wine-reviews

May 4, 2020

[2]:

```
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import pandas as pd
import os
```

1 Data preparation

[3]:

```
filepath = r'C:\Users\chd\Desktop\data\wine-reviews'
```

[4]:

```
data = pd.read_csv(os.path.join(filepath,'winemag-data_first150k.csv'))
```

[5]:

```
attribute = data.columns
print(attribute)
```

```
Index(['Unnamed: 0', 'country', 'description', 'designation', 'points',
       'price', 'province', 'region_1', 'region_2', 'variety', 'winery'],
      dtype='object')
```

2 data summary

[6]:

```
nominal = [attribute[i] for i in [1,2,3,6,7,8,9,10]]
print('   :',nominal)
numeric = [attribute[i] for i in [4,5]]
print('   :',numeric)
```

```
  : ['country', 'description', 'designation', 'province', 'region_1',
'region_2', 'variety', 'winery']
  : ['points', 'price']
```

[7]:

```
for a  in numeric:
    n = data[a].shape[0]-1
    split = [int(i*n) for i in [0,0.25,0.5,0.75,1]]
    data[a] = data[a].fillna(data[a].mean())
    num = [data[a].sort_values().iloc[i]  for i in split]
    print(a+'    :', num)
```

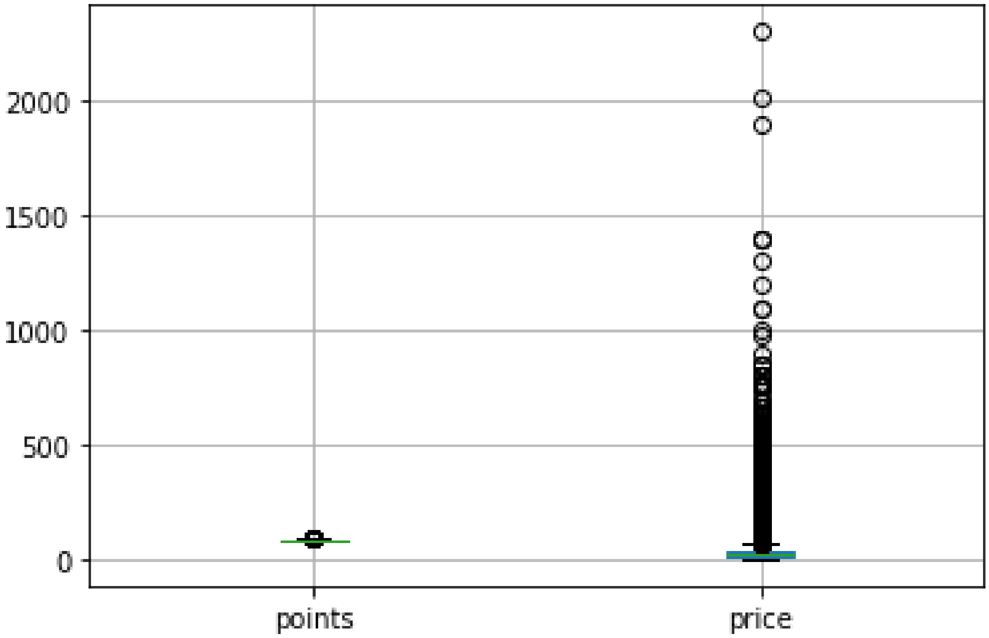
```
points   : [80, 86, 88, 90, 100]
price   : [4.0, 16.0, 26.0, 38.0, 2300.0]
```

[8]:

```
data[numeric].boxplot() #
```

[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29c7fef0248>
```

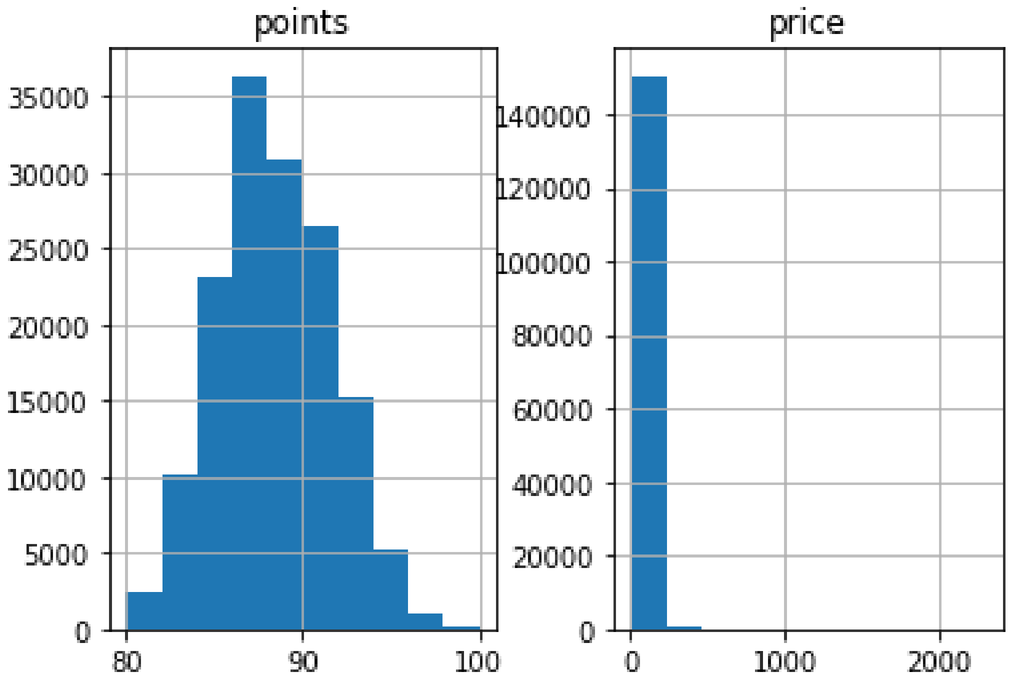


[9]:

```
data[numeric].hist() #
```

[9]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7FED1588>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C00038548>]],
      dtype=object)
```



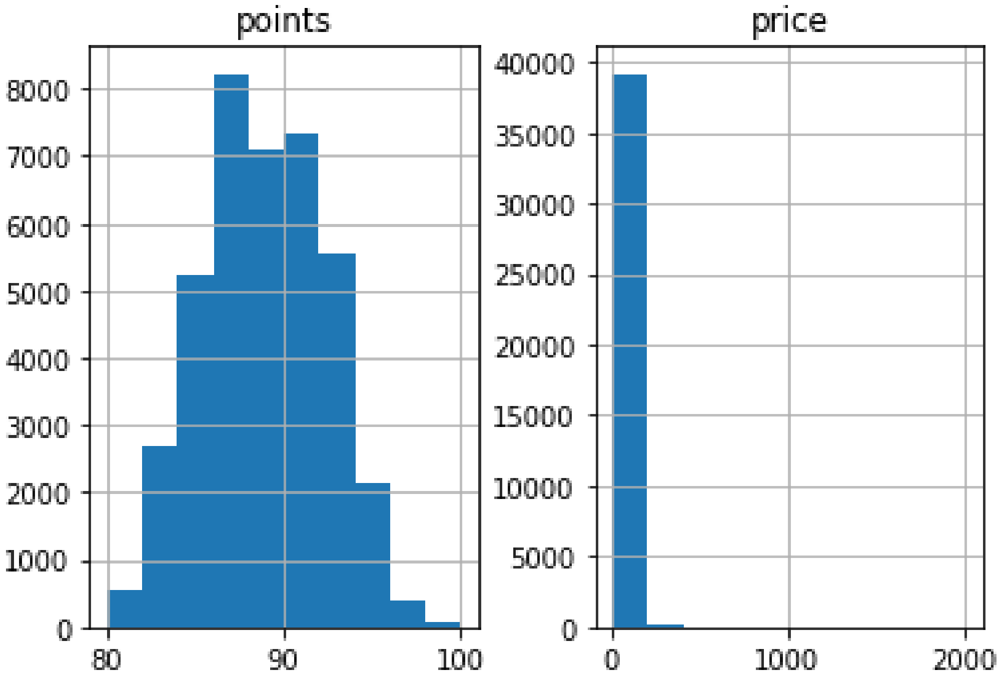
3 Incomplete (Missing) Data

1 Ignore the tuple

```
[10]: data = pd.read_csv(os.path.join(filepath,'winemag-data_first150k.csv'))
      d1 = data.dropna()                              #
```

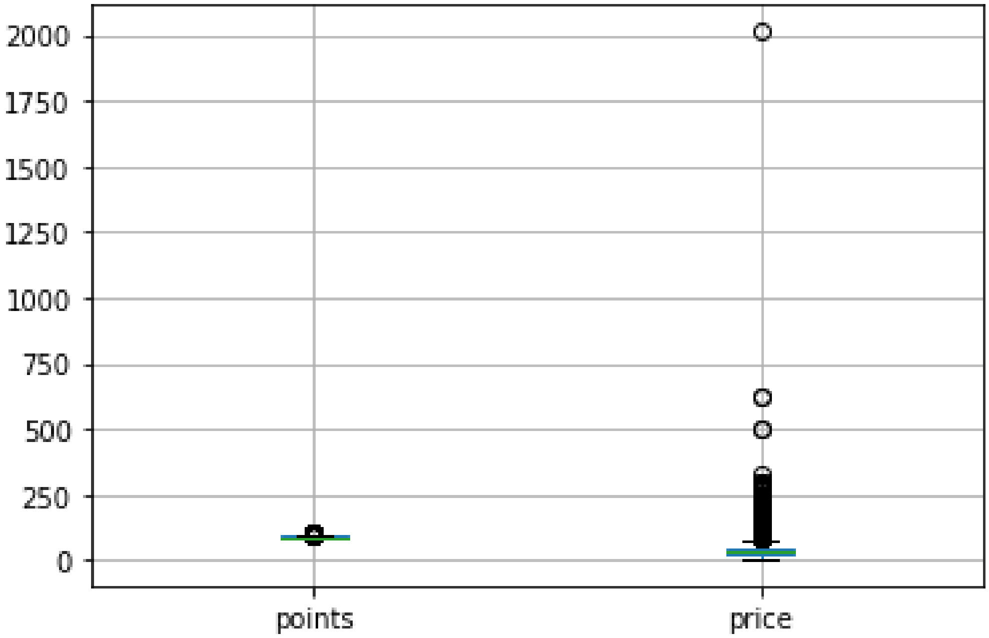
```
[11]: d1[numeric].hist()
```

```
[11]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7FED1048>,
              <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7B832948>]],
            dtype=object)
```



[12]:
```
d1[numeric].boxplot()
```

[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x29c7b832348>
```

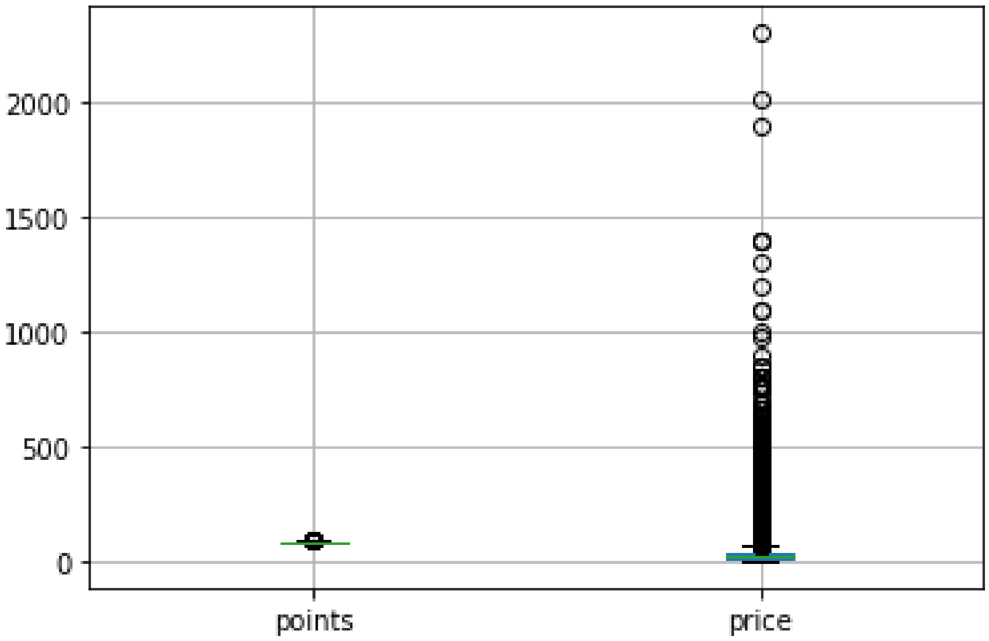


2 Replace with the most frequent data

```
[13]: d2 = pd.read_csv(os.path.join(filepath,'winemag-data_first150k.csv'))
      for i in range(1,11):
          d2[attribute[i]] = d2[attribute[i]].fillna(value= d2[attribute[i]].
       ↪value_counts().index[0]) #
```

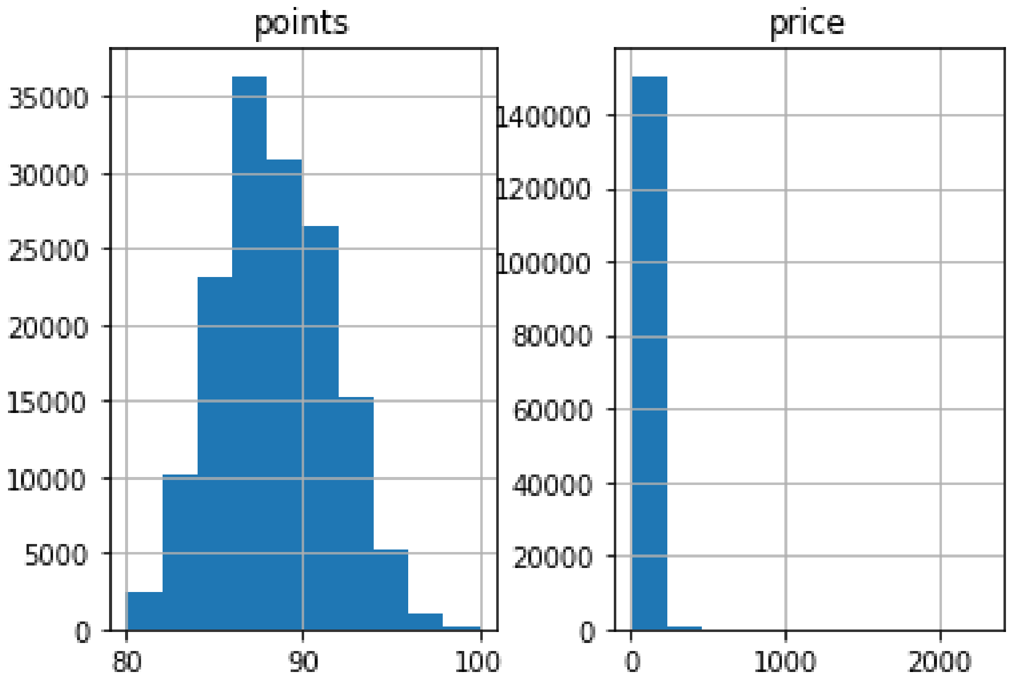
```
[14]: d2[numeric].boxplot()
```

```
[14]: <matplotlib.axes._subplots.AxesSubplot at 0x29c05b19548>
```



```
[15]: d2[numeric].hist()
```

```
[15]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7F827208>,
              <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C01ECA748>]],
            dtype=object)
```



3 Replace with related attribute

There are only two numeric attributes so this substitution does not exist

4 Replace with similar data

[29]:

```
d4 = pd.read_csv(os.path.join(filepath,'winemag-data_first150k.csv')) #
```

123

[30]:

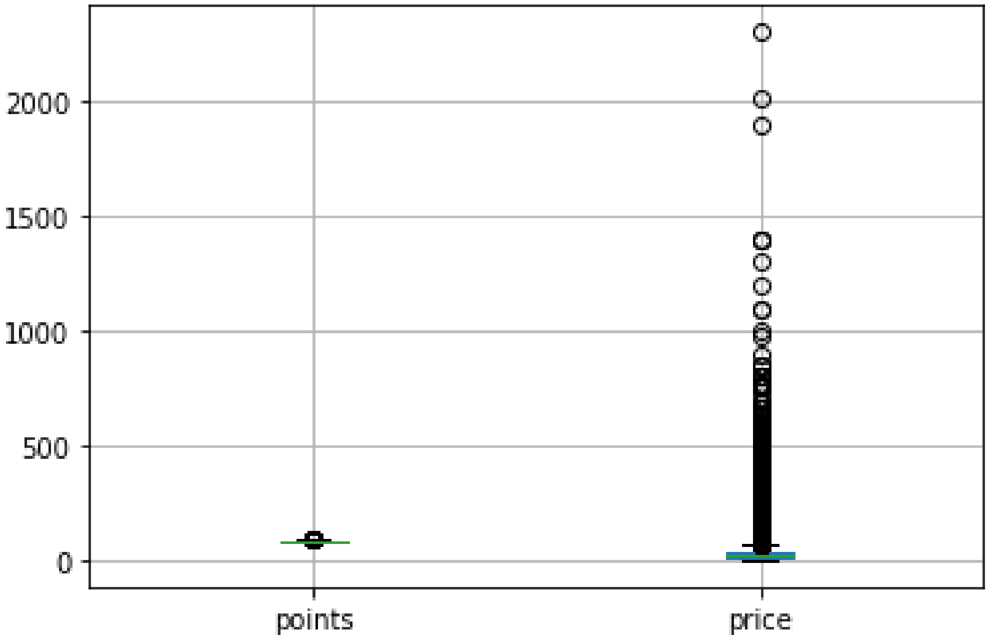
```
def sim(x):
    maxsim = 0
    idx = -1
    for i in range(50):
        tmp = 0
        flag = 1
        for j in list(range(4,11))+[1]:
            if x.iloc[j] == d4.iloc[i,j]:
                tmp+=1
        if tmp>maxsim:
            idx = d4.iloc[i]
            maxsim = tmp
        if maxsim>=3:
            break

    for i in range(1,11):
        if pd.isna(x.iloc[j]):
            x.iloc[i] = idx.iloc[i]
    return x
```

[28]:
```
for i in tqdm.tqdm(range(d4.shape[0])):
    x = d4.iloc[i]
    if x.isnull().any():
        simx = sim(x)
        d4.iloc[i] = simx
```

```
100%|
   | 10/10 [00:00<00:00, 435.50it/s]
```

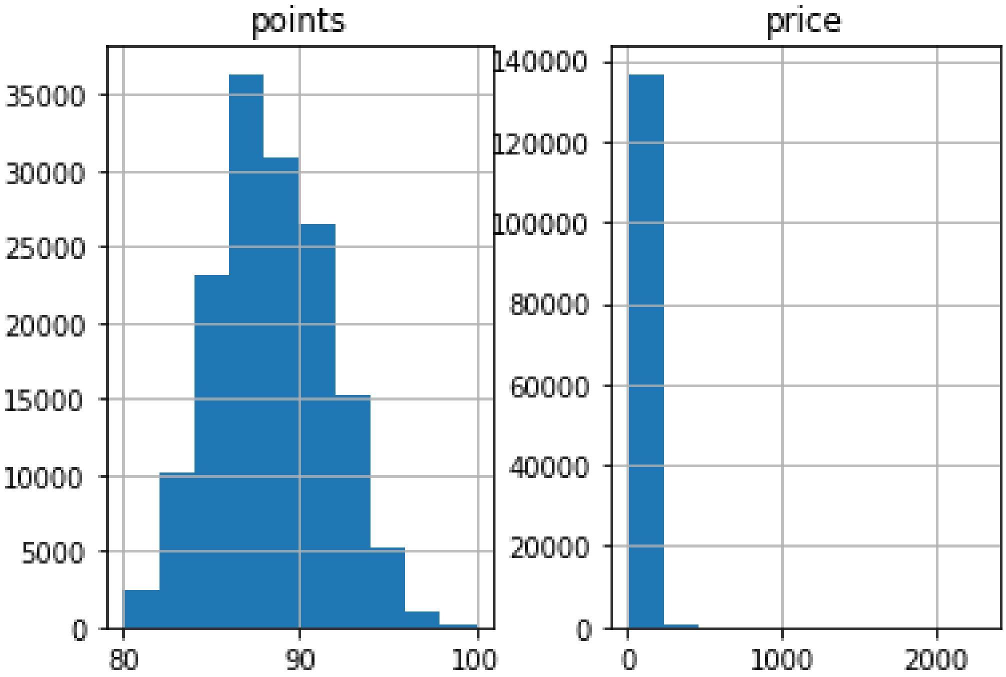
[31]:
```
d4[numeric].boxplot()
```

[31]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x29c785d8c48>
```



[32]:
```
d4[numeric].hist()
```

[32]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C0B904E08>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C0B89CD08>]],
      dtype=object)
```


points
35000
30000
25000
20000
15000
10000
5000
0
80
90
100
price
140000
120000
100000
80000
60000
40000
20000
0
0
1000
2000

[ ]: