Week 2 Assignment

January 6, 2023

0.1 Part 1- Data Coding and Merging

```
In [152]: #Import Libraries

          import pandas as pd
          import numpy as np
          import datetime
          import matplotlib.pyplot as plt
          import seaborn as sns
```

Import the “NHL_Team.csv” data file and name the dataframe as “NHL_Team” in Jupyter Notebook.

```
In [153]: #Import NHL Team Data and Display
          NHL_Team=pd.read_csv("Assignment Data/Week 2/NHL_team.csv")
          NHL_Team.head()

Out[153]:    Unnamed: 0  tid                 name        tname       lname tricode abbr  \
          0           1    1  Toronto Maple Leafs  Maple Leafs     Toronto     TOR  TOR
          1           2    2   Montréal Canadiens    Canadiens    Montréal     MTL  MTL
          2           3    4        Winnipeg Jets         Jets    Winnipeg     WPG  WPG
          3           4    5  Washington Capitals     Capitals  Washington     WSH  WSH
          4           5    6   Chicago Blackhawks   Blackhawks     Chicago     CHI  CHI

                  sname
          0     Toronto
          1    Montréal
          2    Winnipeg
          3  Washington
          4     Chicago
```

a) Delete the following variables: “Unnamed:0”, “abbr”, “tname”, “lname”, and “sname”.

```
In [154]: NHL_Team.shape

Out[154]: (33, 8)

In [155]: NHL_Team.drop(['Unnamed: 0', 'abbr', 'tname', 'lname', 'sname'], axis=1, inplace=Tru
          NHL_Team.head()
```

```
Out[155]:    tid                 name tricode
          0    1  Toronto Maple Leafs     TOR
          1    2   Montréal Canadiens     MTL
          2    4        Winnipeg Jets     WPG
          3    5  Washington Capitals     WSH
          4    6   Chicago Blackhawks     CHI
```

b) Rename the variable "name" to "team_name".

```
In [156]: NHL_Team.rename(columns={'name':'team_name'}, inplace=True)
          NHL_Team.head()
```

```
Out[156]:    tid            team_name tricode
          0    1  Toronto Maple Leafs     TOR
          1    2   Montréal Canadiens     MTL
          2    4        Winnipeg Jets     WPG
          3    5  Washington Capitals     WSH
          4    6   Chicago Blackhawks     CHI
```

Import the "NHL_competition.csv" data file and name the dataframe as "NHL_Competition" in Jupyter Notebook.

```
In [157]: #Import NHL Competition Data and Display Head
          NHL_Competition=pd.read_csv("Assignment Data/Week 2/NHL_competition.csv")
          NHL_Competition.head()
```

```
Out[157]:    Unnamed: 0  comp_id  year  type                     name  tz  start  end
          0           1        1  2013     2  2013 NHL Regular Season  ET    NaN  NaN
          1           2        2  2017     2  2017 NHL Regular Season  ET    NaN  NaN
          2           3     2453  2013     3         2013 NHL Playoff  ET    NaN  NaN
          3           4     2541  2017     3         2017 NHL Playoff  ET    NaN  NaN
          4           5     2661  2012     2  2012 NHL Regular Season  ET    NaN  NaN
```

a) Delete the following variables: "Unnamed: 0", "tz", "start", and "end"

```
In [158]: NHL_Competition.drop(['Unnamed: 0', 'tz', 'start', 'end'], axis=1, inplace=True)
          NHL_Competition.head()
```

```
Out[158]:    comp_id  year  type                     name
          0        1  2013     2  2013 NHL Regular Season
          1        2  2017     2  2017 NHL Regular Season
          2     2453  2013     3         2013 NHL Playoff
          3     2541  2017     3         2017 NHL Playoff
          4     2661  2012     2  2012 NHL Regular Season
```

b) Rename the variable "name" to "competition_name".

```
In [159]: NHL_Competition.rename(columns={'name':'competition_name'}, inplace=True)
          NHL_Competition.head()
```

```
Out[159]:    comp_id  year  type        competition_name
          0        1  2013     2  2013 NHL Regular Season
          1        2  2017     2  2017 NHL Regular Season
          2     2453  2013     3         2013 NHL Playoff
          3     2541  2017     3         2017 NHL Playoff
          4     2661  2012     2  2012 NHL Regular Season
```

Import the "NHL_game.csv" data file and name the dataframe as "NHL_Game" in Jupyter Notebook.

```
In [160]: #Import NHL Game Data and Display Head
          NHL_Game=pd.read_csv("Assignment Data/Week 2/NHL_game.csv")
          NHL_Game.head()
```

```
Out[160]:    X  gid  comp_id        date  ascore  hscore  period  status home_away  tid
          0  1   37        2   10/7/2017     NaN     NaN     NaN     NaN      away   25
          1  2   67        2   10/9/2017     NaN     NaN     NaN     NaN      away   29
          2  3  154        1  10/14/2013     NaN     NaN     NaN     NaN      away   29
          3  4  278        1  10/24/2013     NaN     NaN     NaN     NaN      away   53
          4  5  291        1  10/25/2013     NaN     NaN     NaN     NaN      away    5
```

a) Delete the following variables: "X", "period", and "status".

```
In [161]: NHL_Game.drop(['X', 'period', 'status'], axis=1, inplace=True)
          NHL_Game.head()
```

```
Out[161]:    gid  comp_id        date  ascore  hscore home_away  tid
          0   37        2   10/7/2017     NaN     NaN      away   25
          1   67        2   10/9/2017     NaN     NaN      away   29
          2  154        1  10/14/2013     NaN     NaN      away   29
          3  278        1  10/24/2013     NaN     NaN      away   53
          4  291        1  10/25/2013     NaN     NaN      away    5
```

b) Merge the dataframe "NHL_Team" into the dataframe "NHL_Game" by "tid." Continue to name the merged dataframe as "NHL_Game."

```
In [162]: NHL_Game=pd.merge(NHL_Team, NHL_Game, on=['tid'])
          NHL_Game.head()
```

```
Out[162]:    tid            team_name tricode   gid  comp_id        date  ascore  \
          0    1  Toronto Maple Leafs     TOR   741        1  11/28/2013     NaN
          1    1  Toronto Maple Leafs     TOR   782        1   12/1/2013     NaN
          2    1  Toronto Maple Leafs     TOR  5225     5181   4/25/2017     NaN
          3    1  Toronto Maple Leafs     TOR  6557     5385    1/7/2016     NaN
          4    1  Toronto Maple Leafs     TOR  6914     5385    2/7/2016     NaN

             hscore home_away
          0     NaN      away
          1     NaN      away
```

```
2     NaN      away
3     NaN      away
4     NaN      away
```

c) Merge the dataframe “NHL_Competition” into the dataframe “NHL_Game” by “comp_id.” Continue to name the merged dataframe as “NHL_Game.”

```
In [163]: NHL_Game=pd.merge(NHL_Competition, NHL_Game, on=['comp_id'])
          NHL_Game.head()
```

```
Out[163]:    comp_id  year  type          competition_name  tid            team_name  \
          0        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          1        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          2        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          3        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          4        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs

            tricode   gid        date  ascore  hscore home_away
          0     TOR   741  11/28/2013     NaN     NaN      away
          1     TOR   782   12/1/2013     NaN     NaN      away
          2     TOR  1003  12/17/2013     1.0     3.0      away
          3     TOR  1552   1/26/2014     4.0     5.0      away
          4     TOR  1811    3/2/2014     3.0     4.0      away
```

d) In the merged “NHL_Game” dataframe, create a variable “hgd” to indicate the goal difference between home and away score (hscore – ascore) and delete observations with missing value in the variable “hgd”.

```
In [164]: NHL_Game['hgd']= NHL_Game['hscore'] - NHL_Game['ascore']
          NHL_Game.head()
```

```
Out[164]:    comp_id  year  type          competition_name  tid            team_name  \
          0        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          1        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          2        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          3        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          4        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs

            tricode   gid        date  ascore  hscore home_away  hgd
          0     TOR   741  11/28/2013     NaN     NaN      away  NaN
          1     TOR   782   12/1/2013     NaN     NaN      away  NaN
          2     TOR  1003  12/17/2013     1.0     3.0      away  2.0
          3     TOR  1552   1/26/2014     4.0     5.0      away  1.0
          4     TOR  1811    3/2/2014     3.0     4.0      away  1.0
```

```
In [165]: NHL_Game=NHL_Game[pd.notnull(NHL_Game["hgd"])]
          NHL_Game.shape
```

```
Out[165]: (18506, 13)
```

e) Drop all observations with missing values, if there is still any, from the "NHL_Game" dataframe.

```
In [166]: NHL_Game.dropna()
          NHL_Game.shape
```

```
Out[166]: (18506, 13)
```

```
In [167]: NHL_Game.head()
```

```
Out[167]:    comp_id  year  type        competition_name  tid            team_name  \
          2        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          3        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          4        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          5        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          6        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs

            tricode   gid        date  ascore  hscore home_away  hgd
          2     TOR  1003  12/17/2013     1.0     3.0      away  2.0
          3     TOR  1552   1/26/2014     4.0     5.0      away  1.0
          4     TOR  1811    3/2/2014     3.0     4.0      away  1.0
          5     TOR  1940   3/11/2014     3.0     1.0      away -2.0
          6     TOR  1522   1/24/2014     1.0     7.0      away  6.0
```

g) Convert the type of the "date" variable from "object" to "datetime."

```
In [168]: NHL_Game['date']=pd.to_datetime(NHL_Game['date'])
          NHL_Game['date'].head()
```

```
Out[168]: 2   2013-12-17
          3   2014-01-26
          4   2014-03-02
          5   2014-03-11
          6   2014-01-24
          Name: date, dtype: datetime64[ns]
```

Quiz Question 1 What are the number of observations and the number of variables in the NHL_Game dataframe after performing the first 7 steps?

```
In [169]: # What are the number of observations and the number of variables
          # in the NHL_Game dataframe after performing the first 7 steps?
          NHL_Game.shape
```

```
Out[169]: (18506, 13)
```

h) Sort the NHL games by "date" and show the first 15 observations.

```
In [170]: NHL_Game.sort_values(by=['date'], ascending=[False]).head(15)
```

```
Out[170]:       comp_id  year  type  competition_name  tid              team_name  \
          5383     2541  2017     3  2017 NHL Playoff   59  Vegas Golden Knights
          5212     2541  2017     3  2017 NHL Playoff    5   Washington Capitals
          5228     2541  2017     3  2017 NHL Playoff    5   Washington Capitals
          5367     2541  2017     3  2017 NHL Playoff   59  Vegas Golden Knights
          5369     2541  2017     3  2017 NHL Playoff   59  Vegas Golden Knights
          5229     2541  2017     3  2017 NHL Playoff    5   Washington Capitals
          5214     2541  2017     3  2017 NHL Playoff    5   Washington Capitals
          5384     2541  2017     3  2017 NHL Playoff   59  Vegas Golden Knights
          5217     2541  2017     3  2017 NHL Playoff    5   Washington Capitals
          5388     2541  2017     3  2017 NHL Playoff   59  Vegas Golden Knights
          5215     2541  2017     3  2017 NHL Playoff    5   Washington Capitals
          5331     2541  2017     3  2017 NHL Playoff   25   Tampa Bay Lightning
          5230     2541  2017     3  2017 NHL Playoff    5   Washington Capitals
          5320     2541  2017     3  2017 NHL Playoff   25   Tampa Bay Lightning
          5209     2541  2017     3  2017 NHL Playoff    4         Winnipeg Jets

               tricode   gid        date  ascore  hscore home_away  hgd
          5383     VGK  2730 2018-06-08     4.0     3.0      home -1.0
          5212     WSH  2730 2018-06-08     4.0     3.0      away -1.0
          5228     WSH  2727 2018-06-05     2.0     6.0      home  4.0
          5367     VGK  2727 2018-06-05     2.0     6.0      away  4.0
          5369     VGK  2725 2018-06-03     1.0     3.0      away  2.0
          5229     WSH  2725 2018-06-03     1.0     3.0      home  2.0
          5214     WSH  2723 2018-05-31     3.0     2.0      away -1.0
          5384     VGK  2723 2018-05-31     3.0     2.0      home -1.0
          5217     WSH  2720 2018-05-29     4.0     6.0      away  2.0
          5388     VGK  2720 2018-05-29     4.0     6.0      home  2.0
          5215     WSH  2706 2018-05-24     4.0     0.0      away -4.0
          5331     TBL  2706 2018-05-24     4.0     0.0      home -4.0
          5230     WSH  2703 2018-05-22     0.0     3.0      home  3.0
          5320     TBL  2703 2018-05-22     0.0     3.0      away  3.0
          5209     WPG  2716 2018-05-20     2.0     1.0      home -1.0
```

```
In [171]: NHL_Game['date'].describe()
```

```
Out[171]: count                   18506
          unique                   1607
          top       2017-11-23 00:00:00
          freq                       30
          first     2010-10-07 00:00:00
          last      2018-06-08 00:00:00
          Name: date, dtype: object
```

i) Create two dataframes that separate the "NHL_Game" dataframe by home and away games. Name them "NHL_Home" and "NHL_Away", respectively.

```
a) Rename variables:
    i) For away games, rename ascore to goals_for; rename hscore to goals_against
    ii) For home games, rename hscore to goals_for; rename ascore to goals_against
b) Create a win variable that equals to 1 if the team won the game; 0 if the team lost the gar
```

```
In [172]: # Renaming columns for Away Games
          NHL_Away=NHL_Game[NHL_Game.home_away == 'away']
          NHL_Away=NHL_Away.rename(columns={'ascore':'goals_for','hscore':'goals_against'})

          NHL_Away.head()
```

```
Out[172]:    comp_id  year  type         competition_name  tid           team_name  \
          2        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          3        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          4        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          5        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          6        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs

            tricode   gid       date  goals_for  goals_against home_away  hgd
          2     TOR  1003 2013-12-17        1.0            3.0      away  2.0
          3     TOR  1552 2014-01-26        4.0            5.0      away  1.0
          4     TOR  1811 2014-03-02        3.0            4.0      away  1.0
          5     TOR  1940 2014-03-11        3.0            1.0      away -2.0
          6     TOR  1522 2014-01-24        1.0            7.0      away  6.0
```

```
In [173]: # Renaming columns for Home Games
          NHL_Home=NHL_Game[NHL_Game.home_away == 'home']
          NHL_Home=NHL_Home.rename(columns={'hscore':'goals_for','ascore':'goals_against'})

          NHL_Home.head()
```

```
Out[173]:     comp_id  year  type         competition_name  tid           team_name  \
          42        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          43        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          44        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          45        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          46        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs

             tricode   gid       date  goals_against  goals_for home_away  hgd
          42     TOR   307 2013-10-26            1.0        4.0      home  3.0
          43     TOR   682 2013-11-24            2.0        3.0      home  1.0
          44     TOR  2150 2014-03-25            5.0        3.0      home -2.0
          45     TOR  2067 2014-03-19            5.0        3.0      home -2.0
          46     TOR  2281 2014-04-03            3.0        4.0      home  1.0
```

Using Numpy Select to Set Values using Multiple Conditions: https://datagy.io/pandas-conditional-column/

```
In [174]: # for code reusability, we define function to compare columns, set value and fill th
```

```
'''df1 and df2 are the dataframes to be compared. df is the dataframe'''
def comparator (df_col1, df_col2, df):

    Conditions = [(df_col1<df_col2), (df_col1==df_col2), (df_col1>df_col2)]

    values = [0, 0.5, 1]
    # creating win column for NHL_Home
    df['win'] = np.select(Conditions, values)
    return df.head()
```

```
In [175]: comparator(NHL_Away['goals_for'], NHL_Away['goals_against'], NHL_Away)
```

```
Out[175]:    comp_id  year  type        competition_name  tid            team_name  \
          2        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          3        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          4        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          5        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          6        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs

            tricode   gid        date  goals_for  goals_against home_away  hgd  win
          2     TOR  1003  2013-12-17        1.0            3.0      away  2.0  0.0
          3     TOR  1552  2014-01-26        4.0            5.0      away  1.0  0.0
          4     TOR  1811  2014-03-02        3.0            4.0      away  1.0  0.0
          5     TOR  1940  2014-03-11        3.0            1.0      away -2.0  1.0
          6     TOR  1522  2014-01-24        1.0            7.0      away  6.0  0.0
```

```
In [176]: comparator(NHL_Home['goals_for'], NHL_Home['goals_against'], NHL_Home)
```

```
Out[176]:     comp_id  year  type        competition_name  tid            team_name  \
          42        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          43        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          44        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          45        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs
          46        1  2013     2  2013 NHL Regular Season    1  Toronto Maple Leafs

             tricode   gid        date  goals_against  goals_for home_away  hgd  win
          42     TOR   307  2013-10-26            1.0        4.0      home  3.0  1.0
          43     TOR   682  2013-11-24            2.0        3.0      home  1.0  1.0
          44     TOR  2150  2014-03-25            5.0        3.0      home -2.0  0.0
          45     TOR  2067  2014-03-19            5.0        3.0      home -2.0  0.0
          46     TOR  2281  2014-04-03            3.0        4.0      home  1.0  1.0
```

Quiz Question 2 What is the time range of the NHL_Game dataframe after you performed step 8?

```
In [177]: NHL_Game['date'].describe()
```

```
Out[177]: count                   18506
          unique                   1607
```

```
top       2017-11-23 00:00:00
freq                       30
first     2010-10-07 00:00:00
last      2018-06-08 00:00:00
Name: date, dtype: object
```

j) Append the "NHL_Home" and "NHL_Away" dataframes to be the new "NHL_Game" dataframe. pd.append() method is deprecated but pd.concat() isn't. pandas.concat() function in Python: https://www.geeksforgeeks.org/pandas-concat-function-in-python/

how to make a new line in a jupyter markdown cell: https://stackoverflow.com/questions/41906199/how-to-make-a-new-line-in-a-jupyter-markdown-cell

```
In [178]: NHL_Game=pd.concat([NHL_Home,NHL_Away])
          NHL_Game.sample(8)
```

```
Out[178]:        comp_id         competition_name       date   gid  goals_against  \
          9945      4287  2011 NHL Regular Season 2011-10-16  4401            2.0
          3744         2  2017 NHL Regular Season 2017-12-09   892            3.0
          16213     8011  2014 NHL Regular Season 2015-01-04  8724            1.0
          4406         2  2017 NHL Regular Season 2017-10-22   239            4.0
          8131      2734  2016 NHL Regular Season 2017-02-20  4470            0.0
          3237         2  2017 NHL Regular Season 2017-11-22   636            5.0
          17190     8011  2014 NHL Regular Season 2015-01-01  8704            1.0
          14589     5662  2010 NHL Regular Season 2010-12-29  6751            3.0

                 goals_for  hgd home_away            team_name  tid tricode  type  win  \
          9945         3.0 -1.0      away    Detroit Red Wings   18     DET     2  1.0
          3744         2.0 -1.0      home        Anaheim Ducks   21     ANA     2  0.0
          16213        4.0 -3.0      away   Montréal Canadiens    2     MTL     2  1.0
          4406         2.0 -2.0      home      Arizona Coyotes   43     ARI     2  0.0
          8131         1.0  1.0      home        Anaheim Ducks   21     ANA     2  1.0
          3237         2.0 -3.0      home  Philadelphia Flyers   14     PHI     2  0.0
          17190        3.0  2.0      home    Detroit Red Wings   18     DET     2  1.0
          14589        4.0 -1.0      away        Boston Bruins   20     BOS     2  1.0

                 year
          9945   2011
          3744   2017
          16213  2014
          4406   2017
          8131   2016
          3237   2017
          17190  2014
          14589  2010
```

Question 3 After performing step 9 above, what are the values of the "gid" variable of the fifth, tenth, and fifteenth observations by date in ascending order in the prepared NHL_Game

dataframe?

```
In [179]: NHL_Game.sort_values(by=['date'], ascending=[True]).head(15)
```

```
Out[179]:        comp_id         competition_name       date   gid  goals_against  \
          13564     5662  2010 NHL Regular Season 2010-10-07  5662            2.0
          15183     5662  2010 NHL Regular Season 2010-10-07  5666            4.0
          15794     5662  2010 NHL Regular Season 2010-10-07  5666            3.0
          13955     5662  2010 NHL Regular Season 2010-10-07  5664            3.0
          13611     5662  2010 NHL Regular Season 2010-10-07  5662            3.0
          14163     5662  2010 NHL Regular Season 2010-10-07  5664            2.0
          14056     5662  2010 NHL Regular Season 2010-10-08  5670            0.0
          15112     5662  2010 NHL Regular Season 2010-10-08  5681            2.0
          14796     5662  2010 NHL Regular Season 2010-10-08  5668            3.0
          13672     5662  2010 NHL Regular Season 2010-10-08  5683            4.0
          15381     5662  2010 NHL Regular Season 2010-10-08  5674            3.0
          13770     5662  2010 NHL Regular Season 2010-10-08  5668            4.0
          15926     5662  2010 NHL Regular Season 2010-10-08  5683            2.0
          15843     5662  2010 NHL Regular Season 2010-10-08  5677            1.0
          15675     5662  2010 NHL Regular Season 2010-10-08  5672            3.0

                 goals_for  hgd home_away              team_name    tid tricode  type  \
          13564        3.0  1.0      home    Toronto Maple Leafs      1     TOR     2
          15183        3.0 -1.0      home         Minnesota Wild     35     MIN     2
          15794        4.0 -1.0      away    Carolina Hurricanes     66     CAR     2
          13955        2.0 -1.0      home    Pittsburgh Penguins      8     PIT     2
          13611        2.0  1.0      away     Montréal Canadiens      2     MTL     2
          14163        3.0 -1.0      away    Philadelphia Flyers     14     PHI     2
          14056        4.0  4.0      home        Edmonton Oilers     10     EDM     2
          15112        1.0 -1.0      home        Ottawa Senators     32     OTT     2
          14796        4.0  1.0      home     Colorado Avalanche     22     COL     2
          13672        2.0  2.0      away    Washington Capitals      5     WSH     2
          15381        4.0 -1.0      away           Dallas Stars     46     DAL     2
          13770        3.0  1.0      away     Chicago Blackhawks      6     CHI     2
          15926        4.0  2.0      home      Atlanta Thrashers  11366     ATL     2
          15843        2.0  1.0      home    Carolina Hurricanes     66     CAR     2
          15675        2.0 -1.0      home  Columbus Blue Jackets     52     CBJ     2

                 win  year
          13564  1.0  2010
          15183  0.0  2010
          15794  1.0  2010
          13955  0.0  2010
          13611  0.0  2010
          14163  1.0  2010
          14056  1.0  2010
          15112  0.0  2010
          14796  1.0  2010
```

```
13672  0.0  2010
15381  1.0  2010
13770  0.0  2010
15926  1.0  2010
15843  1.0  2010
15675  0.0  2010
```

My perspective: 05th observation - 5662 10th observation - 5683 15th observation - 5672

k) Generate a team level dataframe that aggregates the total number of games won, the total number of "goals_for" and "goals_against" for each team in each competition (i.e. grouped by tid, competition_name and type). Name this new dataframe "NHL_Team_Stats". Make sure to convert the indexes of the new dataframe back as variables.

```
In [180]: NHL_Team_Stats = NHL_Game.groupby(['tid', 'competition_name', 'type'])['win', 'goals_
          NHL_Team_Stats.sample(8)
```

```
Out[180]:      tid         competition_name  type   win  goals_for  goals_against
          307   45  2017 NHL Regular Season     2  40.5      237.0          240.0
          26     4  2013 NHL Regular Season     2  36.0      235.0          249.0
          261   32  2014 NHL Regular Season     2  42.0      243.0          222.0
          278   35         2017 NHL Playoff     3   1.0        8.0           12.0
          33     5         2010 NHL Playoff     3   4.0       21.0           20.0
          287   41  2013 NHL Regular Season     2  45.0      207.0          175.0
          20     2  2015 NHL Regular Season     2  35.5      211.0          229.0
          232   28  2011 NHL Regular Season     2  26.0      117.0           83.0
```

l) Create a dataframe "NHL_Game_Count" that include the total number of games played by each team in each competition (i.e. grouped by tid, competition_name and type). Name this new variable in the dataframe "game_count".

```
In [181]: NHL_Game['game_count']=1
          NHL_Game.head()
          NHL_Game_Count=NHL_Game.groupby(['tid', 'competition_name', 'type'])['game_count'].su
          NHL_Game_Count.head(8)
```

```
Out[181]:    tid         competition_name  type  game_count
          0    1  2010 NHL Regular Season     2          82
          1    1  2011 NHL Regular Season     2          40
          2    1         2012 NHL Playoff     3           7
          3    1  2012 NHL Regular Season     2          46
          4    1  2013 NHL Regular Season     2          79
          5    1  2014 NHL Regular Season     2          78
          6    1  2015 NHL Regular Season     2          79
          7    1         2016 NHL Playoff     3           6
```

m) Merge dataframes.

```
a) Merge the NHL_Game_Count dataframe into the NHL_Team_Stats dataframe by tid, competition_na
Continue to name the merged dataframe NHL_Team_Stats.
b) Merge the NHL_Team dataframe into the NHL_Team_Stats dataframe by tid. Continue to name the
NHL_Team_Stats.
```

```
In [182]: # comparing columns by visual inspection
          print(NHL_Game_Count.columns.tolist())
          print(NHL_Team_Stats.columns.tolist())

['tid', 'competition_name', 'type', 'game_count']
['tid', 'competition_name', 'type', 'win', 'goals_for', 'goals_against']
```

```
In [183]: NHL_Team_Stats=pd.merge(NHL_Game_Count, NHL_Team_Stats, on=['tid', 'competition_name
          NHL_Team_Stats.head()

Out[183]:    tid         competition_name  type  game_count   win  goals_for  \
          0    1  2010 NHL Regular Season     2          82  36.0      223.0
          1    1  2011 NHL Regular Season     2          40  20.0      129.0
          2    1         2012 NHL Playoff     3           7   3.0       18.0
          3    1  2012 NHL Regular Season     2          46  25.0      144.0
          4    1  2013 NHL Regular Season     2          79  38.0      231.0

             goals_against
          0          259.0
          1          129.0
          2           22.0
          3          129.0
          4          250.0
```

```
In [184]: print(NHL_Team.columns.tolist())
          print(NHL_Team_Stats.columns.tolist())

['tid', 'team_name', 'tricode']
['tid', 'competition_name', 'type', 'game_count', 'win', 'goals_for', 'goals_against']
```

```
In [185]: NHL_Team_Stats=pd.merge(NHL_Team, NHL_Team_Stats, on=['tid'])
          NHL_Team_Stats.head()

Out[185]:    tid            team_name tricode         competition_name  type  \
          0    1  Toronto Maple Leafs     TOR  2010 NHL Regular Season     2
          1    1  Toronto Maple Leafs     TOR  2011 NHL Regular Season     2
          2    1  Toronto Maple Leafs     TOR         2012 NHL Playoff     3
          3    1  Toronto Maple Leafs     TOR  2012 NHL Regular Season     2
          4    1  Toronto Maple Leafs     TOR  2013 NHL Regular Season     2

             game_count   win  goals_for  goals_against
          0          82  36.0      223.0          259.0
```

```
1           40  20.0       129.0          129.0
2            7   3.0        18.0           22.0
3           46  25.0       144.0          129.0
4           79  38.0       231.0          250.0
```

Import the "pp.pk.ppgf.csv" data file and name the dataframe as "NHL_PPPK" in Jupyter Notebook.

```
In [186]: #Import NHL PPPK Data and Display Head
          NHL_PPPK=pd.read_csv("Assignment Data/Week 2/pp.pk.ppgf.csv")
          NHL_PPPK.head()
```

```
Out[186]:   tricode   pp   pk  ppgf  competition_name
          0     ANA   35   27   9.0  2010 NHL Playoff
          1     BOS  126  116  22.0  2010 NHL Playoff
          2     BUF   48   46  13.0  2010 NHL Playoff
          3     CHI   27   39   6.0  2010 NHL Playoff
          4     DET   59   55   6.0  2010 NHL Playoff
```

```
In [187]: NHL_PPPK.shape
```

```
Out[187]: (369, 5)
```

Merge the "NHL_PPPK" dataframe into the "NHL_Team_Stats" dataframe by "tricode" and "competition_name".

```
In [188]: print(NHL_PPPK.columns.tolist())
          print(NHL_Team_Stats.columns.tolist())
```

```
['tricode', 'pp', 'pk', 'ppgf', 'competition_name']
['tid', 'team_name', 'tricode', 'competition_name', 'type', 'game_count', 'win', 'goals_for',
```

```
In [189]: NHL_PPPK=pd.merge(NHL_PPPK, NHL_Team_Stats, on=['tricode', 'competition_name'])
          NHL_PPPK.head()
```

```
Out[189]:   tricode   pp   pk  ppgf  competition_name  tid            team_name  type  \
          0     ANA   35   27   9.0  2010 NHL Playoff   21        Anaheim Ducks     3
          1     BOS  126  116  22.0  2010 NHL Playoff   20        Boston Bruins     3
          2     BUF   48   46  13.0  2010 NHL Playoff   17        Buffalo Sabres     3
          3     CHI   27   39   6.0  2010 NHL Playoff    6    Chicago Blackhawks     3
          4     DET   59   55   6.0  2010 NHL Playoff   18     Detroit Red Wings     3

             game_count   win  goals_for  goals_against
          0           6   2.0       19.0           22.0
          1          24  16.0       76.0           48.0
          2           7   3.0       17.0           22.0
          3           7   3.0       22.0           16.0
          4          11   7.0       36.0           27.0
```

Create new variables in the "NHL_Team_Stats" dataframe.

```
a) Winning percentage (win_pct)=win/ total number of games played
b) Average goals for per game (avg_gf)=total number of goals for / total number of games playe
c) Average goals against per game (avg_ga)=total number of goals against / total number of game
```

```
In [190]: # Winning percentage
          NHL_Team_Stats['win_pct']= NHL_Team_Stats['win']/NHL_Team_Stats['game_count']
          # Average goals for per game
          NHL_Team_Stats['avg_gf']=NHL_Team_Stats['goals_for']/NHL_Team_Stats['game_count']
          # Average goals against per game
          NHL_Team_Stats['avg_ga']=NHL_Team_Stats['goals_against']/NHL_Team_Stats['game_count'

          # checking columns
          NHL_Team_Stats.head()
```

```
Out[190]:    tid            team_name tricode         competition_name  type  \
          0    1  Toronto Maple Leafs     TOR  2010 NHL Regular Season     2
          1    1  Toronto Maple Leafs     TOR  2011 NHL Regular Season     2
          2    1  Toronto Maple Leafs     TOR         2012 NHL Playoff     3
          3    1  Toronto Maple Leafs     TOR  2012 NHL Regular Season     2
          4    1  Toronto Maple Leafs     TOR  2013 NHL Regular Season     2

             game_count   win  goals_for  goals_against   win_pct    avg_gf    avg_ga
          0          82  36.0      223.0          259.0  0.439024  2.719512  3.158537
          1          40  20.0      129.0          129.0  0.500000  3.225000  3.225000
          2           7   3.0       18.0           22.0  0.428571  2.571429  3.142857
          3          46  25.0      144.0          129.0  0.543478  3.130435  2.804348
          4          79  38.0      231.0          250.0  0.481013  2.924051  3.164557
```

In the "NHL_Competition" dataframe, the variable "type" indicates the type of competition: type=2 – regular season. Create a dataframe that contains team statistics for games only during regular seasons. Name this dataframe "NHL_Team_R_Stats".

```
In [191]: # Games played in the regular season

          NHL_Team_R_Stats= NHL_Team_Stats[NHL_Team_Stats.type==2]
          NHL_Team_R_Stats.head()
```

```
Out[191]:    tid            team_name tricode         competition_name  type  \
          0    1  Toronto Maple Leafs     TOR  2010 NHL Regular Season     2
          1    1  Toronto Maple Leafs     TOR  2011 NHL Regular Season     2
          3    1  Toronto Maple Leafs     TOR  2012 NHL Regular Season     2
          4    1  Toronto Maple Leafs     TOR  2013 NHL Regular Season     2
          5    1  Toronto Maple Leafs     TOR  2014 NHL Regular Season     2

             game_count   win  goals_for  goals_against   win_pct    avg_gf    avg_ga
          0          82  36.0      223.0          259.0  0.439024  2.719512  3.158537
          1          40  20.0      129.0          129.0  0.500000  3.225000  3.225000
```

```
3           46  25.0       144.0           129.0  0.543478  3.130435  2.804348
4           79  38.0       231.0           250.0  0.481013  2.924051  3.164557
5           78  29.0       209.0           258.0  0.371795  2.679487  3.307692
```

0.2 Part 2 - Descriptive and Summary Analyses

In the "NHL_Game" dataframe, calculate summary statistics for the "goals_for" variable; calculate summary statistics for the "goals_against" variable based on whether it is home or away game.

```
In [192]: NHL_Game.groupby('home_away')['goals_for' , 'goals_against'].describe()
```

```
Out[192]:           goals_for                                                  \
                        count      mean       std  min  25%  50%  75%   max
          home_away
          away         9253.0  2.689830  1.608916  0.0  1.0  3.0  4.0  10.0
          home         9253.0  2.961958  1.688463  0.0  2.0  3.0  4.0  10.0

                    goals_against
                            count      mean       std  min  25%  50%  75%   max
          home_away
          away             9253.0  2.961958  1.688463  0.0  2.0  3.0  4.0  10.0
          home             9253.0  2.689830  1.608916  0.0  1.0  3.0  4.0  10.0
```

```
In [193]: NHL_Game.groupby('home_away')['goals_for' , 'goals_against'].describe().reset_index(
```

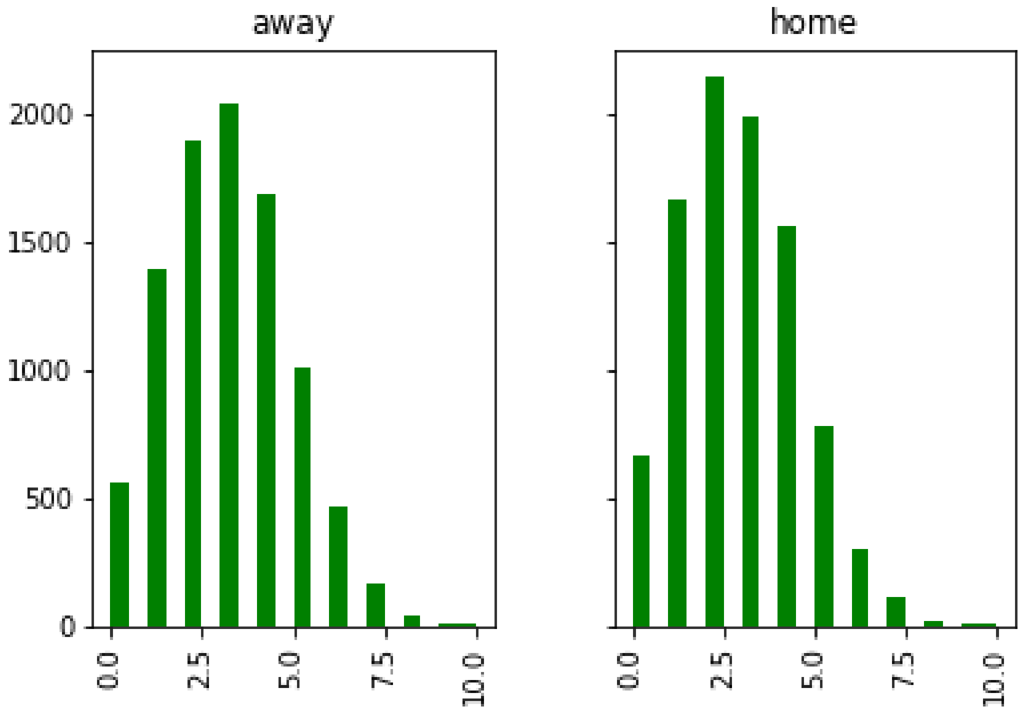
```
Out[193]:   home_away goals_for                                                  \
                          count      mean       std  min  25%  50%  75%   max
          0      away    9253.0  2.689830  1.608916  0.0  1.0  3.0  4.0  10.0
          1      home    9253.0  2.961958  1.688463  0.0  2.0  3.0  4.0  10.0

            goals_against
                    count      mean       std  min  25%  50%  75%   max
          0        9253.0  2.961958  1.688463  0.0  2.0  3.0  4.0  10.0
          1        9253.0  2.689830  1.608916  0.0  1.0  3.0  4.0  10.0
```

Create a histogram of the "goals_against" variable by whether the game is home or away

```
a) Make the color of the histogram green
b) Set the number of bins to be 20
c) Make sure the two sub-histograms share the same ranges for the x-axis and y-axis.
```

```
In [194]: NHL_Game.hist(by='home_away', column='goals_against', color='green', bins=20, sharex
          plt.savefig('NHL_goals_against_HomeAway.png')
```



0.3 Part 3 - Correlation Analyses

In the "NHL_Team_R_Stats" dataframe, make a scatter plot to depict the relationship between the total number of goals for and the winning percentage.

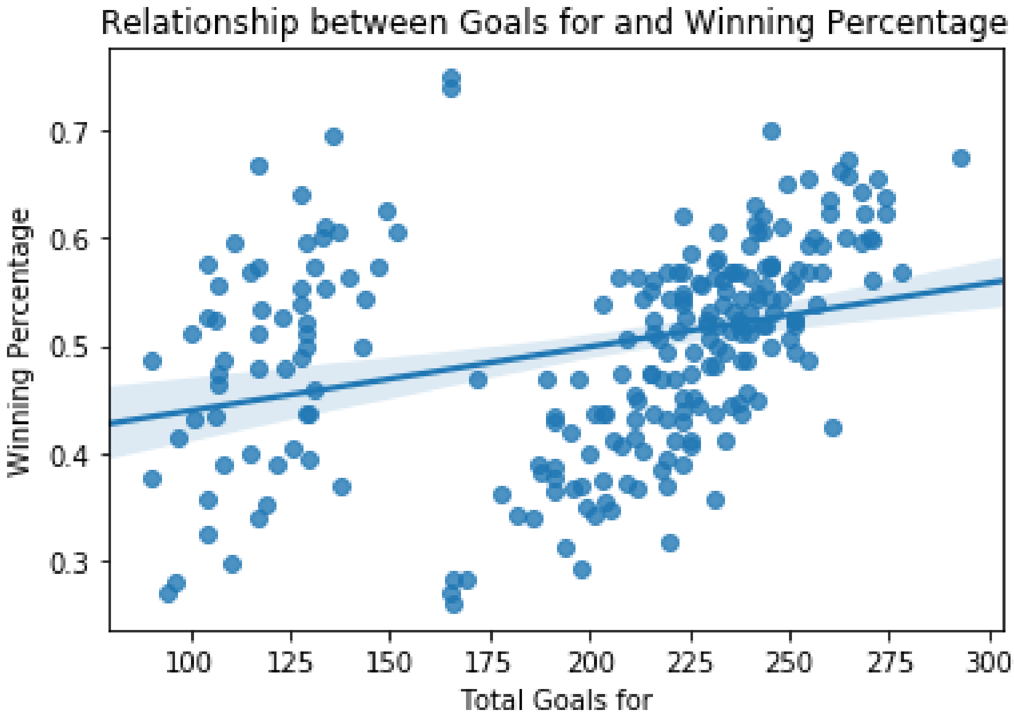
```
a) Plot the total number of goals for on the x-axis and winning percentage on the y-axis.
b) Add a regression line to the scatter plot.
c) Make the title of the graph Relationship between Goals for and Winning Percentage and make
d) Label the x-axis Total Goals for and label the y-axis Winning Percentage.
```

```
In [195]: NHL_Team_R_Stats.head(3)
```

```
Out[195]:    tid           team_name tricode         competition_name  type  \
          0    1  Toronto Maple Leafs     TOR  2010 NHL Regular Season     2
          1    1  Toronto Maple Leafs     TOR  2011 NHL Regular Season     2
          3    1  Toronto Maple Leafs     TOR  2012 NHL Regular Season     2

             game_count   win  goals_for  goals_against   win_pct    avg_gf    avg_ga
          0          82  36.0      223.0          259.0  0.439024  2.719512  3.158537
          1          40  20.0      129.0          129.0  0.500000  3.225000  3.225000
          3          46  25.0      144.0          129.0  0.543478  3.130435  2.804348
```

```
In [196]: sns.regplot(data=NHL_Team_R_Stats,x='goals_for', y='win_pct');
          plt.title('Relationship between Goals for and Winning Percentage');
          plt.xlabel('Total Goals for');
          plt.ylabel('Winning Percentage');
```



In the "NHL_Team_R_Stats" dataframe, calculate the correlation coefficient between total number of goals for and winning percentage.

```
Create a scatter plot of the total number of goals for and winning percentage similar to step 
    a) Plot the total number of goals for on the x-axis and winning percentage on the y-axis.
    b) Add a regression line to the scatter plot.
    c) Make the title of the graph Relationship between Goals for and Winning Percentage and ma
    d) Label the x-axis Total Goals for and label the y-axis Winning Percentage.
```

```
In [197]: NHL_Team_R_Stats.head(3)
```

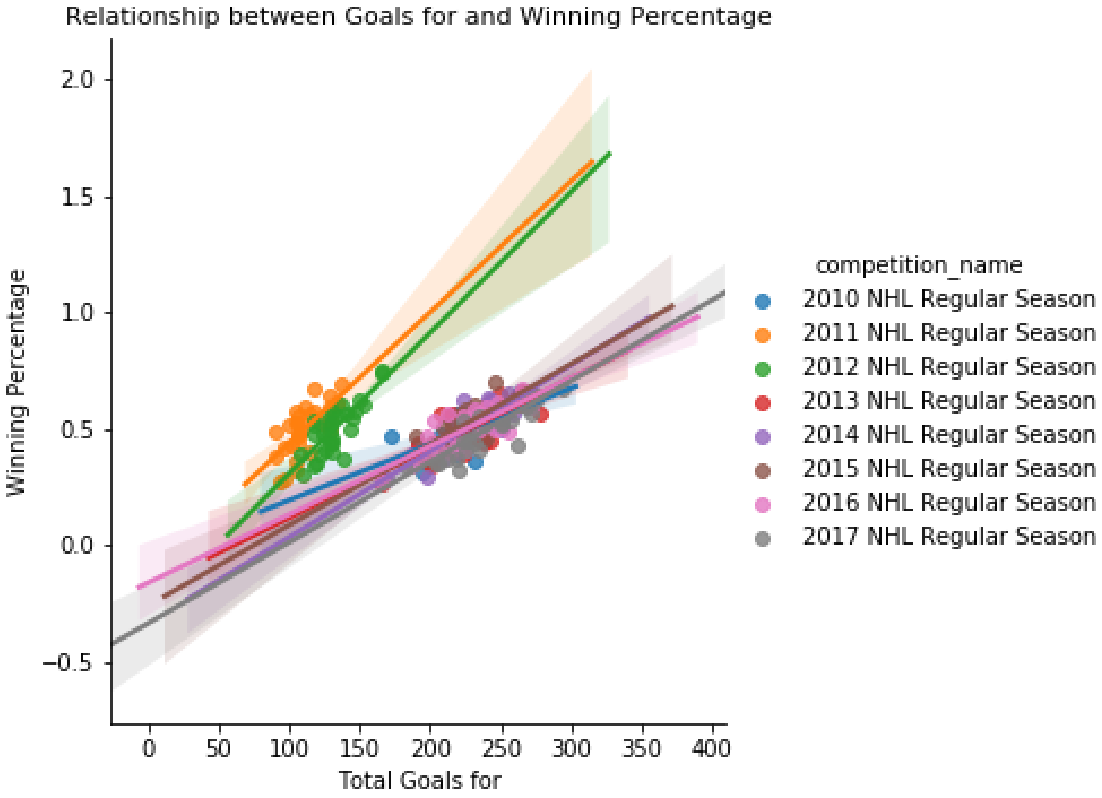
```
Out[197]:    tid           team_name tricode         competition_name  type  \
          0    1  Toronto Maple Leafs     TOR  2010 NHL Regular Season     2
          1    1  Toronto Maple Leafs     TOR  2011 NHL Regular Season     2
          3    1  Toronto Maple Leafs     TOR  2012 NHL Regular Season     2

             game_count   win  goals_for  goals_against   win_pct    avg_gf    avg_ga
          0          82  36.0      223.0          259.0  0.439024  2.719512  3.158537
          1          40  20.0      129.0          129.0  0.500000  3.225000  3.225000
          3          46  25.0      144.0          129.0  0.543478  3.130435  2.804348
```

```
In [198]: keyvars = NHL_Team_R_Stats[['goals_for', 'win_pct']]
          keyvars.corr()
```

```
Out[198]:            goals_for   win_pct
          goals_for   1.000000  0.315665
          win_pct     0.315665  1.000000
```

```
In [199]: # Using lmplot to automatically fit regression line and group scatterplot
          sns.lmplot(x ='goals_for', y ='win_pct', data = NHL_Team_R_Stats, hue ='competition_
          plt.title('Relationship between Goals for and Winning Percentage', fontsize=11);
          plt.xlabel('Total Goals for');
          plt.ylabel('Winning Percentage');
```



For the “NHL_Team_R_Stats” dataframe, delete observations of 2011 and 2012 seasons. Continue to name the dataframe “NHL_Team_R_Stats”.

```
In the new NHL_Team_R_Stats dataframe, create a scatter plot of total number of goals for and 
a) Plot the total number of goals for on the x-axis and winning percentage on the y-axis.
b) Add a regression line to the scatter plot.
c) Make the title of the graph Relationship between Goals for and Winning Percentage and make 
d) Label the x-axis Total Goals for and label the y-axis Winning Percentage.
```

```
In [200]: NHL_Team_R_Stats.head(2)
```

```
Out[200]:    tid            team_name tricode         competition_name  type  \
          0    1  Toronto Maple Leafs     TOR  2010 NHL Regular Season     2
          1    1  Toronto Maple Leafs     TOR  2011 NHL Regular Season     2

             game_count   win  goals_for  goals_against   win_pct    avg_gf    avg_ga
          0          82  36.0      223.0          259.0  0.439024  2.719512  3.158537
          1          40  20.0      129.0          129.0  0.500000  3.225000  3.225000
```

```
In [201]: # Seeing as the NHL_Team_R_Stats only contain games played in Regular seasons only
          # checking all the regular seasons in the dataset
          NHL_Team_R_Stats.competition_name.value_counts()
```

```
Out[201]: 2017 NHL Regular Season    31
          2016 NHL Regular Season    30
          2014 NHL Regular Season    30
          2013 NHL Regular Season    30
          2010 NHL Regular Season    30
          2011 NHL Regular Season    30
          2012 NHL Regular Season    30
          2015 NHL Regular Season    30
          Name: competition_name, dtype: int64
```

```
In [202]: NHL_Team_R_Stats.shape
```

```
Out[202]: (241, 12)
```

```
In [203]: NHL_Team_R_Stats = NHL_Team_R_Stats[(NHL_Team_R_Stats.competition_name !='2011 NHL R
          NHL_Team_R_Stats.head()
```

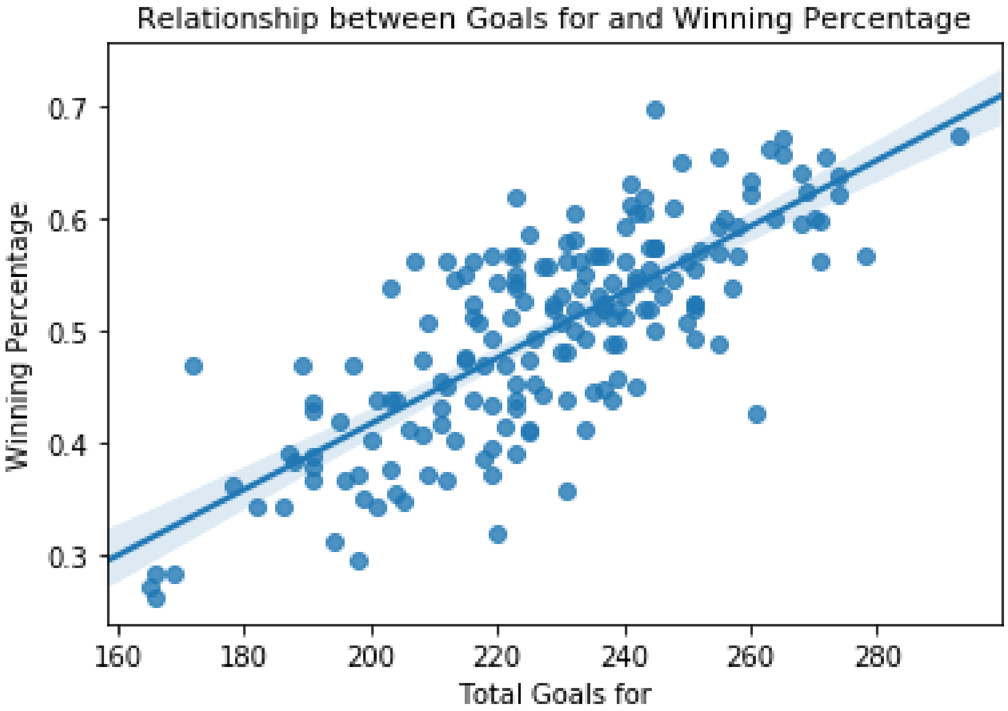
```
Out[203]:    tid            team_name tricode         competition_name  type  \
          0    1  Toronto Maple Leafs     TOR  2010 NHL Regular Season     2
          4    1  Toronto Maple Leafs     TOR  2013 NHL Regular Season     2
          5    1  Toronto Maple Leafs     TOR  2014 NHL Regular Season     2
          6    1  Toronto Maple Leafs     TOR  2015 NHL Regular Season     2
          8    1  Toronto Maple Leafs     TOR  2016 NHL Regular Season     2

             game_count   win  goals_for  goals_against   win_pct    avg_gf    avg_ga
          0          82  36.0      223.0          259.0  0.439024  2.719512  3.158537
          4          79  38.0      231.0          250.0  0.481013  2.924051  3.164557
          5          78  29.0      209.0          258.0  0.371795  2.679487  3.307692
          6          79  29.0      196.0          238.0  0.367089  2.481013  3.012658
          8          82  40.0      255.0          246.0  0.487805  3.109756  3.000000
```

```
In [204]: NHL_Team_R_Stats.shape
```

```
Out[204]: (181, 12)
```

```
In [205]: # Using lmplot to automatically fit regression line and group scatterplot
          sns.regplot(x ='goals_for', y ='win_pct', data = NHL_Team_R_Stats)
          plt.title('Relationship between Goals for and Winning Percentage', fontsize=11);
          plt.xlabel('Total Goals for');
          plt.ylabel('Winning Percentage');
```



Calculate the correlation coefficient between total number of goals for and winning percentage in the updated "NHL_Team_R_Stats" dataframe.

```
Save dataframes as csv files.
a) Name the updated NHL_Game dataframe as NHL_Game2.
b) Name the NHL_Team_Stats dataframe as NHL_Team_Stats.
c) Name the NHL_Team_R_Stats dataframe as NHL_Team_R_Stats.
d) Make sure to exclude the index as a column in the csv files.
```

```
In [206]: keyvars = NHL_Team_R_Stats[['goals_for', 'win_pct']]
          keyvars.corr()
```

```
Out[206]:            goals_for   win_pct
          goals_for   1.000000  0.770626
          win_pct     0.770626  1.000000
```

0.4 Uncomment this Section once your assignment is complete

```
In [207]: # Save Dataframes as .csv files
          NHL_Game.to_csv("NHL_Game2.csv", index=False)
          NHL_Team_Stats.to_csv("NHL_Team_Stats.csv", index=False)
          NHL_Team_R_Stats.to_csv("NHL_Team_R_Stats.csv", index=False)
```

Quiz 2 Question 1 **What are the mean and standard deviation of the total number of goals for in the "NHL_Game" dataframe?**

```
In [208]: NHL_Game.describe()
```

```
Out[208]:            comp_id           gid  goals_against    goals_for           hgd  \
          count  18506.000000  18506.000000   18506.000000  18506.000000  18506.000000
          mean    3734.629309   4739.088188       2.825894      2.825894      0.272128
          std     2805.267754   2737.105786       1.654729      1.654729      2.370648
          min        1.000000      1.000000       0.000000      0.000000     -8.000000
          25%        2.000000   2365.000000       2.000000      2.000000     -1.000000
          50%     4099.000000   4729.000000       3.000000      3.000000      1.000000
          75%     5662.000000   7113.000000       4.000000      4.000000      2.000000
          max     9389.000000   9473.000000      10.000000     10.000000     10.000000

                         tid          type           win          year  game_count
          count  18506.000000  18506.000000  18506.000000  18506.000000     18506.0
          mean      73.391062      2.075219      0.500000   2013.761807         1.0
          std      739.629578      0.263751      0.498363      2.300688         0.0
          min        1.000000      2.000000      0.000000   2010.000000         1.0
          25%       10.000000      2.000000      0.000000   2012.000000         1.0
          50%       21.000000      2.000000      0.500000   2014.000000         1.0
          75%       41.000000      2.000000      1.000000   2016.000000         1.0
          max    11366.000000      3.000000      1.000000   2017.000000         1.0
```

Quiz 2 Question 2 **What is the mean of the total number of goals against for home games? What is the mean of the total number of goals against for away games?**

```
In [209]: NHL_Game.groupby('home_away')['goals_for' , 'goals_against'].describe().reset_index(
```

```
Out[209]:   home_away goals_for                                                 \
                          count      mean       std  min  25%  50%  75%   max
          0      away    9253.0  2.689830  1.608916  0.0  1.0  3.0  4.0  10.0
          1      home    9253.0  2.961958  1.688463  0.0  2.0  3.0  4.0  10.0

            goals_against
                    count      mean       std  min  25%  50%  75%   max
          0        9253.0  2.961958  1.688463  0.0  2.0  3.0  4.0  10.0
          1        9253.0  2.689830  1.608916  0.0  1.0  3.0  4.0  10.0
```

```
In [ ]:
```