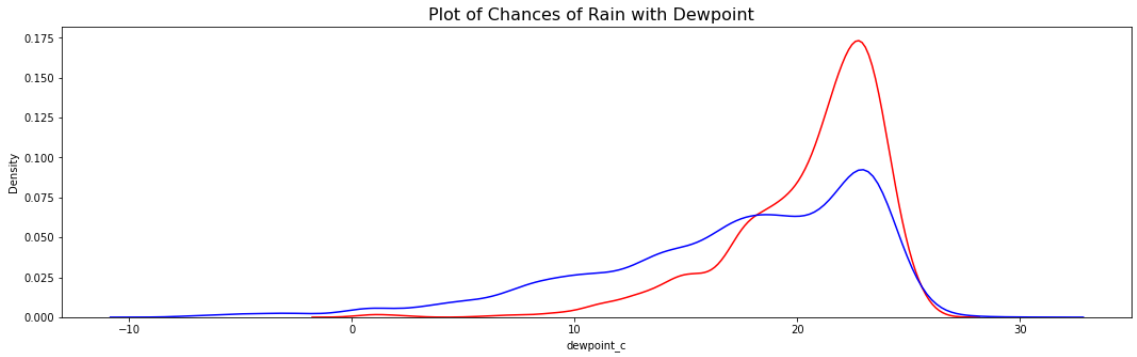
If windchill is less than 5 chances of rain is high.

```
group=df.groupby(["state","is_day"]).mean().head(20)
group
```

state	is_day	time_epoch	temp_c	temp_f	wind_mph	wind_kph	wind_degree	pressure_mb	pressure_
Andhra Pradesh	0	1.634579e+09	25.534740	77.960552	4.129870	6.646429	226.665584	1006.969156	29.7348
	1	1.634581e+09	29.664552	85.396642	6.163619	9.918657	220.436567	1006.740672	29.7283
Arunachal Pradesh	0	1.634581e+09	20.203320	68.366406	3.538477	5.688867	139.683594	1010.156250	29.8291
	1	1.634578e+09	22.617187	72.711384	3.653571	5.882589	118.473214	1010.066964	29.8268
Assam	0	1.634582e+09	24.254968	75.658333	3.017788	4.852885	119.362179	1008.267628	29.7735
	1	1.634578e+09	27.483333	81.469697	3.989583	6.423674	116.986742	1008.073864	29.7672
Bihar	0	1.634580e+09	24.997396	76.994010	6.904167	11.108854	164.132812	1006.671875	29.7270
	1	1.634580e+09	28.311458	82.958333	8.172656	13.144792	161.315104	1006.721354	29.7279
Chhattisgarh	0	1.634583e+09	23.289567	73.920356	4.194148	6.746310	195.005089	1007.002545	29.7355
	1	1.634577e+09	28.212533	82.783467	5.103200	8.206133	179.896000	1006.658667	29.7261
Goa	0	1.634576e+09	25.534722	77.961806	4.554340	7.322396	136.762153	1008.517361	29.7810
	1	1.634582e+09	28.595312	83.469444	6.206250	9.980556	202.541667	1008.272569	29.7733
Gujarat	0	1.634576e+09	25.452431	77.813021	5.055729	8.134375	193.039931	1008.506944	29.7803
	1	1.634584e+09	31.181250	88.124132	5.624132	9.050174	187.750000	1008.572917	29.7827
Haryana	0	1.634578e+09	24.306891	75.751763	4.478365	7.205128	168.022436	1007.530449	29.7511
	1	1.634582e+09	30.022917	86.039962	5.950379	9.571780	217.160985	1007.787879	29.7591
Himachal Pradesh	0	1.634587e+09	19.466990	67.040777	4.703689	7.571456	117.928155	1010.693204	29.8446
	1	1.634588e+09	25.865169	78.558202	5.278427	8.497079	151.853933	1010.078652	29.8267
Jharkhand	0	1.634580e+09	23.576736	74.439757	4.916667	7.914410	188.151042	1007.262153	29.7440
	1	1.634580e+09	27.135417	80.844271	6.754861	10.872222	169.687500	1007.107639	29.7383

