

LA LIGA SCORES - BUSINESS REPORT GRADED PROJECT

Main data frame

```
df=pd.read_csv('C:\\Users\\Intel\\Downloads\\Laliga_scores.csv',header=1)
df
```

	Pos	Team	Seasons	Points	GamesPlayed	GamesWon	GamesDrawn	GamesLost	GoalsFor	GoalsAgainst	Champion	Runner-up	Third	Fourth	Fifth
0	1	Real Madrid	86	4385	2762	1647	552	563	5947	3140	33	23	8	8	3
1	2	Barcelona	86	4262	2762	1581	573	608	5900	3114	25	25	12	12	4
2	3	Atletico Madrid	80	3442	2614	1241	598	775	4534	3309	10	8	16	9	7
3	4	Valencia	82	3386	2664	1187	616	861	4398	3469	6	6	10	11	10
4	5	Athletic Bilbao	86	3368	2762	1209	633	920	4631	3700	8	7	10	5	8
...
56	57	Xerez	1	34	38	8	10	20	38	66	-	-	-	-	-
57	58	Condal	1	22	30	7	8	15	37	57	-	-	-	-	-
58	59	Atletico Tetuan	1	19	30	7	5	18	51	85	-	-	-	-	-
59	60	Cultural Leonesa	1	14	30	5	4	21	34	65	-	-	-	-	-
60	61	Girona	1	-	-	-	-	-	-	-	-	-	-	-	-

61 rows × 18 columns

There are many empty cells in the dataframe.

Q1) Read the data set and replace dashes with 0 to make sure you can perform arithmetic operations on the data. And check the distribution for the 'Best Position' and report the top position.

```
df=pd.read_csv('C:\\Users\\Intel\\Downloads\\Laliga_scores.csv',header=1)
df=df.replace(to_replace='-',
              value=0)
```

df

	Pos	Team	Seasons	Points	GamesPlayed	GamesWon	GamesDrawn	GamesLost	GoalsFor	GoalsAgainst	...	Runner-up	Third	Fourth	Fifth	Sixth	D
0	1	Real Madrid	86	4385	2762	1647	552	563	5947	3140	...	23	8	8	3	4	
1	2	Barcelona	86	4262	2762	1581	573	608	5900	3114	...	25	12	12	4	6	
2	3	Atletico Madrid	80	3442	2614	1241	598	775	4534	3309	...	8	16	9	7	6	
3	4	Valencia	82	3386	2664	1187	616	861	4398	3469	...	6	10	11	10	7	
4	5	Athletic Bilbao	86	3368	2762	1209	633	920	4631	3700	...	7	10	5	8	10	
...
56	57	Xerez	1	34	38	8	10	20	38	66	...	0	0	0	0	0	
57	58	Condal	1	22	30	7	8	15	37	57	...	0	0	0	0	0	
58	59	Atletico Tetuan	1	19	30	7	5	18	51	85	...	0	0	0	0	0	
59	60	Cultural Leonesa	1	14	30	5	4	21	34	65	...	0	0	0	0	0	
60	61	Girona	1	0	0	0	0	0	0	0	...	0	0	0	0	0	

61 rows × 21 columns

df.describe()

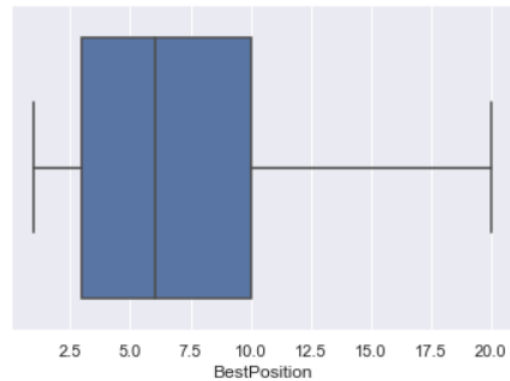
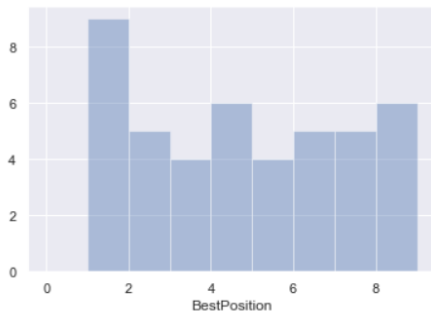
	Pos	Seasons	Points	GamesPlayed	GamesWon	GoalsFor	GoalsAgainst	Debut	BestPosition	Goal_diff_count	Winning Percent
count	61.000000	61.000000	61.000000	61.000000	61.000000	61.000000	61.000000	61.000000	61.000000	61.000000	61.000000
mean	31.000000	24.000000	901.426230	796.819672	303.967213	1140.344262	1140.229508	1958.918033	7.081967	0.114754	30.850613
std	17.752934	26.827225	1134.899121	876.282765	406.991030	1506.740211	1163.710766	27.484114	5.276663	598.095814	8.742574
min	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1929.000000	1.000000	-525.000000	0.000000
25%	16.000000	4.000000	96.000000	114.000000	34.000000	153.000000	221.000000	1935.000000	3.000000	-269.000000	27.192982
50%	31.000000	12.000000	375.000000	423.000000	123.000000	430.000000	632.000000	1951.000000	6.000000	-112.000000	30.281690
75%	46.000000	38.000000	1351.000000	1318.000000	426.000000	1642.000000	1951.000000	1978.000000	10.000000	-34.000000	33.485401
max	61.000000	86.000000	4385.000000	2762.000000	1647.000000	5947.000000	3889.000000	2017.000000	20.000000	2807.000000	59.630702

By using replace function '-' are replaced by 0(zeroes).

```
df['BestPosition']=pd.to_numeric(df['BestPosition'])
best_pos=df['BestPosition']
best_pos.describe()
```

```
count    61.000000
mean      7.081967
std       5.276663
min       1.000000
25%       3.000000
50%       6.000000
75%      10.000000
max      20.000000
Name: BestPosition, dtype: float64
```

```
import seaborn as sns
sns.set(color_codes=True)
sns.distplot(best_pos,kde=False,bins=np.arange(10));
```



```
df[df['BestPosition']==1]['Team']
```

```
0    Real Madrid
1    Barcelona
2    Atletico Madrid
3    Valencia
4    Athletic Bilbao
5    Sevilla
7    Real Sociedad
9    Real Betis
10   Deportivo La Coruna
```

From the above histogram we can conclude that 1st is the best position because 1st is the most secured best position(9 teams) and almost 50 percent of the teams have their best positions within 1-6.

Q2. Print all the teams which have started playing between 1930-1980 using the “Debut” column (Include year 1930 only).

Debut	Be:
1929	
1929	
1929	
1931-32	
1929	
...	
2009-10	
1956-57	
1951-52	
1955-56	
2017-18	

Some values in the debut were in the form(1932-1933),in that case the 1st year is taken.

```
a=df['Debut']
b=[]
for i in a:
    b.append(i[0:4])
df['Debut']=b
df['Debut']=pd.to_numeric(df['Debut'])
```

A for loop is created to restrict all the elements in the series to 4 numbers.

A new dataframe is created for the given condition (‘Debut’ greater than or equal to 1930 and less than 1980).

Ashwath J

```
new_df=df[(df['Debut']>=1930) & (df['Debut'] <1980)].reset_index()  
new_df
```

	index	Pos	Team	Seasons	Points	GamesPlayed	GamesWon	GamesDrawn	GamesLost	GoalsFor	GoalsAgainst	Champion	Runner-up	Third	Fourth
0	3	4	Valencia	82	3386	2664	1187	616	861	4398	3469	6	6	10	11
1	5	6	Sevilla	73	2819	2408	990	531	887	3680	3373	1	4	4	5
2	8	9	Zaragoza	58	2109	1986	698	522	766	2683	2847	0	1	4	5
3	9	10	Real Betis	51	1884	1728	606	440	682	2159	2492	1	0	2	3
4	10	11	Deportivo La Coruna	45	1814	1530	563	392	575	2052	2188	1	5	4	1
5	11	12	Celta Vigo	51	1789	1698	586	389	723	2278	2624	0	0	0	2
6	12	13	Valladolid	42	1471	1466	463	384	619	1767	2180	0	0	0	1
7	14	15	Sporting Gijon	43	1389	1458	471	358	629	1753	2152	0	1	1	2
8	15	16	Osasuna	37	1351	1318	426	327	565	1500	1834	0	0	0	2
9	16	17	Malaga	36	1314	1255	390	330	535	1421	1763	0	0	0	1
10	17	18	Oviedo	38	1174	1192	408	292	492	1642	1951	0	0	3	2
11	18	19	Mallorca	27	1148	988	333	256	399	1182	1371	0	0	2	0
12	19	20	Las Palmas	33	1020	1096	367	242	487	1347	1746	0	1	1	1
13	21	22	Granada	23	667	742	218	175	349	819	1157	0	0	0	0
14	22	23	Rayo Vallecano	17	662	652	189	148	305	760	1088	0	0	0	0
15	23	24	Elche	21	606	678	203	180	295	750	1022	0	0	0	0

The above data frame is for the year 1930-1980.

There are 37 teams within the given range of debut years.

Q3) Print the list of teams which came Top 5 in terms of points

```
df['Points']=pd.to_numeric(df['Points'])
dfq3=df.sort_values(by='Points',ascending=False).head(5)
dfq3
```

	Pos	Team	Seasons	Points	GamesPlayed	GamesWon	GamesDrawn	GamesLost	GoalsFor	GoalsAgainst	Champion	Runner-up	Third	Fourth	Fifth	S
0	1	Real Madrid	86	4385	2762	1647	552	563	5947	3140	33	23	8	8	3	
1	2	Barcelona	86	4262	2762	1581	573	608	5900	3114	25	25	12	12	4	
2	3	Atletico Madrid	80	3442	2614	1241	598	775	4534	3309	10	8	16	9	7	
3	4	Valencia	82	3386	2664	1187	616	861	4398	3469	6	6	10	11	10	
4	5	Athletic Bilbao	86	3368	2762	1209	633	920	4631	3700	8	7	10	5	8	

A new dataframe is created,a sort_values function is used to sort values in the 'Points' column.

Head function is used to select the required number of rows from the top.

```
dfq3[['Team','Points']]
```

	Team	Points
0	Real Madrid	4385
1	Barcelona	4262
2	Atletico Madrid	3442
3	Valencia	3386
4	Athletic Bilbao	3368

Q4. Write a function with the name “Goal_diff_count” which should return all the teams with their Goal Differences. Goal_diff_count = GoalsFor - GoalsAgainst.

A function is created with 2 arguments.

The above function is called with apply method. The output is stored in a new column.

```
df['GoalsFor']=pd.to_numeric(df['GoalsFor'])
df['GoalsAgainst']=pd.to_numeric(df['GoalsAgainst'])

def Goal_diff_count(a,b):
    return a-b
df['Goal_diff_count'] = df.apply(lambda x: Goal_diff_count(x['GoalsFor'], x['GoalsAgainst']), axis=1)
df
```

```
df['Goal_diff_count'] = df.apply(lambda x: Goal_diff_count(x['GoalsFor'], x['GoalsAgainst']), axis=1)
df
```

	GamesWon	GamesDrawn	GamesLost	GoalsFor	GoalsAgainst	Champion	Runner-up	Third	Fourth	Fifth	Sixth	Debut	BestPosition	NewCol	Goal_diff_count
	1647	552	563	5947	3140	33	23	8	8	3	4	1929	1	2807	2807
	1581	573	608	5900	3114	25	25	12	12	4	6	1929	1	2786	2786
	1241	598	775	4534	3309	10	8	16	9	7	6	1929	1	1225	1225
	1187	616	861	4398	3469	6	6	10	11	10	7	1931	1	929	929
	1209	633	920	4631	3700	8	7	10	5	8	10	1929	1	931	931

	8	10	20	38	66	0	0	0	0	0	0	2009	20	-28	-28
	7	8	15	37	57	0	0	0	0	0	0	1956	16	-20	-20
	7	5	18	51	85	0	0	0	0	0	0	1951	16	-34	-34
	5	4	21	34	65	0	0	0	0	0	0	1955	15	-31	-31

```
df[['Team','Goal_diff_count']]
```

	Team	Goal_diff_count
0	Real Madrid	2807
1	Barcelona	2786
2	Atletico Madrid	1225
3	Valencia	929
4	Athletic Bilbao	931
...
56	Xerez	-28
57	Condal	-20
58	Atletico Tetuan	-34
59	Cultural Leonesa	-31
60	Girona	0

61 rows × 2 columns

Q5) Using the same function, find the team which has a maximum and minimum goal difference.

Maximum Goal Difference - Real Madrid

Minimum Goal Difference - Racing Santander

```
a=df['Goal_diff_count'].max()  
a
```

2807

```
b=df['Goal_diff_count'].min()  
b
```

-525

```
df[df['Goal_diff_count']==a]['Team']
```

```
0    Real Madrid  
Name: Team, dtype: object
```

```
df[df['Goal_diff_count']==b]['Team']
```

```
13    Racing Santander  
Name: Team, dtype: object
```

Q6. Create a new column with the name “Winning Percent” and append it to the data set Percentage of Winning = (GamesWon / GamesPlayed)*100 If there are any numerical error, replace it with 0%

A function is created with 2 args a and b.

A if statement is used inside the function to avoid zero division error.

A new column is created with the apply method and the above created function.

```
df['GamesWon']=pd.to_numeric(df['GamesWon'])
df['GamesPlayed']=pd.to_numeric(df['GamesPlayed'])
def percent_of_winning(a,b):
    if b!=0:
        return (a/b)*100
    else:
        return 0
df['Winning Percent']=df.apply(lambda x:percent_of_winning(x['GamesWon'],x['GamesPlayed']),axis=1)
df
```

GamesWon	GamesDrawn	GamesLost	GoalsFor	GoalsAgainst	...	Runner-up	Third	Fourth	Fifth	Sixth	Debut	BestPosition	NewCol	Goal_diff_count	Winning Percent
1647	552	563	5947	3140	...	23	8	8	3	4	1929	1	2807	2807	59.630702
1581	573	608	5900	3114	...	25	12	12	4	6	1929	1	2786	2786	57.241130
1241	598	775	4534	3309	...	8	16	9	7	6	1929	1	1225	1225	47.475134
1187	616	861	4398	3469	...	6	10	11	10	7	1931	1	929	929	44.557057
1209	633	920	4631	3700	...	7	10	5	8	10	1929	1	931	931	43.772629
...
8	10	20	38	66	...	0	0	0	0	0	2009	20	-28	-28	21.052632
7	8	15	37	57	...	0	0	0	0	0	1956	16	-20	-20	23.333333
7	5	18	51	85	...	0	0	0	0	0	1951	16	-34	-34	23.333333
5	4	21	34	65	...	0	0	0	0	0	1955	15	-31	-31	16.666667
0	0	0	0	0	...	0	0	0	0	0	2017	9	0	0	0.000000

7. Print the top 5 teams which have the highest Winning percentage.

Question 7

```
df.sort_values(by='Winning Percent',ascending=False).head(5)
```

	Pos	Team	Seasons	Points	GamesPlayed	GamesWon	GamesDrawn	GamesLost	GoalsFor	GoalsAgainst	...	Runner-up	Third	Fourth	Fifth	Sixth	De
0	1	Real Madrid	86	4385	2762	1647	552	563	5947	3140	...	23	8	8	3	4	1
1	2	Barcelona	86	4262	2762	1581	573	608	5900	3114	...	25	12	12	4	6	1
2	3	Atletico Madrid	80	3442	2614	1241	598	775	4534	3309	...	8	16	9	7	6	1
3	4	Valencia	82	3386	2664	1187	616	861	4398	3469	...	6	10	11	10	7	1
4	5	Athletic Bilbao	86	3368	2762	1209	633	920	4631	3700	...	7	10	5	8	10	1

5 rows × 21 columns

```
dfq7[['Team','Winning Percent']]
```

	Team	Winning Percent
0	Real Madrid	59.630702
1	Barcelona	57.241130
2	Atletico Madrid	47.475134
3	Valencia	44.557057
4	Athletic Bilbao	43.772629

Q8) Group teams based on their “Best position” and print the sum of their points for all positions.

```
In [199]: z=df.groupby('BestPosition')['Points'].sum()  
dfq8=pd.DataFrame(z)  
dfq8
```

Out[199]:

Points	
BestPosition	
1	27933
2	6904
3	5221
4	6563
5	1884
6	2113
7	1186
8	1134
9	96
10	450
11	445
12	511
14	71
15	14
16	81
17	266
19	81
20	34