intro

March 17, 2022

```
[1]: import ncaa_scrape as ncaa
     import pandas as pd
     import metrics
```

0.0.1 Hello! In this notebook, I will be walking you through what collegebaseball package can do.

You can use **get_team_stats** to retrieve single-season batting or pitching statistics for a school. It works for the 2013 - 2022 seasons.

Just give it a school_name/school_id, a season/season_id, and what kind of stats you would like. Acceptable school names in the data/schools.parquet

```
[39]: pd.read_parquet('data/schools.parquet').head()
```

```
[39]:            ncaa_name            bd_name  school_id  max_season  min_season
      0  Abilene Christian  Abilene Christian          2        2022        2013
      1          Air Force          Air Force        721        2022        2013
      2              Akron              Akron          5        2022        2013
      3            Alabama            Alabama          8        2022        2013
      4        Alabama A&M        Alabama A&M          6        2022        2013
```

```
[40]: # ex: school_name & YYYY
      ncaa.get_team_stats("cornell", 2019, "pitching").head()
```

```
[40]:     Jersey                name  Yr pos  GP  GS  App  GS   ERA    IP  …  \
      10      34         John Natoli  Jr   P  19   1   19   1  1.73  36.1  …
      11       1     Nikolas Lillios  Fr   P  15   5    9   0  3.27  11.0  …
      12      26  Trevor Daniel Davis  So   P  14   0   14   0  4.96  16.1  …
      13      32       Jon Zacharias  Fr   P  13   8   13   8  2.83  41.1  …
      14      12       Luke Yacinich  Fr   P  13   4   13   4  8.07  32.1  …

          SHA  SFA  Pitches  GO  FO  W  L  SV  KL  season
      10    1    1      389  32  29  5  2   7  12    2019
      11    0    2       57  13  13  0  0   0   2    2019
      12    2    2      161  14  15  1  0   0   6    2019
      13    3    3      407  45  42  1  3   0  10    2019
      14    2    0      268  33  39  2  4   0   5    2019
```

```
[5 rows x 38 columns]
```

[43]:
```
# ex: school_id and season_id
# we'll go over what lookup functions exists a bit later
school_id = ncaa.lookup_school_id('cornell')
season_id = ncaa.lookup_season_id(2019)

ncaa.get_team_stats(school_id, season_id, "batting").head()
```

[43]:
```
   Jersey              name  Yr  pos  GP  GS     BA  OBPct SlgPct   R  …  \
0      15      Matt Collins  Jr  INF  38  36  0.217  0.314  0.325  13  …
1      33  William Simoneit  Sr    C  38  37  0.299  0.357  0.493  17  …
2       6        Josh Arndt  Sr  INF  38  38  0.214  0.285  0.303  16  …
3      21       Ramon Garza  So  INF  37  37  0.219  0.272  0.299  13  …
4       2   Nicholas Binnie  So    P  34  30  0.257  0.316  0.305  13  …

   HBP  SF  SH   K  DP  CS  Picked  SB  IBB  season
0    2   0   1  42   1   1       0  10    0   14781
1    5   0   0  13   6   0       0   6    2   14781
2    2   4   0  32   7   3       0   6    0   14781
3    2   3   1  23   4   2       0   7    0   14781
4    1   0   3  26   2   0       0   6    0   14781

[5 rows x 28 columns]
```

You can also get a players single-season stats for their entire college career, even if they moved teams.

Just give **get_career_stats** a player_id and the kind of stats you want.

[17]:
```
ncaa.get_career_stats(2111707, 'pitching')
```

[17]:
```
  school_id  GP   G  App  GS   ERA    IP   CG   H  R  … SHA  SFA  Pitches  \
0       167  15  15    9   0  3.27  11.0  0.0  12  5  …   0    2       57
1       167   5   5    5   0  1.69   5.1  0.0   7  1  …   1    0       76

   GO  FO    W    L   SV  KL  season
0  13  13  0.0  0.0  0.0   2    2018
1   6   6  0.0  0.0  0.0   1    2019

[2 rows x 35 columns]
```

n.b. get_career_stats cannot take a player name due to potential ambiguities. No worries though, as getting a
player_id is easy with lookup_player_id(player_name, school_name).

[45]:
```
player_id = ncaa.lookup_player_id('William Simoneit', 'Cornell')
ncaa.get_career_stats(player_id, 'batting')
```

```
[45]:   school_id  GP  G     BA  OBPct  SlgPct   R   AB   H  2B  …  SF  SH   K  \
     0        167  24  0  0.317  0.391   0.610  16   82  26   9  …   0   0  14
     1        167  37  0  0.308  0.380   0.406  17  143  44   8  …   4   0  23
     2        167  38  0  0.299  0.357   0.493  17  144  43  10  …   0   0  13
     3        749  17  0  0.377  0.462   0.642  13   53  20   5  …   2   0  14

         DP  CS  Picked  SB  RBI2out  IBB  season
     0  3.0   0       0   1        0    0    2016
     1  0.0   2       0   7        0    0    2017
     2  6.0   0       0   6        0    2    2018
     3  2.0   0       0   0        0    0    2019

     [4 rows x 26 columns]
```

Calculating advanced stats from these stats is made simple with **add_pitching_metrics** and **add_batting_metrics**.

Just pass any DataFrame obtained from get_team_stats or get_career_stats.

```
[48]: metrics.add_pitching_metrics(ncaa.get_team_stats("cornell", 2019, "pitching")).
        ↪head()
```

```
[48]:     Jersey               name  Yr pos  GP  GS  App  GS   ERA    IP  …  \
     10      34        John Natoli  Jr   P  19   1   19   1  1.73  36.1  …
     11       1   Nikolas Lillios  Fr   P  15   5    9   0  3.27  11.0  …
     13      32     Jon Zacharias  Fr   P  13   8   13   8  2.83  41.1  …
     18       8     Kevin Cushing  Fr   P  11   0   11   0  4.50  12.0  …
     15      31       Colby Wyatt  Jr   P  13  10   13  10  3.68  63.2  …

         OBP-against  BA-against  SLG-against  OPS-against    K/PA     K/9  BB/PA  \
     10        0.248       0.176        0.199        0.447   0.309  11.395  0.060
     11        0.333       0.300        0.375        0.708   0.089   3.273  0.067
     13        0.316       0.247        0.313        0.629   0.172   6.532  0.080
     18        0.404       0.333        0.400        0.804   0.135   5.250  0.115
     15        0.340       0.290        0.405        0.745   0.124   4.948  0.064

          BB/9  BABIP-against    FIP
     10  2.229          0.231  2.410
     11  2.455          0.279  4.042
     13  3.048          0.247  4.120
     18  4.500          0.326  4.284
     15  2.545          0.285  4.548

     [5 rows x 50 columns]
```

```
[49]: metrics.add_pitching_metrics(ncaa.get_career_stats(2111716, 'pitching')).head()
```

```
[49]:   school_id  GP   G  App  GS    ERA    IP   CG   H   R  … OBP-against  \
      0        167  11  11   11   0   4.50  12.0  0.0  15   6  …       0.404
      1        167   3   3    3   0   7.94   5.2  0.0   7   6  …       0.385
      2        167   3   3    3   2  20.57   7.0  0.0  13  17  …       0.574

         BA-against  SLG-against  OPS-against   K/PA     K/9  BB/PA    BB/9  \
      0       0.333        0.400        0.804  0.135   5.250  0.115   4.500
      1       0.304        0.522        0.907  0.308  12.706  0.115   4.765
      2       0.394        0.758        1.332  0.064   3.857  0.191  11.571

        BABIP-against     FIP
      0         0.326   4.174
      1         0.353   5.010
      2         0.244  14.195

      [3 rows x 47 columns]
```

```
[50]: metrics.add_batting_metrics(ncaa.get_team_stats(167, 2019, 'batting')).head()
```

```
[50]:    Jersey              name  Yr  pos  GP  GS     BA  OBPct SlgPct   R  …  \
      9      13         Adam Saks  Sr    P  26  26  0.337  0.414  0.515  13  …
      1      33  William Simoneit  Sr    C  38  37  0.299  0.357  0.493  17  …
      0      15      Matt Collins  Jr  INF  38  36  0.217  0.314  0.325  13  …
      7      25    Jason Apostle  So   OF  28  23  0.233  0.333  0.315   6  …
      4       2   Nicholas Binnie  So    P  34  30  0.257  0.316  0.305  13  …

           OBP    SLG    OPS    ISO     K%    BB%  BABIP   wOBA   wRAA     wRC
      9  0.412  0.515  0.927  0.178  0.092  0.118  0.367  0.416  6.946  24.201
      1  0.374  0.493  0.867  0.194  0.084  0.052  0.296  0.390  5.191  27.666
      0  0.312  0.325  0.637  0.108  0.304  0.109  0.316  0.309 -6.075  13.935
      7  0.318  0.315  0.633  0.082  0.273  0.091  0.347  0.308 -3.958   8.802
      4  0.308  0.305  0.613  0.048  0.222  0.068  0.342  0.295 -6.718  10.247

      [5 rows x 40 columns]
```

```
[51]: metrics.add_batting_metrics(ncaa.get_career_stats(1779078, 'batting')).head()
```

```
[51]:   school_id  GP  G     BA  OBPct  SlgPct   R   AB   H  2B  …    OBP    SLG  \
      2        167  32  0  0.231  0.272   0.368  14  117  27   8  …  0.272  0.368
      3        167  38  0  0.214  0.285   0.303  16  145  31   8  …  0.285  0.303
      0        167  28  0  0.229  0.289   0.277   7   83  19   4  …  0.286  0.277
      1        167  28  0  0.187  0.256   0.280   5   75  14   7  …  0.253  0.280

           OPS    ISO     K%    BB%  BABIP   wOBA    wRAA     wRC
      2  0.640  0.137  0.216  0.048  0.281  0.293  -6.639  11.236
      3  0.588  0.089  0.194  0.085  0.259  0.284 -10.744  12.521
      0  0.563  0.048  0.110  0.066  0.260  0.277  -5.929   6.538
```

```
1  0.533  0.093  0.241  0.084  0.255  0.257  -6.854   4.683

[4 rows x 38 columns]
```

You can get a school's roster with **get_roster**, which takes either school_name/school_id and season/season_id.
get_roster also gives the column stats_player_seq (i.e. player_id), which is quite useful!

```
[28]: ncaa.get_roster('cornell', 2018).head()
```

```
[28]:   jersey stats_player_seq             name position class_year games_played  \
     0     25          1997329   Apostle, Jason       OF         Fr           24
     1      6          1779078      Arndt, Josh      INF         Jr           32
     2     16          1779080   Arnold, Austin        P         Jr            3
     3     29          1997331  Bailey, Garrett        P         Fr            4
     4     35          1652419       Baur, Trey      INF         Sr           26

       games_started
     0            19
     1            27
     2             0
     3             0
     4            22
```

If you want a team's roster over multiple years, you can use **get_multiyear_roster**, which returns a concatenated (not aggregated!) DataFrame.

It also adds on season_ids, which can be handy.

```
[5]: ncaa.get_multiyear_roster('cornell', 2015, 2018).head()
```

```
[5]:   jersey stats_player_seq                name position class_year games_played  \
    0     20          1546998  Balestrieri, Paul        P         So           15
    1     35          1652419         Baur, Trey      INF         Fr            9
    2     24          1652397       Bitar, Ellis        C         Fr           21
    3     37          1547001        Brewer, Ray        P         So            5
    4     32          1324535        Busto, Nick        P         Sr           11

      games_started  season   school  season_id  batting_id  pitching_id
    0             3    2015  cornell      12080       10780        10781
    1             5    2015  cornell      12080       10780        10781
    2            18    2015  cornell      12080       10780        10781
    3             0    2015  cornell      12080       10780        10781
    4             6    2015  cornell      12080       10780        10781
```

```
[6]: ncaa.get_multiyear_roster('cornell', 2015, 2018).tail()
```

```
[6]:     jersey stats_player_seq           name position class_year games_played  \
    132     20          1997324    Urbon, Seth        P         So           12
```

```
133     11          1783016    Wahl, Austin        P          Sr            16
134      5          1652396   Wickham, Dale       OF          Sr            36
135     36          1547000  Willittes, Tim        P          Sr            11
136     31          1884389    Wyatt, Colby        P          So            19

    games_started  season   school  season_id  batting_id  pitching_id
132            10    2018  cornell      12973       11953        11954
133             1    2018  cornell      12973       11953        11954
134            35    2018  cornell      12973       11953        11954
135            11    2018  cornell      12973       11953        11954
136             0    2018  cornell      12973       11953        11954
```

Finally, there are some lookup functions included to make life easier.

You can get a player_id from their name and school with **lookup_player_id**

```
[30]: ncaa.lookup_player_id("Jake Gelof", "Virginia")
```

```
[30]: 2486499
```

the season_id, batting_id, and pitching_id of a given season with **lookup_season_ids**

```
[31]: ncaa.lookup_season_ids(2019)
```

```
[31]: (14781, 14643, 14644)
```

or the season, batting_id, and pitching_id for a given season_id with **lookup_season_ids_reverse**

```
[32]: ncaa.lookup_season_ids_reverse(14781)
```

```
[32]: (2019, 14643, 14644)
```

sometimes, you just need a single season_id. **lookup_season_id** has you covered.

```
[33]: ncaa.lookup_season_id(2019)
```

```
[33]: 14781
```

You can also find the debut and most recent seasons in which a given playeer has made and appearance in.
Just pass a player_id into **lookup_seasons_played**.

```
[34]: ncaa.lookup_seasons_played(2486499)
      # n.b. need to update table for 2022 season
```

```
[34]: (2021, 2021)
```

you can get a school_id by giving **lookup_school_id** the correct (from data/schools.parquet or data/schools/csv) school name

[7]:

```
ncaa.lookup_school_id("Cornell")
```

[7]:

```
167
```