

```
In [2]: import sys
import os

sys.path.insert(0, '/usr/hdp/current/spark2-client/python')
sys.path.insert(0, '/usr/hdp/current/spark2-client/python/lib/py4j-0.10.7-src.zip')

os.environ['SPARK_HOME'] = '/usr/hdp/current/spark2-client/'
os.environ['SPARK_CONF_DIR'] = '/etc/spark2/conf'
os.environ['PYSPARK_PYTHON'] = '/opt/anaconda3/bin/python'

import pyspark
conf = pyspark.SparkConf()
conf.setMaster("yarn")
conf.set("spark.driver.memory", "1g")
conf.set("spark.executor.instances", "5")
conf.set("spark.executor.memory", "20g")
conf.set("spark.executor.cores", "5")

sc = pyspark.SparkContext(conf=conf)
```

```
In [3]: sc
```

Out[3]: **SparkContext**

[Spark UI \(http://cnode225.clemson.cloudlab.us:4042\)](http://cnode225.clemson.cloudlab.us:4042)

**Version**

v2.3.1.3.0.1.0-187

**Master**

yarn

**AppName**

pyspark-shell

```
In [4]: !hdfs dfs -put ./data .
!hdfs dfs -ls
!hdfs dfs -ls ./data
```

```
put: `data/game_shifts.csv': File exists
put: `data/game_plays.csv': File exists
put: `data/team_info.csv': File exists
put: `data/game_plays_players.csv': File exists
put: `data/game_teams_stats.csv': File exists
put: `data/game_goalie_stats.csv': File exists
put: `data/game_skater_stats.csv': File exists
put: `data/player_info.csv': File exists
put: `data/game.csv': File exists
Found 4 items
drwxr-xr-x   - sp876427 hadoop          0 2020-01-10 00:16 .sparkStaging
drwxr-xr-x   - sp876427 hadoop          0 2020-01-02 22:57 data
drwxr-xr-x   - sp876427 hadoop          0 2020-01-02 14:05 sp876427
drwxr-xr-x   - sp876427 hadoop          0 2020-01-02 15:48 text
Found 9 items
-rw-r--r--   3 sp876427 hadoop    1911618 2020-01-02 22:57 data/game.csv
-rw-r--r--   3 sp876427 hadoop    2357517 2020-01-02 22:57 data/game_goalie_stats.csv
-rw-r--r--   3 sp876427 hadoop    592063227 2020-01-02 22:57 data/game_plays.csv
-rw-r--r--   3 sp876427 hadoop    286695830 2020-01-02 22:57 data/game_plays_players.csv
-rw-r--r--   3 sp876427 hadoop    316976976 2020-01-02 22:57 data/game_shifts.csv
-rw-r--r--   3 sp876427 hadoop    30389603 2020-01-02 22:57 data/game_skater_stats.csv
-rw-r--r--   3 sp876427 hadoop    1741828 2020-01-02 22:57 data/game_teams_stats.csv
-rw-r--r--   3 sp876427 hadoop     209552 2020-01-02 22:57 data/player_info.csv
-rw-r--r--   3 sp876427 hadoop        1953 2020-01-02 22:57 data/team_info.csv
```

```
In [5]: !!hdfs dfs -cat data/game.csv \n
        2>/dev/null | head -n 50
```

```

Out[5]: [ "game_id","season","type","date_time","date_time_GMT","away_team_i
d","home_team_id","away_goals","home_goals","outcome","home_rink_side_s
tart","venue","venue_link","venue_time_zone_id","venue_time_zone_offse
t","venue_time_zone_tz",
  "2011030221","20112012","P",2012-04-29,"2012-04-29T19:00:00
Z","1","4",3,4,"home win OT","right","Wells Fargo Center","/api/v1/venu
es/null","America/New_York",-4,"EDT",
  "2011030222","20112012","P",2012-05-01,"2012-05-01T23:30:00
Z","1","4",4,1,"away win REG","right","Wells Fargo Center","/api/v1/ven
ues/null","America/New_York",-4,"EDT",
  "2011030223","20112012","P",2012-05-03,"2012-05-03T23:30:00
Z","4","1",3,4,"home win OT","left","Prudential Center","/api/v1/venue
s/null","America/New_York",-4,"EDT",
  "2011030224","20112012","P",2012-05-06,"2012-05-06T23:30:00
Z","4","1",2,4,"home win REG","left","Prudential Center","/api/v1/venue
s/null","America/New_York",-4,"EDT",
  "2011030225","20112012","P",2012-05-08,"2012-05-08T23:30:00
Z","1","4",3,1,"away win REG","right","Wells Fargo Center","/api/v1/ven
ues/null","America/New_York",-4,"EDT",
  "2011030411","20112012","P",2012-05-31,"2012-05-31T00:00:00Z","2
6","1",2,1,"away win OT","left","Prudential Center","/api/v1/venues/nul
l","America/New_York",-4,"EDT",
  "2011030412","20112012","P",2012-06-03,"2012-06-03T00:00:00Z","2
6","1",2,1,"away win OT","left","Prudential Center","/api/v1/venues/nul
l","America/New_York",-4,"EDT",
  "2011030413","20112012","P",2012-06-05,"2012-06-05T00:00:00Z","1","2
6",0,4,"home win REG","right","Staples Center","/api/v1/venues/null","A
merica/Los_Angeles",-7,"PDT",
  "2011030414","20112012","P",2012-06-07,"2012-06-07T00:00:00Z","1","2
6",3,1,"away win REG","right","Staples Center","/api/v1/venues/null","A
merica/Los_Angeles",-7,"PDT",
  "2011030415","20112012","P",2012-06-10,"2012-06-10T00:00:00Z","2
6","1",1,2,"home win REG","left","Prudential Center","/api/v1/venues/nu
ll","America/New_York",-4,"EDT",
  "2011030416","20112012","P",2012-06-12,"2012-06-12T00:00:00Z","1","2
6",1,6,"home win REG","right","Staples Center","/api/v1/venues/null","A
merica/Los_Angeles",-7,"PDT",
  "2010030311","20102011","P",2011-05-15,"2011-05-15T00:00:00Z","1
4","6",5,2,"away win REG","left","TD Garden","/api/v1/venues/null","Ame
rica/New_York",-4,"EDT",
  "2010030312","20102011","P",2011-05-18,"2011-05-18T00:00:00Z","1
4","6",5,6,"home win REG","left","TD Garden","/api/v1/venues/null","Ame
rica/New_York",-4,"EDT",
  "2010030313","20102011","P",2011-05-20,"2011-05-20T00:00:00Z","6","1
4",2,0,"away win REG","left","St. Pete Times Forum","/api/v1/venues/nul
l","America/New_York",-4,"EDT",
  "2010030314","20102011","P",2011-05-21,"2011-05-21T17:30:00Z","6","1
4",3,5,"home win REG","left","St. Pete Times Forum","/api/v1/venues/nul
l","America/New_York",-4,"EDT",
  "2010030315","20102011","P",2011-05-24,"2011-05-24T00:00:00Z","1
4","6",1,3,"home win REG","left","TD Garden","/api/v1/venues/null","Ame
rica/New_York",-4,"EDT",
  "2010030316","20102011","P",2011-05-26,"2011-05-26T00:00:00Z","6","1
4",4,5,"home win REG","left","St. Pete Times Forum","/api/v1/venues/nul
l","America/New_York",-4,"EDT",
  "2010030317","20102011","P",2011-05-28,"2011-05-28T00:00:00Z","1
4","6",0,1,"home win REG","left","TD Garden","/api/v1/venues/null","Ame

```

```

rica/New_York",-4,"EDT"',
  "2012030221","20122013","P",2013-05-16,"2013-05-16T23:30:00
Z","3","6",2,3,"home win OT","left","TD Garden","/api/v1/venues/nul
l","America/New_York",-4,"EDT"',
  "2012030222","20122013","P",2013-05-19,"2013-05-19T19:00:00
Z","3","6",2,5,"home win REG","left","TD Garden","/api/v1/venues/nul
l","America/New_York",-4,"EDT"',
  "2012030223","20122013","P",2013-05-21,"2013-05-21T23:30:00
Z","6","3",2,1,"away win REG","right","Madison Square Garden","/api/v1/
venues/null","America/New_York",-4,"EDT"',
  "2012030224","20122013","P",2013-05-23,"2013-05-23T23:00:00
Z","6","3",3,4,"home win OT","right","Madison Square Garden","/api/v1/v
enues/null","America/New_York",-4,"EDT"',
  "2012030225","20122013","P",2013-05-25,"2013-05-25T21:30:00
Z","3","6",1,3,"home win REG","left","TD Garden","/api/v1/venues/nul
l","America/New_York",-4,"EDT"',
  "2012030311","20122013","P",2013-06-02,"2013-06-02T00:00:00
Z","6","5",3,0,"away win REG","left","CONSOL Energy Center","/api/v1/ve
nues/null","America/New_York",-4,"EDT"',
  "2012030312","20122013","P",2013-06-04,"2013-06-04T00:00:00
Z","6","5",6,1,"away win REG","left","CONSOL Energy Center","/api/v1/ve
nues/null","America/New_York",-4,"EDT"',
  "2012030313","20122013","P",2013-06-06,"2013-06-06T00:00:00
Z","5","6",1,2,"home win OT","left","TD Garden","/api/v1/venues/nul
l","America/New_York",-4,"EDT"',
  "2012030314","20122013","P",2013-06-08,"2013-06-08T00:00:00
Z","5","6",0,1,"home win REG","left","TD Garden","/api/v1/venues/nul
l","America/New_York",-4,"EDT"',
  "2010030241","20102011","P",2011-04-30,"2011-04-30T02:00:00Z","17","2
8",1,2,"home win OT","left","HP Pavilion at San Jose","/api/v1/venues/n
ull","America/Los_Angeles",-7,"PDT"',
  "2010030242","20102011","P",2011-05-01,"2011-05-01T19:00:00Z","17","2
8",1,2,"home win REG","left","HP Pavilion at San Jose","/api/v1/venues/
null","America/Los_Angeles",-7,"PDT"',
  "2010030243","20102011","P",2011-05-05,"2011-05-05T00:00:00Z","28","1
7",4,3,"away win OT","right","Joe Louis Arena","/api/v1/venues/null","A
merica/Detroit",-4,"EDT"',
  "2010030244","20102011","P",2011-05-06,"2011-05-06T23:00:00Z","28","1
7",3,4,"home win REG","right","Joe Louis Arena","/api/v1/venues/nul
l","America/Detroit",-4,"EDT"',
  "2010030245","20102011","P",2011-05-09,"2011-05-09T00:00:00Z","17","2
8",4,3,"away win REG","left","HP Pavilion at San Jose","/api/v1/venues/
null","America/Los_Angeles",-7,"PDT"',
  "2010030246","20102011","P",2011-05-11,"2011-05-11T00:00:00Z","28","1
7",1,3,"home win REG","right","Joe Louis Arena","/api/v1/venues/nul
l","America/Detroit",-4,"EDT"',
  "2010030247","20102011","P",2011-05-13,"2011-05-13T01:00:00Z","17","2
8",2,3,"home win REG","left","HP Pavilion at San Jose","/api/v1/venues/
null","America/Los_Angeles",-7,"PDT"',
  "2010030231","20102011","P",2011-04-29,"2011-04-29T01:00:00Z","18","2
3",0,1,"home win REG","right","Rogers Arena","/api/v1/venues/null","Ame
rica/Vancouver",-7,"PDT"',
  "2010030232","20102011","P",2011-05-01,"2011-05-01T01:00:00Z","18","2
3",2,1,"away win OT","right","Rogers Arena","/api/v1/venues/null","Amer
ica/Vancouver",-7,"PDT"',
  "2010030233","20102011","P",2011-05-04,"2011-05-04T01:00:00Z","23","1
8",3,2,"away win OT","left","Bridgestone Arena","/api/v1/venues/nul

```

```

1,"America/Chicago",-5,"CDT"',
  "2010030234","20102011","P",2011-05-06,"2011-05-06T00:30:00Z","23","1
8",4,2,"away win REG","left","Bridgestone Arena","/api/v1/venues/nul
1","America/Chicago",-5,"CDT"',
  "2010030235","20102011","P",2011-05-08,"2011-05-08T00:00:00Z","18","2
3",4,3,"away win REG","right","Rogers Arena","/api/v1/venues/null","Ame
rica/Vancouver",-7,"PDT"',
  "2010030236","20102011","P",2011-05-10,"2011-05-10T00:00:00Z","23","1
8",2,1,"away win REG","left","Bridgestone Arena","/api/v1/venues/nul
1","America/Chicago",-5,"CDT"',
  "2012030231","20122013","P",2013-05-16,"2013-05-16T00:00:00Z","17","1
6",1,4,"home win REG","right","United Center","/api/v1/venues/null","Am
erica/Chicago",-5,"CDT"',
  "2012030232","20122013","P",2013-05-18,"2013-05-18T17:00:00Z","17","1
6",4,1,"away win REG","right","United Center","/api/v1/venues/null","Am
erica/Chicago",-5,"CDT"',
  "2012030233","20122013","P",2013-05-20,"2013-05-20T23:30:00Z","16","1
7",1,3,"home win REG","right","Joe Louis Arena","/api/v1/venues/nul
1","America/Detroit",-4,"EDT"',
  "2012030234","20122013","P",2013-05-24,"2013-05-24T00:00:00Z","16","1
7",0,2,"home win REG","right","Joe Louis Arena","/api/v1/venues/nul
1","America/Detroit",-4,"EDT"',
  "2012030235","20122013","P",2013-05-26,"2013-05-26T00:00:00Z","17","1
6",1,4,"home win REG","right","United Center","/api/v1/venues/null","Am
erica/Chicago",-5,"CDT"',
  "2012030236","20122013","P",2013-05-28,"2013-05-28T00:00:00Z","16","1
7",4,3,"away win REG","right","Joe Louis Arena","/api/v1/venues/nul
1","America/Detroit",-4,"EDT"',
  "2012030237","20122013","P",2013-05-30,"2013-05-30T00:00:00Z","17","1
6",1,2,"home win OT","right","United Center","/api/v1/venues/null","Ame
rica/Chicago",-5,"CDT"',
  "2012030121","20122013","P",2013-05-02,"2013-05-02T23:00:00
Z","9","8",4,2,"away win REG","right","Centre Bell","/api/v1/venues/nul
1","America/Montreal",-4,"EDT"',
  "2012030122","20122013","P",2013-05-03,"2013-05-03T23:00:00
Z","9","8",1,3,"home win REG","right","Centre Bell","/api/v1/venues/nul
1","America/Montreal",-4,"EDT"']

```

```

In [6]: !!hdfs dfs -cat data/game_goalie_stats.csv \N
        2>/dev/null | head -n 5

```

```

Out[6]: [{"game_id","player_id","team_id","timeOnIce","assists","goals","pi
m","shots","saves","powerPlaySaves","shortHandedSaves","evenSaves","sho
rtHandedShotsAgainst","evenShotsAgainst","powerPlayShotsAgainst","decis
ion","savePercentage","powerPlaySavePercentage","evenStrengthSavePercen
tage",
  "2011030221","8455710","1",3876,0,0,0,36,32,7,0,25,0,28,8,"L",88.8888
888888889,87.5,89.2857142857143',
  "2011030221","8468524","4",3876,0,0,0,26,23,4,5,14,5,16,5,"W",88.4615
384615385,80,87.5',
  "2011030222","8455710","1",3599,0,0,0,20,19,1,3,15,3,16,1,"W",95,100,
93.75',
  "2011030222","8468524","4",3559,0,0,0,34,31,6,0,25,0,28,6,"L",91.1764
705882353,100,89.2857142857143']

```

```
In [7]: !!hdfs dfs -cat data/game_plays.csv \n
        2>/dev/null | head -n 5
```

```
Out[7]: [ "play_id","game_id","play_num","team_id_for","team_id_against","event",
"secondaryType","x","y","period","periodType","periodTime","periodTimeRemaining",
"dateTime","goals_away","goals_home","description","st_x","st_y","rink_side",
"2011030221_1","2011030221",1,NA,NA,"Game Scheduled",NA,NA,NA,"1","REGULAR",0,1200,
"2012-04-29 18:16:06",0,0,"Game Scheduled",NA,NA,NA',
"2011030221_2","2011030221",2,NA,NA,"Period Ready",NA,NA,NA,"1","REGULAR",0,1200,
"2012-04-29 19:19:49",0,0,"Period Ready",NA,NA,NA',
"2011030221_3","2011030221",3,NA,NA,"Period Start",NA,NA,NA,"1","REGULAR",0,1200,
"2012-04-29 19:20:13",0,0,"Period Start",NA,NA,NA',
"2011030221_4","2011030221",4,"4","1","Faceoff",NA,0,0,"1","REGULAR",0,1200,
"2012-04-29 19:20:13",0,0,"Sean Couturier faceoff won against Patrik Elias",0,0,"right"]
```

```
In [8]: !!hdfs dfs -cat data/game_plays_players.csv \n
        2>/dev/null | head -n 5
```

```
Out[8]: [ "play_id","game_id","play_num","player_id","playerType",
"2011030221_4","2011030221",4,"8476461","Winner",
"2011030221_4","2011030221",4,"8460542","Loser",
"2011030221_5","2011030221",5,"8459670","PlayerID",
"2011030221_6","2011030221",6,"8470171","Blocker"]
```

```
In [9]: !!hdfs dfs -cat data/game_shifts.csv \n
        2>/dev/null | head -n 5
```

```
Out[9]: [ "game_id","player_id","period","shift_start","shift_end",
"2016030161","8477964",3,2729,2730',
"2016030161","8475768",3,2729,2730',
"2016030161","8474125",2,1772,1773',
"2016030161","8470595",4,4212,4214']
```

```
In [117]: !!hdfs dfs -cat data/game_skater_stats.csv \n
          2>/dev/null | head -n 5
```

```
Out[117]: [ "game_id","player_id","team_id","timeOnIce","assists","goals","shots",
"hits","powerPlayGoals","powerPlayAssists","penaltyMinutes","faceOffWins",
"faceoffTaken","takeaways","giveaways","shortHandedGoals","shortHandedAssists",
"blocked","plusMinus","evenTimeOnIce","shortHandedTimeOnIce","powerPlayTimeOnIce",
"2011030221","8467412","1",999,0,0,1,3,0,0,0,0,0,0,0,0,0,0,-1,885,98,16',
"2011030221","8468501","1",1168,0,0,0,4,0,0,0,0,0,0,1,1,0,0,1,0,767,401,0',
"2011030221","8470609","1",558,0,0,2,1,0,0,0,0,2,0,0,0,0,0,1,-1,542,0,16',
"2011030221","8471816","1",1134,0,0,1,4,0,0,0,0,0,0,1,0,0,0,1,-1,935,183,16']
```

```
In [11]: !!hdfs dfs -cat data/game_teams_stats.csv \n
          2>/dev/null | head -n 5
```

```
Out[11]: [ "game_id","team_id","HoA","won","settled_in","head_coach","goals","sh
ots","hits","pim","powerPlayOpportunities","powerPlayGoals","faceOffWin
Percentage","giveaways","takeaways",
  "2011030221","1","away",FALSE,"OT","Peter DeBoer",3,26,31,12,3,1,44.
9,6,7',
  "2011030221","4","home",TRUE,"OT","Peter Laviolette",4,36,27,6,6,1,5
5.1,13,4',
  "2011030222","1","away",TRUE,"REG","Peter DeBoer",4,35,32,12,4,0,50.
9,8,7',
  "2011030222","4","home",FALSE,"REG","Peter Laviolette",1,20,24,32,5,
0,49.1,9,6']
```

```
In [12]: !!hdfs dfs -cat data/player_info.csv \n
          2>/dev/null | head -n 5
```

```
Out[12]: [ "player_id","firstName","lastName","nationality","birthCity","primary
Position","birthDate","link",
  "8467412","Alexei","Ponikarovsky","UKR","Kiev","LW",1980-04-09,"/api/
v1/people/8467412",
  "8468501","Anton","Volchenkov","RUS","Moscow","D",1982-02-25,"/api/v
1/people/8468501",
  "8459670","Kimmo","Timonen","FIN","Kuopio","D",1975-03-18,"/api/v1/pe
ople/8459670",
  "8471233","Travis","Zajac","CAN","Winnipeg","C",1985-05-13,"/api/v1/p
eople/8471233"]
```



```
In [13]: !!hdfs dfs -cat data/team_info.csv \
2>/dev/null | head -n 35
```

```
Out[13]: [{"team_id","franchiseId","shortName","teamName","abbreviation","lin
k",
"1","23","New Jersey","Devils","NJD","/api/v1/teams/1",
"4","16","Philadelphia","Flyers","PHI","/api/v1/teams/4",
"26","14","Los Angeles","Kings","LAK","/api/v1/teams/26",
"14","31","Tampa Bay","Lightning","TBL","/api/v1/teams/14",
"6","6","Boston","Bruins","BOS","/api/v1/teams/6",
"3","10","NY Rangers","Rangers","NYR","/api/v1/teams/3",
"5","17","Pittsburgh","Penguins","PIT","/api/v1/teams/5",
"17","12","Detroit","Red Wings","DET","/api/v1/teams/17",
"28","29","San Jose","Sharks","SJS","/api/v1/teams/28",
"18","34","Nashville","Predators","NSH","/api/v1/teams/18",
"23","20","Vancouver","Canucks","VAN","/api/v1/teams/23",
"16","11","Chicago","Blackhawks","CHI","/api/v1/teams/16",
"9","30","Ottawa","Senators","OTT","/api/v1/teams/9",
"8","1","Montreal","Canadiens","MTL","/api/v1/teams/8",
"30","37","Minnesota","Wild","MIN","/api/v1/teams/30",
"15","24","Washington","Capitals","WSH","/api/v1/teams/15",
"19","18","St Louis","Blues","STL","/api/v1/teams/19",
"24","32","Anaheim","Ducks","ANA","/api/v1/teams/24",
"27","28","Phoenix","Coyotes","PHX","/api/v1/teams/27",
"2","22","NY Islanders","Islanders","NYI","/api/v1/teams/2",
"10","5","Toronto","Maple Leafs","TOR","/api/v1/teams/10",
"13","33","Florida","Panthers","FLA","/api/v1/teams/13",
"7","19","Buffalo","Sabres","BUF","/api/v1/teams/7",
"20","21","Calgary","Flames","CGY","/api/v1/teams/20",
"21","27","Colorado","Avalanche","COL","/api/v1/teams/21",
"25","15","Dallas","Stars","DAL","/api/v1/teams/25",
"29","36","Columbus","Blue Jackets","CBJ","/api/v1/teams/29",
"52","35","Winnipeg","Jets","WPG","/api/v1/teams/52",
"22","25","Edmonton","Oilers","EDM","/api/v1/teams/22",
"54","38","Vegas","Golden Knights","VGK","/api/v1/teams/54",
"12","26","Carolina","Hurricanes","CAR","/api/v1/teams/12",
"53","28","Arizona","Coyotes","ARI","/api/v1/teams/53",
"11","35","Atlanta","Thrashers","ATL","/api/v1/teams/11"]
```

```
In [14]: game = sc.textFile("data/game.csv")
game.cache()
```

```
Out[14]: data/game.csv MapPartitionsRDD[1] at textFile at NativeMethodAccessorIm
pl.java:0
```

```
In [15]: game_goalie_stats = sc.textFile("data/game_goalie_stats.csv")
game_goalie_stats.cache()
```

```
Out[15]: data/game_goalie_stats.csv MapPartitionsRDD[3] at textFile at NativeMet
hodAccessorImpl.java:0
```

```
In [16]: game_plays = sc.textFile("data/game_plays.csv")
game_plays.cache()
```

```
Out[16]: data/game_plays.csv MapPartitionsRDD[5] at textFile at NativeMethodAccesso
rImpl.java:0
```

```
In [17]: game_plays_players = sc.textFile("data/game_plays_players.csv")
game_plays_players.cache()
```

```
Out[17]: data/game_plays_players.csv MapPartitionsRDD[7] at textFile at NativeMe
thodAccessorImpl.java:0
```

```
In [18]: game_shifts = sc.textFile("data/game_shifts.csv")
game_shifts.cache()
```

```
Out[18]: data/game_shifts.csv MapPartitionsRDD[9] at textFile at NativeMethodAcc
essorImpl.java:0
```

```
In [19]: game_skater_shifts = sc.textFile("data/game_skater_shifts.csv")
game_skater_shifts.cache()
```

```
Out[19]: data/game_skater_shifts.csv MapPartitionsRDD[11] at textFile at NativeM
ethodAccessorImpl.java:0
```

```
In [20]: game_teams_shifts = sc.textFile("data/game_teams_shifts.csv")
game_teams_shifts.cache()
```

```
Out[20]: data/game_teams_shifts.csv MapPartitionsRDD[13] at textFile at NativeMe
thodAccessorImpl.java:0
```

```
In [39]: game_skater_stats = sc.textFile("data/game_skater_stats.csv")
game_skater_stats.cache()
```

```
Out[39]: data/game_skater_stats.csv MapPartitionsRDD[52] at textFile at NativeMe
thodAccessorImpl.java:0
```

```
In [21]: game_teams_stats = sc.textFile("data/game_teams_stats.csv")
game_teams_stats.cache()
```

```
Out[21]: data/game_teams_stats.csv MapPartitionsRDD[15] at textFile at NativeMet
hodAccessorImpl.java:0
```

```
In [22]: player_info = sc.textFile("data/player_info.csv")
player_info.cache()
```

```
Out[22]: data/player_info.csv MapPartitionsRDD[17] at textFile at NativeMethodAc
cessorImpl.java:0
```

```
In [23]: team_info = sc.textFile("data/team_info.csv")
team_info.cache()
```

```
Out[23]: data/team_info.csv MapPartitionsRDD[19] at textFile at NativeMethodAcce
ssorImpl.java:0
```

```
In [24]: %%time
game.count()
```

```
CPU times: user 14.8 ms, sys: 1.72 ms, total: 16.5 ms
Wall time: 2.57 s
```

```
Out[24]: 11435
```

```
In [25]: game_teams_statsHeader = game_teams_stats.first()
print(game_teams_statsHeader)
```

```
"game_id", "team_id", "HoA", "won", "settled_in", "head_coach", "goals", "shots", "hits", "pim", "powerPlayOpportunities", "powerPlayGoals", "faceOffWinPercentage", "giveaways", "takeaways"
```

## 1) Team with the highest Faceoff Percentage Throughout all the games

```
In [26]: noHeaderTeamStats = game_teams_stats.filter(lambda x: x != game_teams_statsHeader)
noHeaderTeamStats.take(5)
```

```
Out[26]: [ '"2011030221", "1", "away", FALSE, "OT", "Peter DeBoer", 3, 26, 31, 12, 3, 1, 44.9, 6, 7',
  '"2011030221", "4", "home", TRUE, "OT", "Peter Laviolette", 4, 36, 27, 6, 6, 1, 55.1, 13, 4',
  '"2011030222", "1", "away", TRUE, "REG", "Peter DeBoer", 4, 35, 32, 12, 4, 0, 50.9, 8, 7',
  '"2011030222", "4", "home", FALSE, "REG", "Peter Laviolette", 1, 20, 24, 32, 5, 0, 49.1, 9, 6',
  '"2011030223", "4", "away", FALSE, "OT", "Peter Laviolette", 3, 28, 28, 4, 5, 1, 50.8, 2, 1']
```

```
In [27]: faceOff_percentage = noHeaderTeamStats.map(lambda line: (line.split(",")[1], float(line.split(",")[12])))
faceOff_percentage.take(5)
```

```
Out[27]: [('1', 44.9), ('4', 55.1), ('1', 50.9), ('4', 49.1), ('4', 50.8)]
```

```
In [28]: countByKeyFO = faceOff_percentage.countByKey()  
countByKeyFO
```

```
Out[28]: defaultdict(int,  
    {'1': 733,  
     '4': 745,  
     '26': 783,  
     '14': 790,  
     '6': 812,  
     '3': 802,  
     '5': 815,  
     '17': 751,  
     '28': 805,  
     '18': 787,  
     '23': 744,  
     '16': 793,  
     '9': 746,  
     '8': 751,  
     '30': 748,  
     '15': 804,  
     '19': 788,  
     '24': 774,  
     '27': 314,  
     '2': 736,  
     '10': 731,  
     '13': 717,  
     '7': 711,  
     '20': 724,  
     '21': 729,  
     '25': 736,  
     '29': 731,  
     '52': 649,  
     '22': 717,  
     '54': 191,  
     '12': 719,  
     '53': 410,  
     '11': 82})
```

```
In [29]: def sumValues(x,y):  
         return (x + y)  
  
sumRatings = faceOff_percentage.reduceByKey(sumValues)  
sumRatings.take(35)
```

```
Out[29]: [ ("4", 38103.799999999999),  
          ("5", 40795.199999999998),  
          ("28", 41508.900000000002),  
          ("23", 37134.899999999994),  
          ("9", 37494.9),  
          ("30", 38051.000000000003),  
          ("19", 39991.8),  
          ("24", 39079.800000000001),  
          ("27", 16105.299999999996),  
          ("2", 35825.6000000000035),  
          ("13", 35681.5),  
          ("20", 34989.599999999999),  
          ("25", 37147.500000000001),  
          ("54", 9459.4),  
          ("11", 4010.4999999999999),  
          ("1", 35250.900000000001),  
          ("26", 39973.099999999999),  
          ("14", 39028.999999999985),  
          ("6", 42553.899999999994),  
          ("3", 38818.900000000001),  
          ("17", 38206.900000000001),  
          ("18", 39937.700000000003),  
          ("16", 39749.599999999997),  
          ("8", 37269.100000000006),  
          ("15", 39959.699999999999),  
          ("10", 36899.999999999985),  
          ("7", 34322.700000000001),  
          ("21", 36226.899999999994),  
          ("29", 36572.5),  
          ("52", 31815.9),  
          ("22", 34110.300000000002),  
          ("12", 36792.700000000001),  
          ("53", 20530.499999999996)]
```

```
In [30]: import operator  
  
sumRatings = faceOff_percentage.reduceByKey(operator.add)  
sumRatings.take(5)
```

```
Out[30]: [ ("2", 35825.6000000000035),  
          ("23", 37134.899999999994),  
          ("28", 41508.900000000002),  
          ("19", 39991.8),  
          ("9", 37494.9)]
```

```
In [31]: avgRatings02 = sumRatings.map(lambda x: (x[0], x[1] / countByKeyFO.get(x
[0])))
avgRatings02.take(5)
```

```
Out[31]: [('4"', 51.14604026845636),
('5"', 50.055460122699365),
('28"', 51.563850931677045),
('23"', 49.912499999999994),
('9"', 50.2612600536193)]
```

```
In [32]: team_info_header = team_info.first()
print(team_info_header)

"team_id","franchiseId","shortName","teamName","abbreviation","link"
```

```
In [33]: team_info = team_info.filter(lambda x: x != team_info_header)
team_info.take(5)
```

```
Out[33]: ['1","23","New Jersey","Devils","NJD","/api/v1/teams/1"',
'4","16","Philadelphia","Flyers","PHI","/api/v1/teams/4"',
'26","14","Los Angeles","Kings","LAK","/api/v1/teams/26"',
'14","31","Tampa Bay","Lightning","TBL","/api/v1/teams/14"',
'6","6","Boston","Bruins","BOS","/api/v1/teams/6"']
```

```
In [34]: teams = team_info.map(lambda line: (line.split(",")[0], ((line.rsplit(
",",1)[0]).split(",",1)[1], line.rsplit(",",1)[1])))
teams.take(5)
```

```
Out[34]: [('1"', ('23","New Jersey","Devils","NJD"', '/api/v1/teams/1')),
('4"', ('16","Philadelphia","Flyers","PHI"', '/api/v1/teams/4')),
('26"', ('14","Los Angeles","Kings","LAK"', '/api/v1/teams/26')),
('14"', ('31","Tampa Bay","Lightning","TBL"', '/api/v1/teams/1
4')),
('6"', ('6","Boston","Bruins","BOS"', '/api/v1/teams/6'))]
```

```
In [35]: augmentedFOP = avgRatings02.join(teams)
augmentedFOP.take(5)
```

```
Out[35]: [('2"',
(48.676086956521786,
('22","NY Islanders","Islanders","NYI"', '/api/v1/teams/2'))),
('30"',
(50.87032085561501,
('37","Minnesota","Wild","MIN"', '/api/v1/teams/30'))),
('20"',
(48.3281767955801, ('21","Calgary","Flames","CGY"', '/api/v1/teams/
20'))),
('11"',
(48.908536585365844,
('35","Atlanta","Thrashers","ATL"', '/api/v1/teams/11'))),
('1"',
(48.09126875852662,
('23","New Jersey","Devils","NJD"', '/api/v1/teams/1')))]
```

```
In [36]: augmentedFOP.takeOrdered(10, key = lambda x : -x[1][0])
```

```
Out[36]: [( '6" ',
(52.406280788177334, ( '6", "Boston", "Bruins", "BOS" ', '/api/v1/teams/
6" '))),
( '28" ',
(51.563850931677045,
( '29", "San Jose", "Sharks", "SJS" ', '/api/v1/teams/28" '))),
( '27" ',
(51.29076433121018,
( '28", "Phoenix", "Coyotes", "PHX" ', '/api/v1/teams/27" '))),
( '12" ',
(51.17204450625871,
( '26", "Carolina", "Hurricanes", "CAR" ', '/api/v1/teams/12" '))),
( '4" ',
(51.14604026845636,
( '16", "Philadelphia", "Flyers", "PHI" ', '/api/v1/teams/4" '))),
( '26" ',
(51.05121328224775,
( '14", "Los Angeles", "Kings", "LAK" ', '/api/v1/teams/26" '))),
( '17" ',
(50.874700399467386,
( '12", "Detroit", "Red Wings", "DET" ', '/api/v1/teams/17" '))),
( '30" ',
(50.87032085561501,
( '37", "Minnesota", "Wild", "MIN" ', '/api/v1/teams/30" '))),
( '19" ',
(50.7510152284264, ( '18", "St Louis", "Blues", "STL" ', '/api/v1/teams/
19" '))),
( '18" ',
(50.74675984752228,
( '34", "Nashville", "Predators", "NSH" ', '/api/v1/teams/18" ')))]
```

```
In [79]: augmentedFOP.takeOrdered(10, key = lambda x : x[1][0])
```

```
Out[79]: [( '22'',
(47.57364016736404,
( '25',"Edmonton","Oilers","EDM'', '/api/v1/teams/22'')),
'1'',
(48.09126875852662,
( '23',"New Jersey","Devils","NJD'', '/api/v1/teams/1'')),
'7'',
(48.27383966244727, ( '19',"Buffalo","Sabres","BUF'', '/api/v1/teams/7'')),
'20'',
(48.3281767955801, ( '21',"Calgary","Flames","CGY'', '/api/v1/teams/20'')),
'3'',
(48.40261845386535,
( '10',"NY Rangers","Rangers","NYR'', '/api/v1/teams/3'')),
'2'',
(48.676086956521786,
( '22',"NY Islanders","Islanders","NYI'', '/api/v1/teams/2'')),
'11'',
(48.908536585365844,
( '35',"Atlanta","Thrashers","ATL'', '/api/v1/teams/11'')),
'52'',
(49.02295839753467, ( '35',"Winnipeg","Jets","WPG'', '/api/v1/teams/52'')),
'14'',
(49.40379746835441,
( '31',"Tampa Bay","Lightning","TBL'', '/api/v1/teams/14'')),
'54'',
(49.525654450261776,
( '38',"Vegas","Golden Knights","VGK'', '/api/v1/teams/54'')))]
```

## Most Goals

```
In [44]: skater_stats = game_skater_stats.first()
print(skater_stats)
```

```
"game_id","player_id","team_id","timeOnIce","assists","goals","shots",
"hits","powerPlayGoals","powerPlayAssists","penaltyMinutes","faceOffWins",
"faceoffTaken","takeaways","giveaways","shortHandedGoals","shortHandedAssists",
"blocked","plusMinus","evenTimeOnIce","shortHandedTimeOnIce","powerPlayTimeOnIce"
```



```
In [45]: NH_game_skater_stats = game_skater_stats.filter(lambda x: x != skater_stats)
NH_game_skater_stats.take(5)
```

```
Out[45]: [ '"2011030221","8467412","1",999,0,0,1,3,0,0,0,0,0,0,0,0,0,-1,885,98,16',
          '"2011030221","8468501","1",1168,0,0,0,4,0,0,0,0,0,0,1,1,0,0,1,0,767,401,0',
          '"2011030221","8470609","1",558,0,0,2,1,0,0,0,0,2,0,0,0,0,1,-1,542,0,16',
          '"2011030221","8471816","1",1134,0,0,1,4,0,0,0,0,0,1,0,0,0,1,-1,935,183,16',
          '"2011030221","8472410","1",436,0,0,1,3,0,0,0,3,6,0,0,0,0,0,-1,436,0,0']
```

```
In [61]: player_goals = NH_game_skater_stats.map(lambda line: (line.split(",")[1], int(line.split(",")[5])))
player_goals.take(5)
```

```
Out[61]: [ ('"8467412"', 0),
          ('"8468501"', 0),
          ('"8470609"', 0),
          ('"8471816"', 0),
          ('"8472410"', 0)]
```

```
In [62]: countByKeyGoals = player_goals.countByKey()  
countByKeyGoals
```

```
Out[62]: defaultdict(int,
                        {"8467412": 213,
                         "8468501": 293,
                         "8470609": 350,
                         "8471816": 413,
                         "8472410": 325,
                         "8471233": 625,
                         "8470920": 378,
                         "8473646": 379,
                         "8470610": 599,
                         "8472382": 690,
                         "8474641": 622,
                         "8460626": 200,
                         "8472394": 146,
                         "8464977": 427,
                         "8469454": 282,
                         "8469623": 410,
                         "8460465": 100,
                         "8460542": 384,
                         "8468486": 607,
                         "8459670": 343,
                         "8470640": 488,
                         "8473512": 734,
                         "8470601": 718,
                         "8475640": 100,
                         "8470171": 423,
                         "8474892": 42,
                         "8476177": 479,
                         "8448208": 499,
                         "8474190": 719,
                         "8464975": 345,
                         "8468534": 115,
                         "8474037": 635,
                         "8471269": 383,
                         "8474161": 715,
                         "8476461": 602,
                         "8475170": 636,
                         "8476457": 516,
                         "8471636": 92,
                         "8471236": 282,
                         "8474736": 365,
                         "8470604": 678,
                         "8467452": 477,
                         "8470606": 765,
                         "8470662": 236,
                         "8470121": 362,
                         "8474594": 254,
                         "8471208": 290,
                         "8470617": 461,
                         "8465185": 365,
                         "8471685": 758,
                         "8470755": 536,
                         "8475325": 405,
                         "8474563": 767,
                         "8473453": 686,
                         "8474166": 616,
                         "8474100": 440,
```

"8468508": 802,  
"8468526": 506,  
"8475185": 321,  
"8467346": 177,  
"8466182": 213,  
"8470905": 105,  
"8466142": 349,  
"8471314": 115,  
"8475167": 706,  
"8467925": 281,  
"8469765": 105,  
"8467329": 373,  
"8467988": 227,  
"8471703": 316,  
"8466378": 424,  
"8465122": 190,  
"8468575": 601,  
"8473962": 489,  
"8470176": 286,  
"8470775": 641,  
"8474564": 655,  
"8460503": 90,  
"8473419": 769,  
"8469619": 535,  
"8471717": 552,  
"8471276": 732,  
"8470187": 693,  
"8471514": 301,  
"8470169": 499,  
"8470230": 374,  
"8467967": 495,  
"8465009": 744,  
"8467545": 366,  
"8465200": 189,  
"8466333": 383,  
"8450725": 106,  
"8470596": 247,  
"8475794": 734,  
"8473473": 775,  
"8465978": 476,  
"8470638": 757,  
"8471767": 101,  
"8474095": 142,  
"8471958": 763,  
"8471339": 627,  
"8471873": 692,  
"8473432": 452,  
"8470192": 157,  
"8474151": 729,  
"8475184": 537,  
"8475186": 556,  
"8474176": 674,  
"8467389": 509,  
"8465025": 161,  
"8470619": 773,  
"8467881": 285,  
"8474613": 766,

"8473544": 758,  
"8474584": 560,  
"8475692": 584,  
"8470041": 628,  
"8474749": 274,  
"8476792": 524,  
"8476462": 543,  
"8474230": 259,  
"8458938": 186,  
"8470232": 31,  
"8465951": 342,  
"8469638": 591,  
"8468542": 550,  
"8462042": 557,  
"8471311": 366,  
"8471675": 674,  
"8466160": 364,  
"8470543": 743,  
"8471215": 643,  
"8474091": 594,  
"8471702": 758,  
"8471724": 610,  
"8468114": 270,  
"8465166": 402,  
"8468498": 680,  
"8471707": 715,  
"8466393": 312,  
"8467580": 127,  
"8471856": 257,  
"8468674": 668,  
"8475761": 221,  
"8473484": 97,  
"8471716": 679,  
"8466149": 290,  
"8467331": 362,  
"8470201": 485,  
"8470047": 749,  
"8471794": 560,  
"8470622": 366,  
"8468083": 616,  
"8470318": 615,  
"8471309": 315,  
"8470778": 470,  
"8457063": 168,  
"8459444": 241,  
"8460743": 163,  
"8468509": 691,  
"8464962": 86,  
"8467856": 74,  
"8467514": 394,  
"8468482": 309,  
"8467096": 481,  
"8471338": 575,  
"8474053": 739,  
"8470273": 202,  
"8469581": 169,  
"8471682": 344,

"8469622": 271,  
"8466139": 789,  
"8466138": 739,  
"8468577": 92,  
"8471709": 753,  
"8470794": 782,  
"8459687": 190,  
"8473600": 82,  
"8474218": 594,  
"8470280": 162,  
"8459514": 193,  
"8446485": 55,  
"8473465": 544,  
"8449645": 42,  
"8473608": 67,  
"8471859": 220,  
"8470839": 250,  
"8467330": 462,  
"8470630": 532,  
"8468064": 353,  
"8467370": 555,  
"8468162": 230,  
"8470600": 741,  
"8471887": 694,  
"8471742": 513,  
"8470642": 649,  
"8474164": 122,  
"8474096": 460,  
"8460719": 180,  
"8474094": 277,  
"8469547": 386,  
"8468208": 642,  
"8467876": 642,  
"8470616": 677,  
"8467875": 638,  
"8467463": 182,  
"8468485": 241,  
"8470310": 167,  
"8471498": 565,  
"8471664": 384,  
"8471268": 36,  
"8470358": 595,  
"8470854": 479,  
"8471303": 601,  
"8470654": 404,  
"8470054": 224,  
"8470274": 402,  
"8469465": 645,  
"8469598": 582,  
"8469555": 391,  
"8469626": 133,  
"8474570": 340,  
"8465202": 241,  
"8474569": 647,  
"8464957": 73,  
"8470620": 68,  
"8470724": 459,

"8473558": 157,  
"8474090": 455,  
"8474679": 555,  
"8477059": 135,  
"8469470": 245,  
"8474168": 232,  
"8471693": 348,  
"8466148": 562,  
"8465058": 399,  
"8470607": 774,  
"8471254": 443,  
"8470666": 227,  
"8475323": 607,  
"8475181": 740,  
"8469544": 611,  
"8469665": 638,  
"8471769": 731,  
"8471245": 301,  
"8473564": 679,  
"8473604": 740,  
"8476438": 602,  
"8470281": 760,  
"8474141": 748,  
"8462129": 297,  
"8476381": 576,  
"8473537": 508,  
"8467493": 468,  
"8470760": 503,  
"8474250": 635,  
"8475164": 561,  
"8464956": 333,  
"8476195": 199,  
"8473468": 302,  
"8471862": 303,  
"8473588": 398,  
"8476419": 403,  
"8458951": 333,  
"8474578": 681,  
"8476459": 519,  
"8470599": 387,  
"8471670": 57,  
"8475174": 265,  
"8474068": 663,  
"8460621": 261,  
"8467496": 414,  
"8475797": 58,  
"8471976": 569,  
"8470324": 477,  
"8476851": 518,  
"8474157": 651,  
"8471283": 424,  
"8474056": 725,  
"8469521": 662,  
"8467407": 519,  
"8468635": 326,  
"8471504": 433,  
"8473466": 316,

"8475848": 528,  
"8476244": 214,  
"8474189": 759,  
"8465914": 234,  
"8471719": 81,  
"8470378": 193,  
"8467831": 265,  
"8475254": 90,  
"8469474": 120,  
"8474605": 275,  
"8471663": 85,  
"8470803": 717,  
"8467336": 225,  
"8474716": 635,  
"8474618": 557,  
"8470886": 383,  
"8473504": 663,  
"8475722": 442,  
"8475745": 568,  
"8470104": 414,  
"8470153": 172,  
"8476463": 530,  
"8464989": 705,  
"8469820": 241,  
"8469459": 655,  
"8475650": 276,  
"8473992": 280,  
"8469542": 74,  
"8469506": 715,  
"8465210": 161,  
"8465059": 115,  
"8469639": 420,  
"8471214": 788,  
"8459429": 161,  
"8473563": 744,  
"8474590": 766,  
"8471242": 561,  
"8471240": 188,  
"8464979": 99,  
"8468611": 580,  
"8466158": 330,  
"8467365": 188,  
"8466251": 665,  
"8470611": 523,  
"8475149": 682,  
"8470159": 428,  
"8473991": 694,  
"8470120": 346,  
"8458590": 196,  
"8473913": 7,  
"8473932": 220,  
"8458637": 145,  
"8474551": 229,  
"8467361": 86,  
"8459424": 66,  
"8470597": 64,  
"8471346": 527,



```
"8462535": 91,  
"8470996": 20,  
"8460661": 6,  
"8467344": 368,  
"8474184": 196,  
"8474162": 583,  
"8475160": 662,  
"8471817": 631,  
"8467890": 460,  
"8473534": 626,  
"8474565": 754,  
"8471698": 680,  
"8474031": 669,  
"8473485": 555,  
"8470151": 706,  
"8468434": 135,  
"8475768": 524,  
"8470871": 249,  
"8470655": 724,  
"8470257": 664,  
"8471743": 430,  
"8471490": 220,  
"8471392": 660,  
"8474102": 633,  
"8467917": 285,  
"8475765": 567,  
"8475726": 504,  
"8470613": 753,  
"8471956": 461,  
"8474739": 506,  
"8471476": 652,  
"8474027": 691,  
"8474001": 254,  
"8467899": 255,  
"8467478": 100,  
"8459457": 153,  
"8471729": 633,  
"8470647": 286,  
"8471851": 227,  
"8465005": 126,  
"8468515": 147,  
"8473908": 166,  
"8471677": 690,  
"8460496": 289,  
"8474114": 14,  
"8471386": 16,  
"8468700": 422,  
"8470603": 167,  
"8474600": 634,  
"8475268": 394,  
"8474519": 202,  
"8471228": 264,  
"8471450": 152,  
"8475176": 542,  
"8475225": 640,  
"8477215": 452,  
"8460561": 114,
```

"8467332": 294,  
"8473933": 606,  
"8471676": 677,  
"8474009": 644,  
"8471699": 775,  
"8470621": 690,  
"8469483": 198,  
"8467400": 621,  
"8459442": 287,  
"8465042": 196,  
"8457981": 290,  
"8475764": 682,  
"8470612": 690,  
"8473492": 452,  
"8475151": 534,  
"8475770": 196,  
"8474579": 488,  
"8474873": 135,  
"8470329": 127,  
"8474727": 264,  
"8466285": 545,  
"8468518": 329,  
"8471426": 745,  
"8460527": 40,  
"8471352": 136,  
"8465170": 63,  
"8473722": 219,  
"8459628": 202,  
"8459053": 190,  
"8471735": 750,  
"8468484": 208,  
"8471232": 485,  
"8468085": 561,  
"8471680": 89,  
"8474571": 631,  
"8462036": 93,  
"8462038": 513,  
"8468535": 643,  
"8458537": 283,  
"8471231": 374,  
"8471691": 467,  
"8464966": 254,  
"8475171": 673,  
"8470615": 24,  
"8473673": 421,  
"8474646": 179,  
"8467393": 76,  
"8467844": 127,  
"8468598": 277,  
"8459587": 85,  
"8468095": 191,  
"8469992": 206,  
"8467304": 92,  
"8475105": 48,  
"8459596": 166,  
"8471279": 99,  
"8470713": 150,

"8475153": 125,  
"8459461": 108,  
"8474625": 336,  
"8474709": 670,  
"8470144": 694,  
"8470644": 157,  
"8475166": 700,  
"8474573": 686,  
"8473584": 560,  
"8473565": 203,  
"8473449": 618,  
"8474612": 609,  
"8470852": 555,  
"8473546": 596,  
"8474066": 473,  
"8475231": 518,  
"8464967": 175,  
"8468309": 447,  
"8469779": 91,  
"8468504": 425,  
"8467931": 162,  
"8474666": 52,  
"8475155": 222,  
"8468101": 99,  
"8473514": 112,  
"8475754": 506,  
"8473618": 596,  
"8467371": 518,  
"8476162": 80,  
"8474291": 599,  
"8471686": 683,  
"8476880": 523,  
"8470719": 46,  
"8475690": 461,  
"8467334": 297,  
"8465192": 167,  
"8470700": 11,  
"8471266": 314,  
"8475178": 487,  
"8474681": 156,  
"8473682": 118,  
"8474668": 530,  
"8473914": 582,  
"8472368": 108,  
"8473533": 594,  
"8469760": 388,  
"8458529": 95,  
"8474025": 152,  
"8471668": 37,  
"8459492": 164,  
"8471714": 100,  
"8459462": 72,  
"8470299": 117,  
"8471234": 138,  
"8468483": 512,  
"8468172": 211,  
"8475118": 164,

"8473573": 662,  
"8471273": 625,  
"8460562": 29,  
"8468224": 95,  
"8469455": 635,  
"8475913": 418,  
"8462176": 78,  
"8471768": 427,  
"8468463": 127,  
"8467351": 311,  
"8470222": 310,  
"8471678": 569,  
"8469500": 484,  
"8467439": 112,  
"8474134": 483,  
"8466144": 54,  
"8468778": 157,  
"8471748": 155,  
"8470602": 674,  
"8469508": 520,  
"8470207": 313,  
"8474125": 600,  
"8470667": 374,  
"8473548": 776,  
"8473712": 80,  
"8473579": 543,  
"8474497": 89,  
"8471362": 380,  
"8473463": 432,  
"8475098": 673,  
"8475172": 579,  
"8468639": 399,  
"8462033": 34,  
"8472424": 32,  
"8474192": 141,  
"8470672": 189,  
"8474581": 577,  
"8474577": 308,  
"8473422": 685,  
"8474498": 160,  
"8474024": 84,  
"8462225": 188,  
"8474772": 94,  
"8470750": 430,  
"8475637": 131,  
"8460492": 170,  
"8468478": 318,  
"8466371": 114,  
"8470039": 414,  
"8470105": 515,  
"8475179": 578,  
"8474520": 563,  
"8473574": 489,  
"8475790": 484,  
"8469473": 318,  
"8469457": 260,  
"8470729": 150,

"8469770": 462,  
"8473536": 201,  
"8471996": 42,  
"8475625": 370,  
"8471842": 86,  
"8476871": 422,  
"8470063": 215,  
"8474631": 46,  
"8460541": 56,  
"8471804": 383,  
"8459640": 80,  
"8471388": 201,  
"8471752": 79,  
"8471761": 308,  
"8459427": 78,  
"8473426": 337,  
"8470598": 680,  
"8474589": 546,  
"8471226": 605,  
"8474574": 586,  
"8469591": 235,  
"8469472": 63,  
"8468927": 131,  
"8467878": 144,  
"8473568": 137,  
"8471284": 563,  
"8470018": 170,  
"8471419": 27,  
"8469485": 313,  
"8462077": 137,  
"8466215": 227,  
"8458525": 142,  
"8468636": 189,  
"8466309": 65,  
"8475150": 166,  
"8471262": 514,  
"8474628": 445,  
"8474013": 550,  
"8476302": 80,  
"8473542": 151,  
"8473911": 37,  
"8469759": 528,  
"8470740": 558,  
"8465050": 190,  
"8470966": 647,  
"8475907": 341,  
"8477935": 332,  
"8474642": 362,  
"8474673": 590,  
"8474150": 616,  
"8476346": 414,  
"8479314": 233,  
"8475163": 484,  
"8477497": 491,  
"8470162": 545,  
"8476854": 502,  
"8477447": 240,

"8477986": 210,  
"8475222": 431,  
"8476312": 357,  
"8476400": 192,  
"8476483": 428,  
"8475780": 283,  
"8478131": 162,  
"8477941": 265,  
"8473560": 162,  
"8475878": 76,  
"8477508": 253,  
"8471879": 34,  
"8475182": 121,  
"8476882": 400,  
"8476806": 406,  
"8476453": 512,  
"8474870": 537,  
"8476292": 494,  
"8476975": 382,  
"8477205": 369,  
"8473986": 585,  
"8476300": 160,  
"8477494": 345,  
"8477845": 339,  
"8476869": 264,  
"8470595": 710,  
"8476468": 496,  
"8475855": 392,  
"8475763": 421,  
"8474602": 488,  
"8477839": 319,  
"8475832": 259,  
"8476874": 431,  
"8475208": 389,  
"8470110": 641,  
"8476897": 171,  
"8475795": 51,  
"8475810": 315,  
"8475715": 269,  
"8476919": 405,  
"8475820": 333,  
"8477922": 395,  
"8476442": 443,  
"8475455": 591,  
"8476881": 467,  
"8476404": 160,  
"8474568": 597,  
"8475188": 427,  
"8476406": 237,  
"8473571": 338,  
"8468505": 465,  
"8475807": 385,  
"8468513": 470,  
"8476906": 310,  
"8471762": 284,  
"8475430": 258,  
"8477930": 416,

"8474040": 590,  
"8475462": 431,  
"8476393": 265,  
"8477290": 431,  
"8475744": 486,  
"8477444": 384,  
"8475200": 502,  
"8477220": 385,  
"8476872": 281,  
"8471710": 134,  
"8471260": 659,  
"8471436": 405,  
"8476293": 210,  
"8476539": 317,  
"8476480": 389,  
"8474717": 260,  
"8475798": 522,  
"8475799": 588,  
"8475613": 376,  
"8475287": 401,  
"8471718": 157,  
"8471657": 596,  
"8474207": 456,  
"8474008": 103,  
"8476455": 604,  
"8471669": 664,  
"8470317": 198,  
"8469707": 484,  
"8473446": 588,  
"8470699": 370,  
"8471326": 136,  
"8475158": 685,  
"8475197": 505,  
"8474089": 205,  
"8477492": 481,  
"8476235": 216,  
"8475767": 16,  
"8473700": 173,  
"8475168": 658,  
"8470877": 389,  
"8474610": 144,  
"8476856": 368,  
"8474849": 497,  
"8473994": 695,  
"8475236": 583,  
"8476889": 311,  
"8474098": 402,  
"8477501": 240,  
"8469466": 415,  
"8475906": 393,  
"8477406": 254,  
"8475730": 163,  
"8471274": 700,  
"8474818": 486,  
"8476545": 42,  
"8475147": 170,  
"8474627": 346,

"8476436": 317,  
"8476879": 465,  
"8474884": 449,  
"8474697": 330,  
"8476808": 282,  
"8475758": 446,  
"8476470": 349,  
"8473507": 627,  
"8477455": 191,  
"8471296": 495,  
"8474688": 254,  
"8471241": 190,  
"8474034": 522,  
"8471299": 162,  
"8476884": 202,  
"8476409": 107,  
"8477506": 180,  
"8476458": 436,  
"8475177": 406,  
"8476386": 139,  
"8468001": 559,  
"8476428": 97,  
"8477932": 397,  
"8475755": 269,  
"8475760": 435,  
"8476456": 473,  
"8475253": 94,  
"8475191": 551,  
"8477493": 419,  
"8476875": 245,  
"8475735": 147,  
"8476389": 367,  
"8467423": 382,  
"8467396": 357,  
"8475747": 268,  
"8476116": 318,  
"8462196": 297,  
"8475246": 314,  
"8474535": 57,  
"8475218": 531,  
"8476925": 316,  
"8476887": 456,  
"8477942": 241,  
"8475766": 294,  
"8475793": 645,  
"8478042": 333,  
"8476892": 379,  
"8474145": 375,  
"8475175": 478,  
"8476441": 318,  
"8475714": 449,  
"8476447": 192,  
"8477964": 194,  
"8476158": 95,  
"8477482": 111,  
"8471217": 636,  
"8477300": 30,



"8474850": 66,  
"8478550": 349,  
"8476979": 160,  
"8477952": 185,  
"8475209": 422,  
"8471185": 305,  
"8469476": 348,  
"8477939": 259,  
"8475716": 207,  
"8476941": 207,  
"8477953": 153,  
"8479318": 232,  
"8475786": 271,  
"8476853": 490,  
"8477015": 273,  
"8478483": 261,  
"8475792": 465,  
"8479458": 241,  
"8476448": 379,  
"8478506": 22,  
"8477416": 218,  
"8476449": 165,  
"8476432": 446,  
"8474715": 554,  
"8477505": 370,  
"8476981": 262,  
"8475233": 516,  
"8474685": 522,  
"8477495": 495,  
"8478460": 256,  
"8477240": 156,  
"8468493": 701,  
"8477404": 245,  
"8477947": 74,  
"8478906": 217,  
"8476310": 164,  
"8475314": 438,  
"8475247": 117,  
"8476852": 39,  
"8476429": 172,  
"8474659": 71,  
"8476877": 43,  
"8477850": 234,  
"8478493": 156,  
"8476886": 117,  
"8477244": 147,  
"8471218": 724,  
"8470828": 488,  
"8473412": 657,  
"8470289": 253,  
"8473482": 358,  
"8476392": 386,  
"8469501": 533,  
"8470614": 425,  
"8470834": 636,  
"8476885": 435,  
"8475279": 329,

"8476460": 475,  
"8469490": 301,  
"8471385": 227,  
"8475739": 127,  
"8475868": 146,  
"8470180": 390,  
"8476857": 145,  
"8475060": 62,  
"8474722": 121,  
"8475214": 84,  
"8476207": 199,  
"8475834": 165,  
"8478414": 228,  
"8476336": 135,  
"8477934": 364,  
"8477498": 292,  
"8476454": 552,  
"8474586": 687,  
"8478402": 300,  
"8479465": 195,  
"8476472": 328,  
"8477415": 114,  
"8476988": 217,  
"8471766": 371,  
"8477919": 92,  
"8476866": 7,  
"8477946": 323,  
"8476822": 457,  
"8475772": 476,  
"8476289": 127,  
"8476960": 253,  
"8475193": 532,  
"8470714": 125,  
"8476440": 304,  
"8477591": 222,  
"8477589": 43,  
"8477500": 383,  
"8476452": 36,  
"8475223": 139,  
"8476466": 287,  
"8476326": 125,  
"8475213": 95,  
"8477913": 4,  
"8471760": 48,  
"8476430": 211,  
"8475157": 84,  
"8476525": 280,  
"8477956": 362,  
"8479325": 158,  
"8476374": 190,  
"8476191": 353,  
"8475727": 329,  
"8477213": 240,  
"8474062": 499,  
"8478366": 211,  
"8475227": 196,  
"8476368": 164,

"8476476": 173,  
"8476288": 292,  
"8477433": 112,  
"8478569": 215,  
"8474019": 4,  
"8478400": 95,  
"8475192": 97,  
"8471390": 378,  
"8473658": 125,  
"8475917": 22,  
"8478528": 76,  
"8477401": 114,  
"8477951": 183,  
"8477451": 258,  
"8476994": 179,  
"8477929": 142,  
"8479482": 199,  
"8476918": 252,  
"8474038": 425,  
"8476479": 274,  
"8477476": 227,  
"8478561": 154,  
"8475869": 165,  
"8477402": 184,  
"8477446": 70,  
"8478137": 115,  
"8474130": 151,  
"8476850": 331,  
"8471681": 344,  
"8471348": 264,  
"8469492": 464,  
"8469469": 391,  
"8477126": 115,  
"8474774": 264,  
"8471371": 271,  
"8470626": 641,  
"8475845": 10,  
"8476443": 161,  
"8474048": 168,  
"8475671": 91,  
"8475310": 100,  
"8476439": 244,  
"8475169": 612,  
"8478099": 244,  
"8476331": 220,  
"8477931": 176,  
"8477949": 185,  
"8477846": 149,  
"8475750": 322,  
"8477478": 149,  
"8476624": 228,  
"8477046": 129,  
"8476407": 36,  
"8477969": 114,  
"8474683": 212,  
"8476905": 174,  
"8477043": 133,

```
'"8477944"' : 206,  
'"8476826"' : 205,  
'"8478010"' : 250,  
'"8478519"' : 121,  
'"8479410"' : 179,  
'"8479400"' : 180,  
'"8480143"' : 39,  
'"8479339"' : 260,  
'"8478458"' : 125,  
'"8477940"' : 319,  
'"8478398"' : 201,  
'"8477429"' : 315,  
'"8477504"' : 245,  
'"8479293"' : 217,  
'"8476469"' : 250,  
'"8477960"' : 191,  
'"8476356"' : 383,  
'"8480147"' : 95,  
'"8476983"' : 69,  
'"8480113"' : 160,  
'"8475857"' : 107,  
'"8478020"' : 81,  
'"8476891"' : 160,  
'"8478498"' : 174,  
'"8478046"' : 195,  
'"8477987"' : 71,  
'"8477948"' : 135,  
'"8478439"' : 239,  
'"8478500"' : 252,  
'"8479974"' : 151,  
'"8478067"' : 108,  
'"8479944"' : 72,  
...})
```

```
In [84]: def sumValues(x,y):  
         return (x + y)  
  
sumPlayerGoals = player_goals.reduceByKey(sumValues)  
sumPlayerGoals.take(35)
```

```
Out[84]: [ ('"8468501"', 4),  
          ('"8470920"', 77),  
          ('"8473646"', 35),  
          ('"8474641"', 165),  
          ('"8464977"', 56),  
          ('"8469454"', 103),  
          ('"8469623"', 37),  
          ('"8460542"', 99),  
          ('"8459670"', 23),  
          ('"8470640"', 21),  
          ('"8470601"', 26),  
          ('"8474892"', 5),  
          ('"8448208"', 121),  
          ('"8464975"', 95),  
          ('"8474037"', 228),  
          ('"8474161"', 175),  
          ('"8476461"', 142),  
          ('"8471636"', 0),  
          ('"8471236"', 31),  
          ('"8474736"', 15),  
          ('"8470604"', 247),  
          ('"8467452"', 5),  
          ('"8470662"', 8),  
          ('"8470121"', 14),  
          ('"8474594"', 27),  
          ('"8470755"', 73),  
          ('"8475325"', 25),  
          ('"8474563"', 101),  
          ('"8473453"', 74),  
          ('"8474100"', 63),  
          ('"8468508"', 209),  
          ('"8475185"', 20),  
          ('"8466182"', 8),  
          ('"8470905"', 2),  
          ('"8466142"', 22)]
```

```
In [85]: NH_player_names = player_info.first()  
print(NH_player_names)  
  
"player_id","firstName","lastName","nationality","birthCity","primaryPo  
sition","birthDate","link"
```

```
In [106]: NH1_player_names = player_info.filter(lambda x: x != NH_player_names)
NH1_player_names.take(5)
```

```
Out[106]: [ ("8467412", "Alexei", "Ponikarovsky", "UKR", "Kiev", "LW", 1980-04-09, "/api/v1/people/8467412",
            "8468501", "Anton", "Volchenkov", "RUS", "Moscow", "D", 1982-02-25, "/api/v1/people/8468501",
            "8459670", "Kimmo", "Timonen", "FIN", "Kuopio", "D", 1975-03-18, "/api/v1/people/8459670",
            "8471233", "Travis", "Zajac", "CAN", "Winnipeg", "C", 1985-05-13, "/api/v1/people/8471233",
            "8455710", "Martin", "Brodeur", "CAN", "Montreal", "G", 1972-05-06, "/api/v1/people/8455710") ]
```

```
In [107]: player_names = NH1_player_names.map(lambda line: (line.split(",")[0], ((
line.rsplitt(", ", 1)[0]).split(", ", 1)[1], line.rsplitt(", ", 1)[1])))
player_names.take(5)
```

```
Out[107]: [ ("8467412",
            ("Alexei", "Ponikarovsky", "UKR", "Kiev", "LW", 1980-04-09',
            "/api/v1/people/8467412")),
            ("8468501",
            ("Anton", "Volchenkov", "RUS", "Moscow", "D", 1982-02-25',
            "/api/v1/people/8468501")),
            ("8459670",
            ("Kimmo", "Timonen", "FIN", "Kuopio", "D", 1975-03-18',
            "/api/v1/people/8459670")),
            ("8471233",
            ("Travis", "Zajac", "CAN", "Winnipeg", "C", 1985-05-13',
            "/api/v1/people/8471233")),
            ("8455710",
            ("Martin", "Brodeur", "CAN", "Montreal", "G", 1972-05-06',
            "/api/v1/people/8455710")) ]
```

```
In [111]: player_names1 = player_names.filter(lambda x: x != [0])
player_names1.take(5)
```

```
Out[111]: [ ("8467412",
            ("Alexei", "Ponikarovsky", "UKR", "Kiev", "LW", 1980-04-09',
            "/api/v1/people/8467412")),
            ("8468501",
            ("Anton", "Volchenkov", "RUS", "Moscow", "D", 1982-02-25',
            "/api/v1/people/8468501")),
            ("8459670",
            ("Kimmo", "Timonen", "FIN", "Kuopio", "D", 1975-03-18',
            "/api/v1/people/8459670")),
            ("8471233",
            ("Travis", "Zajac", "CAN", "Winnipeg", "C", 1985-05-13',
            "/api/v1/people/8471233")),
            ("8455710",
            ("Martin", "Brodeur", "CAN", "Montreal", "G", 1972-05-06',
            "/api/v1/people/8455710")) ]
```

```
In [112]: augmentedPG = sumPlayerGoals.join(player_names)
augmentedPG.take(5)
```

```
Out[112]: [( ' "8468501" ',
            (4,
              ( ' "Anton","Volchenkov","RUS","Moscow","D",1982-02-25 ',
                '"/api/v1/people/8468501"' )) ),
            ( ' "8464977" ',
              (56,
                ( ' "Dainius","Zubrus","LTU","Elektrenai","C",1978-06-16 ',
                  '"/api/v1/people/8464977"' )) ),
            ( ' "8475603" ',
              (0,
                ( ' "Ben","Holmstrom","USA","Colorado Springs","C",1987-04-09 ',
                  '"/api/v1/people/8475603"' )) ),
            ( ' "8474641" ',
              (165,
                ( ' "Adam","Henrique","CAN","Brantford","C",1990-02-06 ',
                  '"/api/v1/people/8474641"' )) ),
            ( ' "8476461" ',
              (142,
                ( ' "Sean","Couturier","CAN","Phoenix","C",1992-12-07 ',
                  '"/api/v1/people/8476461"' )) ) )]
```

```
In [114]: augmentedPG.takeOrdered(10, key = lambda x : -x[1][0])
```

```
Out[114]: [( '8471214"',
(434,
( 'Alex", "Ovechkin", "RUS", "Moscow", "LW", 1985-09-17',
'"/api/v1/people/8471214"' ))),
( '8474564"',
(342,
( 'Steven", "Stamkos", "CAN", "Markham", "C", 1990-02-07',
'"/api/v1/people/8474564"' ))),
( '8474141"',
(311,
( 'Patrick", "Kane", "USA", "Buffalo", "RW", 1988-11-19',
'"/api/v1/people/8474141"' ))),
( '8475166"',
(308,
( 'John", "Tavares", "CAN", "Mississauga", "C", 1990-09-20',
'"/api/v1/people/8475166"' ))),
( '8470794"',
(305,
( 'Joe", "Pavelski", "USA", "Plover", "C", 1984-07-11',
'"/api/v1/people/8470794"' ))),
( '8471675"',
(299,
( 'Sidney", "Crosby", "CAN", "Cole Harbour", "C", 1987-08-07',
'"/api/v1/people/8471675"' ))),
( '8473419"',
(292,
( 'Brad", "Marchand", "CAN", "Halifax", "LW", 1988-05-11',
'"/api/v1/people/8473419"' ))),
( '8473548"',
(285,
( 'Phil", "Kessel", "USA", "Madison", "RW", 1987-10-02',
'"/api/v1/people/8473548"' ))),
( '8471215"',
(282,
( 'Evgeni", "Malkin", "RUS", "Magnitogorsk", "C", 1986-07-31',
'"/api/v1/people/8471215"' ))),
( '8474053"',
(279,
( 'Logan", "Couture", "CAN", "Guelph", "C", 1989-03-28',
'"/api/v1/people/8474053"' )))]
```

## SQL, Top 20 in Penalty Minutes

```
In [116]: sqlContext = pyspark.SQLContext(sc)
sqlContext
```

```
Out[116]: <pyspark.sql.context.SQLContext at 0x7fe5942b5ad0>
```



```
In [185]: SQL_game_skater_stats = sqlContext.read.format("com.databricks.spark.csv")\
          .option("header", "true")\
          .option("inferSchema", "true")\
          .load("./data/game_skater_stats.csv")\
          .cache()
```

```
In [120]: SQL_game_skater_stats.printSchema()
```

```
root
|-- game_id: integer (nullable = true)
|-- player_id: integer (nullable = true)
|-- team_id: integer (nullable = true)
|-- timeOnIce: integer (nullable = true)
|-- assists: integer (nullable = true)
|-- goals: integer (nullable = true)
|-- shots: integer (nullable = true)
|-- hits: integer (nullable = true)
|-- powerPlayGoals: integer (nullable = true)
|-- powerPlayAssists: integer (nullable = true)
|-- penaltyMinutes: integer (nullable = true)
|-- faceOffWins: integer (nullable = true)
|-- faceoffTaken: integer (nullable = true)
|-- takeaways: integer (nullable = true)
|-- giveaways: integer (nullable = true)
|-- shortHandedGoals: integer (nullable = true)
|-- shortHandedAssists: integer (nullable = true)
|-- blocked: integer (nullable = true)
|-- plusMinus: integer (nullable = true)
|-- evenTimeOnIce: integer (nullable = true)
|-- shortHandedTimeOnIce: integer (nullable = true)
|-- powerPlayTimeOnIce: integer (nullable = true)
```

```
In [127]: SQL_game_skater_stats.registerTempTable("SQL_game_skater_stats")
```

```
In [170]: %%time
mostPenaltyMinutes = sqlContext.sql("SELECT player_id, plusMinus, SUM(pe
naltyMinutes) AS penaltyMinutes \
                                     FROM SQL_game_skater_stats \
                                     GROUP BY player_id, plusMinus\
                                     ORDER BY penaltyMinutes")
mostPenaltyMinutes.show()
```

player_id	plusMinus	penaltyMinutes
8474013	-4	0
8477402	3	0
8473473	4	0
8476442	2	0
8479974	-2	0
8474707	4	0
8468575	-3	0
8467514	4	0
8475820	-2	0
8473546	-3	0
8473564	-4	0
8474149	3	0
8478915	-1	0
8471413	0	0
8470283	-2	0
8465005	3	0
8467831	3	0
8473604	5	0
8473526	-3	0
8479442	-4	0

only showing top 20 rows

CPU times: user 1.4 ms, sys: 1.99 ms, total: 3.39 ms  
Wall time: 1.03 s

```
In [174]: SQL_player_info = sqlContext.read.format("com.databricks.spark.csv")\
.option("header", "true")\
.option("inferSchema", "true")\
.load("./data/player_info.csv")\
.cache()
```

```
In [175]: SQL_player_info.printSchema()
```

```
root
|-- player_id: integer (nullable = true)
|-- firstName: string (nullable = true)
|-- lastName: string (nullable = true)
|-- nationality: string (nullable = true)
|-- birthCity: string (nullable = true)
|-- primaryPosition: string (nullable = true)
|-- birthDate: timestamp (nullable = true)
|-- link: string (nullable = true)
```

```
In [177]: SQL_player_info.registerTempTable("SQL_player_info")
```

```
In [182]: %%time
mostPenaltyMinutes = sqlContext.sql("SELECT s.player_id, p.firstName, p.
lastName, SUM(s.penaltyMinutes) AS penaltyMinutes \
FROM SQL_game_skater_stats AS s \
INNER JOIN SQL_player_info AS p \
ON p.player_id = s.player_id \
GROUP BY p.firstName, p.lastName, s.
player_id \
ORDER BY penaltyMinutes DESC")
mostPenaltyMinutes.show()
```

player_id	firstName	lastName	penaltyMinutes
8471657	Cody	McLeod	1259
8473432	Derek	Dorsett	1126
8467493	Chris	Neil	1122
8476880	Tom	Wilson	1039
8474849	Antoine	Roussel	967
8474190	Wayne	Simmonds	965
8474709	Matt	Martin	924
8471817	Ryan	Reaves	921
8473473	Milan	Lucic	916
8471283	Brandon	Prust	856
8470834	Dustin	Byfuglien	854
8468486	Scott	Hartnell	852
8475169	Evander	Kane	846
8474056	P.K.	Subban	843
8475160	Kyle	Clifford	836
8470602	Dion	Phaneuf	816
8474736	Zac	Rinaldo	783
8468505	Steve	Ott	772
8473419	Brad	Marchand	771
8470655	David	Backes	768

only showing top 20 rows

CPU times: user 2.18 ms, sys: 1.08 ms, total: 3.26 ms

Wall time: 980 ms

## Games with the highest faceOffPercentageWins W/ Head Coach

```
In [188]: SQL_game_teams_stats = sqlContext.read.format("com.databricks.spark.csv")\
.option("header", "true")\
.option("inferSchema", "true")\
.load("./data/game_teams_stats.csv")\
.cache()
```

```
In [189]: SQL_game_teams_stats.printSchema()
```

```
root
|-- game_id: integer (nullable = true)
|-- team_id: integer (nullable = true)
|-- HoA: string (nullable = true)
|-- won: boolean (nullable = true)
|-- settled_in: string (nullable = true)
|-- head_coach: string (nullable = true)
|-- goals: integer (nullable = true)
|-- shots: integer (nullable = true)
|-- hits: integer (nullable = true)
|-- pim: integer (nullable = true)
|-- powerPlayOpportunities: integer (nullable = true)
|-- powerPlayGoals: integer (nullable = true)
|-- faceOffWinPercentage: double (nullable = true)
|-- giveaways: integer (nullable = true)
|-- takeaways: integer (nullable = true)
```

```
In [190]: SQL_game_teams_stats.registerTempTable("SQL_game_teams_stats")
```

```
In [226]: SQL_team_info = sqlContext.read.format("com.databricks.spark.csv")\
    .option("header", "true")\
    .option("inferSchema", "true")\
    .load("./data/team_info.csv")\
    .cache()
```

```
In [227]: SQL_team_info.printSchema()
```

```
root
|-- team_id: integer (nullable = true)
|-- franchiseId: integer (nullable = true)
|-- shortName: string (nullable = true)
|-- teamName: string (nullable = true)
|-- abbreviation: string (nullable = true)
|-- link: string (nullable = true)
```

```
In [228]: SQL_team_info.registerTempTable("SQL_team_info")
```

```
In [242]: %%time
coachHighestFaceOffs = sqlContext.sql("SELECT game_id, head_coach, faceOffWinPercentage \
                                         FROM SQL_game_teams_stats \
                                         WHERE faceOffWinPercentage > 60.0 \
                                         GROUP BY game_id, head_coach, faceOffWinPercentage \
                                         ORDER BY faceOffWinPercentage DESC")

coachHighestFaceOffs.show()
```

game_id	head_coach	faceOffWinPercentage
2017020780	Bill Peters	79.2
2010030223	Claude Julien	78.2
2011020761	Todd McLellan	76.9
2012020675	Darryl Sutter	76.4
2016020999	Todd McLellan	75.6
2017020550	Bill Peters	75.0
2018021268	Bill Peters	74.1
2011020515	Joe Sacco	74.1
2018020314	Peter Laviolette	74.1
2013020054	Mike Yeo	73.8
2016020354	Jared Bednar	73.6
2014020435	Claude Julien	73.5
2013020990	Kirk Muller	73.4
2018020102	Dave Hakstol	73.3
2016020304	Randy Carlyle	73.1
2016020457	Jared Bednar	72.9
2018020613	Scott Gordon	72.9
2015021011	John Torchetti	72.7
2016020857	Randy Carlyle	72.5
2013021186	Ken Hitchcock	72.4

only showing top 20 rows

CPU times: user 652  $\mu$ s, sys: 2.73 ms, total: 3.38 ms

Wall time: 606 ms

```
In [233]: #%%time
#mostCoachWins = sqlContext.sql("SELECT game_id, head_coach, COUNT(won)\
#                                     FROM SQL_game_teams_stats \
#                                     WHERE won = 1 \
#                                     GROUP BY game_id, head_coach, won")
#
#mostCoachWins.show()
```

## Players With Most Hits

```
In [237]: %%time
mostHits = sqlContext.sql("SELECT player_id, SUM(hits) AS hits \
                           FROM SQL_game_skater_stats \
                           GROUP BY player_id, hits\
                           ORDER BY hits")

mostHits.show()
```

```
+-----+-----+
|player_id|hits|
+-----+-----+
| 8474581|    0|
| 8473574|    0|
| 8479580|    0|
| 8467407|    0|
| 8471862|    0|
| 8474573|    0|
| 8477957|    0|
| 8468504|    0|
| 8468526|    0|
| 8474762|    0|
| 8473999|    0|
| 8471413|    0|
| 8478449|    0|
| 8476779|    0|
| 8471685|    0|
| 8477326|    0|
| 8475637|    0|
| 8479250|    0|
| 8475060|    0|
| 8476323|    0|
+-----+-----+
```

only showing top 20 rows

CPU times: user 2.77 ms, sys: 955  $\mu$ s, total: 3.73 ms  
Wall time: 744 ms

```
In [239]: %%time
mostHits = sqlContext.sql("SELECT s.player_id, p.firstName, p.lastName,
    SUM(s.hits) AS hits \
                                FROM SQL_game_skater_stats AS s \
                                INNER JOIN SQL_player_info AS p \
                                ON p.player_id = s.player_id \
                                GROUP BY p.firstName, p.lastName, s.
    player_id \
                                ORDER BY hits DESC")
mostHits.show()
```

player_id	firstName	lastName	hits
8474709	Matt	Martin	2991
8473504	Cal	Clutterbuck	2474
8470606	Dustin	Brown	2381
8473473	Milan	Lucic	2283
8471214	Alex	Ovechkin	2270
8470655	David	Backes	2138
8468498	Brooks	Orpik	2083
8471817	Ryan	Reaves	2065
8474568	Luke	Schenn	2051
8471426	Troy	Brouwer	1908
8471339	Ryan	Callahan	1835
8470543	Chris	Kunitz	1824
8471392	Roman	Polak	1823
8471958	Dan	Girardi	1789
8470619	Brian	Boyle	1747
8471273	Brandon	Dubinsky	1720
8471716	Justin	Abdelkader	1657
8476880	Tom	Wilson	1613
8470602	Dion	Phaneuf	1607
8473422	Nick	Foligno	1603

only showing top 20 rows

CPU times: user 1.53 ms, sys: 2.03 ms, total: 3.56 ms

Wall time: 606 ms

In [ ]: