

```
In [1]: ▶ #Import the libraries.  
  
#Import the libraries.  
  
import seaborn as sns  
import re  
import numpy as np # linear algebra  
import pandas as pd # data processing, CSV file I/O  
import matplotlib.pyplot as plt  
import plotly.graph_objects as go  
import plotly.figure_factory as ff  
import plotly  
import plotly.express as px
```

```
In [2]: ▶ import warnings  
warnings.filterwarnings('ignore')  
%matplotlib inline
```

```
In [3]: ▶ from IPython.core.interactiveshell import InteractiveShell  
InteractiveShell.ast_node_interactivity = 'all'
```

```
In [4]: ▶ #Import the data  
import pandas as pd  
PA = pd.read_csv(r"C:\Users\16093\ProjectDataset.csv")
```

In [5]: `print(PA)`

```

      state    playerid    wagerid    event_start \
0      State1  30651211.80  1.693004e+06  2021-04-28 00:30:00+00
1      State1  22237170.91  1.696371e+06  2021-04-28 01:45:00+00
2      State1  22237170.91  1.696371e+06  2021-04-28 01:45:00+00
3      State1  22237170.91  1.696371e+06  2021-04-28 01:45:00+00
4      State1  22237170.91  1.696371e+06  2021-04-28 01:45:00+00
...      ...      ...      ...      ...
1048570  State3  20520826.33  2.425001e+07  2021-10-02 01:27:00+00
1048571  State3  20520826.33  2.425009e+07  2021-10-02 01:27:00+00
1048572  State3  20520826.33  2.425009e+07  2021-10-02 00:27:00+00
1048573  State1  22245057.45  2.425012e+07  2021-10-27 00:10:00+00
1048574  State1  22245057.45  2.425012e+07  2021-10-27 00:10:00+00

      placed_date  settled_date    sportname  bet_type  result \
0      4/27/2021    4/27/2021          nhl  straight    won
1      4/27/2021    4/27/2021          nba   parlay    lost
2      4/27/2021    4/27/2021          nba   parlay    lost
3      4/27/2021    4/27/2021          nba   parlay    lost
4      4/27/2021    4/27/2021          nba   parlay    lost
...      ...      ...      ...      ...      ...
1048570   10/1/2021   10/1/2021  college football  parlay    lost
1048571   10/1/2021   10/1/2021  college football  parlay    lost
1048572   10/1/2021   10/1/2021  college football  parlay    lost
1048573  10/26/2021  10/27/2021          mlb   parlay    lost
1048574  10/26/2021  10/27/2021          mlb   parlay    lost

      net_stake    ggr legresult  decimalodds
0         6.64   -4.96      won      1.74627
1         5.00    5.00      won      1.78125
2         5.00    5.00     lost      1.86207
3         5.00    5.00     lost      1.74627
4         5.00    5.00     lost      1.78125
...      ...      ...      ...      ...
1048570   100.00  100.00      won      1.28169
1048571    50.00   50.00      won      1.28169
1048572    50.00   50.00     lost      1.86957
1048573    37.50   37.50      won      1.69444
1048574    37.50   37.50      won      3.30000

```

[1048575 rows x 13 columns]

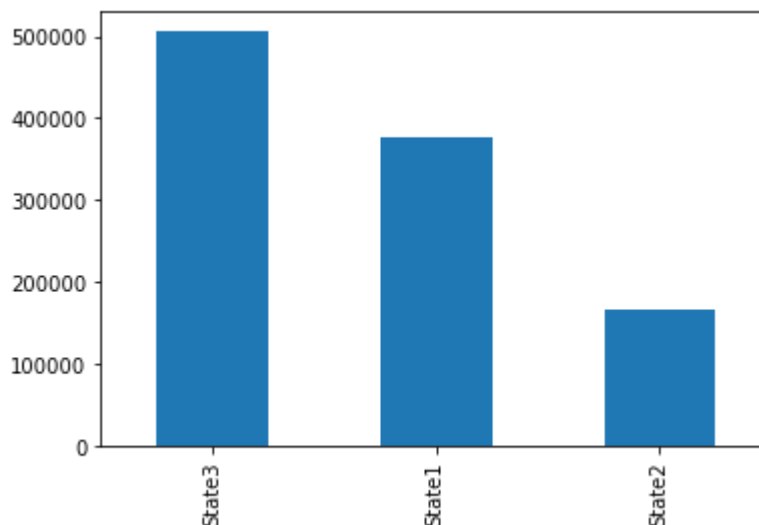
In [6]: PA.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   state                  1048575 non-null object  
1   playerid               1048575 non-null float64 
2   wagerid               1048575 non-null float64 
3   event_start            1048575 non-null object  
4   placed_date            1048575 non-null object  
5   settled_date           1048575 non-null object  
6   sportname              1048575 non-null object  
7   bet_type               1048575 non-null object  
8   result                 1048575 non-null object  
9   net_stake              1048575 non-null float64 
10  ggr                    1048575 non-null float64 
11  legresult              1048575 non-null object  
12  decimalodds            1048569 non-null float64 
dtypes: float64(5), object(8)
memory usage: 104.0+ MB
```

Number of Users Per State

In [7]: PA['state'].value_counts().head(10).plot.bar()

Out[7]: <AxesSubplot:>

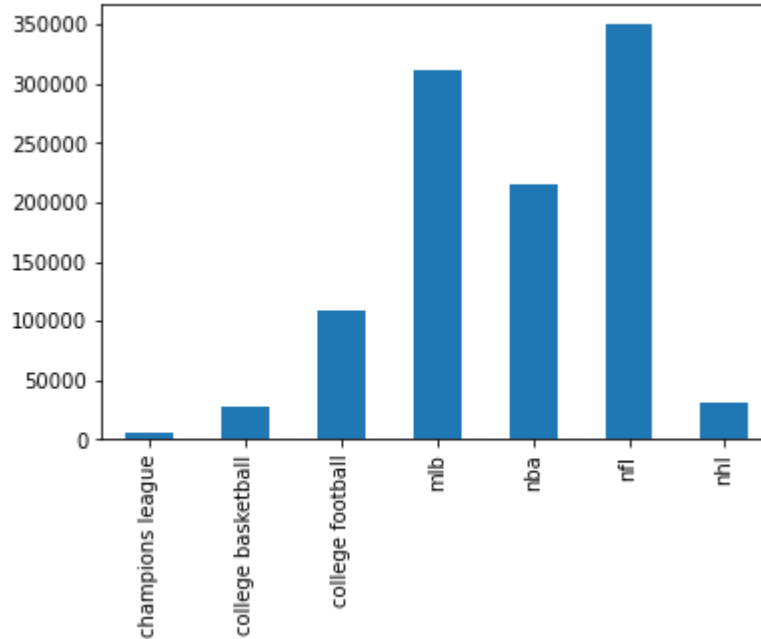


This graph is used to analyze the amount of users per state to help us closer identify which state we can focus on to improve their amount of users. In this case "State2" would be the state that we would primarily focus on to see what we can do to improve the amount of users within the state.

Wagers Per Sport

```
In [8]: PA['sportname'].value_counts().sort_index().plot.bar()
```

Out[8]: <AxesSubplot:>

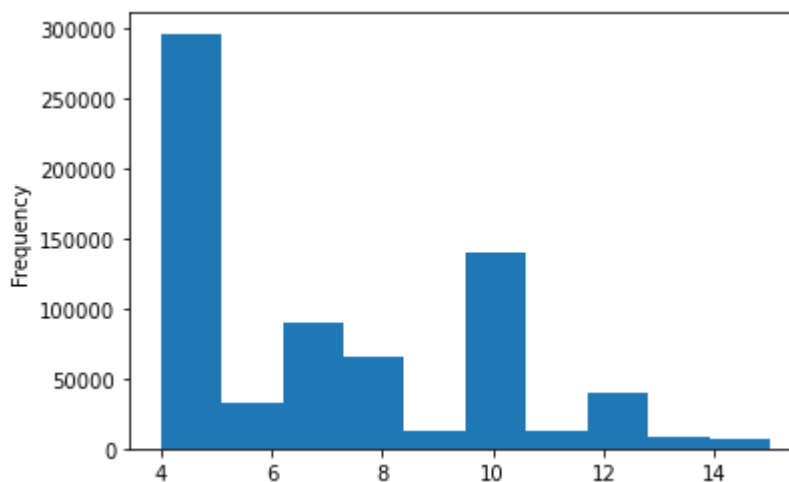


From this graph we can see that the main 3 sports wagered on are all American sports due to our audience being from the United States. This allows us to know which sports need more attention when designing promotions per sport.

Frequency of Stakes Less Than 10

```
In [18]: PA[PA['net_stake'] < 15]['net_stake'].plot.hist()
```

Out[18]: <AxesSubplot:ylabel='Frequency'>



This graph allows us to see that users are more likely to place stakes around 4-5. Resulting in our promotions giving risk free for higher amounts to get users used to placing those types of bets to

increase revenue from higher spending.

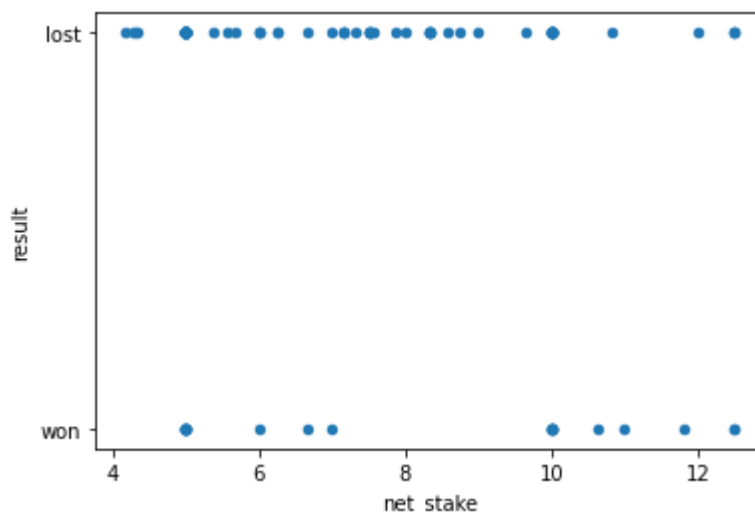
```
In [10]: PA[PA['ggr'] > 10]
```

06	2021-04-27 23:21:00+00	4/27/2021	4/27/2021	mlb	straight	lost	25.0	25.0	lost	6.00000
...
07	2021-10-02 01:27:00+00	10/1/2021	10/1/2021	college football	parlay	lost	100.0	100.0	won	1.28169
07	2021-10-02 01:27:00+00	10/1/2021	10/1/2021	college football	parlay	lost	50.0	50.0	won	1.28169
07	2021-10-02 00:27:00+00	10/1/2021	10/1/2021	college football	parlay	lost	50.0	50.0	lost	1.86957
07	2021-10-27 00:10:00+00	10/26/2021	10/27/2021	mlb	parlay	lost	37.5	37.5	won	1.69444
07	2021-10-27 00:10:00+00	10/26/2021	10/27/2021	mlb	parlay	lost	37.5	37.5	won	3.30000

Result to Net Stake Comparison

```
In [11]: PA[PA['net_stake'] < 15].sample(100).plot.scatter(x='net_stake', y='result')
```

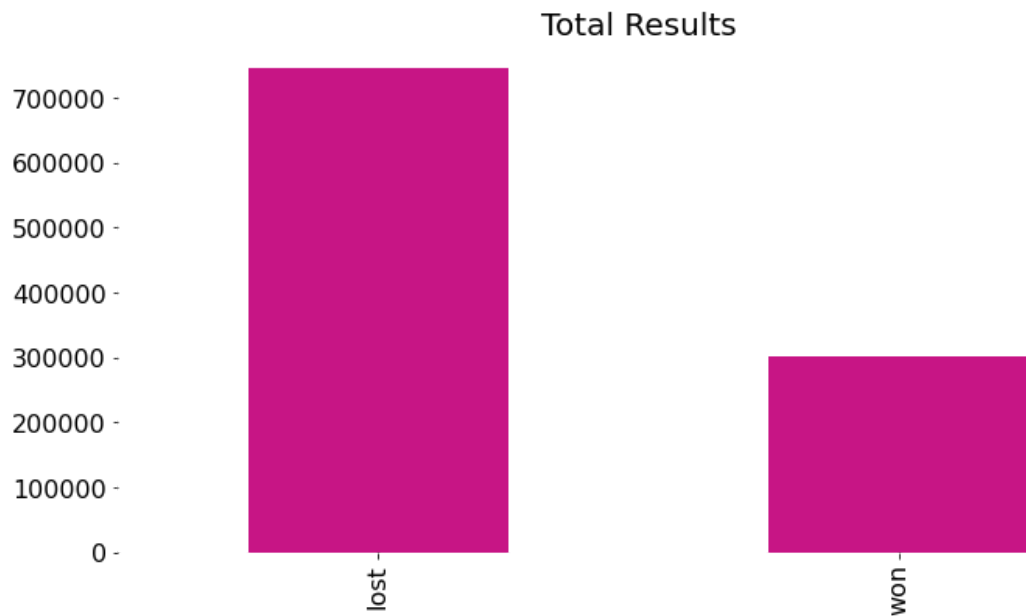
```
Out[11]: <AxesSubplot:xlabel='net_stake', ylabel='result'>
```



Total Win/Loss

```
In [12]: ▶ import matplotlib.pyplot as plt
import seaborn as sns

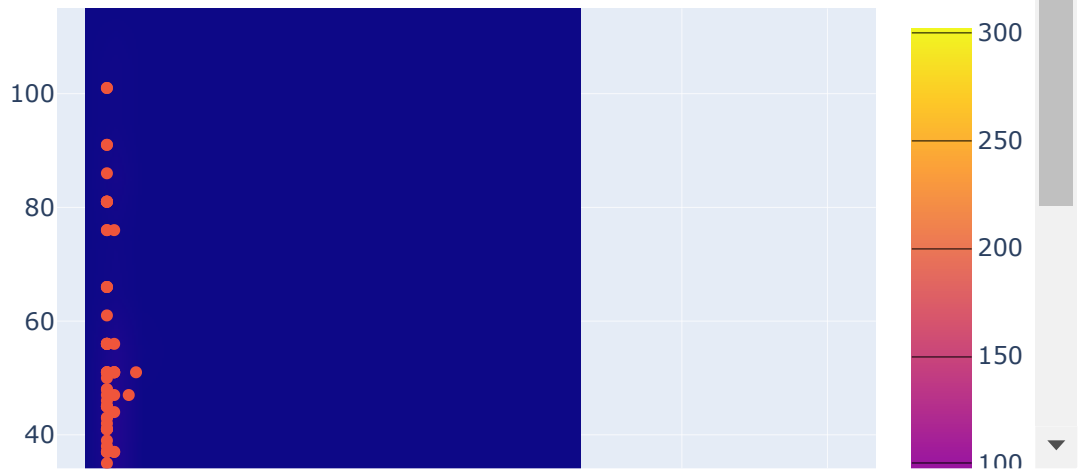
ax = PA['result'].value_counts().sort_index().plot.bar(
    figsize=(12, 6),
    color='mediumvioletred',
    fontsize=16
)
ax.set_title("Total Results", fontsize=20)
sns.despine(bottom=True, left=True)
```



```
In [13]: ▶ from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
```

Net Stake to Decimal Odds

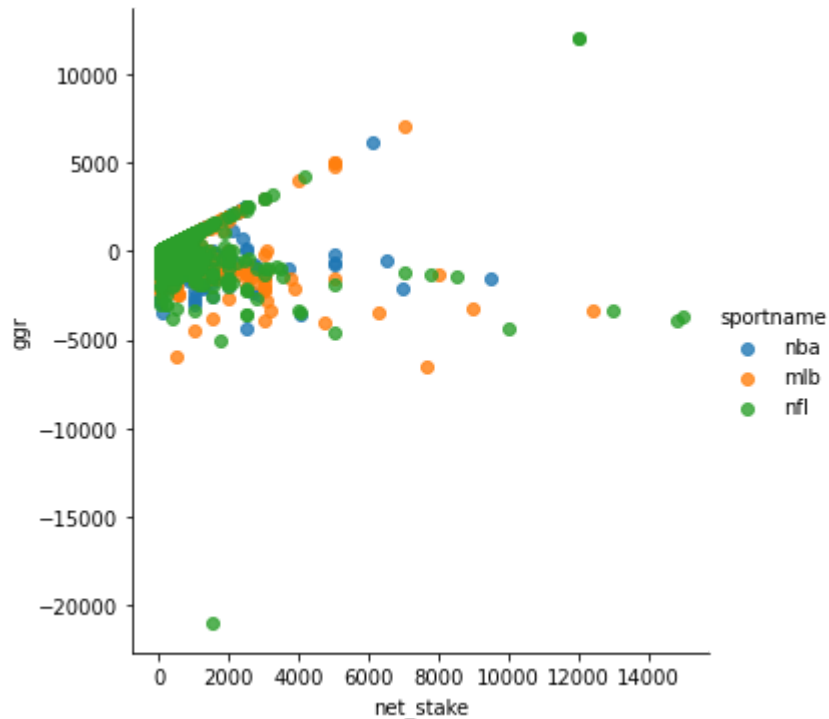
```
In [14]: plt([go.Histogram2dContour(x=PA.head(500)['net_stake'],  
                                     y=PA.head(500)['decimalodds'],  
                                     contours=go.Contours(coloring='heatmap')),  
            go.Scatter(x=PA.head(1000)['net_stake'], y=PA.head(1000)['decimalodds'], m
```



In this graph you see there is higher amount of net stakes with lower decimal odds which goes to show users are trying to place wagers rather safe than sorry. Seeing from promotions we have implemented that users must reach a certain number for their odds to qualify for the promotion, and with promotions like that it can generate higher decimal odds, and wager amount from users trying to make more money.

```
In [16]: sns.lmplot(x='net_stake', y='ggr', hue='sportname',
                    data=PA.loc[PA['sportname'].isin(['mlb', 'nba', 'nfl'])],
                    fit_reg=False)
```

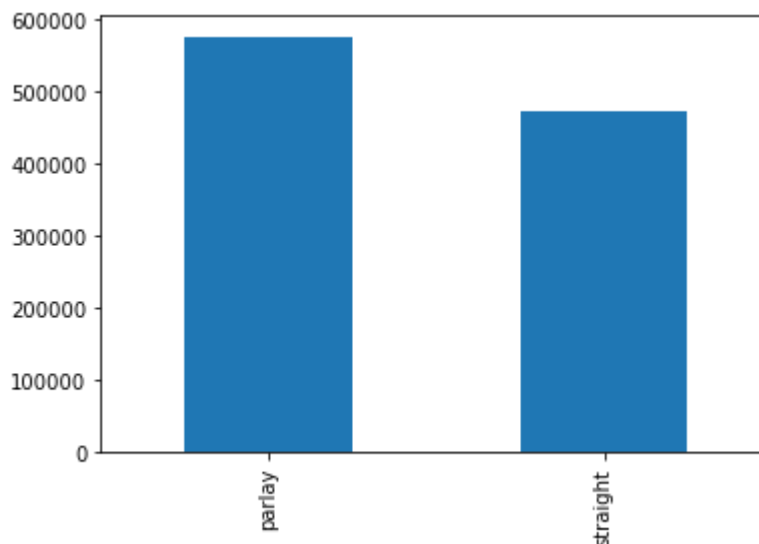
Out[16]: <seaborn.axisgrid.FacetGrid at 0x11686b75760>



For this graph I plot the net stake and gross gaming revenue per sport to help our company get an idea of what sports provide the most revenue. This graph also gives us an idea of the stakes users are placing on these sports to give us an insight as to which sports we can better promote.

```
In [19]: PA['bet_type'].value_counts().sort_index().plot.bar()
```

Out[19]: <AxesSubplot:>



Cross-sell Opportunity

I would use this data to create a cross-sell by seeing what sports users are most likely to wager on, and finding a way to make a promotion involving that particular sport and racing. Obviously we cross-sell casino and sports book by having them on the same app so users are more likely to stumble upon casino, and its' many promotions, but how can we incorporate DFS and racing into the mix? During the football season we could cross promote DFS and Sportsbook together giving promotions such as if you place a certain wager you will qualify for bonuses on DFS resulting in users getting into Sportsbook for the benefit of their DFS. For racing we could incorporate the races into sports for promotions to have users start looking into racing, and possibly becoming more interested in betting on the races as a straight. Inclusion of racing into sportsbook would be another opportunity allowing users to parlay these races into their bets helping drive long term value into racing.

In []: ▶