# 作业一 马的疝病分析

**姓名：魏嘉毅**

**学号：2620160022**

### **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**

****下载数据:**** [地址](http://archive.ics.uci.edu/ml/datasets/Horse+Colic)

共368个样本，27个特征。关于特征的详细说明见下载链接。

### **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**

**语言及环境依赖**

语言： python

依赖的包：pandas， mtplotlib, scipy, numpy

pandas: 数据摘要处理时用到

matplotlib, scipy, numpy：数据可视化时用于生成图

1. **实验方法**

**5.1 数据载入**

**使用read\_csv方法，按空格读入数据，将表格中全部缺失数据替换为nan。**

**5.2 数据摘要**

**将特征值中的数值属性标出，分别对标称属性使用value\_count（）方法和对数值属性使用describe（）方法来统计。**

**标称属性**

**1.0 180**

**2.0 119**

**Name: surgery, dtype: int64**

**1 276**

**9 24**

**Name: Age, dtype: int64**

**529796 2**

**529424 2**

**5279822 2**

**527916 2**

**530526 2**

**528996 2**

**528729 2**

**528151 2**

**529461 2**

**528931 2**

**528890 2**

**528469 2**

**530693 2**

**532349 2**

**528904 2**

**527544 2**

**533697 1**

**533696 1**

**533736 1**

**534719 1**

**530101 1**

**529399 1**

**533692 1**

**528570 1**

**527957 1**

**534157 1**

**521399 1**

**529045 1**

**530612 1**

**528047 1**

**..**

**533887 1**

**535031 1**

**530301 1**

**5294369 1**

**530297 1**

**529272 1**

**535415 1**

**530294 1**

**534899 1**

**529777 1**

**535407 1**

**522979 1**

**528743 1**

**534885 1**

**534857 1**

**530276 1**

**535392 1**

**535338 1**

**527709 1**

**527706 1**

**533847 1**

**528214 1**

**535381 1**

**533750 1**

**527698 1**

**530255 1**

**530254 1**

**533836 1**

**530251 1**

**535043 1**

**Name: Hospital\_Number, dtype: int64**

**3.0 109**

**1.0 78**

**2.0 30**

**4.0 27**

**Name: temperature\_of\_extremities, dtype: int64**

**1.0 115**

**3.0 103**

**4.0 8**

**2.0 5**

**Name: peripheral\_pulse, dtype: int64**

**1.0 79**

**3.0 58**

**4.0 41**

**2.0 30**

**5.0 25**

**6.0 20**

**Name: mucous\_membranes, dtype: int64**

**1.0 188**

**2.0 78**

**3.0 2**

**Name: capillary\_refill\_time, dtype: int64**

**3.0 67**

**2.0 59**

**5.0 42**

**4.0 39**

**1.0 38**

**Name: pain\_level, dtype: int64**

**3.0 128**

**4.0 73**

**1.0 39**

**2.0 16**

**Name: peristalsis, dtype: int64**

**1.0 76**

**3.0 65**

**2.0 65**

**4.0 38**

**Name: abdominal\_distension, dtype: int64**

**2.0 102**

**1.0 71**

**3.0 23**

**Name: nasogastric\_tube, dtype: int64**

**1.0 120**

**3.0 39**

**2.0 35**

**Name: nasogastric\_reflux, dtype: int64**

**4.0 79**

**1.0 57**

**3.0 49**

**2.0 13**

**Name: feces, dtype: int64**

**5.0 79**

**4.0 43**

**1.0 28**

**2.0 19**

**3.0 13**

**Name: abdomen, dtype: int64**

**2.0 48**

**3.0 46**

**1.0 41**

**Name: abdominocentesis\_appearance, dtype: int64**

**1.0 178**

**2.0 77**

**3.0 44**

**Name: outcome, dtype: int64**

**1 191**

**2 109**

**Name: surgical\_lesion, dtype: int64**

**0 56**

**3111 33**

**3205 29**

**2208 20**

**2205 13**

**4205 11**

**2209 11**

**2124 9**

**1400 8**

**31110 7**

**7111 7**

**2113 6**

**2112 5**

**400 5**

**3209 4**

**4300 4**

**2206 4**

**5400 4**

**3112 3**

**4124 3**

**2111 3**

**2207 3**

**7209 3**

**5206 2**

**5124 2**

**3124 2**

**5111 2**

**9400 2**

**6111 2**

**2322 2**

**..**

**3025 2**

**8400 2**

**6112 2**

**11300 1**

**4122 1**

**7113 1**

**6209 1**

**3115 1**

**5000 1**

**3133 1**

**4111 1**

**3400 1**

**300 1**

**12208 1**

**9000 1**

**5205 1**

**1111 1**

**1124 1**

**8300 1**

**2305 1**

**4206 1**

**4207 1**

**21110 1**

**2300 1**

**3207 1**

**11400 1**

**7400 1**

**3113 1**

**3300 1**

**41110 1**

**Name: lesion\_1, dtype: int64**

**0 293**

**3111 3**

**6112 1**

**7111 1**

**1400 1**

**3112 1**

**Name: lesion\_2, dtype: int64**

**0 299**

**2209 1**

**Name: lesion\_3, dtype: int64**

**2 201**

**1 99**

**Name: cp\_data, dtype: int64**

**数值属性：**

**count 240.000000**

**mean 38.167917**

**std 0.732289**

**min 35.400000**

**25% 37.800000**

**50% 38.200000**

**75% 38.500000**

**max 40.800000**

**Name: rectal\_temerature, dtype: float64**

**count 276.000000**

**mean 71.913043**

**std 28.630557**

**min 30.000000**

**25% 48.000000**

**50% 64.000000**

**75% 88.000000**

**max 184.000000**

**Name: pulse, dtype: float64**

**count 242.000000**

**mean 30.417355**

**std 17.642231**

**min 8.000000**

**25% 18.500000**

**50% 24.500000**

**75% 36.000000**

**max 96.000000**

**Name: respiratory\_rate, dtype: float64**

**count 53.000000**

**mean 4.707547**

**std 1.982311**

**min 1.000000**

**25% 3.000000**

**50% 5.000000**

**75% 6.500000**

**max 7.500000**

**Name: nasogastric\_reflux\_PH, dtype: float64**

**count 271.000000**

**mean 46.295203**

**std 10.419335**

**min 23.000000**

**25% 38.000000**

**50% 45.000000**

**75% 52.000000**

**max 75.000000**

**Name: packed\_cell\_volume, dtype: float64**

**count 267.000000**

**mean 24.456929**

**std 27.475009**

**min 3.300000**

**25% 6.500000**

**50% 7.500000**

**75% 57.000000**

**max 89.000000**

**Name: total\_protein, dtype: float64**

**count 102.000000**

**mean 3.019608**

**std 1.968567**

**min 0.100000**

**25% 2.000000**

**50% 2.250000**

**75% 3.900000**

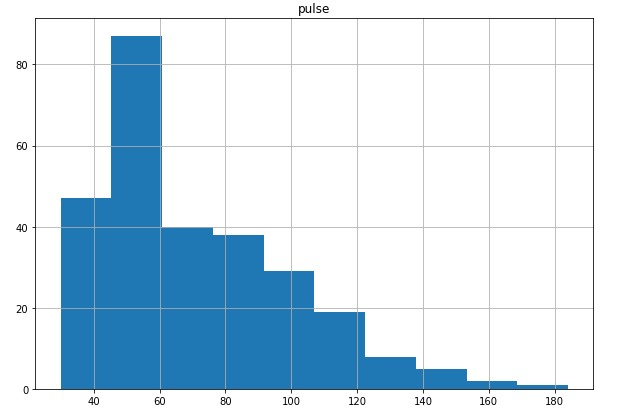
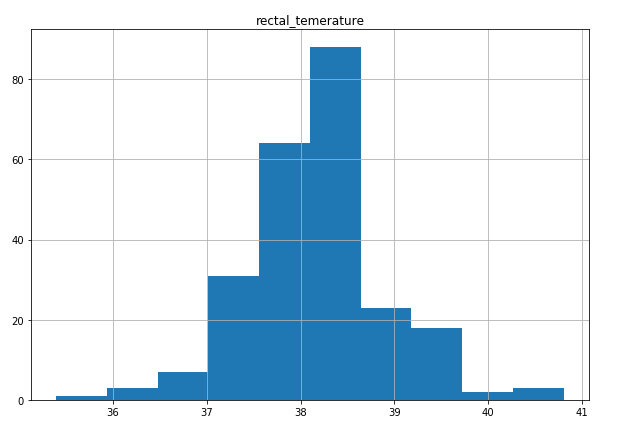
**max 10.100000**

