```
!pip install -r rec.txt
```

```
Collecting numpy==1.19.5 (from -r rec.txt (line 1))
  Using cached numpy-1.19.5.zip (7.3 MB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from -r rec.txt (line 2)) (2.2.
Collecting matplotlib==3.3.4 (from -r rec.txt (line 3))
  Using cached matplotlib-3.3.4.tar.gz (37.9 MB)
  Preparing metadata (setup.py) ... done
Collecting seaborn==0.11.1 (from -r rec.txt (line 4))
  Using cached seaborn-0.11.1-py3-none-any.whl.metadata (2.3 kB)
Collecting scikit-learn==0.24.1 (from -r rec.txt (line 5))
  Using cached scikit-learn-0.24.1.tar.gz (7.4 MB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting scipy==1.5.4 (from -r rec.txt (line 6))
  Using cached scipy-1.5.4.tar.gz (25.2 MB)
  error: subprocess-exited-with-error

  × pip subprocess to install build dependencies did not run successfully.
  │ exit code: 1
  ╰─> See above for output.

  note: This error originates from a subprocess, and is likely not a problem with pip.
  Installing build dependencies ... error
error: subprocess-exited-with-error

× pip subprocess to install build dependencies did not run successfully.
│ exit code: 1
╰─> See above for output.

note: This error originates from a subprocess, and is likely not a problem with pip.
```

```
!pip install catboost
```

```
Collecting catboost
  Downloading catboost-1.2.7-cp310-cp310-manylinux2014_x86_64.whl.metadata (1.2 kB)
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (from catboost) (0.20.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from catboost) (3.7.1)
Requirement already satisfied: numpy<2.0,>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from catboost) (1.2
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.10/dist-packages (from catboost) (2.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from catboost) (1.13.1)
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (from catboost) (5.24.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from catboost) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboo
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catb
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catb
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->cat
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->cat
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catbo
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboos
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catb
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly->catboost)
Downloading catboost-1.2.7-cp310-cp310-manylinux2014_x86_64.whl (98.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.7/98.7 MB 7.4 MB/s eta 0:00:00
Installing collected packages: catboost
Successfully installed catboost-1.2.7
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
import scipy.stats as stats
from sklearn.model_selection import train_test_split
from collections import Counter
from imblearn.over_sampling import SMOTE
```

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier
from catboost import CatBoostClassifier
from xgboost import XGBClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
import joblib
```

```
df = pd.read_csv("processed_weather_data.csv")
pd.set_option("display.max_columns", None)
df
```

	Date	Location	Temp	WindSpeed	Humidity	RainToday	RainTomorrow
0	2008-12-01	Albury	18.15	22.0	46.5	No	No
1	2008-12-02	Albury	16.25	13.0	34.5	No	No
2	2008-12-03	Albury	19.30	22.5	34.0	No	No
3	2008-12-04	Albury	18.60	10.0	30.5	No	No
4	2008-12-05	Albury	24.90	13.5	57.5	No	No
...	...	...	...	...	...	...	...
145455	2017-06-21	Uluru	13.10	12.0	37.5	No	No
145456	2017-06-22	Uluru	14.45	11.0	38.5	No	No
145457	2017-06-23	Uluru	16.15	9.0	38.5	No	No
145458	2017-06-24	Uluru	17.40	10.0	37.5	No	No
145459	2017-06-25	Uluru	14.90	17.0	49.0	No	NaN

145460 rows × 7 columns

```
numerical_feature = [feature for feature in df.columns if df[feature].dtypes != 'O']
discrete_feature=[feature for feature in numerical_feature if len(df[feature].unique())<25]
continuous_feature = [feature for feature in numerical_feature if feature not in discrete_feature]
categorical_feature = [feature for feature in df.columns if feature not in numerical_feature]
print("Numerical Features Count {}".format(len(numerical_feature)))
print("Discrete feature Count {}".format(len(discrete_feature)))
print("Continuous feature Count {}".format(len(continuous_feature)))
print("Categorical feature Count {}".format(len(categorical_feature)))
```

```
Numerical Features Count 3
Discrete feature Count 0
Continuous feature Count 3
Categorical feature Count 4
```

```
# Handle Missing Values
df.isnull().sum()*100/len(df)
```

	0
Date	0.000000
Location	0.000000
Temp	0.594665
WindSpeed	0.833906
Humidity	1.297264
RainToday	2.241853
RainTomorrow	2.245978

dtype: float64

```
print(numerical_feature)
```

```
['Temp', 'WindSpeed', 'Humidity']
```

```
def randomsampleimputation(df, variable):
    df[variable]=df[variable]
    random_sample=df[variable].dropna().sample(df[variable].isnull().sum(),random_state=0)
    random_sample.index=df[df[variable].isnull()].index
    df.loc[df[variable].isnull(),variable]=random_sample
```

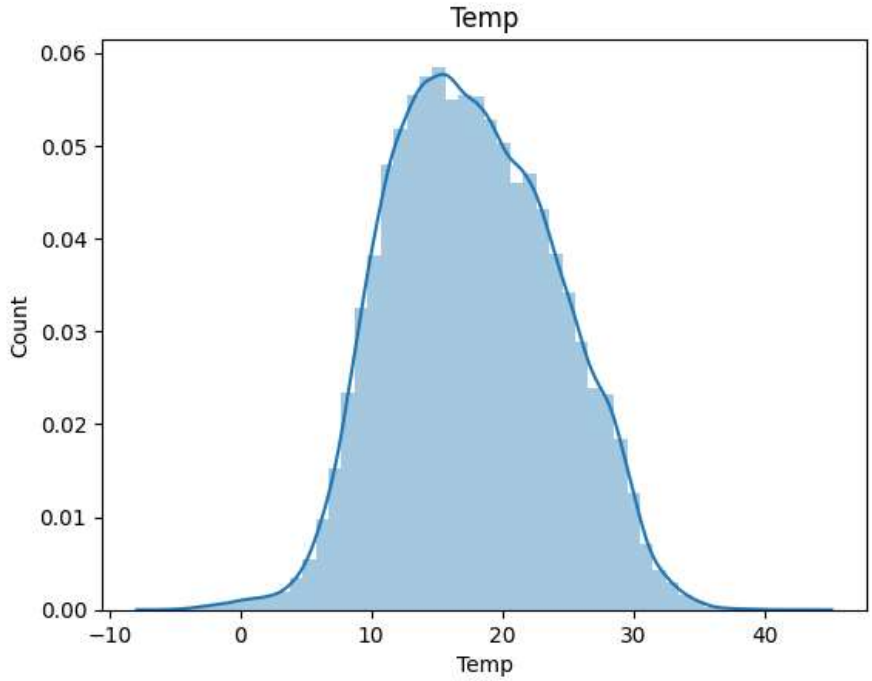
```
df
```

	Date	Location	Temp	WindSpeed	Humidity	RainToday	RainTomorrow
0	2008-12-01	Albury	18.15	22.0	46.5	No	No
1	2008-12-02	Albury	16.25	13.0	34.5	No	No
2	2008-12-03	Albury	19.30	22.5	34.0	No	No
3	2008-12-04	Albury	18.60	10.0	30.5	No	No
4	2008-12-05	Albury	24.90	13.5	57.5	No	No
...	...	...	...	...	...	...	...
145455	2017-06-21	Uluru	13.10	12.0	37.5	No	No
145456	2017-06-22	Uluru	14.45	11.0	38.5	No	No
145457	2017-06-23	Uluru	16.15	9.0	38.5	No	No
145458	2017-06-24	Uluru	17.40	10.0	37.5	No	No
145459	2017-06-25	Uluru	14.90	17.0	49.0	No	NaN

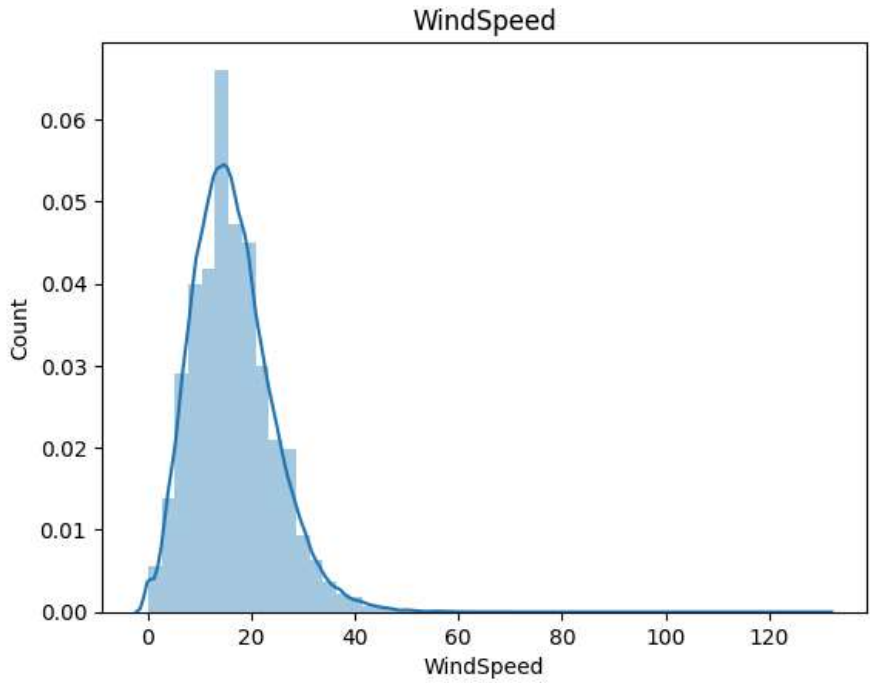
145460 rows × 7 columns

```
for feature in continuous_feature:
    data=df.copy()
    sns.distplot(df[feature])
    plt.xlabel(feature)
    plt.ylabel("Count")
    plt.title(feature)
    plt.figure(figsize=(15,15))
    plt.show()
```

```
<ipython-input-12-f3f36df57cc9>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df[feature])
```

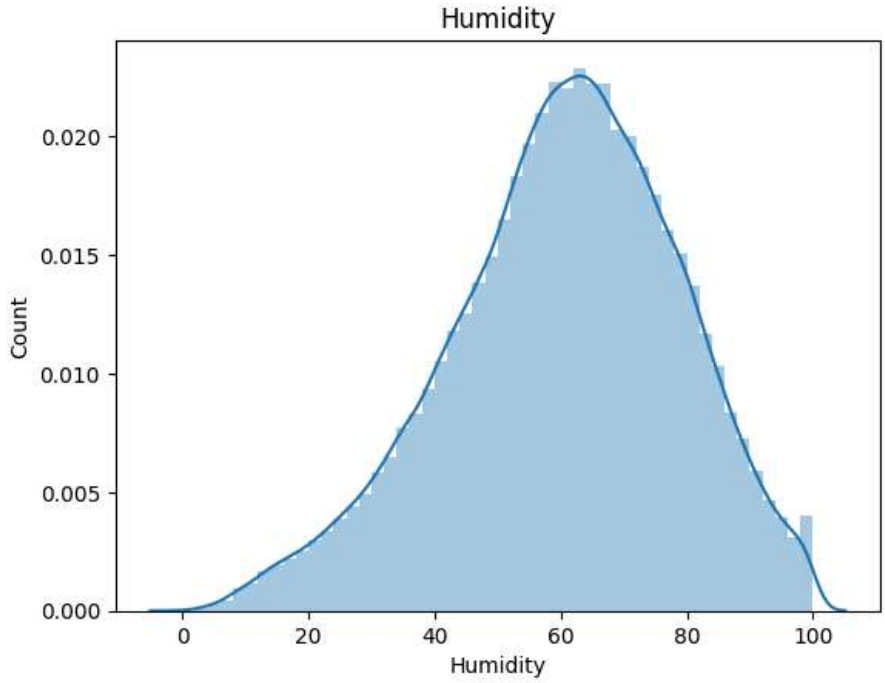



```
<Figure size 1500x1500 with 0 Axes>
<ipython-input-12-f3f36df57cc9>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df[feature])
```



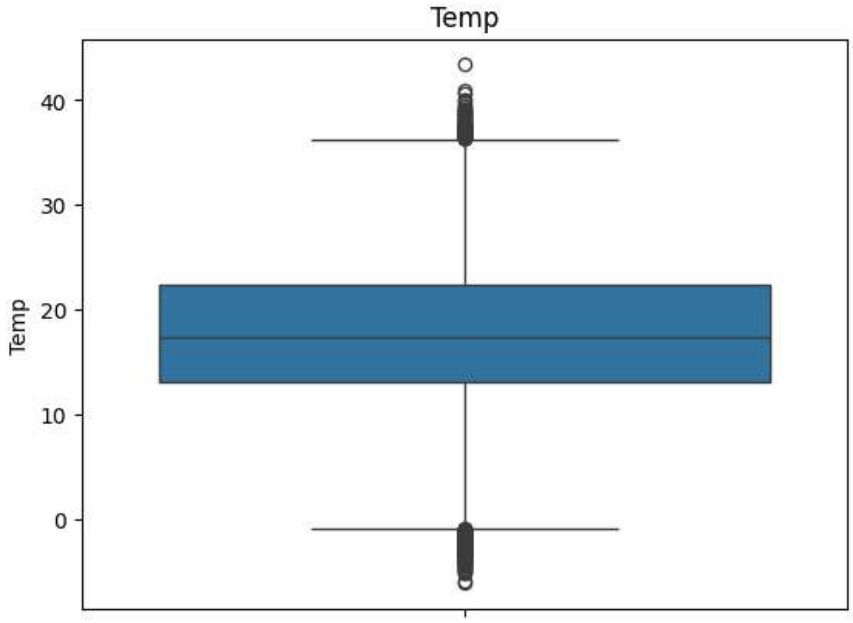

```
<Figure size 1500x1500 with 0 Axes>
<ipython-input-12-f3f36df57cc9>:3: UserWarning:
```

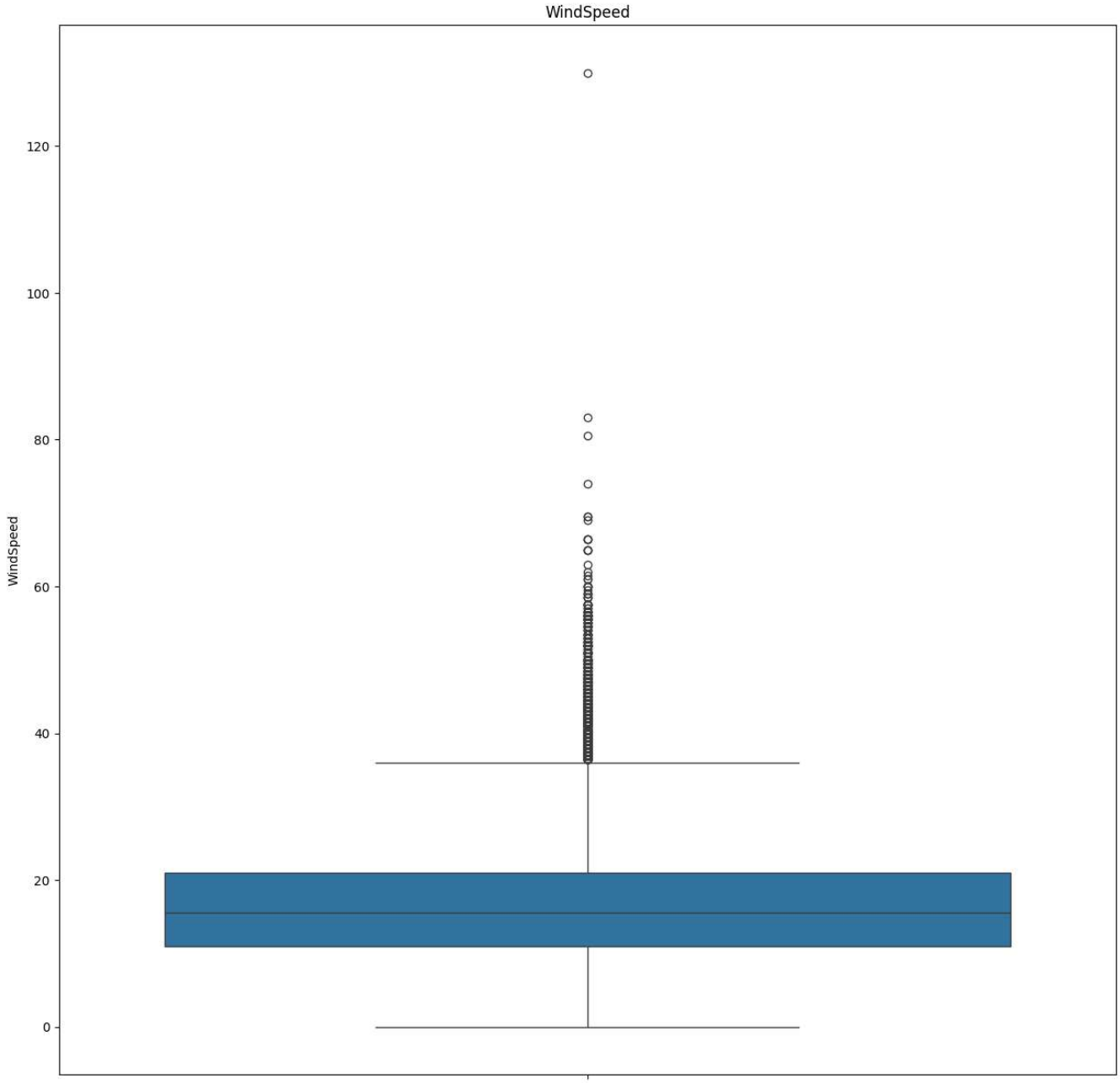
```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df[feature])
```

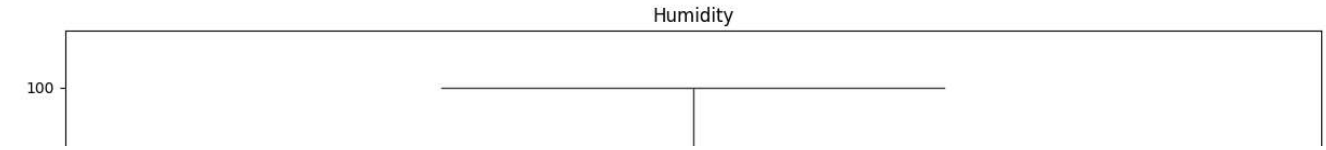


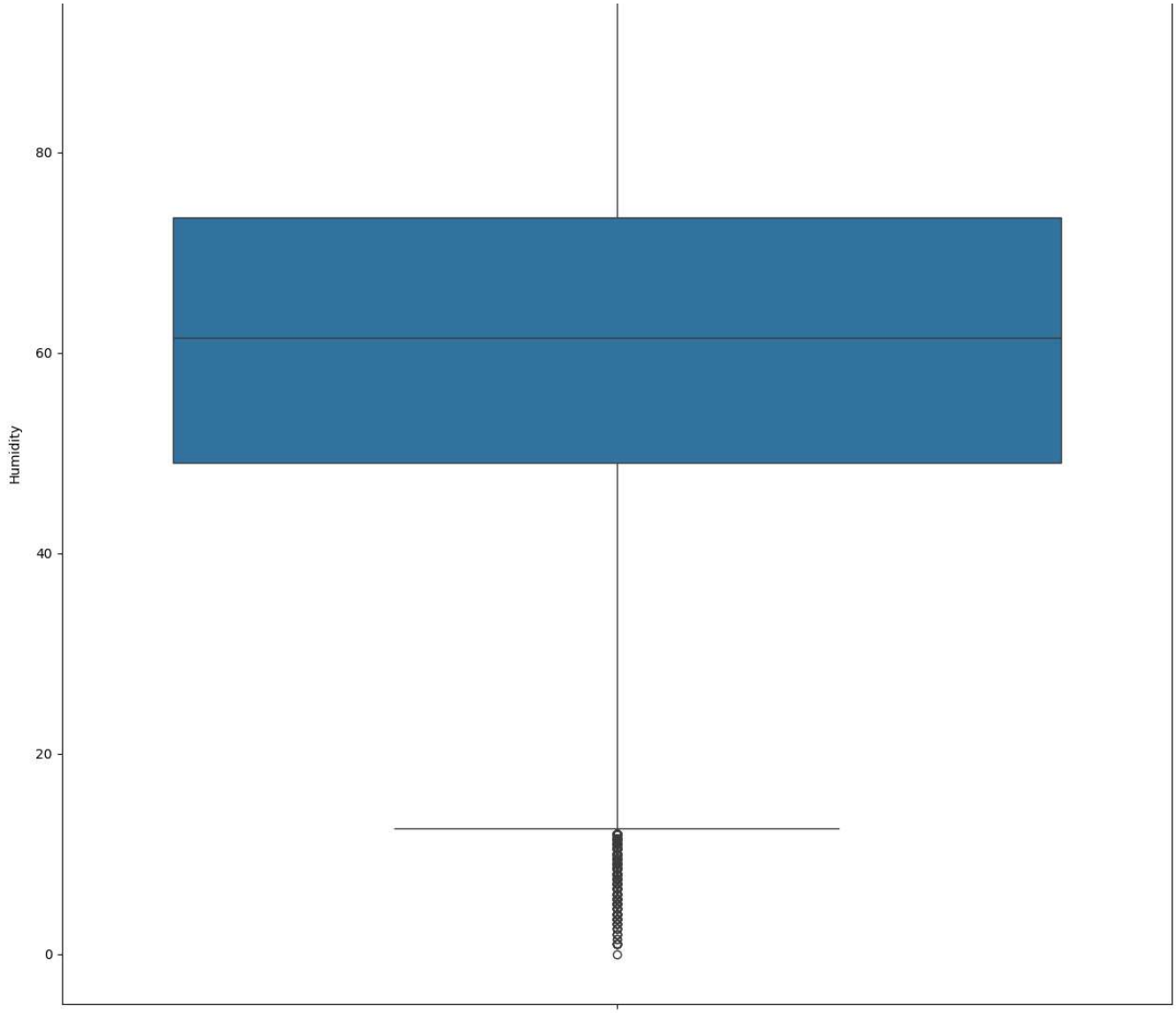

