BDA 507

TERM PROJECT

REPORT

FERAY ECE TOPCU

MEF University

ISTANBUL

2018

1. INTRODUCTION

As a coffee addict, I really appreciate to Starbucks because I can get a cup of coffee whenever or wherever I want. Because of easy accessibility of Starbucks, I've wondered how many Starbucks exist all over the world and especially in Turkey. I have learnt the company now operates more than 24000 retail stores in 70 countries. So, I started to search for a data collection about location of all Starbucks around the world to understand the details.

Fortunately, after some research, I've found a dataset on Kaggle.

Dataset

It can be accessible on this url:

https://www.kaggle.com/starbucks/store-locations/data

This dataset includes a record for every Starbucks or subsidiary store location currently in operation as of February 2017 and contains 25600 rows and 13 columns; it is saved as csv file.

Column Metadata

Column details can be examined on the table as below:

Column Name	Column Definition	Data Type
Brand	Starbucks	String
Store Number	Unique number for each store.	String
Store Name	Names according to their location.	String
Ownership Type	Company owned, licenced or franchise etc.	String
Street Address	Real address of the store.	String
City	Which city store is located. (Name of City)	String
State/Province	Which state/province store is located.	String
Country	Which country store is located. (ISO 3166-1 alpha-2 standa	String
Postcode	Postcode of the store.	String
Phone Number	Phone Number of the store.	Numeric
Timezone	Timezone of the store. (GMT Standard.)	String
Longitude	Longitude of the store.	Numeric
Latitude	Latitude of the store.	Numeric

2. Exploration of the Dataset

- Read CSV file into a dataframe to analyze it.

```
filename = 'directory2.csv'
data = pd.read_csv(filename)
```

- Firstly, **data.info()** is used to check datatype of each column. As it is seen below, just longitude and latitude column is float and other columns are categorical.

```
>>> data.info()  #25600
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25600 entries, 0 to 25599
Data columns (total 13 columns):
Brand             25600 non-null object
Store Number      25600 non-null object
Store Name        25600 non-null object
Ownership Type    25600 non-null object
Street Address    25598 non-null object
City              25585 non-null object
State/Province    25600 non-null object
Country           25600 non-null object
Postcode          24078 non-null object
Phone Number      18739 non-null object
Timezone          25600 non-null object
Longitude         25599 non-null float64
Latitude          25599 non-null float64
dtypes: float64(2), object(11)
memory usage: 2.5+ MB
```

- So, Longitude and Latitude column can be described with **describe()** and observe the columns statistics. When **data.describe()** command is executed, the results are as below:

```
          Longitude      Latitude
count  25599.000000  25599.000000
mean     -27.872234     34.793016
std       96.844046     13.342332
min     -159.460000    -46.410000
25%     -104.665000     31.240000
50%      -79.350000     36.750000
75%      100.630000     41.570000
max      176.920000     64.850000
```

- For describing categorical columns below code chunk is executed:

```
categorical = data.dtypes[data.dtypes == "object"].index
print(data[categorical].describe())
```

The result of this:

	Brand	Store Numbe	Store Name	Ownership Type	St Address	City	State/Pro	Country	PostCode	PhNm	TimeZone
Count	25600	25600	25600	25600	25598	25585	25600	25600	24078	18739	25600
Unique	4	25599	25364	4	25353	5455	338	73	18887	18402	101
Top	Starbucks	19773-160973	Starbucks	Company Owned	Guard Post 8	上海市	CA	US	0	4.20E+11	GMT-05:00 America/New_York
Freq	25249	2	224	11932	11	542	2821	13608	101	27	4889

So, as it is seen on the result table the dataset mainly is composed by Company Owned, Starbucks stores.

- Before start to explore data more, unnecessary columns should be deleted:

```
del data["Postcode"]
del data["Phone Number"]
del data["Street Address"]
```

- The result of describing categorical columns show that there are different brands in dataset. Check what are they with the following code chunk:

```
unique_brand = data.groupby('Brand')["Store Number"].nunique()
print("Unique Brands:\n",unique_brand)
```

The result of this:

```
Unique Brands:
 Brand
Coffee House Holdings        1
Evolution Fresh              2
Starbucks                25248
Teavana                    348
```

- it is seen that 25248 stores are belongs to Starbucks brand. So, just take them to analyze.

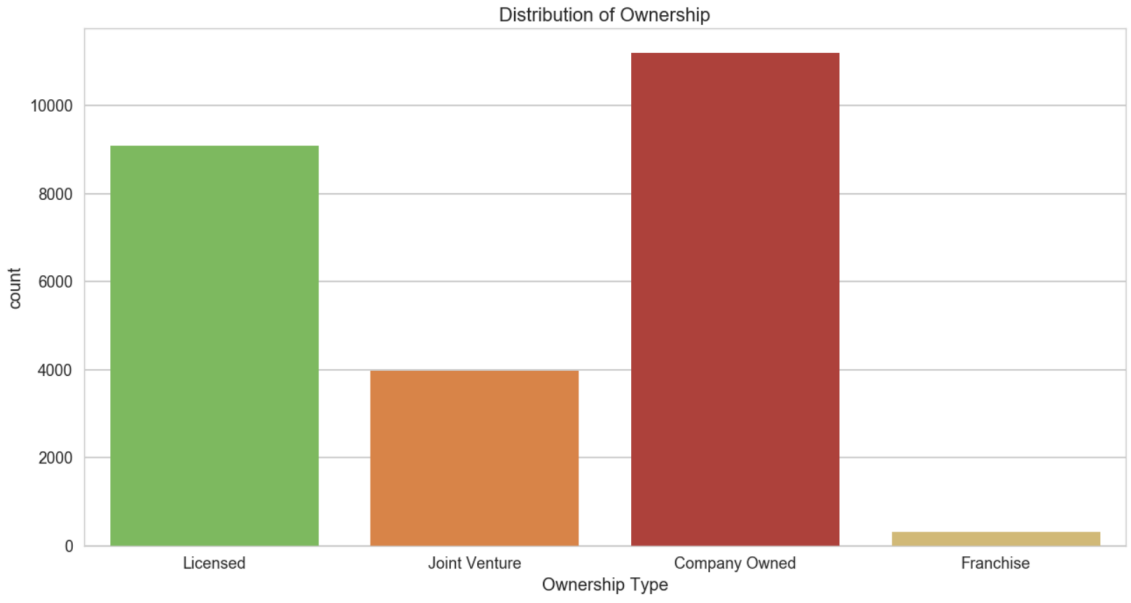
```
import numpy as np
sb_index=np.where(data["Brand"]=="Starbucks")
data=data.loc[sb_index]
print(data.info())
```

The result of **data.info()** after deletion of 3 columns and non-Starbucks brand stores:

```
Data columns (total 10 columns):
Brand             25249 non-null object
Store Number      25249 non-null object
Store Name        25249 non-null object
Ownership Type    25249 non-null object
City              25234 non-null object
State/Province    25249 non-null object
Country           25249 non-null object
Timezone          25249 non-null object
Longitude         25248 non-null float64
Latitude          25248 non-null float64
```

- The result of describing categorical columns show also that there are different Ownership Type in dataset. Show them with a graph:

```
import seaborn as sns
type_colors = ['#78C850','#F08030','#C03028','#E0C068']
sns.set(style="whitegrid", context="talk")
sns.countplot(x='Ownership Type', data=data, palette=type_colors)
plt.title("Distribution of Ownership")
```



As seen on the graph, Company Owned stores are more than 10000 and Licenced are more than 8000. Additionally, Franchise stores are remarkably few.

- The aim of this analysis, explore the number of stores according to countries and cities. Therefore, if there are null values on City and Country columns, it should be cleaned. Check the nullity for each column:

```
print("Nullity Check:\n",pd.isnull(data).any())
print("Sum of null values for each column:\n",pd.isnull(data).sum())
```

```
Nullity Check:
Brand             False
Store Number      False
Store Name        False
Ownership Type    False
City               True
State/Province    False
Country           False
Timezone          False
Longitude          True
Latitude           True
```

```
Sum of null values:
Brand              0
Store Number       0
Store Name         0
Ownership Type     0
City              15
State/Province     0
Country            0
Timezone           0
Longitude          1
Latitude           1
```

As shown above; there are null values in City, Longitude and Latitude columns but they are few when it is compared to row count. However; in any case, City and Country column should not have any null or corrupted values.

- So, clean the null values from City, Longitude and Latitude column and move on with clean data without any nullity:

```
missing= np.where(data["City"].isnull() == True)
data = data.drop(data.index[missing[0]])
missing= np.where(data["Longitude"].isnull() == True)
data = data.drop(data.index[missing[0]])
missing= np.where(data["Latitude"].isnull() == True)
data = data.drop(data.index[missing[0]])
print("Nullity Check:\n",pd.isnull(data).any())
```

```
Nullity Check:
Brand             False
Store Number      False
Store Name        False
Ownership Type    False
City              False
State/Province    False
Country           False
Timezone          False
Longitude         False
Latitude          False
```

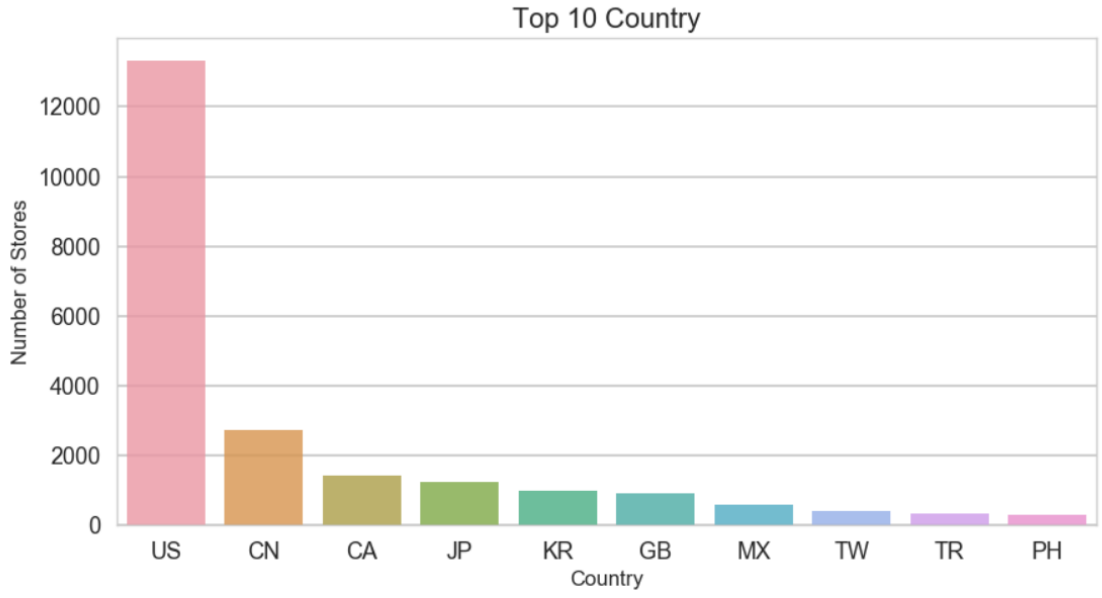
- Let's start to examine the number of stores all around the world. First, find the maximum and minimum number of stores according to countries.

```
country_count = data[['Country','City']].groupby(['Country'])['City'] \
                                .count() \
                                .reset_index(name='count') \
                                .sort_values(['count'], ascending=False) #\

print("Country number of stores: ",country_count['count'].count())
print("Average of stores: ",country_count['count'].mean())
print("Max number of stores: ",country_count['count'].max())
print("Min number of stores: ",country_count['count'].min())
```

Country number of stores: 73
Average of stores: 345.657534247
Max number of stores: 13311
Min number of stores: 1

- Plot a graph with top 10 country according to number of stores:

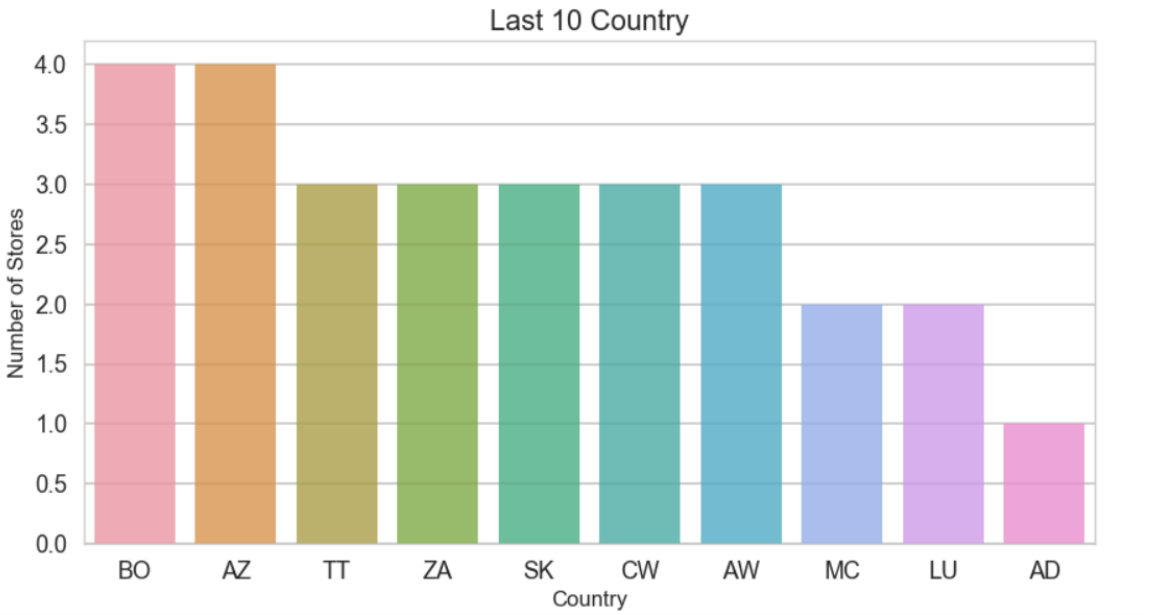
```
country_count=country_count.head(10)
plt.figure(figsize=(10,5))
sns.barplot(country_count['Country'], country_count['count'], alpha=0.8)
plt.title('Top 10 Country')
plt.ylabel('Number of Stores', fontsize=12)
plt.xlabel('Country', fontsize=12)
plt.show()
```



* The explanation of country codes in order; United States, Canada, Japan, South Korea, Great Britain, Mexico, Taiwan, Turkey and Philippines.

According to graph, US is the first with a big difference and has 13111 stores. Canada is following US with more than 2000 stores. Turkey is the 9th country which has the most stores. As an interesting point, Philippines is 10th on this list.

- Let's look at the last 10 countries on this list:



* The explanation of country codes in order; Bolivia, Azerbaijan, Trinidad and Tobago, South Africa, Slovakia, Curaçao, Aruba, Monaco, Luxembourg, Andorra.

Due to the graph; Starbucks tries to be everywhere all around the World at least with one store.

- This is just about curiosity; Is Italy in the general list?

```
sb_index=np.where(country_count['Country']=="IT")
x=country_count.loc[sb_index]
print(x)
```

No way; Italy does not have any Starbucks.

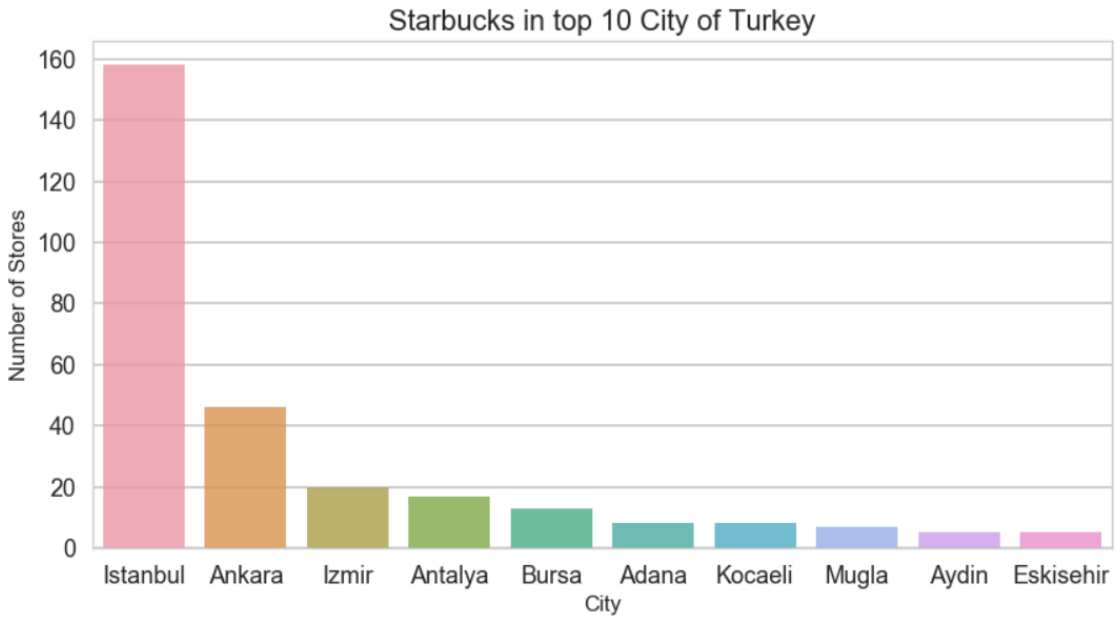
- Let's start to examine the number of stores in Turkey. First, find the maximum and minimum number of stores according to cities.

```
tr_data = pd.DataFrame(data.loc[data['Country'] == 'TR'])

city_count = tr_data[['City']].groupby(['City'])['City'] \
                              .count() \
                              .reset_index(name='count') \
                              .sort_values(['count'], ascending=False)

print("Country number of stores: ",city_count['count'].count())
print("Average of stores: ",city_count['count'].mean())
print("Max number of stores: ",city_count['count'].max())
print("Min number of stores: ",city_count['count'].min())
```

Country number of stores: 27
Average of stores: 11.6428571429
Max number of stores: 158
Min number of stores: 1

- Plot a graph with top 10 country according to number of stores:

```
city_count=city_count.head(10)
plt.figure(figsize=(10,5))
sns.barplot(city_count['City'], city_count['count'], alpha=0.8)
plt.title('Starbucks in top 10 City of Turkey')
plt.ylabel('Number of Stores', fontsize=12)
plt.xlabel('City', fontsize=12)
plt.show()
```



According to graph; Istanbul has 158 Starbucks stores. Ankara is following Istanbul with more than 40 stores. Additionally, It is seen on graph, the number of stores is related with city size in Turkey.

- Look at the last cities in Turkey according to number of Starbucks stores:

```
city_count=city_count.tail(10)
print(city_count)
```

City	count
Izmit	2
Diyarbakir	2
Konya	2
Canakkale	2
Sakarya	2
Bilecik	2
Isparta	1
Denizli	1
Bolu	1
Tekirdag	1

According to this list, Starbucks tries to be everywhere also in Turkey at least with one store.

3. Conclusion

The dataset has 25600 rows and 13 columns as bulk data. There are 25249 rows with Starbucks brand, so I eliminated the other Brands. There are null values on City column so I deleted them for a good descriptive analysis.

As a conclusion of this analysis, we can say Starbucks spreads around the world with its stores in 73 countries. Its largest castle is US which has 13111 stores in 2017. Canada and Japan are following US with around 2000 stores. Turkey is 9^{th} on this list and Interestingly, Philippines is also in top 10 Country List. Additionally, the last 10 country is a proof of the spread policy to the world because although they are the smallest countries in the world, they have Starbucks.

When we look at Turkey, the situation is same; there is Starbucks in 27 cities in Turkey. Istanbul has the most stores (158) and Ankara is following Istanbul with around 40 stores. The number of stores is related with city size in Turkey.

As a future work; this dataset and descriptive analysis can be used for prediction about where will the next Starbucks be opened.

To sum up, Starbucks dataset explain the strategy of spreading around the world and it can be used for prediction about where will be the next Starbucks.

References

[1]2017. Pandas Library Tutorials. [ONLINE] Available at: https://www.tutorialspoint.com/python_pandas/python_pandas_dataframe.htm

[Accessed 18 January 2018].

[2] 2017. Pandas Library Tutorials. [ONLINE] Available at:

https://pandas.pydata.org/pandas-docs/stable/dsintro.html

[Accessed 18 January 2018].

[3] hamelg. 2015. Python for Data Analysis. [ONLINE] Available at: http://hamelg.blogspot.com.tr/2015/11/python-for-data-analysis-part-14.html. [Accessed 9 January 2018].

[4] CHRIS ALBON. 2017. *Drop Column and Row with Pandas*. [ONLINE] Available at: https://chrisalbon.com/python/data_wrangling/pandas_dropping_column_and_rows/ [Accessed 11 January 2018]

[5] John Tukey. 2017. *Python Plotting for EDA*. [ONLINE] Available at: http://pythonplot.com/

[Accessed 18 January 2018].

[6] Ravi Teja Gudapati. 2017. *Seaborn plot and pandas*. [ONLINE] Available at:

https://www.kaggle.com/tejainece/seaborn-barplot-and-pandas-value-counts.

[Accessed 12 January 2018].

Appendix

```
import numpy as np
import pandas as pd
from numpy.ma.core import sort
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sb

import pandas as pd
filename = 'directory2.csv'
data = pd.read_csv(filename)
data.info()  #25600
print(data.describe())
data.columns

#describe the categorical columns.
categorical = data.dtypes[data.dtypes == "object"].index
print(data[categorical].describe())

#I dont need all of columns, so delete them:
del data["Postcode"]
del data["Phone Number"]
del data["Street Address"]

#Find unique brands:
categorical = data.dtypes[data.dtypes == "object"].index
print(categorical)
print(data[categorical]["Brand"].describe())  #unique 4, why?
unique_brand = data.groupby('Brand')["Store Number"].nunique()  ##25248 Starbucks
print("Unique Brands:\n",unique_brand)
#data.info()

#Take just Starbucks brand:
import numpy as np
sb_index=np.where(data["Brand"]=="Starbucks")
data=data.loc[sb_index]
print(data.info()) #25249

##Ownership Type
import seaborn as sns
type_colors = ['#78C850',  # Grass
               '#F08030',  # Fire
               '#C03028',  # Electric
               '#E0C068' # Ground
              ]
sns.set(style="whitegrid", context="talk")
sns.countplot(x='Ownership Type', data=data, palette=type_colors)
plt.title("Distribution of Ownership")
plt.show()

#nullity check:
print("Nullity Check:\n",pd.isnull(data).any())
print("Sum of null values for each column:\n",pd.isnull(data).sum())
#Remove the null values from City by row index
missing= np.where(data["City"].isnull() == True)
print(missing[0])
data = data.drop(data.index[missing[0]])
missing= np.where(data["Longitude"].isnull() == True)
data = data.drop(data.index[missing[0]])
missing= np.where(data["Latitude"].isnull() == True)
data = data.drop(data.index[missing[0]])
print("Nullity Check:\n",pd.isnull(data).any())
data.info()

#Group Data according to Countries and Cities and find descriptives:
```

```
country_count = data[['Country','City']].groupby(['Country'])['City'] \
                              .count() \
                              .reset_index(name='count') \
                              .sort_values(['count'], ascending=False) #\
print("Country number of stores: ",country_count['count'].count())
print("Average of stores: ",country_count['count'].mean())
print("Max number of stores: ",country_count['count'].max())
print("Min number of stores: ",country_count['count'].min())

#last 10
country_count=country_count.tail(10)
plt.figure(figsize=(10,5))
sns.barplot(country_count['Country'], country_count['count'], alpha=0.8)
plt.title('Last 10 Country')
plt.ylabel('Number of Stores', fontsize=12)
plt.xlabel('Country', fontsize=12)
plt.show()

#first 10
country_count=country_count.head(10)
plt.figure(figsize=(10,5))
sns.barplot(country_count['Country'], country_count['count'], alpha=0.8)
plt.title('Top 10 Country')
plt.ylabel('Number of Stores', fontsize=12)
plt.xlabel('Country', fontsize=12)
plt.show()

# Is Italy in list?
sb_index=np.where(country_count['Country']=="IT")
x=country_count.loc[sb_index]
print(x)

#Group Data according to Cities of Turkey and find descriptives:
tr_data = pd.DataFrame(data.loc[data['Country'] == 'TR'])

city_count = tr_data[['City']].groupby(['City'])['City'] \
                              .count() \
                              .reset_index(name='count') \
                              .sort_values(['count'], ascending=False)
print("Country number of stores: ",city_count['count'].count())
print("Average of stores: ",city_count['count'].mean())
print("Max number of stores: ",city_count['count'].max())
print("Min number of stores: ",city_count['count'].min())

#top 10
city_count=city_count.head(10)
plt.figure(figsize=(10,5))
sns.barplot(city_count['City'], city_count['count'], alpha=0.8)
plt.title('Starbucks in top 10 City of Turkey')
plt.ylabel('Number of Stores', fontsize=12)
plt.xlabel('City', fontsize=12)
plt.show()

#last 10
city_count=city_count.tail(10)
print(city_count)
```