build_tables

March 28, 2020

[88]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Building the Tables

We've done the major preprocessing in other scripts, and now it's time to get our final tables together for fielders, catchers and pitchers with all appropriate stats.

[89]:
```
df_bat = pd.read_csv('../core/output/batting_pre.csv')
df_field = pd.read_csv('../core/output/fielding_pre.csv')
df_catch = pd.read_csv('../core/output/catching_pre.csv')
df_pitch = pd.read_csv('../core/output/pitching_pre.csv')
df_meta = pd.read_csv('../core/output/metadata.csv')
```

[90]:
```
df_meta.head()
```

[90]:

	retroID	POS	birthYear	bats	throws	weight	height	debutYear	finalYear
0	aardd001	P	1981.0	R	R	215.0	75.0	2004	2015
1	aaroh101	OF	1934.0	R	R	180.0	72.0	1954	1976
2	aarot101	1B	1939.0	R	R	190.0	75.0	1962	1971
3	aased001	P	1954.0	R	R	190.0	75.0	1977	1990
4	abada001	1B	1972.0	L	L	184.0	73.0	2001	2006

Making Metadata Usable

We are interested in all of these fields, so we want to convert POS, bats and throws to numbers and use dummy variables. Note that we won't be using birthYear as-is, but rather subtracting it from current year to get a player's age for a season. This won't matter for the player career stats tensor so we can drop it here.

[91]:
```
df_meta.drop(columns=['birthYear'], inplace=True)
```

[92]:
```
df_meta_pos = pd.get_dummies(df_meta['POS'], prefix='pos')
df_meta_bats = pd.get_dummies(df_meta['bats'], drop_first=True, prefix='bats')
df_meta_throws = pd.get_dummies(df_meta['throws'], prefix='throws')
```

[93]:
```
dropped_meta_cols = ['POS', 'bats', 'throws']
df_meta.drop(columns=dropped_meta_cols, inplace=True)
```

```
df_meta.head()
```

[93]:

	retroID	weight	height	debutYear	finalYear
0	aardd001	215.0	75.0	2004	2015
1	aaroh101	180.0	72.0	1954	1976
2	aarot101	190.0	75.0	1962	1971
3	aased001	190.0	75.0	1977	1990
4	abada001	184.0	73.0	2001	2006

[94]:
```
df_meta = df_meta.join([df_meta_pos, df_meta_bats, df_meta_throws])
df_meta
```

[94]:

	retroID	weight	height	debutYear	finalYear	pos_1B	pos_2B	pos_3B	pos_C	pos_OF	pos_P	pos_SS	bats_L	bats_R	throws_L	throws_R	throws_S
0	aardd001	215.0	75.0	2004	2015	0	0	0	0	0	1	0	0	1	0	1	0
1	aaroh101	180.0	72.0	1954	1976	0	0	0	0	1	0	0	0	1	0	1	0
2	aarot101	190.0	75.0	1962	1971	1	0	0	0	0	0	0	0	1	0	1	0
3	aased001	190.0	75.0	1977	1990	0	0	0	0	0	1	0	0	1	0	1	0
4	abada001	184.0	73.0	2001	2006	1	0	0	0	0	0	0	1	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15026	zupcb001	220.0	76.0	1991	1994	0	0	0	0	1	0	0	0	1	0	1	0
15027	zupof101	182.0	71.0	1957	1961	0	0	0	1	0	0	0	1	0	1	0	0
15028	zuveg101	195.0	76.0	1951	1959	0	0	0	0	0	1	0	0	1	0	1	0
15029	zuvep001	173.0	72.0	1982	1991	0	0	0	0	0	0	1	0	1	0	1	0
15030	zycht001	190.0	75.0	2015	2017	0	0	0	0	0	1	0	0	1	0	1	0

```
15028         0
15029         0
15030         0

[15031 rows x 17 columns]
```

I didn't drop_first on the 'throws_' columns because I want to get rid of 'throws_S' instead of 'throws_L'

[95]:
```
df_meta.drop(columns=['throws_S'], inplace=True)
```

We want to use weight and height but we can normalize them

[96]:
```
from sklearn.preprocessing import MinMaxScaler
```

[97]:
```
scaler = MinMaxScaler()
```

[98]:
```
df_meta[['weight', 'height']] = scaler.fit_transform(df_meta[['weight',␣
↪'height']])
df_meta
```

