1 Iris Dataset

1.1 Summary Statistics

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
```

In [2]:

```
data = pd.read_csv("iris.data", names=["sepal_length", "sepal_width" , "petal_length", "petal_width", "class" ])
data
```

Out[2]:

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [3]:

```
# the count, mean, std, 25:50:75% percentiles, min, max of the features
data.describe()
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [4]:

```
# the variance of the each feature
print('sepal_length variance = ' + str(data['sepal_length'].var()))
print('sepal_width variance = ' + str(data['sepal_width'].var()))
print('petal_length variance = ' + str(data['petal_length'].var()))
print('petal_width variance = ' + str(data['petal_width'].var()))
```

```
sepal_length variance = 0.6856935123042507
sepal_width variance = 0.1880040268456376
petal_length variance = 3.113179418344519
petal_width variance = 0.582414317673378
```

In [5]:

```
# the range of the each feature
print('sepal_length range = ' + str(data['sepal_length'].max() - data['sepal_length'].min()))
print('sepal_width range = ' + str(data['sepal_width'].max() - data['sepal_width'].min()))
print('petal_length range = ' + str(data['petal_length'].max() - data['petal_length'].min()))
print('petal_width range = ' + str(data['petal_width'].max() - data['petal_width'].min()))
```

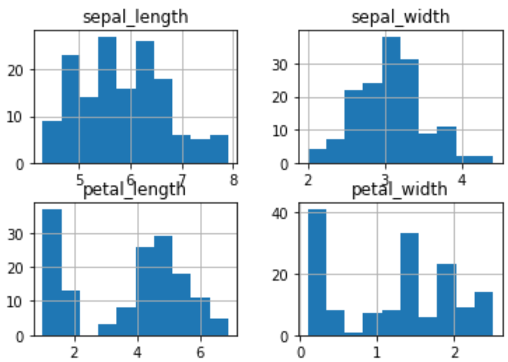
```
sepal_length range = 3.6000000000000005
sepal_width range = 2.4000000000000004
petal_length range = 5.9
petal_width range = 2.4
```

1.2 Data Visualization

Histograms:

To illustrate the feature distributions, create a histogram for each feature in the dataset. You may plot each histogram individually or combine them all into a single plot. When generating histograms for this assignment, use the default number of bins. Recall that a histogram provides a graphical representation of the distribution of the data.

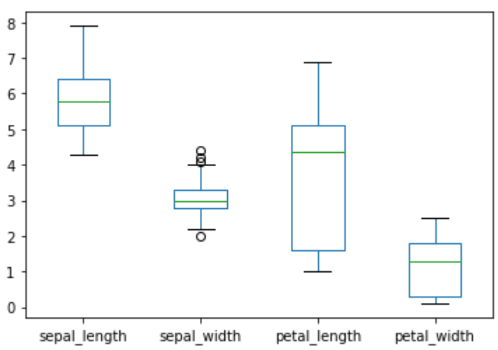
In [6]:
```
pdhist = data.hist()
```



Box Plots:

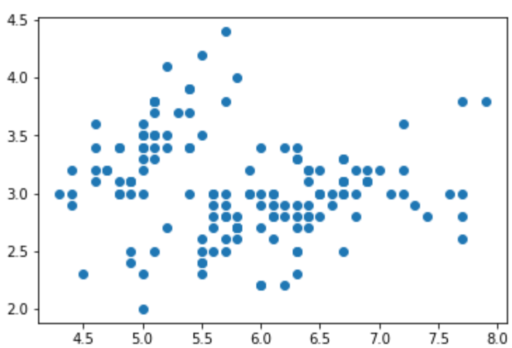
To further understand the data, create a boxplot for each feature in the dataset. Present all the boxplots into a single plot. Recall that a boxplot provides a graphical repre- sentation of the location and variation of the data through their quartiles; they are especially useful for comparing distributions and identifying outliers.

In [7]:
```
box = data.boxplot(grid=False, return_type='axes')
```

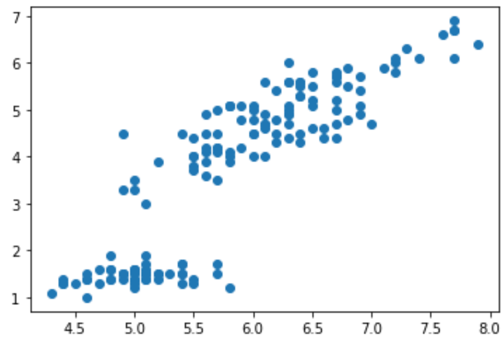


Pairwise Plot: To understand the relationship between the features, create a scatter plot for each pair of the features. If there are are n features in the dataset, there should be nC2 plots.

In [8]:
```
# relationship between sepal_length and sepal_width
pdscatter1 = plt.scatter(data['sepal_length'], data['sepal_width'])
```

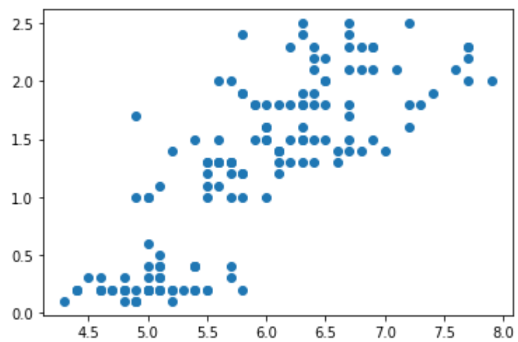


In [9]:
```
# relationship between sepal_length and petal_length
pdscatter2 = plt.scatter(data['sepal_length'], data['petal_length'])
```



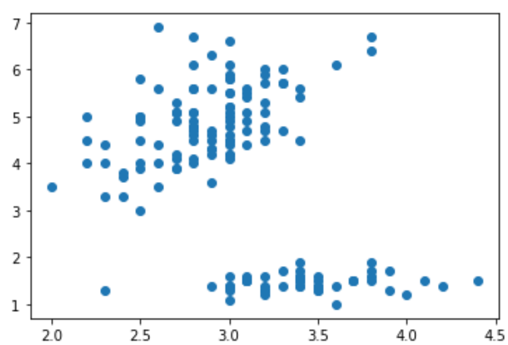
In [10]:

```
# relationship between sepal_length and petal_width
pdscatter3 = plt.scatter(data['sepal_length'], data['petal_width'])
```



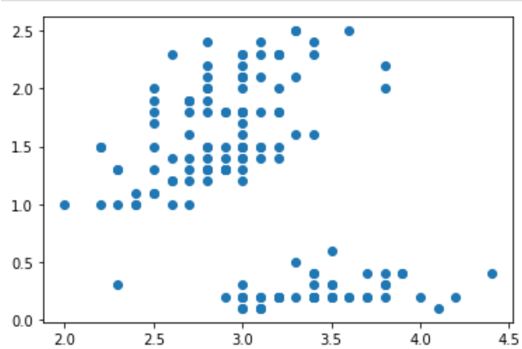
In [11]:

```
# relationship between sepal_width and petal_length
pdscatter4 = plt.scatter(data['sepal_width'], data['petal_length'])
```



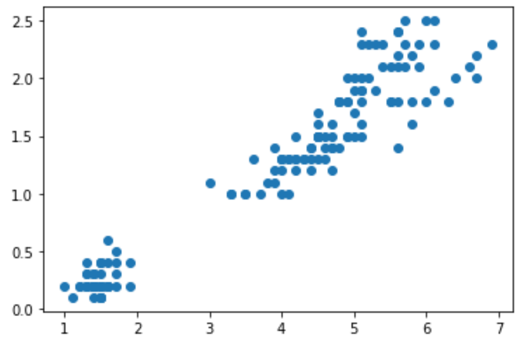
In [12]:

```
# relationship between sepal_width and petal_width
pdscatter5 = plt.scatter(data['sepal_width'], data['petal_width'])
```



In [13]:

```
# relationship between petal_length and petal_width
pdscatter5 = plt.scatter(data['petal_length'], data['petal_width'])
```

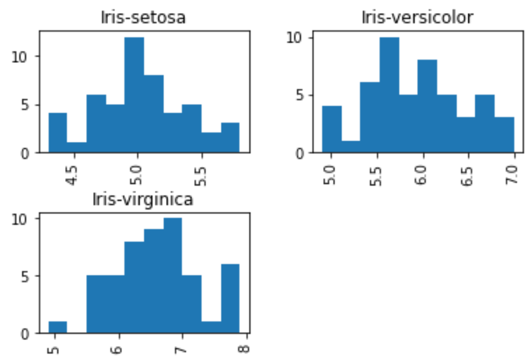


Class-wise Visualization:

Create histograms for each feature in a similar way for each of the different classes present in the data.

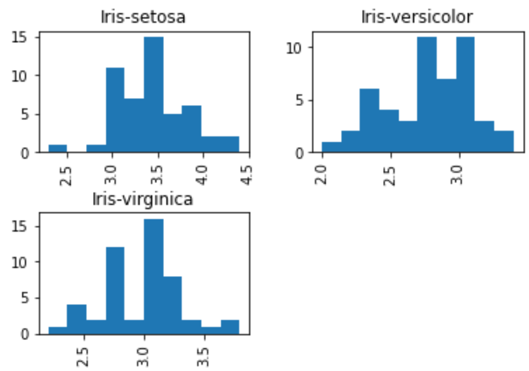
In [14]:
```
data['sepal_length'].hist(by=data['class'])
```

Out[14]:
```
array([[<AxesSubplot:title={'center':'Iris-setosa'}>,
        <AxesSubplot:title={'center':'Iris-versicolor'}>],
       [<AxesSubplot:title={'center':'Iris-virginica'}>, <AxesSubplot:>]],
      dtype=object)
```



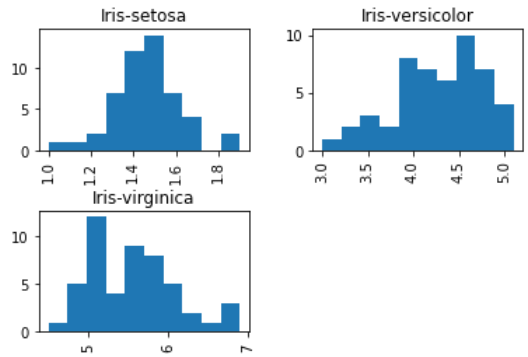
In [15]:
```
data['sepal_width'].hist(by=data['class'])
```

Out[15]:
```
array([[<AxesSubplot:title={'center':'Iris-setosa'}>,
        <AxesSubplot:title={'center':'Iris-versicolor'}>],
       [<AxesSubplot:title={'center':'Iris-virginica'}>, <AxesSubplot:>]],
      dtype=object)
```



In [16]:
```
data['petal_length'].hist(by=data['class'])
```

Out[16]:
```
array([[<AxesSubplot:title={'center':'Iris-setosa'}>,
        <AxesSubplot:title={'center':'Iris-versicolor'}>],
       [<AxesSubplot:title={'center':'Iris-virginica'}>, <AxesSubplot:>]],
      dtype=object)
```

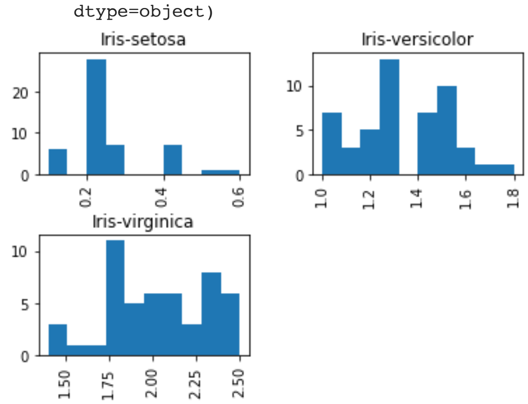


In [17]:

```
data['petal_width'].hist(by=data['class'])
```

Out[17]:

```
array([[<AxesSubplot:title={'center':'Iris-setosa'}>,
        <AxesSubplot:title={'center':'Iris-versicolor'}>],
       [<AxesSubplot:title={'center':'Iris-virginica'}>, <AxesSubplot:>]],
      dtype=object)
```



1.3 Conceptual Questions

1. How many features are there? What are the Types of the features (e.g., numeric, nominal, discrete, continuous)?

There are 4 features. They have continuous numeric values.

2. From the histograms of the whole data, how do the shapes of the histograms for petal length and petal width differ from those for sepal length and sepal width? Is there a particular value of petal length (which ranges from 1.0 to 6.9) where the distribution of petal lengths (as illustrated by the histogram) could be best segmented into two parts?

As we can tell from the histograms:

1. Sepal length and sepal width's shapes are continous, which looks like a skewed distribution.
2. Petal length and petal width's shape are separated into two parts.
3. Petal length can be segmented into two parts around 2.3

3. Based upon these boxplots, is there a pair of features that appear to have significantly different medians? Recall that the degree of overlap between variability is an important initial indicator of the likelihood that differences in means or medians are meaningful. Also, based solely upon the box plots, which feature appears to explain the greatest amount of the data?

As we can tell from the boxplots:

1. Sepal length and Petal width have significantly different medians.
2. Petal length has the highest degree of overlap, which explains the greates amount of data.

4. From the pairwise plots of the features, which features are most correlated from the plots? Mention at least three pairs.

According to the pairwise plots:

1. petal_length and petal_width are correlated.
2. sepal_length and petal_length are correlated.
3. sepal_length and petal_width are correlated.

5. Compare the histograms of each class to the histograms of the whole dataset. What differences do you see in the shapes?

1. All of the histograms of each class adds up to the histograms of the whole dataset.
2. For sepal_length, petal_length, and petal_width, they have totally different ranges for different classes.
3. For example, for petal_length, sentosa has ranges around 1.0-2.2, versicolor has ranges around 3.0-5.0, and verginica has ranges around 4.5-7.0 .
4. There isn't a specific pattern for histograms for each classes. However, histograms of the whole dataset are either fall into two segments or look like a skewed distribution.

Air Quality Dataset

In [18]:

```
data2 = pd.read_excel('AirQualityUCI.xlsx')
data2.head()
```

Out[18]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O
0	2004-03-10	18:00:00	2.6	1360.00	150	11.881723	1045.50	166.0	1056.25	113.0	1692.00	1267.5
1	2004-03-10	19:00:00	2.0	1292.25	112	9.397165	954.75	103.0	1173.75	92.0	1558.75	972.2
2	2004-03-10	20:00:00	2.2	1402.00	88	8.997817	939.25	131.0	1140.00	114.0	1554.50	1074.0
3	2004-03-10	21:00:00	2.2	1375.50	80	9.228796	948.25	172.0	1092.00	122.0	1583.75	1203.2
4	2004-03-10	22:00:00	1.6	1272.25	51	6.518224	835.50	131.0	1205.00	116.0	1490.00	1110.0

2.1 Summary Statistics

In [19]:

```
data2.describe()
```

Out[19]:

	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5
count	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.00
mean	-34.207524	1048.869652	-159.090093	1.865576	894.475963	168.604200	794.872333	58.135898	1391.363266	974.95
std	77.657170	329.817015	139.789093	41.380154	342.315902	257.424561	321.977031	126.931428	467.192382	456.92
min	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.00
25%	0.600000	921.000000	-200.000000	4.004958	711.000000	50.000000	637.000000	53.000000	1184.750000	699.75
50%	1.500000	1052.500000	-200.000000	7.886653	894.500000	141.000000	794.250000	96.000000	1445.500000	942.00
75%	2.600000	1221.250000	-200.000000	13.636091	1104.750000	284.200000	960.250000	133.000000	1662.000000	1255.25
max	11.900000	2039.750000	1189.000000	63.741476	2214.000000	1479.000000	2682.750000	339.700000	2775.000000	2522.75

In [20]:

```
# the variance of the each feature
print('CO(GT) variance = ' + str(data2['CO(GT)'].var()))
print('PT08.S1(CO) variance = ' + str(data2['PT08.S1(CO)'].var()))
print('NMHC(GT) = ' + str(data2['NMHC(GT)'].var()))
print('C6H6(GT) variance = ' + str(data2['C6H6(GT)'].var()))
print('PT08.S2(NMHC) variance = ' + str(data2['PT08.S2(NMHC)'].var()))
print('NOx(GT)   variance = ' + str(data2['NOx(GT)'].var()))
print('PT08.S3(NOx) = ' + str(data2['PT08.S3(NOx)'].var()))
print('NO2(GT) variance = ' + str(data2['NO2(GT)'].var()))
print('PT08.S4(NO2) variance = ' + str(data2['PT08.S4(NO2)'].var()))
print('PT08.S5(O3) = ' + str(data2['PT08.S5(O3)'].var()))
print('T = ' + str(data2['T'].var()))
print('RH variance = ' + str(data2['RH'].var()))
print('AH variance = ' + str(data2['AH'].var()))
```

```
CO(GT) variance = 6030.636106276823
PT08.S1(CO) variance = 108779.26309521518
NMHC(GT) = 19540.99049290499
C6H6(GT) variance = 1712.317143218122
PT08.S2(NMHC) variance = 117180.17665318836
NOx(GT)  variance = 66267.40479317415
PT08.S3(NOx) = 103669.20871905099
NO2(GT) variance = 16111.58746171175
PT08.S4(NO2) variance = 218268.72172917935
PT08.S5(O3) = 208778.37916470043
T = 1866.5370236018796
```

```
RH variance = 2623.042272805839
AH variance = 1519.1808166108053
```

In [21]:

```python
# the range of the each feature
print('Date range = ' + str(data2['Date'].max() - data2['Date'].min()))
print('CO(GT) range = ' + str(data2['CO(GT)'].max() - data2['CO(GT)'].min()))
print('PT08.S1(CO) range = ' + str(data2['PT08.S1(CO)'].max() - data2['PT08.S1(CO)'].min()))
print('NMHC(GT) range = ' + str(data2['NMHC(GT)'].max() - data2['NMHC(GT)'].min()))
print('C6H6(GT) range = ' + str(data2['C6H6(GT)'].max() - data2['C6H6(GT)'].min()))
print('PT08.S2(NMHC) range = ' + str(data2['PT08.S2(NMHC)'].max() - data2['PT08.S2(NMHC)'].min()))
print('NOx(GT) range = ' + str(data2['NOx(GT)'].max() - data2['NOx(GT)'].min()))
print('PT08.S3(NOx) range = ' + str(data2['PT08.S3(NOx)'].max() - data2['PT08.S3(NOx)'].min()))
print('NO2(GT) range = ' + str(data2['NO2(GT)'].max() - data2['NO2(GT)'].min()))
print('PT08.S4(NO2) range = ' + str(data2['PT08.S4(NO2)'].max() - data2['PT08.S4(NO2)'].min()))
print('PT08.S5(O3) range = ' + str(data2['PT08.S5(O3)'].max() - data2['PT08.S5(O3)'].min()))
print('T range = ' + str(data2['T'].max() - data2['T'].min()))
print('RH range = ' + str(data2['RH'].max() - data2['RH'].min()))
print('AH range = ' + str(data2['AH'].max() - data2['AH'].min()))
```

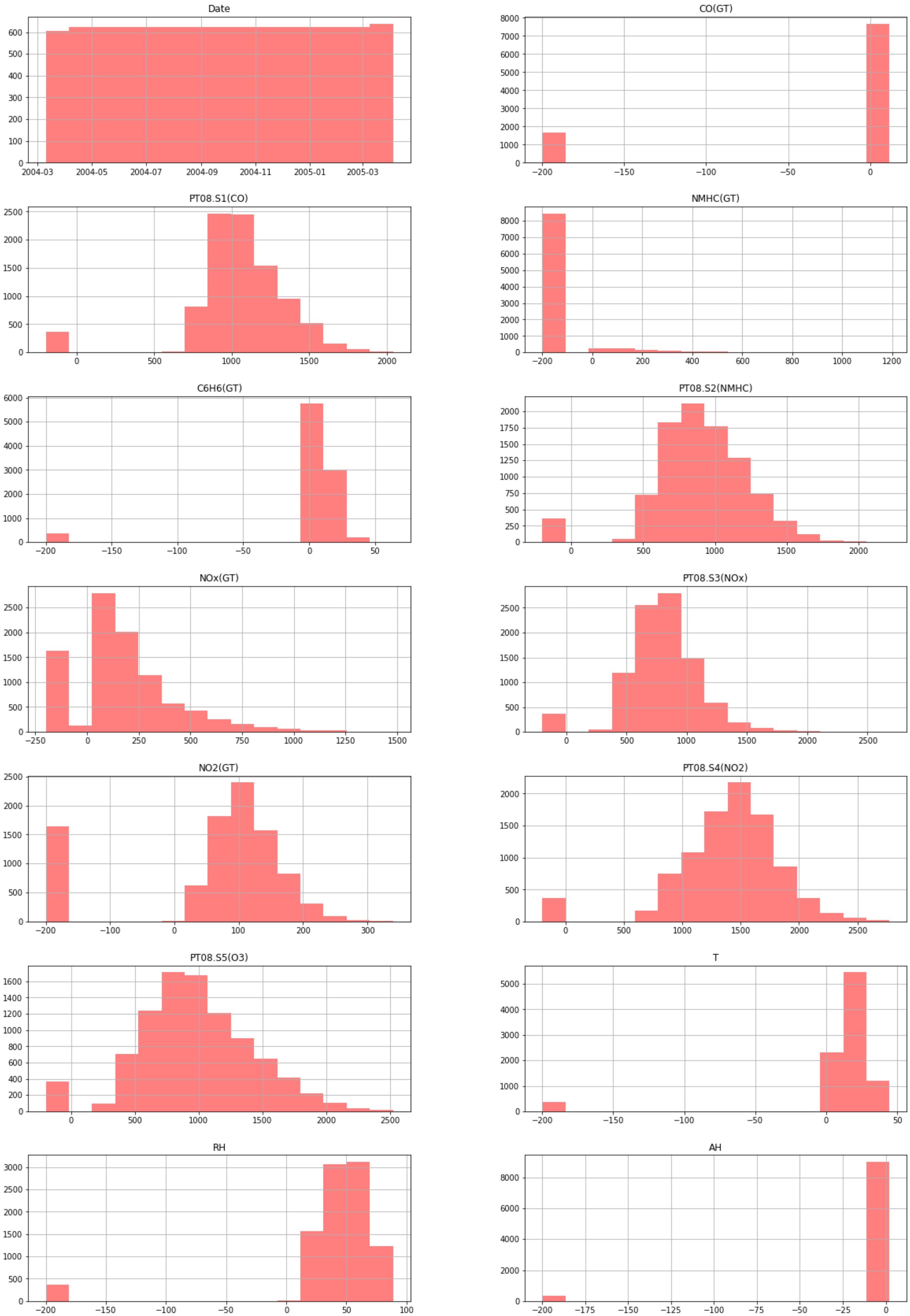
```
Date range = 390 days 00:00:00
CO(GT) range = 211.9
PT08.S1(CO) range = 2239.75
NMHC(GT) range = 1389
C6H6(GT) range = 263.7414764482916
PT08.S2(NMHC) range = 2414.0
NOx(GT) range = 1679.0
PT08.S3(NOx) range = 2882.75
NO2(GT) range = 539.7
PT08.S4(NO2) range = 2975.0
PT08.S5(O3) range = 2722.75
T range = 244.60000038147
RH range = 288.72500038147
AH range = 202.23103571558318
```

2.2 Data Visualization

Histogram

In [30]:

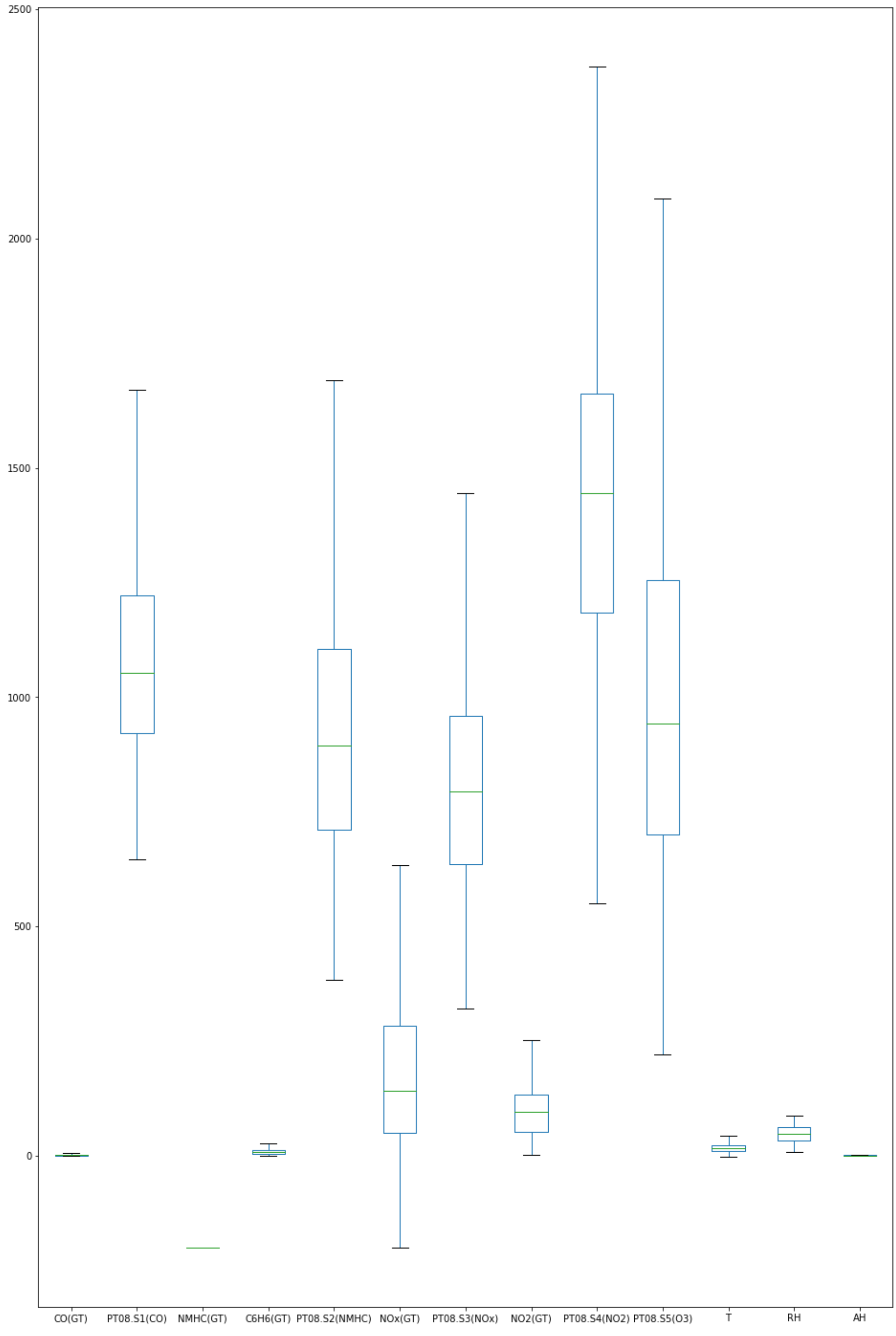
```python
pdhist2 = data2.hist(bins=15, color='r',layout=[-1,2], alpha=0.5, figsize=(20,30))
```


Date
0 100 200 300 400 500 600
2004-03 2004-05 2004-07 2004-09 2004-11 2005-01 2005-03
CO(GT)
0 1000 2000 3000 4000 5000 6000 7000 8000
−200 −150 −100 −50 0
PT08.S1(CO)
0 500 1000 1500 2000 2500
0 500 1000 1500 2000
NMHC(GT)
0 1000 2000 3000 4000 5000 6000 7000 8000
−200 0 200 400 600 800 1000 1200
C6H6(GT)
0 1000 2000 3000 4000 5000 6000
−200 −150 −100 −50 0 50
PT08.S2(NMHC)
0 250 500 750 1000 1250 1500 1750 2000
0 500 1000 1500 2000
NOx(GT)
0 500 1000 1500 2000 2500
−250 0 250 500 750 1000 1250 1500
PT08.S3(NOx)
0 500 1000 1500 2000 2500
0 500 1000 1500 2000 2500
NO2(GT)
0 500 1000 1500 2000 2500
−200 −100 0 100 200 300
PT08.S4(NO2)
0 500 1000 1500 2000
0 500 1000 1500 2000 2500
PT08.S5(O3)
0 200 400 600 800 1000 1200 1400 1600
0 500 1000 1500 2000 2500
T
0 1000 2000 3000 4000 5000
−200 −150 −100 −50 0 50
RH
0 500 1000 1500 2000 2500 3000
−200 −150 −100 −50 0 50 100
AH
0 2000 4000 6000 8000
−200 −175 −150 −125 −100 −75 −50 −25 0

Boxplot

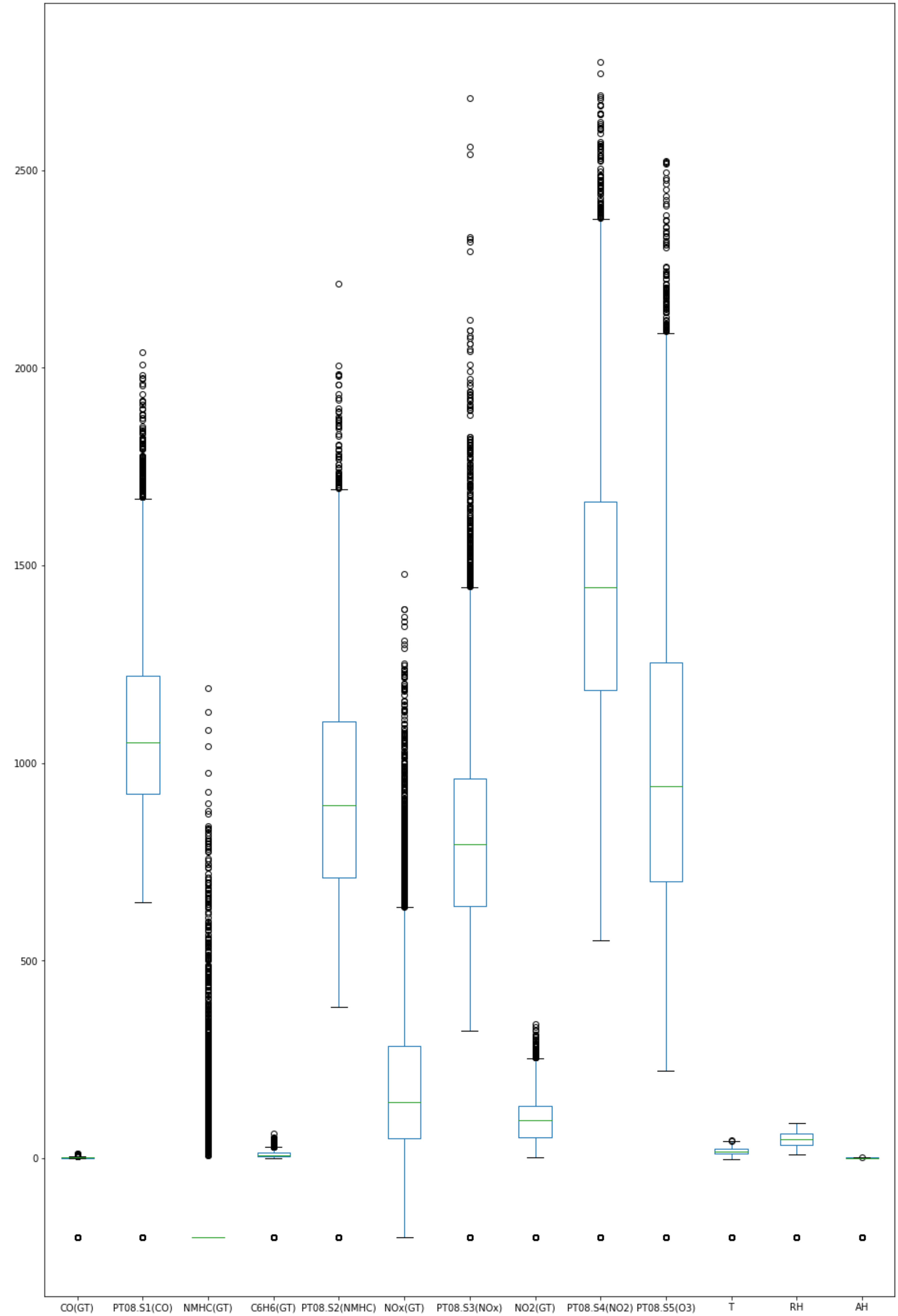
In [23]:

```
box1 = data2.boxplot(grid=False, return_type='axes', showfliers=False, figsize=(16,25))
```


2500
2000
1500
1000
500
0
CO(GT)
PT08.S1(CO)
NMHC(GT)
C6H6(GT)
PT08.S2(NMHC)
NOx(GT)
PT08.S3(NOx)
NO2(GT)
PT08.S4(NO2)
PT08.S5(O3)
T
RH
AH

In [24]:

```
box2 = data2.boxplot(grid=False, return_type='axes', showfliers=True, figsize=(16,25))
```


2500
2000
1500
1000
500
0
CO(GT)
PT08.S1(CO)
NMHC(GT)
C6H6(GT)
PT08.S2(NMHC)
NOx(GT)
PT08.S3(NOx)
NO2(GT)
PT08.S4(NO2)
PT08.S5(O3)
T
RH
AH

2.3 Conceptual Questions

1. From the histograms, what abnormality can you see?

We have a column which has a value of -200 for all histograms, which is outside the pattern of skewed distirbution for C0, C6H6 PT08.S1, PT08.S2, PT08.S3, PT08.S4, PT08.S5, NOx, NO2, tempertature, Relative Humidity(RH), and AH histograms.

2. What abnormality can you see from the summary statistics?

The mininum values for all feastures are -200.

3. How can you remove the abnormality from the data?

Apparently, -200 is not a legitmate values for all of these feastures. Therefore, we should get rid of the data which has a negative value.

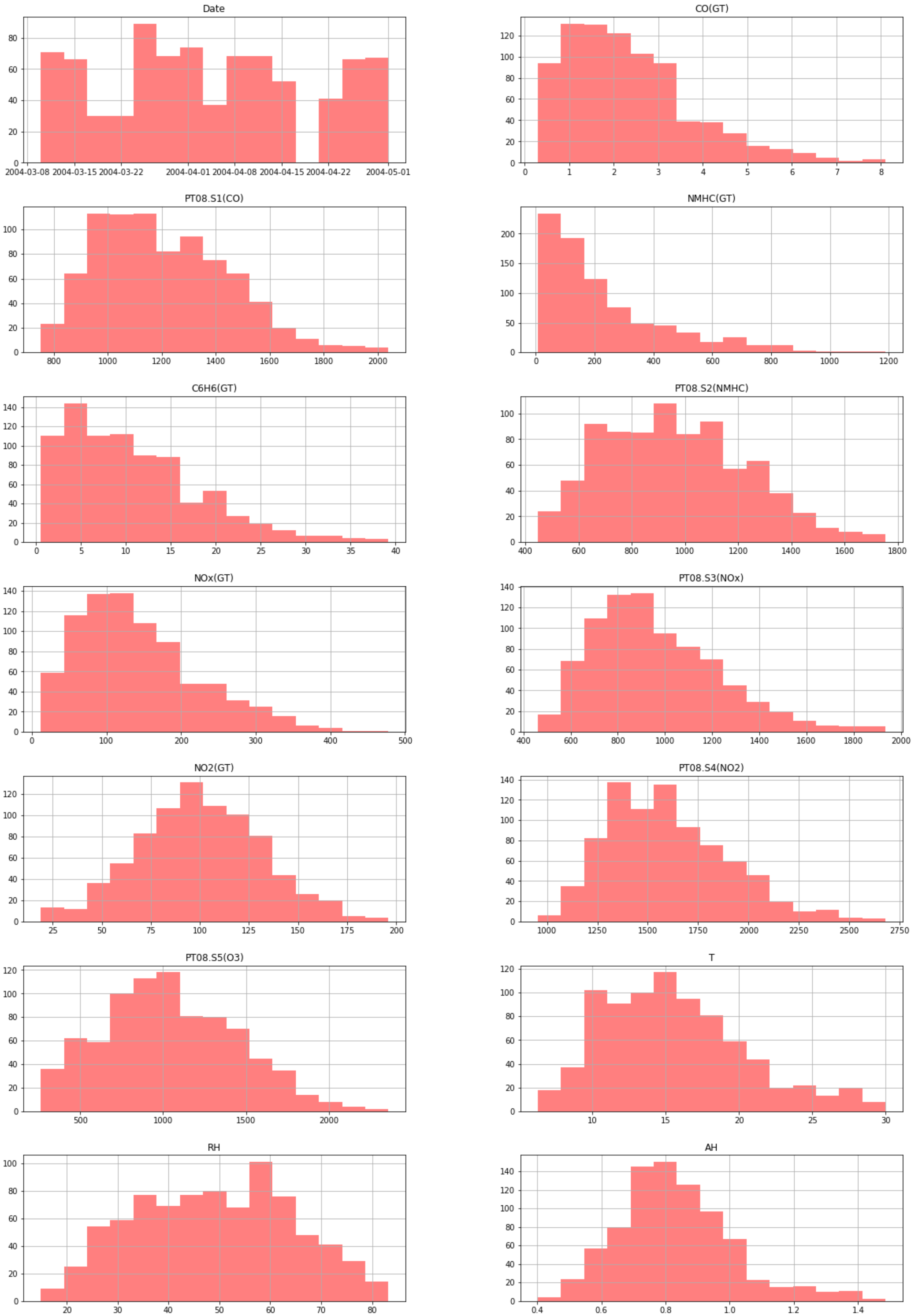
4. Show how the histograms look after removing the abnormalities from the data?

In [31]:

```
table = data2[data2["CO(GT)"] > 0]
table = table[table["NMHC(GT)"] > 0]
table = table[table["C6H6(GT)"] > 0]
table = table[table["NOx(GT)"] > 0]
table = table[table["PT08.S1(CO)"] > 0]
table = table[table["NO2(GT)"] > 0]
table = table[table["PT08.S4(NO2)"] > 0]
table = table[table["PT08.S5(O3)"] > 0]
table = table[table["PT08.S2(NMHC)"] > 0]
table = table[table["AH"] > 0]
table = table[table["RH"] > 0]
table = table[table["T"] > 0]
table.hist(bins=15, color='r',layout=[-1,2], alpha=0.5, figsize=(20,30))
```

Out[31]:

```
array([[<AxesSubplot:title={'center':'Date'}>,
        <AxesSubplot:title={'center':'CO(GT)'}>],
       [<AxesSubplot:title={'center':'PT08.S1(CO)'}>,
        <AxesSubplot:title={'center':'NMHC(GT)'}>],
       [<AxesSubplot:title={'center':'C6H6(GT)'}>,
        <AxesSubplot:title={'center':'PT08.S2(NMHC)'}>],
       [<AxesSubplot:title={'center':'NOx(GT)'}>,
        <AxesSubplot:title={'center':'PT08.S3(NOx)'}>],
       [<AxesSubplot:title={'center':'NO2(GT)'}>,
        <AxesSubplot:title={'center':'PT08.S4(NO2)'}>],
       [<AxesSubplot:title={'center':'PT08.S5(O3)'}>,
        <AxesSubplot:title={'center':'T'}>],
       [<AxesSubplot:title={'center':'RH'}>,
        <AxesSubplot:title={'center':'AH'}>]], dtype=object)
```


Date
2004-03-08 2004-03-15 2004-03-22 2004-04-01 2004-04-08 2004-04-15 2004-04-22 2004-05-01
CO(GT)
0 1 2 3 4 5 6 7 8
PT08.S1(CO)
800 1000 1200 1400 1600 1800 2000
NMHC(GT)
0 200 400 600 800 1000 1200
C6H6(GT)
0 5 10 15 20 25 30 35 40
PT08.S2(NMHC)
400 600 800 1000 1200 1400 1600 1800
NOx(GT)
0 100 200 300 400 500
PT08.S3(NOx)
400 600 800 1000 1200 1400 1600 1800 2000
NO2(GT)
25 50 75 100 125 150 175 200
PT08.S4(NO2)
1000 1250 1500 1750 2000 2250 2500 2750
PT08.S5(O3)
500 1000 1500 2000
T
10 15 20 25 30
RH
20 30 40 50 60 70 80
AH
0.4 0.6 0.8 1.0 1.2 1.4

In [26]:

```
# we verify that we don't have abnormalities now.
table.describe()
```

Out[26]:

	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)
count	827.000000	827.000000	827.000000	827.000000	827.000000	827.000000	827.000000	827.000000	827.000000	827.000000
mean	2.353567	1207.741838	231.025393	10.772367	965.983777	143.501814	963.178053	100.259976	1600.506550	1045.691052
std	1.409496	241.826753	208.461912	7.417127	266.413137	81.829717	265.906153	31.493823	302.290036	400.130277
min	0.300000	752.500000	7.000000	0.542781	447.500000	12.000000	461.250000	19.000000	955.000000	263.000000
25%	1.300000	1016.875000	77.000000	4.804320	753.500000	81.000000	768.875000	78.500000	1369.125000	759.500000
50%	2.000000	1172.000000	157.000000	9.125831	944.250000	128.000000	920.000000	99.000000	1556.250000	1009.000000
75%	3.100000	1380.250000	318.500000	14.803204	1142.375000	187.000000	1131.000000	122.000000	1783.375000	1319.750000
max	8.100000	2039.750000	1189.000000	39.202340	1754.250000	478.000000	1934.500000	196.000000	2679.000000	2358.500000