

research

August 4, 2021

```
[4]: #https://baseballsavant.mlb.com/csv-docs#plate_x
from pybaseball import statcast
from pybaseball import playerid_lookup
from pybaseball import statcast_pitcher, pitching_stats
from pybaseball import statcast_batter, batting_stats_range, batting_stats
from pybaseball import team_ids, teams, team_batting, team_batting_bref
from pybaseball import cache, standings, get_splits
import pandas as pd
import os
import matplotlib
from matplotlib import pyplot as plt
from matplotlib.patches import Rectangle
import matplotlib.patches as mpatchess
cache.enable()

def Average(lst):
    return(sum(lst)/len(lst))

start_date = '2019-01-01' #yy/mm/dd
end_date = '2019-12-31'
data = statcast(start_date, end_date)
#print(data.head())

#### all stats from year 2020
#stats = batting_stats(2020)
#print(stats)
#print("===== AARON JUDGE_
    ↳=====")
#### Player lookup
#print(playerid_lookup('Judge', 'Aaron'))
#print()

#### Data on player
#player_data = statcast_batter('2016-04-01', '2017-07-15', player_id = 592450)
#print(player_data)
#df, player_info_dict = get_splits('judgeaa01', player_info = True)
#print(df)
```

```

#print()

##### Turns homeruns column to a list
#HRList = df['HR'].to_list()
#print("Home runs in the last 365 days for Arron Judge: ", HRList[4])

#### Team batting stats for only 2019 season
#dataTeam = team_batting(2019)
#print(dataTeam)
#teams = team_ids(2019)
#batting = team_batting(2019).add_prefix('batting.')
#teams.merge(batting, left_on=['yearID', 'teamIDfg'], right_on=['batting.
→Season', 'batting.teamIDfg'])

#print("===== TEAM DATA_
→=====")
#dataNYY = team_batting_bref('NYY', 2019)
#print(dataNYY)
#all_hits = dataNYY['HR'].to_list()
#total = 0
#for num in all_hits:
#    total += int(num)
#print("Total home runs for NYY: ", total)

print("===== PITCH DATA NYY_
→===== \n")
dataNYY = team_batting_bref('NYY', 2019)
players = dataNYY['Name'].to_list()
listPlayers = []    #### all NYY players
for playerName in players:
    name = playerName.split()
    listPlayers.append(name)

#### Stores players' ID in dict... Omits 2 players - no data
playerNums = {}
for player in listPlayers:
    playerName = player[0] + " " + player[1]
    playerLookUp = playerid_lookup(player[1], player[0])
    playerNum = playerLookUp['key_mlbam'].to_list()
    if len(playerNum) != 0:
        playerNums[playerName] = playerNum[0]    #### {'key': 'value'}

#print(data.columns.get_loc('plate_x'))    #### finds index
#print(data.columns.get_loc('plate_z'))
#print(data.columns.get_loc('sz_top'))
#print(data.columns.get_loc('sz_bot'))
#print(data.columns.get_loc('balls'))

```

```

#print(data.columns.get_loc('strikes'))

index = data.index
num_rows = len(index)

YankAtBats = []      ##### all yank pitches
xPosList = []        ##### all x positions
zPosList = []        ##### all z positions
DescList = []        ##### all descriptions of pitches
topSZ = []           ##### top of strike zone - predetermined
botSZ = []           ##### bottom of strike zone - predetermined
pType = []           ##### pitch type
Names = []           ##### batter name
Count_3_2 = []
i = 0
gardyT = []
gardyB = []
for x in range(num_rows):      #ONLY FIRST 100000    CHANGE THIS BACK
    ID = (data.iloc[x]).iloc[6]
    name = [key for key, v in playerNums.items() if v == ID]
    if name != [] and name[0] in playerNums:
        Names.append(name[0])
        xPosList.append((data.iloc[x]).iloc[29])
        zPosList.append((data.iloc[x]).iloc[30])
        pType.append((data.iloc[x]).iloc[0])
        DescList.append((data.iloc[x]).iloc[9])
        topSZ.append((data.iloc[x]).iloc[50])
        botSZ.append((data.iloc[x]).iloc[51])
        if ((data.iloc[x]).iloc[24]) == 3 and ((data.iloc[x]).iloc[25]) == 2:
            ##### pulls out all 3-2 counts
            Count_3_2.append(1)
        else:
            Count_3_2.append(0)
        if name[0] == "Brett Gardner":
            gardyT.append((data.iloc[x]).iloc[50])
            gardyB.append((data.iloc[x]).iloc[51])
        atBat = [Names[i], xPosList[i], zPosList[i], pType[i], DescList[i]]
        YankAtBats.append(atBat)
        i += 1

##### Classifications
classification = []
j = 0
for yankPitch in YankAtBats:
    ## strike
    if (-0.71 < xPosList[j] < 0.71) and (botSZ[j] < zPosList[j] < topSZ[j]):

```

```

        if DescList[j] == "hit_into_play" or DescList[j] == "swinging_strike"
        or DescList[j] == "foul" or DescList[j] == "swinging_strike_blocked" or
        DescList[j] == "foul_tip":
            classification.append("strike, good") # swung
        else:
            classification.append("strike, bad") # didn't swing
        ## not strike
    else:
        if DescList[j] == "hit_into_play" or DescList[j] == "swinging_strike"
        or DescList[j] == "foul" or DescList[j] == "swinging_strike_blocked" or
        DescList[j] == "foul_tip":
            classification.append("ball, bad") # swung
        else:
            classification.append("ball, good") # didn't swing
    YankAtBats[j].append(classification[j])
    j += 1

#### New data frame of all NYY pitches (size = 28512, last 53 bad)
newdf = pd.DataFrame(YankAtBats[0:len(YankAtBats)-53], columns = ['Name',
    'xPos', 'zPos', 'Pitch Type', 'Given Description', 'Classification'])
print("FA - Fastball, CU - Curveball, FT - Two-seam Fastball, CH - Changeup,
    \nFC - Cutter, SL - Slider, FS - Splitter, KN - Knuckleball,\nFF - Four-seam
    \nFastball \n")
print(newdf, "\n")

```

This is a large query, it may take a moment to complete

```

0%|          | 0/225 [00:00<?, ?it/s]

Skipping offseason dates
Skipping offseason dates

100%|         | 225/225 [01:38<00:00, 2.28it/s]

===== PITCH DATA NYY =====

FA - Fastball, CU - Curveball, FT - Two-seam Fastball, CH - Changeup,
FC - Cutter, SL - Slider, FS - Splitter, KN - Knuckleball,
FF - Four-seam Fastball

```

	Name	xPos	zPos	Pitch Type	Given Description \
0	Gleyber Torres	-0.49	3.21	FT	hit_into_play
1	Gleyber Torres	0.39	1.85	SL	called_strike
2	Aaron Judge	0.31	2.99	FF	swinging_strike
3	Aaron Judge	0.77	1.03	FC	ball
4	Aaron Judge	0.83	1.81	FC	called_strike
...
28454	Edwin Encarnacion	-0.58	2.31	SL	hit_into_play
28455	Edwin Encarnacion	-1.11	4.14	FF	ball

28456	Edwin Encarnacion	-0.39	1.81	SL	swinging_strike
28457	Edwin Encarnacion	0.30	1.09	CH	ball
28458	Edwin Encarnacion	0.92	3.22	FT	called_strike

```

Classification
0      ball, bad
1      strike, bad
2      strike, good
3      ball, good
4      ball, good
...
28454  strike, good
28455  ball, good
28456  strike, good
28457  ball, good
28458  ball, good

```

[28459 rows x 6 columns]

```

[34]: ##### BRETT GARDNER STATS
print("===== BRETT GARDNER DATA_
→===== \n")
avgTop = Average(gardyT)
avgBot = Average(gardyB)
Gardy = []
S_PitchCount = 0 # pitch count
S_Count = 0 # strike count
B_Count = 0 # ball count
goodStrikes = [] #list of tuples - (xpos, zpos)
badStrikes = []
goodBalls = []
badBalls = []
for x in range(len(newdf)): ##### ***** add pitch coordinates as tuples_
→to color code plot *****
    if ((newdf.iloc[x]).iloc[0]) == "Brett Gardner":
        Gardy.append(newdf.loc[x, :].values.flatten().tolist())
        S_PitchCount += 1
        if ((newdf.iloc[x]).iloc[5]) == "strike, good" or ((newdf.iloc[x]).
→iloc[5]) == "strike, bad":
            S_Count += 1
            if ((newdf.iloc[x]).iloc[5]) == "strike, good":
                goodStrikes.append(((newdf.iloc[x]).iloc[1], (newdf.iloc[x]).
→iloc[2]))
            else:
                badStrikes.append(((newdf.iloc[x]).iloc[1], (newdf.iloc[x]).
→iloc[2]))

```

```

        else:
            B_Count += 1
            if ((newdf.iloc[x]).iloc[5]) == "ball, good":
                goodBalls.append(((newdf.iloc[x]).iloc[1], (newdf.iloc[x]).
→iloc[2]))
            else:
                badBalls.append(((newdf.iloc[x]).iloc[1], (newdf.iloc[x]).
→iloc[2]))

#### Gardner Data Frame
GarddyDF = pd.DataFrame(Garddy, columns = ['Name', 'xPos', 'zPos', 'Pitch Type',
→'Given Description', 'Classification'])
#print(GarddyDF.head(len(GarddyDF.index)).to_string())
print("FA - Fastball, CU - Curveball, FT - Two-seam Fastball, CH - Changeup,
→\nFC - Cutter, SL - Slider, FS - Splitter, KN - Knuckleball,\nFF - Four-seam
→Fastball \n")
print(GarddyDF)
print()

percent_Good = len(goodStrikes) / S_Count
percent_Bad = len(goodBalls) / B_Count
print("% of \"good\" strikes: ", round(percent_Good,3) )
print("% of \"good\" balls:", round(percent_Bad,3) )
print("(% good strikes) + (% good balls): ", round((percent_Good +
→percent_Bad),3) )

#### Gardner Scatter Plot
#fig, ax = plt.subplots()
#plt.scatter(GarddyDF['xPos'], GarddyDF['zPos'])
#plt.xlabel("x position (ft)",fontweight='bold', size=14)
#plt.ylabel("z position (ft)", fontweight='bold',size=14)
#plt.title("Brett Gardner Pitch Data", fontweight='bold',size=18)
#ax.add_patch(Rectangle((-0.79, 1.57), 1.58, 1.77,edgecolor =
→'red',fill=False,lw=2))
#plt.show()

fig = plt.figure()
ax1 = fig.add_subplot(111)
ax1.set_xlim(-1,1)
ax1.set_ylim(0,5)
for x in range(len(goodStrikes)):
    p1 = ax1.scatter(goodStrikes[x][0],goodStrikes[x][1], s=10, c='g',
→marker="o")
for y in range(len(badStrikes)):
    p2 = ax1.scatter(badStrikes[y][0], badStrikes[y][1], s=10, c='r',
→marker="o")

```

```
plt.title("Brett Gardner Strike Data", fontweight='bold',size=18)
plt.legend([p1,p2], ["good strikes", "bad strikes"])
ax1.add_patch(Rectangle((-0.79, avgBot), 1.58, avgTop-avgBot ,edgecolor =_
    ↪'red',fill=False,lw=2))
plt.show()

fig = plt.figure()
ax2 = fig.add_subplot(111)
ax2.set_xlim(-2.75,2.75)
ax2.set_ylim(0,5)
for x in range(len(goodBalls)):
    p21 = ax2.scatter(goodBalls[x][0],goodBalls[x][1], s=10, c='g', marker="o")
for y in range(len(badBalls)):
    p22 = ax2.scatter(badBalls[y][0], badBalls[y][1], s=10, c='r', marker="o", )
plt.title("Brett Gardner Ball Data", fontweight='bold',size=18)
plt.legend([p21,p22], ["good balls", "bad balls"])
ax2.add_patch(Rectangle((-0.79, 1.57), 1.58, 1.77,edgecolor =_
    ↪'red',fill=False,lw=2))
plt.show()
```

===== BRETT GARDNER DATA =====

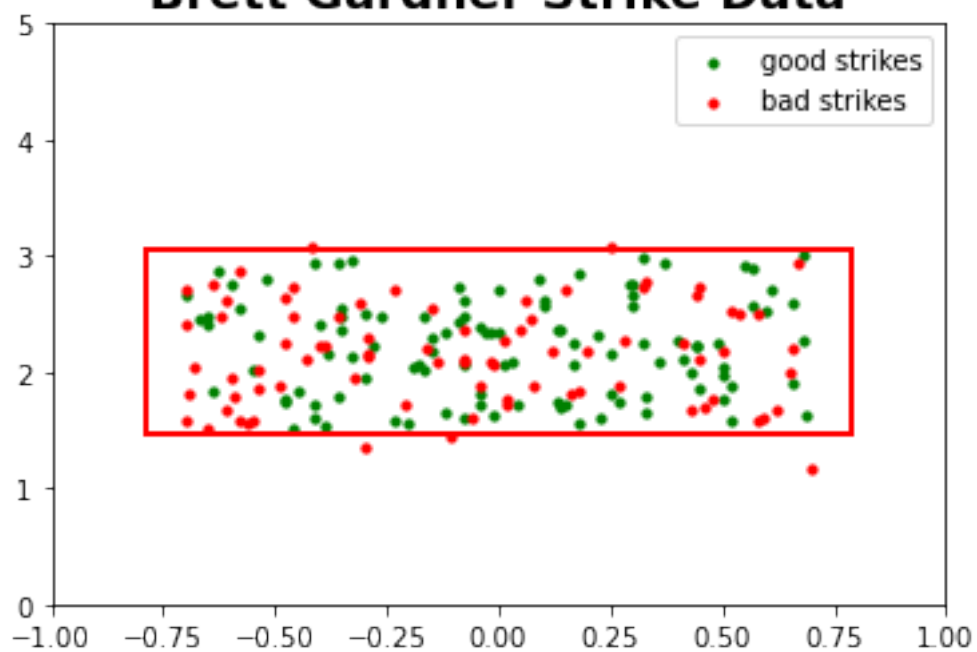
FA - Fastball, CU - Curveball, FT - Two-seam Fastball, CH - Changeup,
 FC - Cutter, SL - Slider, FS - Splitter, KN - Knuckleball,
 FF - Four-seam Fastball

	Name	xPos	zPos	Pitch Type	Given Description	Classification
0	Brett Gardner	0.14	1.69	CH	swinging_strike	strike, good
1	Brett Gardner	0.18	2.85	FF	foul	strike, good
2	Brett Gardner	-0.60	2.76	FF	swinging_strike	strike, good
3	Brett Gardner	1.50	3.60	FC	ball	ball, good
4	Brett Gardner	0.15	2.71	FF	called_strike	strike, bad
..
486	Brett Gardner	-0.08	2.61	FT	foul	strike, good
487	Brett Gardner	-0.40	2.23	CU	called_strike	strike, bad
488	Brett Gardner	-1.01	1.87	CU	ball	ball, good
489	Brett Gardner	1.41	2.35	FT	ball	ball, good
490	Brett Gardner	-0.70	1.59	FF	called_strike	strike, bad

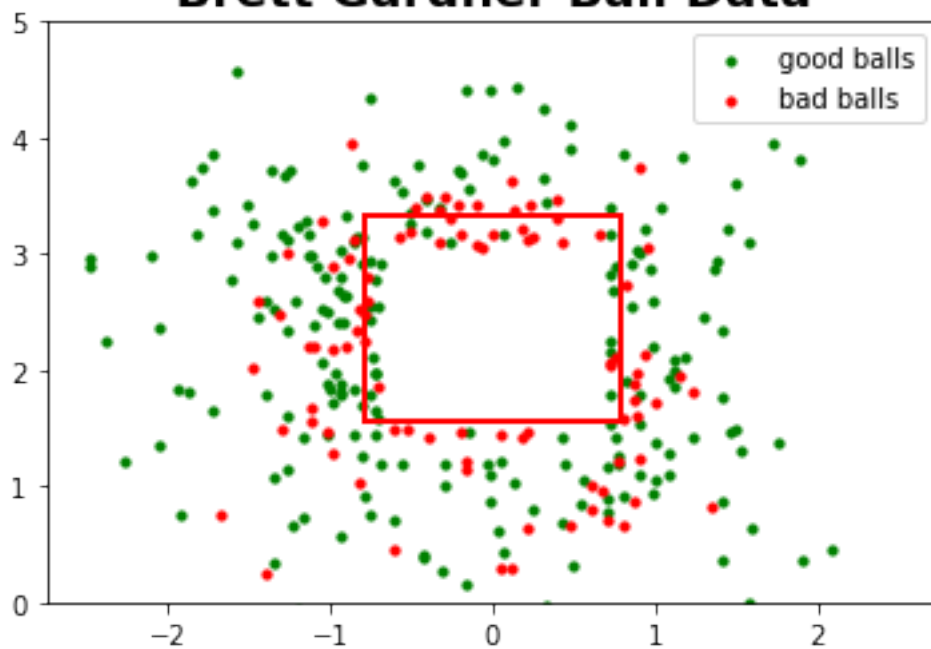
[491 rows x 6 columns]

% of "good" strikes: 0.568
 % of "good" balls: 0.701
 (% good strikes) + (% good balls): 1.269

Brett Gardner Strike Data



Brett Gardner Ball Data




```

[38]: ##### ALL NYY PLAYER STATS
from IPython.core.display import display, HTML
class color:
    PURPLE = '\033[95m'
    CYAN = '\033[96m'
    DARKCYAN = '\033[36m'
    BLUE = '\033[94m'
    GREEN = '\033[92m'
    YELLOW = '\033[93m'
    RED = '\033[91m'
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'
    END = '\033[0m'

display(HTML("<style>.container { width:100% !important; }</style>"))
print("===== NYY PLAYER STAT DATA_
↳===== \n")
nyyList = []
for yank in playerNums:
    num_pitches = 0
    num_strikes = 0
    num_balls = 0
    num_good_strikes = 0
    num_good_balls = 0
    for x in range(len(newdf)):
        if (newdf.iloc[x]).iloc[0] == yank:
            num_pitches += 1
            if ((newdf.iloc[x]).iloc[5]) == "strike, good" or ((newdf.iloc[x]).
↳iloc[5]) == "strike, bad":
                num_strikes += 1
                if ((newdf.iloc[x]).iloc[5]) == "strike, good":
                    num_good_strikes += 1
            else:
                num_balls += 1
                if ((newdf.iloc[x]).iloc[5]) == "ball, good":
                    num_good_balls += 1
    if num_strikes != 0 and num_balls != 0 and num_pitches >= 100:
        p_Good = num_good_strikes / num_strikes
        p_Bad = num_good_balls / num_balls
        nyyList.append([yank,round(num_pitches,3) ,round(num_strikes,3),_
↳round(num_balls,3), round(p_Good,3),round(p_Bad,3),round((p_Good +_
↳p_Bad),3)])

NYYDF = pd.DataFrame(nyyList, columns = ['Name', 'Pitches', 'Strikes', 'Balls',_
↳'g_strike', 'g_ball', 'Sum'])

ax = NYYDF.plot.barh(x='Name', y='g_strike', figsize=(12, 8)) # Bar graph plot

```

```

plt.title("Good Strikes", fontweight='bold',size=18)
plt.ylabel('Name', fontweight='bold', size=12)
plt.axvline(x=0.699, color='r', linestyle='--') # Draws average line
ax2 = NYYDF.plot.barh(x='Name', y='g_ball', figsize=(12, 8)) # Bar graph plot
plt.title("Good Balls", fontweight='bold',size=18)
plt.ylabel('Name', fontweight='bold', size=12)
plt.axvline(x=0.673, color='r', linestyle='--') # Draws average line

print(color.BOLD + "\t\t\t\t Ordered by name" + color.END)
print(NYYDF.sort_values(by='Name', ascending=True), "\n")
print(color.BOLD + "\t\t\t\t Ordered by pitches" + color.END)
sort = NYYDF.sort_values(by='Pitches', ascending=False)
print(sort, "\n")
print(color.BOLD + "\t\t\t\t Ordered by good strikes" + color.END)
sort1 = NYYDF.sort_values(by='g_strike', ascending=False)
print(sort1, "\n")
print(color.BOLD + "\t\t\t\t Ordered by good balls" + color.END)
sort2 = NYYDF.sort_values(by='g_ball', ascending=False)
print(sort2, "\n")
print(color.BOLD + "\t\t\t\t Ordered by sum" + color.END)
sort3 = NYYDF.sort_values(by='Sum', ascending=False)
print(sort3, "\n")

g_strike_AVG = NYYDF["g_strike"].mean() #### Gets average of whole column
g_ball_AVG = NYYDF["g_ball"].mean()
print("Team Good Strike Average: ", round(g_strike_AVG,3), "\nTeam Good Ball_
↪Average: ", round(g_ball_AVG,3))

```

<IPython.core.display.HTML object>

===== NYY PLAYER STAT DATA =====

		Ordered by name					
	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
11	Aaron Hicks	1151	399	752	0.657	0.726	1.383
7	Aaron Judge	2073	727	1346	0.693	0.724	1.417
13	Austin Romine	871	356	515	0.806	0.598	1.404
6	Brett Gardner	2544	991	1553	0.570	0.733	1.304
22	Breyvic Valera	205	78	127	0.744	0.724	1.468
10	Cameron Maybin	1154	445	709	0.649	0.722	1.372
12	Clint Frazier	976	382	594	0.725	0.742	1.468
2	DJ LeMahieu	2618	1021	1597	0.670	0.681	1.351
3	Didi Gregorius	1504	542	962	0.782	0.570	1.352
8	Edwin Encarnacion	2262	826	1436	0.688	0.698	1.385
0	Gary Sanchez	2053	647	1406	0.604	0.632	1.237
17	Giancarlo Stanton	395	121	274	0.612	0.686	1.298
4	Gio Urshela	1908	727	1181	0.761	0.556	1.317

9	Gleyber Torres	2551	953	1598	0.781	0.634	1.415
21	Greg Bird	206	76	130	0.632	0.677	1.309
16	Kendrys Morales	833	313	520	0.633	0.748	1.381
19	Kyle Higashioka	246	103	143	0.845	0.552	1.397
1	Luke Voit	2016	731	1285	0.804	0.715	1.520
20	Miguel Andujar	175	61	114	0.689	0.544	1.232
14	Mike Ford	657	245	412	0.649	0.748	1.397
5	Mike Tauchman	1288	491	797	0.731	0.755	1.486
18	Thairo Estrada	227	95	132	0.632	0.614	1.245
15	Tyler Wade	388	153	235	0.725	0.694	1.419

Ordered by pitches

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
2	DJ LeMahieu	2618	1021	1597	0.670	0.681	1.351
9	Gleyber Torres	2551	953	1598	0.781	0.634	1.415
6	Brett Gardner	2544	991	1553	0.570	0.733	1.304
8	Edwin Encarnacion	2262	826	1436	0.688	0.698	1.385
7	Aaron Judge	2073	727	1346	0.693	0.724	1.417
0	Gary Sanchez	2053	647	1406	0.604	0.632	1.237
1	Luke Voit	2016	731	1285	0.804	0.715	1.520
4	Gio Urshela	1908	727	1181	0.761	0.556	1.317
3	Didi Gregorius	1504	542	962	0.782	0.570	1.352
5	Mike Tauchman	1288	491	797	0.731	0.755	1.486
10	Cameron Maybin	1154	445	709	0.649	0.722	1.372
11	Aaron Hicks	1151	399	752	0.657	0.726	1.383
12	Clint Frazier	976	382	594	0.725	0.742	1.468
13	Austin Romine	871	356	515	0.806	0.598	1.404
16	Kendrys Morales	833	313	520	0.633	0.748	1.381
14	Mike Ford	657	245	412	0.649	0.748	1.397
17	Giancarlo Stanton	395	121	274	0.612	0.686	1.298
15	Tyler Wade	388	153	235	0.725	0.694	1.419
19	Kyle Higashioka	246	103	143	0.845	0.552	1.397
18	Thairo Estrada	227	95	132	0.632	0.614	1.245
21	Greg Bird	206	76	130	0.632	0.677	1.309
22	Breyvic Valera	205	78	127	0.744	0.724	1.468
20	Miguel Andujar	175	61	114	0.689	0.544	1.232

Ordered by good strikes

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
19	Kyle Higashioka	246	103	143	0.845	0.552	1.397
13	Austin Romine	871	356	515	0.806	0.598	1.404
1	Luke Voit	2016	731	1285	0.804	0.715	1.520
3	Didi Gregorius	1504	542	962	0.782	0.570	1.352
9	Gleyber Torres	2551	953	1598	0.781	0.634	1.415
4	Gio Urshela	1908	727	1181	0.761	0.556	1.317
22	Breyvic Valera	205	78	127	0.744	0.724	1.468
5	Mike Tauchman	1288	491	797	0.731	0.755	1.486
15	Tyler Wade	388	153	235	0.725	0.694	1.419

12	Clint Frazier	976	382	594	0.725	0.742	1.468
7	Aaron Judge	2073	727	1346	0.693	0.724	1.417
20	Miguel Andujar	175	61	114	0.689	0.544	1.232
8	Edwin Encarnacion	2262	826	1436	0.688	0.698	1.385
2	DJ LeMahieu	2618	1021	1597	0.670	0.681	1.351
11	Aaron Hicks	1151	399	752	0.657	0.726	1.383
14	Mike Ford	657	245	412	0.649	0.748	1.397
10	Cameron Maybin	1154	445	709	0.649	0.722	1.372
16	Kendrys Morales	833	313	520	0.633	0.748	1.381
18	Thairo Estrada	227	95	132	0.632	0.614	1.245
21	Greg Bird	206	76	130	0.632	0.677	1.309
17	Giancarlo Stanton	395	121	274	0.612	0.686	1.298
0	Gary Sanchez	2053	647	1406	0.604	0.632	1.237
6	Brett Gardner	2544	991	1553	0.570	0.733	1.304

Ordered by good balls

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
5	Mike Tauchman	1288	491	797	0.731	0.755	1.486
16	Kendrys Morales	833	313	520	0.633	0.748	1.381
14	Mike Ford	657	245	412	0.649	0.748	1.397
12	Clint Frazier	976	382	594	0.725	0.742	1.468
6	Brett Gardner	2544	991	1553	0.570	0.733	1.304
11	Aaron Hicks	1151	399	752	0.657	0.726	1.383
7	Aaron Judge	2073	727	1346	0.693	0.724	1.417
22	Breyvic Valera	205	78	127	0.744	0.724	1.468
10	Cameron Maybin	1154	445	709	0.649	0.722	1.372
1	Luke Voit	2016	731	1285	0.804	0.715	1.520
8	Edwin Encarnacion	2262	826	1436	0.688	0.698	1.385
15	Tyler Wade	388	153	235	0.725	0.694	1.419
17	Giancarlo Stanton	395	121	274	0.612	0.686	1.298
2	DJ LeMahieu	2618	1021	1597	0.670	0.681	1.351
21	Greg Bird	206	76	130	0.632	0.677	1.309
9	Gleyber Torres	2551	953	1598	0.781	0.634	1.415
0	Gary Sanchez	2053	647	1406	0.604	0.632	1.237
18	Thairo Estrada	227	95	132	0.632	0.614	1.245
13	Austin Romine	871	356	515	0.806	0.598	1.404
3	Didi Gregorius	1504	542	962	0.782	0.570	1.352
4	Gio Urshela	1908	727	1181	0.761	0.556	1.317
19	Kyle Higashioka	246	103	143	0.845	0.552	1.397
20	Miguel Andujar	175	61	114	0.689	0.544	1.232

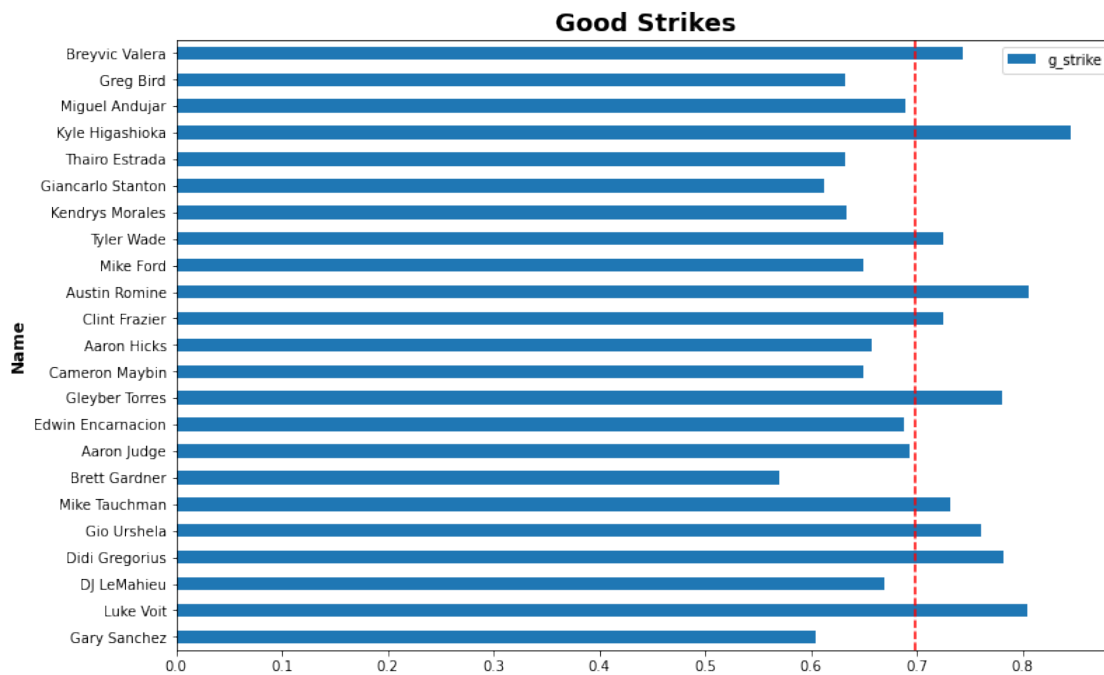
Ordered by sum

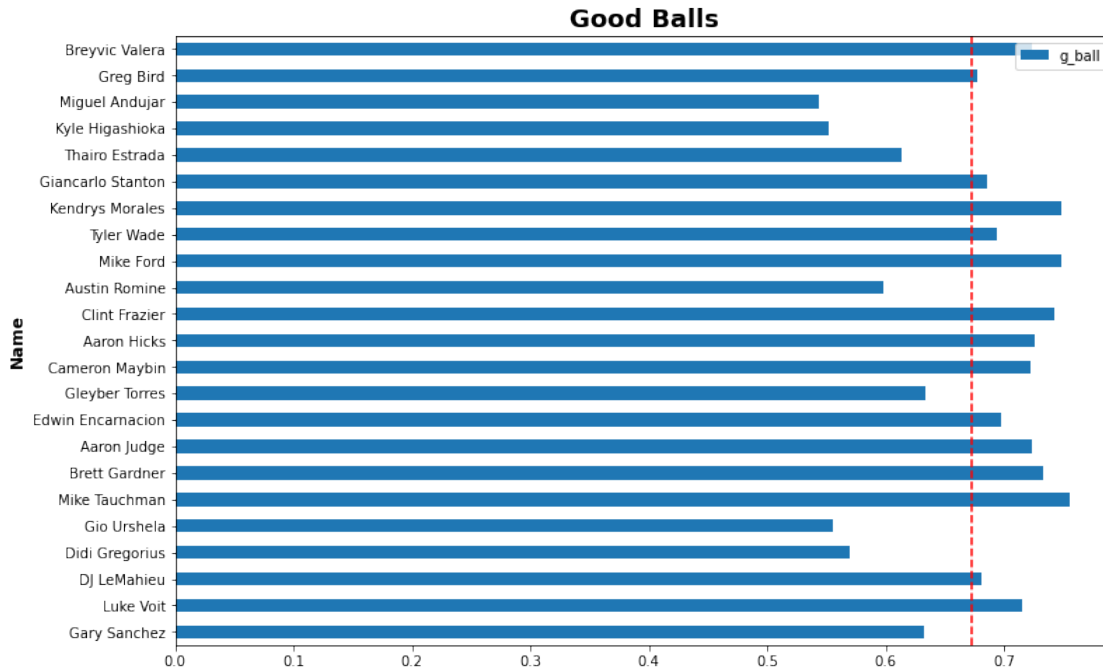
	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
1	Luke Voit	2016	731	1285	0.804	0.715	1.520
5	Mike Tauchman	1288	491	797	0.731	0.755	1.486
22	Breyvic Valera	205	78	127	0.744	0.724	1.468
12	Clint Frazier	976	382	594	0.725	0.742	1.468
15	Tyler Wade	388	153	235	0.725	0.694	1.419

7	Aaron Judge	2073	727	1346	0.693	0.724	1.417
9	Gleyber Torres	2551	953	1598	0.781	0.634	1.415
13	Austin Romine	871	356	515	0.806	0.598	1.404
19	Kyle Higashioka	246	103	143	0.845	0.552	1.397
14	Mike Ford	657	245	412	0.649	0.748	1.397
8	Edwin Encarnacion	2262	826	1436	0.688	0.698	1.385
11	Aaron Hicks	1151	399	752	0.657	0.726	1.383
16	Kendrys Morales	833	313	520	0.633	0.748	1.381
10	Cameron Maybin	1154	445	709	0.649	0.722	1.372
3	Didi Gregorius	1504	542	962	0.782	0.570	1.352
2	DJ LeMahieu	2618	1021	1597	0.670	0.681	1.351
4	Gio Urshela	1908	727	1181	0.761	0.556	1.317
21	Greg Bird	206	76	130	0.632	0.677	1.309
6	Brett Gardner	2544	991	1553	0.570	0.733	1.304
17	Giancarlo Stanton	395	121	274	0.612	0.686	1.298
18	Thairo Estrada	227	95	132	0.632	0.614	1.245
0	Gary Sanchez	2053	647	1406	0.604	0.632	1.237
20	Miguel Andujar	175	61	114	0.689	0.544	1.232

Team Good Strike Average: 0.699

Team Good Ball Average: 0.673





```
[35]: ##### ONLY 3-2 COUNT PITCHES
print("===== NYY 3-2 COUNT DATA_
↳===== \n")
YankAtBats32 = []
for z in range(len(YankAtBats)):
    if Count_3_2[z] == 1:
        YankAtBats32.append(YankAtBats[z])

YankAtBats32df = pd.DataFrame(YankAtBats32[0:len(YankAtBats32)-5], columns =_
↳['Name', 'xPos', 'zPos', 'Pitch Type', 'Given Description',_
↳'Classification'])
#print(YankAtBats32df, "\n")

nyyList2 = []
for yank in playerNums:
    num_pitches = 0
    num_strikes = 0
    num_balls = 0
    num_good_strikes = 0
    num_good_balls = 0
    for x in range(len(YankAtBats32df)):
        if ((YankAtBats32df.iloc[x]).iloc[0]) == yank:
            num_pitches += 1
            if ((YankAtBats32df.iloc[x]).iloc[5]) == "strike, good" or_
↳((YankAtBats32df.iloc[x]).iloc[5]) == "strike, bad":
```

```

        num_strikes += 1
        if ((YankAtBats32df.iloc[x]).iloc[5]) == "strike, good":
            num_good_strikes += 1
    else:
        num_balls += 1
        if ((YankAtBats32df.iloc[x]).iloc[5]) == "ball, good":
            num_good_balls += 1
    if num_strikes != 0 and num_balls != 0 and num_pitches >= 50: ##### min 50
    ↪pitches
        p_Good = num_good_strikes / num_strikes
        p_Bad = num_good_balls / num_balls
        nyyList2.append([yank,round(num_pitches,3) ,round(num_strikes,3),
    ↪round(num_balls,3), round(p_Good,3),round(p_Bad,3),round((p_Good +
    ↪p_Bad),3)])

YankAtBats32df = pd.DataFrame(nyyList2, columns = ['Name','Pitches', ' Strikes',
    ↪'Balls', 'g_strike', 'g_ball', 'Sum'])
print(color.BOLD + "\t\t\t Ordered by name" + color.END)
print(YankAtBats32df.sort_values(by='Name', ascending=True), "\n")
print(color.BOLD + "\t\t\t Ordered by pitches" + color.END)
sort = YankAtBats32df.sort_values(by='Pitches', ascending=False)
print(sort, "\n")
print(color.BOLD + "\t\t\t Ordered by good strikes" + color.END)
sort1 = YankAtBats32df.sort_values(by='g_strike', ascending=False)
print(sort1, "\n")
print(color.BOLD + "\t\t\t Ordered by good balls" + color.END)
sort2 = YankAtBats32df.sort_values(by='g_ball', ascending=False)
print(sort2, "\n")
print(color.BOLD + "\t\t\t Ordered by sum" + color.END)
sort3 = YankAtBats32df.sort_values(by='Sum', ascending=False)
print(sort3, "\n")
g_strike_AVG = YankAtBats32df["g_strike"].mean() ##### Gets average of whole
    ↪column
g_ball_AVG = YankAtBats32df["g_ball"].mean()
print("Team Good Strike Average: ", round(g_strike_AVG,3), "\nTeam Good Ball
    ↪Average: ", round(g_ball_AVG,3))

```

===== NYY 3-2 COUNT DATA =====

		Ordered by name					
	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
11	Aaron Hicks	72	27	45	0.852	0.533	1.385
7	Aaron Judge	130	58	72	0.879	0.625	1.504
6	Brett Gardner	137	86	51	0.953	0.333	1.287
10	Cameron Maybin	78	41	37	0.902	0.432	1.335
12	Clint Frazier	51	22	29	0.955	0.241	1.196
2	DJ LeMahieu	125	51	74	0.922	0.446	1.368

3	Didi Gregorius	69	34	35	1.000	0.314	1.314
8	Edwin Encarnacion	163	71	92	0.944	0.467	1.411
0	Gary Sanchez	126	53	73	0.943	0.288	1.231
4	Gio Urshela	65	28	37	0.964	0.432	1.397
9	Gleyber Torres	146	64	82	0.969	0.366	1.335
13	Kendrys Morales	58	28	30	0.929	0.500	1.429
1	Luke Voit	109	37	72	0.757	0.653	1.410
5	Mike Tauchman	81	33	48	0.939	0.521	1.460

Ordered by pitches

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
8	Edwin Encarnacion	163	71	92	0.944	0.467	1.411
9	Gleyber Torres	146	64	82	0.969	0.366	1.335
6	Brett Gardner	137	86	51	0.953	0.333	1.287
7	Aaron Judge	130	58	72	0.879	0.625	1.504
0	Gary Sanchez	126	53	73	0.943	0.288	1.231
2	DJ LeMahieu	125	51	74	0.922	0.446	1.368
1	Luke Voit	109	37	72	0.757	0.653	1.410
5	Mike Tauchman	81	33	48	0.939	0.521	1.460
10	Cameron Maybin	78	41	37	0.902	0.432	1.335
11	Aaron Hicks	72	27	45	0.852	0.533	1.385
3	Didi Gregorius	69	34	35	1.000	0.314	1.314
4	Gio Urshela	65	28	37	0.964	0.432	1.397
13	Kendrys Morales	58	28	30	0.929	0.500	1.429
12	Clint Frazier	51	22	29	0.955	0.241	1.196

Ordered by good strikes

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
3	Didi Gregorius	69	34	35	1.000	0.314	1.314
9	Gleyber Torres	146	64	82	0.969	0.366	1.335
4	Gio Urshela	65	28	37	0.964	0.432	1.397
12	Clint Frazier	51	22	29	0.955	0.241	1.196
6	Brett Gardner	137	86	51	0.953	0.333	1.287
8	Edwin Encarnacion	163	71	92	0.944	0.467	1.411
0	Gary Sanchez	126	53	73	0.943	0.288	1.231
5	Mike Tauchman	81	33	48	0.939	0.521	1.460
13	Kendrys Morales	58	28	30	0.929	0.500	1.429
2	DJ LeMahieu	125	51	74	0.922	0.446	1.368
10	Cameron Maybin	78	41	37	0.902	0.432	1.335
7	Aaron Judge	130	58	72	0.879	0.625	1.504
11	Aaron Hicks	72	27	45	0.852	0.533	1.385
1	Luke Voit	109	37	72	0.757	0.653	1.410

Ordered by good balls

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
1	Luke Voit	109	37	72	0.757	0.653	1.410
7	Aaron Judge	130	58	72	0.879	0.625	1.504
11	Aaron Hicks	72	27	45	0.852	0.533	1.385

5	Mike Tauchman	81	33	48	0.939	0.521	1.460
13	Kendrys Morales	58	28	30	0.929	0.500	1.429
8	Edwin Encarnacion	163	71	92	0.944	0.467	1.411
2	DJ LeMahieu	125	51	74	0.922	0.446	1.368
4	Gio Urshela	65	28	37	0.964	0.432	1.397
10	Cameron Maybin	78	41	37	0.902	0.432	1.335
9	Gleyber Torres	146	64	82	0.969	0.366	1.335
6	Brett Gardner	137	86	51	0.953	0.333	1.287
3	Didi Gregorius	69	34	35	1.000	0.314	1.314
0	Gary Sanchez	126	53	73	0.943	0.288	1.231
12	Clint Frazier	51	22	29	0.955	0.241	1.196

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
7	Aaron Judge	130	58	72	0.879	0.625	1.504
5	Mike Tauchman	81	33	48	0.939	0.521	1.460
13	Kendrys Morales	58	28	30	0.929	0.500	1.429
8	Edwin Encarnacion	163	71	92	0.944	0.467	1.411
1	Luke Voit	109	37	72	0.757	0.653	1.410
4	Gio Urshela	65	28	37	0.964	0.432	1.397
11	Aaron Hicks	72	27	45	0.852	0.533	1.385
2	DJ LeMahieu	125	51	74	0.922	0.446	1.368
9	Gleyber Torres	146	64	82	0.969	0.366	1.335
10	Cameron Maybin	78	41	37	0.902	0.432	1.335
3	Didi Gregorius	69	34	35	1.000	0.314	1.314
6	Brett Gardner	137	86	51	0.953	0.333	1.287
0	Gary Sanchez	126	53	73	0.943	0.288	1.231
12	Clint Frazier	51	22	29	0.955	0.241	1.196

Team Good Strike Average: 0.922

Team Good Ball Average: 0.439

```
[22]: ##### ONLY FAST BALLS (2 and 4 seam)
print("===== NYY FASTBALL DATA_
↳===== \n")
nyyList3 = []
for yank in playerNums:
    num_pitches = 0
    num_strikes = 0
    num_balls = 0
    num_good_strikes = 0
    num_good_balls = 0
    for x in range(len(newdf)):
        if ((newdf.iloc[x]).iloc[0]) == yank and (((newdf.iloc[x]).iloc[3]) ==_
↳"FT" or ((newdf.iloc[x]).iloc[3]) == "FF"):
            num_pitches += 1
```

```

        if ((newdf.iloc[x]).iloc[5]) == "strike, good" or ((newdf.iloc[x]).
→iloc[5]) == "strike, bad":
            num_strikes += 1
            if ((newdf.iloc[x]).iloc[5]) == "strike, good":
                num_good_strikes += 1
        else:
            num_balls += 1
            if ((newdf.iloc[x]).iloc[5]) == "ball, good":
                num_good_balls += 1
        if num_strikes != 0 and num_balls != 0 and num_pitches >= 100: #### Only
→using 100 here
            p_Good = num_good_strikes / num_strikes
            p_Bad = num_good_balls / num_balls
            nyyList3.append([yank,round(num_pitches,3) ,round(num_strikes,3),
→round(num_balls,3), round(p_Good,3),round(p_Bad,3),round((p_Good +
→p_Bad),3)])

FastBallsDF = pd.DataFrame(nyyList3, columns = ['Name','Pitches', 'Strikes',
→'Balls', 'g_strike', 'g_ball', 'Sum'])

print(color.BOLD + "\t\t\t Ordered by name" + color.END)
print(FastBallsDF.sort_values(by='Name', ascending=True), "\n")
print(color.BOLD + "\t\t\t Ordered by pitches" + color.END)
sort = FastBallsDF.sort_values(by='Pitches', ascending=False)
print(sort, "\n")
print(color.BOLD + "\t\t\t Ordered by good strikes" + color.END)
sort1 = FastBallsDF.sort_values(by='g_strike', ascending=False)
print(sort1, "\n")
print(color.BOLD + "\t\t\t Ordered by good balls" + color.END)
sort2 = FastBallsDF.sort_values(by='g_ball', ascending=False)
print(sort2, "\n")
print(color.BOLD + "\t\t\t Ordered by sum" + color.END)
sort3 = FastBallsDF.sort_values(by='Sum', ascending=False)
print(sort3, "\n")
g_strike_AVG = FastBallsDF["g_strike"].mean() #### Gets average of whole column
g_ball_AVG = FastBallsDF["g_ball"].mean()
print("Team Good Strike Average: ", round(g_strike_AVG,3), "\nTeam Good Ball
→Average: ", round(g_ball_AVG,3))

```

===== NYY FASTBALL DATA
=====

		Ordered by name					
	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
11	Aaron Hicks	471	188	283	0.654	0.749	1.403
7	Aaron Judge	800	341	459	0.707	0.734	1.441
13	Austin Romine	352	151	201	0.834	0.672	1.506

6	Brett Gardner	1218	518	700	0.573	0.754	1.328
10	Cameron Maybin	459	201	258	0.726	0.752	1.478
12	Clint Frazier	424	178	246	0.669	0.793	1.461
2	DJ LeMahieu	1278	563	715	0.631	0.737	1.368
3	Didi Gregorius	628	256	372	0.734	0.548	1.283
8	Edwin Encarnacion	837	344	493	0.680	0.680	1.360
0	Gary Sanchez	788	301	487	0.671	0.639	1.310
17	Giancarlo Stanton	179	68	111	0.588	0.703	1.291
4	Gio Urshela	771	340	431	0.776	0.573	1.350
9	Gleyber Torres	1014	428	586	0.820	0.662	1.482
19	Greg Bird	102	42	60	0.524	0.700	1.224
16	Kendrys Morales	334	145	189	0.607	0.767	1.374
18	Kyle Higashioka	100	47	53	0.830	0.509	1.339
1	Luke Voit	810	336	474	0.821	0.762	1.583
14	Mike Ford	242	101	141	0.723	0.702	1.425
5	Mike Tauchman	612	292	320	0.740	0.825	1.565
15	Tyler Wade	166	68	98	0.721	0.786	1.506

Ordered by pitches

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
2	DJ LeMahieu	1278	563	715	0.631	0.737	1.368
6	Brett Gardner	1218	518	700	0.573	0.754	1.328
9	Gleyber Torres	1014	428	586	0.820	0.662	1.482
8	Edwin Encarnacion	837	344	493	0.680	0.680	1.360
1	Luke Voit	810	336	474	0.821	0.762	1.583
7	Aaron Judge	800	341	459	0.707	0.734	1.441
0	Gary Sanchez	788	301	487	0.671	0.639	1.310
4	Gio Urshela	771	340	431	0.776	0.573	1.350
3	Didi Gregorius	628	256	372	0.734	0.548	1.283
5	Mike Tauchman	612	292	320	0.740	0.825	1.565
11	Aaron Hicks	471	188	283	0.654	0.749	1.403
10	Cameron Maybin	459	201	258	0.726	0.752	1.478
12	Clint Frazier	424	178	246	0.669	0.793	1.461
13	Austin Romine	352	151	201	0.834	0.672	1.506
16	Kendrys Morales	334	145	189	0.607	0.767	1.374
14	Mike Ford	242	101	141	0.723	0.702	1.425
17	Giancarlo Stanton	179	68	111	0.588	0.703	1.291
15	Tyler Wade	166	68	98	0.721	0.786	1.506
19	Greg Bird	102	42	60	0.524	0.700	1.224
18	Kyle Higashioka	100	47	53	0.830	0.509	1.339

Ordered by good strikes

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
13	Austin Romine	352	151	201	0.834	0.672	1.506
18	Kyle Higashioka	100	47	53	0.830	0.509	1.339
1	Luke Voit	810	336	474	0.821	0.762	1.583
9	Gleyber Torres	1014	428	586	0.820	0.662	1.482
4	Gio Urshela	771	340	431	0.776	0.573	1.350

5	Mike Tauchman	612	292	320	0.740	0.825	1.565
3	Didi Gregorius	628	256	372	0.734	0.548	1.283
10	Cameron Maybin	459	201	258	0.726	0.752	1.478
14	Mike Ford	242	101	141	0.723	0.702	1.425
15	Tyler Wade	166	68	98	0.721	0.786	1.506
7	Aaron Judge	800	341	459	0.707	0.734	1.441
8	Edwin Encarnacion	837	344	493	0.680	0.680	1.360
0	Gary Sanchez	788	301	487	0.671	0.639	1.310
12	Clint Frazier	424	178	246	0.669	0.793	1.461
11	Aaron Hicks	471	188	283	0.654	0.749	1.403
2	DJ LeMahieu	1278	563	715	0.631	0.737	1.368
16	Kendrys Morales	334	145	189	0.607	0.767	1.374
17	Giancarlo Stanton	179	68	111	0.588	0.703	1.291
6	Brett Gardner	1218	518	700	0.573	0.754	1.328
19	Greg Bird	102	42	60	0.524	0.700	1.224

Ordered by good balls

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
5	Mike Tauchman	612	292	320	0.740	0.825	1.565
12	Clint Frazier	424	178	246	0.669	0.793	1.461
15	Tyler Wade	166	68	98	0.721	0.786	1.506
16	Kendrys Morales	334	145	189	0.607	0.767	1.374
1	Luke Voit	810	336	474	0.821	0.762	1.583
6	Brett Gardner	1218	518	700	0.573	0.754	1.328
10	Cameron Maybin	459	201	258	0.726	0.752	1.478
11	Aaron Hicks	471	188	283	0.654	0.749	1.403
2	DJ LeMahieu	1278	563	715	0.631	0.737	1.368
7	Aaron Judge	800	341	459	0.707	0.734	1.441
17	Giancarlo Stanton	179	68	111	0.588	0.703	1.291
14	Mike Ford	242	101	141	0.723	0.702	1.425
19	Greg Bird	102	42	60	0.524	0.700	1.224
8	Edwin Encarnacion	837	344	493	0.680	0.680	1.360
13	Austin Romine	352	151	201	0.834	0.672	1.506
9	Gleyber Torres	1014	428	586	0.820	0.662	1.482
0	Gary Sanchez	788	301	487	0.671	0.639	1.310
4	Gio Urshela	771	340	431	0.776	0.573	1.350
3	Didi Gregorius	628	256	372	0.734	0.548	1.283
18	Kyle Higashioka	100	47	53	0.830	0.509	1.339

Ordered by sum

	Name	Pitches	Strikes	Balls	g_strike	g_ball	Sum
1	Luke Voit	810	336	474	0.821	0.762	1.583
5	Mike Tauchman	612	292	320	0.740	0.825	1.565
13	Austin Romine	352	151	201	0.834	0.672	1.506
15	Tyler Wade	166	68	98	0.721	0.786	1.506
9	Gleyber Torres	1014	428	586	0.820	0.662	1.482
10	Cameron Maybin	459	201	258	0.726	0.752	1.478
12	Clint Frazier	424	178	246	0.669	0.793	1.461

7	Aaron Judge	800	341	459	0.707	0.734	1.441
14	Mike Ford	242	101	141	0.723	0.702	1.425
11	Aaron Hicks	471	188	283	0.654	0.749	1.403
16	Kendrys Morales	334	145	189	0.607	0.767	1.374
2	DJ LeMahieu	1278	563	715	0.631	0.737	1.368
8	Edwin Encarnacion	837	344	493	0.680	0.680	1.360
4	Gio Urshela	771	340	431	0.776	0.573	1.350
18	Kyle Higashioka	100	47	53	0.830	0.509	1.339
6	Brett Gardner	1218	518	700	0.573	0.754	1.328
0	Gary Sanchez	788	301	487	0.671	0.639	1.310
17	Giancarlo Stanton	179	68	111	0.588	0.703	1.291
3	Didi Gregorius	628	256	372	0.734	0.548	1.283
19	Greg Bird	102	42	60	0.524	0.700	1.224

Team Good Strike Average: 0.701

Team Good Ball Average: 0.702

```
[ ]: # z strike zone is 1.75 to 3.42 feet
# x strike zone is -0.71 to 0.71 feet
# top: 3.34, bottom: 1.57 ---> (-0.79, 1.57), 1, 1.77
# bottom left corner ->(x,y), width, height
# https://www.baseballprospectus.com/news/article/14098/
  ↳ spinning-yarn-the-real-strike-zone-part-2/
# https://www.geeksforgeeks.org/different-ways-to-create-pandas-dataframe/
# https://stackoverflow.com/questions/17812978/
  ↳ how-to-plot-two-columns-of-a-pandas-data-frame-using-points
# https://www.statology.org/matplotlib-rectangle/
# https://stackoverflow.com/questions/57246963/
  ↳ why-isnt-the-legend-in-matplotlib-correctly-displaying-the-colors
# https://pythonexamples.org/pandas-dataframe-sort-by-column/#2
# https://stackoverflow.com/questions/8924173/
  ↳ how-do-i-print-bold-text-in-python/8930747
# https://www.geeksforgeeks.org/change-figure-size-in-pandas-python/
# IMPORTANT: Cell -> Current Outputs -> Toggle Scrolling
```