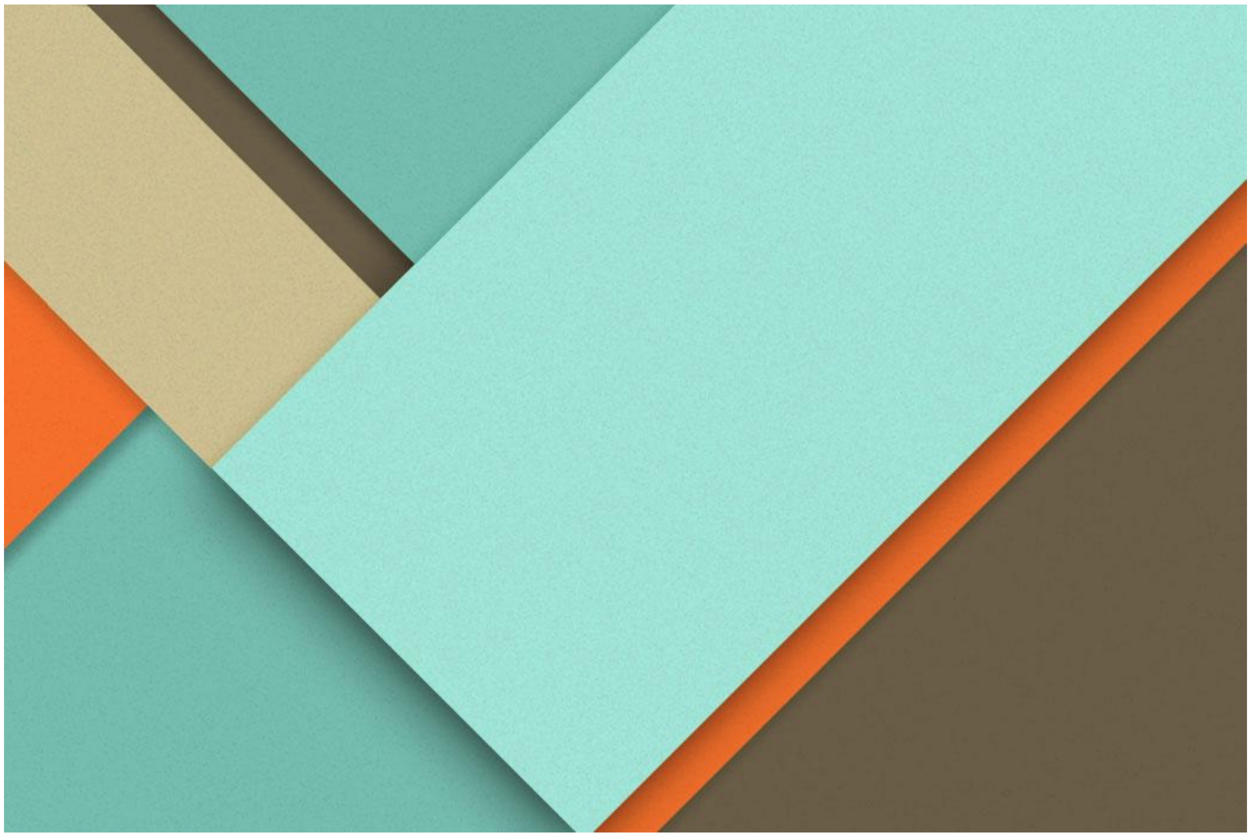


Project Report

DEM Group 11

18.04.2023

—

Alle Pranav Sudhan-(21WU0102058)
Sreeshanth S-(21WU0101017)
Sindhu Priya-(21WU0102037)
Thanmayi Vempala-(21WU0102031)

Introduction

- This report includes an analysis of a dataset in which we identify the dataset's significance and highlight it by making adjustments to the dataset by eliminating unnecessary, redundant data or null values.
- The dataset that was provided to us was based on basketball teams from various states and cities, and it contained details about the players, such as their height, name, position in the game, year in high school or college, among many others.
- The given database has records for **13806** players from various hometowns with **30** columns or attributes.
- Before deciding on the problem statements and the data that can be extracted from the dataset, we removed any unwanted columns where the data was already mentioned in another column or where it wasn't necessary. Below is a list of some of the columns that were omitted:
    - ☐ Hs_clean
    - ☐ Hometown_clean
    - ☐ Previous_school_clean
    - ☐ Season
    - ☐ Year_clean
    - ☐ Position_clean
    - ☐ Height_ft
    - ☐ Height_in
    - ☐ Total_inches

- Several columns in the dataset, like home state and position, record the same value in various ways. We eliminated such confusions (for instance, storing "California" and "Calif" separately when we may keep them together as one).

Problem Statments

1. Figuring out top 10 homestate or hometown wich produces the greatest number of players.

2. Figuring which year (Sophomore, Freshman, Junior) produces maximum number of players.

3. Figuring out how the number of players on each team vary across different conferences and divisions.

Summary of each individual work

Tasks done by Alle Pranav Sudan:

1. Data Pre-processing
2. Checked for any irrelevant columns and then removed them if necessary.
3. Checked for any inconsistent data types and converted them if necessary.
4. Made the Report
5. Helped in the PPT

Tasks done by Sreeshanth S:

1. Data Pre-processing
2. Data Analysis
3. Data Formatting
4. Data Visualisation
5. Label the axes appropriately and include a title for the visualization.
6. Use color-coding or legends to distinguish between different groups or categories in the data.
7. Interpret the results of the visualization and draw conclusions based on the research question or hypothesis.

Tasks done by Sindhu Priya:

1. Data Pre-processing
2. Data visualisation
3. Determined the research question or hypothesis to be tested.

4. Choose the appropriate type of visualization based on the type of data and the research question.
5. Used the matplotlib or seaborn library and created the visualizations.
6. Helped in making the Report

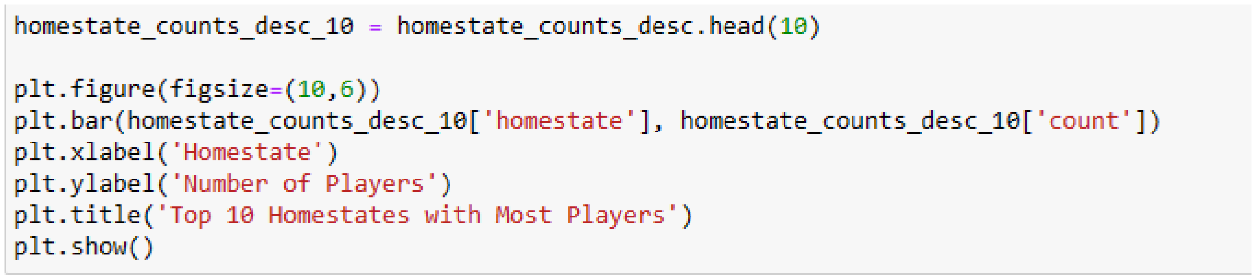
Tasks done by Thanmayi:

1. Data Analysis
2. Explored the data.
3. Analyzed the dataset.
4. Data Preprocessing
5. Helped in making the Report
6. Made the PPT

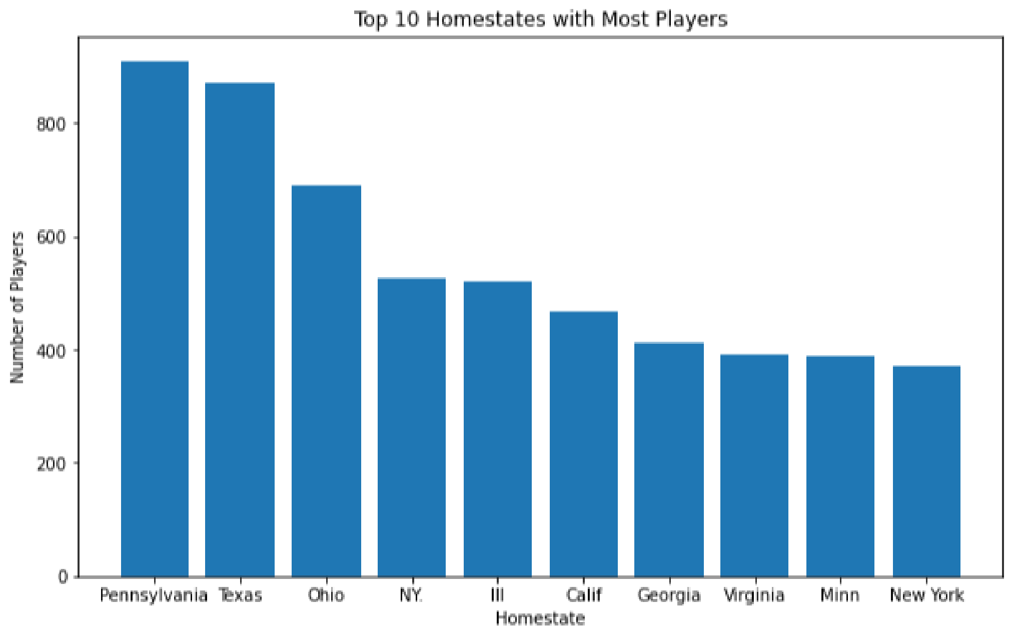
The Analysis and Visualisation

1. Figuring out top 10 homestates or hometowns which produces the greatest number of players.

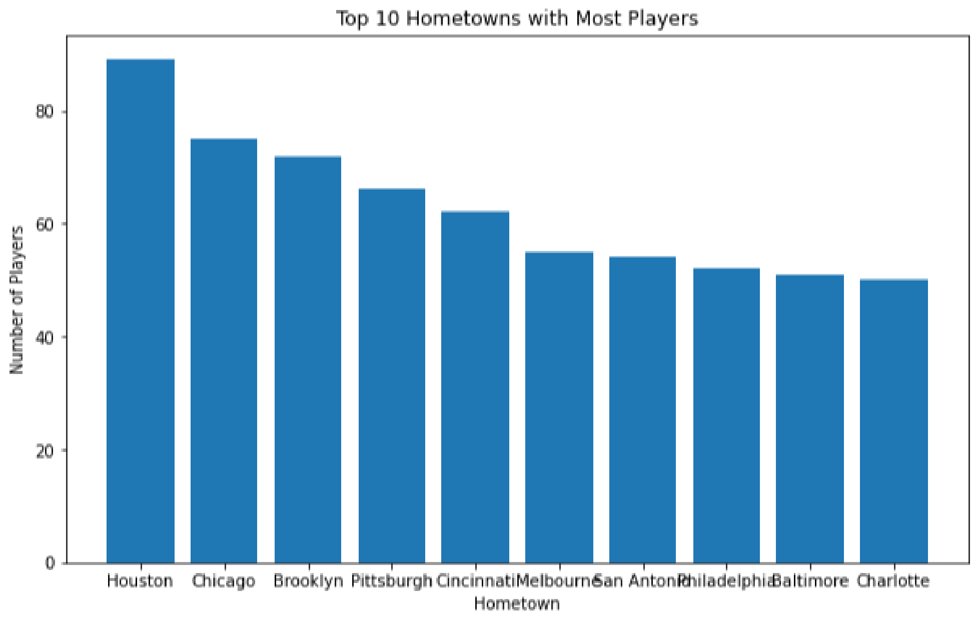
- To depict the number of players from each year, we use a bar plot.
- We then count the number of participants in each group after organizing the players by year.
- The number of players in each year can be visualized using a bar plot, which makes it easy to see how the numbers fluctuate from year to year. The height of each bar, which represents a year, reflects the number of players in that year.
- This makes it simple to determine which year has the most players. A bar plot also enables us to compare player counts across several years at once.



```python
homestate_counts_desc_10 = homestate_counts_desc.head(10)

plt.figure(figsize=(10,6))
plt.bar(homestate_counts_desc_10['homestate'], homestate_counts_desc_10['count'])
plt.xlabel('Homestate')
plt.ylabel('Number of Players')
plt.title('Top 10 Homestates with Most Players')
plt.show()
```



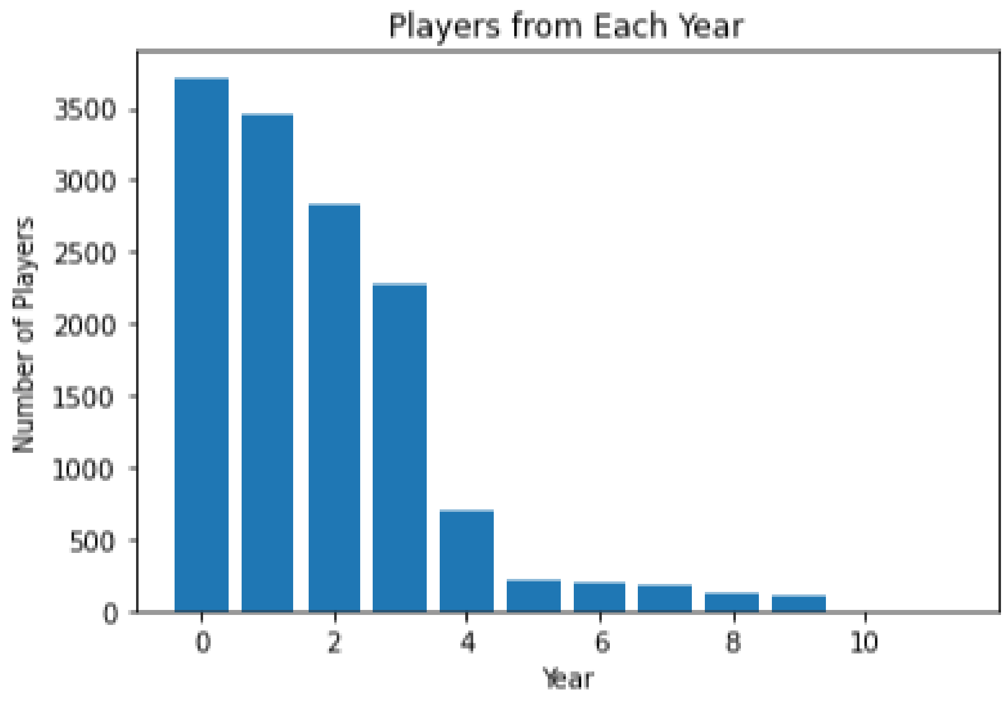
```
hometown_counts_desc_10 = hometown_counts_desc.head(10)

plt.figure(figsize=(10,6))
plt.bar(hometown_counts_desc_10['hometown'], hometown_counts_desc_10['count'])
plt.xlabel('Hometown')
plt.ylabel('Number of Players')
plt.title('Top 10 Hometowns with Most Players')
plt.show()
```



2. Figuring which year (Sophomore, Freshman, Junior) produces maximum number of players.

- To figure out which year (Sophomore, Freshman, Junior) produces the maximum number of players, we need to collect data on the players and sort it based on their year.
- Then, we can calculate the total number of players in each year and create a graph to visually represent the data.
- We should analyze the data, consider any external factors that might be affecting it, and draw conclusions about which year produces the maximum number of players and any contributing factors.

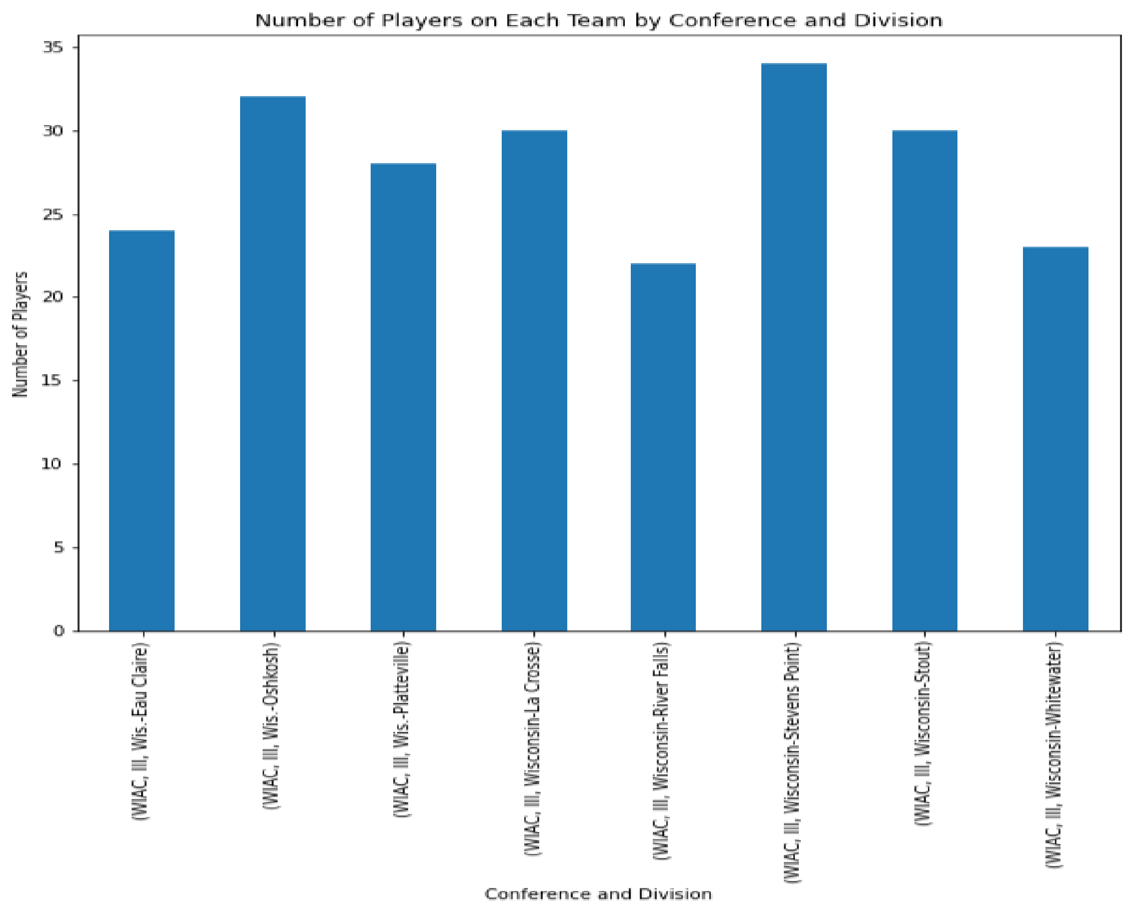
```
plt.bar(year_counts_desc.index, year_counts_desc["count"])
plt.xlabel("Year")
plt.ylabel("Number of Players")
plt.title("Players from Each Year")
plt.show()
```



3. Figuring out how the number of players on each team vary across different conferences and divisions.

- To figure out how the number of players on each team varies across different conferences and divisions, we need to collect data on the players and sort it based on their team's conference and division.
- Then, we can calculate the average number of players per team in each conference and division and create a graph to visually represent the data. We should analyze the data, consider any external factors that might be affecting it, and draw conclusions about how the number of players on each team varies across different conferences and divisions and any contributing factors.

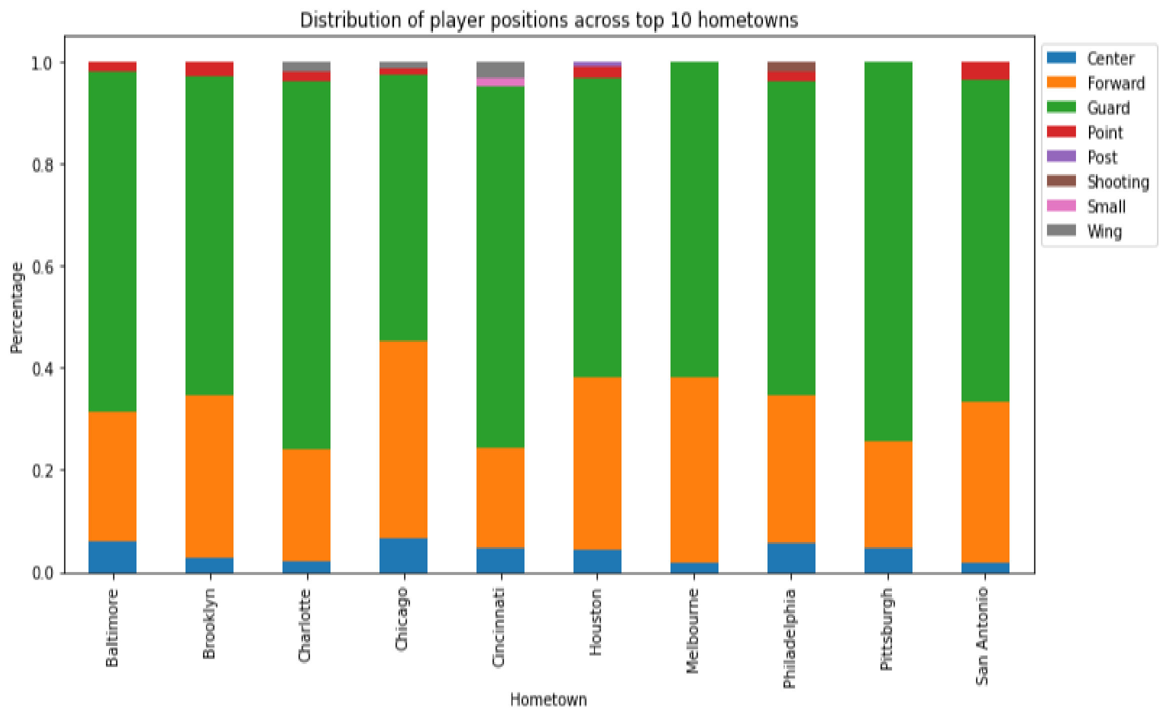
```python
# Group the data by conference and division, and count the number of players on each team
team_counts = df.groupby(["conference", "division", "team"]).size()

# Plot a bar chart of the team counts by conference and division
fig, ax = plt.subplots(figsize=(10, 8))
team_counts.groupby(["conference", "division"]).plot(x="team", y="count", kind="bar", ax=ax)
ax.set_xlabel("Conference and Division")
ax.set_ylabel("Number of Players")
ax.set_title("Number of Players on Each Team by Conference and Division")
plt.show()
```



Data Analysis from the Dataset- wbb_rosters_2022_23

1. **To investigate the distribution of player positions across the top 10 hometowns of athletes.**

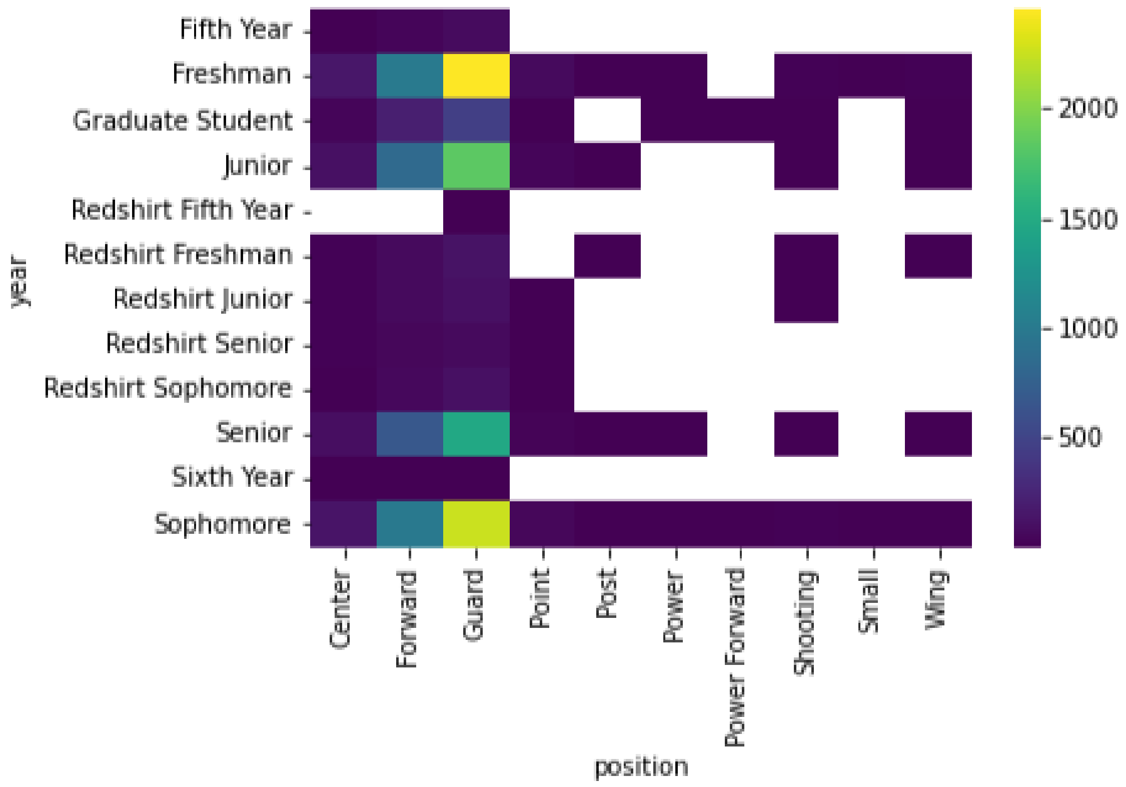
```python
import matplotlib.pyplot as plt

# Get the top 10 hometowns by number of players
top_hometowns = df['hometown'].value_counts().nlargest(10).index.tolist()

# Create a subset of the original dataframe with only the top hometowns
subset = df[df['hometown'].isin(top_hometowns)]

# Group by hometown and position, and count the number of players
grouped = subset.groupby(['hometown', 'position']).size().reset_index(name='count')

# Pivot the data to create a matrix with hometowns as rows, positions as columns, and counts as values
pivoted = grouped.pivot(index='hometown', columns='position', values='count')

# Normalize the values for each hometown to create a percentage distribution
normalized = pivoted.div(pivoted.sum(axis=1), axis=0)

# Create a stacked bar chart for the normalized data
ax = normalized.plot(kind='bar', stacked=True, figsize=(12, 6))

# Add labels and legend
ax.set_xlabel('Hometown')
ax.set_ylabel('Percentage')
ax.set_title('Distribution of player positions across top 10 hometowns')
ax.legend(loc='upper left', bbox_to_anchor=(1, 1))
plt.show()
```



2. **To investigate the correlation between year of the player and their professional position in basketball.**

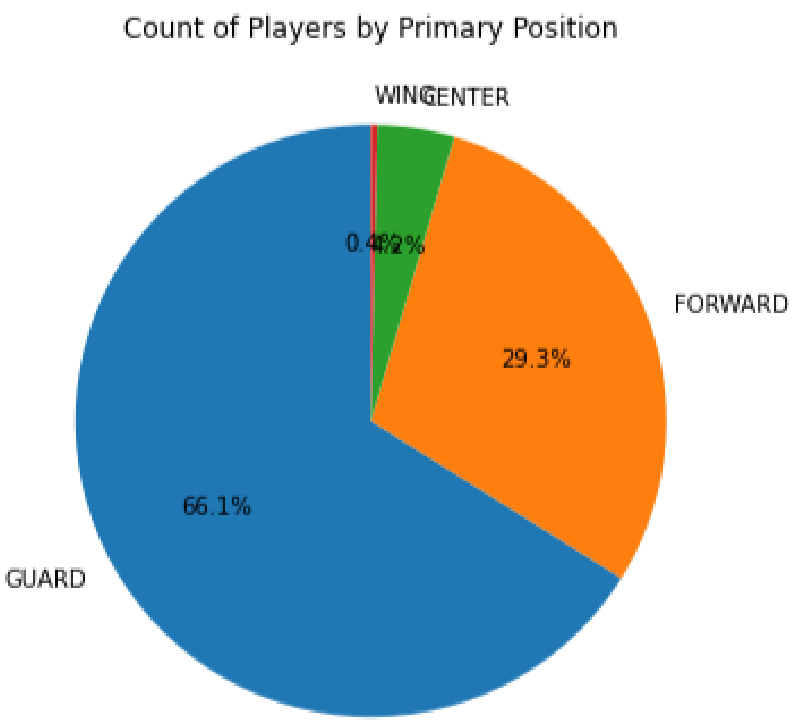
```
import seaborn as sns

# Group the data by year and position, and count the number of players in each group
heatmap_data = df.groupby(["year", "position"]).size().reset_index(name="count")

# Pivot the data to create a matrix where the rows are years, columns are positions, and values are the counts
heatmap_data = heatmap_data.pivot("year", "position", "count")

# Plot the heatmap using seaborn
sns.heatmap(heatmap_data, cmap="viridis")
```

```
<AxesSubplot:xlabel='position', ylabel='year'>
```



3. Pie Chart for count of players by Primary Position

```
position_counts = df['primary_position'].value_counts()

fig, ax = plt.subplots(figsize=(6, 6))
ax.set_title('Count of Players by Primary Position')

ax.pie(position_counts, labels=position_counts.index, autopct='%1.1f%%', startangle=90)

plt.show()
```


Count of Players by Primary Position
WING
CENTER
FORWARD
GUARD