[98]:

	retroID	weight	height	debutYear	finalYear	pos_1B	pos_2B	pos_3B	pos_C	pos_OF	pos_P	pos_SS	bats_L	bats_R	throws_L
0	aardd001	0.569672	0.60	2004	2015	0	0	0	0	0	1	0	0	1	0
1	aaroh101	0.426230	0.45	1954	1976	0	0	0	0	1	0	0	0	1	0
2	aarot101	0.467213	0.60	1962	1971	1	0	0	0	0	0	0	0	1	0
3	aased001	0.467213	0.60	1977	1990	0	0	0	0	0	1	0	0	1	0
4	abada001	0.442623	0.50	2001	2006	1	0	0	0	0	0	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15026	zupcb001	0.590164	0.65	1991	1994	0	0	0	0	1	0	0	0	1	0
15027	zupof101	0.434426	0.40	1957	1961	0	0	0	1	0	0	0	1	0	0
15028	zuveg101	0.487705	0.65	1951	1959	0	0	0	0	0	1	0	0	1	0
15029	zuvep001	0.397541	0.45	1982	1991	0	0	0	0	0	0	1	0	1	0
15030	zycht001	0.467213	0.60	2015	2017	0	0	0	0	0	1	0	0	1	0

```
       throws_R
0             1
1             1
2             1
3             1
4             0
...         ...
15026         1
15027         1
15028         1
15029         1
15030         1

[15031 rows x 16 columns]
```

The metadata is now ready to go into the final tensor.

Combining Batting Data

[99]:
```
df_bat
```

[99]:
```
        retroID     G     AB     R     H   2B  3B   HR   RBI   SB    CS    BB  \
0      aardd001   331      4     0     0    0   0    0     0    0   0.0     0
1      aaroh101  3298  12364  2174  3771  624  98  755  2297  240  73.0  1402
2      aarot101   437    944   102   216   42   6   13    94    9   8.0    86
3      aased001   448      5     0     0    0   0    0     0    0   0.0     0
4      abada001    15     21     1     2    0   0    0     0    0   1.0     4
...         ...   ...    ...   ...   ...  ...  ..  ...   ...  ...   ...   ...
15187  zupcb001   319    795    99   199   47   4    7    80    7   5.0    57
15188  zupof101    16     18     3     3    1   0    0     0    0   0.0     2
15189  zuveg101   266    142     5    21    2   1    0     7    0   1.0     9
15190  zuvep001   209    491    41   109   17   2    2    20    2   0.0    34
15191  zycht001    70      0     0     0    0   0    0     0    0   0.0     0

         SO  IBB  HBP  SH   SF  GIDP  NL
0         2    0    0   1    0     0   1
1      1383  293   32  21  121   328   1
2       145    3    0   9    6    36   1
3         3    0    0   0    0     0   1
4         5    0    0   0    0     1   1
...     ...  ...  ...  ..  ...   ...  ..
15187   137    3    6  20    8    15   0
15188     6    0    0   0    0     0   0
15189    39    0    0  16    0     3   1
15190    50    1    2  18    0     8   1
15191     0    0    0   0    0     0   0

[15192 rows x 19 columns]
```

[100]:
```
df = pd.merge(df_meta, df_bat, how='inner', on=['retroID'])
```

[187]:
```
df.shape
```

[187]:
```
(15031, 34)
```

[101]:
```
df.head(10)
```

[101]:

	retroID	weight	height	debutYear	finalYear	pos_1B	pos_2B	pos_3B	\
0	aardd001	0.569672	0.60	2004	2015	0	0	0	
1	aaroh101	0.426230	0.45	1954	1976	0	0	0	
2	aarot101	0.467213	0.60	1962	1971	1	0	0	
3	aased001	0.467213	0.60	1977	1990	0	0	0	
4	abada001	0.442623	0.50	2001	2006	1	0	0	
5	abadf001	0.590164	0.50	2010	2017	0	0	0	
6	abbog001	0.508197	0.75	1973	1984	0	0	0	
7	abboj001	0.508197	0.60	1989	1999	0	0	0	
8	abboj002	0.467213	0.55	1997	2001	0	0	0	
9	abbok001	0.508197	0.65	1991	1996	0	0	0	

	pos_C	pos_OF	...	SB	CS	BB	SO	IBB	HBP	SH	SF	GIDP	NL
0	0	0	...	0	0.0	0	2	0	0	1	0	0	1
1	0	1	...	240	73.0	1402	1383	293	32	21	121	328	1
2	0	0	...	9	8.0	86	145	3	0	9	6	36	1
3	0	0	...	0	0.0	0	3	0	0	0	0	0	1
4	0	0	...	0	1.0	4	5	0	0	0	0	1	1
5	0	0	...	0	0.0	0	5	0	0	0	0	1	1
6	0	0	...	0	0.0	0	0	0	0	0	0	0	0
7	0	0	...	0	0.0	0	10	0	0	3	0	0	1
8	0	1	...	6	5.0	38	91	2	3	5	7	12	1
9	0	0	...	0	0.0	1	19	0	0	6	0	0	1

[10 rows x 34 columns]

We noticed that df_bat and df_meta don't have the same number of rows, so we want to find out what's going on there.

[102]:
```
df_bat[~df_bat['retroID'].isin(df_meta['retroID'])]
```

[102]:

	retroID	G	AB	R	H	2B	3B	HR	RBI	SB	CS	BB	SO	IBB	HBP	SH	\
121	albeb101	6	18	1	5	1	0	0	0	0	0.0	0	2	0	0	0	
358	aragj101	1	0	0	0	0	0	0	0	0	0.0	0	0	0	0	0	
445	atkil101	1	1	1	0	0	0	0	0	0	0.0	0	0	0	0	0	
611	banij001	1	1	0	1	0	0	0	0	0	0.0	0	0	0	0	0	
633	barbr101	1	1	0	0	0	0	0	0	0	0.0	0	0	0	0	0	
...	...	..	..	..	..	..	..	..	..	...	..	...	..	..	...	...	..	
14543	westj101	1	1	0	0	0	0	0	0	0	0.0	0	1	0	0	0	

```
14730  willh101  10   9  0  2   0   0   0    0   0  0.0   1   4    0    0   0
14811  wilsi101   1   1  0  0   0   0   0    0   0  0.0   0   0    0    0   0
15009  wrigr002   1   3  0  0   0   0   0    0   0  0.0   0   1    0    0   0
15050  yeabb101   3   0  0  0   0   0   0    0   0  0.0   1   0    0    0   0

       SF  GIDP  NL
121     0     0   0
358     0     0   1
445     0     0   0
611     0     0   1
633     0     0   0
...    ..   ...  ..
14543   0     0   1
14730   0     0   1
14811   0     0   0
15009   0     1   0
15050   0     0   1

[161 rows x 19 columns]
```

[103]:
```
df_bat[~df_bat['retroID'].isin(df_meta['retroID'])]['G'].max()
```

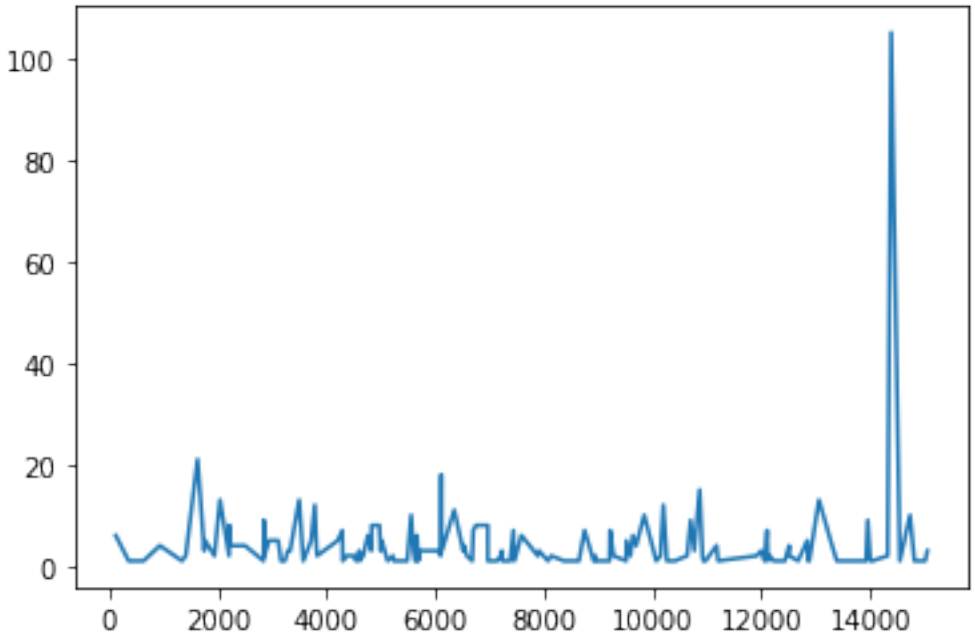
[103]:
```
105
```

[104]:
```
df_bat[~df_bat['retroID'].isin(df_meta['retroID'])]['G'].mean()
```

[104]:
```
4.105590062111801
```

[105]:
```
plt.plot(df_bat[~df_bat['retroID'].isin(df_meta['retroID'])]['G'])
```

[105]:
```
[<matplotlib.lines.Line2D at 0x11ffb5bd0>]
```



For the most part, we're talking about players who have played under 20 total games. We can easily drop these data points and not really affect the overall result.

Combining the Tensors

Catchers

[106]:
```
df_catch.shape[0] + df_field.shape[0] + df_pitch.shape[0]
```

[106]: 23591

[107]:
```
df_catch
```

[107]:

	retroID	GS	InnOuts	PO	A	E	DP	PB	WP	SB	CS	ZR
0	adamb105	1	27	6	0	0	0	0	0	1	0	0
1	adamb106	0	0	249	90	12	15	7	0	0	0	0
2	adamd101	3	78	9	2	0	0	1	0	0	0	0
3	adled101	65	1840	453	26	4	2	8	19	37	16	0
4	afent001	20	613	123	5	1	3	6	0	17	3	0
...	...	...	...	...	...	..	..	..	..	...	..	..
1524	zimmd101	27	744	150	18	6	1	5	12	10	10	3
1525	zimmj101	298	8560	2131	150	21	26	19	84	110	80	4
1526	zinta001	0	3	2	0	0	0	0	0	0	0	0
1527	zunim001	535	14489	4356	264	21	22	39	0	248	98	0
1528	zupof101	1	114	31	1	2	0	1	1	2	1	0

[1529 rows x 12 columns]

[108]:
```
np.intersect1d(df_catch.columns, df.columns)
```

[108]:
```
array(['CS', 'SB', 'retroID'], dtype=object)
```

The 'caught stealing' and 'stolen bases' stats appear both offensively and defensively (CS/SB against) for catchers. We need to keep them separate when merging the metadata and we can do so by just adding a prefix to the defensive stats.

[109]:
```
df_catch.rename(columns={'CS': 'CS_A', 'SB': 'SB_A'}, inplace=True)
```

[110]:
```
catchers = pd.merge(df_catch, df, how='inner', on=['retroID'])
```

[111]:
```
catchers
```

[111]:

	retroID	GS	InnOuts	PO	A	E	DP	PB	WP	SB_A	...	SB	CS	\
0	adamb105	1	27	6	0	0	0	0	0	1	...	0	0.0	
1	adamb106	0	0	249	90	12	15	7	0	0	...	4	2.0	
2	adamd101	3	78	9	2	0	0	1	0	0	...	0	0.0	
3	adled101	65	1840	453	26	4	2	8	19	37	...	0	0.0	
4	afent001	20	613	123	5	1	3	6	0	17	...	0	0.0	
...	...	...	...	...	...	..	..	..	..	...	...	..	...	
1524	zimmd101	27	744	150	18	6	1	5	12	10	...	45	25.0	
1525	zimmj101	298	8560	2131	150	21	26	19	84	110	...	1	2.0	
1526	zinta001	0	3	2	0	0	0	0	0	0	...	0	0.0	
1527	zunim001	535	14489	4356	264	21	22	39	0	248	...	2	4.0	
1528	zupof101	1	114	31	1	2	0	1	1	2	...	0	0.0	

	BB	SO	IBB	HBP	SH	SF	GIDP	NL
0	0	5	0	0	0	0	0	0
1	6	27	0	0	3	0	0	1
2	1	3	0	0	0	0	1	0
3	18	80	5	2	2	1	9	1
4	5	32	0	0	1	1	1	1
...	...	...	...	...	..	..	...	..
1524	246	678	27	13	37	14	99	1
1525	78	154	12	11	31	4	38	1
1526	5	34	0	0	0	1	0	1
1527	138	714	1	45	8	11	38	0
1528	2	6	0	0	0	0	0	0

[1529 rows x 45 columns]

[112]:
```
catchers.columns
```

[112]:
```
Index(['retroID', 'GS', 'InnOuts', 'PO', 'A', 'E', 'DP', 'PB', 'WP', 'SB_A',
       'CS_A', 'ZR', 'weight', 'height', 'debutYear', 'finalYear', 'pos_1B',
       'pos_2B', 'pos_3B', 'pos_C', 'pos_OF', 'pos_P', 'pos_SS', 'bats_L',
```

```
       'bats_R', 'throws_L', 'throws_R', 'G', 'AB', 'R', 'H', '2B', '3B', 'HR',
       'RBI', 'SB', 'CS', 'BB', 'SO', 'IBB', 'HBP', 'SH', 'SF', 'GIDP', 'NL'],
      dtype='object')
```

There's no reason to waste columns on position for the catchers.

[113]:
```
catchers.drop(columns=['pos_1B', 'pos_2B', 'pos_3B', 'pos_C',
       'pos_OF', 'pos_P', 'pos_SS'], inplace=True)
```

[114]:
```
catchers.columns
```

[114]:
```
Index(['retroID', 'GS', 'InnOuts', 'PO', 'A', 'E', 'DP', 'PB', 'WP', 'SB_A',
       'CS_A', 'ZR', 'weight', 'height', 'debutYear', 'finalYear', 'bats_L',
       'bats_R', 'throws_L', 'throws_R', 'G', 'AB', 'R', 'H', '2B', '3B', 'HR',
       'RBI', 'SB', 'CS', 'BB', 'SO', 'IBB', 'HBP', 'SH', 'SF', 'GIDP', 'NL'],
      dtype='object')
```

Pitchers

[147]:
```
np.intersect1d(df_pitch.columns, df.columns)
```

[147]:
```
array(['G', 'retroID'], dtype=object)
```

We have quite a few common columns for pitching data and metadata. We'll do what we did for catching and just add 'A' to the end (for 'against'). Since there are quite a few, we'll define a conversion dictionary ahead of time. Before we do that, we see that we can drop the 'G' (games) column as it should be the same between the tables. We can also drop the position information.

[148]:
```
batting_data_for_pitchers = df.drop(columns=['G', 'pos_1B', 'pos_2B',
                     'pos_3B', 'pos_C', 'pos_OF', 'pos_P', 'pos_SS'])
```

[149]:
```
pitching_data_conversion_dict = {
    'BB': 'BB_A',
    'GIDP': 'GIDP_A',
    'H': 'H_A',
    'HBP': 'HBP_A',
    'HR': 'HR_A',
    'IBB': 'IBB_A',
    'R': 'R_A',
    'SF': 'SF_A',
    'SH': 'SH_A',
    'SO': 'SO_A'
}
```

[150]:
```
df_pitch.rename(columns=pitching_data_conversion_dict, inplace=True)
```

[151]:
```
pitchers = pd.merge(df_pitch, batting_data_for_pitchers, how='inner',␣
 ↪on=['retroID'])
```

[152]:
```
pitchers
```

[152]:

	retroID	BAOpp	ERA	W	L	G	GS	CG	SHO	SV	...	SB	CS	\
0	aardd001	0.2574	5.1944	16	18	331	0	0	0	69	...	0	0.0	
1	aased001	0.2508	3.4931	66	60	448	91	22	5	82	...	0	0.0	
2	abadf001	0.2501	4.0733	8	27	363	6	0	0	2	...	0	0.0	
3	abbog001	0.2786	4.3317	62	83	248	206	37	5	0	...	0	0.0	
4	abboj001	0.2804	4.4964	87	108	263	254	31	6	0	...	0	0.0	
...	...	...	...	..	...	...	...	..	...	..	...	..	...	
7830	zolds101	0.2700	3.6890	43	53	250	93	30	5	8	...	1	0.0	
7831	zubeb101	0.2717	5.3617	43	42	224	65	23	3	6	...	0	0.0	
7832	zumaj001	0.2286	3.4420	13	12	171	0	0	0	5	...	0	0.0	
7833	zuveg101	0.2760	4.1280	32	36	265	31	9	2	40	...	0	1.0	
7834	zycht001	0.2183	2.8000	7	3	70	1	0	0	1	...	0	0.0	

	BB	SO	IBB	HBP	SH	SF	GIDP	NL
0	0	2	0	0	1	0	0	1
1	0	3	0	0	0	0	0	1
2	0	5	0	0	0	0	1	1
3	0	0	0	0	0	0	0	0
4	0	10	0	0	3	0	0	1
...	..	..	...	...	..	..	...	..
7830	10	52	0	1	9	0	4	0
7831	10	66	0	0	20	0	8	0
7832	0	0	0	0	0	0	0	0
7833	9	39	0	0	16	0	3	1
7834	0	0	0	0	0	0	0	0

[7835 rows x 51 columns]

Fielders

[32]:
```
np.intersect1d(df_field.columns, df.columns)
```

[32]:
```
array(['retroID'], dtype=object)
```

We don't need to worry about common columns between the general fielding stats and metadata.

[136]:
```
fielders = pd.merge(df_field, df, how='inner', on=['retroID'])
```

[137]:
```
fielders = fielders[~fielders['retroID'].isin(pitchers['retroID'])]
```

[138]:
```
fielders
```

[138]:

	retroID	GS	InnOuts	PO	A	E	DP	weight	height	\
1	aaroh101	2977	78414	7436	429	144	218	0.426230	0.45	
2	aarot101	206	6472	1317	113	22	124	0.467213	0.60	
4	abada001	4	138	37	1	1	3	0.442623	0.50	

```
8       abboj002   140     3688   299     2    8    0  0.467213    0.55
10      abbok002   504    13474   938  1262   79  275  0.426230    0.40
...          ...   ...      ...   ...   ...  ...  ...       ...     ...
14218   zoske001     8      404    16    42    2    8  0.405738    0.45
14220   zubej001    26      702   167    12    2   11  0.467213    0.50
14221   zulej001    36     1019   296    15    5   20  0.631148    0.75
14223   zupcb001   198     5842   483    22   12    5  0.590164    0.65
14225   zuvep001   136     3844   267   415   23   84  0.397541    0.45

        debutYear  ...   SB    CS    BB    SO  IBB  HBP  SH   SF  GIDP  NL
1            1954  ...  240  73.0  1402  1383  293   32  21  121   328   1
2            1962  ...    9   8.0    86   145    3    0   9    6    36   1
4            2001  ...    0   1.0     4     5    0    0   0    0     1   1
8            1997  ...    6   5.0    38    91    2    3   5    7    12   1
10           1993  ...   22  11.0   133   571   11   17  21   12    37   1
...           ...  ...  ...   ...   ...   ...  ...  ...  ..  ...   ...  ..
14218        1991  ...    0   0.0     1    13    0    0   1    1     1   1
14220        1996  ...    1   0.0    12    20    1    1   1    1     4   1
14221        2000  ...    0   2.0    10    51    1    6   0    1     5   1
14223        1991  ...    7   5.0    57   137    3    6  20    8    15   0
14225        1982  ...    2   0.0    34    50    1    2  18    0     8   1

[6392 rows x 40 columns]
```

[140]:
```
catchers
```

[140]:
```
        retroID   GS  InnOuts    PO    A   E  DP  PB  WP  SB_A  ...  SB    CS  \
0     adamb105    1       27     6    0   0   0   0   0     1  ...   0   0.0
1     adamb106    0        0   249   90  12  15   7   0     0  ...   4   2.0
2     adamd101    3       78     9    2   0   0   1   0     0  ...   0   0.0
3     adled101   65     1840   453   26   4   2   8  19    37  ...   0   0.0
4     afent001   20      613   123    5   1   3   6   0    17  ...   0   0.0
...        ...  ...      ...   ...  ...  ..  ..  ..  ..   ...  ...  ..   ...
1524  zimmd101   27      744   150   18   6   1   5  12    10  ...  45  25.0
1525  zimmj101  298     8560  2131  150  21  26  19  84   110  ...   1   2.0
1526  zinta001    0        3     2    0   0   0   0   0     0  ...   0   0.0
1527  zunim001  535    14489  4356  264  21  22  39   0   248  ...   2   4.0
1528  zupof101    1      114    31    1   2   0   1   1     2  ...   0   0.0

        BB   SO  IBB  HBP  SH  SF  GIDP  NL
0        0    5    0    0   0   0     0   0
1        6   27    0    0   3   0     0   1
2        1    3    0    0   0   0     1   0
3       18   80    5    2   2   1     9   1
4        5   32    0    0   1   1     1   1
...    ...  ...  ...  ...  ..  ..   ...  ..
1524   246  678   27   13  37  14    99   1
```

```
1525   78  154   12   11  31   4    38   1
1526    5   34    0    0   0   1     0   1
1527  138  714    1   45   8  11    38   0
1528    2    6    0    0   0   0     0   0

[1529 rows x 38 columns]
```

[169]:
```
fielders[fielders['retroID'].isin(catchers['retroID'])]
```

[169]:
```
        retroID   GS  InnOuts     PO     A    E    DP    weight  height  \
54     adamb105    2       54     20     1    0     1  0.508197    0.55
55     adamb106    0        0      0     0    0     0  0.446721    0.50
108    ainse101    0        0     11     0    0     0  0.426230    0.40
144    alexg101   22      500     83     6    3     4  0.487705    0.55
151    alfaj002    1       31      8     2    0     1  0.610656    0.55
...         ...  ...      ...    ...   ...  ...   ...       ...     ...
14111  yorkr101    0        0  11425  1030  136  1077  0.545082    0.50
14139  younj001  843    23486   1679   756  115   113  0.426230    0.45
14183  zaung001    0       33      3     2    0     1  0.385246    0.35
14199  zimmd101  813    21993   1491  2204  150   417  0.364754    0.30
14210  zinta001    5      198     65     5    1     4  0.508197    0.55

       debutYear  ...   SB    CS   BB   SO  IBB  HBP  SH  SF  GIDP  NL
54          1977  ...    0   0.0    0    5    0    0   0   0     0   0
55          1910  ...    4   2.0    6   27    0    0   3   0     0   1
108         1910  ...   20  11.5  125  122    0    3  44   0     0   1
144         1975  ...    8  12.0  154  381   12    5   4  19    34   1
151         2016  ...    3   0.0   22  179    8   18   0   1     4   1
...          ...  ...   ..   ...  ...  ...  ...  ...  ..  ..   ...  ..
14111       1934  ...   38  26.0  792  867    0   12  25   0   155   0
14139       1976  ...   60  55.0  332  589   30   36  18  33    80   1
14183       1995  ...   23  19.0  479  544   30   29  14  31    87   1
14199       1954  ...   45  25.0  246  678   27   13  37  14    99   1
14210       2002  ...    0   0.0    5   34    0    0   0   1     0   1

[655 rows x 40 columns]
```

We have some catchers that are also in the fielders table.

[168]:
```
fielders[(fielders['retroID'].isin(catchers['retroID']) & fielders['pos_C'] ==␣
↪1)]
```

[168]:
```
        retroID   GS  InnOuts    PO  A  E   DP    weight  height  debutYear  \
108    ainse101    0        0    11  0  0    0  0.426230    0.40       1910
144    alexg101   22      500    83  6  3    4  0.487705    0.55       1975
151    alfaj002    1       31     8  2  0    1  0.610656    0.55       2016
156    allaa001    0       33    15  0  0    2  0.590164    0.70       1986
```

```
169    alleg001    1       54     4   4   0    2  0.446721    0.40       1979
...         ...  ...      ...   ...  ..  ..  ...       ...     ...        ...
13914  wingi101    0        0     7   2   1    0  0.344262    0.35       1911
13956  wockj001  249     6120  1429  90  17  133  0.467213    0.45       1974
14066  wronr001    0        6     2   0   0    0  0.446721    0.50       1988
14074  wyneb001    0        6     0   0   0    0  0.467213    0.50       1976
14183  zaung001    0       33     3   2   0    1  0.385246    0.35       1995

       ...  SB    CS   BB   SO  IBB  HBP  SH  SF  GIDP  NL
108    ...  20  11.5  125  122    0    3  44   0     0   1
144    ...   8  12.0  154  381   12    5   4  19    34   1
151    ...   3   0.0   22  179    8   18   0   1     4   1
156    ...  23  18.0   87  223    4    9  35  16    27   1
169    ...   3   7.0  130  192    3    5  15  11    35   0
...    ...  ..   ...  ...  ...  ...  ...  ..  ..   ...  ..
13914  ...  17  16.0  121   84    0    4  36   0     0   1
13956  ...   5  11.0  277  278   14    7   5  12    52   1
14066  ...   1   0.0    5   41    2    1   2   2     3   1
14074  ...  10  13.0  626  428   41   17  58  36   119   0
14183  ...  23  19.0  479  544   30   29  14  31    87   1

[430 rows x 40 columns]
```

[170]:
```
to_inspect = fielders[(fielders['retroID'].isin(catchers['retroID']) &␣
 ↪fielders['pos_C'] == 1)]['retroID']
```

[171]:
```
catchers[catchers['retroID'].isin(to_inspect)]
```

[171]:
```
       retroID    GS  InnOuts    PO    A   E  DP  PB  WP  SB_A  ...  SB    CS  \
6     ainse101     0        0  1528  361  72  31  34   0     0  ...  20  11.5
7     alexg101   205     5481  1008  100  35   8  24   0   212  ...   8  12.0
8     alfaj002   130     3430  1135   71  13   9  14   0    72  ...   3   0.0
10    allaa001   453    11965  2395  208  52  24  41   0   292  ...  23  18.0
11    alleg001   342     8959  1762  146  32  24  27   0   233  ...   3   7.0
...        ...   ...      ...   ...  ...  ..  ..  ..  ..   ...  ...  ..   ...
1498  wingi101     0        0  1716  536  79  53  36   0     0  ...  17  16.0
1502  wockj001   216     5957  1212  119  39  17  27   0   188  ...   5  11.0
1508  wronr001    47     1360   296   32   8   3   4   0    25  ...   1   0.0
1509  wyneb001  1164    31563  6281  583  75  88  61   0   708  ...  10  13.0
1521  zaung001   910    24700  6134  418  88  59  51   0   651  ...  23  19.0

       BB   SO  IBB  HBP  SH  SF  GIDP  NL
6     125  122    0    3  44   0     0   1
7     154  381   12    5   4  19    34   1
8      22  179    8   18   0   1     4   1
10     87  223    4    9  35  16    27   1
11    130  192    3    5  15  11    35   0
```

```
...   ...  ...  ...  ...  ..  ..   ...  ..
1498  121   84    0    4  36   0     0   1
1502  277  278   14    7   5  12    52   1
1508    5   41    2    1   2   2     3   1
1509  626  428   41   17  58  36   119   0
1521  479  544   30   29  14  31    87   1

[430 rows x 38 columns]
```

It looks like the information in the catchers table is a better indicator of the player's career.

[175]:
```
fielders[fielders['retroID'] == 'alexg101'][['debutYear', 'finalYear']]
```

[175]:
```
     debutYear  finalYear
144       1975       1981
```

[174]:
```
catchers[catchers['retroID'] == 'alexg101'][['debutYear', 'finalYear']]
```

[174]:
```
   debutYear  finalYear
7       1975       1981
```

Thee years line up. So for any catcher who appears in the fielders table with his position as catcher, we're going to drop him from the fielders table and only use the catchers information. We'll keep catchers in the fielders table if they're in a different position.

[176]:
```
fielders[fielders['pos_C'] == 1]
```

[176]:
```
        retroID   GS  InnOuts    PO   A   E   DP    weight  height  debutYear  \
108    ainse101    0        0    11   0   0    0  0.426230    0.40       1910
144    alexg101   22      500    83   6   3    4  0.487705    0.55       1975
151    alfaj002    1       31     8   2   0    1  0.610656    0.55       2016
156    allaa001    0       33    15   0   0    2  0.590164    0.70       1986
169    alleg001    1       54     4   4   0    2  0.446721    0.40       1979
...         ...  ...      ...   ...  ..  ..  ...       ...     ...        ...
13914  wingi101    0        0     7   2   1    0  0.344262    0.35       1911
13956  wockj001  249     6120  1429  90  17  133  0.467213    0.45       1974
14066  wronr001    0        6     2   0   0    0  0.446721    0.50       1988
14074  wyneb001    0        6     0   0   0    0  0.467213    0.50       1976
14183  zaung001    0       33     3   2   0    1  0.385246    0.35       1995

       ...  SB    CS   BB   SO  IBB  HBP  SH  SF  GIDP  NL
108    ...  20  11.5  125  122    0    3  44   0     0   1
144    ...   8  12.0  154  381   12    5   4  19    34   1
151    ...   3   0.0   22  179    8   18   0   1     4   1
156    ...  23  18.0   87  223    4    9  35  16    27   1
169    ...   3   7.0  130  192    3    5  15  11    35   0
...    ...  ..   ...  ...  ...  ...  ...  ..  ..   ...  ..
13914  ...  17  16.0  121   84    0    4  36   0     0   1
```

```
13956  ...   5  11.0  277  278   14    7   5  12    52   1
14066  ...   1   0.0    5   41    2    1   2   2     3   1
14074  ...  10  13.0  626  428   41   17  58  36   119   0
14183  ...  23  19.0  479  544   30   29  14  31    87   1

[430 rows x 40 columns]
```

All fielders with a position of 'C' are in the catchers table, so we don't have to worry about leaving any by filtering with the following predicate.

[184]:
```
fielders = fielders[~(fielders['retroID'].isin(catchers['retroID']) &␣
↪fielders['pos_C'] == 1)]
```

[185]:
```
fielders.shape
```

[185]:
```
(5962, 40)
```