Luciana Solorzano, lusolorz@umich.edu

Link to repo: https://github.com/mandykim22/SI206_Final.git

1. The goals for your project including what APIs/websites you planned to work with and what data you planned to gather (10 points)

For the first part of the project we planned to work with the Sports Radar API and collect players on every team for the March Madness tournament and create our own bracket based on players' stats on each team. We planned to gather players for each team and stats for each player.

2. The goals that were achieved including what APIs/websites you actually worked with and what data you did gather (10 points)

We decided to scrape this wikipedia page to gather the list of teams playing in the tournament. We then collected the links available for each team and scraped the corresponding 64 links to get the players' names and heights (example link for Alabama Crimson Tide). We then organized teams by regions and created a pie chart for the tallest teams and the lowest teams. We also created a bar chart for the average height of each team.

3. The problems that you faced (10 points)

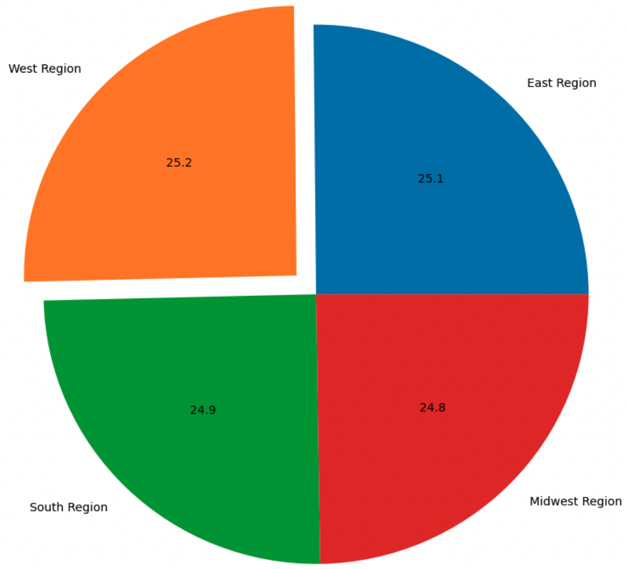
For the first part of the project the API we decided to work with had a QPS limit of 1 so we were unable to gather the ~980 player stats. We then decided to switch to scraping a website to retrieve relevant data. Because there were no player stats on the relevant wikipedia pages, we decided to collect the players' heights instead and change the visualization from a bracket to a pie chart and bar graph.

4. The calculations from the data in the database (i.e. a screen shot) (10 points)

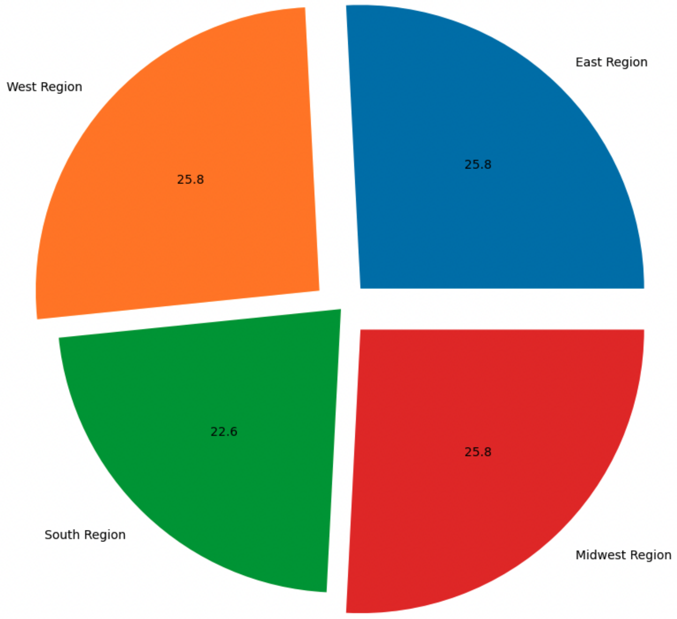
Calculation for March Madness (too large to take a screenshot…copied from txt file generated to google doc)

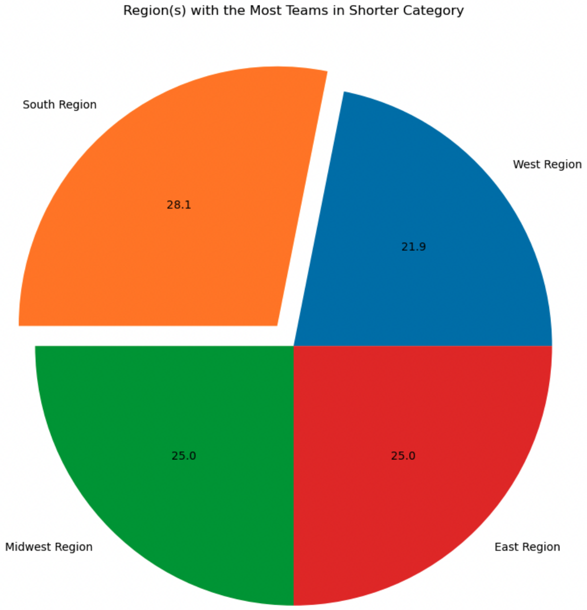
5. The visualization that you created (i.e. screen shot or image file) (10 points)

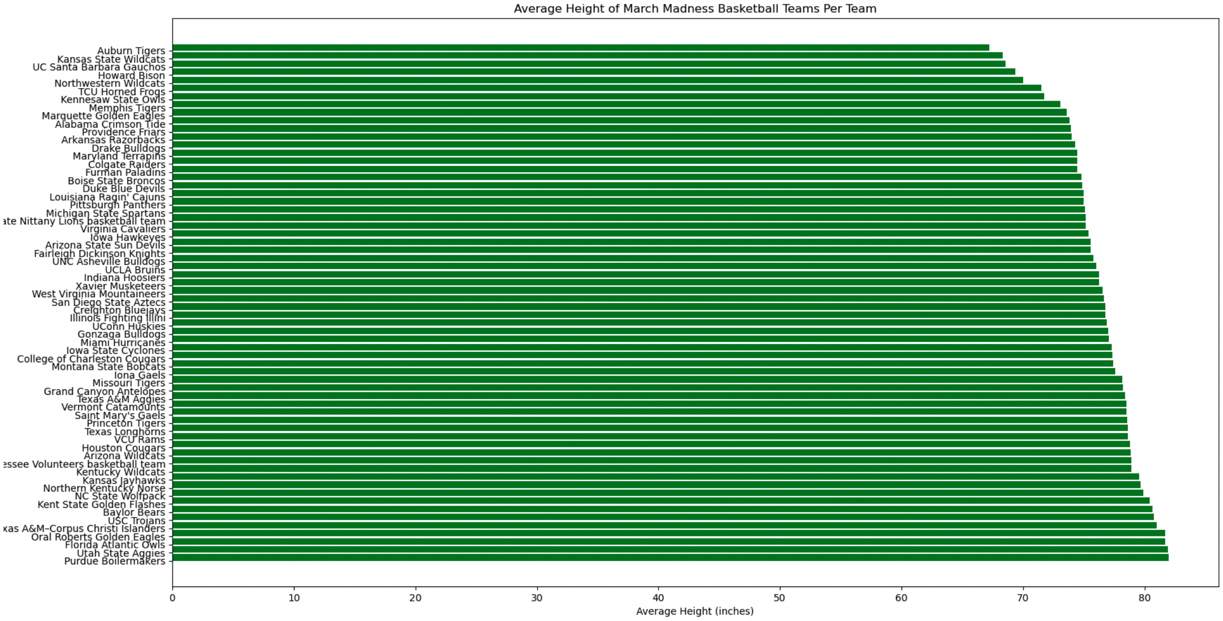
Percentage Per Region of Overall Height
Across all March Madness Teams
(The Tallest Region)



Region(s) with the Most Teams in Taller Category







6. Instructions for running your code (10 points)

Our file for filling the database has to be run 70 times as we have to retrieve ~980 data values. In order to automate this process, we created a bashscript that is included in our repository (titles ‘march_madness_fill_db.sh’). This bashscript also handles running our march_madness_calcs.py. Running this command:

./march_madness_fill_db.sh march_madness_scrape.py

Should run our entire program regarding march madness including providing visualization and writing to our calcs txt file. You will need to exit out of the visualization windows in order to continue with the program.

7. Documentation for each function that you wrote. This includes describing the input and output for each function (20 points)

march_madness_scrape.py:

```
def get_teams_playing(url, conn, cur):
```

Input: url for wikipedia page with March Madness Bracket
Function: scrapes wikipedia page for team names of 64 starting teams for March Madness tournament. Also collects region and team wiki page link
Output: returns list of tuples with team name, id, region, and link to team wiki page

```
def open_db(db):
```

Input:database name
Function: Sets conn and cur for March_Madness database/connects to database
Output: returns conn, cur

```
def creat_table_regions(conn, cur):
```

Input: conn, cur
Function: creates Regions tables and inserts 4 regions into table
Output: N/A

```
def create_teams_table(conn, cur):
```

Input: con, cur
Function: creates Teams_scrape table
Output: N/A

```
def create_table_players(conn, cur):
```

Input: conn, cur
Function: creates Players tables
Output: N/A

```
def add_to_teams_table(teams, conn, cur):
```

Input: teams (returned from get_teams_playing, conn, cur
Function: Adds teams to table Teams_scrape with default team_chance = 0 (updated later)
Output: N/A

```
def insert_into_players(conn, cur):
```

Input: conn, cur
Function: Add players to Player table
Output: N/A

march_madness_calcs.py

```
def open_db(db):
```

Same as previous open_db

```
def update_team_chance_col(conn, cur):
```

Input: conn, cur
Function: updates the team_chance column in Teams_scrape table with the average height of the team based on values in the Players table
Output: N/A

```
def write_regions_to_txt(conn, cur):
```

Input: conn, cur
Function: Writes region and all associated teams to calcs.txt for all regions
Output: calcs.txt file with regions data written

```
def write_rosters_to_txt(conn, cur):
```

Input: conn, cur
Function: Writes team and all associated players to calcs.txt for each team along with the average height of the team
Output: calcs.txt file with teams data (rosters) written

```
def visualize_write_team_heights_average(conn, cur):
```

Input: conn, cur
Function: Creates pie charts to determine the tallest region, the region(s) with the most teams in the list of taller teams, and the region with the most teams in the list of shorter teams. Also creates a bar chart with Teams and their average heights
Output: 3 pie charts and one bar chart

8. You must also clearly document all resources you used. The documentation should be of the following form (20 points)

Date	**Issue Description**	**Location of Resource**	**Result**
4/19/23	Exceeded QPS Limit	https://stackoverflow.com/questions/38601326/google-cloud-pub-sub-on-google-app-engine-hits-qps-limit-too-soon	Switched to webscraping
4/19/23	How to create bash script for loop	https://www.cyberciti.biz/faq/bash-for-loop/	Used example from link
4/20/23	Matplot lib not functioning to show plots	https://stackoverflow.com/questions/66121948/matplotlib-plots-not-showing-in-vs-code	Ran: pip install matplotlib --upgrade
4/21/23	How to plot pie chart with excluded points	https://www.analyticsvidhya.com/blog/2020/02/beginner-guide-matplotlib-data-visualization-exploration-python/	Used code from link and changed dictionary value
4/21/23	List names on bar graph without '2022-23'	https://www.tutorialspoint.com/How-to-match-any-non-digit-character-in-Python-using-Regular-Expression	Used re with `pattern= [r'\D+']`