HousePrice - Đống Đa

In [2]:
```
import pandas as pd
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt
```

Yêu cầu

- Vẽ biểu đồ phân tích mối liên hệ giữa diện tích với giá nhà, giữa số phòng ngủ với giá nhà, giữa số toilet với giá nhà.
- Vẽ biểu đồ so sánh giá nhà trung bình trên 1 m2 giữa các hình thức nhà (type_of_land).
- Vẽ biểu đồ thể hiện tỉ lệ % bài đăng (bản ghi) giữa các hình thức nhà (type_of_land).
- Vẽ biểu đồ thể hiện sự thay đổi giá nhà trung bình trên 1m2 theo số lượng phòng ngủ.

In [3]:
```
# Bo qua canh bao UserWaring
import warnings
warnings.filterwarnings('ignore', category=UserWarning, module='openpyxl')

df = pd.read_excel('../Data/house_price_đống đa.xlsx', sheet_name='Sheet1', engine='openpyxl')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 18 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   title              1000 non-null   object
 1   address            1000 non-null   object
 2   area               994 non-null    float64
 3   price              944 non-null    float64
 4   postDate           1000 non-null   datetime64[ns]
 5   land_certificate   693 non-null    object
 6   house_direction    44 non-null     object
 7   balcony_direction  24 non-null     object
 8   toilet             551 non-null    float64
 9   bedroom            635 non-null    float64
 10  floor              376 non-null    float64
 11  type_of_land       1000 non-null   object
 12  street_name        808 non-null    object
 13  ward_name          802 non-null    object
 14  district_name      1000 non-null   object
 15  city_name          1000 non-null   object
 16  lat                1000 non-null   float64
 17  long               1000 non-null   float64
dtypes: datetime64[ns](1), float64(7), object(10)
memory usage: 140.8+ KB
```

In [3]:
```
df.head(10)
```

Out[3]:

	title	address	area	price	postDate	land_certificate	house_direction	balcony_direction	toilet	bedroom	floor	type_
0	Bán nhà Trần Quang Diệu mới coong đẹp 50m2x6 t...	Đường Trần Quang Diệu, Phường Trung Liệt, Đốn...	50.0	14700.0	2021-01-01	Sổ đỏ	NaN	NaN	4.0	6.0	6.0	I
1	Bán nhà mặt phố Tây Sơn - 6 tầng. Kinh doanh t...	Đường Tây Sơn, Phường Trung Liệt, Đống Đa, Hà...	35.0	12500.0	2021-02-19	Sổ đỏ	NaN	NaN	NaN	NaN	6.0	Bán r
2	Bán nhà số 36 Đoàn Kết - phố Khâm Thiên - Đống...	Số 36 Đoàn Kết, Phố Khâm Thiên, Phường Thổ Qu...	57.0	4200.0	2021-03-10	Sổ đỏ	Nam	Nam	3.0	8.0	4.0	I
3	Bán nhà 6 tầng mới kinh doanh mặt ngõ	Ngõ 1194, Đường Láng, Phường Láng Thượng, Đố...	62.0	11000.0	2021-03-13	Sổ đỏ	Đông	Đông-Nam	6.0	5.0	6.0	I

	title	address	area	price	postDate	land_certificate	house_direction	balcony_direction	toilet	bedroom	floor	type_
	1194 Đườ...											
4	Bán nhà mặt phố Thái Hà 70m2, 5 tầng, 4.5m mặt...	Phố Thái Hà, Phường Trung Liệt, Đống Đa, Hà Nội	70.0	36000.0	2021-03-13	Sổ đỏ	NaN	NaN	NaN	6.0	5.0	Bán r
5	Bán gấp mặt phố Chùa Bộc, Đống Đa, thang máy, ...	Phố Chùa Bộc, Phường Quang Trung, Đống Đa, Hà Nội	41.0	13500.0	2021-04-04	Sổ đỏ	NaN	NaN	5.0	3.0	6.0	Bán r
6	Bán nhà mặt phố Hoàng Cầu - Mai Anh Tuấn. Đối ...	Phố Hoàng Cầu, Phường Láng Hạ, Đống Đa, Hà Nội	50.0	13000.0	2021-04-06	Sổ đỏ	NaN	NaN	5.0	4.0	5.0	Bán r
7	Bán nhà phường Thổ Quan ngõ	Đường Trung Phụng, Phường Thổ Quan,	26.0	2900.0	2021-04-06	Sổ đỏ	NaN	NaN	4.0	3.0	4.0	I

	title	address	area	price	postDate	land_certificate	house_direction	balcony_direction	toilet	bedroom	floor	type_
	thoáng 26m2 x 4 tầ...	Đống Đa, ...										
8	Chính chủ bán nhà Xã Đàn 40m2, chỉ 3. X tỷ	Phố Xã Đàn, Phường Kim Liên, Đống Đa, Hà Nội	40.0	3950.0	2021-04-06	Sổ đỏ	NaN	NaN	NaN	NaN	5.0	I
9	7 tầng thang máy gara ở vip 41m2, MT 4.6m ở vi...	Đường Hồ Đắc Di, Phường Nam Đồng, Đống Đa, Hà...	41.0	8900.0	2021-04-06	NaN	Đông-Nam	NaN	NaN	3.0	7.0	I

