马的疝病分析报告

姓名：许志凤

学号：2120161066

1. 问题描述

疝病是描述马胃肠痛的术语，这种病不一定源自马的胃肠问题，其他问题也可能引发马疝病。所给数据集是医院检测的一些指标。

2. 数据说明

共 368 个样本，28 个特征。

每条记录由 28 个变量组成，包括：21 个标称型变量和 7 个数值型变量

1. 手术(surg)？ 1 =是的，有手术；2 =未经手术治疗
2. 年龄(age)？ 1 =成人马；2 =年轻（<6 个月）
3. 医院号码(HN)
4. 直肠温度(rt)
5. 脉冲(pulse)
6. 呼吸频率(rr) 正常值是 8 到 10
7. 四肢温度(toe)？1=正常； 2=暖； 3=酷； 4=冷。
8. 外设脉冲(pp)？1=正常； 2=增加； 3=减少； 4=缺席。
9. 粘膜(mm)？1 =正常粉红色； 2 =明亮的粉红色； 3 =淡粉红色； 4 =淡紫色； 5 =亮红/注射； 6 =暗紫色。
10. 毛细管再填充时间(crt)？1 = <3 秒； 2 => = 3 秒。
11. 疼痛(pain) - 马的疼痛程度的主观判断？1 =警惕，没有痛苦；2 =郁闷；3 =间歇性轻度疼痛；4 =间歇性剧烈疼痛；5 =连续剧烈疼痛。
12. 蠕动(peris)？1 =超动力；2 =正常；3 = hypomotile ；4 =缺席
13. 腹胀(ad)？1 =无；2 =轻微；3 =中等；4 =严重
14. 鼻胃管(nt)？1 =无；2 =轻微；3 =显着
15. 鼻胃反流(nr)？1 =无；2 => 1 升；3 = <1 升

16. 鼻胃回流 PH(nrPH)？刻度为 0 到 14，7 为中性，正常值在 3 到 4 范围内

17. 直肠检查 - 粪便(ref)？1 =正常；2 =增加；3 =减少；4 =缺席- 无粪便可能表示阻塞

18. 腹部 (abd)？1 =正常；2 =其他；3 =大肠固体粪便；4 =扩大的小肠；5 =扩大的大肠

19. 填充细胞体积(pcv)：正常范围为 30 至 50.水平随着循环受损或动物脱水而升高。

20. 总蛋白质(tp)：正常值在 6-7.5（gms / dL）范围内

21. 腹腔镜外观(aa)？1 =清除；2 =多云；3 =血清浆

22. 腹部总蛋白(atp)

23. 结果(outcome)？1 =活着；2 =死亡；3 =被安乐死了

24. 手术病变(sl)？1 =是的；2 =否

25. 25,26,27(tol1,tol2,tol3)：病变类型第一个数字是病变部位 1 =胃 2 = sm 肠 3 = lg 冒号 4 = lg 结肠和盲肠 5 =盲肠 6 =横结肠 7 =回肠/降结肠 8 =子宫 9 =膀胱 11 =所有肠道部位 00 =无

    - 第二个数字是类型 1 =简单 2 =扼杀 3 =发炎 4 =其他

    - 第三个数字是亚型 1 =机械 2 =麻痹 0 = n / a

    - 第四个数字是具体代码 1 =封闭 2 =内在的 3 =外在的 4 =动力学 5 =旋转/扭转 6 = intussuption7 =血栓栓塞 8 =疝气 9 =脂肪瘤/狭窄监禁 10 =位移 0 = n / a

28. cp_data- 这是病例数据吗(cpd)？1 =是的;2 =否

3. 数据分析要求

3.1 数据可视化和摘要

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。

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

3.2 数据缺失的处理

数据集中有 30%的值是缺失的，因此需要先处理数据中的缺失值。

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。

4. 实验环境及语言

4.1 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

5、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：

```
{
    "aa": {
                "1.0": 41,
                "2.0": 48,
                "3.0": 46,
                "?": 165
                },
```

```
                "surg": {
                                        "1.0": 180,
                                        "2.0": 119,
                                        "?": 1
                                        },
}
```

手术(surg)是表示是否进行了手术 1 =是的，有手术；2 =未经手术治疗

年龄(age)是指马是否已成年 1 =成人马；2 =年轻（<6 个月）

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：

```
{
   "rr": {
            "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
            },
      "toe":{同上格式},
        ......
 }
```

具体数据如下：

```
{
 "rt": {
  "min": 36.0,
  "max": 39.7,
  "midian": 38.0,
  "quartiles": [
   37.7,
   38.0,
   38.3   ],
  "miss_num": 9,
  "mean": 37.99830508474576
 },
 "crt": {
  "min": 1.0,
  "max": 5.0,
  "midian": 3.0,
  "quartiles": [
   2.0,
   3.0,
   4.0   ],
  "miss_num": 8,
  "mean": 2.7333333333333334
 },
 "rr": {
  "min": 10.0,
  "max": 96.0,
  "midian": 28.0,
  "quartiles": [
   18.0,
   28.0,
   40.0   ],
  "miss_num": 13,
  "mean": 30.98181818181818
 },
 "tol3": {
  "min": 0.0,
  "max": 0.0,
  "midian": 0.0,
  "quartiles": [
   0.0,
   0.0,
   0.0   ],
  "miss_num": 0,
  "mean": 0.0 },
 "tol2": {
  "min": 0.0,
  "max": 3205.0,
  "midian": 0.0,
  "quartiles": [
   0.0,
   0.0,
   0.0   ],
  "miss_num": 0,
  "mean": 126.73529411764706
 },
 "tol1": {
  "min": 0.0,
  "max": 31110.0,
  "midian": 3111.0,
  "quartiles": [
   2111.0,
   3111.0,
   3209.0   ],
  "miss_num": 0,
  "mean": 3619.75
 },
 "pcv": {
  "min": 4.0,
  "max": 74.0,
  "midian": 42.0,
  "quartiles": [
   35.0,
   42.0,
   48.0   ],
  "miss_num": 8,
  "mean": 42.77333333333333
 },
 "pulse": {
  "min": 34.0,
  "max": 150.0,
  "midian": 60.0,
  "quartiles": [
   45.0,
   60.0,
   80.0    ],
  "miss_num": 2,
  "mean": 65.92424242424242
 },
 "HN": {
  "min": 514279.0,
  "max": 5299049.0,
  "midian": 530173.0,
  "quartiles": [
   528919.0,
   530107.0,
   534686.0   ],
  "miss_num": 0,
  "mean": 1229003.1029411764
 },
 "tp": {
  "min": 3.5,
  "max": 79.0,
  "midian": 7.199999999999999,
  "quartiles": [
   6.0,
   6.9,
   56.0   ],
  "miss_num": 10,
  "mean": 26.217241379310348
 },
 "atp": {
  "min": 0.9,
  "max": 8.0,
  "midian": 2.0,
  "quartiles": [
   1.4,
   2.0,
   3.9   ],
  "miss_num": 37,
  "mean": 2.7129032258064516}
}
```

```
{
  "aa": {
    "1.0": 52,
    "2.0": 62,
    "3.0": 60,
    "?": 194
  },
  "surg": {
    "1.0": 214,
    "2.0": 152,
    "?": 2
  },
  "pp": {
    "1.0": 151,
    "2.0": 6,
    "3.0": 116,
    "4.0": 12,
    "?": 83
  },
  "ref": {
    "1.0": 68,
    "2.0": 14,
    "3.0": 61,
    "4.0": 97,
    "?": 128
  },
  "nrPH": {
    "4.5": 3,
    "5.5": 4,
    "2.0": 10,
    "3.0": 3,
    "4.0": 3,
    "5.0": 7,
    "6.0": 4,
    "7.0": 9,
    "8.0": 1,
    "8.5": 1,
    "6.2": 1,
    "1.0": 2,
    "5.7": 1,
    "?": 299,
    "1.5": 2,
    "5.8": 1,
    "5.3": 1,
    "4.3": 1,
    "5.9": 1,
    "3.5": 1,
    "4.4": 1,
    "5.4": 1,
    "7.2": 2,
    "7.5": 3,
    "6.5": 6
  },
  "ad": {
    "1.0": 101,
    "2.0": 75,
    "3.0": 85,
    "4.0": 42,
    "?": 65
  },
  "crt": {
    "1.0": 232,
    "2.0": 96,
    "3.0": 2,
    "?": 38
  },
  "mm": {
    "1.0": 98,
    "2.0": 38,
    "3.0": 81,
    "4.0": 50,
    "5.0": 28,
    "6.0": 25,
    "?": 48
  },
  "nt": {
    "1.0": 89,
    "2.0": 121,
    "3.0": 27,
    "?": 131
  },
  "age": {
    "1.0": 340,
    "9.0": 28
  },
  "cpd": {
    "1.0": 124,
    "2.0": 244
  },
  "abd": {
    "1.0": 31,
    "2.0": 24,
    "3.0": 19,
    "4.0": 55,
    "5.0": 96,
    "?": 143
  },
  "nr": {
    "1.0": 141,
    "2.0": 45,
    "3.0": 49,
    "?": 133
  },
  "outcome": {
    "1.0": 225,
    "2.0": 89,
    "3.0": 52,
    "?": 2
  },
  "peris": {
    "1.0": 49,
    "2.0": 22,
    "3.0": 154,
    "4.0": 91,
    "?": 52
  },
  "s1": {
    "1.0": 232,
    "2.0": 136
  },
  "toe": {
    "1.0": 95,
    "2.0": 39,
    "3.0": 135,
    "4.0": 34,
    "?": 65
  }
}
```

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]

此部分用到的脚本为[cleaning.py]

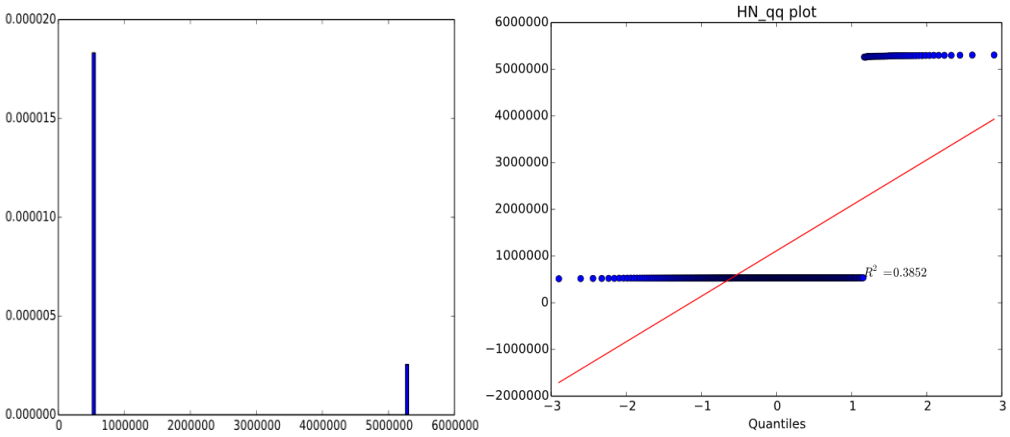
(2)．绘制直方图，qq 图，盒图

+ 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

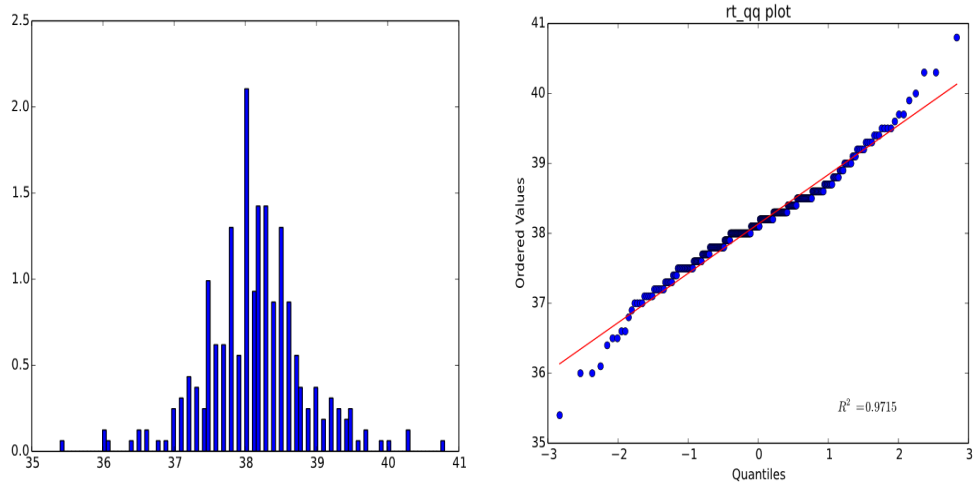
代码详见[show.py]

(3)．结果展示

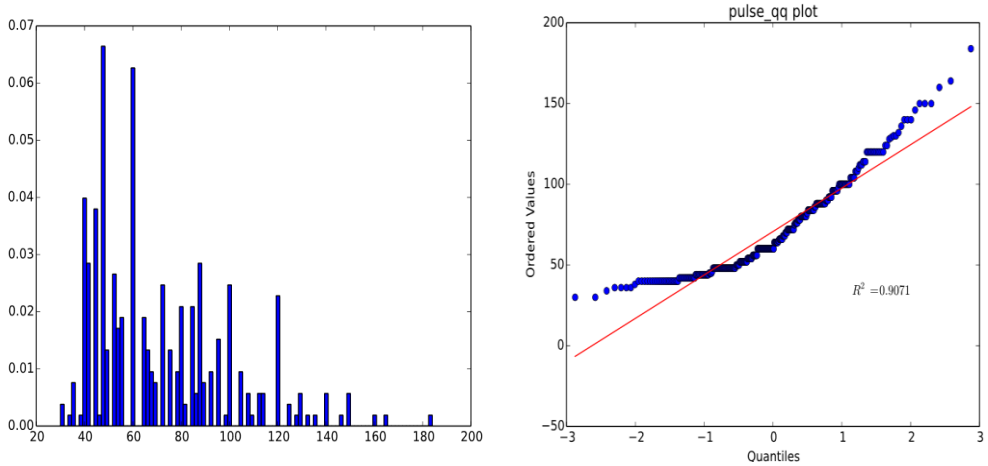
结果图分别以 svg 和 jpg 两种格式保存



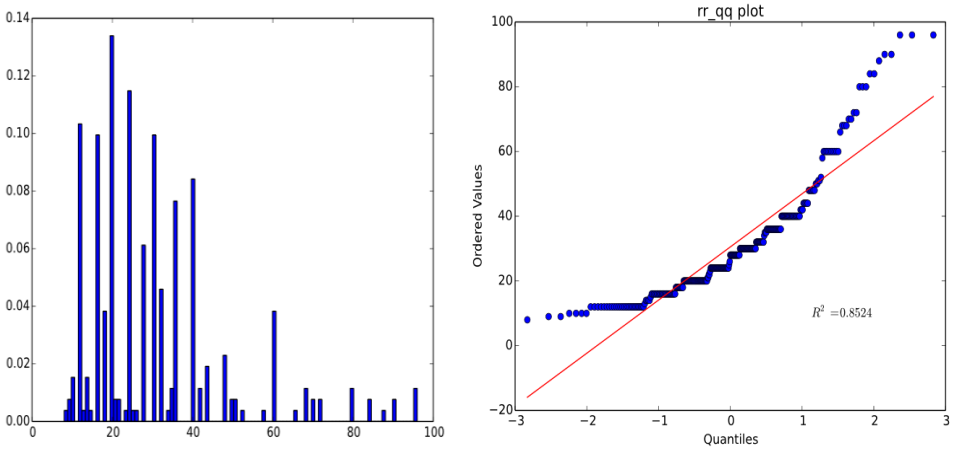
医院编号（HN）的直方图与 qq 图



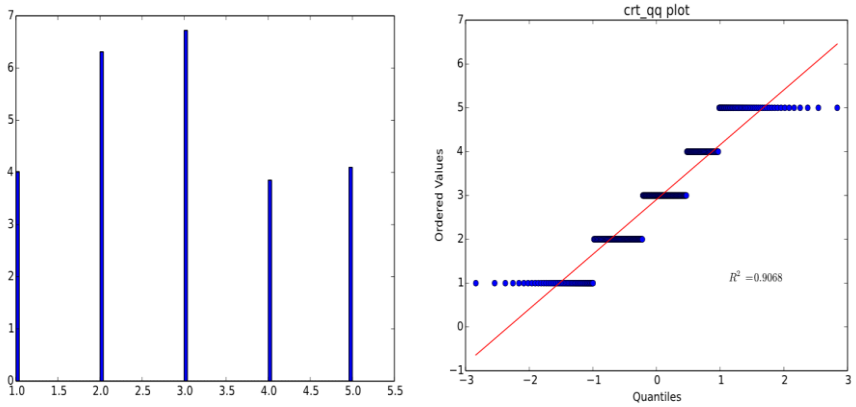
直肠温度（rt）的直方图与 qq 图



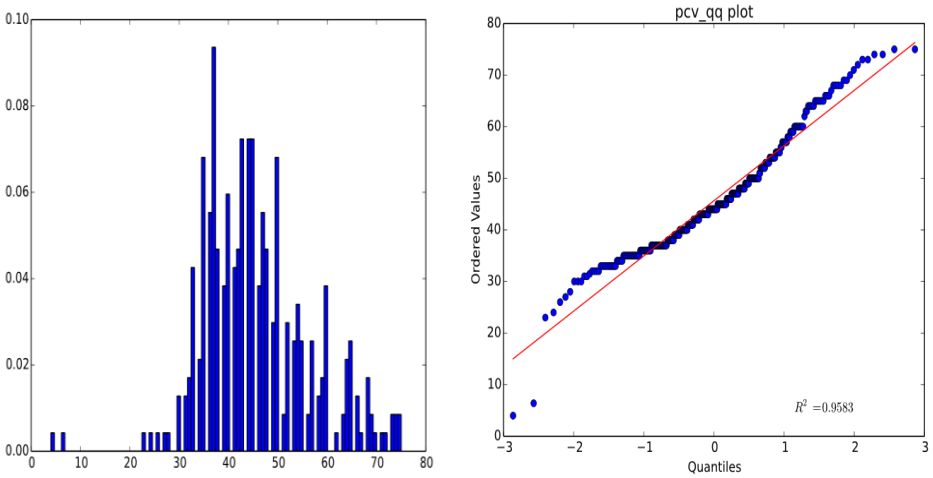
脉冲（pluse）的直方图与 qq 图



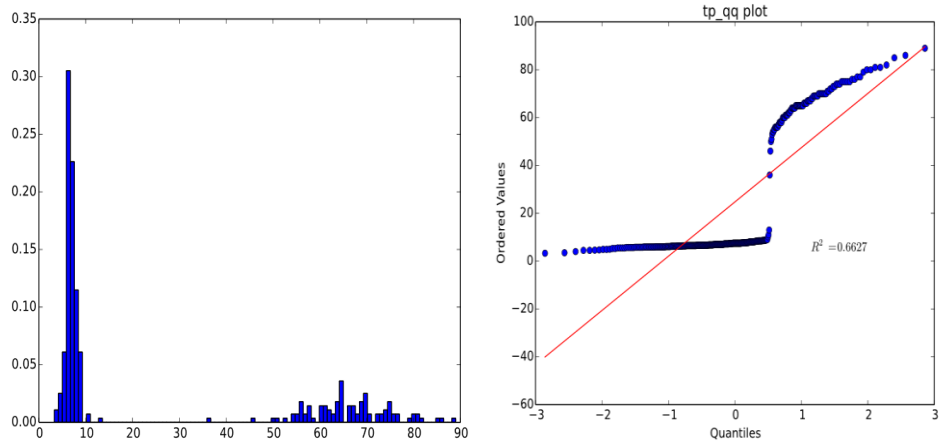
呼吸频率(rr)的直方图与 qq 图



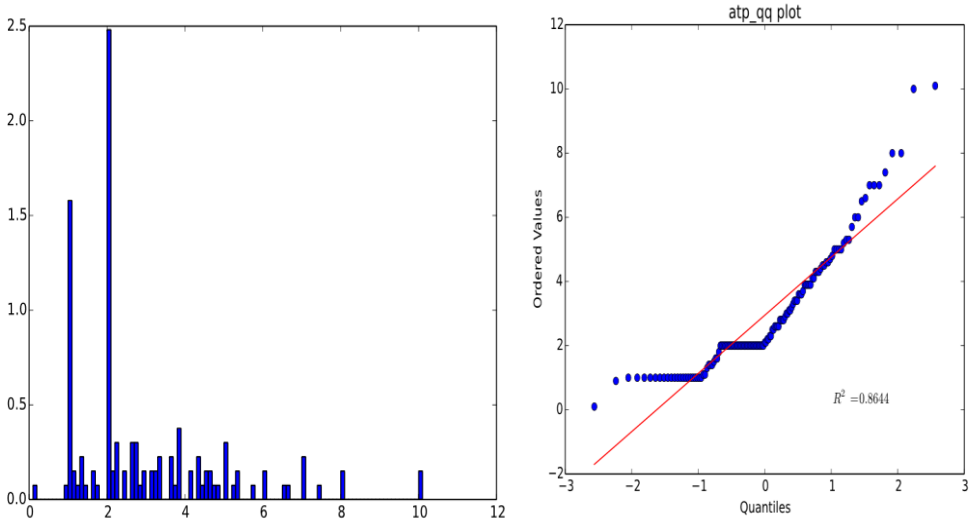
疼痛(pain)的直方图与 qq 图



填充细胞体积(pcv)的直方图与 qq 图

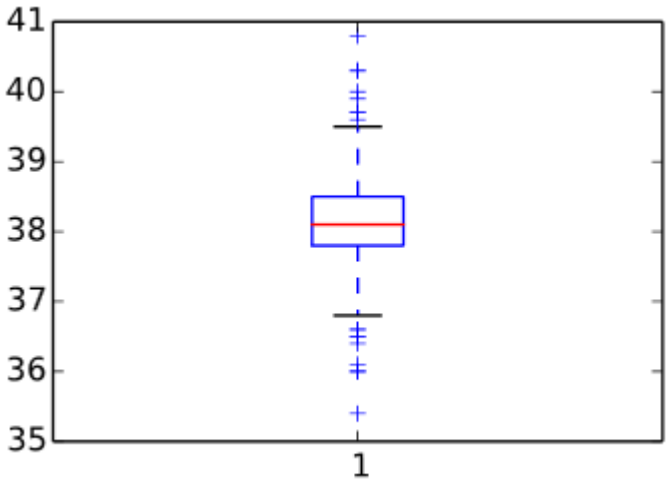


总蛋白质(tp)的直方图与 qq 图

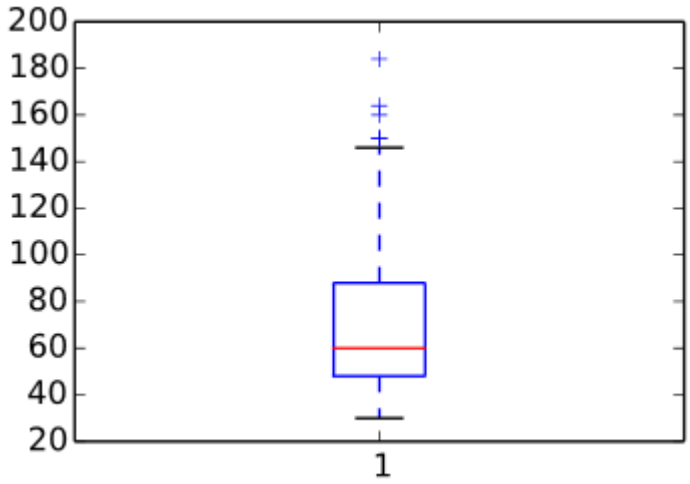


腹部总蛋白质(atp)的直方图与 qq 图

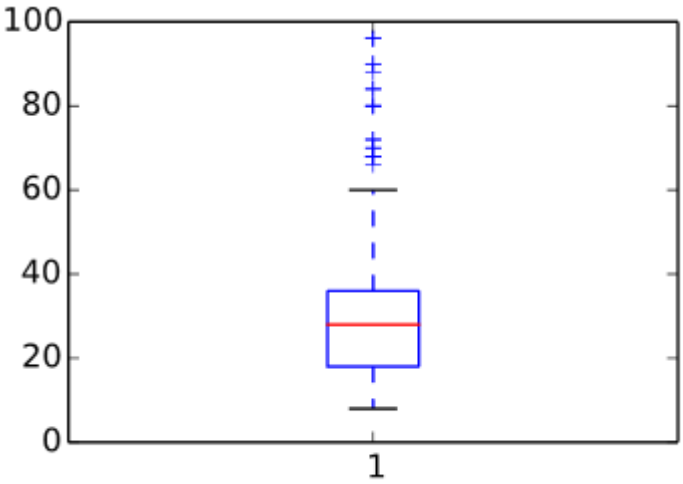
绘制盒图，识别离群点



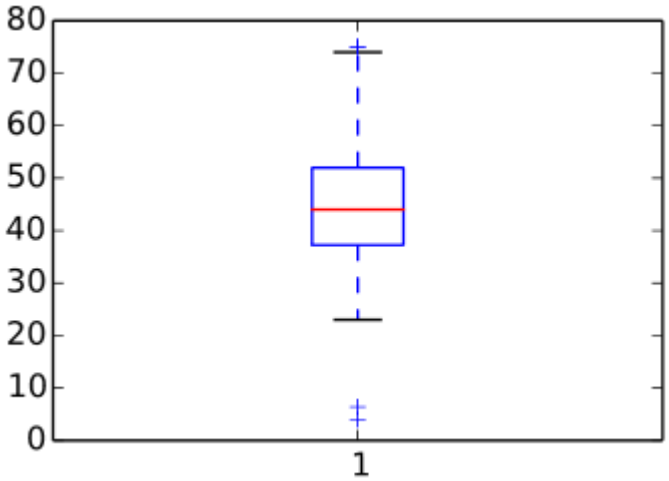
直肠温度（rt）的盒图



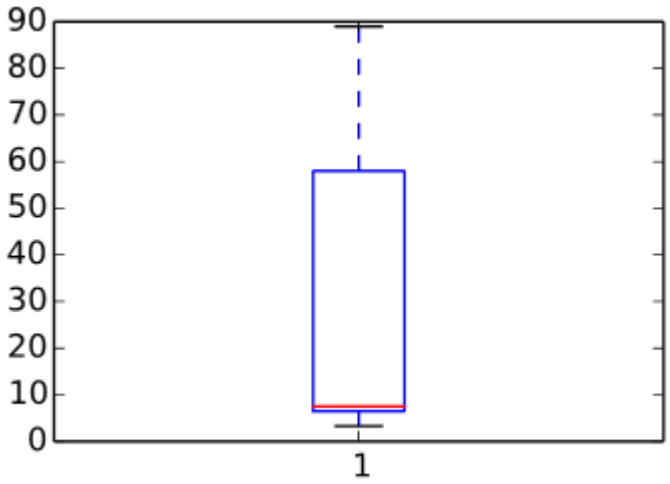
脉冲（pluse）的盒图



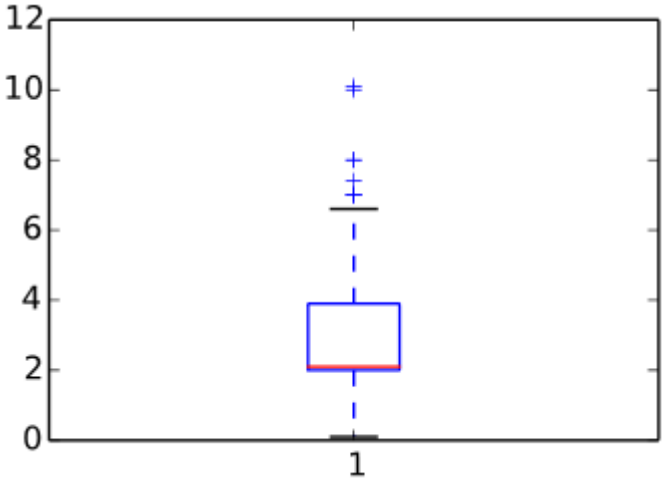
呼吸频率(rr)的盒图



填充细胞体积(pcv)的盒图



总蛋白质(tp)的盒图



腹部总蛋白质(atp)的盒图

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
```

剔除之前的数据

```
2 1 530101 38.50 66 28 3 3 ? 2 5 4 4 ? ? ? 3 5 45.00 8.40 ? ? 2 2 11300 00000 00000 2
1 1 534817 39.2 88 20 ? ? 4 1 3 4 2 ? ? ? 4 2 50 85 2 2 3 2 02208 00000 00000 2
2 1 530334 38.30 40 24 1 1 3 1 3 3 1 ? ? ? 1 1 33.00 6.70 ? ? 1 2 00000 00000 00000 1
1 9 5290409 39.10 164 84 4 1 6 2 2 4 4 1 2 5.00 3 ? 48.00 7.20 3 5.30 2 1 02208 00000 00000 1
2 1 530255 37.30 104 35 ? ? 6 2 ? ? ? ? ? ? ? ? 74.00 7.40 ? ? 2 2 04300 00000 00000 2
2 1 528355 ? ? ? 2 1 3 1 2 3 2 2 1 ? 3 3 ? ? ? ? 1 2 00000 00000 00000 2
1 1 526802 37.90 48 16 1 1 1 1 3 3 3 1 1 ? 3 5 37.00 7.00 ? ? 1 1 03124 00000 00000 2
1 1 529607 ? 60 ? 3 ? ? 1 ? 4 2 2 1 ? 3 4 44.00 8.30 ? ? 2 1 02208 00000 00000 2
2 1 530051 ? 80 36 3 4 3 1 4 4 4 2 1 ? 3 5 38.00 6.20 ? ? 3 1 03205 00000 00000 2
2 9 5299629 38.30 90 ? 1 ? 1 1 5 3 1 2 1 ? 3 ? 40.00 6.20 1 2.20 1 2 00000 00000 00000 1
1 1 528548 38.10 66 12 3 3 5 1 3 3 1 2 1 3.00 2 5 44.00 6.00 2 3.60 1 1 02124 00000 00000 1
2 1 527927 39.10 72 52 2 ? 2 1 2 1 2 1 1 ? 4 4 50.00 7.80 ? ? 1 1 02111 00000 00000 2
1 1 528031 37.20 42 12 2 1 1 1 3 3 3 3 1 ? 4 5 ? 7.00 ? ? 1 2 04124 00000 00000 2
2 9 5291329 38.00 92 28 1 1 2 1 1 3 2 3 ? 7.20 1 1 37.00 6.10 1 ? 2 2 00000 00000 00000 1
1 1 534917 38.2 76 28 3 1 1 1 3 4 1 2 2 ? 4 4 46 81 1 2 1 1 02112 00000 00000 2
1 1 530233 37.60 96 48 3 1 4 1 5 3 3 2 3 4.50 4 ? 45.00 6.80 ? ? 2 1 03207 00000 00000 2
1 9 5301219 ? 128 36 3 3 4 2 4 4 3 3 ? ? 4 5 53.00 7.80 3 4.70 2 2 01400 00000 00000 1
2 1 526639 37.50 48 24 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? 1 2 00000 00000 00000 2
1 1 5290481 37.60 64 21 1 1 2 1 2 3 1 1 1 ? 2 5 40.00 7.00 1 ? 1 1 04205 00000 00000 1
2 1 532110 39.4 110 35 4 3 6 ? ? 3 3 ? ? ? ? ? 55 8.7 ? ? 1 2 00000 00000 00000 2
1 1 530157 39.90 72 60 1 1 5 2 5 4 4 3 1 ? 4 4 46.00 6.10 2 ? 1 1 02111 00000 00000 2
```

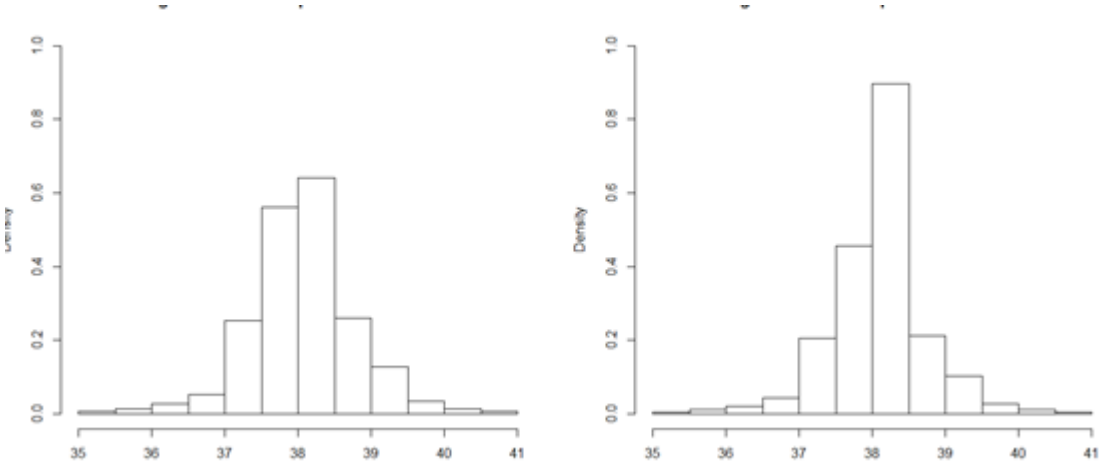
剔除之后的数据

```
1 1 528548 38.1 66 12 3 3 5 1 3 3 1 2 1 3 2 5 44 6 2 3.6 1 1 2124 0 0 1
2 1 529461 40.3 114 36 3 3 1 2 2 3 3 2 1 7 1 5 57 8.1 3 4.5 3 1 7400 0 0 1
1 1 529667 39 64 36 3 1 4 2 3 3 2 1 2 7 4 5 44 7.5 3 5 1 1 2113 0 0 1
2 1 529461 40.3 114 36 3 3 1 2 2 3 3 2 1 7 1 5 57 8.1 3 4.5 2 1 3205 0 0 1
1 1 527563 37.8 52 24 1 3 3 1 4 4 1 2 3 5.7 2 5 48 6.6 1 3.7 2 1 5400 0 0 2
1 1 5299603 38.3 60 16 3 1 1 1 2 1 1 2 2 3 1 4 30 6 1 3 1 1 31110 0 0 2
1 1 528999 37.9 120 60 3 3 3 1 5 4 4 2 2 7.5 4 5 52 6.6 3 1.8 2 1 3205 0 0 1
```

缺失状况非常严重，已经无法做进一步的数据挖掘。

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 rt 的直方图对比

左图是变换前，右图是变换之后的，从图中可以看出，数据更加集中于均值的位置，这可能是由于是利用中位数插值的结果。

3） 通过属性的相关关系来填补缺失值

```
symnum(cor(mydata[,1:13],use='complete.obs'))
```

得到属性之间的相关性如下图：

```
> symnum(cor(Adata[,1:23],use='complete.obs'))
                          s A H rc. pl rs. t.. pr. m c pn pr abdmnl. nsgstrc.t nsgstrc.r n.. r.. ab p.. tt. abdmnc. a.. o
surgery                   1
Age                       ? 1
Hospital.Number             ? 1
rectal.temperature        * ?   1
pulse                     , ? . .   1
respiratory.rate            ? .     ,  1
temperature.of.extremities  ?   .   .      1
peripheral.pulse          . ? ,     .          1
mucous.membranes          , ? . .   .              1
capillary.refill.time     , ? . *   .      .       . 1
pain                      . ? . ,       .   .       . . 1
peristalsis                 ? +         .   .   ,   .   ,  1
abdominal.distension      . ? . .   *   *   .   .   . .     .  1
nasogastric.tube            ?                 ,   . .                 1
nasogastric.reflux        , ?   ,   .       ,         . .     .                   1
nasogastric.reflux.PH     . ? , .   ,   +             ,     ,   +                       1
rectal.examination...feces  . ? . .     .             ,   ,   .           ,       .       .   1
abdomen                     ? 1     .   .       ,     .   .   +   .                       ,   .   1
packed.cell.volume        , ? + .   ,   .       ,     .       ,   ,                   .       ,       + 1
total.protein             + ? . *   .   .             .   * .     .                   .       ,       .   ,   1
abdominocentesis.appearance . ? . ,   ,   ,   .         ,     .   +       .       .       ,   .   .   ,   ,   1
abdomcentesis.total.protein . ?   ,                       + ,             .                                   ,           1
outcome                   , ? . .   ,   .       ,     .   .     .   .       .                   ,       .   ,   ,   .           1
attr(,"legend")
[1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1 \t    ## NA: '?'
```

使用以下代码得到其线性模型

```
lm(formula=rt~ort, data=mydata)
```

```
> lm(formula=rectal.temperature~surgery, data=Adata)

Call:
lm(formula = rectal.temperature ~ surgery, data = Adata)

Coefficients:
(Intercept)      surgery
  38.122193     0.009133
```

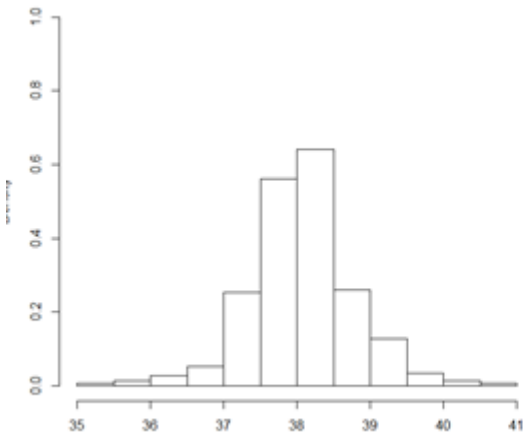
```
                                                  mber   rectal.temperature     pulse
Number   rectal.temperature     pulse            4279   Min.   :35.40      Min.   : 30.00
514279   Min.   :35.40      Min.   : 30.00      8915   1st Qu.:37.80      1st Qu.: 48.00
528915   1st Qu.:37.80      1st Qu.: 48.00      0299   Median :38.13      Median : 60.00
530299   Median :38.10      Median : 60.00      2334   Mean   :38.13      Mean   : 70.76
.12334   Mean   :38.13      Mean   : 70.76      4728   3rd Qu.:38.40      3rd Qu.: 88.00
534728   3rd Qu.:38.50      3rd Qu.: 88.00      5629   Max.   :40.80      Max.   :184.00
805629   Max.   :40.80      Max.   :184.00                                 NA's   :26
         NA's   :69         NA's   :26          stalsis    abdominal.distension nasogastri
istalsis    abdominal.distension nasogastri     :1.000   Min.   :1.000        Min.   :1.
   :1.000   Min.   :1.000        Min.   :1
```

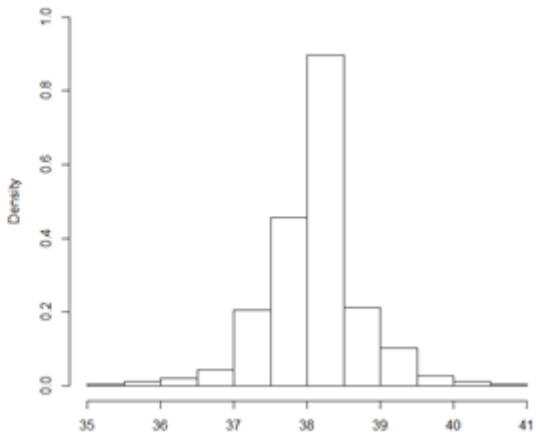
处理之前和之后的对比

使用线性模型来填充数据：

```
preprocess3 = mydata[-manyNAs(mydata),]
fillPO4 <- function(rt){
    if(is.na(rt))
        return(NA)
    else return (42.897 + 1.293 * oP)
}
```

```
preprocess3[is.na(preprocess3$rt),'rt'] =
sapply(preprocess3[is.na(preprocess3$rt),'ort'],fillrt)
```





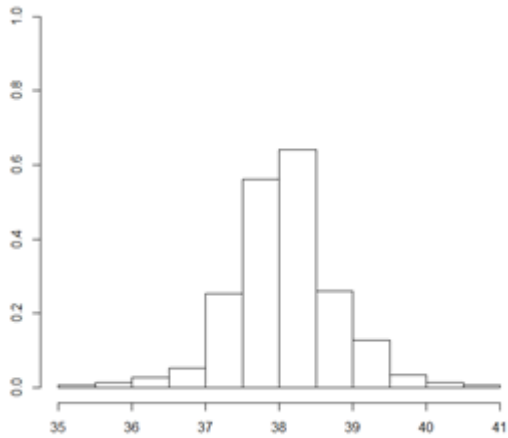
针对于 rt 的分布基本上和中位数插值是相同的。

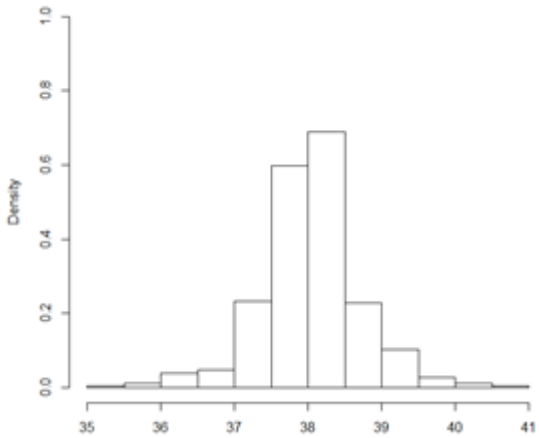
4） 使用数据对象之间的相似型填补缺失值

```
preprocess4 = knnImputation(mydata,k=3)
write.table(preprocess4,'C:\Users\xzf0724\Python/knnImputationData.
txt',col.names = F,row.names = F, quote = F)
```


```
{
 "rt": {
  "min": 36.0,
  "max": 39.7,
  "midian": 38.0,
  "quartiles": [
   37.7,
   38.0,
   38.3   ],
  "miss_num": 9,
  "mean": 37.99830508474576
 },
 "ort": {
  "min": 1.0,
  "max": 5.0,
  "midian": 3.0,
  "quartiles": [
   2.0,
   3.0,
   4.0   ],
  "miss_num": 8,
  "mean": 2.7333333333333334
 },
 "tol2": {
  "min": 0.0,
  "max": 3205.0,
  "midian": 0.0,
  "quartiles": [
   0.0,
   0.0,
   0.0   ],
  "miss_num": 0,
  "mean": 126.73529411764706
 },
 "tol1": {
  "min": 0.0,
  "max": 31110.0,
  "midian": 3111.0,
  "quartiles": [
   2111.0,
   3111.0,
   3209.0   ],
  "miss_num": 0,
  "mean": 3619.75
 },
 "HN": {
  "min": 514279.0,
  "max": 5299049.0,
  "midian": 530173.0,
  "quartiles": [
   528919.0,
   530107.0,
   534686.0   ],
  "miss_num": 0,
  "mean": 1229003.1029411764
 },
 "tp": {
  "min": 3.5,
  "max": 79.0,
  "midian": 7.199999999999999,
  "quartiles": [
   6.0,
   6.9,
   56.0   ],
  "miss_num": 10,
  "mean": 26.217241379310348
 },
```


这是操作之后的结果可以看出所有案例的缺失值都已经被处理了





也是针对 rt 的分布，这个处理的结果与 2)，3）方法处理出来的相比更接近与原先数据的分布，相比较而言更加的准确。