# 作业一 海藻数据的分析

**姓名：张燕妮**

**学号：2120151065**

1. **问题描述**

**某些高浓度的有害藻类对河流生态环境的破坏是一个严重的问题。它们不仅破坏河流的生物，也破坏水质。能够监测并在早期对海藻的繁殖进行预测对提高河流质量是很有必要的。**

**针对这一问题的预测目标，在大约一年的时间内，在不同时间内收集了欧洲多条河流的水样。对于每个水样，测定了它们的不同化学性质以及7种有害藻类的存在频率。在水样收集过程中，也记录了一些其他特性，如收集的季节、河流大小和水流速度。**

**二、数据说明**

**有200个水样，每条记录是同一条河流在该年的同一个季节的三个月内收集的水样的平均值。**

**每条记录由11个变量构成，3个是标称变量，分别描述水样收集的季节，河流大小和河水速度，剩下的8个变量是水样的化学参数：**

* **最大pH值(mxPH)**
* **最小含氧量(mnO2)**
* **平均氯化物含量(Cl)**
* **平均硝酸盐含量(NO3)**
* **平均氨含量(NH4)**
* **平均正磷酸盐含量(oPO4)**
* **平均磷酸盐含量(PO4)**
* **平均叶绿素含量(Chla)**

**a1-a7为7种不同有害藻类在相应水样中的频率数目。**

**三、数据分析要求**

**1. 数据可视化和摘要**

**(1) 数据摘要**

* **对标称属性，给出每个可能取值的频数，**
* **数值属性，给出最大、最小、均值、中位数、四分位数及缺失值的个数。**

**(2) 数据的可视化**

**针对数值属性，**

* **绘制直方图，如mxPH，用qq图检验其分布是否为正态分布。**
* **绘制盒图，对离群值进行识别**

**对7种海藻，分别绘制其数量与标称变量，如size的条件盒图**

**2. 数据缺失的处理**

**分别使用下列四种策略对缺失值进行处理:**

* **将缺失部分剔除**
* **用最高频率值来填补缺失值**
* **通过属性的相关关系来填补缺失值**
* **通过数据对象之间的相似性来填补缺失值**

**处理后，可视化地对比新旧数据集。**

**四、实验环境及语言**

**语言及环境依赖**

语言： python

依赖的包：xlrd, pylab, matplotlib, scipy, numpy

xlrd: 数据摘要处理时用到

pylab, matplotlib, scipy, numpy：数据可视化时用于生成图

**五、实现方法**

**5.1 数据摘要**

**5.1.1 概述**

数据摘要部分主要进行的处理有以下两个方面：

1. 对标称属性，给出每个可能取值的频数

2. 数值属性，给出最大、最小、均值、中位数、四分位数及缺失值的个数

**5.1.2 实现方法**

语言：python

结果：json，图

依赖包：xlrd

**（1）. 把原始数据导入excel表格\*\***

通过excel表格的“数据”-->“来自文本” 导入

**（2）. 利用python脚本实现数据摘要的获得\*\***

脚本为：statistics.py ，该脚本涉及到的关键函数如下：

init() #从excel获取数据，分离出每一个属性的数据集

nominalDataFrequency #对标称属性，给出每个可能取值的频数，

statistic(chemicalParameters,frequency) #数值属性，给出最大、最小、均值、中位数、四分位数及缺失值的个数

**（3）. 统计结果保存为json数据**

nominalDataFrequency.json 是标称属性每个可能取值的频数统计结果

其格式如下：

```

{

"riverSize": {

"small": 71,

"large": 45,

"medium": 84

},

"season": {

"autumn": 40,

"spring": 53,

"winter": 62,

"summer": 45

},

"riverSpeed": {

"high": 84,

"medium": 83,

"low": 33

}

}

```

**"riverSize"下记录的是河流大小包含的三个属性各自的频数**

**"season"下记录的是水样收集的季节，包含的四个属性各自的频数**

**"riverSpeed"下记录的是河水速度包含的三个属性各自的频数**

**statistic\_max\_min\_etc.json 是数值属性，最大、最小、均值、中位数、四分位数及缺失值的个数的统计结果**

其格式如下：

```

{

"Cl": {

"min": 最大值,

"max": 最小值,

"midian": 中位数,

"quartiles": [

四分位数（三个值）

],

"miss\_num": 1缺失值个数

"mean": 均值

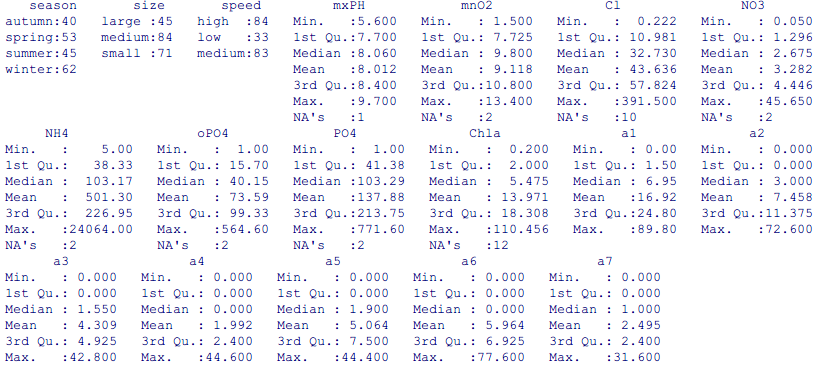
},

"mxPH":{同上格式},

......

}

```



**5.2 数据可视化**

**5.2.1 概述**

数据可视部分主要进行的处理是把数值部分的每一个属性包含的数据分别可视化（绘制成直方图，qq图，盒图）：

**5.2.2 实现方法**

语言：python

结果：直方图，qq图，盒图

依赖包：pylab, matplotlib, scipy, numpy

**（1）. 提取出数值数据，以json格式保存**

利用“数据摘要”的方法，把原数据中的数值数据提取出来，以json格式保存，结果是[data.json](https://github.com/jennyzhang8800/DataMining/blob/master/作业一/数据可视化/结果/data.json)

此部分用到的脚本为[cleaning.py](https://github.com/jennyzhang8800/DataMining/blob/master/作业一/数据可视化/程序/cleaning.py)

**（2）. 绘制直方图，qq图，盒图**

+ 直方图的绘制方法：主要用到的是pylab.hist()方法

+ qq图的绘制方法：主要用到的是scipy.stats.probplot方法

+ 盒图的给制方法：主要用到的是boxplot()方法

代码详见[show.py](https://github.com/jennyzhang8800/DataMining/blob/master/作业一/数据可视化/程序/show.py)

