College Football Bowl Games

Dataset link: https://www.kaggle.com/code/mattop/college-football-bowl-games-eda-wordcloud/notebook (https://www.kaggle.com/code/mattop/college-football-bowl-games-eda-wordcloud/notebook)

Prepared by: Appurva Talashilkar

Problem Statement: Overall performance of each team from the year(1902-2022)

Tabular data includes:

year: Year of game. date: Date of game. day: Day of the week. winner_tie: Winner of game/tied. winner_rank: Rank of winning team. winner_points: Total points scored by winning team. loser_tie: Loser of game/tied. loser_rank: Rank of losing team. loser_points: Total points scored by losing team. attendance: Total number of people at bowl game mvp: Most valuable player, position, team, name sponsor: Sponsor of game.

In [16]:

```
#LOADING LIBRARIES
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

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

In [17]:

```
df=pd.read_csv("collegefootballbowl.csv")
df
```

Out[17]:

	id	year	date	day	winner_tie	winner_rank	winner_points	loser_tie	loser_r
0	1	2021	12/29/2021	Wed	Oklahoma	14	47	Oregon	
1	2	2020	12/29/2020	Tue	Texas	20	55	Colorado	I
2	3	2019	12/31/2019	Tue	Texas	NaN	38	Utah	
3	4	2018	12/28/2018	Fri	Washington State	12	28	Iowa State	
4	5	2017	12/28/2017	Thu	Texas Christian	13	39	Stanford	
...	...	...	...	...	...	...	...	...	
1522	1523	2004	12/30/2004	Thu	Northern Illinois	NaN	34	Troy	I
1523	1524	2003	12/30/2003	Tue	Fresno State	NaN	17	UCLA	I
1524	1525	2002	12/31/2002	Tue	Fresno State	NaN	30	Georgia Tech	I
1525	1526	2001	12/31/2001	Mon	Michigan State	NaN	44	Fresno State	

	id	year	date	day	winner_tie	winner_rank	winner_points	loser_tie	loser_r
1526	1527	2000	12/31/2000	Sun	Air Force	NaN	37	Fresno State	I

1527 rows × 14 columns

EDA(Exploratory Data Analysis

In [18]:

```
df.head()
```

Out[18]:

	id	year	date	day	winner_tie	winner_rank	winner_points	loser_tie	loser_rank	los
0	1	2021	12/29/2021	Wed	Oklahoma	14	47	Oregon	15	
1	2	2020	12/29/2020	Tue	Texas	20	55	Colorado	NaN	
2	3	2019	12/31/2019	Tue	Texas	NaN	38	Utah	12	
3	4	2018	12/28/2018	Fri	Washington State	12	28	Iowa State	25	
4	5	2017	12/28/2017	Thu	Texas Christian	13	39	Stanford	15	

In [19]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1527 entries, 0 to 1526
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             1527 non-null   int64  
 1   year           1527 non-null   int64  
 2   date           1527 non-null   object 
 3   day            1527 non-null   object 
 4   winner_tie     1527 non-null   object 
 5   winner_rank    754 non-null    object 
 6   winner_points  1527 non-null   int64  
 7   loser_tie      1527 non-null   object 
 8   loser_rank     671 non-null    object 
 9   loser_points   1527 non-null   int64  
 10  attendance     1518 non-null   float64
 11  mvp            1358 non-null   object 
 12  sponsor        820 non-null    object 
 13  bowl_name      1527 non-null   object 
dtypes: float64(1), int64(4), object(9)
memory usage: 167.1+ KB
```

In [20]:

```
df.describe()
```

Out[20]:

	id	year	winner_points	loser_points	attendance
count	1527.000000	1527.000000	1527.000000	1527.000000	1518.000000
mean	764.000000	1991.286182	30.253438	17.092338	49487.574440
std	440.951244	24.437900	12.111077	10.395141	23552.602532
min	1.000000	1901.000000	0.000000	0.000000	0.000000
25%	382.500000	1976.000000	21.000000	10.000000	31383.000000
50%	764.000000	1998.000000	30.000000	16.000000	49056.000000
75%	1145.500000	2011.000000	38.000000	24.000000	68321.500000
max	1527.000000	2021.000000	70.000000	61.000000	106869.000000

In [21]:

```
df.columns
```

Out[21]:

```
Index(['id', 'year', 'date', 'day', 'winner_tie', 'winner_rank',
       'winner_points', 'loser_tie', 'loser_rank', 'loser_points',
       'attendance', 'mvp', 'sponsor', 'bowl_name'],
      dtype='object')
```

In [22]:

```
df.dropna()
```

Out[22]:

	id	year	date	day	winner_tie	winner_rank	winner_points	loser_tie	lose
0	1	2021	12/29/2021	Wed	Oklahoma	14	47	Oregon	
3	4	2018	12/28/2018	Fri	Washington State	12	28	Iowa State	
4	5	2017	12/28/2017	Thu	Texas Christian	13	39	Stanford	
5	6	2016	12/29/2016	Thu	Oklahoma State	13	38	Colorado	
6	7	2015	1/2/2016	Sat	Texas Christian	11	47	Oregon	
...	...	...	...	...	...	...	...	...	
1427	1428	1996	12/30/1996	Mon	Colorado	8	33	Washington	
1429	1430	1994	12/30/1994	Fri	Michigan	20	24	Colorado State	
1434	1435	1989	12/29/1989	Fri	Penn State	18	50	Brigham Young	
1435	1436	1988	12/30/1988	Fri	Oklahoma State	12	62	Wyoming	

	id	year	date	day	winner_tie	winner_rank	winner_points	loser_tie	lose
1504	1505	2008	12/23/2008	Tue	Texas Christian	11	17	Boise State	

269 rows × 14 columns

In [23]:

```
df.isnull().sum()
```

Out[23]:

```
id                 0
year               0
date               0
day                0
winner_tie         0
winner_rank      773
winner_points      0
loser_tie          0
loser_rank       856
loser_points       0
attendance         9
mvp              169
sponsor          707
bowl_name          0
dtype: int64
```

In [24]:

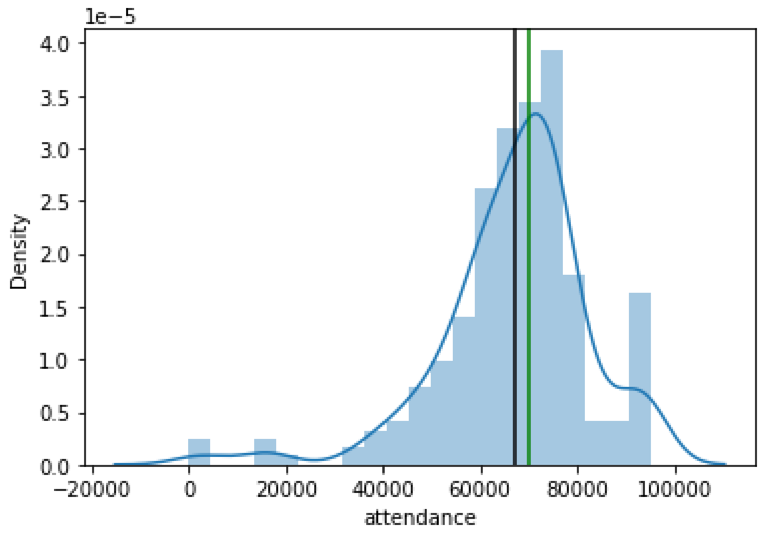
```
df.dropna(inplace=True)
df.head()
```

Out[24]:

	id	year	date	day	winner_tie	winner_rank	winner_points	loser_tie	loser_rank	los
0	1	2021	12/29/2021	Wed	Oklahoma	14	47	Oregon	15	
3	4	2018	12/28/2018	Fri	Washington State	12	28	Iowa State	25	
4	5	2017	12/28/2017	Thu	Texas Christian	13	39	Stanford	15	
5	6	2016	12/29/2016	Thu	Oklahoma State	13	38	Colorado	11	
6	7	2015	1/2/2016	Sat	Texas Christian	11	47	Oregon	15	

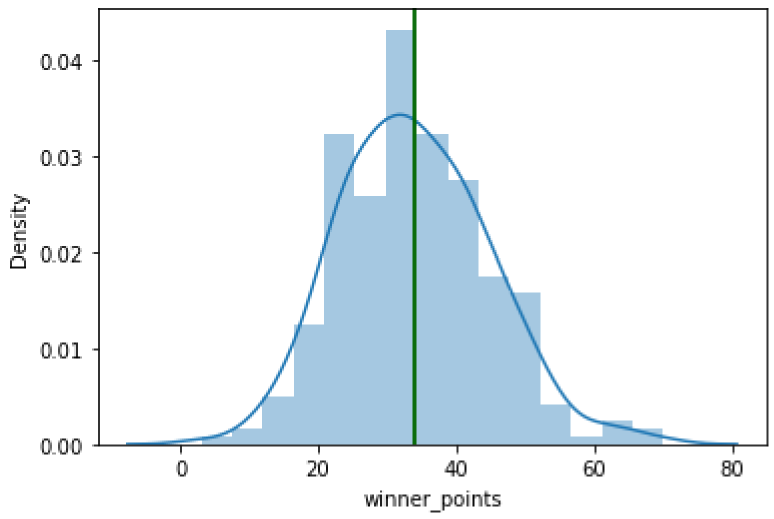
In [25]:

```
sns.distplot(df.attendance)
plt.axvline(df.attendance.mean(), color="black")
plt.axvline(df.attendance.median(), color="green")
plt.show()
```



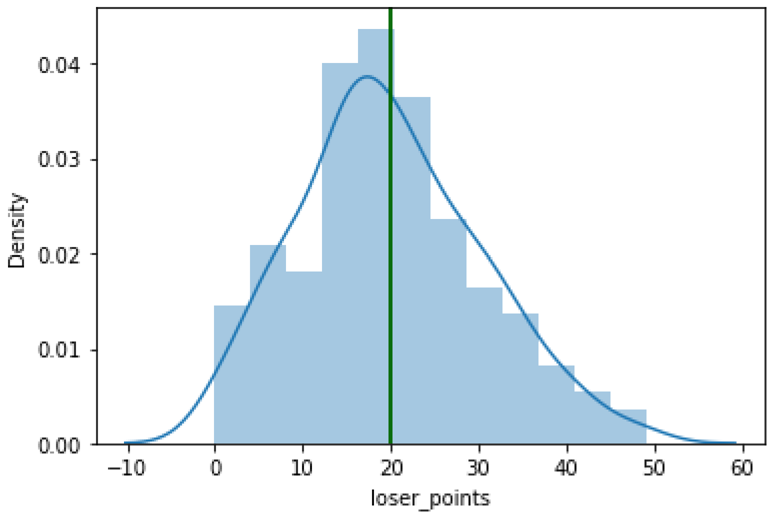
In [26]:

```
sns.distplot(df.winner_points)
plt.axvline(df.winner_points.mean(), color="black")
plt.axvline(df.winner_points.median(), color="green")
plt.show()
```



In [27]:

```
sns.distplot(df.loser_points)
plt.axvline(df.loser_points.mean(), color="black")
plt.axvline(df.loser_points.median(), color="green")
plt.show()
```



In [28]:

```
df.attendance.describe(percentiles = [0.25,0.50,0.75,0.85,0.90,1])
```

Out[28]:

```
count      269.000000
mean     67146.888476
std      15555.305191
min          0.000000
25%      59815.000000
50%      70016.000000
75%      74682.000000
85%      78576.000000
90%      83031.600000
100%     95173.000000
max      95173.000000
Name: attendance, dtype: float64
```

In [29]:

```
df['day'].value_counts()
```

Out[29]:

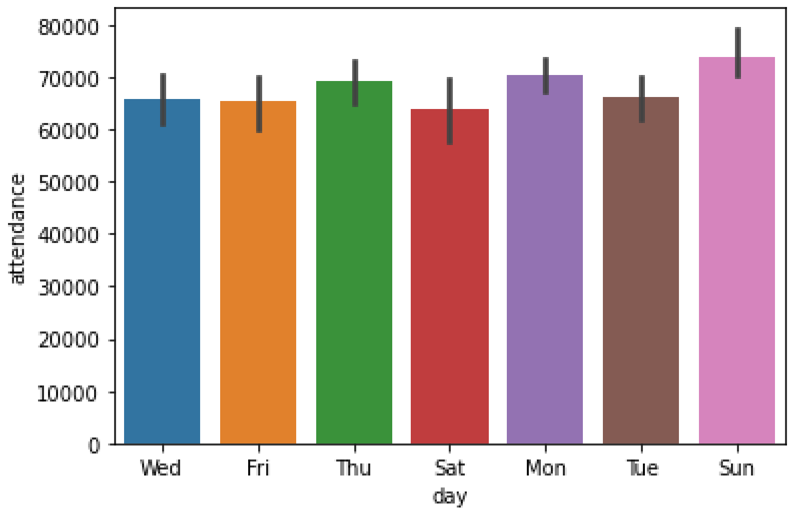
```
Mon     62
Fri     50
Sat     41
Thu     38
Wed     38
Tue     37
Sun      3
Name: day, dtype: int64
```

In [30]:

```
sns.barplot(x = 'day',y = 'attendance',data = df)
```

Out[30]:

```
<AxesSubplot:xlabel='day', ylabel='attendance'>
```

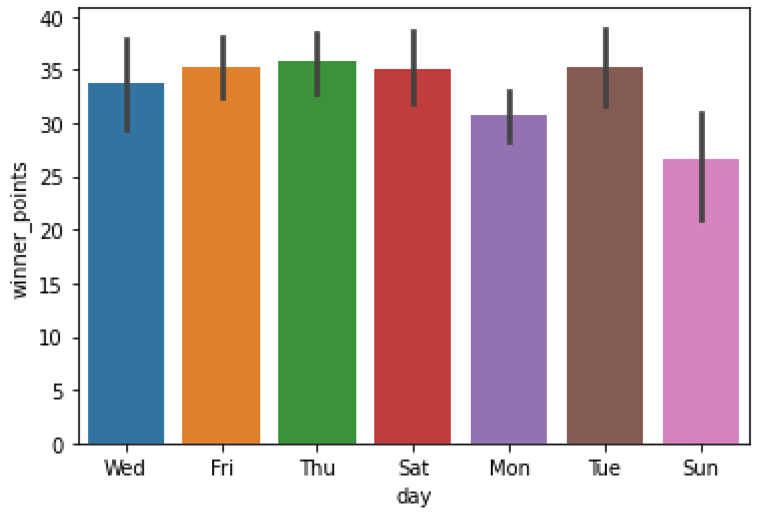


In [31]:

```
sns.barplot(x = 'day',y = 'winner_points',data = df)
```

Out[31]:

```
<AxesSubplot:xlabel='day', ylabel='winner_points'>
```

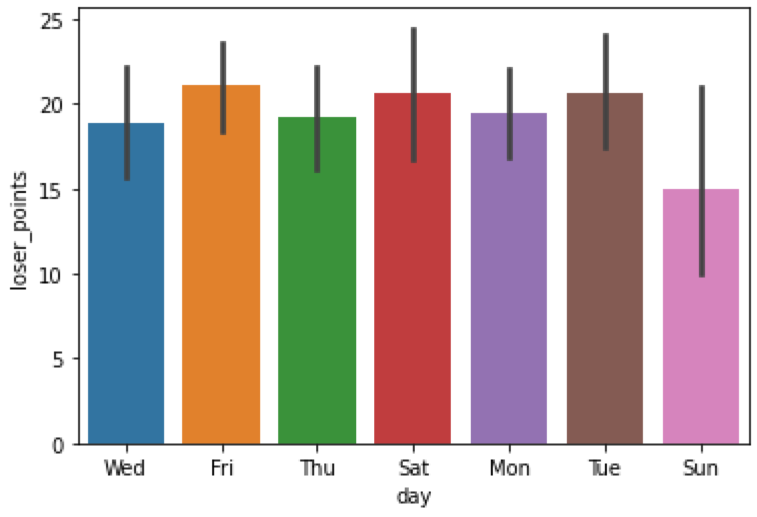


In [32]:

```
sns.barplot(x = 'day',y = 'loser_points',data = df)
```

Out[32]:

```
<AxesSubplot:xlabel='day', ylabel='loser_points'>
```



In [33]:

```
df.columns
```

Out[33]:

```
Index(['id', 'year', 'date', 'day', 'winner_tie', 'winner_rank',
       'winner_points', 'loser_tie', 'loser_rank', 'loser_points',
       'attendance', 'mvp', 'sponsor', 'bowl_name'],
      dtype='object')
```

In [34]:

```
x=df.drop(['date','day','winner_tie','loser_tie','mvp','sponsor','bowl_name','winner
y=df['attendance']
x
```

Out[34]:

	id	year	winner_points	loser_points
0	1	2021	47	32
3	4	2018	28	26
4	5	2017	39	37
5	6	2016	38	8
6	7	2015	47	41
...	...	...	...	...
1427	1428	1996	33	21
1429	1430	1994	24	14
1434	1435	1989	50	39
1435	1436	1988	62	14
1504	1505	2008	17	16

269 rows × 4 columns

In [35]:

```
y
```

Out[35]:

```
0        59121.0
3        60675.0
4        57653.0
5        59815.0
6        64569.0
          ...
1427     54749.0
1429     59453.0
1434     61113.0
1435     60718.0
1504     34628.0
Name: attendance, Length: 269, dtype: float64
```

TTS(TRAIN TEST SPLIT)

In [36]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_stat
X_train = X_train.replace((np.inf, -np.inf, np.nan), 0).reset_index(drop=True).repla
```

In [37]:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
num_vars = ['winner_points','loser_points']
X_train[num_vars] = scaler.fit_transform(X_train[num_vars])
X_test[num_vars] = scaler.transform(X_test[num_vars])
X_test.head()
```

Out[37]:

	id	year	winner_points	loser_points
1409	1410	2014	0.626866	0.875000
1062	1063	2012	0.447761	0.479167
337	338	2020	0.462687	0.354167
917	918	2017	0.761194	1.000000
915	916	2019	0.373134	0.562500

In [38]:

```
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(215, 4) (54, 4) (215,) (54,)
```

In [39]:

```
df.describe()
```

Out[39]:

	id	year	winner_points	loser_points	attendance
count	269.000000	269.000000	269.000000	269.000000	269.000000
mean	661.245353	2005.609665	33.918216	19.929368	67146.888476
std	414.018331	9.609544	10.884777	10.381466	15555.305191
min	1.000000	1985.000000	3.000000	0.000000	0.000000
25%	246.000000	1998.000000	26.000000	13.000000	59815.000000
50%	730.000000	2006.000000	34.000000	20.000000	70016.000000
75%	1028.000000	2014.000000	41.000000	27.000000	74682.000000
max	1505.000000	2021.000000	70.000000	49.000000	95173.000000

In [ ]:

In [40]:

```
X_train.columns
```

Out[40]:

```
Index(['id', 'year', 'winner_points', 'loser_points'], dtype='object')
```

In [ ]:

```

```

In [41]:

```
X_train.describe()
```

Out[41]:

	id	year	winner_points	loser_points
count	215.000000	215.000000	215.000000	215.000000
mean	653.906977	2005.111628	0.458938	0.408818
std	418.007111	9.736558	0.162556	0.217117
min	1.000000	1985.000000	0.000000	0.000000
25%	244.000000	1997.500000	0.343284	0.270833
50%	724.000000	2006.000000	0.462687	0.375000
75%	1051.500000	2013.000000	0.567164	0.562500
max	1505.000000	2021.000000	1.000000	1.000000

In [42]:

```
X_test.describe()
```

Out[42]:

	id	year	winner_points	loser_points
count	54.000000	54.000000	54.000000	54.000000
mean	690.462963	2007.592593	0.471531	0.440586
std	400.195976	8.896721	0.163204	0.213014
min	8.000000	1987.000000	0.238806	0.000000
25%	339.500000	2000.250000	0.373134	0.322917
50%	821.000000	2007.500000	0.440299	0.437500
75%	936.500000	2017.000000	0.544776	0.578125
max	1427.000000	2020.000000	0.940299	1.020833

In [43]:

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
lr.score(X_train,y_train)
```

Out[43]:

```
0.03354321250299419
```

In [44]:

```
from sklearn.linear_model import Ridge
ridge=Ridge()
ridge=Ridge(alpha=189)
ridge.fit(X_train,y_train)
print('Train: ',ridge.score(X_train,y_train))
print('Test: ',ridge.score(X_test,y_test))
```

```
Train:  0.033286218997407246
Test:  0.02479032575121498
```

DECISION TREE

THERE IS NO CO LINEARITY IN DATA SO WE WILL USE DT

In [45]:

```
catcols=list(df.select_dtypes(include=['object']).head())
df=pd.get_dummies(df, columns=catcols, drop_first=True)
df.head()
```

Out[45]:

	id	year	winner_points	loser_points	attendance	date_1/1/1987	date_1/1/1988	date_1/1/199
0	1	2021	47	32	59121.0	0	0	(
3	4	2018	28	26	60675.0	0	0	(
4	5	2017	39	37	57653.0	0	0	(
5	6	2016	38	8	59815.0	0	0	(
6	7	2015	47	41	64569.0	0	0	(

5 rows × 675 columns

In [46]:

```
dtx=df.drop(['attendance'], axis=1)
dty=df['attendance']
```

In [ ]:

In [47]:

```
from sklearn.model_selection import train_test_split
dtxt,dtxte,dtyt,dtyte = train_test_split(dtx,dty, test_size = 0.2, random_state = 2)
```

In [48]:

```
from sklearn.tree import DecisionTreeRegressor
dtmodel = DecisionTreeRegressor(random_state=0)
dtmodel.fit(dtxt,dtyt)
```

Out[48]:

```
DecisionTreeRegressor(random_state=0)
```

In [49]:

```
print(dtmodel.score(dtxt,dtyt))
print(dtmodel.score(dtxte,dtyte))
```

```
1.0
0.37473410128085227
```

There is overfiiting in data because of absence of colinearity

In [ ]: