# hw1

April 7, 2021

# 1 3220200915

# 2 Wines Review

### 2.0.1 Github https://github.com/lucien1998/DataMingLDY

[35]:

```
import seaborn as sns
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from fancyimpute import KNN
WineReviews_data = pd.read_csv('winemag-data-130k-v2.csv')
```

# 3 3.1

## 3.1 3.1.1

### 3.1.1 3.1.1 1

[36]:

```
# country(     )
print(WineReviews_data['country'].value_counts())
```

```
US                        54504
France                    22093
Italy                     19540
Spain                      6645
Portugal                   5691
Chile                      4472
Argentina                  3800
Austria                    3345
Australia                  2329
Germany                    2165
New Zealand                1419
```

```
South Africa               1401
Israel                      505
Greece                      466
Canada                      257
Hungary                     146
Bulgaria                    141
Romania                     120
Uruguay                     109
Turkey                       90
Slovenia                     87
Georgia                      86
England                      74
Croatia                      73
Mexico                       70
Moldova                      59
Brazil                       52
Lebanon                      35
Morocco                      28
Peru                         16
Ukraine                      14
Serbia                       12
Czech Republic               12
Macedonia                    12
Cyprus                       11
India                         9
Switzerland                   7
Luxembourg                    6
Armenia                       2
Bosnia and Herzegovina        2
China                         1
Egypt                         1
Slovakia                      1
Name: country, dtype: int64
```

[37]:

```
# province(     )
print(WineReviews_data['province'].value_counts())
```

```
California                     36247
Washington                      8639
Bordeaux                        5941
Tuscany                         5897
Oregon                          5373
                                ...
Slovenska Istra                    1
Middle and South Dalmatia          1
Kentucky                           1
Dalmatian Coast                    1
Corinthia                          1
```

```
Name: province, Length: 425, dtype: int64
```

[38]:
```
# region1(      )
print(WineReviews_data['region_1'].value_counts())
```

```
Napa Valley                    4480
Columbia Valley (WA)           4124
Russian River Valley           3091
California                     2629
Paso Robles                    2350
                               …
Offida Rosso                      1
Vino de la Tierra de Zamora       1
Coteaux d'Ancenis                 1
Coteaux du Lyonnais               1
Mazoyeres-Chambertin              1
Name: region_1, Length: 1229, dtype: int64
```

[39]:
```
# region_2(      )
print(WineReviews_data['region_2'].value_counts())
```

```
Central Coast        11065
Sonoma                9028
Columbia Valley       8103
Napa                  6814
Willamette Valley     3423
California Other      2663
Finger Lakes          1777
Sierra Foothills      1462
Napa-Sonoma           1169
Central Valley        1062
Southern Oregon        917
Oregon Other           727
Long Island            680
North Coast            584
Washington Other       534
South Coast            272
New York Other         231
Name: region_2, dtype: int64
```

[40]:
```
# taster_name(      )
print(WineReviews_data['taster_name'].value_counts())
```

```
Roger Voss            25514
Michael Schachner     15134
Kerin O’Keefe         10776
Virginie Boone         9537
Paul Gregutt           9532
```

```
Matt Kettmann          6332
Joe Czerwinski         5147
Sean P. Sullivan       4966
Anna Lee C. Iijima     4415
Jim Gordon             4177
Anne Krebiehl MW       3685
Lauren Buzzeo          1835
Susan Kostrzewa        1085
Mike DeSimone           514
Jeff Jenssen            491
Alexander Peartree      415
Carrie Dykes            139
Fiona Adams              27
Christina Pickard         6
Name: taster_name, dtype: int64
```

[41]:
```python
# taster_twitter_handle(     )
print(WineReviews_data['taster_twitter_handle'].value_counts())
```

```
@vossroger          25514
@wineschach         15134
@kerinokeefe        10776
@vboone              9537
@paulgwine           9532
@mattkettmann        6332
@JoeCz               5147
@wawinereport        4966
@gordone_cellars     4177
@AnneInVino          3685
@laurbuzz            1835
@suskostrzewa        1085
@worldwineguys       1005
@bkfiona               27
@winewchristina         6
Name: taster_twitter_handle, dtype: int64
```

[42]:
```python
# variety(     )
print(WineReviews_data['variety'].value_counts())
```

```
Pinot Noir                  13272
Chardonnay                  11753
Cabernet Sauvignon           9472
Red Blend                    8946
Bordeaux-style Red Blend     6915
                            …
Gros Plant                      1
Chardonnay-Pinot Gris           1
Colorino                        1
```

```
Caprettone                        1
Macabeo-Moscatel                  1
Name: variety, Length: 707, dtype: int64
```

[43]:
```
# country(     )
print(WineReviews_data['winery'].value_counts())
```

```
Wines & Winemakers          222
Testarossa                  218
DFJ Vinhos                  215
Williams Selyem             211
Louis Latour                199
                           …
Seaside                       1
Stellenbosch Vineyards        1
Château le Reysse             1
Marston Family                1
Carneros Hills                1
Name: winery, Length: 16757, dtype: int64
```

### 3.1.2 3.1.1 2 points price

[44]:
```
#
WineReviews_data.describe()
```

[44]:
```
          Unnamed: 0         points          price
count  129971.000000  129971.000000  120975.000000
mean    64985.000000      88.447138      35.363389
std     37519.540256       3.039730      41.022218
min         0.000000      80.000000       4.000000
25%     32492.500000      86.000000      17.000000
50%     64985.000000      88.000000      25.000000
75%     97477.500000      91.000000      42.000000
max    129970.000000     100.000000    3300.000000
```

[45]:
```
#
WineReviews_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129971 entries, 0 to 129970
Data columns (total 14 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   Unnamed: 0             129971 non-null  int64
 1   country                129908 non-null  object
 2   description            129971 non-null  object
 3   designation            92506 non-null   object
 4   points                 129971 non-null  int64
 5   price                  120975 non-null  float64
 6   province               129908 non-null  object
 7   region_1               108724 non-null  object
 8   region_2               50511 non-null   object
 9   taster_name            103727 non-null  object
 10  taster_twitter_handle  98758 non-null   object
 11  title                  129971 non-null  object
 12  variety                129970 non-null  object
 13  winery                 129971 non-null  object
dtypes: float64(1), int64(2), object(11)
memory usage: 13.9+ MB
```

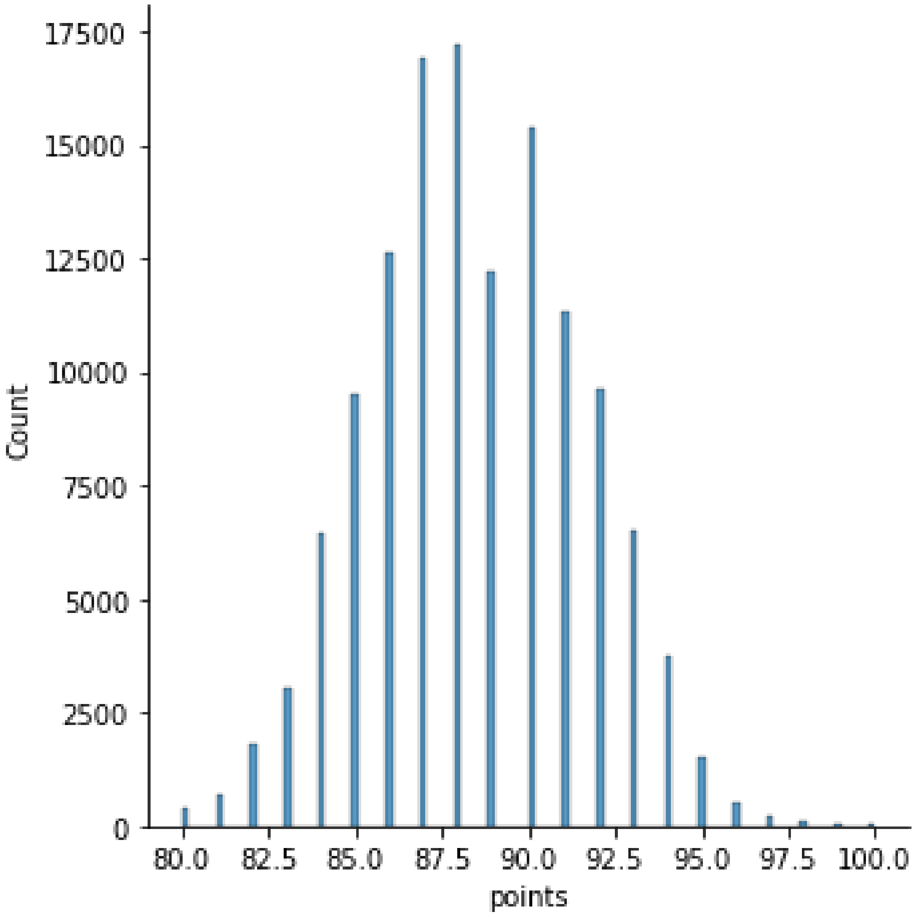
129971 country 63 description 0 designation 37465 points 0 price 8996 province 63 re

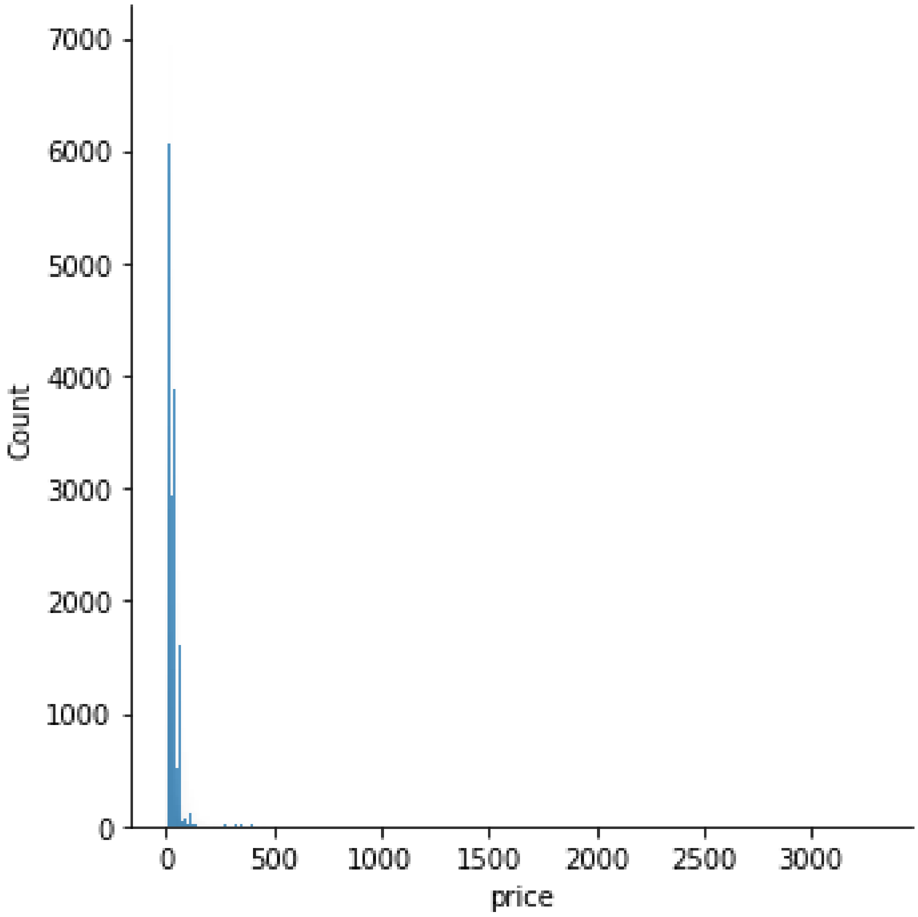
## 3.2 3.1.2

### 3.2.1 3.1.2 1

[46]:

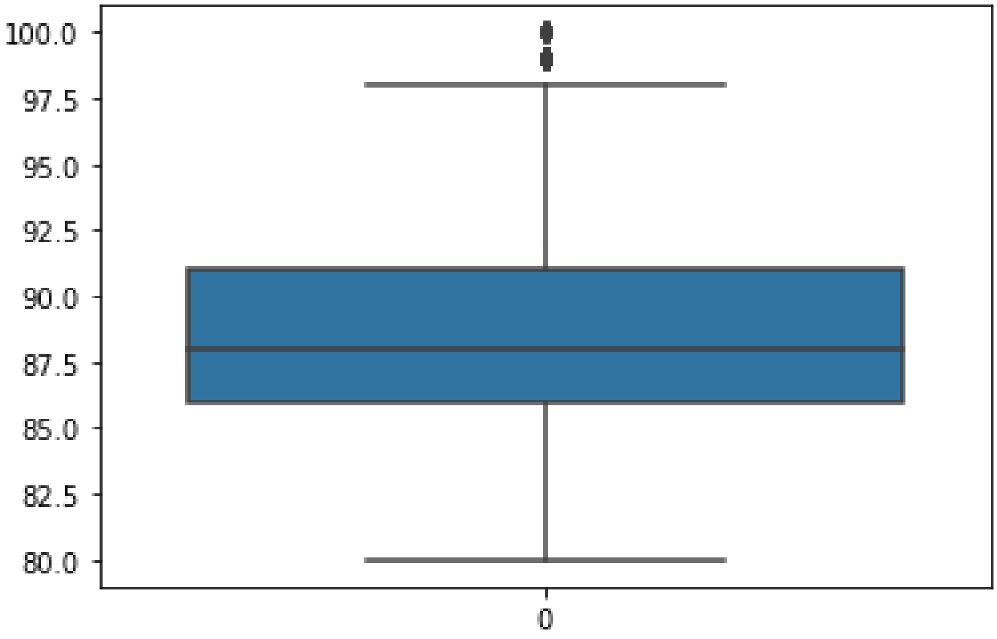
```
sns.displot(WineReviews_data['points'])
plt.show()
sns.displot(WineReviews_data['price'])
plt.show()
```

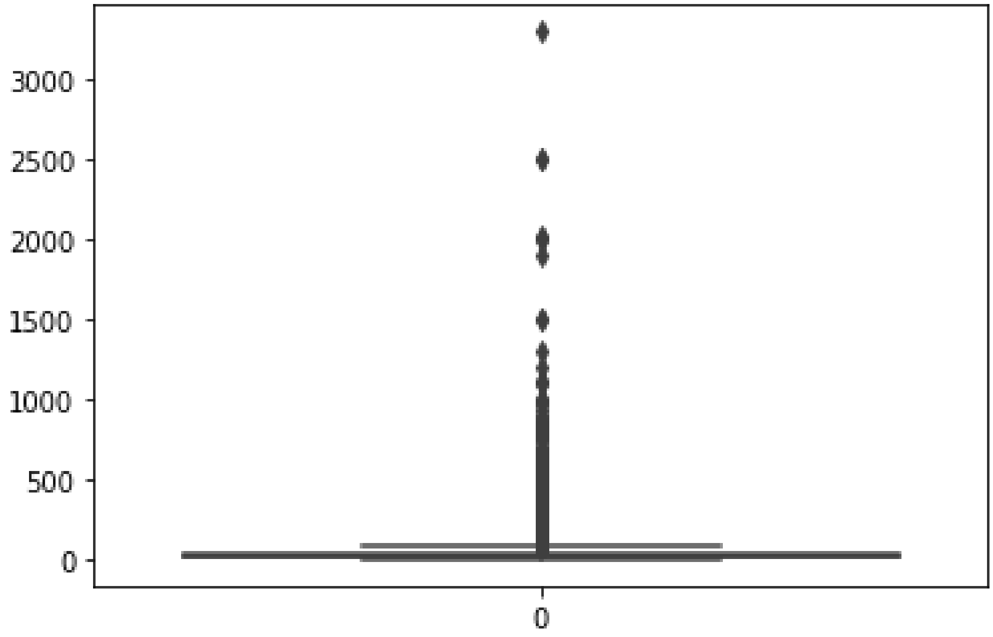

17500
15000
12500
10000
7500
5000
2500
0
Count
80.0
82.5
85.0
87.5
90.0
92.5
95.0
97.5
100.0
points



### 3.2.2 3.1.2 2

[47]:
```
sns.boxplot(data=WineReviews_data['points'])
plt.show()
sns.boxplot(data=WineReviews_data['price'])
plt.show()
```


100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
0

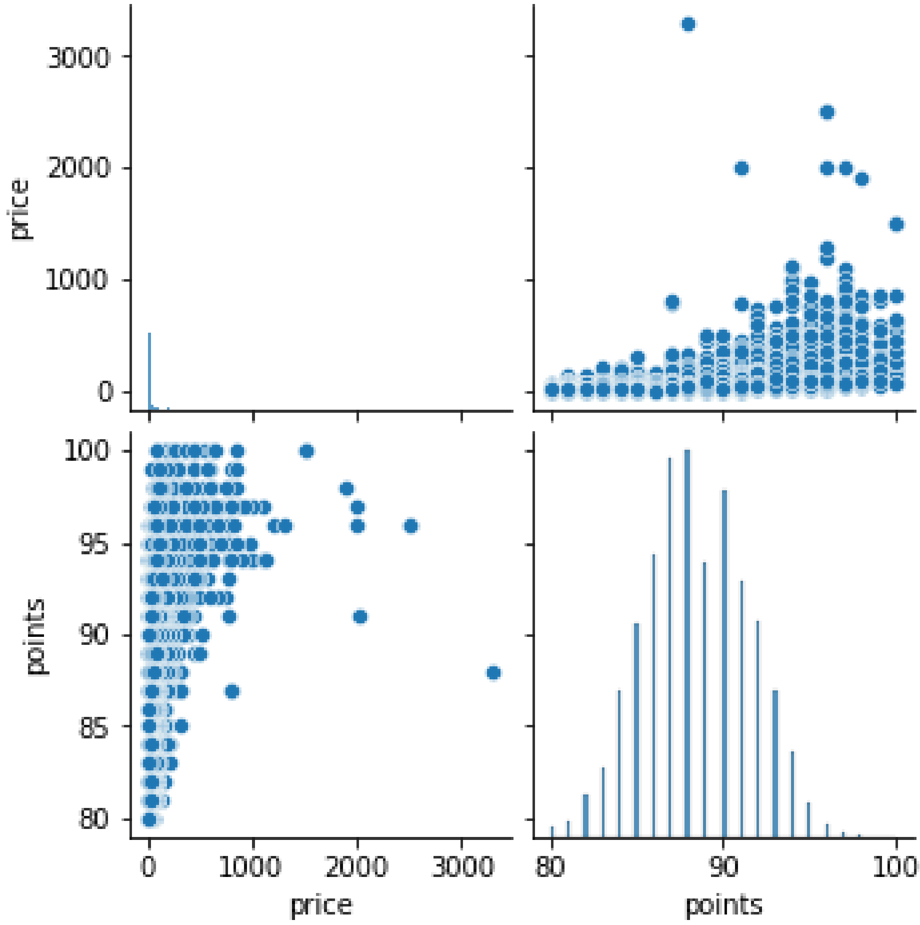

3000
2500
2000
1500
1000
500
0
0

# 4 3.2

## 4.1 3.2.1

```
[48]: print(" 1  \n")
sns.pairplot(WineReviews_data, vars=["price","points"])
plt.show()
print(WineReviews_data['price'])
print("-----------------------------------------------------------------------\n")
print(" 2  \n")
WineReviews_data_after = WineReviews_data.dropna()
sns.pairplot(WineReviews_data_after, vars=["price","points"])
plt.show()
print(WineReviews_data_after['price'])
```

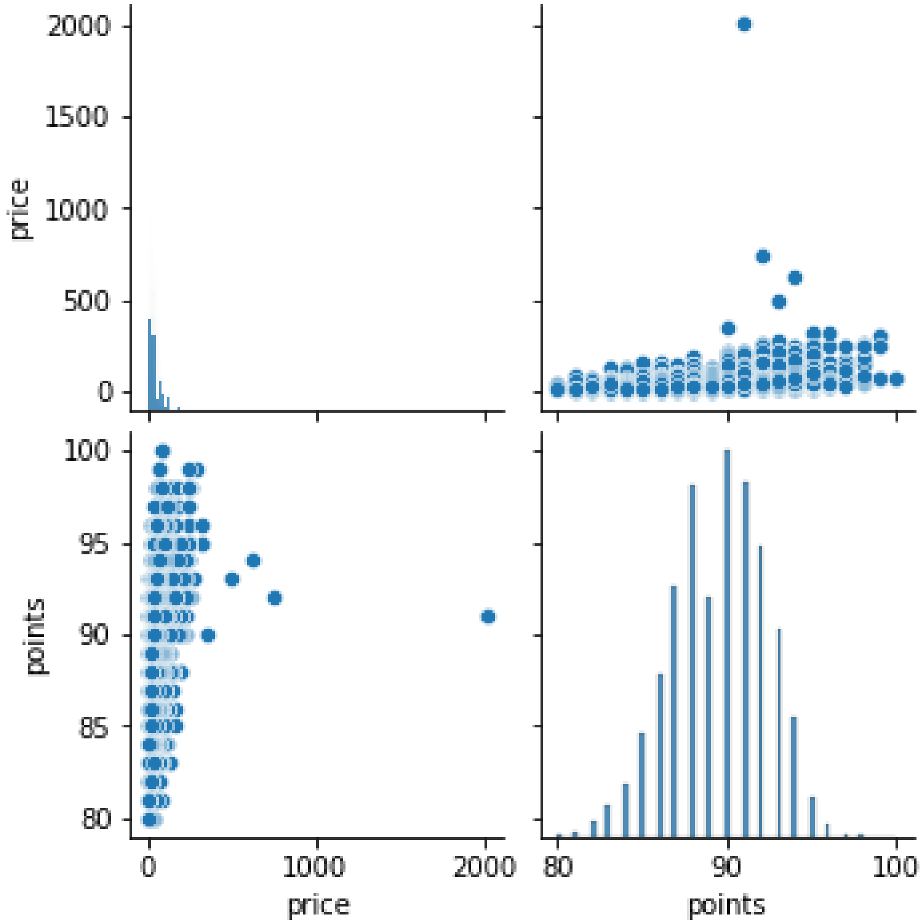
```
1
```



```
0           NaN
1          15.0
```

```
2         14.0
3         13.0
4         65.0
          …
129966    28.0
129967    75.0
129968    30.0
129969    32.0
129970    21.0
Name: price, Length: 129971, dtype: float64
-----------------------------------------------------------------------
```

```
 2
```



```
4          65.0
10         19.0
23         22.0
```

```
25         69.0
35         50.0
          …
129919    105.0
129926     41.0
129945     20.0
129949     35.0
129950     35.0
Name: price, Length: 22387, dtype: float64
```

