Filtering Our Data

This notebook has been adapted from...

https://github.com/callysto/basketball-and-data-science/blob/main/content/01-introduction.ipynb, with permmission.

(Open in Callysto | Colab)

Let’s Get Our Data

In [ ]:

```
import pandas as pd

# URL of the CSV file containing data for Pascal Siakam
url = 'https://raw.githubusercontent.com/pbeens/Data-Dunkers/main/Data/Pascal_Siaka

# Read the CSV file into a pandas DataFrame
df = pd.read_csv(url)

# Display the DataFrame
display(df)
```

Only Include Seasons with the Raptors

There are a few things you will notice from the data above. The first is that it includes data from the year Pascal Siakam was traded to the Indiana Pacers, as well as a line that has totals and averages for his career. To eliminate these, we can simply make sure we only include anything up to and including the 2022-23 season.

In [ ]:

```
# Filter the DataFrame to exclude seasons after 2022-23
df = df[df['SEASON_ID'] <= '2022-23']
display(df)
```

Reducing the Number of Columns

Let's reduce the number of columns to make it easier to work with. To do this we set up a filter, first by specifying the columns we want to keep, then using that filter to filter the DataFrame.

In [ ]:

```
# Filter the DataFrame to only include specific columns
columns_to_keep = [
    'SEASON_ID','PLAYER_AGE', 'GP', 'GS', 'MIN', 'AST', 'STL', 'FGM', 'FGA', 'FG_PC
    'FG3A', 'FG3_PCT', 'FTM', 'FTA', 'FT_PCT', 'PTS', 'FG2M', 'FG2A', 'FG2_PCT'
]
```

```
df = df[columns_to_keep]
display(df)
```

Free Throw Percentage > 75%

We can filter the data to only display rows where Pascal Siakam's free throw percentage was greater than 75%.

In [ ]:

```
filter = df['FT_PCT'] > 0.75 # free throw % above 75%
display(df[filter])
```

...and When He Started Every Game

What if we only want to display the seasons where he started every game. Logically, that would be where the number of games played (GS) equals the number of games started (GS). We can filter using `==` which means "is equal to".

In [ ]:

```
filter = df['GS'] == df['GP'] # games started equals games
display(df[filter])
```

It is even possible to include multiple conditions. " `&` " means "and".

In [ ]:

```
filter_1 = df['FT_PCT'] > 0.75 # free throw % above 75%
filter_2 = df['GS'] == df['GP'] # games started equals games
display(df[filter_1 & filter_2])
```

These are the symbols we use for comparison operations in Python:

Symbol	Meaning
>	greater than
<	less than
==	is equal to
!=	not equal to
>=	greater than or equal to
<=	less than or equal to
&	and
|	or

Exercise

In the cell below, use code to display only rows where Assists Per Game ('AST') was greater than 5 or Steals Per Game ('STL') was greater than 1.

In [ ]:

```
import pandas as pd

url = 'https://raw.githubusercontent.com/pbeens/Data-Dunkers/main/Data/Pascal_Siaka

df = pd.read_csv(url)

# Filter the DataFrame to exclude seasons after 2022-23
df = df[df['SEASON_ID'] <= '2022-23']

# filter_1 =
# filter_2 =
display(df)
```

Next Lesson: Sorting Data (GitHub link)