# Assignment_1

April 24, 2021

[1]:
```
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
import wget
```

## 0.1 Get the dataset

Download from my github, I have upload the cleaned dataset to my github, so, In this assignment, I needn't clean it again. And use the wget module in python to download it in the current directory, If that doesn't work, please check your connection.

[2]:
```
print('Beginning file download with wget module')
url = 'https://raw.githubusercontent.com/wlof-2/
 ↪Statistic_Machine_Learning_Course/main/data/AUS_Weather.csv'
# wget.download(url, './data.csv')
dataset = pd.read_csv('./data.csv')
# save the dataset file with the path './data.csv'
```

```
Beginning file download with wget module
```

[3]:
```
dataset
```

[3]:
```
              Date Location  MinTemp  MaxTemp  Rainfall  Evaporation  \
0       2008-12-01   Albury     13.4     22.9       0.6          4.8
1       2008-12-02   Albury      7.4     25.1       0.0          4.8
2       2008-12-03   Albury     12.9     25.7       0.0          4.8
3       2008-12-04   Albury      9.2     28.0       0.0          4.8
4       2008-12-05   Albury     17.5     32.3       1.0          4.8
…              …      …        …      …        …         …
145455  2017-06-21    Uluru      2.8     23.4       0.0          4.8
145456  2017-06-22    Uluru      3.6     25.3       0.0          4.8
145457  2017-06-23    Uluru      5.4     26.9       0.0          4.8
145458  2017-06-24    Uluru      7.8     27.0       0.0          4.8
145459  2017-06-25    Uluru     14.9     22.6       0.0          4.8

        Sunshine WindGustDir  WindGustSpeed WindDir9am  … Pressure3pm  \
0            8.4           W           44.0          W  …      1007.1
1            8.4         WNW           44.0        NNW  …      1007.8
2            8.4         WSW           46.0          W  …      1008.7
```

```
3             8.4          NE          24.0        SE  …      1012.8
4             8.4           W          41.0       ENE  …      1006.0
…          …        …          …        …  …       …
145455        8.4           E          31.0        SE  …      1020.3
145456        8.4         NNW          22.0        SE  …      1019.1
145457        8.4           N          37.0        SE  …      1016.8
145458        8.4          SE          28.0       SSE  …      1016.5
145459        8.4           W          39.0       ESE  …      1017.9

        Cloud9am  Cloud3pm  Temp9am  Temp3pm  RainToday  RainTomorrow  Day  \
0            8.0       5.0     16.9     21.8         No            No    1
1            5.0       5.0     17.2     24.3         No            No    2
2            5.0       2.0     21.0     23.2         No            No    3
3            5.0       5.0     18.1     26.5         No            No    4
4            7.0       8.0     17.8     29.7         No            No    5
…          …      …     …      …        …   …
145455       5.0       5.0     10.1     22.4         No            No   21
145456       5.0       5.0     10.9     24.5         No            No   22
145457       5.0       5.0     12.5     26.1         No            No   23
145458       3.0       2.0     15.1     26.0         No            No   24
145459       8.0       8.0     15.0     20.9         No            No   25

        Month  Year
0          12  2008
1          12  2008
2          12  2008
3          12  2008
4          12  2008
…       …   …
145455      6  2017
145456      6  2017
145457      6  2017
145458      6  2017
145459      6  2017

[145460 rows x 26 columns]
```

## 0.2 General description of the dataset

This dataset contains about 10 years of daily weather observations from many locations across Australia. RainTomorrow is the target variable to predict. It means – did it rain the next day, Yes or No? This column is Yes if the rain for that day was 1mm or more. And the feature is some weather information today, for example, temperature, wind, sunshine etc.

[4]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
```

```
Data columns (total 26 columns):
Date             145460 non-null object
Location         145460 non-null object
MinTemp          145460 non-null float64
MaxTemp          145460 non-null float64
Rainfall         145460 non-null float64
Evaporation      145460 non-null float64
Sunshine         145460 non-null float64
WindGustDir      145460 non-null object
WindGustSpeed    145460 non-null float64
WindDir9am       145460 non-null object
WindDir3pm       145460 non-null object
WindSpeed9am     145460 non-null float64
WindSpeed3pm     145460 non-null float64
Humidity9am      145460 non-null float64
Humidity3pm      145460 non-null float64
Pressure9am      145460 non-null float64
Pressure3pm      145460 non-null float64
Cloud9am         145460 non-null float64
Cloud3pm         145460 non-null float64
Temp9am          145460 non-null float64
Temp3pm          145460 non-null float64
RainToday        145460 non-null object
RainTomorrow     145460 non-null object
Day              145460 non-null int64
Month            145460 non-null int64
Year             145460 non-null int64
dtypes: float64(16), int64(3), object(7)
memory usage: 28.9+ MB
```

[5]:

```
# Encode object type labels with value between 0 and n_classes-1 to calculate
↪the standard deviations of these features. However, the means, min, max of
↪these variables actually are meaningless.
le =  LabelEncoder()
for i in dataset:
    if dataset[i].dtype=='object':
        dataset[i] = le.fit_transform(dataset[i])
    else:
        continue
```

[6]:

```
print(dataset.dtypes)
```

```
Date               int32
Location           int32
MinTemp          float64
MaxTemp          float64
Rainfall         float64
Evaporation      float64
```

```
Sunshine          float64
WindGustDir         int32
WindGustSpeed     float64
WindDir9am          int32
WindDir3pm          int32
WindSpeed9am      float64
WindSpeed3pm      float64
Humidity9am       float64
Humidity3pm       float64
Pressure9am       float64
Pressure3pm       float64
Cloud9am          float64
Cloud3pm          float64
Temp9am           float64
Temp3pm           float64
RainToday           int32
RainTomorrow        int32
Day                 int64
Month               int64
Year                int64
dtype: object
```

[7]:
```
# For some variables, such as locations, it is categorical variable, and we␣
↪encode it with number, so, when analysis the means, standard deviations, min␣
↪and max values fo the variables, we should analysis categorical type and␣
↪numerical type respectively.
categorical = [i for i in dataset.columns if dataset[i].dtype=='int32']
numerical = [i for i in dataset.columns if dataset[i].dtype=='float64' or␣
↪dataset[i].dtype=='int64']
```

[8]:
```
dataset[numerical].mean()
```

[8]:
```
MinTemp            12.192053
MaxTemp            23.215962
Rainfall            2.307990
Evaporation         5.179779
Sunshine            7.989889
WindGustSpeed      39.962189
WindSpeed9am       14.030751
WindSpeed3pm       18.669758
Humidity9am        68.901251
Humidity3pm        51.553396
Pressure9am      1017.644768
Pressure3pm      1015.250115
Cloud9am            4.659755
Cloud3pm            4.709913
Temp9am            16.987101
```

```
Temp3pm             21.668916
Day                 15.712258
Month                6.399615
Year              2012.769751
dtype: float64
```

[9]:
```
dataset[numerical].max()
```

[9]:
```
MinTemp             33.9
MaxTemp             48.1
Rainfall           371.0
Evaporation        145.0
Sunshine            14.5
WindGustSpeed      135.0
WindSpeed9am       130.0
WindSpeed3pm        87.0
Humidity9am        100.0
Humidity3pm        100.0
Pressure9am       1041.0
Pressure3pm       1039.6
Cloud9am             9.0
Cloud3pm             9.0
Temp9am             40.2
Temp3pm             46.7
Day                 31.0
Month               12.0
Year              2017.0
dtype: float64
```

[10]:
```
dataset[numerical].min()
```

[10]:
```
MinTemp             -8.5
MaxTemp             -4.8
Rainfall             0.0
Evaporation          0.0
Sunshine             0.0
WindGustSpeed        6.0
WindSpeed9am         0.0
WindSpeed3pm         0.0
Humidity9am          0.0
Humidity3pm          0.0
Pressure9am        980.5
Pressure3pm        977.1
Cloud9am             0.0
Cloud3pm             0.0
Temp9am             -7.2
Temp3pm             -5.4
```

```
Day                1.0
Month              1.0
Year            2007.0
dtype: float64
```

[11]:
```
dataset.std()
```

[11]:
```
Date            884.988002
Location         14.228687
MinTemp           6.365780
MaxTemp           7.088358
Rainfall          8.389771
Evaporation       3.178819
Sunshine          2.757790
WindGustDir       4.694110
WindGustSpeed    13.120931
WindDir9am        4.515839
WindDir3pm        4.538135
WindSpeed9am      8.861796
WindSpeed3pm      8.716716
Humidity9am      18.855360
Humidity3pm      20.471345
Pressure9am       6.728484
Pressure3pm       6.663994
Cloud9am          2.281490
Cloud3pm          2.106768
Temp9am           6.449299
Temp3pm           6.850658
RainToday         0.413683
RainTomorrow      0.413669
Day               8.794789
Month             3.427262
Year              2.537684
dtype: float64
```

### 0.2.1 Means, standard deviations, min and max values of variables

1. we find that Location and WindGustSpeed have relatively large standard deviations, for location, the climate of different area are usually different, so, some areas may often rains, but others never, such as desert. For the WindGustSpeed, because the gust usually have different strength, so, they have big difference at different day. We can see windspeed's standard deviations are smaller than the WindGustSpeed

2. Humidity9am and Humidity3am have the biggest standard deviations, I think because the humidity could be affected by many factor, for example, the humidity different in sunny and rainy day, also very different in desert and forest.

3. For the temperature, the max and min different a lot, this might be different in different

climate.

4. Rainfall's max is very large, and min is relatively small, but the standard deviations is relatively small, and means is also small, This means that the rainfall is at a relatively stable level throughout the year. And the Evaporation is similiar with the Rainfall.

5. We can see, the sunshine is relatively stable, and the Cloud is similiar.

6. Pressure is different in different day but the different is not large.

7. some information for the standard deviations also have no means, for example, the day which is about date.

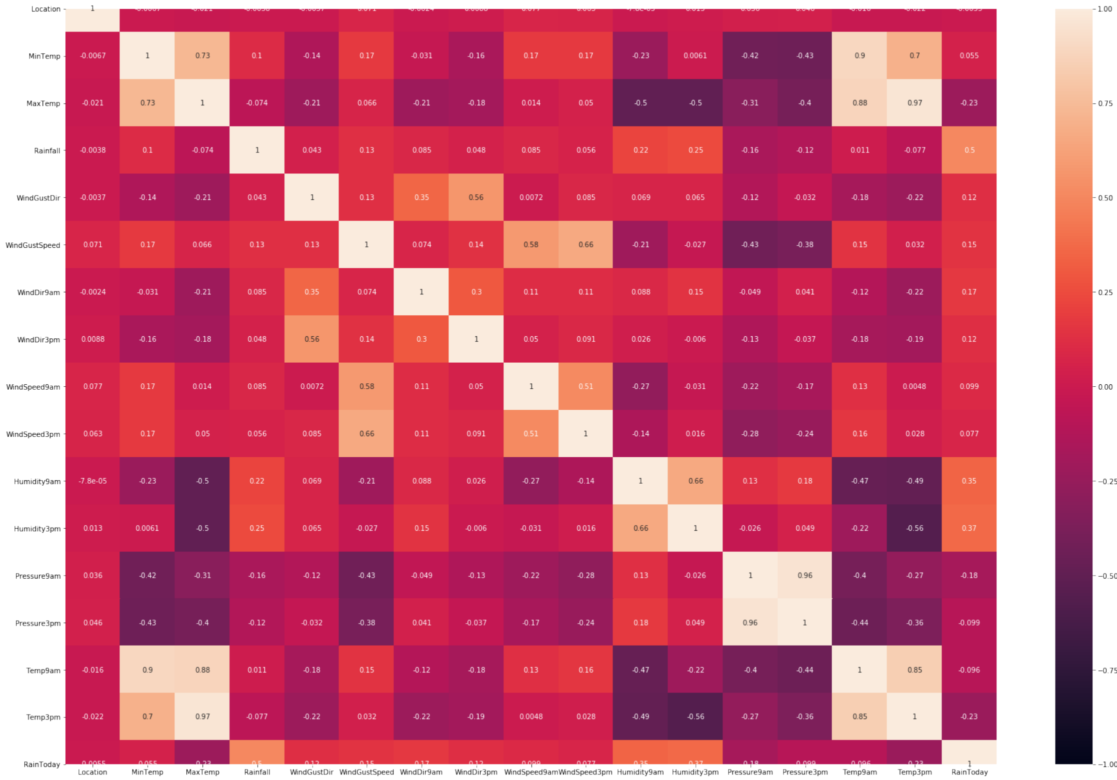
### 0.3 Explanations of the most important variables

From the above analysis, we can see, some variables are very important, such as Humidity, windspeed, but some variables we can see make no contribution, these variables we could drop out, and the others are important variables. Some variables have no relationship with weather making no contribution in this problem, for example, 'Date', 'Day', 'Month', 'Year', and some variables are too stable, therefore they have little contribution to the result, such as 'Sunshine', 'Cloud', and 'Evaporation', so we drop these variables, and get the important variables. For the 'RainTomorrow', we deal it as target.

[12]:
```
dataset_important = dataset.drop(['Date', 'Day', 'Month', 'Year', 
 ↪'RainTomorrow', 'Sunshine', 'Evaporation', 'Cloud9am', 'Cloud3pm'], axis=1)
RainTomorrow = dataset.loc[:, ['RainTomorrow']]
```

[13]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

[14]:
```
# Correlations between the most important variables between each other
plt.figure(figsize=(30,20))
heatmap = sns.heatmap(dataset_important.corr(), vmin=-1, vmax=1, annot=True)
plt.show()
```




## 0.4 Analysis of Correlations

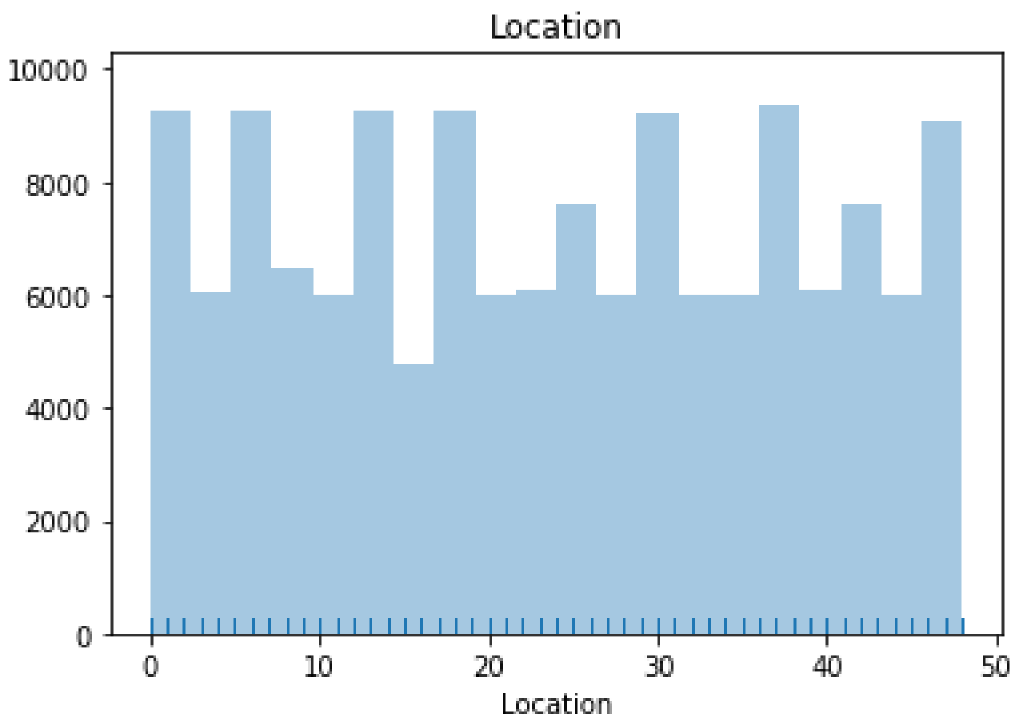
We analyze the Correlations in descending order
1. Temp9am(89%) and Temp3pm(98%) has high correlation with MaxTemp, This may be caused by little temperature change during the day.
2. Windspeed9am(58%) and Windspeed3pm(66%) has relatively high correlation with WindGust-Speed, because wind allways comes with the Gust wind
3. WindDir9am(35%) and WindDir3pm(56%) has relatively high correlation with WindDir, because the wind direction hardly change in the day.
4. Humidity9am and Humidity3pm also has not low correlation with the rainfall, because in rainy day, these variable will be high

[15]:
```
dataset_important.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 17 columns):
Location         145460 non-null int32
MinTemp          145460 non-null float64
MaxTemp          145460 non-null float64
Rainfall         145460 non-null float64
WindGustDir      145460 non-null int32
WindGustSpeed    145460 non-null float64
```

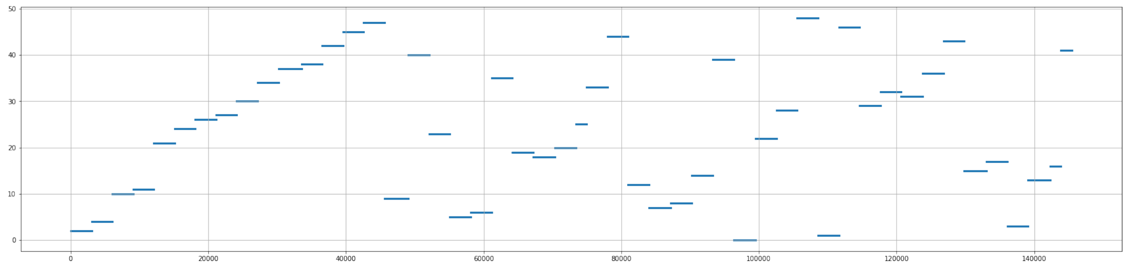
```
WindDir9am       145460 non-null int32
WindDir3pm       145460 non-null int32
WindSpeed9am     145460 non-null float64
WindSpeed3pm     145460 non-null float64
Humidity9am      145460 non-null float64
Humidity3pm      145460 non-null float64
Pressure9am      145460 non-null float64
Pressure3pm      145460 non-null float64
Temp9am          145460 non-null float64
Temp3pm          145460 non-null float64
RainToday        145460 non-null int32
dtypes: float64(12), int32(5)
memory usage: 16.1 MB
```

[16]:
```
Location = dataset_important['Location']
sns.distplot(Location, bins=20, hist=True, kde=False, norm_hist=False, rug=True,
             vertical=False, axlabel=None, label=None, ax=None,
             fit=None)
plt.title('Location')
plt.show()
```




[17]:
```
distplot = dataset_important['Location']
fig = plt.figure(figsize = (30,15))
ax1 = fig.add_subplot(2,1,1)  #   1
```

```
ax1.scatter(distplot.index, distplot.values, s =4)
plt.grid()
```




From the histogram and scatter plot, for the distribution of Location, we could know it is Discretely distributed.

## 0.5 Analysis of the variables' distribution

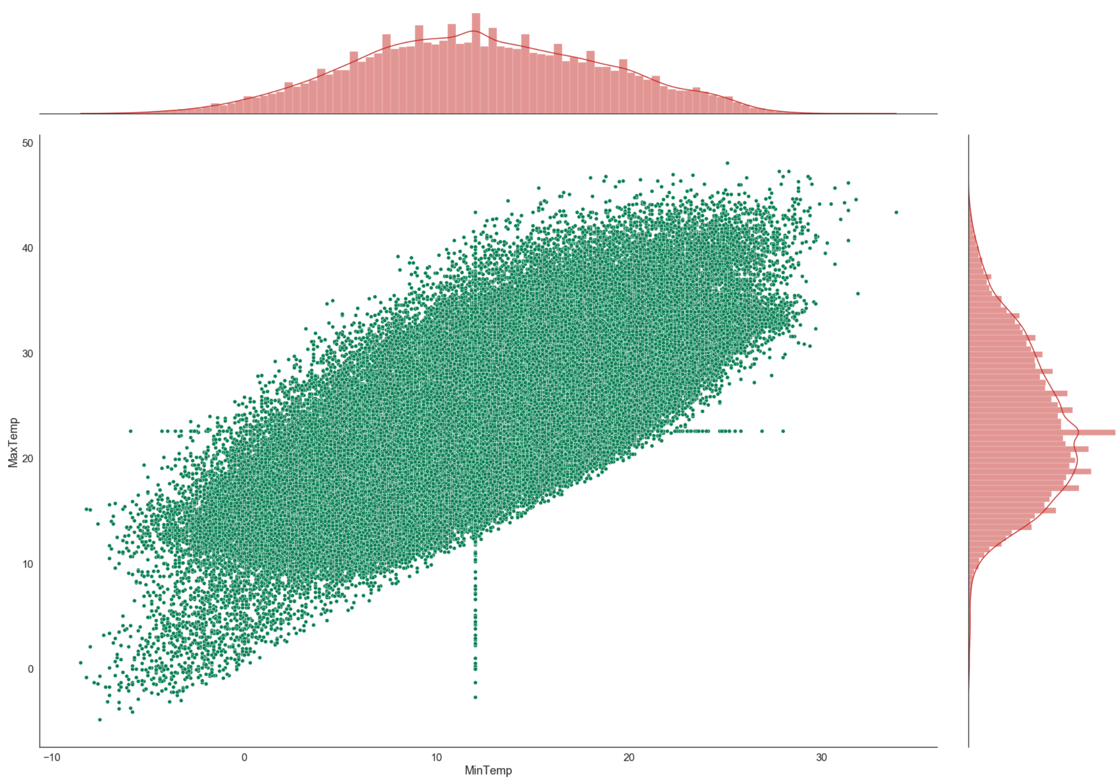
[18]:
```
# Determine whether the continuous variable conforms to the normal distribution
from scipy import stats
for i in dataset_important.columns:
    if dataset_important[i].dtype=='float64':
        u = dataset_important[i].mean()
        std = dataset_important[i].std()
        print(i)
        print(stats.kstest(dataset_important[i], 'norm', (u, std)))
```

```
MinTemp
KstestResult(statistic=0.02020132575032385, pvalue=5.501346820100461e-52)
MaxTemp
KstestResult(statistic=0.03997212354682511, pvalue=2.6963400408630606e-202)
Rainfall
KstestResult(statistic=0.3916213615158128, pvalue=0.0)
WindGustSpeed
KstestResult(statistic=0.11686839486405898, pvalue=0.0)
WindSpeed9am
KstestResult(statistic=0.0955281730184242, pvalue=0.0)
WindSpeed3pm
KstestResult(statistic=0.10284188580373943, pvalue=0.0)
Humidity9am
KstestResult(statistic=0.049539818797650126, pvalue=1.6828863214963e-310)
Humidity3pm
KstestResult(statistic=0.03321798481815452, pvalue=7.72059154132862e-140)
Pressure9am
KstestResult(statistic=0.05537672092496049, pvalue=0.0)
Pressure3pm
KstestResult(statistic=0.05545439139632846, pvalue=0.0)
```

```
Temp9am
KstestResult(statistic=0.026237124168665193, pvalue=2.1225028352392918e-87)
Temp3pm
KstestResult(statistic=0.047536125945820906, pvalue=6.330801749347389e-286)
```

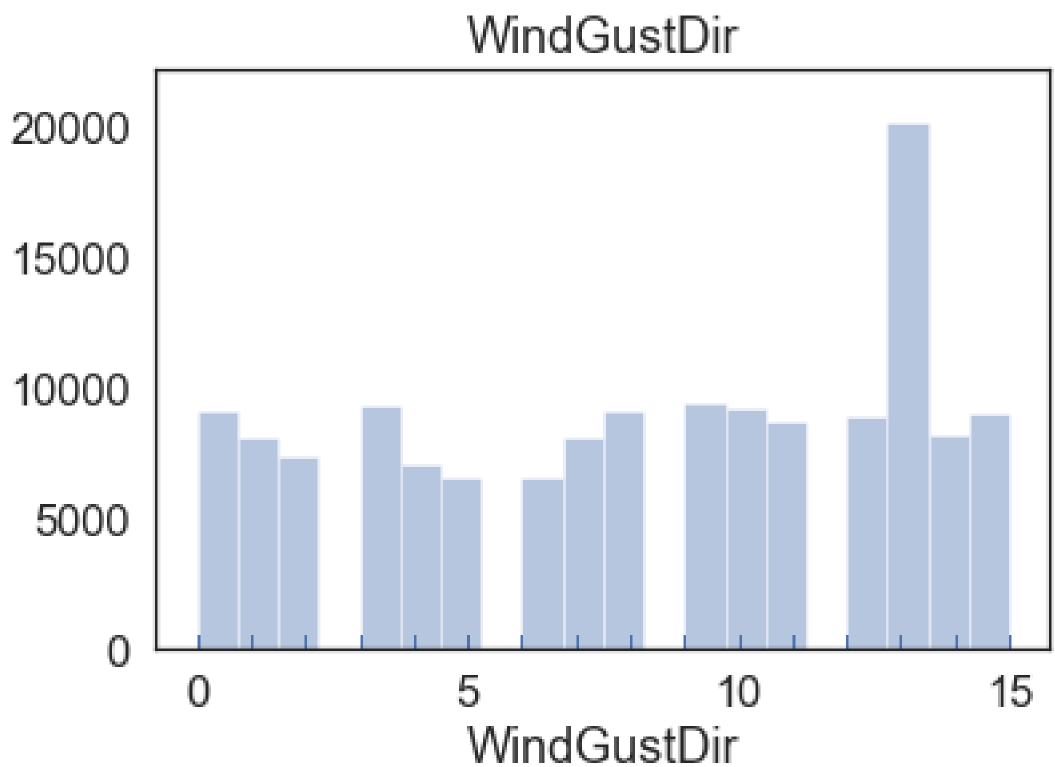
we can see some numerical variables are the normal distribution because in the Kolmogorov-Smirnov test, the P value is bigger than 0.05, these variables are, 'MinTemp', 'MaxTemp', 'Humidity9am', 'Humidity3pm', 'Temp9am', 'Temp3pm', for the variables that don't conform to the normal distribution, we can analyze them by histogram and scatter plot.

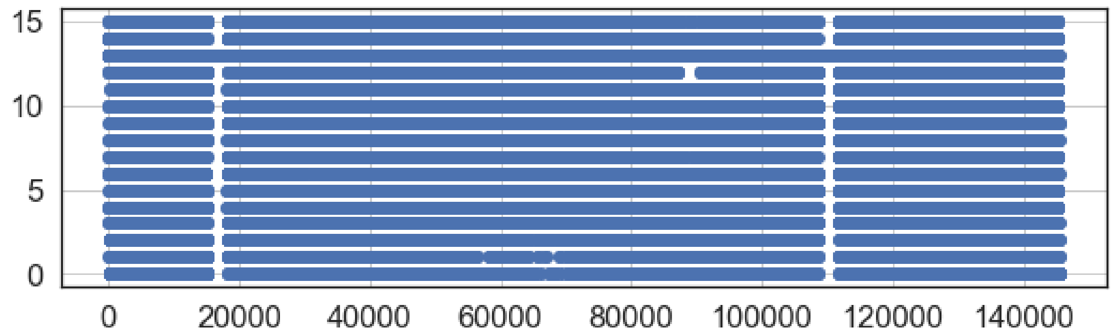
[19]:
```python
# The histogram and scatter plot of MaxTemp and MinTemp
sns.set(style="white",font_scale=1.5)
g = sns.jointplot(x='MinTemp', y='MaxTemp', data=dataset_important,
                  color='#098154',
                  marginal_kws=dict(bins=100,
                                    kde=True,
                                    color='#c72e29',
                                   ),
                  )
g.fig.set_size_inches(30,20)
```



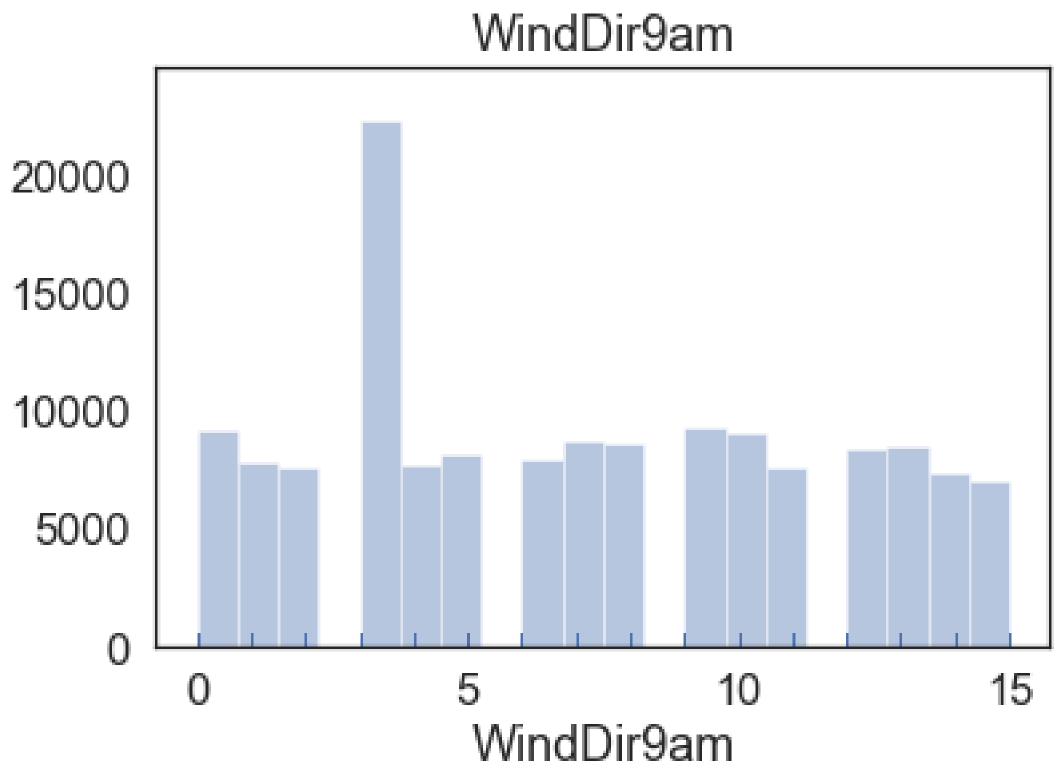

From the histogram and scatter plot of MaxTemp and MinTemp, we can see the nuclear density map of these two variables, and find that the similarity between the curve and the normal distribution curve. For the scatter, we can see some node are deviated from the right track, I think it might because when I deal with the original data, some data are lost, so, I replaced them by the means of the variables.

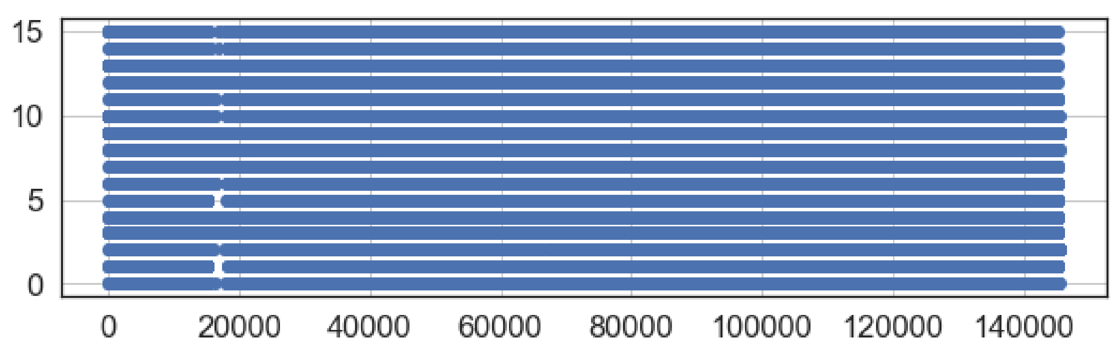
[20]:
```
WindGustDir = dataset_important['WindGustDir']
sns.distplot(WindGustDir, bins=20, hist=True, kde=False, norm_hist=False,␣
↪rug=True,
             vertical=False, axlabel=None, label=None, ax=None,
             fit=None)
plt.title('WindGustDir')
plt.show()

fig = plt.figure(figsize = (10,6))
ax1 = fig.add_subplot(2,1,1)
ax1.scatter(WindGustDir.index, WindGustDir.values)
plt.grid()
```





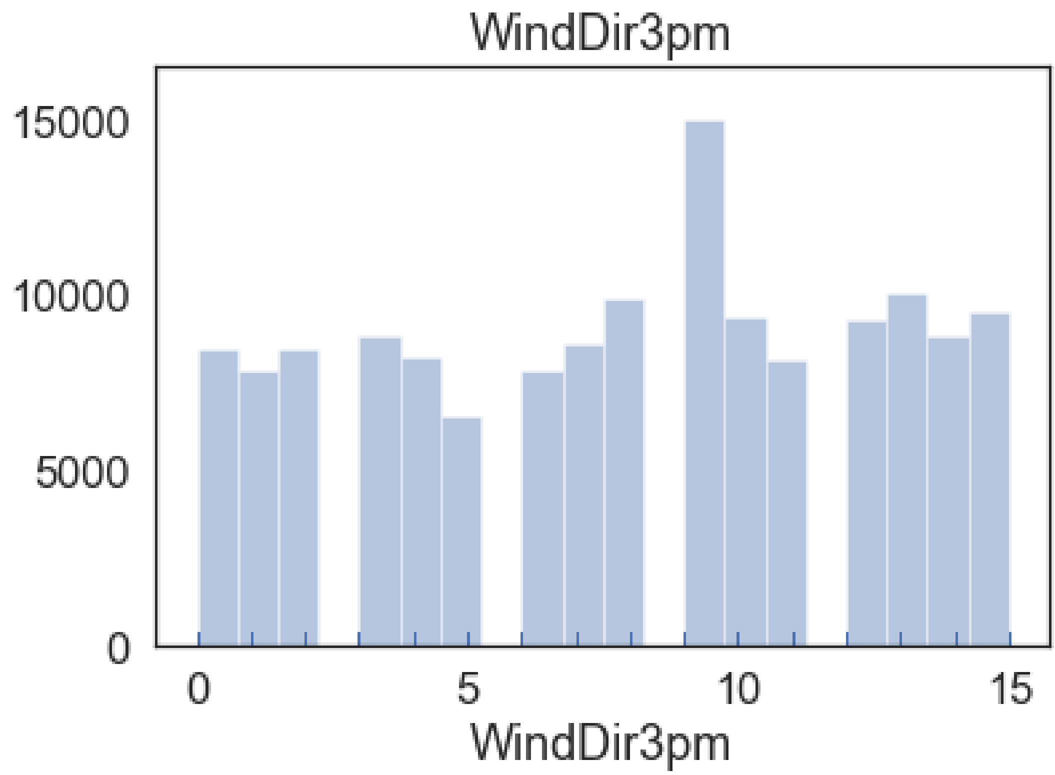

[21]:
```
WindDir9am = dataset_important['WindDir9am']
sns.distplot(WindDir9am, bins=20, hist=True, kde=False, norm_hist=False,␣
 ↪rug=True,
             vertical=False, axlabel=None, label=None, ax=None,
             fit=None)
plt.title('WindDir9am')
plt.show()

fig = plt.figure(figsize = (10,6))
ax1 = fig.add_subplot(2,1,1)
ax1.scatter(WindDir9am.index, WindDir9am.values)
plt.grid()
```

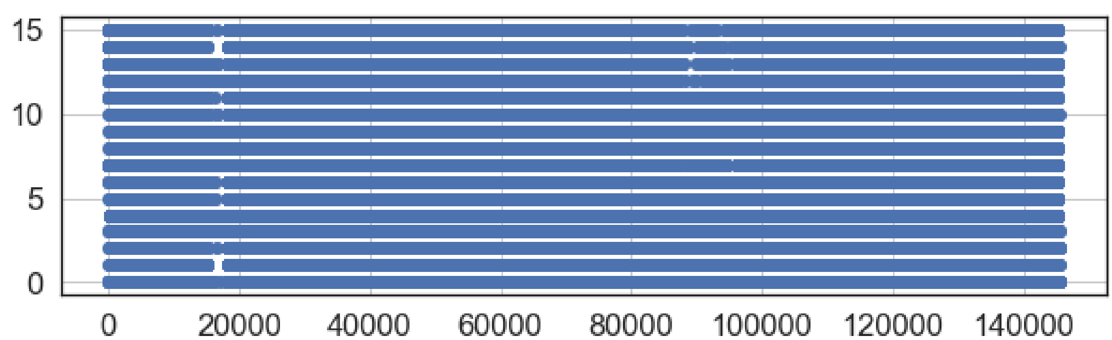



[22]:
```
WindDir3pm = dataset_important['WindDir3pm']
sns.distplot(WindDir3pm, bins=20, hist=True, kde=False, norm_hist=False,␣
 ↪rug=True,
             vertical=False, axlabel=None, label=None, ax=None,
             fit=None)
plt.title('WindDir3pm')
plt.show()

fig = plt.figure(figsize = (10,6))
```

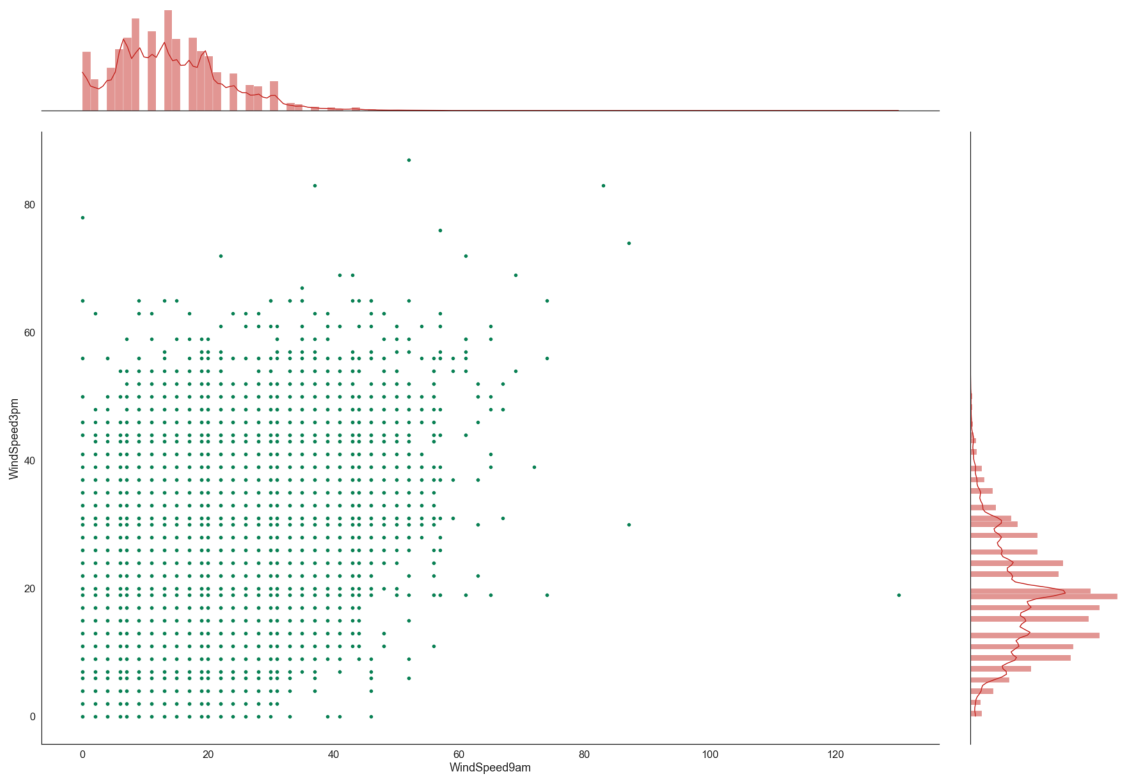
```
ax1 = fig.add_subplot(2,1,1)
ax1.scatter(WindDir3pm.index, WindDir3pm.values)
plt.grid()
```




Because the WindGustDir are categorical varibale, we can get an information is that, the wind direction have a high probability of pointing in a direction, this might have relation to the climate of Australia. And the WindDir9am, WindDir3pm is similiar with the WindGustDir.

[23]:
```
# The histogram and scatter plot of WindSpeed9am and WindSpeed3pm
sns.set(style="white",font_scale=1.5)
g = sns.jointplot(x='WindSpeed9am', y='WindSpeed3pm', data=dataset_important,
                  color='#098154',
                  marginal_kws=dict(bins=100,
                                    kde=True,
                                    color='#c72e29',
                                   ),
                  )
g.fig.set_size_inches(30,20)
```

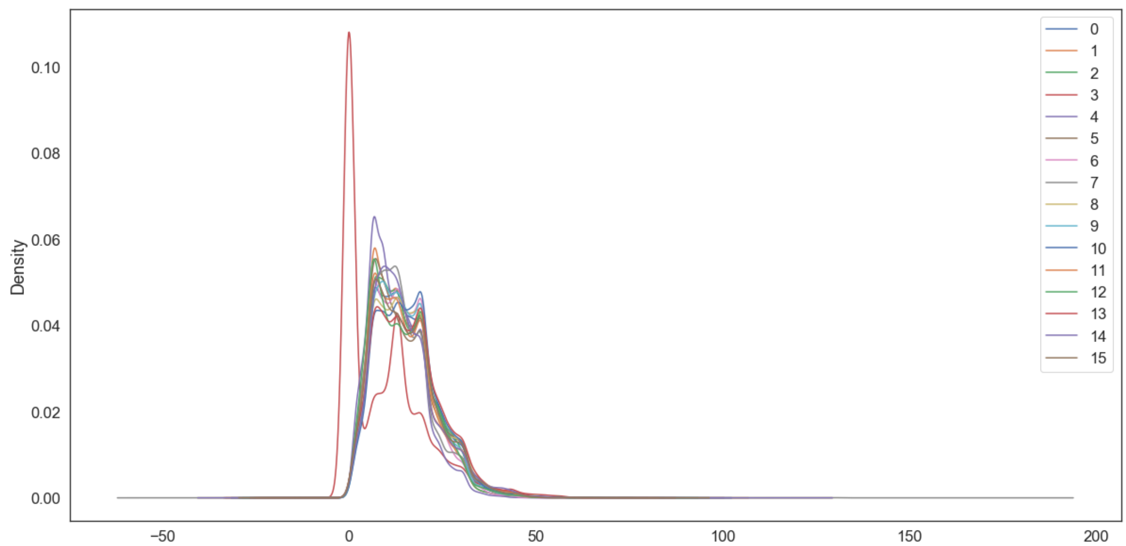



According to Kolmogorov-Smirnov test, we find that the WindSpeed9am and WindSpeed3pm are not the normal distribution, from the histogram and scatter plot above, we can find that the The distribution of points is uneven, and for the nuclear density map, it is also different from the normal distribution, I think because the Wind speed is unpredictable, even the changes in the same day are also very large, and different area the climate are very different, the speed of an area might conforms to the normal distribution

[24]:
```
dataset_important.groupby('WindDir9am')['WindSpeed9am'].plot(kind='kde',␣
↪legend=True, figsize=(20, 10))
```

[24]:
```
WindDir9am
0      AxesSubplot(0.125,0.125;0.775x0.755)
```

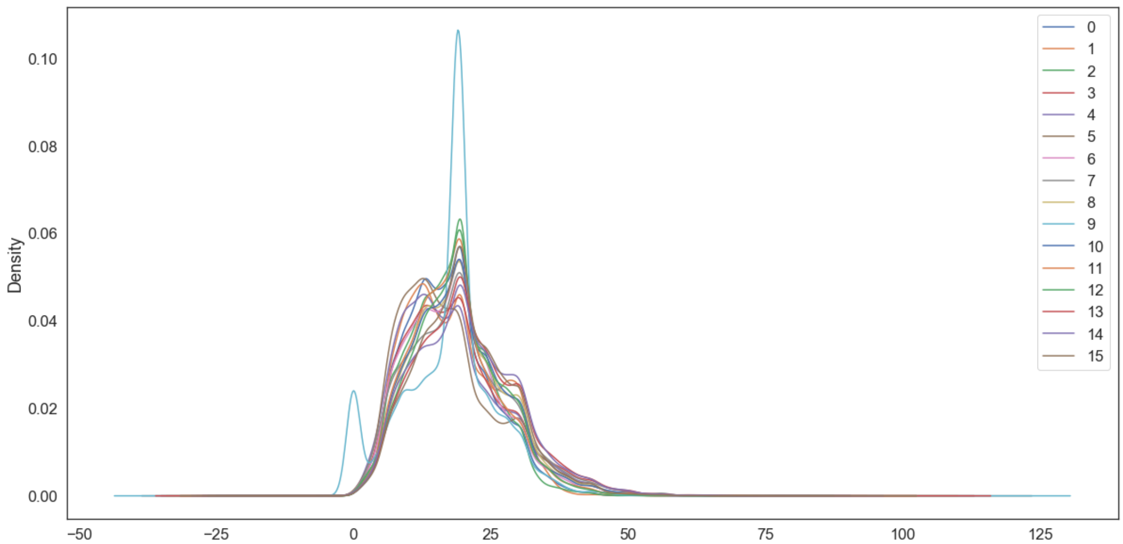
```
1      AxesSubplot(0.125,0.125;0.775x0.755)
2      AxesSubplot(0.125,0.125;0.775x0.755)
3      AxesSubplot(0.125,0.125;0.775x0.755)
4      AxesSubplot(0.125,0.125;0.775x0.755)
5      AxesSubplot(0.125,0.125;0.775x0.755)
6      AxesSubplot(0.125,0.125;0.775x0.755)
7      AxesSubplot(0.125,0.125;0.775x0.755)
8      AxesSubplot(0.125,0.125;0.775x0.755)
9      AxesSubplot(0.125,0.125;0.775x0.755)
10     AxesSubplot(0.125,0.125;0.775x0.755)
11     AxesSubplot(0.125,0.125;0.775x0.755)
12     AxesSubplot(0.125,0.125;0.775x0.755)
13     AxesSubplot(0.125,0.125;0.775x0.755)
14     AxesSubplot(0.125,0.125;0.775x0.755)
15     AxesSubplot(0.125,0.125;0.775x0.755)
Name: WindSpeed9am, dtype: object
```




[25]:
```python
dataset_important.groupby('WindDir3pm')['WindSpeed3pm'].plot(kind='kde',
    legend=True, figsize=(20, 10))
```

[25]:
```
WindDir3pm
0      AxesSubplot(0.125,0.125;0.775x0.755)
1      AxesSubplot(0.125,0.125;0.775x0.755)
2      AxesSubplot(0.125,0.125;0.775x0.755)
3      AxesSubplot(0.125,0.125;0.775x0.755)
4      AxesSubplot(0.125,0.125;0.775x0.755)
5      AxesSubplot(0.125,0.125;0.775x0.755)
6      AxesSubplot(0.125,0.125;0.775x0.755)
7      AxesSubplot(0.125,0.125;0.775x0.755)
```

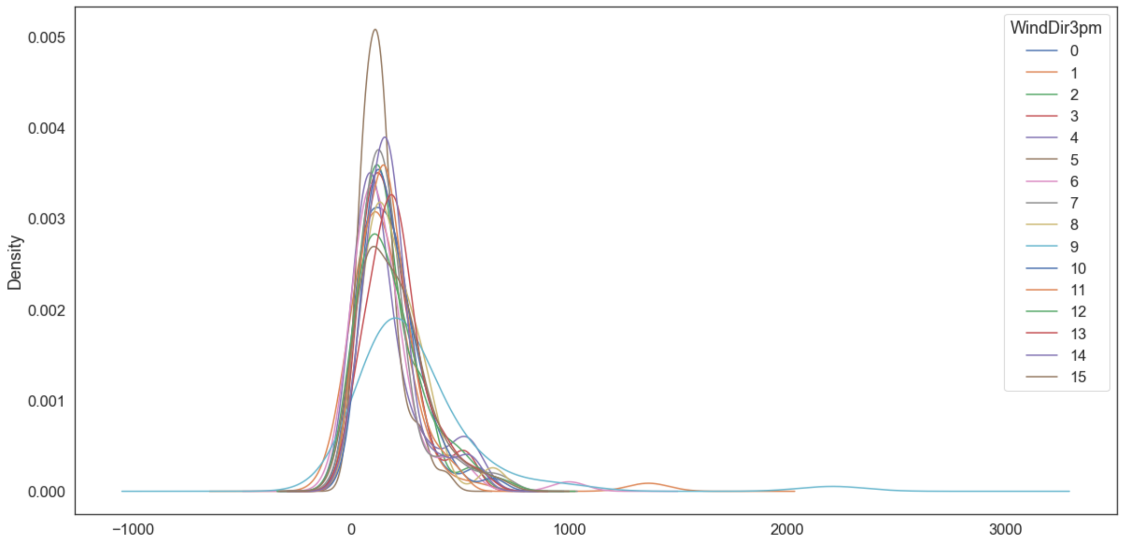
```
8      AxesSubplot(0.125,0.125;0.775x0.755)
9      AxesSubplot(0.125,0.125;0.775x0.755)
10     AxesSubplot(0.125,0.125;0.775x0.755)
11     AxesSubplot(0.125,0.125;0.775x0.755)
12     AxesSubplot(0.125,0.125;0.775x0.755)
13     AxesSubplot(0.125,0.125;0.775x0.755)
14     AxesSubplot(0.125,0.125;0.775x0.755)
15     AxesSubplot(0.125,0.125;0.775x0.755)
Name: WindSpeed3pm, dtype: object
```




We group the Wind Speed by the Wind Dirction and find that the distribution of the windSpeed still complex, but we can see the curves are more similiar to the normal distribution than the curve without grouping. So, I think the distribution of windSpeed is Mixed Gaussian distribution, but the parameters are currently unknown

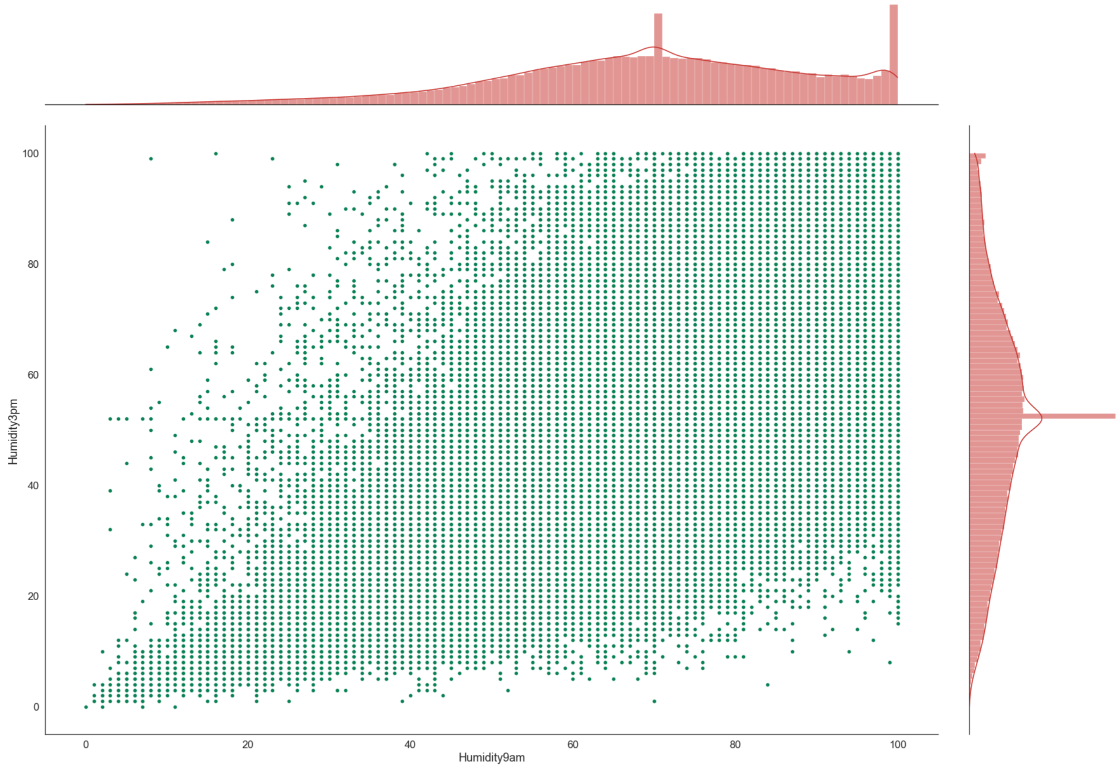
[55]:
```
group =  dataset_important.groupby(['Location', 'WindDir3pm']).count()
(group['WindSpeed3pm'].unstack()).plot(kind='kde', legend=True, figsize=(20,␣
 ↪10))
# ['WindSpeed3pm'].plot(kind='kde', legend=True, figsize=(20, 10))
```

[55]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1995f96ca08>
```



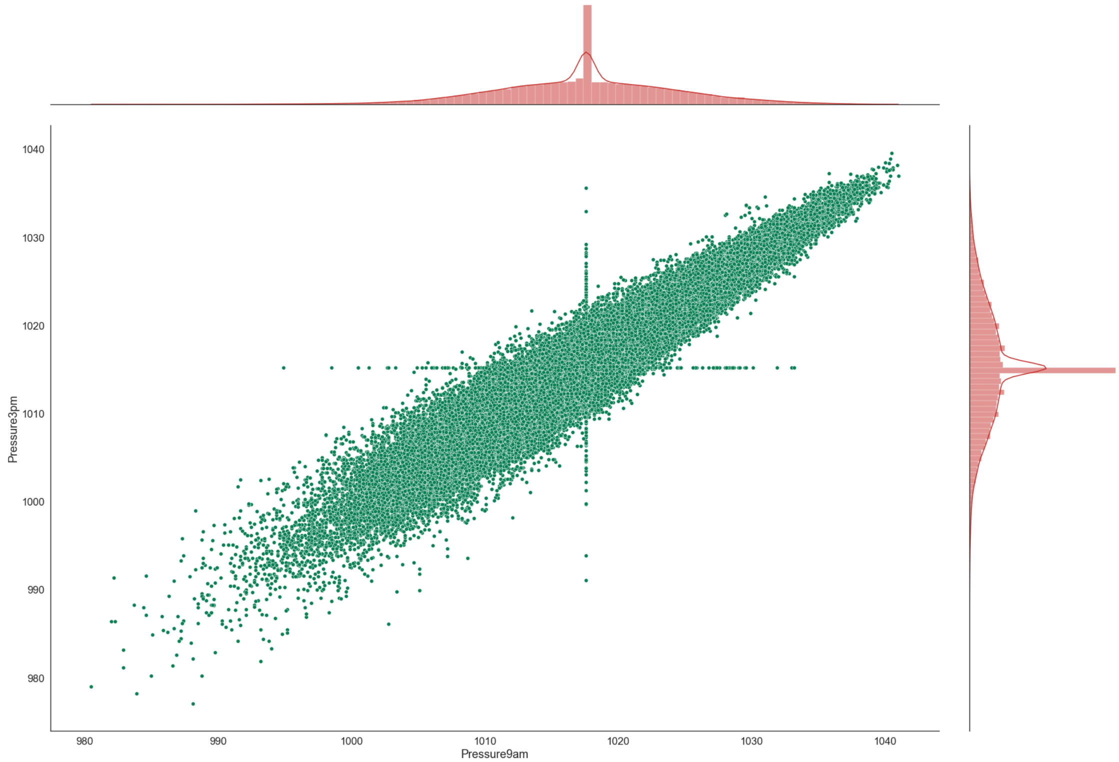

we can see, when we group the Wind speed with Wind Direction and Location, the curves are more similiar with the normal distribution, therefore, windSpeed is Mixed Gaussian distribution.

[26]:
```python
# The histogram and scatter plot of Humidity9am and Humidity3pm
sns.set(style="white",font_scale=1.5)
g = sns.jointplot(x='Humidity9am', y='Humidity3pm', data=dataset_important,
                  color='#098154',
                  marginal_kws=dict(bins=100,
                                    kde=True,
                                    color='#c72e29',
                                   ),
                  )
g.fig.set_size_inches(30,20)
```




We can't find some special relation from the scatter plots, but from the histogram plots, we find the curves are very similiar with the normal distribution, Kolmogorov-Smirnov test also show that the Humidity is normal distribution.

[27]:
```
# The histogram and scatter plot of Pressure9am and Pressure3pm
sns.set(style="white",font_scale=1.5)
g = sns.jointplot(x='Pressure9am', y='Pressure3pm', data=dataset_important,
                  color='#098154',
                  marginal_kws=dict(bins=100,
                                    kde=True,
                                    color='#c72e29',
                                   ),
                  )
g.fig.set_size_inches(30,20)
```




```
[28]: print(stats.normaltest(dataset_important['Pressure9am']))
      print(stats.normaltest(dataset_important['Pressure3pm']))
```

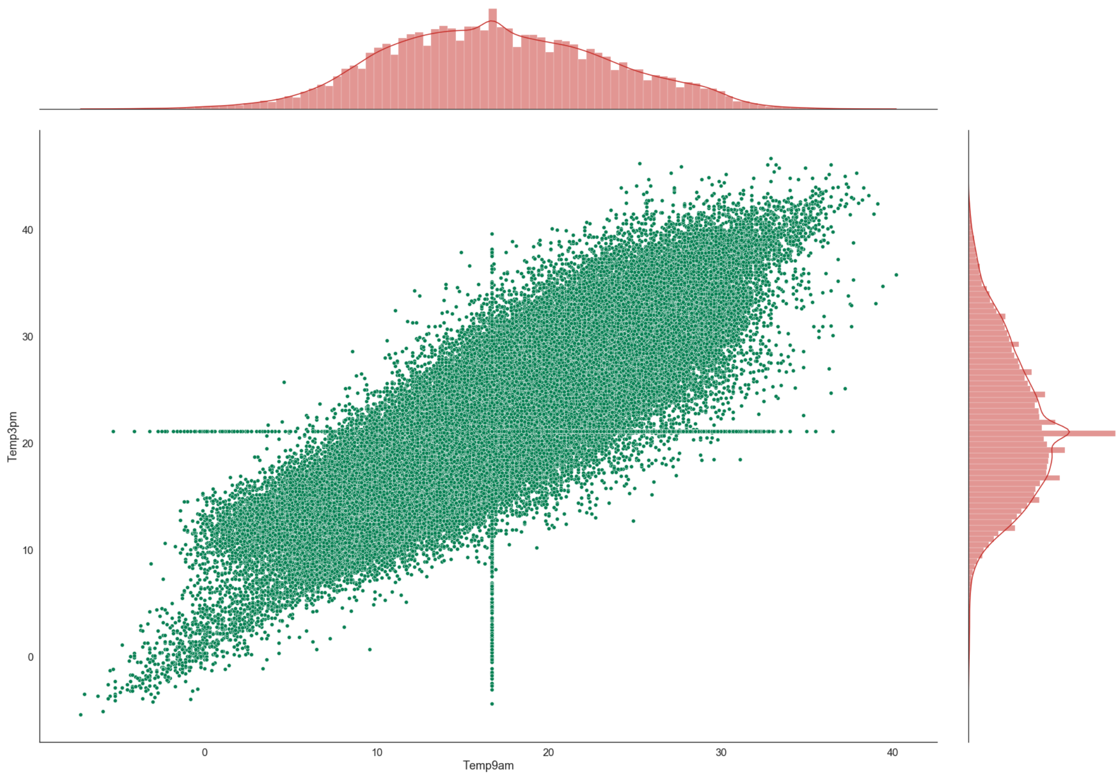
```
NormaltestResult(statistic=1567.000624115162, pvalue=0.0)
NormaltestResult(statistic=1001.1234012079072, pvalue=4.0627076847467417e-218)
```

1. From the scatter plot, we cloud find that the Pressure9am and Pressure3pm are linearly dependent, and some point are noise of the data, I think these noise might come from some original data are lost, so, I replace them with the means, and the lost data is little.

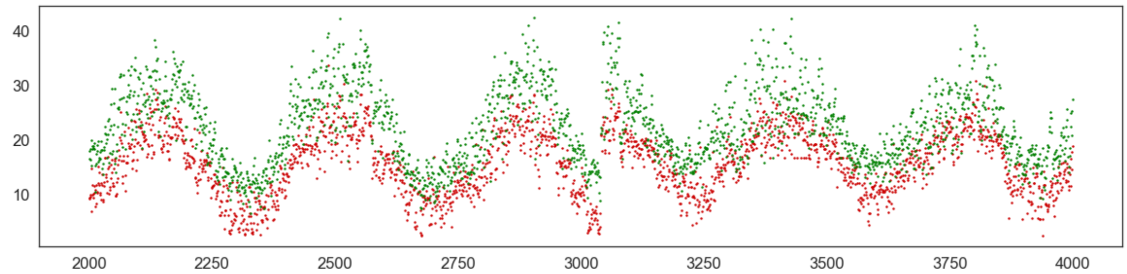
2. Kolmogorov-Smirnov test also show that the Pressure9am and Pressure3pm are not the normal distribution, but from the histogram plots, we find the curves are similiar with the normal distribution, so, we add a Normaltest, and find that Pressure3pm is normal distribution, but Pressure9am isn't, I think the Pressure9am is also normal distribution, but the noise have a lot influence to it, so, the P-value is too small.

```
[29]: # The histogram and scatter plot of Temp9am and Temp3pm
      sns.set(style="white",font_scale=1.5)
      g = sns.jointplot(x='Temp9am', y='Temp3pm', data=dataset_important,
                        color='#098154',
                        marginal_kws=dict(bins=100,
                                          kde=True,
                                          color='#c72e29',
```

```
                                    ),
                  )
g.fig.set_size_inches(30,20)
```



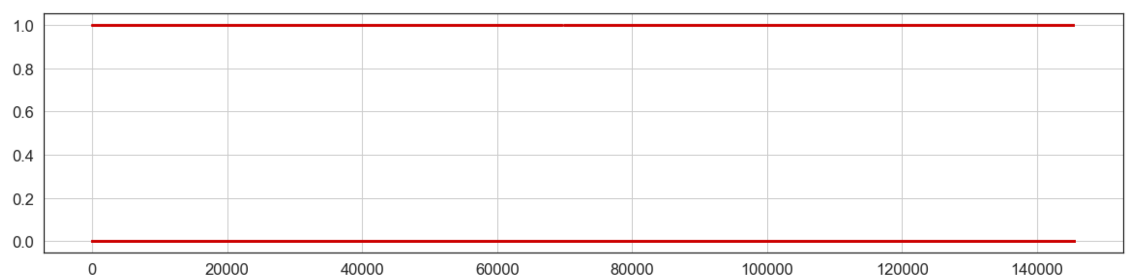

[30]:
```
Temp9am = dataset_important['Temp9am']
Temp3pm = dataset_important['Temp3pm']
Temp9am = Temp9am[2000:4000]
Temp3pm = Temp3pm[2000:4000]
fig = plt.figure(figsize = (20,10))
ax1 = fig.add_subplot(2,1,1)
ax1.scatter(Temp9am.index, Temp9am.values, s =2,color=(0.8,0.,0.) )
plt.grid()

ax1 = fig.add_subplot(2,1,1)
ax1.scatter(Temp3pm.index, Temp3pm.values, s= 2, color=(0.,0.5,0.))
plt.grid()
```



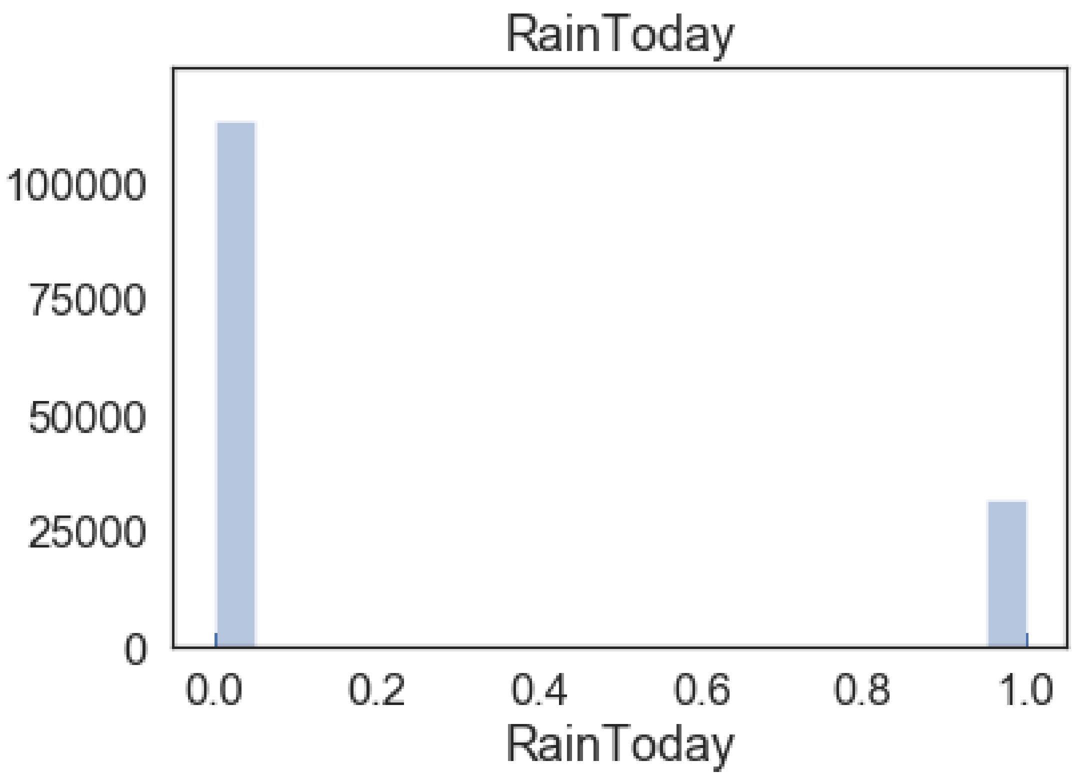

1. Kolmogorov-Smirnov test show that the Temp9am and Temp3pm are the normal distribution, we can also get it from the histogram and scatter plot, we can also find that Temp3pm and Temp9am are ilinearly dependent.

2. For the scatter plot, some points also deviate from the line, but we then add a scatter that show some points in the data, and find that these wrong points are randomly distributed, these points come from the noise of the data.

[31]:
```
RainToday = dataset_important['RainToday']
fig = plt.figure(figsize = (20,10))
ax1 = fig.add_subplot(2,1,1)
ax1.scatter(RainToday.index, RainToday.values, s =2,color=(0.8,0.,0.) )
plt.grid()
```




[32]:
```
sns.distplot(RainToday, bins=20, hist=True, kde=False, norm_hist=False,␣
 ↪rug=True,
             vertical=False, axlabel=None, label=None, ax=None,
             fit=None)
plt.title('RainToday')
plt.show()
```




What we can see is that the sunny day are more than the rainy day in Australia.

## 0.6 Conclusion

We analyze the data of daily weather observations from many locations across Australia. And find that some important variables, such as locayion, Temperature, Humidity, and find that also all the numerical variables are normal distribution, even some complex variables are Mixed Gaussian distribution, for some Weird point in the scatter plots, they are the result of the noise in the data, and have little influence on data

[ ]:

```

```