

Data_Preprocessing_Đáp_án_thực_hành

August 31, 2021

1 Bài tập bổ sung

Phần bài tập này là các câu hỏi mở rộng, làm tiếp theo bài toán ở trên. Học viên cần viết mã để thực hiện các yêu cầu dưới đây:

Bài tập 0: Sử dụng `sns.boxplot()` để quan sát đặc điểm phân bố dữ liệu của các trường số, mỗi trường này có outlier ko?

```
[ ]: # Sử dụng boxplot để quan sát phân bố của dữ liệu và phát hiện ngoại lai của
      ↳ từng trường dữ liệu
      # Gợi ý: sns.boxplot(data_field)

vars = ['price', 'total_sqft_float', 'price_per_sqft', 'balcony', 'bath', 'bhk']
plt.figure(figsize=(16,12))
for i,var in enumerate(vars):
    plt.subplot(3,2,i+1)
    sns.boxplot(df8[var])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```

FutureWarning

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```

FutureWarning

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```

FutureWarning

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```

FutureWarning

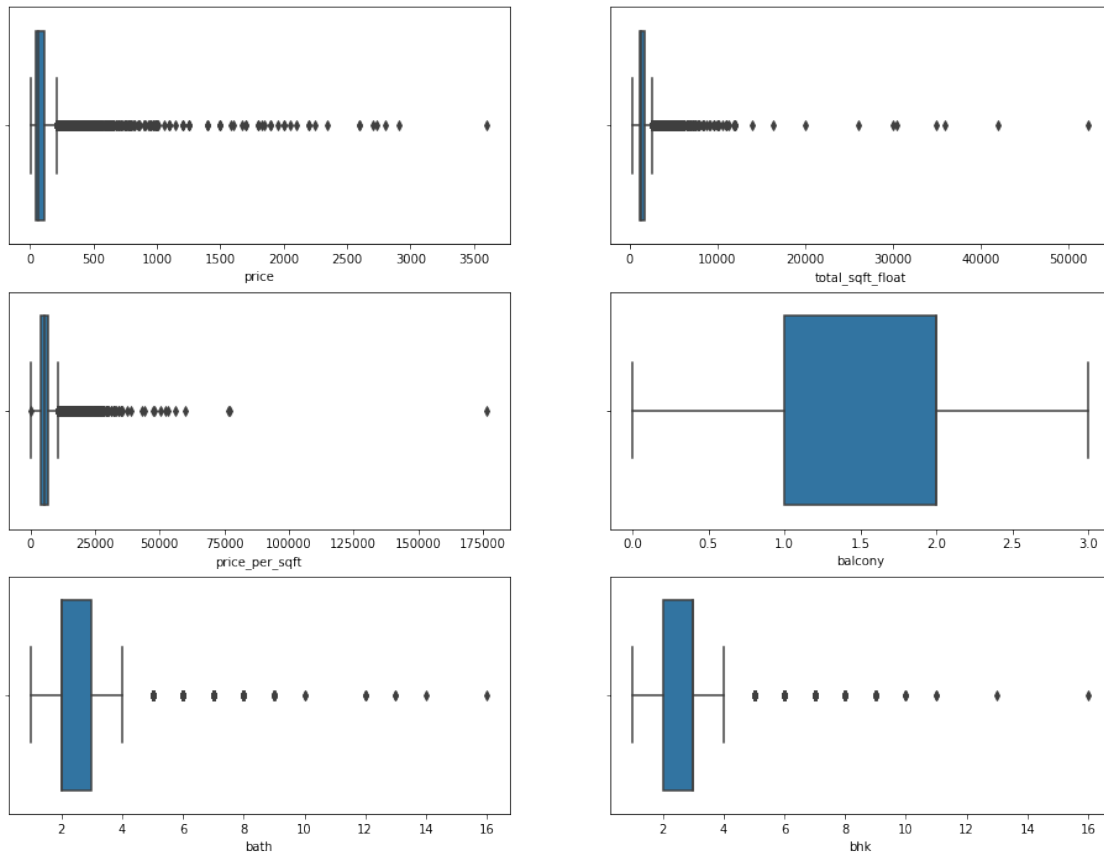
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
```

Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning



Bài tập 1: Viết hàm bỏ đi các điểm dữ liệu có price per sqft dựa trên mean, std của các ngôi nhà dựa trên từng vị trí

Gợi ý: Xét trên từng vị trí (location), ngôi nhà thỏa mãn phải có $price_per_sqft \in [mean - std, mean + std]$

```
[ ]: # Viết hàm tại đây
def remove_pps_outliers(df):
    df_out = pd.DataFrame()
    for key, subdf in df.groupby('location'):
        m=np.mean(subdf.price_per_sqft)
```

```

    st=np.std(subdf.price_per_sqft)
    reduced_df = subdf[(subdf.price_per_sqft>(m-st))&(subdf.
→price_per_sqft<=(m+st))]
    df_out = pd.concat([df_out, reduced_df], ignore_index = True)
    return df_out
#-----
df9 = remove_pps_outliers(df8)
df9.shape

```

[]: (8888, 11)

Bài tập 2: Loại bỏ outlier xét theo trường bhk (số phòng)

Xét theo từng khu vực địa lí và theo từng loại nhà với số lượng phòng khác nhau, có một số ngôi nhà có giá không hợp lí (outliers), hãy tìm cách loại bỏ các outlier này. Cần ghi rõ quy tắc ghi nhận outlier

```

[ ]: def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk]={
                'mean':np.mean(bhk_df.price_per_sqft),
                'std':np.std(bhk_df.price_per_sqft),
                'count':bhk_df.shape[0]}
        for bhk, bhk_df in location_df.groupby('bhk'):
            stats=bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices = np.append(exclude_indices, bhk_df[bhk_df.
→price_per_sqft<(stats['mean'])].index.values)
        return df.drop(exclude_indices, axis='index')

df10 = remove_bhk_outliers(df9)
df10.shape

```

[]: (7194, 11)

Bài tập 3: Loại bỏ outlier khi xét trường 'bathroom'

```

[ ]: df10.bath.unique() #Có thể quan sát thấy một số căn nhà có số phòng tắm quá lớn
→(VD: 12!!!)

```

[]: array([3., 2., 1., 4., 5., 8., 9., 6., 7., 12.])

```

[ ]: df10[df10.bath > df10.bhk+2]

```

```
[ ]:
```

	area_type	availability	location	size \
1861	Built-up Area	Ready To Move	Chikkabanavar	4 Bedroom
5836	Built-up Area	Ready To Move	Nagasandra	4 Bedroom
7098	Super built-up Area	Ready To Move	Sathya Sai Layout	6 BHK
7569	Super built-up Area	Ready To Move	Thanisandra	3 BHK

	total_sqft	bath	balcony	price	total_sqft_float	bhk	price_per_sqft
1861	2460	7.0	2.000000	80.0	2460.0	4	3252.032520
5836	7000	8.0	1.584376	450.0	7000.0	4	6428.571429
7098	11338	9.0	1.000000	1000.0	11338.0	6	8819.897689
7569	1806	6.0	2.000000	116.0	1806.0	3	6423.034330

```
[ ]: df11 = df10[df10.bath < df10.bhk+2]
df11.shape
```

```
[ ]: (7120, 11)
```

```
[ ]: df11.head()
```

```
[ ]:
```

	area_type	availability	location	size \
0	Super built-up Area	Ready To Move	Devarabeesana Halli	3 BHK
1	Built-up Area	Ready To Move	Devarabeesana Halli	3 BHK
2	Super built-up Area	Ready To Move	Devarabeesana Halli	3 BHK
4	Super built-up Area	Ready To Move	Devarachikkanahalli	2 BHK
5	Plot Area	Ready To Move	Devarachikkanahalli	2 Bedroom

	total_sqft	bath	balcony	price	total_sqft_float	bhk	price_per_sqft
0	1672	3.0	2.0	150.0	1672.0	3	8971.291866
1	1750	3.0	3.0	149.0	1750.0	3	8514.285714
2	1750	3.0	2.0	150.0	1750.0	3	8571.428571
4	1250	2.0	2.0	40.0	1250.0	2	3200.000000
5	1200	2.0	2.0	83.0	1200.0	2	6916.666667

```
[ ]: #Quan sát lại kết quả sau khi xử lý với boxplot
plt.figure(figsize=(16,12))
for i,var in enumerate(vars):
    plt.subplot(3,2,i+1)
    sns.boxplot(df11[var])
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an

explicit keyword will result in an error or misinterpretation.

FutureWarning

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

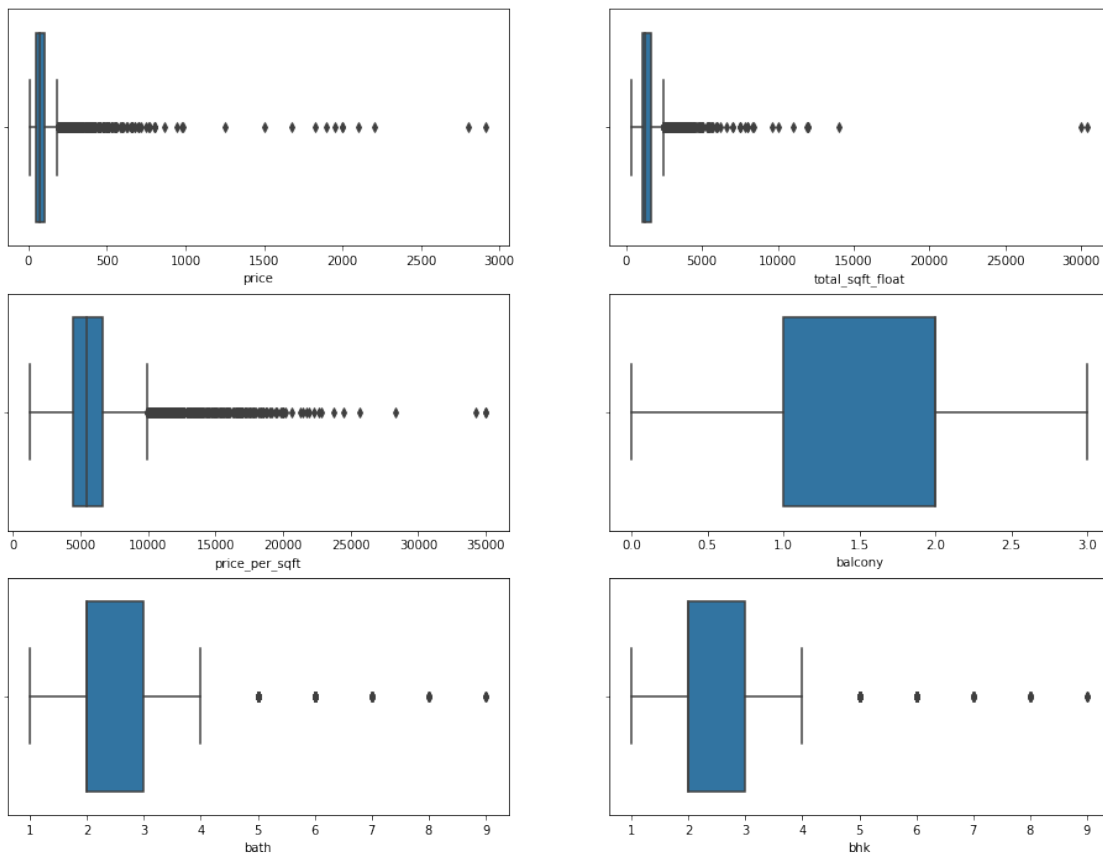
FutureWarning

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning



Bài tập 4: Xem xét bỏ đi các trường không cần thiết

Gợi ý: bỏ đi ['area_type', 'availability', 'location', 'size', 'total_sqft']

```
[ ]: df12 = df11.drop(['area_type', 'availability', 'location', 'size', 'total_sqft'], axis=1)
df12.head()
```

```
[ ]:      bath  balcony  price  total_sqft_float  bhk  price_per_sqft
0      3.0      2.0  150.0          1672.0      3      8971.291866
1      3.0      3.0  149.0          1750.0      3      8514.285714
2      3.0      2.0  150.0          1750.0      3      8571.428571
4      2.0      2.0   40.0          1250.0      2      3200.000000
5      2.0      2.0   83.0          1200.0      2      6916.666667
```

```
[ ]: #Lưu kết quả xử lý cuối cùng:
df12.to_csv("clean_data.csv", index=False)
```

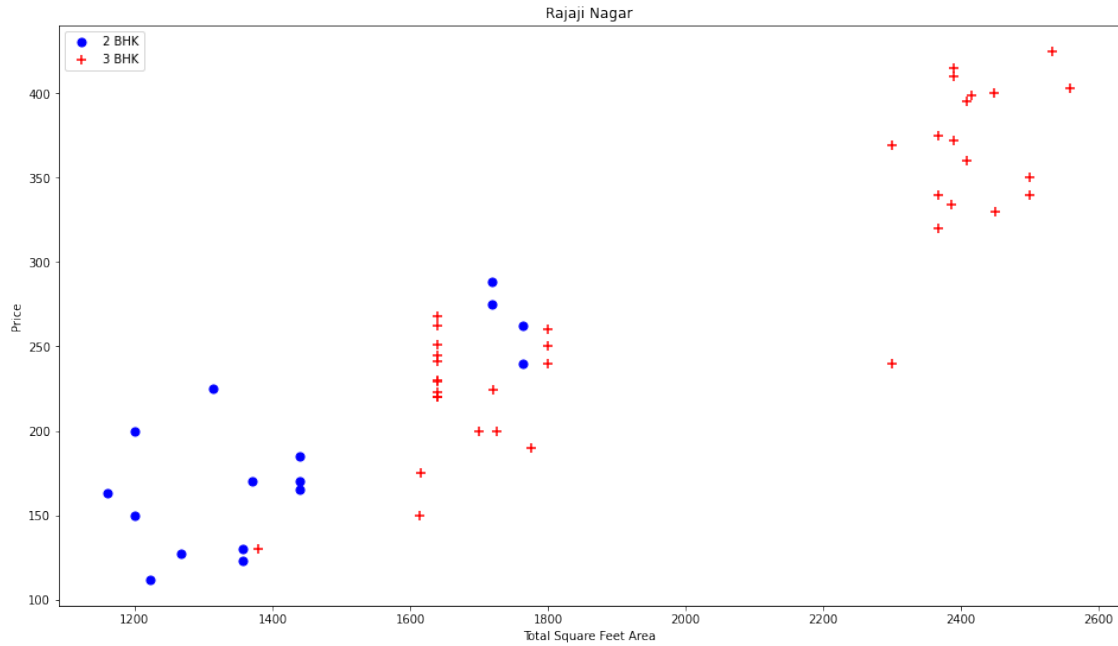
Bài tập 5*: Viết hàm trực quan hóa thể hiện mối tương quan giữa tổng diện tích (total_sqft) và giá nhà (price) theo từng vị trí địa lý (location) (tùy chọn minh họa theo 2 vị trí nào đó), của những căn nhà có 2 hoặc 3 phòng. Và cần phân biệt rõ điểm dữ liệu nào tương ứng với nhà có 2 phòng, điểm nào tương ứng với nhà có 3 phòng?

Gợi ý: Kết quả tương tự như hình dưới/ hoặc biểu đồ khác có ý nghĩa tương đương

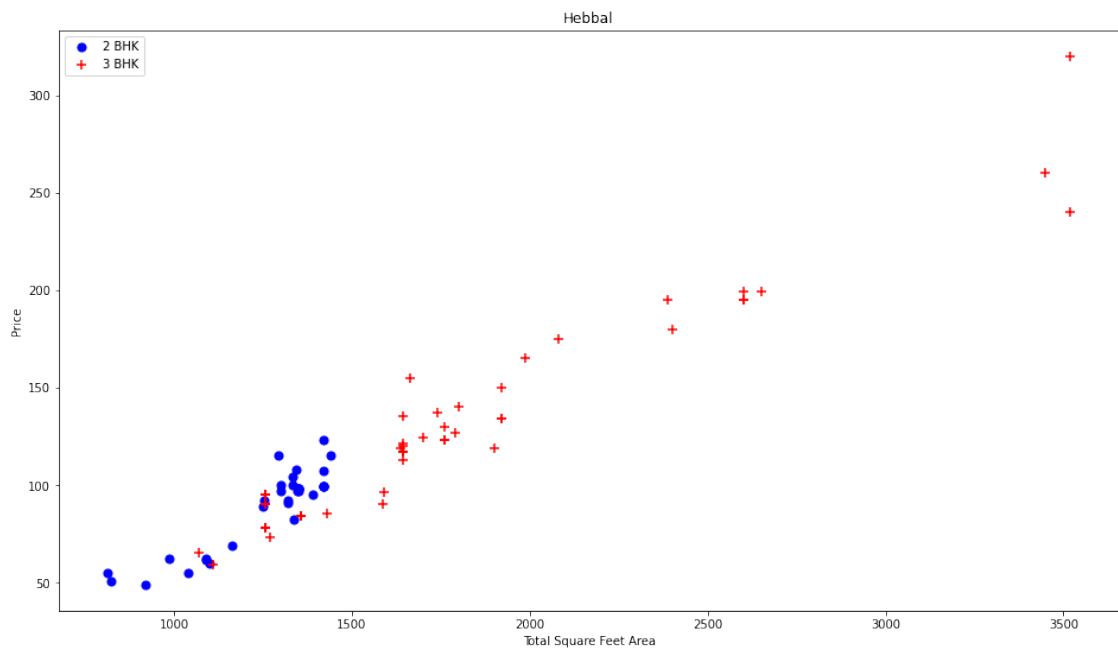
```
[ ]: #Gợi ý: Sử dụng plt.scatter() .... hoặc câu lệnh khác tương đương. Làm với df9

def plot_scatter_chart(df, location):
    bhk2 = df[(df.location==location) & (df.bhk==2)]
    bhk3 = df[(df.location==location) & (df.bhk==3)]
    plt.figure(figsize=(16,9))
    plt.scatter(bhk2.total_sqft_float, bhk2.price, color='Blue', label='2 BHK', s=50)
    plt.scatter(bhk3.total_sqft_float, bhk3.price, color='Red', label='3 BHK', s=50, marker="+")
    plt.xlabel("Total Square Feet Area")
    plt.ylabel("Price")
    plt.title(location)
    plt.legend()

plot_scatter_chart(df9, "Rajaji Nagar")
```



```
[ ]: plot_scatter_chart(df9, "Hebbal")
```



Bài tập 6*: Thực hiện các câu lệnh để trả lời các câu hỏi dưới đây:

```
[ ]: # Thống kê giá nhà theo từng loại khu vực (area_type). Làm với df9
# Gợi ý: Phần này có thể đưa ra kết quả dạng bảng hoặc biểu đồ (cột, histogram,
→...). Sử dụng các lệnh: df.groupby(), df.sortvalues(), ...
# xem xét theo từng khu vực, thì giá nhà trung bình (price_per_sqft) là bao
→nhiều, tương quan về giá nhà trung bình giữa các khu vực

import random

df91 = df9.groupby('area_type')['price_per_sqft'].mean().
→reset_index(name='money')
df91 = df91.sort_values(by = 'money')

df91['money'] = df91['money'].apply(lambda x : round(x, 2))
n = df91['area_type'].unique().__len__()+1
all_colors = list(plt.cm.colors.cnames.keys())

random.seed(100) #Chọn màu ngẫu nhiên cho các cột :)
c = random.choices(all_colors, k=n)

plt.figure(figsize=(16,10), dpi= 80)
plt.bar(df91['area_type'], df91['money'], color=c, width=.5)
for i, val in enumerate(df91['money'].values):
    plt.text(i, val, float(val), horizontalalignment='center',
→verticalalignment='bottom', fontdict={'fontweight':500, 'size':12})

plt.gca().set_xticklabels(df2['area_type'], rotation=60, horizontalalignment=
→'right')
plt.title("Biểu đồ thể hiện giá nhà đất trung bình theo khu vực", fontsize=22)
plt.ylabel('amount of money')
# plt.ylim(25,35)
plt.show()
```