```
<Figure size 1500x1500 with 0 Axes>
```

```
#A for loop is used to plot a boxplot for all the continuous features to see the outliers
for feature in continuous_feature:
    data=df.copy()
    sns.boxplot(data[feature])
    plt.title(feature)
    plt.figure(figsize=(15,15))
```


Temp
Temp
40
30
20
10
0


WindSpeed
WindSpeed
120
100
80
60
40
20
0


Humidity
100




```
<Figure size 1500x1500 with 0 Axes>
```

```
for feature in continuous_feature:
    if(df[feature].isnull().sum()*100/len(df))>0:
        df[feature] = df[feature].fillna(df[feature].median())
```

```
df.isnull().sum()*100/len(df)
```

	0
Date	0.000000
Location	0.000000
Temp	0.000000
WindSpeed	0.000000
Humidity	0.000000
RainToday	2.241853
RainTomorrow	2.245978

dtype: float64

```
discrete_feature
```

```
[]
```

```
def mode_nan(df,variable):
    mode=df[variable].value_counts().index[0]
    df[variable].fillna(mode,inplace=True)

```

```
df["RainToday"] = pd.get_dummies(df["RainToday"], drop_first = True)
df["RainTomorrow"] = pd.get_dummies(df["RainTomorrow"], drop_first = True)
df
```

	Date	Location	Temp	WindSpeed	Humidity	RainToday	RainTomorrow
0	2008-12-01	Albury	18.15	22.0	46.5	False	False
1	2008-12-02	Albury	16.25	13.0	34.5	False	False
2	2008-12-03	Albury	19.30	22.5	34.0	False	False
3	2008-12-04	Albury	18.60	10.0	30.5	False	False
4	2008-12-05	Albury	24.90	13.5	57.5	False	False
...	...	...	...	...	...	...	...
145455	2017-06-21	Uluru	13.10	12.0	37.5	False	False
145456	2017-06-22	Uluru	14.45	11.0	38.5	False	False
145457	2017-06-23	Uluru	16.15	9.0	38.5	False	False
145458	2017-06-24	Uluru	17.40	10.0	37.5	False	False
145459	2017-06-25	Uluru	14.90	17.0	49.0	False	False

145460 rows × 7 columns

```
for feature in categorical_feature:
    print(feature, (df.groupby([feature])["RainTomorrow"].mean().sort_values(ascending = False)).index)
```

```
Date Index(['2007-11-01', '2007-12-15', '2008-02-03', '2008-01-31', '2008-01-30',
       '2008-01-19', '2008-01-18', '2008-01-16', '2008-01-12', '2007-11-30',
       ...
       '2008-05-08', '2008-05-09', '2008-01-03', '2008-01-02', '2008-05-10',
       '2008-05-11', '2008-05-12', '2008-05-13', '2008-05-14', '2007-12-25'],
      dtype='object', name='Date', length=3436)
Location Index(['Portland', 'Walpole', 'Cairns', 'Dartmoor', 'NorfolkIsland',
```

```
       'MountGambier', 'Albany', 'Witchcliffe', 'CoffsHarbour', 'MountGinini',
       'NorahHead', 'Darwin', 'Sydney', 'SydneyAirport', 'Ballarat',
       'GoldCoast', 'Watsonia', 'Newcastle', 'Hobart', 'Wollongong',
       'Williamtown', 'Launceston', 'Brisbane', 'MelbourneAirport', 'Adelaide',
       'Sale', 'Albury', 'Perth', 'Melbourne', 'Nuriootpa', 'Penrith',
       'BadgerysCreek', 'PerthAirport', 'Tuggeranong', 'Richmond', 'Bendigo',
       'Canberra', 'WaggaWagga', 'Townsville', 'Katherine', 'PearceRAAF',
       'SalmonGums', 'Nhil', 'Moree', 'Cobar', 'Mildura', 'AliceSprings',
       'Uluru', 'Woomera'],
      dtype='object', name='Location')
RainToday Index([True, False], dtype='bool', name='RainToday')
RainTomorrow Index([True, False], dtype='bool', name='RainTomorrow')
```

```
df1 = df.groupby(["Location"])["RainTomorrow"].value_counts().sort_values().unstack()
```

