```
# https://pypi.python.org/pypi/libarchive
!apt-get -qq install -y libarchive-dev && pip install -U libarchive
import libarchive
```

```
Selecting previously unselected package libarchive-dev:amd64.
(Reading database ... 155113 files and directories currently installed.)
Preparing to unpack .../libarchive-dev_3.2.2-3.1ubuntu0.7_amd64.deb ...
Unpacking libarchive-dev:amd64 (3.2.2-3.1ubuntu0.7) ...
Setting up libarchive-dev:amd64 (3.2.2-3.1ubuntu0.7) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Collecting libarchive
  Downloading libarchive-0.4.7.tar.gz (23 kB)
Collecting nose
  Downloading nose-1.3.7-py3-none-any.whl (154 kB)
     |████████████████████████████████| 154 kB 10.5 MB/s
Building wheels for collected packages: libarchive
  Building wheel for libarchive (setup.py) ... done
  Created wheel for libarchive: filename=libarchive-0.4.7-py3-none-any.whl size=31646 sha256=9474
  Stored in directory: /root/.cache/pip/wheels/63/b1/c6/b3da79bec2012175bd43603eed98ef8548ac1733b
Successfully built libarchive
Installing collected packages: nose, libarchive
Successfully installed libarchive-0.4.7 nose-1.3.7
```

```
!pip install matplotlib-venn
```

```
Requirement already satisfied: matplotlib-venn in /usr/local/lib/python3.7/dist-packages (0.11.6)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from matplotlib-ve
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from matplotl
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from matplotlib-ve
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from k
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplo
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dat
```

```
!apt-get -qq install -y libfluidsynth1
```

```
Selecting previously unselected package libfluidsynth1:amd64.
(Reading database ... 155169 files and directories currently installed.)
Preparing to unpack .../libfluidsynth1_1.1.9-1_amd64.deb ...
Unpacking libfluidsynth1:amd64 (1.1.9-1) ...
Setting up libfluidsynth1:amd64 (1.1.9-1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.3) ...
/sbin/ldconfig.real: /usr/local/lib/python3.7/dist-packages/ideep4py/lib/libmkldnn.so.0 is not a s
```

```
# https://pypi.python.org/pypi/pydot
!apt-get -qq install -y graphviz && pip install pydot
import pydot
```

```
Requirement already satisfied: pydot in /usr/local/lib/python3.7/dist-packages (1.3.0)
Requirement already satisfied: pyparsing>=2.1.4 in /usr/local/lib/python3.7/dist-packages (from py
```

```
!pip install cartopy
import cartopy
```

```
Collecting cartopy
  Downloading Cartopy-0.20.2.tar.gz (10.8 MB)
     |████████████████████████████████| 10.8 MB 9.8 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... error
WARNING: Discarding https://files.pythonhosted.org/packages/f6/55/1e1c737dc9436b320deead73d1c455d
  Downloading Cartopy-0.20.1.tar.gz (10.8 MB)
     |████████████████████████████████| 10.8 MB 40.5 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... error
WARNING: Discarding https://files.pythonhosted.org/packages/fc/59/aa52698e3838f4cd0e7eaa75bd86837
  Downloading Cartopy-0.20.0.tar.gz (10.8 MB)
     |████████████████████████████████| 10.8 MB 53.8 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... error
WARNING: Discarding https://files.pythonhosted.org/packages/0f/c0/58453b036e79046d211f083880d58dc
  Downloading Cartopy-0.19.0.post1.tar.gz (12.1 MB)
     |████████████████████████████████| 12.1 MB 49.2 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
    Preparing wheel metadata ... done
Collecting pyshp>=2
  Downloading pyshp-2.2.0-py3-none-any.whl (44 kB)
     |████████████████████████████████| 44 kB 3.3 MB/s
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.7/dist-packages (from carto
Requirement already satisfied: shapely>=1.5.6 in /usr/local/lib/python3.7/dist-packages (from car
Building wheels for collected packages: cartopy
  Building wheel for cartopy (PEP 517) ... done
  Created wheel for cartopy: filename=Cartopy-0.19.0.post1-cp37-cp37m-linux_x86_64.whl size=12516
  Stored in directory: /root/.cache/pip/wheels/98/01/f7/bd10aeb96fe4b518cde5f7c4f5e12c7202f85b735
Successfully built cartopy
Installing collected packages: pyshp, cartopy
Successfully installed cartopy-0.19.0.post1 pyshp-2.2.0
```

```
#Data manipulation libraries :
import numpy as np  #numpy
import pandas as pd  #pandas
import tensorflow as tf
#System libraries
import glob #The glob module finds all the pathnames matching a specified pattern according to the rule

#Map plotting
import folium #Interactive Maps viz

#Plotting
import seaborn as sns
```

```
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

#math operations lib
import math
from math import pi

#date manipulation
import datetime as dt

#Missing data detector lib
import missingno as mn

#Impute missing data
import sklearn.impute
sklearn.impute.SimpleImputer
#from sklearn.preprocessing import Imputer

#Deep learning with keras
from keras import backend as K
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout

#Splitting data to test and train
from sklearn.model_selection import train_test_split

import datetime
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from tensorflow import keras
%matplotlib inline
plt.style.use('seaborn-whitegrid')

from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
from collections import Counter
import time
import datetime as dt
from datetime import datetime
import collections
import os # accessing directory structure
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
from matplotlib.pyplot import rcParams

from sklearn import linear_model
```

```
data = pd.read_csv('/content/weatherHistory.csv')
data.head()
```

	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km
0	2006-04-01 00:00:00.000 +0200	Partly Cloudy	rain	9.472222	7.388889	0.89	14.1197	251	15.826
1	2006-04-01 01:00:00.000 +0200	Partly Cloudy	rain	9.355556	7.227778	0.86	14.2646	259	15.826
2	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	rain	9.377778	9.377778	0.89	3.9284	204	14.956
3	2006-04-01 03:00:00.000 +0200	Partly Cloudy	rain	8.288889	5.944444	0.83	14.1036	269	15.826

```
data.dtypes
```

```
Formatted Date               object
Summary                      object
Precip Type                  object
Temperature (C)             float64
Apparent Temperature (C)    float64
Humidity                    float64
Wind Speed (km/h)           float64
Wind Bearing (degrees)        int64
Visibility (km)             float64
Loud Cover                    int64
Pressure (millibars)        float64
Daily Summary                object
dtype: object
```

```
#Categorical variables:
categorical = data.select_dtypes(include = ["object"]).keys()
print(categorical)
```

```
Index(['Formatted Date', 'Summary', 'Precip Type', 'Daily Summary'], dtype='object')
```

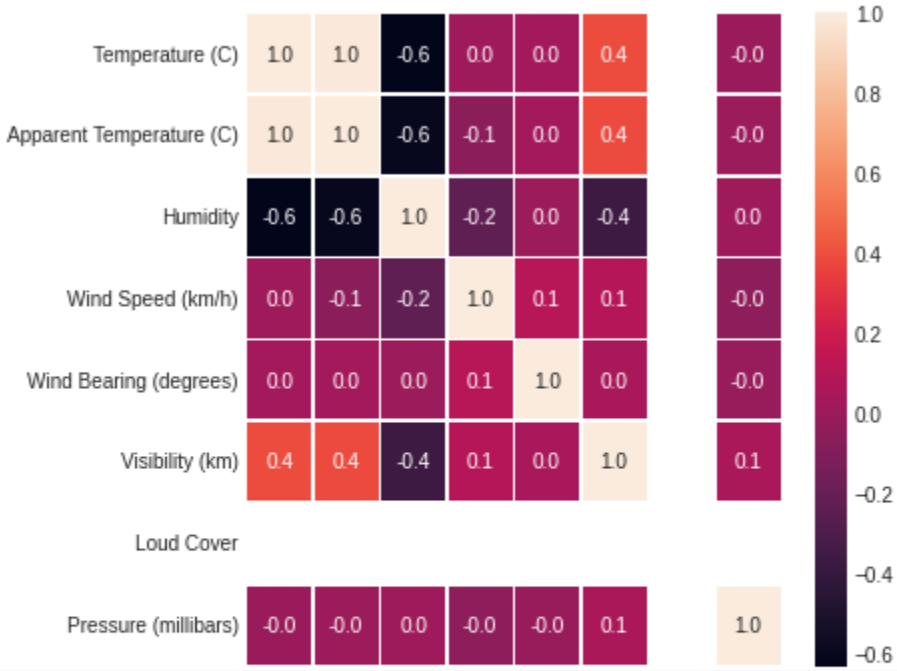
```
#Quantitative variables:
quantitative = data.select_dtypes(include = ["int64","float64"]).keys()
print(quantitative)
```

```
Index(['Temperature (C)', 'Apparent Temperature (C)', 'Humidity',
       'Wind Speed (km/h)', 'Wind Bearing (degrees)', 'Visibility (km)',
       'Loud Cover', 'Pressure (millibars)'],
      dtype='object')
```

```
#'Formatted Date' transformation:

data['Date'] = pd.to_datetime(data['Formatted Date'])
data['year'] = pd.to_datetime(data['Formatted Date'])
data['month'] = pd.to_datetime(data['Formatted Date'])
data['day'] = pd.to_datetime(data['Formatted Date'])
data['hour'] = pd.to_datetime(data['Formatted Date'])
'''data['year'] = data['Date'].dt.year
data['month'] = data['Date'].dt.month
data['day'] = data['Date'].dt.day
data['hour'] = data['Date'].dt.hour'''
data.head()
```

	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km
0	2006-04-01 00:00:00.000 +0200	Partly Cloudy	rain	9.472222	7.388889	0.89	14.1197	251	15.826
1	2006-04-01 01:00:00.000 +0200	Partly Cloudy	rain	9.355556	7.227778	0.86	14.2646	259	15.826
2	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	rain	9.377778	9.377778	0.89	3.9284	204	14.956
3	2006-04-01 03:00:00.000 +0200	Partly Cloudy	rain	8.288889	5.944444	0.83	14.1036	269	15.826
4	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	rain	8.755556	6.977778	0.83	11.0446	259	15.826

```
f,ax = plt.subplots(figsize=(6, 6))
sns.heatmap(data.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
plt.show()
```

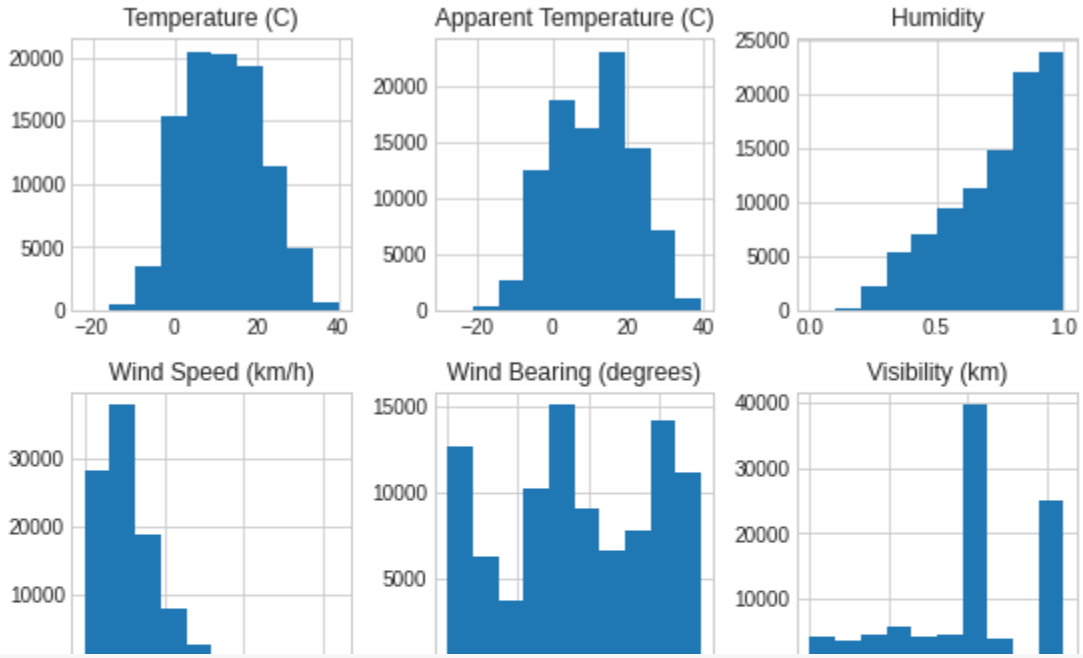



```
#quantitative variables, missing
data[quantitative].describe()
```

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover
count	96453.000000	96453.000000	96453.000000	96453.000000	96453.000000	96453.000000	96453.0
mean	11.932678	10.855029	0.734899	10.810640	187.509232	10.347325	0.0
std	9.551546	10.696847	0.195473	6.913571	107.383428	4.192123	0.0
min	-21.822222	-27.716667	0.000000	0.000000	0.000000	0.000000	0.0
25%	4.688889	2.311111	0.600000	5.828200	116.000000	8.339800	0.0
50%	12.000000	12.000000	0.780000	9.965900	180.000000	10.046400	0.0
75%	18.838889	18.838889	0.890000	14.135800	290.000000	14.812000	0.0

```
rcParams['figure.figsize'] = 9, 9
data[quantitative].hist()
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f082fb68090>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f082f651690>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f082f608c90>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f082f5cc2d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f082f57f8d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f082f537ed0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f082f4fa590>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f082f4acad0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f082f4acb10>]],
      dtype=object)
```



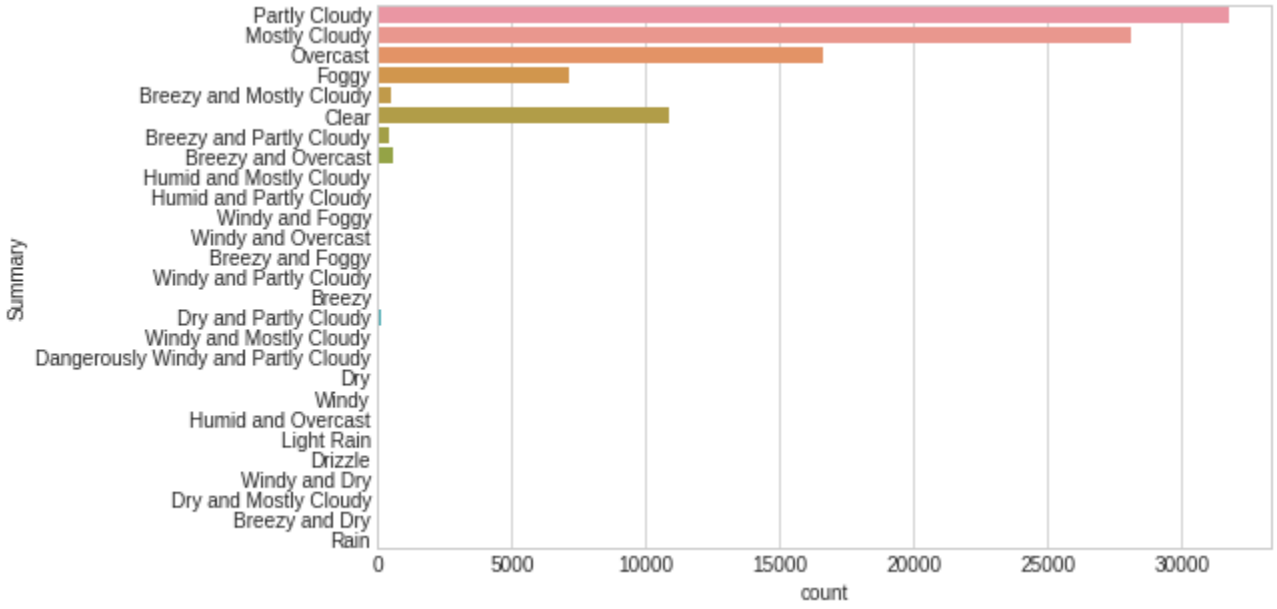

```
data=data.drop('Loud Cover',axis=1)
```

```
pressure_median = data['Pressure (millibars)'].median()
def pressure(x):
    if x==0:
        return x + pressure_median
    else:
        return x

data["Pressure (millibars)"] = data.apply(lambda row:pressure(row["Pressure (millibars)"]) , axis = 1)

rcParams['figure.figsize'] = 5, 3
data['Pressure (millibars)'].hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f082f293510>
```

```
rcParams['figure.figsize'] = 8, 5
sns.countplot(y=data['Summary'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f082f293590>
```




```
data['Daily Summary'].value_counts(dropna=False)
```

```
Mostly cloudy throughout the day.
20085
Partly cloudy throughout the day.
9981
Partly cloudy until night.
6169
Partly cloudy starting in the morning.
5184
Foggy in the morning.
4201

...
Breezy starting overnight continuing until morning and foggy overnight.
24
Mostly cloudy throughout the day and breezy starting overnight continuing until afternoon.
24
Partly cloudy starting in the morning and breezy starting in the afternoon continuing until
evening.       24
Rain until afternoon.
24
Foggy starting overnight continuing until morning and breezy in the afternoon.
23
Name: Daily Summary, Length: 214, dtype: int64
```

```
data['Summary'].value_counts(dropna=False)
```

```
Partly Cloudy                         31733
Mostly Cloudy                         28094
Overcast                              16597
```

```
Clear                                10890
Foggy                                 7148
Breezy and Overcast                    528
Breezy and Mostly Cloudy               516
Breezy and Partly Cloudy               386
Dry and Partly Cloudy                   86
Windy and Partly Cloudy                 67
Light Rain                              63
Breezy                                  54
Windy and Overcast                      45
Humid and Mostly Cloudy                 40
Drizzle                                 39
Breezy and Foggy                        35
Windy and Mostly Cloudy                 35
Dry                                     34
Humid and Partly Cloudy                 17
Dry and Mostly Cloudy                   14
Rain                                    10
Windy                                    8
Humid and Overcast                       7
Windy and Foggy                          4
Windy and Dry                            1
Dangerously Windy and Partly Cloudy      1
Breezy and Dry                           1
Name: Summary, dtype: int64
```

```
data['Precip Type'].value_counts(dropna=False)
```

```
rain    85224
snow    10712
NaN       517
Name: Precip Type, dtype: int64
```

```
data.fillna(method='ffill', inplace=True)
```

```
sns.countplot(x=data['Precip Type'])
```

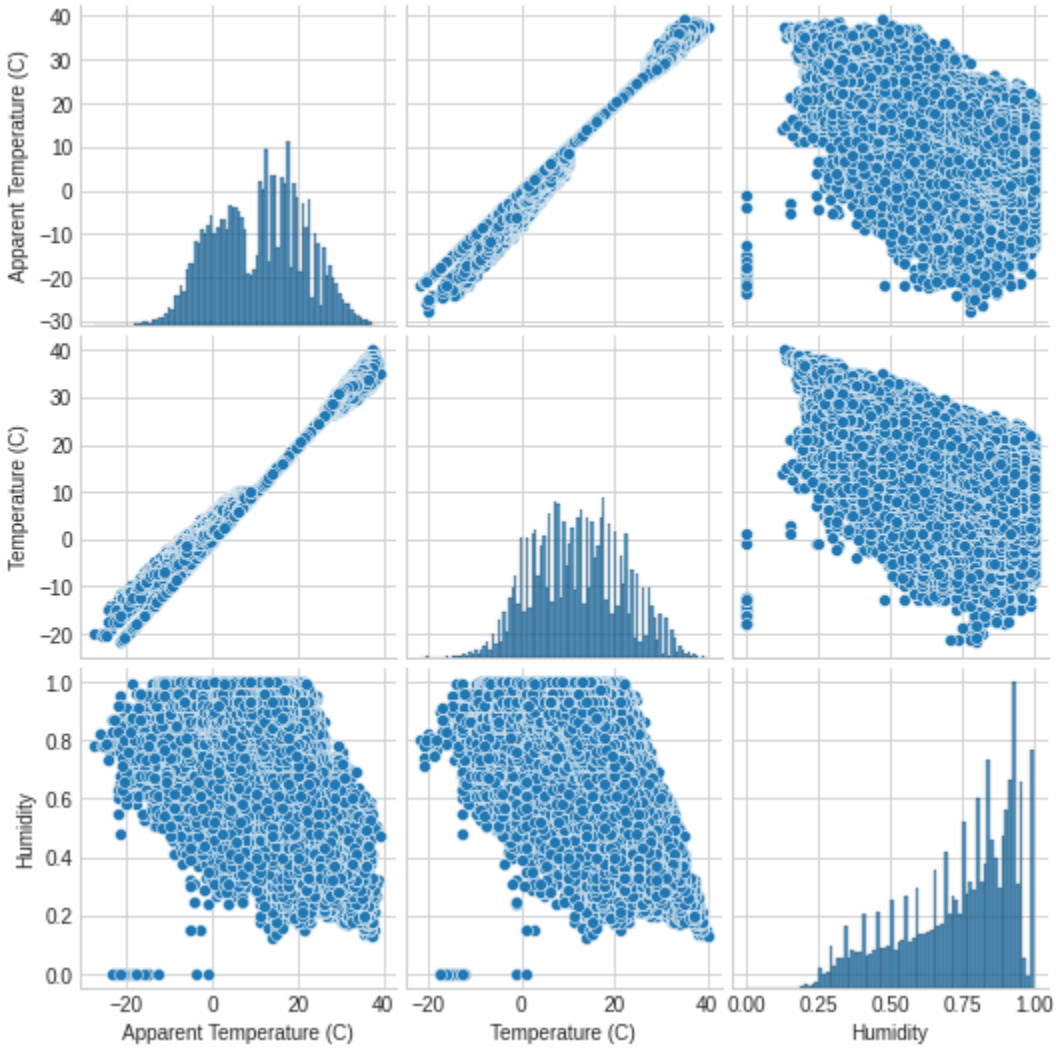
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f082e0725d0>
```

```
# Calculate total number of cells in dataframe
totalCells = np.product(data.shape)
# Count number of missing values per column
missingCount = data.isnull().sum()
# Calculate total number of missing values
totalMissing = missingCount.sum()
# Calculate percentage of missing values
print("The weather history dataset contains", round(((totalMissing/totalCells) * 100), 2), "%", "missin
```

```
The weather history dataset contains 0.0 % missing values.
```

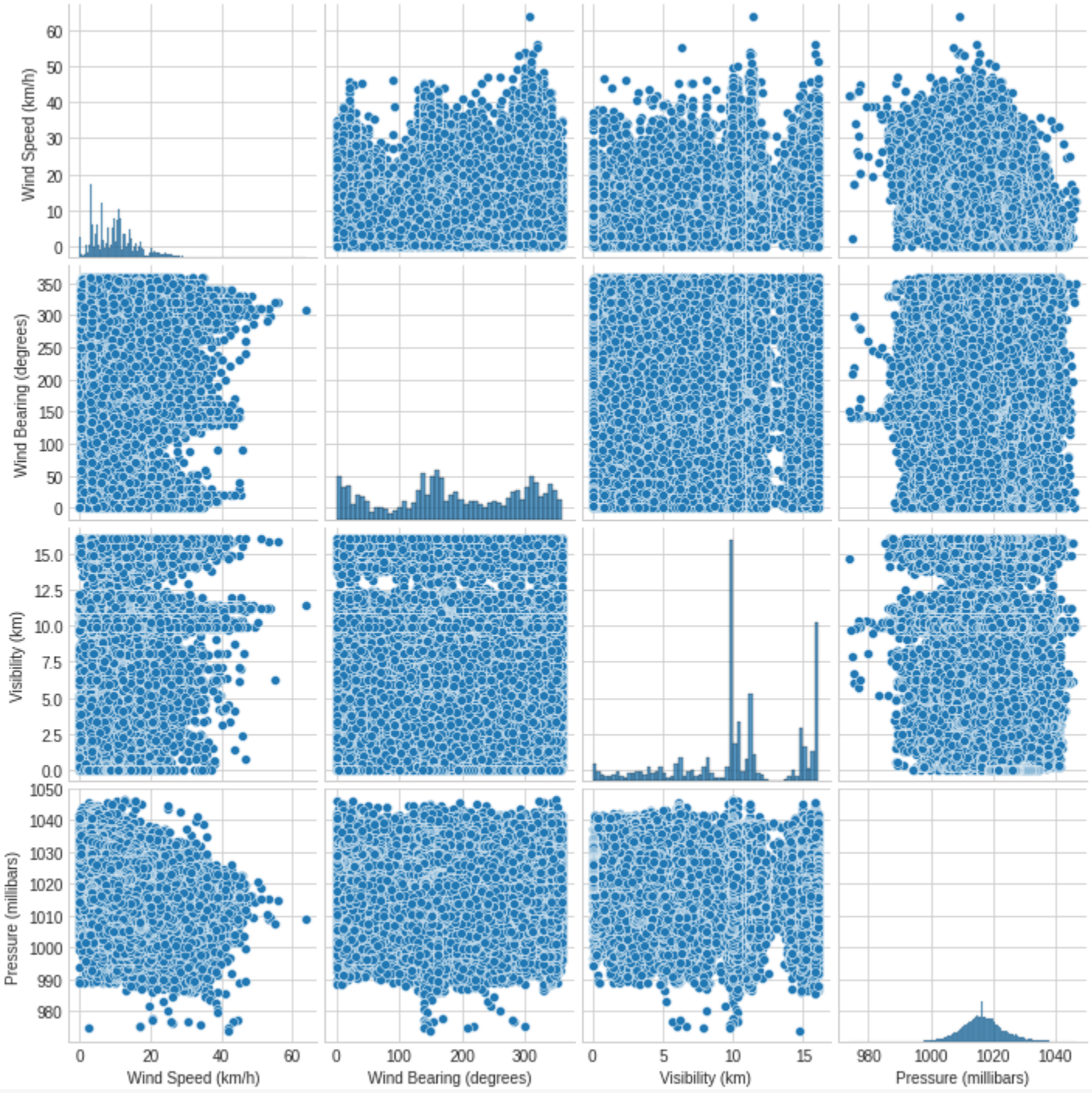
```
plt.figure(figsize=(18,8)) # this creates a figure 8 inch wide, 4 inch high
sns.pairplot(data[['Apparent Temperature (C)','Temperature (C)', 'Humidity' ]])
plt.show()
```

```
<Figure size 1296x576 with 0 Axes>
```

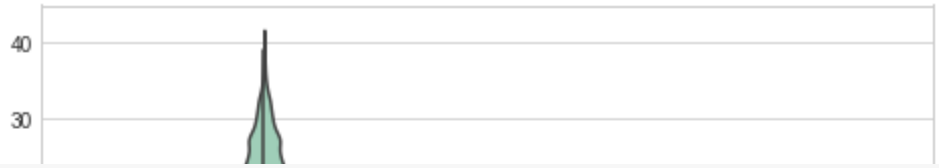



```
plt.figure(figsize=(18,8)) # this creates a figure 8 inch wide, 4 inch high
sns.pairplot(data[['Wind Speed (km/h)', 'Wind Bearing (degrees)', 'Visibility (km)', 'Pressure (milliba
plt.show()
```

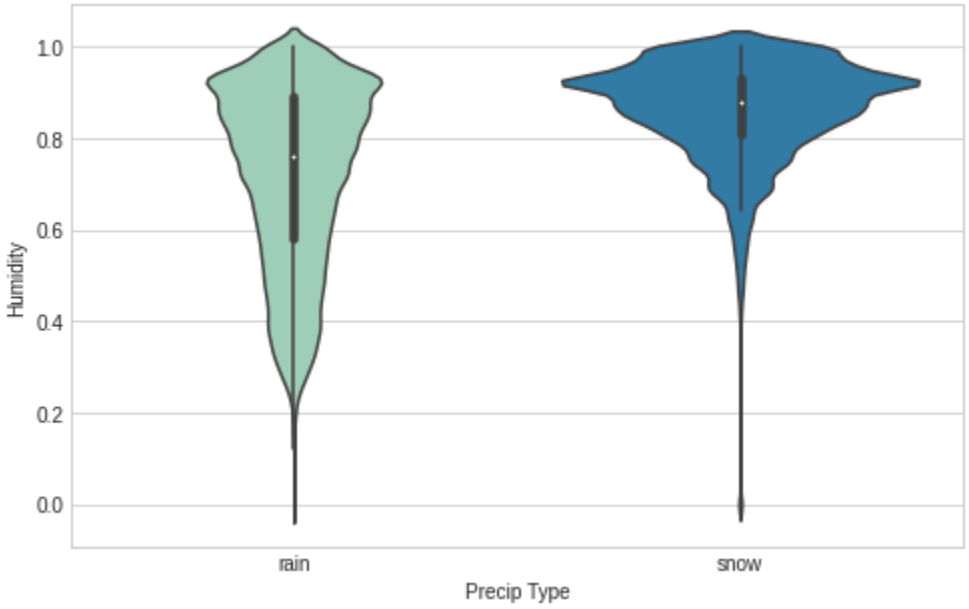
<Figure size 1296x576 with 0 Axes>




```
sns.violinplot(x="Precip Type", y="Temperature (C)", data=data, palette="YlGnBu");
```




```
sns.violinplot(x="Precip Type", y="Humidity", data=data, palette="YlGnBu");
```




```
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

ls = linear_model.LinearRegression()



X = data["Humidity"].values.reshape(-1,1)
y = data["Temperature (C)"].values.reshape(-1,1)

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.33,
                                                    shuffle=True, random_state=0)
print("Linear Regression")
ls.fit(X_train, y_train)
print("alpha = ",ls.coef_[0])
print("beta = ",ls.intercept_)
print("\n\nCalculating some regression quality metrics")
y_pred = ls.predict(X_test)
print("MSE = ",mean_squared_error(y_test, y_pred))
print("R2 = ",r2_score(y_test, y_pred))
```

```
Linear Regression
alpha =  [-30.86657506]
beta =  [34.61438171]
```

```
 Calculating some regression quality metrics
 MSE =  54.71624111922442
 R2 =  0.4004047204664315
```

```
def change_category_to_number(DailySummaryCat):
    if DailySummaryCat=='Partly cloudy throughout the day.':
        return 1
    elif DailySummaryCat=='Mostly cloudy throughout the day.':
        return 2
    elif DailySummaryCat=='Foggy in the evening.':
        return 3
    elif DailySummaryCat=='Foggy overnight and breezy in the morning.':
        return 4
    elif DailySummaryCat=='Overcast throughout the day.':
        return 5
    elif DailySummaryCat=='Partly cloudy until night.':
        return 6
    elif DailySummaryCat=='Motly cloudy until night.':
        return 7
    elif DailySummaryCat=='Foggy starting overnight continuing until morning.':
        return 8
    elif DailySummaryCat=='Foggy in the morning.':
        return 9
    elif DailySummaryCat=='Partly cloudy until evening.':
        return 10
    elif DailySummaryCat=='Partly cloudy starting in the morning.':
        return 11
    elif DailySummaryCat=='Mostly cloudy starting overnight continuing until night.':
        return 12
    elif DailySummaryCat=='Partly cloudy starting in the afternoon.':
        return 13
    elif DailySummaryCat=='Partly cloudy starting overnight.':
        return 14
    elif DailySummaryCat=='Mostly cloudy starting overnight.':
        return 15
    elif DailySummaryCat=='Mostly cloudy until night and breezy in the afternoon.':
        return 16
    elif DailySummaryCat=='Mostly cloudy until evening.':
        return 17
    elif DailySummaryCat=='Foggy throughout the day.':
        return 18
    elif DailySummaryCat=='Partly cloudy starting in the morning.':
        return 19
    elif DailySummaryCat=='Partly cloudy starting in the morning continuing until evening.':
        return 20
    elif DailySummaryCat=='Foggy until morning.':
        return 21
    elif DailySummaryCat=='Partly cloudy starting in the morning continuing until night.':
        return 22
    elif DailySummaryCat=='Mostly cloudy starting in the morning.':
        return 23
    elif DailySummaryCat=='Foggy starting in the evening.':
```

```
    return 24
elif DailySummaryCat=='Partly cloudy starting in the afternoon continuing until evening.':
    return 25
elif DailySummaryCat=='Foggy overnight.':
    return 26
elif DailySummaryCat=='Clear throughout the day.':
    return 27
elif DailySummaryCat=='Partly cloudy starting overnight continuing until night.':
    return 28
elif DailySummaryCat=='Partly cloudy overnight.':
    return 29
elif DailySummaryCat=='Partly cloudy starting overnight continuing until evening.':
    return 30
elif DailySummaryCat=='Foggy until night.':
    return 31
elif DailySummaryCat=='Partly cloudy in the morning.':
    return 32
elif DailySummaryCat=='Foggy starting overnight continuing until afternoon.':
    return 33
elif DailySummaryCat=='Foggy until afternoon.':
    return 34
elif DailySummaryCat=='Breezy and mostly cloudy overnight.':
    return 35
elif DailySummaryCat=='Partly cloudy overnight and breezy starting in the morning continuing until
    return 36
elif DailySummaryCat=='Breezy in the morning and foggy in the evening.':
    return 37
elif DailySummaryCat=='Mostly cloudy until evening and breezy in the evening.':
    return 38
elif DailySummaryCat=='Mostly cloudy starting in the evening.':
    return 39
elif DailySummaryCat=='Mostly cloudy throughout the day and breezy starting overnight continuing un
    return 40
elif DailySummaryCat=='Breezy starting in the morning continuing until night.':
    return 41
elif DailySummaryCat=='Overcast throughout the day and breezy starting overnight continuing until m
    return 42
elif DailySummaryCat=='Breezy starting overnight continuing until morning and foggy in the evening.
    return 43
elif DailySummaryCat=='Light rain until morning.':
    return 44
elif DailySummaryCat=='Mostly cloudy until night and breezy starting in the afternoon continuing un
    return 45
elif DailySummaryCat=='Mostly cloudy starting in the morning continuing until afternoon.':
    return 46
elif DailySummaryCat=='Breezy until afternoon and overcast throughout the day.':
    return 47
elif DailySummaryCat=='Partly cloudy until evening and breezy in the afternoon.':
    return 48
elif DailySummaryCat=='Breezy starting overnight continuing until morning and partly cloudy startin
    return 49
elif DailySummaryCat=='Light rain starting overnight.':
    return 50
```

```
    elif DailySummaryCat=='Partly cloudy starting overnight continuing until evening and breezy startin
        return 51
    elif DailySummaryCat=='Foggy starting in the morning continuing until evening and breezy in the eve
        return 52
    elif DailySummaryCat=='Partly cloudy throughout the day and breezy in the afternoon.':
        return 53
    elif DailySummaryCat=='Mostly cloudy starting overnight continuing until evening and breezy startin
        return 54
    elif DailySummaryCat=='Partly cloudy starting overnight continuing until evening and breezy in the
        return 55
    elif DailySummaryCat=='Overcast throughout the day and breezy overnight.':
        return 56
    elif DailySummaryCat=='Light rain in the morning.':
        return 57
    elif DailySummaryCat=='Rain until morning.':
        return 58
    elif DailySummaryCat=='Breezy in the morning and mostly cloudy starting in the evening.':
        return 59
    elif DailySummaryCat=='Mostly cloudy starting in the morning and breezy overnight.':
        return 60
    elif DailySummaryCat=='Partly cloudy starting overnight and breezy starting in the morning continui
        return 61
    elif DailySummaryCat=='Partly cloudy starting in the morning and breezy starting in the afternoon c
        return 62
    elif DailySummaryCat=='Partly cloudy starting in the morning continuing until evening and breezy in
        return 63
    elif DailySummaryCat=='Foggy starting overnight continuing until morning and breezy in the afternoo
        return 64
```

```
data['DailySummaryCat'] = data['Daily Summary'].apply(change_category_to_number)
```

```
data.fillna(method='ffill', inplace=True)
```

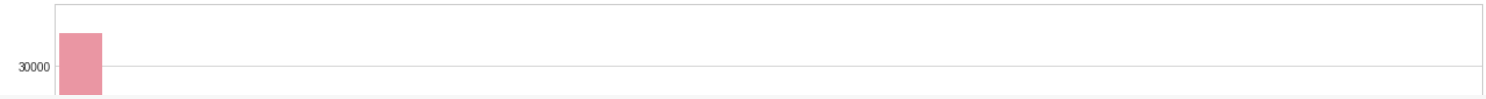
```
plt.figure(figsize=(18,8)) # this creates a figure 8 inch wide, 4 inch high
ax = sns.countplot(x=data['DailySummaryCat'])
ax.set_xticklabels(ax.get_xticklabels(), rotation=90, ha="right")
plt.tight_layout()
plt.show()
```


```
def change_category(Summary):
    if Summary=='Partly Cloudy':
        return 1
    elif Summary=='Mostly Cloudy':
        return 2
    elif Summary=='Foggy':
        return 3
    elif Summary=='Clear':
        return 4
    elif Summary=='Overcast':
        return 5
    elif Summary=='Breezy and Overcast':
        return 6
    elif Summary=='Breezy and Partly Cloudy':
        return 7
    elif Summary=='Breezy and Mostly Cloudy':
        return 8
    elif Summary=='Dry and Partly Cloudy':
        return 9
    elif Summary=='Windy and Partly Cloudy':
        return 10
    elif Summary=='Light Rain':
        return 11
    elif Summary=='Breezy':
        return 12
    elif Summary=='Windy and Overcast':
        return 13
    elif Summary=='Humid and Mostly Cloudy':
        return 14
    elif Summary=='Drizzle':
        return 15
    elif Summary=='Windy and Mostly Cloudy':
        return 16
    elif Summary=='Breezy and Foggy':
        return 17
    elif Summary=='Dry':
        return 18
```

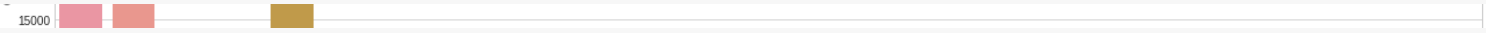
```
    elif Summary=='Humid and Partly Cloudy':
        return 19
    elif Summary=='Dry and Mostly Cloudy':
        return 20
    elif Summary=='Rain':
        return 21
    elif Summary=='Windy':
        return 22
    elif Summary=='Humid and Overcast':
        return 23
    elif Summary=='Windy and Foggy':
        return 24
    elif Summary=='Dangerously Windy and Partly Cloudy':
        return 25
    elif Summary=='Windy and Dry':
        return 26
    elif Summary=='Breezy and Dry':
        return 27
```

```
data['SummaryCat'] = data['Summary'].apply(change_category)
```

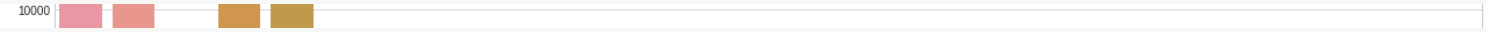
```
plt.figure(figsize=(18,8)) # this creates a figure 8 inch wide, 4 inch high
ax = sns.countplot(x=data['SummaryCat'])
ax.set_xticklabels(ax.get_xticklabels(), rotation=90, ha="right")
plt.tight_layout()
plt.show()
```



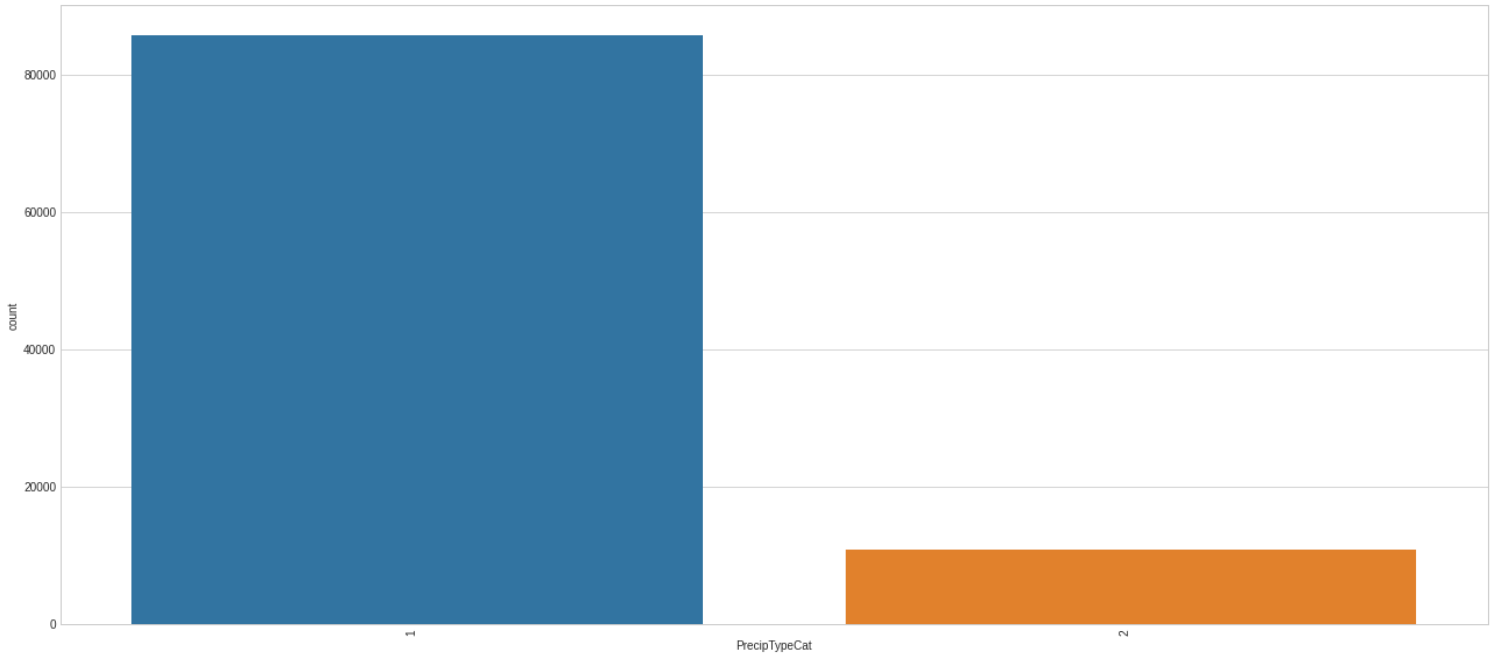

```
def change_category(PrecipTypeCat):
    if PrecipTypeCat=='rain':
        return 1
    elif PrecipTypeCat=='snow':
        return 2
```




```
data['PrecipTypeCat'] = data['Precip Type'].apply(change_category)
```




```
plt.figure(figsize=(18,8)) # this creates a figure 8 inch wide, 4 inch high
ax = sns.countplot(x=data['PrecipTypeCat'])
ax.set_xticklabels(ax.get_xticklabels(), rotation=90, ha="right")
plt.tight_layout()
plt.show()
```




```
data.drop(['Formatted Date','Summary','Daily Summary','Precip Type','Date'],axis=1,inplace=True)
data.head()
```

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Pressure (millibars)	year
0	9.472222	7.388889	0.89	14.1197	251	15.8263	1015.13	2006-04-01 00:00:00+02:00
1	9.355556	7.227778	0.86	14.2646	259	15.8263	1015.63	2006-04-01 01:00:00+02:00
2	9.377778	9.377778	0.89	3.9284	204	14.9569	1015.94	2006-04-01 02:00:00+02:00
3	8.288889	5.944444	0.83	14.1036	269	15.8263	1016.41	2006-04-01 03:00:00+02:00
4	8.755556	6.977778	0.83	11.0446	259	15.8263	1016.51	2006-04-01

```
X = data.drop('DailySummaryCat', axis=1)
y = data['DailySummaryCat']
```

```
print(X)
```

```
       Temperature (C)  Apparent Temperature (C)  ...  SummaryCat  PrecipTypeCat
0             9.472222                  7.388889  ...           1              1
1             9.355556                  7.227778  ...           1              1
2             9.377778                  9.377778  ...           2              1
3             8.288889                  5.944444  ...           1              1
4             8.755556                  6.977778  ...           2              1
...                ...                       ...  ...         ...            ...
96448        26.016667                 26.016667  ...           1              1
96449        24.583333                 24.583333  ...           1              1
96450        22.038889                 22.038889  ...           1              1
96451        21.522222                 21.522222  ...           1              1
96452        20.438889                 20.438889  ...           1              1

[96453 rows x 13 columns]
```

```
print(y)
```

```
0         1.0
1         1.0
2         1.0
3         1.0
4         1.0
         ...
96448    11.0
96449    11.0
96450    11.0
96451    11.0
96452    11.0
Name: DailySummaryCat, Length: 96453, dtype: float64
```