江明明 2120161000

数据挖掘实验一报告

实验任务：

1.数据可视化和摘要

1） 对于标签属性，我们计算了每个可能取值的频数。
2） 对于数值属性，我们计算了最小值、最大值、均值、中位数、四分位数等。
3） 在数据可视化部分，我们对数值属性绘制了直方图，并使用 qq 图判断该属性是否服从正态分布，绘制盒图，对离群点进行识别。

2.缺失值处理，分别使用下列四种策略对缺失值进行处理:

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

实验过程：

本实验使用 python 的 pandas 库进行数据预处理与可视化，数据中共有 28 个属性，其中第 1,2,3,7,8,9,10,11,12,13,14,15,17,18,21,23,24,25,26,27,28 属性为标签属性，第 4,5,6,16,19,20,22 属性为数值属性。

读取数据：

使用 pandas 的 read_csv()方法读取数据文件，使用 attrname 保存属性名，使用 labelattr 保存标称属性，valueattr 保存数值属性。

```
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt

#读取数据
df=pd.read_csv('horse-colic.data_backup.txt',sep=' ')
df=df.drop(df.columns[[28]], axis=1)
df=df.apply(pd.to_numeric,errors='coerce')
attrname=['surgery','Age','Hospital_Number','rectal_temperature','pulse',
'respiratory_rate','temperature_of_extremities','peripheral_pulse',
'mucous_membranes','capillary_refill_time','pain','peristalsis',
'abdominal_distension','nasogastric_tube','nasogastric_reflux',
'nasogastric_reflux_PH','rectal_examination','abdomen','packed_cell_volume',
'total_protein','abdominocentesis_appearance','abdomcentesis_total_protein',
'outcome','surgical_lesion','#1_lesion','#2_lesion','#3_lesion','cp_data']
labelattr=[1,2,3,7,8,9,10,11,12,13,14,15,17,18,21,23,24,25,26,27,28]
valueattr=[4,5,6,16,19,20,22]
```

统计标称属性不同值的频数：

使用 pandas.DataFrame.value_counts()方法统计不同可能值的频数。

```
#分析标签属性的出现频数
for attrid in labelattr:
    print(df[attrname[attrid-1]].value_counts())
    print()
```

以下为部分属性的频数统计结果：

```
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
```

数值属性的最大值、最小值、均值、中位数、四分位数：

使用 pandas.Series.max()、min()、mean()、median()、quantile()等方法计算数值属性的

最大值、最小值、均值、中位数、四分位数。

```
#数值属性的最小值，1/4分位数，中位数，均值，3/4分位数，最大值
for attrid in valueattr:
    print(attrname[attrid - 1])
    series=df[attrname[attrid - 1]].apply(pd.to_numeric, errors='coerce')
    series=series[series.notnull()]
    print('min:',series.min())
    print('1/4 quantile:',series.quantile(0.25))
    print('mean:',series.mean())
    print('median:',series.median())
    print('3/4 quantile:',series.quantile(0.75))
    print('max:',series.max())
    print()
```

以下为部分结果：

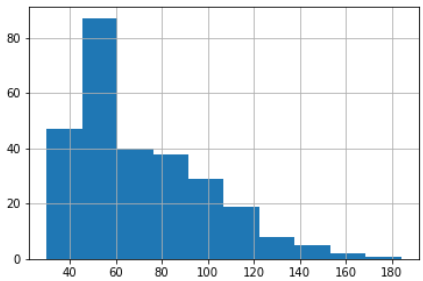
```
rectal_temperature
min: 35.4
1/4 quantile: 37.8
mean: 38.16791666666669
median: 38.2
3/4 quantile: 38.5
max: 40.8

pulse
min: 30.0
1/4 quantile: 48.0
mean: 71.91304347826087
median: 64.0
3/4 quantile: 88.0
max: 184.0
```

