winemag-data_first150k.csv 数据集分析

1.数据摘要

1.1 读取数据

根据数据类型具体分析数值属性

```
Unnamed: 0       int64
country         object
description     object
designation     object
points           int64
price          float64
province        object
region_1        object
region_2        object
variety         object
winery          object
dtype: object
```

1.2 标称属性可能取值的频数

(1) country

	country
US	62397
Italy	23478
France	21098
Spain	8268
Chile	5816
Argentina	5631
Portugal	5322
Australia	4957
New Zealand	3320
Austria	3057
Germany	2452
South Africa	2258
Greece	884
Israel	630
Hungary	231
Canada	196
Romania	139
Slovenia	94
Uruguay	92
Croatia	89
Bulgaria	77
Moldova	71
Mexico	63
Turkey	52
Georgia	43
Lebanon	37
Cyprus	31
Brazil	25
Macedonia	16
Serbia	14
Morocco	12
England	9
Luxembourg	9

(2) province

	province
California	44508
Washington	9750
Tuscany	7281
Bordeaux	6111
Northern Spain	4892
Mendoza Province	4742
Oregon	4589
Burgundy	4308
Piedmont	4093
Veneto	3962
South Australia	3004
Sicily & Sardinia	2545
New York	2428
Northeastern Italy	1982
Loire Valley	1786
Alsace	1680
Marlborough	1655
Southwest France	1601
Central Italy	1530
Southern Italy	1439
Champagne	1370
Catalonia	1352
Rhône Valley	1318
Colchagua Valley	1201
Languedoc-Roussillon	1082
Douro	1075
Provence	1021
Port	903
Maipo Valley	895
Other	889
...	...

(3) variety

	variety
Chardonnay	14482
Pinot Noir	14291
Cabernet Sauvignon	12800
Red Blend	10062
Bordeaux-style Red Blend	7347
Sauvignon Blanc	6320
Syrah	5825
Riesling	5524
Merlot	5070
Zinfandel	3799
Sangiovese	3345
Malbec	3208
White Blend	2824
Rosé	2817
Tempranillo	2556
Nebbiolo	2241
Portuguese Red	2216
Sparkling Blend	2004
Shiraz	1970
Corvina, Rondinella, Molinara	1682
Rhône-style Red Blend	1505
Pinot Gris	1365
Barbera	1365
Cabernet Franc	1363
Sangiovese Grosso	1346
Pinot Grigio	1305
Viognier	1263
Bordeaux-style White Blend	1261
Champagne Blend	1238
Port	1058
...	...

1.3 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

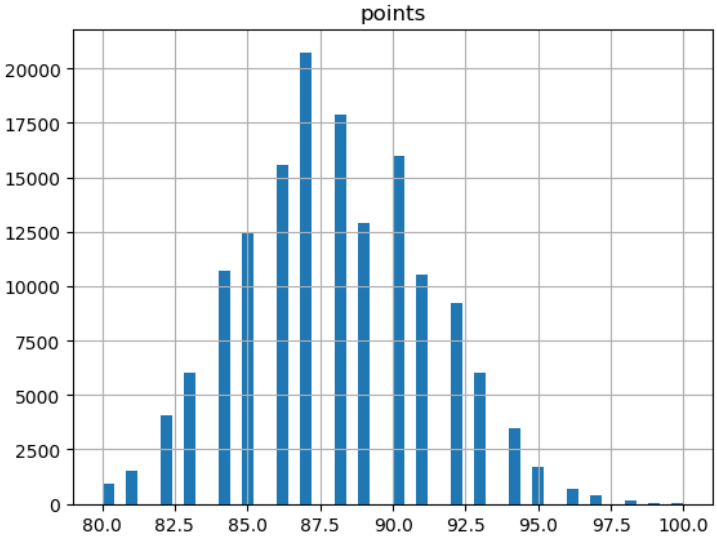
选取数值属性，分别使用.max(),.min(),.mean(),.median(),.quantile()等函数获取属性最大、最小、均值、中位数、四分位数。使用.isnull().sum()函数获取缺失值个数。

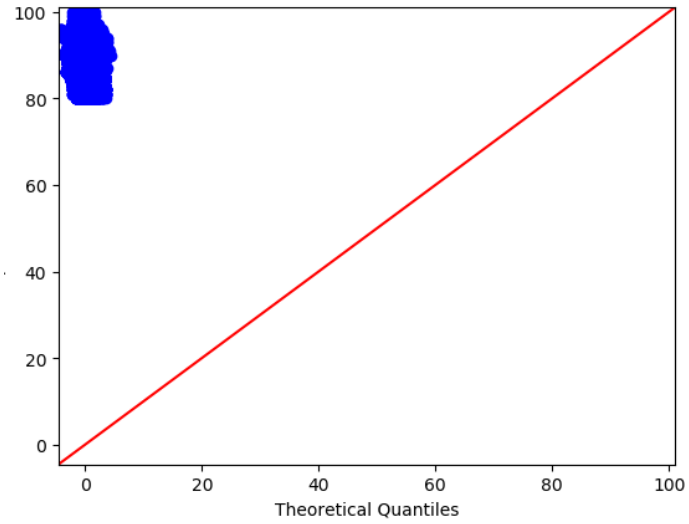
```
数值属性最大值:
points     100.0
price     2300.0
dtype: float64
数值属性最小值:
points    80.0
price      4.0
dtype: float64
数值属性均值:
points    87.888418
price     33.131482
dtype: float64
数值属性中位数:
points    88.0
price     24.0
dtype: float64
数值属性四分位数:
        0.25  0.50  0.75
points  86.0  88.0  90.0
price   16.0  24.0  40.0
数值属性缺失值:
price     13695
points        0
```

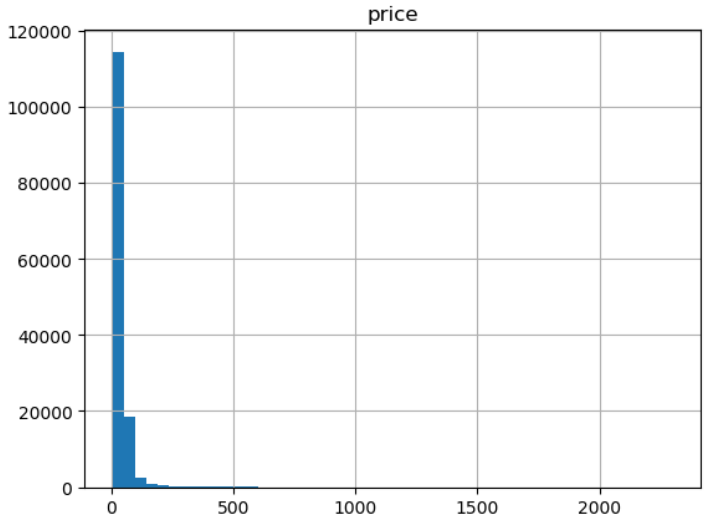
2.数据的可视化

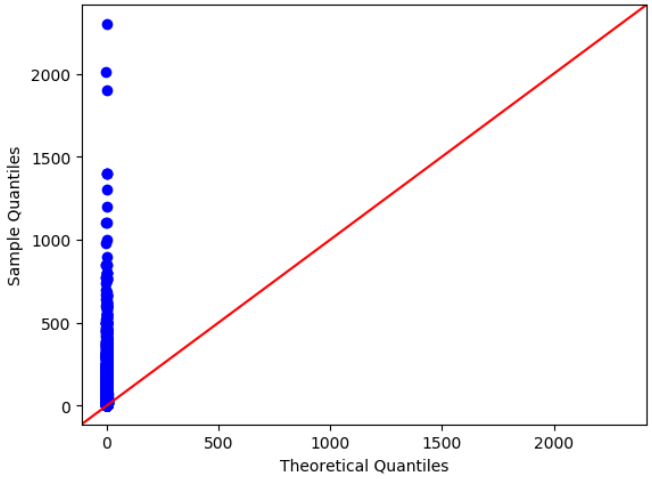
2.1 绘制直方图

导入 `matplotlib.pyplot` 模块，用于数据可视化。导入 `statsmodels.api` 统计分析库，用于用 qq 图检验其分布。使用 `hist(bins = XX)`函数，绘制直方图。使用 `qqplot(df,line='45')`绘制 qq 图检验数据分布是否为正态分布。


points
20000
17500
15000
12500
10000
7500
5000
2500
0
80.0
82.5
85.0
87.5
90.0
92.5
95.0
97.5
100.0