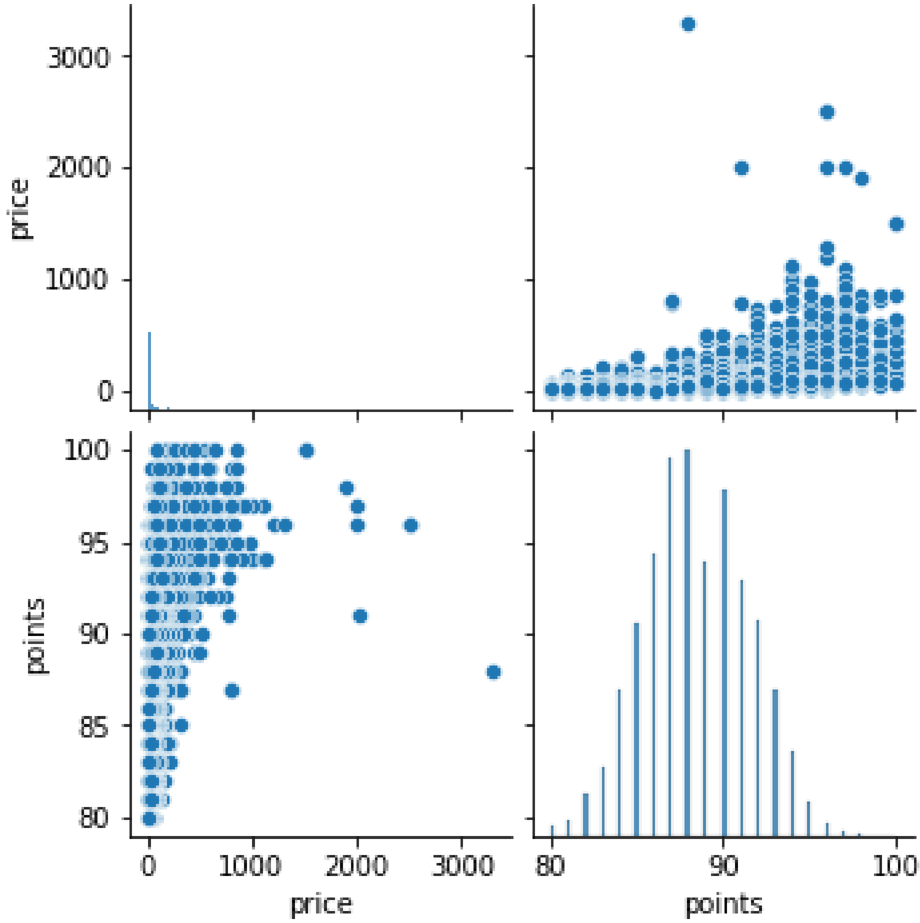
### 4.2 3.2.2

[49]:

```
print(" 1  \n")
WineReviews_data2 = WineReviews_data.copy(deep=True)
sns.pairplot(WineReviews_data2, vars=["price","points"])
plt.show()
print(WineReviews_data2['price'])
print("\n===================\n")
WineReviews_data2.info()
print("-----------------------------------------------------------------------\n")
print(" 2  \n")
WineReviews_data2['price'].fillna(WineReviews_data2['price'].mode().
 ↪iloc[0],inplace=True)
WineReviews_data2['country'].fillna(WineReviews_data2['country'].mode().
 ↪iloc[0],inplace=True)
WineReviews_data2['description'].fillna(WineReviews_data2['description'].mode().
 ↪iloc[0],inplace=True)
WineReviews_data2['designation'].fillna(WineReviews_data2['designation'].mode().
 ↪iloc[0],inplace=True)
WineReviews_data2['province'].fillna(WineReviews_data2['province'].mode().
 ↪iloc[0],inplace=True)
WineReviews_data2['country'].fillna(WineReviews_data2['country'].mode().
 ↪iloc[0],inplace=True)
WineReviews_data2['region_1'].fillna(WineReviews_data2['region_1'].mode().
 ↪iloc[0],inplace=True)
WineReviews_data2['region_2'].fillna(WineReviews_data2['region_2'].mode().
 ↪iloc[0],inplace=True)
WineReviews_data2['taster_name'].fillna(WineReviews_data2['taster_name'].mode().
 ↪iloc[0],inplace=True)
WineReviews_data2['taster_twitter_handle'].
 ↪fillna(WineReviews_data2['taster_twitter_handle'].mode().
 ↪iloc[0],inplace=True)
WineReviews_data2['variety'].fillna(WineReviews_data2['variety'].mode().
 ↪iloc[0],inplace=True)
sns.pairplot(WineReviews_data2, vars=["price","points"])
plt.show()
```

```
print(WineReviews_data2['price'])
print("\n===================\n")
WineReviews_data2.info()
print("\n        price    price points         \n             ")
```

```
1
```



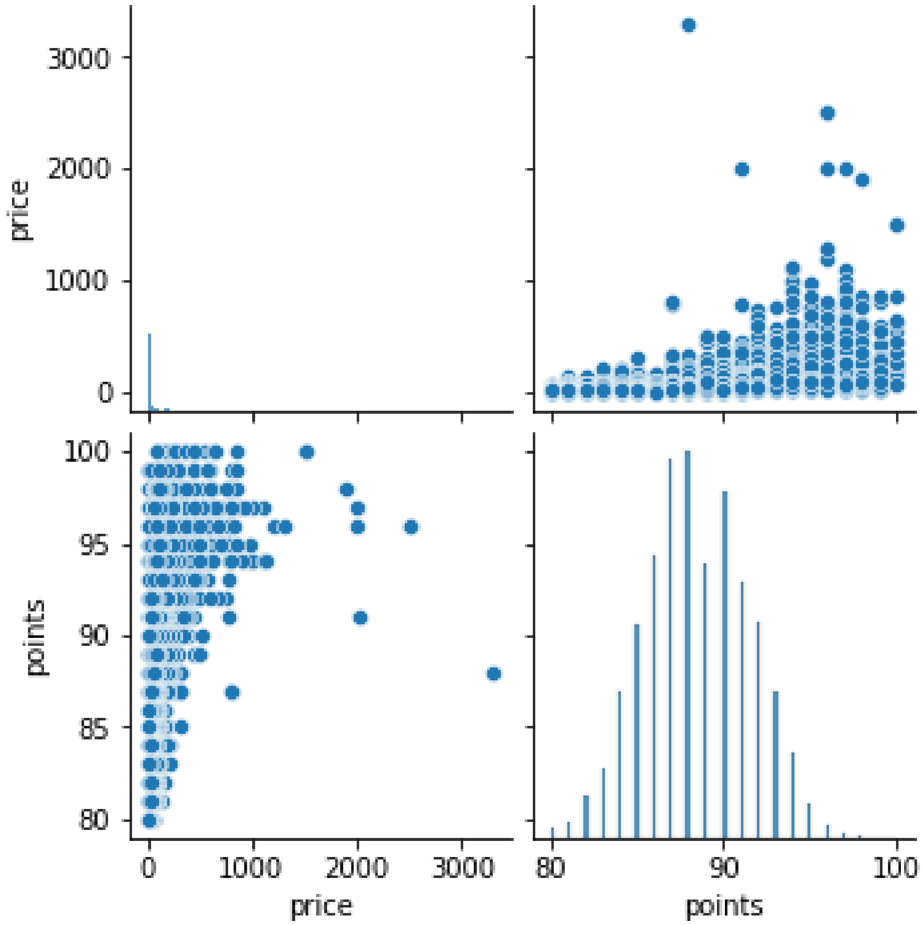
```
0            NaN
1           15.0
2           14.0
3           13.0
4           65.0
            …
129966      28.0
129967      75.0
129968      30.0
129969      32.0
```

```
129970    21.0
Name: price, Length: 129971, dtype: float64

===================

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129971 entries, 0 to 129970
Data columns (total 14 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   Unnamed: 0             129971 non-null  int64
 1   country                129908 non-null  object
 2   description            129971 non-null  object
 3   designation            92506 non-null   object
 4   points                 129971 non-null  int64
 5   price                  120975 non-null  float64
 6   province               129908 non-null  object
 7   region_1               108724 non-null  object
 8   region_2               50511 non-null   object
 9   taster_name            103727 non-null  object
 10  taster_twitter_handle  98758 non-null   object
 11  title                  129971 non-null  object
 12  variety                129970 non-null  object
 13  winery                 129971 non-null  object
dtypes: float64(1), int64(2), object(11)
memory usage: 13.9+ MB
-----------------------------------------------------------------------

 2
```



```
0           20.0
1           15.0
2           14.0
3           13.0
4           65.0
            …
129966    28.0
129967    75.0
129968    30.0
129969    32.0
129970    21.0
Name: price, Length: 129971, dtype: float64

===================

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129971 entries, 0 to 129970
Data columns (total 14 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   Unnamed: 0             129971 non-null  int64
 1   country                129971 non-null  object
 2   description            129971 non-null  object
 3   designation            129971 non-null  object
 4   points                 129971 non-null  int64
 5   price                  129971 non-null  float64
 6   province               129971 non-null  object
 7   region_1               129971 non-null  object
 8   region_2               129971 non-null  object
 9   taster_name            129971 non-null  object
 10  taster_twitter_handle  129971 non-null  object
 11  title                  129971 non-null  object
 12  variety                129971 non-null  object
 13  winery                 129971 non-null  object
dtypes: float64(1), int64(2), object(11)
memory usage: 13.9+ MB

       price   price points
```

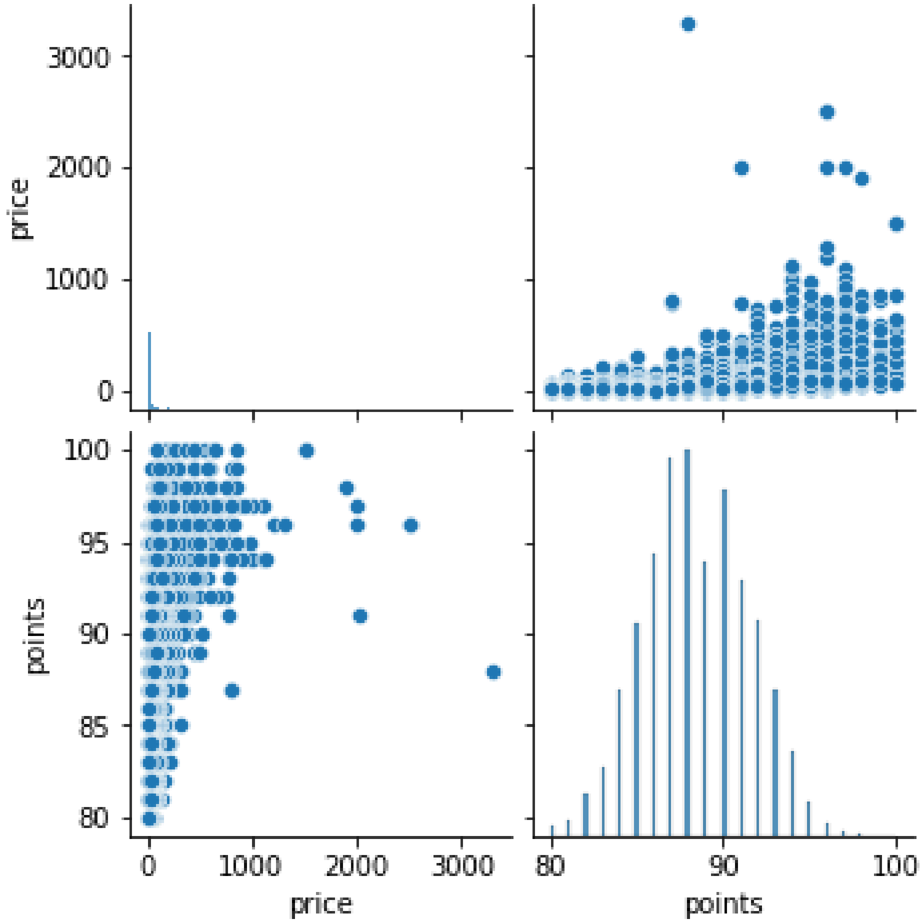
### 4.3 3.2.3

[50]:

```python
print(" 1  \n")
WineReviews_data3 = WineReviews_data.copy(deep=True)
sns.pairplot(WineReviews_data3, vars=["price","points"])
plt.show()
print(WineReviews_data3['price'])
print("-----------------------------------------------------------------------\n")
print(" 2  \n")
def set_missing_prices(df):
    #
    price_df = df[['price', 'points']]
    known_price = price_df[price_df.price.notnull()].iloc[:,:].values
    unknown_price = price_df[price_df.price.isnull()].iloc[:,:].values
    y = known_price[:, 0]  # y price
    x = known_price[:, 1:]  # x
    rfr = RandomForestRegressor(random_state=0, n_estimators=2000, n_jobs=-1)
    #
    rfr.fit(x, y)
    #
    predictedprices = rfr.predict(unknown_price[:, 1:])
    #
    df.loc[(df.price.isnull()), 'price'] = predictedprices
    return df
```

```
WineReviews_data3 = set_missing_prices(WineReviews_data3)
sns.pairplot(WineReviews_data3, vars=["price","points"])
plt.show()
print(WineReviews_data3['price'])
```

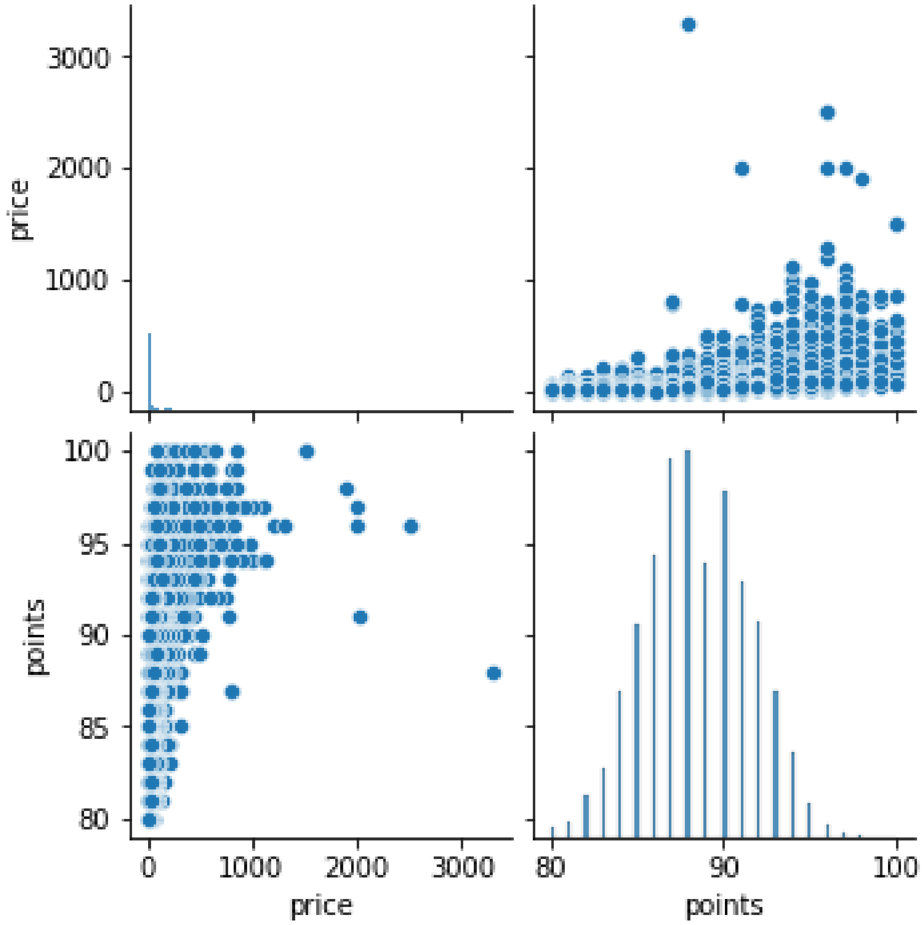
```
1
```



```
0            NaN
1           15.0
2           14.0
3           13.0
4           65.0
            …
129966      28.0
129967      75.0
129968      30.0
129969      32.0
```

```
129970    21.0
Name: price, Length: 129971, dtype: float64
-----------------------------------------------------------------------
```

```
 2
```



```
0         24.903054
1         15.000000
2         14.000000
3         13.000000
4         65.000000
                …
129966    28.000000
129967    75.000000
129968    30.000000
129969    32.000000
129970    21.000000
```

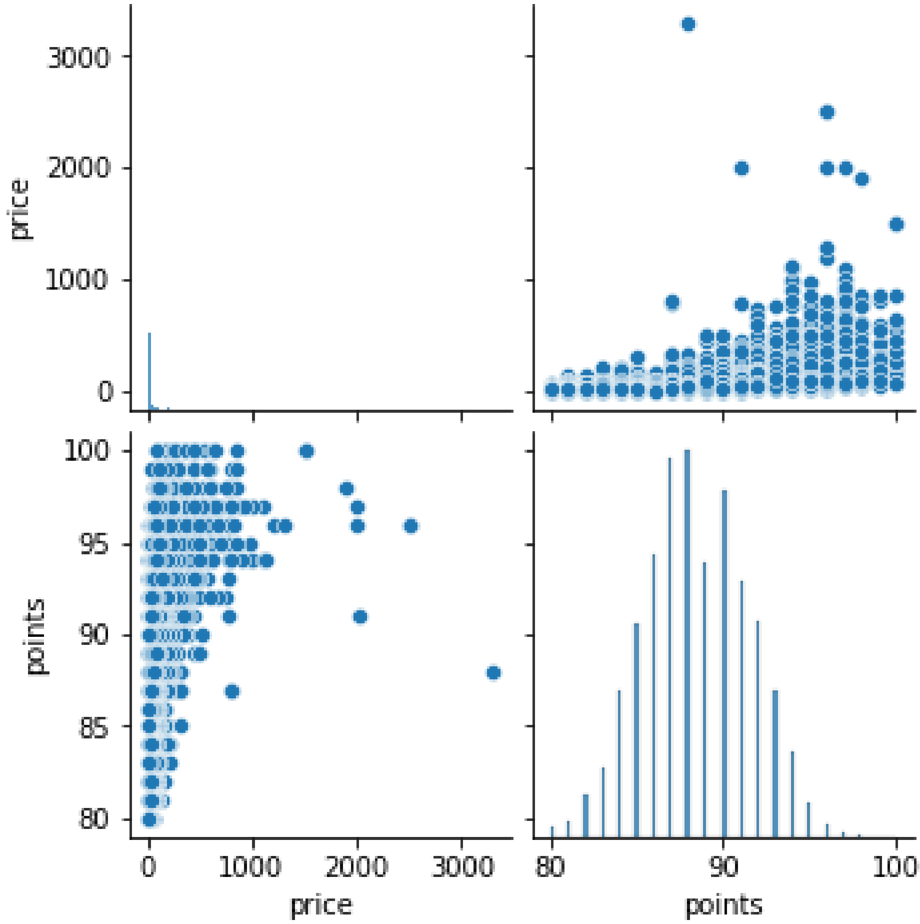
```
Name: price, Length: 129971, dtype: float64
```

## 4.4 3.2.4

[51]:

```
print(" 1  \n")
WineReviews_data4 = WineReviews_data.copy(deep=True)
sns.pairplot(WineReviews_data4, vars=["price","points"])
plt.show()
print(WineReviews_data4['price'])
print("------------------------------------------------------------------------\n")
print(" 2  \n")
new_data = WineReviews_data4[['price', 'points']][:10000]
fill_knn = KNN(k=3).fit_transform(new_data)
print(fill_knn)
```

```
1
```



```
0          NaN
1         15.0
2         14.0
3         13.0
4         65.0
          …
129966    28.0
129967    75.0
129968    30.0
129969    32.0
129970    21.0
Name: price, Length: 129971, dtype: float64
------------------------------------------------------------------------
```

```
 2
```

```
Imputing row 1/10000 with 1 missing, elapsed time: 13.797
Imputing row 101/10000 with 0 missing, elapsed time: 13.799
Imputing row 201/10000 with 1 missing, elapsed time: 13.801
Imputing row 301/10000 with 0 missing, elapsed time: 13.803
Imputing row 401/10000 with 0 missing, elapsed time: 13.804
Imputing row 501/10000 with 0 missing, elapsed time: 13.806
Imputing row 601/10000 with 0 missing, elapsed time: 13.807
Imputing row 701/10000 with 0 missing, elapsed time: 13.808
Imputing row 801/10000 with 0 missing, elapsed time: 13.808
Imputing row 901/10000 with 0 missing, elapsed time: 13.810
Imputing row 1001/10000 with 0 missing, elapsed time: 13.811
Imputing row 1101/10000 with 0 missing, elapsed time: 13.812
Imputing row 1201/10000 with 0 missing, elapsed time: 13.813
Imputing row 1301/10000 with 0 missing, elapsed time: 13.815
Imputing row 1401/10000 with 0 missing, elapsed time: 13.816
Imputing row 1501/10000 with 0 missing, elapsed time: 13.817
Imputing row 1601/10000 with 0 missing, elapsed time: 13.818
Imputing row 1701/10000 with 0 missing, elapsed time: 13.819
Imputing row 1801/10000 with 0 missing, elapsed time: 13.820
Imputing row 1901/10000 with 0 missing, elapsed time: 13.822
Imputing row 2001/10000 with 0 missing, elapsed time: 13.822
Imputing row 2101/10000 with 0 missing, elapsed time: 13.824
Imputing row 2201/10000 with 0 missing, elapsed time: 13.825
Imputing row 2301/10000 with 0 missing, elapsed time: 13.826
Imputing row 2401/10000 with 0 missing, elapsed time: 13.827
Imputing row 2501/10000 with 0 missing, elapsed time: 13.828
Imputing row 2601/10000 with 0 missing, elapsed time: 13.829
Imputing row 2701/10000 with 0 missing, elapsed time: 13.830
Imputing row 2801/10000 with 0 missing, elapsed time: 13.832
Imputing row 2901/10000 with 0 missing, elapsed time: 13.834
Imputing row 3001/10000 with 0 missing, elapsed time: 13.835
Imputing row 3101/10000 with 0 missing, elapsed time: 13.836
Imputing row 3201/10000 with 0 missing, elapsed time: 13.838
Imputing row 3301/10000 with 0 missing, elapsed time: 13.839
Imputing row 3401/10000 with 0 missing, elapsed time: 13.840
Imputing row 3501/10000 with 0 missing, elapsed time: 13.841
Imputing row 3601/10000 with 0 missing, elapsed time: 13.842
Imputing row 3701/10000 with 0 missing, elapsed time: 13.843
Imputing row 3801/10000 with 0 missing, elapsed time: 13.844
Imputing row 3901/10000 with 0 missing, elapsed time: 13.846
Imputing row 4001/10000 with 0 missing, elapsed time: 13.848
Imputing row 4101/10000 with 0 missing, elapsed time: 13.849
Imputing row 4201/10000 with 0 missing, elapsed time: 13.850
Imputing row 4301/10000 with 0 missing, elapsed time: 13.851
Imputing row 4401/10000 with 0 missing, elapsed time: 13.852
Imputing row 4501/10000 with 1 missing, elapsed time: 13.853
Imputing row 4601/10000 with 0 missing, elapsed time: 13.854
Imputing row 4701/10000 with 0 missing, elapsed time: 13.856
Imputing row 4801/10000 with 0 missing, elapsed time: 13.857
Imputing row 4901/10000 with 0 missing, elapsed time: 13.858
Imputing row 5001/10000 with 0 missing, elapsed time: 13.859
Imputing row 5101/10000 with 0 missing, elapsed time: 13.860
Imputing row 5201/10000 with 0 missing, elapsed time: 13.862
Imputing row 5301/10000 with 0 missing, elapsed time: 13.863
Imputing row 5401/10000 with 0 missing, elapsed time: 13.864
Imputing row 5501/10000 with 0 missing, elapsed time: 13.866
Imputing row 5601/10000 with 0 missing, elapsed time: 13.867
Imputing row 5701/10000 with 1 missing, elapsed time: 13.868
Imputing row 5801/10000 with 0 missing, elapsed time: 13.869
Imputing row 5901/10000 with 0 missing, elapsed time: 13.870
Imputing row 6001/10000 with 0 missing, elapsed time: 13.871
Imputing row 6101/10000 with 0 missing, elapsed time: 13.872
Imputing row 6201/10000 with 0 missing, elapsed time: 13.873
Imputing row 6301/10000 with 0 missing, elapsed time: 13.874
Imputing row 6401/10000 with 0 missing, elapsed time: 13.875
Imputing row 6501/10000 with 0 missing, elapsed time: 13.877
Imputing row 6601/10000 with 1 missing, elapsed time: 13.879
Imputing row 6701/10000 with 0 missing, elapsed time: 13.880
Imputing row 6801/10000 with 0 missing, elapsed time: 13.882
Imputing row 6901/10000 with 0 missing, elapsed time: 13.883
Imputing row 7001/10000 with 0 missing, elapsed time: 13.884
Imputing row 7101/10000 with 0 missing, elapsed time: 13.885
Imputing row 7201/10000 with 0 missing, elapsed time: 13.886
Imputing row 7301/10000 with 0 missing, elapsed time: 13.887
Imputing row 7401/10000 with 0 missing, elapsed time: 13.888
Imputing row 7501/10000 with 0 missing, elapsed time: 13.889
Imputing row 7601/10000 with 0 missing, elapsed time: 13.890
Imputing row 7701/10000 with 0 missing, elapsed time: 13.891
Imputing row 7801/10000 with 0 missing, elapsed time: 13.892
Imputing row 7901/10000 with 0 missing, elapsed time: 13.893
Imputing row 8001/10000 with 0 missing, elapsed time: 13.894
Imputing row 8101/10000 with 0 missing, elapsed time: 13.895
Imputing row 8201/10000 with 0 missing, elapsed time: 13.896
Imputing row 8301/10000 with 0 missing, elapsed time: 13.897
Imputing row 8401/10000 with 0 missing, elapsed time: 13.899
Imputing row 8501/10000 with 0 missing, elapsed time: 13.900
Imputing row 8601/10000 with 1 missing, elapsed time: 13.901
Imputing row 8701/10000 with 0 missing, elapsed time: 13.903
Imputing row 8801/10000 with 1 missing, elapsed time: 13.904
Imputing row 8901/10000 with 1 missing, elapsed time: 13.905
Imputing row 9001/10000 with 0 missing, elapsed time: 13.907
Imputing row 9101/10000 with 0 missing, elapsed time: 13.908
Imputing row 9201/10000 with 0 missing, elapsed time: 13.909
Imputing row 9301/10000 with 0 missing, elapsed time: 13.910
Imputing row 9401/10000 with 0 missing, elapsed time: 13.911
Imputing row 9501/10000 with 0 missing, elapsed time: 13.912
Imputing row 9601/10000 with 0 missing, elapsed time: 13.913
Imputing row 9701/10000 with 0 missing, elapsed time: 13.914
Imputing row 9801/10000 with 0 missing, elapsed time: 13.915
Imputing row 9901/10000 with 0 missing, elapsed time: 13.916
[[20.33333333 87.        ]
 [15.         87.        ]
 [14.         87.        ]
 …
 [43.         89.        ]
 [75.         91.        ]
 [52.         91.        ]]
```