```
!pip install plotly --user
```

```
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (5.15.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly) (8.2.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from plotly) (23.2)
```

Importing libraries

```
import pandas as pd # data processing
import numpy as np # linear algebra
import matplotlib.pyplot as plt
import seaborn as sns
import plotly
import plotly.express as px
import plotly.graph_objs as go
from plotly.offline import init_notebook_mode, plot, iplot
from plotly import tools
from warnings import filterwarnings
filterwarnings('ignore')
```

Importing Data set

```
ball_data = pd.read_csv("IPL Ball-by-Ball 2008-2020.csv")
match_data = pd.read_csv("IPL Matches 2008-2020.csv")
print("Data ready for exploration")
```

```
Data ready for exploration
```

Double-click (or enter) to edit

```
match_data.head()
```

	id	city	date	player_of_match	venue	neutral_venue	team1
0	335982	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	0	Kings XI Punjab
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils
3	335985	Mumbai	2008-	MV Boucher	Wankhede	0	Mumbai Cl

```
ball_data.head()
```

	id	inning	over	ball	batsman	non_striker	bowler	batsman_runs	extra_runs
0	335982	1	6	5	RT Ponting	BB McCullum	AA Noffke	1	0
1	335982	1	6	6	BB McCullum	RT Ponting	AA Noffke	1	0

```
match_data.isnull().sum()
```

```
id                 0
city              13
date               0
player_of_match    4
venue              0
neutral_venue      0
team1              0
team2              0
toss_winner        0
toss_decision      0
winner             4
result             4
result_margin     17
eliminator         4
method           797
umpire1            0
umpire2            0
dtype: int64
```

```
match_data.shape
```

```
(816, 17)
```

```
match_data.columns
```

```
Index(['id', 'city', 'date', 'player_of_match', 'venue', 'neutral_venue',
       'team1', 'team2', 'toss_winner', 'toss_decision', 'winner', 'result',
       'result_margin', 'eliminator', 'method', 'umpire1', 'umpire2'],
      dtype='object')
```

```
print('Total Matches Played:',match_data.shape[0])
print(' \n Venues Played At:',match_data['city'].unique())
print(' \n Teams :',match_data['team1'].unique())
```

```
Total Matches Played: 816

 Venues Played At: ['Bangalore' 'Chandigarh' 'Delhi' 'Mumbai' 'Kolkata' 'Jaipur' 'Hyderabad'
 'Chennai' 'Cape Town' 'Port Elizabeth' 'Durban' 'Centurion' 'East London'
 'Johannesburg' 'Kimberley' 'Bloemfontein' 'Ahmedabad' 'Cuttack' 'Nagpur'
 'Dharamsala' 'Kochi' 'Indore' 'Visakhapatnam' 'Pune' 'Raipur' 'Ranchi'
 'Abu Dhabi' nan 'Rajkot' 'Kanpur' 'Bengaluru' 'Dubai' 'Sharjah']

 Teams : ['Royal Challengers Bangalore' 'Kings XI Punjab' 'Delhi Daredevils'
 'Mumbai Indians' 'Kolkata Knight Riders' 'Rajasthan Royals'
 'Deccan Chargers' 'Chennai Super Kings' 'Kochi Tuskers Kerala'
 'Pune Warriors' 'Sunrisers Hyderabad' 'Gujarat Lions'
 'Rising Pune Supergiants' 'Rising Pune Supergiant' 'Delhi Capitals']
```

1) Number of matches played in various seasons :

```
match_data['Season'] = pd.DatetimeIndex(match_data['date']).year
match_data.head()
```

	id	city	date	player_of_match	venue	neutral_venue	team1	
0	335982	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	0	Kings XI Punjab	
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	
3	335985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Ch

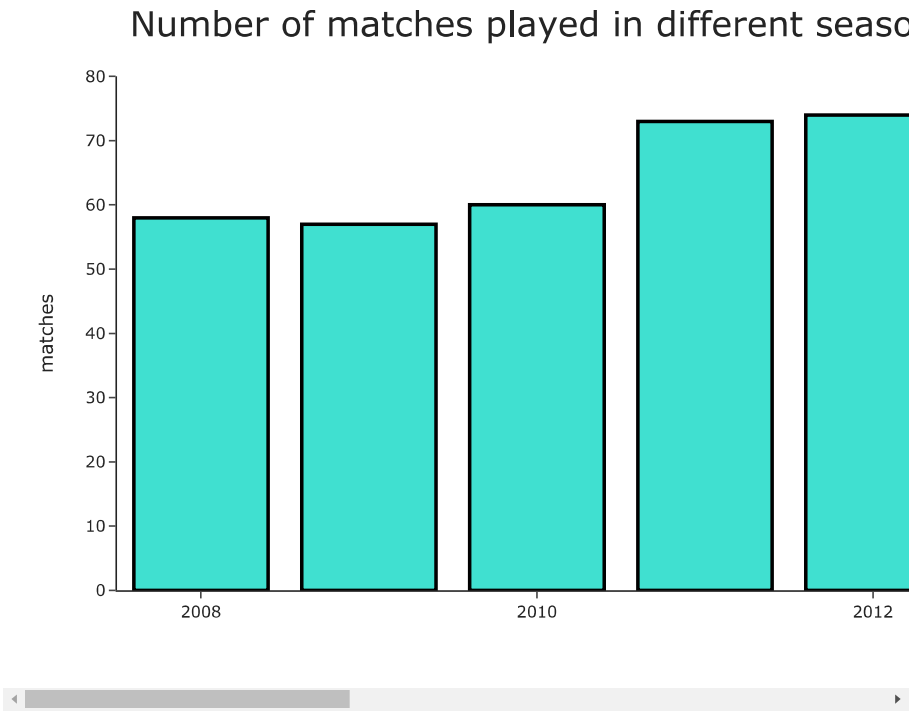
```
import pandas as pd
match_data['Season'] = pd.to_datetime(match_data['date']).dt.year

match_per_season = match_data.groupby(['Season'])['id'].count().reset_index().rename(columns={'id': 'matches'})
match_per_season
```

	Season	matches
0	2008	58
1	2009	57
2	2010	60
3	2011	73
4	2012	74
5	2013	76
6	2014	60
7	2015	59
8	2016	60
9	2017	59
10	2018	60
11	2019	60
12	2020	60

```
colors = ['turquoise',] * 13
colors[5] = 'crimson'

fig=px.bar(data_frame=match_per_season,x=match_per_season.Season,y=match_per_season.matches,labels=dict(x="Season",y="Count"),)
fig.update_layout(title="Number of matches played in different seasons ",
                  titlefont={'size': 26},template='simple_white'
                  )
fig.update_traces(marker_line_color='black',
                  marker_line_width=2.5, opacity=1,marker_color=colors)
fig.show()
```



Each season, almost 60 matches were played. However, we see a spike in the number of matches from 2011 to 2013. This is because two new franchises, the Pune Warriors and Kochi Tuskers Kerala, were introduced, increasing the number of teams to 10.

2) Total number of runs scored across seasons

```
season_data=match_data[['id','date']].merge(ball_data, left_on = 'id', right_on = 'id', how = 'left').drop('id', axis = 1)
season_data.head()
```

	date	inning	over	ball	batsman	non_striker	bowler	batsman_runs	extra_runs	tc
0	2008-04-18	1	6	5	RT Ponting	BB McCullum	AA Noffke	1	0	
1	2008-04-18	1	6	6	BB McCullum	RT Ponting	AA Noffke	1	0	

```
season_data = ball_data
Season = season_data.groupby(['Season'])['total_runs'].sum().reset_index()
p=season.set_index('Season')
fig = px.line(p, x=p.index, y="total_runs")
fig.update_layout(title="Total Runs Across the Seasons ",
titlefont={'size': 26},template='simple_white')
fig.show()
```

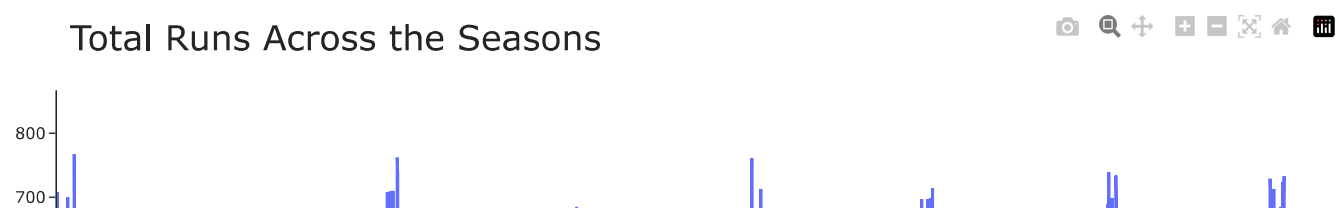
```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-58-4b66579a4aff> in <cell line: 2>()
      1 season_data = ball_data
----> 2 Season = season_data.groupby(['Season'])['total_runs'].sum().reset_index()
      3 p=season.set_index('Season')
      4 fig = px.line(p, x=p.index, y="total_runs")
      5 fig.update_layout(title="Total Runs Across the Seasons ",

                                  2 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/groupby/grouper.py in get_grouper(obj, key, axis, level, sort, observed, mutated,
validate, dropna)
    886                 in_axis, level, gpr = False, gpr, None
    887             else:
--> 888                 raise KeyError(gpr)
    889         elif isinstance(gpr, Grouper) and gpr.key is not None:
    890             # Add key to exclusions

KeyError: 'Season'
```

SEARCH STACK OVERFLOW

```
season=season_data.groupby(['date'])['total_runs'].sum().reset_index()
p=season.set_index('date')
fig = px.line(p, x=p.index, y="total_runs")
fig.update_layout(title="Total Runs Across the Seasons ",
                  titlefont={'size': 26},template='simple_white'
                  )
fig.show()
```



3) Runs scored per match across seasons :

```
runs_per_season=pd.concat([match_per_season,season.iloc[:,1]],axis=1)
runs_per_season['Runs scored per match']=runs_per_season['total_runs']/runs_per_season['matches']
runs_per_season.set_index('Season',inplace=True)
runs_per_season.style.background_gradient(cmap='PuBu',subset=['Runs scored per match'])
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/IPython/core/formatters.py in __call__(self, obj)
    339                 pass
    340             else:
--> 341                 return printer(obj)
    342             # Finally look for special method names
    343             method = get_real_method(obj, self.print_method)

7 frames
/usr/local/lib/python3.10/dist-packages/pandas/io/formats/style.py in _update_ctx(self, attrs)
   1572         """
   1573         if not self.index.is_unique or not self.columns.is_unique:
-> 1574             raise KeyError(
   1575                 "`Styler.apply` and `.applymap` are not compatible "
   1576                 "with non-unique index or columns."

KeyError: '`Styler.apply` and `.applymap` are not compatible with non-unique index or columns.'
```

SEARCH STACK OVERFLOW

```
<pandas.io.formats.style.Styler at 0x7f0cb76a35b0>
```