Capstone

January 16, 2019

In [1]:
```
### Import necessary Libraries
```

In [2]:
```
!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # module to convert an address into latitude and l
import requests # library to handle requests
import pandas as pd # library for data analsysis
import numpy as np # library to handle data in a vectorized manner
import random # library for random number generation
import matplotlib.pyplot as plt

# libraries for displaying images
from IPython.display import Image
from IPython.core.display import HTML

# tranforming json file into a pandas dataframe library
from pandas.io.json import json_normalize

!conda install -c conda-forge folium=0.5.0 --yes
import folium # plotting library

print('Folium installed')
print('Libraries imported.')
```

```
Solving environment: done

## Package Plan ##

  environment location: /home/jupyterlab/conda

  added / updated specs:
    - geopy


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
```

```
    geopy-1.18.1               |             py_0          51 KB  conda-forge
    openssl-1.0.2p             |    h14c3975_1002         3.1 MB  conda-forge
    geographiclib-1.49         |             py_0          32 KB  conda-forge
    conda-4.6.0                |        py36_1000         878 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         4.0 MB

The following NEW packages will be INSTALLED:

    geographiclib: 1.49-py_0         conda-forge
    geopy:         1.18.1-py_0       conda-forge

The following packages will be UPDATED:

    conda:         4.5.12-py36_1000  conda-forge --> 4.6.0-py36_1000      conda-forge
    openssl:       1.0.2p-h470a237_2 conda-forge --> 1.0.2p-h14c3975_1002 conda-forge


Downloading and Extracting Packages
geopy-1.18.1         | 51 KB     | ###################################### | 100%
openssl-1.0.2p       | 3.1 MB    | ###################################### | 100%
geographiclib-1.49   | 32 KB     | ###################################### | 100%
conda-4.6.0          | 878 KB    | ###################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Collecting package metadata: done
Solving environment: done

## Package Plan ##

  environment location: /home/jupyterlab/conda

  added / updated specs:
    - folium=0.5.0


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    altair-2.3.0               |        py36_1001         533 KB  conda-forge
    branca-0.3.1               |             py_0          25 KB  conda-forge
    folium-0.5.0               |             py_0          45 KB  conda-forge
    vincent-0.4.4              |             py_1          28 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         631 KB
```

```
The following NEW packages will be INSTALLED:

  altair             conda-forge/linux-64::altair-2.3.0-py36_1001
  branca             conda-forge/noarch::branca-0.3.1-py_0
  folium             conda-forge/noarch::folium-0.5.0-py_0
  vincent            conda-forge/noarch::vincent-0.4.4-py_1



Downloading and Extracting Packages
vincent-0.4.4        | 28 KB     | ##################################### | 100%
folium-0.5.0         | 45 KB     | ##################################### | 100%
altair-2.3.0         | 533 KB    | ##################################### | 100%
branca-0.3.1         | 25 KB     | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Folium installed
Libraries imported.
```

```
In [3]: CLIENT_ID = '1KUUP5MRFUQ4EF1EM2LCWRF3OV15IHEBW1HGL41ALCNOGLJJ' # your Foursquare ID
        CLIENT_SECRET = 'QRH1GPJOHIMFKSARJPRR2FKTTWJVC5M2GSTAHOMR4NBESCQV' # your Foursquare Sec
        VERSION = '20180604'
        LIMIT = 100
        print('Your credentails:')
        print('CLIENT_ID: ' + CLIENT_ID)
        print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: 1KUUP5MRFUQ4EF1EM2LCWRF3OV15IHEBW1HGL41ALCNOGLJJ
CLIENT_SECRET:QRH1GPJOHIMFKSARJPRR2FKTTWJVC5M2GSTAHOMR4NBESCQV
```

```
In [60]: summary = pd.DataFrame(columns=['City', 'Chain', 'Amount'])
         queries = ["McDonalds", "Starbucks", "Vapiano", "Extrablatt"]
         cities = ["Berlin", "Hamburg", "Munich", "Cologne", "Frankfurt", "Stuttgart", "Düsseldo
         i = 0
         for city in cities:
             for search_query in queries:
                 radius = 10000
                 intent= 'browse'
                 url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={
                 results = requests.get(url).json()
                 # assign relevant part of JSON to venues
                 venues = results['response']['venues']

                 # tranform venues into a dataframe
```

```
            try:
                amount = json_normalize(venues).id.count()
            except:
                amount = 0
            summary.loc[i] = [city, search_query, amount]
            i = i+1
    summary
```

Out[60]:

	City	Chain	Amount
0	Berlin	McDonalds	50
1	Berlin	Starbucks	23
2	Berlin	Vapiano	8
3	Berlin	Extrablatt	1
4	Hamburg	McDonalds	40
5	Hamburg	Starbucks	11
6	Hamburg	Vapiano	5
7	Hamburg	Extrablatt	0
8	Munich	McDonalds	50
9	Munich	Starbucks	19
10	Munich	Vapiano	4
11	Munich	Extrablatt	0
12	Cologne	McDonalds	46
13	Cologne	Starbucks	5
14	Cologne	Vapiano	5
15	Cologne	Extrablatt	3
16	Frankfurt	McDonalds	48
17	Frankfurt	Starbucks	17
18	Frankfurt	Vapiano	3
19	Frankfurt	Extrablatt	2
20	Stuttgart	McDonalds	24
21	Stuttgart	Starbucks	6
22	Stuttgart	Vapiano	2
23	Stuttgart	Extrablatt	0
24	Düsseldorf	McDonalds	44
25	Düsseldorf	Starbucks	6
26	Düsseldorf	Vapiano	4
27	Düsseldorf	Extrablatt	3
28	Dortmund	McDonalds	15
29	Dortmund	Starbucks	4
30	Dortmund	Vapiano	2
31	Dortmund	Extrablatt	1
32	Essen	McDonalds	32
33	Essen	Starbucks	6
34	Essen	Vapiano	2
35	Essen	Extrablatt	7
36	Leipzig	McDonalds	22
37	Leipzig	Starbucks	2
38	Leipzig	Vapiano	1

```
         39     Leipzig  Extrablatt      0
```

```
In [44]: summary.to_csv('summary.csv')
```

```
In [62]: data = summary
```

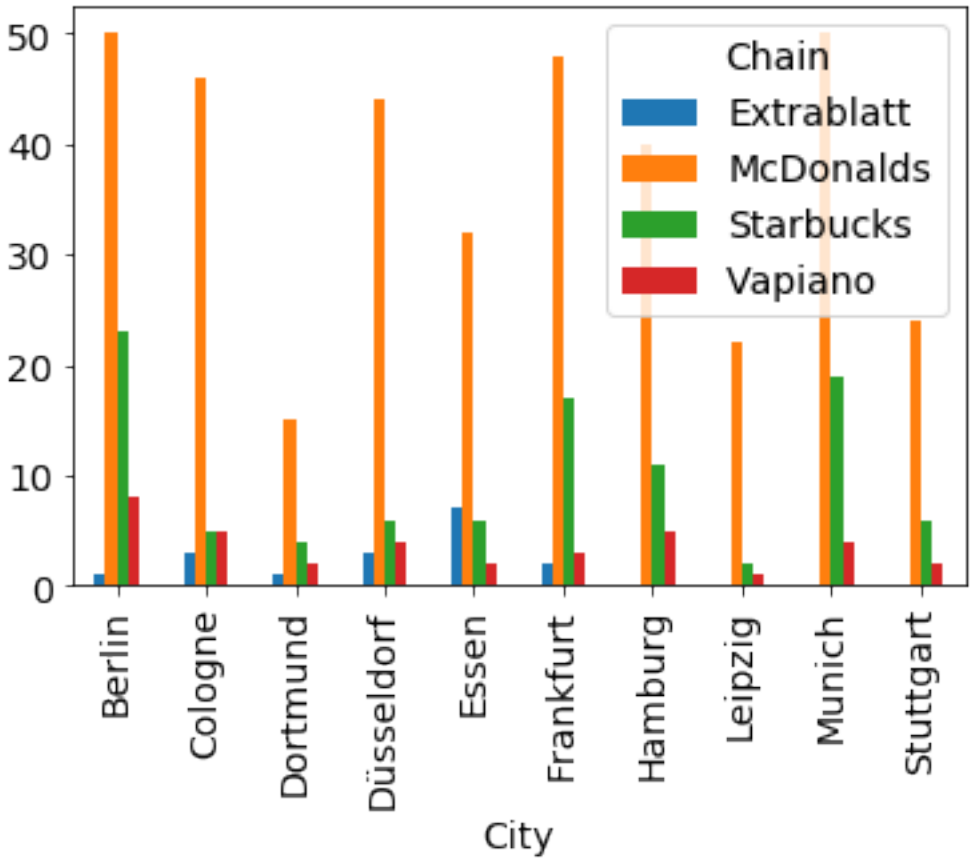
```
In [ ]: #data = pd.read_csv('summary.csv')
```

```
In [64]: pivot = data.pivot("City", "Chain", "Amount")
         pivot
```

```
Out[64]: Chain      Extrablatt McDonalds Starbucks Vapiano
         City
         Berlin              1        50        23       8
         Cologne             3        46         5       5
         Dortmund            1        15         4       2
         Düsseldorf          3        44         6       4
         Essen               7        32         6       2
         Frankfurt           2        48        17       3
         Hamburg             0        40        11       5
         Leipzig             0        22         2       1
         Munich              0        50        19       4
         Stuttgart           0        24         6       2
```

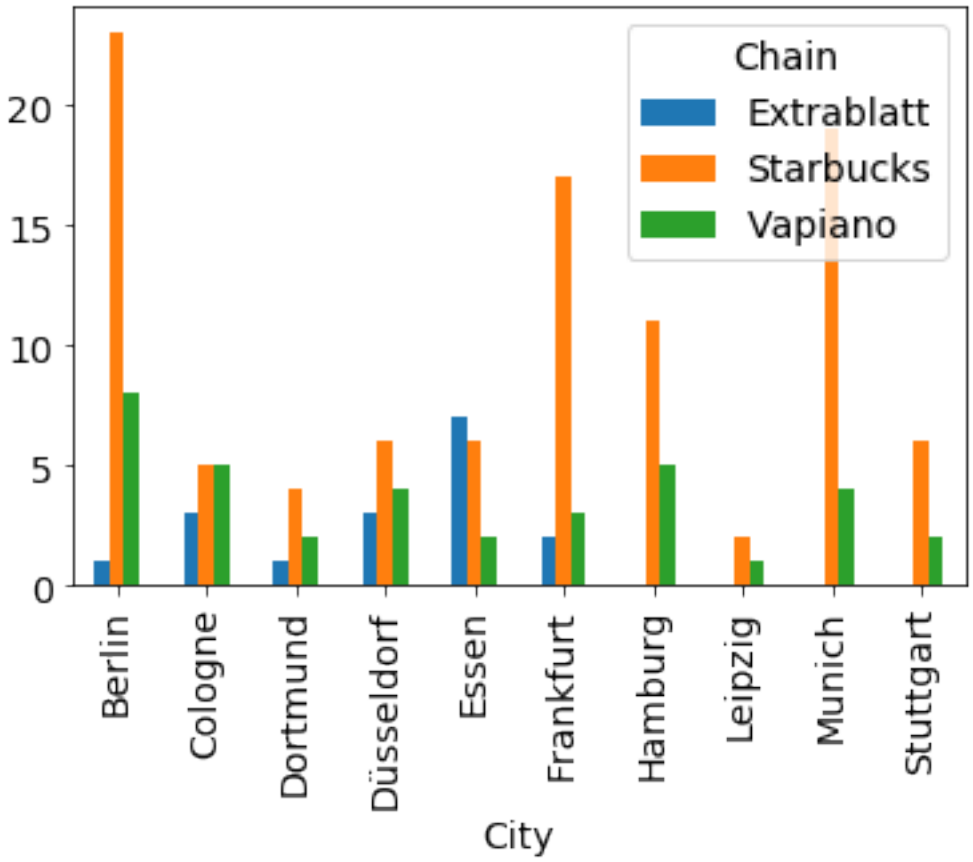
```
In [65]: # Here comes the plot with all the restaurants

         pivot.plot(kind='bar')
         plt.show()
```



In [71]:
```
# Graph without McDonalds

pivot.drop(columns='McDonalds').plot(kind='bar')
plt.show()
```




```
In [86]: # How about the rate of Vapiano to Starbucks, to have an idea
         rate = pivot['Vapiano']/pivot['Starbucks']
         rate.sort_values(0)
```

```
Out[86]: City
         Frankfurt     0.176471
         Munich        0.210526
         Essen         0.333333
         Stuttgart     0.333333
         Berlin        0.347826
         Hamburg       0.454545
         Dortmund           0.5
         Leipzig            0.5
         Düsseldorf    0.666667
         Cologne               1
         dtype: object
```

```
In [87]: # How about the rate of McDonalds to Starbucks
         rate = pivot['McDonalds']/pivot['Starbucks']
         rate.sort_values(0)
```

Out[87]:
```
City
Berlin        2.17391
Munich        2.63158
Frankfurt     2.82353
Hamburg       3.63636
Dortmund         3.75
Stuttgart           4
Essen         5.33333
Düsseldorf    7.33333
Cologne           9.2
Leipzig            11
dtype: object
```