In [15]:
```
df_1 = df.dropna(subset=['area','price'])
print(df_1.loc[191])
# print(df_1['type_of_land'].value_counts())
# df_2 = df_1[['area','price']]
# print(df_2)
# z = np.abs(stats.zscore(df_2)) # Hệ số chuẩn Z Score
# df_3 = df_2[(z<3).all(axis=1)] # Loại bỏ các giá trị ngoại lai (outliners), return index
# print(df_2[(z<3).all(axis=1)])
# # Tham số axis=0, nhóm các dòng (index) lại để tính, axis=1 thì nhóm các cột lại để tính
# # https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.all.html
# plt.scatter(df_3['area'], df_3['price'])
# plt.title('Relationship between Area and Price', fontsize=16)
# plt.xlabel('Area', fontsize=14)
# plt.ylabel('Price', fontsize=14)
# slope, intercept = np.polyfit(df_3.area, df_3.price, deg=1)
```

```
# plt.plot(df_3.area, slope*df_3.area + intercept)
# plt.show()
```

```
title                tuyển dụng lao động làm việc tại nhật
address                                Hà Nội\nQuận Đống Đa
area                                                   0.0
price                                                 30.0
postDate                             2021-01-01 00:00:00
land_certificate                                       NaN
house_direction                                        NaN
balcony_direction                                      NaN
toilet                                                 NaN
bedroom                                                NaN
floor                                                  NaN
type_of_land                           Bất động sản khác
street_name                                            NaN
ward_name                                              NaN
district_name                                 Quận Đống Đa
city_name                                  Thành phố Hà Nội
lat                                               21.018072
long                                             105.829949
Name: 191, dtype: object
```

Z-Score? Điều cần biết

> Trong khoảng từ -3 đến 3, nghĩa là chúng nằm trong độ lệch chuẩn trên và dưới giá trị trung bình, hay còn gọi là **giá trị ngoại lai**

Hệ số chuẩn Z-score

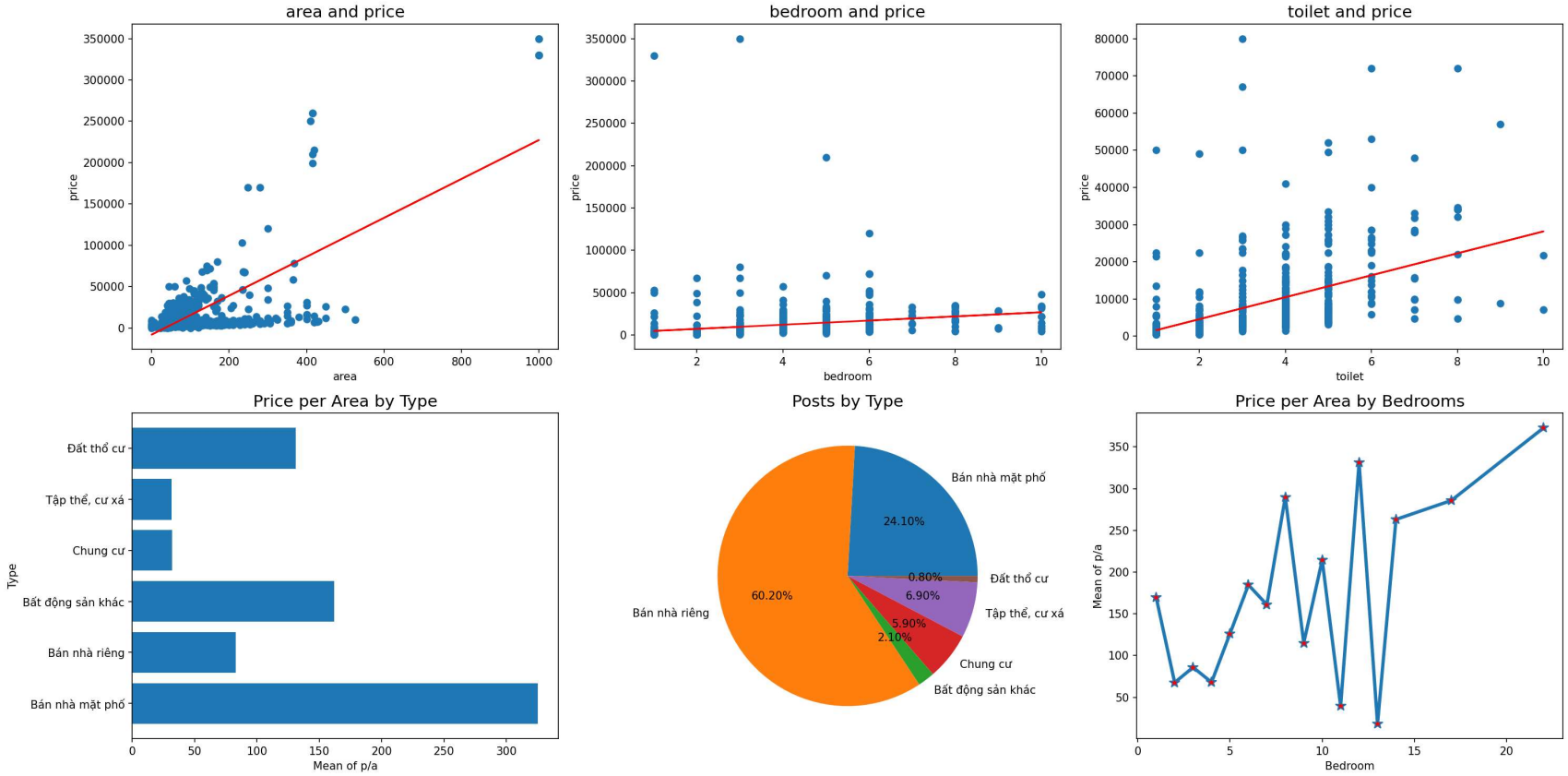
stats.zscore

In [115…

```
# 3 Yêu cầu đầu tiên
def o_c(str1, str2): # (outliners cleaner) Tạo 1 hàm tự động cleaning outliners cho nhanh :D
    df_1 = df.dropna(subset=[str1,str2])
    df_2 = df_1[[str1,str2]]
    z = np.abs(stats.zscore(df_2))
    return df_2[(z<3).all(axis=1)]


fig, axs = plt.subplots(2, 3, figsize=(20,10), dpi=150)
need_list = [['area','price'],['bedroom','price'],['toilet','price']]
```

```
for i in range(len(need_list)): # Tạo vòng lặp làm 3 yêu cầu đầu tiên cho nhàn :D
    df_bp = o_c(need_list[i][0],need_list[i][1])
    axs[0,i].scatter(df_bp[need_list[i][0]], df_bp[need_list[i][1]])
    axs[0,i].set_title("{} and {}".format(need_list[i][0], need_list[i][1]), fontsize=15)
    axs[0,i].set_xlabel(need_list[i][0])
    axs[0,i].set_ylabel(need_list[i][1])
    # Đường hồi quy tuyến tính: slope-độ dốc, intercept-hệ số chặn
    slope, intercept = np.polyfit(df_bp[need_list[i][0]], df_bp[need_list[i][1]], deg=1)
    axs[0,i].plot(df_bp[need_list[i][0]], slope*df_bp[need_list[i][0]] + intercept, color='r')

# Yêu cầu số 4
df_3 = o_c('area', 'price')
df_4 = (df.loc[df_3.index])[['area','price','type_of_land']]

df_4['type_of_land'] = [(c.replace('\n', '')).strip() for c in df_4['type_of_land']]
mean_price = df_4.groupby(['type_of_land'])[['price']].mean()
mean_area = df_4.groupby(['type_of_land'])[['area']].mean()
axs[1,0].barh(mean_price.index, mean_price['price']/mean_area['area'])
axs[1,0].set_xlabel('Mean of p/a')
axs[1,0].set_ylabel('Type')
axs[1,0].set_title('Price per Area by Type', fontsize=15)

# Yêu cầu số 5
df_1 = df.dropna(subset='type_of_land')
df_1['type_of_land'] = [(c.replace('\n', '')).strip() for c in df_1['type_of_land']] # List Comprehension
total = df_1['type_of_land'].shape[0]
ratio_post = df_1.groupby(['type_of_land'])[['type_of_land']].count()
axs[1,1].pie(ratio_post['type_of_land'],labels=ratio_post.index,autopct='%1.2f%%', textprops={'fontsize': 10}, radius
axs[1,1].set_title('Posts by Type', fontsize=15)

# Yêu cầu số 6
df_4 = (df.loc[df_3.index])[['area','price','bedroom']]

mean_price = df_4.groupby(['bedroom'])[['price']].mean()
mean_area = df_4.groupby(['bedroom'])[['area']].mean()
axs[1,2].plot(mean_price.index, mean_price.price/mean_area.area, linewidth=3, marker='*', markersize=10, markerfacec
axs[1,2].set_title('Price per Area by Bedrooms', fontsize=15)
axs[1,2].set_xlabel('Bedroom')
axs[1,2].set_ylabel('Mean of p/a')
```

```
fig.tight_layout()
plt.show()
```



In [ ]:

In [ ]:

In [ ]: