Final Capstone - Harvard and Duke Forests environmental growing conditions

Overview:

This project is a statistical analysis of Harvard and Duke forests environmental conditions. These two forests are approximately 700 miles from each other on the eastern seaboard. This dataset is of interest due to the rapidly changing climatic conditions, digging into the experimental environmenal conditions can provide key insights into future outcomes of these two forest. This data and analysis is very important for US Forest Service and Conservation agencies to understand as they asses and make predictions about the future of our forests.

Goals:

1. Combine 3 datasets into 1
2. Predict difference between Duke and Harvard forests
3. Model the relationship between air temperature and PAR
4. Use classification models to predict treatment changes
5. Use Random Forest to predict Tree Species with treatments ('Only for Harvard forest')
6. PCA model of the dataset - how is it related or different
7. Time series modeling of Air temperature and photosysnthetically active radiation

Loading Modules

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
import seaborn as sns
sns.set(style="ticks", color_codes=True)

import chardet
import codecs

from sklearn.preprocessing import scale
import sklearn.linear_model as skl_lm
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
import statsmodels.formula.api as smf

import folium
from folium import plugins
from scipy import stats

%matplotlib inline
plt.style.use('seaborn-white')

import warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', None)
```

Loading in datasets for Harvard and Duke forest

The havard_forest_complete.csv file is a joined file of hf199-01: hf environment and hf199-03: hf growth. To see the code on the merge see: Harvard Forest and Plant ID merged.ipynb

In [4]:
```
# Harvard Forest
df = pd.read_csv('/Users/mille/Desktop/Final Capstone/havard_forest_complete.c
sv', low_memory=False)
print('Dataframe dimensions:', df.shape)
```

```
Dataframe dimensions: (1407177, 18)
```

In [5]:
```
# Duke Forest
df1 = pd.read_csv('/Users/mille/Desktop/Final Capstone/hf199-04-df-env.csv', l
ow_memory=False)
print('Dataframe dimensions:', df1.shape)
```

```
Dataframe dimensions: (750768, 15)
```

In [6]:
```
df.head()
```

Out[6]:

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	trea
0	0	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	1	1	S
1	1	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	2	1	S
2	2	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	3	1	S
3	3	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	4	1	S
4	4	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	5	1	S

In [7]:
```
df.drop(columns=['Unnamed: 0'])
```

Out[7]:

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
0	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	1	1	S
1	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	2	1	S
2	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	3	1	S
3	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	4	1	S
4	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	5	1	S
5	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	6	1	S
6	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	7	1	S
7	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	8	1	S
8	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	9	1	S
9	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	10	1	S
10	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	11	1	S
11	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	12	1	S
12	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	13	1	S
13	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	14	1	S
14	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	15	1	S
15	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	16	1	S
16	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	17	1	S
17	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	18	1	S
18	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	19	1	S

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
19	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	20	1	S
20	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	21	1	S
21	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	22	1	S
22	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	23	1	S
23	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	24	1	S
24	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	1	1	S
25	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	2	1	S
26	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	3	1	S
27	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	4	1	S
28	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	5	1	S
29	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	6	1	S
...	...	...	...	...	...	...	...	...	...	...	...
1407147	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	19	11	S
1407148	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	20	11	S
1407149	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	21	11	S
1407150	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	22	11	S
1407151	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	23	11	S
1407152	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	24	11	S
1407153	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	1	11	S

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
1407154	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	2	11	S
1407155	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	3	11	S
1407156	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	4	11	S
1407157	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	5	11	S
1407158	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	6	11	S
1407159	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	7	11	S
1407160	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	8	11	S
1407161	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	9	11	S
1407162	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	10	11	S
1407163	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	11	11	S
1407164	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	12	11	S
1407165	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	13	11	S
1407166	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	14	11	S
1407167	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	15	11	S
1407168	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	16	11	S
1407169	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	17	11	S
1407170	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	18	11	S
1407171	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	19	11	S
1407172	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	20	11	S

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
1407173	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	21	11	S
1407174	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	22	11	S
1407175	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	23	11	S
1407176	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	24	11	S

1407177 rows × 17 columns

Adding datetime to Harvard Forest dataset

In [8]:
```
df['date']  = pd.to_datetime(df[['year', 'month', 'day']])
```

In [9]:
```
df.drop(columns=['Unnamed: 0'])
```

Out[9]:

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
0	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	1	1	S
1	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	2	1	S
2	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	3	1	S
3	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	4	1	S
4	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	5	1	S
5	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	6	1	S
6	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	7	1	S
7	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	8	1	S
8	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	9	1	S
9	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	10	1	S
10	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	11	1	S
11	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	12	1	S
12	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	13	1	S
13	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	14	1	S
14	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	15	1	S
15	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	16	1	S
16	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	17	1	S
17	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	18	1	S
18	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	19	1	S

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
19	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	20	1	S
20	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	21	1	S
21	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	22	1	S
22	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	23	1	S
23	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	24	1	S
24	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	1	1	S
25	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	2	1	S
26	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	3	1	S
27	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	4	1	S
28	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	5	1	S
29	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	6	1	S
...	...	...	...	...	...	...	...	...	...	...	...
1407147	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	19	11	S
1407148	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	20	11	S
1407149	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	21	11	S
1407150	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	22	11	S
1407151	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	23	11	S
1407152	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	24	11	S
1407153	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	1	11	S

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
1407154	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	2	11	S
1407155	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	3	11	S
1407156	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	4	11	S
1407157	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	5	11	S
1407158	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	6	11	S
1407159	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	7	11	S
1407160	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	8	11	S
1407161	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	9	11	S
1407162	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	10	11	S
1407163	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	11	11	S
1407164	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	12	11	S
1407165	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	13	11	S
1407166	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	14	11	S
1407167	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	15	11	S
1407168	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	16	11	S
1407169	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	17	11	S
1407170	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	18	11	S
1407171	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	19	11	S
1407172	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	20	11	S

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
1407173	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	21	11	S
1407174	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	22	11	S
1407175	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	23	11	S
1407176	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	24	11	S

1407177 rows × 17 columns

Understanding the Harvad Forest - environmental variables

year: year month: month day: day of month time: hour of day chamber: chamber number (1-12) treatment: light treatment G: chamber in open gap S: chamber under closed canopy warming: warming treatment 3: 3 degrees C 5: 5 degrees C A: ambient C: control AT: air temperature (unit: celsius / missing value: NA) Q: photosynthetically active radiation (unit: micromolePerMeterSquaredPerSecond / missing value: NA) Rh: relative humidity (%) (unit: dimensionless / missing value: NA) SM: volumetric water content (fractional) (unit: dimensionless / missing value: NA) ST: soil temperature at 5cm depth (unit: celsius / missing value: NA)

Q: photosynthetically active radiation (unit: micromolePerMeterSquaredPerSecond/missing value:NA)

Why is Photosynthetically Active Radiation Important?

Photosynthetically Active Radiation is needed for photosynthesis and plant growth. Higher PAR promotes plant growth, and monitoring monitoring PAR is important to ensure plants are receiving adequate light for this process.

PAR values range from 0 to 3,000 millimoles per square meter. At night, PAR is zero. During mid-day in the summer, PAR often reaches 2,000 to 3,000 millimoles per square meter.

PAR of 0 is Night

Mid-day in summer PAR 2,000 to 3,000

https://s.campbellsci.com/documents/ca/manuals/li190sb_man.pdf (https://s.campbellsci.com/documents/ca/manuals/li190sb_man.pdf)

https://curiousplant.com/light-carnivorous-plants-part-2/ (https://curiousplant.com/light-carnivorous-plants-part-2/)

In [10]:
```
df['Q'].describe()
```

Out[10]:
```
count    1.099241e+06
mean     9.723897e+01
std      3.367903e+02
min     -1.753000e+03
25%      0.000000e+00
50%      5.033000e+00
75%      7.214000e+01
max      4.410000e+03
Name: Q, dtype: float64
```

In [11]:
```
df.groupby('hour')
df.groupby('hour').get_group(23).max()
```

Out[11]:
```
Unnamed: 0                 1407175
date           2011-11-15 00:00:00
Species                       unkn
Tag                          20011
Chamber                      S12_C
site                            HF
year                          2011
month                           11
day                             30
hour                            23
chamber                         12
treatment                        S
warming                          C
AT                           28.65
Q                             3466
Rh                           104.2
SM                           0.303
ST                         27.6033
dtype: object
```

In [12]:
```
filtered_data = df[df.hour == 1]
```

In [13]:
```
filtered_data = df[df.Q < 0 ]
```

In [14]:
```
filtered_data
```

Out[14]:

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	ch
3	3	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	4	1
26	26	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	3	1
45	45	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	22	1
67	67	2010-04-29	acru	1.0	S01_5	HF	2010	4	29	20	1
72	72	2010-04-30	acru	1.0	S01_5	HF	2010	4	30	1	1
75	75	2010-04-30	acru	1.0	S01_5	HF	2010	4	30	4	1
99	99	2010-05-17	acru	1.0	S01_5	HF	2010	5	17	4	1
205	205	2010-06-16	acru	1.0	S01_5	HF	2010	6	16	23	1
209	209	2010-07-07	acru	1.0	S01_5	HF	2010	7	7	3	1
535	535	2011-10-13	acru	5.0	G09_5	HF	2011	10	13	2	9
536	536	2011-10-13	acru	5.0	G09_5	HF	2011	10	13	3	9
537	537	2011-10-13	acru	5.0	G09_5	HF	2011	10	13	4	9
558	558	2011-10-20	acru	5.0	G09_5	HF	2011	10	20	1	9
559	559	2011-10-20	acru	5.0	G09_5	HF	2011	10	20	2	9
561	561	2011-10-20	acru	5.0	G09_5	HF	2011	10	20	4	9
579	579	2011-10-20	acru	5.0	G09_5	HF	2011	10	20	22	9
582	582	2011-10-26	acru	5.0	G09_5	HF	2011	10	26	1	9
585	585	2011-10-26	acru	5.0	G09_5	HF	2011	10	26	4	9

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	ch
608	608	2011-11-03	acru	5.0	G09_5	HF	2011	11	3	3	9
611	611	2011-11-03	acru	5.0	G09_5	HF	2011	11	3	6	9
628	628	2011-11-03	acru	5.0	G09_5	HF	2011	11	3	23	9
631	631	2011-11-08	acru	5.0	G09_5	HF	2011	11	8	2	9
634	634	2011-11-08	acru	5.0	G09_5	HF	2011	11	8	5	9
635	635	2011-11-08	acru	5.0	G09_5	HF	2011	11	8	6	9
649	649	2011-11-08	acru	5.0	G09_5	HF	2011	11	8	20	9
650	650	2011-11-08	acru	5.0	G09_5	HF	2011	11	8	21	9
651	651	2011-11-08	acru	5.0	G09_5	HF	2011	11	8	22	9
656	656	2011-11-15	acru	5.0	G09_5	HF	2011	11	15	3	9
658	658	2011-11-15	acru	5.0	G09_5	HF	2011	11	15	5	9
659	659	2011-11-15	acru	5.0	G09_5	HF	2011	11	15	6	9
...	...	...	...	...	...	...	...	...	...	...	...
1405593	1405593	2011-11-08	acru	11468.0	S11_C	HF	2011	11	8	1	11
1405636	1405636	2011-11-15	acru	11468.0	S11_C	HF	2011	11	15	20	11
1405639	1405639	2011-11-15	acru	11468.0	S11_C	HF	2011	11	15	23	11
1405785	1405785	2011-11-08	acru	11470.0	S11_C	HF	2011	11	8	1	11
1405828	1405828	2011-11-15	acru	11470.0	S11_C	HF	2011	11	15	20	11
1405831	1405831	2011-11-15	acru	11470.0	S11_C	HF	2011	11	15	23	11

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	ch
1405977	1405977	2011-11-08	acru	11471.0	S11_C	HF	2011	11	8	1	11
1406020	1406020	2011-11-15	acru	11471.0	S11_C	HF	2011	11	15	20	11
1406023	1406023	2011-11-15	acru	11471.0	S11_C	HF	2011	11	15	23	11
1406217	1406217	2011-11-08	acru	11475.0	S11_C	HF	2011	11	8	1	11
1406260	1406260	2011-11-15	acru	11475.0	S11_C	HF	2011	11	15	20	11
1406263	1406263	2011-11-15	acru	11475.0	S11_C	HF	2011	11	15	23	11
1406409	1406409	2011-11-08	fram	11477.0	S11_C	HF	2011	11	8	1	11
1406452	1406452	2011-11-15	fram	11477.0	S11_C	HF	2011	11	15	20	11
1406455	1406455	2011-11-15	fram	11477.0	S11_C	HF	2011	11	15	23	11
1406649	1406649	2011-11-08	acru	11496.0	S11_C	HF	2011	11	8	1	11
1406692	1406692	2011-11-15	acru	11496.0	S11_C	HF	2011	11	15	20	11
1406695	1406695	2011-11-15	acru	11496.0	S11_C	HF	2011	11	15	23	11
1406723	1406723	2010-10-21	piun	19005.0	S10_C	HF	2010	10	21	3	10
1406725	1406725	2010-10-21	piun	19005.0	S10_C	HF	2010	10	21	5	10
1406773	1406773	2010-11-08	piun	19005.0	S10_C	HF	2010	11	8	5	10
1406791	1406791	2010-11-08	piun	19005.0	S10_C	HF	2010	11	8	23	10
1406867	1406867	2010-10-21	piun	20003.0	S11_C	HF	2010	10	21	3	11
1406890	1406890	2010-10-28	piun	20003.0	S11_C	HF	2010	10	28	2	11

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	ch
1406956	1406956	2010-11-18	piun	20003.0	S11_C	HF	2010	11	18	20	11
1406957	1406957	2010-11-18	piun	20003.0	S11_C	HF	2010	11	18	21	11
1407059	1407059	2010-10-21	acru	20011.0	S11_C	HF	2010	10	21	3	11
1407082	1407082	2010-10-28	acru	20011.0	S11_C	HF	2010	10	28	2	11
1407148	1407148	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	20	11
1407149	1407149	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	21	11

61659 rows × 18 columns

In [15]:
```
df.sort_values('month', axis=0, ascending=False)
```

Out[15]:

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	ch
703588	703588	2010-11-29	magr	3996.0	S04_5	HF	2010	11	29	8	4
620649	620649	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	13	3
620660	620660	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	24	3
620659	620659	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	23	3
620658	620658	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	22	3
620657	620657	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	21	3
620656	620656	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	20	3
620655	620655	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	19	3
620654	620654	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	18	3
620653	620653	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	17	3
620652	620652	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	16	3
620651	620651	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	15	3
620650	620650	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	14	3
620648	620648	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	12	3
620635	620635	2010-11-08	quru	3608.0	S03_3	HF	2010	11	8	23	3
620647	620647	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	11	3
620646	620646	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	10	3
620645	620645	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	9	3

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	ch
620644	620644	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	8	3
620643	620643	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	7	3
620642	620642	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	6	3
620641	620641	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	5	3
620640	620640	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	4	3
620639	620639	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	3	3
620638	620638	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	2	3
620637	620637	2011-11-08	quru	3608.0	S03_3	HF	2011	11	8	1	3
620661	620661	2011-11-15	quru	3608.0	S03_3	HF	2011	11	15	1	3
620662	620662	2011-11-15	quru	3608.0	S03_3	HF	2011	11	15	2	3
620663	620663	2011-11-15	quru	3608.0	S03_3	HF	2011	11	15	3	3
620664	620664	2011-11-15	quru	3608.0	S03_3	HF	2011	11	15	4	3
...	...	...	...	...	...	...	...	...	...	...	...
515068	515068	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	14	8
515069	515069	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	15	8
515070	515070	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	16	8
515071	515071	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	17	8
515072	515072	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	18	8
515073	515073	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	19	8

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	ch
515074	515074	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	20	8
515075	515075	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	21	8
515076	515076	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	22	8
515077	515077	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	23	8
515078	515078	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	24	8
515063	515063	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	9	8
515062	515062	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	8	8
515061	515061	2010-04-21	acru	3377.0	S08_3	HF	2010	4	21	7	8
1284177	1284177	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	16	4
1284171	1284171	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	10	4
1284172	1284172	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	11	4
1284173	1284173	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	12	4
1284174	1284174	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	13	4
1284175	1284175	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	14	4
1284176	1284176	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	15	4
1284178	1284178	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	17	4
1284185	1284185	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	24	4
1284179	1284179	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	18	4

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	ch
1284180	1284180	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	19	4
1284181	1284181	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	20	4
1284182	1284182	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	21	4
1284183	1284183	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	22	4
1284184	1284184	2010-04-30	acru	10768.0	S04_5	HF	2010	4	30	23	4
0	0	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	1	1

1407177 rows × 18 columns

Adding a new feature - ppfd (Photosynthetic Photon Flux Density)

Created by using the Q: photosynthetically active radiation * Avogadro's number

In [16]:
```
def calculate_ppfd(Q):
    ppfd = Q * (6.0221409**23)
    return ppfd

df['ppfd'] = df.Q.apply(calculate_ppfd)
```

In [17]:
```
df.describe()
```

Out[17]:

	Unnamed: 0	Tag	year	month	day	
count	1.407177e+06	1.407177e+06	1.407177e+06	1.407177e+06	1.407177e+06	1.407177
mean	7.035880e+05	6.488845e+03	2.010447e+03	8.398515e+00	1.662539e+01	1.254754
std	4.062172e+05	3.720405e+03	5.266105e-01	2.324370e+00	8.881952e+00	6.915155
min	0.000000e+00	1.000000e+00	2.009000e+03	4.000000e+00	1.000000e+00	1.000000
25%	3.517940e+05	2.957000e+03	2.010000e+03	6.000000e+00	8.000000e+00	7.000000
50%	7.035880e+05	3.996000e+03	2.010000e+03	9.000000e+00	1.800000e+01	1.300000
75%	1.055382e+06	1.019000e+04	2.011000e+03	1.000000e+01	2.600000e+01	1.900000
max	1.407176e+06	2.001100e+04	2.011000e+03	1.100000e+01	3.000000e+01	2.400000

Add feature - CO2 uptake and CO2 release are both related to Q.

These are boolean values created by a conditional statement.

When Q is positive number CO2_uptake will read TRUE. When Q is negative number CO2_release will read TRUE.

In [18]:
```
df['CO2_uptake'] = np.where(df['Q']>=0, True, False) #during the day CO2 into
 the plant
```

In [19]:
```
df['CO2_release'] = np.where(df['Q']<=0, True, False) #at night CO2 is being r
eleased
```

In [20]:
```
df.isnull().sum().sum() #total na's
```

Out[20]:
```
1485217
```

Large number of missing values within the environmental variables.

1. Use a groupby method and fillna by the monthly mean per environmental variable.
2. This likely will introduce variance into the models. However, by doing this I hope to reduce the bias and minize the variance.

In [21]:
```
df.isnull().sum()
```

Out[21]:
```
Unnamed: 0          0
date                0
Species             0
Tag                 0
Chamber             0
site                0
year                0
month               0
day                 0
hour                0
chamber             0
treatment           0
warming             0
AT              28645
Q              307936
Rh             757305
SM              53790
ST              29605
ppfd           307936
CO2_uptake          0
CO2_release         0
dtype: int64
```

In [23]:
```
df['Q'] = df.groupby('month').fillna((df['Q'].mean()))
```

In [24]:
```
df.head()
```

Out[24]:

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	trea
0	0	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	1	1	S
1	1	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	2	1	S
2	2	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	3	1	S
3	3	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	4	1	S
4	4	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	5	1	S

In [25]:
```
g = df.groupby('month').mean()
g
```

Out[25]:

	Unnamed: 0	Tag	year	day	hour	chamber	
month							
4	782598.589151	7171.094086	2010.377056	21.686305	12.500000	4.324144	10.172
5	730058.465704	6789.717911	2010.523094	18.071968	12.880173	4.469591	18.147
6	712563.088452	6567.208095	2010.589584	14.231248	12.500000	4.803566	16.897
7	667784.096889	6097.067032	2010.555839	17.560938	12.500000	4.830189	24.346
8	645180.462294	5673.751460	2010.000000	2.000000	12.500000	4.360064	21.211
9	700802.449504	6412.761079	2010.269141	20.296936	12.500000	4.716012	20.011
10	690671.366376	6404.501565	2010.516896	15.745448	12.500000	4.827679	12.383
11	693926.832879	6457.635298	2010.508231	13.853479	12.500000	4.880582	7.7984

In [26]:
```
df['Rh'] = df.groupby('month').fillna((df['Rh'].mean()))
df['AT'] = df.groupby('month').fillna((df['AT'].mean()))
df['SM'] = df.groupby('month').fillna((df['SM'].mean()))
df['ST'] = df.groupby('month').fillna((df['ST'].mean()))
df['ppfd'] = df.groupby('month').fillna((df['ppfd'].mean()))
```

In [27]:
```
df.drop(columns=['Unnamed: 0'])
```

Out[27]:

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
0	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	1	1	S
1	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	2	1	S
2	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	3	1	S
3	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	4	1	S
4	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	5	1	S
5	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	6	1	S
6	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	7	1	S
7	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	8	1	S
8	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	9	1	S
9	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	10	1	S
10	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	11	1	S
11	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	12	1	S
12	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	13	1	S
13	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	14	1	S
14	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	15	1	S
15	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	16	1	S
16	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	17	1	S
17	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	18	1	S
18	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	19	1	S

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
19	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	20	1	S
20	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	21	1	S
21	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	22	1	S
22	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	23	1	S
23	2010-04-15	acru	1.0	S01_5	HF	2010	4	15	24	1	S
24	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	1	1	S
25	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	2	1	S
26	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	3	1	S
27	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	4	1	S
28	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	5	1	S
29	2010-04-21	acru	1.0	S01_5	HF	2010	4	21	6	1	S
...	...	...	...	...	...	...	...	...	...	...	...
1407147	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	19	11	S
1407148	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	20	11	S
1407149	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	21	11	S
1407150	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	22	11	S
1407151	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	23	11	S
1407152	2010-11-18	acru	20011.0	S11_C	HF	2010	11	18	24	11	S
1407153	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	1	11	S

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
1407154	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	2	11	S
1407155	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	3	11	S
1407156	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	4	11	S
1407157	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	5	11	S
1407158	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	6	11	S
1407159	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	7	11	S
1407160	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	8	11	S
1407161	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	9	11	S
1407162	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	10	11	S
1407163	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	11	11	S
1407164	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	12	11	S
1407165	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	13	11	S
1407166	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	14	11	S
1407167	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	15	11	S
1407168	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	16	11	S
1407169	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	17	11	S
1407170	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	18	11	S
1407171	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	19	11	S
1407172	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	20	11	S

	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	treat
1407173	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	21	11	S
1407174	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	22	11	S
1407175	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	23	11	S
1407176	2010-11-29	acru	20011.0	S11_C	HF	2010	11	29	24	11	S

1407177 rows × 20 columns

Now we have a complete dataset without any nulls.

In [28]:
```
df.isnull().sum()
```

Out[28]:
```
Unnamed: 0     0
date           0
Species        0
Tag            0
Chamber        0
site           0
year           0
month          0
day            0
hour           0
chamber        0
treatment      0
warming        0
AT             0
Q              0
Rh             0
SM             0
ST             0
ppfd           0
CO2_uptake     0
CO2_release    0
dtype: int64
```

Duke Forest dataset

In [29]:
```
df1.head()
```

Out[29]:

	site	year	times	month	day	JD	JD2009	dayFraction	chamber	treatment	
0	DF	2009	125.54	5	6	-240	125	0.54	1	G	24.226
1	DF	2009	125.54	5	6	-240	125	0.54	2	G	24.513
2	DF	2009	125.54	5	6	-240	125	0.54	3	G	24.070
3	DF	2009	125.54	5	6	-240	125	0.54	4	G	24.443
4	DF	2009	125.54	5	6	-240	125	0.54	5	G	24.305

Adding datetime stamp

In [30]:
```
df1['date']  = pd.to_datetime(df1[['year', 'month', 'day']])
```

In [31]:
```
df1.head() #Duke
```

Out[31]:

	site	year	times	month	day	JD	JD2009	dayFraction	chamber	treatment	
0	DF	2009	125.54	5	6	-240	125	0.54	1	G	24.226
1	DF	2009	125.54	5	6	-240	125	0.54	2	G	24.513
2	DF	2009	125.54	5	6	-240	125	0.54	3	G	24.070
3	DF	2009	125.54	5	6	-240	125	0.54	4	G	24.443
4	DF	2009	125.54	5	6	-240	125	0.54	5	G	24.305

Large number of missing null values in Duke Forest dataset as well.

Using the same approach as above. Taking the groupby of month and fillna per environmental variable with a mean.

In [32]:
```
df1.isnull().sum()
```

Out[32]:
```
site                0
year                0
times               0
month               0
day                 0
JD                  0
JD2009              0
dayFraction         0
chamber             0
treatment           0
AT              77654
Q              158106
Rh             750768
SM                162
ST              77614
date                0
dtype: int64
```

In [33]:
```
df1['AT'] = df1.groupby('month').fillna((df1['AT'].mean()))
df1['Q'] = df1.groupby('month').fillna((df1['Q'].mean()))
df1['Rh'] = df1.groupby('month').fillna((df1['Rh'].mean()))
df1['SM'] = df1.groupby('month').fillna((df1['SM'].mean()))
df1['ST'] = df1.groupby('month').fillna((df1['ST'].mean()))
```

In [34]:
```
df1.isnull().sum()
```

Out[34]:
```
site           0
year           0
times          0
month          0
day            0
JD             0
JD2009         0
dayFraction    0
chamber        0
treatment      0
AT             0
Q              0
Rh             0
SM             0
ST             0
date           0
dtype: int64
```

In [35]:
```
df1['CO2_uptake'] = np.where(df1['Q']>=0, True, False) #during the day CO2 int
o the plant
df1['CO2_release'] = np.where(df1['Q']<=0, True, False) #at night CO2 is being
released
```

In [36]:
```
def calculate_ppfd(Q):
    ppfd = Q * (6.0221409**23)
    return ppfd

df1['ppfd'] = df1.Q.apply(calculate_ppfd)
```

In [37]:
```
df1.isnull().sum()
```

Out[37]:
```
site           0
year           0
times          0
month          0
day            0
JD             0
JD2009         0
dayFraction    0
chamber        0
treatment      0
AT             0
Q              0
Rh             0
SM             0
ST             0
date           0
CO2_uptake     0
CO2_release    0
ppfd           0
dtype: int64
```

In [38]:
```
df1.dtypes
```

Out[38]:
```
site                   object
year                    int64
times                 float64
month                   int64
day                     int64
JD                      int64
JD2009                  int64
dayFraction           float64
chamber                 int64
treatment              object
AT                     object
Q                      object
Rh                     object
SM                     object
ST                     object
date           datetime64[ns]
CO2_uptake               bool
CO2_release              bool
ppfd                  float64
dtype: object
```

In [39]:
```
df.dtypes
```

Out[39]:
```
Unnamed: 0              int64
date           datetime64[ns]
Species                object
Tag                   float64
Chamber                object
site                   object
year                    int64
month                   int64
day                     int64
hour                    int64
chamber                 int64
treatment              object
warming                object
AT                     object
Q                      object
Rh                     object
SM                     object
ST                     object
ppfd                   object
CO2_uptake               bool
CO2_release              bool
dtype: object
```

Renaming dataframes so that I can use them later. But preparing to merge datasets.

In [40]:
```
df2 = df1.drop(columns=['times', 'JD', 'JD2009', 'dayFraction'])# Duke Forest

df3 = df.drop(columns=['Tag', 'Species', 'hour', 'warming']) # Harvard Forest
```

Combining Harvard and Duke forest datasets

In [41]:
```
HfDf = pd.concat([df2, df3], ignore_index=True)
```

Re-ordering the columns so that it is easier to read and date is on the far left.

In [42]:
```
HfDf = HfDf[['date', 'year', 'month', 'day', 'site', 'chamber', 'treatment',
'AT', 'Q', 'Rh', 'SM',
            'ST', 'ppfd', 'CO2_uptake', 'CO2_release']]
```

In [43]:
```
HfDf.head(10)
```

Out[43]:

	date	year	month	day	site	chamber	treatment	AT	Q	Rh	SM
0	2009-05-06	2009	5	6	DF	1	G	24.2267	24.2267	24.2267	24.2267
1	2009-05-06	2009	5	6	DF	2	G	24.5133	24.5133	24.5133	24.5133
2	2009-05-06	2009	5	6	DF	3	G	24.07	24.07	24.07	24.07
3	2009-05-06	2009	5	6	DF	4	G	24.4433	24.4433	24.4433	24.4433
4	2009-05-06	2009	5	6	DF	5	G	24.305	24.305	24.305	24.305
5	2009-05-06	2009	5	6	DF	6	G	24.28	24.28	24.28	24.28
6	2009-05-06	2009	5	6	DF	7	G	23.865	23.865	23.865	23.865
7	2009-05-06	2009	5	6	DF	8	G	23.76	23.76	23.76	23.76
8	2009-05-06	2009	5	6	DF	9	G	24.175	24.175	24.175	24.175
9	2009-05-06	2009	5	6	DF	10	G	17.3824	17.3824	17.3824	17.3824

In [44]:
```
#Converting light treatment

HfDf['treatment'] = HfDf['treatment'].apply({ 'G': 0, 'S': 1}.get)
HfDf['site'] = HfDf['site'].apply({ 'HF': 0, 'DF': 1}.get)
```

In [45]:
```
HfDf["Rh"] = HfDf.Rh.convert_objects(convert_numeric=True)
HfDf["Q"] = HfDf.Q.convert_objects(convert_numeric=True)
HfDf["SM"] = HfDf.SM.convert_objects(convert_numeric=True)
HfDf["ST"] = HfDf.ST.convert_objects(convert_numeric=True)
HfDf["AT"] = HfDf.AT.convert_objects(convert_numeric=True)
```

In [46]:
```
HfDf.dtypes
```

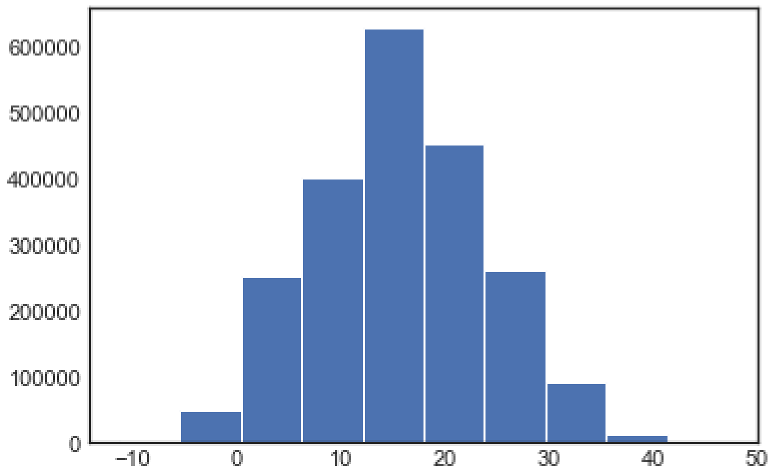
Out[46]:
```
date          datetime64[ns]
year                   int64
month                  int64
day                    int64
site                   int64
chamber                int64
treatment              int64
AT                   float64
Q                    float64
Rh                   float64
SM                   float64
ST                   float64
ppfd                  object
CO2_uptake              bool
CO2_release             bool
dtype: object
```

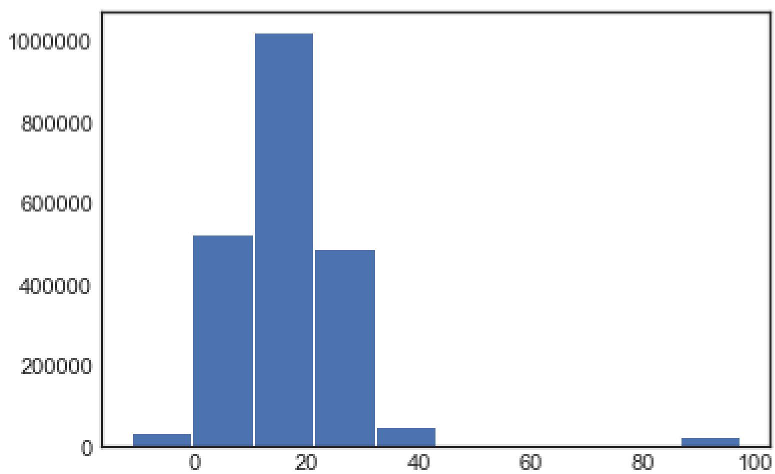
Exploratory Data Analysis

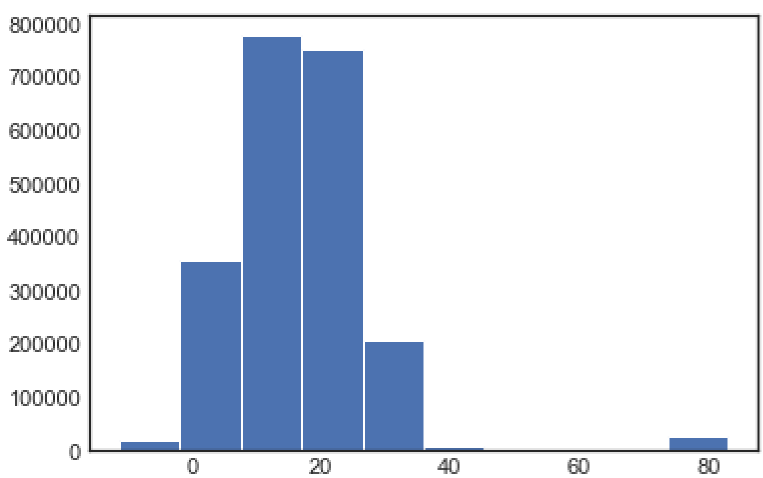
In [47]:
```
#Just continious features
cont_data=HfDf.ix[:,'AT':'ST']
```

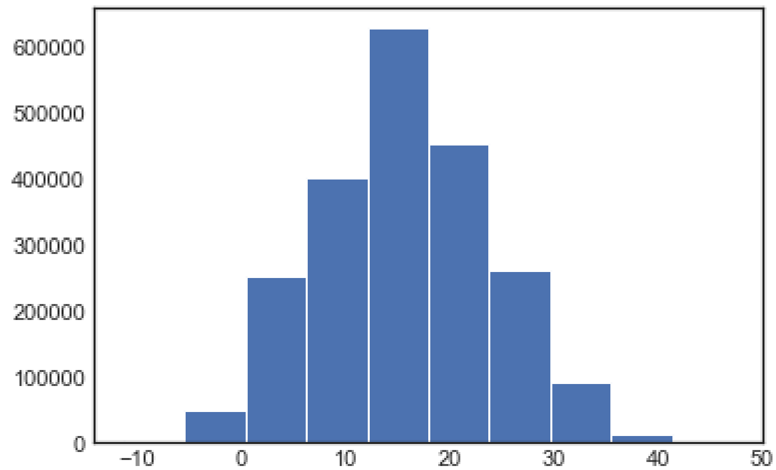
In [48]:
```
#All binary features
binary_data=HfDf.ix[:,'year': 'treatment']
```

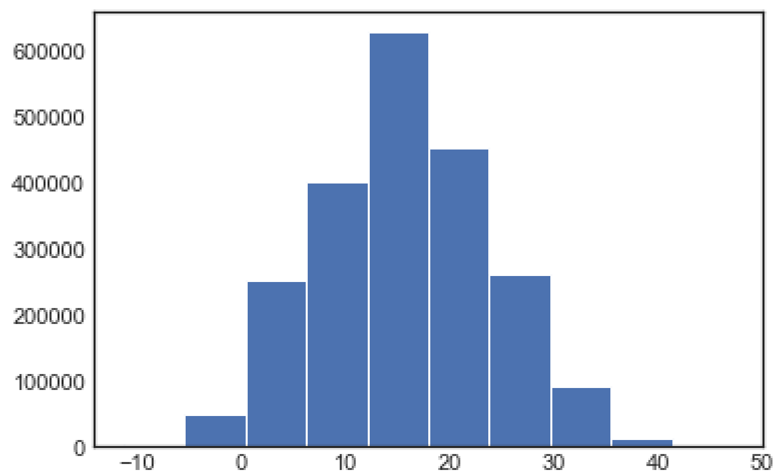
In [49]:
```
for i, col in enumerate(cont_data.columns):
    plt.figure(i)
    plt.hist(cont_data[col])
```


600000
500000
400000
300000
200000
100000
0
−10
0
10
20
30
40
50


1000000
800000
600000
400000
200000
0
0
20
40
60
80
100


800000
700000
600000
500000
400000
300000
200000
100000
0
0
20
40
60
80





Histograms:

AT: Air temperature is normally distributed, has a nice bell shape.

Q: Photosynthetically active radiation (PAR) - bell shape shifted to the right. When I initially had fillna with 0 it was just a column at the 0 mark.

Rh: Relative humdity - a boarder histogram that is similar to Q.

SM: volumetric water content - it is a very even histogram

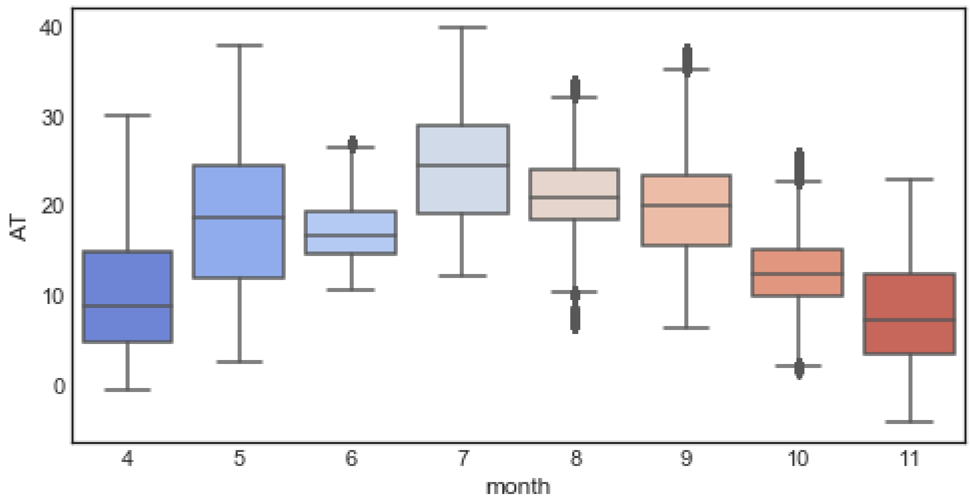
ST: Soil temperature: follows a very similar pattern to AT.

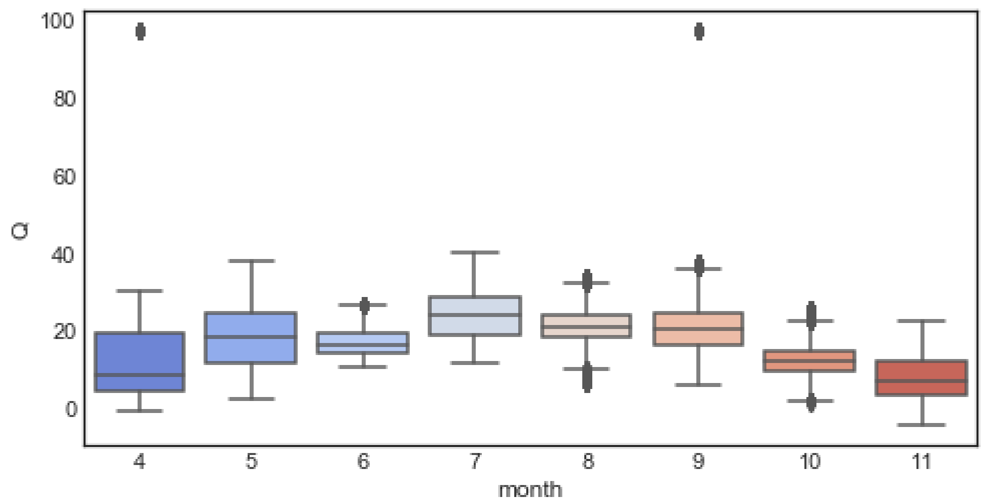
EDA Harvard Forest

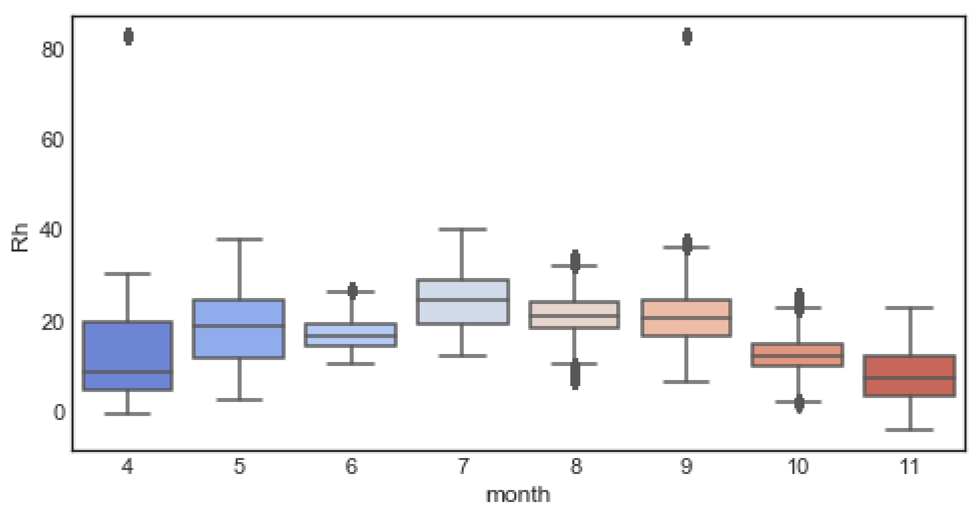
In [52]:
```
df['treatment'] = df['treatment'].apply({ 'G': 0, 'S': 1}.get)
df['treatment'] = df['treatment'].apply({ 'C': 0, 'A': 1, '3': 3, '5': 5}.get)
```

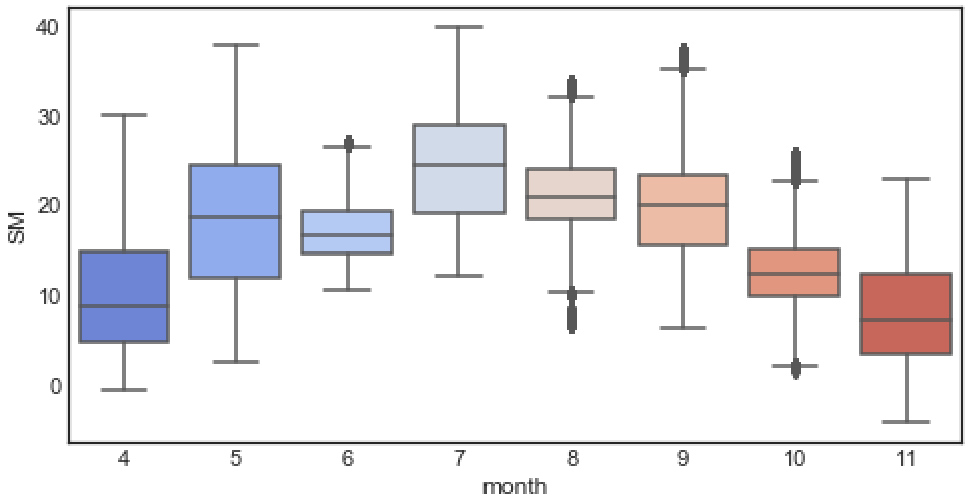
In [53]:
```
df["Rh"] = df.Rh.convert_objects(convert_numeric=True)
df["Q"] = df.Q.convert_objects(convert_numeric=True)
df["SM"] = df.SM.convert_objects(convert_numeric=True)
df["ST"] = df.ST.convert_objects(convert_numeric=True)
df["AT"] = df.AT.convert_objects(convert_numeric=True)

df["chamber"] = df.chamber.convert_objects(convert_numeric=True)
df["treatment"] = df.treatment.convert_objects(convert_numeric=True)
df["warming"] = df.warming.convert_objects(convert_numeric=True)
```

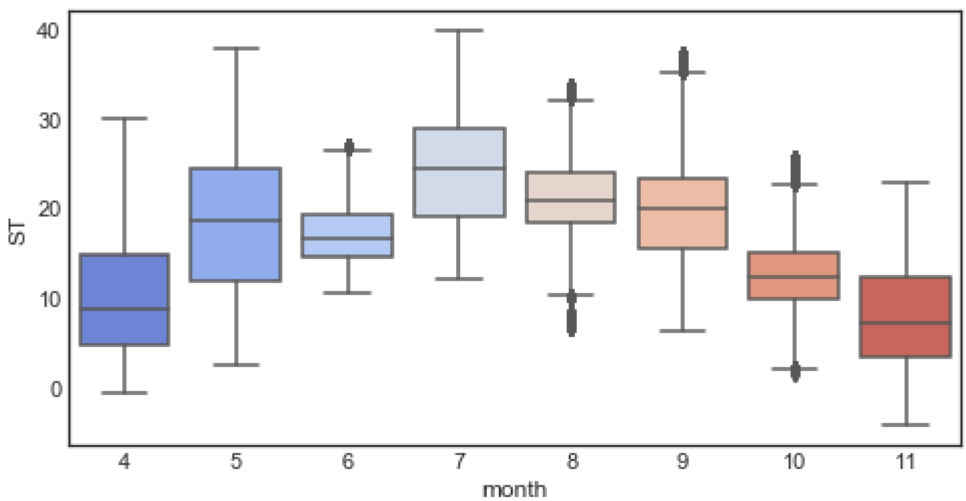
In [54]:
```
cont_hf=df.ix[:,'AT':'ST']

df['month']=df['month'].astype('category') #To convert target class into categ
ory

for i, col in enumerate(cont_hf.columns):
    plt.figure(i,figsize=(8,4))
    sns.boxplot(x=df['month'], y=col, data=df, palette="coolwarm")
```


40
30
20
10
0
AT
4
5
6
7
8
9
10
11
month


100
80
60
40
20
0
Q
4
5
6
7
8
9
10
11
month


80
60
40
20
0
Rh
4
5
6
7
8
9
10
11
month





Harvard Forest - Environmental variables by Month - Barplots

AT: Air temperature: Show a seasonal warming and cooling pattern. However, it months of May, July, August and November there is much more variability in air temperature. May, likely has spring rain and late season storms in Boston. July/August, could be influenced by hurricanes reaching up the north east. November,annomalous late season warming.

Q: Photosynthetically active radiation (PAR) - The data seems very consisted, except for 2 annomalous data points in April and August.

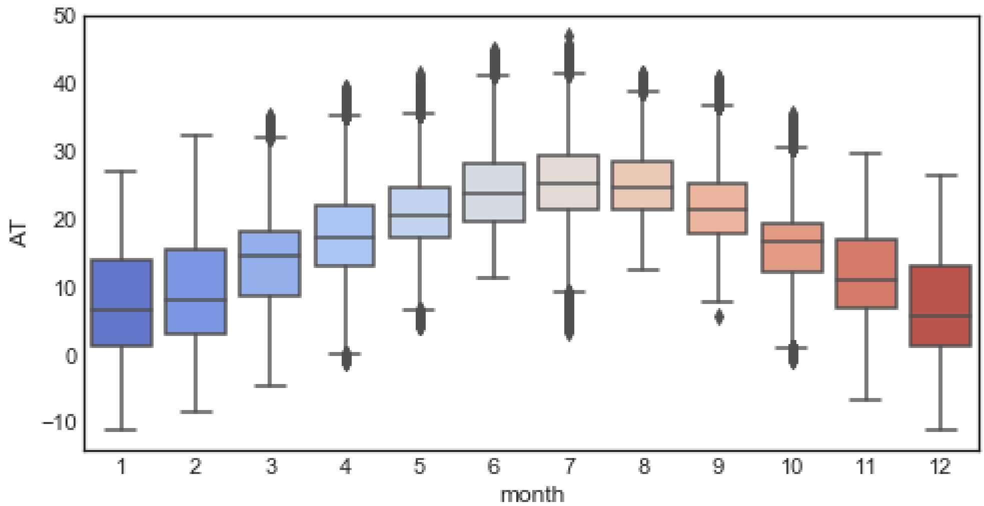
Rh: Relative humdity and ST: Soil temperature: follows a very similar pattern to AT.

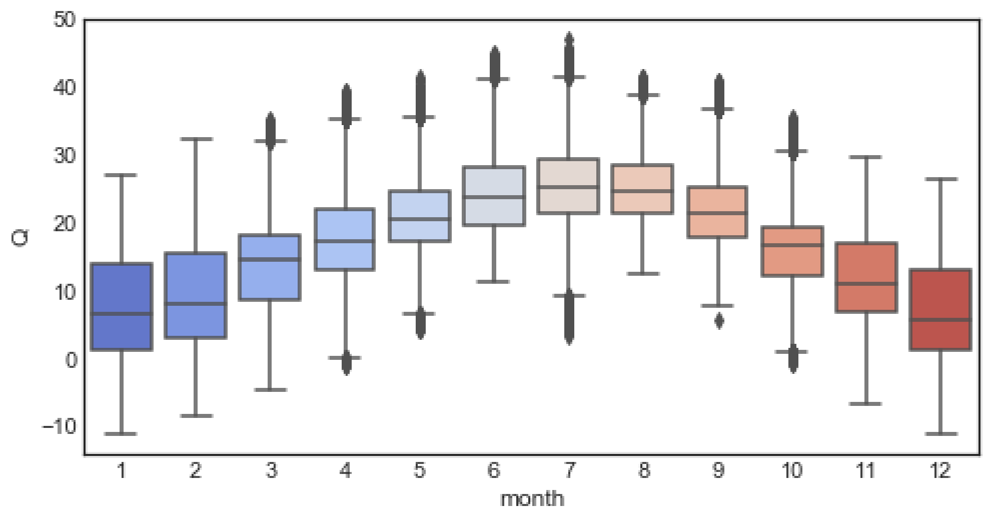
Duke Forest EDA

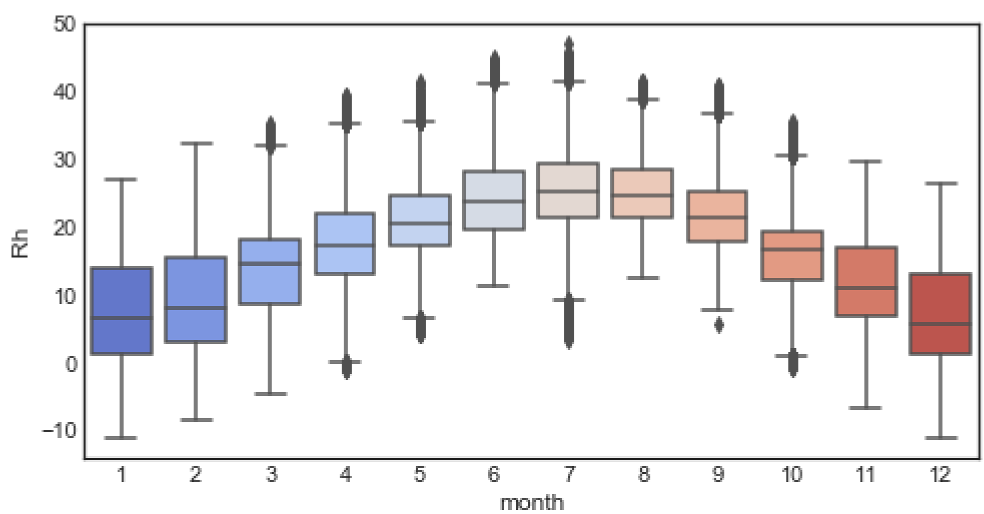
In [55]:
```
df1['treatment'] = df1['treatment'].apply({ 'G': 0, 'S': 1}.get)
df1['treatment'] = df1['treatment'].apply({ 'C': 0, 'A': 1, '3': 3, '5': 5}.ge
t)
```

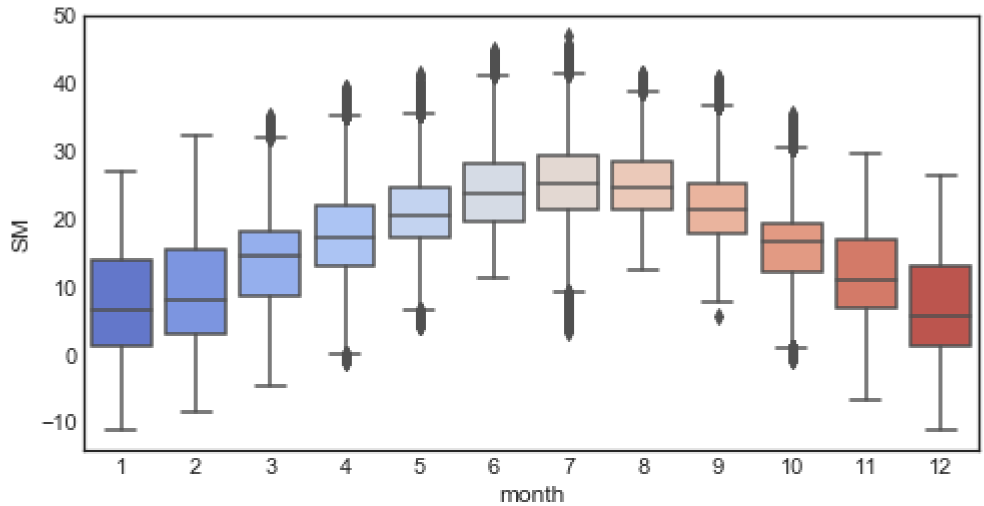
In [56]:
```
df1["Rh"] = df1.Rh.convert_objects(convert_numeric=True)
df1["Q"] = df1.Q.convert_objects(convert_numeric=True)
df1["SM"] = df1.SM.convert_objects(convert_numeric=True)
df1["ST"] = df1.ST.convert_objects(convert_numeric=True)
df1["AT"] = df1.AT.convert_objects(convert_numeric=True)
df1["chamber"] = df1.chamber.convert_objects(convert_numeric=True)
df1["treatment"] = df1.treatment.convert_objects(convert_numeric=True)
```

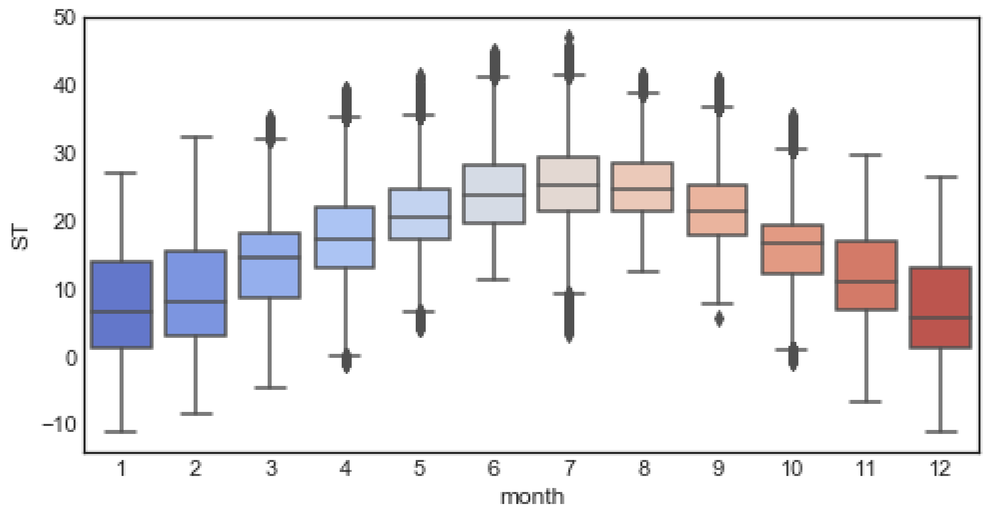
In [57]:
```
cont_df=df1.ix[:,'AT':'ST']

df1['month']=df1['month'].astype('category') #To convert target class into cat
egory

for i, col in enumerate(cont_df.columns):
    plt.figure(i,figsize=(8,4))
    sns.boxplot(x=df1['month'], y=col, data=df1, palette="coolwarm")
```


AT
50
40
30
20
10
0
−10
1
2
3
4
5
6
7
8
9
10
11
12
month


Q
50
40
30
20
10
0
−10
1
2
3
4
5
6
7
8
9
10
11
12
month


Rh
50
40
30
20
10
0
−10
1
2
3
4
5
6
7
8
9
10
11
12
month



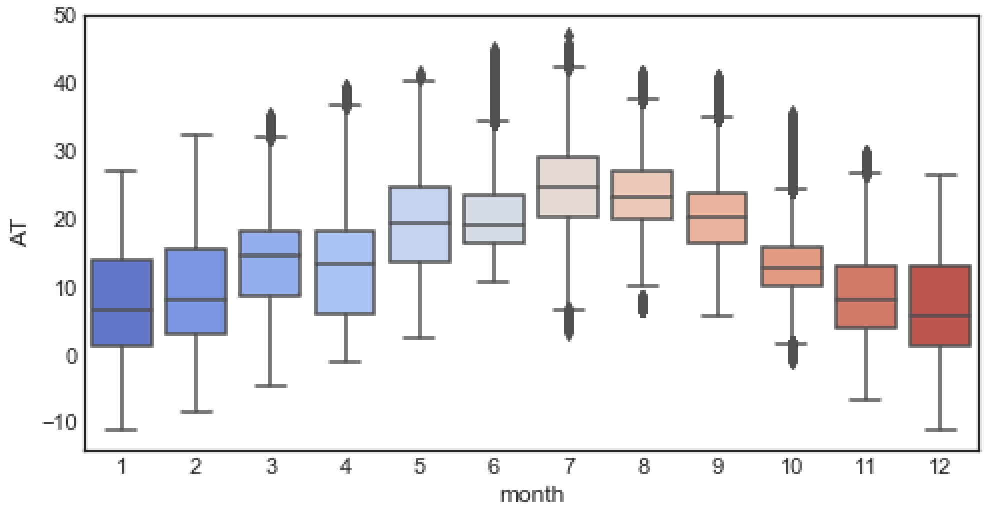


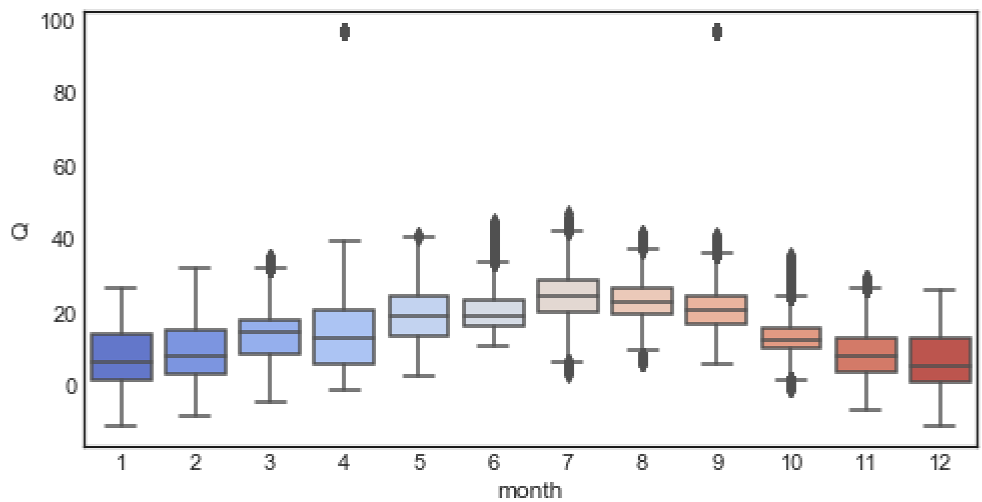
Duke Forest - Environmental variables by Month - Barplots

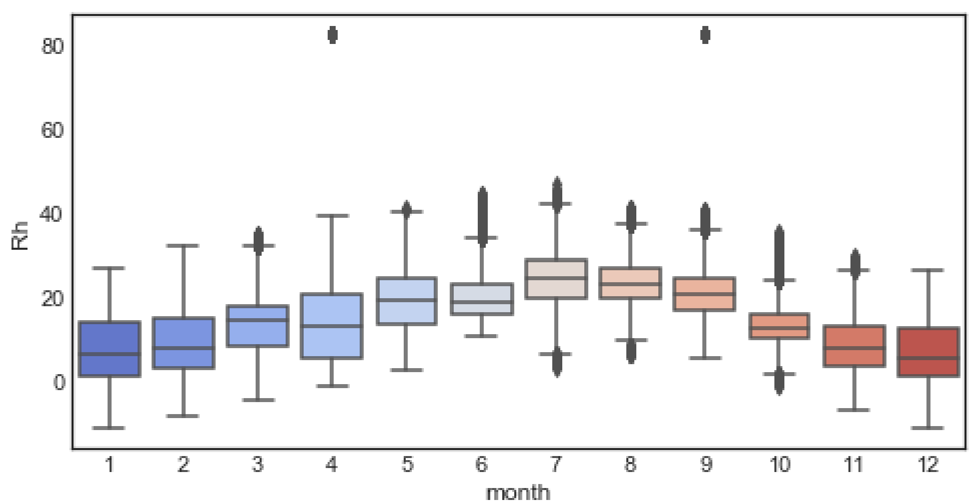
All of the environmental variables for Duke follow the same pattern. There is a bit of concern that fillna with mean leveled out the data too much? Hard to tell. Looks a bit suspicious thou.

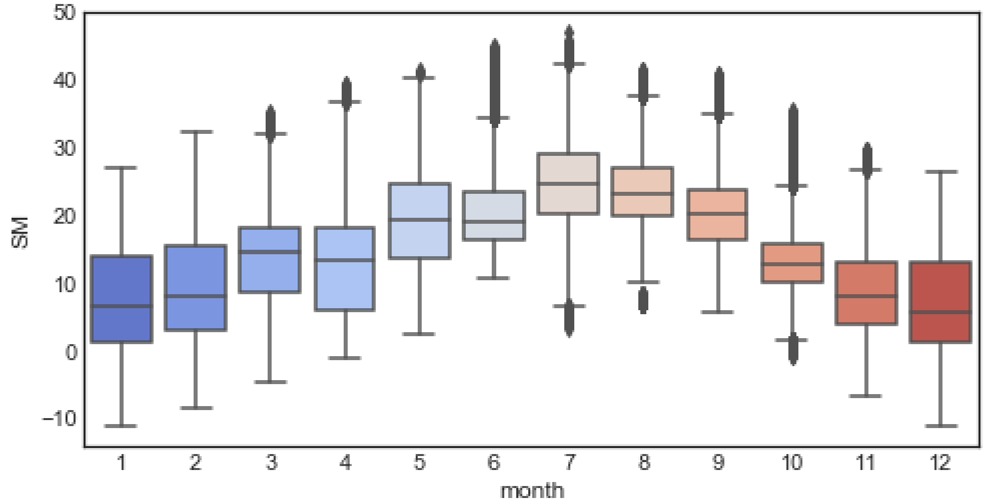
The entire dataset

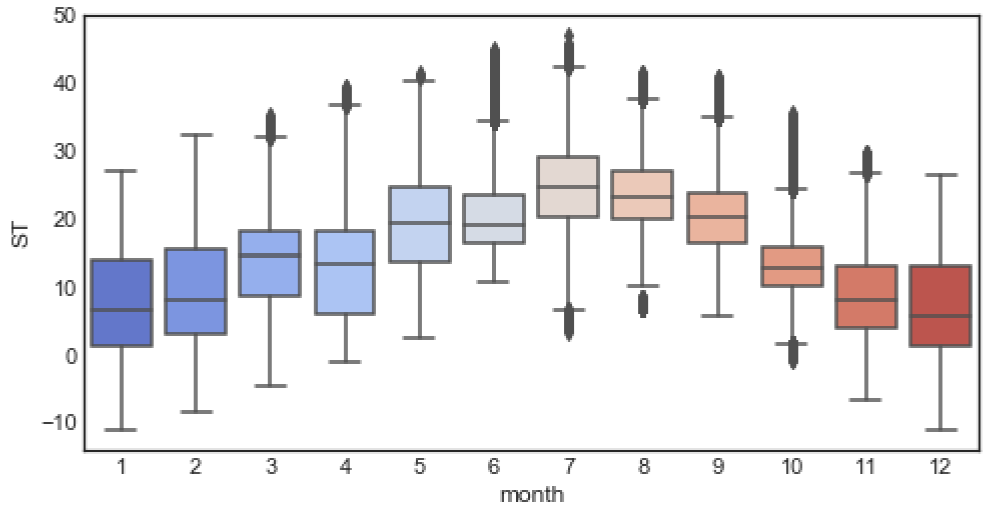
In [58]:
```
HfDf['month']=HfDf['month'].astype('category') #To convert target class into c
ategory

for i, col in enumerate(cont_data.columns):
    plt.figure(i,figsize=(8,4))
    sns.boxplot(x=HfDf['month'], y=col, data=HfDf, palette="coolwarm")
```


AT
month
50
40
30
20
10
0
−10
1 2 3 4 5 6 7 8 9 10 11 12


Q
month
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12


Rh
month
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12





Environmental variables by Month - Barplots

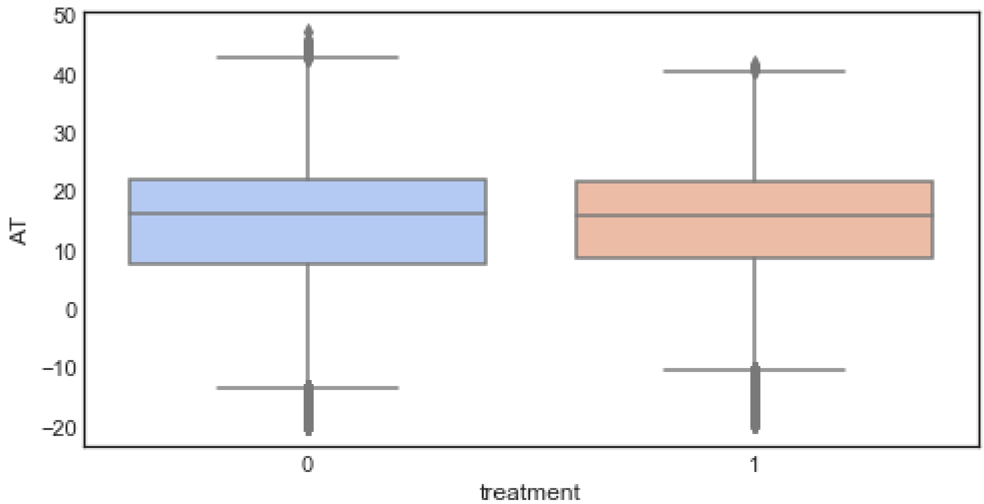
AT: Air temperature: by combining the datasets does pull the AT down a little bit. This is Harvard forest.

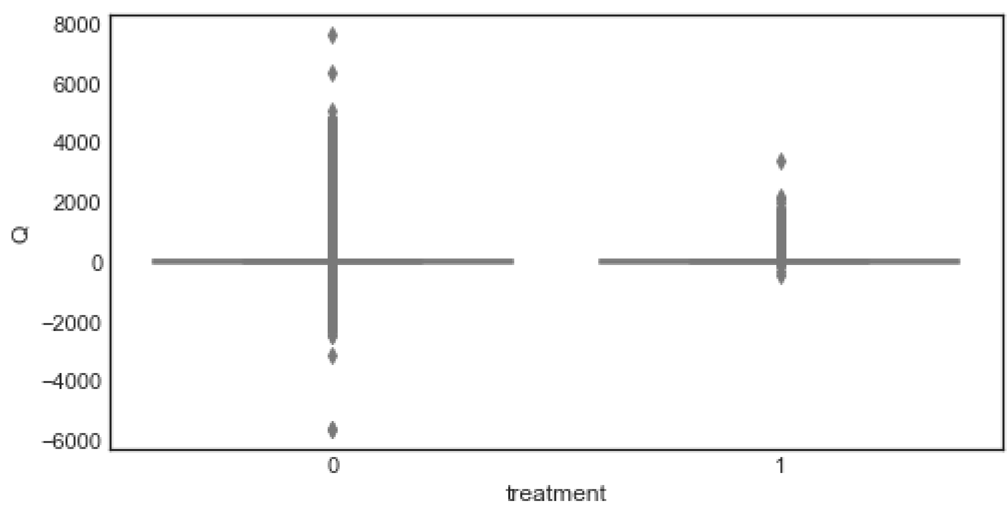
Q: Photosynthetically active radiation (PAR) - Two annolamous data points again at April and August, it is from the Harvard dataset.

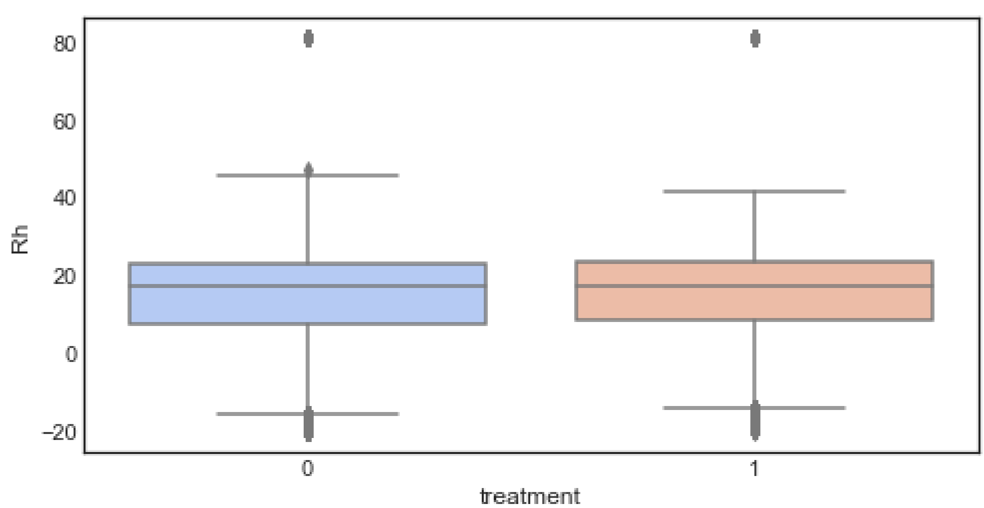
Rh: Relative humdity - Two annolamous data points again at April and August, it is from the Harvard dataset.

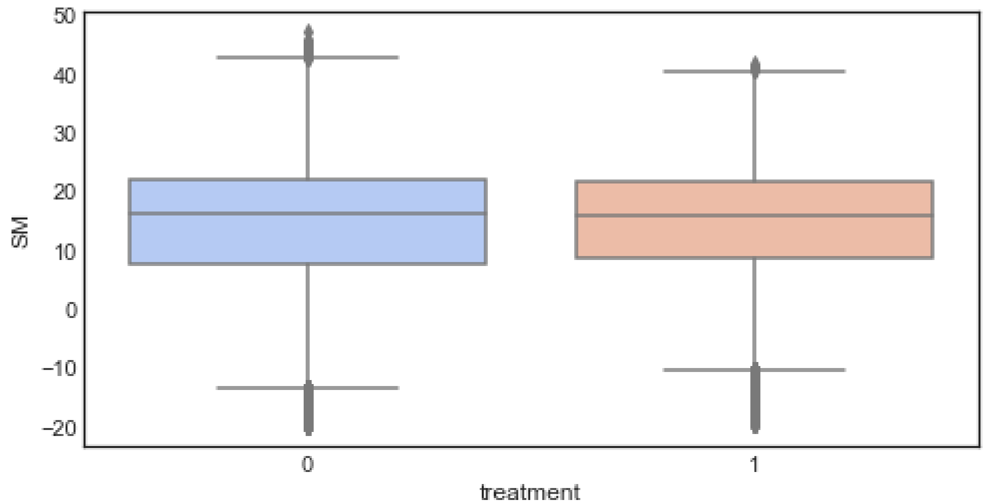
SM and ST are very similar to AT.

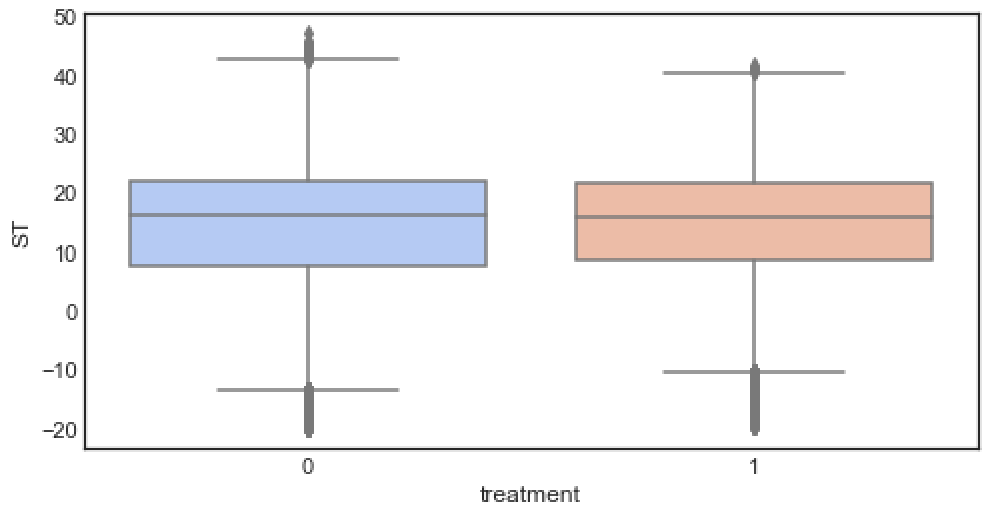
In [62]:
```
HfDf['treatment']=HfDf['treatment'].astype('category') #To convert target clas
s into category

for i, col in enumerate(cont_data.columns):
    plt.figure(i,figsize=(8,4))
    sns.boxplot(x=HfDf['treatment'], y=col, data=HfDf, palette="coolwarm")
```


50
40
30
20
10
0
−10
−20
AT
0
1
treatment


8000
6000
4000
2000
0
−2000
−4000
−6000
Q
0
1
treatment


80
60
40
20
0
−20
Rh
0
1
treatment



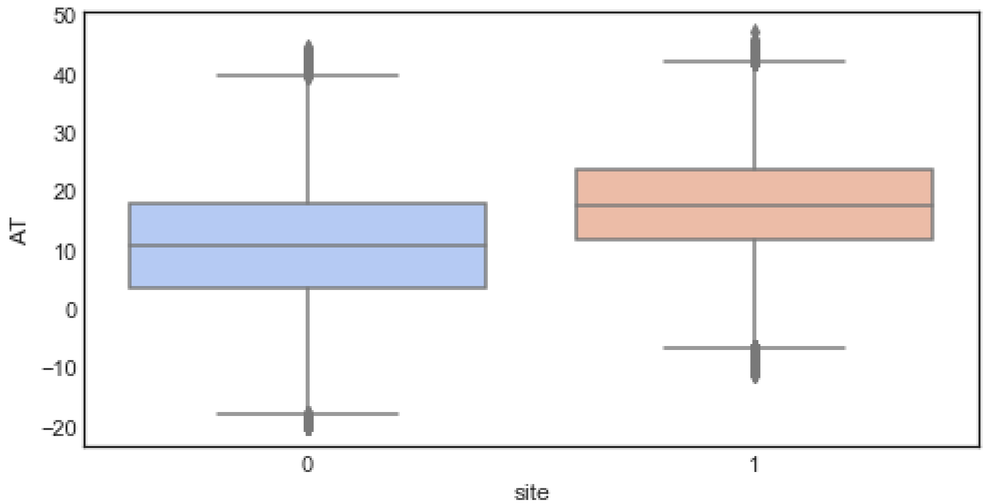


Comparing the treatment types:

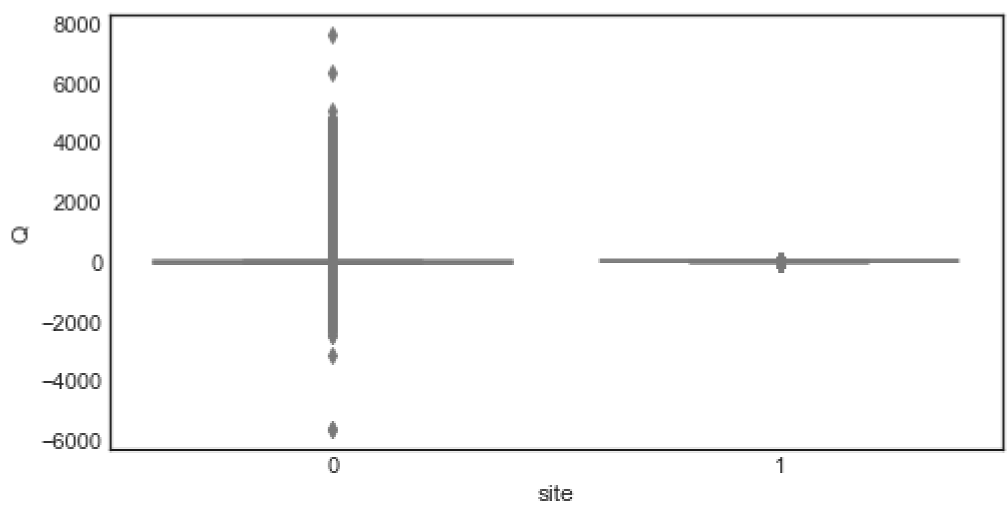
light treatment 0 - G: chamber in open gap 1 - S: chamber under closed canopy

Overall they seem relatively similar. However, the ranges on the barplots are wider across all environmental variables for open gap instead of closed canopy. Q (PAR) had the most noticable effect, since Q is a measure of photosythensis having more light throught a gap in the forest makes sense.

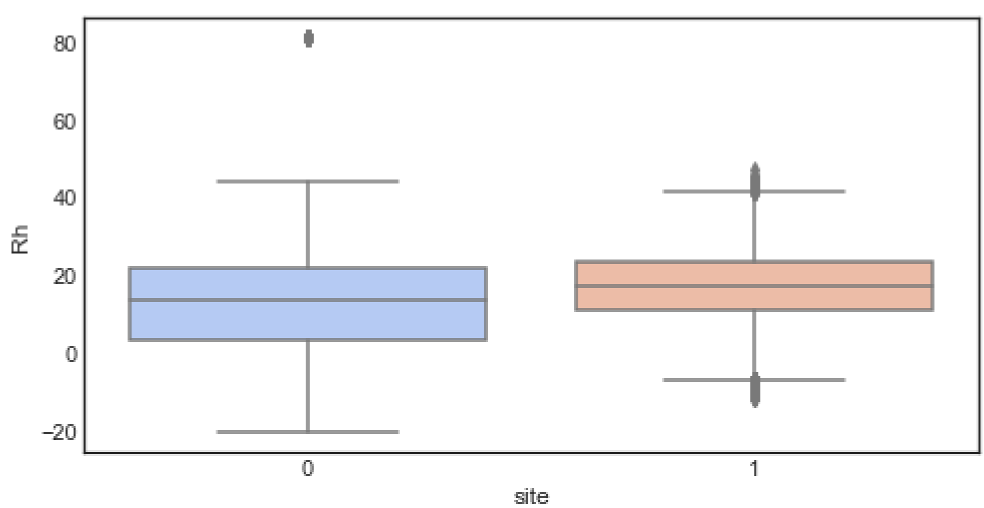
In [63]:
```
HfDf['site']=HfDf['site'].astype('category') #To convert target class into cat
egory

for i, col in enumerate(cont_data.columns):
    plt.figure(i,figsize=(8,4))
    sns.boxplot(x=HfDf['site'], y=col, data=HfDf, palette="coolwarm")
```



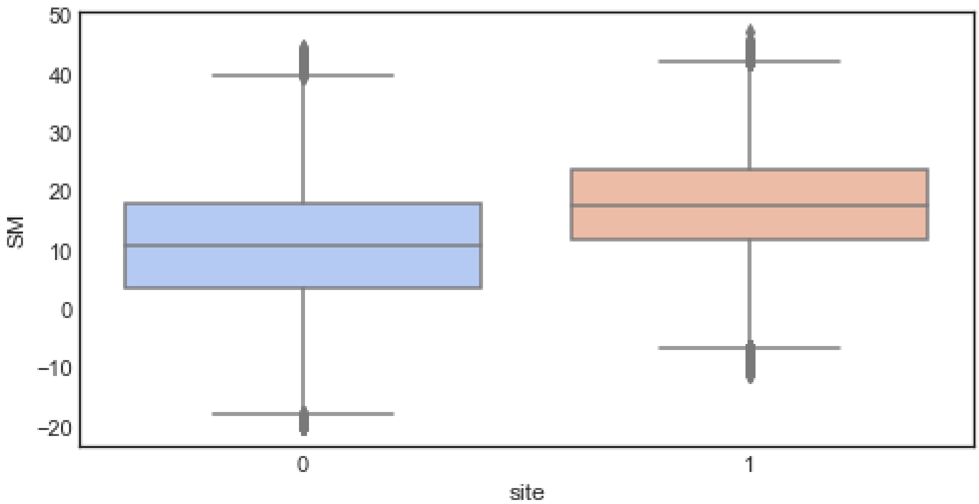
50
40
30
20
10
0
−10
−20
AT
0
1
site

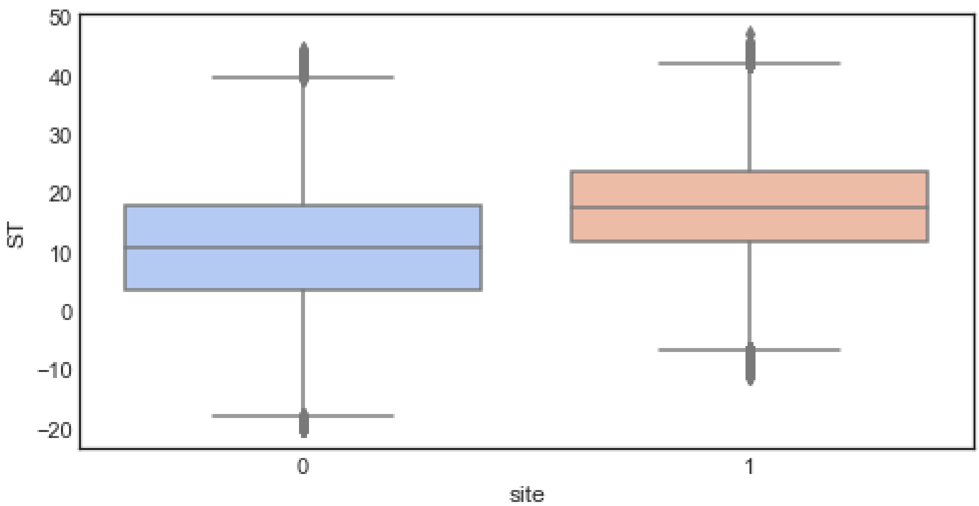


8000
6000
4000
2000
0
−2000
−4000
−6000
Q
0
1
site



80
60
40
20
0
−20
Rh
0
1
site





Comparing sites:

Harvard Forest 0

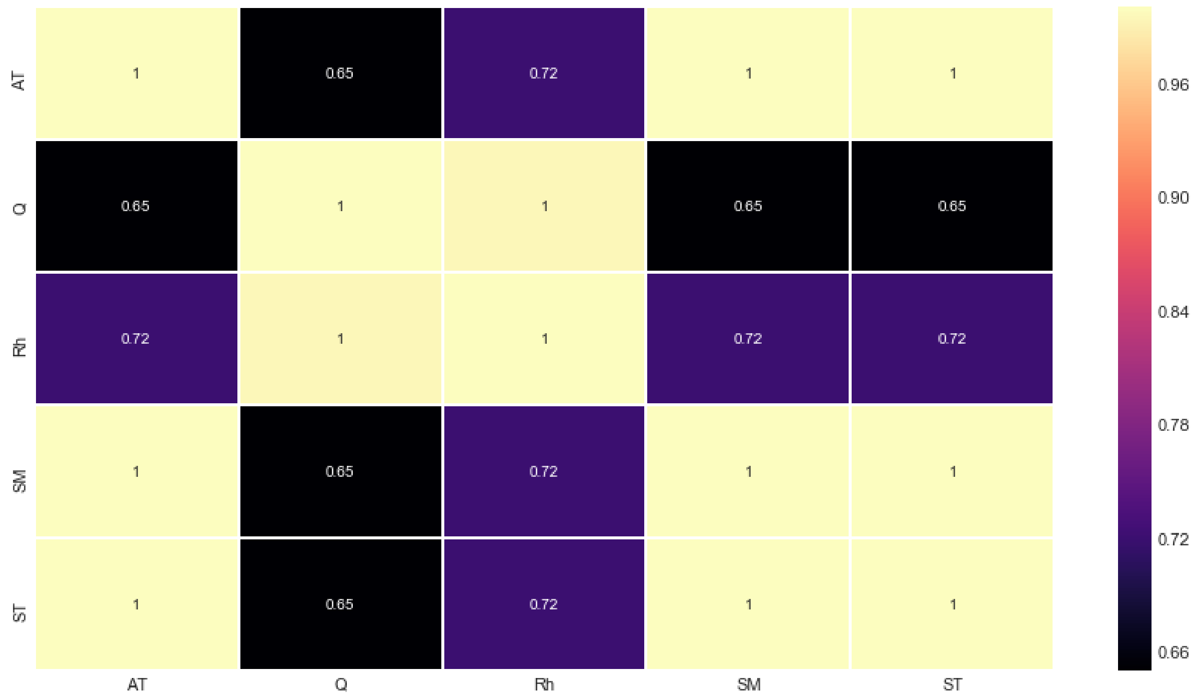
The barplots are all slightly lower for harvard forest. Which makes sense since it is 700 miles due north of Duke forest. What is interesting is that the Q (PAR) is higher in Harvard Forest. Perphas there was clear-cut forest area, downed trees, and younger trees studied.

Duke Forest 1

The barplots are all slightly higher for Duke forest. This make sense since it is more southernly in a warmer climate and more humid conditions. It is curious why the Q (PAR) was lower in the Duke forest. Was it due to instrumental readings? The time of year or month? Is the forest at Duke more dense and has less canopy gaps for light?

In [59]:
```
plt.figure(figsize=(15,8))
sns.heatmap(cont_data.corr(),cmap='magma',linecolor='white',linewidths=1,annot
=True)
```

Out[59]: <matplotlib.axes._subplots.AxesSubplot at 0x2aa7376d6a0>



Seaborn heatmap correlation

I suspect that the 1 to 1 correlation is due to variance of adding mean values into the dataset.

There is a relationship between AT (air temperature) and Q (PAR) as well as AT and Rh (Relative humdity)

Supervised Learning

In [60]:
```
%matplotlib inline
pd.options.display.float_format = '{:.3f}'.format

from sklearn import linear_model
from sklearn import preprocessing
import warnings
warnings.filterwarnings(action="ignore", module="scipy", message="^internal ge
lsd")
from mpl_toolkits.mplot3d import axes3d

from sklearn.preprocessing import scale
import sklearn.linear_model as skl_lm
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

In [61]:
```
# %load ../standard_import.txt
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

import sklearn.linear_model as skl_lm
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.metrics import confusion_matrix, classification_report, precision
_score
from sklearn import preprocessing
from sklearn import neighbors

import statsmodels.api as sm
import statsmodels.formula.api as smf

%matplotlib inline
plt.style.use('seaborn-white')
```

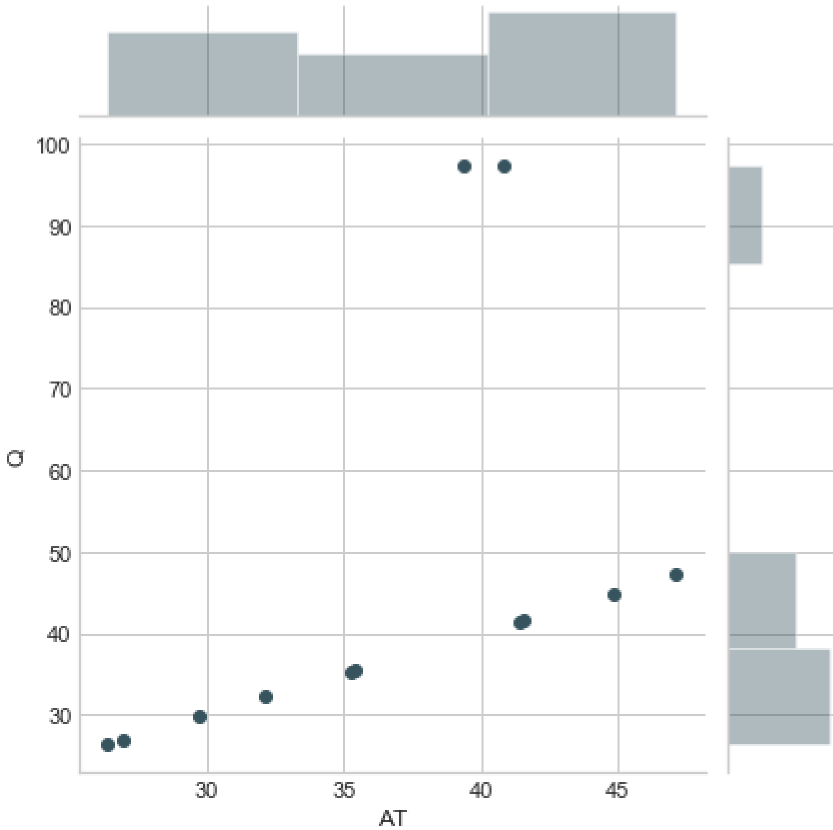
Linear regression

In [62]:
```
dmax = HfDf.groupby('month').max()
dmin = HfDf.groupby('month').min()
davg = HfDf.groupby('month').mean()
```

Plot of the monthly max of AT and Q

In [63]:
```
sns.set_palette("GnBu_d")
sns.set_style('whitegrid')

sns.jointplot(x='AT',y='Q',data=dmax) # Day with AT
```

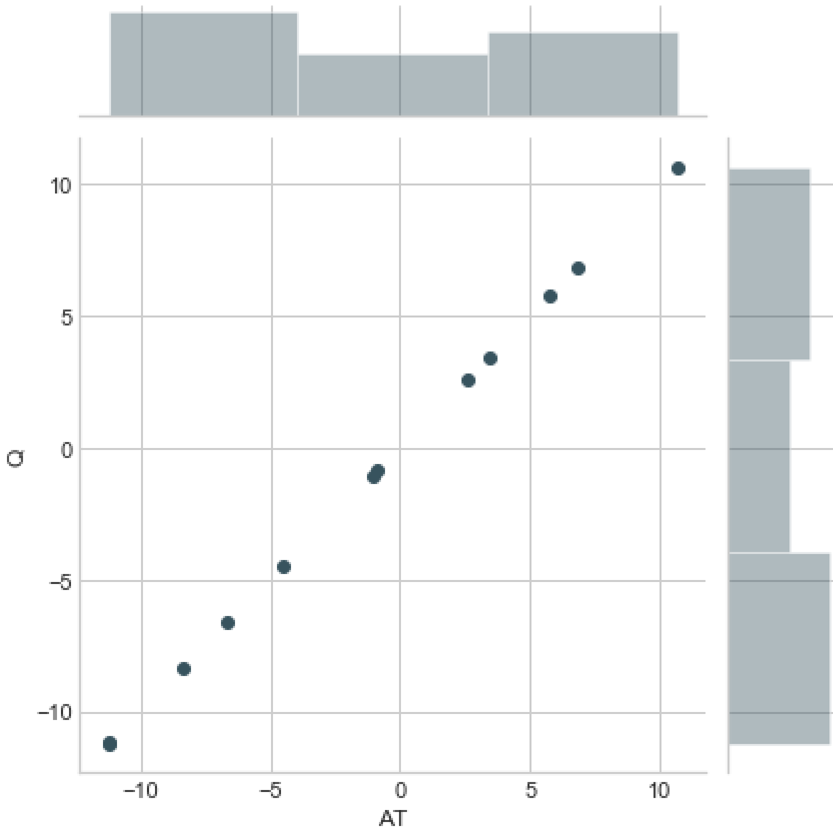
Out[63]: <seaborn.axisgrid.JointGrid at 0x2aa7376d470>



Plot of monthly min of AT and Q

In [64]:
```
sns.set_palette("GnBu_d")
sns.set_style('whitegrid')

sns.jointplot(x='AT',y='Q',data=dmin) # Day with AT
```

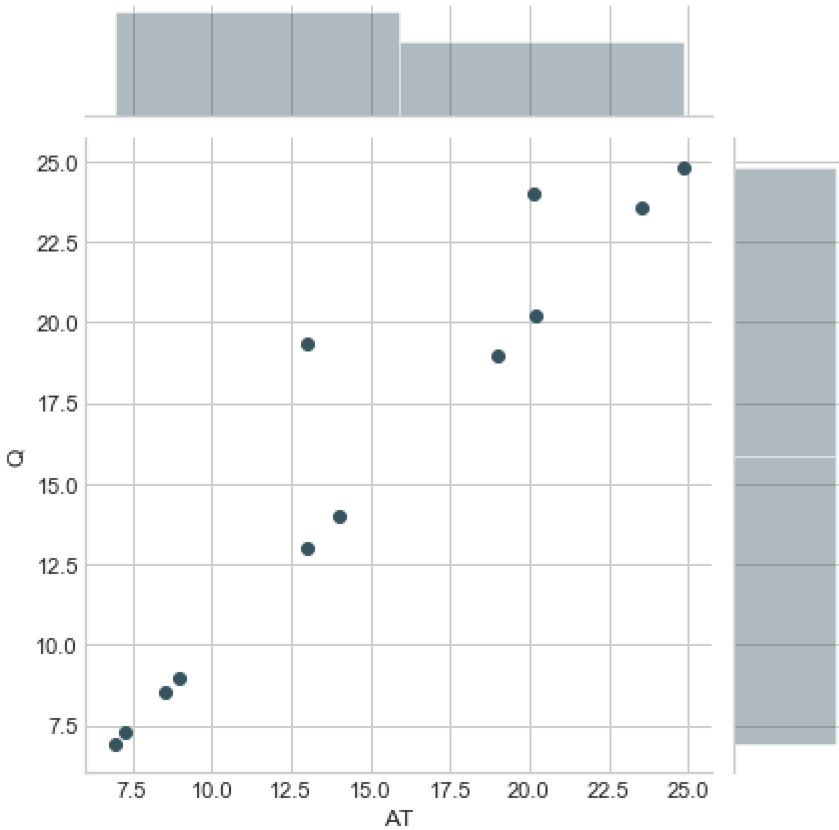
Out[64]: <seaborn.axisgrid.JointGrid at 0x2aa660ff550>



Plot of monthly mean of AT and Q

In [65]:
```
sns.set_palette("GnBu_d")
sns.set_style('whitegrid')

sns.jointplot(x='AT',y='Q',data=davg) # Day with AT
```

Out[65]: <seaborn.axisgrid.JointGrid at 0x2aa660a9fd0>



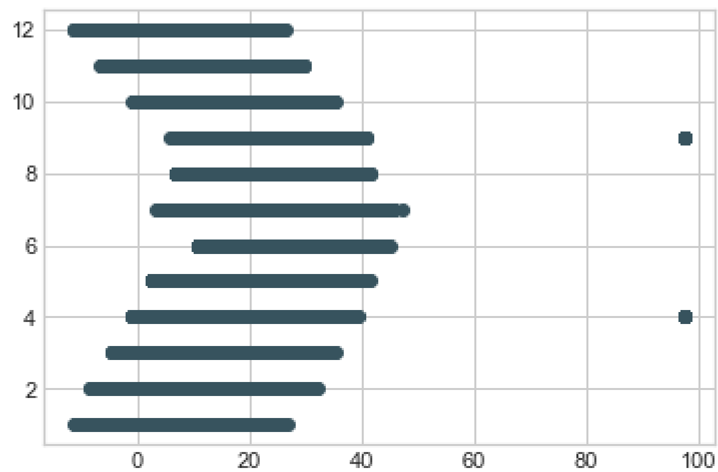
In all senarios of min, max, and mean there is a linear relationship between air temperature and par

In [66]:
```
print(HfDf.loc[HfDf['Q']==max(HfDf['Q']),:])

plt.scatter(HfDf['Q'],HfDf['month'])
plt.show()
```

```
               date  year month  day  site  chamber  treatment     AT      Q  \
750851  2010-04-30  2010     4   30     0        1          1 14.920 97.239
751023  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751024  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751025  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751026  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751027  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751028  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751029  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751030  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751031  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751032  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751033  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751034  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751035  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751036  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751037  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751038  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751039  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751040  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751041  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751042  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751043  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751044  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751045  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751046  2010-04-21  2010     4   21     0        4          0 14.920 97.239
751529  2010-04-30  2010     4   30     0        9          1 14.920 97.239
751749  2010-04-21  2010     4   21     0        3          0 14.920 97.239
751750  2010-04-21  2010     4   21     0        3          0 14.920 97.239
751751  2010-04-21  2010     4   21     0        3          0 14.920 97.239
751752  2010-04-21  2010     4   21     0        3          0 14.920 97.239
...            ...   ...   ...  ...   ...      ...        ...    ...    ...
2119693 2009-09-28  2009     9   28     0        4          1 14.920 97.239
2119694 2009-09-28  2009     9   28     0        4          1 14.920 97.239
2119695 2009-09-28  2009     9   28     0        4          1 14.920 97.239
2119696 2009-09-28  2009     9   28     0        4          1 14.920 97.239
2119697 2009-09-28  2009     9   28     0        4          1 14.920 97.239
2124869 2010-04-30  2010     4   30     0        6          1 14.920 97.239
2125209 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125210 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125211 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125212 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125213 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125214 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125215 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125216 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125217 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125218 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125219 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125220 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125221 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125222 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125223 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125224 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125225 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125226 2009-09-28  2009     9   28     0        6          1 14.920 97.239
```

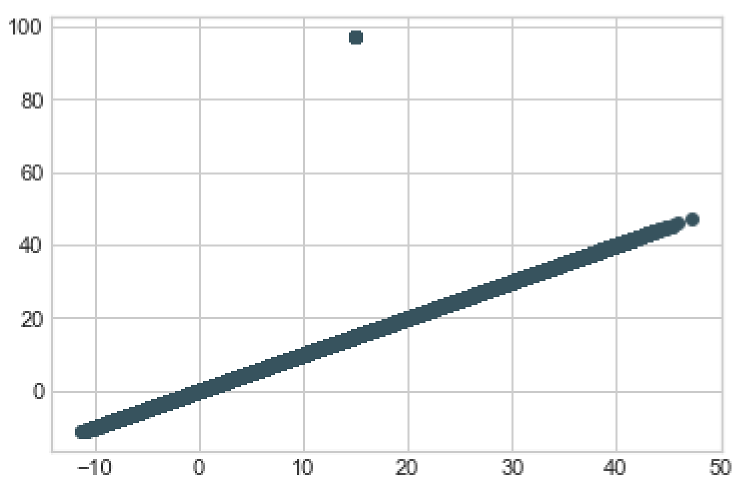
```
2125227 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125228 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125229 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125230 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125231 2009-09-28  2009     9   28     0        6          1 14.920 97.239
2125232 2009-09-28  2009     9   28     0        6          1 14.920 97.239

            Rh     SM     ST   ppfd  CO2_uptake  CO2_release
750851  82.969 14.920 14.920 14.920       False        False
751023  82.969 14.920 14.920 14.920       False        False
751024  82.969 14.920 14.920 14.920       False        False
751025  82.969 14.920 14.920 14.920       False        False
751026  82.969 14.920 14.920 14.920       False        False
751027  82.969 14.920 14.920 14.920       False        False
751028  82.969 14.920 14.920 14.920       False        False
751029  82.969 14.920 14.920 14.920       False        False
751030  82.969 14.920 14.920 14.920       False        False
751031  82.969 14.920 14.920 14.920       False        False
751032  82.969 14.920 14.920 14.920       False        False
751033  82.969 14.920 14.920 14.920       False        False
751034  82.969 14.920 14.920 14.920       False        False
751035  82.969 14.920 14.920 14.920       False        False
751036  82.969 14.920 14.920 14.920       False        False
751037  82.969 14.920 14.920 14.920       False        False
751038  82.969 14.920 14.920 14.920       False        False
751039  82.969 14.920 14.920 14.920       False        False
751040  82.969 14.920 14.920 14.920       False        False
751041  82.969 14.920 14.920 14.920       False        False
751042  82.969 14.920 14.920 14.920       False        False
751043  82.969 14.920 14.920 14.920       False        False
751044  82.969 14.920 14.920 14.920       False        False
751045  82.969 14.920 14.920 14.920       False        False
751046  82.969 14.920 14.920 14.920       False        False
751529  82.969 14.920 14.920 14.920       False        False
751749  82.969 14.920 14.920 14.920       False        False
751750  82.969 14.920 14.920 14.920       False        False
751751  82.969 14.920 14.920 14.920       False        False
751752  82.969 14.920 14.920 14.920       False        False
...        ...    ...    ...    ...         ...          ...
2119693 82.969 14.920 14.920 14.920       False        False
2119694 82.969 14.920 14.920 14.920       False        False
2119695 82.969 14.920 14.920 14.920       False        False
2119696 82.969 14.920 14.920 14.920       False        False
2119697 82.969 14.920 14.920 14.920       False        False
2124869 82.969 14.920 14.920 14.920       False        False
2125209 82.969 14.920 14.920 14.920       False        False
2125210 82.969 14.920 14.920 14.920       False        False
2125211 82.969 14.920 14.920 14.920       False        False
2125212 82.969 14.920 14.920 14.920       False        False
2125213 82.969 14.920 14.920 14.920       False        False
2125214 82.969 14.920 14.920 14.920       False        False
2125215 82.969 14.920 14.920 14.920       False        False
2125216 82.969 14.920 14.920 14.920       False        False
2125217 82.969 14.920 14.920 14.920       False        False
2125218 82.969 14.920 14.920 14.920       False        False
2125219 82.969 14.920 14.920 14.920       False        False
2125220 82.969 14.920 14.920 14.920       False        False
```

```
2125221 82.969 14.920 14.920 14.920       False        False
2125222 82.969 14.920 14.920 14.920       False        False
2125223 82.969 14.920 14.920 14.920       False        False
2125224 82.969 14.920 14.920 14.920       False        False
2125225 82.969 14.920 14.920 14.920       False        False
2125226 82.969 14.920 14.920 14.920       False        False
2125227 82.969 14.920 14.920 14.920       False        False
2125228 82.969 14.920 14.920 14.920       False        False
2125229 82.969 14.920 14.920 14.920       False        False
2125230 82.969 14.920 14.920 14.920       False        False
2125231 82.969 14.920 14.920 14.920       False        False
2125232 82.969 14.920 14.920 14.920       False        False

[28645 rows x 15 columns]
```



In [67]:
```
newdf = HfDf.loc[HfDf['Q']!=7643.000,:]
```

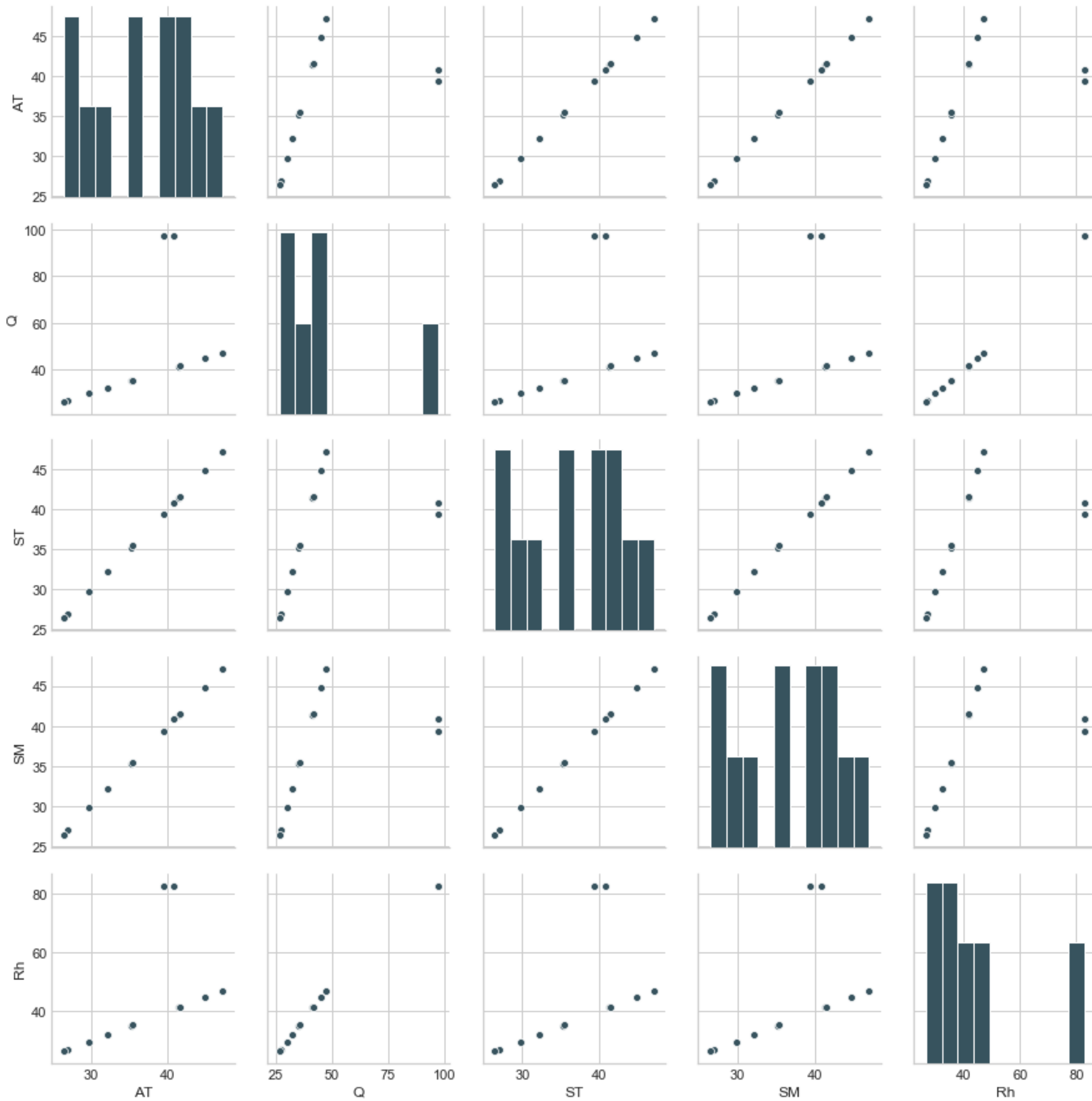
In [68]:
```
plt.scatter(newdf['AT'],newdf['Q'])
plt.show()
```



Seaborn pairplots using max, min, and mean for all the environmental variables

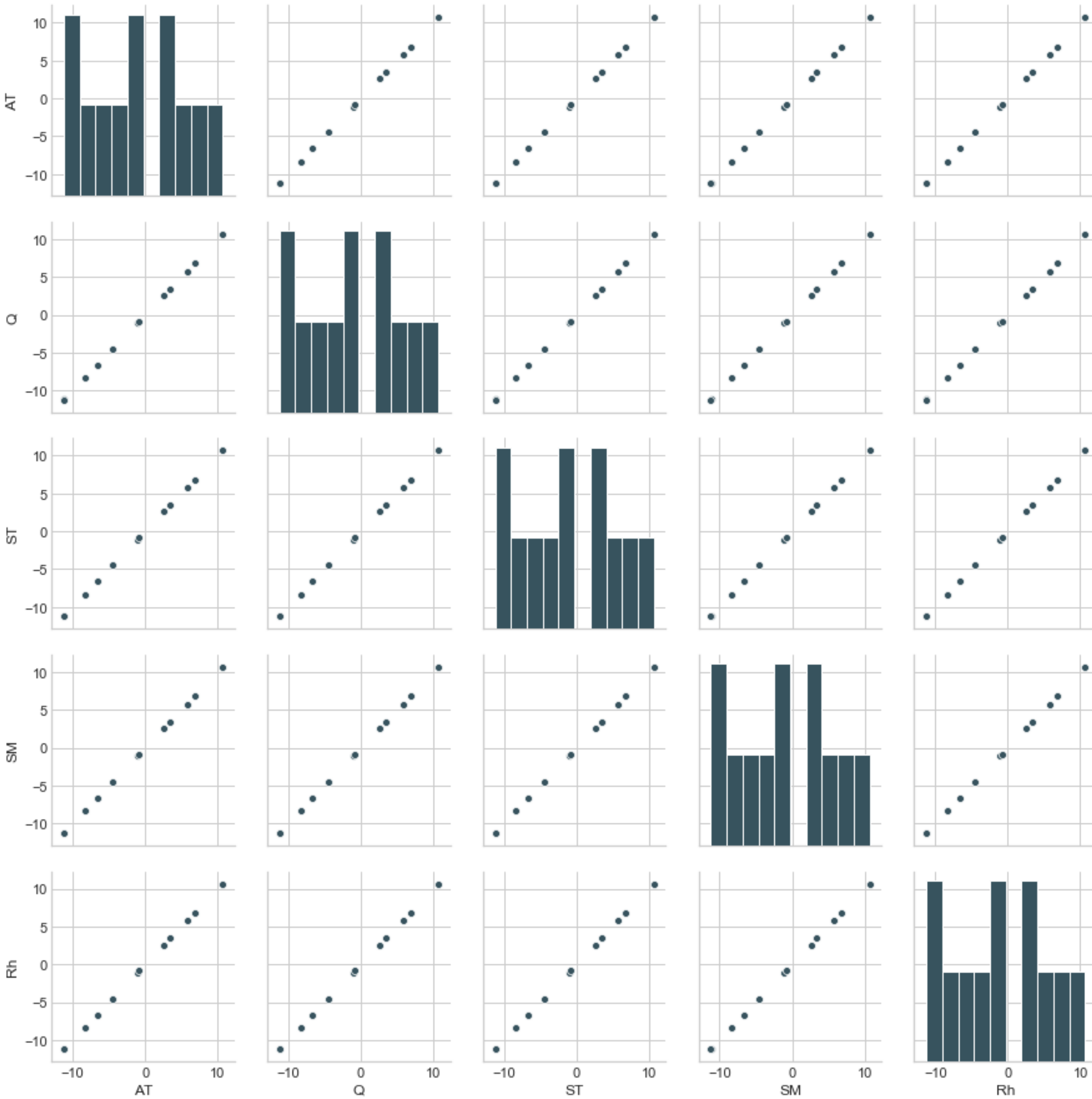
In [69]:
```
sns.pairplot(dmax[['AT', 'Q', 'ST', 'SM', 'Rh']]) #Max values
```

Out[69]: <seaborn.axisgrid.PairGrid at 0x2aa1dc17a58>



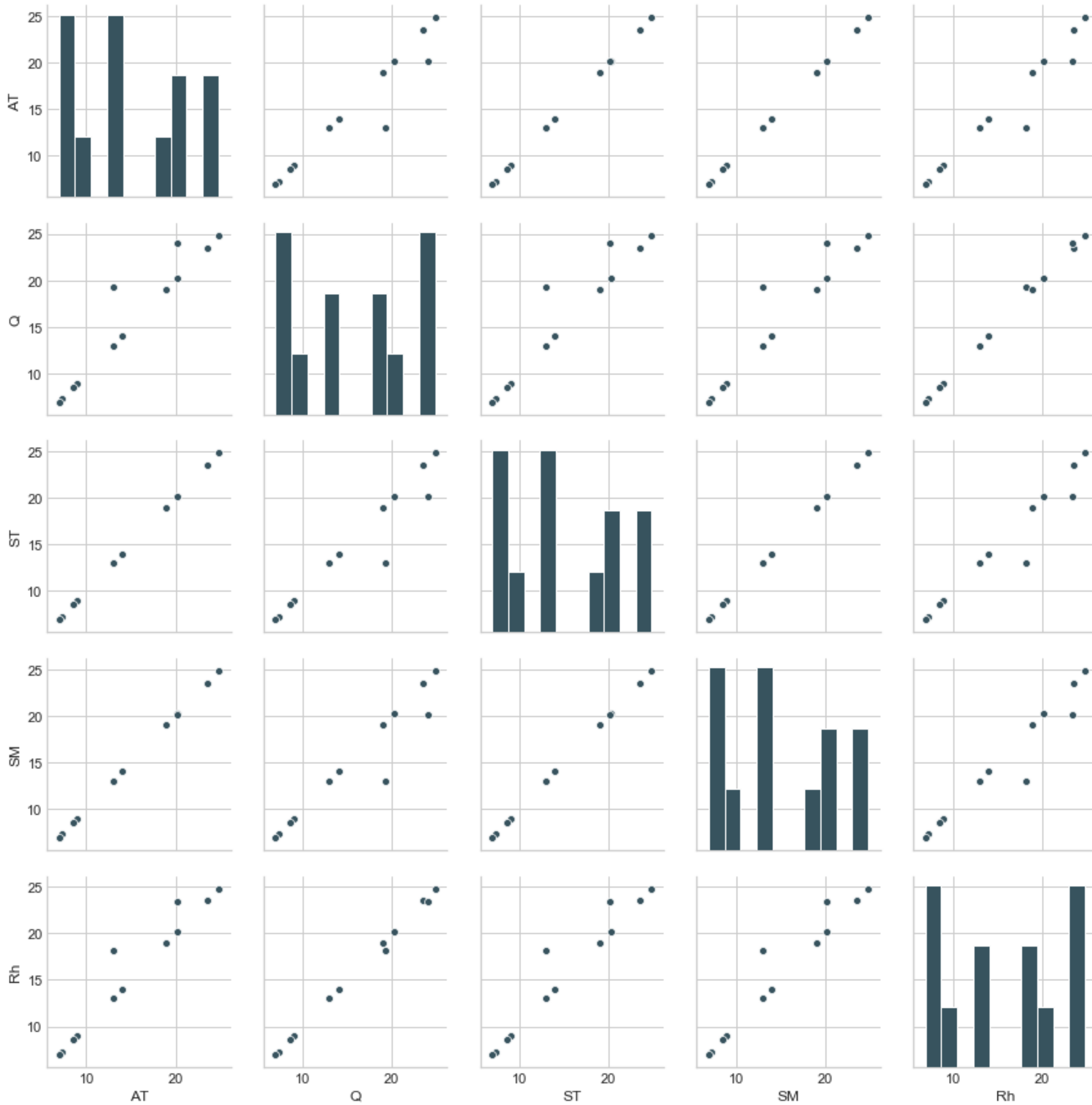
In [70]:
```
sns.pairplot(dmin[['AT', 'Q', 'ST', 'SM', 'Rh']]) #Min values
```

Out[70]: <seaborn.axisgrid.PairGrid at 0x2aa5d7ec2b0>



In [71]:
```
sns.pairplot(davg[['AT', 'Q', 'ST', 'SM', 'Rh']]) #Mean values
```

Out[71]: <seaborn.axisgrid.PairGrid at 0x2aa6e2b3da0>



Linear regression

Using Q (PAR) as the target variable (y).

In [87]:
```
y = davg['Q']
X = davg[['AT', 'Rh']]
```

In [88]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, rando
m_state=0)
from sklearn.linear_model import LinearRegression

lm = LinearRegression()
lm.fit(X_train,y_train)
```

Out[88]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

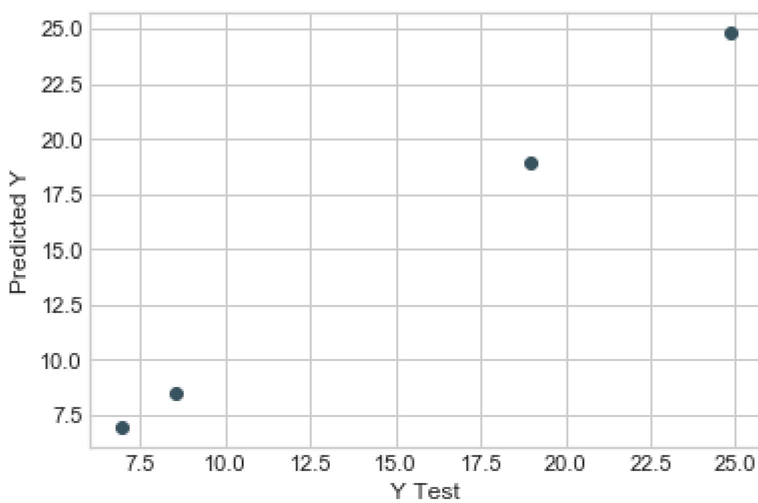
In [89]:
```
# The coefficients
print('Coefficients: \n', lm.coef_)
```

```
Coefficients:
 [-0.2097018  1.2097018]
```

In [90]:
```
predictions = lm.predict( X_test)
```

In [91]:
```
plt.scatter(y_test,predictions)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[91]:
```
Text(0,0.5,'Predicted Y')
```



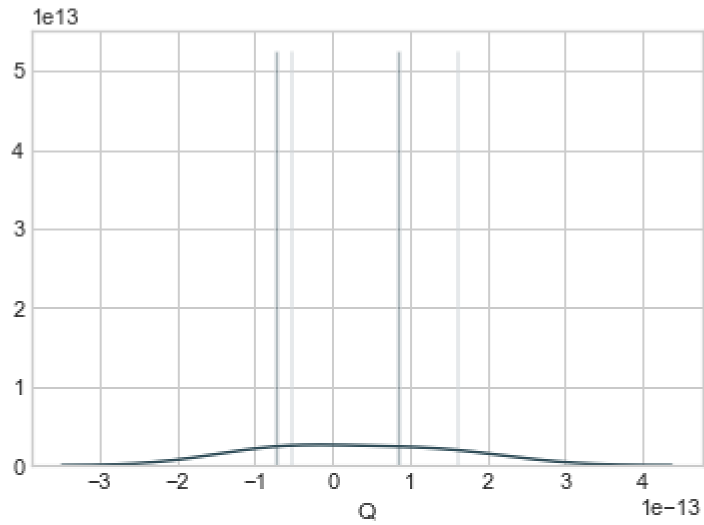
Scatter plot of our y_test by predictions is not great.

It does indicate that Q increases porportionally to AT and Rh.

In [92]:
```
from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 9.414691248821327e-14
MSE: 1.059597967892863e-26
RMSE: 1.0293677515314257e-13
```

In [93]:
```
sns.distplot((y_test-predictions),bins=50);
```



In [94]:
```
coeffecients = pd.DataFrame(lm.coef_,X.columns)
coeffecients.columns = ['Coeffecient']
coeffecients
```

Out[94]:

	Coeffecient
AT	-0.210
Rh	1.210

Linear regression model shows a 1.210 coefficient with Rh. This may be due to the two outliers.

Logistic Regression

In [95]:
```
HfDf.head(2)
```

Out[95]:

	date	year	month	day	site	chamber	treatment	AT	Q	Rh	SM	S
0	2009-05-06	2009	5	6	1	1	0	24.227	24.227	24.227	24.227	24.2
1	2009-05-06	2009	5	6	1	2	0	24.513	24.513	24.513	24.513	24.5

Does the light treatment have an effect?

treatment: light treatment 0 G: chamber in open gap 1 S: chamber under closed canopy

In [98]:
```
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
```

In [99]:
```
from sklearn.model_selection import train_test_split
```

In [101]:
```
X = HfDf.ix[:, 'chamber':'ST'].drop(columns=['treatment'])
y = HfDf['treatment']
```

In [102]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, rand
om_state=42)
```

In [103]:
```
from sklearn.linear_model import LogisticRegression
```

In [104]:
```
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
```

Out[104]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

In [105]:
```
predictions = logmodel.predict(X_test)
```

In [106]:
```
from sklearn.metrics import classification_report
```

In [107]:
```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.53      0.84      0.65    370200
           1       0.51      0.18      0.27    341922

   micro avg       0.52      0.52      0.52    712122
   macro avg       0.52      0.51      0.46    712122
weighted avg       0.52      0.52      0.47    712122
```

In [108]:
```
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
```

In [109]:
```
kfold = KFold(n_splits=7, random_state=42)
```

In [110]:
```
result = cross_val_score(logmodel, X, y, cv=kfold, scoring='accuracy')
print(result.mean())
```

```
0.4542956125438623
```

Does the light treatment have an effect?

Yes it does!

treatment: light treatment 0 G: chamber in open gap - analyzing the entire dataset open gap in the forest performs better. Meaning that trees respond better to this treatment.

1 S: chamber under closed canopy - Under closed canopy conditions both forest combined did not perform as well. Trees along the eastern coast need light.

Naive Bayes

Site - Forests

0 Harvard 1 Duke

In [117]:
```
data = HfDf.ix[:, 'AT':'ST']
target = HfDf['site']
```

In [118]:
```
from sklearn.naive_bayes import BernoulliNB
bnb = BernoulliNB()
y_pred = bnb.fit(data, target).predict(data)
```

In [119]:
```
# Test your model with different holdout groups.

from sklearn.model_selection import train_test_split
# Use train_test_split to create the necessary training and test groups
X_train, X_test, y_train, y_test = train_test_split(data, target, test_size=0.
2, random_state=20)
print('With 20% Holdout: ' + str(bnb.fit(X_train, y_train).score(X_test, y_tes
t)))
print('Testing on Sample: ' + str(bnb.fit(data, target).score(data, target)))
```

```
With 20% Holdout: 0.6571529858267936
Testing on Sample: 0.6567720678701264
```

In [120]:
```
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report,confusion_matrix
```

In [121]:
```
f1_score(target, y_pred, average='macro')
```

Out[121]: 0.4286899711147681

In [122]:
```
print(classification_report(target, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.66      0.99      0.79   1407177
           1       0.62      0.04      0.07    750768

   micro avg       0.66      0.66      0.66   2157945
   macro avg       0.64      0.51      0.43   2157945
weighted avg       0.64      0.66      0.54   2157945
```

Based on the data the Harvard Forest site is easier to predict then the Duke forest. This maybe due to many more data points. It also could be related to how chambers and treatments could be differnt enough in both forests. This clearly says these forest are different based on our environmental variables used.

Random Forest

In [123]:
```
#Supervised Learning imports

from sklearn import ensemble
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.linear_model import SGDClassifier
from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import learning_curve
from sklearn import svm
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold, cross_val_score
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
```

In [124]:
```
df.head(5)
```

Out[124]:

	Unnamed: 0	date	Species	Tag	Chamber	site	year	month	day	hour	chamber	tr
0	0	2010-04-15	acru	1.000	S01_5	HF	2010	4	15	1	1	No
1	1	2010-04-15	acru	1.000	S01_5	HF	2010	4	15	2	1	No
2	2	2010-04-15	acru	1.000	S01_5	HF	2010	4	15	3	1	No
3	3	2010-04-15	acru	1.000	S01_5	HF	2010	4	15	4	1	No
4	4	2010-04-15	acru	1.000	S01_5	HF	2010	4	15	5	1	No

In [125]:
```
df.dtypes
```

Out[125]:
```
Unnamed: 0              int64
date           datetime64[ns]
Species                object
Tag                   float64
Chamber                object
site                   object
year                    int64
month                category
day                     int64
hour                    int64
chamber                 int64
treatment              object
warming               float64
AT                    float64
Q                     float64
Rh                    float64
SM                    float64
ST                    float64
ppfd                   object
CO2_uptake               bool
CO2_release              bool
dtype: object
```

In [126]:
```
df.groupby('warming').size()
```

Out[126]:
```
warming
3.000    371529
5.000    420708
dtype: int64
```

In [127]:
```
df.isnull().sum()
```

Out[127]:
```
Unnamed: 0           0
date                 0
Species              0
Tag                  0
Chamber              0
site                 0
year                 0
month                0
day                  0
hour                 0
chamber              0
treatment      1407177
warming         614940
AT                   0
Q                    0
Rh                   0
SM                   0
ST                   0
ppfd                 0
CO2_uptake           0
CO2_release          0
dtype: int64
```

In [128]:
```
df['treatment'] = df['treatment'].apply({ 'G': 0, 'S': 1}.get)
df['treatment'] = df['treatment'].apply({ 'C': 0, 'A': 1, '3': 3, '5': 5}.get)
```

In [129]:
```
df["Rh"] = df.Rh.convert_objects(convert_numeric=True)
df["Q"] = df.Q.convert_objects(convert_numeric=True)
df["SM"] = df.SM.convert_objects(convert_numeric=True)
df["ST"] = df.ST.convert_objects(convert_numeric=True)
df["AT"] = df.AT.convert_objects(convert_numeric=True)

df["chamber"] = df.chamber.convert_objects(convert_numeric=True)
df["treatment"] = df.treatment.convert_objects(convert_numeric=True)
df["warming"] = df.warming.convert_objects(convert_numeric=True)
```

In [130]:
```
df.dtypes
```

Out[130]:
```
Unnamed: 0              int64
date           datetime64[ns]
Species                object
Tag                   float64
Chamber                object
site                   object
year                    int64
month                category
day                     int64
hour                    int64
chamber                 int64
treatment              object
warming               float64
AT                    float64
Q                     float64
Rh                    float64
SM                    float64
ST                    float64
ppfd                   object
CO2_uptake               bool
CO2_release              bool
dtype: object
```

Random Forest - Harvard Forest with tree species

Too many missing values for treatment and warming - both needed to be dropped.

In [132]:
```
from sklearn import ensemble
from sklearn.model_selection import train_test_split

X = df.ix[:, 'year':'ST'].drop(columns=['treatment', 'warming'])
Y = df['Species']

rfc = ensemble.RandomForestClassifier()

X_train, X_test, y_train, y_test = train_test_split(X, Y,test_size=0.1,random_
state=0)
train = rfc.fit(X_train, y_train)

rfc.fit(X_train, y_train)

# Apply to testing data
y_hat = rfc.predict(X_test)


print('Training set score:', rfc.score(X_train, y_train))
print('\nTest set score:', rfc.score(X_test, y_test))
print(classification_report(y_test, y_hat))
```

```
Training set score: 0.44117259224341254

Test set score: 0.42928410011512386
              precision    recall  f1-score   support

        acba       0.00      0.00      0.00       680
        acru       0.53      0.86      0.66     56089
        acsa       0.30      0.01      0.01      1438
        acun       0.00      0.00      0.00       154
        beal       0.20      0.20      0.20      3090
        bele       0.17      0.05      0.08      6651
        bepa       0.18      0.20      0.19      7270
        beun       0.29      0.61      0.39     10244
        cagl       0.00      0.00      0.00         6
        fagr       0.00      0.00      0.00       633
        fram       0.20      0.02      0.04      1171
        ilvo       0.00      0.00      0.00        32
        list       0.27      0.02      0.04      3102
        litu       0.00      0.00      0.00      4840
        magr       0.00      0.00      0.00      1218
        maun       0.00      0.00      0.00        64
        mavi       0.00      0.00      0.00       385
        nysy       0.00      0.00      0.00      1587
        pipa       0.00      0.00      0.00       214
        pire       0.34      0.01      0.03      4206
        pist       0.40      0.11      0.17     13903
        pita       0.00      0.00      0.00      1051
        piun       0.31      0.15      0.20      3816
        prpe       0.00      0.00      0.00       296
        prse       0.08      0.01      0.01       268
        qual       0.15      0.14      0.15      8629
        qufa       0.00      0.00      0.00       215
        quni       0.00      0.00      0.00       544
        quph       0.00      0.00      0.00       118
        quru       0.31      0.05      0.08      7326
        quun       0.00      0.00      0.00       775
        quve       0.00      0.00      0.00       338
        ulam       0.00      0.00      0.00         6
        unkn       0.00      0.00      0.00       359

   micro avg       0.43      0.43      0.43    140718
   macro avg       0.11      0.07      0.07    140718
weighted avg       0.35      0.43      0.35    140718
```

Species: species code. Most ambiguous individuals died before they grew large enough for positive identification. **Only including the highest values in discussion f1-scores**

acba: Acer barbatum, southern sugar maple

acru: Acer rubrum, red maple - 0.66 It is a wide spread dicideous tree across north earstern US. And can handle a wide range of temperature.

acsa: Acer saccharum, sugar maple acun: Acer spp, ambiguous maple

beal: Betula alleghaniensis, yellow birch - 0.20 large and important lumber species of birch native to North-eastern North America. Its native range extends from Newfoundland to Prince Edward Island, Nova Scotia, New Brunswick, southern Quebec and Ontario, and the southeast corner of Manitoba in Canada, west to Minnesota, and south in the Appalachian Mountains to northern Georgia.[20] While its range extends as far south as Georgia, it is most abundant in the northern part of its range. In southern Pennsylvania, it is rare and generally only found along bodies of water in cool, mature woods, and it only occurs at high elevations from Maryland southward.[5][10] It grows in USDA zones 3-7.[5] B. alleghaniensis prefers to grow in cooler conditions and is often found on north facing slopes, swamps, stream banks, and rich woods.[4][21] It does not grow well in dry regions or regions with hot summers and will often last only 30-50 years in such conditions. It grows soil pH ranging from 4-8.[9]

bele: Betula lenta, black birch bepo: Betula populifolia, gray birch

bepa: Betula papyrifera, paper birch - 0.19 Betula papyrifera is mostly confined to Canada and the far northern United States. It is found in interior (var. humilus) and south-central (var. kenaica) Alaska and in all provinces and territories of Canada, except Nunavut, as well as the far northern continental United States. Isolated patches are found as far south as the Hudson Valley of New York and Pennsylvania, as well as Washington, D.C. High elevation stands are also in mountains to North Carolina, New Mexico, and Colorado. The most southerly stand in the Western United States is located in Long Canyon in the City of Boulder Open Space and Mountain Parks. This is an isolated Pleistocene relict that most likely reflects the southern reach of boreal vegetation into the area during the last Ice Age.[9]

beun: Betula spp, ambiguous birch cagl: Carya glabra, pignut hickory caov: Carya ovata, shagbark hickory fagr: Fagus grandifolia, beech fram: Fraxinus americana, white ash ilvo: Ilix vomitoria, yaupon holly list: Liquidambar styraciflua, sweetgum litu: Liriodendron tulipifera, tulip tree; tulip poplar magr: Magnolia grandiflora, southern magnolia mavi: Magnolia virginiana, sweetbay magnolia nysy: Nyssa sylvatica, black gum; sour gum pipa: Pinus palustris, longleaf pine pire: Pinus resinosa, red pine

pist: Pinus strobus, white pine - 0.17 Pinus strobus is found in the nearctic temperate broadleaf and mixed forests biome of eastern North America. It prefers well-drained or sandy soils and humid climates, but can also grow in boggy areas and rocky highlands. In mixed forests, this dominant tree towers over many others, including some of the large broadleaf hardwoods. It provides food and shelter for numerous forest birds, such as the red crossbill, and small mammals such as squirrels. Eastern white pine forests originally covered much of north-central and north-eastern North America. Only one percent of the old-growth forests remain after the extensive logging operations of the 18th century to early 20th century.

pita: Pinus taeda, loblolly pine

piun: Pinus spp, ambiguous pine - 0.20

prse: Prunus serotina, black cherry prpe: Prunus pensylvanica, pin cherry

qual: Quercus alba, white oak - 0.15 Q. alba is fairly tolerant of a variety of habitats, and may be found on ridges, in valleys, and in between, in dry and moist habitats, and in moderately acid and alkaline soils. It is mainly a lowland tree, but reaches altitudes of 5,249 ft in the Appalachian Mountains. It is often a component of the forest canopy in an oak-heath forest.[13][14] Frequent fires in the Central Plains region of the United States prevented oak forests, including Q. alba, from expanding into the Midwest. However, a decrease in the frequency of these natural fires after European settlement caused rapid expansion of oak forests into the Great Plains, negatively affecting the natural prairie vegetation [15].

qufa: Quercus falcata, southern red oak; Spanish oak quni: Quercus nigra, water oak quph: Quercus phellos, willow oak quru: Quercus rubra, red oak quve: Quercus velutina, black oak quun: Quercus spp, ambiguous oak ulam: Ulmus americana, American elm unkn: Unidentified tree

It is interesting that the tree species with the highest f1-scores are all very northern tree species. It seems that these trees favor the warming under the experimental conditions. More research is need to tease out what is going on here?

In [134]:
```
from sklearn import ensemble
from sklearn.model_selection import train_test_split

X = HfDf.ix[:, 'year':'ST'].drop(columns=['treatment'])
Y = HfDf['treatment']

rfc = ensemble.RandomForestClassifier()

X_train, X_test, y_train, y_test = train_test_split(X, Y,test_size=0.1,random_
state=0)
train = rfc.fit(X_train, y_train)

rfc.fit(X_train, y_train)

# Apply to testing data
y_hat = rfc.predict(X_test)


print('Training set score:', rfc.score(X_train, y_train))
print('\nTest set score:', rfc.score(X_test, y_test))
print(classification_report(y_test, y_hat))
```

```
Training set score: 0.9754998326596813

Test set score: 0.838022196992516
              precision    recall  f1-score   support

           0       0.83      0.87      0.85    112174
           1       0.85      0.81      0.83    103621

   micro avg       0.84      0.84      0.84    215795
   macro avg       0.84      0.84      0.84    215795
weighted avg       0.84      0.84      0.84    215795
```

Does the light treatment have an effect?

treatment: light treatment 0 G: chamber in open gap 1 S: chamber under closed canopy

Random Forest shows that light treatment has an effect but the difference between treatments is not as extreme as Logistic Regression showed.

Unsupervised - PCA

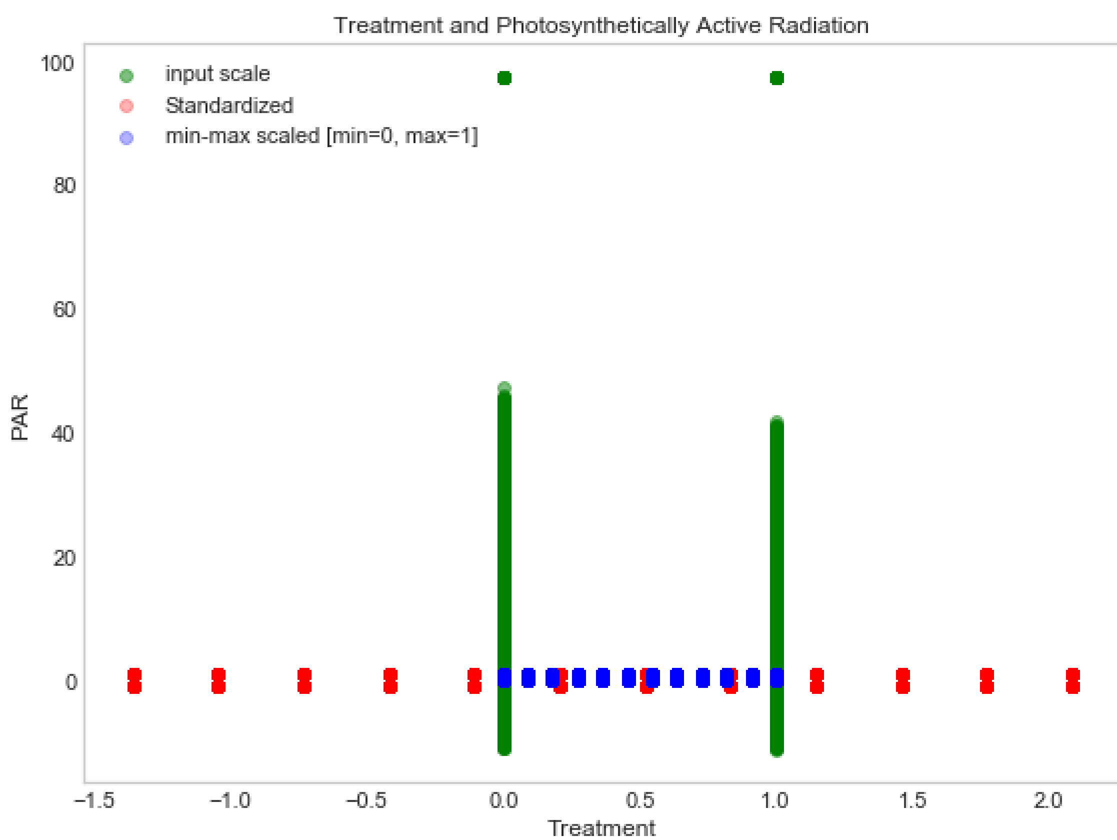
In [135]:
```
HfDf.head(3)
```

Out[135]:

	date	year	month	day	site	chamber	treatment	AT	Q	Rh	SM	S
0	2009-05-06	2009	5	6	1	1	0	24.227	24.227	24.227	24.227	24.2
1	2009-05-06	2009	5	6	1	2	0	24.513	24.513	24.513	24.513	24.5
2	2009-05-06	2009	5	6	1	3	0	24.070	24.070	24.070	24.070	24.0

In [138]:
```
from sklearn import preprocessing

std_scale = preprocessing.StandardScaler().fit(HfDf[['chamber', 'treatment', 'AT',
                                                     'Q', 'Rh', 'SM', 'ST']])
df_std = std_scale.transform(HfDf[['chamber', 'treatment', 'AT',
                                                     'Q', 'Rh', 'SM', 'ST']])

minmax_scale = preprocessing.MinMaxScaler().fit(HfDf[['chamber', 'treatment', 'AT',
                                                     'Q', 'Rh', 'SM', 'ST']])
df_minmax = minmax_scale.transform(HfDf[['chamber', 'treatment', 'AT',
                                                     'Q', 'Rh', 'SM', 'ST']])
```

In [140]:
```
from matplotlib import pyplot as plt

def plot():
    plt.figure(figsize=(8,6))

    plt.scatter(HfDf['treatment'], HfDf['Q'],
            color='green', label='input scale', alpha=0.5)

    plt.scatter(df_std[:,0], df_std[:,1], color='red',
            label='Standardized', alpha=0.3)

    plt.scatter(df_minmax[:,0], df_minmax[:,1],
            color='blue', label='min-max scaled [min=0, max=1]', alpha=0.3)

    plt.title('Treatment and Photosynthetically Active Radiation')
    plt.xlabel('Treatment')
    plt.ylabel('PAR')
    plt.legend(loc='upper left')
    plt.grid()

    plt.tight_layout()

plot()
plt.show()
```

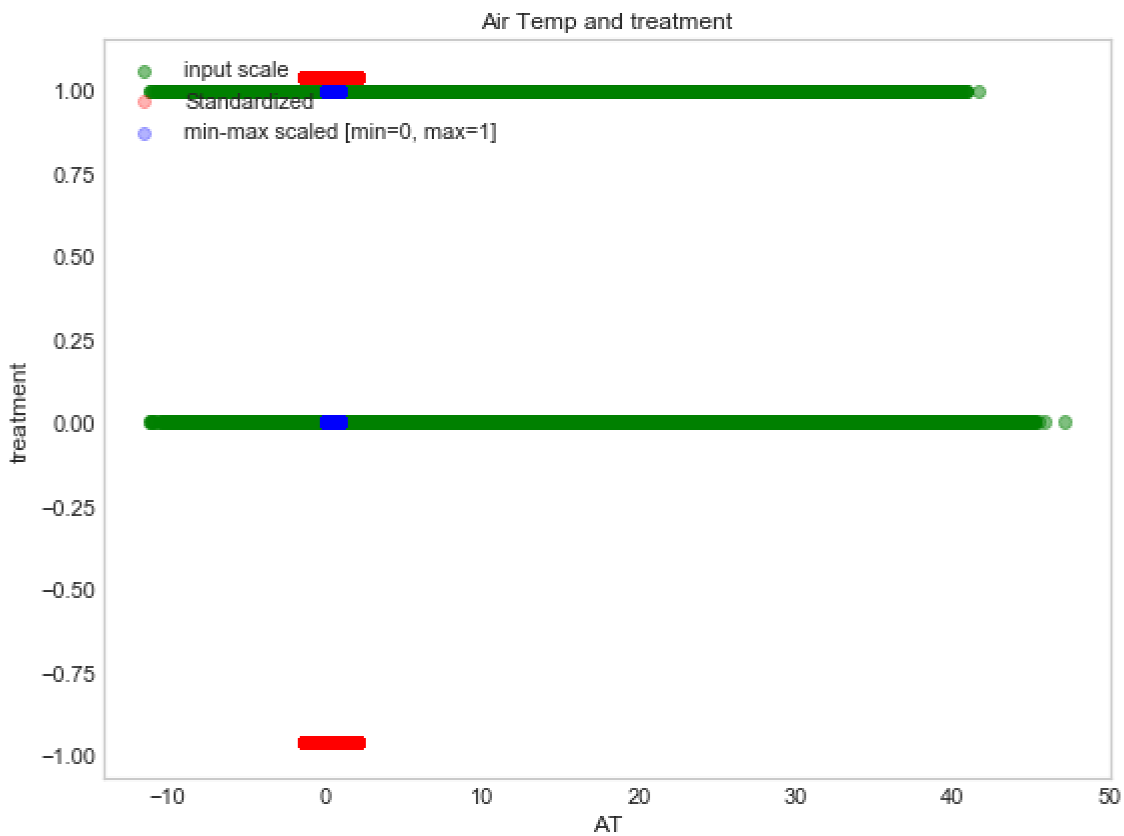


PCA of Treatment and Q (PAR)

It looks like a fieldgoal post.

It does show a difference in green lines of Q between both treatments.

In [144]:
```
from matplotlib import pyplot as plt

def plot():
    plt.figure(figsize=(8,6))

    plt.scatter(HfDf['AT'], HfDf['treatment'],
            color='green', label='input scale', alpha=0.5)

    plt.scatter(df_std[:,0], df_std[:,1], color='red',
            label='Standardized', alpha=0.3)

    plt.scatter(df_minmax[:,0], df_minmax[:,1],
            color='blue', label='min-max scaled [min=0, max=1]', alpha=0.3)

    plt.title('Air Temp and treatment')
    plt.xlabel('AT')
    plt.ylabel('treatment')
    plt.legend(loc='upper left')
    plt.grid()

    plt.tight_layout()

plot()
plt.show()
```



Not a great figure. But does show the differences in AT with treatment.

In [145]:
```
HfDf.head(3)
```

Out[145]:

	date	year	month	day	site	chamber	treatment	AT	Q	Rh	SM	S
0	2009-05-06	2009	5	6	1	1	0	24.227	24.227	24.227	24.227	24.2
1	2009-05-06	2009	5	6	1	2	0	24.513	24.513	24.513	24.513	24.5
2	2009-05-06	2009	5	6	1	3	0	24.070	24.070	24.070	24.070	24.0

In [164]:
```
trees3 = HfDf.drop(columns=['date', 'year', 'month','day', 'site', 'chamber',
'treatment', 'ppfd', 'CO2_uptake', 'CO2_release'])
```

In [165]:
```
trees3.head(1)
```

Out[165]:

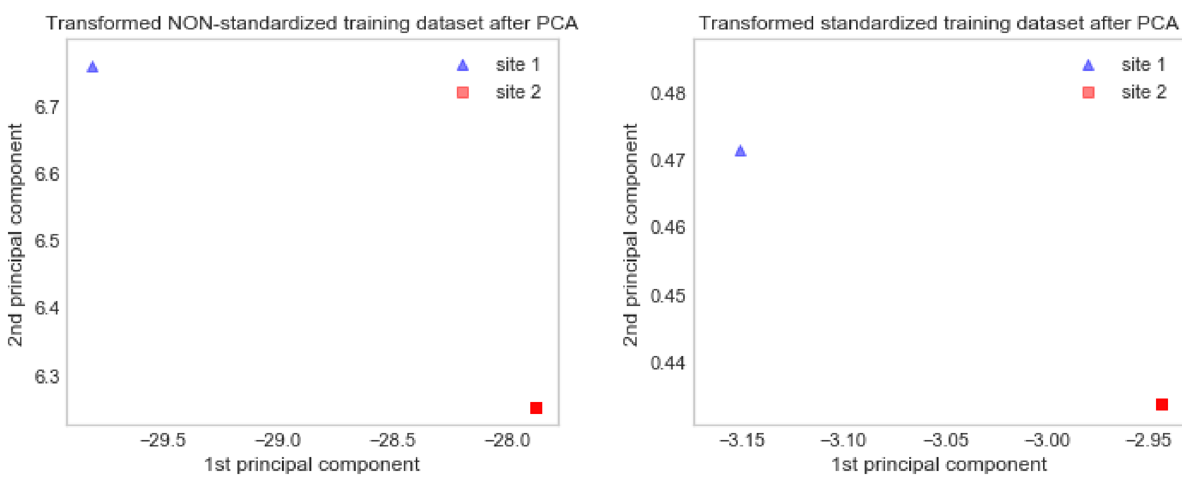
	AT	Q	Rh	SM	ST
0	24.227	24.227	24.227	24.227	24.227

In [166]:
```
X_forest = trees3.values[:,1:]
y_forest = trees3.values[:,0]

X_train, X_test, y_train, y_test = train_test_split(X_forest, y_forest,
    test_size=0.90, random_state=12345)
```

In [167]:
```
from sklearn import preprocessing

std_scale = preprocessing.StandardScaler().fit(X_train)
X_train_std = std_scale.transform(X_train)
X_test_std = std_scale.transform(X_test)
```

In [168]:
```
from sklearn.decomposition import PCA

# on non-standardized data
pca = PCA(n_components=2).fit(X_train)
X_train = pca.transform(X_train)
X_test = pca.transform(X_test)


# om standardized data
pca_std = PCA(n_components=2).fit(X_train_std)
X_train_std = pca_std.transform(X_train_std)
X_test_std = pca_std.transform(X_test_std)
```

In [169]:
```
from matplotlib import pyplot as plt

fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(10,4))


for l,c,m in zip(range(1,10), ('blue', 'red'), ('^', 's')):
    ax1.scatter(X_train[y_train==l, 0], X_train[y_train==l, 1],
        color=c,
        label='site %s' %l,
        alpha=0.5,
        marker=m
        )

for l,c,m in zip(range(1,10), ('blue', 'red'), ('^', 's')):
    ax2.scatter(X_train_std[y_train==l, 0], X_train_std[y_train==l, 1],
        color=c,
        label='site %s' %l,
        alpha=0.5,
        marker=m
        )

ax1.set_title('Transformed NON-standardized training dataset after PCA')
ax2.set_title('Transformed standardized training dataset after PCA')

for ax in (ax1, ax2):

    ax.set_xlabel('1st principal component')
    ax.set_ylabel('2nd principal component')
    ax.legend(loc='upper right')
    ax.grid()
plt.tight_layout()

plt.show()
```



PCA does show a clear difference between site 1 and 2 (Harvard and Duke forests).

Time Series Machine Learning

In [170]:
```
import warnings
import itertools
import numpy as np
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
import pandas as pd
import statsmodels.api as sm
import matplotlib
matplotlib.rcParams['axes.labelsize'] = 14
matplotlib.rcParams['xtick.labelsize'] = 12
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'

#https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-
and-forecasting-with-python-4835e6bf050b
```

In [171]:
```
import numpy as np
import pandas as pd
import scipy
from datetime import datetime
import datetime as dt
import matplotlib.pyplot as plt
%matplotlib inline
```

In [261]:
```
df3.head(3)
```

Out[261]:

	Unnamed: 0	date	Chamber	site	year	month	day	chamber	treatment	AT	(
0	0	2010-04-15	S01_5	HF	2010	4	15	1	S	11.000	11.00
1	1	2010-04-15	S01_5	HF	2010	4	15	1	S	11.040	11.04
2	2	2010-04-15	S01_5	HF	2010	4	15	1	S	10.415	10.41

In [263]:
```
ts1 = df3.drop(columns=['Unnamed: 0', 'Chamber', 'site', 'year', 'month','day'
, 'chamber', 'treatment', 'ppfd', 'CO2_uptake', 'CO2_release'])
```

In [264]:
```
ts1.head(1)
```

Out[264]:

	date	AT	Q	Rh	SM	ST
0	2010-04-15	11.000	11.000	11.000	11.000	11.000

Time Series for AT - Harvard Forest

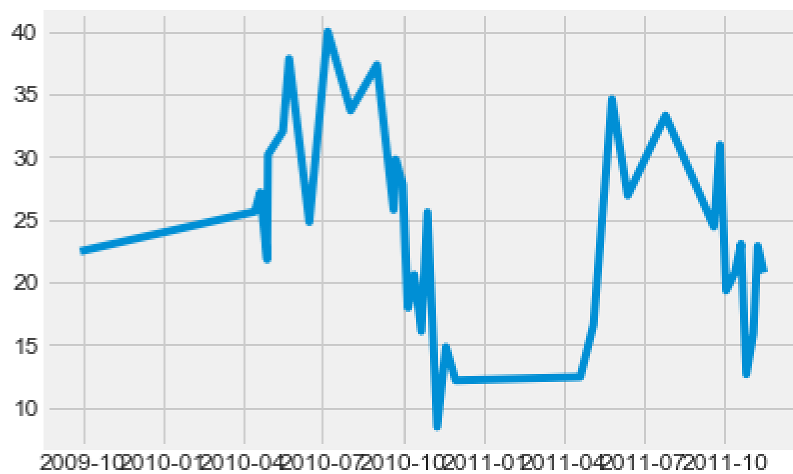
In [265]:
```
from datetime import datetime
con=ts1['date']
ts1['date']=pd.to_datetime(ts1['date'])
ts1.set_index('date', inplace=True)
#check datatype of index
ts1.index
```

Out[265]:
```
DatetimeIndex(['2010-04-15', '2010-04-15', '2010-04-15', '2010-04-15',
               '2010-04-15', '2010-04-15', '2010-04-15', '2010-04-15',
               '2010-04-15', '2010-04-15',
               ...
               '2010-11-29', '2010-11-29', '2010-11-29', '2010-11-29',
               '2010-11-29', '2010-11-29', '2010-11-29', '2010-11-29',
               '2010-11-29', '2010-11-29'],
              dtype='datetime64[ns]', name='date', length=1407177, freq=None)
```

In [231]:
```
ts=df.groupby(['date'], sort=True)['AT'].max()
```

In [232]:
```
plt.plot(ts)
```

Out[232]:
```
[<matplotlib.lines.Line2D at 0x2aa8e158ba8>]
```

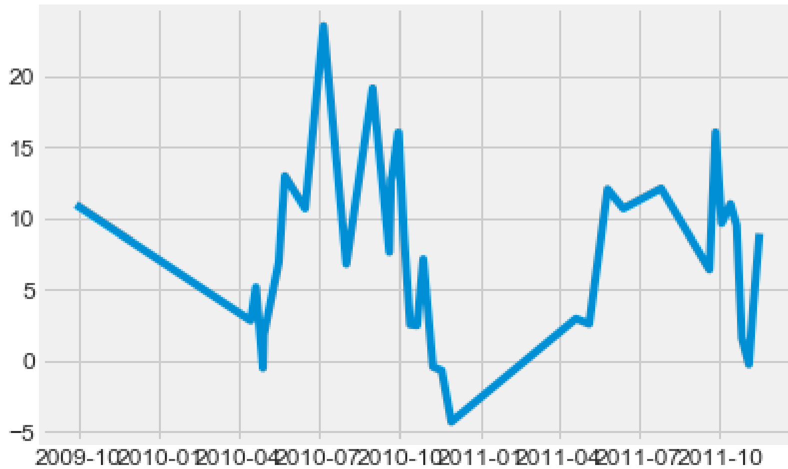


Time series plot of air temperature MAX in Harvard forest since 2009 to 2011.

Max AT reached in 2010 at 40 C.

In [233]:
```
ts1=df.groupby(['date'], sort=True)['AT'].min()
plt.plot(ts1)
```

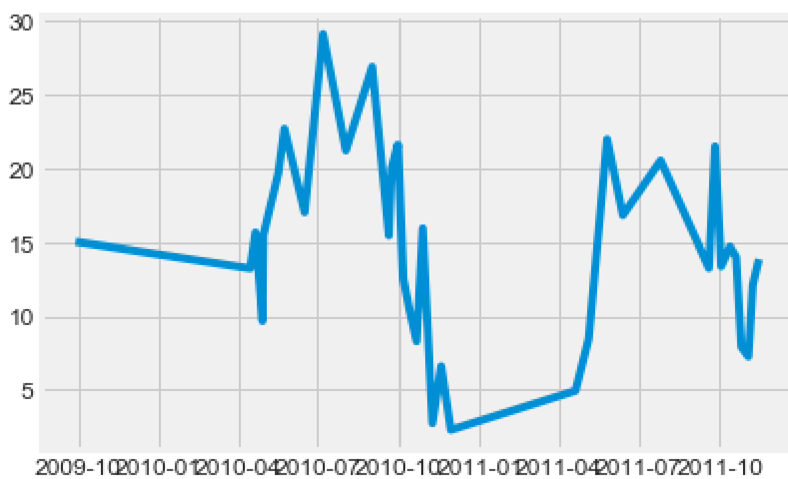
Out[233]: [<matplotlib.lines.Line2D at 0x2aa8e195438>]



Time series plot of AT min. The min temperature recorded at 2011 at -4 C.

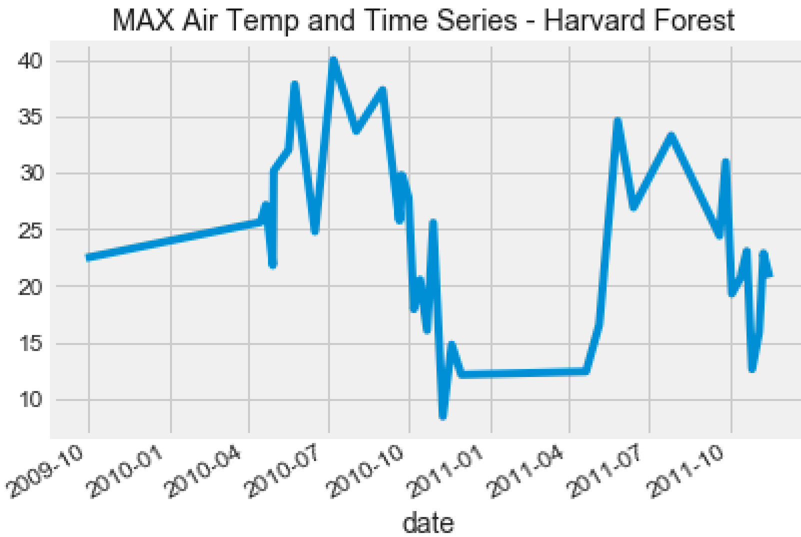
In [234]:
```
ts2=df.groupby(['date'], sort=True)['AT'].mean()
plt.plot(ts2)
```

Out[234]: [<matplotlib.lines.Line2D at 0x2aa8e1dd320>]



Time series - AT mean temp

In [279]:
```
ts.plot()
plt.title('MAX Air Temp and Time Series - Harvard Forest ')
plt.show()
```



In [259]:
```
ts1.head(10)
```

Out[259]:
```
date
2009-09-28   10.937
2010-04-15    2.823
2010-04-21    5.170
2010-04-29   -0.509
2010-04-30    1.847
2010-05-17    6.911
2010-05-24   12.953
2010-06-16   10.670
2010-07-07   23.497
2010-08-02    6.803
Name: AT, dtype: float64
```

In [182]:
```
from statsmodels.tsa.arima_model import ARIMA
model = ARIMA(df.AT, order=(1,0,0))
model_fit = model.fit()
print(model_fit.summary())
print('Residuals Description')
print(model_fit.resid.describe())
```

```
                              ARMA Model Results
==============================================================================
=
Dep. Variable:                     AT   No. Observations:               140717
7
Model:                     ARMA(1, 0)   Log Likelihood             -2818131.75
8
Method:                       css-mle   S.D. of innovations              1.79
3
Date:                Sat, 22 Jun 2019   AIC                        5636269.51
7
Time:                        18:41:51   BIC                        5636305.98
8
Sample:                             0   HQIC                       5636279.41
8

==============================================================================
=
                 coef    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const         14.9199      0.053    282.387      0.000      14.816      15.02
3
ar.L1.AT       0.9714      0.000   4852.586      0.000       0.971       0.97
2
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.0294           +0.0000j            1.0294            0.0000
-----------------------------------------------------------------------------
Residuals Description
count    1407177.000
mean          -0.000
std            1.793
min          -25.843
25%           -0.643
50%           -0.113
75%            0.477
max           30.456
dtype: float64
```

In [183]:
```
from statsmodels.tsa.arima_model import ARIMA
model = ARIMA(df.AT, order=(0,0,1))
model_fit = model.fit()
print(model_fit.summary())
print('Residuals Description')
print(model_fit.resid.describe())
```

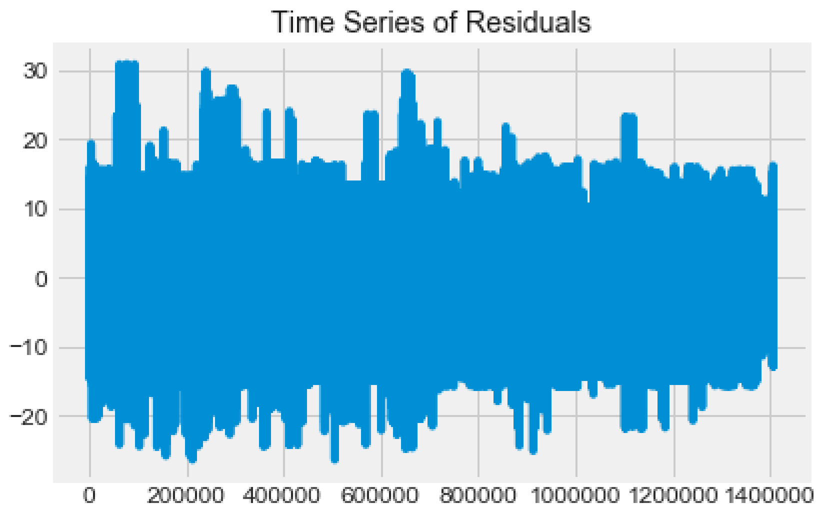
```
                              ARMA Model Results                              
==============================================================================
=
Dep. Variable:                     AT   No. Observations:               140717
7
Model:                     ARMA(0, 1)   Log Likelihood            -4079348.09
5
Method:                       css-mle   S.D. of innovations              4.39
3
Date:                Sat, 22 Jun 2019   AIC                        8158702.18
9
Time:                        18:42:56   BIC                        8158738.66
1
Sample:                             0   HQIC                       8158712.09
1

==============================================================================
=
                 coef    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const         14.9202      0.007   2167.485      0.000      14.907      14.93
4
ma.L1.AT       0.8588      0.000   2624.017      0.000       0.858       0.85
9
                                    Roots                                    
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1           -1.1644           +0.0000j            1.1644            0.5000
-----------------------------------------------------------------------------
Residuals Description
count   1407177.000
mean          0.000
std           4.393
min         -19.388
25%          -3.096
50%          -0.140
75%           2.964
max          22.031
dtype: float64
```

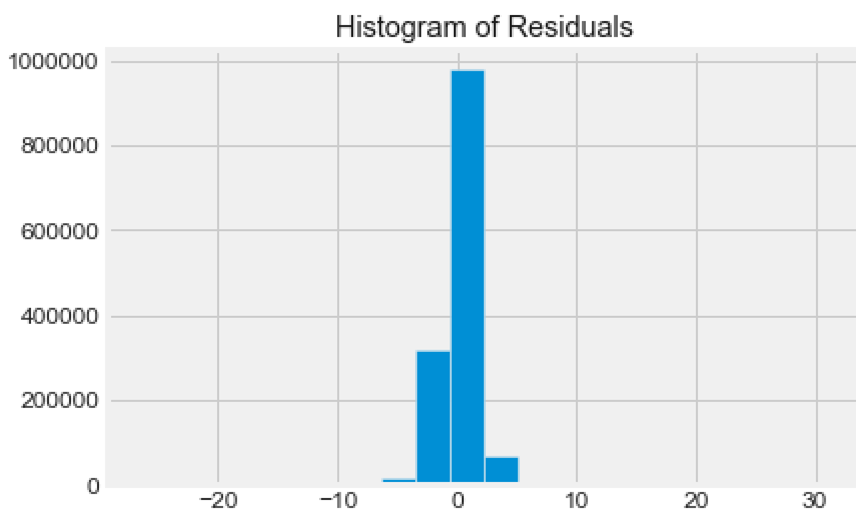
In [187]:
```
from statsmodels.tsa.arima_model import ARIMA
model_011 = ARIMA(df.AT, order=(0,1,1))
model_011_fit = model_011.fit()
print(model_011_fit.summary())
print('Residuals Description')
print(model_011_fit.resid.describe())
```

```
                              ARIMA Model Results
==============================================================================
=
Dep. Variable:                   D.AT   No. Observations:              140717
6
Model:                 ARIMA(0, 1, 1)   Log Likelihood            -2818612.14
3
Method:                       css-mle   S.D. of innovations              1.79
3
Date:                Sat, 22 Jun 2019   AIC                        5637230.28
5
Time:                        18:53:15   BIC                        5637266.75
7
Sample:                             1   HQIC                       5637240.18
7

==============================================================================
=
                 coef    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const      -8.499e-06      0.002     -0.005      0.996      -0.003       0.00
3
ma.L1.D.AT     0.1052      0.001    139.441      0.000       0.104       0.10
7
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1           -9.5067           +0.0000j            9.5067            0.5000
-----------------------------------------------------------------------------
Residuals Description
count   1407176.000
mean         -0.000
std           1.793
min         -26.401
25%          -0.613
50%          -0.043
75%           0.502
max          30.875
dtype: float64
```

In [188]:
```
residuals = pd.DataFrame(model_011_fit.resid)

residuals.plot(legend=False)
plt.title('Time Series of Residuals')

residuals.hist(bins=20)
plt.title('Histogram of Residuals')
```

Out[188]: Text(0.5,1,'Histogram of Residuals')





The residuals for the Harvard forest show a slight up-tick in the AT temperature in the past 3 years. It would be interesting to see that past 7 years to get deeper insights.

Duke Forest Time series

In [268]:
```
df1.head(5)
```

Out[268]:

	site	year	times	month	day	JD	JD2009	dayFraction	chamber	treatment	A
0	DF	2009	125.540	5	6	-240	125	0.540	1	None	24.22
1	DF	2009	125.540	5	6	-240	125	0.540	2	None	24.51
2	DF	2009	125.540	5	6	-240	125	0.540	3	None	24.07
3	DF	2009	125.540	5	6	-240	125	0.540	4	None	24.44
4	DF	2009	125.540	5	6	-240	125	0.540	5	None	24.30

In [270]:
```
tsf = df1.drop(columns=['site', 'year', 'times', 'month','day', 'JD', 'JD2009'
, 'ppfd', 'CO2_uptake', 'CO2_release', 'dayFraction', 'chamber', 'treatment'])
```

In [271]:
```
tsf.head(3)
```

Out[271]:

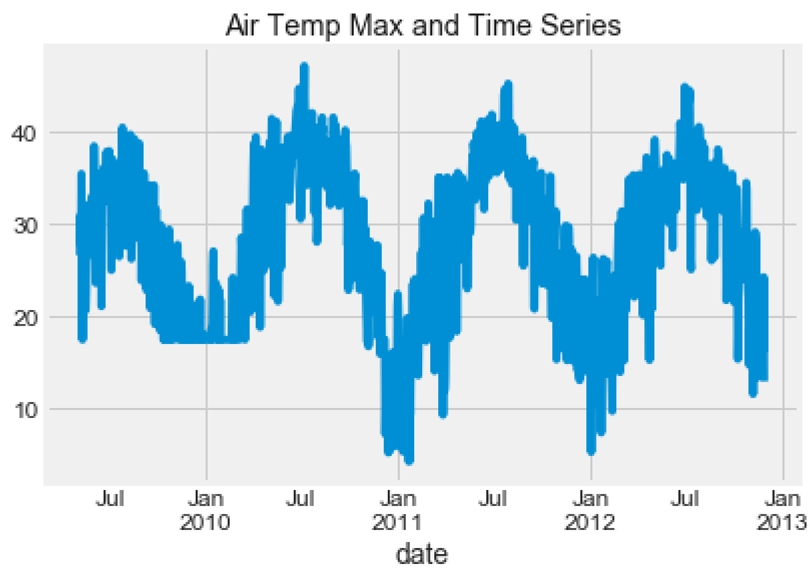
	AT	Q	Rh	SM	ST	date
0	24.227	24.227	24.227	24.227	24.227	2009-05-06
1	24.513	24.513	24.513	24.513	24.513	2009-05-06
2	24.070	24.070	24.070	24.070	24.070	2009-05-06

In [273]:
```
tsf = tsf[['date', 'AT', 'Q', 'Rh', 'SM', 'ST']]
```

In [274]:
```
from datetime import datetime
con=tsf['date']
tsf['date']=pd.to_datetime(tsf['date'])
tsf.set_index('date', inplace=True)
#check datatype of index
tsf.index
```

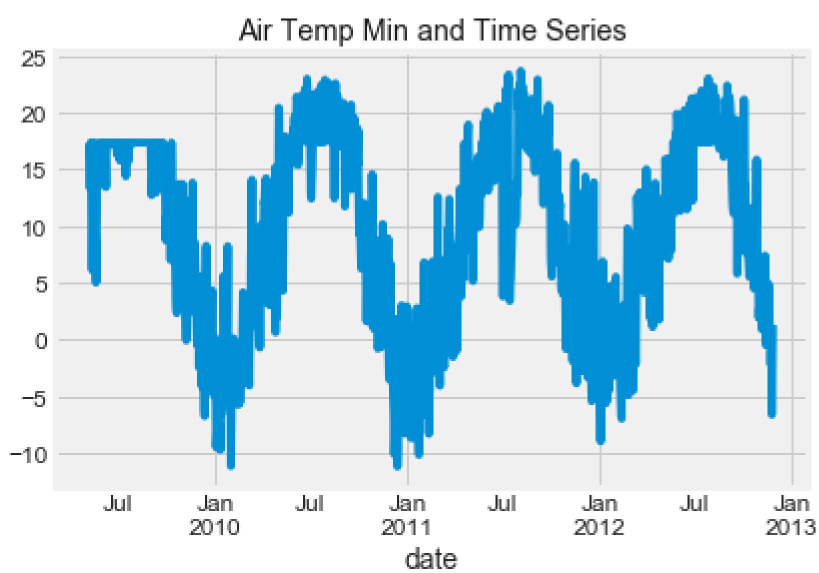
Out[274]:
```
DatetimeIndex(['2009-05-06', '2009-05-06', '2009-05-06', '2009-05-06',
               '2009-05-06', '2009-05-06', '2009-05-06', '2009-05-06',
               '2009-05-06', '2009-05-06',
               ...
               '2012-11-27', '2012-11-27', '2012-11-27', '2012-11-27',
               '2012-11-27', '2012-11-27', '2012-11-27', '2012-11-27',
               '2012-11-27', '2012-11-27'],
              dtype='datetime64[ns]', name='date', length=750768, freq=None)
```

In [277]:
```
tsf1=tsf.groupby(['date'], sort=True)['AT'].max()

tsf1.plot()
plt.title('Air Temp Max and Time Series')
plt.show()
```



The max air temperature in Duke forest was nearly 50 C in July 2010.

In [278]:
```
tsf2=tsf.groupby(['date'], sort=True)['AT'].min()

tsf2.plot()
plt.title('Air Temp Min and Time Series')
plt.show()
```



Duke forest min air temperature in Jan 2010 was below -10 C.

In [301]:
```
tsf.head(3)
```

Out[301]:

	AT	Q	Rh	SM	ST
date					
2009-05-06	24.227	24.227	24.227	24.227	24.227
2009-05-06	24.513	24.513	24.513	24.513	24.513
2009-05-06	24.070	24.070	24.070	24.070	24.070

In [313]:
```
df1.head(3)
```

Out[313]:

	site	year	times	month	day	JD	JD2009	dayFraction	chamber	treatment	A
0	DF	2009	125.540	5	6	-240	125	0.540	1	None	24.22
1	DF	2009	125.540	5	6	-240	125	0.540	2	None	24.51
2	DF	2009	125.540	5	6	-240	125	0.540	3	None	24.07

In [326]:
```
tsf13 = df1.drop(columns=['site', 'times', 'month', 'date','day', 'JD', 'JD200
9', 'ppfd', 'CO2_uptake', 'CO2_release', 'dayFraction', 'chamber', 'treatment'
])
```

In [327]:
```
tsf13.head(3)
```

Out[327]:

	year	AT	Q	Rh	SM	ST
0	2009	24.227	24.227	24.227	24.227	24.227
1	2009	24.513	24.513	24.513	24.513	24.513
2	2009	24.070	24.070	24.070	24.070	24.070

In [283]:
```
from statsmodels.tsa.arima_model import ARIMA
model = ARIMA(tsf.AT, order=(0,0,1))
model_fit = model.fit()
print(model_fit.summary())
print('Residuals Description')
print(model_fit.resid.describe())
```

```
                              ARMA Model Results                              
==============================================================================
=
Dep. Variable:                     AT   No. Observations:               75076
8
Model:                     ARMA(0, 1)   Log Likelihood            -2375560.27
2
Method:                       css-mle   S.D. of innovations              5.72
7
Date:                Sat, 22 Jun 2019   AIC                        4751126.54
4
Time:                        20:20:23   BIC                        4751161.13
0
Sample:                             0   HQIC                       4751136.17
3

==============================================================================
=
                 coef    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const         17.3824      0.012   1495.864      0.000      17.360      17.40
5
ma.L1.AT       0.7580      0.000   1550.674      0.000       0.757       0.75
9
                                    Roots                                    
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1           -1.3192           +0.0000j            1.3192            0.5000
-----------------------------------------------------------------------------
Residuals Description
count    750768.000
mean         -0.000
std           5.727
min         -33.442
25%          -3.506
50%           0.393
75%           3.774
max          31.440
dtype: float64
```

In [284]:
```
from statsmodels.tsa.arima_model import ARIMA
model = ARIMA(tsf.AT, order=(0,1,0))
model_fit = model.fit()
print(model_fit.summary())
print('Residuals Description')
print(model_fit.resid.describe())
```

```
                              ARIMA Model Results
==============================================================================
=
Dep. Variable:                   D.AT   No. Observations:               75076
7
Model:                 ARIMA(0, 1, 0)   Log Likelihood           -1934356.48
0
Method:                           css   S.D. of innovations             3.18
2
Date:                Sat, 22 Jun 2019   AIC                       3868716.96
1
Time:                        20:21:19   BIC                       3868740.01
8
Sample:                             1   HQIC                      3868723.38
0

==============================================================================
=
                 coef    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const      -2.245e-05      0.004     -0.006      0.995      -0.007       0.00
7
==============================================================================
=
Residuals Description
count   750767.000
mean        -0.000
std          3.182
min        -28.309
25%         -1.236
50%          0.000
75%          1.320
max         27.506
dtype: float64
```

In [285]:
```
from statsmodels.tsa.arima_model import ARIMA
model = ARIMA(tsf.AT, order=(1,0,0))
model_fit = model.fit()
print(model_fit.summary())
print('Residuals Description')
print(model_fit.resid.describe())
```

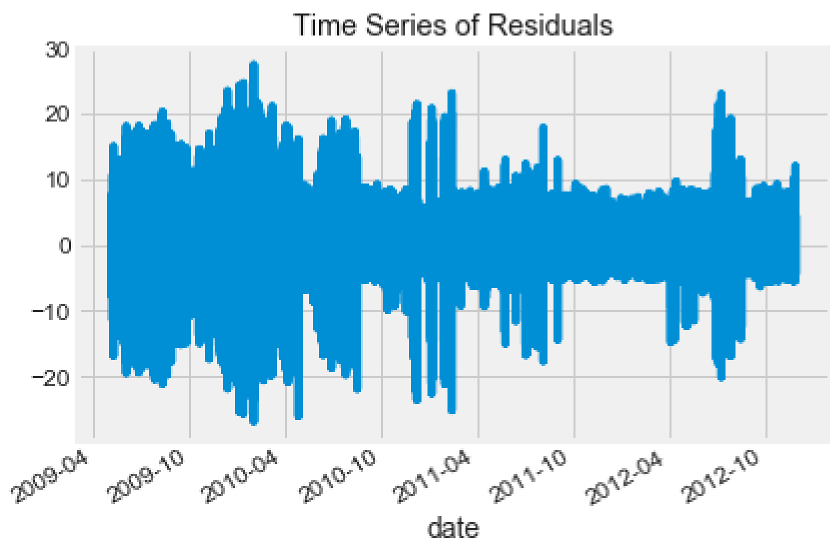
```
                              ARMA Model Results
==============================================================================
=
Dep. Variable:                     AT   No. Observations:               75076
8
Model:                     ARMA(1, 0)   Log Likelihood            -1922405.41
3
Method:                       css-mle   S.D. of innovations              3.13
2
Date:                Sat, 22 Jun 2019   AIC                        3844816.82
6
Time:                        20:21:55   BIC                        3844851.41
2
Sample:                             0   HQIC                       3844826.45
5

==============================================================================
=
                 coef    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const         17.3824      0.058    301.478      0.000      17.269      17.49
5
ar.L1.AT       0.9373      0.000   2330.469      0.000       0.937       0.93
8
                                    Roots
==============================================================================
                  Real          Imaginary           Modulus         Frequency
------------------------------------------------------------------------------
AR.1            1.0669           +0.0000j            1.0669            0.0000
------------------------------------------------------------------------------
Residuals Description
count   750768.000
mean        -0.000
std          3.132
min        -28.309
25%         -1.325
50%          0.000
75%          1.480
max         25.781
dtype: float64
```

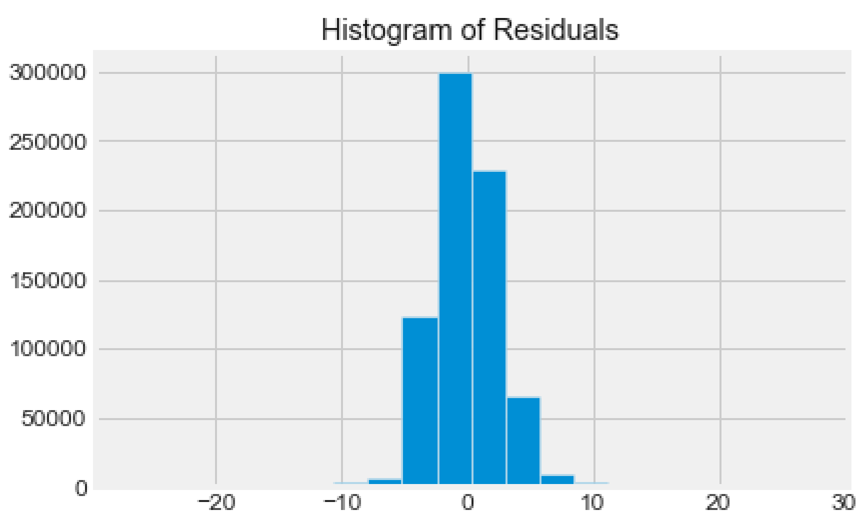
In [332]:
```
from statsmodels.tsa.arima_model import ARIMA
model_011 = ARIMA(tsf.AT, order=(0,1,1))
model_011_fit = model_011.fit()
print(model_011_fit.summary())
print('Residuals Description')
print(model_011_fit.resid.describe())
```

```
                              ARIMA Model Results
==============================================================================
=
Dep. Variable:                   D.AT   No. Observations:              75076
7
Model:                 ARIMA(0, 1, 1)   Log Likelihood            -1911566.46
4
Method:                       css-mle   S.D. of innovations              3.08
7
Date:                Sat, 22 Jun 2019   AIC                        3823138.92
8
Time:                        21:37:25   BIC                        3823173.51
5
Sample:                             1   HQIC                       3823148.55
7

==============================================================================
=
                 coef    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const       -2.23e-05      0.002     -0.010      0.992      -0.004       0.00
4
ma.L1.D.AT    -0.3896      0.002   -202.773      0.000      -0.393      -0.38
6
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1            2.5665           +0.0000j            2.5665            0.0000
-----------------------------------------------------------------------------
Residuals Description
count   750767.000
mean        -0.000
std          3.087
min        -26.885
25%         -1.531
50%         -0.000
75%          1.557
max         27.443
dtype: float64
```

In [333]:
```
residuals = pd.DataFrame(model_011_fit.resid)

residuals.plot(legend=False)
plt.title('Time Series of Residuals')

residuals.hist(bins=20)
plt.title('Histogram of Residuals')
```

Out[333]: `Text(0.5,1,'Histogram of Residuals')`





Within a 3-year span of the residual for the Duke forest indicate in 2012 tigher residuals or less variance. It would be interesting to see the last 7 years of environmental data to see if the AT ticked furture upward.