

Project Overview:

In this project the group based their report upon WNBA sports data to compare the WNBA season between 2018 and 2019.

Full jupyter notebook can be found [here](#).

Extracting the data:

We extracted four CSV files from Sports Reference's [WNBA data](#).

1. [2019 Game Data](#)
2. [2019 Player Data](#)
3. [2018 Game Data](#)
4. [2018 Player Data](#)

Data Cleanup, Analysis, and Transformation:

Player Data

The payer data was summarized by creating a DataFrame with only 9 columns that we wanted to evaluate for both seasons of 2018 and 2019. Additionally, to be able to match our schema in SQL the headers were updated and a Season column was added to further join analysis.

2018 Player Data reduction

Extract "Player", "Tm", "Pos", "G", "FG", "FG%", "FT", "FT%" and "PTS"

```
reduced_player18_df = player_2018_df.loc[:, ["Player", "Tm", "Pos", "G", "FG", "FG%", "FT", "FT%", "PTS"]]
```

```
reduced_player18_df.head(10)
```

	Player	Tm	Pos	G	FG	FG%	FT	FT%	PTS
0	Natalie Achonwa	IND	C	34	137	0.527	76	0.800	350
1	Kayla Alexander	IND	C	30	33	0.541	14	0.824	80
2	Lindsay Allen	LVA	G	24	28	0.384	17	0.708	74
3	Rebecca Allen	NYL	F	28	38	0.376	21	0.840	107
4	Ariel Atkins	WAS	G	29	120	0.432	42	0.824	327
5	Seimone Augustus	MIN	G	33	156	0.467	24	0.706	357
6	Rachel Banham	CON	G	33	55	0.414	33	0.868	173
7	Alana Beard	LAS	G-F	30	51	0.392	17	0.810	121
8	Hind Ben Abdelkader	IND	G	14	8	0.186	7	0.875	29
9	Alex Bentley	ATL	G	16	56	0.376	9	0.529	139

```
# 2019 Player Data reduction
```

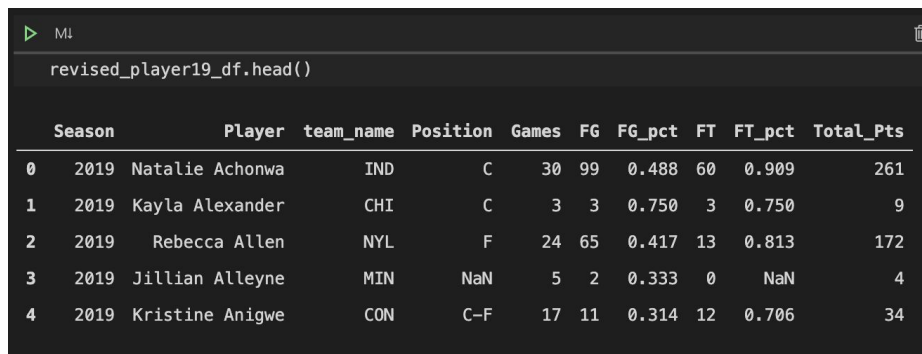
```
# Extract "Player", "Tm", "Pos", "G", "FG", "FG%", "FT", "FT%" and "PTS"
```

```
reduced_player19_df = player_2019_df.loc[:, ["Player", "Tm", "Pos", "G", "FG", "FG%", "FT",  
"FT%", "PTS"]]
```

```
reduced_player19_df.head(10)
```

	Player	Tm	Pos	G	FG	FG%	FT	FT%	PTS
0	Natalie Achonwa	IND	C	30	99	0.488	60	0.909	261
1	Kayla Alexander	CHI	C	3	3	0.750	3	0.750	9
2	Rebecca Allen	NYL	F	24	65	0.417	13	0.813	172
3	Jillian Alleyne	MIN	NaN	5	2	0.333	0	NaN	4
4	Kristine Anigwe	CON	C-F	17	11	0.314	12	0.706	34
5	Kristine Anigwe	DAL	C-F	10	11	0.333	10	0.667	32
6	Kristine Anigwe	TOT	C-F	27	22	0.324	22	0.688	66
7	Ariel Atkins	WAS	G	33	123	0.416	43	0.811	340
8	Seimone Augustus	MIN	G	12	20	0.313	3	0.750	45
9	Rachel Banham	CON	G	29	37	0.322	9	0.692	105

Final DF example:



```
revised_player19_df.head()
```

	Season	Player	team_name	Position	Games	FG	FG_pct	FT	FT_pct	Total_Pts
0	2019	Natalie Achonwa	IND	C	30	99	0.488	60	0.909	261
1	2019	Kayla Alexander	CHI	C	3	3	0.750	3	0.750	9
2	2019	Rebecca Allen	NYL	F	24	65	0.417	13	0.813	172
3	2019	Jillian Alleyne	MIN	NaN	5	2	0.333	0	NaN	4
4	2019	Kristine Anigwe	CON	C-F	17	11	0.314	12	0.706	34

Game Data

Since we were using four CSV files, we transform the game data into DataFrames as well by removing the box score column from the data. In addition, column headers were updated to correspond with the schema in SQL and added a Season column for further joins and analysis.

```
# 2018 Game Data reduction
```

```
# Extract "Date", "Visitor/Neutral", "PTS", "Visitor/Neutral", "PTS.1"
```

```
reduced_game18_df = wnba_2018_df.loc[:, ["Date", "Visitor/Neutral", "PTS", "Home/Neutral",
"PTS.1"]]
reduced_game18_df.head(10)
```

We used SQL schema to run queries comparing season 2018 with 2019 by updating the Game Data DFs so team names match by the 3 letter acronyms on the Player Data DFs.

#Replace each full team name with 3-letter acronym (i.e. DallasWings --> DAL)

```
replacements = {
    "Dallas Wings": "DAL",
    "Chicago Sky": "CHI",
    "New York Liberty": "NYL",
    "Las Vegas Aces": "LVA",
    "Atlanta Dream": "ATL",
    "Los Angeles Sparks": "LAS",
    "Phoenix Mercury": "PHO",
    "Seattle Storm": "SEA",
    "Indiana Fever": "IND",
    "Washington Mystics": "WAS",
    "Minnesota Lynx": "MIN",
    "Connecticut Sun": "CON",
}
reduced_game18_df["Visitor/Neutral"].replace(replacements, inplace=True)
reduced_game18_df["Home/Neutral"].replace(replacements, inplace=True)
reduced_game19_df["Visitor/Neutral"].replace(replacements, inplace=True)
reduced_game19_df["Home/Neutral"].replace(replacements, inplace=True)
```

	Date	Visitor/Neutral	PTS	Home/Neutral	PTS.1
0	Fri, May 24, 2019	Dallas Wings	72	Atlanta Dream	76
1	Fri, May 24, 2019	Indiana Fever	81	New York Liberty	80



	Date	Visitor/Neutral	PTS	Home/Neutral	PTS.1
0	Fri, May 24, 2019	DAL	72	ATL	76
1	Fri, May 24, 2019	IND	81	NYL	80

Final DF example:

```
ML
revised_game18_df.head()
```

	Season	Date	away_team	away_team_pts	home_team	home_team_pts
0	2018	Fri, May 18, 2018	DAL	78	PHO	86
1	2018	Sat, May 19, 2018	CHI	82	IND	64
2	2018	Sun, May 20, 2018	NYL	76	CHI	80
3	2018	Sun, May 20, 2018	LVA	65	CON	101
4	2018	Sun, May 20, 2018	ATL	78	DAL	101

Load: the final database, tables/collections, and why this was chosen.

We imported pandas and SQL Alchemy via Jupyter Notebook to show the files. SQL was used with this analysis because the data is considered to be a Relational Databases.

This was chosen because the data that we were able to find via Sports Reference maintained consistency and year over year the same data points are tracked. This database would allow for easy joins and allow for many different metrics to be compared over time.

See SQL schema below:

Game_Data_2018 season INT date DATE home_team varchar(50) away_team varchar(50) home_team_pts INT away_team_pts INT	Game_Data_2019 season INT date DATE home_team varchar(50) away_team varchar(50) home_team_pts INT away_team_pts INT
Player_Profile_2019 Season INT Player varchar(50) team_name varchar(50) position varchar(50) Games INT FG INT FG_pct INT FT INT FT_pct INT Total_Pts INT	Player_Profile_2018 Season INT Player varchar(50) team_name varchar(50) position varchar(50) Games INT FG INT FG_pct INT FT INT FT_pct INT Total_Pts INT