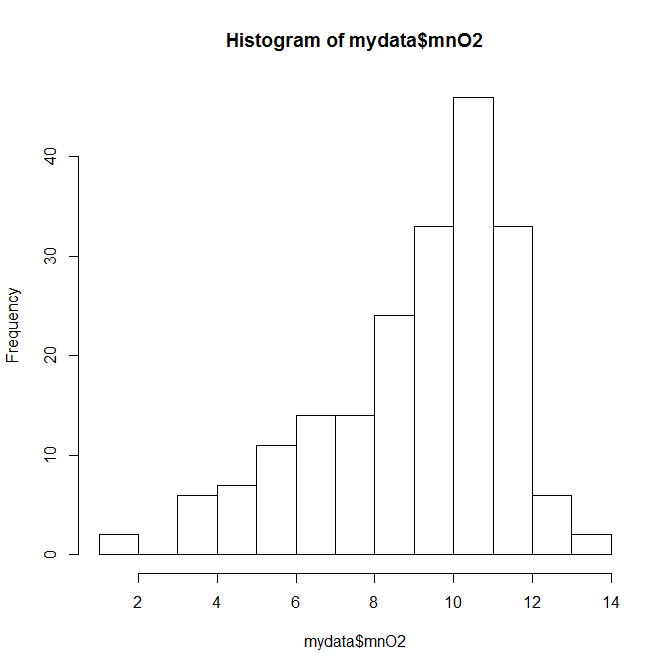
**（3）. 结果展示**

结果图分别以svg和jpg两种格式保存

(https://github.com/jennyzhang8800/DataMining/tree/master/作业一/数据可视化/结果/plot) 2



**mxPH的直方图与QQ图**



**mnO2的直方图与QQ图**



**Cl的直方图与QQ图**



**NO3的直方图与QQ图**



**NH4的直方图与QQ图**



**oPO4的直方图与QQ图**



**PO4的直方图与QQ图**



**Chla的直方图与QQ图**

**绘制盒图，识别离群点**

**mxPH盒图 mnO2盒图**



**Cl盒图** **NO3盒图**



**NH4** **oPO4**



**PO4盒图** **Chla盒图**

**依次绘制a1-a7海藻的条件盒图，如图所示：**



**a1海藻与河流大小的条件盒图 a2海藻与河流大小的条件盒图**

**a3海藻与河流大小的条件盒图 a4海藻与河流大小的条件盒图**

**a5海藻与河流大小的条件盒图 a6海藻与河流大小的条件盒图**



**a7海藻与河流大小的条件盒图**

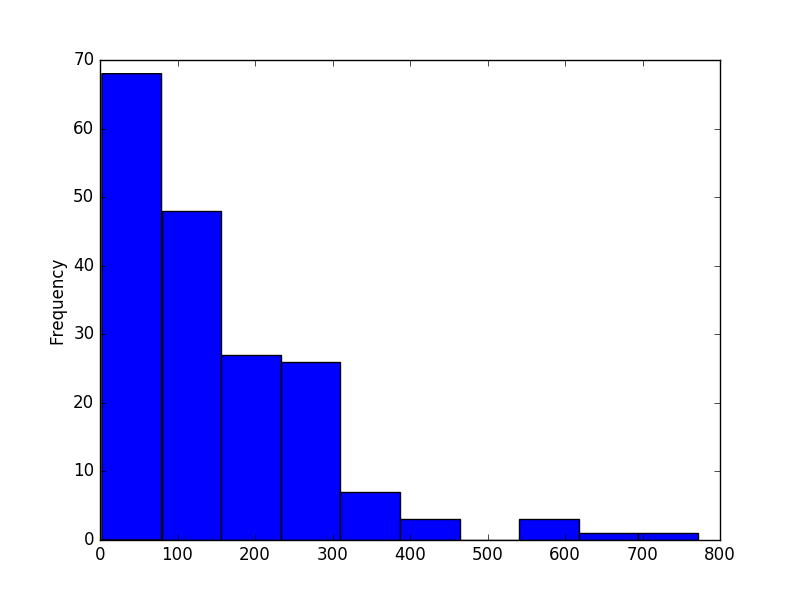
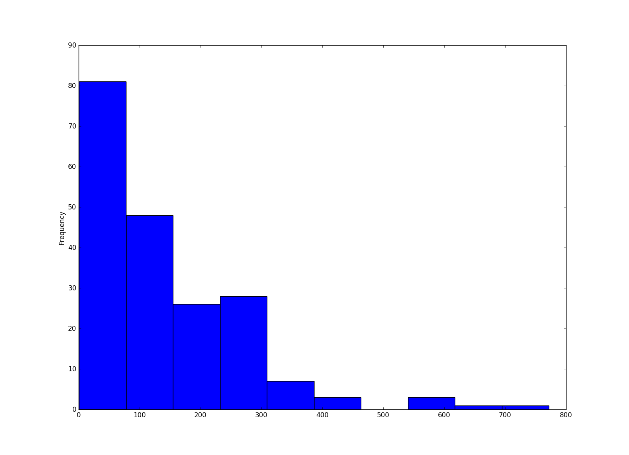
**从图中可以看出a1有更高的频数，但是 a3,a5,a6在中型河流中更多一些。**

**5.3. 数据缺失的处理**

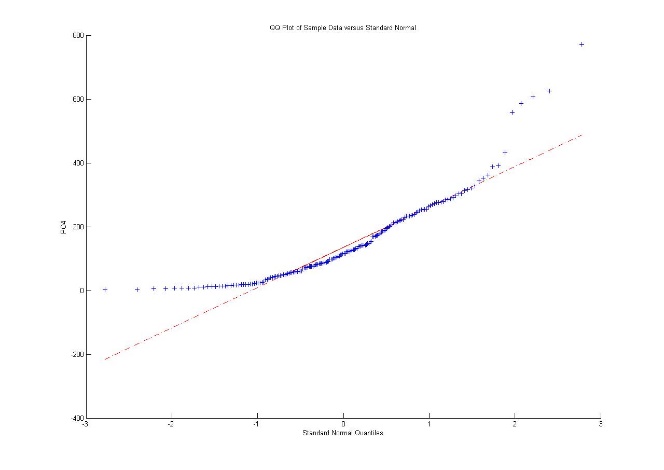
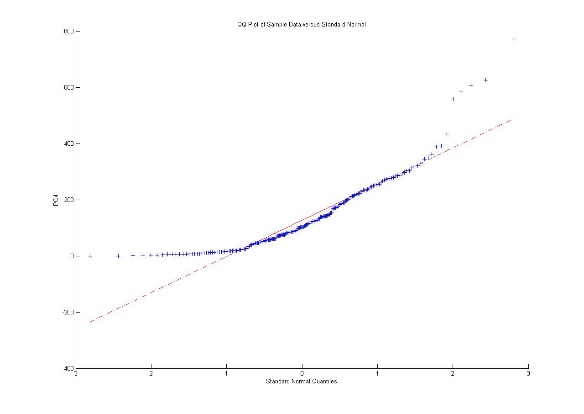
**1） 将缺失部分剔除**

**剔除缺失数据与写入文件的命令如下：**

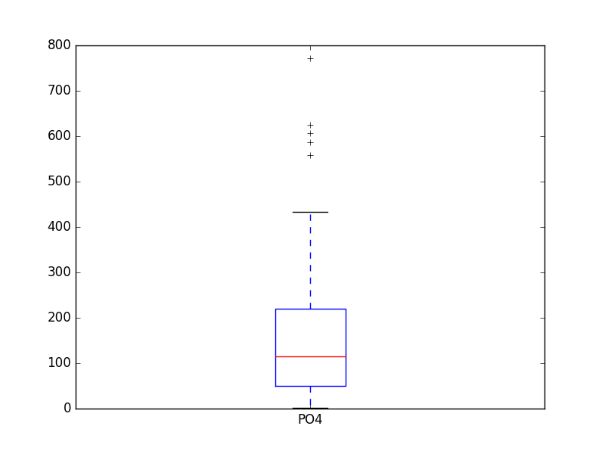
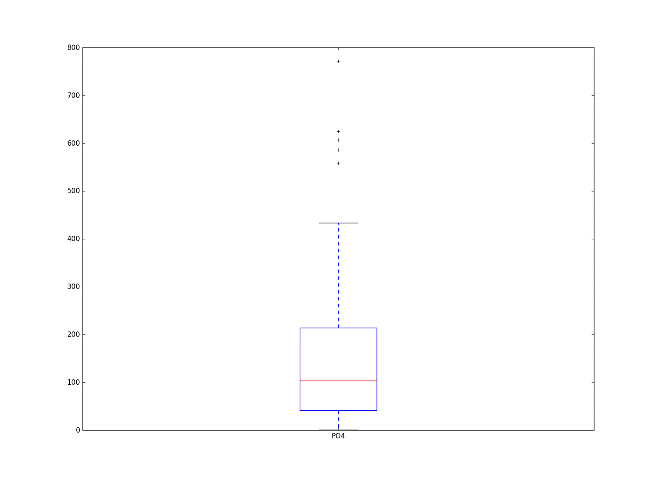
|  |
| --- |
| omitdata = na.omit(mydata) 剔出缺失数据  write.table(omitdata,'OmitedData.txt',col.names = F,row.names = F, quote = F) 写入文件 |



处理前后PO4的直方图对比



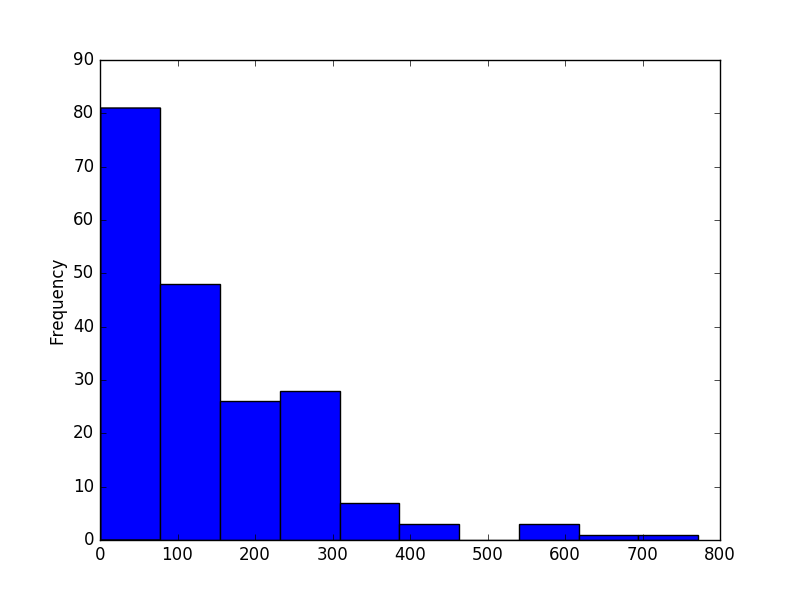
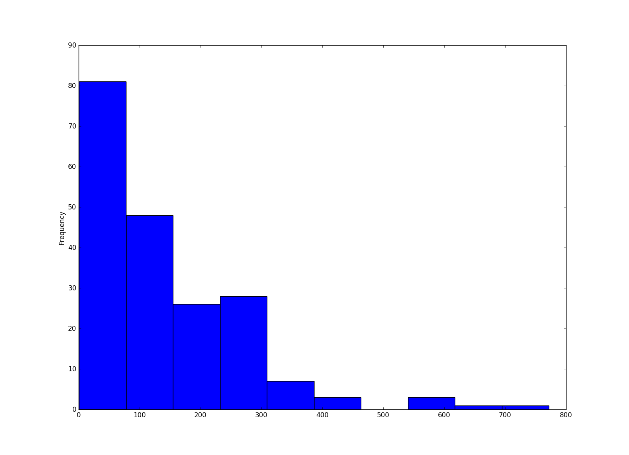
处理前后PO4属性的qq图对比



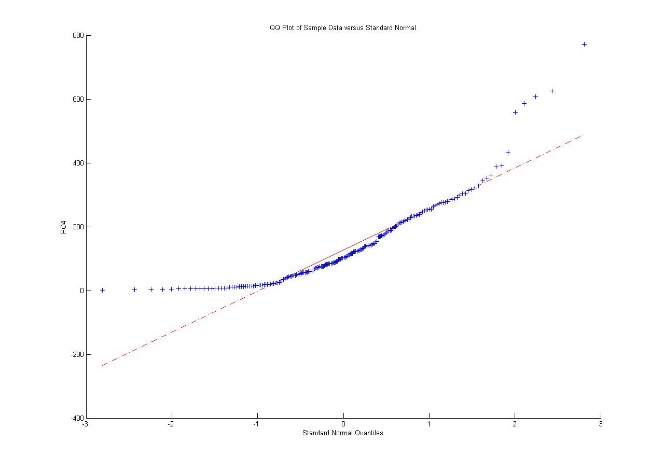
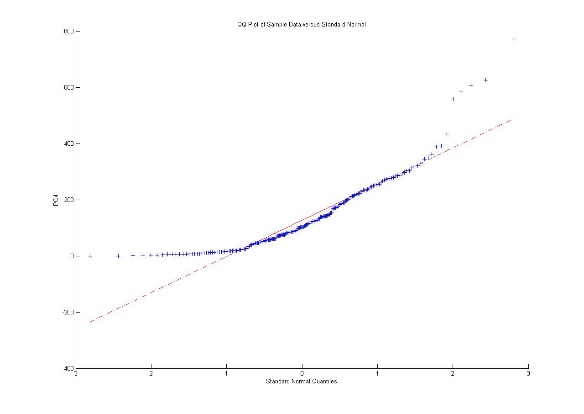
处理前后PO4属性的盒图对比

**2) 使用高频数值来填补缺失值**

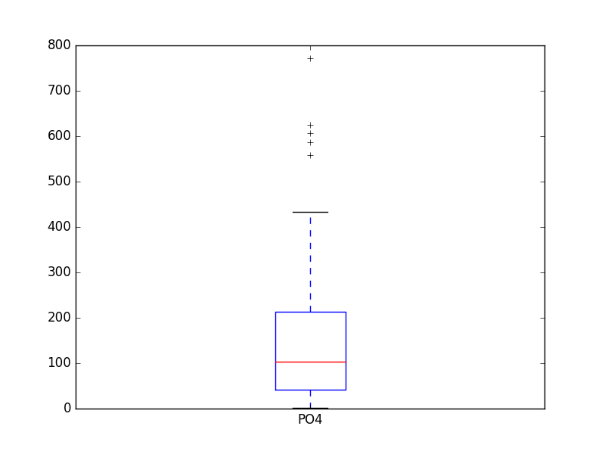
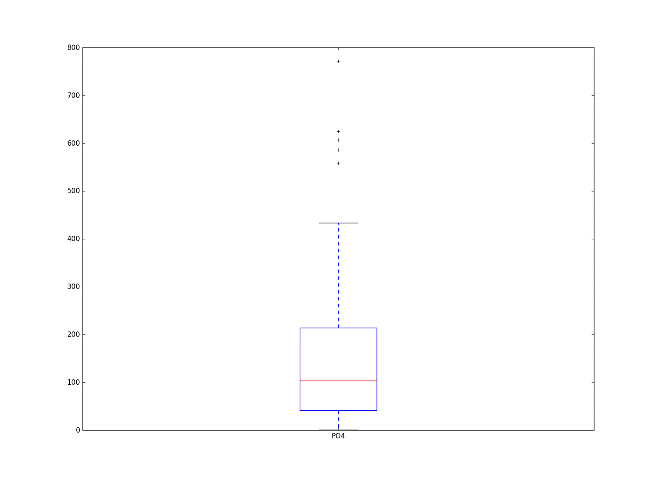
|  |
| --- |
| library(DMwR)  preprocess2 = mydata[-manyNAs(mydata),]  preprocess2 = centralImputation(Preprocess2)  write.table(preprocess2,'D:/DataMining/CentralImputationData.txt',col.names = F,row.names = F, quote = F) |



处理前后PO4的直方图对比



处理前后PO4属性的qq图对比

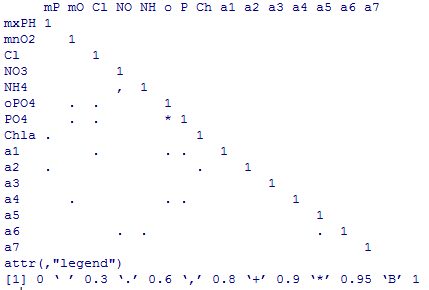


处理前后PO4属性的盒图对比

**3) 通过属性的相关关系来填补缺失值**

|  |
| --- |
| symnum(cor(mydata[,4:18],use='complete.obs')) |

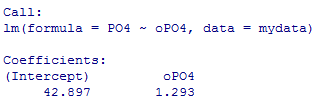
**得到属性之间的相关性如下图：**



**从图中可以看出oPO4与PO4相关度超过0.9，所以可以用这两个属性作相关分析，互相填补缺失数据。**

**使用以下代码得到其线性模型，如图所示为oP04和P04的模型分析：**

|  |
| --- |
| lm(formula=PO4~oPO4, data=mydata) |

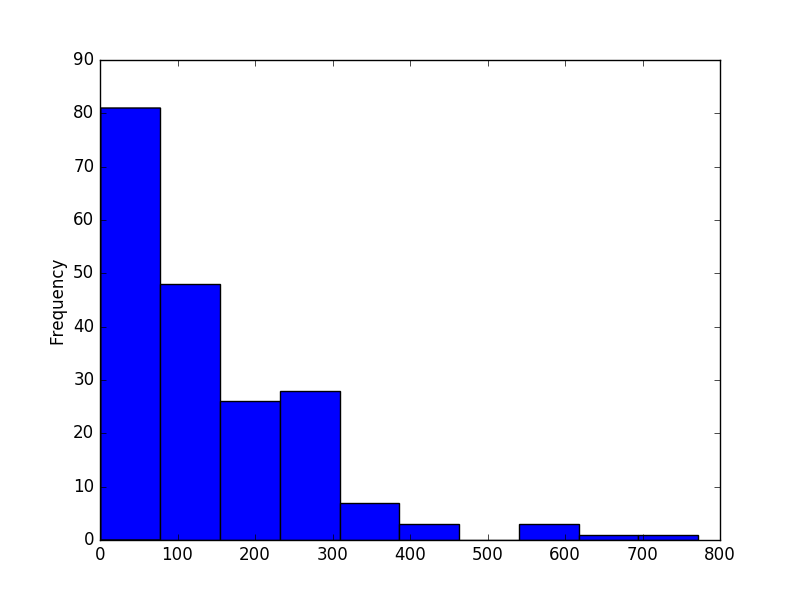
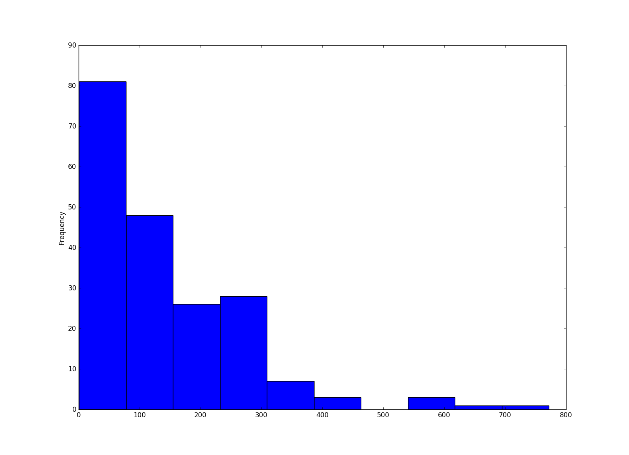


**oPO4与PO4的线性模型分析**

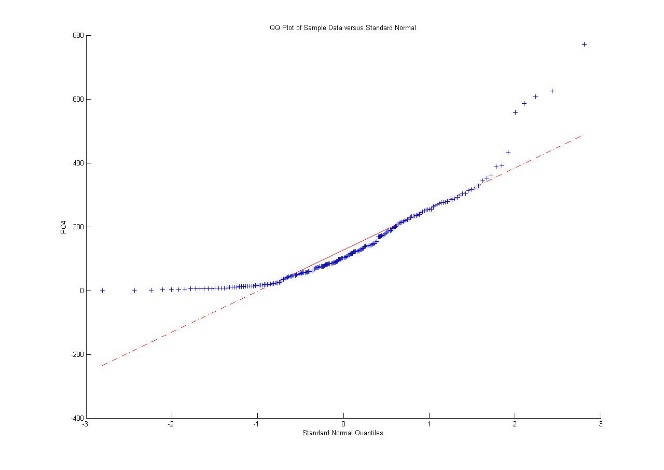
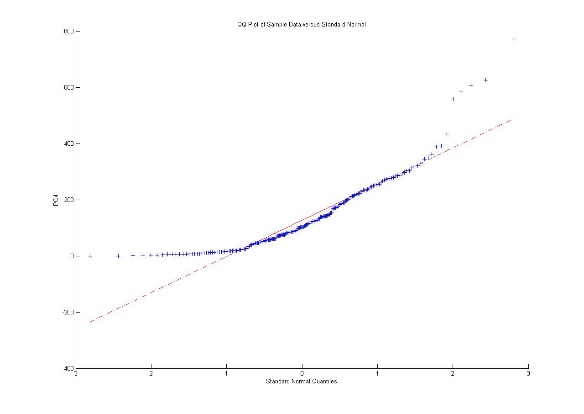
**得到结果如上图所示，表示得到的线性模型为PO4=42.897 + oPO4\* 1.293。**

**使用线性模型来填充PO4与oPO4的数据：**

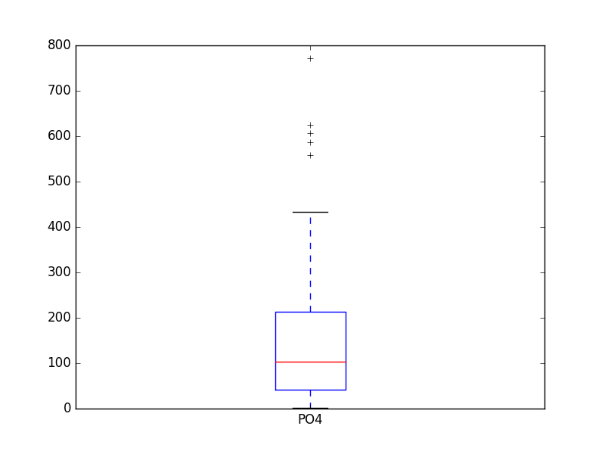
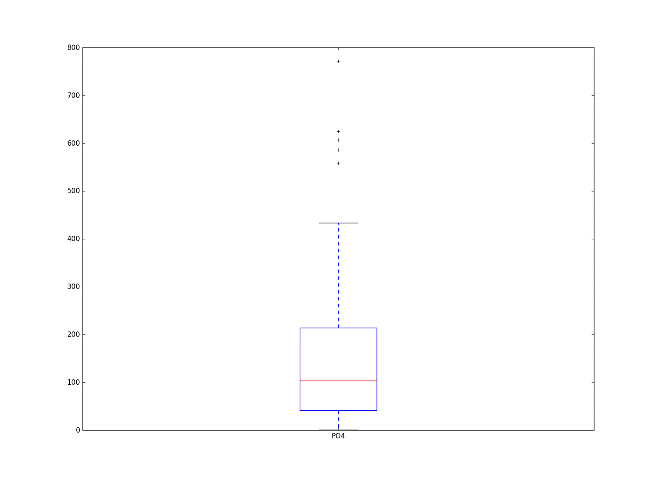
|  |
| --- |
| preprocess3 = mydata[-manyNAs(mydata),]  fillPO4 <- function(oP){  if(is.na(oP))  return(NA)  else return (42.897 + 1.293 \* oP)  }  preprocess3[is.na(preprocess3$PO4),'PO4'] = sapply(preprocess3[is.na(preprocess3$PO4),'oPO4'],fillPO4) |



处理前后PO4的直方图对比



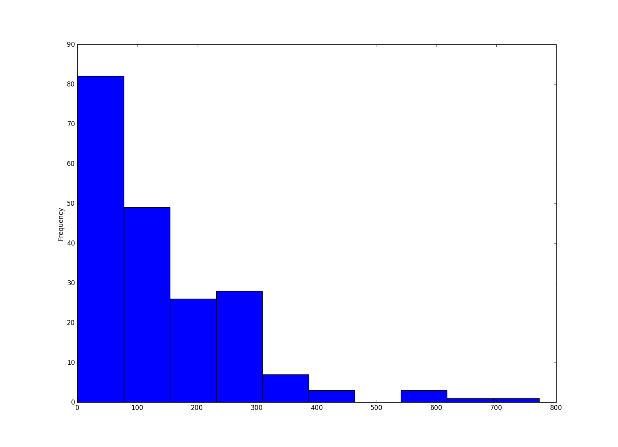
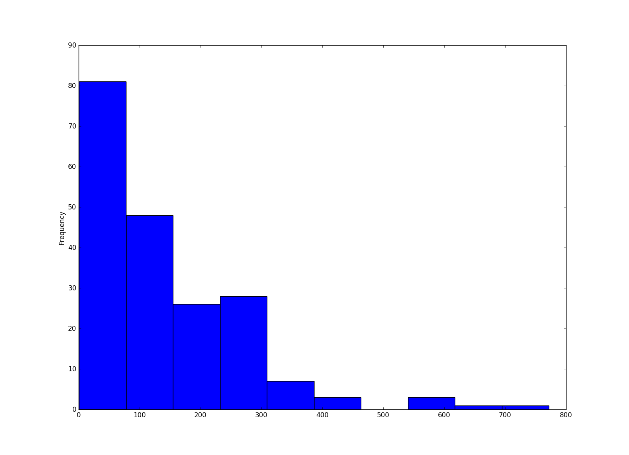
处理前后PO4属性的qq图对比



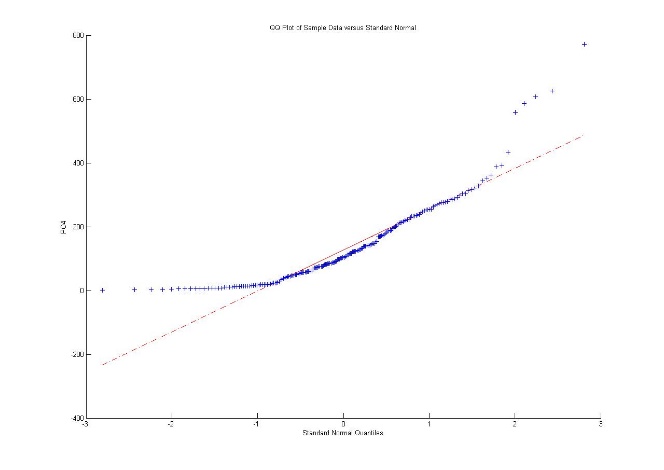
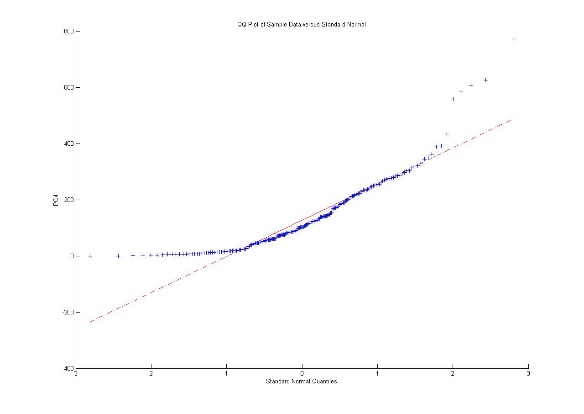
处理前后PO4属性的盒图对比

**4） 使用数据对象之间的相似型填补缺失值**

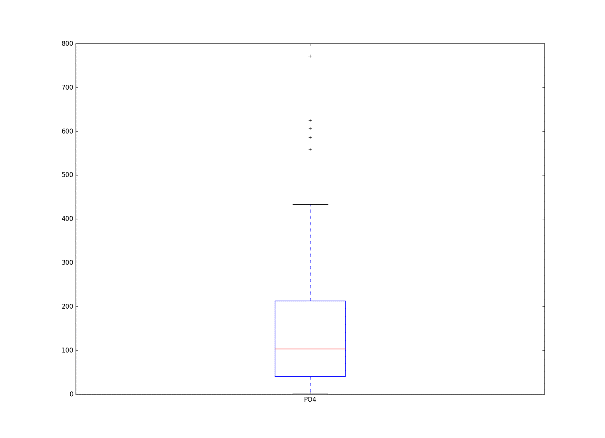
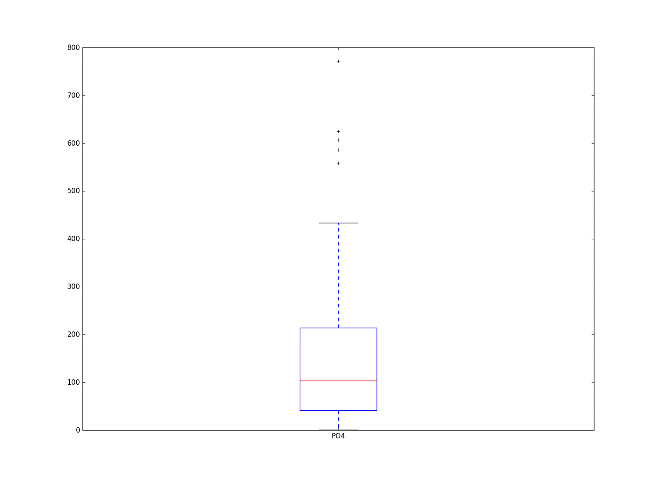
|  |
| --- |
| preprocess4 = knnImputation(mydata,k=10)  write.table(preprocess4,'D:/Mine/Study/DataMining/knnImputationData.txt',col.names = F,row.names = F, quote = F) |



处理前后PO4的直方图对比



处理前后PO4属性的qq图对比



处理前后PO4属性的盒图对比