Project description

The purpose of the project is to select locations to open a small chain (or cafe) in Los Angeles. Within the project we will conduct market analysis, prepare recommendations for the selection of a location, and decide on the type of cafe (or chain) that will be better for investors (or owners) within the scope of the current market situation.

Data preprocessing

Importing the libraries

```
import numpy as np, pandas as pd, seaborn
from sklearn.feature_extraction import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
pio.renderers.default = 'svg'
from sklearn import preprocessing
from sklearn import metrics
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import DBSCAN
from sklearn.cluster import MeanShift
from sklearn.cluster import estimate_bandwidth
from sklearn.cluster import SpectralClustering
from sklearn.cluster import AffinityPropagation
from sklearn.cluster import Birch
from sklearn.cluster import OPTICS
from sklearn import cluster, mixture
from sklearn.metrics import silhouette_score, silhouette_samples
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import Normalizer
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from scipy import stats
from scipy.stats import norm, skew
from scipy.stats.stats import pearsonr
from scipy.stats.stats import kurtosis
from scipy.stats.stats import kurtosistest
from scipy.stats.stats import ttest_ind
from scipy.stats.stats import ttest_rel
from scipy.stats.stats import chi2_contingency
from scipy.stats.stats import f_oneway
from scipy.stats.stats import levene
from scipy.stats.stats import shapiro
from scipy.stats.stats import kruskal
from scipy.stats.stats import mannwhitneyu
from scipy.stats.stats import wilcoxon
from scipy.stats.stats import ttest_ind_from_stats
from scipy.stats.stats import power_divergence
from scipy.stats.stats import binom_test
from scipy.stats.stats import gaussian_kde
from scipy.stats.stats import kstest
from scipy.stats.stats import ttest_1samp
from scipy.stats.stats import ttest_rel
from scipy.stats.stats import chisquare
from scipy.stats.stats import power_divergence
import math as mt
import sys
import os, warnings
warnings.filterwidth('ignore')
pd.set_option('display.max_colwidth', -1)
pd.set_option('display.max_columns', 50)
pd.set_option('display.max_rows', 50)
%matplotlib inline
%matplotlib notebook
%matplotlib inline
%matplotlib notebook
%matplotlib inline
%matplotlib notebook
```

```
In [1]: inspect_data
```

```
Out[1]:
   id                object name                 address  zipcode  object type  number
0   0  CHARLIE'S COFFEE SHOP  2700 N DIXIE DRIVE 45414    45414         food     20
1   1              CHIPOTLE  2700 N DIXIE DRIVE 45414    45414         food     21
2   2          JIMMY JOHNS'  2600 CORPORATE EXCH DR 45…    45431         food     22
3   3          TEAM FUN ZONE  1010 N MAIN ST 45390 PAL…    45390   Recreation     23
4   4                  PALLA  1000 DAYTON PARK DR 45431    45431   Recreation     24
5   5                                                                           25
6   6  ROSE LANE  1012 OF LIFE DR  1012 OF LIFE DR 45…    45431   Recreation     26
7   7  ROSE LANE  1012 OF LIFE DR  1012 OF LIFE DR 45…    45431   Recreation     27
8   8          JUST BORN  1015 FOUNDATION DR 45431    45431   Recreation     28
9   9  ROSE LANE  1020 GREEN DRIVE  1020 GREEN DR 45…    45431   Recreation     29
10  10  ROSE LANE  1020 GREEN DRIVE  1020 GREEN DR 45…    45431   Recreation     30
```

```
RAW rows: 11 columns
```

```
In [2]: count_object = inspect_data[['object type']].value_counts()
        print("Number of unique object types:", len(count_object))
        print("\n Frequency of each object type:")
        print(count_object.head(10))
        count_object.head(10).plot(kind='bar', figsize=(8, 4))
```

```
Out[2]:
Number of unique object types: 7

 Frequency of each object type:
object type
food          7
Recreation    3
Name: count, dtype: int64
```

```
In [3]: count_object_name = inspect_data[['object type', 'name']].value_counts()
        print("Number of unique object types:", len(count_object_name))
        print("\n Frequency of each object type and name:")
        print(count_object_name.head(10))
```

```
Out[3]:
Number of unique object types: 11

 Frequency of each object type and name:
object type  name
food         CHARLIE'S COFFEE SHOP    1
             CHIPOTLE                 1
             JIMMY JOHNS'             1
             PALLA                    1
             ROSE LANE                1
             ROSE LANE                1
             ROSE LANE                1
             ROSE LANE                1
             ROSE LANE                1
             TEAM FUN ZONE            1
Name: count, dtype: int64
```

```
In [4]: # Top 10 Figures
        top_10_figures = inspect_data[['object type']].value_counts().head(10)
        print("Top 10 Figures:")
        print(top_10_figures)
        top_10_figures.plot(kind='bar', figsize=(8, 4))
        plt.title('The number of establishments')
        plt.xlabel('Restaurant Type')
        plt.ylabel('Number of establishments')
        plt.show()
```

```
Out[4]:
Top 10 Figures:
object type
food          7
Recreation    3
Name: count, dtype: int64
```

Most Eyes Restaurant and **Meet the Restaurant** have the most establishments in it according to our data.

The number of streets that only have one restaurant

```
In [5]: count_the_number_of_streets_that_only_have_one_restaurant = inspect_data[['address']].value_counts()
        print("The number of streets that only have one restaurant: ", count_the_number_of_streets_that_only_have_one_restaurant.shape[0])
        # Top 10 Figures Most_Eyes_Restaurant_Meet_the_Restaurant
        top_10_figures = inspect_data[['address']].value_counts().head(10)
        print("Top 10 Figures:")
        top_10_figures.plot(kind='bar', figsize=(8, 4))
        plt.title('The number of streets that only have one restaurant')
        plt.xlabel('Restaurant')
        plt.ylabel('Number of streets')
        plt.show()
```

```
Out[5]:
The number of streets that only have one restaurant:  9
Top 10 Figures:
```


Most of the streets in it have only one restaurant according to our data.

The number of seats for streets with a lot of establishments

```
In [6]: # Top 10 Figures Most_Eyes_Restaurant_Meet_the_Restaurant
        top_10_figures = inspect_data[['address']].value_counts().head(10)
        print("Top 10 Figures:")
        top_10_figures.plot(kind='bar', figsize=(8, 4))
        plt.title('The number of seats for streets with a lot of establishments')
        plt.xlabel('Restaurant')
        plt.ylabel('Number of seats')
        plt.show()
```

```
Out[6]:
Top 10 Figures:
```


The average number of seats for the top ten streets is little higher than the average for entire dataset. Mostly more than 40 seats.

```
In [7]: # Top 10 Figures Most_Eyes_Restaurant_Meet_the_Restaurant
        top_10_figures = inspect_data[['address']].value_counts().head(10)
        print("Top 10 Figures:")
        top_10_figures.plot(kind='bar', figsize=(8, 4))
        plt.title('The distribution of the number of seats')
        plt.xlabel('Restaurant')
        plt.ylabel('Number of seats')
        plt.show()
```

```
Out[7]:
Top 10 Figures:
```


number
number
number