Project Objectives

The primary objective of this project is to analyse the WPL Auction 2023 dataset sourced from Kaggle, aiming to provide valuable insights for informed decision-making. Through comprehensive exploratory data analysis (EDA), the project intends to gain a deep understanding of the dataset's structure and content. Utilizing statistical analysis and data visualization techniques, the project aims to uncover trends, patterns, and anomalies in player auctions, with a specific focus on evaluating team strategies, identifying promising players signed at reasonable prices, and detecting instances of overpaid players. Ultimately, the project seeks to provide actionable recommendations to cricket teams, fans, and stakeholders, enhancing their understanding of IPL auctions and enabling more informed choices in team building and player recruitment.

General Description of Data

We have two datasets for our project:

Auction Dataset: This dataset provides information about the players who participated in the auction. It includes details like the player's name, which country they represent, the team they were bought by, the price they were bought for (both in Indian Rupees and US Dollars). This dataset helps us understand how much teams were willing to pay for each player.

Player Information Dataset: This dataset contains information about the players themselves. It includes data like the player's team, their role in the game (like batsman or bowler), their full name, date of birth, age, and the national team they belong to. It also tells us their batting and bowling styles, as well as the different teams they have played for in the past. This dataset helps us know more about the players' backgrounds and playing styles.

By using both of these datasets, we can analyse and gain insights into how different players were valued in the auction and understand more about the players' backgrounds and abilities.

Reading the dataset

```
auction_df = pd.read_csv('auction_2023.csv')
players_info_df = pd.read_csv('players_info_2023.csv')
display(auction_df.head())
display(players_info_df.head())
```

	Player	National Side	Team	Price	Price (₹)	Price (US$)
0	Smriti Mandhana	India	Royal Challengers Bangalore	₹3.4 crore (US$430,000)	₹3.4 crore	430000
1	Harmanpreet Kaur	India	Mumbai Indians	₹1.8 crore (US$230,000)	₹1.8 crore	230000
2	Sophie Devine	New Zealand	Royal Challengers Bangalore	₹50 lakh (US$63,000)	₹50 lakh	63000
3	Ashleigh Gardner	Australia	Gujarat Giants	₹3.2 crore (US$400,000)	₹3.2 crore	400000
4	Ellyse Perry	Australia	Royal Challengers Bangalore	₹1.7 crore (US$210,000)	₹1.7 crore	210000

	Team	Player	Role	Full Name	Born	Age	National Side	Batting Style	Bowling	Teams Played
0	Up Warriorz	Shabnim Ismail	Bowler	Shabnim Ismail	October 5, 1988 Cape Town, Cape Province	34 Years, 4 Months, 22 Days	South Africa	Left Handed	Right-arm fast medium	South Africa Women, Melbourne Renegades Women,...
1	Up Warriorz	Rajeshwari Gayakwad	Bowler	Rajeshwari Shivanand Gayakwad	June 1, 1991 Bijapur, Karnataka	31 Years, 8 Months, 25 Days	India	Right Handed	Slow left-arm orthodox	India Women, Karnataka, Supernovas, Trailblaze...
2	Up Warriorz	Deepti Sharma	Bowler	Deepti Bhagwan Sharma	August 24, 1997 Saharanpur, Uttar Pradesh	25 Years, 6 Months, 3 Days	India	Left Handed	Off break	India Women, Trailblazers, Western Storm, Indi...
3	Up Warriorz	Shweta Sehrawat	Bowler	Shweta Sanjay Sehrawat	May 10, 2006 Bulandshahr, Uttar Pradesh	18 Years, 11 Months, 30 Days	India	Right Handed	Offbreak	Delhi Women, India A Women Under-19, India B W...
4	Up Warriorz	Parshavi Chopra	All Rounder	Parshavi Chopra	February 26, 2004 Delhi, Delhi	16 Years, 9 Months, 15 Days	India	Right Handed	Leg break	India A Women Under-19, India Women Under-19

```python
display(auction_df.info())
display(players_info_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 87 entries, 0 to 86
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Player         87 non-null     object
 1   National Side  87 non-null     object
 2   Team           87 non-null     object
 3   Price          87 non-null     object
 4   Price (₹)      87 non-null     object
 5   Price (US$)    87 non-null     int64
dtypes: int64(1), object(5)
memory usage: 4.2+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 87 entries, 0 to 86
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Team           87 non-null     object
 1   Player         87 non-null     object
 2   Role           87 non-null     object
 3   Full Name      87 non-null     object
 4   Born           87 non-null     object
 5   Age            87 non-null     object
 6   National Side  87 non-null     object
 7   Batting Style  87 non-null     object
 8   Bowling        87 non-null     object
 9   Teams Played   87 non-null     object
dtypes: object(10)
memory usage: 6.9+ KB
```

Analysis

To initiate our analysis, our first step involves data preprocessing. Initially, we will extract the age in years from the "Age" column by splitting the values within this column using the keyword "Years" and selecting the first part of the split values.

```
# Extract Age in Years from Age column. Split value in Age column by Years, keyword and take the first value.
players_info_df['Age(Years)'] = players_info_df['Age'].apply(lambda x: x.split('Years')[0].strip() if 'Years' in x else x.split('years')[0].strip()).astype(int)
players_info_df['Age(Years)'].value_counts()
```

```
25    11
29     7
26     6
22     6
30     6
23     5
27     5
31     5
20     4
33     4
32     4
24     4
19     4
34     3
28     3
18     3
15     2
16     2
21     2
17     1
Name: Age(Years), dtype: int64
```

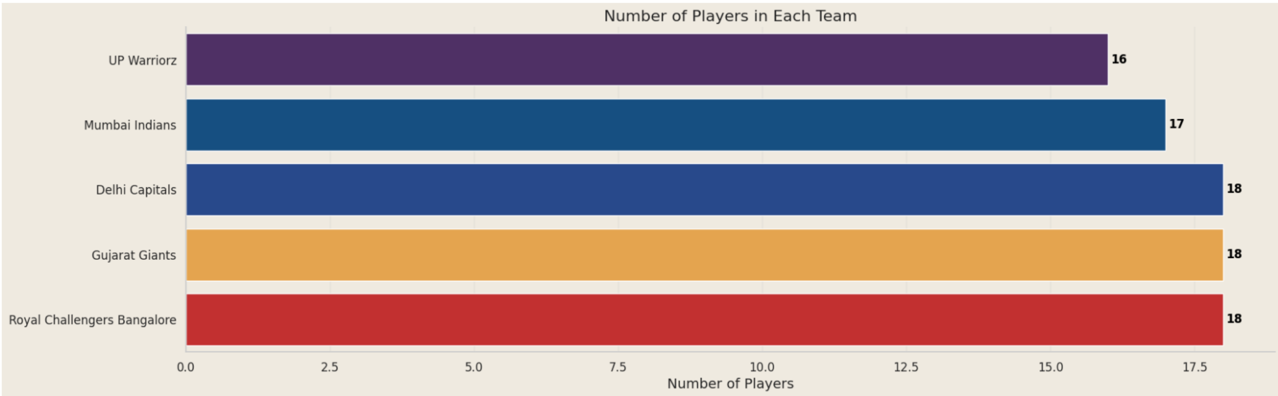
```
players_info_df.drop(columns=['Team', 'National Side'], inplace=True)
auction_df = auction_df.merge(players_info_df, on='Player', how='left')
display(auction_df.head())
```

	Player	National Side	Team	Price	Price (₹)	Price (US$)	Role	Full Name	Born	Age	Batting Style	Bowling	Teams Played	Age(Years)
0	Smriti Mandhana	India	Royal Challengers Bangalore	₹3.4 crore (US$430,000)	₹3.4 crore	430000	Batsman	Smriti Shriniwas Mandhana	July 18, 1996 Mumbai, Maharashtra	26 Years, 7 Months, 9 Days	Left Handed	Off break	India Women, Brisbane Heat Women, Hobart Hurri...	26
1	Harmanpreet Kaur	India	Mumbai Indians	₹1.8 crore (US$230,000)	₹1.8 crore	230000	Batsman	Harmanpreet Kaur Bhullar	March 8, 1989 Punjab	33 Years, 11 Months, 19 Days	Right Handed	Off break	India Women, Sydney Thunder Women, Supernovas,...	33
2	Sophie Devine	New Zealand	Royal Challengers Bangalore	₹50 lakh (US$63,000)	₹50 lakh	63000	All Rounder	Sophie Frances Monique Devine	September 1, 1989 Wellington	33 Years, 5 Months, 25 Days	Right Handed	Right-arm medium	New Zealand Women, Adelaide Strikers Women, Su...	33
3	Ashleigh Gardner	Australia	Gujarat Giants	₹3.2 crore (US$400,000)	₹3.2 crore	400000	All Rounder	Ashleigh Katherine Gardner	April 15, 1997 Bankstown, Sydney, New South Wales	25 Years, 10 Months, 11 Days	Right Handed	Off break	Australia Women, Sydney Sixers Women, New Sout...	25
4	Ellyse Perry	Australia	Royal Challengers Bangalore	₹1.7 crore (US$210,000)	₹1.7 crore	210000	Bowler	Ellyse Alexandra Perry	November 3, 1990 Wahroonga, Sydney, New South ...	32 Years, 3 Months, 23 Days	Right Handed	Right-arm fast medium	Australia Women, Sydney Sixers Women, Supernov...	32

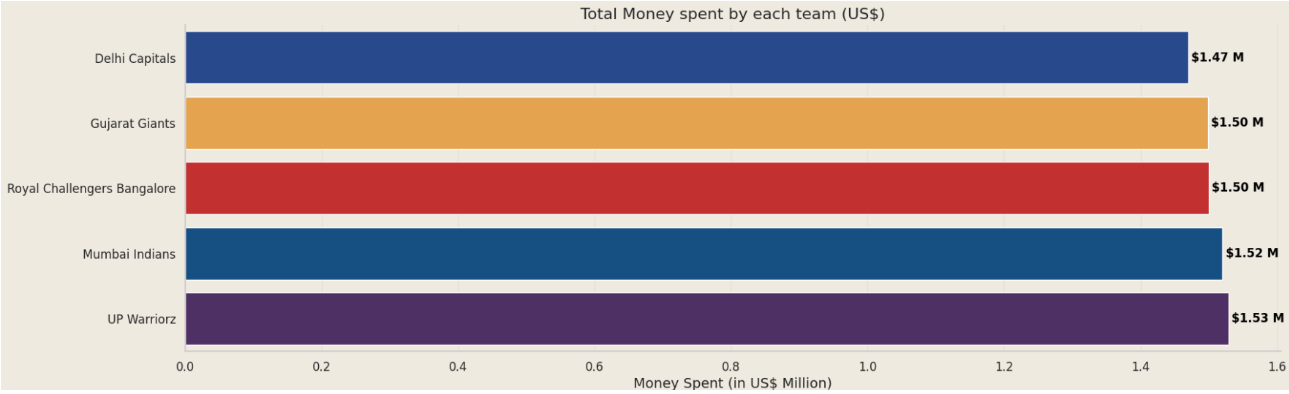
We will begin by conducting a team-wise analysis, initially focusing on the distribution of players among the top five teams. Following that, we will calculate the average player price for each team, providing valuable insights into the team composition and financial strategies employed in the auction.

Team Wise Analysis

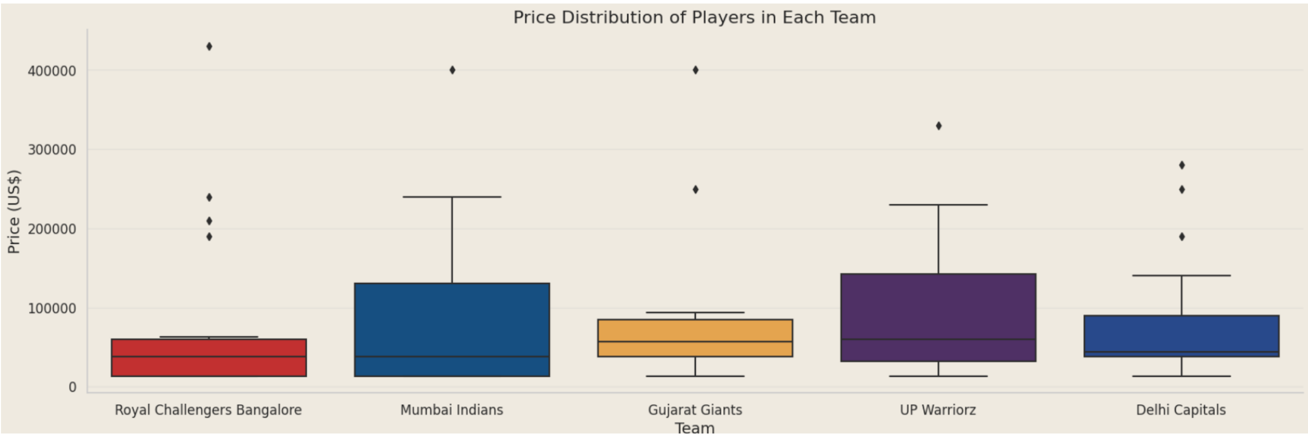
```python
team_specific_palette = {
    'UP Warriorz': '#50276E',
    'Mumbai Indians': '#045093',
    'Delhi Capitals': '#17449B',
    'Gujarat Giants': '#FDA736',
    'Royal Challengers Bangalore': '#DA1818'
}

fig = plt.figure(figsize=(20, 6))

players_per_team = auction_df.groupby('Team')['Player'].count().reset_index().sort_values(by='Player', ascending=True)

ax = sns.barplot(y='Team', x='Player', data=players_per_team, palette=team_specific_palette)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold')
plt.title('Number of Players in Each Team', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('')
plt.xlabel('Number of Players', fontsize=14)
plt.show();
```




```python
fig = plt.figure(figsize=(20, 6))

money_spent_by_each_team = auction_df.groupby('Team')['Price (US$)'].sum().reset_index().sort_values(by='Price (US$)', ascending=True)
money_spent_by_each_team['Price (US$)'] = money_spent_by_each_team['Price (US$)'] / 1000000
money_spent_by_each_team['Price (US$) Label'] = money_spent_by_each_team['Price (US$)'].apply(lambda x: f'${x:.2f} M')

ax = sns.barplot(y='Team', x='Price (US$)', data=money_spent_by_each_team, palette=team_specific_palette)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold', labels=money_spent_by_each_team['Price (US$) Label'])
plt.title('Total Money spent by each team (US$)', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('')
plt.xlabel('Money Spent (in US$ Million)', fontsize=14)
plt.show();
```

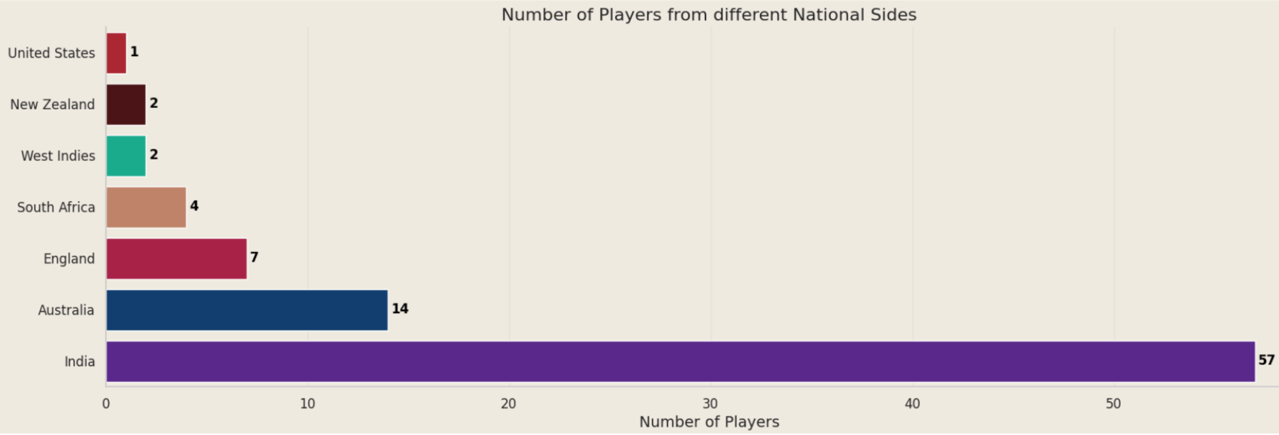



```
fig = plt.figure(figsize=(20, 6))

ax = sns.boxplot(x='Team', y='Price (US$)', data=auction_df, palette=team_specific_palette)
plt.title('Price Distribution of Players in Each Team', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('Price (US$)', fontsize=14)
plt.xlabel('Team', fontsize=14)
plt.show();
```

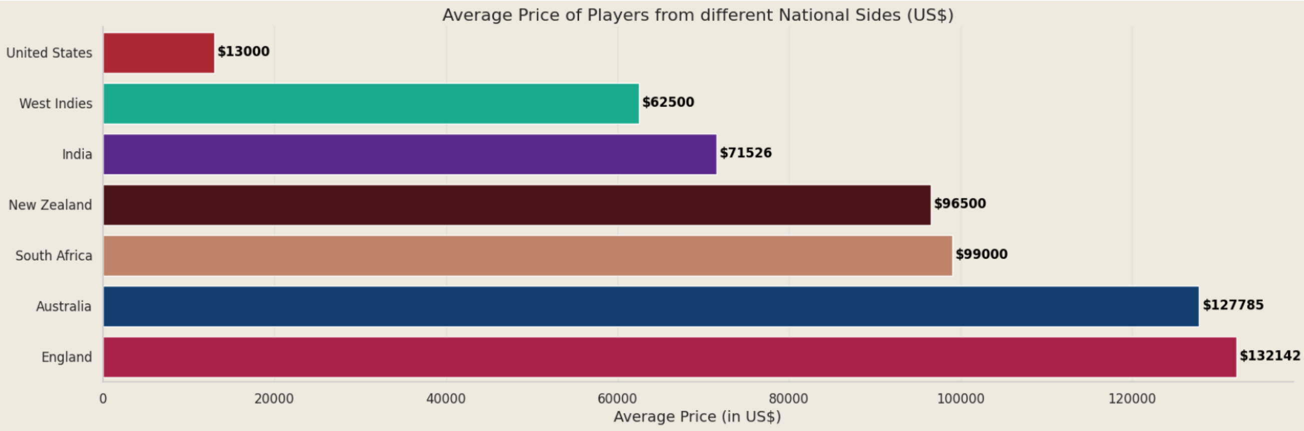


Our next step involves analysing players from various national sides. We will then determine the average prices they received in the auction. Finally, we will examine the distribution of player prices based on their respective national sides, shedding light on the financial dynamics of players from different countries.

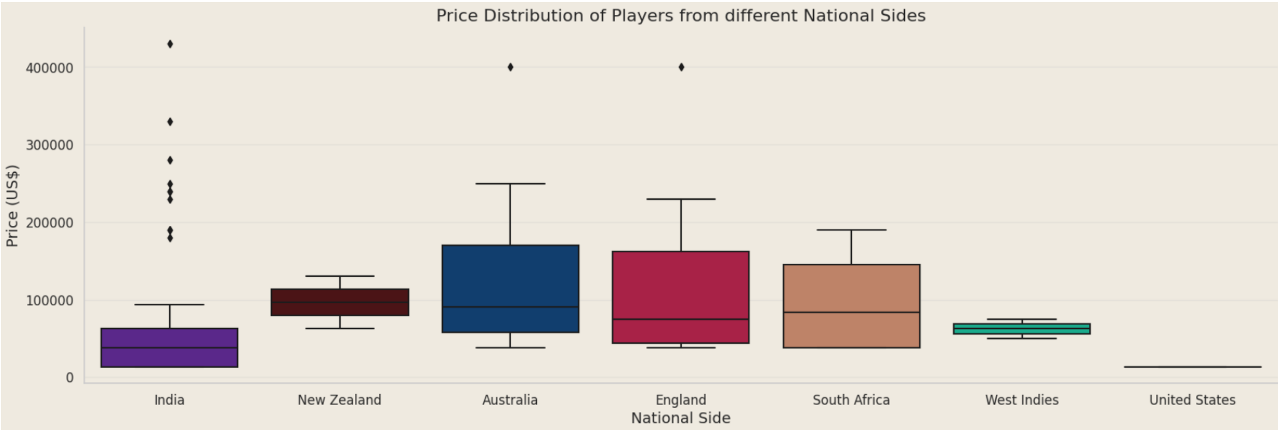
```python
national_side_palette = ['#c1121f', '#540b0e', '#02c39a', '#cd7e59', '#BE0C3D', '#023e7d', '#5a189a', '#b4d2b1',]
players_per_national_side = auction_df.groupby('National Side')['Player'].count().reset_index().sort_values(by='Player', ascending=True)
national_side_palettes = {
    team:color for team, color in zip(players_per_national_side['National Side'].values, national_side_palette)
}

fig = plt.figure(figsize=(20, 6))
ax = sns.barplot(y='National Side', x='Player', data=players_per_national_side, palette=national_side_palettes)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold')
plt.title('Number of Players from different National Sides', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('')
plt.xlabel('Number of Players', fontsize=14)
plt.show();
```




```python
average_price_per_national_side = auction_df.groupby('National Side')['Price (US$)'].mean().reset_index().sort_values(by='Price (US$)', ascending=True)
average_price_per_national_side['Price (US$)'] = average_price_per_national_side['Price (US$)'].astype(int)
average_price_per_national_side['Price (US$) Label'] = average_price_per_national_side['Price (US$)'].apply(lambda x: f'${x}')

fig = plt.figure(figsize=(20, 6))
ax = sns.barplot(y='National Side', x='Price (US$)', data=average_price_per_national_side, palette=national_side_palettes)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold', labels=average_price_per_national_side['Price (US$) Label'])
plt.title('Average Price of Players from different National Sides (US$)', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('')
plt.xlabel('Average Price (in US$)', fontsize=14)
plt.show();
```



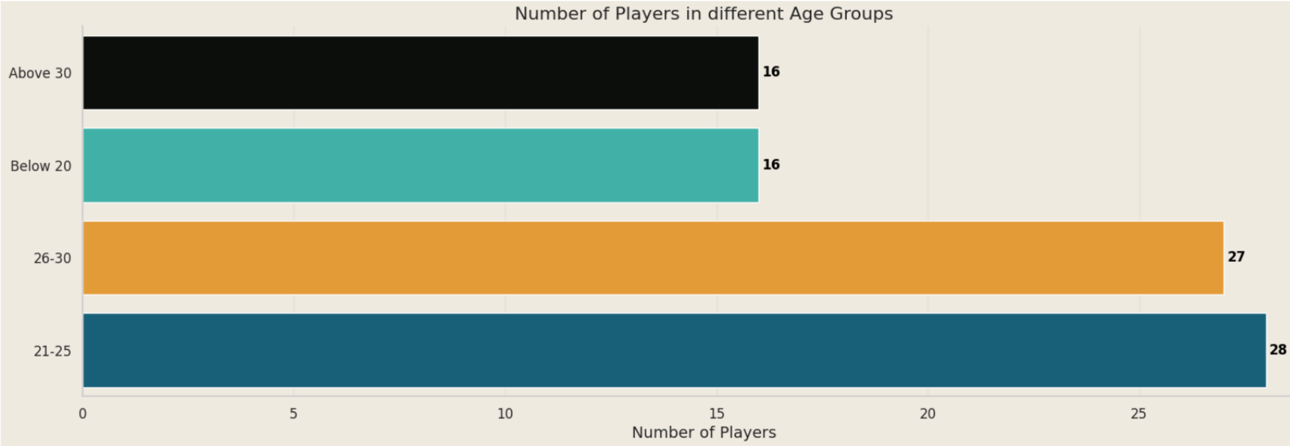
```python
fig = plt.figure(figsize=(20, 6))

ax = sns.boxplot(x='National Side', y='Price (US$)', data=auction_df, palette=national_side_palettes)
plt.title('Price Distribution of Players from different National Sides', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('Price (US$)', fontsize=14)
plt.xlabel('National Side', fontsize=14)
plt.show();
```



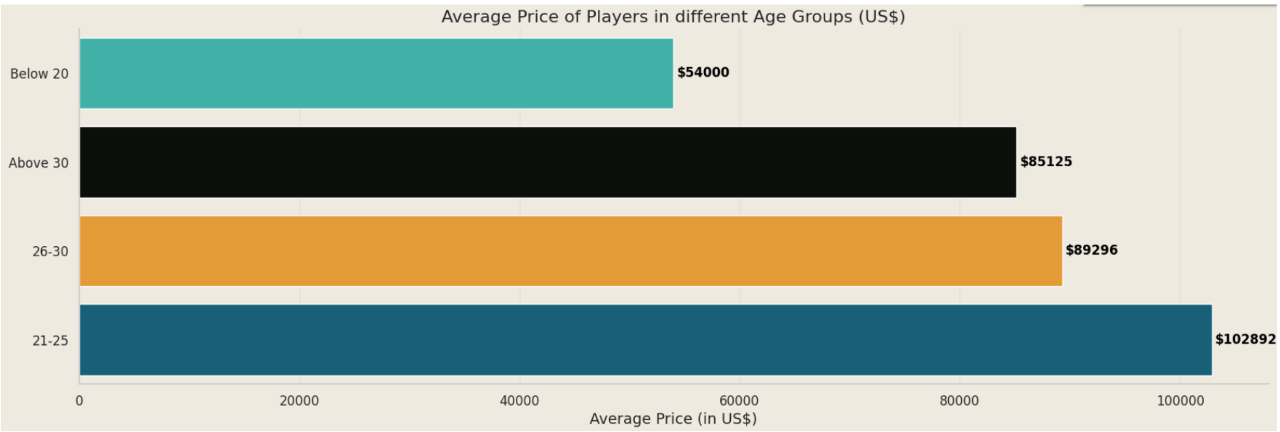
Our upcoming analysis will be age-group oriented. Initially, we will categorize players into distinct age groups. Subsequently, we will calculate the average player salaries within each age group. Finally, we will scrutinize the price distribution among these age groups, offering comprehensive insights into player salaries across various age brackets.

Age Group Wise Analysis

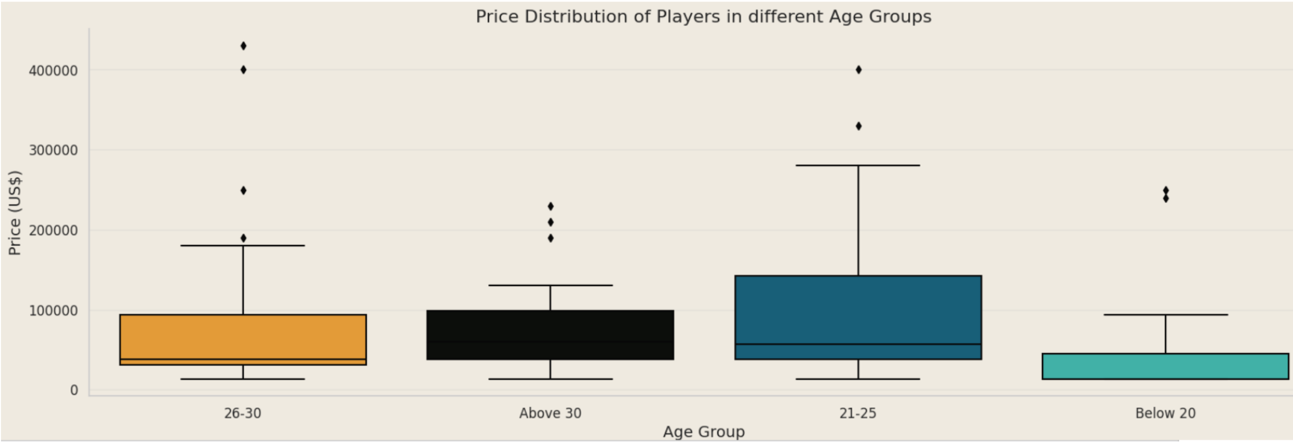
```python
age_group_palette = ['#0c0f0a', '#2ec4b6', '#ff9f1c', '#086788']

def player_age_age_group(age):
    if age <= 20:
        return 'Below 20'
    elif age <= 25:
        return '21-25'
    elif age > 25 and age <= 30:
        return '26-30'
    else:
        return 'Above 30'

auction_df['Age Group'] = auction_df['Age(Years)'].apply(player_age_age_group)
players_count_by_age_group = auction_df.groupby('Age Group')['Player'].count().reset_index().sort_values(by='Player', ascending=True)
age_group_palettes = {
    group:color for group, color in zip(players_count_by_age_group['Age Group'].values, age_group_palette)
}

fig = plt.figure(figsize=(20, 6))
ax = sns.barplot(y='Age Group', x='Player', data=players_count_by_age_group, palette=age_group_palettes)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold')
plt.title('Number of Players in different Age Groups', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('')
plt.xlabel('Number of Players', fontsize=14)
plt.show();
```




```python
average_price_by_age_group = auction_df.groupby('Age Group')['Price (US$)'].mean().reset_index().sort_values(by='Price (US$)', ascending=True)
average_price_by_age_group['Price (US$)'] = average_price_by_age_group['Price (US$)'].astype(int)
average_price_by_age_group['Price (US$) Label'] = average_price_by_age_group['Price (US$)'].apply(lambda x: f'${x}')

fig = plt.figure(figsize=(20, 6))
ax = sns.barplot(y='Age Group', x='Price (US$)', data=average_price_by_age_group, palette=age_group_palettes)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold', labels=average_price_by_age_group['Price (US$) Label'])
plt.title('Average Price of Players in different Age Groups (US$)', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('')
plt.xlabel('Average Price (in US$)', fontsize=14)
plt.show();
```



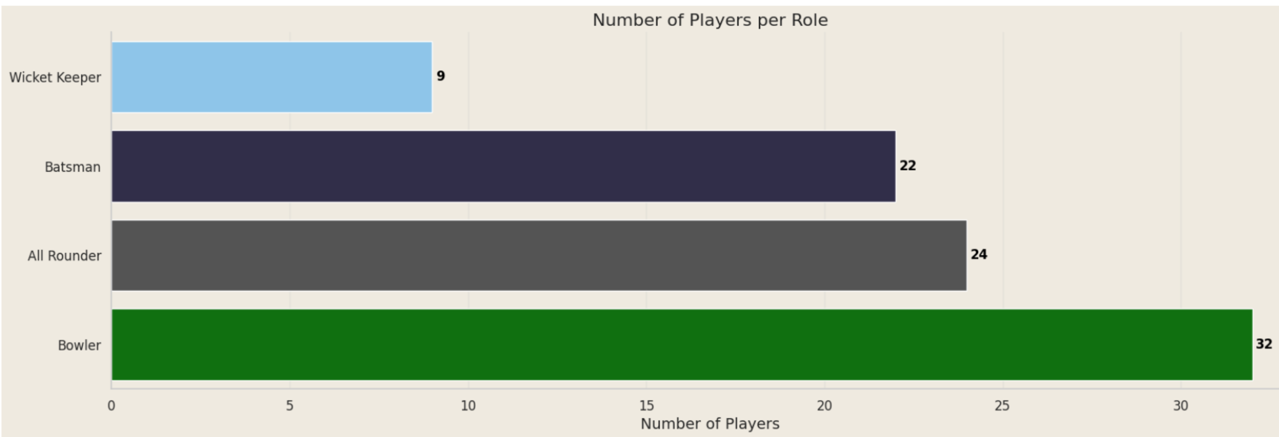

```python
fig = plt.figure(figsize=(20, 6))
ax = sns.boxplot(x='Age Group', y='Price (US$)', data=auction_df, palette=age_group_palettes)
plt.title('Price Distribution of Players in different Age Groups', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('Price (US$)', fontsize=14)
plt.xlabel('Age Group', fontsize=14)
plt.show();
```



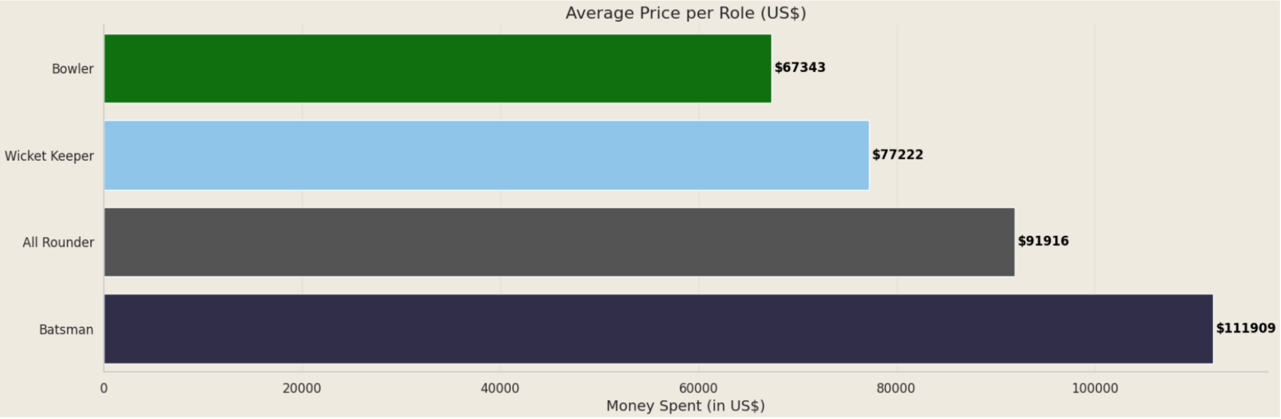
We will now delve into player role analysis, categorizing players as batsmen, bowlers, all-rounders, or wicketkeepers. Next, we'll calculate the average salaries for players in each role. Finally, we'll examine the price distribution for players based on their respective roles, providing valuable insights into how different roles are valued in the auction.

Player Role Analysis

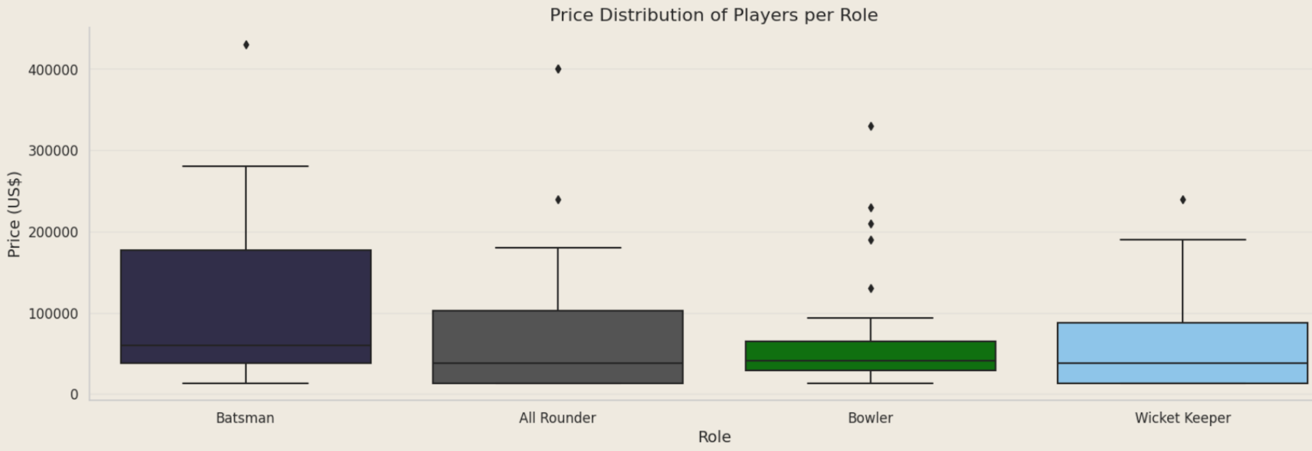
```
role_palette = ['#7fc8f8', '#2e294e', '#545454', '#008000']

players_per_role = players_info_df.groupby('Role')['Player'].count().reset_index().sort_values(by='Player', ascending=True)
role_palettes = {
    role:color for role, color in zip(players_per_role['Role'].values, role_palette)
}

fig = plt.figure(figsize=(20, 6))
ax = sns.barplot(y='Role', x='Player', data=players_per_role, palette=role_palettes)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold')
plt.title('Number of Players per Role', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('')
plt.xlabel('Number of Players', fontsize=14)
plt.show();
```



```python
average_price_per_role = auction_df.groupby('Role')['Price (US$)'].mean().reset_index().sort_values(by='Price (US$)', ascending=True)
average_price_per_role['Price (US$)'] = average_price_per_role['Price (US$)'].astype(int)
average_price_per_role['Price (US$) Label'] = average_price_per_role['Price (US$)'].apply(lambda x: f'${x}')

fig = plt.figure(figsize=(20, 6))
ax = sns.barplot(y='Role', x='Price (US$)', data=average_price_per_role, palette=role_palettes)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold', labels=average_price_per_role['Price (US$) Label'])
plt.title('Average Price per Role (US$)', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('')
plt.xlabel('Money Spent (in US$)', fontsize=14)
plt.show();
```




```python
fig = plt.figure(figsize=(20, 6))

ax = sns.boxplot(x='Role', y='Price (US$)', data=auction_df, palette=role_palettes)
plt.title('Price Distribution of Players per Role', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('Price (US$)', fontsize=14)
plt.xlabel('Role', fontsize=14)
plt.show();
```



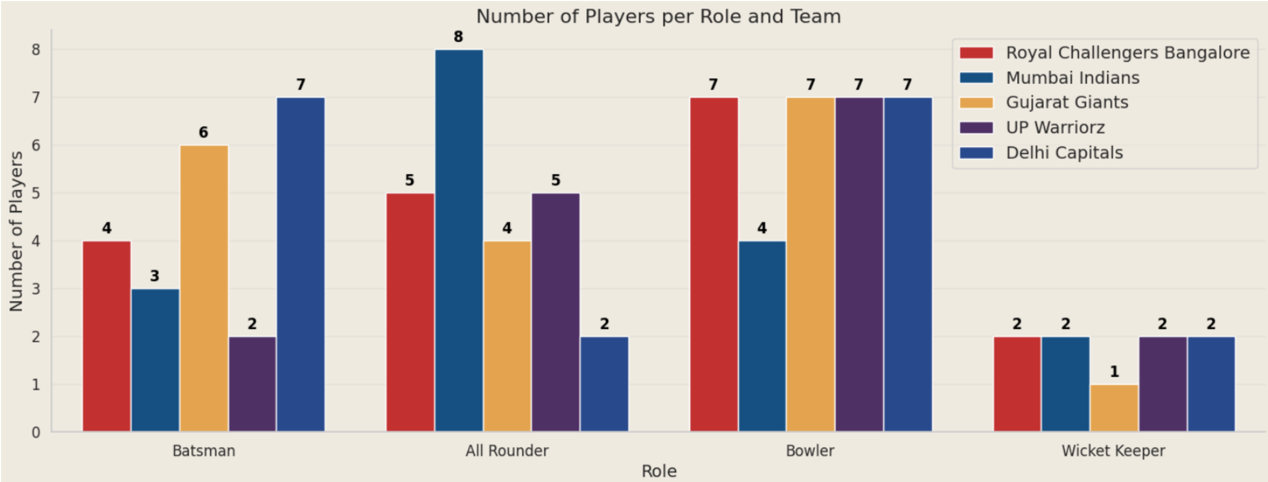
Our analysis will now focus on identifying the top 5 highest-priced players in the WPL 2023 auction, shedding light on the most expensive acquisitions in the tournament.

```
fig = plt.figure(figsize=(20, 6))
auction_df['Price (US$) Label'] = auction_df['Price (US$)'].apply(lambda x: f'${x}')
top_5_players_by_price = auction_df.sort_values(by='Price (US$)', ascending=False).head(5)
ax = sns.barplot(y='Player', x='Price (US$)', data=top_5_players_by_price.sort_values(by='Price (US$)', ascending=True))
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold', labels=top_5_players_by_price.sort_values(by='Price (US$)', ascending=True)['Price (US$) Label'])
plt.title('Top 5 Players based on Price (US$)', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('')
plt.xlabel('Money Spent (in US$)', fontsize=14)
plt.show();
```

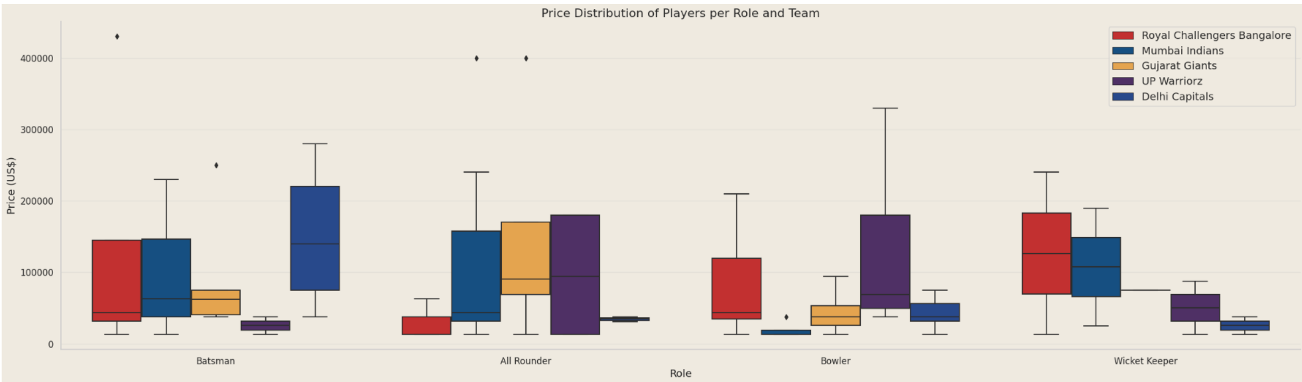
In this phase of our analysis, we will conduct a team and player role analysis. Firstly, we'll determine the number of batters, bowlers, all-rounders, and wicketkeepers in each team to understand team compositions. Then, we will explore the price distribution of players according to their roles within each team, offering insights into the financial strategies of teams and the value placed on specific player roles.

Team and Player Role Analysis

```
fig = plt.figure(figsize=(18, 6))

ax = sns.countplot(x='Role', data=auction_df, hue='Team', palette=team_specific_palette)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold')
ax.legend(loc='upper right', fontsize=14)
plt.title('Number of Players per Role and Team', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('Number of Players')
plt.xlabel('Role', fontsize=14)
plt.show();
```



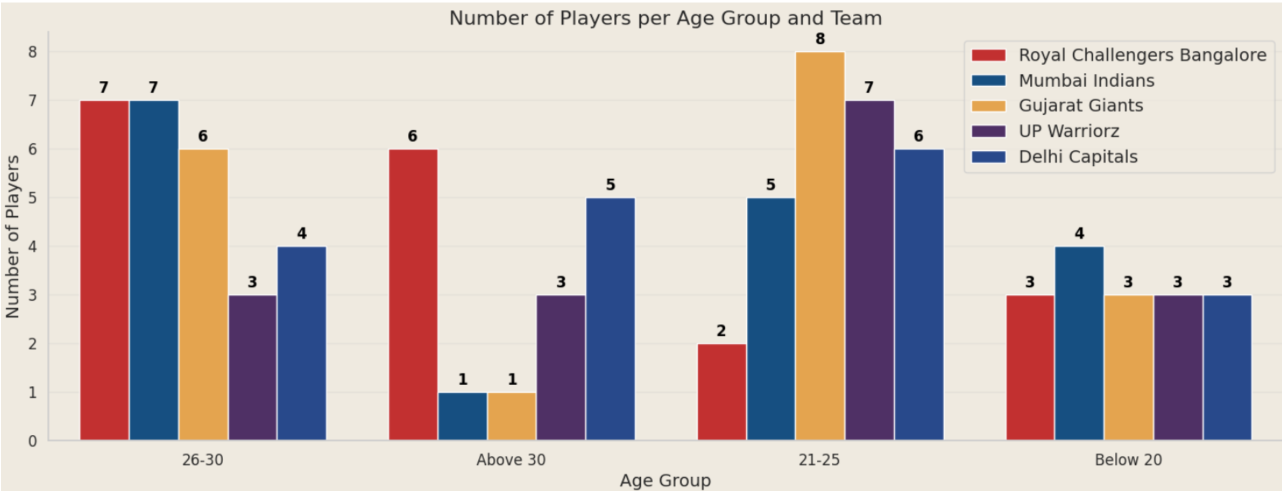

```
fig = plt.figure(figsize=(30, 8))
ax = sns.boxplot(x='Role', y='Price (US$)', data=auction_df, hue='Team', palette=team_specific_palette)
ax.legend(loc='upper right', fontsize=14)
plt.title('Price Distribution of Players per Role and Team', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('Price (US$)', fontsize=14)
plt.xlabel('Role', fontsize=14)
plt.show();
```



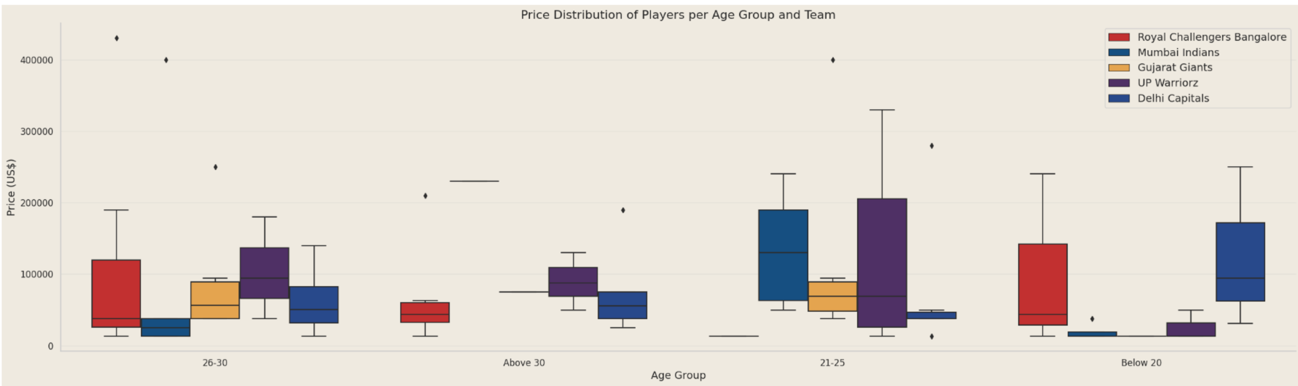
In this phase of our analysis, we will perform a team and age group analysis. Initially, we'll examine the composition of each team in terms of the number of batters, bowlers, all-rounders, and wicketkeepers across different age groups. Subsequently, we'll investigate the price distribution of players within each age group and team, providing insights into how teams value players of various ages and roles.

Team and Age Group Analysis

```python
fig = plt.figure(figsize=(18, 6))

ax = sns.countplot(x='Age Group', data=auction_df, hue='Team', palette=team_specific_palette)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold')
ax.legend(loc='upper right', fontsize=14)
plt.title('Number of Players per Age Group and Team', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('Number of Players')
plt.xlabel('Age Group', fontsize=14)
plt.show();
```



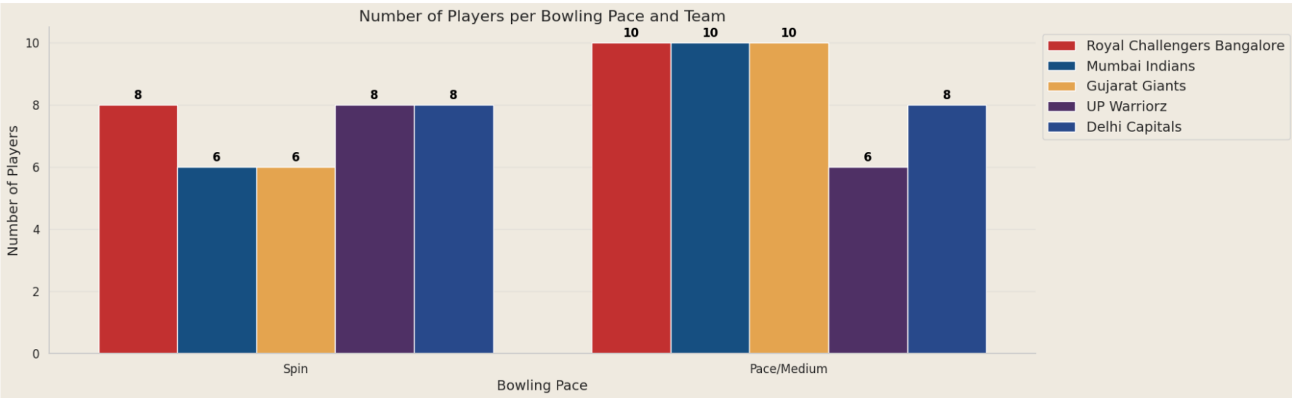

```python
fig = plt.figure(figsize=(30, 8))
ax = sns.boxplot(x='Age Group', y='Price (US$)', data=auction_df, hue='Team', palette=team_specific_palette)
ax.legend(loc='upper right', fontsize=14)
plt.title('Price Distribution of Players per Age Group and Team', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('Price (US$)', fontsize=14)
plt.xlabel('Age Group', fontsize=14)
plt.show();
```



In the upcoming phase of our analysis, we will conduct a team and bowling pace analysis. We'll categorize players into different groups based on their bowling style, distinguishing between spin, medium pace, and pace attack bowlers. This analysis will provide valuable insights into the bowling strategies adopted by each team.

Team and Bowling Pace Analysis

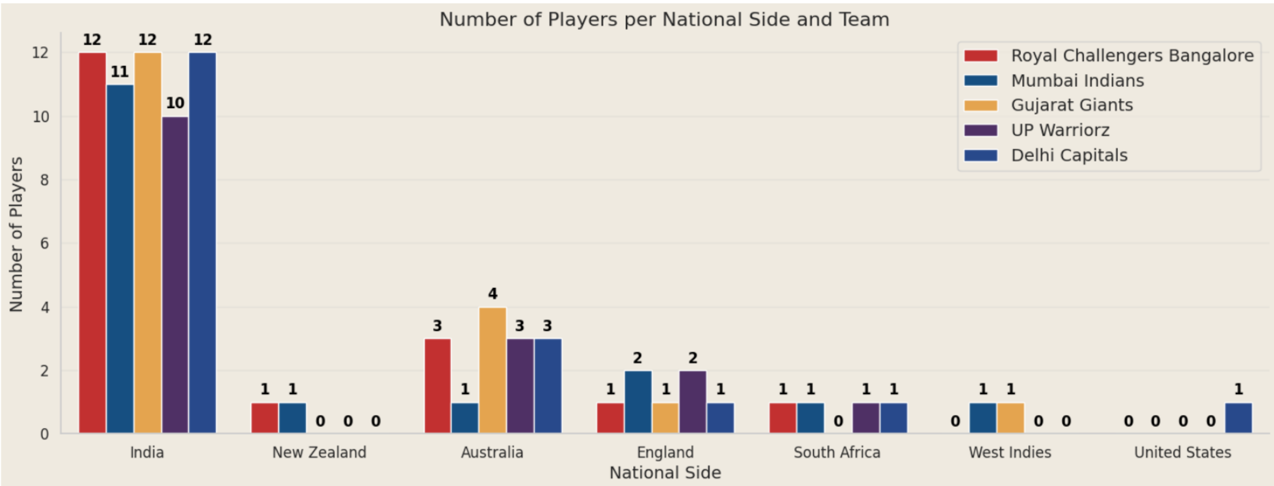
```python
def bowling_to_bowling_pace(bowling_style):
    if bowling_style == '-':
        return None
    elif 'medium' in bowling_style or 'fast' in bowling_style:
        return 'Pace/Medium'
    else:
        return 'Spin'

auction_df['Bowling Pace'] = auction_df['Bowling'].apply(bowling_to_bowling_pace)


fig = plt.figure(figsize=(18, 6))
ax = sns.countplot(x='Bowling Pace', data=auction_df, hue='Team', palette=team_specific_palette)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold')
ax.legend(loc='upper left', fontsize=14, bbox_to_anchor=(1, 1))
plt.title('Number of Players per Bowling Pace and Team', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('Number of Players')
plt.xlabel('Bowling Pace', fontsize=14)
plt.show();
```



In this segment of our analysis, we will explore the distribution of players from various national teams within each of the top 5 WPL teams. This examination will provide insights into the international diversity and representation among the players in these teams.

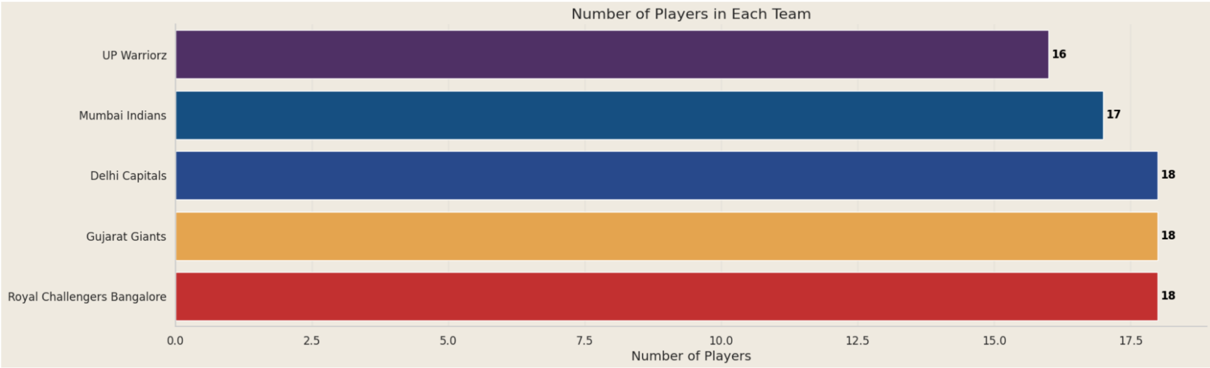
Team and National Side Analysis

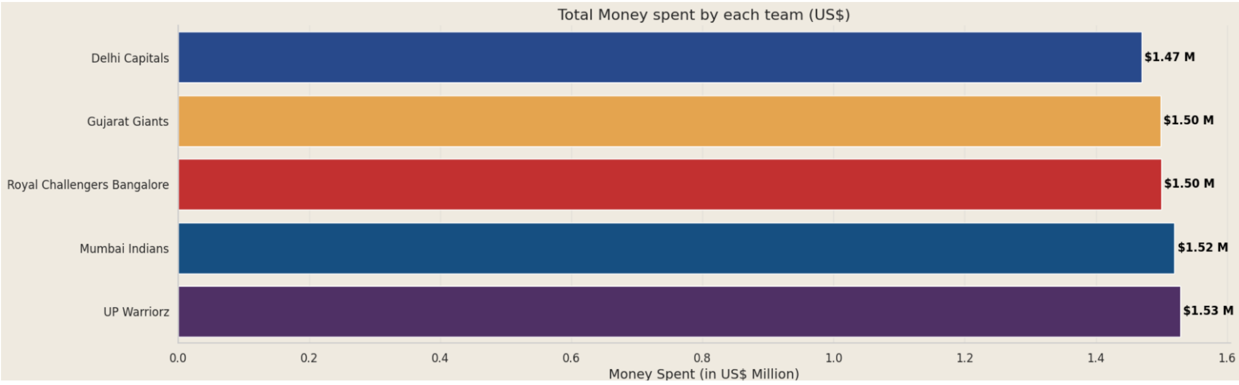
```python
fig = plt.figure(figsize=(18, 6))
ax = sns.countplot(x='National Side', data=auction_df, hue='Team', palette=team_specific_palette)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=12, color='black', padding=3, weight='bold')
ax.legend(loc='upper right', fontsize=14)
plt.title('Number of Players per National Side and Team', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylabel('Number of Players', fontsize=14)
plt.xlabel('National Side', fontsize=14)
plt.show();
```



Findings & Inferences

Team Wise Analysis

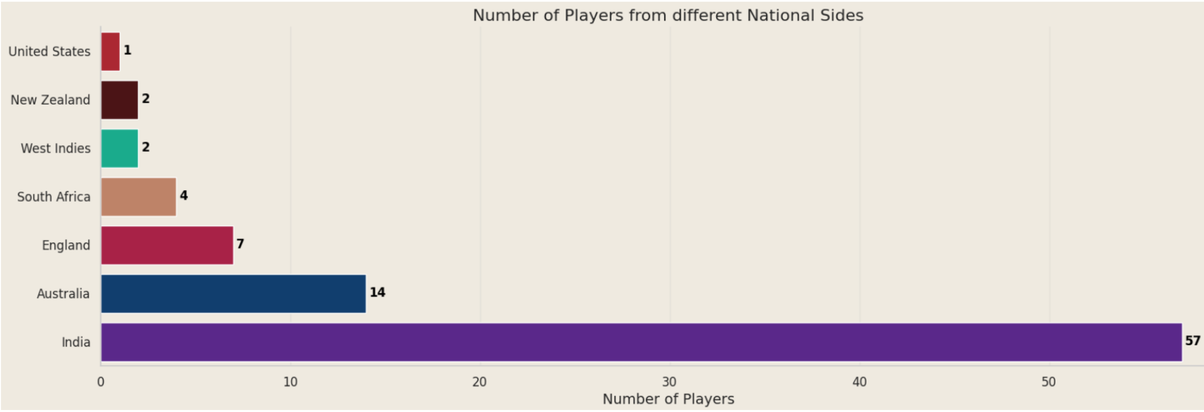


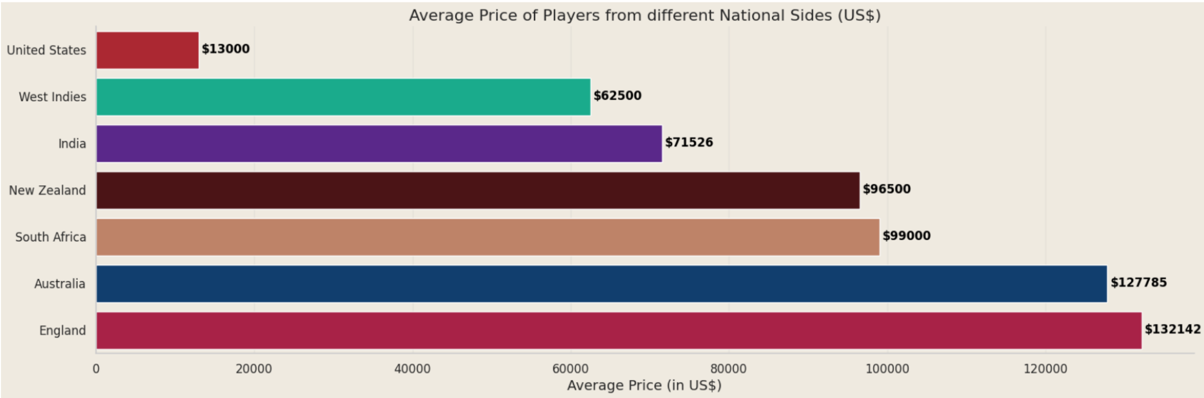


📌 Insights:

- 5 teams participated in this auction to buy 87 players
- UP Warriorz spent the most money($153M) and bought the minimum number of players(16).
- Average price of players bought by UP Warriorz is the highest($95562).
- Highest bid came from RCB which was for over $400K.

`National Side Analysis`

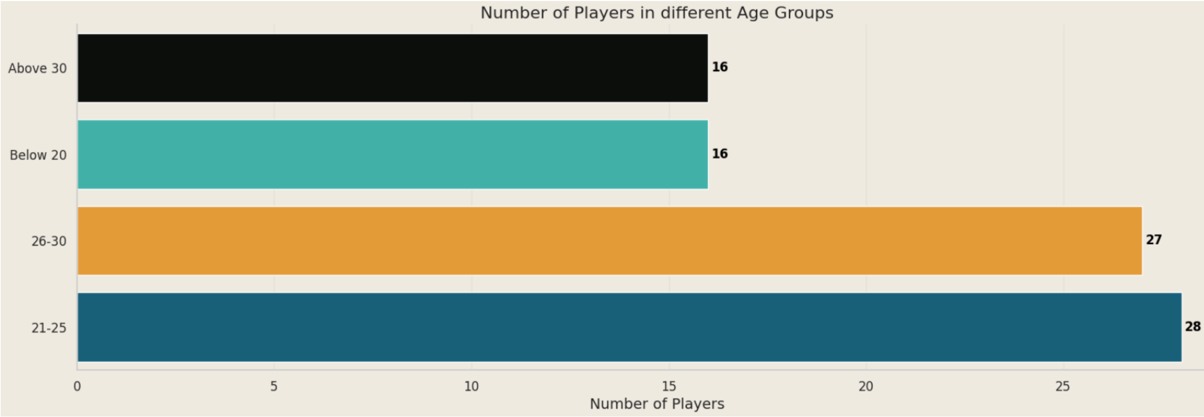


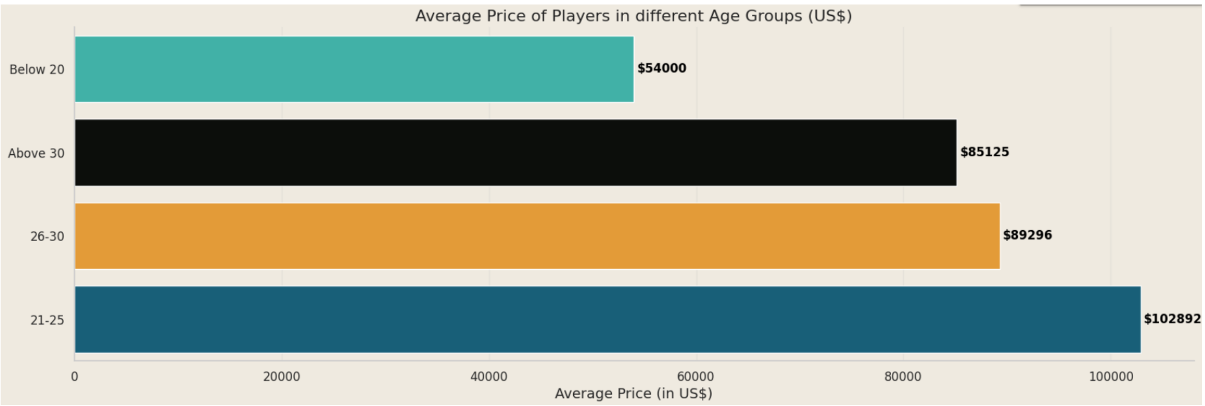


📌 **Insights:**

- Players from 7 national sides will be part of first WPL.
- Indian players are the most in number(57) followed by Australian(14) and England(7).
- Average price for English players is the highest followed by Australians.
- Highest bid was made for Indian Player

Age Group Wise Analysis

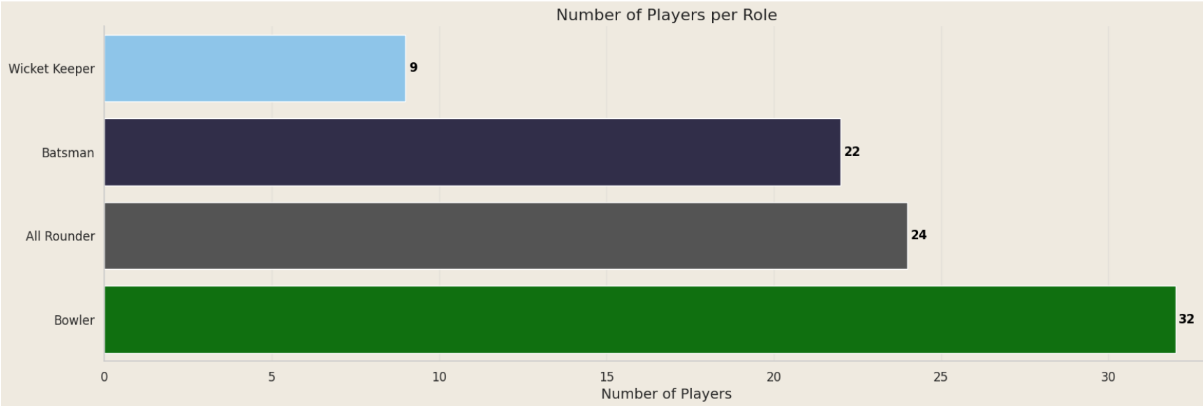


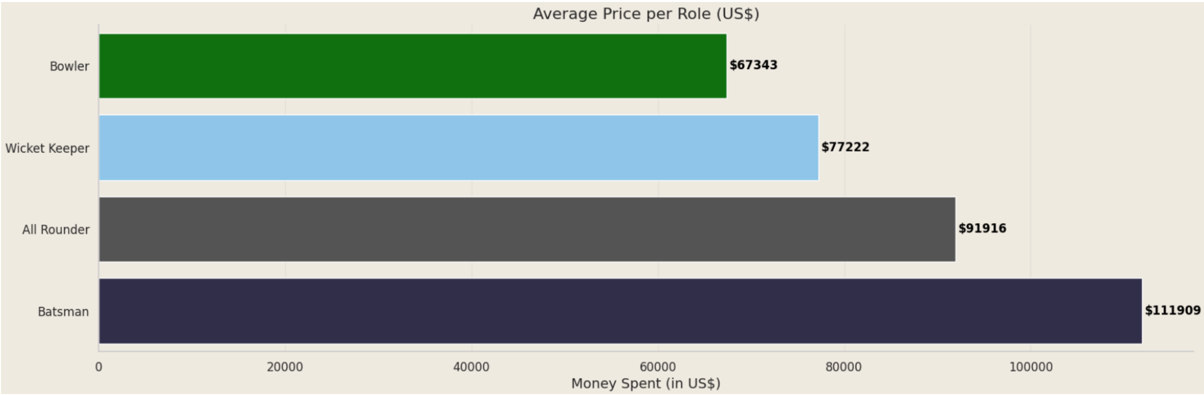


📌 **Insights:**

- Most of the players are in the age group of 21-25. Above 30 and below 20 both has only 16 players each.
- Players in the age group of 21-25 are the most expensive and players below 20 are the least expensive.
- Highest bids was made for a player in the age group of 26-30.

Player Role Analysis





📌 **Insights:**

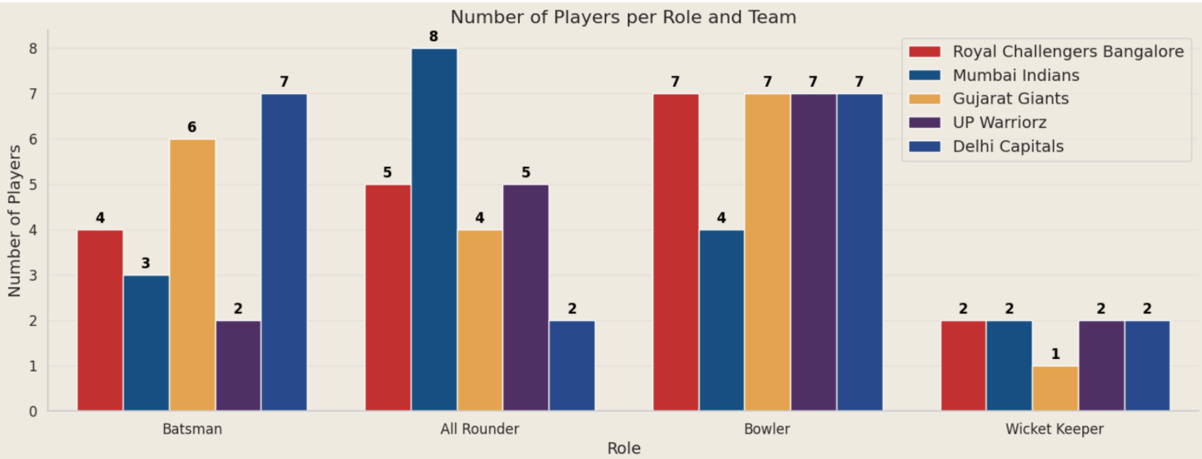
- Number of bowlers are the most(32) followed by all-rounders(24) and batters(22).
- Average price of batters is the highest($111909) followed by all-rounders.
- Highest bid was made for a batsman.

```
Top 5 Costliest Players
```

📌 **Insights:**

- Smriti Mandhana is the most expensive player in this auction.
- Out of top 5, 3 are Indian players. Other two are from Australia and England.
- Out of top 5, 2 each are batters and all-rounders and 1 is a bowler.

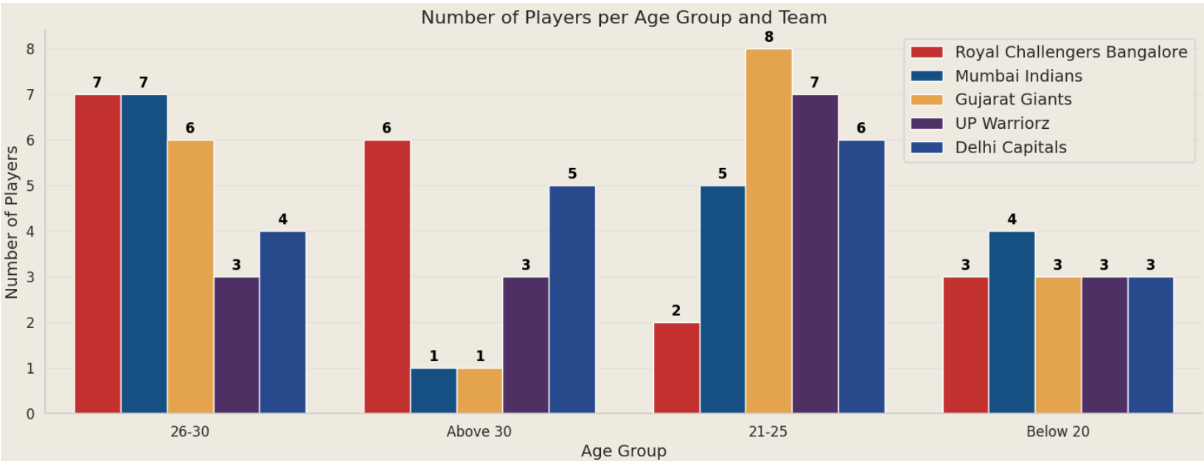
```
Team and Player Role Analysis
```



📌 **Insights:**

- Mumbai Indians has the most number of all rounders.
- Delhi Capitals has the most number of batters and bowlers. They seems to rely on specialist players instead of all-rounders.
- Delhi capitals has spent the least money on all-rounders.
- UP Warriorz has spend more on All rounders and bowlers but least on batsmen.

`Team and Age Group Analysis`



📌 **Insights:**

- 13 out of 18 players for RCB are above 26 years. This is the most among all the teams
- RCB and Delhi Capitals spent the most on players below 20 years.
- Mumbai Indians and Gujarat Giants has only 1 player each above 30 years.

Managerial Insights | Implications

The analysis of the WPL 2023 dataset yields several valuable managerial insights and implications for the cricket tournament:

Team Composition Optimization: By understanding the distribution of players in terms of roles (batters, bowlers, all-rounders, wicketkeepers), team managers can optimize their squad compositions. This can help in building well-balanced teams with the right mix of skills.

Financial Strategy Refinement: Examining price distributions for different player roles and age groups allows team managers to refine their financial strategies. They can identify cost-effective players and allocate budgets more efficiently.

National Representation: Understanding the number of players from different national teams within each squad helps managers assess the international diversity and make informed decisions regarding player recruitment and team dynamics.

Bowling Tactics: Categorizing bowlers into spin, medium pace, and pace attack groups assists in formulating bowling tactics. Teams can tailor their strategies to exploit opponents' weaknesses or adapt to various pitch conditions.

Player Valuation: Identifying the top 5 costliest players provides a benchmark for player valuation. Managers can assess whether the investment in high-priced players translates into on-field performance.

Age Group Insights: Analysing player age groups can aid in long-term team planning. Teams can balance youth and experience while identifying potential retirements or recruitment needs.

In conclusion, the dataset analysis equips team managers with actionable insights to make informed decisions regarding team composition, financial strategies, and tactics. These insights can significantly impact the success of teams in the WPL 2023 tournament.