数据预处理

曹文强
2120150977
atlantic8@outlook.com

数据载入

我采用 python 作为数据处理工具，涉及 pandas，numpy，csv，matplotlib 等库。因为需要使用 csv 库中的 read_csv()方法，我先将数据间的空格全部转换为’,’，为了便于处理，我将表格中的缺失值全部换成 np.nan。

数据摘要

- 标称属性
  读入数据为 DataFrame 实例，取出一列为 Series 实例。将 Series 对象转换成 list 对象，使用 count()方法即可。

```
count non-numeric attributes value
('winter', 57)
('spring', 48)
('autumn', 36)
('summer', 43)
('small', 59)
('medium', 83)
('large', 42)
('medium', 77)
('high', 76)
('low', 31)
```

- 数值属性
  缺失值计数：

```
count missing value
('mxPH', 1)
('mnO2', 2)
('Cl', 10)
('NO3', 2)
('NH4', 2)
('oPO4', 2)
('PO4', 2)
('Chla', 12)
```

使用 Series 对象的 max,min,median,quantile 等方法,也可以使用 DataFrame 的 describe 方法将这个 DataFrame 对象的所有列的摘要信息打印出来。

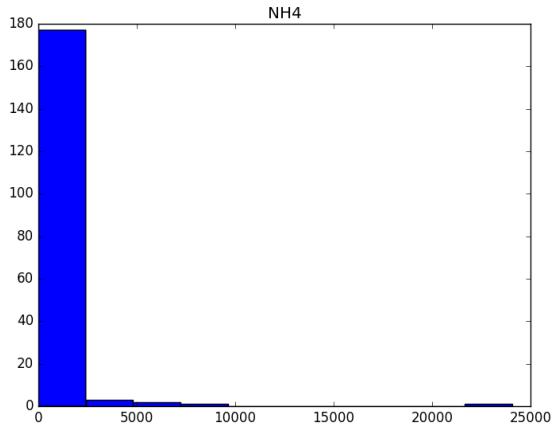
```
attribute: mxPH
('max value: ', 9.5)
('min value: ', 7.0)
('median value: ', 8.1)
('quantile(0.25): ', 7.7774999999999999)
('quantile(0.75): ', 8.4000000000000004)
attribute: mnO2
('max value: ', 13.4)
('min value: ', 1.5)
('median value: ', 9.75)
('quantile(0.25): ', 7.6750000000000007)
('quantile(0.75): ', 10.699999999999999)
attribute: Cl
('max value: ', 391.5)
('min value: ', 0.80000000000000004)
('median value: ', 35.08)
('quantile(0.25): ', 11.854749999999999)
('quantile(0.75): ', 58.515000000000001)
attribute: NO3
('max value: ', 45.649999999999999)
('min value: ', 0.050000000000000003)
('median value: ', 2.8200000000000003)
('quantile(0.25): ', 1.36425)
('quantile(0.75): ', 4.54)
```

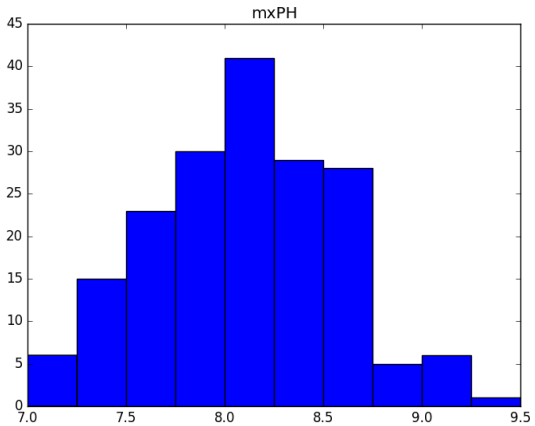
```
attribute: NH4
('max value: ', 24064.0)
('min value: ', 5.7999999999999998)
('median value: ', 115.714)
('quantile(0.25): ', 49.375)
('quantile(0.75): ', 235.25)
attribute: oPO4
('max value: ', 564.59997999999996)
('min value: ', 1.25)
('median value: ', 46.283500000000004)
('quantile(0.25): ', 18.562750000000001)
('quantile(0.75): ', 102.82849999999999)
attribute: PO4
('max value: ', 771.59997999999996)
('min value: ', 2.5)
('median value: ', 115.6)
('quantile(0.25): ', 50.341250000000002)
('quantile(0.75): ', 220.25125)
attribute: Chla
('max value: ', 110.456)
('min value: ', 0.20000000000000001)
('median value: ', 5.522)
('quantile(0.25): ', 2.0750000000000002)
('quantile(0.75): ', 18.307500000000001)
```

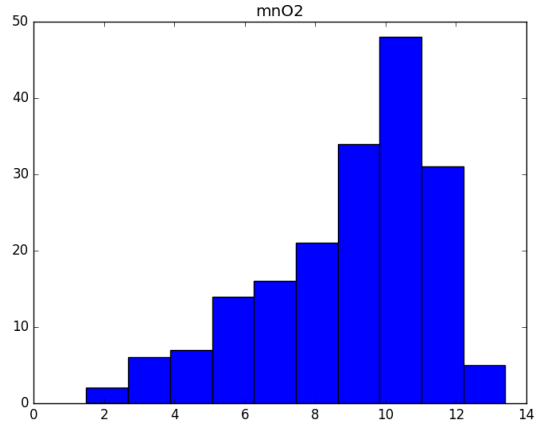
数据可视化

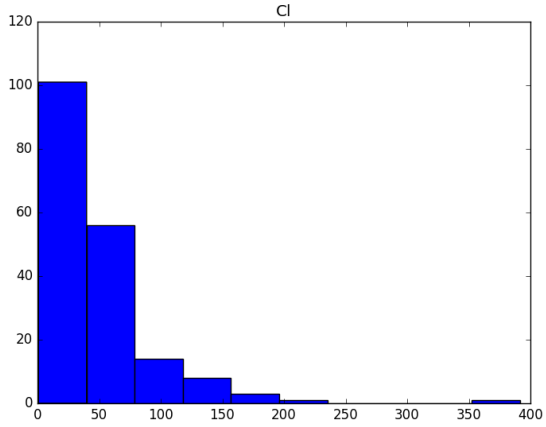
我使用 python 的 matplotlib 库作为画图工具

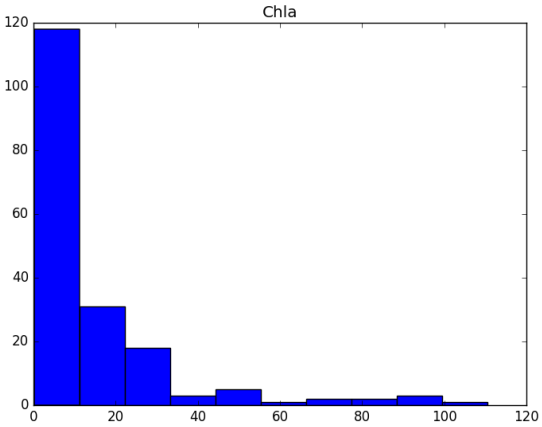
- 直方图

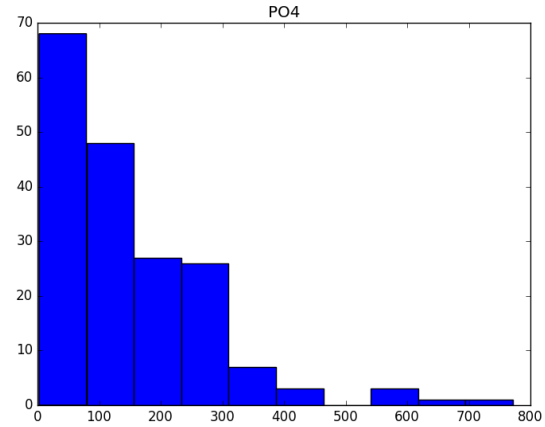


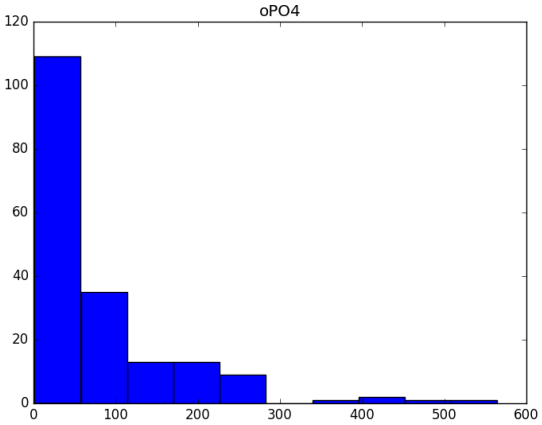


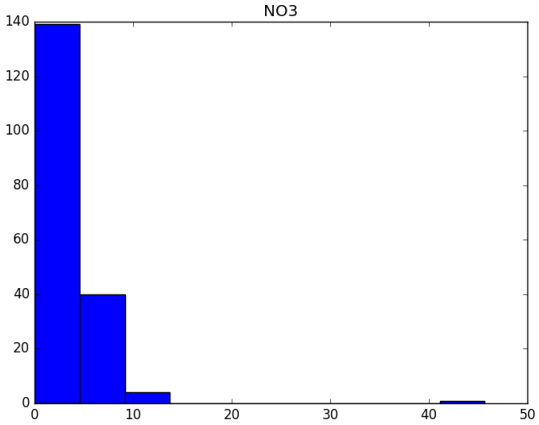












- qq 图

比较已知样本的分布和猜测分布的图，猜测的概率分布通常为正态分布。比如猜测样本是正态分布的，则有：假设样本有 n 个，则用标准正态分布函数获取 n 个分位值。取法是:

Q-Q 图原理方法简介

1.分位数

设 $x_1,x_2,\cdots,x_n$ 为来自标准正态分布总体 X 的 n 个样本，分位数 q_i 定义如下：

$P\{X \le q_1\} = \int_{\infty}^{q_1} \frac{1}{\sqrt{2x}} e^{-\frac{x^2}{2}} dx = p_i$，这里 $p_i = \left(i - \frac{1}{2}\right)/n, i = 1,2,\cdots,n$ 显然 q_i 由 p_i 所唯一确定。

2.Q-Q 图

由统计学可以证明：若数据 $x_1,x_2,\cdots,x_n$ 的分布与正态分布非常接近，则点 $(q_i,x_i)(i=1,2,\cdots,n)$ 应大致成一条直线。

3.作 Q-Q 图步骤

a.将原始观察数据依小到大排成顺序 $x_{(1)},x_{(2)},\cdots,x_{(n)}$，与它们相对应的概率值为

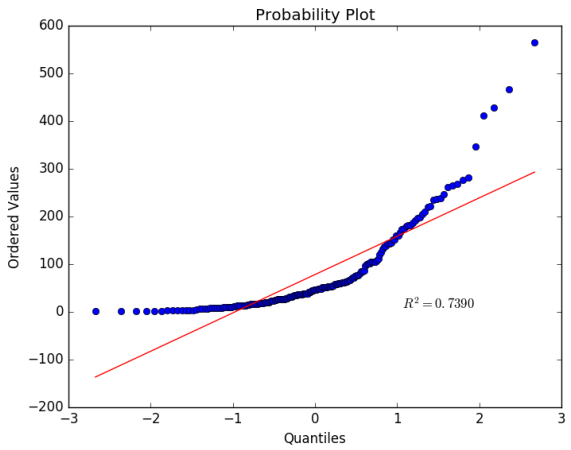
$P_1 = \left(1 - \frac{1}{2}\right)/n$，$P_2 = \left(2 - \frac{1}{2}\right)/n$，…，$P_n = \left(n - \frac{1}{2}\right)/n$；

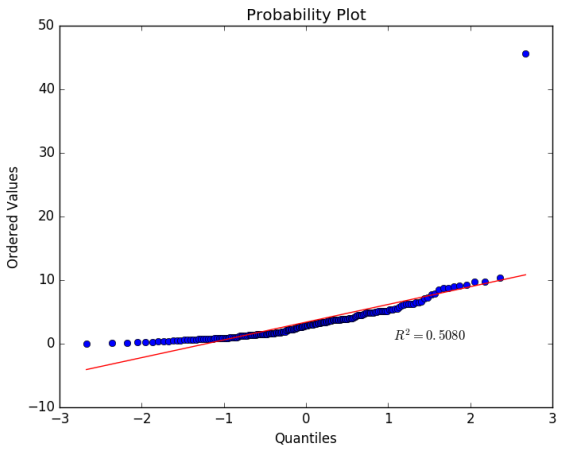
b.计算标准正态分位数 $q_1,q_2,\cdots,q_n$；

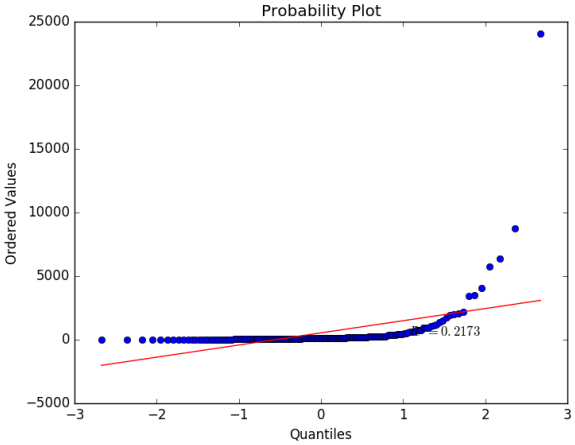
c.把数对 $(q_i,x_i)(i=1,2,\cdots,n)$ 画在坐标平面上，并观察它们是否成直线。

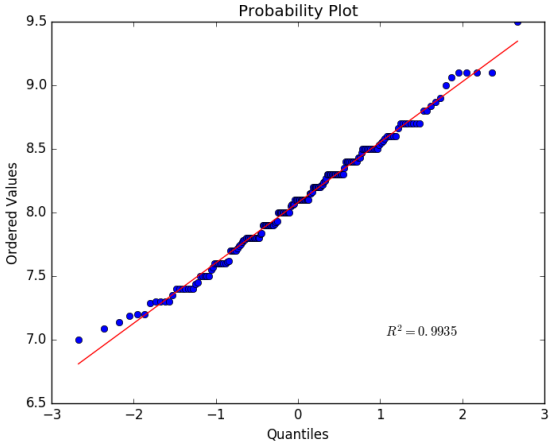
将样本和这个 n 个值都从小到大排列，一一对应。这样就能获得 n 对坐标。标准正态分布函数生成的值作 x，样本值作 y，则可在直角坐标系中绘制出 n 个点。**如果所有点连成的线越接近直线 y=x，那么就能说样本分布越近似猜测的分布**。

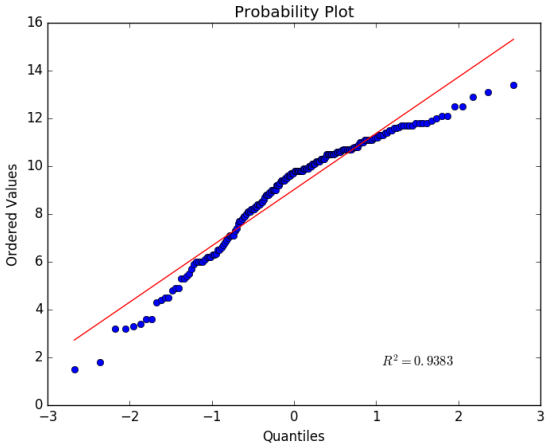
python 的 SciPy 库中已经实现了画 qq 图的函数，就是 stats.probplot()函数。

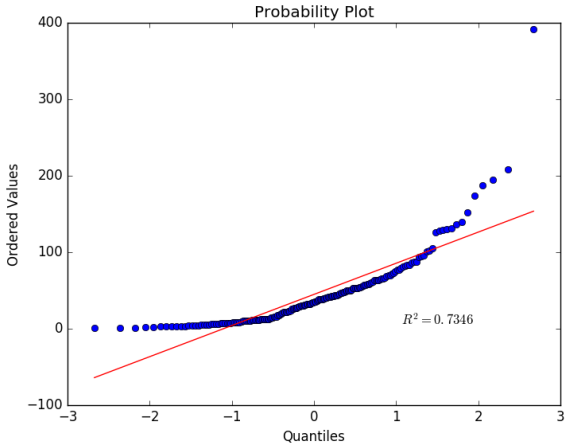


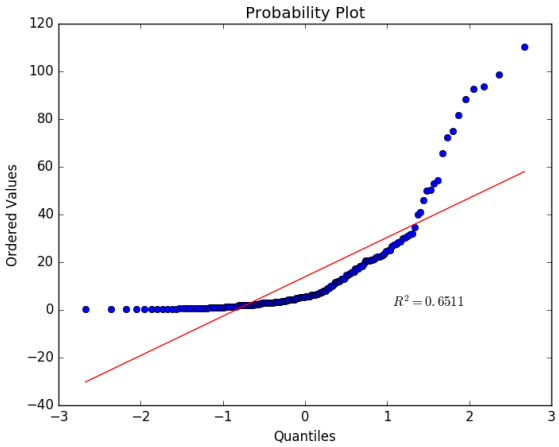


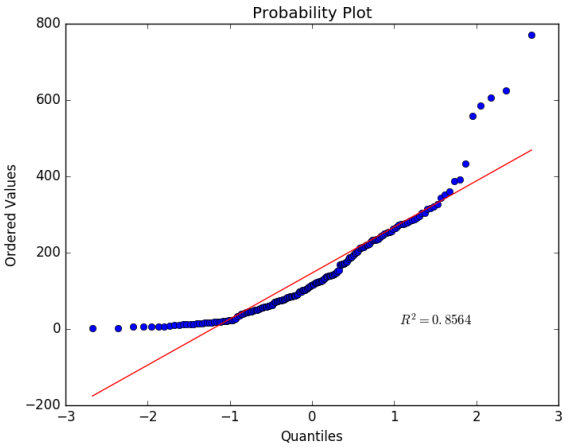




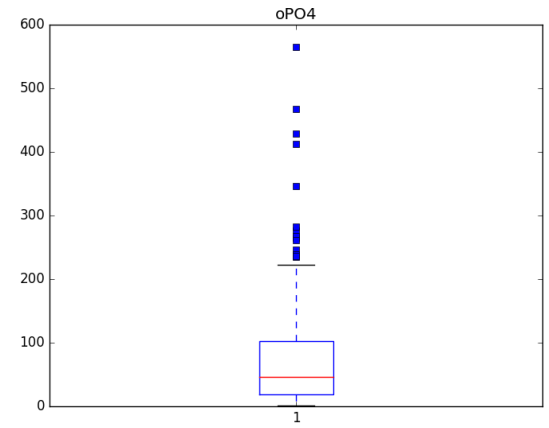


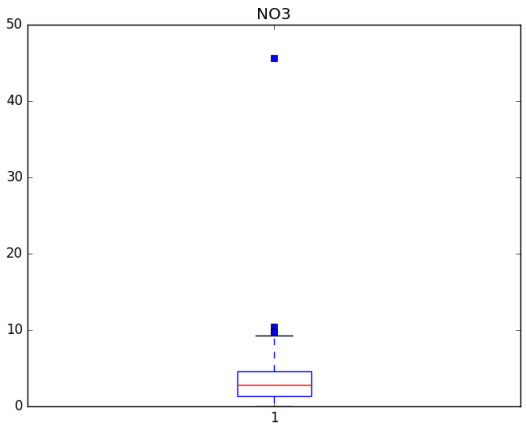


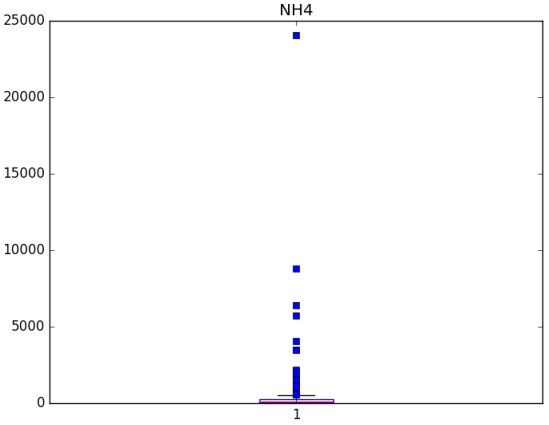


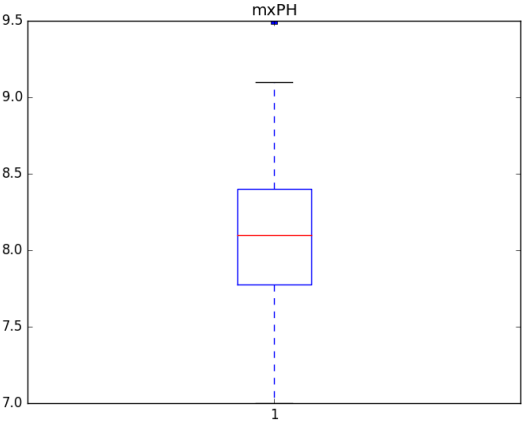


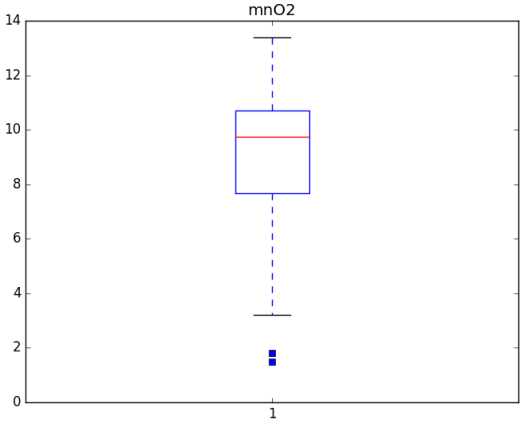
- 盒图，离群值进行识别
  盒图使用 matplotlib 库中的 boxplot 函数实现。

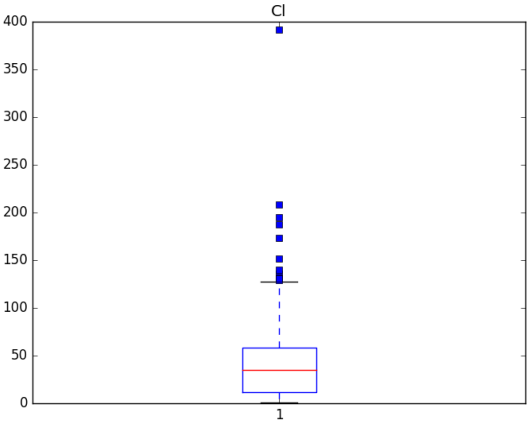

oPO4
600
500
400
300
200
100
0
1

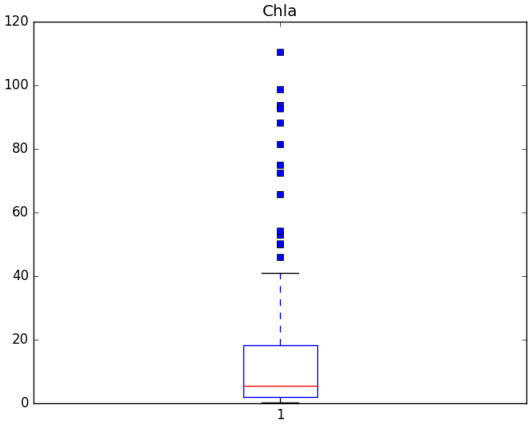

NO3
50
40
30
20
10
0
1

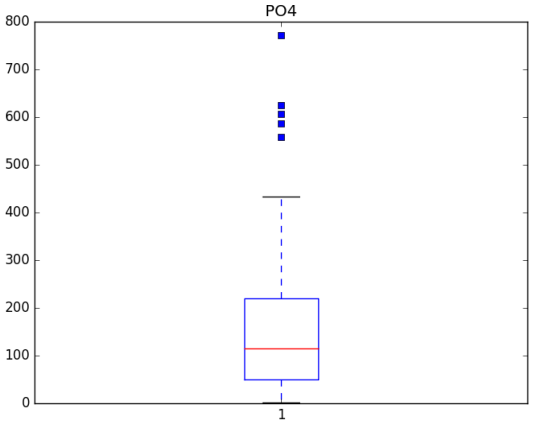

NH4
25000
20000
15000
10000
5000
0
1


mxPH
9.5
9.0
8.5
8.0
7.5
7.0
1

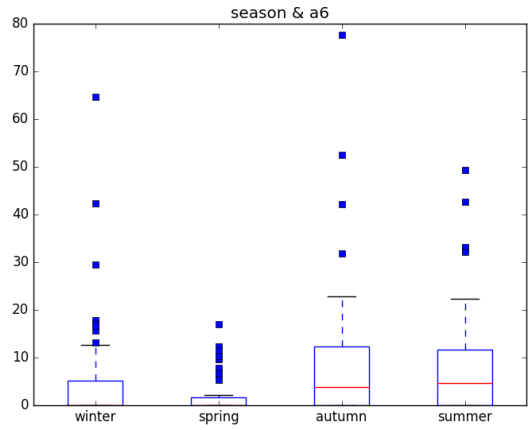

mnO2
14
12
10
8
6
4
2
0
1

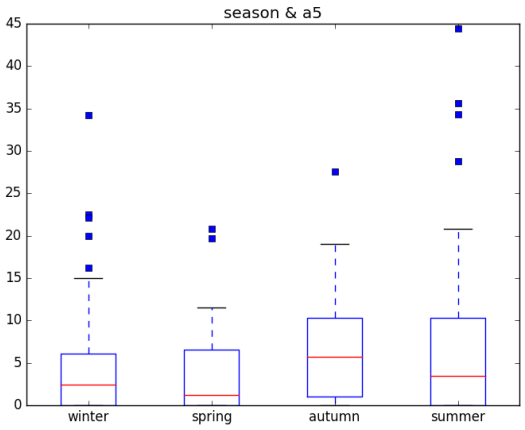

Cl
400
350
300
250
200
150
100
50
0
1

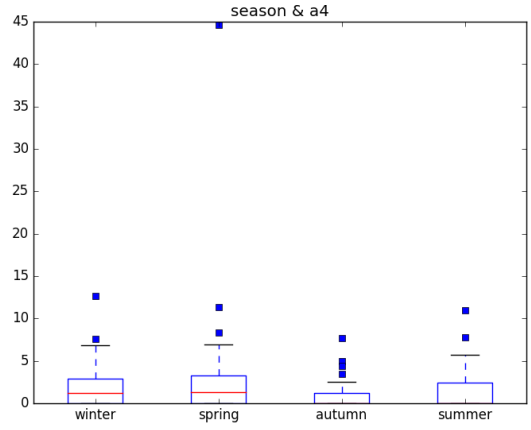


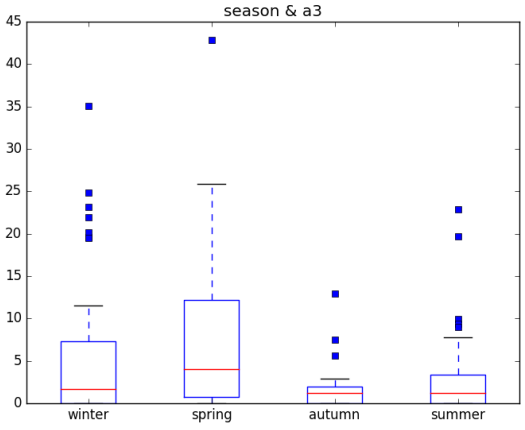


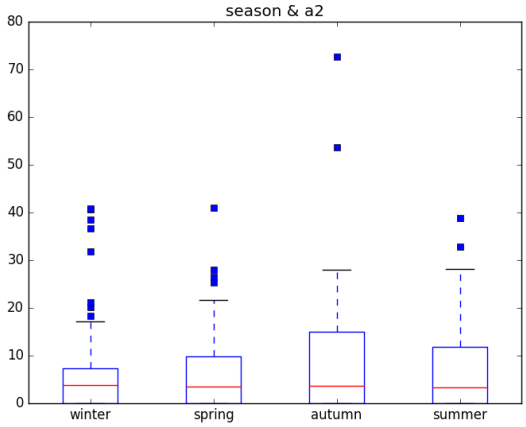
- 数量与标称变量的条件盒图
  - 季节

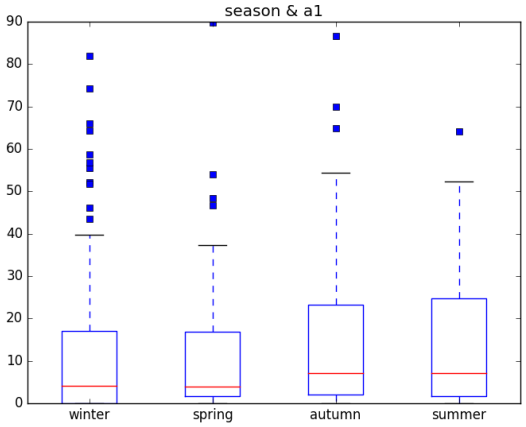


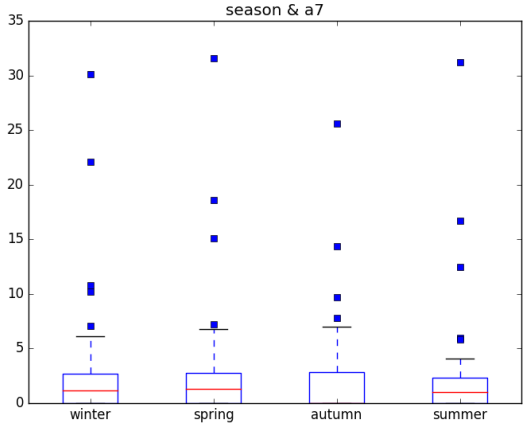




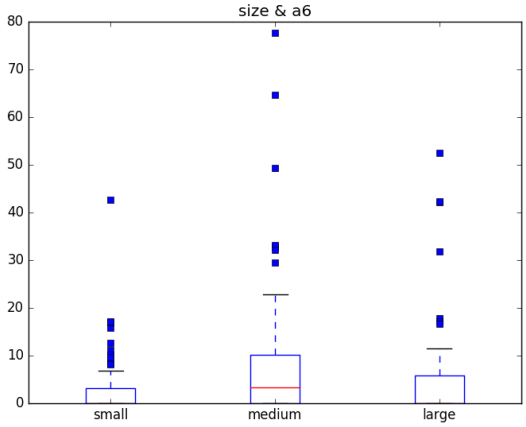


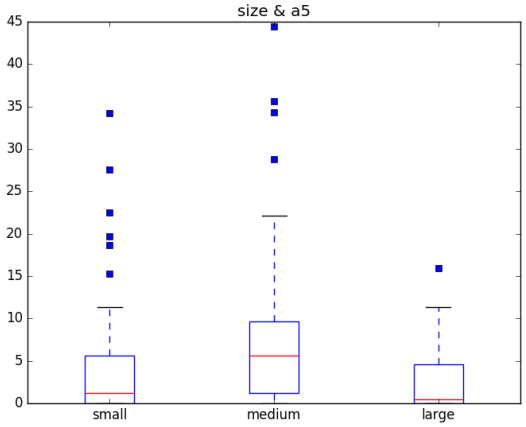

season & a2
winter
spring
autumn
summer

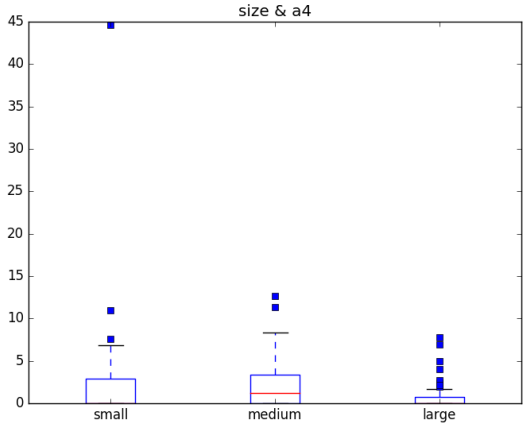

season & a1
winter
spring
autumn
summer

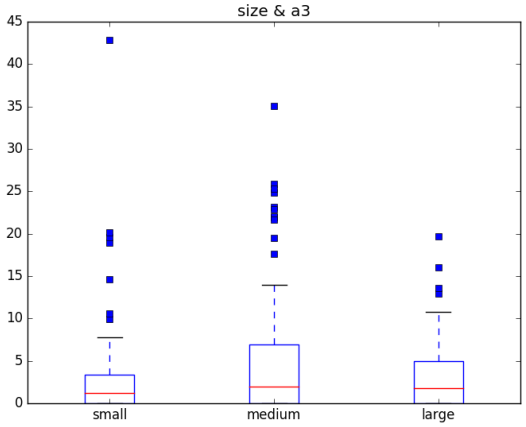

season & a7
winter
spring
autumn
summer

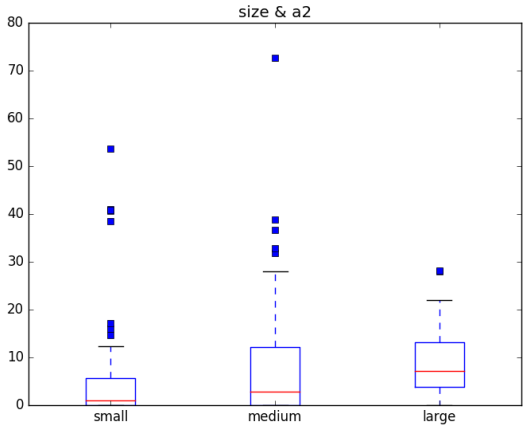
■ 大小

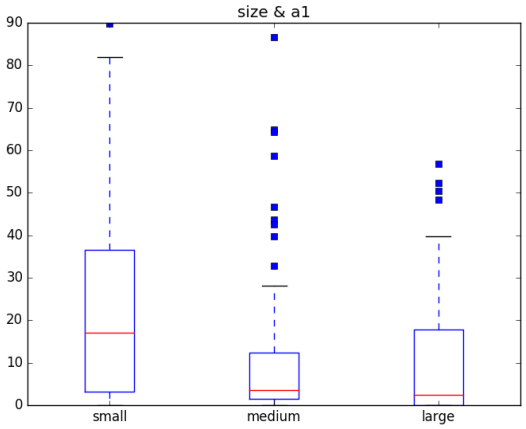

size & a6
small
medium
large

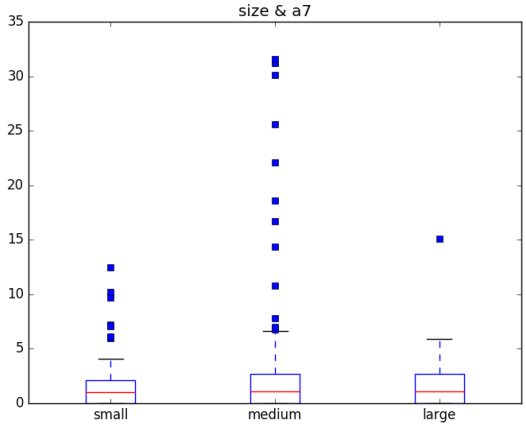

size & a5
small
medium
large


size & a4
45
40
35
30
25
20
15
10
5
0
small
medium
large

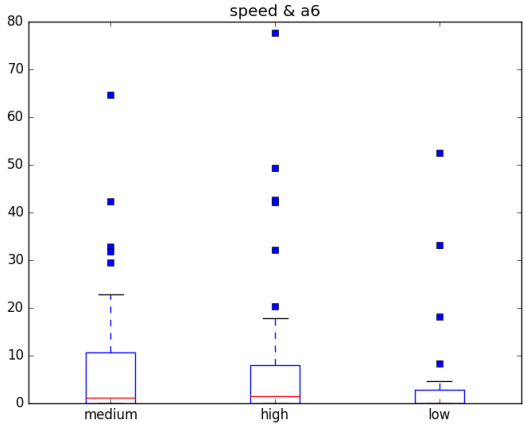

size & a3
45
40
35
30
25
20
15
10
5
0
small
medium
large

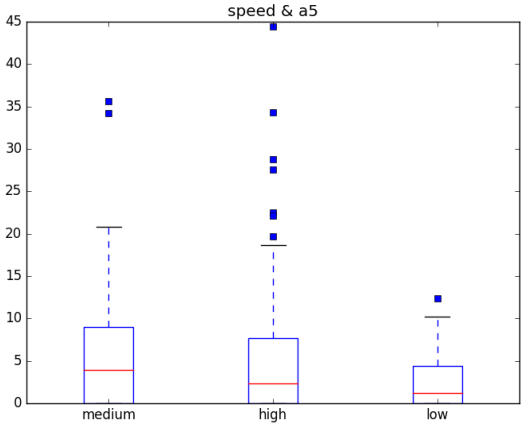

size & a2
80
70
60
50
40
30
20
10
0
small
medium
large

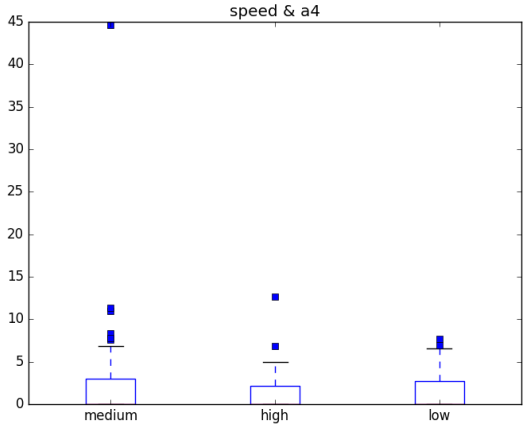

size & a1
90
80
70
60
50
40
30
20
10
0
small
medium
large

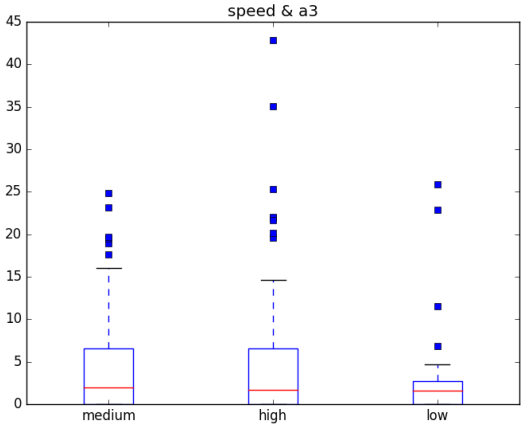

size & a7
35
30
25
20
15
10
5
0
small
medium
large

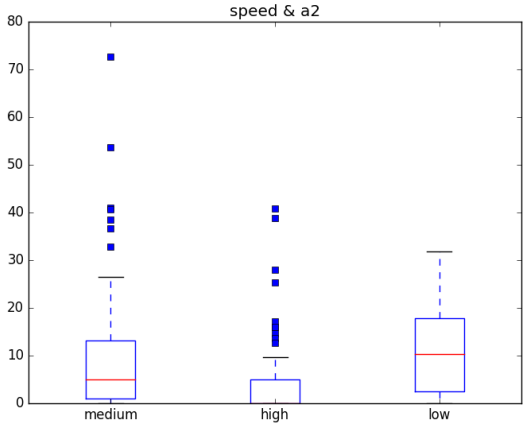
■ 速度

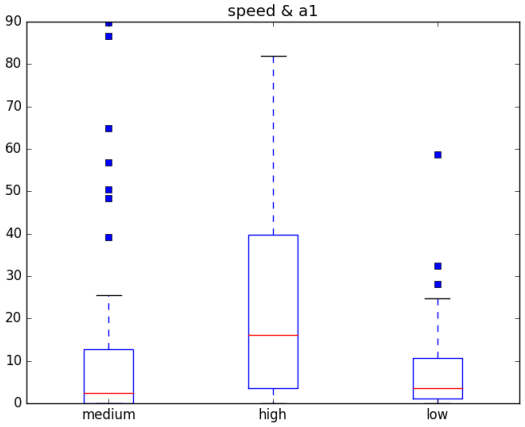

speed & a6
80
70
60
50
40
30
20
10
0
medium
high
low

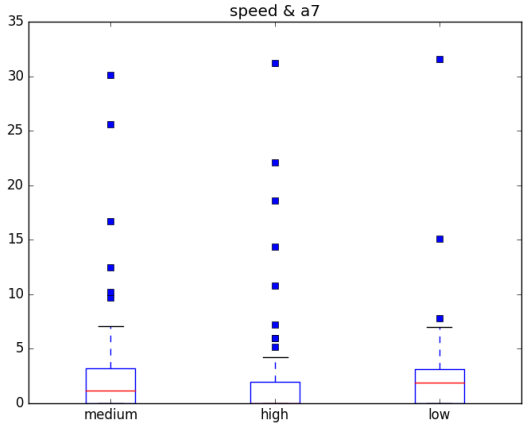

speed & a5
45
40
35
30
25
20
15
10
5
0
medium
high
low


speed & a4
45
40
35
30
25
20
15
10
5
0
medium
high
low


speed & a3
45
40
35
30
25
20
15
10
5
0
medium
high
low


speed & a2
80
70
60
50
40
30
20
10
0
medium
high
low


speed & a1
90
80
70
60
50
40
30
20
10
0
medium
high
low



缺失数据处理

由于篇幅有限，本节生成的图就不展示了，又需要可以运行我的程序获取。

- 将缺失部分剔除
  使用 DataFrame 的 dropna 方法实现。上面就是使用剔除缺失数据得到的结果。
- 用最高频率值来填补缺失值
  简单的统计数据然后填补。
- 通过属性的相关关系来填补缺失值
  因为缺失数据过多的条目的利用价值不大，所以我将第 61 行和 198 行的数据去掉。把其他的确实值填补好。
  首先，相关系数求取可以通过 DataFrame 的 corr 方法取得。结果如下：

	mxPH	mnO2	Cl	NO3	NH4	oPO4	PO4	Chla
mxPH	1.000000	-0.168612	0.136108	-0.130981	-0.093536	0.158999	0.189908	0.445962
mnO2	-0.168612	1.000000	-0.278333	0.099444	-0.087478	-0.416163	-0.487486	-0.153265
Cl	0.136108	-0.278333	1.000000	0.225041	0.071913	0.391054	0.457449	0.149856
NO3	-0.130981	0.099444	0.225041	1.000000	0.721444	0.144588	0.168601	0.139679
NH4	-0.093536	-0.087478	0.071913	0.721444	1.000000	0.227237	0.208180	0.088947
oPO4	0.158999	-0.416163	0.391054	0.144588	0.227237	1.000000	0.914365	0.115621
PO4	0.189908	-0.487486	0.457449	0.168601	0.208180	0.914365	1.000000	0.253621
Chla	0.445962	-0.153265	0.149856	0.139679	0.088947	0.115621	0.253621	1.000000

把这个结果看成矩阵，矩阵的对角线值为 1。这是因为某个属性与自己的相关系数为 1。接着使用相关系数最大的列来填补缺失值。

填补方法包括直接填补和拟合函数，其中拟合函数也包括线性拟合和非线性拟合，这可以通过一些插值运算来完成，为简便考虑，我使用了直接填补方法。

- 通过数据对象之间的相似性来填补缺失值
  数据相似性填充的前提是计算数据行之间的相似性，因为数据行之间基本都是数值，所以我考虑向量的相似性模型，直观的方法就是使用欧几里得距离作为度量方法：

$$\mathrm{dist}(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

对于包含 NaN 的向量，对应 NaN 的位不考虑。对于选取到的填充标准，我采用直接填充的方法。其他的方法也包括用函数拟合去实现，如果希望增强填充值的健壮性，可以使用 knn 的思想，即选取最接近的 k 个填充值，然后求取平均数。

关于数据填充，在数据量大的情况下，应该将有缺失值的数据直接删除，因为采用填充的方法可能会对数据的产生不良影响。