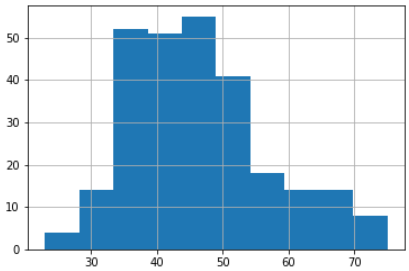
绘制直方图：

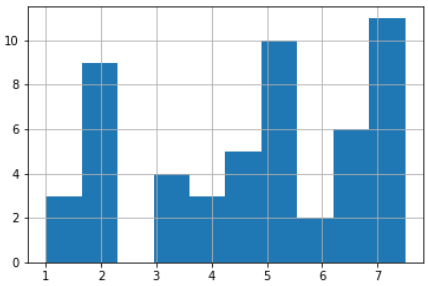
使用 pandas.Series.hist()方法绘制属性的直方图。

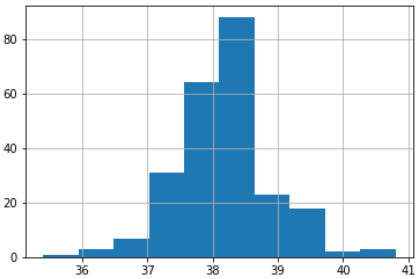
```
#直方图
for attrid in valueattr:
#    print(attrname[attrid - 1])
    series=df[attrname[attrid - 1]]
    series=series[series != '?'].apply(pd.to_numeric, errors='coerce')
    fig=series.hist().get_figure()
    fig.savefig('./pngs/hist_'+attrname[attrid -1]+'.png')
    fig.clear()
```

以下为部分属性的直方图：







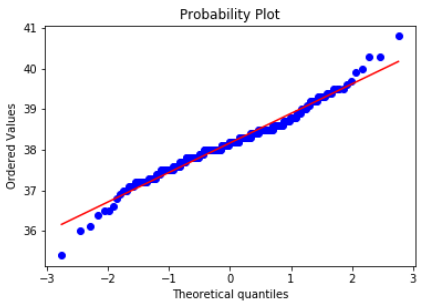


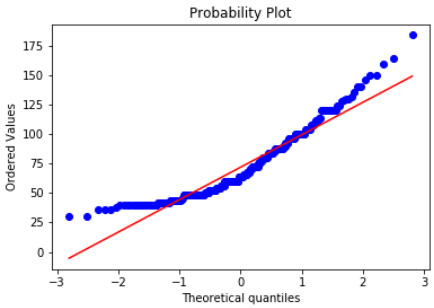
绘制 QQ 图：

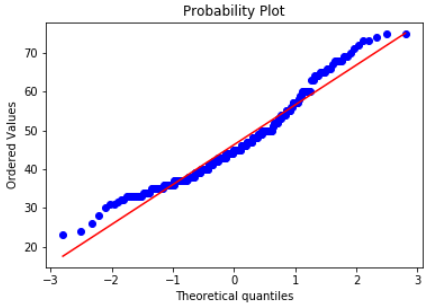
使用 scipy.stats.probplot(series, dist="norm", plot=plt)方法绘制属性的 QQ 图。

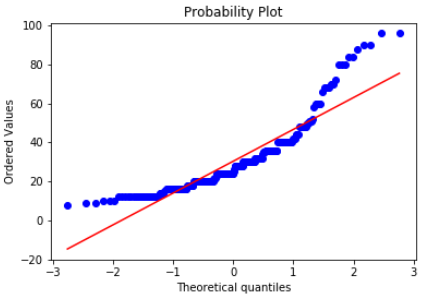
```
#qq图
for attrid in valueattr:
    print(attrname[attrid - 1])
    series=df[attrname[attrid - 1]]
    series=series[series != '?'].apply(pd.to_numeric, errors='coerce')
    stats.probplot(series, dist="norm", plot=plt)
    fig=plt.gcf()
    fig.savefig('./pngs/qqplot_'+attrname[attrid -1]+'.png')
    fig.clear()
```

以下为部分属性的 QQ 图：







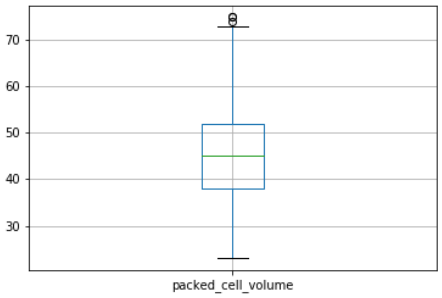


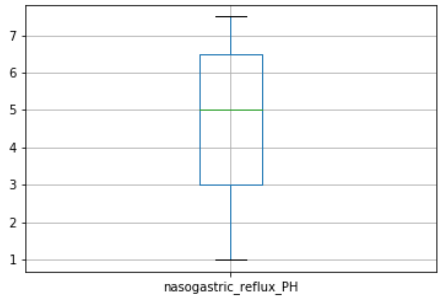
绘制盒图：

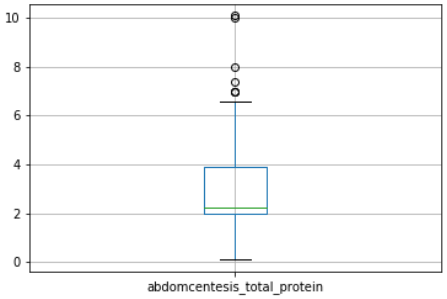
使用 pandas.DataFrame.boxplot()绘制盒图。

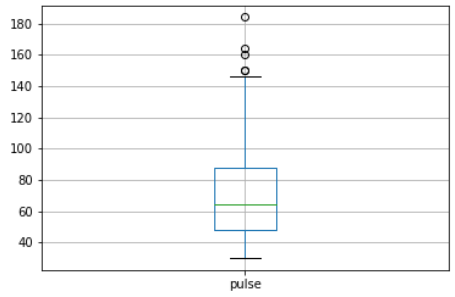
```
#盒图
for attrid in valueattr:
    series=df[attrname[attrid - 1]]
    series=series[series != '?'].apply(pd.to_numeric, errors='coerce')
    fig=pd.DataFrame(series).boxplot().get_figure()
    fig.savefig('./pngs/boxplot_'+attrname[attrid -1]+'.png')
    fig.clear()
```

以下为部分属性的盒图：









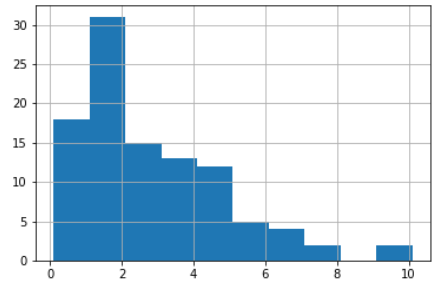
缺失值处理：

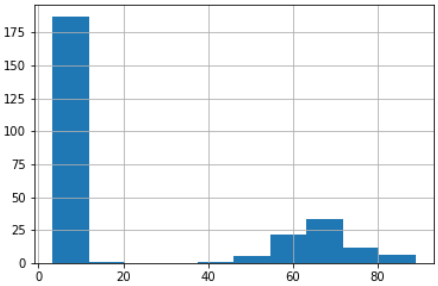
使用 pandas.DataFrame.dropna()方法剔除含有缺失值记录。
使用 pandas.DataFrame.fillna()方法对缺失值进行填补。

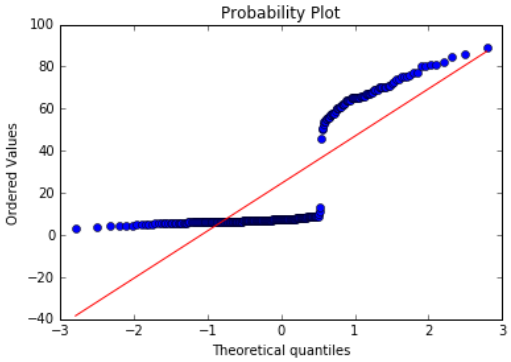
```
drop_any_df=df.dropna()
```

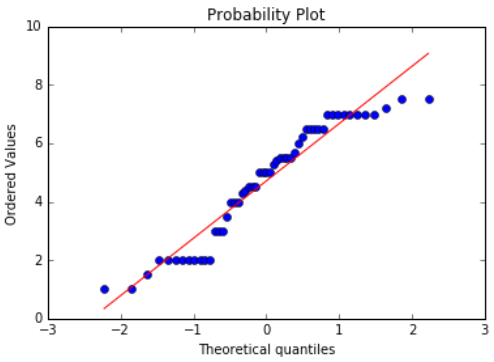
```
high_frequency_fillna_df=df.fillna(df.mode().iloc[0])
```

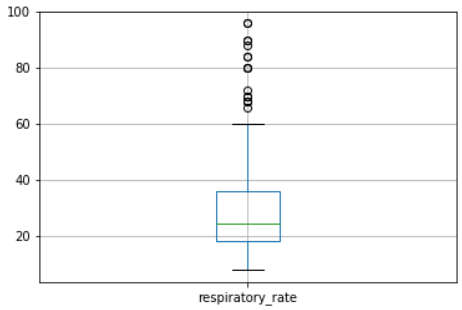
进行缺失值处理后，使用前面的程序绘制相应的图。
部分结果如下：

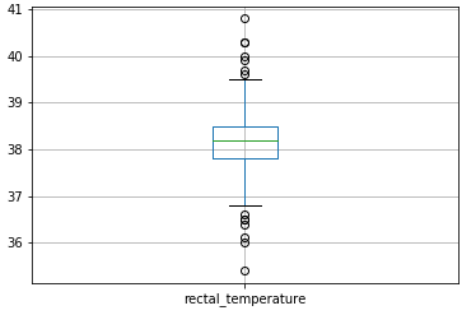

0
5
10
15
20
25
30
0
2
4
6
8
10


0
25
50
75
100
125
150
175
0
20
40
60
80


Probability Plot
Ordered Values
Theoretical quantiles
100
80
60
40
20
0
−20
−40
−3
−2
−1
0
1
2
3


Probability Plot
Ordered Values
Theoretical quantiles
10
8
6
4
2
0
−3
−2
−1
0
1
2
3


100
80
60
40
20
respiratory_rate


41
40
39
38
37
36
rectal_temperature