

# Python Script

## Assignment 5.2: Heat Maps, Contour Charts and Spatial Charts

DSC640

Taniya Adhikari

```
In [2]: import datetime as dt
from pathlib import Path
import math
import os
import sqlite3
import json
import geopandas as gpd
import pygeos
import pyproj
import shapely
import shapely.ops as ops
from shapely.geometry import Point, Polygon
from shapely.geometry.polygon import Polygon
from functools import partial
import geoplots as gplt
import geoplots.crs as gcrs

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
# Data Preprocessing
from sklearn.preprocessing import MinMaxScaler
```

## Heat Maps

```
In [4]: df = pd.read_csv("ppg2008.csv")
df.columns = df.columns.str.strip()
```

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

Out[5]:

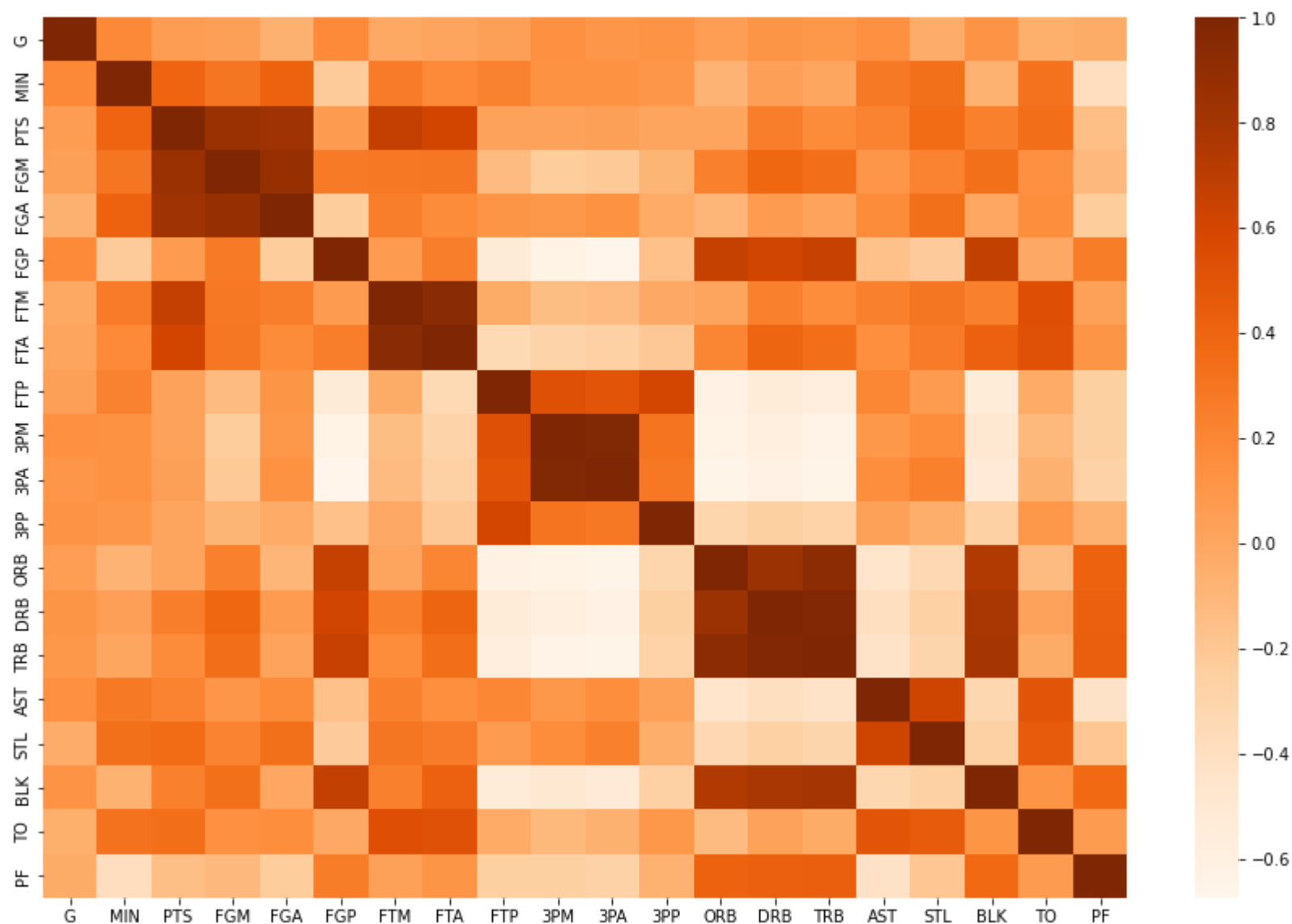
	Name	G	MIN	PTS	FGM	FGA	FGP	FTM	FTA	FTP	...	3PA	3PP	ORB	DRB	TRB	AST	STL	BLK	TO	PF
0	Dwyane Wade	79	38.6	30.2	10.8	22.0	0.491	7.5	9.8	0.765	...	3.5	0.317	1.1	3.9	5.0	7.5	2.2	1.3	3.4	2.3
1	LeBron James	81	37.7	28.4	9.7	19.9	0.489	7.3	9.4	0.780	...	4.7	0.344	1.3	6.3	7.6	7.2	1.7	1.1	3.0	1.7
2	Kobe Bryant	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	4.9	1.5	0.5	2.6	2.3
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	2.4	0.8	0.8	1.9	2.2
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	2.7	1.0	1.4	2.5	3.1

5 rows × 21 columns

In [14]:

```
plt.rcParams['figure.figsize'] = [15,10]
fig, ax = plt.subplots()
sns.heatmap(df.drop(['Name'], axis = 1).corr(), ax = ax, cmap="Oranges")
plt.suptitle("Python - Heatmap: Basketball Statistics",
             size=20, x=0.08, y=.95, horizontalalignment='left', verticalalignment='top')
plt.show()
```

## Python - Heatmap: Basketball Statistics



## Spatial Charts

```
In [8]: contiguous_usa = gpd.read_file(gplt.datasets.get_path('contiguous_usa'))
```

```
In [9]: df2 = pd.read_csv("costcos-geocoded.csv")
df2.head()
```

```
Out[9]:
```

	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

```
In [10]: location = gpd.GeoDataFrame(df2, geometry=gpd.points_from_xy(df2['Longitude'], df2['Latitude']))
```

```
In [11]: location2 = pd.DataFrame(location['State'].value_counts()).reset_index()
location2.columns = ['State', 'Count']

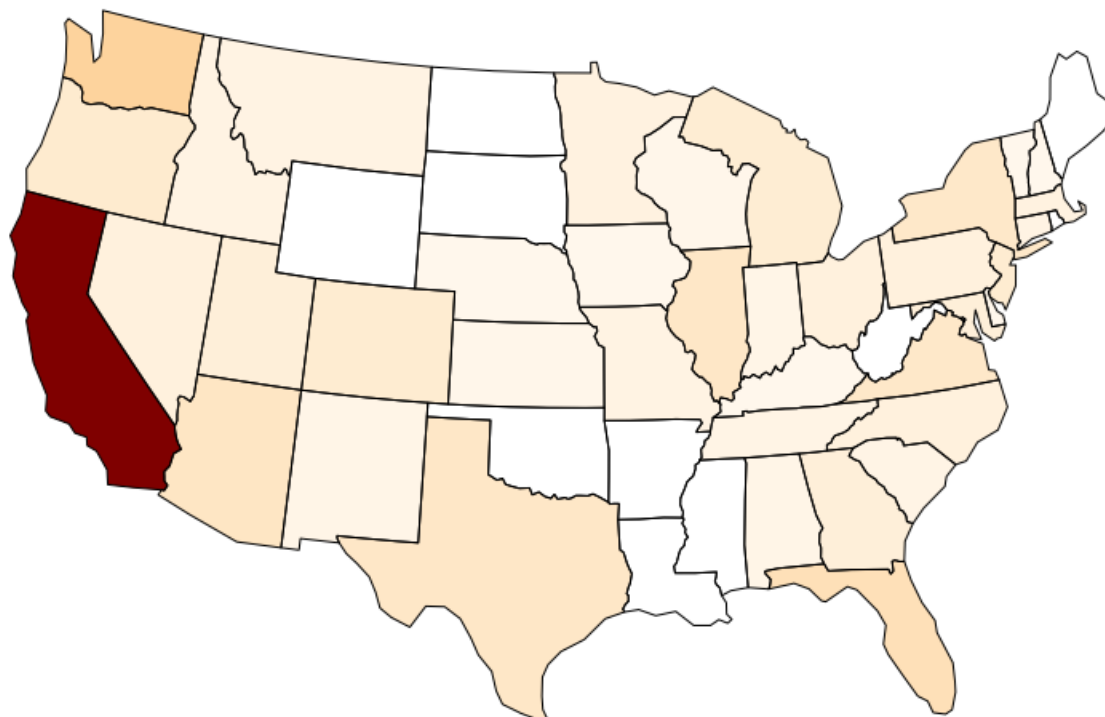
geodata = contiguous_usa.merge(location2, how='outer', left_on=['state'], right_on=['State'])
```

```
In [12]: ax = gplt.polyplot(contiguous_usa,
                           projection=gcrs.AlbersEqualArea(),
                           figsize = (19,16),
                           zorder = 2)
gplt.choropleth(geodata,
                hue = geodata['Count'],
                cmap = 'OrRd',
                projection=gcrs.WebMercator(),
                ax = ax)
plt.title("Python - Spatial CHarts: Number of Locations by State", fontsize=20)
plt.show()
```

C:\Users\bibek\anaconda3\envs\my\_env\lib\site-packages\geoplot\geoplot.py:66: UserWarning: The data being passed to "hue" includes null values. You probably want to remove these before plotting this data with geoplot.

```
warnings.warn(
```

## Python - Spatial CHarts: Number of Locations by State



## Contour Maps

In [15]:

```
def f(x, y):
    return -np.sin(x) ** 10 + np.cos(10 + y * x) * np.cos(x)

x = np.linspace(0.1, 5, 50)
y = np.linspace(0.1, 5, 40)

X, Y = np.meshgrid(x, y)
Z = f(X, Y)
```

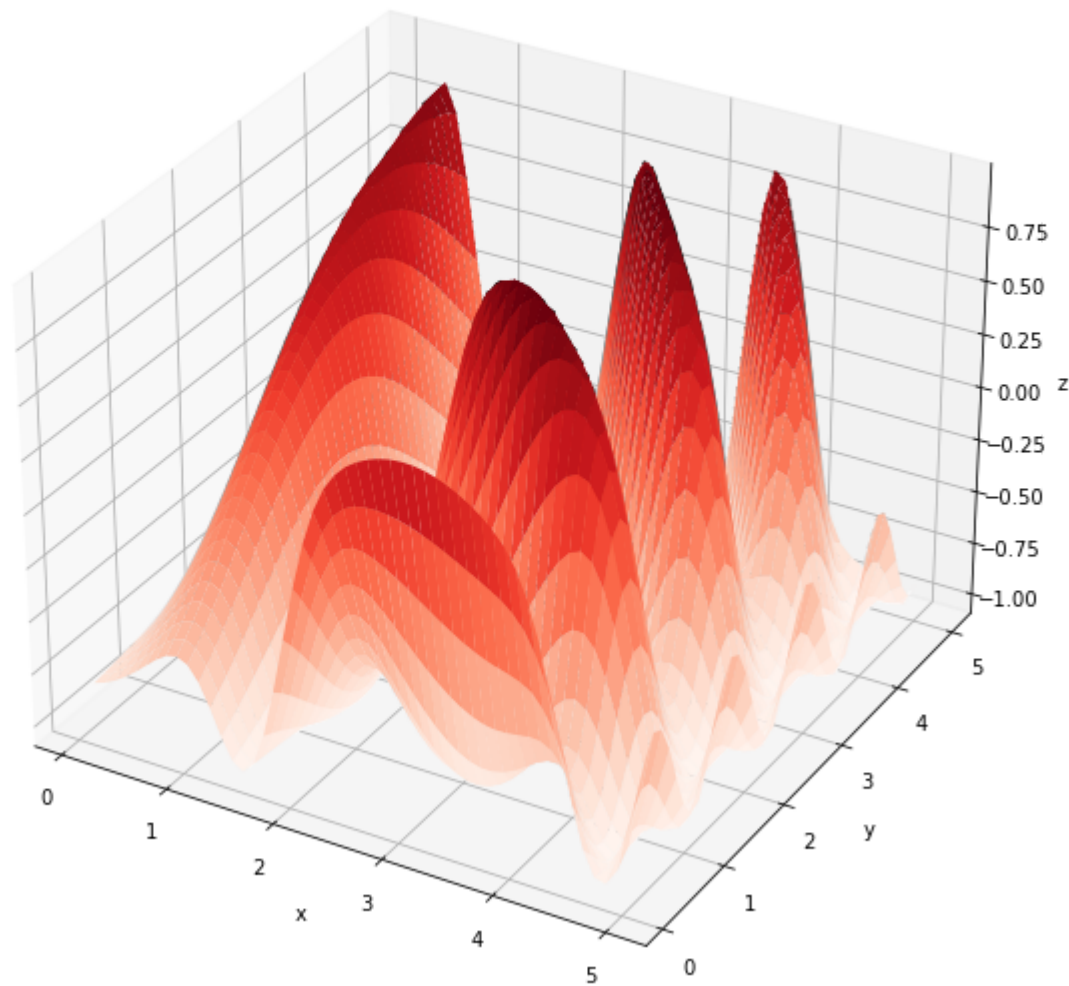
In [25]:

```
ax = plt.axes(projection='3d')
ax.contour3D(X, Y, Z, 40, cmap='Greys')
ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
               cmap='Reds', edgecolor='none')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z');

plt.suptitle("Python - Contour Chart",
            size=20, x=.2, horizontalalignment='left', verticalalignment='top')

right_side = ax.spines["right"]
right_side.set_visible(False)
top = ax.spines["top"]
top.set_visible(False)
```

## Python - Contour Chart



In [ ]: