Descriptive Statistics

Another important aspect of Descriptive Analysis is visualization of data to understand it better and d

```
# Importing Libraries
import math

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import scipy.stats as stats

import warnings
warnings.filterwarnings('ignore')
```

```
deliveries = pd.read_csv('/content/deliveries.csv')
matches = pd.read_csv('/content/matches.csv')
matches.head()
```

	id	season	city	date	team1	team2	toss_winner	toss_decision	resu
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	norm
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	norm
2	3	2017	Rajkot	2017-04-07	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	norm
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	norm

Bar Plot

A bar chart is a graph with rectangular bars. The graph usually compares different categories. Althou
(bars standing up) or horizontally (bars laying flat from left to right), the most usual type of bar graph

The horizontal (x) axis represents the categories; The vertical (y) axis represents a value for those cat

A bar graph is useful for looking at a set of data and making comparisons. For example, it's easier to s chunk of your budget by glancing at the above chart rather than looking at a string of numbers.

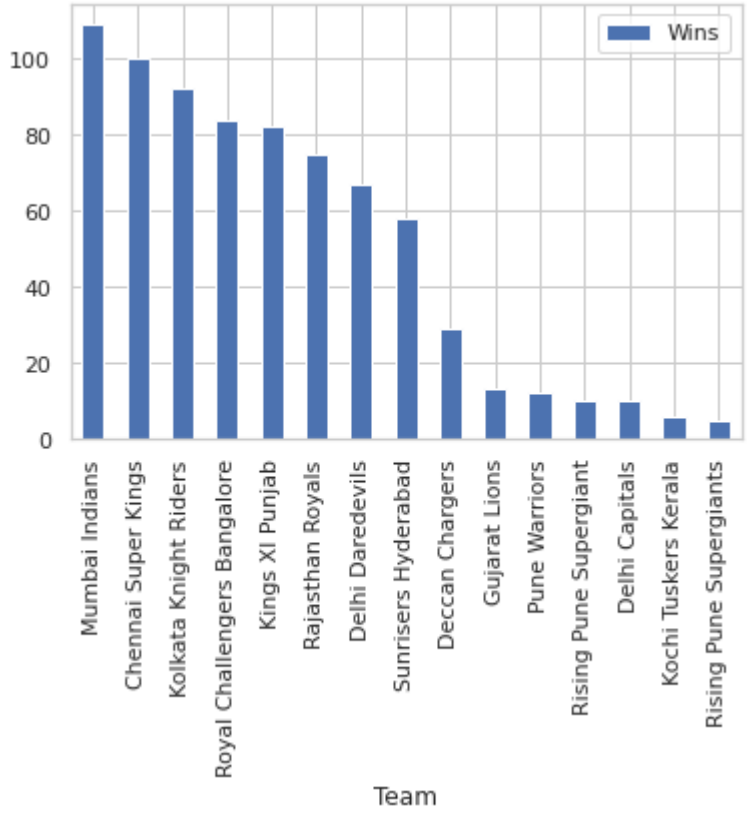
Bar charts can also represent more complex categories with stacked bar charts or grouped bar charts

Although they bar charts and histograms look the same, but they have one important difference: **they** used to plot discrete data whereas the continous data is plotted using Histograms.

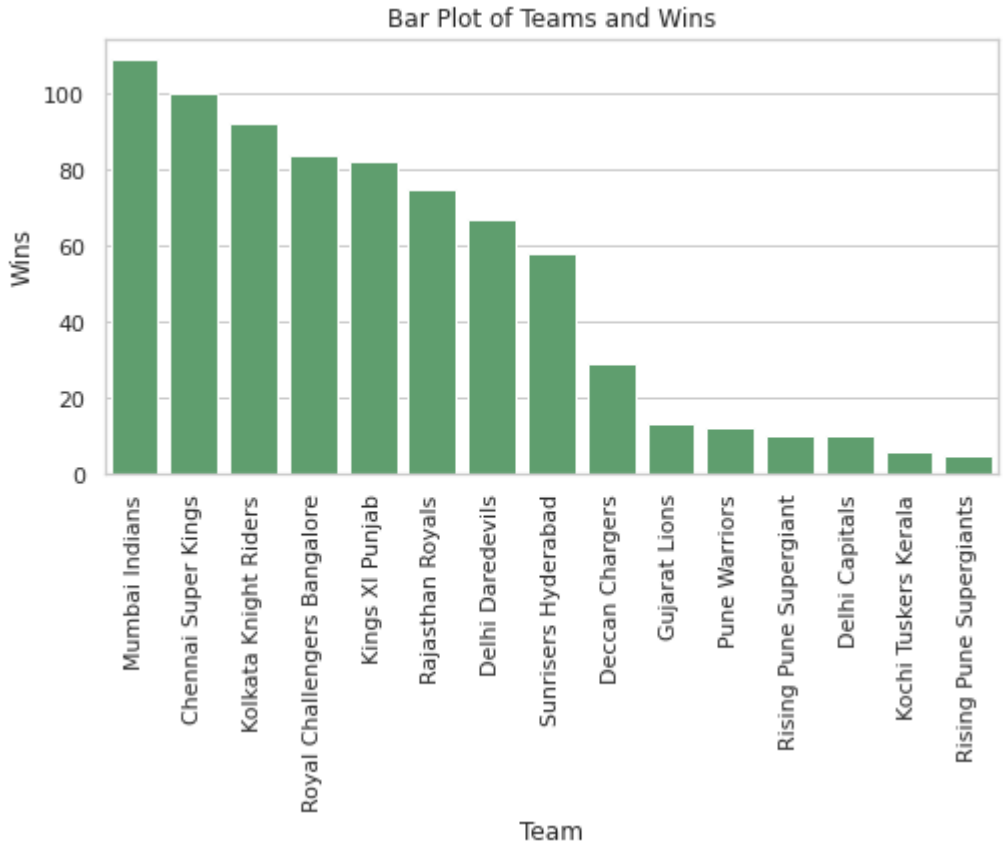
```
Wins = matches.winner.value_counts().rename_axis('Team').to_frame('Wins').reset_index()
```

```
# using Default Pandas Function
Wins.plot.bar(x = 'Team', rot = 90)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f86b615bb70>
```

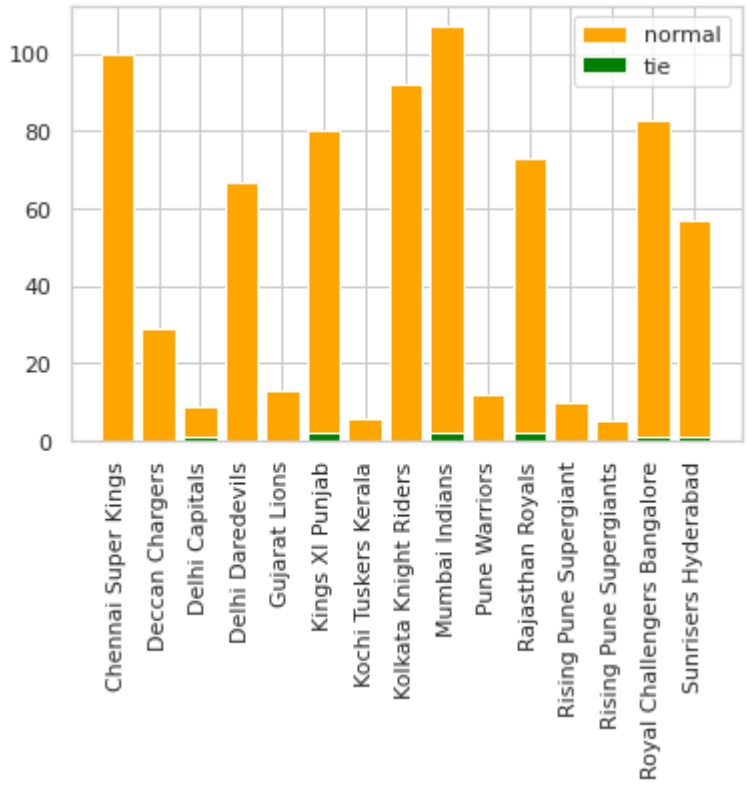



```
# Bar Plot to visualize the number of wins by a team

sns.set(style="whitegrid")

# Initialize the matplotlib figure
f, ax = plt.subplots(figsize=(8, 4))

# Plot the total wins by each team
sns.barplot(x="Team", y="Wins", data=Wins, color="g")
ax.set_xticklabels(Wins.Team, rotation = 90)
plt.title('Bar Plot of Teams and Wins')
plt.show()
```

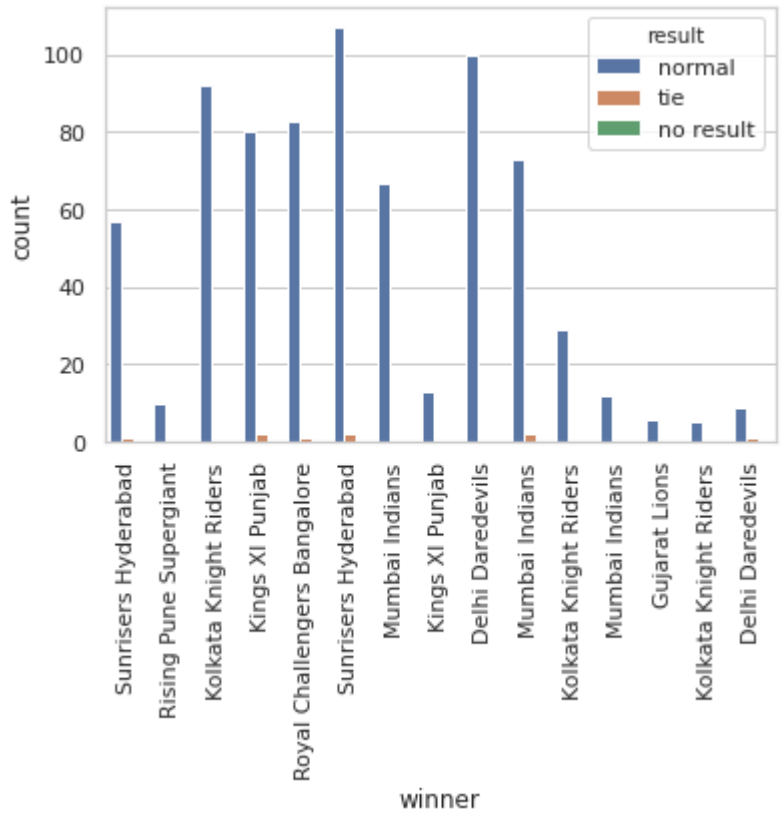



```
xtab = pd.crosstab(matches['winner'],matches['result']).reset_index()
```

```
# Stacked Chart
ax = plt.subplot(111)
ax.bar(xtab.winner,xtab.normal, color = 'orange')
ax.bar(xtab.winner,xtab.tie, color = 'green')
ax.set_xticklabels(xtab.winner,rotation = 90)
ax.legend(['normal','tie'])
plt.show()
```




```
# Categorical Count Plot
fi = sns.countplot(matches.winner, hue = matches.result)
fi.set_xticklabels(matches.winner,rotation = 90)
plt.show()
```

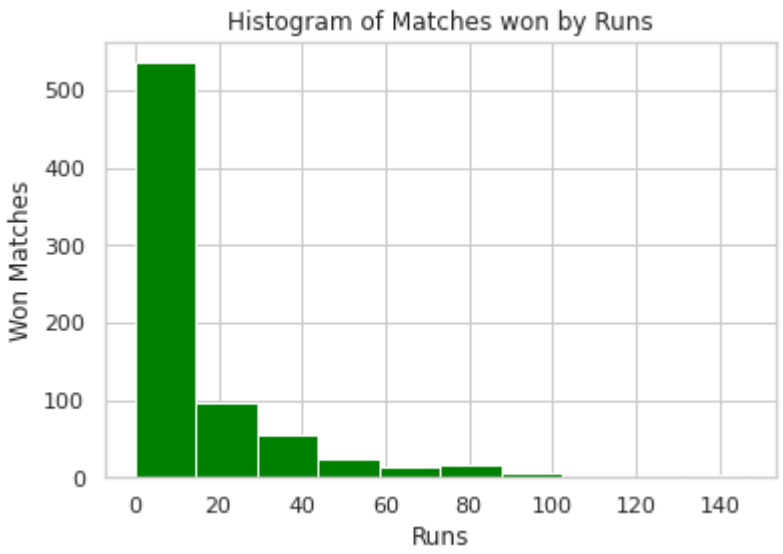


Histograms

Histograms are similar to bar charts; they are a way to display counts of data. A bar graph charts actu
of the bar indicates the number of items in that category. A histogram displays the same categorical v

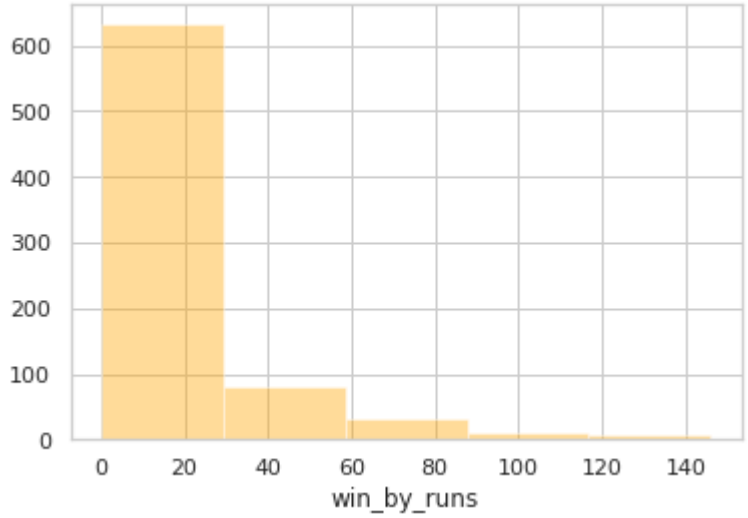
A histogram divides the variable into bins, counts the data points in each bin, and shows the bins on t

```
# Matplotlib histogram for matches won by runs
plt.hist(matches.win_by_runs, color = 'green')

# Add labels
plt.title('Histogram of Matches won by Runs')
plt.xlabel('Runs')
plt.ylabel('Count of Won Matches')
plt.show()
```




```
# seaborn histogram
sns.distplot(matches.win_by_runs, hist=True, kde=False, bins=int(140/28), color = 'orang
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f86b3bdff98>
```

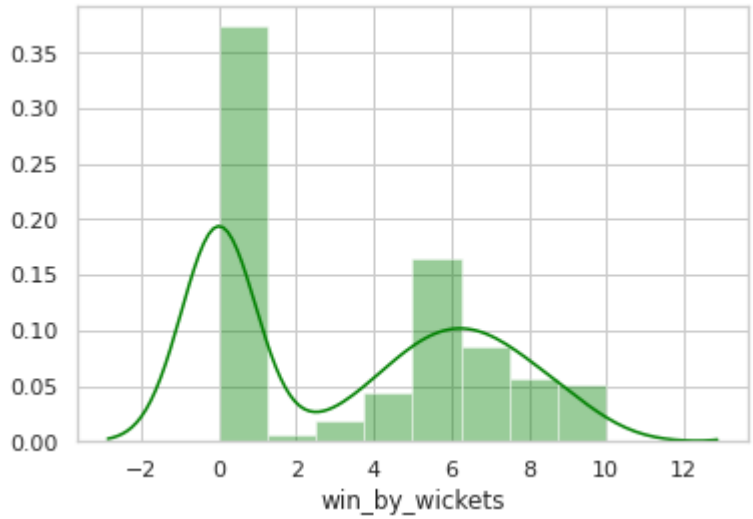


Density Plot

A density plot is a smoothed, continuous version of a histogram estimated from the data. The most c
Kernel Density Estimation (KDE is a non-parametric way to estimate the probability density function o
continuous curve (the kernel) is drawn at every individual data point and all of these curves are then a
density estimation. The kernel most often used is a Gaussian (which produces a Gaussian bell curve

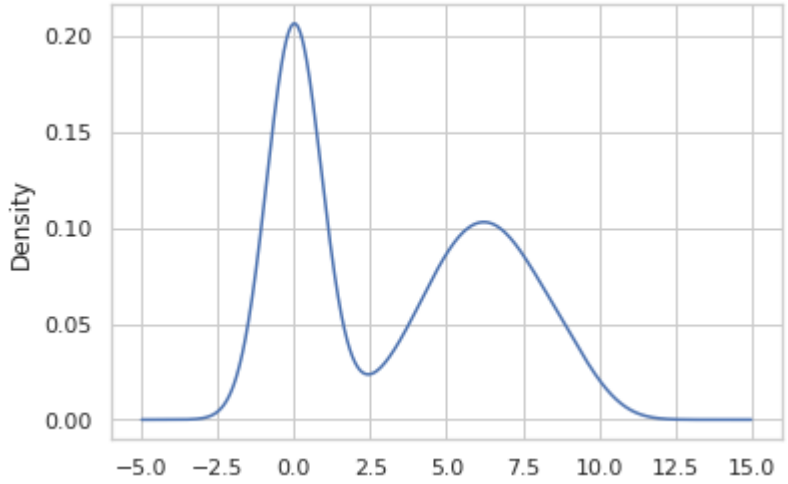
```
1  sns.distplot(matches.win_by_wickets, hist=True, kde=True, color = 'green') # Seaborn
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f86b3cde160>
```




```
1  matches.win_by_wickets.plot(kind='density') # Matplotlib
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f86b3ce7b70>
```

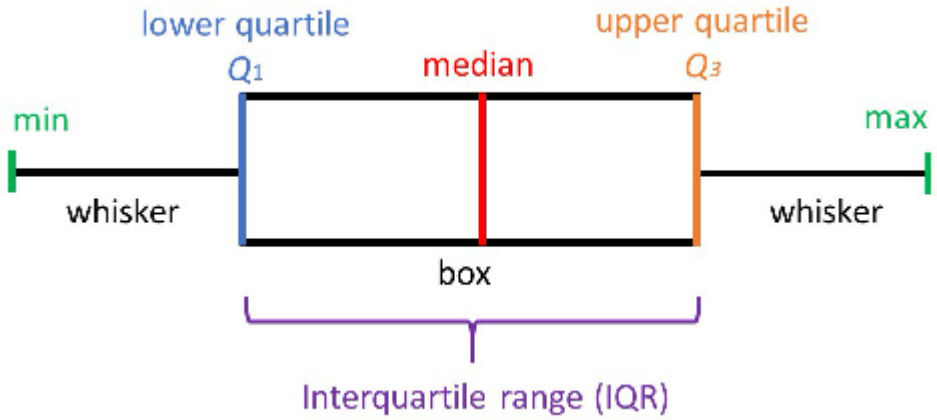


Box-Whisker Plot

Box and whisker plots (or box plots) represents five-number summary of the dataset. The five-numbe

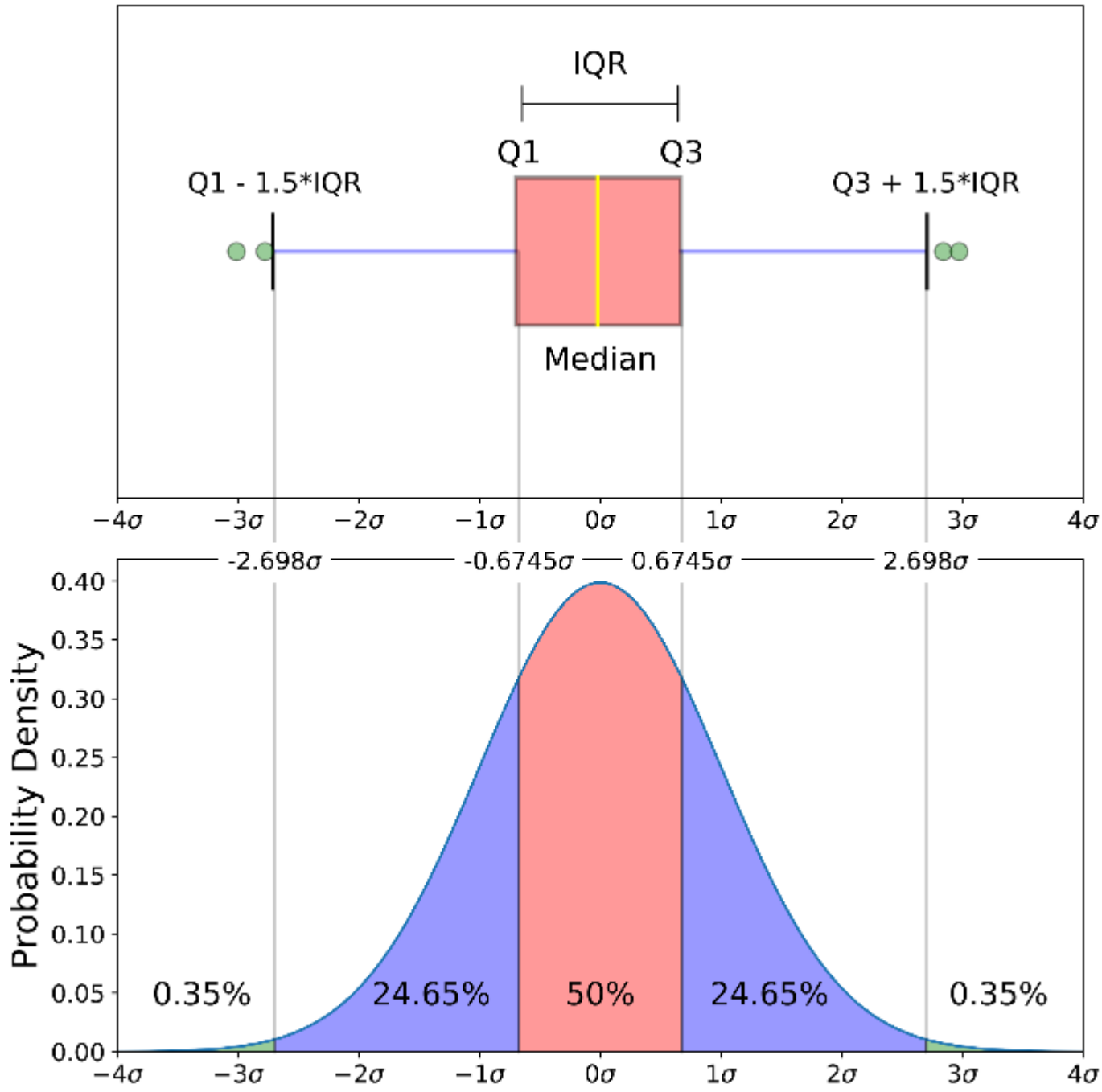
- Minimum
- First quartile (25th percentile)
- Median (50th percentile)
- Third quartile (75th percentile)
- Maximum

The following is a representation of box-and-whisker plot:



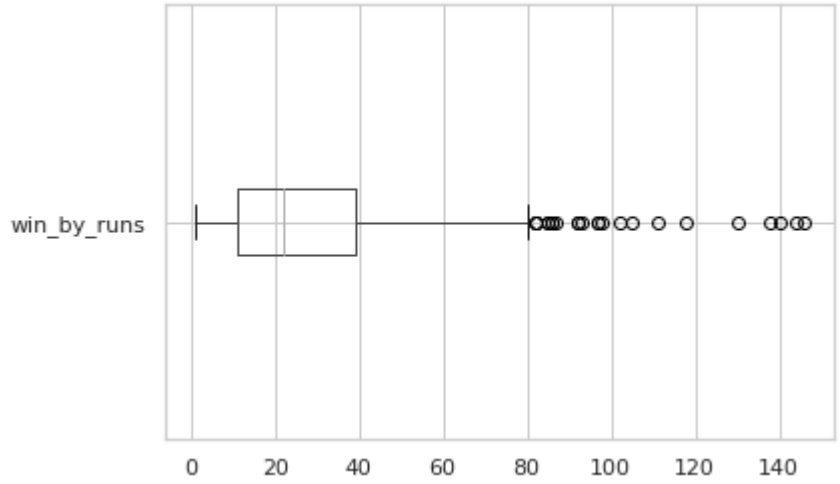
Data sets can sometimes contain outliers that are suspected to be anomalies (perhaps because of da flukes). If outliers are present, the whisker on the appropriate side is drawn to 1.5 * IQR rather than the Small circles or unfilled dots are drawn on the chart to indicate where suspected outliers lie. Filled cir

Box Plot of Normal Distribution:



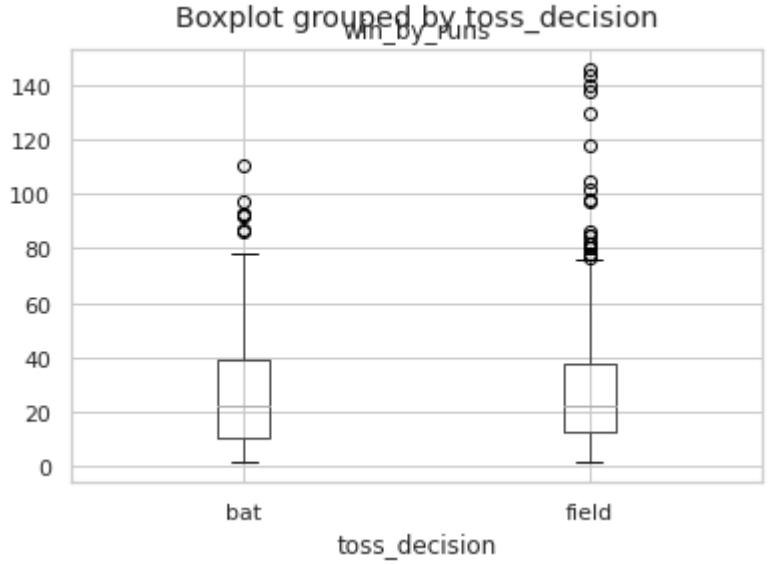

```
1  matches.loc[matches['win_by_runs']>0,'win_by_runs'].to_frame().boxplot(vert=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f86b3605e48>
```



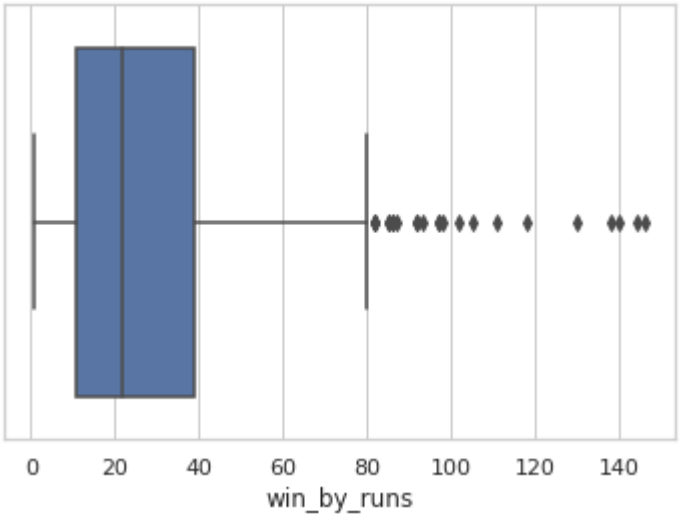

```
1  runs = matches.loc[matches['win_by_runs']>0,:]
2  runs.boxplot(column= 'win_by_runs', by = 'toss_decision')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f86b3580d30>
```



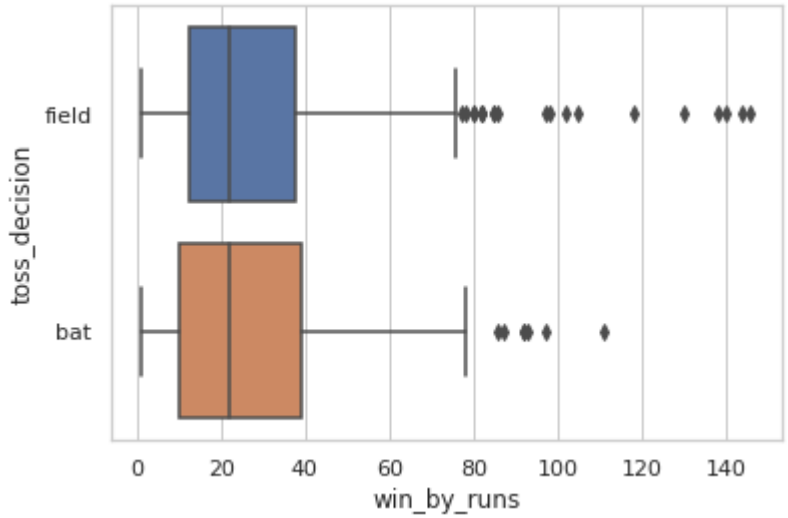

```
1  sns.boxplot(x = 'win_by_runs', data = runs)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f86b35987f0>
```



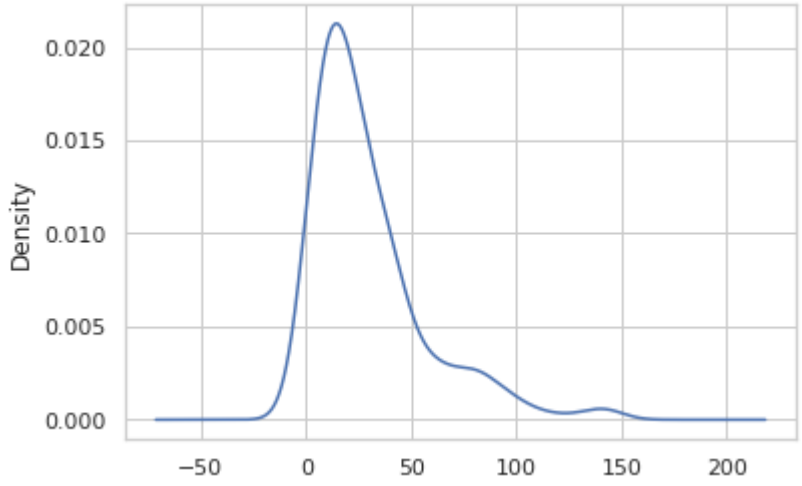

```
1  sns.boxplot(x = 'win_by_runs', y='toss_decision', data = runs)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f86b3300e48>
```



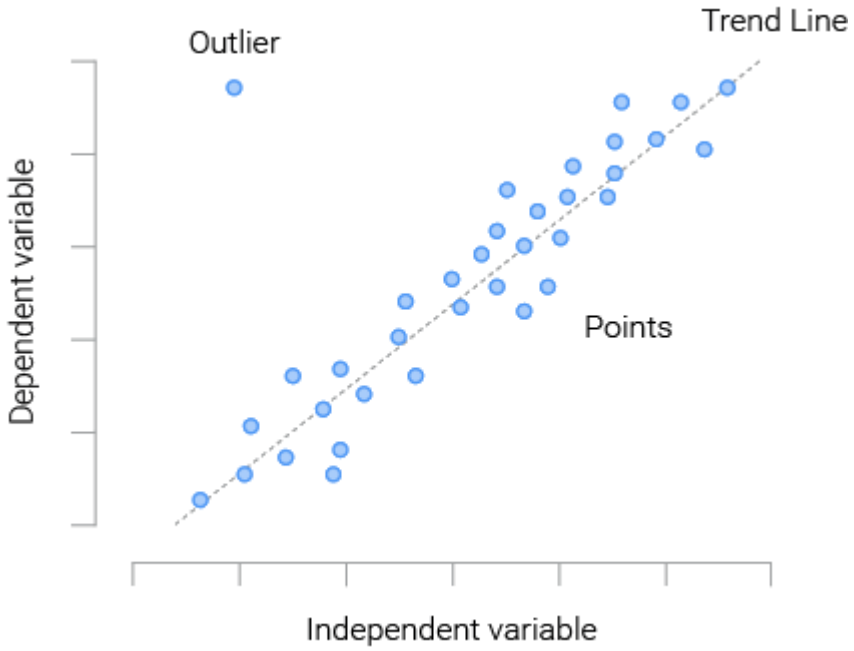

```
1  runs.win_by_runs.plot(kind= 'density') # right-skewed
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f86b320d470>
```



Scatter Plot

Scatter plot Scatter plots help us see relationship between variables or how much one variable is affe the plot, whether there is a positive or negative or no association between the variables. Following is a and Variable B.



We can see that there is a positive association between Variable A and Variable B. i.e. Variable A incre

Some datasets many contain a few data points that doesn't fit the pattern. They're called outliers.

A scatter chart can suggest a linear relationship between 2 variables(i.e. a straight line).

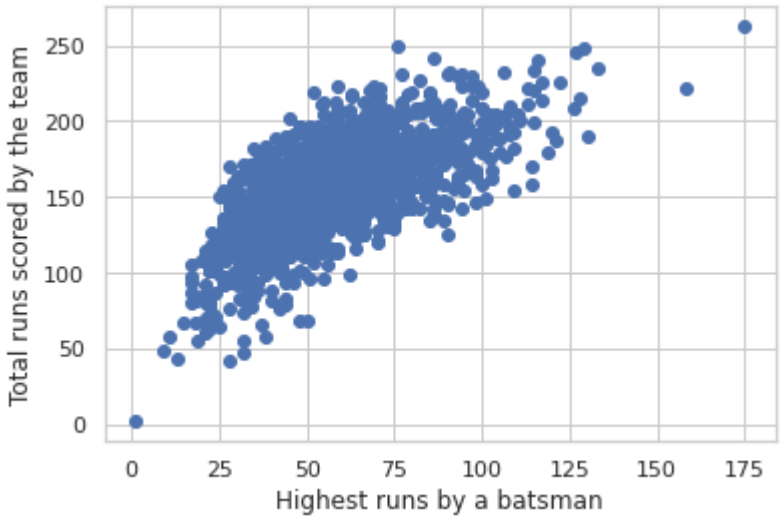
```
1  deliveries.head()
```

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowle
0	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	DA Warner	S Dhawan	TS Mill
1	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	DA Warner	S Dhawan	TS Mill
2	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	3	DA Warner	S Dhawan	TS Mill
3	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	4	DA Warner	S Dhawan	TS Mill
4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	TS Mill

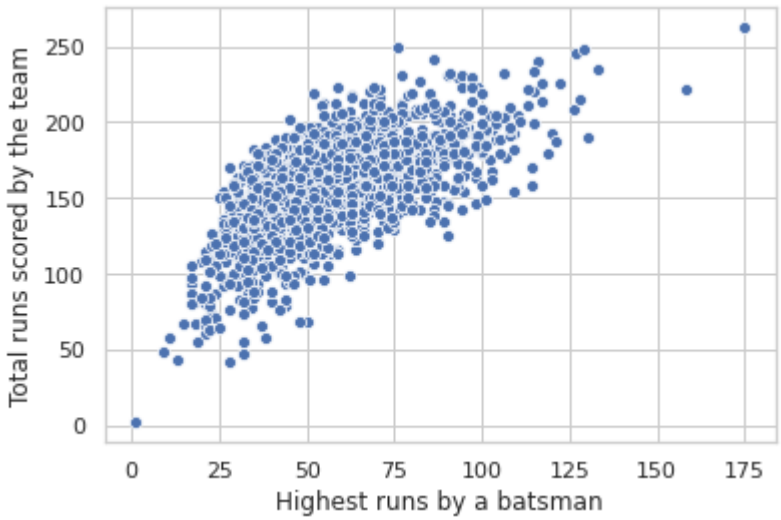
```
highest_score_batsman = deliveries.groupby(['match_id', 'batting_team', 'batsman'])['bat

total_runs = deliveries.groupby(['match_id', 'batting_team'])['total_runs'].sum()

match_total_highest_combined = pd.concat([highest_score_batsman, total_runs], axis=1)
match_total_highest_combined.head()
```

		batsman_runs	total_runs
match_id	**batting_team**		
1	**Royal Challengers Bangalore**	32	172
	Sunrisers Hyderabad	62	207
2	**Mumbai Indians**	38	184
	Rising Pune Supergiant	84	187
3	**Gujarat Lions**	68	183

```
plt.scatter(match_total_highest_combined.batsman_runs, match_total_highest_combined.tota
plt.xlabel("Highest runs by a batsman")
plt.ylabel("Total runs scored by the team")
plt.show()
```




```
sns.scatterplot(match_total_highest_combined.batsman_runs, match_total_highest_combined.
plt.xlabel("Highest runs by a batsman")
plt.ylabel("Total runs scored by the team")
plt.show()
```



Pair Plot

The pair plot technique allows us to visualize distributions of individual numerical features, as well as numerical features. Pair plot is a handy technique because it is very easy to implement and allows us between each of our features by pairing them together.

Seaborn pair plot command will not run if there are NaN values in the data frame.

Output of Pair Plot has two basic types of graphs: histograms for each numerical feature showing the the numerical values plotted against each other.

The histograms are displayed in a diagonal fashion starting in the upper left cell, or the (0, 0)th cell, of right cell, or the (n-1, n-1)th cell, each histogram can be looked by the corresponding labels on either t

The scatter plots can be located in all other cells on the grid. We can identify the individual features p corresponding labels on either the x-axis or the y-axis.

```
1  sns.pairplot(matches.iloc[:,1:]) # not including id column here
```

```
<seaborn.axisgrid.PairGrid at 0x7f86b1162d30>
```