PRACTICE CASE

STATISTIC

Link Github :

https://github.com/dhea1323/IYKRA-DheaFajriatiAnas_PracticeCaseStatistics.git

by : Dhea Fajriati Anas

SCENARIO

The data-set contains aggregate individual statistics for 67 NBA seasons. from basic box-score attributes such as points, assists, rebounds etc., to more advanced money-ball like features such as Value Over Replacement.

In this task not all data will be used, only data in 2017. So it is necessary to do filtering at the beginning. Besides that there are some players who make team transfers in the NBA transfer market so that there is duplication of player data. Therefore you can use the df.drop_duplicates() syntax to solve this to produce the same output as the trainer. Delete columns that have as many missing values as the entire row of data. Then you can do additional preprocessing if needed or you can immediately process the data.

Some goals of this project:

1. Who is the youngest and oldest player in the NBA in 2017 for each team (Tm) ?
2. Which player has the most minutes played (MP) in each position (Pos)?
3. Which team has the highest average total rebound percentage (TRB%), assist percentage (AST%), steal percentage (STL%), and block percentage (BLK%)?
4. Who is the best player in your opinion based on his record stats? note: you can refer to variables point (PTS), assists, rebounds, or anything else. A combination of several variables would be nice.
5. Which team has the best average stat record of their players? Note: you can refer to points, assists, rebounds, or anything else. A combination of several variables would be nice

DATASET VARIABLE (1)

- **Year:** Since the NBA season is split over two calendar years, the year given is the last year for that season. So, the year for the 1999-00 season would be 2000.
- **Player:** A player's name.
- **Pos:** Position (Center, Power Forward, Small Forward, Shooting Guard, Point Guard)
- **Age** - Age; player age on February 1 of the given season.
- **Tm:** The team, in which player rostered in a given season. A player can have more than a team in a season due to the frequent trade. For further information about team abbreviations and history visit this link.
- **G** (**Games**): Number of games played in a season
- **GS** (**Game Started**): Number of games played in a season as a Starting Lineups (available since the 1982 season).
- **MP** (**Minutes Played**): Total minutes of play in a season (available since the 1951-52 season)
- **PER:** The player efficiency rating (PER) is a rating of a player's per-minute productivity
- **TS%** - True Shooting Percentage; the formula is PTS / (2 * TSA). True shooting percentage is a measure of shooting efficiency that takes into account field goals, 3-point field goals, and free throws.
- **3PAr** = (3PA / FGA), and is a measure of what % of a player's shots come from long-distance, another good gauge of how they're utilized offensively
- **FTR (Free Throw Rate)** : refers to a team's or player's ability to draw fouls, get to the line, and ultimately make those free throw attempts.
- **ORB%** - Offensive Rebound Percentage (available since the 1970-71 season in the NBA); the formula is 100 * (ORB * (Tm MP / 5)) / (MP * (Tm ORB + Opp DRB)). Offensive rebound percentage is an estimate of the percentage of available offensive rebounds a player grabbed while he was on the floor.
- **DRB%** - Defensive Rebound Percentage (available since the 1970-71 season in the NBA); the formula is 100 * (DRB * (Tm MP / 5)) / (MP * (Tm DRB + Opp ORB)). Defensive rebound percentage is an estimate of the percentage of available defensive rebounds a player grabbed while he was on the floor.
- **TRB%** - Total Rebound Percentage (available since the 1970-71 season in the NBA); the formula is 100 * (TRB * (Tm MP / 5)) / (MP * (Tm TRB + Opp TRB)). Total rebound percentage is an estimate of the percentage of available rebounds a player grabbed while he was on the floor.
- **AST%** - Assist Percentage (available since the 1964-65 season in the NBA); the formula is 100 * AST / (((MP / (Tm MP / 5)) * Tm FG) - FG). Assist percentage is an estimate of the percentage of teammate field goals a player assisted while he was on the floor.

DATASET VARIABLE (2)

- **STL%** - Steal Percentage (available since the 1973-74 season in the NBA); the formula is 100 * (STL * (Tm MP / 5)) / (MP * Opp Poss). Steal Percentage is an estimate of the percentage of opponent possessions that end with a steal by the player while he was on the floor.
- **BLK%** - Block Percentage (available since the 1973-74 season in the NBA); the formula is 100 * (BLK * (Tm MP / 5)) / (MP * (Opp FGA - Opp 3PA)). Block percentage is an estimate of the percentage of opponent two-point field goal attempts blocked by the player while he was on the floor.
- **TOV%** - Turnover Percentage (available since the 1977-78 season in the NBA); the formula is 100 * TOV / (FGA + 0.44 * FTA + TOV). Turnover percentage is an estimate of turnovers per 100 plays.
- **Usg%** - Usage Percentage (available since the 1977-78 season in the NBA); the formula is 100 * ((FGA + 0.44 * FTA + TOV) * (Tm MP / 5)) / (MP * (Tm FGA + 0.44 * Tm FTA + Tm TOV)). Usage percentage is an estimate of the percentage of team plays used by a player while he was on the floor.
- **OWS** - Offensive Win Shares; please see the article Calculating Win Shares for more information.
- **DWS** - Defensive Win Shares; please see the article Calculating Win Shares for more information.
- **WS** - Win Shares; an estimate of the number of wins contributed by a player. Please see the article Calculating Win Shares for more information.
- **WS/48** - Win Shares Per 48 Minutes (available since the 1951-52 season in the NBA); an estimate of the number of wins contributed by the player per 48 minutes (league average is approximately 0.100). Please see the article Calculating Win Shares for more information.
- **BPM** - Box Plus/Minus (available since the 1973-74 season in the NBA); a box score estimate of the points per 100 possessions that a player contributed above a league-average player, translated to an average team. Please see the article About Box Plus/Minus (BPM) for more information.
- **VORP** - Value Over Replacement Player (available since the 1973-74 season in the NBA); 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 prorated to an 82-game season. Multiply by 2.70 to convert to wins over replacement. Please see the article About Box Plus/Minus (BPM) for more information.
- **FG** - Field Goals (includes both 2-point field goals and 3-point field goals)
- **FGA** - Field Goal Attempts (includes both 2-point field goal attempts and 3-point field goal attempts)

DATASET VARIABLE (3)

- **FG%** - Field Goal Percentage; the formula is FG / FGA.
- **3P** - 3-Point Field Goals (available since the 1979-80 season in the NBA)
- **3PA** - 3-Point Field Goal Attempts (available since the 1979-80 season in the NBA)
- **3P%** - 3-Point Field Goal Percentage (available since the 1979-80 season in the NBA); the formula is 3P / 3PA.
- **2P** - 2-Point Field Goals
- **2PA** - 2-Point Field Goal Attempts
- **2P%** - 2-Point Field Goal Percentage; the formula is 2P / 2PA.
- **eFG%** - Effective Field Goal Percentage; the formula is (FG + 0.5 * 3P) / FGA. This statistic adjusts for the fact that a 3-point field goal is worth one more point than a 2-point field goal. For example, suppose Player A goes 4 for 10 with 2 threes, while Player B goes 5 for 10 with 0 threes. Each player would have 10 points from field goals, and thus would have the same effective field goal percentage (50%).
- **FT** - Free Throws
- **FTA** - Free Throw Attempts
- **FT%** - Free Throw Percentage; the formula is FT / FTA.
- **ORB** - Offensive Rebounds (available since the 1973-74 season in the NBA)
- **DRB** - Defensive Rebounds (available since the 1973-74 season in the NBA)
- **TRB** - Total Rebounds (available since the 1950-51 season)
- **AST** - Assists
- **STL** - Steals (available since the 1973-74 season in the NBA)
- **BLK** - Blocks (available since the 1973-74 season in the NBA)
- **TOV** - Turnovers (available since the 1977-78 season in the NBA)
- **PF** - Personal Fouls
- **PTS** - Points

HANDS-ON

DATA PRE-PROCESSING

Does the data have missing values or not?

```
df.isnull().sum()
```

```
Unnamed: 0      0
Year            0
Player          0
Pos             0
Age             0
Tm              0
G               0
GS              0
MP              0
PER             0
TS%             2
3PAr            2
FTr             2
ORB%            0
DRB%            0
TRB%            0
AST%            0
STL%            0
BLK%            0
TOV%            2
USG%            0
blanl         595
OWS             0
DWS             0
WS              0
WS/48           0
blank2        595
OBPM            0
DBPM            0
BPM             0
VORP            0
FG              0
FGA             0
FG%             2
3P              0
3PA             0
3P%            46
2P              0
2PA             0
2P%             5
eFG%            2
FT              0
FTA             0
FT%            24
ORB             0
DRB             0
TRB             0
AST             0
STL             0
BLK             0
TOV             0
PF              0
PTS             0
dtype: int64
```

How to handle missing values?

1. Drop **blanl and blank2** column because the column is 'blank' according to the name.
2. For the following variables: **TS%, 3PAr, FTr, TOV%, FG%, 3P%, 2P%, eFG%, FT%,** previously I want to fill in the null values with their respective formulas, like 3PAr = 3PA/FGA, etc. But, it didn't works and still null values. When I checked the column data that being used for their formula, surprisingly all of them were 0. Thus, I concluded that if the result is 0 then the column will be filled with null. CMIIW. **So, I filled all null values with 0 manuallly**.

   Ex.

```
#FTR (Free Throw Rate) = FTA/FGA
df[['FTA','FGA']][df['FTr'].isnull()]
```

	FTA	FGA
60	0.0	0.0
248	0.0	0.0

```
df['FTr'] = df['FTr'].fillna(0)
```

HANDS-ON

DATA PRE-PROCESSING

Does the data have duplicates value?

```
#Check duplicate values
df['Player'].value_counts()
```

```
Ersan Ilyasova      4
Lance Stephenson    4
Omri Casspi         4
Mike Dunleavy       3
Taj Gibson          3
                   ..
Jarrod Uthoff       1
Andre Drummond      1
Jodie Meeks         1
Bobby Brown         1
James Young         1
Name: Player, Length: 486, dtype: int64
```

How to handle duplicates value?

Drop duplicates except for the last occurrence

```
#Drop duplicate values
df_dup = df.drop_duplicates(['Player'], keep='last')
```

QUESTION 1

Who is the youngest and oldest player in the NBA in 2017 for each team (Tm)?

Here's the youngest player of each team (30 team). I used min() to find the lowest age from each team and sort values by team.

	Tm	Player	Age
0	ATL	DeAndre' Bembry	22.0
1	BOS	Al Horford	20.0
2	BRK	Andrew Nicholson	21.0
3	CHI	Anthony Morrow	21.0
4	CHO	Aaron Harrison	21.0
5	CLE	Andrew Bogut	21.0
6	DAL	A.J. Hammons	21.0
7	DEN	Alonzo Gee	19.0
8	DET	Andre Drummond	20.0
9	GSW	Anderson Varejao	20.0
10	HOU	Bobby Brown	20.0
11	IND	Aaron Brooks	20.0
12	LAC	Alan Anderson	19.0
13	LAL	Brandon Ingram	19.0
14	MEM	Andrew Harrison	20.0
15	MIA	Dion Waiters	20.0
16	MIL	Gary Payton	19.0
17	MIN	Adreian Payne	20.0
18	NOP	Alexis Ajinca	20.0
19	NYK	Carmelo Anthony	21.0
20	OKC	Alex Abrines	20.0
21	ORL	Aaron Gordon	20.0
22	PHI	Alex Poythress	21.0
23	PHO	Alan Williams	19.0
24	POR	Al-Farouq Aminu	21.0
25	SAC	Anthony Tolliver	19.0
26	SAS	Bryn Forbes	20.0
27	TOR	Bruno Caboclo	21.0
28	UTA	Alec Burks	21.0
29	WAS	Bojan Bogdanovic	21.0

HANDS-ON

QUESTION 1

Here's the oldest player of each team (30 team). I used max() to find the highest age from each team and sort values by team (A-Z).

	Tm	Player	Age
0	ATL	Tim Hardaway	36.0
1	BOS	Tyler Zeller	31.0
2	BRK	Trevor Booker	36.0
3	CHI	Robin Lopez	35.0
4	CHO	Treveon Graham	31.0
5	CLE	Tristan Thompson	38.0
6	DAL	Yogi Ferrell	38.0
7	DEN	Wilson Chandler	36.0
8	DET	Tobias Harris	34.0
9	GSW	Zaza Pachulia	36.0
10	HOU	Troy Williams	34.0
11	IND	Thaddeus Young	32.0
12	LAC	Wesley Johnson	39.0
13	LAL	Tyler Ennis	37.0
14	MEM	Zach Randolph	40.0
15	MIA	Willie Reed	36.0
16	MIL	Tony Snell	39.0
17	MIN	Zach LaVine	34.0
18	NOP	Tim Frazier	33.0
19	NYK	Willy Hernangomez	32.0
20	OKC	Victor Oladipo	36.0
21	ORL	Terrence Ross	32.0
22	PHI	Timothe Luwawu-Cabarrot	32.0
23	PHO	Tyson Chandler	34.0
24	POR	Tim Quarterman	28.0
25	SAC	Willie Cauley-Stein	31.0
26	SAS	Tony Parker	39.0
27	TOR	Serge Ibaka	31.0
28	UTA	Trey Lyles	35.0
29	WAS	Trey Burke	32.0

Here's the code to show the tables :

```
df_young = df1[['Player','Tm','Age']].groupby(['Tm']).min().sort_values(by='Tm').reset_index()
df_old = df1[['Player','Tm','Age']].groupby(['Tm']).max().sort_values(by='Tm').reset_index()
```

QUESTION 2

Which player has the most minutes played (MP) in each position (Pos)?

Here's the most minutes played in each position. I used max() to find the highest age from each Pos and sort values by highest MP.

	Pos	Player	MP
0	SF	Wilson Chandler	3048.0
1	C	Zaza Pachulia	3030.0
2	PG	Yogi Ferrell	2947.0
3	PF	Zach Randolph	2803.0
4	SG	Zach LaVine	2796.0

Here's the code to show the table :

```
df_mp = df1[['Player','Pos','MP']].groupby(['Pos']).max().sort_values(by='MP', ascending=False).reset_index()
df_mp
```

HANDS-ON

QUESTION 3

Which team has the highest average total rebound percentage (TRB%), assist percentage (AST%), steal percentage (STL%), and block percentage (BLK%)?

Here's the highest average from TRB%, AST%, STL%, and BLK%. I used mean() to find the average from each team. Then, sort every variable values by the highest score.

	Tm	TRB%
0	WAS	12.735294

	Tm	AST%
6	DEN	15.723529

	Tm	STL%
22	MIN	2.413333

	Tm	BLK%
5	MIL	2.741176

Here's the example code to show the highest TRB%:

```
df_highestaverage = df1[['Tm','TRB%','AST%','STL%','BLK%']].groupby(['Tm']).mean().reset_index
```

```
#Highest average total rebound percentage (TRB%)
df_highestaverage[['Tm','TRB%']].sort_values(by='TRB%', ascending=False).head(1)
```

QUESTION 4

Who is the best player in your opinion based on his record stats? note: you can refer to variables point (PTS), assists, rebounds, or anything else. A combination of several variables would be nice.

Here's the best player based on average of all numeric column (new column named Overall), except age and year.

	Overall	Player
0	360.796867	Russell Westbrook
1	328.918111	James Harden
2	302.964289	Karl-Anthony Towns
3	298.494067	Anthony Davis
4	281.170644	LeBron James

Here's the code to show the table:

```
df1['Overall'] = df1.iloc[:,5:].mean(axis=1)
```

```
df1[['Overall','Player']].sort_values(by='Overall', ascending=False).reset_index(drop=True).head(5)
```

HANDS-ON

QUESTION 5

Which team has the best average stat record of their players? Note: you can refer to points, assists, rebounds, or anything else. A combination of several variables would be nice.

Here's the best team based on sum Overall column of each team.

	Tm	Overall
0	GSW	1575.927800
1	WAS	1514.843933
2	LAC	1507.355156
3	BOS	1507.239800
4	SAS	1506.784778

Here's the code to show the table:

```
df1[['Tm','Overall']].groupby(by='Tm').sum().sort_values(by='Overall',ascending=False).reset_index().head(5)
```