Soccer

October 12, 2019

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sqlite3
import seaborn as sns
conn = sqlite3.connect('database.sqlite')
c = conn.cursor()
# Country Data COUNTRY table
COUNTRY_df = pd.read_sql_query("select * from COUNTRY;", conn)
# League data from LEAGUE
LEAGUE_df = pd.read_sql_query("select * from LEAGUE;", conn)
# Match data from Matches - just pulling 1 year for now -2015/2016 might be␣
 ↪latest year
# You can remove restriction if you want
#MATCH_df = pd.read_sql_query("select * from MATCH where SEASON = '2015/2016';
 ↪", conn)
# Match data from Matches - just pulling 1 and relevant data for now - we can␣
 ↪remove restriction later if we want
MATCH_df = pd.read_sql_query("SELECT DATE, LEAGUE_ID, HOME_TEAM_API_ID,␣
 ↪AWAY_TEAM_API_ID, HOME_TEAM_GOAL, AWAY_TEAM_GOAL FROM MATCH WHERE SEASON =␣
 ↪'2015/2016';",
conn)
PLAYER_df = pd.read_sql_query("select * from PLAYER;", conn)
PLAYER_ATTRIBUTES_df = pd.read_sql_query("select * from PLAYER_ATTRIBUTES;",␣
 ↪conn)
TEAM_df = pd.read_sql_query("select * from TEAM;", conn)
TEAM_ATTRIBUTES_df = pd.read_sql_query("select * from TEAM_ATTRIBUTES;", conn)
```

1 We calculate the "differential" for both the home team and away team

[2]:
```
MATCH_df["HOME_DIFF"] = MATCH_df["home_team_goal"]- MATCH_df["away_team_goal"]
MATCH_df["AWAY_DIFF"] = MATCH_df["away_team_goal"]- MATCH_df["home_team_goal"]
```

```
MATCH_home_df=MATCH_df.loc[:,["date","home_team_api_id",␣
 ↪"HOME_DIFF","league_id","away_team_api_id"]]
MATCH_away_df=MATCH_df.loc[:,["date","away_team_api_id",␣
 ↪"AWAY_DIFF","league_id","home_team_api_id"]]
```

[3]:
```
MATCH_away_df['W'] = 0
MATCH_away_df['L'] = 0
MATCH_away_df['T'] = 0
MATCH_home_df['W'] = 0
MATCH_home_df['L'] = 0
MATCH_home_df['T'] = 0


MATCH_away_df.loc[MATCH_away_df.AWAY_DIFF >0, 'W'] = 1
MATCH_away_df.loc[MATCH_away_df.AWAY_DIFF <0, 'L'] = 1
MATCH_away_df.loc[MATCH_away_df.AWAY_DIFF ==0, 'T'] = 1

MATCH_home_df.loc[MATCH_home_df.HOME_DIFF >0, 'W'] = 1
MATCH_home_df.loc[MATCH_home_df.HOME_DIFF <0, 'L'] = 1
MATCH_home_df.loc[MATCH_home_df.HOME_DIFF ==0, 'T'] = 1

MATCH_away_df.head(5)
```

[3]:

	date	away_team_api_id	AWAY_DIFF	league_id	\
0	2015-07-24 00:00:00	8342	-1	1	
1	2015-07-25 00:00:00	9985	-1	1	
2	2015-07-25 00:00:00	1773	-2	1	
3	2015-07-25 00:00:00	8203	-2	1	
4	2015-07-25 00:00:00	9994	-2	1	

	home_team_api_id	W	L	T
0	9997	0	1	0
1	8571	0	1	0
2	9987	0	1	0
3	8573	0	1	0
4	10000	0	1	0

2 We combine the home team data with the away team data to create 1 list of games, with the differential

[4]:
```
MATCH_home_df = MATCH_home_df.rename(columns={"home_team_api_id":␣
 ↪"team_api_id","HOME_DIFF":"DIFF","away_team_api_id":"opponent"})
MATCH_away_df = MATCH_away_df.rename(columns={"away_team_api_id":␣
 ↪"team_api_id","AWAY_DIFF":"DIFF","home_team_api_id":"opponent"})

MATCH_away_df.head()
```

```
frames = [MATCH_home_df, MATCH_away_df]
all_games_unsorted = pd.concat(frames)
all_games = all_games_unsorted.sort_values(by=["team_api_id","date"])
all_games.reset_index(inplace=True)
all_games.head()
```

[4]:

	index	date	team_api_id	DIFF	league_id	opponent	W	L	T
0	1995	2015-07-18 00:00:00	1601	-2	15722	8019	0	1	0
1	2084	2015-07-24 00:00:00	1601	2	15722	8028	1	0	0
2	2173	2015-08-03 00:00:00	1601	1	15722	8245	1	0	0
3	2188	2015-08-10 00:00:00	1601	0	15722	8033	0	0	1
4	2197	2015-08-16 00:00:00	1601	-1	15722	1957	0	1	0

[5]:
```
### We want to figure out the what the differential was for the next game as␣
 ↪well.
### We will make a copy of the table, and add "ng_" as a suffix to the columns␣
 ↪for "next game"
next_game_unsorted = all_games.copy()
next_game_unsorted = next_game_unsorted.rename(columns={"DIFF": "ng_DIFF",
"team_api_id": "ng_team_api_id2",
"league_id": "ng_league_id",
"date": "ng_date",
"W": "ng_W",
"L": "ng_L",
"T": "ng_T" })

next_game = next_game_unsorted.sort_values(by=["ng_team_api_id2","ng_date"])
next_game.head()
```

[5]:

	index	ng_date	ng_team_api_id2	ng_DIFF	ng_league_id	\
0	1995	2015-07-18 00:00:00	1601	-2	15722	
1	2084	2015-07-24 00:00:00	1601	2	15722	
2	2173	2015-08-03 00:00:00	1601	1	15722	
3	2188	2015-08-10 00:00:00	1601	0	15722	
4	2197	2015-08-16 00:00:00	1601	-1	15722	

	opponent	ng_W	ng_L	ng_T
0	8019	0	1	0
1	8028	1	0	0
2	8245	1	0	0
3	8033	0	0	1
4	1957	0	1	0

3 We will merge the original data to the copy, but shift the copy data by one row, so it will line up with the next game

[6]:
```
#df_complete = pd.merge(all_games, next_game.shift(-1).fillna(0).astype(int),␣
 ↪left_index=True, right_index=True)
df_complete = pd.merge(all_games, next_game.shift(-1).fillna(0),␣
 ↪left_index=True, right_index=True)
df_complete.head()
```

[6]:
```
   index_x                 date  team_api_id  DIFF  league_id  opponent_x  W  \
0     1995  2015-07-18 00:00:00         1601    -2      15722        8019  0
1     2084  2015-07-24 00:00:00         1601     2      15722        8028  1
2     2173  2015-08-03 00:00:00         1601     1      15722        8245  1
3     2188  2015-08-10 00:00:00         1601     0      15722        8033  0
4     2197  2015-08-16 00:00:00         1601    -1      15722        1957  0

   L  T  index_y              ng_date  ng_team_api_id2  ng_DIFF  ng_league_id  \
0  1  0   2084.0  2015-07-24 00:00:00           1601.0      2.0       15722.0
1  0  0   2173.0  2015-08-03 00:00:00           1601.0      1.0       15722.0
2  0  0   2188.0  2015-08-10 00:00:00           1601.0      0.0       15722.0
3  0  1   2197.0  2015-08-16 00:00:00           1601.0     -1.0       15722.0
4  1  0   2204.0  2015-08-22 00:00:00           1601.0      3.0       15722.0

   opponent_y  ng_W  ng_L  ng_T
0      8028.0   1.0   0.0   0.0
1      8245.0   1.0   0.0   0.0
2      8033.0   0.0   0.0   1.0
3      1957.0   0.0   1.0   0.0
4    177361.0   1.0   0.0   0.0
```

4 Oops! For the last game of the season (where we don't have a next game), checking the next game would have returned the next team's first game. We should just zero that out.

[7]:
```
#df.loc[df.First_name == 'Bill', 'name_match'] = 'Match'
df_complete.sort_values(by=["team_api_id","date"]).head()
df_complete.loc[df_complete.team_api_id != df_complete.ng_team_api_id2,
"ng_DIFF"] = 0
df_complete.loc[df_complete.team_api_id != df_complete.ng_team_api_id2,
"ng_league_id"] = 0
df_complete.loc[df_complete.team_api_id != df_complete.ng_team_api_id2,
"opponent_y"] = 0
df_complete.loc[df_complete.team_api_id != df_complete.ng_team_api_id2,
"ng_date"] = 0
df_complete.loc[df_complete.team_api_id != df_complete.ng_team_api_id2,
```

```
"ng_W"] = 0
df_complete.loc[df_complete.team_api_id != df_complete.ng_team_api_id2,
"ng_L"] = 0
df_complete.loc[df_complete.team_api_id != df_complete.ng_team_api_id2,
"ng_T"] = 0
df_complete.head()
```

[7]:

	index_x	date	team_api_id	DIFF	league_id	opponent_x	W
0	1995	2015-07-18 00:00:00	1601	-2	15722	8019	0
1	2084	2015-07-24 00:00:00	1601	2	15722	8028	1
2	2173	2015-08-03 00:00:00	1601	1	15722	8245	1
3	2188	2015-08-10 00:00:00	1601	0	15722	8033	0
4	2197	2015-08-16 00:00:00	1601	-1	15722	1957	0

	L	T	index_y	ng_date	ng_team_api_id2	ng_DIFF	ng_league_id
0	1	0	2084.0	2015-07-24 00:00:00	1601.0	2.0	15722.0
1	0	0	2173.0	2015-08-03 00:00:00	1601.0	1.0	15722.0
2	0	0	2188.0	2015-08-10 00:00:00	1601.0	0.0	15722.0
3	0	1	2197.0	2015-08-16 00:00:00	1601.0	-1.0	15722.0
4	1	0	2204.0	2015-08-22 00:00:00	1601.0	3.0	15722.0

	opponent_y	ng_W	ng_L	ng_T
0	8028.0	1.0	0.0	0.0
1	8245.0	1.0	0.0	0.0
2	8033.0	0.0	0.0	1.0
3	1957.0	0.0	1.0	0.0
4	177361.0	1.0	0.0	0.0

5 Let's delete all the extra columns we created

[8]:
```
#df_complete.drop(['index_x','index_y','ng_date','ng_league_id'], axis=1,␣
↪inplace=True)
df_complete.drop(['ng_date','ng_league_id'], axis=1, inplace=True)
df_complete.head()
```

[8]:

	index_x	date	team_api_id	DIFF	league_id	opponent_x	W
0	1995	2015-07-18 00:00:00	1601	-2	15722	8019	0
1	2084	2015-07-24 00:00:00	1601	2	15722	8028	1
2	2173	2015-08-03 00:00:00	1601	1	15722	8245	1
3	2188	2015-08-10 00:00:00	1601	0	15722	8033	0
4	2197	2015-08-16 00:00:00	1601	-1	15722	1957	0

	L	T	index_y	ng_team_api_id2	ng_DIFF	opponent_y	ng_W	ng_L	ng_T
0	1	0	2084.0	1601.0	2.0	8028.0	1.0	0.0	0.0
1	0	0	2173.0	1601.0	1.0	8245.0	1.0	0.0	0.0
2	0	0	2188.0	1601.0	0.0	8033.0	0.0	0.0	1.0
3	0	1	2197.0	1601.0	-1.0	1957.0	0.0	1.0	0.0

```
4  1  0   2204.0          1601.0      3.0    177361.0   1.0   0.0   0.0
```

6 Now, lets find the most dominant team. We will do that by grouping by the bext team

[9]:
```
df_teams = df_complete.groupby(['league_id','team_api_id'], as_index=False).
 ↪sum()
team_merge = pd.merge(df_teams, TEAM_df, on="team_api_id")
team_merge.tail()
```

[9]:
```
     league_id  team_api_id  index_x  DIFF  opponent_x   W   L   T   index_y  \
183      24558        10190   116479   -25      354444  10  18   8  116478.0
184      24558        10191   116489    -9      354440  10  15  11  116491.0
185      24558        10192   116498    31      354436  20   7   9  113596.0
186      24558        10199   116463     9      354408  15  12   9  116089.0
187      24558        10243   116522   -23      354232   7  16  13  113997.0

     ng_team_api_id2  ng_DIFF  opponent_y  ng_W  ng_L  ng_T     id  \
183         366841.0    -27.0    346548.0   9.0  18.0   8.0  49479
184         366877.0     -7.0    344484.0  10.0  14.0  11.0  49837
185         366914.0     31.0    344193.0  20.0   7.0   8.0  49117
186         367170.0      9.0    344229.0  15.0  12.0   8.0  49121
187         368754.0    -23.0    344040.0   7.0  16.0  12.0  49124

     team_fifa_api_id  team_long_name team_short_name
183             898.0   FC St. Gallen             GAL
184            1715.0         FC Thun             THU
185             900.0  BSC Young Boys              YB
186             897.0       FC Luzern             LUZ
187             894.0       FC Zürich             ZUR
```

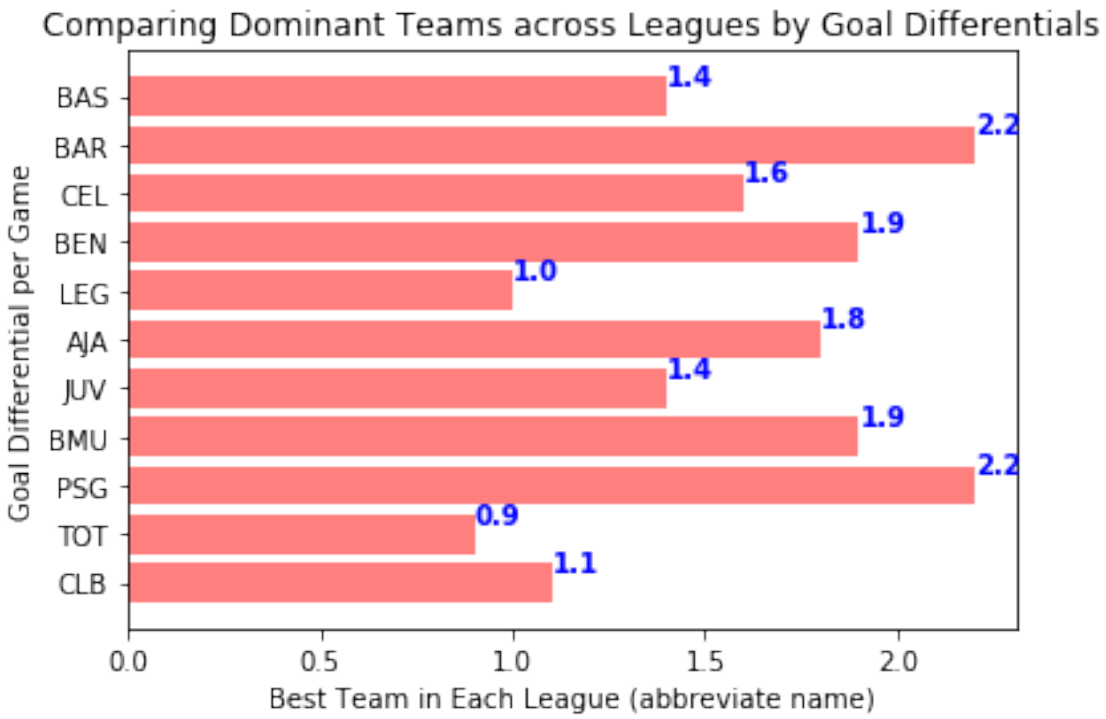
7 For each league, we will append the team that has the maximum Differential. These are our dominant teams, their seasonal goal differentials, and their goal differentials per game.

[10]:
```
leagues = team_merge.league_id.unique()
x = []
y = []
a = []
b = []
c = []
name = []
#df_test['DIFF'] = df_test['DIFF'].apply(np.array)
for l in leagues:
```

```
    df_test = team_merge[team_merge.league_id == l]
    x.append(df_test['team_api_id'].loc[df_test['DIFF'].idxmax()])
    name.append(df_test['team_short_name'].loc[df_test['DIFF'].idxmax()])
    #change "team_api_id" to "team_short_name" for Dominant Team Comparison␣
 ↪barchart
    y.append(df_test['DIFF'].loc[df_test['DIFF'].idxmax()])
    a.append(df_test['W'].loc[df_test['DIFF'].idxmax()])
    b.append(df_test['L'].loc[df_test['DIFF'].idxmax()])
    c.append(df_test['T'].loc[df_test['DIFF'].idxmax()])
d = [x + y + z for x, y, z in zip(a, b, c)]
diff_per_game = [i/j for i,j in zip(y, d)]
diff_per_game = [ round(elem, 1) for elem in diff_per_game]
diff_per_game
```

[10]:
```
[1.1, 0.9, 2.2, 1.9, 1.4, 1.8, 1.0, 1.9, 1.6, 2.2, 1.4]
```

[11]:
```
# # x_axis = np.arange(0, 5, 0.1)
fig,ax = plt.subplots()
ax.barh(name, diff_per_game, color='r', alpha=0.5, align="center")
plt.ylabel("Goal Differential per Game")
plt.xlabel("Best Team in Each League (abbreviate name)")
plt.title("Comparing Dominant Teams across Leagues by Goal Differentials")
for i, v in enumerate(diff_per_game):
    ax.text(v, i + .25, str(v), color='blue', fontweight='bold')
plt.show()
# tick_locations = [value for value in x]
# plt.xticks(tick_locations, x)
#y = [34, 34, 83, 63, 55, 60, 30, 66, 62, 83, 50]
#x = array([8673, 8342, 9772, 9847, 8593, 8634, 9931, 9885, 9925, 9823, 8586],␣
 ↪dtype=int64)
plt.savefig("DomTeamsGoalDiff.png")
```




```
<Figure size 432x288 with 0 Axes>
```

[12]:
```
dom_match = df_complete[df_complete.opponent_x.isin(x)]
#dom_match.opponent_y.value_counts()
```

[13]:
```
df_teams.head()

df_DIFF = df_teams[["team_api_id","DIFF"]]

df_DIFF = df_DIFF.rename(columns={"DIFF": "total_DIFF"})


df_DIFF.head()
```

[13]:

	team_api_id	total_DIFF
0	1773	-11
1	8203	-2
2	8342	34
3	8475	-17
4	8571	-4

[14]:
```
# set team_api_id's DIFF
merge_table_all = pd.merge(dom_match, df_DIFF,left_on="team_api_id",
right_on="team_api_id")
```

```
merge_table_c = merge_table_all.rename(columns={"opponent_x": "dom_team",
"team_api_id": "team",
"opponent_y": "ng_opponent",
"total_DIFF": "team_season_DIFF"

})
merge_table_c.head()
```

[14]:

	index_x	date	team	DIFF	league_id	dom_team	W	L	T	\
0	2231	2015-09-20 00:00:00	1601	-3	15722	8673	0	1	0	
1	2127	2016-02-28 00:00:00	1601	-2	15722	8673	0	1	0	
2	32	2015-10-28 00:00:00	1773	-2	1	8342	0	1	0	
3	189	2016-03-13 00:00:00	1773	-1	1	8342	0	1	0	
4	2212	2015-08-30 00:00:00	1957	0	15722	8673	0	0	1	

	index_y	ng_team_api_id2	ng_DIFF	ng_opponent	ng_W	ng_L	ng_T	\
0	2005.0	1601.0	-1.0	2186.0	0.0	1.0	0.0	
1	2133.0	1601.0	-1.0	2186.0	0.0	1.0	0.0	
2	45.0	1773.0	-2.0	9991.0	0.0	1.0	0.0	
3	1997.0	1957.0	0.0	0.0	0.0	0.0	0.0	
4	2218.0	1957.0	-2.0	8025.0	0.0	1.0	0.0	

	team_season_DIFF
0	-9
1	-9
2	-11
3	-11
4	-17

[15]:
```
merge_table_b = pd.merge(merge_table_c, df_DIFF,left_on="dom_team",
right_on="team_api_id")
merge_table_b = merge_table_b.rename(columns={"total_DIFF":
"dom_season_DIFF"})
merge_table_b.drop(['team_api_id'], axis=1, inplace=True)
merge_table_b.head()
```

[15]:

	index_x	date	team	DIFF	league_id	dom_team	W	L	T	\
0	2231	2015-09-20 00:00:00	1601	-3	15722	8673	0	1	0	
1	2127	2016-02-28 00:00:00	1601	-2	15722	8673	0	1	0	
2	2212	2015-08-30 00:00:00	1957	0	15722	8673	0	0	1	
3	2108	2016-02-14 00:00:00	1957	-4	15722	8673	0	1	0	
4	2031	2015-10-25 00:00:00	2182	1	15722	8673	1	0	0	

	index_y	ng_team_api_id2	ng_DIFF	ng_opponent	ng_W	ng_L	ng_T	\
0	2005.0	1601.0	-1.0	2186.0	0.0	1.0	0.0	
1	2133.0	1601.0	-1.0	2186.0	0.0	1.0	0.0	
2	2218.0	1957.0	-2.0	8025.0	0.0	1.0	0.0	

```
3   2114.0           1957.0      1.0       8025.0   1.0   0.0   0.0
4   2034.0           2182.0      0.0       8025.0   0.0   0.0   1.0

   team_season_DIFF  dom_season_DIFF
0                -9               30
1                -9               30
2               -17               30
3               -17               30
4                -1               30
```

[16]:
```
merge_table = pd.merge(merge_table_b, df_DIFF,left_on="ng_opponent",
right_on="team_api_id")
merge_table = merge_table.rename(columns={"total_DIFF":
"ng_opp_season_DIFF"})
merge_table.head(5)
```

[16]:
```
   index_x                 date  team  DIFF  league_id  dom_team  W  L  T  \
0     2231  2015-09-20 00:00:00  1601    -3      15722      8673  0  1  0
1     2127  2016-02-28 00:00:00  1601    -2      15722      8673  0  1  0
2     2222  2015-09-11 00:00:00  8021     0      15722      8673  0  0  1
3     2118  2016-02-21 00:00:00  8021    -1      15722      8673  0  1  0
4     2086  2015-07-26 00:00:00  8033    -5      15722      8673  0  1  0

   index_y  ng_team_api_id2  ng_DIFF  ng_opponent  ng_W  ng_L  ng_T  \
0   2005.0           1601.0     -1.0       2186.0   0.0   1.0   0.0
1   2133.0           1601.0     -1.0       2186.0   0.0   1.0   0.0
2   2228.0           8021.0      2.0       2186.0   1.0   0.0   0.0
3   2124.0           8021.0      1.0       2186.0   1.0   0.0   0.0
4   2172.0           8033.0     -1.0       2186.0   0.0   1.0   0.0

   team_season_DIFF  dom_season_DIFF  team_api_id  ng_opp_season_DIFF
0                -9               30         2186                  15
1                -9               30         2186                  15
2                 4               30         2186                  15
3                 4               30         2186                  15
4                -9               30         2186                  15
```

[17]:
```
merge_table.drop(['team_api_id'], axis=1, inplace=True)

#merge_table.value_count("ng_opponent")
#merge_table.ng_opponent.value_counts()
merge_table = merge_table.rename(columns={"ng_team_api_id2": "shift_team",
"league_id": "league",
"ng_opponent": "ng_oppo",
"team_season_DIFF": "team_szn_DIFF",
"dom_season_DIFF": "dom_szn_DIFF",
"ng_opp_season_DIFF":
"ng_oppo_szn_DIFF" })
```

[18]:
```
### DATA TABLES

# df_teams_WLT
# Has the total W/L/T and winning percentage of each team

# merge_table
# Has all the rows

# Note:
# shift_team is the team we merged on when we shifted. If it's not the same
# as team it is the last game of the season
```

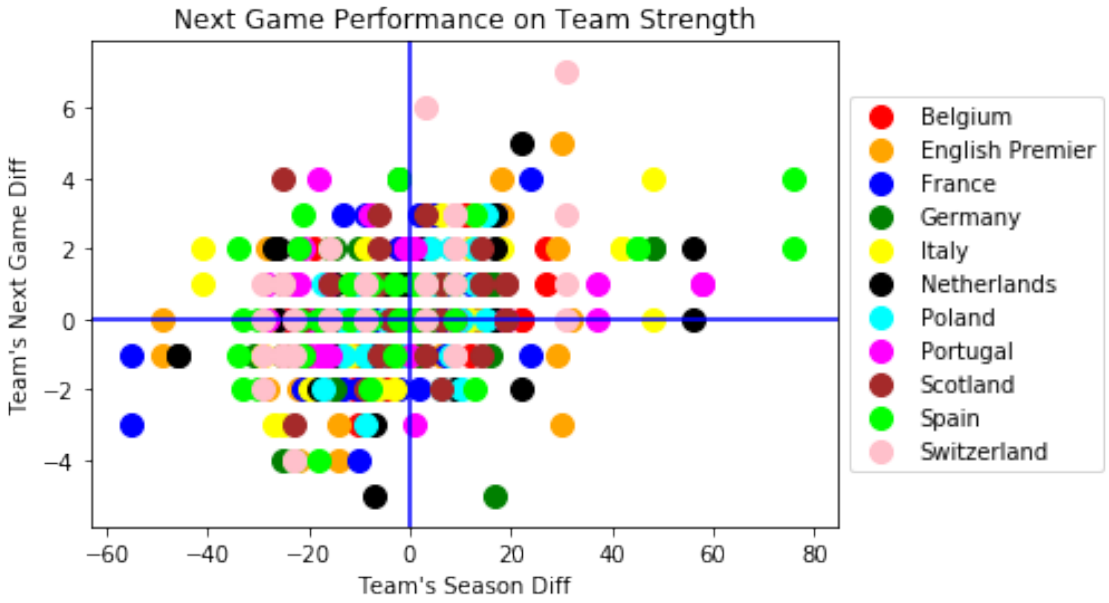
[19]:
```
LEAGUE_df
```

[19]:

	id	country_id	name
0	1	1	Belgium Jupiler League
1	1729	1729	England Premier League
2	4769	4769	France Ligue 1
3	7809	7809	Germany 1. Bundesliga
4	10257	10257	Italy Serie A
5	13274	13274	Netherlands Eredivisie
6	15722	15722	Poland Ekstraklasa
7	17642	17642	Portugal Liga ZON Sagres
8	19694	19694	Scotland Premier League
9	21518	21518	Spain LIGA BBVA
10	24558	24558	Switzerland Super League

[20]:
```
# scatter_x = merge_table["team_szn_DIFF"]
# scatter_y = merge_table["ng_DIFF"]
# group = merge_table["league"]
# group_unique = group.unique()
# cdict = {1:"red",1729:"orange",4769:"blue",7809:"green",10257:"yellow",13274:
 ↪"black",15722:"cyan",17642:"magenta",19694:"brown",21518:"lime",24558:"pink"}
# ldict = {1:"Belgium",1729:"Eng Premier",4769:"France",7809:"Germany",10257:
 ↪"Italy",13274:"Netherlands",15722:"Poland",17642:"Portugal",19694:
 ↪"Scotland",21518:"Spain",24558:"Switzerland"}
# fig, ax = plt.subplots()
# for index, row in merge_table.iterrows():
#     g = group[index]
#     ax.scatter(scatter_x[index], scatter_y[index], c = cdict[g], label =␣
 ↪ldict[g], s = 100)
l1 = plt.scatter(merge_table.loc[merge_table["league"]==1,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==1, ["ng_DIFF"]], c␣
 ↪= "r", label = "Belgium", s = 100)
l2 = plt.scatter(merge_table.loc[merge_table["league"]==1729,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==1729,␣
 ↪["ng_DIFF"]], c = "orange", label = "English Premier", s = 100)
```

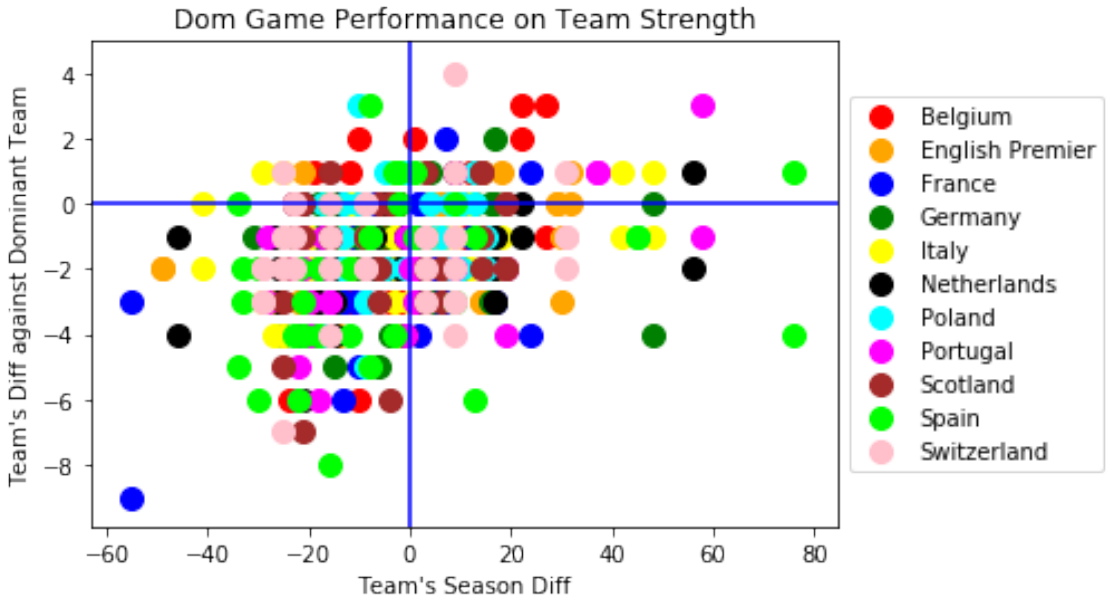
```
l3 = plt.scatter(merge_table.loc[merge_table["league"]==4769,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==4769,␣
 ↪["ng_DIFF"]], c = "blue", label = "France", s = 100)
l4 = plt.scatter(merge_table.loc[merge_table["league"]==7809,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==7809,␣
 ↪["ng_DIFF"]], c = "green", label = "Germany", s = 100)
l5 = plt.scatter(merge_table.loc[merge_table["league"]==10257,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==10257,␣
 ↪["ng_DIFF"]], c = "yellow", label = "Italy", s = 100)
l6 = plt.scatter(merge_table.loc[merge_table["league"]==13274,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==13274,␣
 ↪["ng_DIFF"]], c = "black", label = "Netherlands", s = 100)
l7 = plt.scatter(merge_table.loc[merge_table["league"]==15722,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==15722,␣
 ↪["ng_DIFF"]], c = "cyan", label = "Poland", s = 100)
l8 = plt.scatter(merge_table.loc[merge_table["league"]==17642,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==17642,␣
 ↪["ng_DIFF"]], c = "magenta", label = "Portugal", s = 100)
l9 = plt.scatter(merge_table.loc[merge_table["league"]==19694,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==19694,␣
 ↪["ng_DIFF"]], c = "brown", label = "Scotland", s = 100)
l10 = plt.scatter(merge_table.loc[merge_table["league"]==21518,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==21518,␣
 ↪["ng_DIFF"]], c = "lime", label = "Spain", s = 100)
l11 = plt.scatter(merge_table.loc[merge_table["league"]==24558,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==24558,␣
 ↪["ng_DIFF"]], c = "pink", label = "Switzerland", s = 100)

plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.xlabel("Team's Season Diff")
plt.ylabel("Team's Next Game Diff")
plt.axhline(0, color='b')
plt.axvline(0, color='b')
plt.title("Next Game Performance on Team Strength")
plt.show()
# plt.scatter(merge_table["team_szn_DIFF"], merge_table["ng_DIFF"], marker="o",␣
 ↪facecolors="red", edgecolors="black",
#             alpha=0.75)
```



[21]:
```
#plt.scatter(merge_table['team_szn_DIFF'], merge_table['DIFF'],␣
 ↪label=merge_table['league'])
```

[22]:
```
# scatter_x = merge_table["team_szn_DIFF"]
# scatter_y = merge_table["DIFF"]
# group = merge_table["league"]
# group_unique = group.unique()
# cdict = {1:"red",1729:"orange",4769:"blue",7809:"green",10257:"yellow",13274:
 ↪"black",15722:"cyan",17642:"magenta",19694:"brown",21518:"lime",24558:"pink"}
# fig, ax = plt.subplots()
# for index, row in merge_table.iterrows():
#     g = group[index]
#     ax.scatter(scatter_x[index], scatter_y[index], c = cdict[g], s = 100)
    #ax.scatter(merge_table['team_szn_DIFF'], merge_table['DIFF'], c=cdict[g],␣
 ↪s=100)
n1 = plt.scatter(merge_table.loc[merge_table["league"]==1,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==1, ["DIFF"]], c =␣
 ↪"r", label = "Belgium", s = 100)
n2 = plt.scatter(merge_table.loc[merge_table["league"]==1729,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==1729, ["DIFF"]], c␣
 ↪= "orange", label = "English Premier", s = 100)
n3 = plt.scatter(merge_table.loc[merge_table["league"]==4769,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==4769, ["DIFF"]], c␣
 ↪= "blue", label = "France", s = 100)
n4 = plt.scatter(merge_table.loc[merge_table["league"]==7809,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==7809, ["DIFF"]], c␣
 ↪= "green", label = "Germany", s = 100)
```

```
n5 = plt.scatter(merge_table.loc[merge_table["league"]==10257,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==10257, ["DIFF"]],␣
 ↪c = "yellow", label = "Italy", s = 100)
n6 = plt.scatter(merge_table.loc[merge_table["league"]==13274,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==13274, ["DIFF"]],␣
 ↪c = "black", label = "Netherlands", s = 100)
n7 = plt.scatter(merge_table.loc[merge_table["league"]==15722,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==15722, ["DIFF"]],␣
 ↪c = "cyan", label = "Poland", s = 100)
n8 = plt.scatter(merge_table.loc[merge_table["league"]==17642,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==17642, ["DIFF"]],␣
 ↪c = "magenta", label = "Portugal", s = 100)
n9 = plt.scatter(merge_table.loc[merge_table["league"]==19694,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==19694, ["DIFF"]],␣
 ↪c = "brown", label = "Scotland", s = 100)
n10 = plt.scatter(merge_table.loc[merge_table["league"]==21518,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==21518, ["DIFF"]],␣
 ↪c = "lime", label = "Spain", s = 100)
n11 = plt.scatter(merge_table.loc[merge_table["league"]==24558,␣
 ↪["team_szn_DIFF"]],merge_table.loc[merge_table["league"]==24558, ["DIFF"]],␣
 ↪c = "pink", label = "Switzerland", s = 100)

plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.xlabel("Team's Season Diff")
plt.ylabel("Team's Diff against Dominant Team")
plt.axhline(0, color='b')
plt.axvline(0, color='b')
plt.title("Dom Game Performance on Team Strength")
plt.show()
#plt.scatter(merge_table["team_szn_DIFF"], merge_table["DIFF"], marker="o",␣
 ↪facecolors="red", edgecolors="black",alpha=0.75)
```

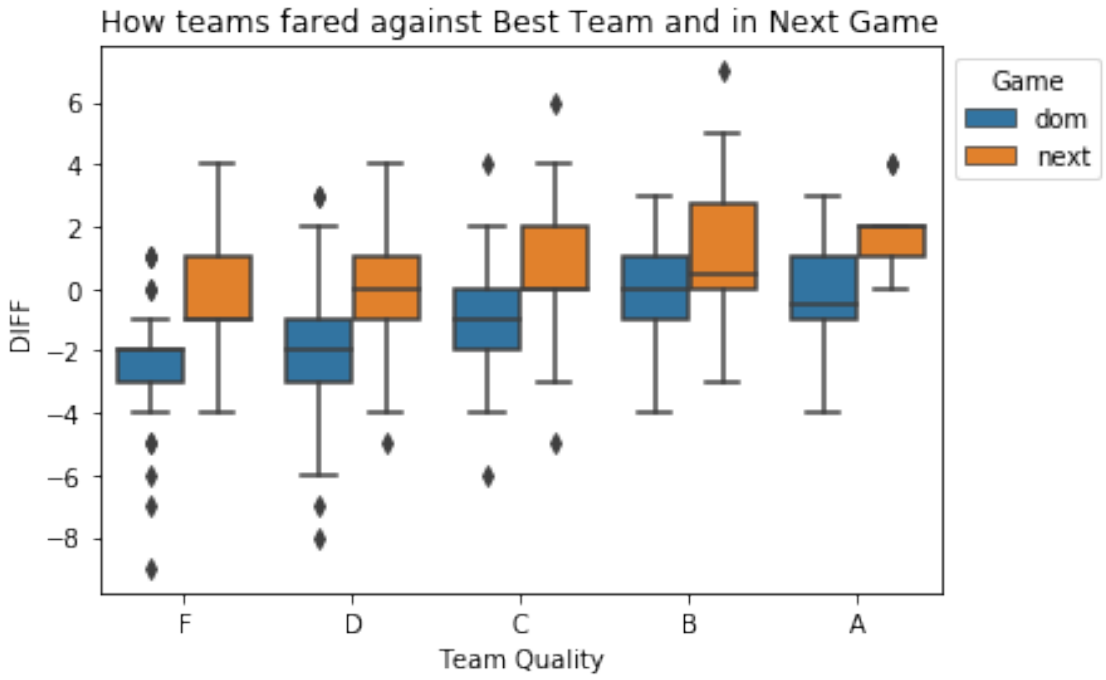


[23]:
```
bins = [-60, -25, 0, 20, 35, 80]
group_names = ["F", "D", "C", "B", "A"]
merge_table["Team Quality"] = pd.cut(merge_table["team_szn_DIFF"], bins,␣
 ↪labels=group_names)
merge_table.head()
#carving out dom_game and next_game results from merge_table
#then concatenating them with an added column to identify "Dom" or "Next"
#so we can arrange two boxplots in the same chart
dom_game = merge_table[["Team Quality","DIFF"]]
dom_game["Game"] = "dom"
dom_game.head()
next_game = merge_table[["Team Quality", "ng_DIFF"]]
next_game = next_game.rename(columns={"ng_DIFF":"DIFF"})
next_game["Game"] = "next"
both_games = pd.concat([dom_game,next_game],sort=False)
both_games.head()
```

```
C:\Users\Shalesh Kumbhat\Anaconda3\envs\PythonData\lib\site-
packages\ipykernel_launcher.py:9: 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: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
  if __name__ == '__main__':
```

[23]:
```
  Team Quality  DIFF Game
0            D  -3.0  dom
1            D  -2.0  dom
2            C   0.0  dom
3            C  -1.0  dom
4            D  -5.0  dom
```

[24]:
```
#data = both_games.melt(id_vars=['Team Quality'], var_name='Game',␣
 ↪value_name='DIFF')
sns.boxplot(data=both_games, x='Team Quality', y='DIFF', hue='Game')
plt.legend(title='Game', loc='upper left', bbox_to_anchor=(1, 1))
plt.title("How teams fared against Best Team and in Next Game")
plt.savefig("Dom-Next-Game BoxPlot.png")
```



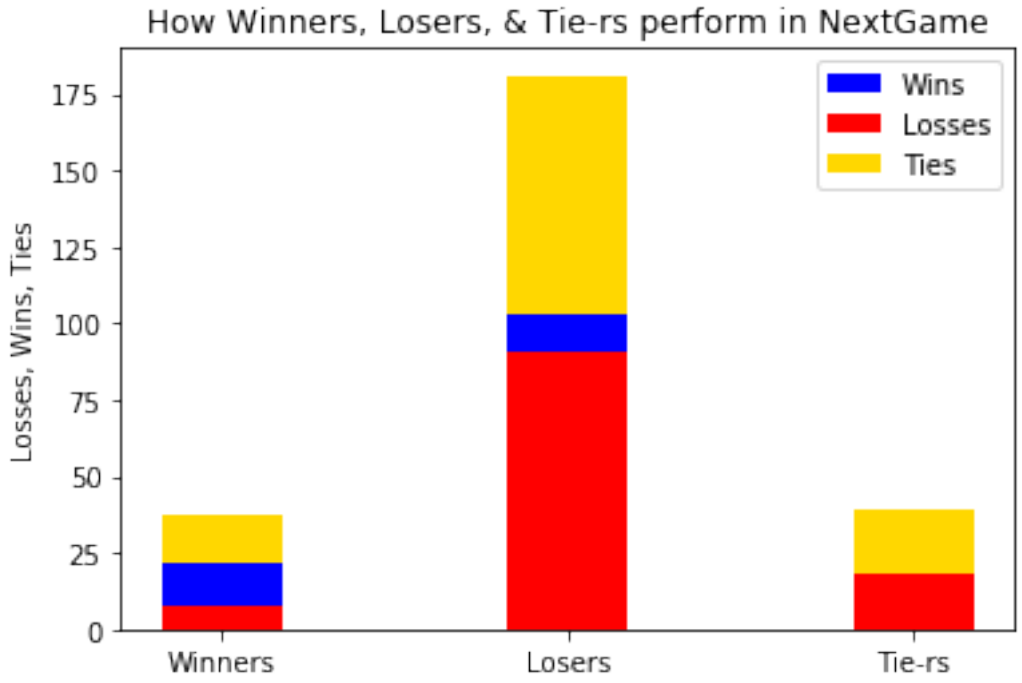
[25]:
```
# ng_bplot = sns.boxplot(y='ng_DIFF', x='Team Quality', data=merge_table,␣
 ↪width=0.5, palette="colorblind")
# plt.ylabel("Goal Differential")
# plt.title("How various quality teams fared in the next game")
# plt.axhline(0, color='b')
# plt.savefig("NextGame BoxPlot.png")

# dom_bplot = sns.boxplot(y='DIFF', x='Team Quality', data=merge_table, width=0.
 ↪5, palette="colorblind")
# plt.ylabel("Goal Differential")
# plt.title("How various quality teams fared against the Dominant Team")
```

```
# plt.axhline(0, color='b')
# plt.savefig("DomGame BoxPlot.png")
```

[26]:
```
winners_df = merge_table.loc[merge_table["DIFF"] > 0, ["team", "ng_DIFF",␣
 ↪"team_season_DIFF","Team Quality"]]
winners_wins = winners_df[winners_df["ng_DIFF"]>0]["ng_DIFF"].count()
winners_losses = winners_df[winners_df["ng_DIFF"]<0]["ng_DIFF"].count()
winners_ties = winners_df[winners_df["ng_DIFF"]==0]["ng_DIFF"].count()
winners_results = [winners_wins,winners_losses,winners_ties]

losers_df = merge_table.loc[merge_table["DIFF"] < 0, ["team", "ng_DIFF",␣
 ↪"team_season_DIFF","Team Quality"]]
losers_wins = losers_df[losers_df["ng_DIFF"]>0]["ng_DIFF"].count()
losers_losses = losers_df[losers_df["ng_DIFF"]<0]["ng_DIFF"].count()
losers_ties = losers_df[losers_df["ng_DIFF"]==0]["ng_DIFF"].count()
losers_results = [losers_wins,losers_losses,losers_ties]

ties_df = merge_table.loc[merge_table["DIFF"] == 0, ["team", "ng_DIFF",␣
 ↪"team_season_DIFF","Team Quality"]]
ties_wins = ties_df[ties_df["ng_DIFF"]>0]["ng_DIFF"].count()
ties_losses = ties_df[ties_df["ng_DIFF"]<0]["ng_DIFF"].count()
ties_ties = ties_df[ties_df["ng_DIFF"]==0]["ng_DIFF"].count()
ties_results = [ties_wins,ties_losses,ties_ties]
results_df = pd.DataFrame({'Winners': winners_results, 'Losers':␣
 ↪losers_results, 'Tie-rs': ties_results})
results_df = results_df.transpose()
results_df = results_df.rename(columns={"0":"Wins", "1":"Losses", "2":"Ties"})
results_df

N = 3
ind = np.arange(N)
width = 0.35
p1 = plt.bar(ind, results_df[0], width, color = "blue")
p2 = plt.bar(ind, results_df[1], width, color = "red")
p3 = plt.bar(ind, results_df[2], width, color = "gold", bottom=results_df[0])
plt.ylabel('Losses, Wins, Ties')
plt.title('How Winners, Losers, & Tie-rs perform in NextGame')
plt.xticks(ind, ('Winners', 'Losers', 'Tie-rs'))
#plt.yticks(np.arange(0, 81, 10))
plt.legend((p1[0], p2[0], p3[0]), ('Wins', 'Losses','Ties'))
plt.show()
plt.savefig("NextGame-Stacked.png")
```

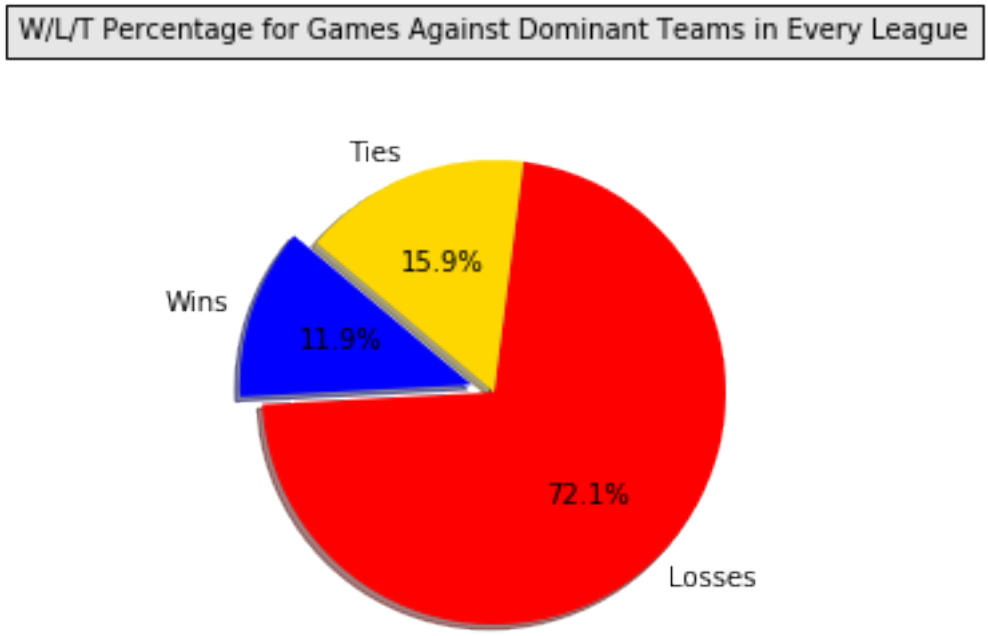
```
C:\Users\Shalesh Kumbhat\Anaconda3\envs\PythonData\lib\site-
packages\pandas\core\indexing.py:1494: FutureWarning:
Passing list-likes to .loc or [] with any missing label will raise
KeyError in the future, you can use .reindex() as an alternative.
```

```
See the documentation here:
https://pandas.pydata.org/pandas-docs/stable/indexing.html#deprecate-loc-
reindex-listlike
  return self._getitem_tuple(key)
```

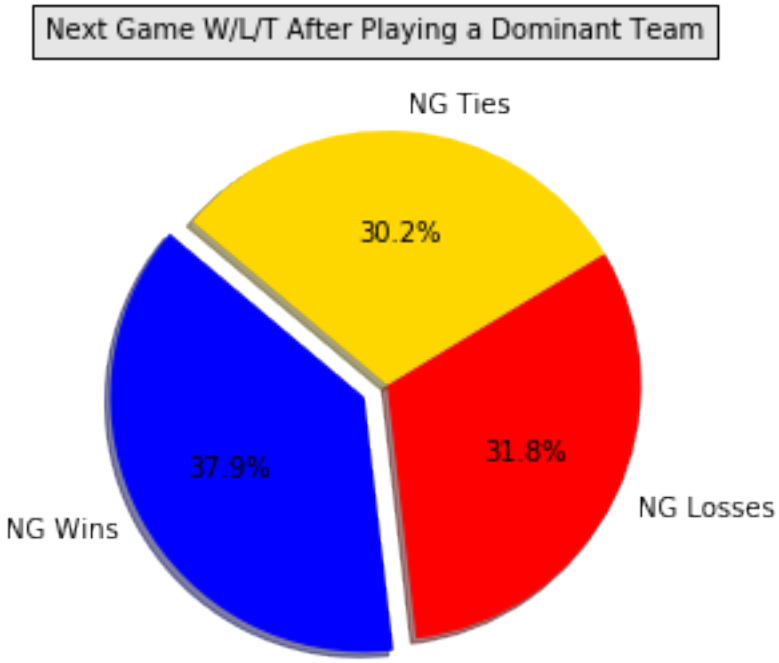



```
<Figure size 432x288 with 0 Axes>
```

[27]:
```
# Dom team game
#plt.figure(1)
labels = ['Wins', 'Losses', 'Ties']
sizes = [11.94, 72.15, 15.92]
colors = ['blue', 'red', 'gold']
explode = (0.1, 0, 0)  # explode 1st slice
p1 = plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.
 ↪1f%%', shadow=True, startangle=140)
plt.title("W/L/T Percentage for Games Against Dominant Teams in Every League",␣
 ↪bbox={'facecolor':'0.9', 'pad':5}, fontsize=10, y=1.08)

#plt.figure(2)

plt.show()
# plt.savefig("DomGame PieChart.png")
```



[28]:
```
# Next game
labels1 = ['NG Wins', 'NG Losses', 'NG Ties']
sizes1 = [37.93, 31.83, 30.24]
colors1 = ['blue', 'red', 'gold']
explode1 = (0.1, 0, 0)  # explode 1st slice
plt.pie(sizes1, explode=explode1, labels=labels1, colors=colors1,
autopct='%1.1f%%', shadow=True, startangle=140)
plt.axis('equal')
# plt.title(figure_title, )
plt.title("Next Game W/L/T After Playing a Dominant Team", bbox={'facecolor':'0.
 ↪9', 'pad':5}, fontsize=10, y=1.08)
plt.show()
plt.savefig("NextGame PieChart.png")
```




```
<Figure size 432x288 with 0 Axes>
```

[29]:
```
# df_teams.head() has the
df_teams_WLT = df_teams[["league_id","team_api_id","W","L","T"]].copy()
#df_teams_WLT["win_%"]=df_teams_WLT[['T']] + df_teams_WLT[['L']]
#+ df_teams_WLT.L + df_teams_WLT.L
#df_teams_WLT.head()
#df_teams_WLT[["win_percent"]]=df_teams_WLT["W"


df_teams_WLT['total_games'] = df_teams_WLT.apply(lambda x: x['W'] + x['L']
+ x['T'], axis=1)
df_teams_WLT['win_rate'] = df_teams_WLT.apply(lambda x: x['W'] /
x['total_games'], axis=1)
df_teams_WLT['tie_rate'] = df_teams_WLT.apply(lambda x: x['T'] /
x['total_games'], axis=1)
df_teams_WLT['loss_rate'] = df_teams_WLT.apply(lambda x: x['L'] /
x['total_games'], axis=1)
df_teams_WLT.head()
```

[29]:

	league_id	team_api_id	W	L	T	total_games	win_rate	tie_rate	\
0	1	1773	7	15	8	30	0.233333	0.266667	
1	1	8203	10	13	7	30	0.333333	0.233333	
2	1	8342	21	8	1	30	0.700000	0.033333	

```
3          1         8475   9  15  6           30  0.300000  0.200000
4          1         8571  10  11  9           30  0.333333  0.300000

   loss_rate
0   0.500000
1   0.433333
2   0.266667
3   0.500000
4   0.366667
```

[30]:
```
df_teams_WLT["points"] = df_teams_WLT["W"]*3 + df_teams_WLT["T"]
bins = [0, 36, 42, 49, 60.0,999]
group_names = ["F", "D", "C", "B", "A"]
df_teams_WLT["Team Win Pctg"] = pd.cut(df_teams_WLT["points"], bins,␣
 ↪labels=group_names)
df_teams_WLT.head(5)
#plt.bar(df_teams_WLT["Team Win Pctg"], merge_table["ng_WIN"].sum(), color='r',␣
 ↪alpha=0.5, align="center")
#df_teams_WLT["points"] = (df_teams_WLT.W*3 + df_teams_WLT.T)
#+ df_teams_WLT["T"]
#df_teams_WLT["points"] = df_teams_WLT["points"] + df_teams_WLT["tie_points"]

df_teams_WLT.head()
```

[30]:
```
   league_id  team_api_id   W   L  T  total_games  win_rate  tie_rate  \
0          1         1773   7  15  8           30  0.233333  0.266667
1          1         8203  10  13  7           30  0.333333  0.233333
2          1         8342  21   8  1           30  0.700000  0.033333
3          1         8475   9  15  6           30  0.300000  0.200000
4          1         8571  10  11  9           30  0.333333  0.300000

   loss_rate  points Team Win Pctg
0   0.500000      29             F
1   0.433333      37             D
2   0.266667      64             A
3   0.500000      33             F
4   0.366667      39             D
```

[31]:
```
#df_teams_WLT.head()

#60     / 49   /42      / 36
# 37.6 / 75.2 / 112.8 / 150

#df_teams[df_teams_WLT.points > 40].count()#  122
#df_teams[df_teams_WLT.points > 59].count() #36

#df_teams[df_teams_WLT.points > 60].count() #36
```

```
#df_teams[df_teams_WLT.points > 49].count() #73
#df_teams[df_teams_WLT.points > 42].count() #113
#df_teams[df_teams_WLT.points > 36].count() #### 150
#94



#df_teams[df_teams_WLT.points > 70].count() #94

#df_teams[df_teams_WLT.points > 50].count()  62


#df_teams[df_teams_WLT.loss_rate > .56].count()  #10 for 05%
# df_teams[df_teams_WLT.loss_rate > .48].count()  #34

#------# df_teams[df_teams_WLT.loss_rate > .48].count()  #34
#------#df_teams[df_teams_WLT.loss_rate > .42].count()
#------df_teams[df_teams_WLT.loss_rate > .36].count()  #141 for 75%  113
# ---df_teams[df_teams_WLT.loss_rate > .28].count()  #141 for 75%  141
```

[32]:
```
#merge_table # 377
#merge_table_bins = pd.merge(merge_table, df_teams_WLT,left_on="team_api_id",␣
 ↪right_on="team_api_id")
#merge_table_bins.head()
#df_teams_WLT=df_teams_WLT.sort_values(by=["team_api_id"])
# 188
#all_games = all_games_unsorted.sort_values(by=["team_api_id","date"])
#df_teams_WLT # 188 rows

merge_table_group = merge_table.groupby(['league','team'], as_index=False).sum()

merge_table_group['total_ng'] = merge_table_group.apply(lambda x: x['W'] +␣
 ↪x['L'] + x['T'], axis=1)
merge_table_group['win_rate_ng'] = merge_table_group.apply(lambda x: x['W'] /␣
 ↪x['total_ng'], axis=1)
merge_table_group['loss_rate_ng'] = merge_table_group.apply(lambda x: x['L'] /␣
 ↪x['total_ng'], axis=1)
merge_table_group['tie_rate_ng'] = merge_table_group.apply(lambda x: x['T'] /␣
 ↪x['total_ng'], axis=1)

merge_table_group.head(5)


#merge_table_group # 177 -  11

#df_teams_WLT.head() #  team_api_id
#merge_table_group.head()  # team
```

```
merge_table_bins = pd.merge(merge_table_group, df_teams_WLT,left_on="team",␣
 ↪right_on="team_api_id")
merge_table_bins.columns
# team / BIN / LEAGUE /


#df_teams_WLT # 188

#df_teams_WLT.sort_values(by=["team_api_id"])


#unique_df_teams  = df_teams.team_api_id.unique()
#unique_df_teams
```

[32]:
```
Index(['league', 'team', 'index_x', 'DIFF', 'dom_team', 'W_x', 'L_x', 'T_x',
       'index_y', 'shift_team', 'ng_DIFF', 'ng_oppo', 'ng_W', 'ng_L', 'ng_T',
       'team_szn_DIFF', 'dom_szn_DIFF', 'ng_oppo_szn_DIFF', 'total_ng',
       'win_rate_ng', 'loss_rate_ng', 'tie_rate_ng', 'league_id',
       'team_api_id', 'W_y', 'L_y', 'T_y', 'total_games', 'win_rate',
       'tie_rate', 'loss_rate', 'points', 'Team Win Pctg'],
      dtype='object')
```

[33]:
```
merge_table_bins.
 ↪drop(['shift_team','ng_oppo','dom_team','league','team_api_id'], axis=1,␣
 ↪inplace=True)
merge_table_bins.head(6)
```

[33]:

	team	index_x	DIFF	W_x	L_x	T_x	index_y	ng_DIFF	ng_W	ng_L	...	\
0	1773	32	-2	0	1	0	45.0	-2.0	0.0	1.0	...	
1	8203	155	-6	0	2	0	256.0	3.0	2.0	0.0	...	
2	8475	350	-5	0	2	0	368.0	-3.0	0.0	2.0	...	
3	8571	300	-4	0	2	0	322.0	-1.0	0.0	1.0	...	
4	8573	163	-3	0	2	0	179.0	3.0	1.0	0.0	...	
5	8635	120	5	2	0	0	99.0	0.0	0.0	0.0	...	

	league_id	W_y	L_y	T_y	total_games	win_rate	tie_rate	loss_rate	\
0	1	7	15	8	30	0.233333	0.266667	0.500000	
1	1	10	13	7	30	0.333333	0.233333	0.433333	
2	1	9	15	6	30	0.300000	0.200000	0.500000	
3	1	10	11	9	30	0.333333	0.300000	0.366667	
4	1	14	9	7	30	0.466667	0.233333	0.300000	
5	1	15	5	10	30	0.500000	0.333333	0.166667	

	points	Team Win Pctg
0	29	F
1	37	D
2	33	F

```
3      39              D
4      49              C
5      55              B

[6 rows x 28 columns]
```

[34]:
```
merge_table_group_bin = merge_table_bins.groupby(['Team Win Pctg'],␣
 ↪as_index=False).sum()
merge_table_group_bin_mean = merge_table_bins.groupby(['Team Win Pctg'],␣
 ↪as_index=False).mean()

merge_table_group_bin.head(12)
merge_table_group_bin.columns
#merge_table_group_bin.
 ↪drop(['league','team','dom_team','shift_team','team_api_id'], axis=1,␣
 ↪inplace=True)
merge_table_group_bin_mean.head()
```

[34]:
```
  Team Win Pctg          team      index_x      DIFF       W_x       L_x  \
0             F  24965.710526  4028.157895 -4.315789  0.105263  1.710526
1             D  14273.621622  3227.810811 -4.351351  0.135135  1.621622
2             C   8507.725000  3671.475000 -3.225000  0.125000  1.575000
3             B  13264.777778  3926.888889 -2.722222  0.388889  1.444444
4             A   9259.307692  3925.807692 -1.846154  0.653846  1.230769

        T_x      index_y   ng_DIFF      ng_W  ...  tie_rate_ng     league_id  \
0  0.315789  4037.789474 -0.657895  0.473684  ...     0.129386  12545.947368
1  0.243243  3249.756757 -0.648649  0.621622  ...     0.108108  10793.621622
2  0.375000  3676.350000  0.575000  0.800000  ...     0.168750  12033.850000
3  0.416667  3944.916667  0.583333  0.944444  ...     0.208333  11275.666667
4  0.346154  3946.000000  2.730769  1.384615  ...     0.153846  11370.192308

         W_y        L_y       T_y  total_games  win_rate  tie_rate  loss_rate  \
0   7.263158  17.973684  8.710526    33.947368  0.216251  0.257968   0.525781
1   9.756757  15.189189  9.918919    34.864865  0.282627  0.285042   0.432331
2  11.925000  13.850000  9.925000    35.700000  0.337515  0.277784   0.384701
3  14.888889  11.777778  9.166667    35.833333  0.418803  0.255567   0.325629
4  20.846154   7.730769  8.576923    37.153846  0.563878  0.229430   0.206693

      points
0  30.500000
1  39.189189
2  45.700000
3  53.833333
4  71.115385

[5 rows x 28 columns]
```

[35]:
```
merge_table_group_bin_mean.
 ↪drop(['team','DIFF','W_x','L_x','T_x','ng_DIFF','ng_W','ng_L'], axis=1,␣
 ↪inplace=True)
merge_table_group_bin_mean.head()
```

[35]:

	Team Win Pctg	index_x	index_y	ng_T	team_szn_DIFF
0	F	4028.157895	4037.789474	0.789474	-49.657895
1	D	3227.810811	3249.756757	0.432432	-25.864865
2	C	3671.475000	3676.350000	0.700000	-8.425000
3	B	3926.888889	3944.916667	0.694444	9.305556
4	A	3925.807692	3946.000000	0.576923	62.000000

	dom_szn_DIFF	ng_oppo_szn_DIFF	total_ng	win_rate_ng	loss_rate_ng
0	112.605263	-16.078947	2.131579	0.052632	0.817982
1	107.837838	-14.486486	2.000000	0.074324	0.817568
2	123.925000	-9.800000	2.075000	0.062500	0.768750
3	136.111111	-7.972222	2.250000	0.171296	0.620370
4	132.538462	-27.269231	2.230769	0.298077	0.548077

	tie_rate_ng	league_id	W_y	L_y	T_y	total_games
0	0.129386	12545.947368	7.263158	17.973684	8.710526	33.947368
1	0.108108	10793.621622	9.756757	15.189189	9.918919	34.864865
2	0.168750	12033.850000	11.925000	13.850000	9.925000	35.700000
3	0.208333	11275.666667	14.888889	11.777778	9.166667	35.833333
4	0.153846	11370.192308	20.846154	7.730769	8.576923	37.153846

	win_rate	tie_rate	loss_rate	points
0	0.216251	0.257968	0.525781	30.500000
1	0.282627	0.285042	0.432331	39.189189
2	0.337515	0.277784	0.384701	45.700000
3	0.418803	0.255567	0.325629	53.833333
4	0.563878	0.229430	0.206693	71.115385

[36]:
```
# merge_table_group_bin.head()

# merge_table_group_bin.columns

merge_table_group_bin['total_ng'] = merge_table_group_bin.apply(lambda x:␣
 ↪x['ng_W'] + x['ng_L'] + x['ng_T'], axis=1)
merge_table_group_bin['win_rate_ng'] = merge_table_group_bin.apply(lambda x:␣
 ↪x['ng_W'] / x['total_ng'], axis=1)
merge_table_group_bin['loss_rate_ng'] = merge_table_group_bin.apply(lambda x:␣
 ↪x['ng_L'] / x['total_ng'], axis=1)
merge_table_group_bin['tie_rate_ng'] = merge_table_group_bin.apply(lambda x:␣
 ↪x['ng_T'] / x['total_ng'], axis=1)

merge_table_group_bin.head()
```

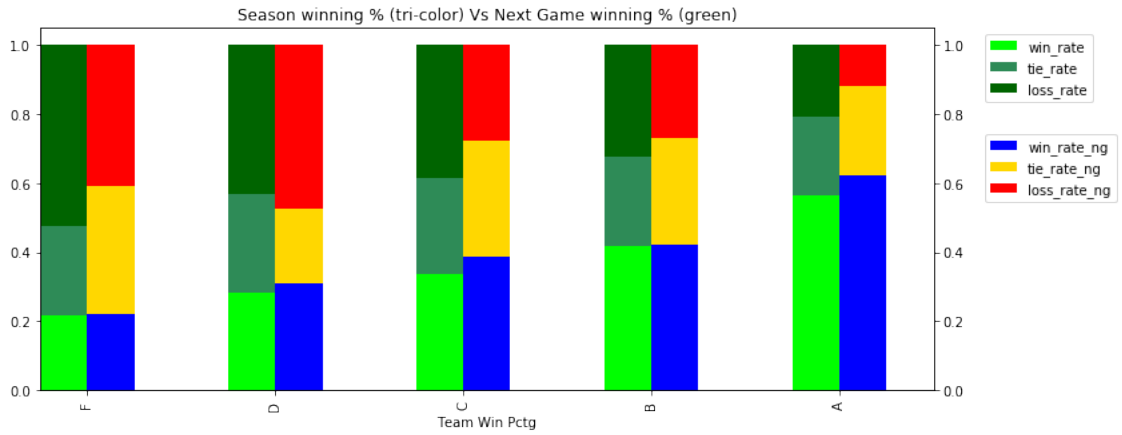
[36]:
```
  Team Win Pctg    team  index_x  DIFF  W_x  L_x  T_x   index_y  ng_DIFF  \
0             F  948697   153070  -164    4   65   12  153436.0    -25.0
1             D  528124   119429  -161    5   60    9  120241.0    -24.0
2             C  340309   146859  -129    5   63   15  147054.0     23.0
3             B  477532   141368   -98   14   52   15  142017.0     21.0
4             A  240742   102071   -48   17   32    9  102596.0     71.0

   ng_W  ...  tie_rate_ng  league_id  W_y  L_y  T_y  total_games   win_rate  \
0  18.0  ...     0.370370     476746  276  683  331         1290   8.217527
1  23.0  ...     0.216216     399364  361  562  367         1290  10.457207
2  32.0  ...     0.337349     481354  477  554  397         1428  13.500619
3  34.0  ...     0.308642     405924  536  424  330         1290  15.076918
4  36.0  ...     0.258621     295625  542  201  223          966  14.660819

    tie_rate  loss_rate  points
0   9.802786  19.979687    1159
1  10.546560  15.996233    1450
2  11.111352  15.388029    1828
3   9.200430  11.722652    1938
4   5.965170   5.374011    1849

[5 rows x 28 columns]
```

[37]:
```
merge_table_group_bin_mean.set_index('Team Win Pctg',inplace=True)
merge_table_group_bin.set_index('Team Win Pctg',inplace=True)
```

[38]:
```
fig=plt.figure()

ax=fig.add_subplot(111)
ax2=ax.twinx()


width=.25

#ax = x.plot(kind='bar', legend=False)
patches, labels = ax.get_legend_handles_labels()
#ax.legend(patches, labels, loc='best')
ax.legend(patches, labels, loc='lower left')
ax2.legend(patches, labels, loc='lower left')


colors = ["lime","seagreen","darkgreen"]
colors2 = ["blue", "gold","red"]

merge_table_group_bin_mean.loc[:,['win_rate','tie_rate','loss_rate']].plot.
 ↪bar(stacked=True,width=width,position=1, color=colors, ax=ax,align='center')
#figsize=(15,15),
```

```
#merge_table_group_bin.loc[:,['ng_W','ng_L','ng_T']].plot.
 ↪bar(stacked=True,width=width, color=colors2,␣
 ↪figsize=(15,10),position=1,ax=ax2)
merge_table_group_bin.loc[:,['win_rate_ng','tie_rate_ng','loss_rate_ng']].plot.
 ↪bar(stacked=True,width=width,position=0, color=colors2,␣
 ↪figsize=(12,5),ax=ax2,align='center')

ax.legend(patches, labels, loc='best')
ax2.legend(patches, labels, loc='right')
#merge_table_group_bin.loc[:,['W_x','L_x','T_x']].plot.bar(stacked=True,␣
 ↪color=colors, figsize=(10,7),position=1,ax=ax)

#plt.legend(handles,labels, bbox_to_anchor=(0.85,1.025), loc="upper left")

#merge_table_group_bin.loc[:,['W_x','L_x','T_x']].plot.bar(stacked=True,␣
 ↪color=colors, figsize=(10,7))

#plt.show()
#merge_table_group_bin
ax.legend(fancybox=True, framealpha=0.5)
ax.legend(loc='upper left',bbox_to_anchor=(1.05, 1))
ax2.legend(fancybox=True, framealpha=0.5)
ax2.legend(loc='lower left',bbox_to_anchor=(1.05,0.5))
plt.title("Season winning % (tri-color) Vs Next Game winning % (green)")


#plt.legend(loc='best')
```

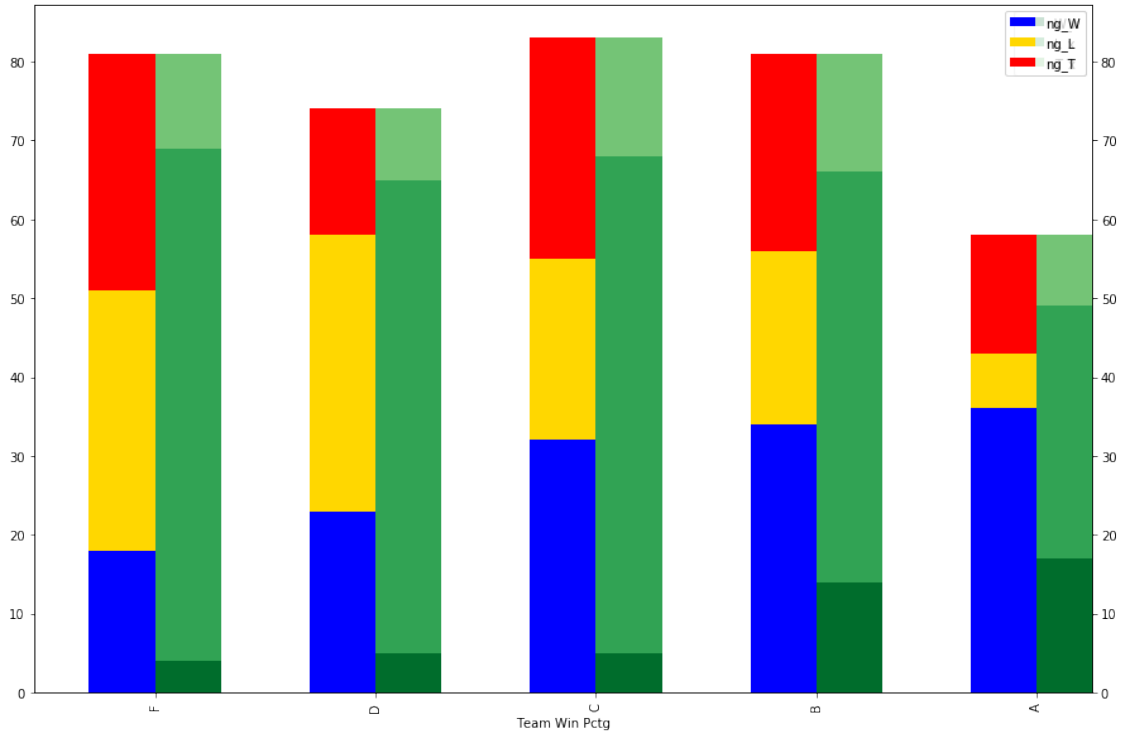
[38]: `Text(0.5, 1.0, 'Season winning % (tri-color) Vs Next Game winning % (green)')`



[39]:
```
fig=plt.figure()

ax=fig.add_subplot(111)
```

```
ax2=ax.twinx()


width=.3

colors = ["#006D2C", "#31A354","#74C476"]
colors2 = ["blue", "gold","red"]

merge_table_group_bin.loc[:,['W_x','L_x','T_x']].plot.
 ↪bar(stacked=True,width=width, color=colors, figsize=(15,10),position=0,ax=ax)
merge_table_group_bin.loc[:,['ng_W','ng_L','ng_T']].plot.
 ↪bar(stacked=True,width=width, color=colors2,␣
 ↪figsize=(15,10),position=1,ax=ax2)
[' ',' ',' '].plot.bar(stacked=True,width=width, color=colors2,␣
 ↪figsize=(15,10),position=2,ax=ax2)


#merge_table_group_bin.loc[:,['W_x','L_x','T_x']].plot.bar(stacked=True,␣
 ↪color=colors, figsize=(10,7),position=1,ax=ax)


#merge_table_group_bin.loc[:,['W_x','L_x','T_x']].plot.bar(stacked=True,␣
 ↪color=colors, figsize=(10,7))

plt.show()
#merge_table_group_bin
```

```
      ␣
↪---------------------------------------------------------------------------

       AttributeError                            Traceback (most recent call␣
↪last)

       <ipython-input-39-5fdce4361001> in <module>
        12 merge_table_group_bin.loc[:,['W_x','L_x','T_x']].plot.
↪bar(stacked=True,width=width, color=colors, figsize=(15,10),position=0,ax=ax)
        13 merge_table_group_bin.loc[:,['ng_W','ng_L','ng_T']].plot.
↪bar(stacked=True,width=width, color=colors2, figsize=(15,10),position=1,ax=ax2)
   ---> 14 [' ',' ',' '].plot.bar(stacked=True,width=width, color=colors2,␣
↪figsize=(15,10),position=2,ax=ax2)
        15
        16


       AttributeError: 'list' object has no attribute 'plot'
```



[ ]:
```
#df_complete.drop(['index_x','index_y','ng_date','ng_league_id'], axis=1,␣
 ↪inplace=True)
merge_table_group_bin.
 ↪drop(['league','team','DIFF','dom_team','shift_team','team_api_id','win_rate','loss_rate','
 ↪axis=1, inplace=True)
merge_table_group_bin.head()
```

[ ]:
```
colors = ["#006D2C", "#31A354","#74C476"]
#merge_table_group_bin.loc[:,['W_x','L_x','T_x']].plot.bar(stacked=True,␣
 ↪color=colors, figsize=(10,7))
#index = ["F","D","C","B","A"]
hello = merge_table_group_bin.loc[:,['ng_W','ng_L','ng_T']]#.plot.
 ↪bar(stacked=True, color=colors, figsize=(10,7))
hello
#hello.plot.bar(stacked=True, color=colors, figsize=(10,7))

#dfx = pd.DataFrame(columns=['a', 'b', 'c', 'd'])
#dfx.plot()

#pd.DataFrame({'lab':['A', 'B', 'C','D','F']})
#merge_table_group_bin.head()
```

[ ]:
```
data = [[2000, 2000, 2000, 2001, 2001, 2001, 2002, 2002, 2002],
        ['Jan', 'Feb', 'Mar', 'Jan', 'Feb', 'Mar', 'Jan', 'Feb', 'Mar'],
        [1, 2, 3, 4, 5, 6, 7, 8, 9]]
```

```
rows = zip(data[0], data[1], data[2])
headers = ['Year', 'Month', 'Value']
df = pd.DataFrame(rows, columns=headers)

df
```

[ ]:
```
pivot_df = df.pivot(index='Year', columns='Month', values='Value')
pivot_df
```

[ ]:
```
colors = ["#006D2C", "#31A354","#74C476"]
pivot_df.loc[:,['Jan','Feb', 'Mar']].plot.bar(stacked=True, color=colors,␣
 ↪figsize=(10,7))

# Years = BINS
# Months = W L T
```

[ ]:
```
#plt.bar(merge_table_bins["Team Win Pctg"], merge_table_bins["W_y"].unique(),␣
 ↪color='r', alpha=0.5, align="center")

#pivot_df = df.pivot(index='Year', columns='Month', values='Value')


# This gives error . . .
```

[ ]:
```
                                                    Add
1773 7 15 8 30 0.233333 0.500000 0.266667 F


8203 10 13 7 30 0.333333 0.433333 0.233333 F


8342 21 8 1 30 0.700000 0.266667 0.033333 A


8475 9 15 6 30 0.300000 0.500000 0.200000 F


8571 10 11 9 30 0.333333 0.366667 0.300000 F



# add to the bins, ng_

    next game   current game
    w  L   T
A   ?  ?   ?
B   ?  ?   ?
```

```
C   ?   ?   ?
D   ?   ?   ?
F   ?   ?   ?
```