Analysis of Stephen Curry’s Shot Data (2014 - 2015 Season)

Charalampos Deligiannakis, Feb 2025

Analysis of Stephen Curry's Shot Data (2014-2015 Season)

In this notebook, we will analyze Stephen Curry's shot data from the 2014-2015 NBA season. We will explore various aspects of his performance including shot attempts, accuracy by distance, clutch performance, and shooting trends across different game periods. All of these visualizations and analyses are aimed at understanding Curry's shooting patterns during the regular season. The language I used is Python.

The dataset contains shot logs that capture various details about each of Curry's shots, including shot distance, result (made or missed), the game period, and more. We will load the data and perform several analyses based on specific goals.

```
# DATA:
#   Steph Curry
#   Field Goal Shots (2s and 3s)
#   Season 2014 - 2015
#   Only Regular Games

# GOALS:
#   Attempts Based On Distance
#   Accuracy Based On Distance
#   Accuracy Based On Period
#   Average Points Based On Period From FGs
#   Clutch Accuracy (Accuracy Based On Last 5 Minutes Where |Final Margin| <= 10)
#   3PT Accuracy Through Games
#   2PT Accuracy Through Games
#   3PT Accuracy Based On Defender Distance
#   Accuracy Based On Number Of Dribbles Before Shooting - Accuracy Based On Tou
#   2PT Accuracy For Home and Away Games - 3PT Accuracy For Home and Away Gar

# NEW COLUMNS: (New Columns Will Be Created Containing Any Information That Wil
#   Distance Categories [0 - 8, 8 - 22, 22 - 30, 30+]
```

```
# NOTE: Some variables are repeated just for the independence of each code segment
```

1. Import Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
```

2. Load the Data

Let's load the dataset which contains shot logs for Stephen Curry. The data includes various columns, such as *SHOT_DIST* (shot distance), *SHOT_RESULT* (whether the shot was made or missed), *PERIOD* (game period), and *PTS_TYPE* (type of shot, either 2 or 3 points).

```
# Load data from csv file
df = pd.read_csv('/kaggle/input/stephen-curry-regular-fg-shots-2014-2015/curry_shot
```

3. Preprocess the Data

We will categorize the shots into distance categories. These categories will be helpful for analyzing shot attempts and accuracy based on the distance from the basket.

```
# Create New Columns
distanceBins = [0, 8, 22, 30, float('inf')]
distanceLabels = ['Paint Area', 'Mid-Range 2PT', '3PT', 'Deep 3PT']
df['distance_category'] = pd.cut(df['SHOT_DIST'], bins=distanceBins, labels=distanceL
```

4. Plot Styling

We will apply some styling to the plots, using the Golden State Warriors' team colors for the background, grid, and text.

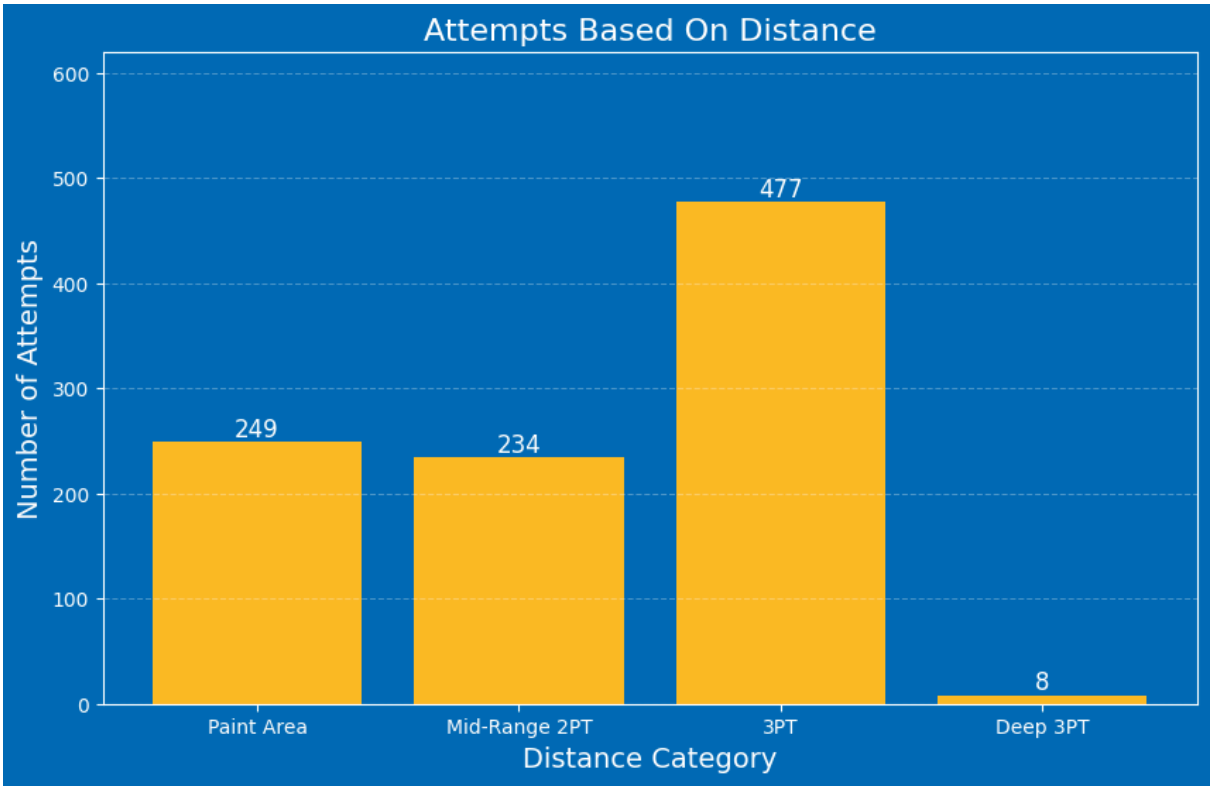
```
# Plot Styling
gsw_gold = '#FDB927'
gsw_blue = '#006BB6'
white = '#FFFFFF'
plt.rcParams.update({
    'axes.facecolor': gsw_blue,
```

```
    'figure.facecolor': gsw_blue,
    'axes.edgecolor': white,
    'axes.labelcolor': white,
    'xtick.color': white,
    'ytick.color': white,
    'text.color': white,
    'axes.titlecolor': white,
    'axes.prop_cycle': plt.cycler('color', [gsw_gold])
})
```

5. Attempts Based on Distance

We will now visualize how many shots Curry attempted from different distances.

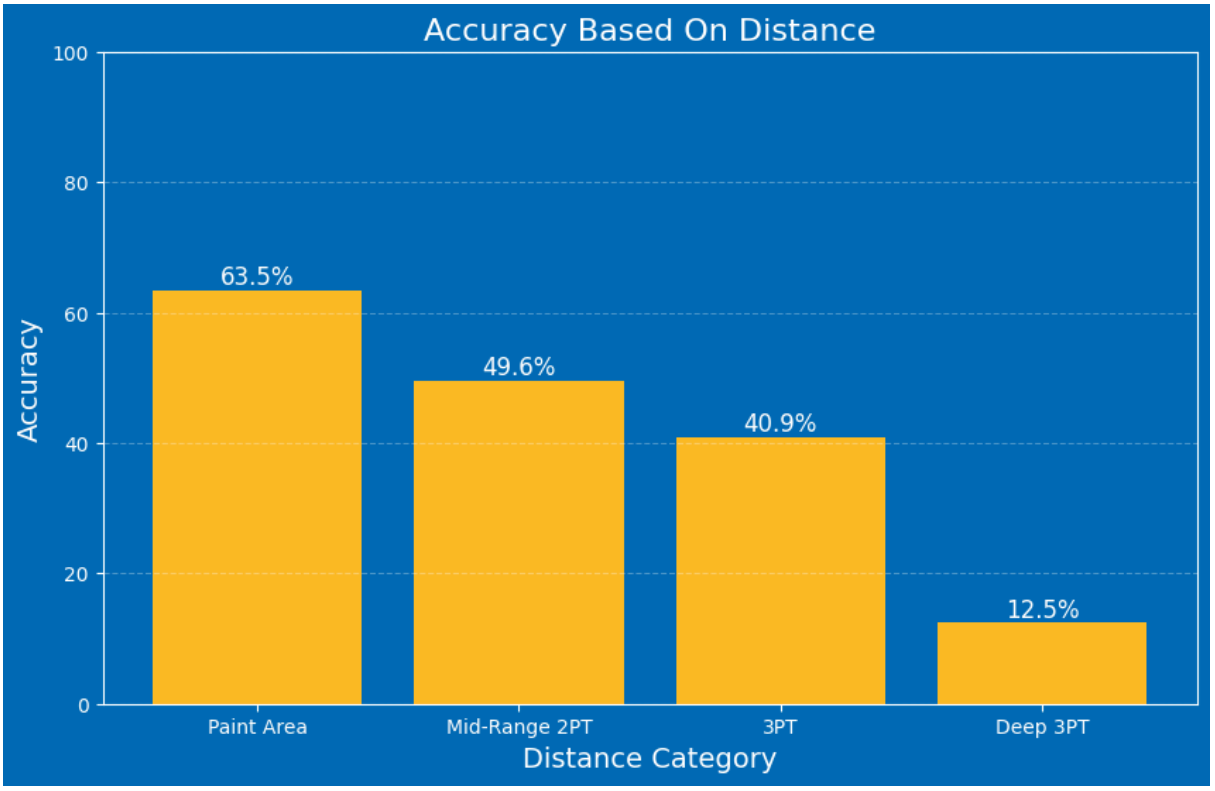
```
totalAttemptsByDistance = df.groupby('distance_category', observed=False).size().valu
plt.figure(figsize=(10, 6))
bars = plt.bar(distanceLabels, totalAttemptsByDistance)
plt.title("Attempts Based On Distance", fontsize=16)
plt.xlabel("Distance Category", fontsize=14)
plt.ylabel("Number of Attempts", fontsize=14)
plt.ylim(0, 1.3*max(totalAttemptsByDistance))
plt.grid(axis='y', linestyle='--', alpha=0.3, color=white)
for bar in bars: # Annotating bars
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
             f'{bar.get_height()}', ha='center', va='bottom', fontsize=12)
```



6. Accuracy Based on Distance

Next, we will plot Curry's shooting accuracy based on the distance of his shots.

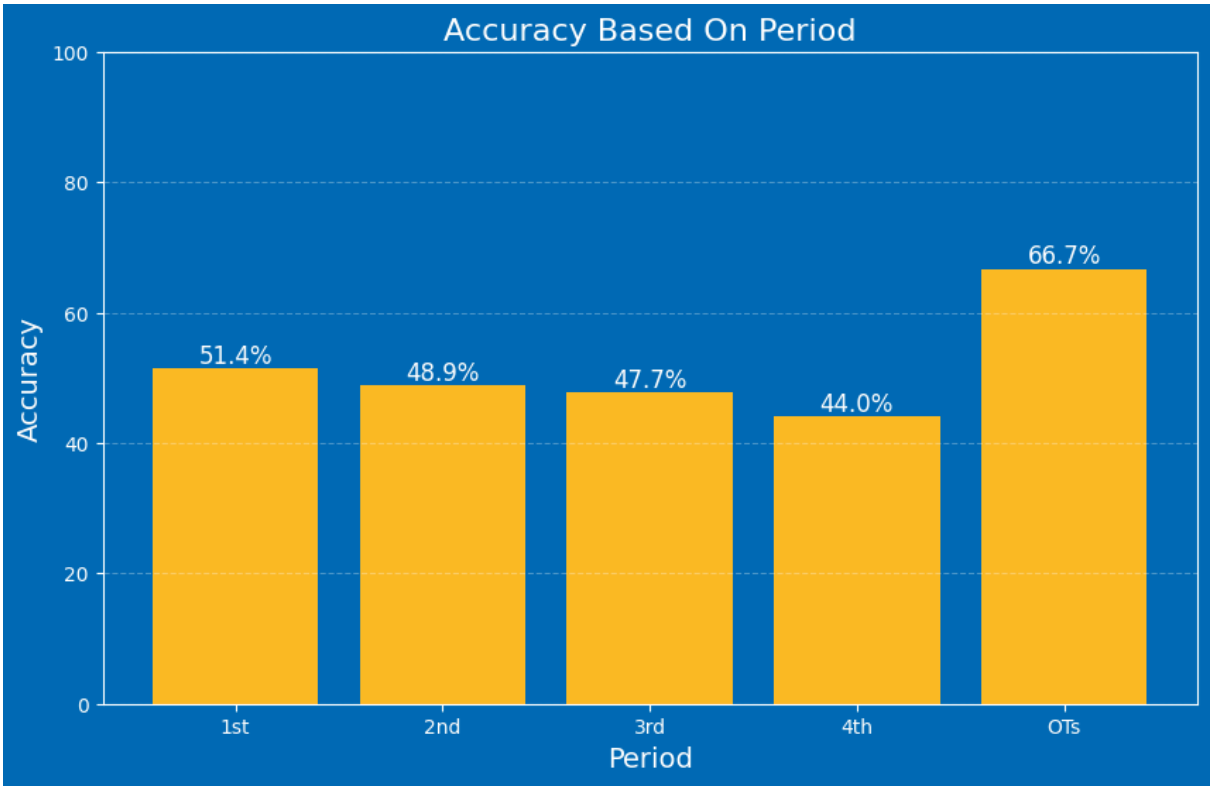
```
# Accuracy Based On Distance
totalAttemptsByDistanceDF = df.groupby('distance_category', observed=False).size().v
madeAttemptsByDistance = df[df['SHOT_RESULT'] == 'made'].groupby('distance_cate
accuracyByDistance = (madeAttemptsByDistance / totalAttemptsByDistance * 100)
plt.figure(figsize=(10, 6))
bars = plt.bar(distanceLabels, accuracyByDistance)
plt.title("Accuracy Based On Distance", fontsize=16)
plt.xlabel("Distance Category", fontsize=14)
plt.ylabel("Accuracy", fontsize=14)
plt.ylim(0, 100)
plt.grid(axis='y', linestyle='--', alpha=0.3, color=white)
for bar in bars: # Annotating bars
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
             f'{bar.get_height():.1f}%', ha='center', va='bottom', fontsize=12)
```



7. Accuracy Based on Period

We will now investigate Curry's shooting accuracy in different periods of the game (1st, 2nd, 3rd, 4th, and Overtime).

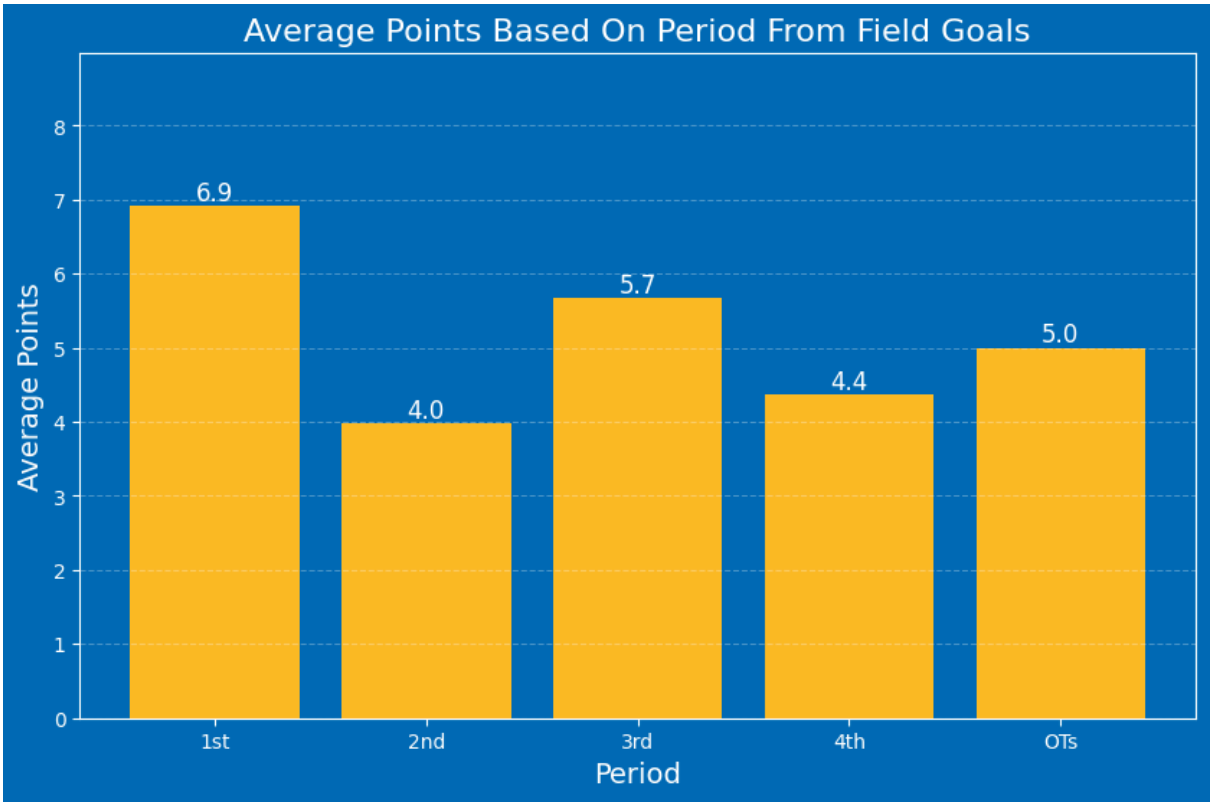
```
# Accuracy Based On Period
totalAttemptsByPeriod = df.groupby('PERIOD', observed=False).size().values
madeAttemptsByPeriod = df[df['SHOT_RESULT'] == 'made'].groupby('PERIOD', observ
accuracyByPeriod = (madeAttemptsByPeriod / totalAttemptsByPeriod) * 100
periodLabels = ['1st', '2nd', '3rd', '4th', 'OTs']
plt.figure(figsize=(10, 6))
bars = plt.bar(periodLabels, accuracyByPeriod)
plt.title("Accuracy Based On Period", fontsize=16)
plt.xlabel("Period", fontsize=14)
plt.ylabel("Accuracy", fontsize=14)
plt.ylim(0, 100)
plt.grid(axis='y', linestyle='--', alpha=0.3, color=white)
for bar in bars: # Annotating bars
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
             f'{bar.get_height():.1f}%', ha='center', va='bottom', fontsize=12)
```



8. Average Points Based on Period from Field Goals

Next, we will calculate the average points Curry scored in different periods of the game based on field goals.

```
# Average Points Based On Period From FGs
totalPeriodsPlayed = df.groupby('PERIOD', observed=False)['GAME_ID'].nunique()
totalPointsPerPeriod = df.groupby('PERIOD', observed=False)['PTS'].sum().values
averagePointsByPeriod = totalPointsPerPeriod / totalPeriodsPlayed
periodLabels = ['1st', '2nd', '3rd', '4th', 'OTs']
plt.figure(figsize=(10, 6))
bars = plt.bar(periodLabels, averagePointsByPeriod)
plt.title("Average Points Based On Period From Field Goals", fontsize=16)
plt.xlabel("Period", fontsize=14)
plt.ylabel("Average Points", fontsize=14)
plt.ylim(0, 1.3*max(averagePointsByPeriod))
plt.grid(axis='y', linestyle='--', alpha=0.3, color=white)
for bar in bars: # Annotating bars
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
             f'{bar.get_height():.1f}', ha='center', va='bottom', fontsize=12)
```

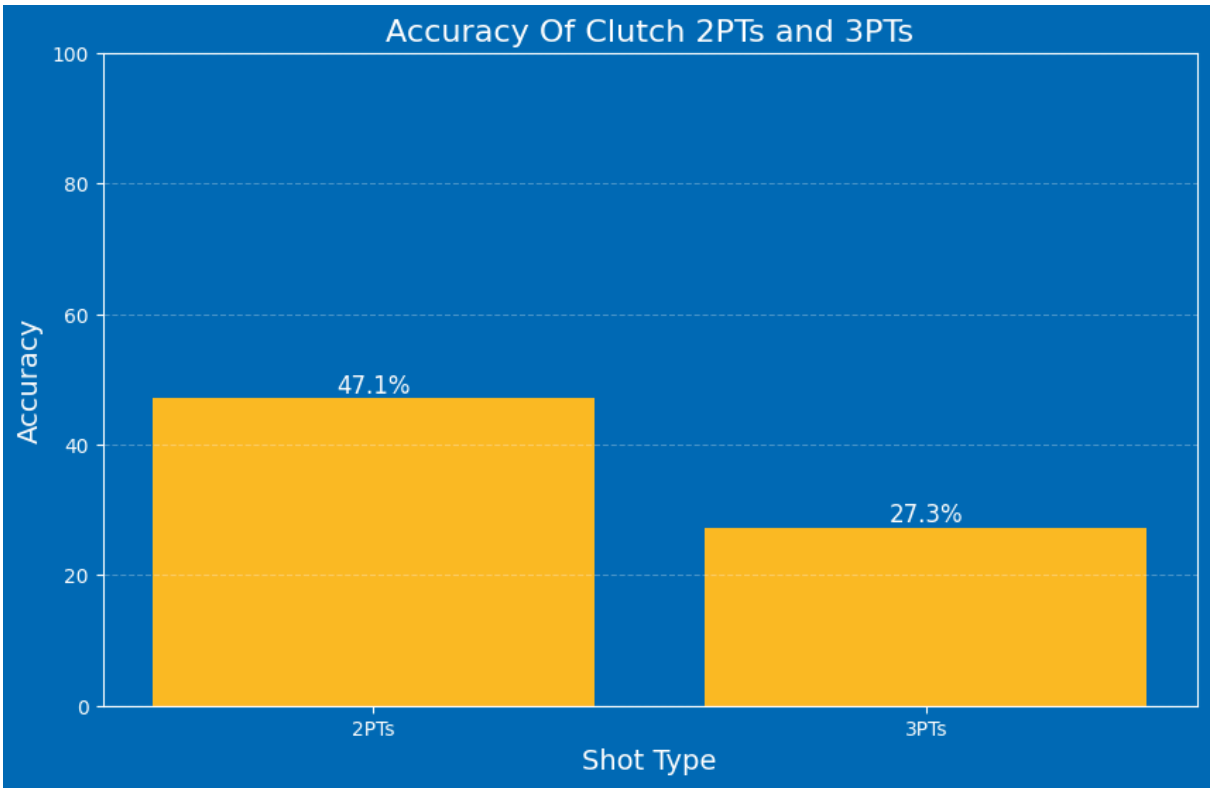


9. Clutch Accuracy (Accuracy Based on Last 5 Minutes)

Now, we will look into Curry's accuracy during clutch moments, i.e., the last 5 minutes of the game where the final margin is 10 points or less.

```
# Accuracy Of Clutch 2PTs and 3PTs (Accuracy Based On Last 5 Minutes Where Final
clutchAttempts = df[
    (df['PERIOD'] == 4) &
    (pd.to_datetime(df['GAME_CLOCK'], format='%M:%S') <= pd.to_datetime("5:00", fo
    (df['FINAL_MARGIN'].abs() <= 10)
]
totalAttempts = clutchAttempts.groupby('PTS_TYPE').size()
madeAttempts = clutchAttempts[clutchAttempts['SHOT_RESULT'] == 'made'].groupby(
clutchAccuracy = (madeAttempts / totalAttempts * 100).values
shotTypeLabels = ['2PTs', '3PTs']
plt.figure(figsize=(10, 6))
bars = plt.bar(shotTypeLabels, clutchAccuracy)
plt.title("Accuracy Of Clutch 2PTs and 3PTs", fontsize=16)
plt.xlabel("Shot Type", fontsize=14)
plt.ylabel("Accuracy", fontsize=14)
plt.ylim(0, 100)
plt.grid(axis='y', linestyle='--', alpha=0.3, color=white)
```

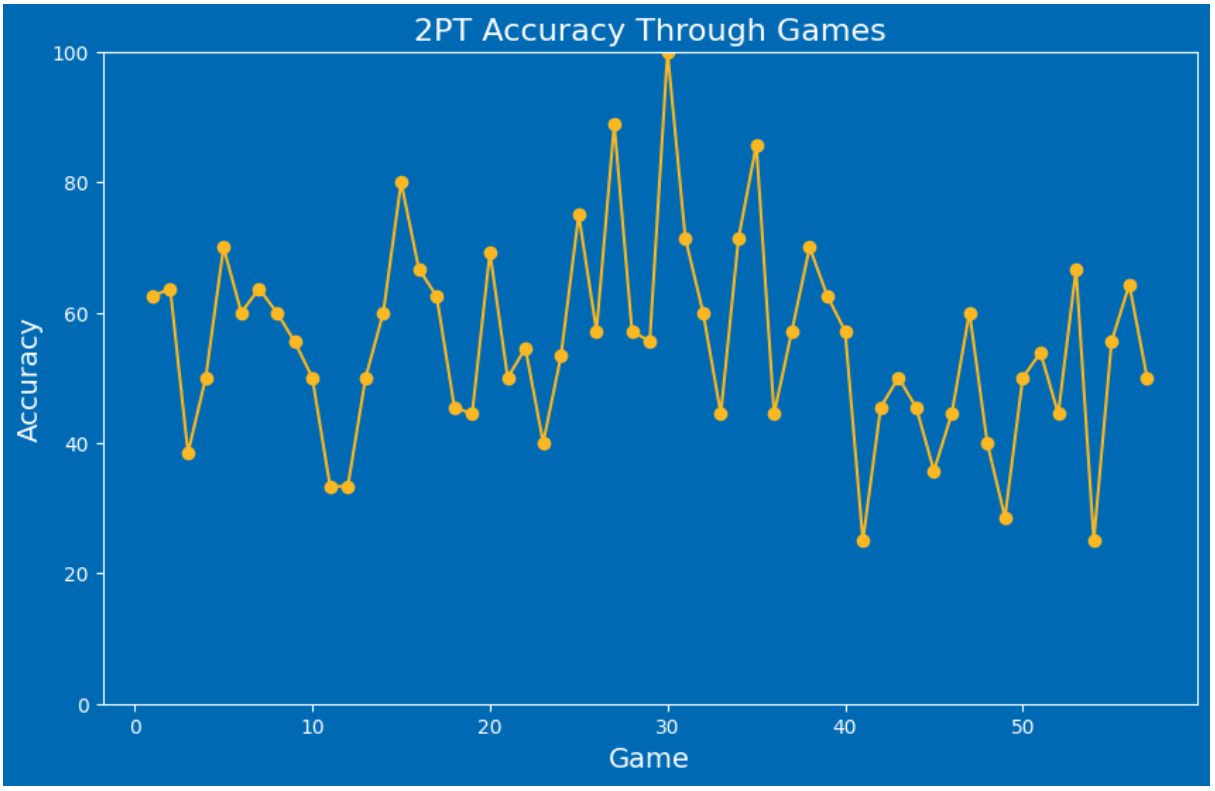
```
for bar in bars: # Annotating bars
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
             f'{bar.get_height():.1f}%', ha='center', va='bottom', fontsize=12)
```



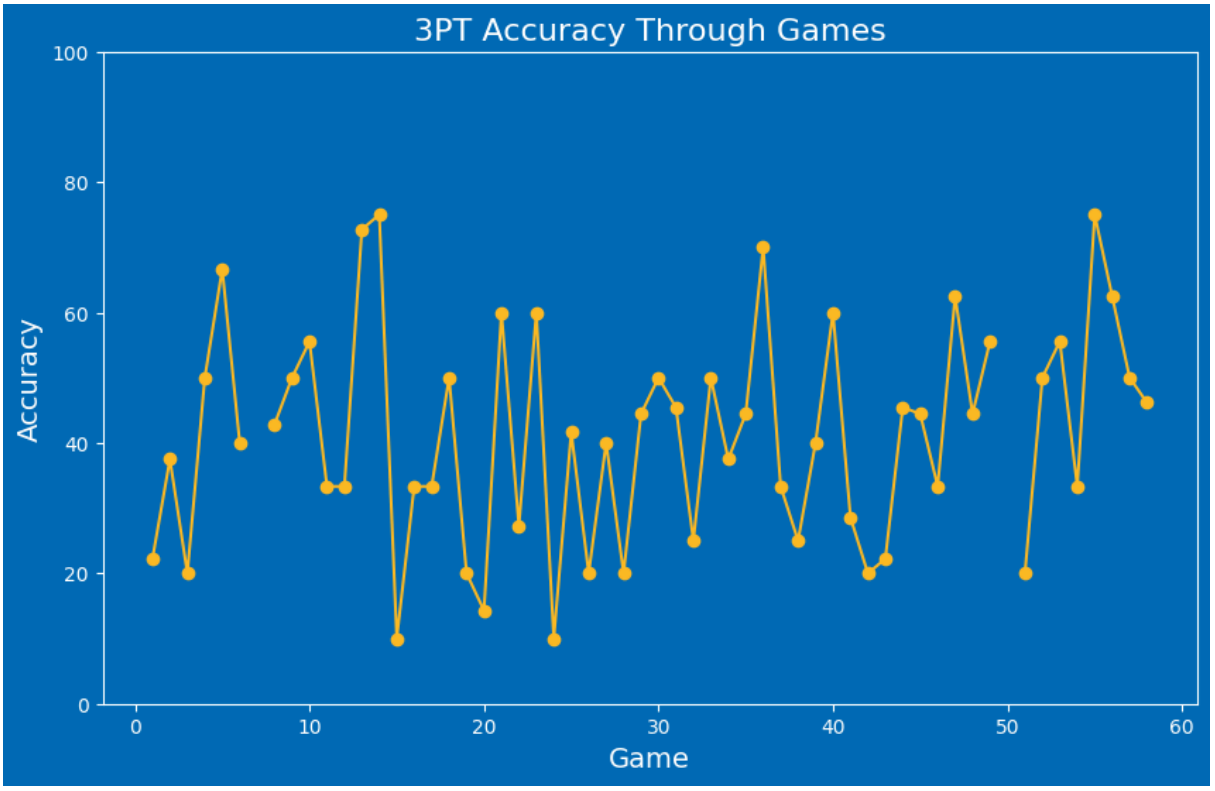
10. 2PT and 3PT Accuracy Through Games

Next, we will visualize Curry's 2-point and 3-point accuracy over the course of the season.

```
# 2PT Accuracy Through Games
twoPointsAttempted = df[df['PTS_TYPE'] == 2]
twoPointsMade = twoPointsAttempted[twoPointsAttempted['SHOT_RESULT'] == 'made
twoPointsAttemptedPerGame = twoPointsAttempted.groupby("GAME_ID", observed=Fa
twoPointsMadePerGame = twoPointsMade.groupby("GAME_ID", observed=False).size(
twoPointsAccuracyPerGame = twoPointsMadePerGame / twoPointsAttemptedPerGame
games = list(range(1, len(twoPointsAttemptedPerGame)+1))
plt.figure(figsize=(10, 6))
plt.plot(games, twoPointsAccuracyPerGame, linestyle='-', marker='o')
plt.title('2PT Accuracy Through Games', fontsize=16)
plt.xlabel('Game', fontsize=14)
plt.ylabel('Accuracy', fontsize=14)
plt.ylim(0, 100)
```



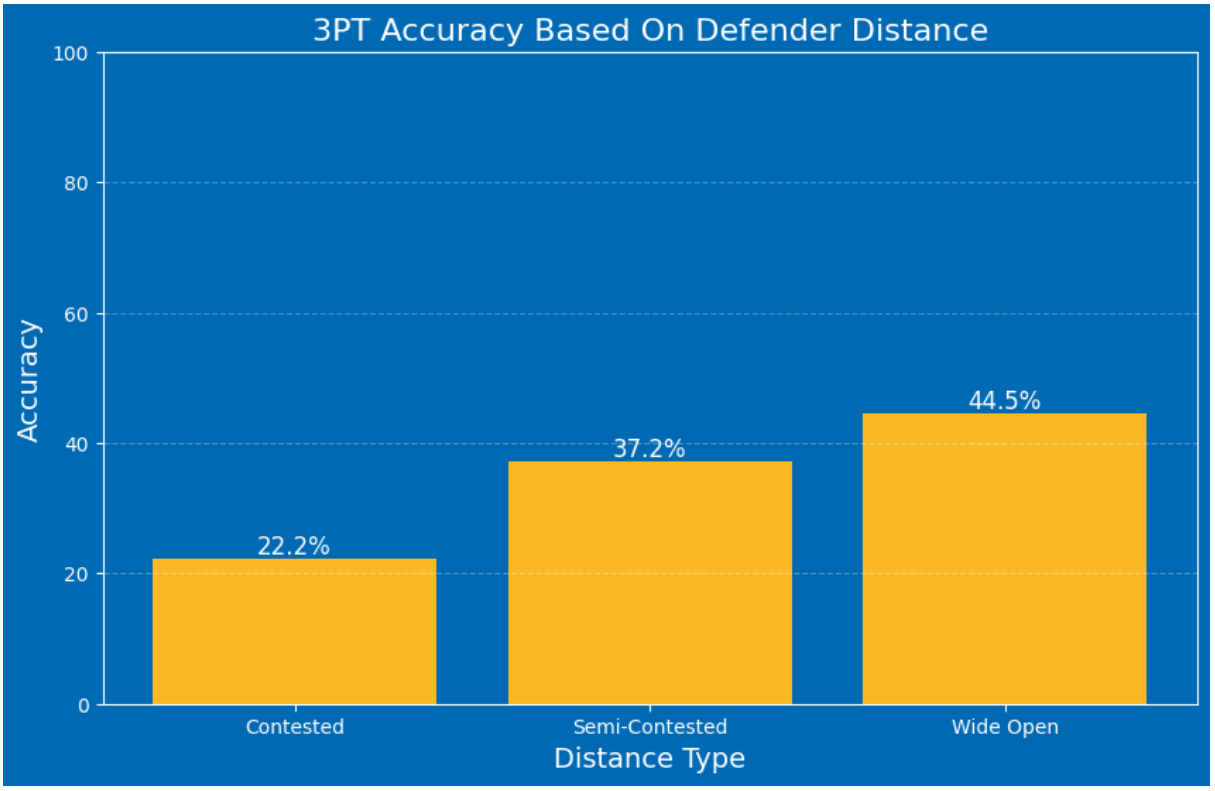

```
# 3PT Accuracy Through Games
threePointsAttempted = df[df['PTS_TYPE'] == 3]
threePointsMade = threePointsAttempted[threePointsAttempted['SHOT_RESULT'] == 'r
threePointsAttemptedPerGame = threePointsAttempted.groupby("GAME_ID", observed
threePointsMadePerGame = threePointsMade.groupby("GAME_ID", observed=False).si
threePointsAccuracyPerGame = threePointsMadePerGame / threePointsAttemptedPerG
games = list(range(1, len(threePointsAttemptedPerGame)+1))
plt.figure(figsize=(10, 6))
plt.plot(games, threePointsAccuracyPerGame, linestyle='-', marker='o')
plt.title('3PT Accuracy Through Games', fontsize=16)
plt.xlabel('Game', fontsize=14)
plt.ylabel('Accuracy', fontsize=14)
plt.ylim(0, 100)
```



11. 3PT Accuracy Based on Defender Distance

Now, let's examine Curry's 3-point accuracy based on the distance of the defender from him.

```
# 3PT Accuracy Based On Defender Distance
bins = [0, 2, 4, float('inf')]
labels = ['Contested', 'Semi-Contested', 'Wide Open']
threePoints = df[df['PTS_TYPE'] == 3]
threePointsAttemptsByDistance = threePoints.groupby(pd.cut(threePoints['CLOSE_DEF
threePointsMadeByDistance = threePoints[threePoints['SHOT_RESULT'] == 'made'].gro
threePointsAccuracyByDistance = threePointsMadeByDistance / threePointsAttemptsB
plt.figure(figsize=(10, 6))
bars = plt.bar(labels, threePointsAccuracyByDistance)
plt.title("3PT Accuracy Based On Defender Distance", fontsize=16)
plt.xlabel("Distance Type", fontsize=14)
plt.ylabel("Accuracy", fontsize=14)
plt.ylim(0, 100)
plt.grid(axis='y', linestyle='--', alpha=0.3, color=white)
for bar in bars: # Annotating bars
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
             f'{bar.get_height():.1f}%', ha='center', va='bottom', fontsize=12)
```

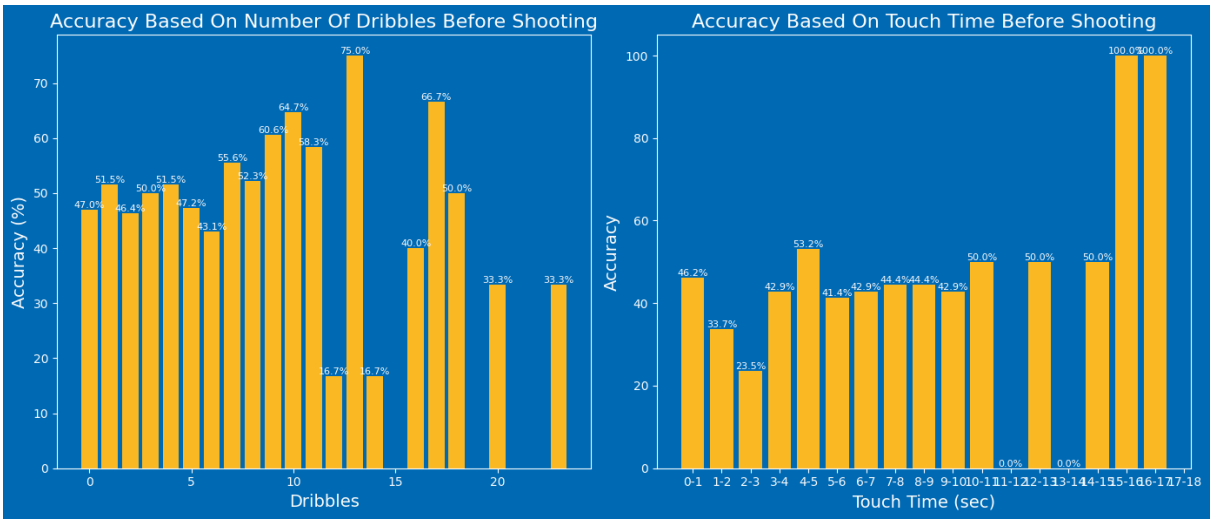


12. Accuracy Based on Number of Dribbles and Touch Time Before Shooting

We will analyze how Curry's accuracy changes based on the number of dribbles and the touch time before shooting.

```
# Accuracy Based On Number Of Dribbles Before Shooting -- Accuracy Based On Touc
totalAttemptsByDribbles = df.groupby('DRIBBLES', observed=False).size()
madeAttemptsByDribbles = df[df['SHOT_RESULT'] == 'made'].groupby('DRIBBLES', ob
accuracyByDribbles = madeAttemptsByDribbles / totalAttemptsByDribbles * 100
dribbles = totalAttemptsByDribbles.index
bins = list(range(0, int(df['TOUCH_TIME'].max()+1)))
labels = [f"{x}-{x+1}" for x in bins[:-1]]
labels.append(str(max(bins))+'+')
bins.append(float('inf'))
totalAttemptsByTouchTime = df.groupby(pd.cut(threePoints['TOUCH_TIME'], bins=bins
madeAttemptsByTouchTime = df[df['SHOT_RESULT'] == 'made'].groupby(pd.cut(three
accuracyByTouchTime = madeAttemptsByTouchTime / totalAttemptsByTouchTime * 10
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))
bars1 = ax1.bar(dribbles, accuracyByDribbles)
ax1.set_title("Accuracy Based On Number Of Dribbles Before Shooting", fontsize=16)
ax1.set_xlabel("Dribbles", fontsize=14)
ax1.set_ylabel("Accuracy (%)", fontsize=14)
```

```
for bar in bars1:
    ax1.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
             f'{bar.get_height():.1f}%', ha='center', va='bottom', fontsize=8)
bars2 = ax2.bar(labels, accuracyByTouchTime)
ax2.set_title("Accuracy Based On Touch Time Before Shooting", fontsize=16)
ax2.set_xlabel("Touch Time (sec)", fontsize=14)
ax2.set_ylabel("Accuracy", fontsize=14)
for bar in bars2:
    ax2.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
             f'{bar.get_height():.1f}%', ha='center', va='bottom', fontsize=8)
plt.tight_layout()
```

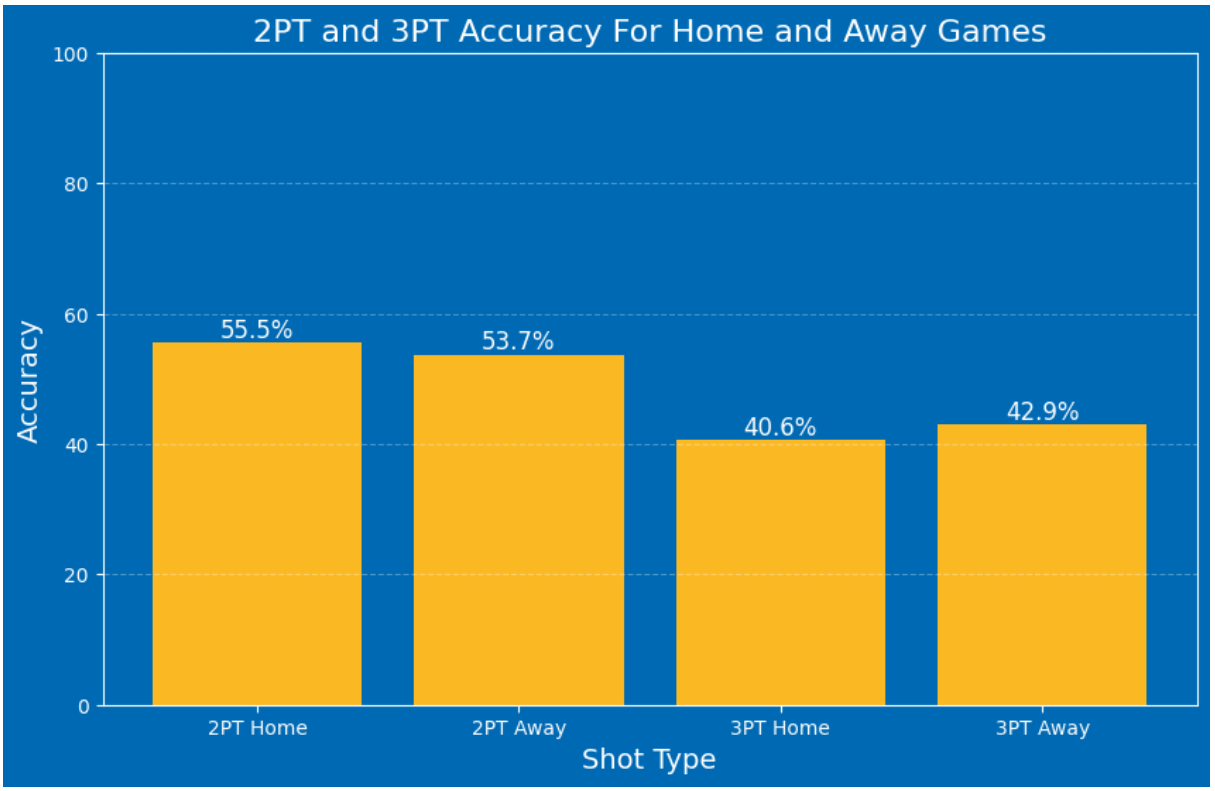


13. 2PT and 3PT Accuracy for Home and Away Games

Finally, we will analyze Curry's shooting accuracy for both 2PT and 3PT shots in home and away games.

```
# 2PT Accuracy For Home and Away Games - 3PT Accuracy For Home and Away Gam
twoPoints = df[df['PTS_TYPE'] == 2]
threePoints = df[df['PTS_TYPE'] == 3]
totalTwoPoints = twoPoints.groupby('LOCATION')['SHOT_RESULT'].count()
madeTwoPoints = twoPoints[twoPoints['SHOT_RESULT'] == 'made'].groupby('LOCATIC
twoPointsAccuracy = (madeTwoPoints / totalTwoPoints * 100).fillna(0)
totalThreePoints = threePoints.groupby('LOCATION')['SHOT_RESULT'].count()
madeThreePoints = threePoints[threePoints['SHOT_RESULT'] == 'made'].groupby('LOC
threePointsAccuracy = (madeThreePoints / totalThreePoints * 100).fillna(0)
accuracyByLocation = list(twoPointsAccuracy.values) + list(threePointsAccuracy.value
labels = ['2PT Home', '2PT Away', '3PT Home', '3PT Away']
```

```
plt.figure(figsize=(10, 6))
bars = plt.bar(labels, accuracyByLocation)
plt.title("2PT and 3PT Accuracy For Home and Away Games", fontsize=16)
plt.xlabel("Shot Type", fontsize=14)
plt.ylabel("Accuracy", fontsize=14)
plt.ylim(0, 100)
plt.grid(axis='y', linestyle='--', alpha=0.3, color=white)
for bar in bars: # Annotating bars
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
             f'{bar.get_height():.1f}%', ha='center', va='bottom', fontsize=12)
```



14. Conclusion

This concludes our analysis of Stephen Curry's shot data for the 2014-2015 season. Through various visualizations, we explored his shooting performance across different distances, periods of the game, clutch situations, and home vs. away games. The findings provide valuable insights into his shooting patterns and efficiency.