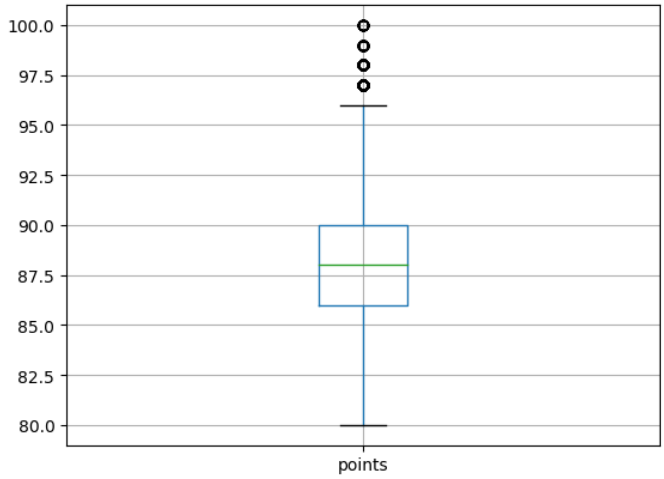

100
80
60
40
20
0
0
20
40
60
80
100
Theoretical Quantiles

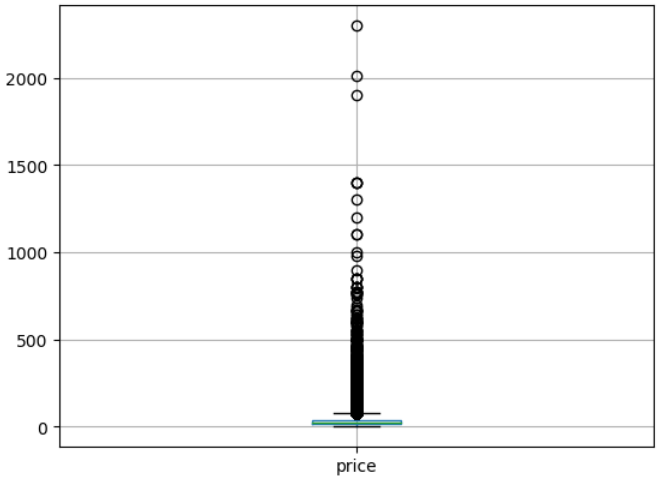

price
120000
100000
80000
60000
40000
20000
0
0
500
1000
1500
2000



2.2 绘制盒图

使用 `boxplot()`函数，绘制盒图，，对离群值进行识别

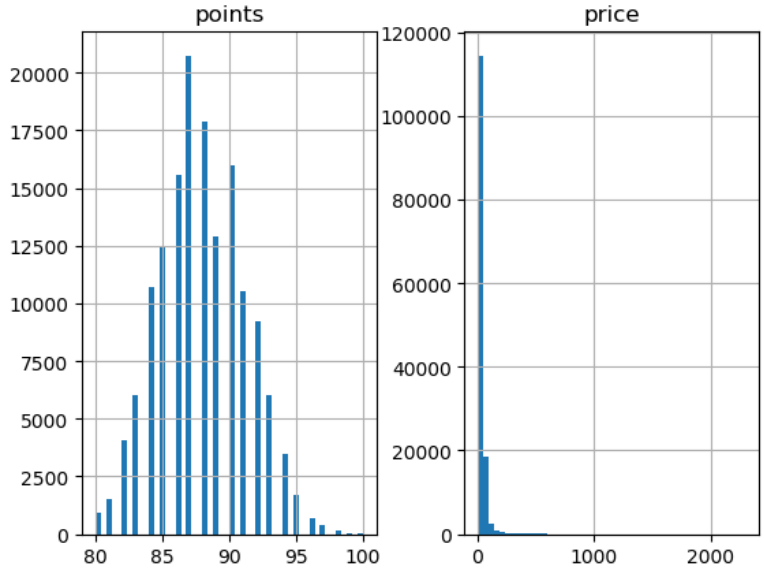




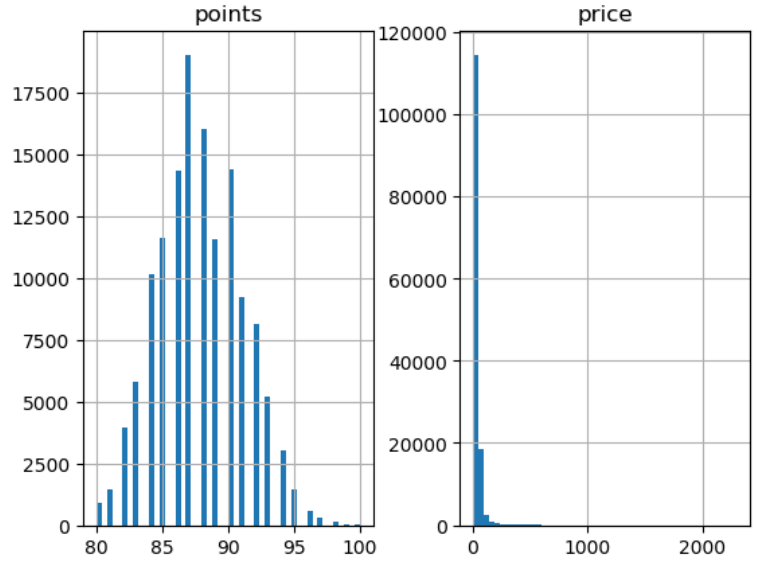
3 数据缺失的处理

3.1 将缺失部分剔除

使用 dropna()函数将缺失值进行剔除，并用直方图可视化地对比新旧数据集。
旧数据集



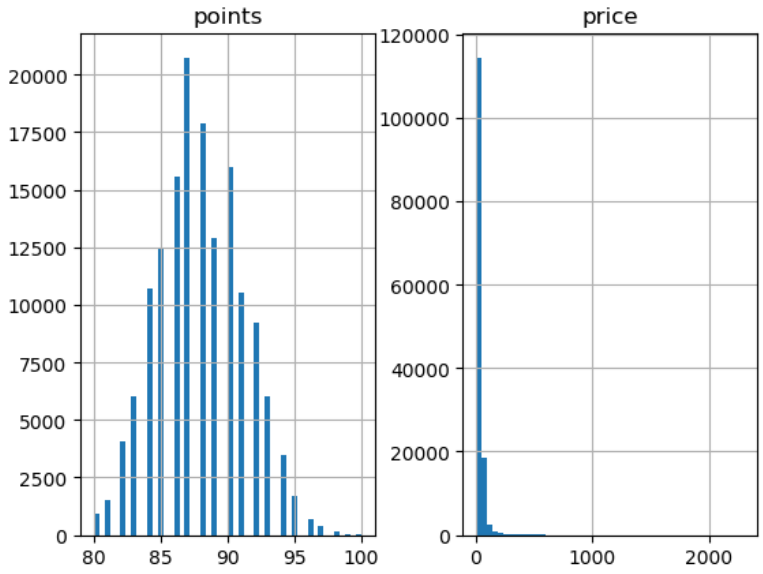
新数据集



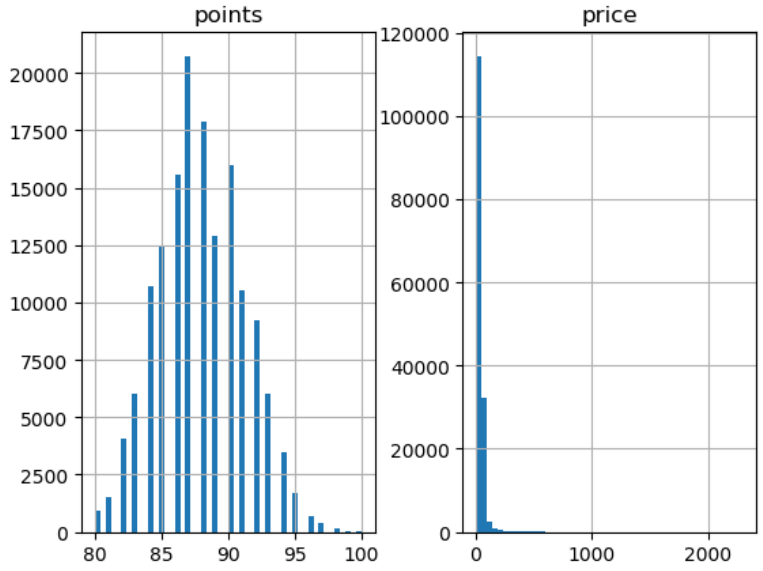
3.2 用最高频率值来填补缺失值

使用 mode()函数获取众数，使用 fillna()函数填补缺失值。并用直方图可视化地对比新旧数据集。

旧数据集



新数据集



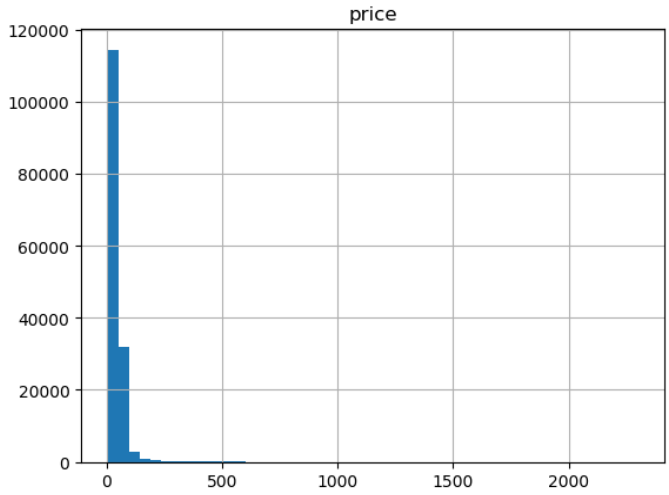
3.3 通过属性的相关关系来填补缺失值

使用 corr()函数获取数值属性相关系数。并输出到 corr.csv 文件，查看相关关系。根据相关关系进行填充。使用 apply()函数，对强相关属性进行填充，并用直方图可视化地对比新旧数据集。

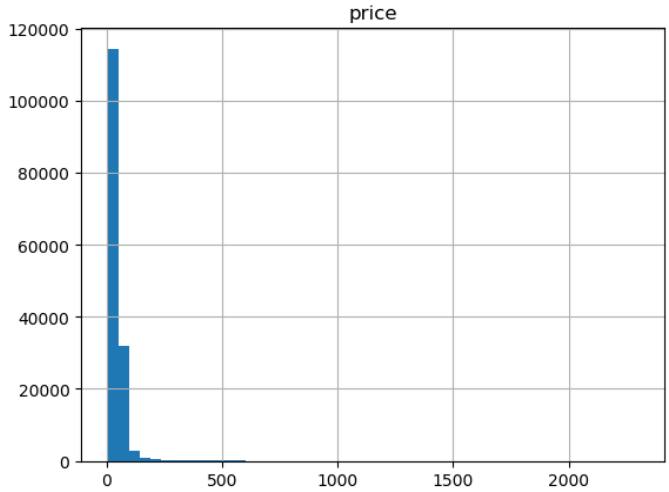
相关性

	points	price
points	1.000000	0.459863
price	0.459863	1.000000

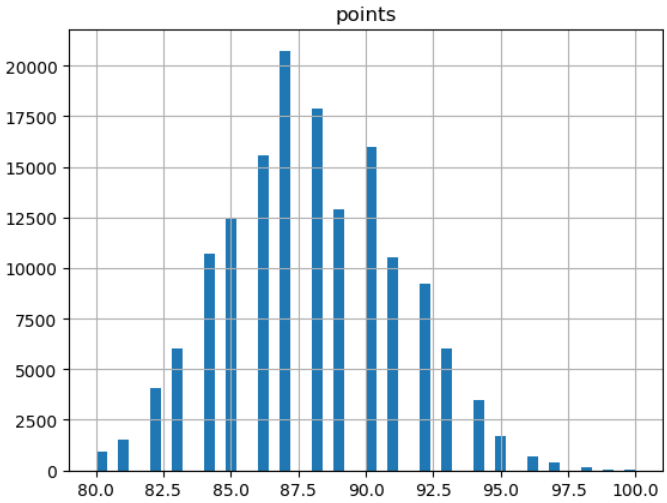
price 填充前



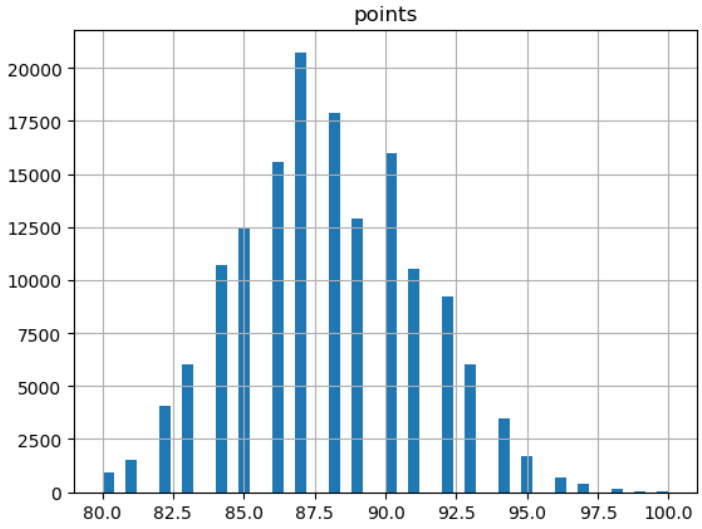
price 填充后



points 填充前



points 填充前


points
20000
17500
15000
12500
10000
7500
5000
2500
0
80.0
82.5
85.0
87.5
90.0
92.5
95.0
97.5
100.0