# Peer review assignment 3

---

## Project Description

This week, the client has more questions for you to answer to further develop the profile of potential clients. This includes finding which cities to target and how to evaluate them. You will have access to crime rates, population, housing growth city wide data and segmentation distribution dumps to help in your analysis.

From the graphs, we can deduce Diverse Workers, Mass Markets, Young Affluent Mobiles , Young Urban Masses and Young Upscale Families occupies top 5. Even though Well-heeled Affluents wants to buy security devices the most, they will have more choices to choose from.

## Data Dictionary

| Field | Description |
|---|---|
| City | City Names |
| Segment | Segment Names |
| Total | Summation |

## Import Libraries

In [1]:

```
import numpy as np
from numpy import count_nonzero, median, mean
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import random


%matplotlib inline
#sets the default autosave frequency in seconds
%autosave 60
sns.set_style('dark')
sns.set(font_scale=1.2)

plt.rc('axes', titlesize=9)
plt.rc('axes', labelsize=14)
plt.rc('xtick', labelsize=12)
plt.rc('ytick', labelsize=12)

import warnings
warnings.filterwarnings('ignore')

pd.set_option('display.max_columns',None)
#pd.set_option('display.max_rows',None)
pd.set_option('display.width', 1000)
pd.set_option('display.float_format','{:.2f}'.format)

random.seed(0)
np.random.seed(0)
np.set_printoptions(suppress=True)
```

```
Autosaving every 60 seconds
```

# Exploratory Data Analysis

In [2]:
```
df = pd.read_csv("cities.csv")
```

In [3]:
```
df
```

Out[3]:

| | City | ComfortableRetirees | DiverseWorkers | ElderMidscaleClass | EliteHouseholds | MassMarkets | Modest |
|---|---|---|---|---|---|---|---|
| **0** | May | 203 | 118 | 91 | 1 | 30 | |
| **1** | Aaronsburg | 4 | 4 | 67 | 0 | 90 | |
| **2** | Abbeville | 1426 | 2019 | 1782 | 53 | 1684 | |
| **3** | Abbot | 104 | 47 | 31 | 0 | 17 | |
| **4** | Abbotsford | 28 | 205 | 27 | 5 | 200 | |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **15743** | Zumbro Falls | 12 | 6 | 31 | 17 | 172 | |
| **15744** | Zumbrota | 25 | 55 | 88 | 62 | 239 | |
| **15745** | Zuni | 124 | 899 | 139 | 13 | 145 | |
| **15746** | Zwingle | 11 | 14 | 32 | 3 | 50 | |
| **15747** | Zwolle | 33 | 646 | 214 | 4 | 67 | |

15748 rows × 14 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15748 entries, 0 to 15747
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   City                  15748 non-null  object
 1   ComfortableRetirees   15748 non-null  int64 
 2   DiverseWorkers        15748 non-null  int64 
 3   ElderMidscaleClass    15748 non-null  int64 
 4   EliteHouseholds       15748 non-null  int64 
 5   MassMarkets           15748 non-null  int64 
 6   ModestFamilies        15748 non-null  int64 
 7   ProsperousAcheivers   15748 non-null  int64 
 8   UpscaleMatures        15748 non-null  int64 
 9   WellheeledAffluents   15748 non-null  int64 
 10  YoungAffluentMobiles  15748 non-null  int64 
 11  YoungUpscaleFamilies  15748 non-null  int64 
 12  YoungUrbanMasses      15748 non-null  int64 
 13  Total                 15748 non-null  int64 
dtypes: int64(13), object(1)
memory usage: 1.7+ MB
```

In [5]:
```
df.describe()
```

Out[5]:

| | ComfortableRetirees | DiverseWorkers | ElderMidscaleClass | EliteHouseholds | MassMarkets | ModestFamilies | Pro |
|---|---|---|---|---|---|---|---|
| **count** | 15748.00 | 15748.00 | 15748.00 | 15748.00 | 15748.00 | 15748.00 | |
| **mean** | 222.43 | 575.78 | 325.54 | 240.18 | 568.00 | 249.11 | |
| **std** | 850.03 | 2839.37 | 998.51 | 1109.76 | 2442.70 | 1957.72 | |
| **min** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | |
| **25%** | 10.00 | 7.00 | 23.00 | 1.00 | 18.00 | 0.00 | |
| **50%** | 42.00 | 55.00 | 83.00 | 7.00 | 89.00 | 2.00 | |
| **75%** | 156.00 | 297.25 | 260.00 | 66.00 | 358.00 | 23.00 | |
| **max** | 43857.00 | 128468.00 | 39708.00 | 53139.00 | 88380.00 | 121997.00 | |

In [6]:
```
df.columns
```

Out[6]:
```
Index(['City', 'ComfortableRetirees', 'DiverseWorkers', 'ElderMidscaleClass', 'EliteHouseh
olds', 'MassMarkets', 'ModestFamilies', 'ProsperousAcheivers', 'UpscaleMatures', 'Wellheel
edAffluents', 'YoungAffluentMobiles', 'YoungUpscaleFamilies', 'YoungUrbanMasses', 'Tota
l'], dtype='object')
```

In [7]:
```
df["TargetGroups"] = df["DiverseWorkers"] + df["MassMarkets"] + df["YoungAffluentMobiles"]
+ df["YoungUpscaleFamilies"] + df["YoungUrbanMasses"]
```

In [8]:
```
df.head()
```

Out[8]:

| | City | ComfortableRetirees | DiverseWorkers | ElderMidscaleClass | EliteHouseholds | MassMarkets | ModestFami |
|---|---|---|---|---|---|---|---|
| **0** | May | 203 | 118 | 91 | 1 | 30 | |
| **1** | Aaronsburg | 4 | 4 | 67 | 0 | 90 | |
| **2** | Abbeville | 1426 | 2019 | 1782 | 53 | 1684 | |
| **3** | Abbot | 104 | 47 | 31 | 0 | 17 | |
| **4** | Abbotsford | 28 | 205 | 27 | 5 | 200 | |

In [9]:
```
df["Percentage"] = (df["TargetGroups"]/df["Total"]) * 100
```

In [10]:
```
df.head()
```

Out[10]:

| | City | ComfortableRetirees | DiverseWorkers | ElderMidscaleClass | EliteHouseholds | MassMarkets | ModestFami |
|---|---|---|---|---|---|---|---|
| **0** | May | 203 | 118 | 91 | 1 | 30 | |
| **1** | Aaronsburg | 4 | 4 | 67 | 0 | 90 | |
| **2** | Abbeville | 1426 | 2019 | 1782 | 53 | 1684 | |
| **3** | Abbot | 104 | 47 | 31 | 0 | 17 | |
| **4** | Abbotsford | 28 | 205 | 27 | 5 | 200 | |

In [11]:
```
df[["City","Total","Percentage"]].sort_values(by="Total", ascending=False, axis=0).head()
```

Out[11]:

| | City | Total | Percentage |
|---|---|---|---|
| **6505** | Houston | 567640 | 50.23 |
| **2494** | Chicago | 550579 | 84.72 |
| **1669** | Brooklyn | 503335 | 68.74 |
| **8107** | Los Angeles | 435664 | 65.04 |
| **9866** | New York | 405625 | 61.16 |

In [12]:
```
df[["City","Total","Percentage"]].sort_values(by="Percentage", ascending=False).head()
```

Out[12]:

| | City | Total | Percentage |
|---|---|---|---|
| **6863** | Jbsa Randolph | 8 | 100.00 |
| **8202** | Lukachukai | 294 | 100.00 |
| **11230** | Point Mugu Nawc | 6 | 100.00 |
| **1731** | Brush Valley | 3 | 100.00 |
| **15197** | Westfall | 3 | 100.00 |

# Conclusion

From the tables, we decided on dense populations in cities since we can reach as many customers in one city.

Selected cities are Houston, Chicago, Brooklyn, Los Angeles, New York

### Python code done by Dennis Lam