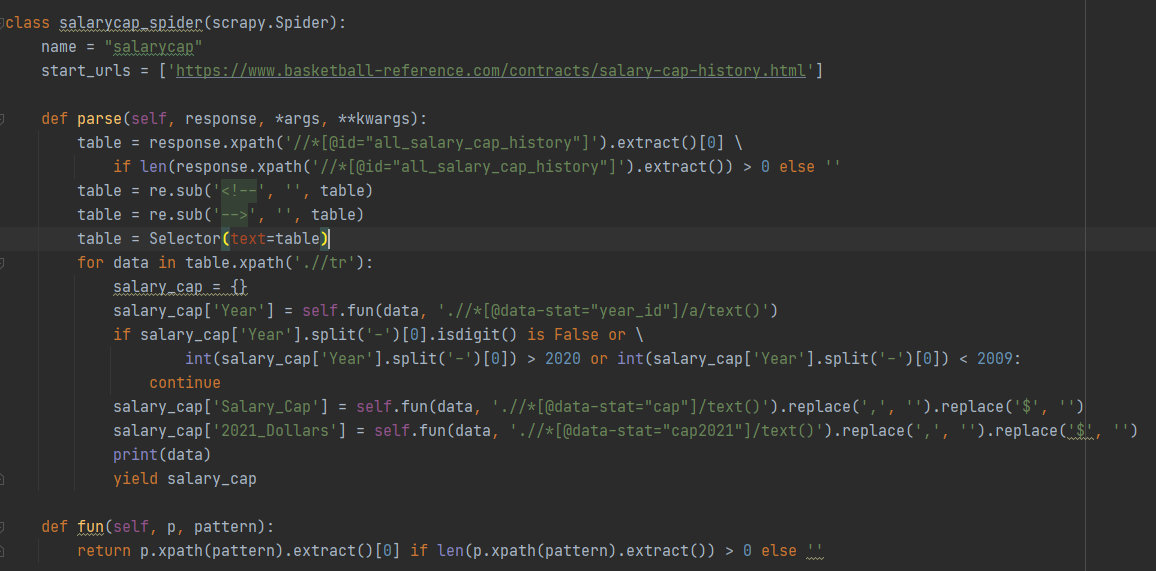
## Part A

1. **https://www.basketball-reference.com/players/a/**. For each player listed on this page, my crawler will enter his detail page and acquire more information if his last year is larger than 2009. For each table in player’s detail page, I set one specific crawler to get his information using the method aforementioned. **https://www.basketball-reference.com/teams/**. For each team listed on this page, my crawler will enter his detail page and acquire more information. For each table in team’s detail page, I set one specific crawler to get his information using the method aforementioned.
2. Here is a part of my code, you can see it in detail under folder **/spider/NBA/spiders*.***



1. For each player, I parse his basic information, performance per game, total performance, performance per 36 minutes, performance per 100 poss, advanced performance, adjusted shooting data, play-by-play data, shooting data, game highs, playoff series and his salaries. There are 12 tables in total. You can find them in fold **/data.**
2. For each team, I parse basic team information, index of seasonal statistics, basic seasonal statistics, basic seasonal statistics of opponent, seasonal statistics ranks, seasonal statistics ranks of opponent, seasonal statistics year by year, seasonal statistics year by year of opponent. There are 9 tables in total. You can find them in fold **/data.**
3. Salary cap information is useful when evaluating players’salary(https://www.basketball- reference.com/contracts/salary-cap-history.html). The reason is that salary cap is not only an indicator of upper limit of salary, it also shows the average level or the development of the league and nation’s economy. If salary cap is high, it means that players earn more as a whole. The code is showed below which is in **salarycap\_spider.py**.



## Part B

1. Merge the data from salary.csv, p36m.csv, p100p.csv, pergame.csv, total.csv, advanced.csv, team\_seasonal\_index.csv

Data clean and transform

* Intercept the start year of the season as season data. e.g. 2009-10 -> 2009
* Remove playoff data
* Use the Team attribute to unify the expression of team names
* Fill the missing value of Playoffs field with ‘Not in’
* Drop all rows with null values
* For salary variable, replace ‘< minimum’ with minimum salary
* Convert stats and salary to INT

Note: You can find the code in **partb.py**

Data dictionary

* Name: player name
* Born: date of birth, YYYY-MM-DD
* Is\_playoff: distinguish regular season and playoff stats. playoff if=1
* Season: year of Season, YYYY
* Age: age of player
* Team: full team name of player
* Lg: league name
* Pos: position of player
* G: games played
* GS: games started
* MP: minutes played per 36 minutes
* FG: field goals per 36 minutes
* FGA: field goals attempts per 36 minutes
* FG%: field goal percentage
* 3P: 3-point field goals per 36 minutes
* 3PA: 3-point field goals attempts per 36 minutes
* 3P%: 3-point field goals percentage
* 2P: 2-point field goals per 36 minutes
* 2PA: 2-point field goals attempts per 36 minutes
* 2P%: 2-point field goals percentage
* FT: free throws per 36 minutes
* FTA: free throws attempts per 36 minutes
* FT%: free throws percentage
* ORB: offensive rebounds per 36 minutes
* DRB: defensive rebounds per 36 minutes
* TRB: total rebounds per 36 minutes
* AST: assists per 36 minutes
* STL: steals per 36 minutes
* BLK: blocks per 36 minutes
* TOV: turnovers per 36 minutes
* PF: personal fouls per 36 minutes
* PTS: points per 36 minutes
* Salary: salary
* TRB\_p100p: total rebounds per 100 team possessions
* AST\_p100p: assist per 100 team possessions
* PTS\_p100p: points per 100 team possessions
* TRB\_pergame: total rebounds per game
* AST\_pergame: assist per game
* PTS\_pergame: points per game
* PER: player efficiency rating - a measure of per-minute production standardized such that the league average is 15
* TS%: true shooting percentage - a measure of shooting efficiency that take into account 2-point field goals, 3-point field goals, and free throws.
* WS: win shares – an estimates of the number of wins contributed by a player.
* VORP: value over replacement player – a box score estimate of the points per 100 TEAM possessions that a player contributed above a replacement-level(-2.0) player, translated to an average team and proated to an 82-game season.
* W/L%: win-loss percentage
* SRS: simple rating system; a team rating that takes into account average point differential and strength of schedule. The rating is denominated in points above/below average, where zero is average
* Playoffs: the result of playoffs. If not in=not in playoff this year

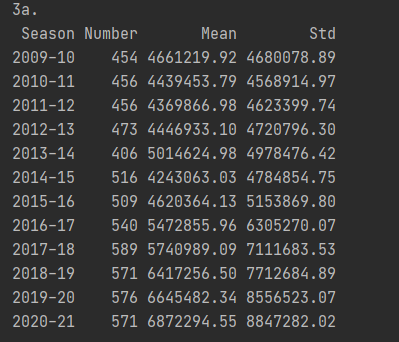
1. a) 571 (Number of players that appears in salary table in 2020-21 season)

b) SG: 199, PG: 139, C: 133, PF: 178, SF: 177

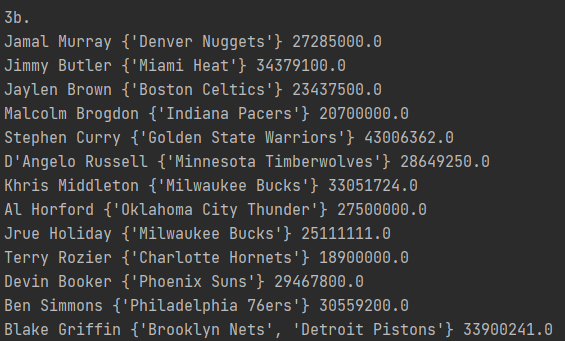
c) average age: 26.20, average weight: 217.16lb, average experience: 4.12 years, average salary: 6872294.55.

d) 22647206.79

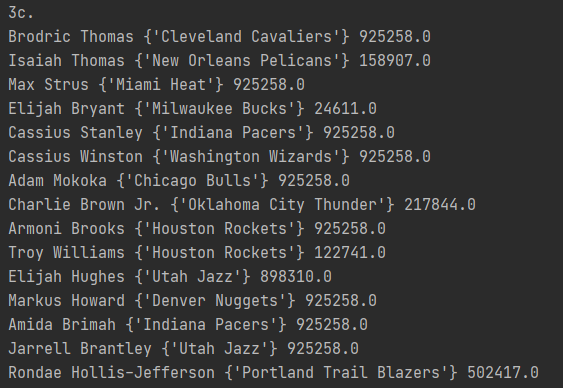
1. a) The result is showed below.



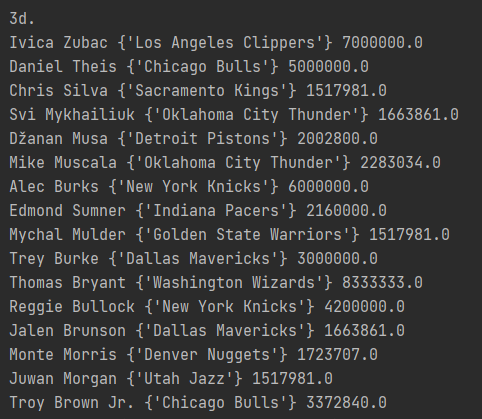
b) Part of the result is showed below. You can see the full answer in **partb.py**.



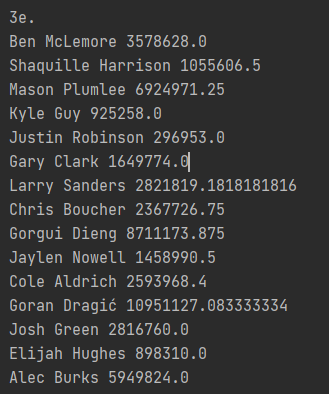
c) Part of the result is showed below. You can see the full answer in **partb.py**. Some of these players have the salary less than minimum. I change it to 925258 according to Internet.



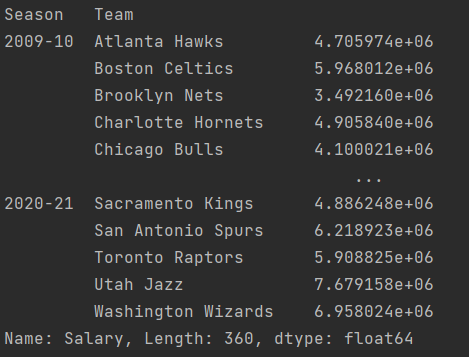
d) Part of the result is showed below. You can see the full answer in **partb.py**.



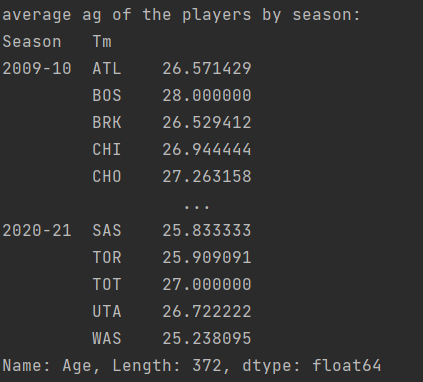
e) Part of the result is showed below. You can see the full answer in **partb.py**.



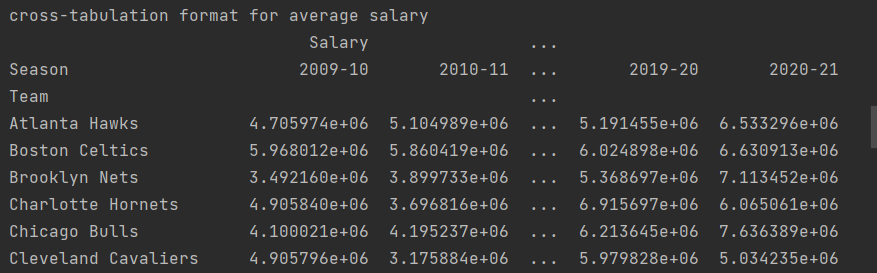
1. a) Part of the result is showed below. You can see the full answer in **partb.py**. And the result file saved in **/output/avg\_salary.csv.**



b) Part of the result is showed below. You can see the full answer in **partb.py**. And the result file saved in **/output/avg\_age.csv** and **/output/avg\_var\_exp.csv**.



c) Part of the result is showed below. You can see the full answer in **partb.py**. And the result file saved in **/output/avg\_salary\_ct.csv, /output/avg\_age\_ct.csv, /output/avg\_ exp\_ct.csv** and **/output/var\_exp\_ct.csv**.



1. Salary cap. The table of summary statistics is showed below.

