**数据探索性分析与预处理**

**——马的疝病分析**

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。

1. **数据说明**

数据集共有368个样本，28个属性。属性值包含标称属性和数值属性两类。数据集中30%的数据是缺失的。

1. **数据分析要求**
   1. **数据可视化和摘要**

**3.1.1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。

**3.1.2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

* 1. **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：

(a) 将缺失部分剔除

(b) 用最高频率值来填补缺失值

(c) 通过属性的相关关系来填补缺失值

(d) 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。

1. **提交材料说明**

|  |  |
| --- | --- |
| 文件名称 | 内容说明 |
| 李盛楠-2120161009-作业一实验报告.docx | 实验报告 |
| horse\_data\_analysis.py | 实验代码 |
|  | 针对数值属性绘制的直方图、盒图、qq图以及分别用四种不同策略处理缺失值后，新数据集与原数据集的对比图。（原本存储在image文件夹中） |
|  | 分别用四种不同策略处理缺失值后产生的新数据集。（原本存储在data\_output文件夹中） |
| numerical\_attribute.txt | 包含针对数值属性给出的最大值、最小值、均值、中位数、四分位数以及缺失值个数。 |
| frequency.txt | 包含针对标称属性的每个可能取值的频数统计。 |
| data.txt | 原始数据集 |
| data.csv | 转换格式后的数据集 |

1. **分析实验过程与代码解读**
   1. **环境和使用的库**

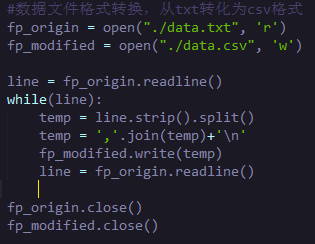
代码用python 2.7实现



其中，除了numpy等常用库以外，pandas用来进行进行数据分析，matplotlib的pyplot子库提供了和matlab类似的绘图API，以满足实验对于数据可视化的要求。

* 1. **实验过程分析与代码解读**
     1. **数据读取及格式处理**

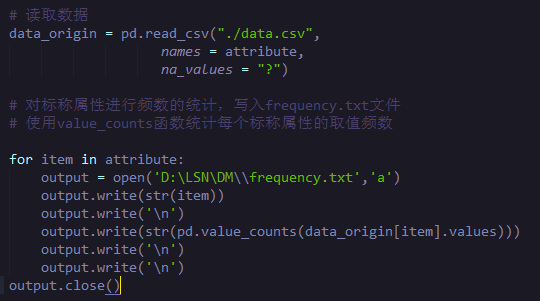
首先要读取下载的数据集。我是将数据集存成txt格式，由于数据条目明显的行列关系，即每个属性为一列，每个样本为一行，故将其转换为csv格式，以方便后续实验操作。具体代码如下图所示：



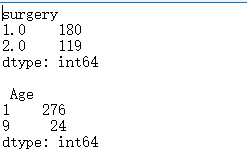
* + 1. **数据摘要**

数据摘要主要分为两部分，针对标称属性和数值属性具有不同的摘要指标。

针对标称属性，其具有有限个可能的取值。因此，针对标称属性的数据摘要工作就转化为了一个计数问题。具体地，可以统计每个可能出现的值的出现频数，从而找到一些客观规律或者可以为之后的其他工作打下基础。pd.value\_counts(data\_origin[item].values)为本实验中核心计算频数的一步，其中data\_origin存储着读取的数据。具体代码如下图所示。

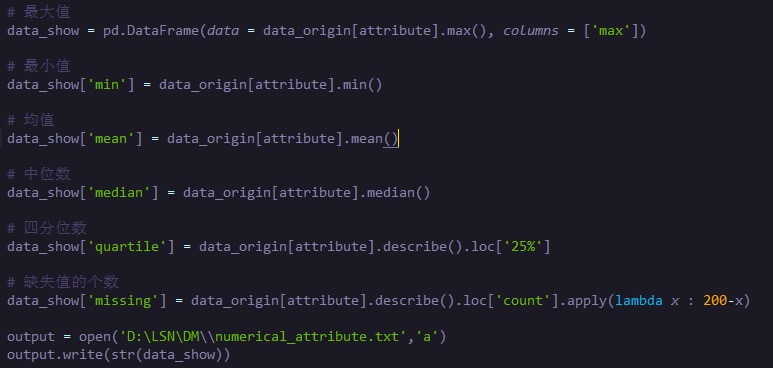


为了方便观察和提交作业，本实验将针对标称属性不同取值的统计结果写入了frequency.txt文件。部分文件内容如下图所示。

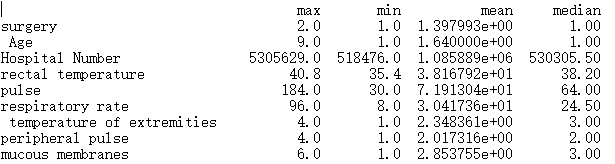


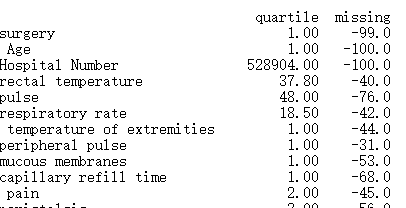
图中展示了关于属性surgery和属性age每个可能的取值所对应的频数。以age为例，有1和9两个取值，1对应的频数为276，9对应的频数为24。

针对数值属性，需要找到其最大值、最小值、均值、中位数、四分位数以及缺失值的个数，来完成数据摘要工作。针对这部分，主要是通过建立对象，再调用python的pandas库里面的函数，直接就可以求得。求最小值的代码为data\_show['min'] = data\_origin[attribute].min()，最大值、均值、中位数类似。data\_show['quartile'] = data\_origin[attribute].describe().loc['25%']，求四分位数。data\_show['missing'] = data\_origin[attribute].describe().loc['count'].apply(lambda x : 200-x)，求缺失值个数。其中data\_show为最终被写入numerical\_attribute.txt文件的结果。具体的代码如下图所示。



上图中最后两行则表示将数据结果写入文件，部分内容如下图所示，详情可查询附件numerical\_attribute.txt。



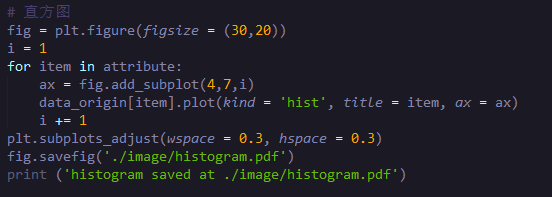


* + 1. **数据可视化**

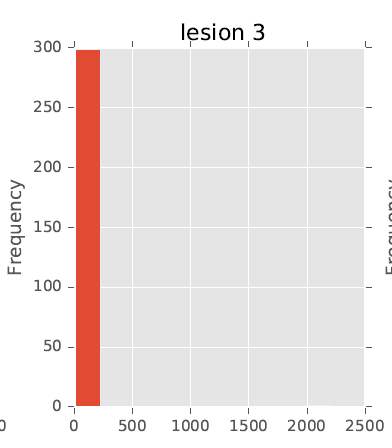
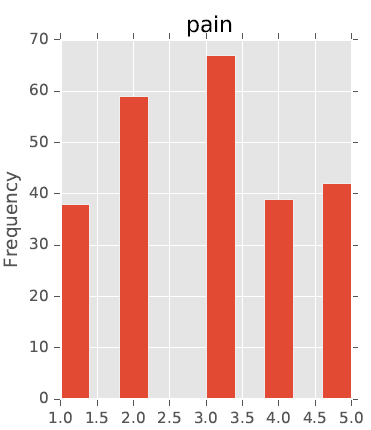
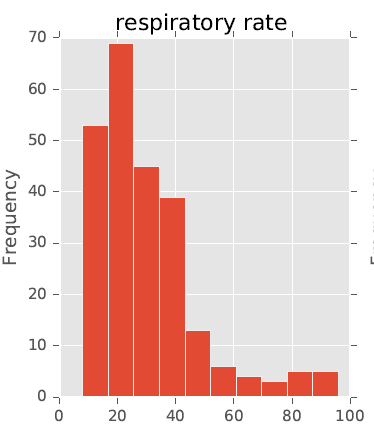
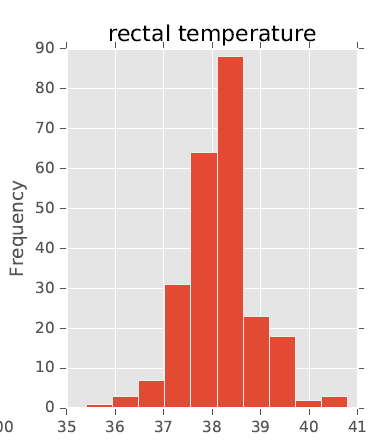
数据可视化主要是针对数值数据进行绘图。通过绘制直方图、qq图以及盒图。

通过qq图可以观察某属性的数值分布是否符合正态分布。具体地，由于qq图是一种散点图,对应于正态分布的qq图,就是由标准正态分布的分位数为横坐标,样本值为纵坐标的散点图. 要利用qq图鉴别样本数据是否近似于正态分布,只需看qq图上的点是否近似地在一条直线附近,而且该直线的斜率为标准差,截距为均值。

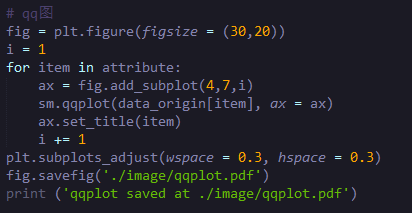
另外还可以通过盒图可以识别数据的离群点。通过盒图的画法就可以直观的知道在盒图中能够看到最小值、第一个四分位数、均值、第三个四分位数和最大值。因此，我们可以看出数据的大致分布，继而观察到离群点。



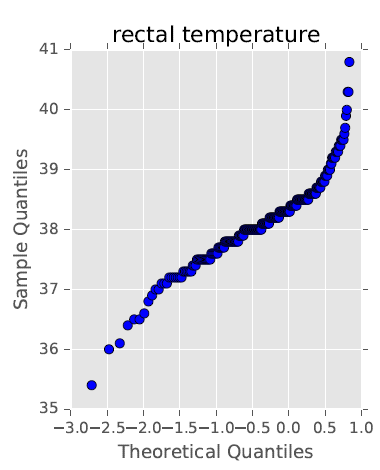
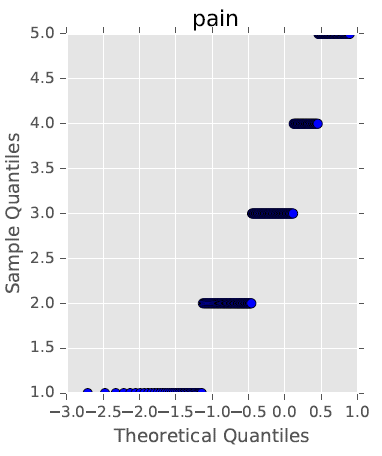
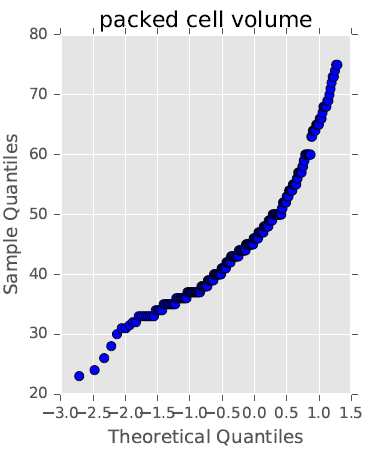
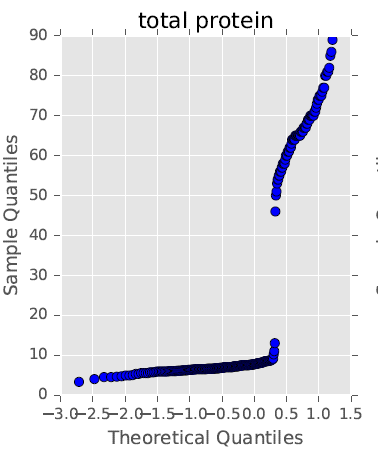
代码中用data\_origin[item].plot(kind = 'hist', title = item, ax = ax)绘制直方图。其中plot函数用于画图，kind这个参数用来控制所绘图的种类，这里选择’hist’表示直方图，title则是标明直方图所对应的属性名称。ax = fig.add\_subplot(4,7,i)将28个直方图以子图的形式画入同一个pdf文件中。全部属性的直方图共28个，可以查询histogram.pdf。下面只展示几个属性的直方图。



从属性rectal temperature和属性respiratory rate的直方图分布来看，它们都属于属性数值比较集中在某一区间内；从属性pain和属性lesion 3的直方图来看，它们数据属性数值集中分布在某一个或者几个值，不存在其他取值出现的情况。

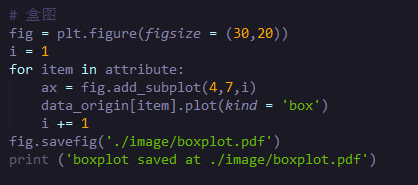


代码中用sm.qqplot(data\_origin[item], ax = ax)绘制qq图，并且也用item，也就是属性的名称，设置了每个属性所对应qq图的标题名称。与直方图相同，代码ax = fig.add\_subplot(4,7,i)将28个qq图以子图的形式画入同一个pdf文件中。全部属性的qq图共28个，可以查询qqplot.pdf。下面只展示几个属性的qq图。

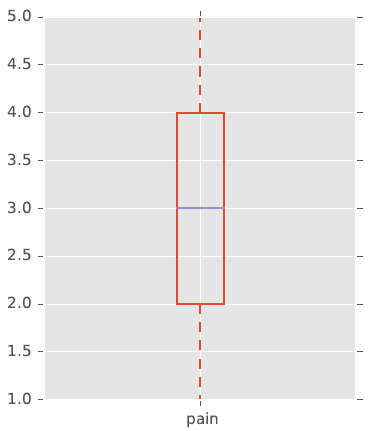
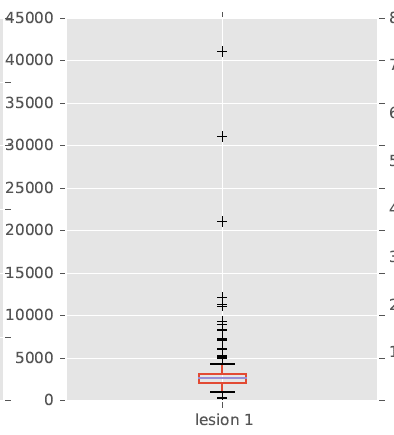
   

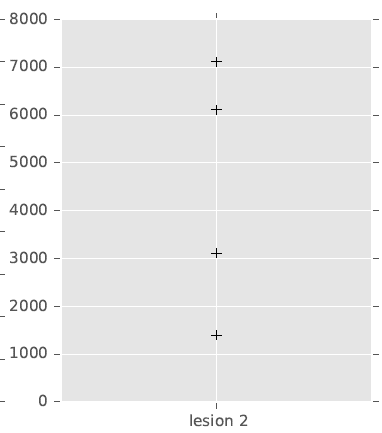
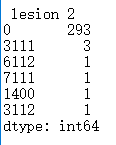
从属性rectal temperature和属性packed cell volume的qq图来看，它们的较好的分布在一条线上面；从属性pain和属性total protein的qq图来看，它们的数据数据分布在水平的几段，或者是非水平的两段。

利用qq图可以来判断数据分布是否符合正态分布，从上面的例子来看属性packed cell volume的qq图最接近一条直线，因此大致能判断其符合正态分布。



代码中用data\_origin[item].plot(kind = 'box') 绘制盒图。其中plot函数用于画图，kind这个参数用来控制所绘图的种类，这里选择’box’表示盒图，title则是标明盒图所对应的属性名称。ax = fig.add\_subplot(4,7,i)将28个盒图以子图的形式画入同一个pdf文件中。全部属性的盒图共28个，可以查询boxplot.pdf。下面只展示几个属性的盒图。

从属性pain的盒图来看来看，数据集在该属性的取值分布较为合理，未观察到离群点。从属性lesion 1的盒图来看，数据集在该属性的取值存在较多的离群点。从属性lesion 2的盒图上只存在四个异常点。对比之前对于该属性各个取值的频数统计可知，除去在此属性处存在缺失值的数据之外的300个数据中293个都为0，而其它几个数据存在较大取值，因此，在盒图的绘制中将其处理为离群点。其实对于这样的属性来说，用盒图观察离群点的意义不是太大。

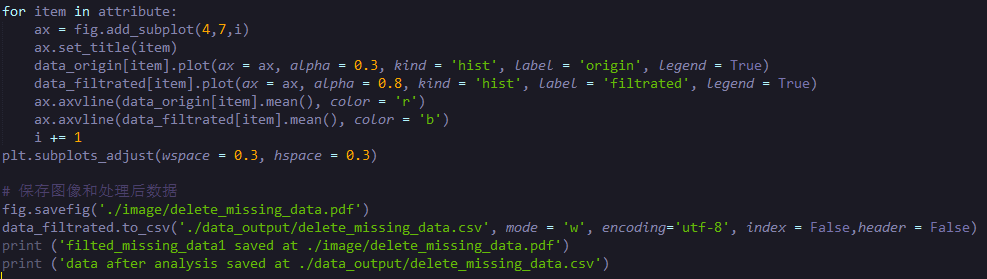
* + 1. **数据缺失值处理及其与原数据对比**

实验要求用四种不同的策略分别对数据中的缺失值进行处理，并作图与原数据集进行对比。

1. 将缺失部分剔除

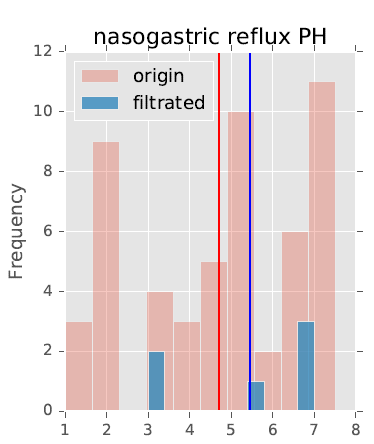
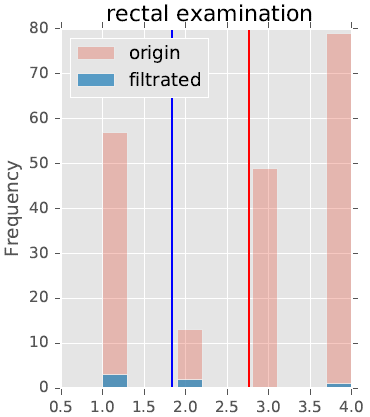
将缺失部分剔除是一个最简单的策略。具体地，对于数据集进行检索，若该数据条目中存在带有缺失值的属性，则整条删除该数据条目。这样做确实很方便、简单，最大程度的保留了数据的真实性，因为不需要通过策略去猜测缺失值的取值。然而也存在很大的问题，在数据不丰富的情况下，这样的简单剔除会大大减少可用数据量。

以本数据集为例，其中有30%的数据缺失，但对应到整条数据条目上就删去了约98%的数据条目。最终仅剩六条完整数据，这样的一个新数据集不能单独用于任何后续的任务。因此，这样的处理方法在这种数据集规模小，切确实数据较多的数据集是不太适用的。



代码data\_filtrated = data\_origin.dropna()用来删除带有缺失值的数据条目。之后通过绘制直方图来对比新数据集与原数据集。 ax = fig.add\_subplot(4,7,i) 将28个对比图以子图的形式画入同一个pdf文件中。ax.set\_title(item)每个子图以item，也就是属性名命名。data\_origin[item].plot(ax = ax, alpha = 0.3, kind = 'hist', label = 'origin', legend = True)用来绘制原数据集的直方图。ax设置了子图在pdf中的位置，alpha设置了图的透明度，kind选择’hist’表示绘制直方图，label表明了数据是原数据集，legend为True表示设置图例。用同样的方法绘制新数据集的直方图，代码为data\_filtrated[item].plot(ax = ax, alpha = 0.8, kind = 'hist', label = 'filtrated', legend = True)。代码ax.axvline(data\_origin[item].mean(), color = 'r')和ax.axvline(data\_filtrated[item].mean(), color = 'b')分别将原数据集和新数据集的均值以垂直于x轴的形式画在直方图中。

最后，利用fig.savefig('./image/delete\_missing\_data.pdf')来将图片存储到pdf，命名为delete\_missing\_data.pdf，可以查看相关附件。下面展示几个属性的对比直方图。其中，除了半透明红色直方图外，红色的竖线代表着原始数据的均值；蓝色直方图和蓝色竖线对应着剔除了缺失值的新数据集的数据和均值。可以看出，以这种方式得到的新数据集的数据条目大大减少了，但是均值的改变不是很大。

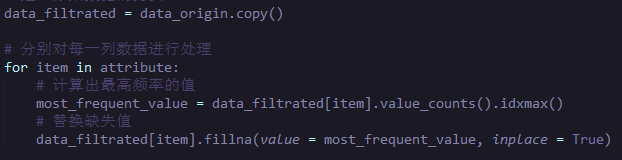
data\_filtrated.to\_csv('./data\_output/delete\_missing\_data.csv', mode = 'w', encoding='utf-8', index = False,header = False)将处理后的新数据集文件存为csv类型，命名为delete\_missing\_data.csv，可以查看相关附件。

1. 用最高频率值来填补缺失值

运用最高频率值来填补缺失值是一个很直观且容易理解的方法。因为对于某个属性来说，大多数数据条目在该属性的取值具有一些代表性，例如该数据集中age这个属性，其中276条数据条目的age值为1，24条数据条目的age值为9。那么我们可以大致认为该数据集反应的是一个age为1的马群的数据，因此缺失值可以用1来填补。

当然，这样做也存在一些问题，即当前存在的数据存在这样的频数分布，但是不能说名缺失值一定是最高频率的值，只能说存在较高的概率。

具体地，针对每个属性统计其最高频率值，即对每个属性的每个取值出现的次数进行分别的计数，找到出现次数最多的取值用于填补所有数据条目该属性的缺失值。

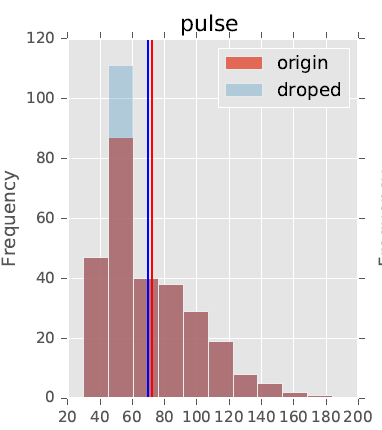
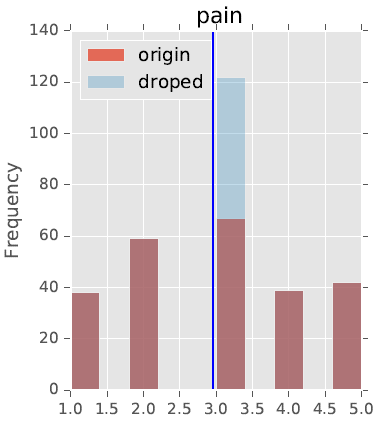


代码data\_filtrated = data\_origin.copy()用来拷贝原始的数据，防止填补操作对原始数据造成覆盖。用for循环实现针对每个属性都要进行的最高频率值的选取和缺失值的替换，循环中体现了这两部分的功能，most\_frequent\_value = data\_filtrated[item].value\_counts().idxmax()用于找到最高频数值。而后通过代码 data\_filtrated[item].fillna(value = most\_frequent\_value, inplace = True)替换缺失值，其中inplace=True很清楚的表示了替换这个动作。



同样的，分别将原数据集和新数据集的均值以垂直于x轴的形式画在直方图中，用到的代码与之前相同分别是ax.axvline(data\_origin[item].mean(), color = 'r')和ax.axvline(data\_filtrated [item].mean(), color = 'b')。

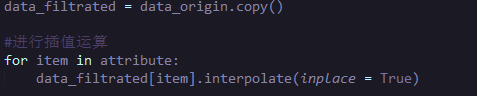
最后，利用fig.savefig('./image/most\_frequency\_missing\_data.pdf')来将图片存储到pdf，命名为most\_frequency\_missing\_data.pdf，可以查看相关附件。下面展示几个属性的对比直方图。其中，红色直方图和红色竖线代表着原始数据和其均值；蓝色直方图和蓝色竖线对应着用最高频数值填补缺失值后的新数据集的数据和其均值。可以看出，以这种方式得到的新数据集的数据分布改变非常小，基本可以形容为将原有直方图的最高的柱体的高度继续提高。这也与我们用最高频数值填补的做法吻合，因为我们就相当于增加了属性最高频数值出现的频数。均值的改变更少，有些甚至没有改变均值的大小。

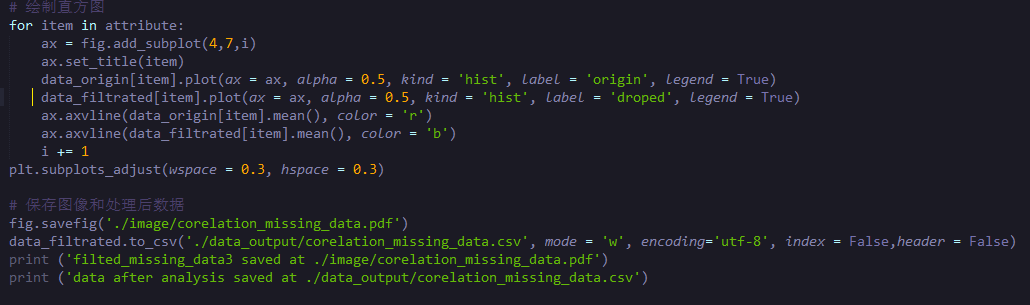
data\_filtrated.to\_csv('./data\_output/most\_frequency\_missing\_data.csv', mode = 'w', encoding='utf-8', index = False,header = False)将处理后的新数据集文件存为csv类型，命名为most\_frequency\_missing\_data.csv，可以查看相关附件。

1. 通过属性的相关关系来填补缺失值

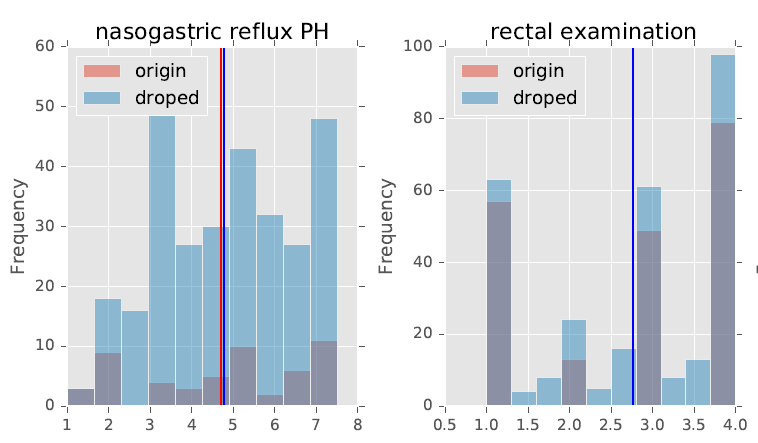
通过相关关系来填补缺失值，即通过属性的相关关系来填补缺失值。



首先，还是用data\_filtrated = data\_origin.copy()拷贝原始数据集。相关关系填补缺失值用的是pandas库中的interpolate()函数，用for循环实现针对每个属性的遍历。data\_filtrated[item].interpolate(inplace = True)完成利用相关关系填补缺失值的操作，interpolate()进行插值，参数inplace为True，表示用差值替换缺失值。



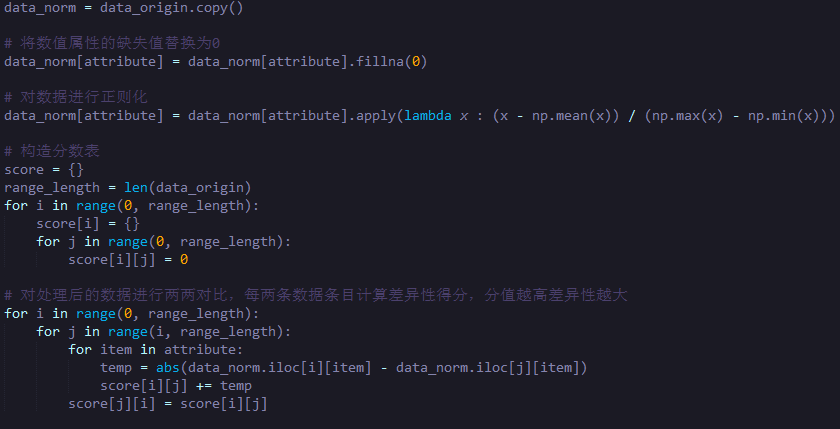
将新数据集和原数据集绘制成直方图，并将新数据集数据存入文件。相关操作与策略(a)中相同，不在此赘述。新数据集存入corelation\_missing\_data.csv文件，对比直方图存入corelation\_missing\_data.pdf文件，均可查询相关附件。下面展示几个对比直方图。其中，红色直方图和红色竖线代表着原始数据和其均值；蓝色直方图和蓝色竖线对应着用属性的相关关系来填补缺失值后的新数据集的数据和其均值。可以看出，以这种方式得到的新数据集的均值基本没有变化，填补的值的多样性也比较丰富，可能出现原数据集中没有出现过的值。然而也可以发现，原数据集中频数较高的值并不是被最多用于填补缺失值的。原数据集中频数最高的值在新数据中并不一定仍保持最高频数。



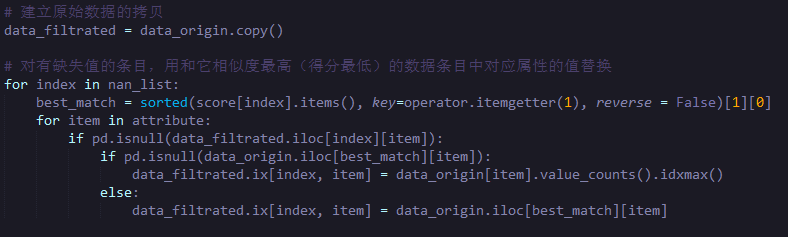
1. 通过数据对象之间的相似性来填补缺失值

利用数据对象之间的相似性来填补缺失值是一种较为合理的方式。具体地，对比数据条目之间的相似度，即各个属性值的相似度，用最相似的数据条目对应的属性来填补缺失值。虽然其他属性值的相同并不一定使其在缺失值所对应的属性处仍具有相同的值，但至少概率是非常大的。从实际的生活中我们也可以发现，如果在很多方面都相同的两个事物，很可能具有更多的相同属性。

实验中以每条数据条目为一个单位，去计算它与其他条目的相似性。然后找到一条与其最相似的数据，再用这条最相似数据的属性取值填补缺失值。



首先用data\_norm = data\_origin.copy()拷贝原始数据，便于下面进行修改、正则化。data\_norm[attribute] = data\_norm[attribute].fillna(0)将缺失值替换为0。代码data\_norm[attribute] = data\_norm[attribute].apply(lambda x : (x - np.mean(x)) / (np.max(x) - np.min(x)))利用函数apply()对数据进行正则化处理，以方便后续计算相似性。然后构造了一个分数表，用于存储相似性分数。最后，通过代码 temp = abs(data\_norm.iloc[i][item] - data\_norm.iloc[j][item])进行数据条目之间对应属性值之间的相似性对比。无论差值为正为负，均不影响，因为0为完全一致，而与0相差越大越不相似，故取绝对值。最终得到score，存储了所有的相似性得分，其中score[i][j]=score[j][i]，该值越小，则证明i与j相似度越高。



用best\_match = sorted(score[index].items(), key=operator.itemgetter(1), reverse = False)[1][0]排序score，找到最相似的数据条目。之后进行缺失值的替换。

将新数据集和原数据集绘制成直方图，并将新数据集数据存入文件。相关操作与策略(a)中相同，不在此赘述。新数据集存入similarity\_missing\_data.csv文件，对比直方图存入similarity\_missing\_data.pdf文件，均可查询相关附件。下面展示几个对比直方图。其中，红色直方图和红色竖线代表着原始数据和其均值；蓝色直方图和蓝色竖线对应着用数据对象相关性来填补缺失值后的新数据集的数据和其均值。可以看出，新数据集直方图与原数据集直方图的重合度还是很高的，均值也几乎没有变化。因此可以说，大部分维持了原有的分布。但也有像下面左图这种情况，某个值被多次用于填补缺失值，继而对均值产生了较大的影响。对于这种属性，用该方法填补缺失值确实有些不妥。

