GRIP: The Sparks Foundation

Data Science and Business Analytics Intern

Author: Sudeshna Saha

In [1]:

```
#import all necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
#load thye data
matches=pd.read_csv("matches.csv")
```

In [3]:

```
matches.head()
```

Out[3]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	d
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	
2	3	2017	Rajkot	2017-04-07	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	
4	5	2017	Bangalore	2017-04-08	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	

In [4]:

```
matches.shape
```

Out[4]:

```
(756, 18)
```

In [5]:

```
#checking the basic information
matches.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   id               756 non-null    int64 
 1   season           756 non-null    int64 
 2   city             749 non-null    object
 3   date             756 non-null    object
 4   team1            756 non-null    object
 5   team2            756 non-null    object
 6   toss_winner      756 non-null    object
 7   toss_decision    756 non-null    object
 8   result           756 non-null    object
 9   dl_applied       756 non-null    int64 
 10  winner           752 non-null    object
 11  win_by_runs      756 non-null    int64 
 12  win_by_wickets   756 non-null    int64 
 13  player_of_match  752 non-null    object
 14  venue            756 non-null    object
 15  umpire1          754 non-null    object
 16  umpire2          754 non-null    object
 17  umpire3          119 non-null    object
dtypes: int64(5), object(13)
memory usage: 106.4+ KB
```

In [6]:

```
#check the summary of the data
matches.describe()
```

Out[6]:

	id	season	dl_applied	win_by_runs	win_by_wickets
count	756.000000	756.000000	756.000000	756.000000	756.000000
mean	1792.178571	2013.444444	0.025132	13.283069	3.350529
std	3464.478148	3.366895	0.156630	23.471144	3.387963
min	1.000000	2008.000000	0.000000	0.000000	0.000000
25%	189.750000	2011.000000	0.000000	0.000000	0.000000
50%	378.500000	2013.000000	0.000000	0.000000	4.000000
75%	567.250000	2016.000000	0.000000	19.000000	6.000000
max	11415.000000	2019.000000	1.000000	146.000000	10.000000

In [7]:

```
#null values
matches.isnull().sum()
```

Out[7]:

```
id                   0
season               0
city                 7
date                 0
team1                0
team2                0
toss_winner          0
toss_decision        0
result               0
dl_applied           0
winner               4
win_by_runs          0
win_by_wickets       0
player_of_match      4
venue                0
umpire1              2
umpire2              2
umpire3            637
dtype: int64
```

In [8]:

```
#dropping the column whcih has so many null values
col=["umpire3"]
matches.drop(columns=col,axis=1,inplace=True)
```

In [9]:

```
np.unique(matches["season"])
```

Out[9]:

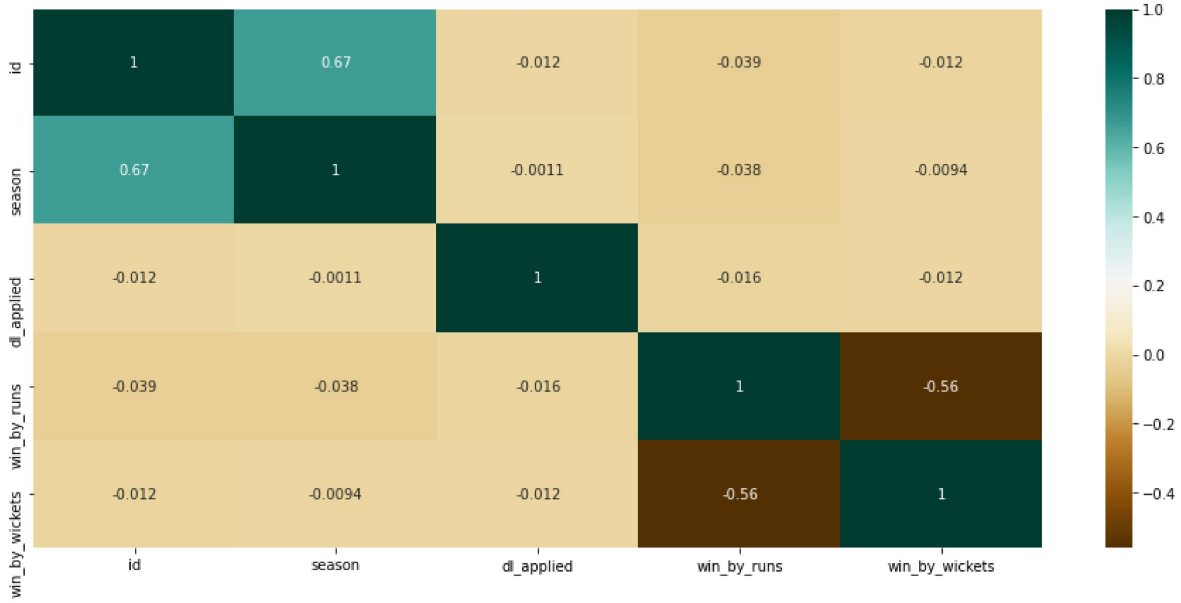
```
array([2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018,
       2019], dtype=int64)
```

In [10]:

```
plt.figure(figsize=(16,7))
sns.heatmap(matches.corr(),cmap="BrBG",annot=True)
```

Out[10]:

<AxesSubplot:>



In [11]:

```
#Load the another data
deliveries=pd.read_csv("deliveries.csv")
```

In [12]:

```
deliveries.head()
```

Out[12]:

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

5 rows × 21 columns

In [13]:

```
deliveries.shape
```

Out[13]:

```
(179078, 21)
```

In [14]:

```
#basic information of the data
deliveries.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 179078 entries, 0 to 179077
Data columns (total 21 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   match_id          179078 non-null  int64 
 1   inning            179078 non-null  int64 
 2   batting_team      179078 non-null  object
 3   bowling_team      179078 non-null  object
 4   over              179078 non-null  int64 
 5   ball              179078 non-null  int64 
 6   batsman           179078 non-null  object
 7   non_striker       179078 non-null  object
 8   bowler            179078 non-null  object
 9   is_super_over     179078 non-null  int64 
 10  wide_runs         179078 non-null  int64 
 11  bye_runs          179078 non-null  int64 
 12  legbye_runs       179078 non-null  int64 
 13  noball_runs       179078 non-null  int64 
 14  penalty_runs      179078 non-null  int64 
 15  batsman_runs      179078 non-null  int64 
 16  extra_runs        179078 non-null  int64 
 17  total_runs        179078 non-null  int64 
 18  player_dismissed  8834 non-null    object
 19  dismissal_kind    8834 non-null    object
 20  fielder           6448 non-null    object
dtypes: int64(13), object(8)
memory usage: 28.7+ MB
```

In [15]:

```
deliveries.describe()
```

Out[15]:

	match_id	inning	over	ball	is_super_over	wide_ru
count	179078.000000	179078.000000	179078.000000	179078.000000	179078.000000	179078.000(
mean	1802.252957	1.482952	10.162488	3.615587	0.000452	0.036
std	3472.322805	0.502074	5.677684	1.806966	0.021263	0.251
min	1.000000	1.000000	1.000000	1.000000	0.000000	0.000(
25%	190.000000	1.000000	5.000000	2.000000	0.000000	0.000(
50%	379.000000	1.000000	10.000000	4.000000	0.000000	0.000(
75%	567.000000	2.000000	15.000000	5.000000	0.000000	0.000(
max	11415.000000	5.000000	20.000000	9.000000	1.000000	5.000(

In [16]:

```
deliveries.isnull().sum()
```

Out[16]:

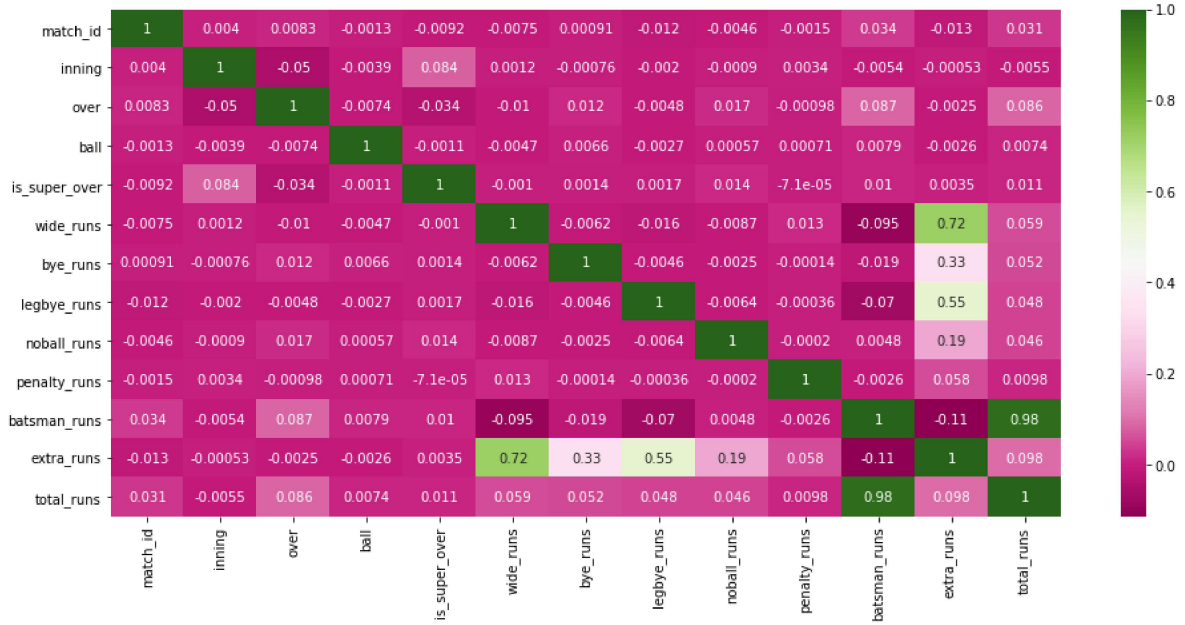
```
match_id                0
inning                  0
batting_team            0
bowling_team            0
over                    0
ball                    0
batsman                 0
non_striker             0
bowler                  0
is_super_over           0
wide_runs               0
bye_runs                0
legbye_runs             0
noball_runs             0
penalty_runs            0
batsman_runs            0
extra_runs              0
total_runs              0
player_dismissed    170244
dismissal_kind      170244
fielder             172630
dtype: int64
```

In [17]:

```
plt.figure(figsize=(16,7))
sns.heatmap(deliveries.corr(),cmap="PiYG",annot=True)
```

Out[17]:

```
<AxesSubplot:>
```



In [ ]:

```

```