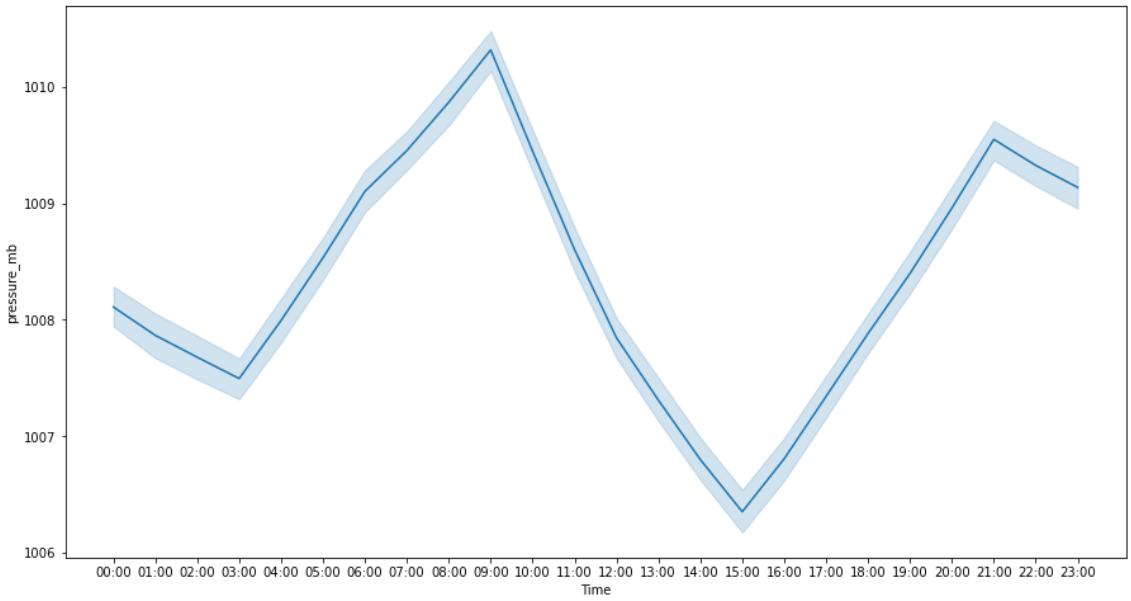
20 rows × 26 columns

```
plt.figure(figsize=(18,5))

sns.distplot(df['dewpoint_c'][df['chance_of_rain']==1],hist=False,color='red')

sns.distplot(df['dewpoint_c'][df['chance_of_rain']==0],hist=False,color='blue')

plt.title('Plot of Chances of Rain with Dewpoint  ', fontsize = 16)
plt.show()
```



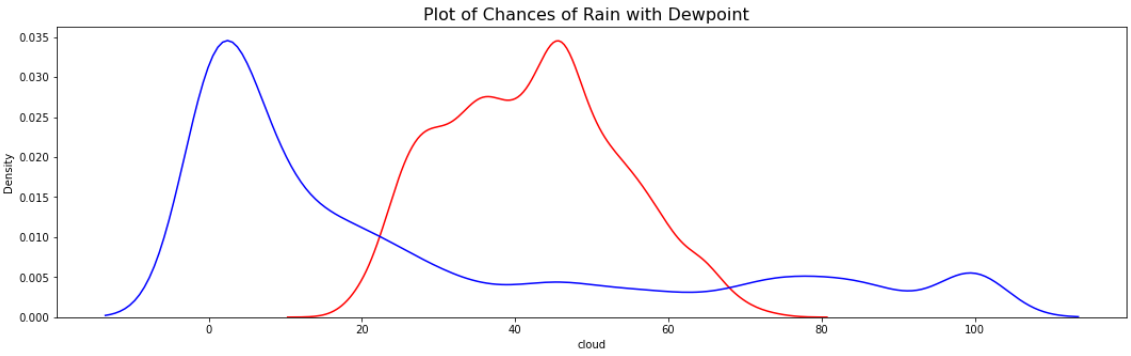
As dewpoint increases chances of rain also increases, dew point with range 18 to 25 has the highest chances of rain, while dewpoint with range 10 to 25 have low chances to rain

```
figure = plt.figure(figsize=(15,8))
sns.lineplot(y="pressure_mb",x="Time",data=df)
plt.show()
```



As time increases the pressure also increases, there is significient drop of pressure at 15:00 and a significient increase in pressure at 21:00

```
plt.figure(figsize=(18,5))

sns.distplot(df['cloud'][df['chance_of_rain']==1],hist=False,color='red')

sns.distplot(df['cloud'][df['chance_of_rain']==0],hist=False,color='blue')

plt.show()
```

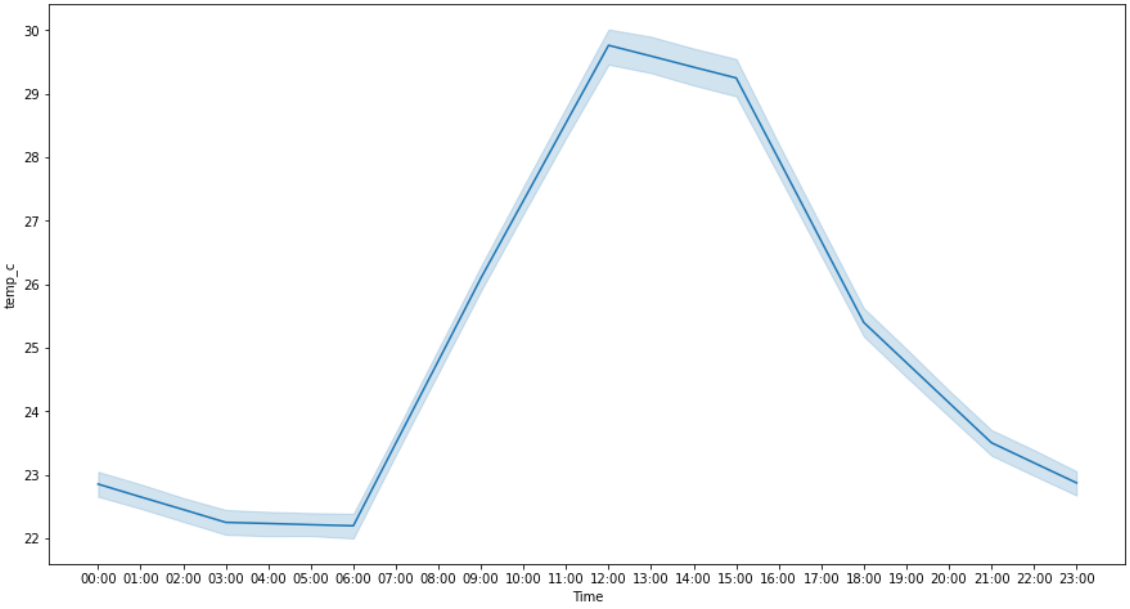


Increase in cloud has more chance of rain and there is less chance of rain when there less amount of cloud.

```
figure = plt.figure(figsize=(15,8))
sns.lineplot(x="temp_c",y="wind_kph",data=df)
plt.show()
```

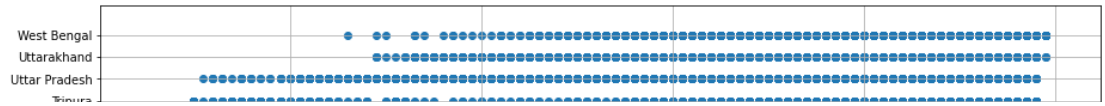
Wind increases between 2k/h to 14k/h when there is low temprature and there is slight increase in wind temprature is high.

```
figure = plt.figure(figsize=(15,8))
sns.lineplot(y="temp_c",x="Time",data=df)
plt.show()
```



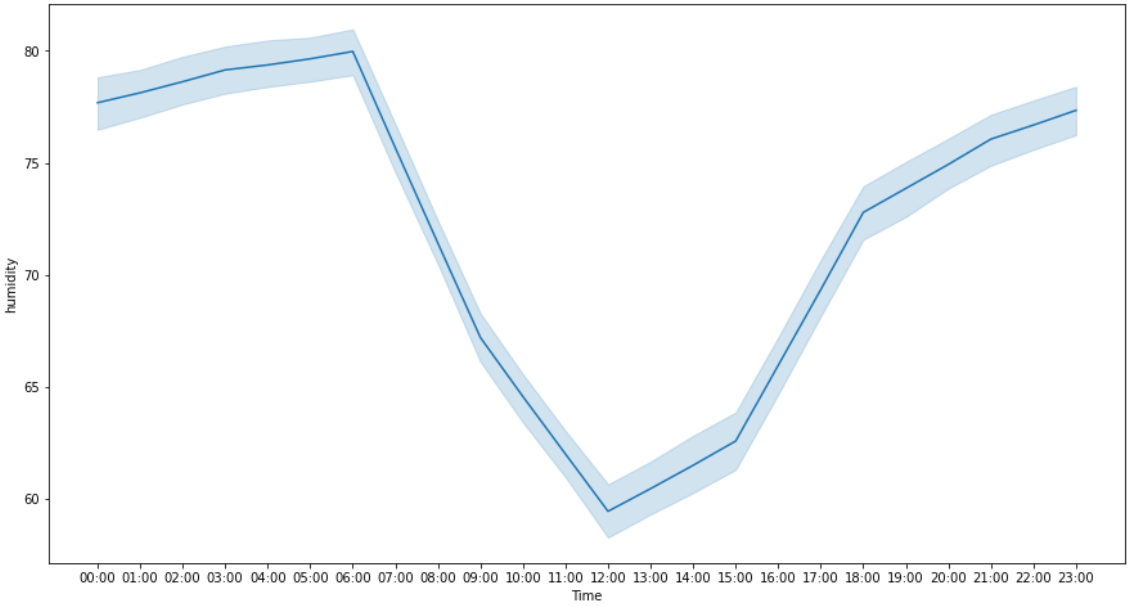
Temprature increases from 6:00 to 12:00 and decreases from 12:00 to 23:00

```
plt.figure(figsize=(15,10))
plt.scatter(data=df,y="state",x="humidity")
plt.xticks(rotation=90)
plt.grid(True)
plt.show()
```



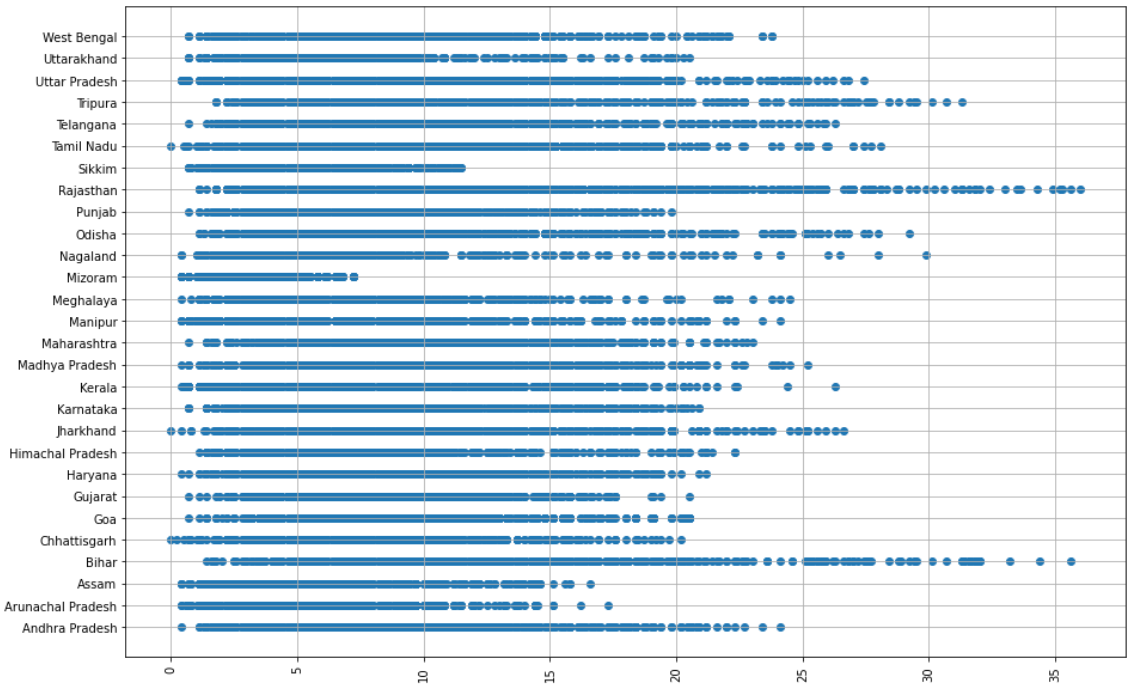
More humidity in Arunachal Pradesh, Assam, Karnataka, Meghalaya, Nizoram and Nagaland. Lowest humidity is found in Rajasthan state


```
figure = plt.figure(figsize=(15,8))
sns.lineplot(y="humidity",x="Time",data=df)
plt.show()
```



Humidity is higest at 6:00, low at 12:00 and gradually increases till 23:00

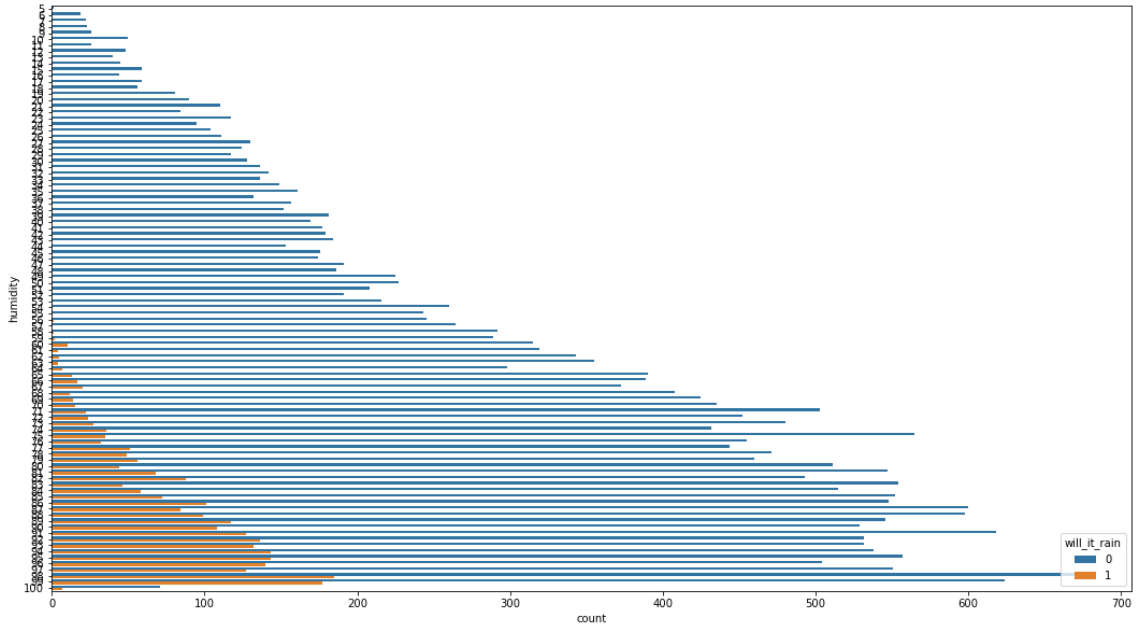
```
plt.figure(figsize=(15,10))
plt.scatter(data=df,x="wind_kph",y="state")
plt.xticks(rotation=90)
plt.grid(True)
plt.show()
```



Highest wind in k/h is found in Rajasthan. Mizoram has low wind in k/h

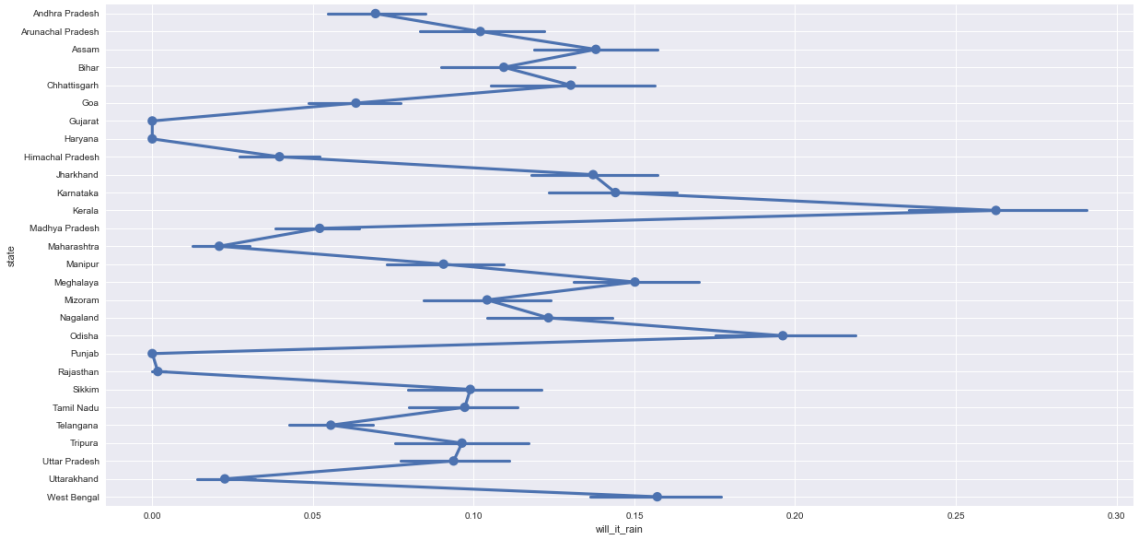
```
plt.figure(figsize=(18,10))
sns.countplot(data=df, y='humidity',hue='will_it_rain')
```

```
<AxesSubplot:xlabel='count', ylabel='humidity'>
```



With increase in humidity chances of rain also increases.

```
plt.figure(figsize=(20,10))
sns.pointplot(df["will_it_rain"], df["state"])
plt.grid(True)
plt.show()
```



Precection of raining is high for state Kerala and Odisha. Predictions for will it rain is low for state Gujarat, Haryana and punjab.