Stephen Curry Performance Analysis

Overview:

Stephen Curry, the star player for the Golden State Warriors, is known for his incredible shooting ability from long distances. This analysis will focus on the longest and median shot distance numbers for Curry, as well as the total number of field goals attempted from certain distances and the resulting field goals made and percentage. Additionally, 3 points field goals and 2 points field goals comparison within different time range and location will be provided.

Key Points:

CurryShots and NBA_teamlist datasets in csv format are available for this study.

Data wrangling including column transformation, date format change, check for missing/unmatched values and tables merge are carried out in Jupiter notebook using python (code included at end of the report).

Various graphics and charts are created in PowerBI including court image showing Curry's shot attempts by location, clustered column chart, line and clustered column chart, etc.

Key Observations:

Curry attempted a total of 12,155 field goals in the 8 seasons (2010-2018), with 43.71% of those attempts coming from beyond the three-point line.

Steph Curry's longest shot distance made in the 8 seasons was 39 feet, and his median shot distance was 21 feet.

Curry's 3 points field goal performance has maintained between 40% to 47% over the 8 seasons he played regardless of his team’s performance.

The field goal score percentage by weekday chart shows Steph Curry's 3 points FG performance is highest on Thursdays and lowest on Tuesdays.

Steph Curry's overall FG score percentage is higher playing at home than playing away, which is to be expected.

When compared to 2 points field goal score percentage, Curry’s 3 points field goal score percentage is lower but only by a slim margin of 8% on average. This shows that Steph Curry is an all-round player.

Conclusion:

This performance analysis provides valuable insights into Steph Curry's shooting abilities, including his longest and median shot distance, and field goals attempted by a distance. It’s powerful tool for teams, coaches, and analysts looking to gain a deeper understanding of player and team performance. These insights can also be used to inform game strategies, training, and overall analysis for both Curry and his team, the Golden State Warriors.

Stephen Curry
Performance Analysis

Score Percentage and Shot Distance

Season

2016

Shot_Distance Group (ft)	total FG attempted	total FG made	total FG percentage
0 to 4 feet	493	308	62.47%
05 to 12 feet	136	68	50.00%
13 to 22 feet	241	105	43.57%
23 and beyond	1046	467	44.65%
Total	**1916**	**948**	**49.48%**

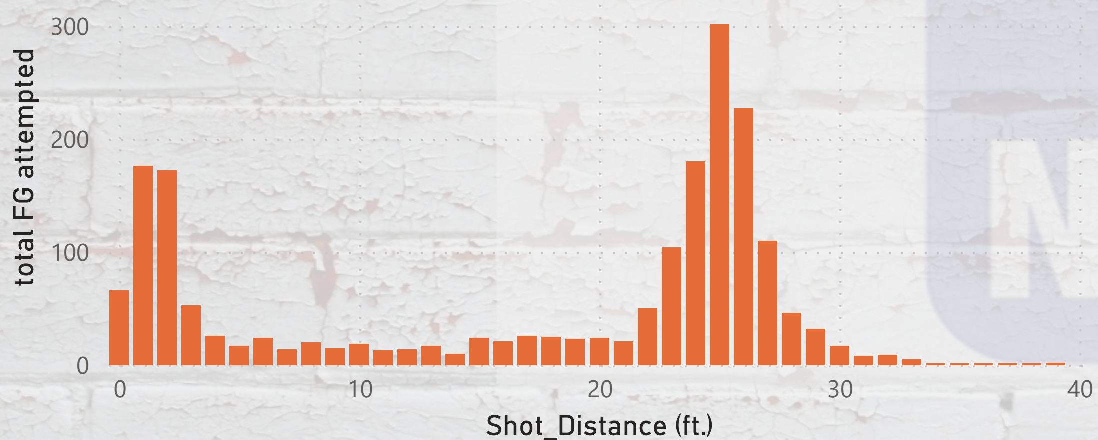
39

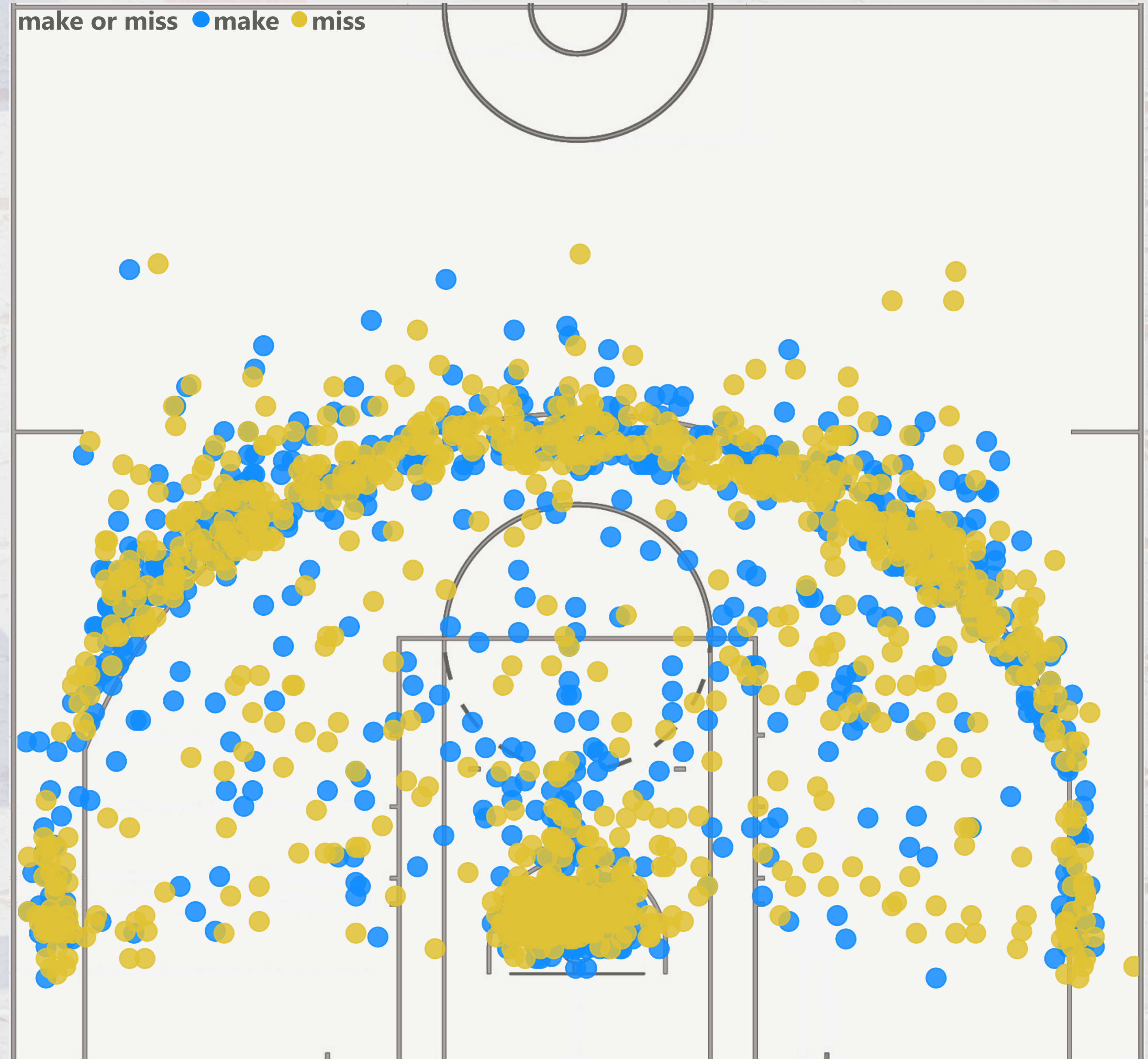
longest shot dist (ft)

23

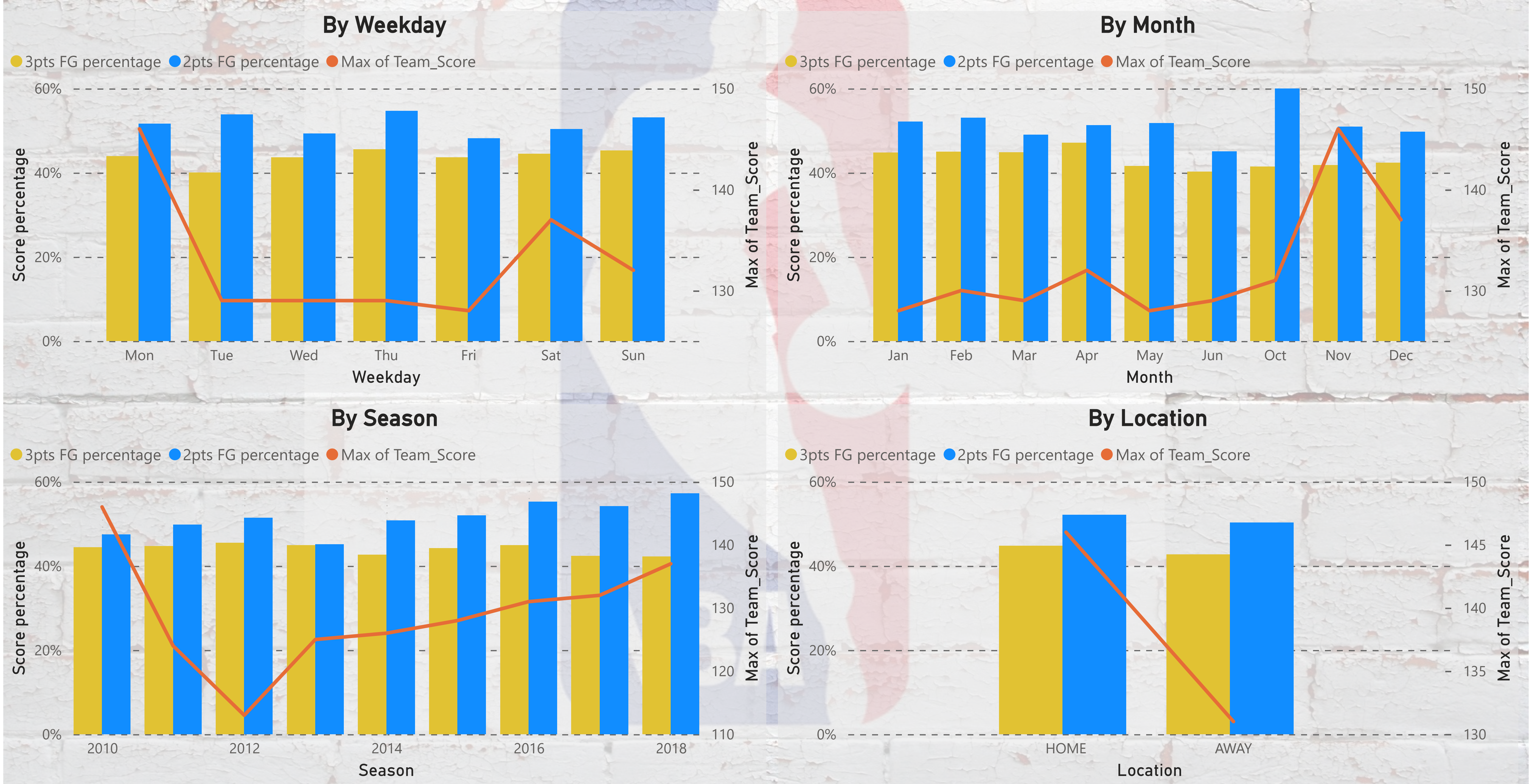
median shot dist (ft)

total FG attempted by Shot_Distance (ft.)





3pts FG Vs. 2pts FG Score Percentage



In [1]:
```python
import pandas as pd
```

In [2]:
```python
# Read the CSV files
data = pd.read_csv(r"C:\...\PowerBI\CurryShots.csv")
NBA_teamlist = pd.read_csv(r"C:\...\PowerBI\NBA_teamlist.csv")
```

In [3]:
```python
data
```

Out[3]:

	ID	Player	Season	Top.px. (Location)	Left.px. (location)	Date	Opponent	Location	Quarter	Game_Clock	Outcome (1 if made, 0 otherwise)	Shot_Value	Shot_Distance.ft.	Team_Score	Opponent_Score
0	curryst01	Stephen Curry	2010	299	339	102809	HOU	HOME	1	11:25	0	3	27	2	0
1	curryst01	Stephen Curry	2010	195	118	102809	HOU	HOME	1	9:31	1	2	19	4	2
2	curryst01	Stephen Curry	2010	179	180	102809	HOU	HOME	1	6:02	0	2	14	8	12
3	curryst01	Stephen Curry	2010	132	68	102809	HOU	HOME	2	9:49	0	2	19	28	32
4	curryst01	Stephen Curry	2010	198	172	102809	HOU	HOME	2	2:19	0	2	16	48	48
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
12150	curryst01	Stephen Curry	2018	50	0	60818	CLE	AWAY	4	6:19	1	3	24	102	74
12151	curryst01	Stephen Curry	2018	139	125	60818	CLE	AWAY	4	5:48	0	2	14	102	74
12152	curryst01	Stephen Curry	2018	380	124	60818	CLE	AWAY	4	5:13	0	3	34	102	76
12153	curryst01	Stephen Curry	2018	61	254	60818	CLE	AWAY	4	4:27	0	2	1	102	77
12154	curryst01	Stephen Curry	2018	109	253	60818	CLE	AWAY	4	3:49	0	2	6	102	77

12155 rows × 15 columns

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12155 entries, 0 to 12154
Data columns (total 15 columns):
 #   Column                          Non-Null Count  Dtype 
---  ------                          --------------  ----- 
 0   ID                              12155 non-null  object
 1   Player                          12155 non-null  object
 2   Season                          12155 non-null  int64 
 3   Top.px. (Location)              12155 non-null  int64 
 4   Left.px. (location)             12155 non-null  int64 
 5   Date                            12155 non-null  int64 
 6   Opponent                        12155 non-null  object
 7   Location                        12155 non-null  object
 8   Quarter                         12155 non-null  object
 9   Game_Clock                      12155 non-null  object
 10  Outcome (1 if made, 0 otherwise)  12155 non-null  int64 
 11  Shot_Value                      12155 non-null  int64 
 12  Shot_Distance.ft.               12155 non-null  int64 
 13  Team_Score                      12155 non-null  int64 
 14  Opponent_Score                  12155 non-null  int64 
dtypes: int64(9), object(6)
memory usage: 1.4+ MB
```

In [5]:
```
NBA_teamlist
```

Out[5]:

	Abbreviation	Franchise
0	ATL	Atlanta Hawks
1	BRK	Brooklyn Nets
2	BOS	Boston Celtics
3	CHA	Charlotte Hornets
4	CHI	Chicago Bulls
5	CLE	Cleveland Cavaliers
6	DAL	Dallas Mavericks
7	DEN	Denver Nuggets
8	DET	Detroit Pistons
9	GSW	Golden State Warriors
10	HOU	Houston Rockets
11	IND	Indiana Pacers
12	LAC	Los Angeles Clippers
13	LAL	Los Angeles Lakers
14	MEM	Memphis Grizzlies
15	MIA	Miami Heat
16	MIL	Milwaukee Bucks
17	MIN	Minnesota Timberwolves
18	NOP	New Orleans Pelicans
19	NYK	New York Knicks
20	OKC	Oklahoma City Thunder
21	ORL	Orlando Magic
22	PHI	Philadelphia 76ers
23	PHO	Phoenix Suns
24	POR	Portland Trail Blazers
25	SAC	Sacramento Kings
26	SAS	San Antonio Spurs
27	TOR	Toronto Raptors
28	UTA	Utah Jazz
29	WAS	Washington Wizards
30	NOH	New Orleans Hornets
31	NJN	New Jersey Nets

In [6]:
```
NBA_teamlist.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Abbreviation  32 non-null     object
 1   Franchise     32 non-null     object
dtypes: object(2)
memory usage: 640.0+ bytes
```

In [7]:
```
# Perform data transformations
data = data.rename(columns={"Outcome (1 if made, 0 otherwise)": "Outcome"})
data["make or miss"] = data["Outcome"].map({1: "make", 0: "miss"})
data = data.drop(columns=["ID", "Player", "Game_Clock", "Outcome"])
# Change date presentation from "xxxxxx" or "xxxxx" to "xx/xx/xx"
data["Date"] = data["Date"].astype(str)
data['Date'] = data['Date'].apply(lambda x: '0' + x if len(x) == 5 else x)
data["Date"] = data["Date"].astype(str).str[:2] + "/" + data["Date"].astype(str).str[2:4] + "/" + data["Date"].astype(str).str[4:]
```

In [8]:
```
data
```

Out[8]:

	Season	Top.px. (Location)	Left.px. (location)	Date	Opponent	Location	Quarter	Shot_Value	Shot_Distance.ft.	Team_Score	Opponent_Score	make or miss
0	2010	299	339	10/28/09	HOU	HOME	1	3	27	2	0	miss
1	2010	195	118	10/28/09	HOU	HOME	1	2	19	4	2	make
2	2010	179	180	10/28/09	HOU	HOME	1	2	14	8	12	miss
3	2010	132	68	10/28/09	HOU	HOME	2	2	19	28	32	miss
4	2010	198	172	10/28/09	HOU	HOME	2	2	16	48	48	miss
...	...	...	...	...	...	...	...	...	...	...	...	...
12150	2018	50	0	06/08/18	CLE	AWAY	4	3	24	102	74	make
12151	2018	139	125	06/08/18	CLE	AWAY	4	2	14	102	74	miss
12152	2018	380	124	06/08/18	CLE	AWAY	4	3	34	102	76	miss
12153	2018	61	254	06/08/18	CLE	AWAY	4	2	1	102	77	miss
12154	2018	109	253	06/08/18	CLE	AWAY	4	2	6	102	77	miss

12155 rows × 12 columns

In [9]:
```
# Merge with NBA_teamlist table
data = data.rename(columns={"Opponent": "Opponent abrev."})
datamerge = pd.merge(data, NBA_teamlist, left_on="Opponent abrev.", right_on="Abbreviation", how="left")
datamerge = datamerge.rename(columns={"Franchise": "Opponent"})
```

In [10]:
```
datamerge
```

Out[10]:

	Season	Top.px. (Location)	Left.px. (location)	Date	Opponent abrev.	Location	Quarter	Shot_Value	Shot_Distance.ft.	Team_Score	Opponent_Score	make or miss	Abbreviation	Opponent
0	2010	299	339	10/28/09	HOU	HOME	1	3	27	2	0	miss	HOU	Houston Rockets
1	2010	195	118	10/28/09	HOU	HOME	1	2	19	4	2	make	HOU	Houston Rockets
2	2010	179	180	10/28/09	HOU	HOME	1	2	14	8	12	miss	HOU	Houston Rockets
3	2010	132	68	10/28/09	HOU	HOME	2	2	19	28	32	miss	HOU	Houston Rockets
4	2010	198	172	10/28/09	HOU	HOME	2	2	16	48	48	miss	HOU	Houston Rockets
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
12150	2018	50	0	06/08/18	CLE	AWAY	4	3	24	102	74	make	CLE	Cleveland Cavaliers
12151	2018	139	125	06/08/18	CLE	AWAY	4	2	14	102	74	miss	CLE	Cleveland Cavaliers
12152	2018	380	124	06/08/18	CLE	AWAY	4	3	34	102	76	miss	CLE	Cleveland Cavaliers
12153	2018	61	254	06/08/18	CLE	AWAY	4	2	1	102	77	miss	CLE	Cleveland Cavaliers
12154	2018	109	253	06/08/18	CLE	AWAY	4	2	6	102	77	miss	CLE	Cleveland Cavaliers

12155 rows × 14 columns

In [11]:
```
datamerge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12155 entries, 0 to 12154
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Season               12155 non-null  int64
 1   Top.px. (Location)   12155 non-null  int64
 2   Left.px. (location)  12155 non-null  int64
 3   Date                 12155 non-null  object
 4   Opponent abrev.      12155 non-null  object
 5   Location             12155 non-null  object
 6   Quarter              12155 non-null  object
 7   Shot_Value           12155 non-null  int64
 8   Shot_Distance.ft.    12155 non-null  int64
 9   Team_Score           12155 non-null  int64
 10  Opponent_Score       12155 non-null  int64
 11  make or miss         12155 non-null  object
 12  Abbreviation         12039 non-null  object
 13  Opponent             12039 non-null  object
dtypes: int64(7), object(7)
memory usage: 1.4+ MB
```

In [12]:
```
#Check the rows with null values in the merged table
nan_rows = datamerge[datamerge["Opponent"].isna()]
nan_rows
```

Out[12]:

	Season	Top.px. (Location)	Left.px. (location)	Date	Opponent abrev.	Location	Quarter	Shot_Value	Shot_Distance.ft.	Team_Score	Opponent_Score	make or miss	Abbreviation	Opponent
5728	2015	296	329	11/15/14	CHO	HOME	1	3	26	0	0	miss	NaN	NaN
5729	2015	256	74	11/15/14	CHO	HOME	1	3	26	9	7	miss	NaN	NaN
5730	2015	136	82	11/15/14	CHO	HOME	1	2	18	14	12	make	NaN	NaN
5731	2015	133	74	11/15/14	CHO	HOME	1	2	19	20	18	make	NaN	NaN
5732	2015	192	104	11/15/14	CHO	HOME	1	2	20	22	20	make	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10104	2017	61	233	02/01/17	CHO	HOME	3	2	1	94	64	make	NaN	NaN
10105	2017	194	444	02/01/17	CHO	HOME	3	3	25	98	69	miss	NaN	NaN
10106	2017	224	130	02/01/17	CHO	HOME	3	2	21	102	79	make	NaN	NaN
10107	2017	191	26	02/01/17	CHO	HOME	3	3	26	105	79	make	NaN	NaN
10108	2017	287	403	02/01/17	CHO	HOME	3	3	29	108	83	make	NaN	NaN

116 rows × 14 columns

In [13]:
```
# There are 116 rows with null values all related to Opponent abrev. CHO.
# The Opponent abbreviation CHO in the data table does not have a match in the NBA_teamlist table. A quick google search shown
# CHO is used for Charlotte Hornets. Check the NBA_teamlist and fund that the abbreviation CHA is used for Charlotte Hornets.
# Replace CHO with CHA for data merge
data = data.rename(columns={"Opponent": "Opponent abrev."})
data["Opponent abrev."] = data["Opponent abrev."].replace("CHO", "CHA")

# Merge data table with NBA_teamlist table
data = pd.merge(data, NBA_teamlist, left_on="Opponent abrev.", right_on="Abbreviation", how="left")
data = data.rename(columns={"Franchise": "Opponent"})

# Reorder columns and add an index column
data = data.reset_index().rename(columns={"index": "Shot-id"})
data = data.reindex(columns=["Shot-id", "Season", "Top.px. (Location)", "Left.px. (location)", "Date", "Opponent", "Location", "Quarter", "make or miss", "Shot_Value

# Rename column
data = data.rename(columns={"Shot_Distance.ft.": "Shot_Distance (ft.)"})
```

In [14]:
```
data
```

Out[14]:

	Shot-id	Season	Top.px. (Location)	Left.px. (location)	Date	Opponent	Location	Quarter	make or miss	Shot_Value	Shot_Distance (ft.)	Team_Score	Opponent_Score
0	0	2010	299	339	10/28/09	Houston Rockets	HOME	1	miss	3	27	2	0
1	1	2010	195	118	10/28/09	Houston Rockets	HOME	1	make	2	19	4	2
2	2	2010	179	180	10/28/09	Houston Rockets	HOME	1	miss	2	14	8	12
3	3	2010	132	68	10/28/09	Houston Rockets	HOME	2	miss	2	19	28	32
4	4	2010	198	172	10/28/09	Houston Rockets	HOME	2	miss	2	16	48	48
...	...	...	...	...	...	...	...	...	...	...	...	...	...
12150	12150	2018	50	0	06/08/18	Cleveland Cavaliers	AWAY	4	make	3	24	102	74
12151	12151	2018	139	125	06/08/18	Cleveland Cavaliers	AWAY	4	miss	2	14	102	74
12152	12152	2018	380	124	06/08/18	Cleveland Cavaliers	AWAY	4	miss	3	34	102	76
12153	12153	2018	61	254	06/08/18	Cleveland Cavaliers	AWAY	4	miss	2	1	102	77
12154	12154	2018	109	253	06/08/18	Cleveland Cavaliers	AWAY	4	miss	2	6	102	77

12155 rows × 13 columns

In [15]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12155 entries, 0 to 12154
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   Shot-id              12155 non-null  int64 
 1   Season               12155 non-null  int64 
 2   Top.px. (Location)   12155 non-null  int64 
 3   Left.px. (location)  12155 non-null  int64 
 4   Date                 12155 non-null  object
 5   Opponent             12155 non-null  object
 6   Location             12155 non-null  object
 7   Quarter              12155 non-null  object
 8   make or miss         12155 non-null  object
 9   Shot_Value           12155 non-null  int64 
 10  Shot_Distance (ft.)  12155 non-null  int64 
 11  Team_Score           12155 non-null  int64 
 12  Opponent_Score       12155 non-null  int64 
dtypes: int64(8), object(5)
memory usage: 1.2+ MB
```

In [16]:
```
# Save clean data to disk
data.to_csv('CurryShotsClean.csv', index=False)
```