Case Study - Condo Market in Singapore

Learning Objectives:

- Become familiar with different visualization techniques
- Understand the confounding effect and the way to control for it
- Explore possible relationships among multiple variables

Background: Buying a condo might be a dream for some Singaporeans. Depending on the location and area of the property, the corresponding price differs substantially. For those who want to buy condos for residing or for investments, a deeper understanding of Singapore's real estate market is crucial. The file `condo.csv` contains the prices of condos in Singapore for the past several years. Moreover, some attributes of such condos are also included in the file.

In [1]:
```
import pandas as pd

df = pd.read_csv("condo.csv")
df.head(10)
```

Out[1]:

	name	price	unit_price	district_code	segment	type	area	level	remaining_years
0	SEASCAPE	4388000	2028	4	CCR	Resale	2164	06 to 10	87.0
1	COMMONWEALTH TOWERS	1300000	1887	3	RCR	Resale	689	16 to 20	93.0
2	THE TRILINQ	1755000	1304	5	OCR	Resale	1346	06 to 10	92.0
3	THE CREST	2085000	2201	3	RCR	Resale	947	01 to 05	92.0
4	THE ANCHORAGE	1848888	1468	3	RCR	Resale	1259	01 to 05	999.0
5	MOUNT FABER LODGE	4400000	1188	4	RCR	Resale	3703	06 to 10	999.0
6	BLUE HORIZON	990000	1022	5	OCR	Resale	969	21 to 25	80.0
7	DOVER PARKVIEW	1088000	1162	5	RCR	Resale	936	06 to 10	73.0

	name	price	unit_price	district_code	segment	type	area	level	remaining_years
8	CARIBBEAN AT KEPPEL BAY	1470000	1751	4	RCR	Resale	840	06 to 10	79.0
9	THE INTERLACE	4550000	868	4	RCR	Resale	5242	16 to 20	89.0

Task 1: Explore the relationship between the condo price and the condo type

For a Singaporean who wants to invest in the real estate market, would you suggest him or her to buy resale condos or newly built condos? Please analyze the historical data in the file `condo.csv` to arrive at your conclusion. Specifically, you need to explore the relationship between the condo price and the type of the condo (resale versus new). **Note**: if you want to draw histograms to explore the distribution of the condo price, please set bins as `np.arange(0.5e6, 5e6, 0.1e6)`.

In [2]:
```
df["type"].unique()
```

Out[2]:
```
array(['Resale', 'New Sale'], dtype=object)
```

In [3]:
```
resale_filt = (df["type"] == "Resale")
df_resale = df.loc[resale_filt].copy()
```

In [4]:
```
new_filt = (df["type"] == "New Sale")
df_new = df.loc[new_filt].copy()
```

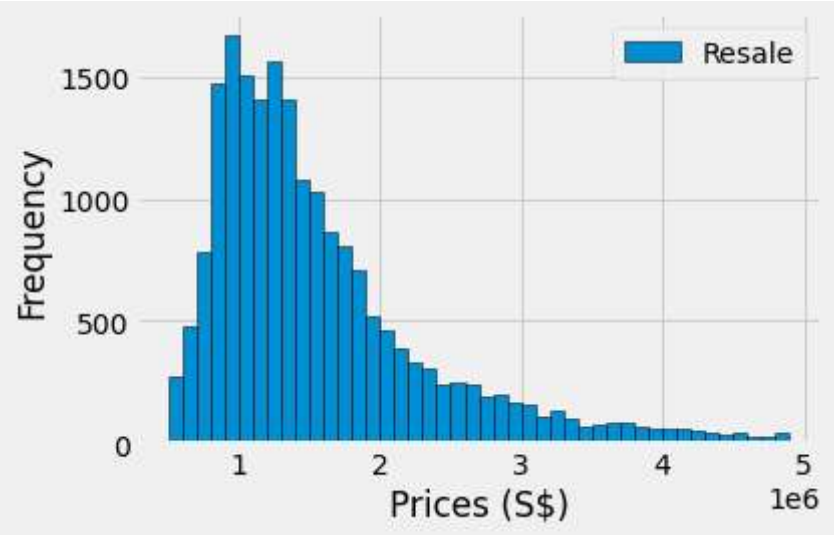
In [5]:
```
from matplotlib import pyplot as plt
import numpy as np
plt.style.use("fivethirtyeight")
```

In [14]:
```
"""
Visualize the New Sale and Resale columns using box plot
"""

# Complete your Code Here


draw_list = ["Resale", "New Sale"]
draw_data = [df.loc[resale_filt, 'price'], df.loc[new_filt, 'price']]

plt.boxplot(x = draw_data)
plt.yscale("log")
plt.xticks([1,2], draw_list)
```

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

In [26]:

```
"""
Visualize the New Sale and Resale columns using stacked Histogram
"""

# Complete your Code Here

## For the Resale

setbins = np.arange(0.5e6, 5e6, 0.1e6)
plt.hist(df_resale["price"], bins = setbins, edgecolor = "black", label = "Resale")
plt.hist
plt.xlabel('Prices (S$)')
plt.ylabel('Frequency')
plt.legend()

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



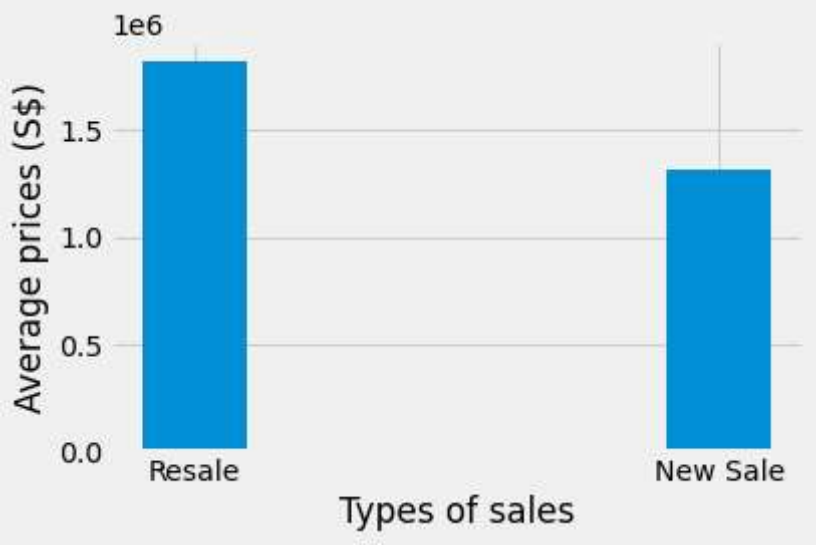
In [27]:

```
## For the New Sale
```

```
setbins = np.arange(0.5e6, 5e6, 0.1e6)
plt.hist(df_new["price"], bins = setbins, edgecolor = "black", label = "New Sale")
plt.hist
plt.xlabel('Prices (S$)')
plt.ylabel('Frequency')
plt.legend()


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

In [41]:

```
# Complete your Code Here
plt.bar(['Resale', 'New Sale'],                                    # x-data
        [df_resale["price"].mean(),
         df_new["price"].mean()],                                  # y-data
         width=0.2)                                                # Width of bars

plt.xlabel('Types of sales')
plt.ylabel('Average prices (S$)')

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

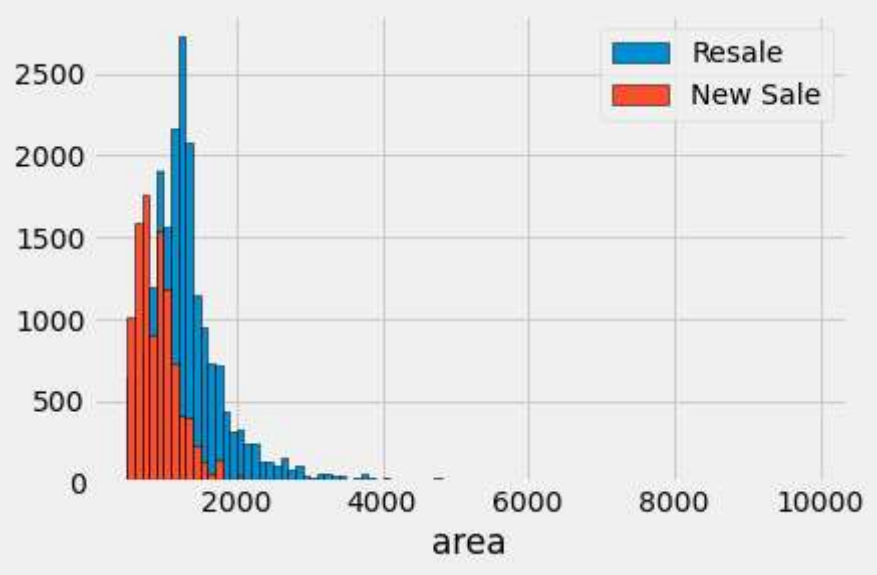


This is a counter-intuitive finding that resale condos are more expensive than the new condos on average. Can you think about a possible reason for this?

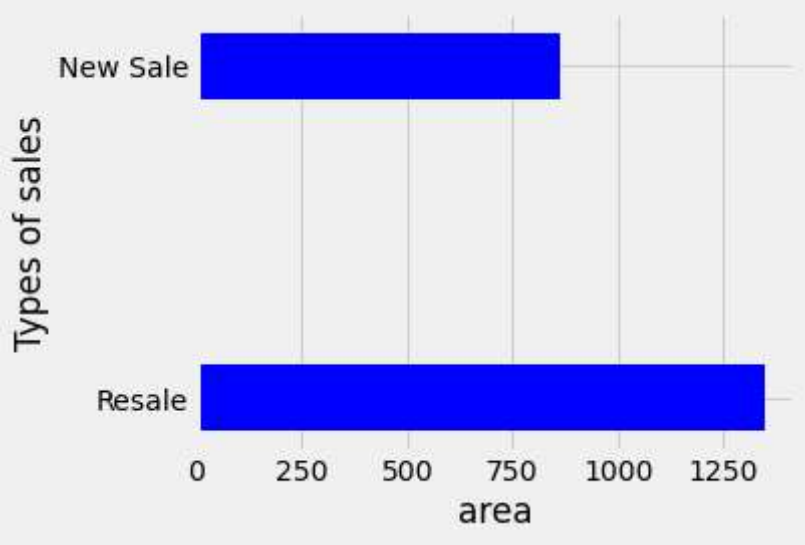
Task 2: Explore the relationship between the condo area and the condo type

Now let's focus on another attribute of a condo in the data set, the area of a condo. Please explore the distribution of the area of a condo by the condo type. Please draw a visualization to present your findings? If you want to draw histograms to explore the distribution of the area of a condo, please set the bins as `np.arange(0.5e3, 10e3, 0.1e3)` .

In [28]:
```
setbins = np.arange(0.5e3, 10e3, 0.1e3)
plt.hist(df_resale["area"], bins = setbins, edgecolor = "black", label = "Resale")
plt.hist(df_new["area"], bins = setbins, edgecolor = "black", label = "New Sale") #comp

plt.xlabel('area')
plt.legend(loc = 'best')
plt.show()
```



In [43]:
```
plt.barh(['Resale', 'New Sale'],
         [df_resale['area'].mean(),df_new['area'].mean()],
         height=0.2, color = "b")      #complete the syntax

plt.ylabel('Types of sales')
plt.xlabel('area')

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

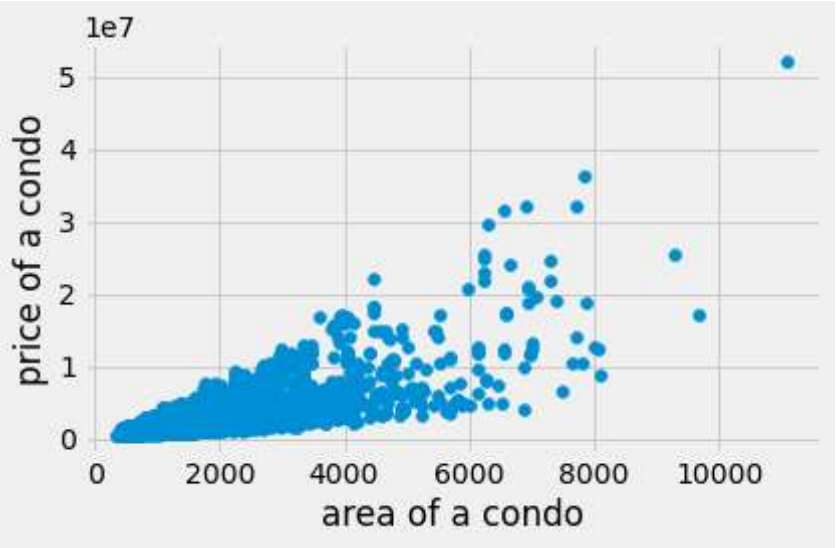


Task 3: Explore the relationship between the condo area and the condo price

explore the relationship between the area of a condo and its price. You can draw a scatter plot to uncover the possible pattern between the two variables.

In [34]:

```
plt.scatter(df.area, df.price)    # complete the syntax

plt.xlabel('area of a condo')
plt.ylabel('price of a condo')

#plt.yscale("log")
plt.tight_layout()
plt.show()
```



From Tasks 1-3, now you should know the relationship found between the condo price and the condo type is not trustworthy. There is a third variable that is related to both the condo price and the condo type. In the above analysis, this third variable is the area of the condo. We call this third variable as the *confounder* or *confounding variable*.

Task 4: Grouping

Now let's explore the ways of controlling for the confounding effect of the area of a condo in the analysis. A useful method is to do a stratified analysis. Since the confounder variable, the area of a condo, is continuous, to simplify the discussion, a discretization is carried out. Please form a new variable in the data set by grouping the condos into 3 different categories in terms of their areas. The three categories are defined as follows:

1. Small: the area of a condo less than 800 square feet
2. Median: the area of a condo between 800 and 1200 square feet
3. Large: the area of a condo larger than 1200 square feet

In [35]:
```
area_small = df["area"] < 800
area_small
```

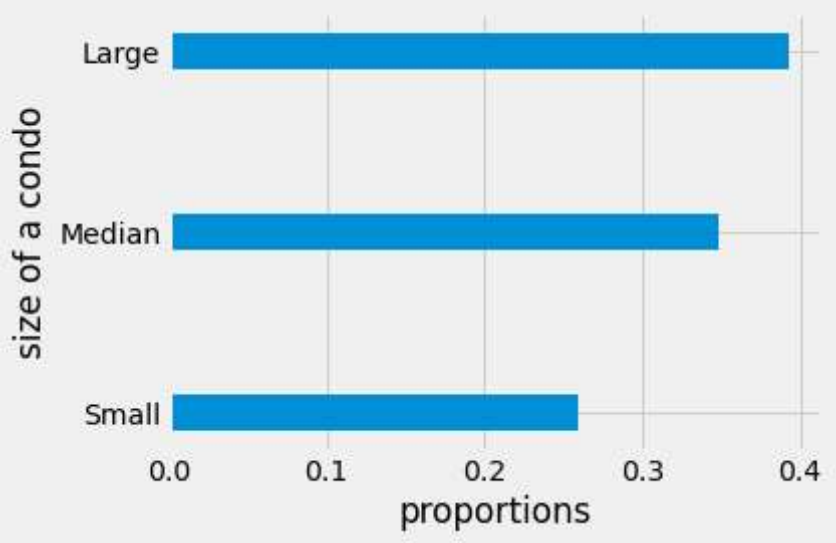
Out[35]:
```
0        False
1         True
2        False
3        False
4        False
         ...
32163    False
32164     True
32165    False
32166    False
32167     True
Name: area, Length: 32168, dtype: bool
```

In [36]:
```
area_median = (df["area"] >= 800) & (df["area"] <= 1200)
area_median
```

Out[36]:
```
0        False
1        False
2        False
3         True
4        False
         ...
32163     True
32164    False
32165    False
32166     True
32167    False
Name: area, Length: 32168, dtype: bool
```

In [39]:
```
area_large = (df["area"] > 1200)

df["area_gp"] = 1 * area_small + 2 * area_median + 3 * area_large
df["area_gp"].unique()

df["area_gp"].value_counts(normalize=True)

# Complete your Code Here
prop_area = df["area_gp"].value_counts(normalize=True)                # complete the sy
plt.barh([1,2,3],prop_area.loc[[1,2,3]], height = 0.2)
```

```
plt.yticks([1,2,3], ['Small','Median','Large'])
plt.xlabel('proportions')
plt.ylabel('size of a condo')

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



In [40]:
```
df["area_gpstr"] = df["area_gp"].map({1:'Small',2:'Median',3:'Large'})
df.head(10)
```

Out[40]:

	name	price	unit_price	district_code	segment	type	area	level	remaining_years
0	SEASCAPE	4388000	2028	4	CCR	Resale	2164	06 to 10	87.0
1	COMMONWEALTH TOWERS	1300000	1887	3	RCR	Resale	689	16 to 20	93.0
2	THE TRILINQ	1755000	1304	5	OCR	Resale	1346	06 to 10	92.0
3	THE CREST	2085000	2201	3	RCR	Resale	947	01 to 05	92.0
4	THE ANCHORAGE	1848888	1468	3	RCR	Resale	1259	01 to 05	999.0
5	MOUNT FABER LODGE	4400000	1188	4	RCR	Resale	3703	06 to 10	999.0
6	BLUE HORIZON	990000	1022	5	OCR	Resale	969	21 to 25	80.0

	name	price	unit_price	district_code	segment	type	area	level	remaining_years
7	DOVER PARKVIEW	1088000	1162	5	RCR	Resale	936	06 to 10	73.0
8	CARIBBEAN AT KEPPEL BAY	1470000	1751	4	RCR	Resale	840	06 to 10	79.0
9	THE INTERLACE	4550000	868	4	RCR	Resale	5242	16 to 20	89.0

Task 5: The relationship between the condo price and the condo type, controlling for the condo area

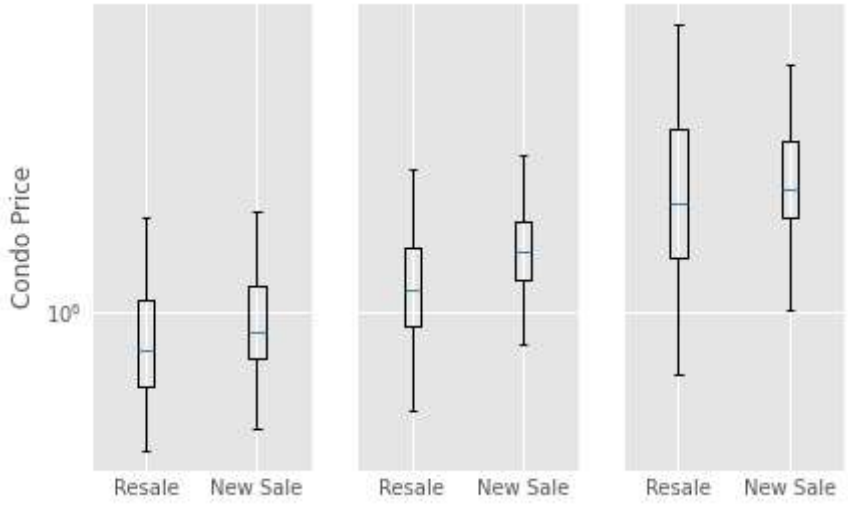
With the new categorical variable and the condo type, 6 possible combinations of the two variables can be generated to represent a condo's profile. For example, we can define a condo's profile as small and resale. Now please explore the relationship between the condo price and condo type by 3 different categories of the condo area. What is your conclusion?

In [ ]:
```
### Use below code for filtering the dataste
```

In [ ]:
```
"""
size_set = ["Small", "Median", "Large"]
type_set = ["Resale", 'New Sale']

filt_SR = (df['area_gpstr'] == size_set[0]) & (df['type'] == type_set[0])
filt_MR = (df['area_gpstr'] == size_set[1]) & (df['type'] == type_set[0])
filt_LR = (df['area_gpstr'] == size_set[2]) & (df['type'] == type_set[0])

filt_SN = (df['area_gpstr'] == size_set[0]) & (df['type'] == type_set[1])
filt_MN = (df['area_gpstr'] == size_set[1]) & (df['type'] == type_set[1])
filt_LN = (df['area_gpstr'] == size_set[2]) & (df['type'] == type_set[1])

"""
```

In [44]:
```
size_set = ["Small", "Median", "Large"]
type_set = ["Resale", 'New Sale']

filt_SR = (df['area_gpstr'] == size_set[0]) & (df['type'] == type_set[0])
filt_MR = (df['area_gpstr'] == size_set[1]) & (df['type'] == type_set[0])
filt_LR = (df['area_gpstr'] == size_set[2]) & (df['type'] == type_set[0])

filt_SN = (df['area_gpstr'] == size_set[0]) & (df['type'] == type_set[1])
filt_MN = (df['area_gpstr'] == size_set[1]) & (df['type'] == type_set[1])
filt_LN = (df['area_gpstr'] == size_set[2]) & (df['type'] == type_set[1])

# Complete your Code Here

plt.style.use('ggplot')
```

```
f, (ax1,ax2,ax3) = plt.subplots(1, 3, sharey = True)

data_small = [df.loc[filt_SR,"price"],df.loc[filt_SN,"price"]]
ax1.boxplot(x = data_small, showfliers=False)
ax1.set_xticks([1,2])
ax1.set_xticklabels(type_set)
ax1.set_yscale('log')
ax1.set_ylabel('Condo Price')


data_median = [df.loc[filt_MR,"price"],df.loc[filt_MN,"price"]]             # Complete t
ax2.boxplot(x = data_median, showfliers=False)                              # Complete t
ax2.set_xticks([1,2])
ax2.set_xticklabels(type_set)
ax2.set_yscale('log')


data_large = [df.loc[filt_LR,"price"],df.loc[filt_LN,"price"]]             # Complete th
ax3.boxplot(x = data_large, showfliers=False)                               # Complete th
ax3.set_xticks([1,2])
ax3.set_xticklabels(type_set)
ax3.set_yscale('log')

plt.show()
```



In [ ]: