Python Sample Code_Word Cloud

February 10, 2020

This is a sample Python code to generate a word cloud, an efficient way to perform data analysis, pull insights, and visualize the findings.

Shih-Hung (Angela) Ma

[1]:
```
##to import necessary packages

import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import datetime, nltk, warnings
import matplotlib.cm as cm
import itertools
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

from os import path
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/angelama032697/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /Users/angelama032697/nltk_data…
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

[2]:
```
##to extract data from the data set

data_f = pd.read_csv('Florida_all_data.csv')
```

[3]:
```
##to retain data that have a positive value in price

data_f = data_f[data_f.Price_per_sqft>=0]
```

[4]:
```
data_f.shape
```

[4]:
```
(2963, 74)
```

[5]:
```
##since the goal is to analyze the relationship between amenities and price per␣
 ↪sqft,
##we only need to use these two columns

data_f = data_f[['Amenities', 'Price_per_sqft']]
```

[6]:
```
#to drop rows that has no value in Amenities

data_f = data_f.dropna()
```

[7]:
```
data_f.Price_per_sqft.sort_values(ascending=False).head()
```

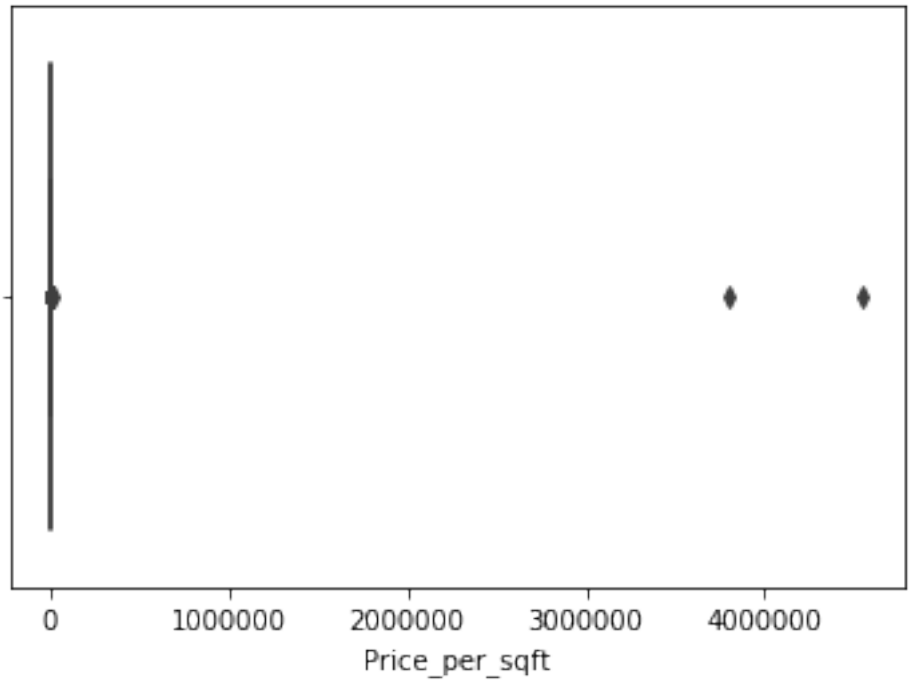
[7]:
```
6361    4551300.00
6362    3800000.00
469       19200.00
3020       2699.06
3021       2692.20
Name: Price_per_sqft, dtype: float64
```

[8]:
```
data_f.Price_per_sqft.describe().round()
```

[8]:
```
count       2661.0
mean        3241.0
std       114916.0
min            0.0
25%           45.0
50%           73.0
75%          114.0
max      4551300.0
Name: Price_per_sqft, dtype: float64
```

[9]:
```
sns.boxplot(x=data_f['Price_per_sqft'])
```

[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x12303a748>
```

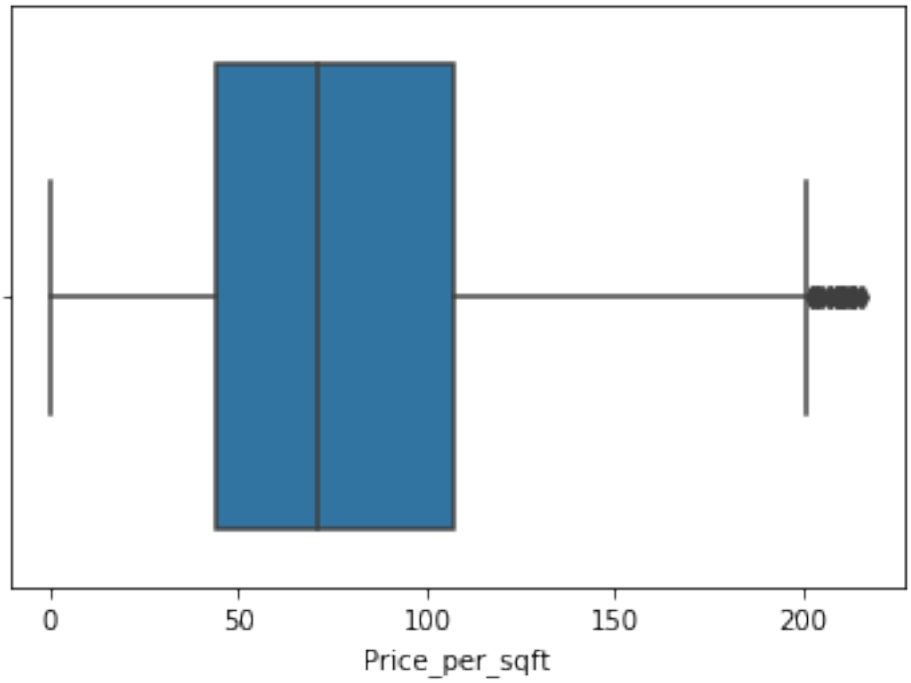


[10]:
```
##Since the above graph shows extreme values, to remove outliers

data_f = data_f[data_f.Price_per_sqft<=216]
```

[11]:
```
sns.boxplot(x=data_f['Price_per_sqft'])
```

[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x122fdc048>
```

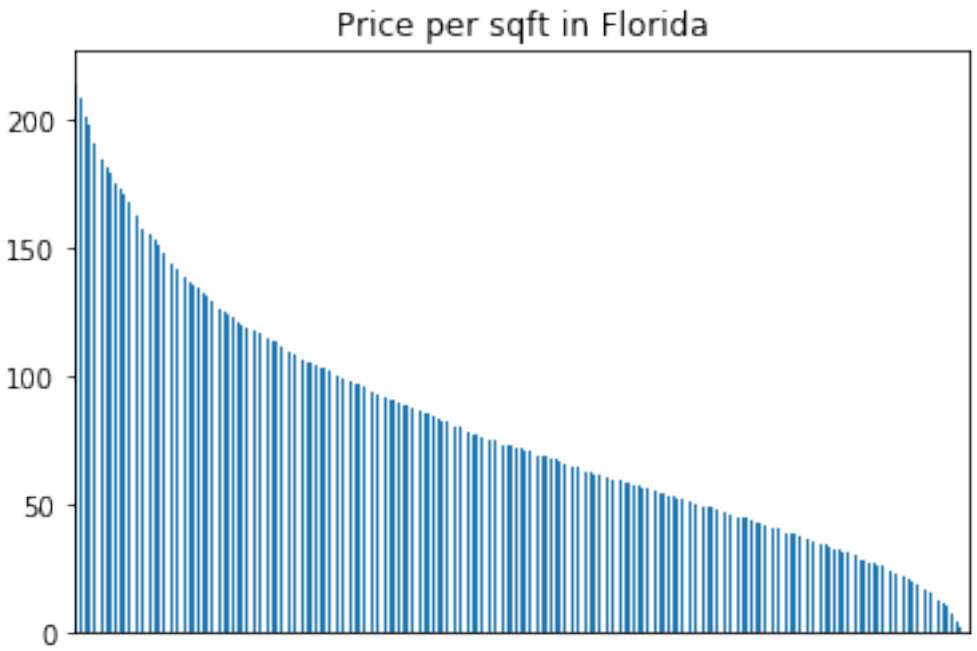


[12]:
```
data_f.shape

#there are 2535 records left
```

[12]: (2549, 2)

[13]:
```
plt.figure()
data_f.Price_per_sqft.sort_values(ascending=False).plot.bar()
plt.xticks([])
ax = plt.gca()
ax.set_title('Price per sqft in Florida')
plt.show()
```



[14]:
```
text = " ".join(review for review in data_f.Amenities)
print ("There are {} words in the combination of all review.".format(len(text)))
```

```
There are 360302 words in the combination of all review.
```

[15]:
```
# Generate a word cloud image
wordcloud = WordCloud(background_color="white").generate(text)

# Display the generated image:
# the matplotlib way:
plt.figure(figsize = (12, 12), facecolor = None)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Alternative code

[16]:
```
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in data_f.Amenities:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split(',')

    tokens = str(tokens).replace(' ', '')

def show_wordcloud(tokens, title = None):
        wordcloud = WordCloud(background_color ='white',
                stopwords = stopwords,
                min_font_size = 10,
                max_words = 10).generate(tokens)


# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
```

```
plt.show()
```

Alternative code: to make the word cloud more clear to see, reset max_words

[17]:
```
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in data_f.Amenities:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

wordcloud = WordCloud(background_color ='white',
                stopwords = stopwords,
                min_font_size = 10,
                max_words = 200).generate(text)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
```

```
plt.tight_layout(pad = 0)

plt.show()
```

[18]:
```
##mapping the value of 1 to properties having a certain amenity and the value␣
 ↪of 0 to properties that do now own the amenity
##in order to perform data analysis on the four most common words in amenities

data_f['fitnesscenter'] = data_f['Amenities'].str.contains(r'Fitness', na=True)
data_f['fitnesscenter'] = data_f['fitnesscenter'].map({True: 1, False: 0})

data_f['businesscenter'] = data_f['Amenities'].str.contains(r'Business',␣
 ↪na=True)
data_f['businesscenter'] = data_f['businesscenter'].map({True: 1, False: 0})

data_f['laundryfacilities'] = data_f['Amenities'].str.contains(r'Laundry',␣
 ↪na=True)
data_f['laundryfacilities'] = data_f['laundryfacilities'].map({True: 1, False:␣
 ↪0})

data_f['propertymanager'] = data_f['Amenities'].str.contains(r'Manager',␣
 ↪na=True)
data_f['propertymanager'] = data_f['propertymanager'].map({True: 1, False: 0})
```

[19]:
```
data_f.head()
```

[19]:

	Amenities	Price_per_sqft	\
11	Business Center, Courtyard, Fitness Center, Gr…	152.81	
12	Business Center, Courtyard, Fitness Center, Gr…	56.01	
13	Fitness Center, Laundry Facilities, Gated, Gam…	52.06	
21	Business Center, Controlled Access, Clubhouse,…	56.90	
22	24 Hour Access, Clubhouse, Courtyard, Fitness …	87.77	

	fitnesscenter	businesscenter	laundryfacilities	propertymanager
11	1	1	0	0
12	1	1	0	0
13	1	0	1	0
21	1	1	1	1
22	1	0	1	0

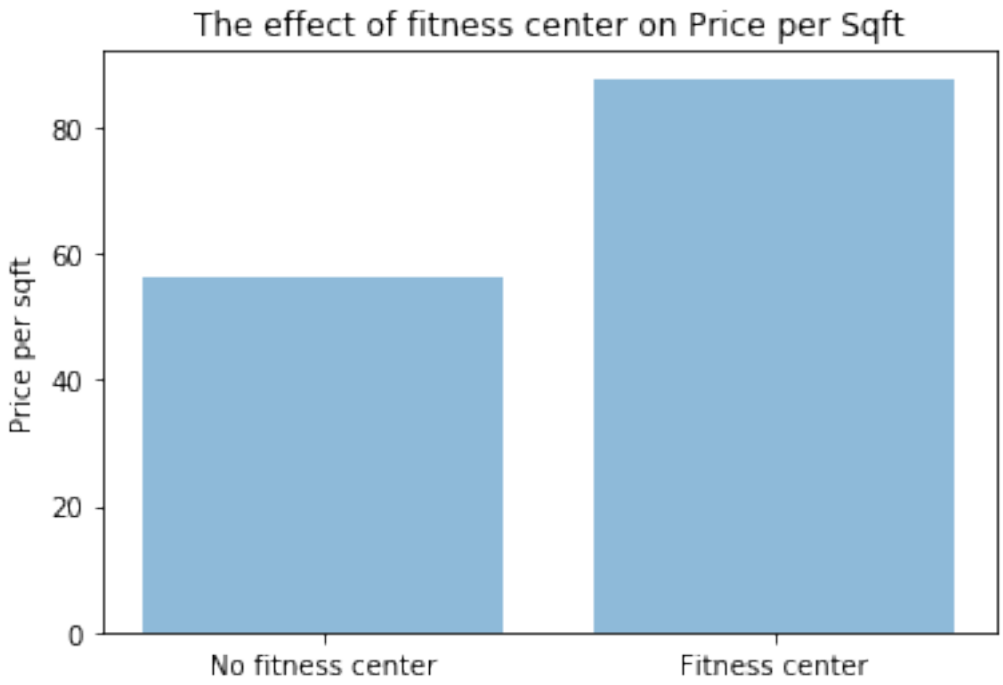
The below is to visualize the effect of these four amenities on price per sqft.

[20]:
```
data_f.groupby('fitnesscenter')['Price_per_sqft'].mean()
```

[20]:
```
fitnesscenter
0    56.332766
1    87.736663
Name: Price_per_sqft, dtype: float64
```

[21]:
```
objects = ('No fitness center', 'Fitness center')
y_pos = np.arange(len(objects))
performance = [56.332766,87.736663]
plt.bar(y_pos, performance, align='center', alpha=0.5)
plt.xticks(y_pos, objects)
plt.ylabel('Price per sqft')
plt.title('The effect of fitness center on Price per Sqft')

plt.show()
```

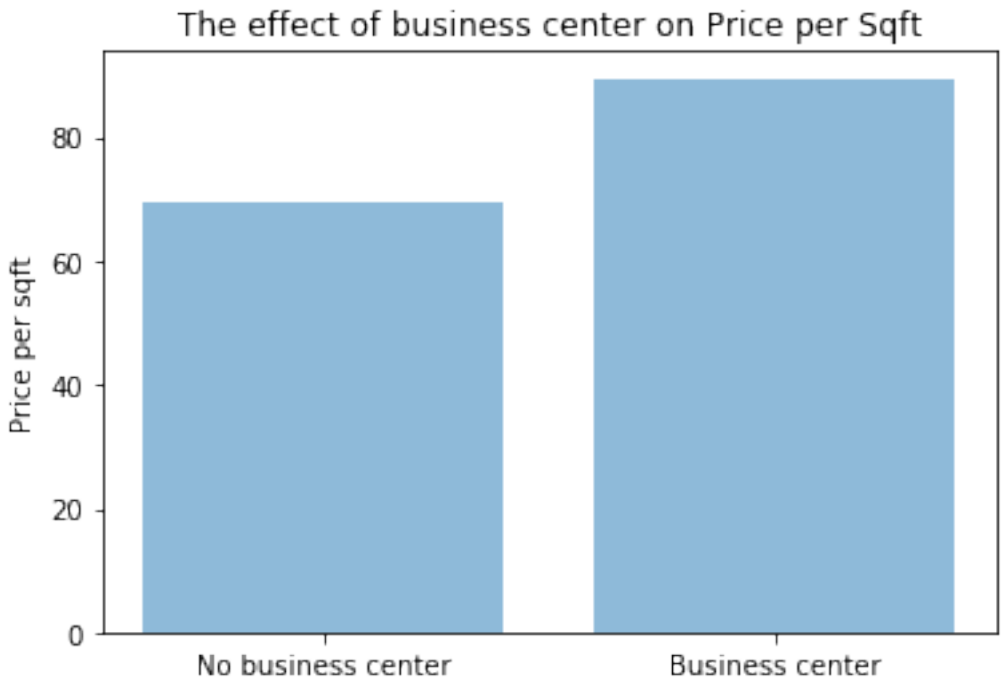


[22]:
```
data_f.groupby('businesscenter')['Price_per_sqft'].mean()
```

[22]:
```
businesscenter
0    69.506978
1    89.458308
Name: Price_per_sqft, dtype: float64
```

[23]:
```
objects = ('No business center', 'Business center')
y_pos = np.arange(len(objects))
performance = [69.506978,89.458308]
plt.bar(y_pos, performance, align='center', alpha=0.5)
plt.xticks(y_pos, objects)
plt.ylabel('Price per sqft')
plt.title('The effect of business center on Price per Sqft')

plt.show()
```

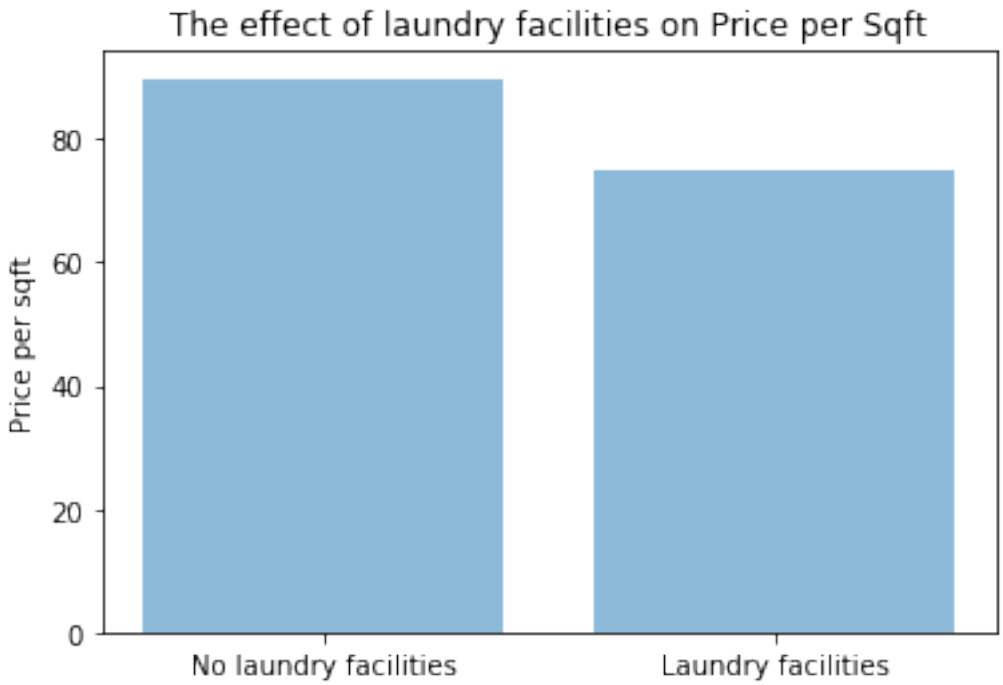


[24]:
```
data_f.groupby('laundryfacilities')['Price_per_sqft'].mean()
```

[24]:
```
laundryfacilities
0    89.814080
1    74.845474
Name: Price_per_sqft, dtype: float64
```

[25]:
```
objects = ('No laundry facilities', 'Laundry facilities')
y_pos = np.arange(len(objects))
performance = [89.814080,74.845474]
plt.bar(y_pos, performance, align='center', alpha=0.5)
plt.xticks(y_pos, objects)
plt.ylabel('Price per sqft')
plt.title('The effect of laundry facilities on Price per Sqft')

plt.show()
```

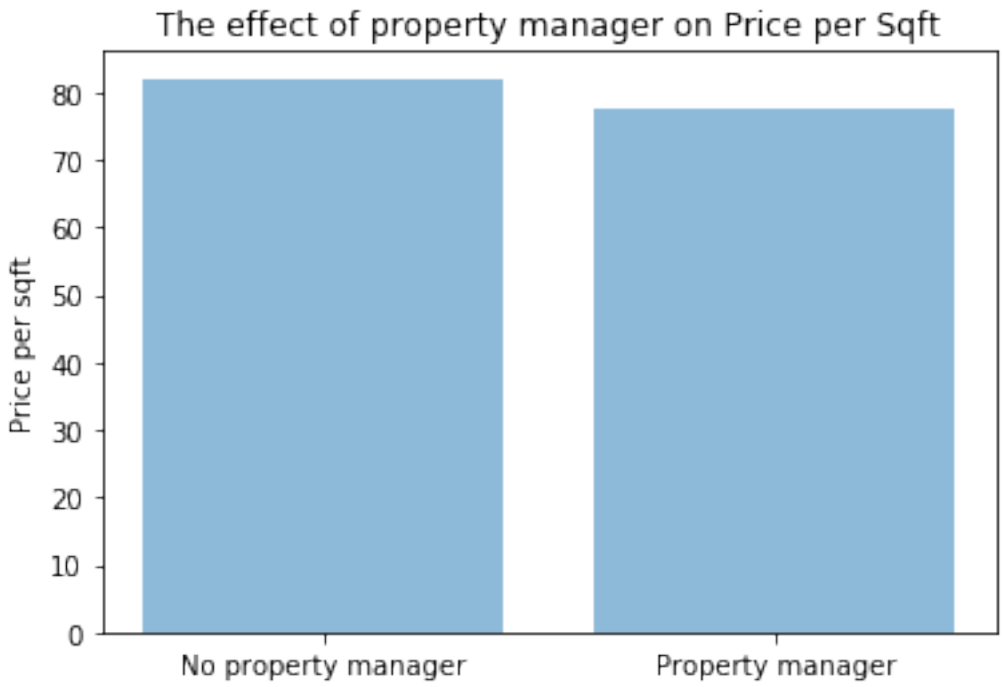


[26]:
```
data_f.groupby('propertymanager')['Price_per_sqft'].mean()
```

[26]:
```
propertymanager
0    82.041958
1    77.691192
Name: Price_per_sqft, dtype: float64
```

[27]:
```
objects = ('No property manager', 'Property manager')
y_pos = np.arange(len(objects))
performance = [82.041958,77.691192]
plt.bar(y_pos, performance, align='center', alpha=0.5)
plt.xticks(y_pos, objects)
plt.ylabel('Price per sqft')
plt.title('The effect of property manager on Price per Sqft')

plt.show()
```


The effect of property manager on Price per Sqft
Price per sqft
0
10
20
30
40
50
60
70
80
No property manager
Property manager

[ ]: