CMSE 201 Final Project

✅ Jeremy Frazee

✅ Section_001

✅ 4/20/2021

Hat Trick: How, When, and Why Hockey Teams Win Games!

Background and Motivation

The National Hockey League (NHL) is the premiere hockey league for professional hockey in the United States of America and around the world. Consisting of 31 member teams, the league collects certain data on games and the players that play in the league. For my final project, I will use NHL data, including player statistics, team statistics, game goal statistics, and power play (where one team has a player advantage because the other has incurred a penalty) data. The data comes from this source: https://www.kaggle.com/martinellis/nhl-game-data

I wanted to do a project on analyzing NHL data because I am an avid watcher of the NHL and hockey generally. Ever since I was young, I have been a fan of hockey. Learning to skate at the age of five, I played youth league and then high school hockey after that before entering college.Throughout my time playing hockey and watching hockey games, there are a suprising number of assumptions that are made about why and how teams win games and players score goals. I thought that this analysis project would be a great way to test some of those assumptions with data analysis to see if they are accurate. This leads me to my questions:

- Which NHL Teams Have Won The Most Games?
- What Were The Top Scoring Seasons?
- What Is The Impact of The Power Play on Goal Scoring and Winning Games?
- How many games were won in OT vs. in regulation?
- Is There a Relationship Between Time on Ice and Goal Scoring?
- Is There a Relationship Between Power Play Time and Goal Scoring?
- Is There A Relationship Between Shots On Net And Goal Scoring?
- Is There A Relationship Between Face Off Win Percentage and Goal Scoring?

Methodology

In [1]:
```
# These are the modules that I will use

%matplotlib inline

import pandas as pd

import numpy as np


import matplotlib.pyplot as plt


import matplotlib.style as style


style.use('fivethirtyeight') #I will use this style for my graphs
```

The following cells were used to create scatter plots, bar charts, horizontal bar charts, and regressions for various NHL related data. The bar charts compare: the top winning teams in the time period, the top goal scoring seasons, the number of game winning goals on the power play vs not, and the number of games won in regulation vs the power play. The scatter plots compare:goals among goal scoring players vs time on ice, goals among goal scorers vs. power play time on ice, and goals among goal scorers vs shots on net, and face off win percentage vs goals scored among goal scoring players. The data is limited to NHL data since 2000.

In [2]:
```
# I will be using four datasets for this analysis:
# skater statistics, team statistics, goal statistics, and other game data

game_skater_stats = pd.read_csv("game_skater_stats.csv")


game_teams_stats = pd.read_csv("game_teams_stats.csv")


game_goals = pd.read_csv("game_goals.csv")


game = pd.read_csv("game.csv")
```

```
/Users/jeremyfrazee/opt/anaconda3/lib/python3.8/site-packages/IPython/core/interactiveshell.py:3146: DtypeWarni
ng: Columns (2) have mixed types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [26]:
```
print(game_skater_stats)
```

```
           game_id  player_id  team_id  timeOnIce  assists  goals  shots  \
0       2016020045    8468513        4        955        1      0      0
1       2016020045    8476906        4       1396        1      0      4
2       2016020045    8474668        4        915        0      0      1
3       2016020045    8473512        4       1367        3      0      0
4       2016020045    8471762        4        676        0      0      3
...            ...        ...      ...        ...      ...    ...    ...
945825  2018030417    8475807        6        501        0      0      1
945826  2018030417    8475149        6       1062        0      0      2
945827  2018030417    8475745        6        913        0      0      2
945828  2018030417    8478443        6       1006        0      0      1
945829  2018030417    8478046        6        909        0      0      2

        hits  powerPlayGoals  powerPlayAssists  ...  faceoffTaken  takeaways  \
0        2.0               0                 0  ...             0        1.0
1        2.0               0                 0  ...             0        1.0
2        1.0               0                 0  ...             0        2.0
3        0.0               0                 2  ...            27        0.0
4        2.0               0                 0  ...             0        0.0
...      ...             ...               ...  ...           ...        ...
945825   2.0               0                 0  ...             0        0.0
945826   0.0               0                 0  ...             1        1.0
945827   3.0               0                 0  ...            11        1.0
945828   1.0               0                 0  ...             0        0.0
945829   0.0               0                 0  ...             0        1.0

        giveaways  shortHandedGoals  shortHandedAssists  blocked  plusMinus  \
0             1.0                 0                   0      1.0          1
1             2.0                 0                   0      2.0          0
2             0.0                 0                   0      0.0         -1
3             0.0                 0                   0      0.0         -1
4             1.0                 0                   0      0.0         -1
...           ...               ...                 ...      ...        ...
945825        0.0                 0                   0      0.0         -1
945826        0.0                 0                   0      0.0         -1
945827        0.0                 0                   0      0.0          0
945828        1.0                 0                   0      2.0         -1
945829        1.0                 0                   0      0.0          0

        evenTimeOnIce  shortHandedTimeOnIce  powerPlayTimeOnIce
0                 858                    97                   0
1                1177                     0                 219
2                 805                     0                 110
3                1083                    19                 265
4                 613                    63                   0
...               ...                   ...                 ...
945825            501                     0                   0
945826            957                     0                 105
945827            913                     0                   0
945828           1006                     0                   0
945829            909                     0                   0

[945830 rows x 22 columns]
```

In [25]:
```
print(game_teams_stats)
```

```
          game_id  team_id   HoA    won settled_in        head_coach  goals  \
0      2016020045        4  away  False        REG      Dave Hakstol    4.0
1      2016020045       16  home   True        REG  Joel Quenneville    7.0
2      2017020812       24  away   True         OT      Randy Carlyle    4.0
3      2017020812        7  home  False         OT      Phil Housley    3.0
4      2015020314       21  away   True        REG       Patrick Roy    4.0
...           ...      ...   ...    ...        ...               ...    ...
52605  2018030416       19  home  False        REG      Craig Berube    1.0
52606  2018030417       19  away   True        REG      Craig Berube    4.0
52607  2018030417        6  home  False        REG     Bruce Cassidy    1.0
52608  2018030417       19  away   True        REG      Craig Berube    4.0
52609  2018030417        6  home  False        REG     Bruce Cassidy    1.0

       shots  hits   pim  powerPlayOpportunities  powerPlayGoals  \
0       27.0  30.0   6.0                     4.0             2.0
1       28.0  20.0   8.0                     3.0             2.0
2       34.0  16.0   6.0                     3.0             1.0
3       33.0  17.0   8.0                     2.0             1.0
4       29.0  17.0   9.0                     3.0             1.0
...      ...   ...   ...                     ...             ...
52605   29.0  29.0  20.0                     4.0             0.0
52606   20.0  36.0   2.0                     0.0             0.0
52607   33.0  28.0   0.0                     1.0             0.0
52608   20.0  36.0   2.0                     0.0             0.0
52609   33.0  28.0   0.0                     1.0             0.0

       faceOffWinPercentage  giveaways  takeaways  blocked startRinkSide
0                      50.9       12.0        9.0     11.0          left
1                      49.1       16.0        8.0      9.0          left
2                      43.8        7.0        4.0     14.0         right
3                      56.2        5.0        6.0     14.0         right
4                      45.7       13.0        5.0     20.0          left
...                     ...        ...        ...      ...           ...
52605                  58.7       12.0       11.0      9.0         right
52606                  49.0        7.0        8.0     21.0         right
52607                  51.0       13.0        6.0      7.0         right
52608                  49.0        7.0        8.0     21.0         right
52609                  51.0       13.0        6.0      7.0         right

[52610 rows x 17 columns]
```

In [24]:
```
print(game_goals)
```

```
                 play_id    strength gameWinningGoal emptyNet
0           2016020045_6        Even           False    False
1          2016020045_97        Even           False    False
2         2016020045_103  Power Play           False    False
3         2016020045_140  Power Play           False    False
4         2016020045_197  Power Play           False    False
...                  ...         ...             ...      ...
148987     2018030417_81        Even           False    False
148988    2018030417_100        Even            True    False
148989    2018030417_257        Even           False    False
148990    2018030417_277        Even           False    False
148991    2018030417_290        Even           False    False

[148992 rows x 4 columns]
```

In [23]:
```
print(game)
```

```
          game_id    season type          date_time_GMT  away_team_id  \
0      2016020045  20162017    R  2016-10-19T00:30:00Z             4
1      2017020812  20172018    R  2018-02-07T00:00:00Z            24
2      2015020314  20152016    R  2015-11-24T01:00:00Z            21
3      2015020849  20152016    R  2016-02-17T00:00:00Z            52
4      2017020586  20172018    R  2017-12-30T03:00:00Z            20
...           ...       ...  ...                   ...           ...
26300  2018030415  20182019    P  2019-06-07T00:00:00Z            19
26301  2018030416  20182019    P  2019-06-10T00:00:00Z             6
26302  2018030416  20182019    P  2019-06-10T00:00:00Z             6
26303  2018030417  20182019    P  2019-06-13T00:00:00Z            19
26304  2018030417  20182019    P  2019-06-13T00:00:00Z            19

       home_team_id  away_goals  home_goals       outcome  \
0                16           4           7  home win REG
1                 7           4           3   away win OT
2                52           4           1  away win REG
3                12           1           2  home win REG
4                24           1           2  home win REG
...             ...         ...         ...           ...
26300             6           2           1  away win REG
26301            19           5           1  away win REG
26302            19           5           1  away win REG
26303             6           4           1  away win REG
26304             6           4           1  away win REG

      home_rink_side_start              venue          venue_link  \
0                    right      United Center  /api/v1/venues/null
1                     left     KeyBank Center  /api/v1/venues/null
2                    right         MTS Centre  /api/v1/venues/null
3                    right          PNC Arena  /api/v1/venues/null
4                     left       Honda Center  /api/v1/venues/null
...                    ...                ...                  ...
26300                 left          TD Garden  /api/v1/venues/5085
26301                 left  Enterprise Center  /api/v1/venues/5076
26302                 left  Enterprise Center  /api/v1/venues/5076
26303                 left          TD Garden  /api/v1/venues/5085
26304                 left          TD Garden  /api/v1/venues/5085

        venue_time_zone_id  venue_time_zone_offset venue_time_zone_tz
0          America/Chicago                      -5                CDT
1         America/New_York                      -4                EDT
2         America/Winnipeg                      -5                CDT
3         America/New_York                      -4                EDT
4      America/Los_Angeles                      -7                PDT
...                    ...                     ...                ...
26300     America/New_York                      -5                EST
26301      America/Chicago                      -6                CST
26302      America/Chicago                      -6                CST
26303     America/New_York                      -5                EST
26304     America/New_York                      -5                EST

[26305 rows x 15 columns]
```

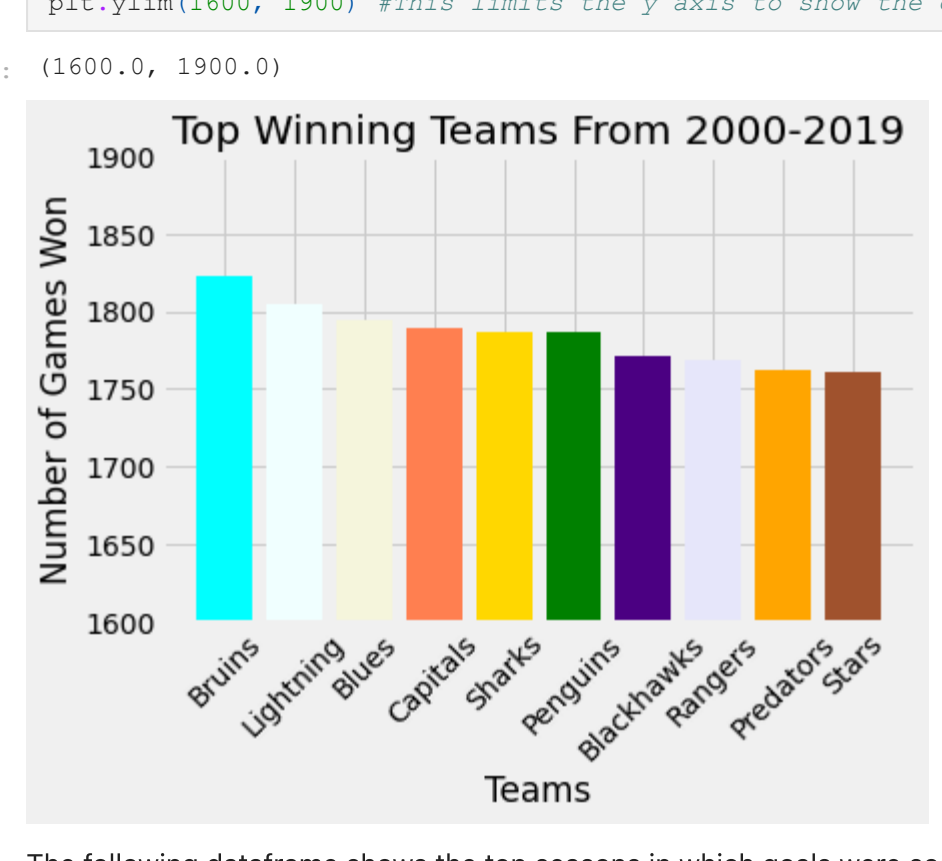
The following dataframe shows the top ten teams that have won games in the aforementioned time period.

In [3]:
```
team_id = game_teams_stats["team_id"]

won = game_teams_stats["won"]

frame1 = { 'team_id': team_id, 'won': won }

result1 = pd.DataFrame(frame1)

result1["won"] = result1["won"].astype(int)


result12 = result1.groupby(['team_id']).count()

result13 = result12.sort_values(by =['won'],ascending=False)

result14 = result13[0:10] #This shows the top ten in the set

print(result14)
```

```
          won
team_id
6        1822
14       1804
19       1794
15       1789
28       1787
5        1786
16       1771
3        1768
18       1762
25       1761
```

In [4]:
```
# This plot shows goals scored over time

team_names = ['Bruins','Lightning','Blues','Capitals','Sharks','Penguins','Blackhawks','Rangers','Predators',
              'Stars']

colors = ['aqua','azure','beige','coral','gold','green','indigo','lavender','orange','sienna']

result15 = result14['won']

frame2 = { 'team_names': team_names, 'result15': result15 }

result2 = pd.DataFrame(frame2)

lnwon = result2["result15"]

plt.xticks(rotation = 45)

plt.bar(result2["team_names"],lnwon,color=colors)


plt.title("Top Winning Teams From 2000-2019")

plt.xlabel("Teams")

plt.ylabel("Number of Games Won")


plt.ylim(1600, 1900) #This limits the y axis to show the change
```

Out[4]: (1600.0, 1900.0)



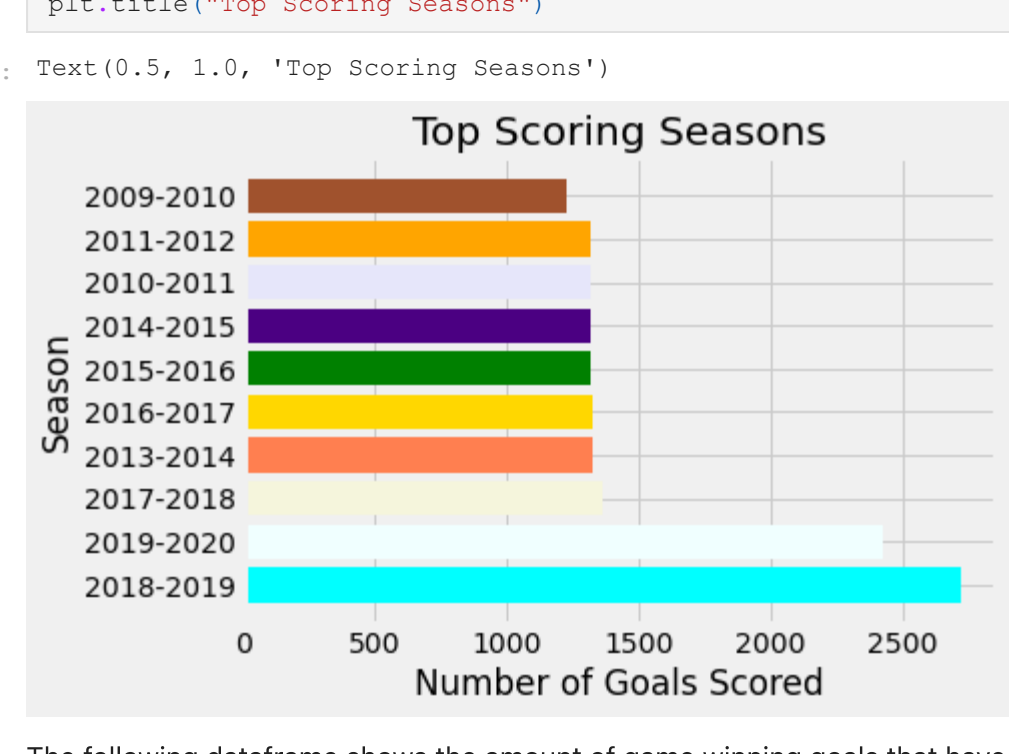
The following dataframe shows the top seasons in which goals were scored among the time period in the data set.

In [5]:
```
season = game["season"]

totalgoals = game["away_goals"]+game["home_goals"]

frame3 = { 'season': season, 'totalgoals': totalgoals }

result3 = pd.DataFrame(frame3)


result31 = result3.groupby(['season']).count()

result32 = result31.sort_values(by =['totalgoals'],ascending=False)

result33 = result32[0:10] #This shows the top ten in the set

print(result33)
```

```
          totalgoals
season
20182019        2720
20192020        2425
20172018        1363
20132014        1323
20162017        1323
20152016        1321
20142015        1319
20102011        1319
20112012        1316
20092010        1230
```

In [6]:
```
# This graph shows the most goals scored from season to season

season_names = ["2018-2019","2019-2020","2017-2018","2013-2014","2016-2017","2015-2016","2014-2015",
                "2010-2011","2011-2012","2009-2010"]


lntotalgoals = np.log10(result33["totalgoals"])

plt.barh(season_names,result33["totalgoals"],color=colors)

plt.xlabel("Number of Goals Scored")

plt.ylabel("Season")

plt.title("Top Scoring Seasons")
```

Out[6]: Text(0.5, 1.0, 'Top Scoring Seasons')



The following dataframe shows the amount of game winning goals that have been scored on the power play vs in regulation.

In [7]:
```
strength = game_goals["strength"]

gameWinningGoal = game_goals["gameWinningGoal"]


frame4 = { 'strength': strength, 'gameWinningGoal': gameWinningGoal }

result5 = pd.DataFrame(frame4)

result51 = result5[strength=="Power Play"]

result51["gameWinningGoal"] = result51["gameWinningGoal"].astype(int)

result52 = result51.groupby(['gameWinningGoal']).count() #Groups by and counts gamewinning goals

print(result52)
```

```
                 strength
gameWinningGoal
0                   29295
1                    5229
```

```
<ipython-input-7-547dca9825b5>:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  result51["gameWinningGoal"] = result51["gameWinningGoal"].astype(int)
```

In [8]:
```
# This graph shows if the game winning goals were scored on the power play across the various seasons

gamewinninggoalnames = ["No", "Yes"]

noyescol = ['red','green']


plt.bar(gamewinninggoalnames,result52['strength'],color=noyescol)

plt.xlabel("If Goal Was Game Winning")

plt.ylabel("Frequency")


plt.title("Number of Game Winning Goals on PowerPlay From 2000-2019")
```

Out[8]: Text(0.5, 1.0, 'Number of Game Winning Goals on PowerPlay From 2000-2019')

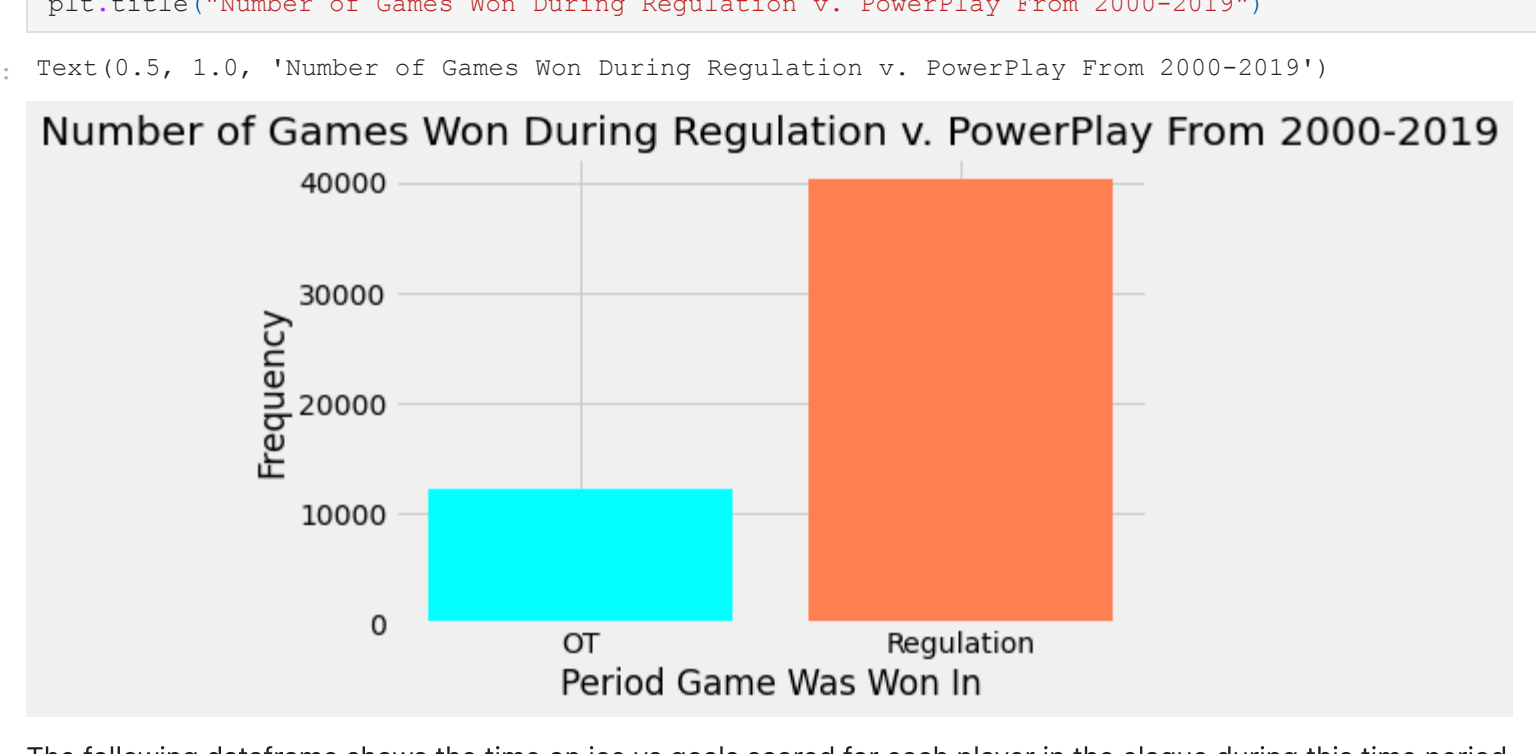
The following dataframe shows the number of games won on the power play vs in regulation

In [9]:
```
wonR = game_teams_stats["won"]

settled_in = game_teams_stats["settled_in"]


frame6 = { 'wonR': wonR, 'settled_in': settled_in }

result6 = pd.DataFrame(frame6)

result6["wonR"] = result6["wonR"].astype(int)


result61 = result6.groupby(['settled_in']).count() #This groups by the period the games were won in and counts

result62 = result61[1:3] #This shows OT v. Regulation

print(result62)
```

```
             wonR
settled_in
OT          12256
REG         40354
```

In [10]:
```
twocol = ['cyan','coral']

final_game_team_resultsframes = ['OT','Regulation']

plt.bar(final_game_team_resultsframes,result62['wonR'],color=twocol)


plt.ylabel("Frequency")


plt.xlabel("Period Game Was Won In")


plt.title("Number of Games Won During Regulation v. PowerPlay From 2000-2019")
```

Out[10]: Text(0.5, 1.0, 'Number of Games Won During Regulation v. PowerPlay From 2000-2019')



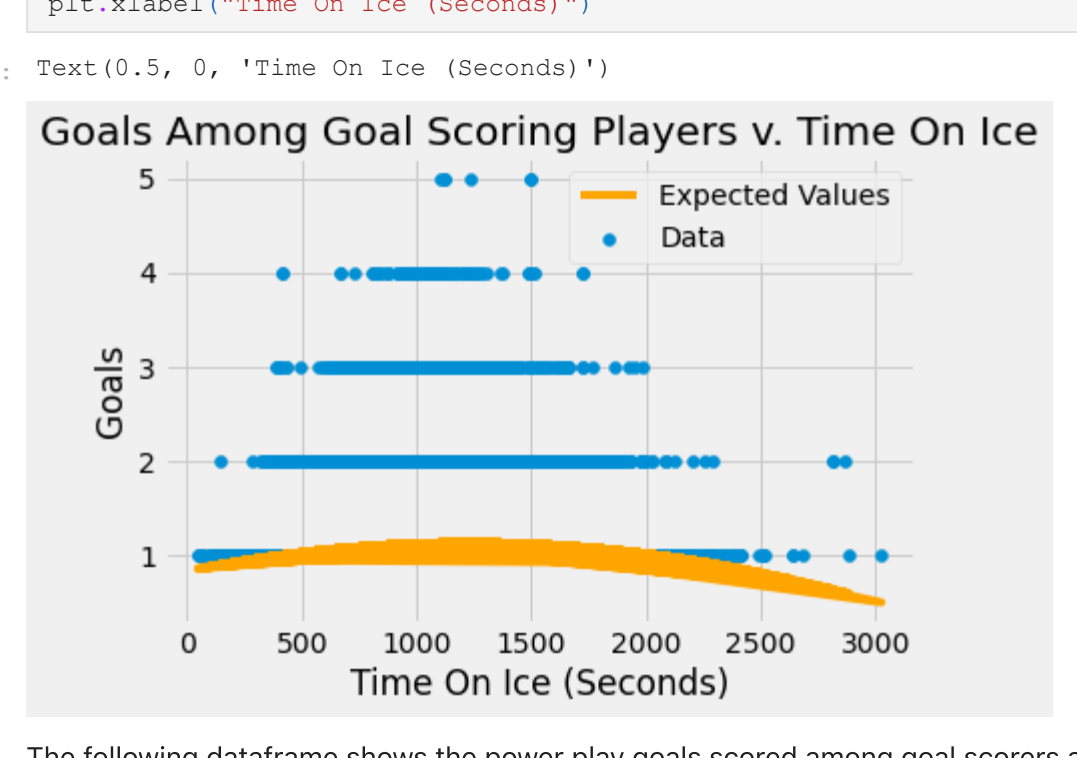
The following dataframe shows the time on ice vs goals scored for each player in the elague during this time period.

In [11]:
```
timeonice = game_skater_stats["timeOnIce"]

goals = game_skater_stats['goals']

frame7 = { 'timeonice': timeonice, 'goals': goals }

result7 = pd.DataFrame(frame7)

result71 = result7[goals>0] #Shows goals among goal scorers and time on ice

print(result71)
```

```
        timeonice  goals
12           1092      1
14           1033      2
17           1326      1
18            723      1
20            652      1
...           ...    ...
945795       1556      1
945800        828      1
945807       1136      1
945811        983      1
945820       1076      1

[130814 rows x 2 columns]
```

In [12]:
```
# Plots data and expected values for time on ice and goals scored among goal scorers

plt.scatter(result71['timeonice'],result71['goals'],label='Data')

poly_parameters = np.polyfit(result71['timeonice'], result71['goals'], 2)


# This uses polyfit to create the expected values

my_poly_function = np.poly1d(poly_parameters)

expected_y_poly = my_poly_function(result71['timeonice'])


plt.plot(result71['timeonice'],expected_y_poly,color='orange',label='Expected Values')

plt.legend()

plt.title("Goals Among Goal Scoring Players v. Time On Ice")

plt.ylabel("Goals")

plt.xlabel("Time On Ice (Seconds)")
```

Out[12]: Text(0.5, 0, 'Time On Ice (Seconds)')



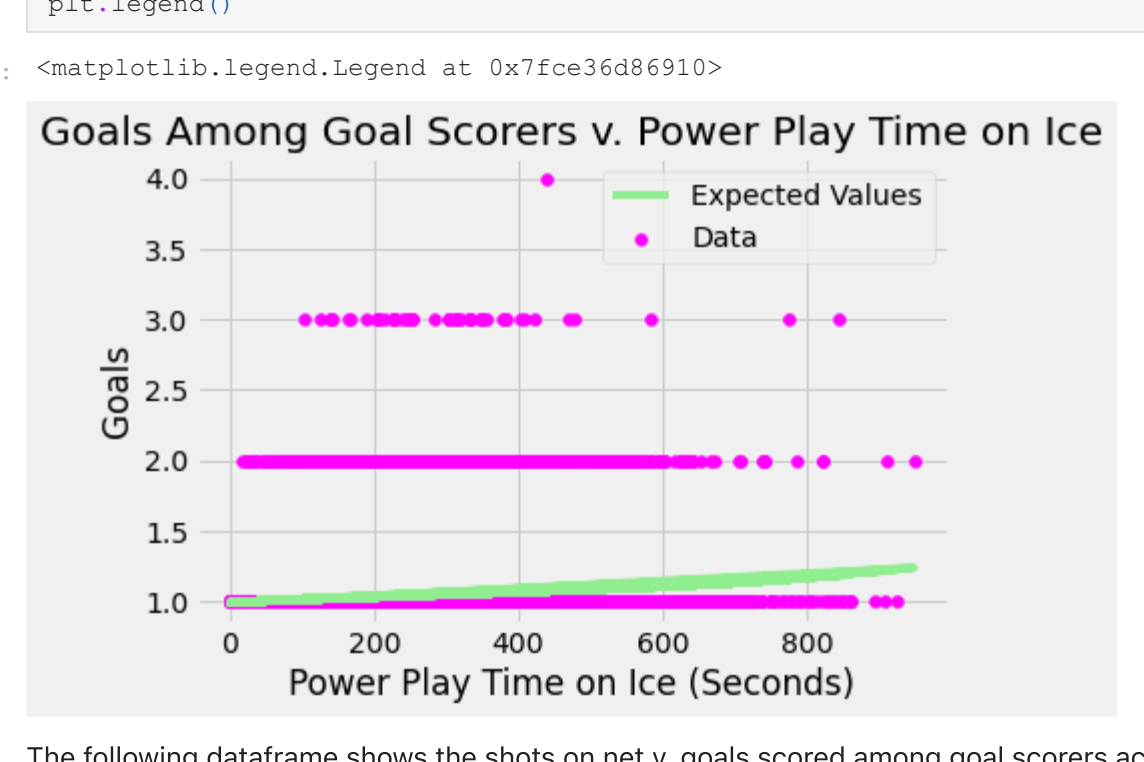
The following dataframe shows the power play goals scored among goal scorers and players' respective power play time on ice during the time period in the data set.

In [13]:
```
pptimeonice = game_skater_stats["powerPlayTimeOnIce"]

goals2 = game_skater_stats["powerPlayGoals"]


frame8 = { 'powerPlayTimeOnIce': pptimeonice, 'powerPlayGoals': goals2 }

frame81 = pd.DataFrame(frame8)


frame82 = frame81[goals2>0] #This shows power play goals among goal scorers


print(frame82)
```

```
        powerPlayTimeOnIce  powerPlayGoals
14                     170               1
17                     217               1
18                      96               1
32                     154               1
48                     156               1
...                    ...             ...
945517                 103               1
945520                  23               1
945528                 307               1
945686                 130               1
945722                 130               1

[33741 rows x 2 columns]
```

In [14]:
```
# This plots the goals among goal scorers vs power play time on ice

# This creates the expected values
poly_parameters2 = np.polyfit(frame82["powerPlayTimeOnIce"], frame82['powerPlayGoals'], 3)

my_poly_function2 = np.poly1d(poly_parameters2)

expected_y_poly2 = my_poly_function2(frame82['powerPlayTimeOnIce'])




plt.scatter(frame82['powerPlayTimeOnIce'],frame82['powerPlayGoals'],color='magenta',label='Data')
plt.plot(frame82['powerPlayTimeOnIce'],expected_y_poly2,color='lightgreen',label='Expected Values')

plt.title("Goals Among Goal Scorers v. Power Play Time on Ice")


plt.ylabel("Goals")

plt.xlabel("Power Play Time on Ice (Seconds)")



plt.legend()
```

Out[14]: <matplotlib.legend.Legend at 0x7fce36d86910>

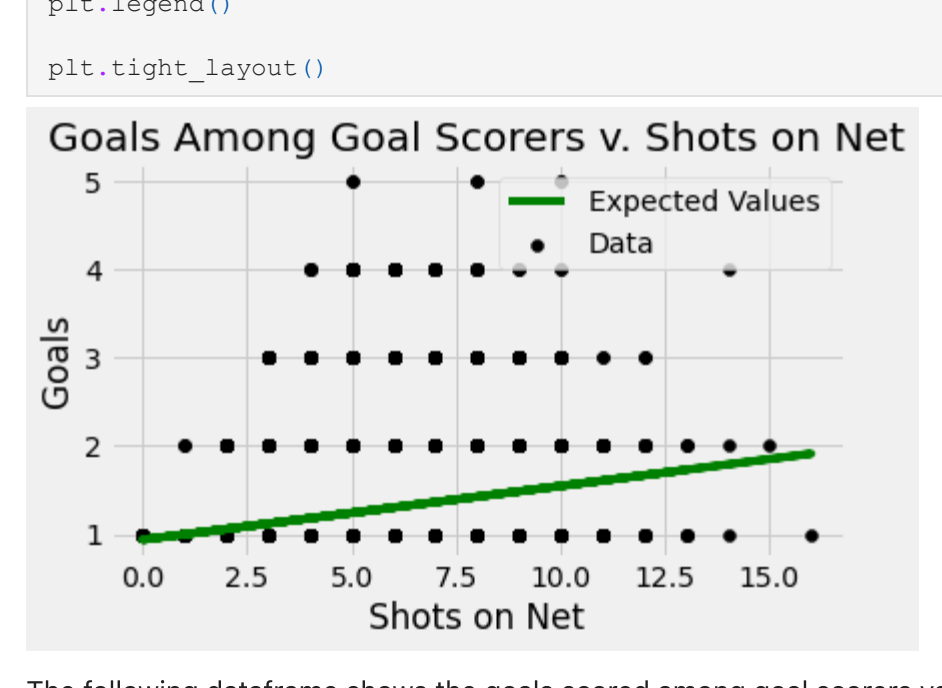


The following dataframe shows the shots on net v. goals scored among goal scorers across all the seasons in the data set.

In [15]:
```
shots = game_skater_stats['shots']

goals3 = game_skater_stats['goals']


frame9 = { 'shots': shots, 'goals3': goals3 }

result9 = pd.DataFrame(frame9)


result91 = result9[goals3>0] # This shows the shots and goals scored among goal scorers

print(result91)
```

```
        shots  goals3
12          2       1
14          3       2
17          6       1
18          2       1
20          1       1
...       ...     ...
945795      3       1
945800      2       1
945807      4       1
945811      3       1
945820      3       1

[130814 rows x 2 columns]
```

In [16]:
```
# This creates the data and expected values graphs for shots on net and goals scored among goal scorers

# This creates the expected values using polyfit

poly_parameters3 = np.polyfit(result91['shots'], result91['goals3'], 2)

my_poly_function3 = np.poly1d(poly_parameters3)


expected_y_poly3 = my_poly_function3(result91['shots'])




plt.scatter(result91['shots'],result91['goals3'],color='black',label="Data")
plt.plot(result91['shots'],expected_y_poly3,color='green',label="Expected Values")


plt.title("Goals Among Goal Scorers v. Shots on Net")

plt.ylabel("Goals")

plt.xlabel("Shots on Net")

plt.legend()

plt.tight_layout()
```

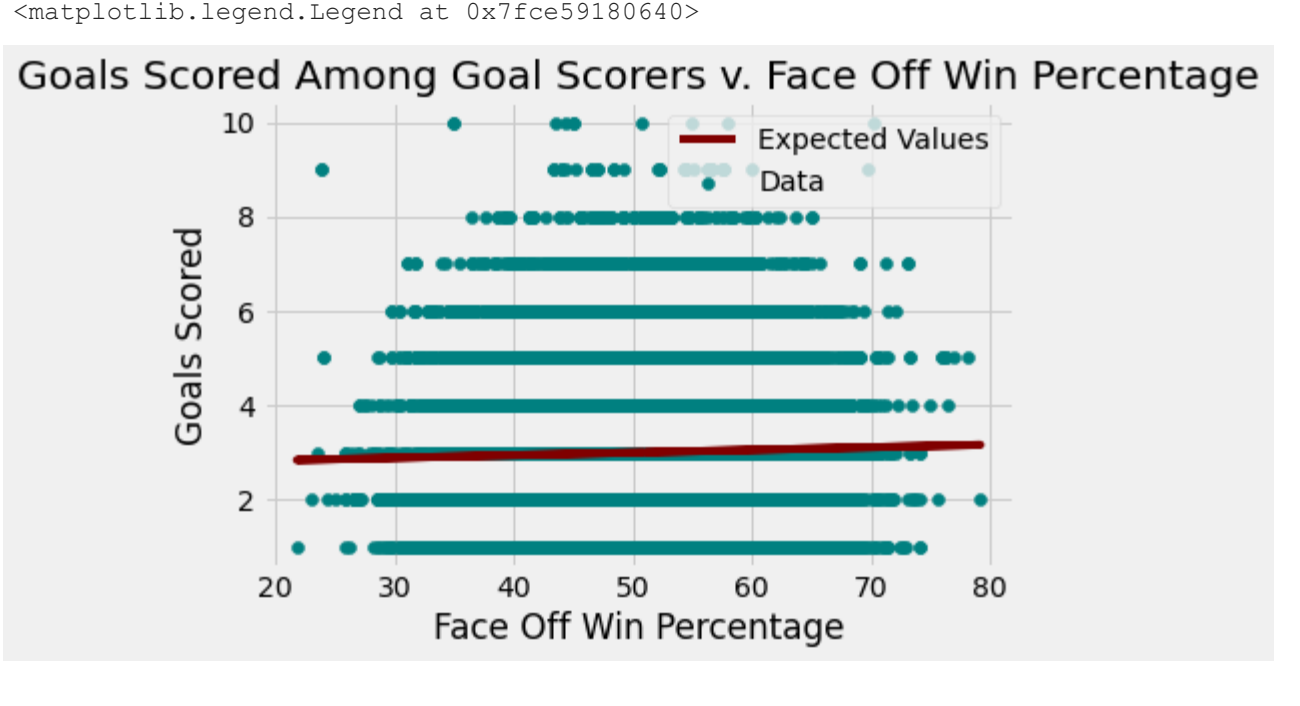


The following dataframe shows the goals scored among goal scorers vs their respective face off win percentage across the various seasons reprented in the data set.

In [17]:
```
faceOffWinPercentage = game_teams_stats["faceOffWinPercentage"]

goals2 = game_teams_stats["goals"]


frame10 = { 'faceOffWinPercentage': faceOffWinPercentage, 'goals2': goals2 }

result10 = pd.DataFrame(frame10)


result101 = result10[goals2>0]


result102 = result101.dropna() #this shows face off win percentage vs goals scored among goal scorers


print(result102)
```

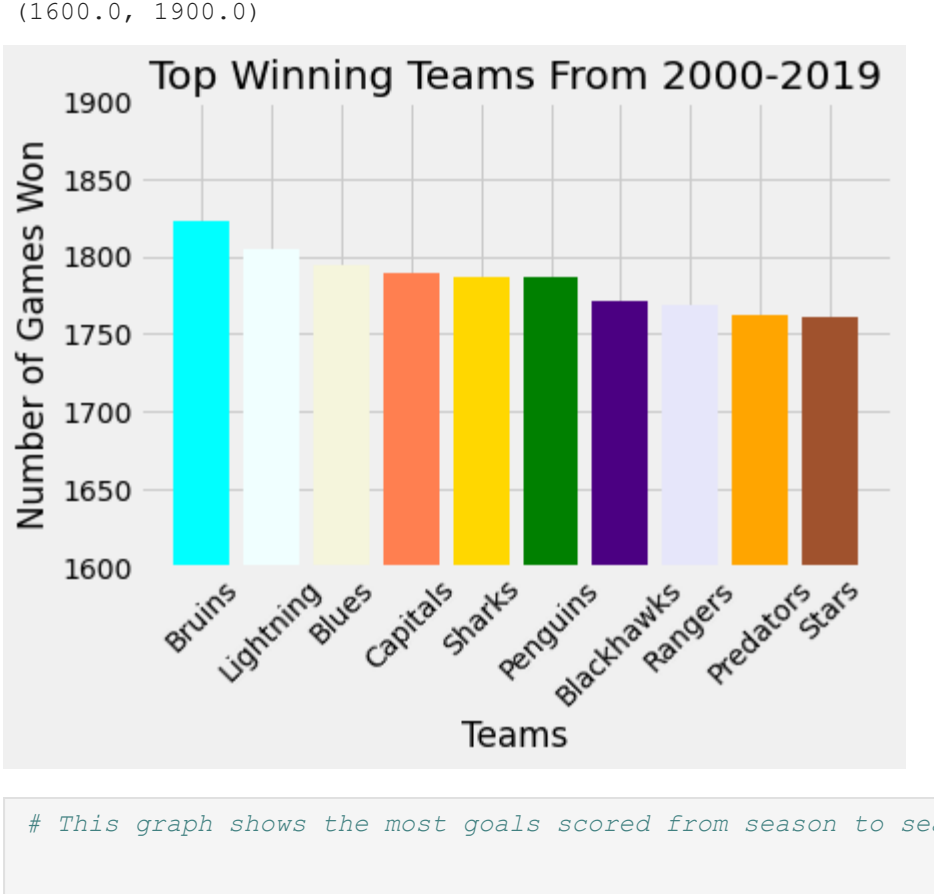
```
       faceOffWinPercentage  goals2
0                      50.9     4.0
1                      49.1     7.0
2                      43.8     4.0
3                      56.2     3.0
4                      45.7     4.0
...                     ...     ...
52605                  58.7     1.0
52606                  49.0     4.0
52607                  51.0     1.0
52608                  49.0     4.0
52609                  51.0     1.0

[28481 rows x 2 columns]
```

In [18]:
```python
# This shows the goals scored among goal scorers v. face off win percentage with expected values with a regress
# This uses polyfit to create the expected values
poly_parameters4 = np.polyfit(result102['faceOffWinPercentage'], result102['goals2'], 1)


my_poly_function4 = np.poly1d(poly_parameters4)

expected_y_poly4 = my_poly_function4(result102['faceOffWinPercentage'])


plt.scatter(result102["faceOffWinPercentage"],result102["goals2"],color='teal',label="Data")


plt.plot(result102['faceOffWinPercentage'],expected_y_poly4,color='maroon',label='Expected Values')


plt.title("Goals Scored Among Goal Scorers v. Face Off Win Percentage")

plt.ylabel("Goals Scored")

plt.xlabel("Face Off Win Percentage")


plt.legend()
```
Out[18]: <matplotlib.legend.Legend at 0x7fce59180640>

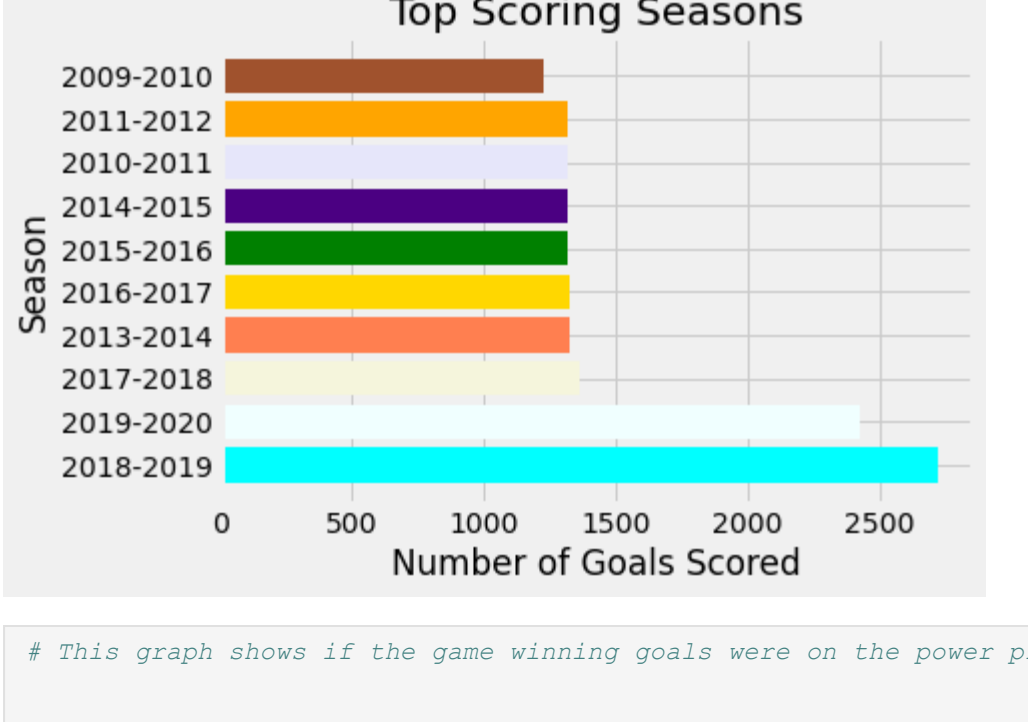


Results

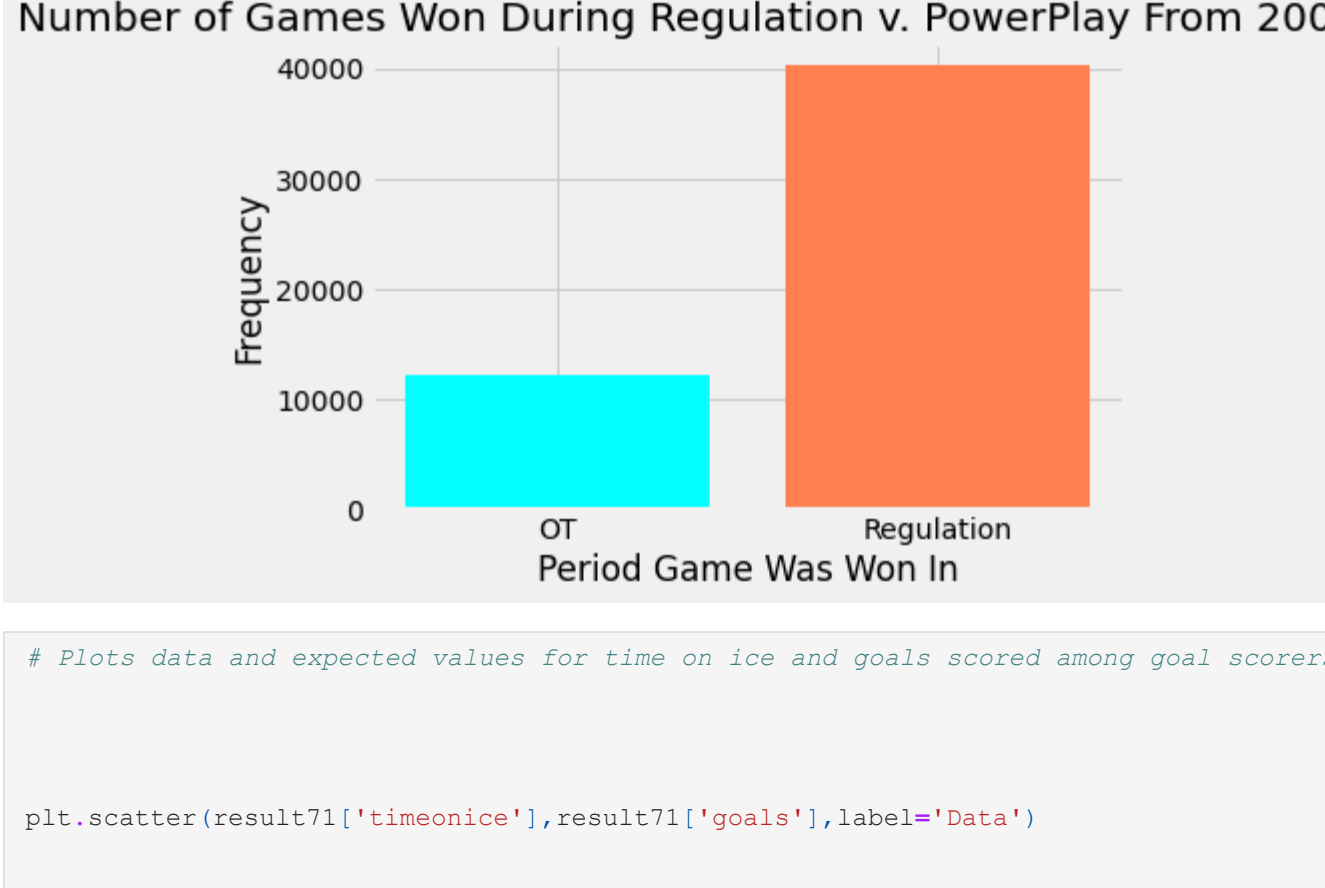
In [19]:
```python
# This plot shows goals scored over time


plt.xticks(rotation = 45)

plt.bar(result2["team_names"],lnwon,color=colors)


plt.title("Top Winning Teams From 2000-2019")

plt.xlabel("Teams")

plt.ylabel("Number of Games Won")



plt.ylim(1600, 1900)
```
Out[19]: (1600.0, 1900.0)



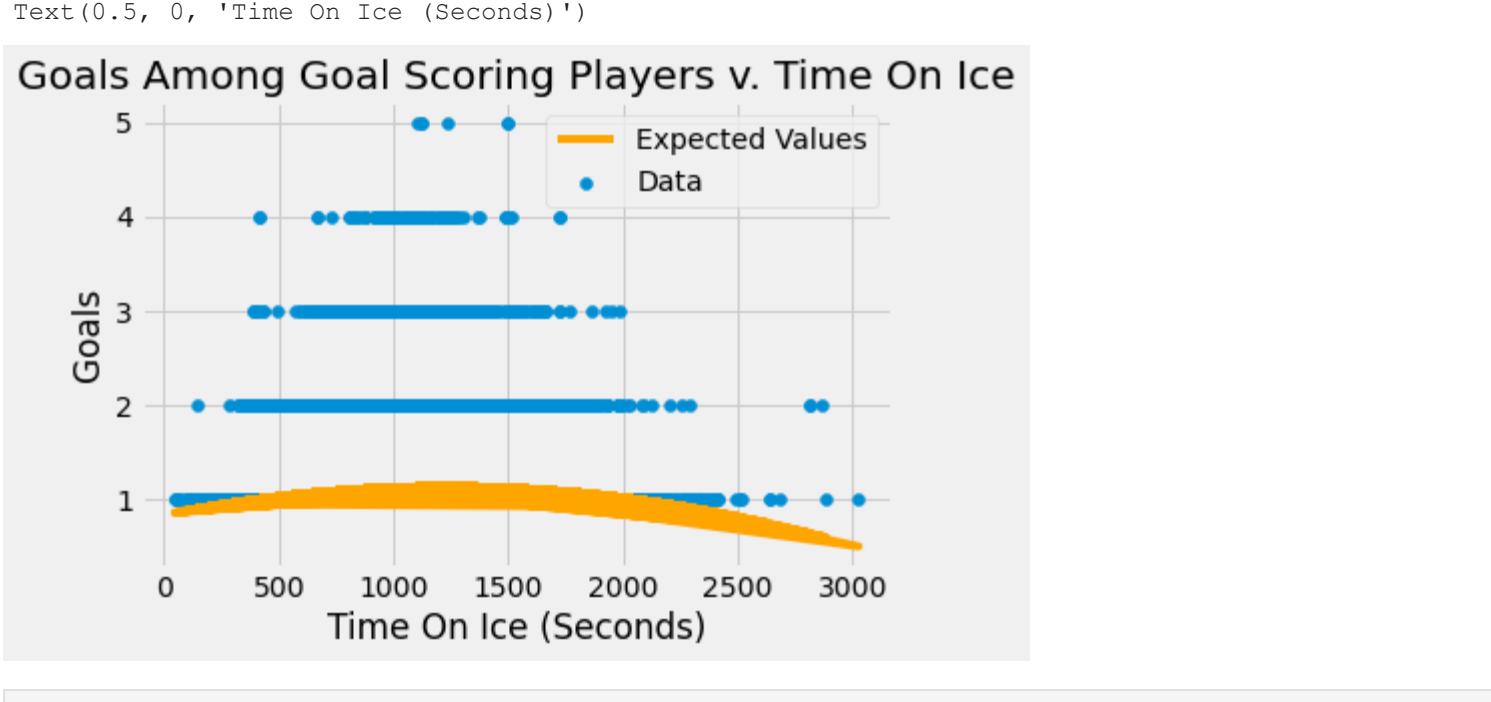
In [20]:
```python
# This graph shows the most goals scored from season to season


season_names = ['2018-2019','2019-2020','2017-2018','2013-2014','2016-2017','2015-2016','2014-2015',
                '2010-2011','2011-2012','2009-2010']



lntotalgoals = np.log10(result33['totalgoals'])

plt.barh(season_names,result33['totalgoals'],color=colors)


plt.xlabel("Number of Goals Scored")

plt.ylabel("Season")


plt.title("Top Scoring Seasons")
```
Out[20]: Text(0.5, 1.0, 'Top Scoring Seasons')



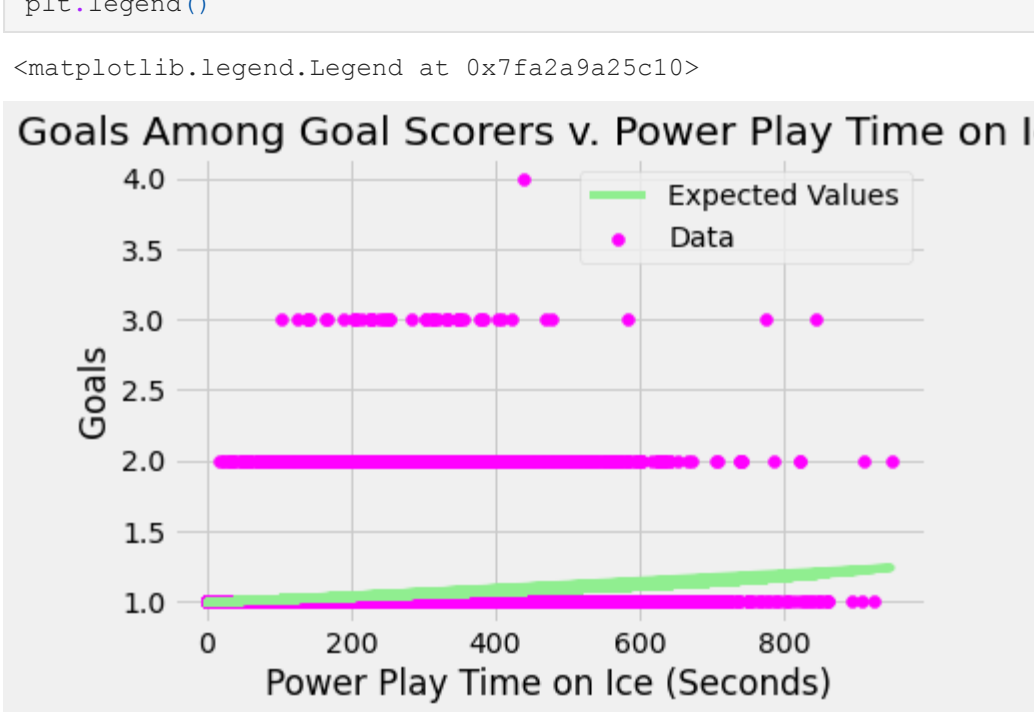
In [21]:
```python
# This graph shows if the game winning goals were on the power play across the various seasons


plt.bar(gamewinninggoalsnames,result52['strength'],color=noyescol)


plt.xlabel("If Goal Was Game Winning")


plt.ylabel("Frequency")


plt.title("Number of Game Winning Goals on PowerPlay From 2000-2019")
```
Out[21]: Text(0.5, 1.0, 'Number of Game Winning Goals on PowerPlay From 2000-2019')



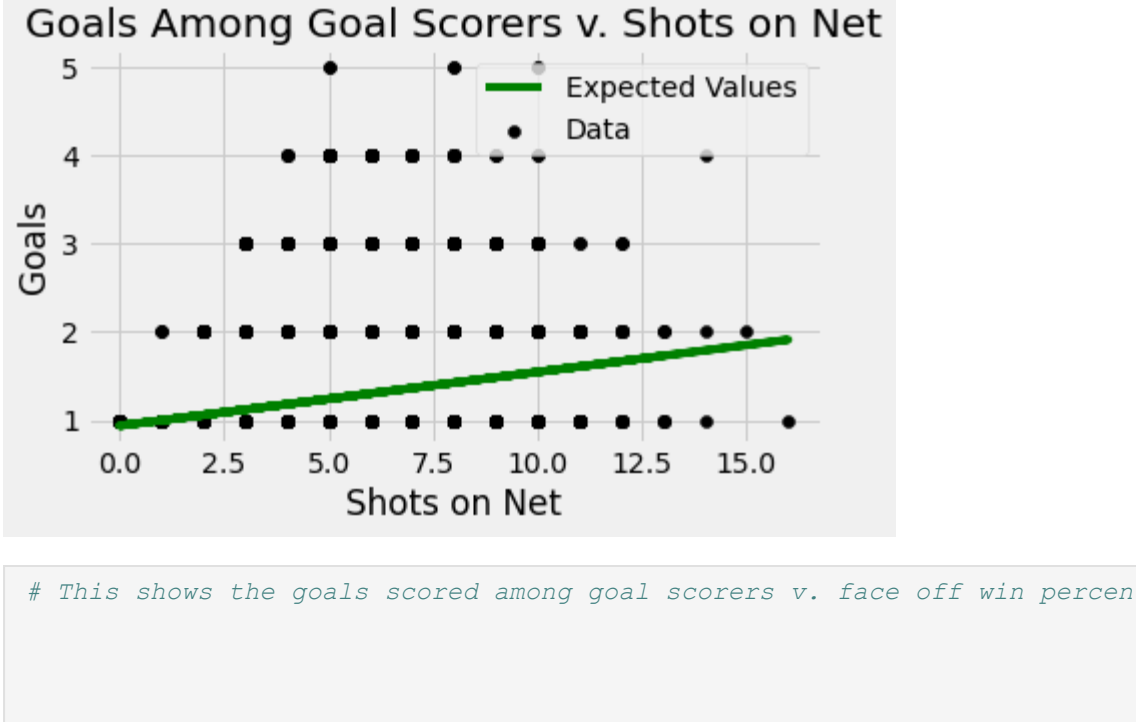
In [22]:
```python
# This graph shows the number of games won during the Powerplay vs during regulation in the time period


plt.bar(final_game_team_resultdfnames,result62['won4'],color=twocol)


plt.ylabel("Frequency")

plt.xlabel("Period Game Was Won In")

plt.title("Number of Games Won During Regulation v. PowerPlay From 2000-2019")
```
Out[22]: Text(0.5, 1.0, 'Number of Games Won During Regulation v. PowerPlay From 2000-2019')



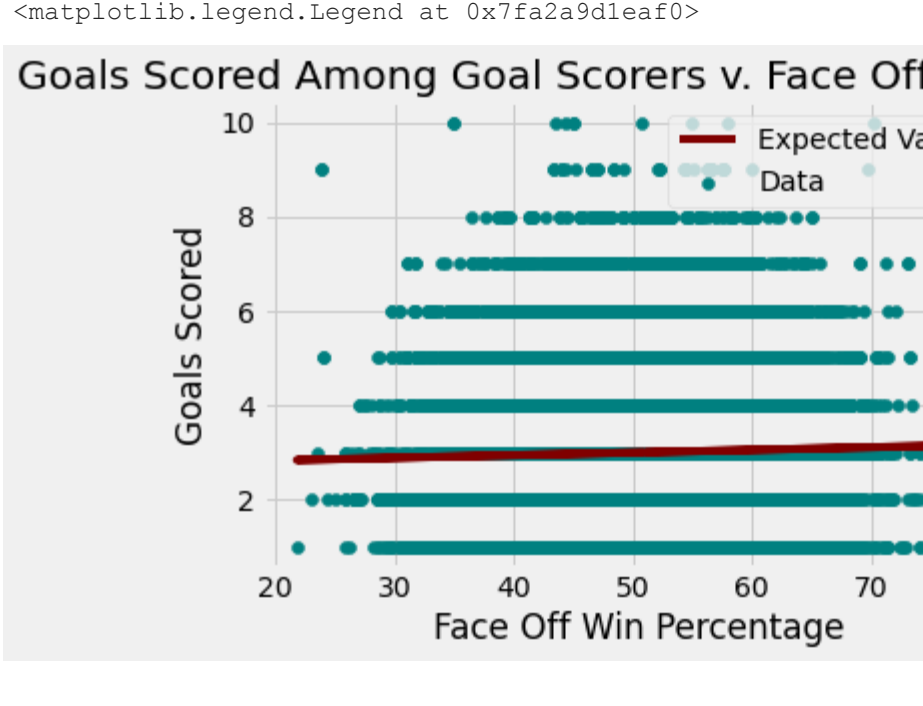
In [23]:
```python
# Plots data and expected values for time on ice and goals scored among goal scorers


plt.scatter(result71['timeonice'],result71['goals'],label='Data')


plt.plot(result71['timeonice'],expected_y_poly,color='orange',label='Expected Values')

plt.legend()

plt.title("Goals Among Goal Scoring Players v. Time On Ice")

plt.ylabel("Goals")

plt.xlabel("Time On Ice (Seconds)")
```
Out[23]: Text(0.5, 0, 'Time On Ice (Seconds)')



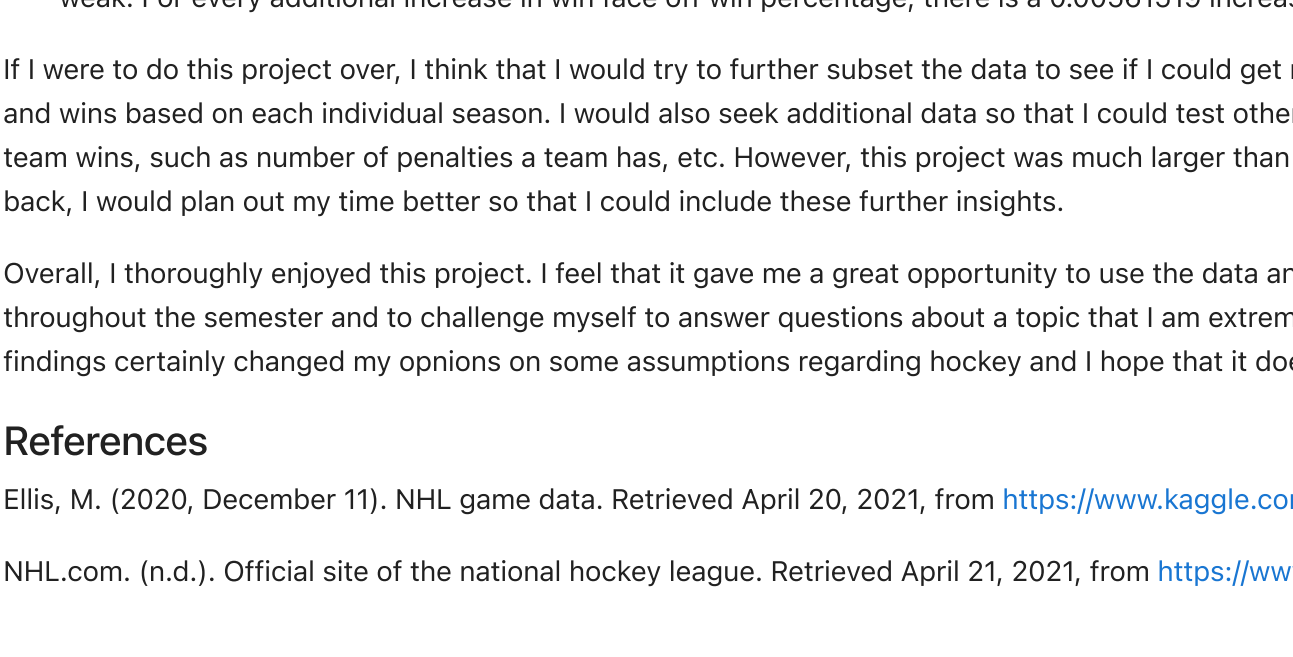
In [40]:
```python
# This plots the goals among goal scorers vs power play time on ice


plt.scatter(frame82['powerPlayTimeOnIce'],frame82['powerPlayGoals'],color='magenta',label='Data')

plt.plot(frame82['powerPlayTimeOnIce'],expected_y_poly2,color='lightgreen',label='Expected Values')


plt.title("Goals Among Goal Scorers v. Power Play Time on Ice")


plt.ylabel("Goals")

plt.xlabel("Power Play Time on Ice (Seconds)")


plt.legend()
```
Out[40]: <matplotlib.legend.Legend at 0x7fa2a9a25c10>



In [24]:
```python
# This creates the data and expected values graphs for shots on net and goals scored among goal scorers
plt.scatter(result91['shots'],result91['goals3'],color='black',label="Data")
plt.plot(result91['shots'],expected_y_poly3,color='green',label="Expected Values")


plt.title("Goals Among Goal Scorers v. Shots on Net")


plt.ylabel("Goals")

plt.xlabel("Shots on Net")

plt.legend()
plt.tight_layout()
```



In [42]:
```python
# This shows the goals scored among goal scorers v. face off win percentage with expected values with a regress


plt.scatter(result102["faceOffWinPercentage"],result102["goals2"],color='teal',label="Data")


plt.plot(result102['faceOffWinPercentage'],expected_y_poly4,color='maroon',label='Expected Values')


plt.title("Goals Scored Among Goal Scorers v. Face Off Win Percentage")

plt.ylabel("Goals Scored")

plt.xlabel("Face Off Win Percentage")


plt.legend()
```
Out[42]: <matplotlib.legend.Legend at 0x7fa2a9d1eaf0>



Discussion and Conclusion

Through this data analysis, I learned the following results based on the questions that I asked:

- Boston Bruins won most games which ended up being 1822, followed by the Tampa Bay Lightning which was 1804.
- 2018-2019 season was highest scoring season. The total goals were 2720. The next higheset scoring season was 2019-2020. The goals scored were 2425.
- Game winning goals are rarely scored on the Power Play compared to being scored when teams are even.The amount of game winning goals was 5229 on the power play. This is compared to for the alternative which is 29295 games.
- 12256 games were won in OT vs 40306 games that were won in regulation.
- There seems to be a negative parabolic relationship between time on ice and goal scoring. However, the relationship is weak. For every 1 second increase in time on ice there is a -1.96336612e-07 decrease in goals scored.
- There seems to be a slight positive correlation between power play time on ice and goal scoring. However, the relationship is weak. For every 1 second increase in power play time on ice there is a 3.78469208e-10 increase in goals scored.
- There seems to be a stronger positive correlation between shots on net and goal scoring. However, the relationship is relatively weak. For every 1 additional shot, there is a increase 3.57769579e-05 in goals scored.
- There seems to be a slight positive correlation between face off win percentages and goal scoring. However, the relationship is weak. For every additional increase in win face off win percentage, there is a 0.00561519 increase in goal scoring.

If I were to do this project over, I think that I would try to further subset the data to see if I could get more insights into goal scoring and wins based on each individual season. I would also seek additional data so that I could test other impacts on goal scoring and team wins, such as number of penalties a team has, etc. However, this project was much larger than I initially planned. If I could go back, I would plan out my time better so that I could include these further insights.

Overall, I thoroughly enjoyed this project. I feel that it gave me a great opportunity to use the data analysis skills I have honed throughout the semester and to challenge myself to answer questions about a topic that I am extremely fond of. I know that my findings certainly changed my opnions on some assumptions regarding hockey and I hope that it does the same for others.

References

Ellis, M. (2020, December 11). NHL game data. Retrieved April 20, 2021, from https://www.kaggle.com/martinellis/nhl-game-data

NHL.com. (n.d.). Official site of the national hockey league. Retrieved April 21, 2021, from https://www.nhl.com/