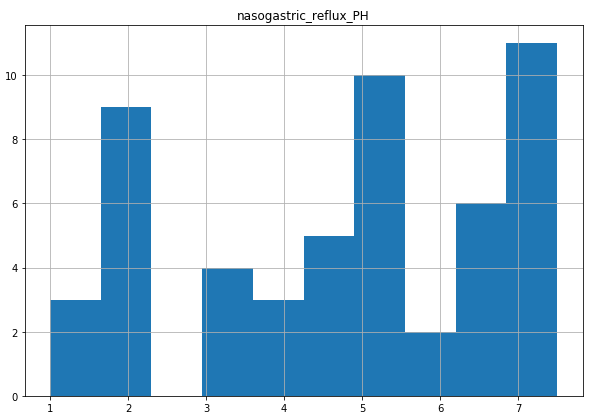
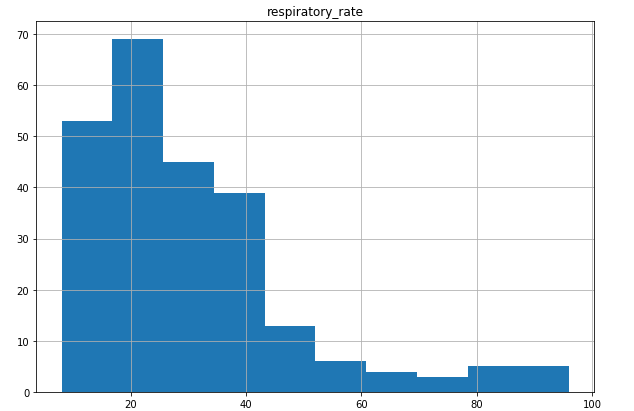
**Name: abdomcentesis\_total\_protein,**

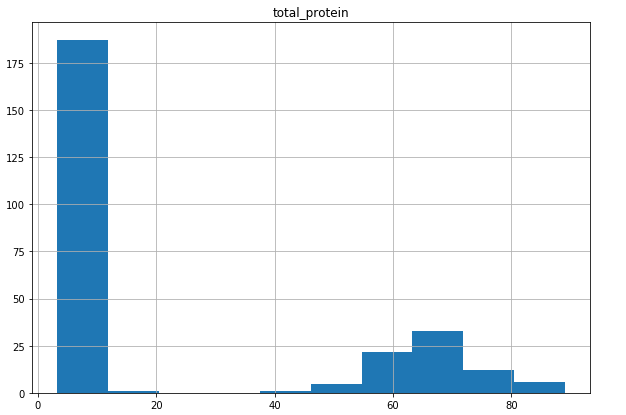
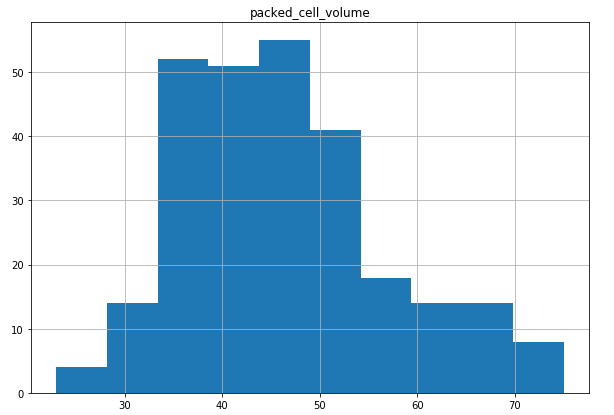
**5.3 数据可视化**

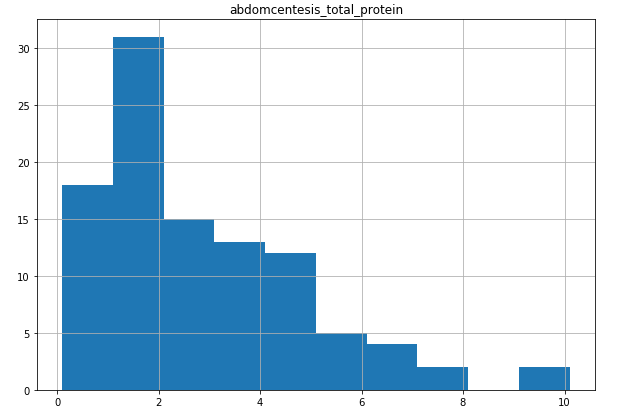
**使用python的mtplotllib库作为画图工具**

**直方图**



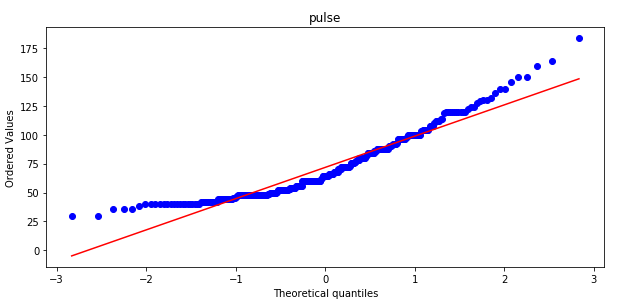
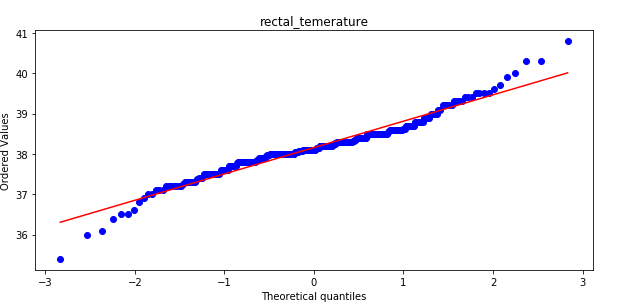


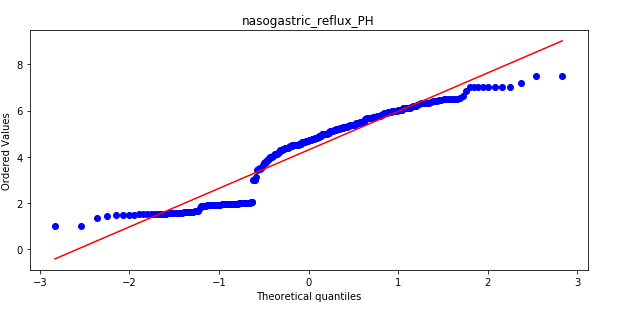
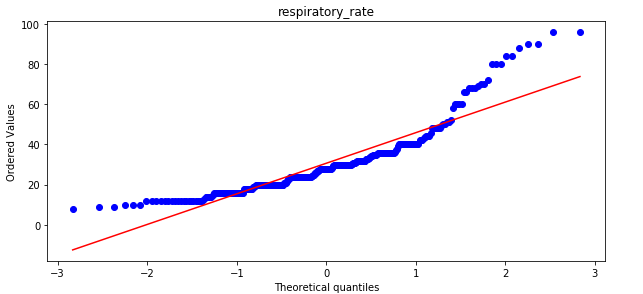


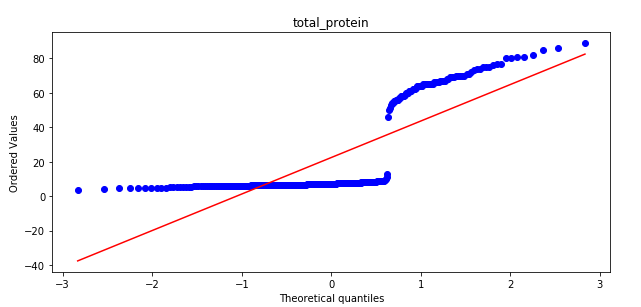
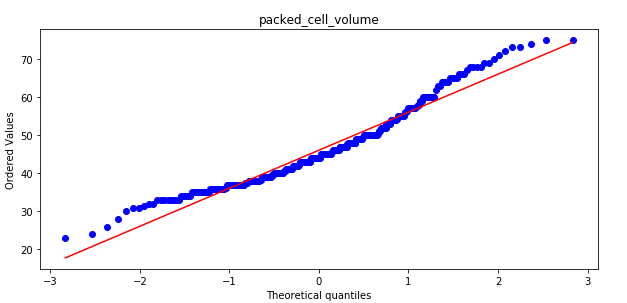


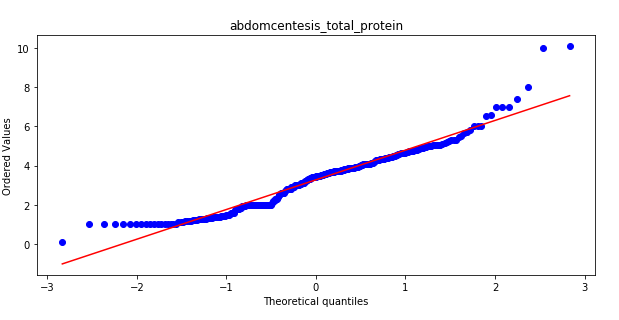
QQ图

使用Python中的 scipy.probplot（）函数来进行绘制



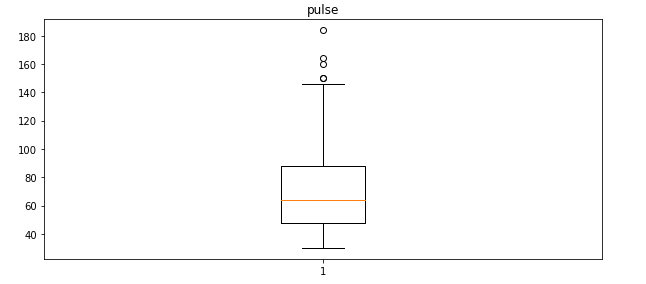
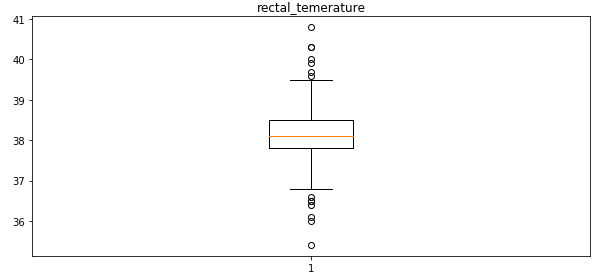


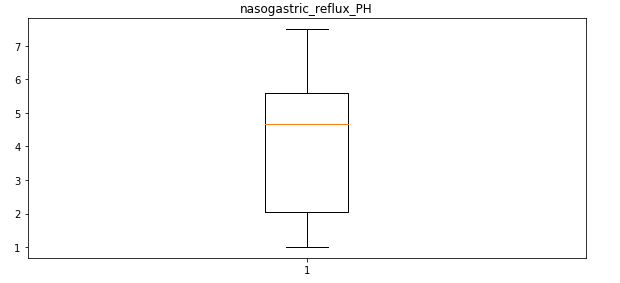
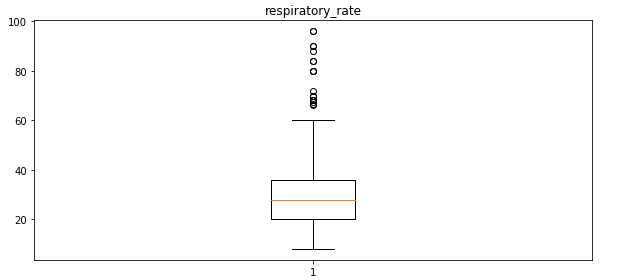


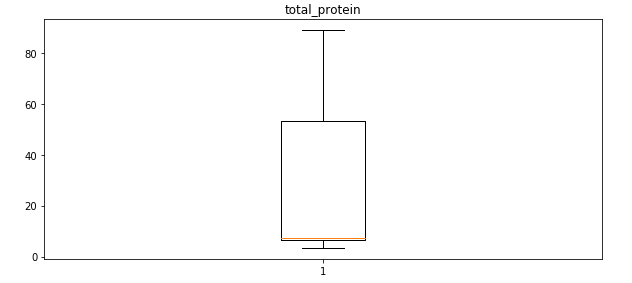
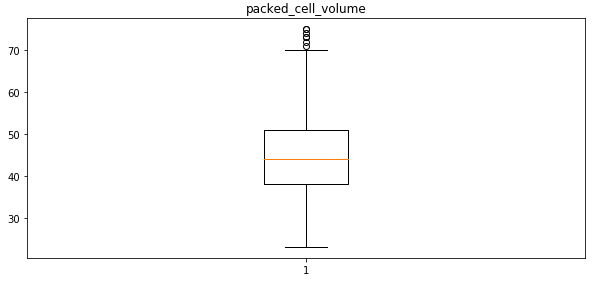


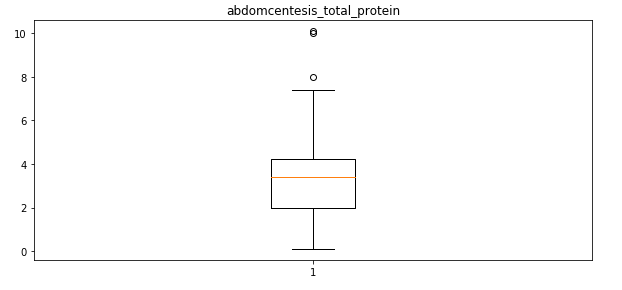
盒图

使用Python中的boxplot（）函数即可绘制









5.4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

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。