```
df1
```

RainTomorrow	False	True
Location		
Adelaide	2505	688
Albany	2138	902
Albury	2422	618
AliceSprings	2796	244
BadgerysCreek	2426	583
Ballarat	2259	781
Bendigo	2478	562
Brisbane	2484	709
Cairns	2090	950
Canberra	2807	629
Cobar	2623	386
CoffsHarbour	2140	869
Dartmoor	2087	922
Darwin	2341	852
GoldCoast	2265	775
Hobart	2432	761
Katherine	1313	265
Launceston	2341	699
Melbourne	2557	636
MelbourneAirport	2356	653
Mildura	2682	327
Moree	2615	394
MountGambier	2120	920
MountGinini	2221	819
Newcastle	2308	731
Nhil	1336	242
NorahHead	2196	808
NorfolkIsland	2090	919
Nuriootpa	2417	592
PearceRAAF	2504	505
Penrith	2444	595
Perth	2548	645
PerthAirport	2442	567
Portland	1914	1095
Richmond	2449	560
Sale	2366	643
SalmonGums	2529	472
Sydney	2479	865
SydneyAirport	2235	774
Townsville	2521	519
Tuggeranong	2471	568

Uluru	1462	116
WaggaWagga	2473	536
Walpole	2057	949
Watsonia	2271	738
Williamtown	2309	700
Witchcliffe	2130	879
Wollongong	2327	713
Woomera	2807	202

Next steps: Generate code with df1 | View recommended plots | New interactive sheet

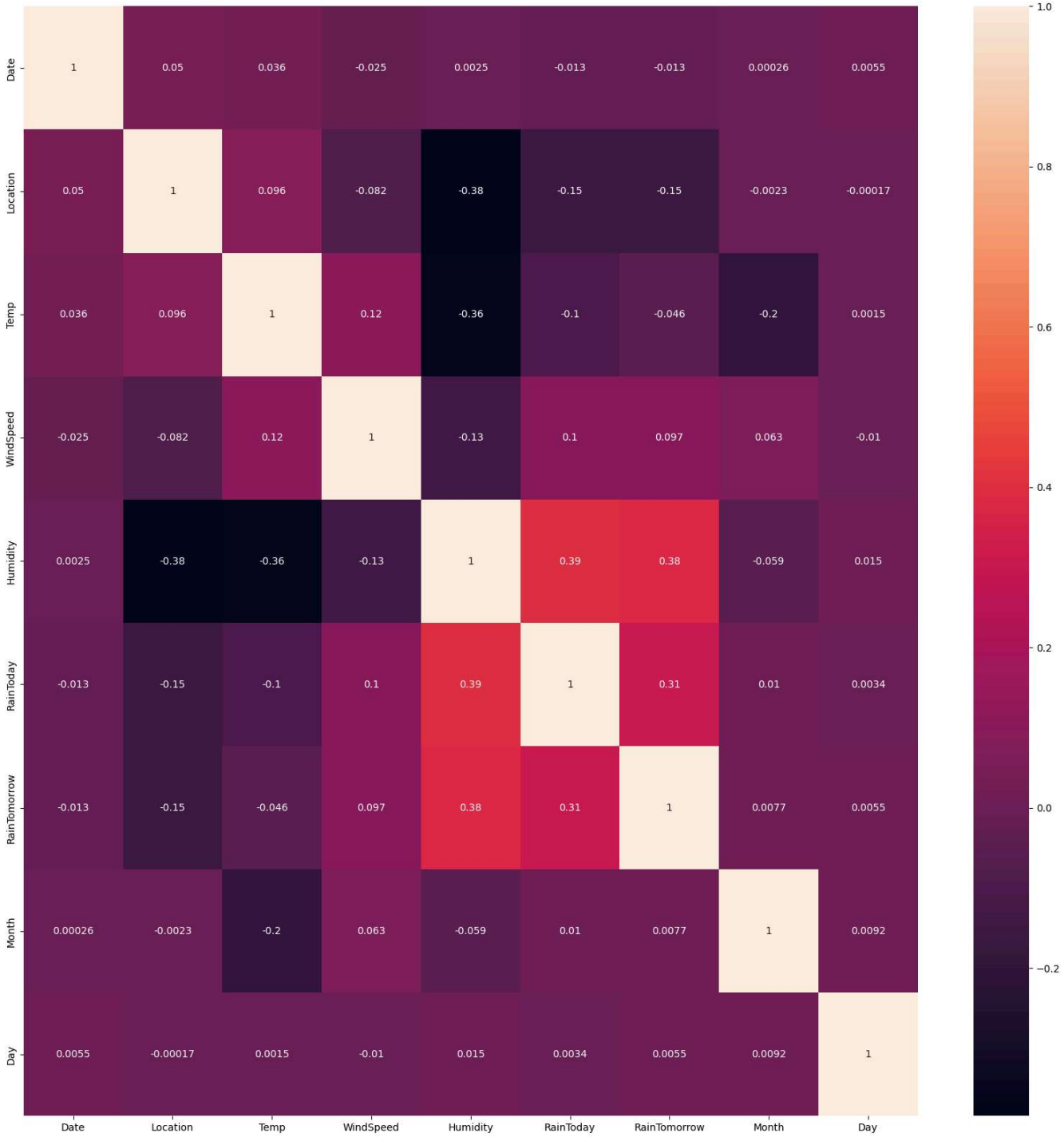
```
location = {'Portland':1, 'Cairns':2, 'Walpole':3, 'Dartmoor':4, 'MountGambier':5,
        'NorfolkIsland':6, 'Albany':7, 'Witchcliffe':8, 'CoffsHarbour':9, 'Sydney':10,
        'Darwin':11, 'MountGinini':12, 'NorahHead':13, 'Ballarat':14, 'GoldCoast':15,
        'SydneyAirport':16, 'Hobart':17, 'Watsonia':18, 'Newcastle':19, 'Wollongong':20,
        'Brisbane':21, 'Williamtown':22, 'Launceston':23, 'Adelaide':24, 'MelbourneAirport':25,
        'Perth':26, 'Sale':27, 'Melbourne':28, 'Canberra':29, 'Albury':30, 'Penrith':31,
        'Nuriootpa':32, 'BadgerysCreek':33, 'Tuggeranong':34, 'PerthAirport':35, 'Bendigo':36,
        'Richmond':37, 'WaggaWagga':38, 'Townsville':39, 'PearceRAAF':40, 'SalmonGums':41,
        'Moree':42, 'Cobar':43, 'Mildura':44, 'Katherine':45, 'AliceSprings':46, 'Nhil':47,
        'Woomera':48, 'Uluru':49}
df["Location"] = df["Location"].map(location)
```

```
df['Date'] = pd.to_datetime(df['Date'])
df['Month'] = df['Date'].dt.month
df['Day'] = df['Date'].dt.day
df
```

	Date	Location	Temp	WindSpeed	Humidity	RainToday	RainTomorrow	Month	Day
0	2008-12-01	30	18.15	22.0	46.5	False	False	12	1
1	2008-12-02	30	16.25	13.0	34.5	False	False	12	2
2	2008-12-03	30	19.30	22.5	34.0	False	False	12	3
3	2008-12-04	30	18.60	10.0	30.5	False	False	12	4
4	2008-12-05	30	24.90	13.5	57.5	False	False	12	5
...	...	...	...	...	...	...	...	...	...
145455	2017-06-21	49	13.10	12.0	37.5	False	False	6	21
145456	2017-06-22	49	14.45	11.0	38.5	False	False	6	22
145457	2017-06-23	49	16.15	9.0	38.5	False	False	6	23
145458	2017-06-24	49	17.40	10.0	37.5	False	False	6	24
145459	2017-06-25	49	14.90	17.0	49.0	False	False	6	25

145460 rows × 9 columns

```
corrmat = df.corr()
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(corrmat,annot=True)
```

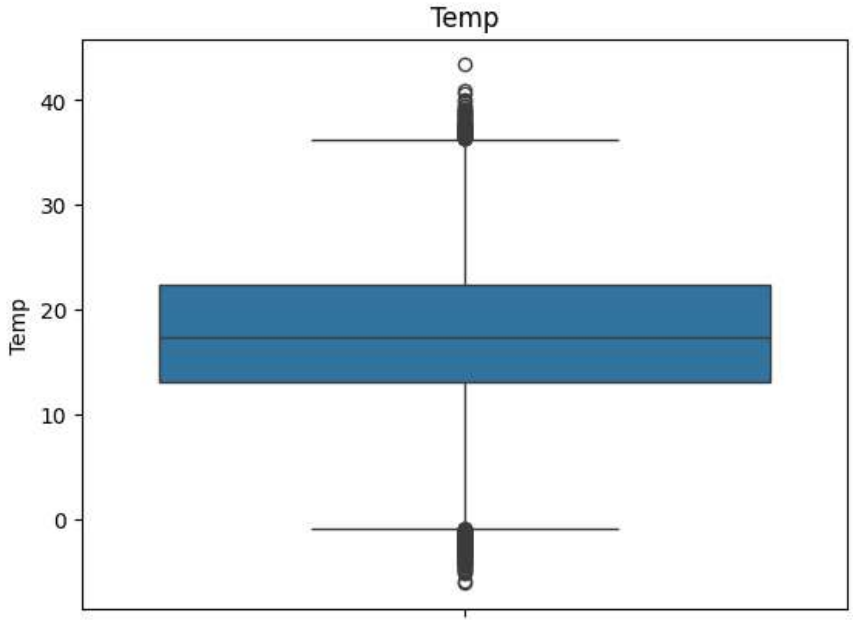

Date 1 0.05 0.036 -0.025 0.0025 -0.013 -0.013 0.00026 0.0055
Location 0.05 1 0.096 -0.082 -0.38 -0.15 -0.15 -0.0023 -0.00017
Temp 0.036 0.096 1 0.12 -0.36 -0.1 -0.046 -0.2 0.0015
WindSpeed -0.025 -0.082 0.12 1 -0.13 0.1 0.097 0.063 -0.01
Humidity 0.0025 -0.38 -0.36 -0.13 1 0.39 0.38 -0.059 0.015
RainToday -0.013 -0.15 -0.1 0.1 0.39 1 0.31 0.01 0.0034
RainTomorrow -0.013 -0.15 -0.046 0.097 0.38 0.31 1 0.0077 0.0055
Month 0.00026 -0.0023 -0.2 0.063 -0.059 0.01 0.0077 1 0.0092
Day 0.0055 -0.00017 0.0015 -0.01 0.015 0.0034 0.0055 0.0092 1
Date Location Temp WindSpeed Humidity RainToday RainTomorrow Month Day
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

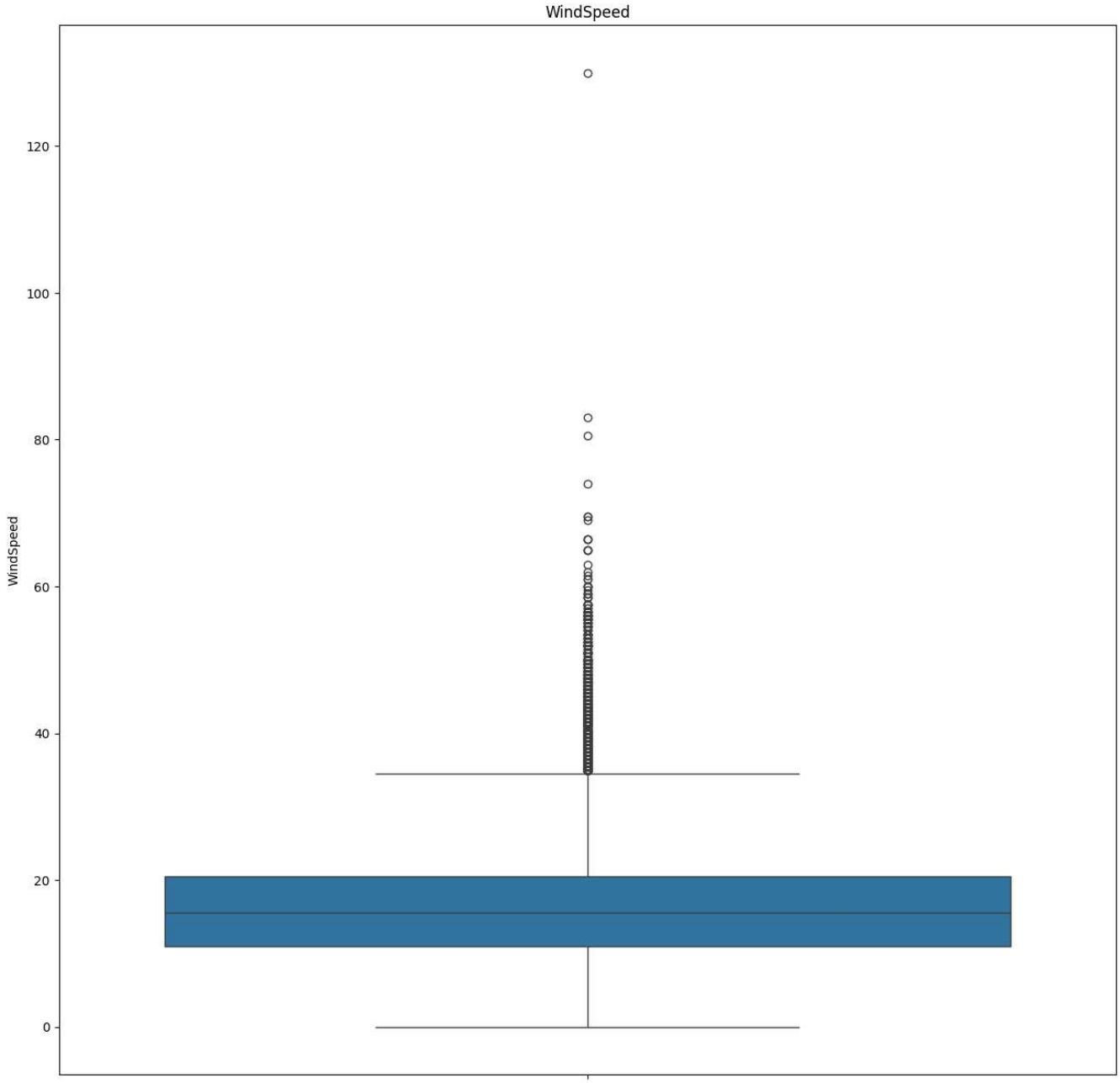
```
df
```

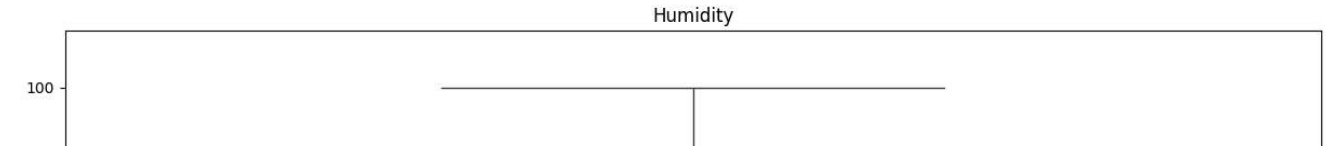
	Date	Location	Temp	WindSpeed	Humidity	RainToday	RainTomorrow	Month	Day
0	2008-12-01	30	18.15	22.0	46.5	False	False	12	1
1	2008-12-02	30	16.25	13.0	34.5	False	False	12	2
2	2008-12-03	30	19.30	22.5	34.0	False	False	12	3
3	2008-12-04	30	18.60	10.0	30.5	False	False	12	4
4	2008-12-05	30	24.90	13.5	57.5	False	False	12	5
...	...	...	...	...	...	...	...	...	...
145455	2017-06-21	49	13.10	12.0	37.5	False	False	6	21
145456	2017-06-22	49	14.45	11.0	38.5	False	False	6	22
145457	2017-06-23	49	16.15	9.0	38.5	False	False	6	23
145458	2017-06-24	49	17.40	10.0	37.5	False	False	6	24
145459	2017-06-25	49	14.90	17.0	49.0	False	False	6	25

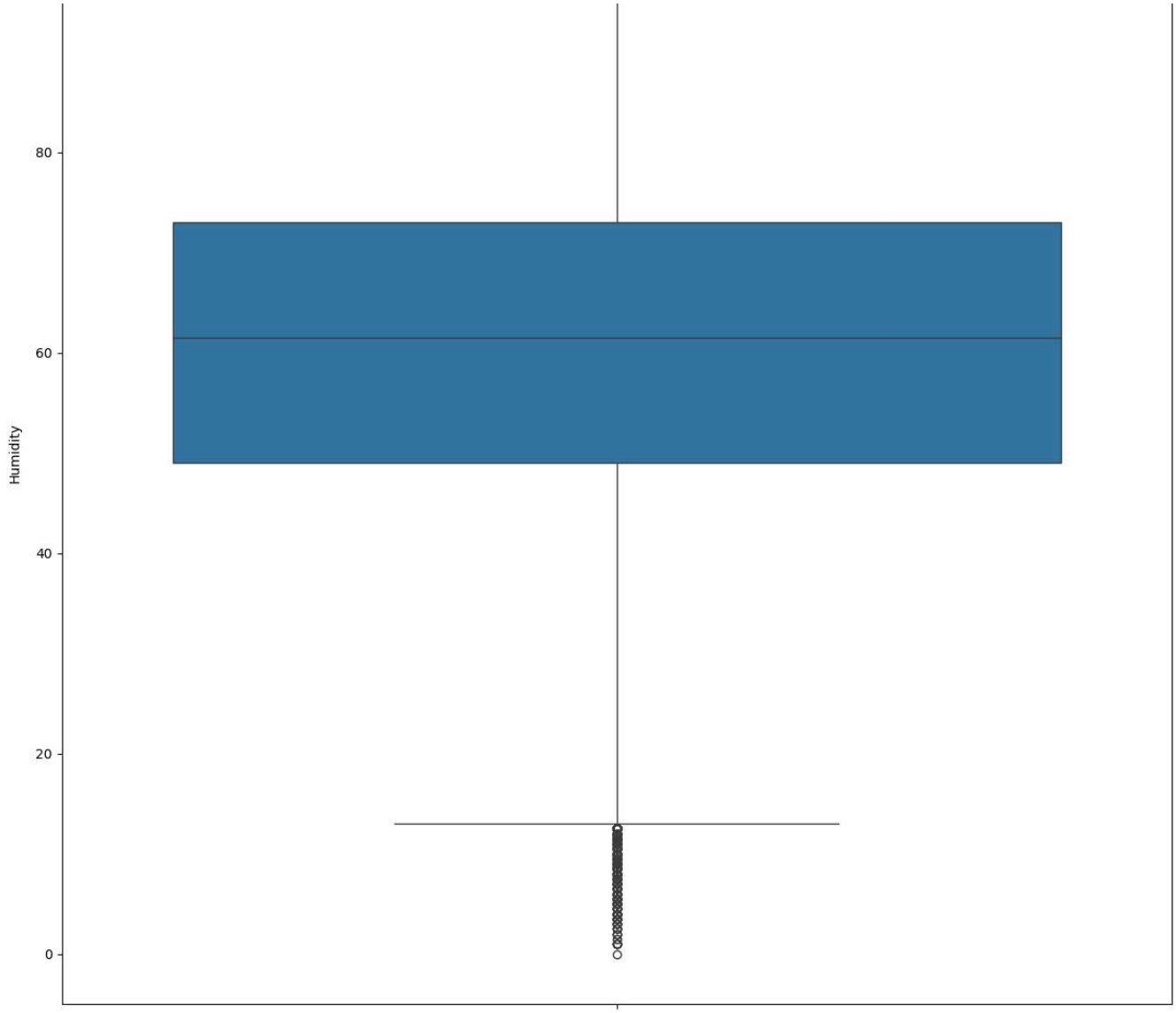
145460 rows × 9 columns

```
for feature in continuous_feature:
    data=df.copy()
    sns.boxplot(data[feature])
    plt.title(feature)
    plt.figure(figsize=(15,15))
```


Temp
Temp
40
30
20
10
0


WindSpeed
WindSpeed
120
100
80
60
40
20
0


Humidity
100




```
<Figure size 1500x1500 with 0 Axes>
```

```
for feature in continuous_feature:
    print(feature)
```

```
Temp
WindSpeed
Humidity
```

```
IQR=df.Temp.quantile(0.75)-df.Temp.quantile(0.25)
lower_bridge=df.Temp.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Temp.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
```

```
-0.9500000000000011 36.25
```

```
df.loc[df['Temp']>=36.975,'Temp']=36.975
df.loc[df['Temp']<=2.7,'Temp']=2.7
```

```
IQR=df.WindSpeed.quantile(0.75)-df.WindSpeed.quantile(0.25)
lower_bridge=df.WindSpeed.quantile(0.25)-(IQR*1.5)
upper_bridge=df.WindSpeed.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
```

```
-3.25 34.75
```

```
df.loc[df['WindSpeed']>=38.75,'WindSpeed']=38.75
df.loc[df['WindSpeed']<=-7.25,'WindSpeed']=-7.25
```

```
IQR=df.Humidity.quantile(0.75)-df.Humidity.quantile(0.25)
lower_bridge=df.Humidity.quantile(0.25)-(IQR*1.5)
upper_bridge=df.Humidity.quantile(0.75)+(IQR*1.5)
print(lower_bridge, upper_bridge)
```

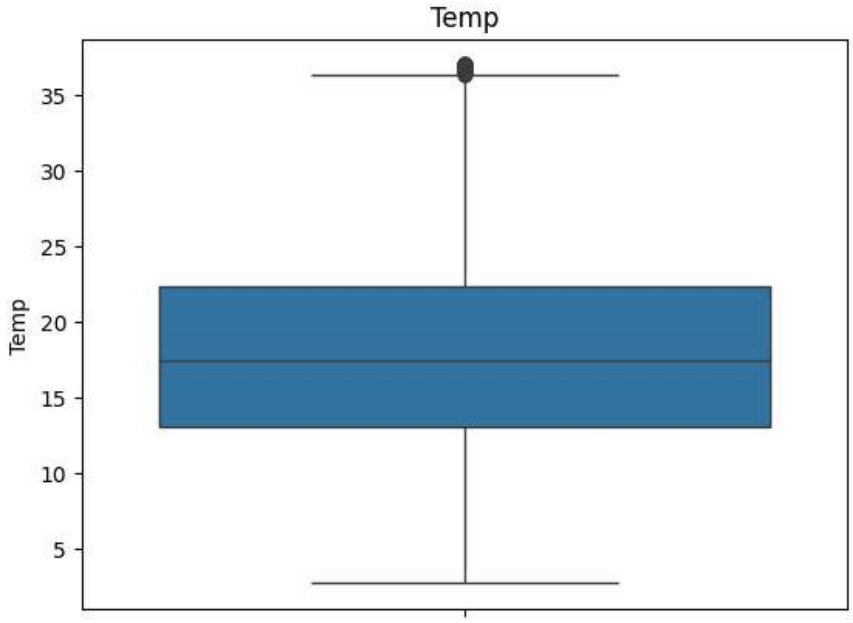
```
13.0 109.0
```

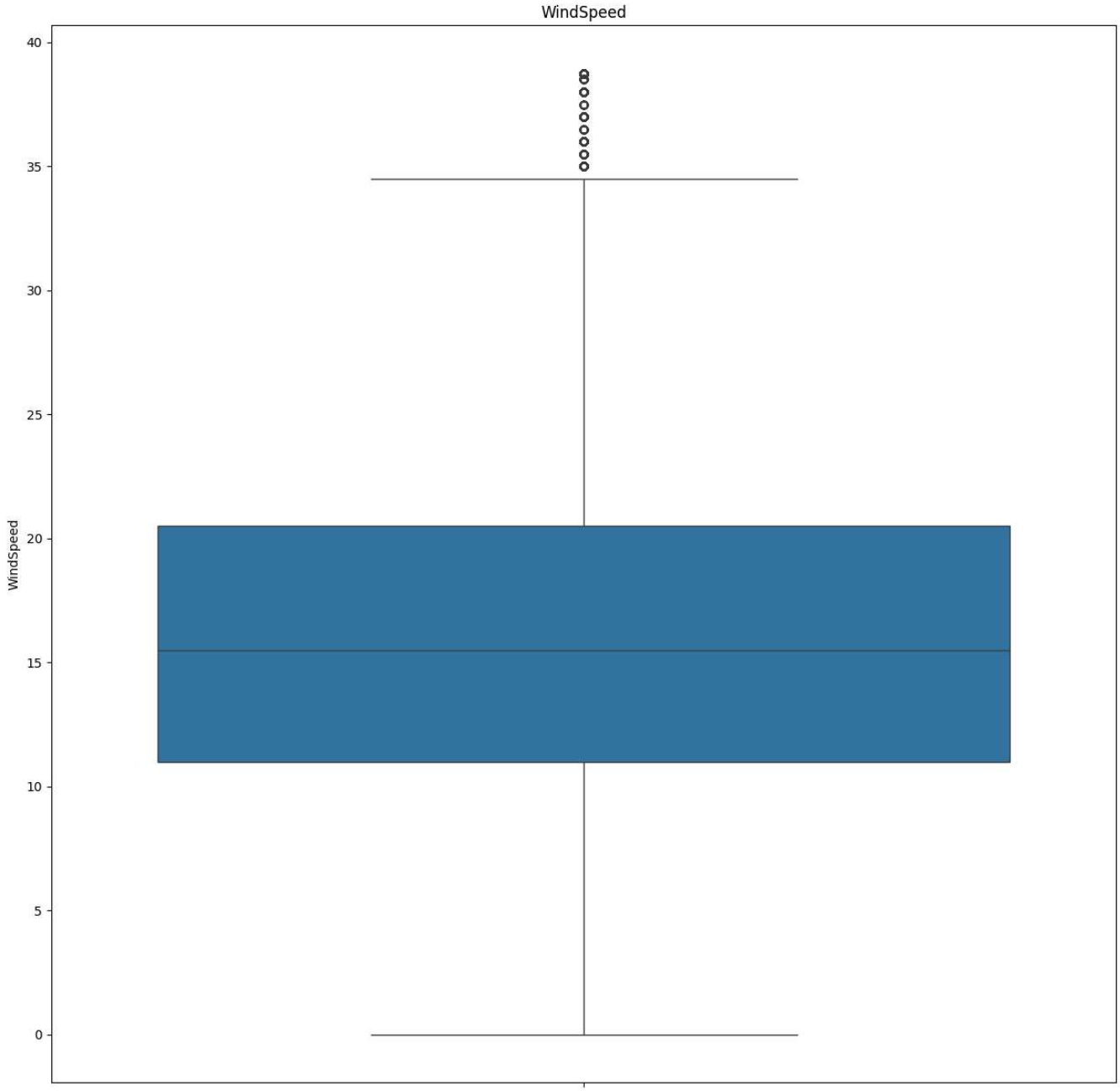
```
df.loc[df['Humidity']>=122,'Humidity']=122
df.loc[df['Humidity']<=18,'Humidity']=18
```

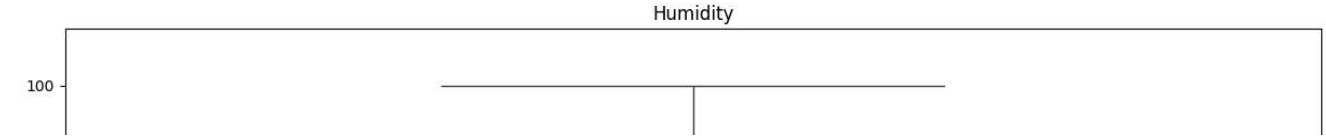
```
'''df.loc[df['Temp3pm']>=40.45,'Temp3pm']=40.45
df.loc[df['Temp3pm']<=2.45,'Temp3pm']=2.45'''
```

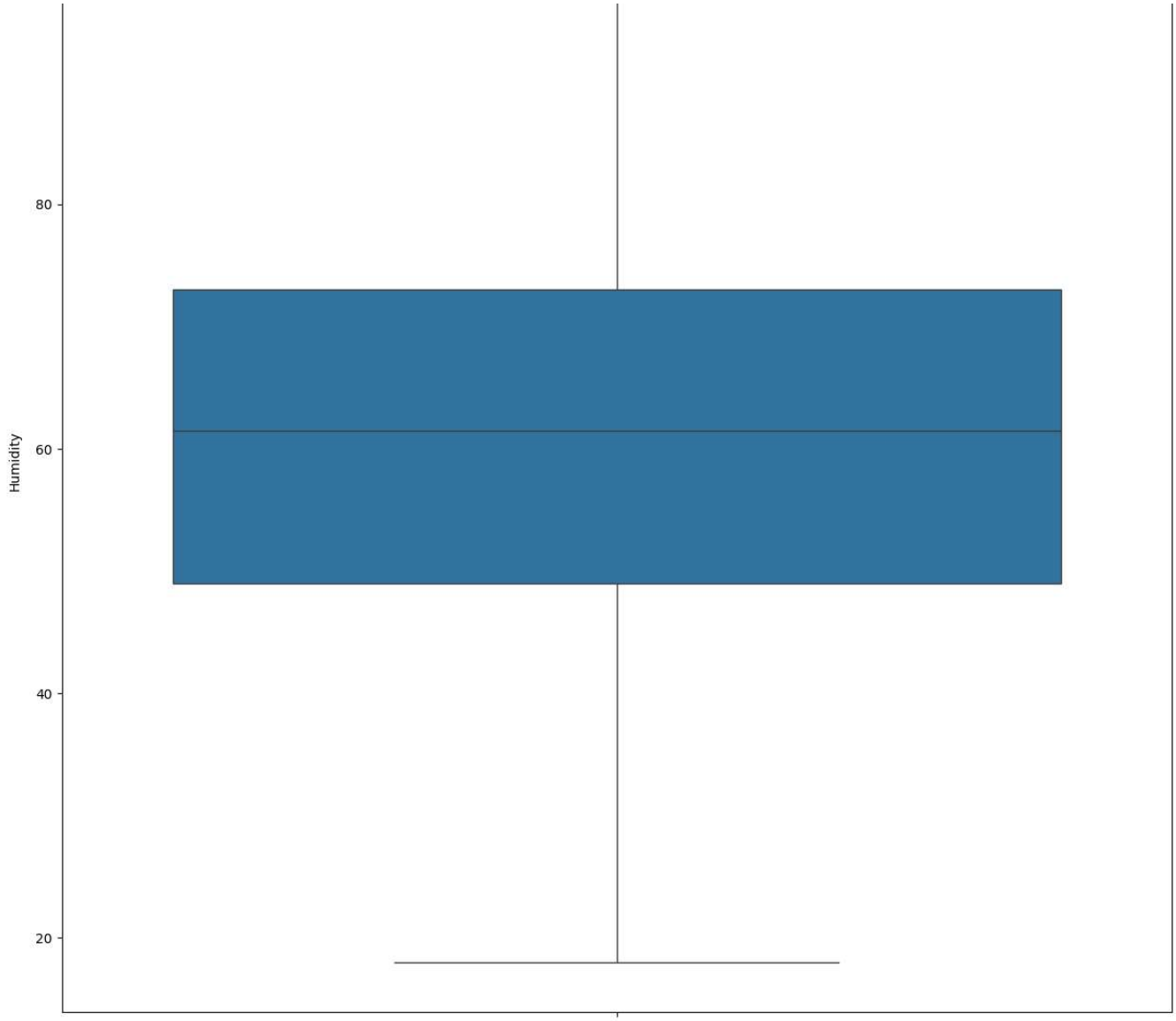
```
'df.loc[df['Temp3pm']>=40.45,'Temp3pm']=40.45\ndf.loc[df['Temp3pm']<=2.45,'Temp3pm']=2.45'
```

```
for feature in continuous_feature:
    data=df.copy()
    sns.boxplot(data[feature])
    plt.title(feature)
    plt.figure(figsize=(15,15))
```


Temp
Temp
35
30
25
20
15
10
5


WindSpeed
WindSpeed
40
35
30
25
20
15
10
5
0


Humidity
100




```
<Figure size 1500x1500 with 0 Axes>
```

```
def qq_plots(df, variable):
    plt.figure(figsize=(15,6))
    plt.subplot(1, 2, 1)
    df[variable].hist()
    plt.subplot(1, 2, 2)
    stats.probplot(df[variable], dist="norm", plot=plt)
    plt.show()
```

```
for feature in continuous_feature:
    print(feature)
    plt.figure(figsize=(15,6))
    plt.subplot(1, 2, 1)
    df[feature].hist()
    plt.subplot(1, 2, 2)
    stats.probplot(df[feature], dist="norm", plot=plt)
    plt.show()
```