Initial data exploration

Load

```
import pandas

data = pandas.read_csv('../data/raw/weatherAUS.csv')
```

Data shape and head

Response variable to predict: RainTomorrow (Yes, No)

```
print('Dimensions of data: {0}'.format(data.shape))
data.head(n=10)
```

```
Dimensions of data: (145460, 23)
```

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	\
0	2008-12-01	Albury	13.4	22.9	0.6	NaN	NaN	
1	2008-12-02	Albury	7.4	25.1	0.0	NaN	NaN	
2	2008-12-03	Albury	12.9	25.7	0.0	NaN	NaN	
3	2008-12-04	Albury	9.2	28.0	0.0	NaN	NaN	
4	2008-12-05	Albury	17.5	32.3	1.0	NaN	NaN	
5	2008-12-06	Albury	14.6	29.7	0.2	NaN	NaN	
6	2008-12-07	Albury	14.3	25.0	0.0	NaN	NaN	
7	2008-12-08	Albury	7.7	26.7	0.0	NaN	NaN	
8	2008-12-09	Albury	9.7	31.9	0.0	NaN	NaN	
9	2008-12-10	Albury	13.1	30.1	1.4	NaN	NaN	

	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am	\
0	W	44.0	W	...	71.0	
1	WNW	44.0	NNW	...	44.0	
2	WSW	46.0	W	...	38.0	
3	NE	24.0	SE	...	45.0	
4	W	41.0	ENE	...	82.0	
5	WNW	56.0	W	...	55.0	
6	W	50.0	SW	...	49.0	
7	W	35.0	SSE	...	48.0	

```
8          NNW           80.0          SE      ...              42.0
9            W           28.0           S      ...              58.0

   Humidity3pm  Pressure9am  Pressure3pm  Cloud9am  Cloud3pm  Temp9am
\
0         22.0       1007.7       1007.1       8.0       NaN     16.9

1         25.0       1010.6       1007.8       NaN       NaN     17.2

2         30.0       1007.6       1008.7       NaN       2.0     21.0

3         16.0       1017.6       1012.8       NaN       NaN     18.1

4         33.0       1010.8       1006.0       7.0       8.0     17.8

5         23.0       1009.2       1005.4       NaN       NaN     20.6

6         19.0       1009.6       1008.2       1.0       NaN     18.1

7         19.0       1013.4       1010.1       NaN       NaN     16.3

8          9.0       1008.9       1003.6       NaN       NaN     18.3

9         27.0       1007.0       1005.7       NaN       NaN     20.1


   Temp3pm  RainToday  RainTomorrow
0     21.8         No            No
1     24.3         No            No
2     23.2         No            No
3     26.5         No            No
4     29.7         No            No
5     28.9         No            No
6     24.6         No            No
7     25.5         No            No
8     30.2         No           Yes
9     28.2        Yes            No

[10 rows x 23 columns]
```

Data summaries

Many rows have missing data, and it's not clear whether they are missing at random or not at random. These will need to be handled. Sunshine, Evaporation, Cloud9am, and Cloud3pm have a lot of missing data.

```
print('Total number of rows: {0}'.format(data.shape[0]))
print('Rows if rows with missing data were dropped:
```

```
{0}'.format(len(data.dropna())))
data.describe()
```

```
Total number of rows: 145460
Rows if rows with missing data were dropped: 56420

             MinTemp        MaxTemp       Rainfall   Evaporation  \
count  143975.000000  144199.000000  142199.000000  82670.000000
mean       12.194034      23.221348       2.360918      5.468232
std         6.398495       7.119049       8.478060      4.193704
min        -8.500000      -4.800000       0.000000      0.000000
25%         7.600000      17.900000       0.000000      2.600000
50%        12.000000      22.600000       0.000000      4.800000
75%        16.900000      28.200000       0.800000      7.400000
max        33.900000      48.100000     371.000000    145.000000

           Sunshine  WindGustSpeed   WindSpeed9am   WindSpeed3pm  \
count  75625.000000  135197.000000  143693.000000  142398.000000
mean       7.611178      40.035230      14.043426      18.662657
std        3.785483      13.607062       8.915375       8.809800
min        0.000000       6.000000       0.000000       0.000000
25%        4.800000      31.000000       7.000000      13.000000
50%        8.400000      39.000000      13.000000      19.000000
75%       10.600000      48.000000      19.000000      24.000000
max       14.500000     135.000000     130.000000      87.000000

         Humidity9am    Humidity3pm   Pressure9am    Pressure3pm  \
count  142806.000000  140953.000000  130395.00000  130432.000000
mean       68.880831      51.539116    1017.64994    1015.255889
std        19.029164      20.795902       7.10653       7.037414
min         0.000000       0.000000     980.50000     977.100000
25%        57.000000      37.000000    1012.90000    1010.400000
50%        70.000000      52.000000    1017.60000    1015.200000
75%        83.000000      66.000000    1022.40000    1020.000000
max       100.000000     100.000000    1041.00000    1039.600000

           Cloud9am      Cloud3pm        Temp9am       Temp3pm
count  89572.000000  86102.000000  143693.000000  141851.00000
mean       4.447461      4.509930      16.990631      21.68339
std        2.887159      2.720357       6.488753       6.93665
min        0.000000      0.000000      -7.200000      -5.40000
25%        1.000000      2.000000      12.300000      16.60000
50%        5.000000      5.000000      16.700000      21.10000
75%        7.000000      7.000000      21.600000      26.40000
max        9.000000      9.000000      40.200000      46.70000
```

Categorical variables:

```
data.describe(include=['O'])
```

```
              Date  Location WindGustDir WindDir9am WindDir3pm
RainToday  \
count       145460    145460      135134     134894     141232
142199
unique        3436        49          16         16         16
2
top     2015-06-18  Canberra           W          N         SE
No
freq            49      3436        9915      11758      10838
110319

       RainTomorrow
count        142193
unique            2
top              No
freq         110316
```

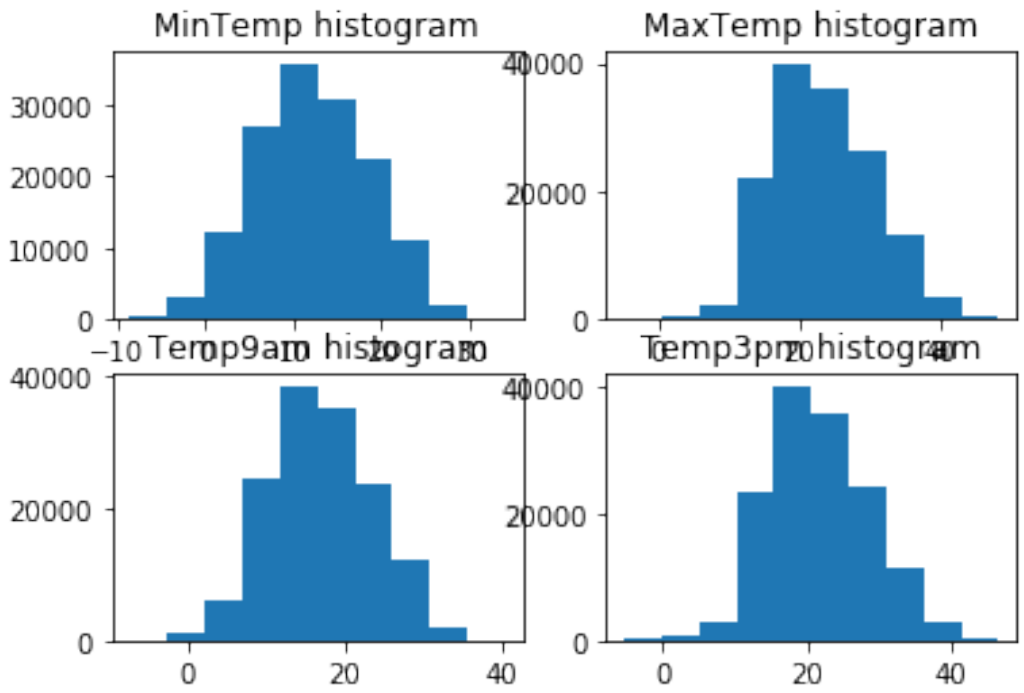
Some plots

```
import matplotlib.pyplot as pyplot

%matplotlib inline
```

Temperature

```
fig, axs = pyplot.subplots(2,2)

axs[0,0].set_title('MinTemp histogram')
axs[0,0].hist(data['MinTemp'])

axs[0,1].set_title('MaxTemp histogram')
axs[0,1].hist(data['MaxTemp'])

axs[1,0].set_title('Temp9am histogram')
axs[1,0].hist(data['Temp9am'])

axs[1,1].set_title('Temp3pm histogram')
axs[1,1].hist(data['Temp3pm'])
```

```
(array([  172.,   634.,  3112., 23341., 40168., 35761., 24036.,
11314.,
         3030.,   283.]),
 array([-5.4 , -0.19,  5.02, 10.23, 15.44, 20.65, 25.86, 31.07, 36.28,
       41.49, 46.7 ]),
 <a list of 10 Patch objects>)
```

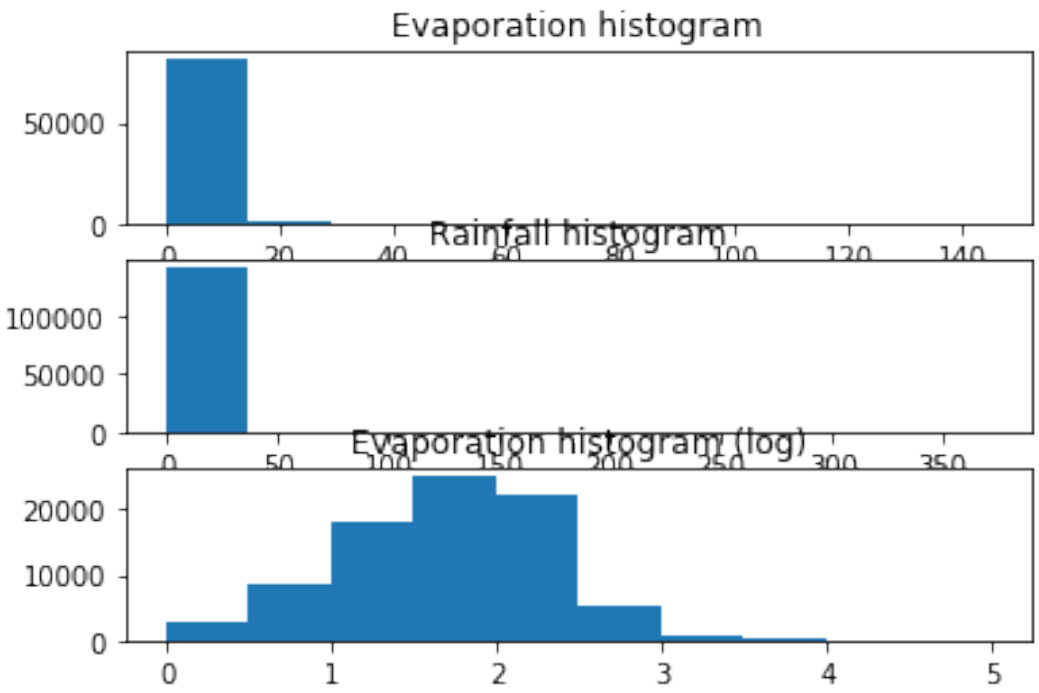


Rainfall, Evaporation

The data for these features are very skewed. Can apply a log transformation on Evaporation to make it less skewed. It may be more suitable to use RainToday in place of Rainfall.

```
import numpy as np

def log_transformation(data):
    return data.apply(np.log1p)

fig, axs = pyplot.subplots(3,1)

axs[0].set_title('Evaporation histogram')
axs[0].hist(data['Evaporation'])

axs[1].set_title('Rainfall histogram')
axs[1].hist(data['Rainfall'])

axs[2].set_title('Evaporation histogram (log)')
axs[2].hist(log_transformation(data['Evaporation']))
```

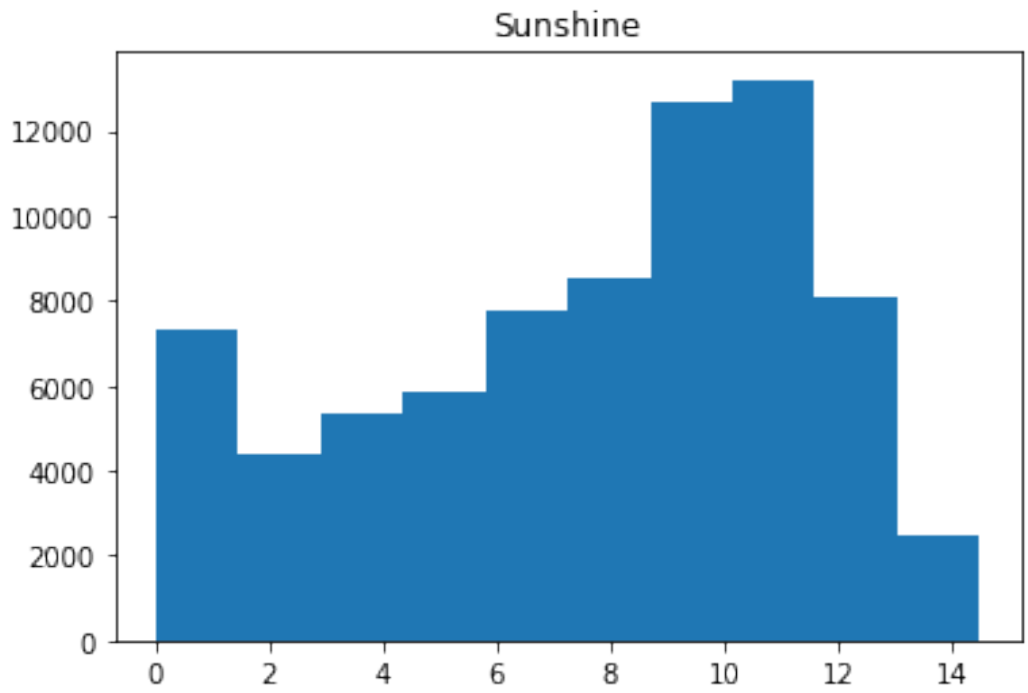
```
(array([2.6450e+03, 8.4680e+03, 1.8182e+04, 2.5154e+04, 2.2130e+04,
        5.3470e+03, 5.5600e+02, 1.4200e+02, 4.5000e+01, 1.0000e+00]),
 array([0.        , 0.49836066, 0.99672132, 1.49508199, 1.99344265,
        2.49180331, 2.99016397, 3.48852464, 3.9868853 , 4.48524596,
        4.98360662]),
 <a list of 10 Patch objects>)
```



Sunshine

```
fig, axs = pyplot.subplots(1,1)

axs.set_title('Sunshine')
axs.hist(data['Sunshine'])
```

```
(array([ 7297.,  4373.,  5314.,  5879.,  7737.,  8548., 12685.,
13206.,
         8093.,  2493.]),
 array([ 0.  ,  1.45,  2.9 ,  4.35,  5.8 ,  7.25,  8.7 , 10.15, 11.6 ,
       13.05, 14.5 ]),
 <a list of 10 Patch objects>)
```

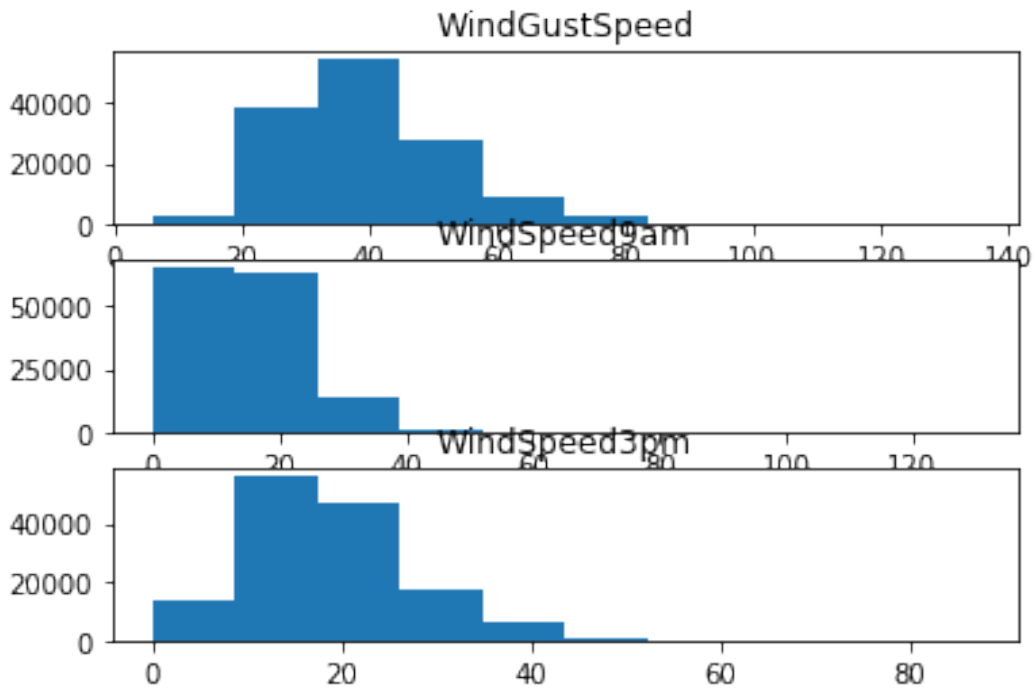


Wind

```
# WindGustSpeed  WindSpeed9am     WindSpeed3pm
```

```
fig, axs = pyplot.subplots(3,1)

axs[0].set_title('WindGustSpeed')
axs[0].hist(data['WindGustSpeed'])

axs[1].set_title('WindSpeed9am')
axs[1].hist(data['WindSpeed9am'])

axs[2].set_title('WindSpeed3pm')
axs[2].hist(data['WindSpeed3pm'])
```

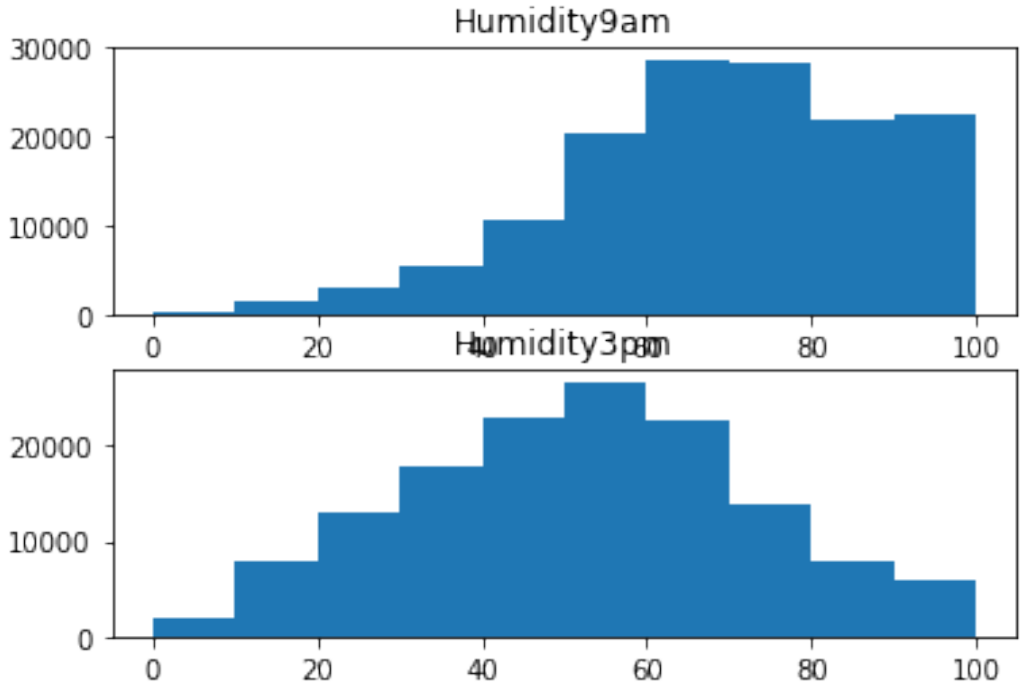
```
(array([1.4103e+04, 5.6370e+04, 4.7134e+04, 1.7637e+04, 5.8450e+03,
        1.0870e+03, 1.6000e+02, 5.3000e+01, 6.0000e+00, 3.0000e+00]),
 array([ 0. ,  8.7, 17.4, 26.1, 34.8, 43.5, 52.2, 60.9, 69.6, 78.3,
87. ]),
 <a list of 10 Patch objects>)
```



Humidity

```
# Humidity9am    Humidity3pm

fig, axs = pyplot.subplots(2,1)

axs[0].set_title('Humidity9am')
axs[0].hist(data['Humidity9am'])

axs[1].set_title('Humidity3pm')
axs[1].hist(data['Humidity3pm'])
```

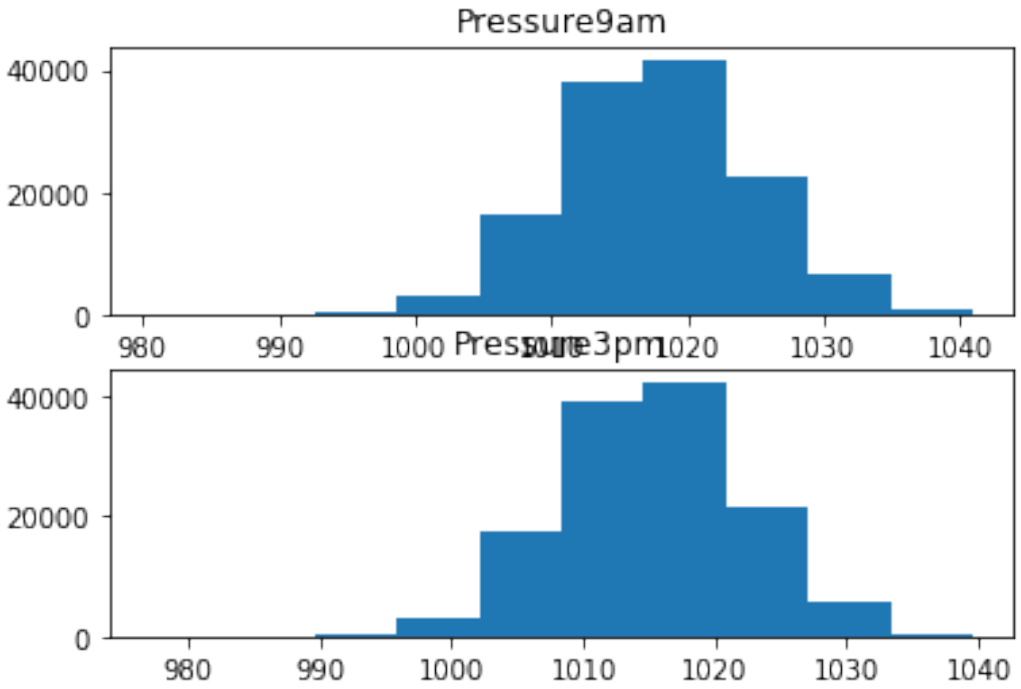
```
(array([ 1846.,  8040., 13030., 17975., 23028., 26748., 22752.,
13931.,
         7800.,  5803.]),
 array([  0.,  10.,  20.,  30.,  40.,  50.,  60.,  70.,  80.,  90.,
100.]),
 <a list of 10 Patch objects>)
```



Pressure

```
# Pressure9am    Pressure3pm

fig, axs = pyplot.subplots(2,1)

axs[0].set_title('Pressure9am')
axs[0].hist(data['Pressure9am'])

axs[1].set_title('Pressure3pm')
axs[1].hist(data['Pressure3pm'])
```

```
(array([1.3000e+01, 8.4000e+01, 5.1200e+02, 2.9200e+03, 1.7509e+04,
        3.9194e+04, 4.2330e+04, 2.1521e+04, 5.7970e+03, 5.5200e+02]),
 array([ 977.1 ,  983.35,  989.6 ,  995.85, 1002.1 , 1008.35, 1014.6 ,
        1020.85, 1027.1 , 1033.35, 1039.6 ]),
 <a list of 10 Patch objects>)
```

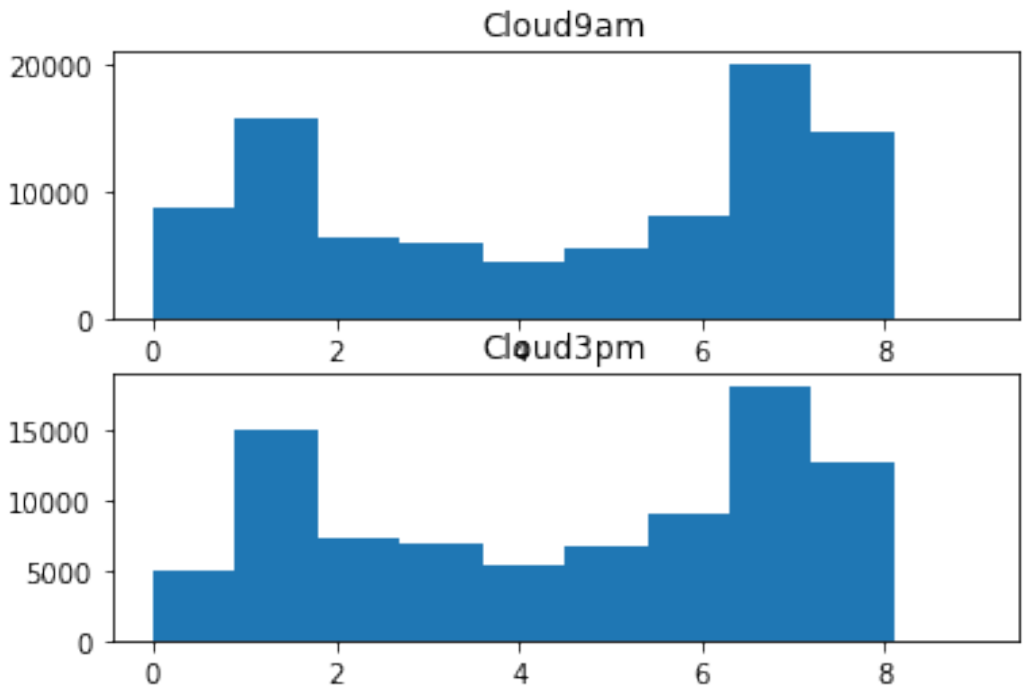


Clouds

Cloud9am and Cloud3pm appear to be bimodal.

```
# Cloud9am Cloud3pm

fig, axs = pyplot.subplots(2,1)

axs[0].set_title('Cloud9am')
axs[0].hist(data['Cloud9am'])

axs[1].set_title('Cloud3pm')
axs[1].hist(data['Cloud3pm'])
```

```
(array([4.9740e+03, 1.4976e+04, 7.2260e+03, 6.9210e+03, 5.3220e+03,
        6.8150e+03, 8.9780e+03, 1.8229e+04, 1.2660e+04, 1.0000e+00]),
 array([0. , 0.9, 1.8, 2.7, 3.6, 4.5, 5.4, 6.3, 7.2, 8.1, 9. ]),
 <a list of 10 Patch objects>)
```


Cloud9am
20000
10000
0
0
2
6
8
Cloud3pm
15000
10000
5000
0
0
2
4
6
8