作业一:马的疝病分析

姓名:　赵佰承

学号: 2120160969

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，27 个特征。

原始数据位于 ”data/ horse-colic.data.txt” 中，其中缺失的数据用‘？’表示，数据各属性的具体描述见 “data/ dataDescription.txt” 文件。

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。

具体的程序见文件夹中”data_summary.py”文件，具体代码功能的解释见代码的注释说明。

首先将原始数据转化成 csv 格式，并保存在 “ data/ horse-colic.data.csv” 中，将”?”识别为缺失值。从数据中确定出标称属性与数值属性。其中：'surgery','Age','temperature_of_extremities', 'peripheral_pulse','mucous_membranes','nasogastric_reflux', 'rectal_examination_feces','abdomen', 'abdominocentesis_appearance', 'outcome','cp_data' 为标称属性，其中属于数值属性的有 'rectal_temperature','pulse','respiratory_rate','nasogastric_reflux_PH', 'packed_cell_volume','total_protein','abdomcentesis_total_protein'。

最终得到每个标称属性取值的频数保存在”data/numinalDataFrequency.json”中，如下图所示：

{"surgery": {"2.0": 119, "1.0": 180}, "Age": {"1.0": 276, "9.0": 24},
"temperature_of_extremities": {"3.0": 109, "1.0": 78, "4.0": 27, "2.0": 30},
"peripheral_pulse": {"3.0": 103, "1.0": 115, "4.0": 8, "2.0": 5},
"mucous_membranes": {"4.0": 41, "3.0": 58, "6.0": 20, "1.0": 79, "5.0": 25, "2.0": 30},
"nasogastric_reflux": {"2.0": 35, "1.0": 120, "3.0": 39},
"rectal_examination_feces": {"3.0": 49, "4.0": 79, "1.0": 57, "2.0": 13},
"abdomen": {"5.0": 79, "2.0": 19, "1.0": 28, "3.0": 13, "4.0": 43},
"abdominocentesis_appearance": {"2.0": 48, "3.0": 46, "1.0": 41},
"outcome": {"2.0": 77, "3.0": 44, "1.0": 178}, "surgical_lesion": {"2.0": 109, "1.0": 191},
"cp_data": {"2.0": 201, "1.0": 99}}

得到的数值属性最大、最小、均值、中位数、四分位数及缺失值的个数信息存在”data/numericDataStatistic.json”中，如下图所示：

{"rectal_temperature":
{"max": 40.8,
"min": 35.4,
"mean": 38.16791666666669,
"midian": 38.2,
"quartiles": [37.8, 38.2, 38.5],
"miss_num": 60},
"pulse":
{"max": 184.0,
"min": 30.0,
"mean": 71.91304347826087,
"midian": 64.0,
"quartiles": [48.0, 64.0, 88.0],
"miss_num": 24},
"respiratory_rate":
{"max": 96.0,
"min": 8.0,
"mean": 30.417355371900825,
"midian": 24.5,
"quartiles": [18.0, 24.0, 36.0],
"miss_num": 58},
"nasogastric_reflux_PH":
{"max": 7.5,
"min": 1.0,
"mean": 4.707547169811321,
"midian": 5.0,
"quartiles": [3.0, 5.0, 6.5],
"miss_num": 247},
"packed_cell_volume":
{"max": 75.0,
"min": 23.0,
"mean": 46.29520295202952,
"midian": 45.0,
"quartiles": [38.0, 45.0, 52.0]
"miss_num": 29},
"total_protein":
{"max": 89.0,
"min": 3.3,
"mean": 24.456928838951317,
"midian": 7.5,
"quartiles": [6.5, 7.5, 57.0],
"miss_num": 33},
"abdomcentesis_total_protein":
{"max": 10.1,
"min": 0.1,
"mean": 3.0196078431372553,
"midian": 2.25,
"quartiles": [2.0, 2.1, 3.9],
"miss_num": 198}}

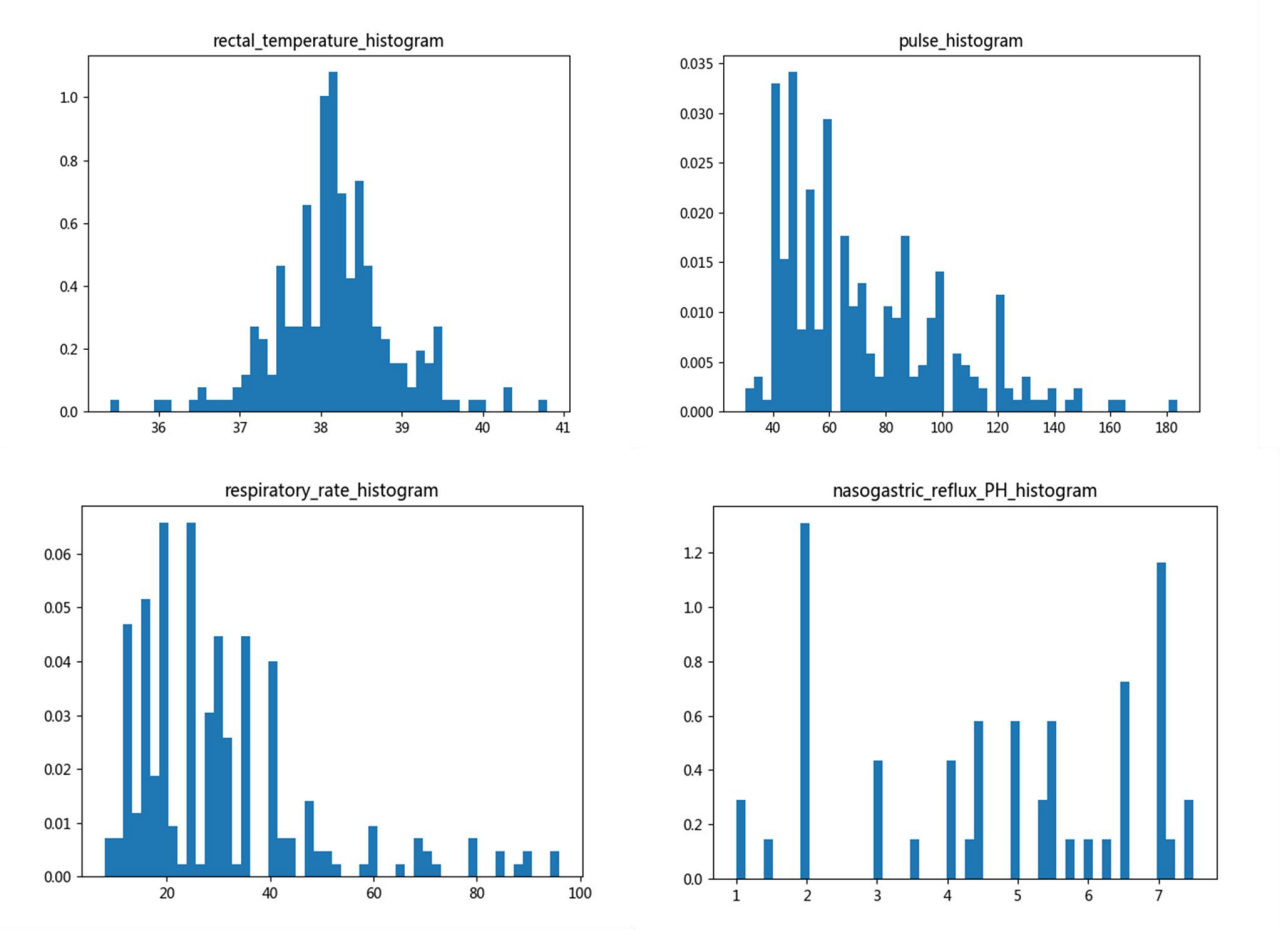
数据的可视化

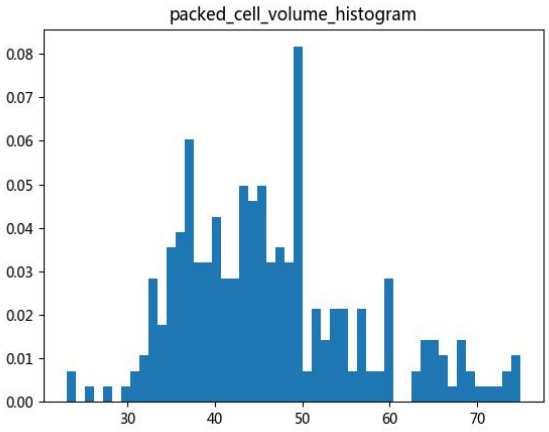
针对数值属性，

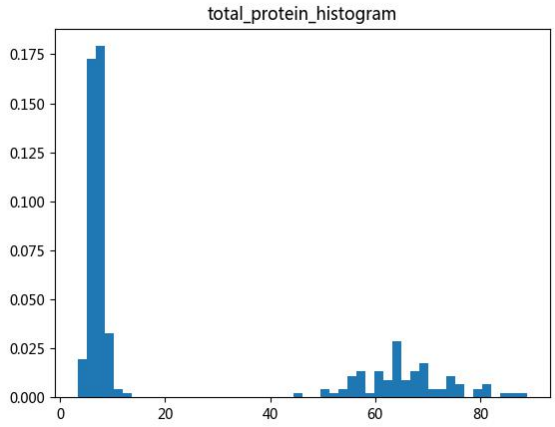
- 绘制直方图，用 qq 图检验其分布是否为正态分布。
- 绘制盒图，对离群值进行识别

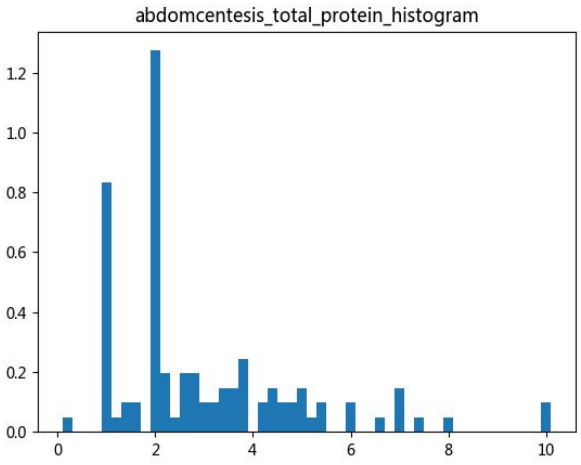
该部分的代码见文件夹中”data_visual.py”文件，该部分同构定义一个 show()函数来实现对直方图和 qq 图的绘制，其中直方图绘制主要使用 pylab.hist()方法，qq 图主要使用 scipy.stats.proplot()方法，盒图主要使用 boxplot()方法。绘制结果如下图所示：

下列为各个数值属性的直方图



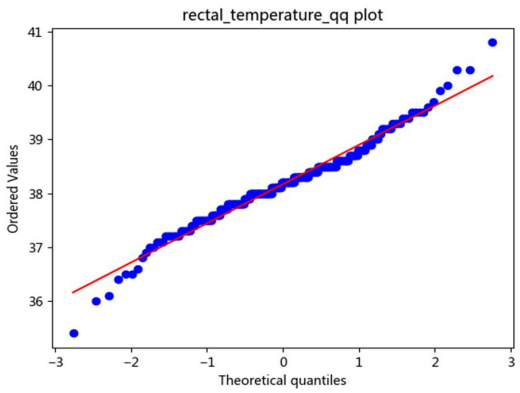


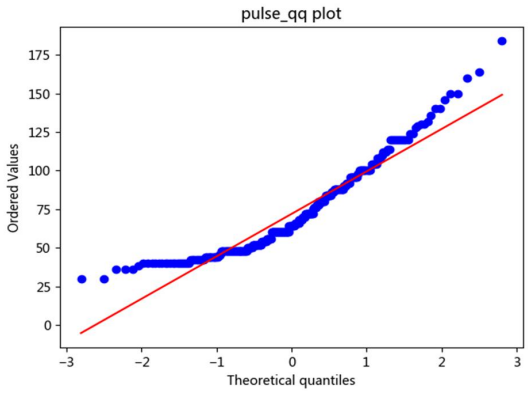


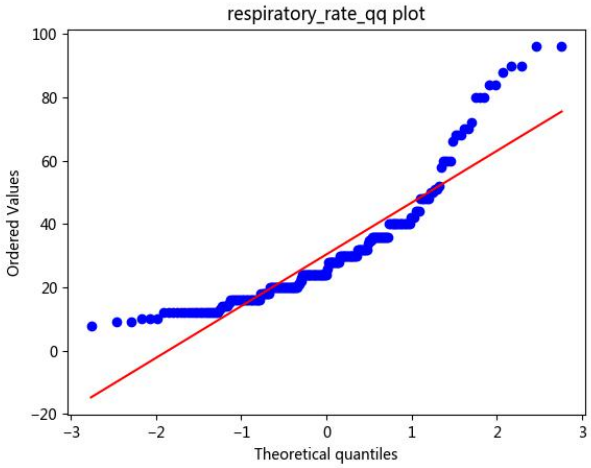


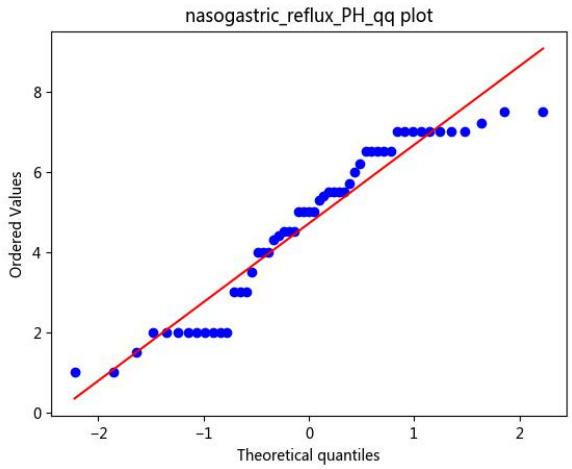
由直方图可知，'rectal_temperature'的分布较集中，方差较小。

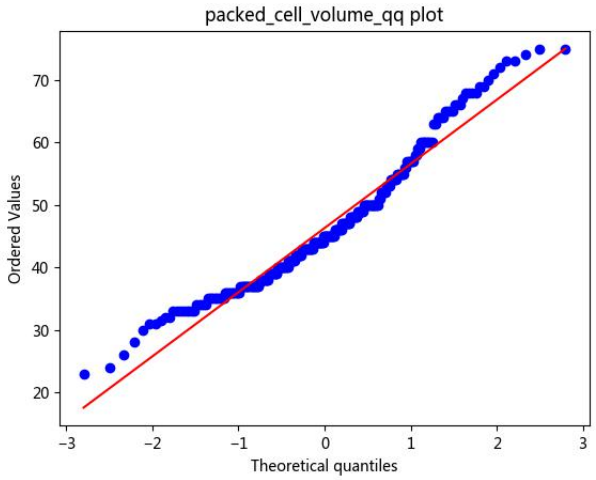
下列为各个数值属性的 qq 图

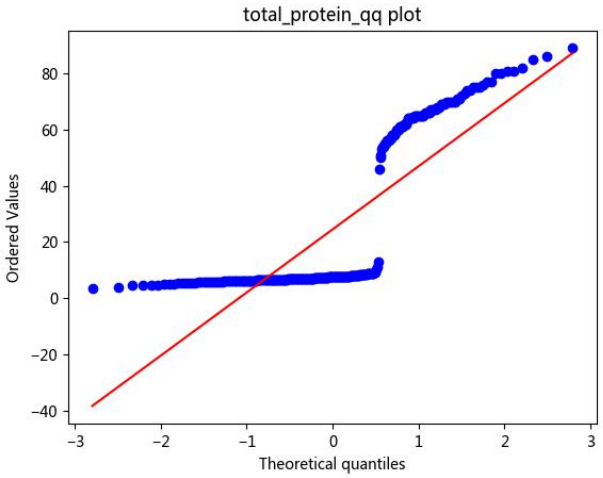


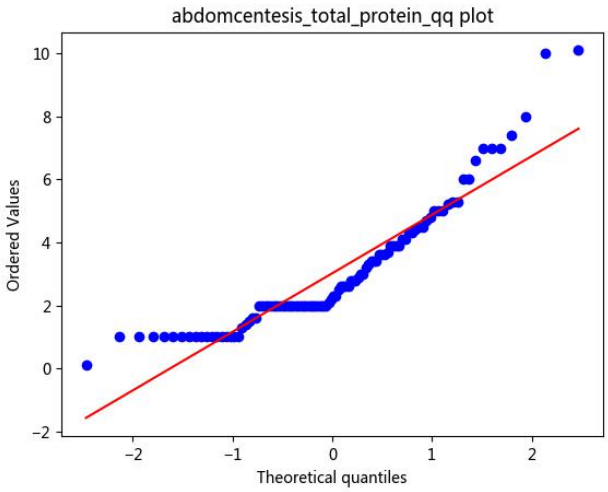



respiratory_rate_qq plot
Ordered Values
Theoretical quantiles


nasogastric_reflux_PH_qq plot
Ordered Values
Theoretical quantiles


packed_cell_volume_qq plot
Ordered Values
Theoretical quantiles

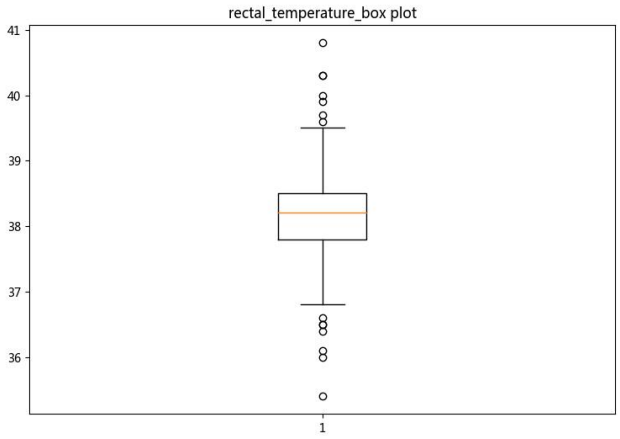

total_protein_qq plot
Ordered Values
Theoretical quantiles

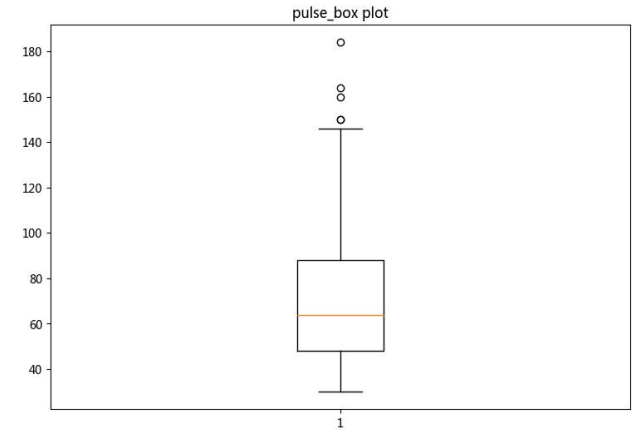

abdomcentesis_total_protein_qq plot
Ordered Values
Theoretical quantiles

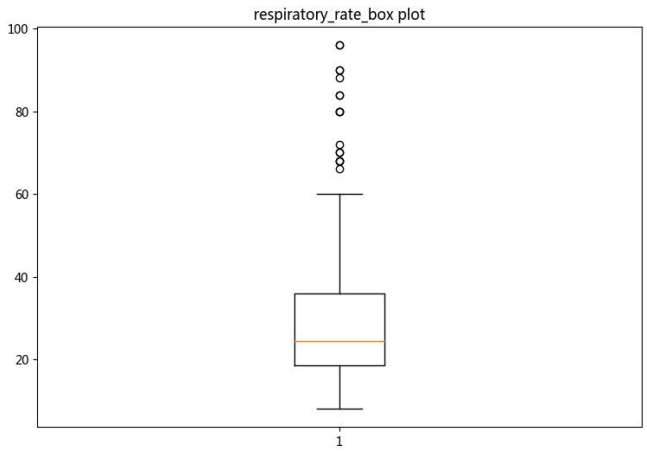
由 qq 图可知，如果数据值服从正态分布，则其 qq 就基本上是一条对角线，所以服从正态分布的数值属性有：

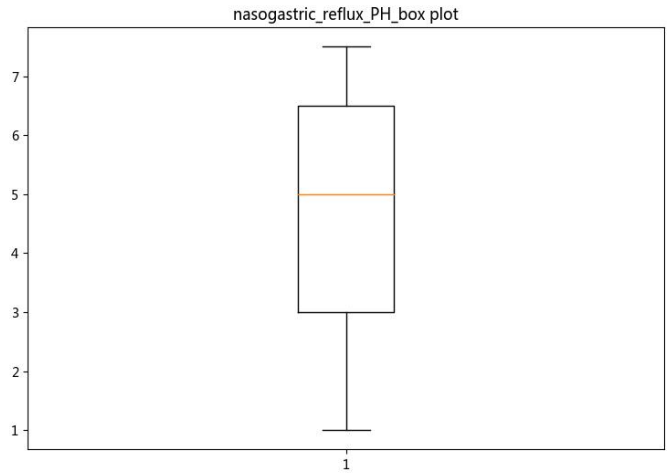
'rectal_temperature','nasogastric_reflux_PH', 'packed_cell_volume'

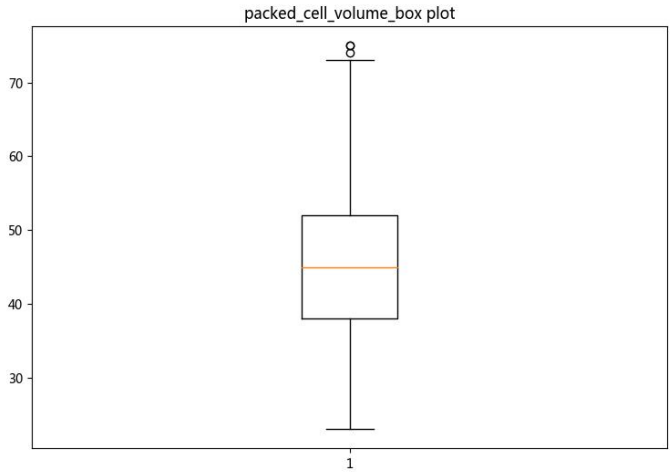
下列为各数值属性的盒图，对离群点进行识别

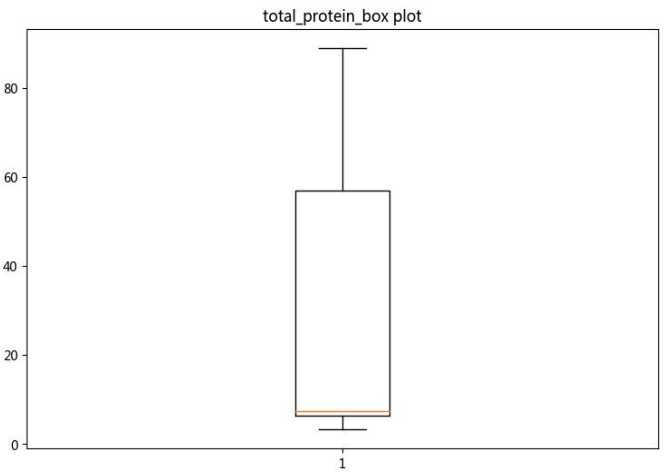


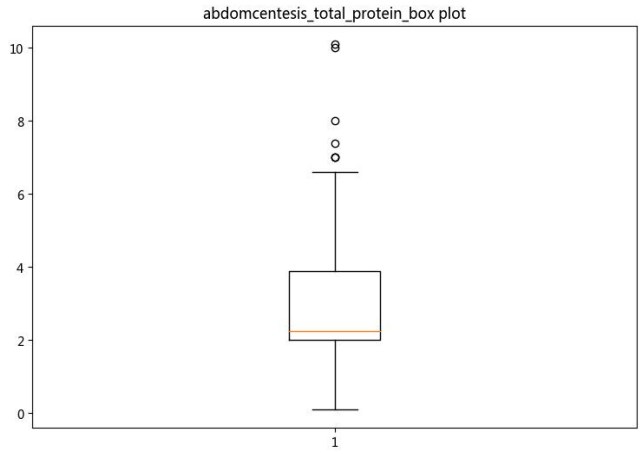












3.2 数据缺失的处理

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:

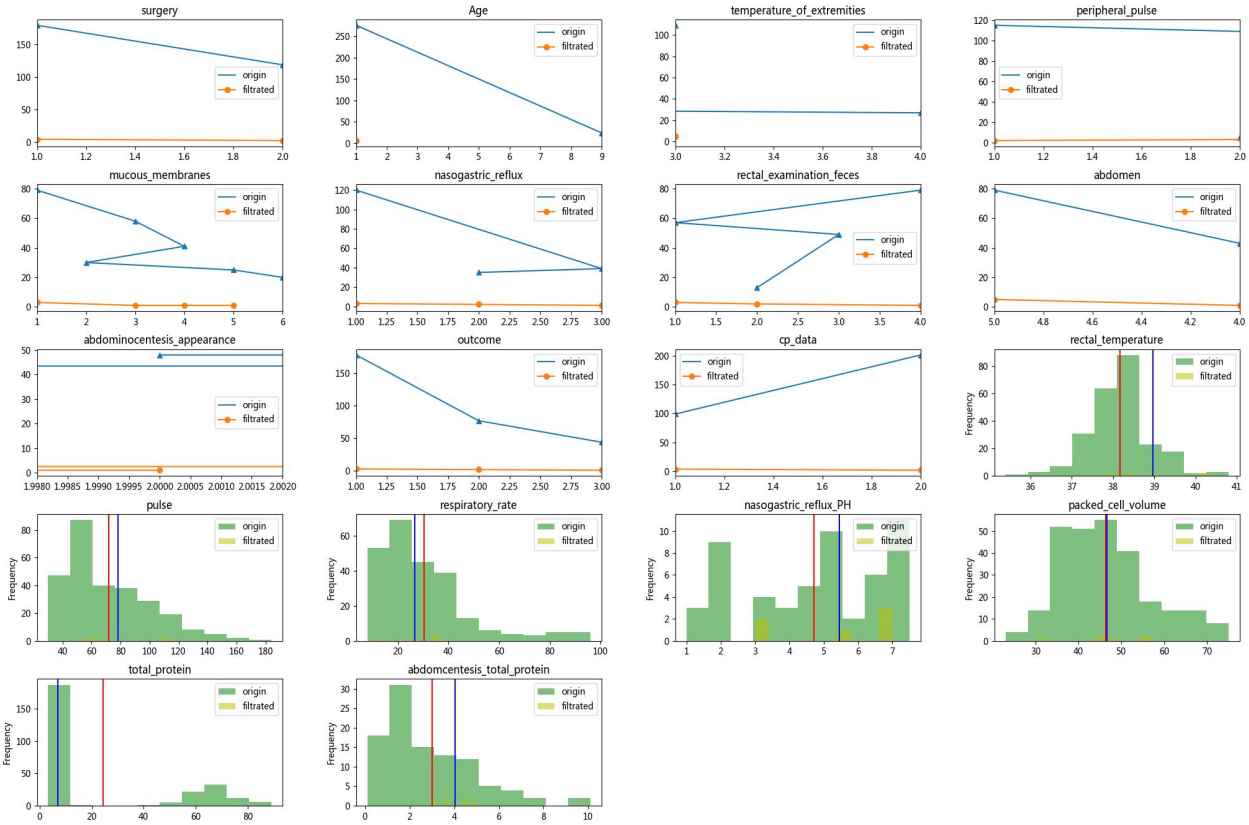
- 将缺失部分剔除
- 用最高频率值来填补缺失值
- 通过属性的相关关系来填补缺失值
- 通过数据对象之间的相似性来填补缺失值

处理后，可视化地对比新旧数据集。

该部分由两部分代码构成，分别为"figCompare.py"与"deal_absent_data.py"，其中"figCompare.py"用户绘制原始数据与处理后数据的可视化对比图，"deal_absent_data.py"中含有上文所述的 4 种处理缺失数据的方法。代码具体说明见代码中的注释。

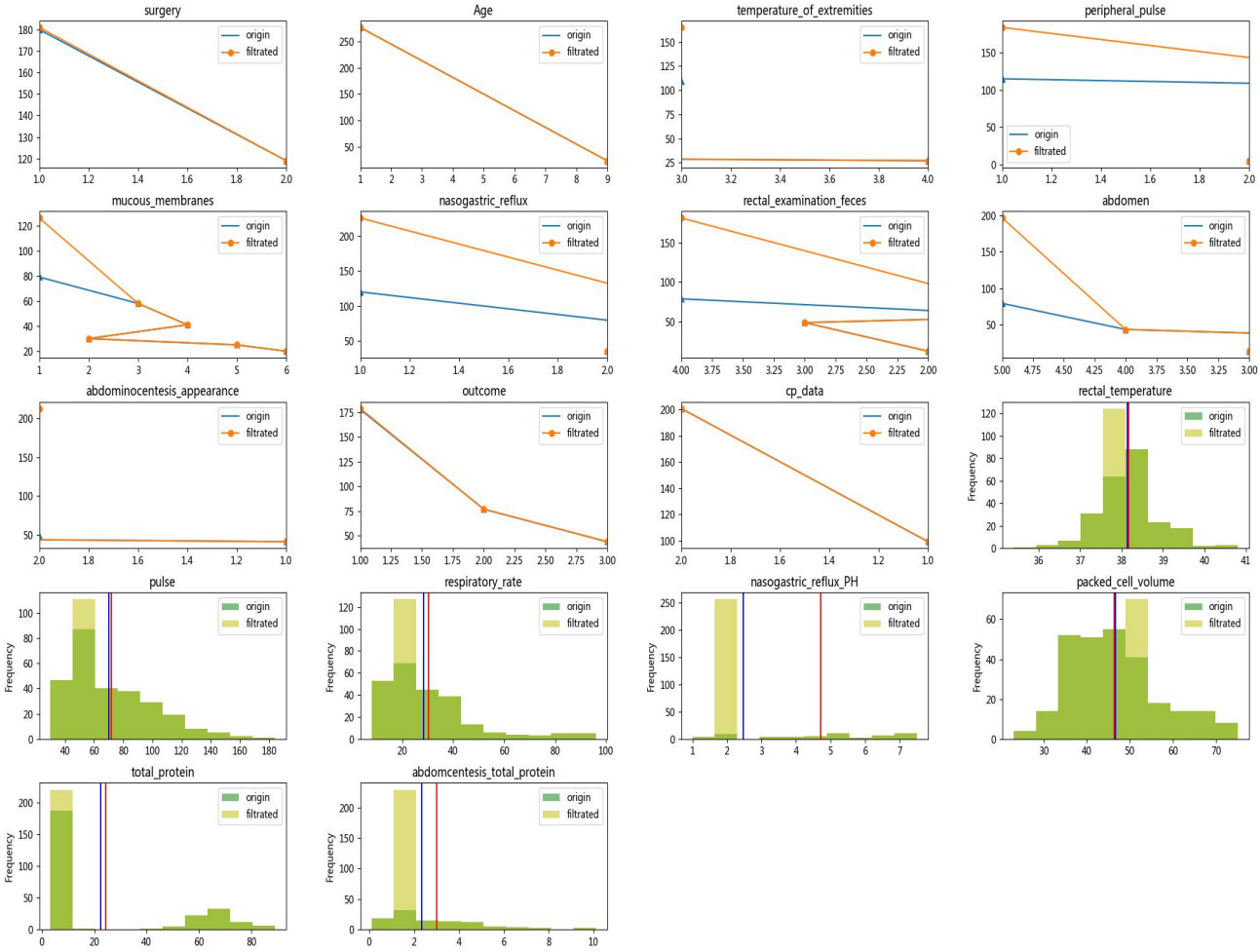
- （1）将缺失部分剔除

由于数据集中缺失的数据较多，把含有缺失项的元组都剔除之后，所剩的数据很少，严重破坏了原始数据对比处理前后的数据，如下图所示，可见这种处理方式不合适。处理后的数据保存在"data/data_filtrated_delete.csv"路径下，对比图在"image/handle_data/missing_data_delete.png"路径下。

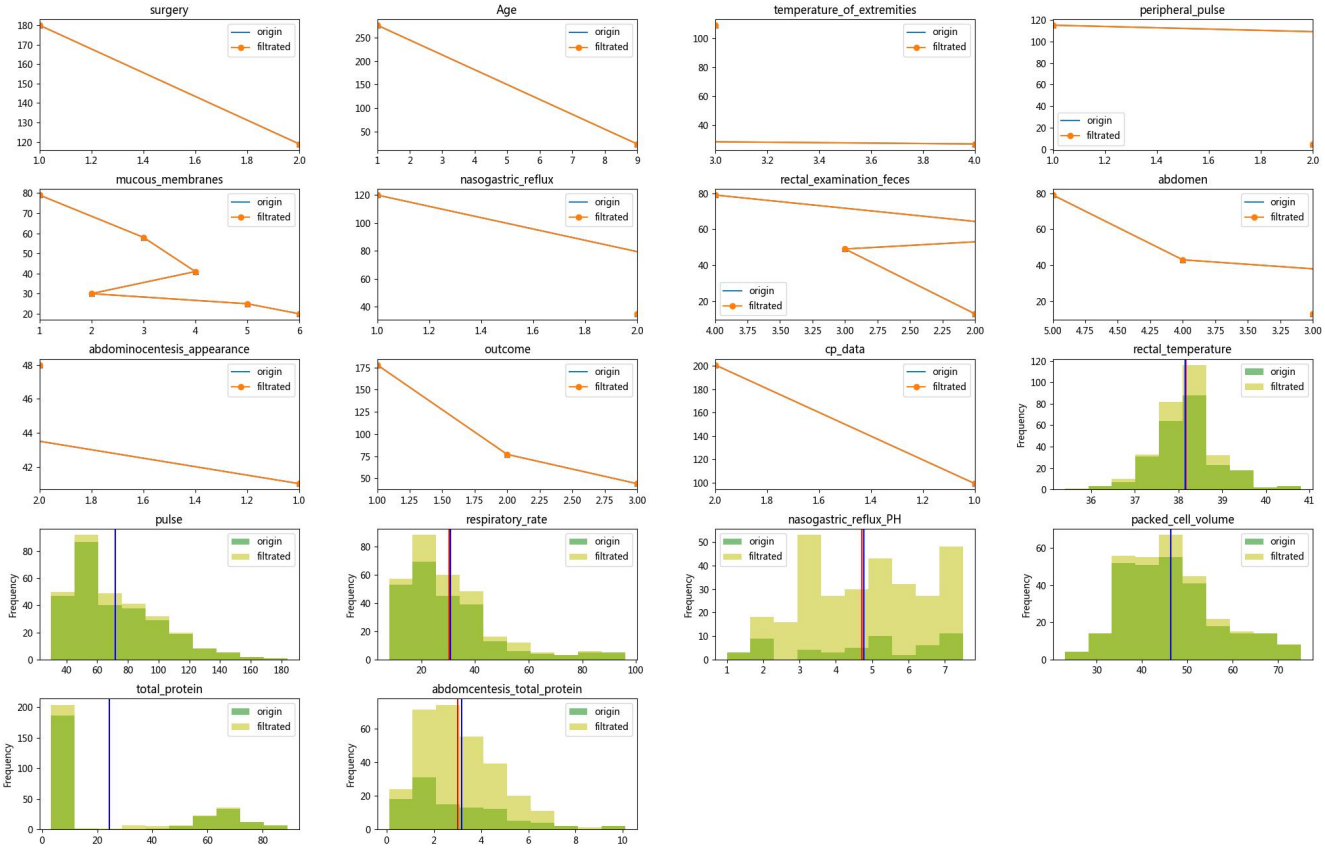


- （2）用最高频率值来填补缺失值

先计算出各属性的项的最高频率，并将该值替代缺失值。处理后的数据保存在"data/data_filtrated_filledByMost.csv"路径下，对比图在"image/handle_data/missing_data_filledByMost.png"路径下。由对比图可知，填补后的数据一定程度上扭曲了原始数据的分布，处理后的数据与原始数据相差甚大，尤其是"nasogastric_flux_PH"，所以这种处理缺失值得方式也不合适。

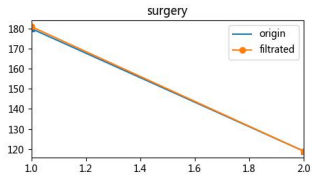


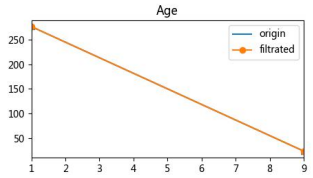
- （3）通过属性的相关关系来填补缺失值
- 该种处理方式在 python 中较为简单，可以直接使用 interpolate()函数来完成利用相关关系来填补缺失值。处理后的数据保存在"data/data_filtrated_filledByCorelation.csv"路径下，对比图在"image/handle_data/missing_data_filledByCorelation.png"路径下。由对比图可以发现，处理后数据的分布和原始数据的分布比较接近，一定程度上具有还原丢失数据的能力。对比图如下图所示:

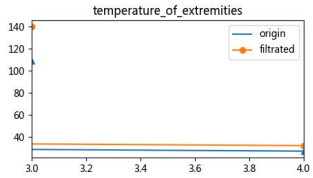


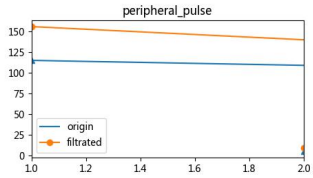
- （4）通过数据对象之间的相似性来填补缺失值

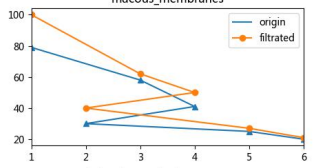
该方法较之前几种方法困难，首先需要定义两个元组之间的相似性，在定义相似性之前，需要将所有的数据进行标准化，让数值属性的值分布在[-1,1]上。标称属性不相同时，dist 值增加 1，数值属性间的距离则用其之差来表示。由于计算相似性列表比较耗时，所以第一次计算得到相似性列表 dist 时，将其保存在"data/dist.json"路径下，下次使用时可以直接调用。对于缺失值得元组，用与之 dist 值最小的项来代替。处理后的数据保存在"data/data_filtrated_filledBySimilar.csv"路径下，对比图在"image/handle_data/missing_data_filledBySimilar.png"路径下。对比图如下图所示：

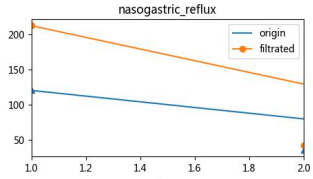

surgery
origin
filtrated

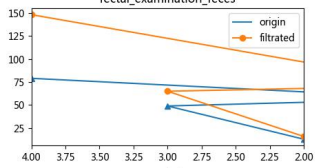

Age
origin
filtrated

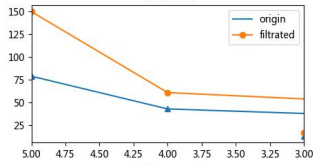

temperature_of_extremities
origin
filtrated

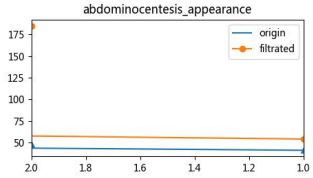

peripheral_pulse
origin
filtrated

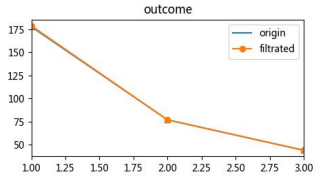

mucous_membranes
origin
filtrated

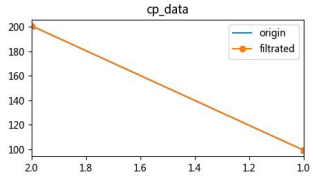

nasogastric_reflux
origin
filtrated

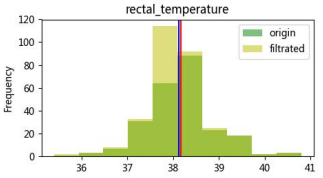

rectal_examination_feces
origin
filtrated

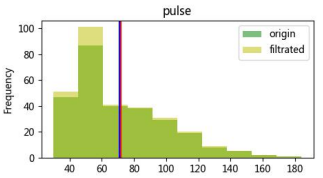

abdomen
origin
filtrated

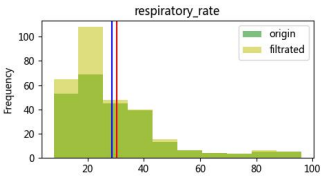

abdominocentesis_appearance
origin
filtrated

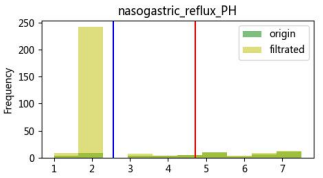

outcome
origin
filtrated

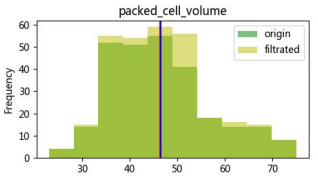

cp_data
origin
filtrated

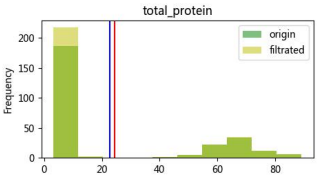

rectal_temperature
origin
filtrated
Frequency

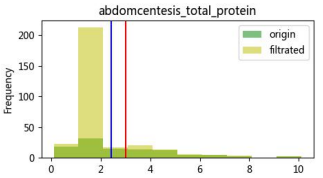

pulse
origin
filtrated
Frequency


respiratory_rate
origin
filtrated
Frequency


nasogastric_reflux_PH
origin
filtrated
Frequency


packed_cell_volume
origin
filtrated
Frequency


total_protein
origin
filtrated
Frequency


abdomcentesis_total_protein
origin
filtrated
Frequency