

DSC640 Week7-8

Assignment 4.2

Name: Madhuri Basava

Date: 07/23/2024

```
In [1]: # Import the necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Ignore warnings
import warnings
warnings.filterwarnings('ignore')

# Set the style of matplotlib
%matplotlib inline
```

```
In [2]: # Load the players dataset into the data frame
ppg_df = pd.read_csv('ppg2008.csv')
ppg_df.head(10)
```

Out[2]:

	Name	G	MIN	PTS	FGM	FGA	FGP	FTM	FTA	FTP	...	3PA	3PP	ORB	DRB	TRB
0	Dwyane Wade	79	38.6	30.2	10.8	22.0	0.491	7.5	9.8	0.765	...	3.5	0.317	1.1	3.9	5.0
1	LeBron James	81	37.7	28.4	9.7	19.9	0.489	7.3	9.4	0.780	...	4.7	0.344	1.3	6.3	7.6
2	Kobe Bryant	82	36.2	26.8	9.8	20.9	0.467	5.9	6.9	0.856	...	4.1	0.351	1.1	4.1	5.2
3	Dirk Nowitzki	81	37.7	25.9	9.6	20.0	0.479	6.0	6.7	0.890	...	2.1	0.359	1.1	7.3	8.4
4	Danny Granger	67	36.2	25.8	8.5	19.1	0.447	6.0	6.9	0.878	...	6.7	0.404	0.7	4.4	5.1
5	Kevin Durant	74	39.0	25.3	8.9	18.8	0.476	6.1	7.1	0.863	...	3.1	0.422	1.0	5.5	6.5
6	Kevin Martin	51	38.2	24.6	6.7	15.9	0.420	9.0	10.3	0.867	...	5.4	0.415	0.6	3.0	3.6
7	Al Jefferson	50	36.6	23.1	9.7	19.5	0.497	3.7	5.0	0.738	...	0.1	0.000	3.4	7.5	11.0
8	Chris Paul	78	38.5	22.8	8.1	16.1	0.503	5.8	6.7	0.868	...	2.3	0.364	0.9	4.7	5.5
9	Carmelo Anthony	66	34.5	22.8	8.1	18.3	0.443	5.6	7.1	0.793	...	2.6	0.371	1.6	5.2	6.8

10 rows × 21 columns



In [3]: `ppg_df.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 21 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0   Name    50 non-null      object
 1   G        50 non-null      int64
 2   MIN      50 non-null      float64
 3   PTS      50 non-null      float64
 4   FGM      50 non-null      float64
 5   FGA      50 non-null      float64
 6   FGP      50 non-null      float64
 7   FTM      50 non-null      float64
 8   FTA      50 non-null      float64
 9   FTP      50 non-null      float64
10   3PM      50 non-null      float64
11   3PA      50 non-null      float64
12   3PP      50 non-null      float64
13   ORB      50 non-null      float64
14   DRB      50 non-null      float64
15   TRB      50 non-null      float64
16   AST      50 non-null      float64
17   STL      50 non-null      float64
18   BLK      50 non-null      float64
19   TO       50 non-null      float64
20   PF       50 non-null      float64
dtypes: float64(19), int64(1), object(1)
memory usage: 8.3+ KB

```

```

In [4]: # Create Heat Map

# Trim whitespace from column names
ppg_df.columns = ppg_df.columns.str.strip()

# set `NAME` column as index in the dataframe
ppg_df.set_index('Name', inplace=True)

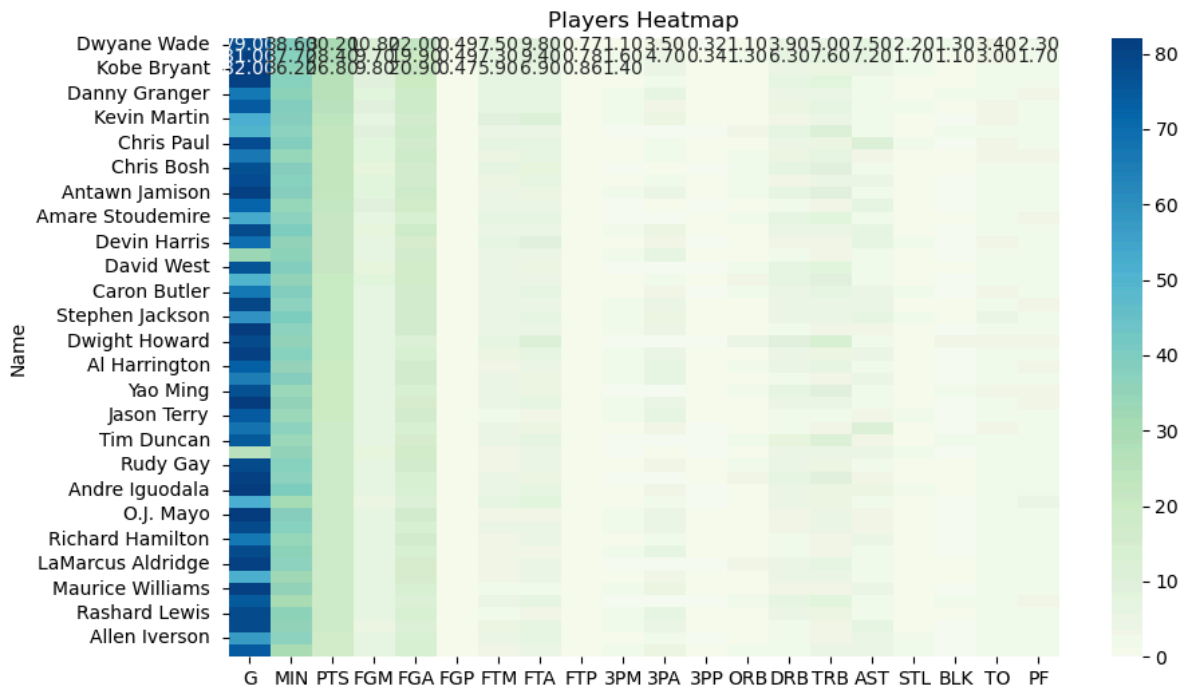
# Create the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(ppg_df, annot=True, fmt=".2f", cmap="GnBu")

# Rotate y-axis labels for better readability
plt.yticks(rotation=0)

# Set title
plt.title('Players Heatmap')

# Show plot
plt.show()

```



In []:

In [5]: *#Load the costco geocoded dataset into a Pandas data frame.*
geocoded_df=pd.read_csv("costcos-geocoded.csv")
geocoded_df

Out[5]:

	Address	City	State	Zip Code	Latitude	Longitude
0	1205 N. Memorial Parkway	Huntsville	Alabama	35801-5930	34.743095	-86.600955
1	3650 Galleria Circle	Hoover	Alabama	35244-2346	33.377649	-86.812420
2	8251 Eastchase Parkway	Montgomery	Alabama	36117	32.363889	-86.150884
3	5225 Commercial Boulevard	Juneau	Alaska	99801-7210	58.359200	-134.483000
4	330 West Dimond Blvd	Anchorage	Alaska	99515-1950	61.143266	-149.884217
...
412	19610 SE 1st St	Vancouver	Washington	98607	45.621299	-122.459135
413	10990 Harbor Hill Dr	Gig Harbor	Washington	98335	47.357748	-122.603888
414	27520 Covington Way SE	Covington	Washington	98042	47.354838	-122.121185
415	2150 Deming Way	Middleton	Wisconsin	53562-5507	43.100195	-89.522751
416	950 Port Washington Rd	Grafton	Wisconsin	53024-9201	43.324691	-87.921615

417 rows × 6 columns

In [6]: geocoded_df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 417 entries, 0 to 416
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Address     417 non-null   object
1   City        417 non-null   object
2   State       417 non-null   object
3   Zip Code    417 non-null   object
4   Latitude    417 non-null   float64
5   Longitude   417 non-null   float64
dtypes: float64(2), object(4)
memory usage: 19.7+ KB
```

```
In [7]: !pip install geopandas
        !pip install geodatasets
```

Requirement already satisfied: geopandas in c:\users\madhu\anaconda3\lib\site-packages (1.0.1)

Requirement already satisfied: numpy>=1.22 in c:\users\madhu\appdata\roaming\python\python39\site-packages (from geopandas) (1.26.1)

Requirement already satisfied: pyogrio>=0.7.2 in c:\users\madhu\anaconda3\lib\site-packages (from geopandas) (0.9.0)

Requirement already satisfied: packaging in c:\users\madhu\appdata\roaming\python\python39\site-packages (from geopandas) (23.2)

Requirement already satisfied: pandas>=1.4.0 in c:\users\madhu\anaconda3\lib\site-packages (from geopandas) (2.1.1)

Requirement already satisfied: pyproj>=3.3.0 in c:\users\madhu\anaconda3\lib\site-packages (from geopandas) (3.6.1)

Requirement already satisfied: shapely>=2.0.0 in c:\users\madhu\anaconda3\lib\site-packages (from geopandas) (2.0.5)

Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\madhu\anaconda3\lib\site-packages (from pandas>=1.4.0->geopandas) (2.8.2)

Requirement already satisfied: pytz>=2020.1 in c:\users\madhu\anaconda3\lib\site-packages (from pandas>=1.4.0->geopandas) (2023.3.post1)

Requirement already satisfied: tzdata>=2022.1 in c:\users\madhu\anaconda3\lib\site-packages (from pandas>=1.4.0->geopandas) (2023.3)

Requirement already satisfied: certifi in c:\users\madhu\appdata\roaming\python\python39\site-packages (from pyogrio>=0.7.2->geopandas) (2023.7.22)

Requirement already satisfied: six>=1.5 in c:\users\madhu\appdata\roaming\python\python39\site-packages (from python-dateutil>=2.8.2->pandas>=1.4.0->geopandas) (1.15.0)

Requirement already satisfied: geodatasets in c:\users\madhu\anaconda3\lib\site-packages (2024.7.0)

Requirement already satisfied: pooch in c:\users\madhu\anaconda3\lib\site-packages (from geodatasets) (1.8.2)

Requirement already satisfied: platformdirs>=2.5.0 in c:\users\madhu\anaconda3\lib\site-packages (from pooch->geodatasets) (3.10.0)

Requirement already satisfied: packaging>=20.0 in c:\users\madhu\appdata\roaming\python\python39\site-packages (from pooch->geodatasets) (23.2)

Requirement already satisfied: requests>=2.19.0 in c:\users\madhu\appdata\roaming\python\python39\site-packages (from pooch->geodatasets) (2.31.0)

Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\madhu\appdata\roaming\python\python39\site-packages (from requests>=2.19.0->pooch->geodatasets) (3.3.2)

Requirement already satisfied: idna<4,>=2.5 in c:\users\madhu\appdata\roaming\python\python39\site-packages (from requests>=2.19.0->pooch->geodatasets) (3.4)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\madhu\appdata\roaming\python\python39\site-packages (from requests>=2.19.0->pooch->geodatasets) (2.0.7)

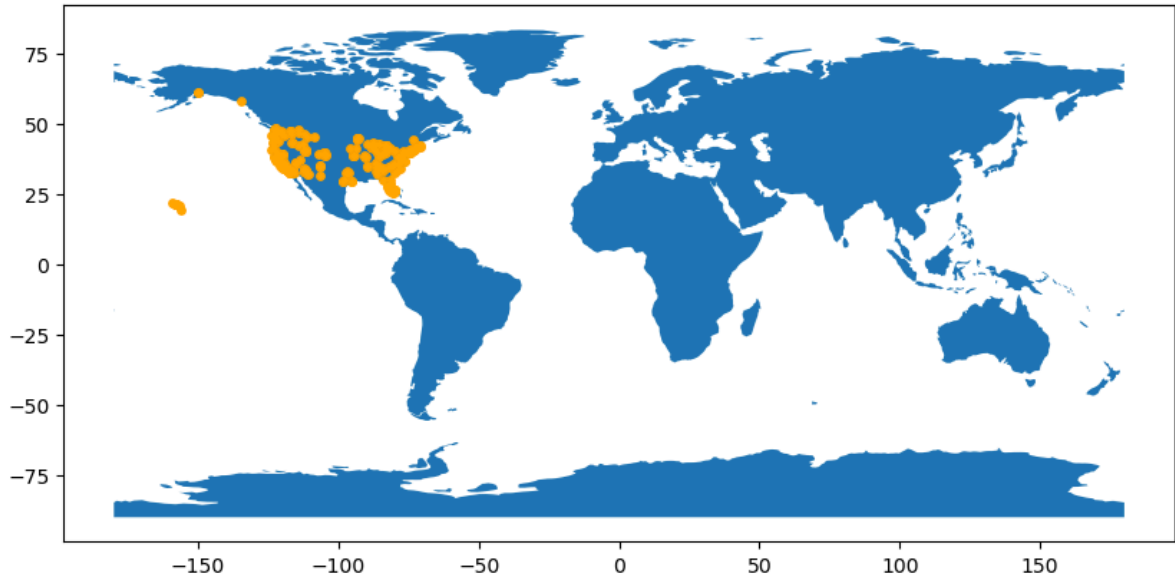
Requirement already satisfied: certifi>=2017.4.17 in c:\users\madhu\appdata\roaming\python\python39\site-packages (from requests>=2.19.0->pooch->geodatasets) (2023.7.22)

```
In [8]: from shapely.geometry import Point
import geopandas as gpd
from geopandas import GeoDataFrame
import geodatasets

geometry = [Point(xy) for xy in zip(geocoded_df['Longitude'], geocoded_df['Latitude'])]
gdf = GeoDataFrame(geocoded_df, geometry=geometry)

# This is a simple map with geopandas
```

```
world = gpd.read_file(geodatasets.data.naturalearth.land['url'])
gdf.plot(ax=world.plot(figsize=(10, 6)), marker='o', color='orange', markersize=15)
```



In [10]: *# Create lollipop chart*

```
# Load the players dataset into the data frame
ppg_df = pd.read_csv('ppg2008.csv')

# Trim whitespace from column names
ppg_df.columns = ppg_df.columns.str.strip()

plt.figure(figsize=(10, 6))
plt.stem(ppg_df['Name'], ppg_df['PTS'], linefmt='b-', markerfmt='go', basefmt='k-')
plt.xticks(rotation=45, ha='right')
plt.xlabel('Player')
plt.ylabel('Points per Game')
plt.title('Points per Game by Player (Lollipop Chart)')
plt.tight_layout()

# Show plot
plt.show()
```

Points per Game by Player (Lollipop Chart)

