

# NBA Shooting Chart

## Documentation

**Project title:** NBA Individual Shooting Chart 2023/2024

**Tools used:** Python, Tableau

**Author:** Riccardo Kiho

**Date:** December 10, 2024

## Summary

This project analyzes NBA player's shooting stats from 2023/2024 season to evaluate individual performance and identify top players. Using Python for **web scraping** data and Tableau for creating an interactive dashboard which includes shooting chart, shot percentages, and a list of top 10 players by total points and shooting percentages. Users can tailor their view with multiple filters such as *shot type*, *shot made*, *shot range* etc.

## Data

**Dataset:** NBA Shots 2024

**Source:** [https://github.com/DomSamangy/NBA\\_Shots\\_04\\_24](https://github.com/DomSamangy/NBA_Shots_04_24)

**Size:** 218702 rows, 26 columns

## Methods

**Data cleaning:** Used Python to check for duplicate rows - found 14 and dropped them.

```
import pandas as pd

file_path = '/Users/piccardokiho/Desktop/NBA_2024_Shots.csv'
df = pd.read_csv(file_path)

duplicates = df.duplicated()
duplicated_rows = df[duplicates]

print("Duplicated rows:")
print(duplicated_rows)

df_no_duplicates = df.drop_duplicates()
df_no_duplicates.to_csv(file_path, index=False)

print("Duplicates have been removed and the updated "
      "dataset has been saved to the same file.")
```

```
Duplicated rows:
  SEASON_1 SEASON_2 TEAM_ID ... QUARTER MINS_LEFT SECS_LEFT
8032      2024   2023-24  1610612762 ...      4         0        46
8151      2024   2023-24  1610612762 ...      2         9        46
8535      2024   2023-24  1610612762 ...      3         7        33
26527     2024   2023-24  1610612759 ...      2        11        16
32912     2024   2023-24  1610612758 ...      3         2        21
53318     2024   2023-24  1610612755 ...      4         3         1
83014     2024   2023-24  1610612740 ...      3        10         4
83353     2024   2023-24  1610612740 ...      3        10        31
108984     2024   2023-24  1610612743 ...      2         0        28
127216     2024   2023-24  1610612746 ...      1         0         2
161816     2024   2023-24  1610612743 ...      2         2        49
187863     2024   2023-24  1610612741 ...      1         0         6
192245     2024   2023-24  1610612766 ...      1         9         5
212832     2024   2023-24  1610612737 ...      2         0         1

[14 rows x 26 columns]
Duplicates have been removed and the updated dataset has been saved to the same file.

Process finished with exit code 0
```

**Data cleaning:** *Uploaded data to Tableau and checked for data types - all fine!*

**Web Scrapping:** *Since I wanted to display player portraits and team logos on my dashboard, I needed to scrape them from web. Since each NBA player has it's unique ID, I figured out that combining that ID with a base URL, I'm able to get each player's headshot. So I used Python **Pandas** to list all ID's(PLAYER\_ID field) and then **Requests** module to scrape png's.*

Scraping Portraits	Scraping Team Logos
<pre> import os import requests import pandas as pd  base_url = "https://cdn.nba.com/headshots/nba/latest/1040x760/"  # Define the folder to save images folder_path = "headshots"  # Create the folder if it doesn't exist if not os.path.exists(folder_path):     os.makedirs(folder_path)  #Get Id's df = pd.read_csv('/Users/piccardokiho/Desktop/NBA_2024-Shots.csv') id_list = df['PLAYER_ID'].unique().tolist() str_id_list = list(map(str, id_list))  for player_id in str_id_list:     image_url = f"{base_url}{player_id}.png" # Construct the URL     image_path = os.path.join(folder_path, f"{player_id}.png") </pre>	<pre> import os import requests import pandas as pd  folder_path = "team_logos"  #Get Id's df = pd.read_csv('/Users/piccardokiho/Desktop/NBA_2024-Shots.csv') id_list = df['TEAM_ID'].unique().tolist() str_id_list = list(map(str, id_list))  for team_id in str_id_list:     logo_url = f'https://cdn.nba.com/logos/nba/{team_id}/global/L/logo.svg'     image_path = os.path.join(folder_path, f"{team_id}.svg")      try:         response = requests.get(logo_url)         if response.status_code == 200:             with open(image_path, 'wb') as img_file:                 img_file.write(response.content)             print(f"Downloaded {team_id}.png")         else:             print(f"Failed to download {team_id}.png: Status code {response.status_code}")      except requests.exceptions.RequestException as e:         print(f"Error downloading {team_id}.png: {e}")  print("Download process complete!") </pre>
<pre> try:     # Send a GET request to fetch the image     response = requests.get(image_url)      # Check if the request was successful (status code 200)     if response.status_code == 200:         # Write the image content to a file         with open(image_path, 'wb') as img_file:             img_file.write(response.content)         print(f"Downloaded {player_id}.png")     else:         print(f"Failed to download {player_id}.png: Status code {response.status_code}")  except requests.exceptions.RequestException as e:     print(f"Error downloading {player_id}.png: {e}")  print("Download process complete!") </pre>	<p>Later I discovered that Player Portraits are much more efficient to display via web URL in Tableau since there were 568 of them. I just created Calculated field - [base URL] + [Player ID].</p> <p>Team Logos are scraped similarly, but operated as 'Shapes' in Tableau. I moved 'logos' folder with it's png files into 'Tableau Repository/Shapes'</p>

**Visualisation:** *Rest of the project is pure Tableau work. I created multiple new fields to calculate accuracy percentages, total points etc. I had to adjust fields 'Loc X' and 'Loc Y' to exactly match my background image (basketball court) in order to display shot coordinates accurately. I experimented with different layouts and colour pallets until I found suitable.*

## Conclusion

Although this dashboard has the potential to incorporate more metrics and advanced functionality, I'm happy with its current state. The primary objective was to explore various Tableau features and improve my data visualization skills.

.