作业1: 数据探索性分析与数据预处理

一、Wine Reviews

1、数据说明

In [12]:
```
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
import tqdm
import numpy as np
from sklearn.linear_model import LinearRegression
import scipy.stats as stats
import warnings
warnings.filterwarnings('ignore')
```

In [6]:
```
data = pd.read_csv('./wine/winemag-data_first150k.csv', index_col=0)
print('属性类别数：', len(data.columns))
print('总行数：', len(data))
print('示例数据：')
data.head(5)
```

```
属性类别数： 10
总行数： 150930
示例数据：
```

Out[6]:

	country	description	designation	points	price	province	region_1	region_2	vari
0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Napa	Caber Sauvig
1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Northern Spain	Toro	NaN	Tinta T
2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonoma	Sauvig Bl
3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Oregon	Willamette Valley	Willamette Valley	Pinot N
4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	NaN	Prove red bl

2、数据摘要

In [3]:

```
num_fields = data.select_dtypes(include=np.number).columns.values
nom_fields = data.select_dtypes(exclude=np.number).columns.values
print('标称属性:', nom_fields)
print('数值属性:', num_fields)
```

```
标称属性: ['country' 'description' 'designation' 'province' 'region_1' 'regi
on_2'
 'variety' 'winery']
数值属性: ['points' 'price']
```

1）标称属性

以“country”属性为例，进行频数统计，其余标称属性类似。

In [4]:
```
#修改该field即可
field = 'country'
print('频数统计:')
data[field].value_counts()
```

频数统计:

Out[4]:
```
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
Lithuania                     8
India                         8
Czech Republic                6
Ukraine                       5
South Korea                   4
Switzerland                   4
Bosnia and Herzegovina        4
Slovakia                      3
China                         3
Egypt                         3
Tunisia                       2
Montenegro                    2
Japan                         2
Albania                       2
US-France                     1
Name: country, dtype: int64
```

2）数值属性

In [5]:
```
data.describe()
```

Out[5]:

	points	price
count	150930.000000	137235.000000
mean	87.888418	33.131482
std	3.222392	36.322536
min	80.000000	4.000000
25%	86.000000	16.000000
50%	88.000000	24.000000
75%	90.000000	40.000000
max	100.000000	2300.000000

5数概括

points：80、86、88、90、100
price；4、16、24、40、2300

缺失值个数统计

In [7]:
```
print('null of points:',data['points'].isnull().sum())
print('null of price:',data['price'].isnull().sum())
```

```
null of points: 0
null of price: 13695
```

3、数据可视化

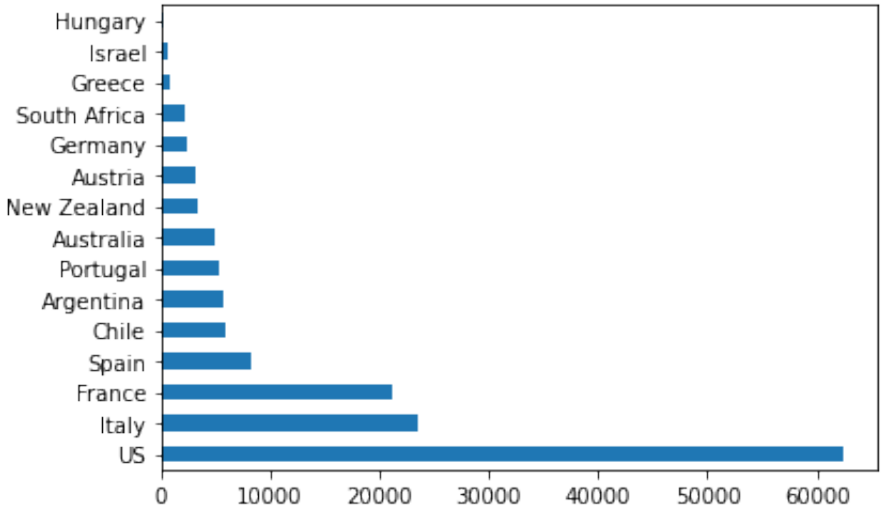
1）标称属性

同样以“country”属性为例，绘制直方图检查数据分布，其余标称属性类似。

In [10]:
```
field = 'country'
print('聚会太多导致显示效果不好，所以取前15个聚会为例（其余聚会的频数都小于200）：')
data[field].value_counts().head(15).plot.barh()
```

聚会太多导致显示效果不好，所以取前15个聚会为例（其余聚会的频数都小于200）：

Out[10]: <AxesSubplot:>



2）数值属性

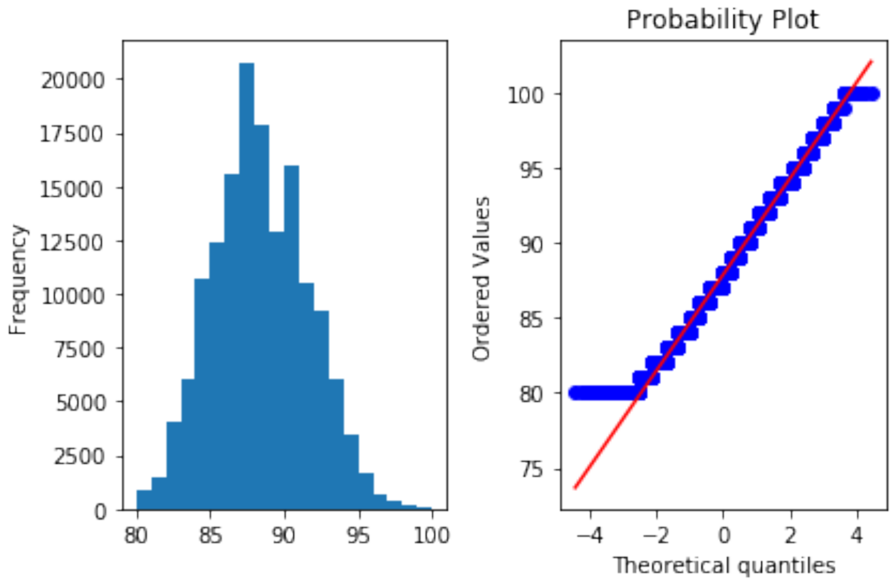
绘制直方图和Q-Q图检查数据分布，并绘制盒图检查离群点。

直方图和Q-Q图

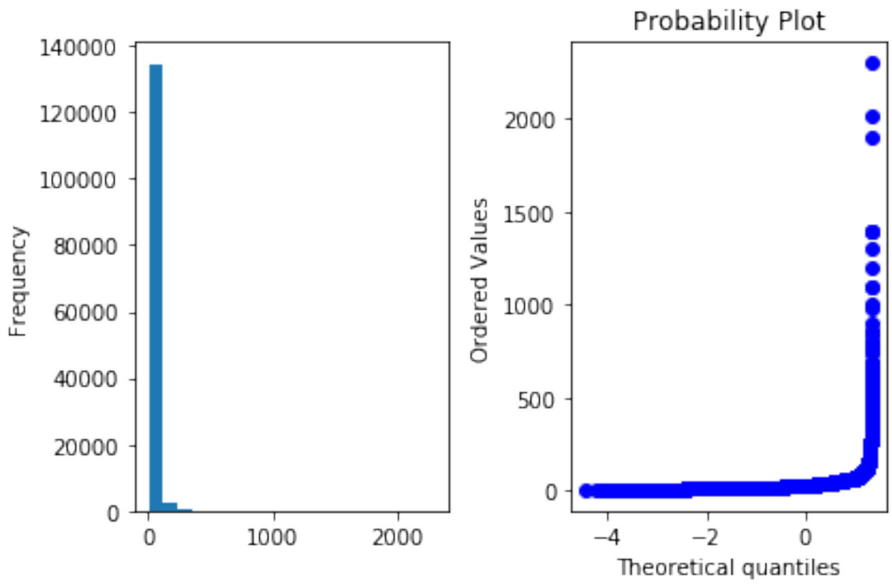
In [8]:

```
for field in num_fields:
    print(field, '直方图和Q-Q图:')
    plt.subplot(1, 2, 1)
    data[field].plot.hist(bins=20)
    plt.subplot(1, 2, 2)
    stats.probplot(data[field], plot=plt)
    plt.tight_layout()  # 调整整体空白
    plt.show()
```

points 直方图和Q-Q图：



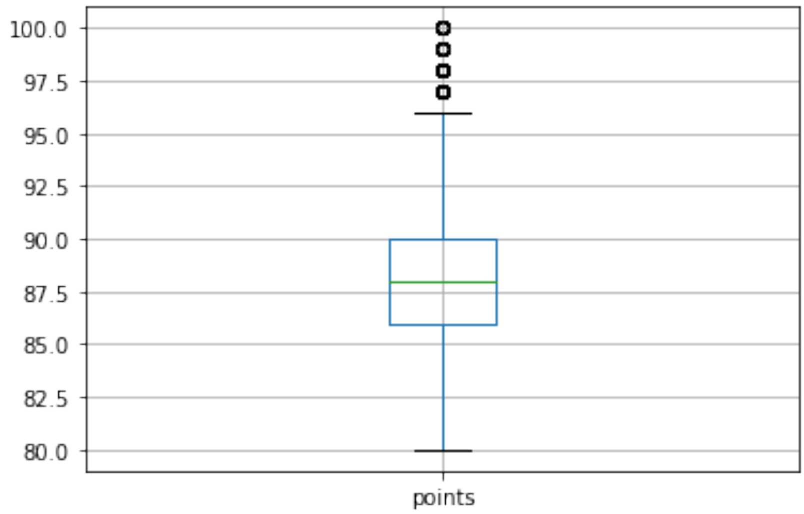
price 直方图和Q-Q图：

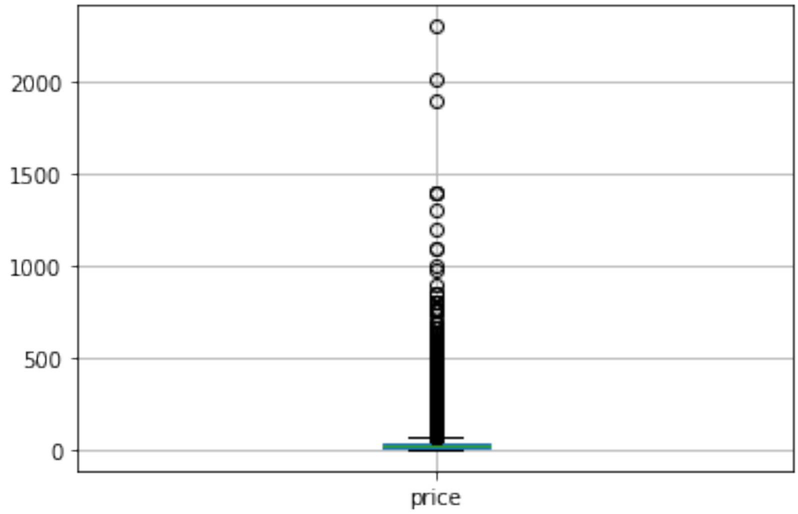


通过直方图和Q-Q图可以看出“points”属性符合正态分布，而“price”属性不符合。

盒图

In [9]:
```
for field in num_fields:
    data.boxplot(field)
    plt.show()
```





4、数据缺失处理

首先对缺失数据进行统计。

In [10]:

```
missing_data = data.isnull().sum()
missing_data = missing_data[missing_data != 0]
missing_data
```

Out[10]:

```
country            5
designation    45735
price          13695
province           5
region_1       25060
region_2       89977
dtype: int64
```

1）将缺失部分剔除

将包含缺失值的整行直接删除。

In [11]:

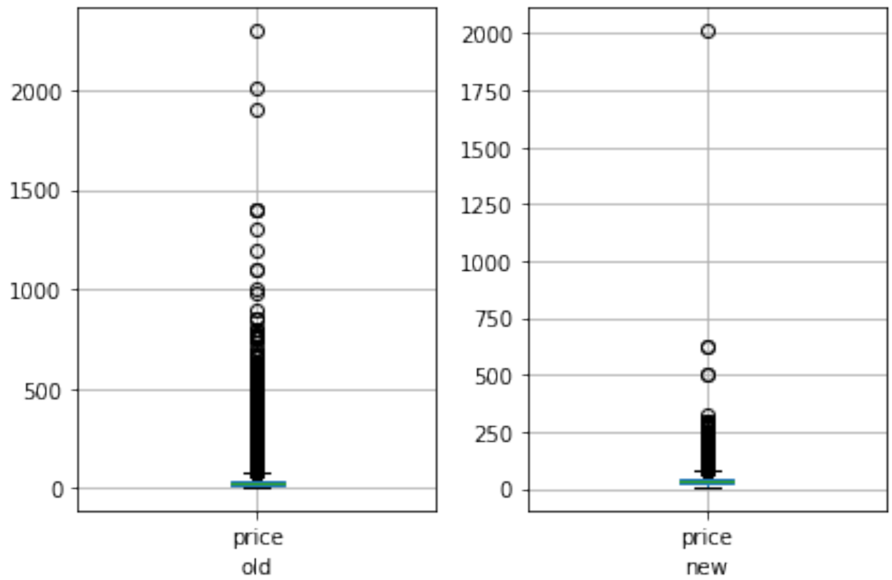
```
print('原始数据行数:', len(data))
drop_data = data.dropna(how='any')
print('将缺失部分剔除后数据行数:', len(drop_data))
```

原始数据行数: 150930
将缺失部分剔除后数据行数: 39241

In [12]:

```
print('以 price 属性为例，通过盒图对比新旧数据:')
field = 'price'
plt.subplot(1, 2, 1)
data.boxplot(field)
plt.xlabel('old')
plt.subplot(1, 2, 2)
drop_data.boxplot(field)
plt.xlabel('new')
plt.tight_layout()  # 调整整体空白
plt.show()
```

以 price 属性为例，通过盒图对比新旧数据:



In [13]:

```
drop_data.isna().sum()
```

Out[13]:

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

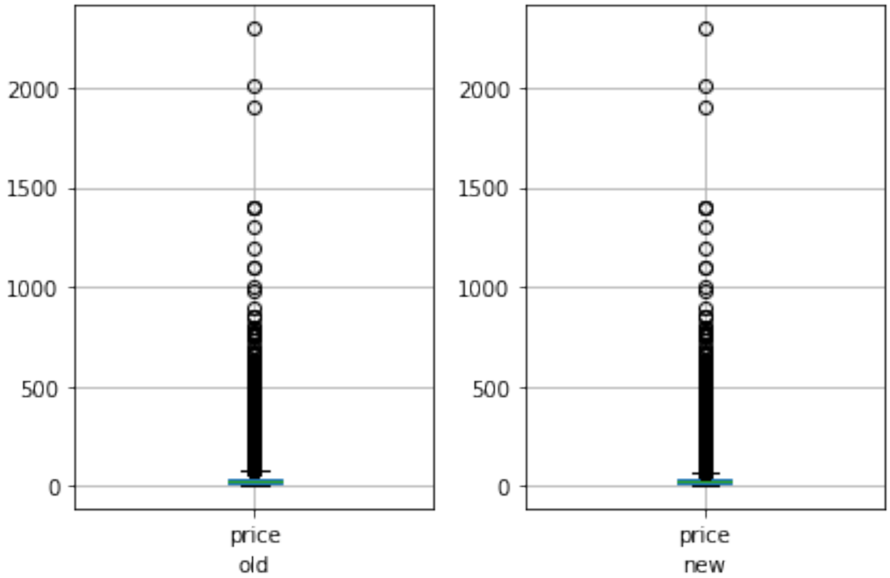
2）用最高频率值来填补缺失值

In [14]:

```
print('以 price 属性为例，通过盒图对比新旧数据：')
field = 'price'
mode = data[field].mode()[0]
new_data = data.fillna({field: mode})
print(field, '属性的最高频率值为：', mode)

plt.subplot(1, 2, 1)
data.boxplot(field)
plt.xlabel('old')
plt.subplot(1, 2, 2)
new_data.boxplot(field)
plt.xlabel('new')
plt.tight_layout()  # 调整整体空白
plt.show()
```

```
以 price 属性为例，通过盒图对比新旧数据：
price 属性的最高频率值为： 20.0
```



In [15]:

```
data[data[field].isna()][field].head(5)
```

Out[15]:

```
32     NaN
56     NaN
72     NaN
82     NaN
116    NaN
Name: price, dtype: float64
```

In [16]:

```
new_data[data[field].isna()][field].head(5)
```

Out[16]:

```
32     20.0
56     20.0
72     20.0
82     20.0
116    20.0
Name: price, dtype: float64
```

可以看出新数据中 price 属性的缺失值已经被填充。

3）通过属性的相关关系来填补缺失值

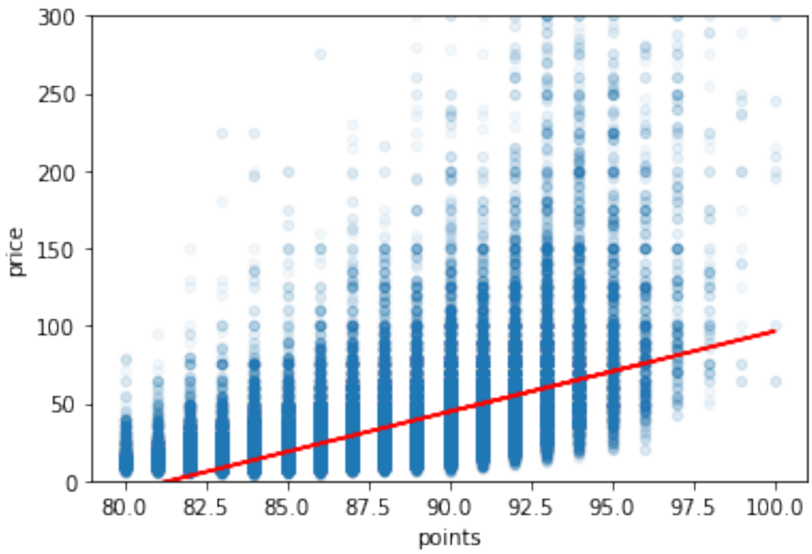
In [17]:
```
data.corr()
```

Out[17]:

	points	price
points	1.000000	0.459863
price	0.459863	1.000000

可以看出“price”和“points”属性之间存在正相关关系，因此可以建立线性回归模型通过“points”值预测缺失的“price”值。

In [18]:
```
drop_data = data.dropna(subset=['price'])
x = drop_data['points']
y = drop_data['price']
x = np.array(x).reshape(-1, 1)
model  = LinearRegression()
model.fit(x, y)
drop_data.plot(kind="scatter", x="points", y="price", alpha=0.05)
plt.plot(x, model.predict(x), 'r-')
plt.ylim(0,300)
plt.show()
```



In [19]:
```
new_data = data.copy()
for index, row in new_data[data['price'].isna()].iterrows():
    new_data['price'][index] = model.predict(np.array(row['points']).resha
```

In [20]:
```
data[data['price'].isna()].head(5)
```

Out[20]:

	country	description	designation	points	price	province	region_1	region_2	va
32	Italy	Underbrush, scorched earth, menthol and plum s...	Vigna Piaggia	90	NaN	Tuscany	Brunello di Montalcino	NaN	Sangi
56	France	Delicious while also young and textured, this ...	Le Pavé	90	NaN	Loire Valley	Sancerre	NaN	Sauv
72	Italy	This offers aromas of red rose, wild berry, da...	Bussia Riserva	91	NaN	Piedmont	Barolo	NaN	Nel
82	Italy	Berry, baking spice, dried iris, mint and a hi...	Palliano Riserva	91	NaN	Piedmont	Roero	NaN	Nel
116	Spain	Aromas of brandied cherry and crème de cassis ...	Dulce Tinto	86	NaN	Levante	Jumilla	NaN	Mona

In [21]:

```
new_data[data['price'].isna()].head(5)
```

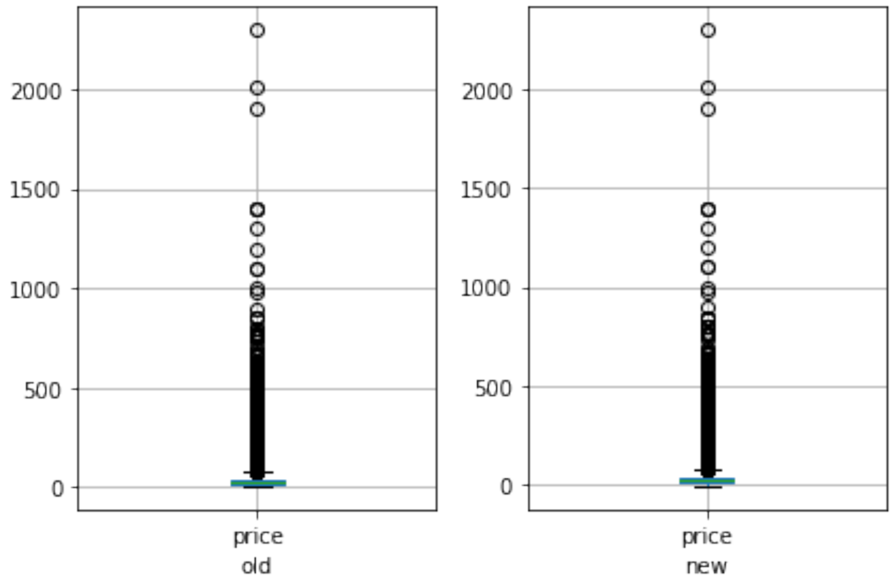
Out[21]:

	country	description	designation	points	price	province	region_1	region_2
32	Italy	Underbrush, scorched earth, menthol and plum s...	Vigna Piaggia	90	44.600434	Tuscany	Brunello di Montalcino	NaN
56	France	Delicious while also young and textured, this ...	Le Pavé	90	44.600434	Loire Valley	Sancerre	NaN
72	Italy	This offers aromas of red rose, wild berry, da...	Bussia Riserva	91	49.785122	Piedmont	Barolo	NaN
82	Italy	Berry, baking spice, dried iris, mint and a hi...	Palliano Riserva	91	49.785122	Piedmont	Roero	NaN
116	Spain	Aromas of brandied cherry and crème de cassis ...	Dulce Tinto	86	23.861683	Levante	Jumilla	NaN

可以看出新数据中缺失的“price”值已经通过“points”值预测填充。

In [22]:

```
plt.subplot(1, 2, 1)
data.boxplot('price')
plt.xlabel('old')
plt.subplot(1, 2, 2)
new_data.boxplot('price')
plt.xlabel('new')
plt.tight_layout()   # 调整整体空白
plt.show()
```



4）通过数据对象之间的相似性来填补缺失值

以填充“price”为例，使用相同“variety”的数据对象的“price”均值来填充缺失数据，如果没有相同的“variety”，则接下来依次考虑相同的“winery”、“designation”、“region_1”、“province”。

In [23]:

```
full_data = data[data['price'].notna()]
new_data = data.copy()
consider_fields = ['variety', 'winery', 'designation', 'region_1', 'provinc
for i, row in tqdm.tqdm(list(new_data[data['price'].isna()].iterrows())):
    for field in consider_fields:
        tmp_data = full_data[full_data[field]==row[field]]
        if len(tmp_data) > 0:
            new_data['price'][i] = tmp_data['price'].mean()
            break
```

```
100%|██████████| 13695/13695 [06:13<00:00, 36.70it/s]
```

In [24]:

```
data[data['price'].isna()].head(5)['price']
```

Out[24]:

```
32     NaN
56     NaN
72     NaN
82     NaN
116    NaN
Name: price, dtype: float64
```

In [25]:

```
new_data[data['price'].isna()].head(5)['price']
```

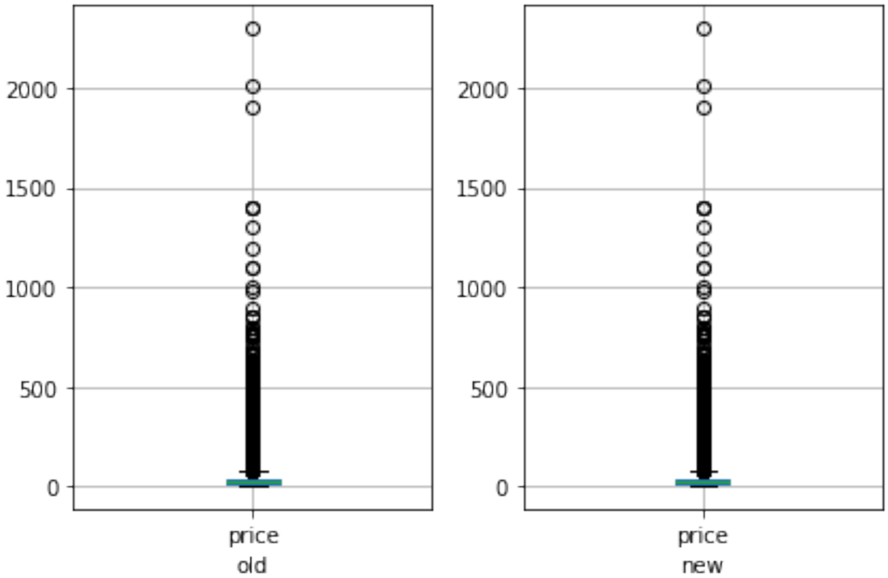
Out[25]:

```
32      36.216047
56      18.615296
72      66.406802
82      66.406802
116     17.459459
Name: price, dtype: float64
```

可以看出新数据中缺失的“price”值已经通过相似对象的“price”属性的均值进行填充。

In [26]:

```
plt.subplot(1, 2, 1)
data.boxplot('price')
plt.xlabel('old')
plt.subplot(1, 2, 2)
new_data.boxplot('price')
plt.xlabel('new')
plt.tight_layout()  # 调整整体空白
plt.show()
```



通过上述分析可以看出在其他属性值相同的情况下，有缺失的属性值的变动很大，说明这些缺失值无法通过其他行来进行填补。