NIL College Athlete Web Scraping Project

James Hamlette

The University of Nebraska at Omaha

ECON 8320: Graduate Tools for Data Analysis Term Project
May 11, 2023

Outline

1. Introduction
2. Code
3. Analysis
4. Other

Introduction

NIL Athlete Data Scraping

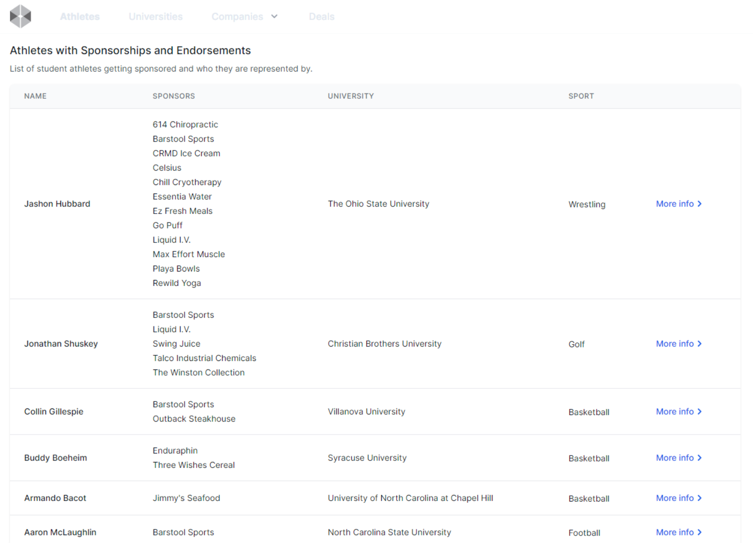
Project and Goal

- Scrape NIL endorsement/sponsor deals and related data
- Few websites list NIL deals and less provide usable data
- Quickly growing data and constantly changing

Using On3 and NIL College Athletes Websites

- https://www.on3.com/nil/rankings/player/nil-100/
- https://nilcollegeathletes.com/athletes

Websites



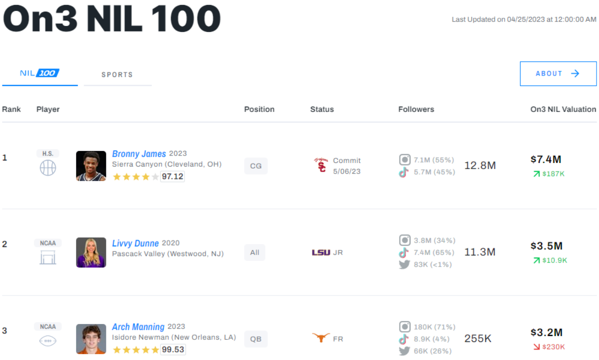
Athletes with Sponsorships and Endorsements

List of student athletes getting sponsored and who they are represented by.

NAME	SPONSORS	UNIVERSITY	SPORT	
Jashon Hubbard	614 Chiropractic Barstool Sports CRMD Ice Cream Celsius Chill Cryotherapy Essentia Water Ez Fresh Meals Go Puff Liquid I.V. Max Effort Muscle Playa Bowls Rewild Yoga	The Ohio State University	Wrestling	More info ›
Jonathan Shuskey	Barstool Sports Liquid I.V. Swing Juice Talco Industrial Chemicals The Winston Collection	Christian Brothers University	Golf	More info ›
Collin Gillespie	Barstool Sports Outback Steakhouse	Villanova University	Basketball	More info ›
Buddy Boeheim	Enduraphin Three Wishes Cereal	Syracuse University	Basketball	More info ›
Armando Bacot	Jimmy's Seafood	University of North Carolina at Chapel Hill	Basketball	More info ›
Aaron McLaughlin	Barstool Sports	North Carolina State University	Football	More info ›

Figure: NIL Athletes Website

Websites Cont.



On3 NIL 100

Last Updated on 04/25/2023 at 12:00:00 AM

NIL 100 SPORTS ABOUT →

Rank		Player	Position	Status	Followers		On3 NIL Valuation
1	H.S.	Bronny James 2023 Sierra Canyon (Cleveland, OH) ★★★★☆ 97.12	CG	Commit 5/06/23	7.1M (55%) 5.7M (45%)	12.8M	$7.4M ↗ $187K
2	NCAA	Livvy Dunne 2020 Pascack Valley (Westwood, NJ)	All	LSU JR	3.8M (34%) 7.4M (65%) 83K (<1%)	11.3M	$3.5M ↗ $10.9K
3	NCAA	Arch Manning 2023 Isidore Newman (New Orleans, LA) ★★★★★ 99.53	QB	FR	180K (71%) 8.9K (4%) 66K (26%)	255K	$3.2M ↘ $230K

Figure: On3 Website

Code

Nil Deals Code

```
###########ALL NIL College Athlete Website Scraping, Cleaning, and Analyzing Code################

#Load relevant libraries
import requests
from bs4 import BeautifulSoup
import numpy as np
import pandas as pd
import re
from urllib.parse import urljoin

#Define function that takes a url link and creates columns of data as follows
def collectNames(startURL):
    myPage = requests.get(startURL)
    parsed = BeautifulSoup(myPage.text)

    #Start with the names of the athletes via tag "a"
    a = parsed.find_all('td', class_="px-2 md:px-6 py-4 whitespace-nowrap text-sm font-medium text-gray-900")
    n=[i.a.text.strip() for i in a]

    #Append names to "ndata"
    ndata=[]
    for x in n:
        ndata.append(x)
    ndata=pd.DataFrame(ndata, columns=['Name'])
```

Nil Deals Code Cont.

```
#Append Sponsor to "t3"
   t=[]
   for i in a:
       try:
           d=i.find_next_sibling()
           t.append(d.text)
       except:
           t.append("not listed")
   t2=list(t)
   t2new= [item.strip().replace('\n','') for item in t2]
   t3=pd.DataFrame(t2new, columns=['Sponsors'])

   #Append University and Sport to "datab"
   b= parsed.find_all('span', class_="truncate")
   blist= []
   for i in b:
       blist.append(i.text)
   blist2=list(blist)
   #Values are in succeeding positions, so create lists for every other to split
   left = []
   right = []
   for i, j in enumerate(blist2):
       if i%2==0:
           left.append(j.strip())
       else:
           right.append(j.strip())
```

Nil Deals Code part 3

```

    #zip the two lists of Universities and Sports Together
    b3 = list(zip(left, right))
    b4=[list(i) for i in b3]
    datab= pd.DataFrame(b4, columns=['University', 'Sport'])

    #Inner join of ndata and datab
    all_data= ndata.join([datab,t3])

    #Parse through all remaining pages if there is one, then concatenate using recursive function
    try:
        nextPage= urljoin( 'https://nilcollegeathletes.com', parsed.find('div', class_="-mt-px flex w-0 flex-1 justify-end").a['href'])
    except:
        nextPage=None
    if nextPage:
        return pd.concat([all_data,collectNames(nextPage)], axis=0)
    else:
        return all_data
###############################################END OF FUNCTION###########################################################
main_data=collectNames('https://nilcollegeathletes.com/athletes')
main_data
main_data.to_csv("main_data.csv")
```

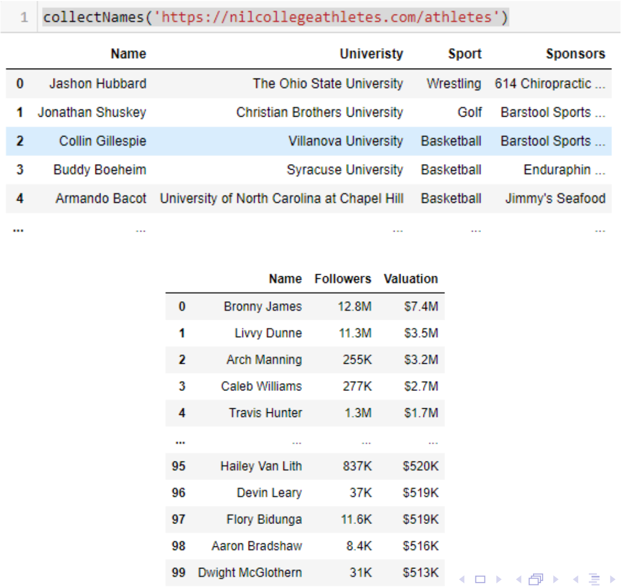
ON3 Top 100 Scraping

```python
##################### ON3 TOP 100 OF ALL ATHLETES SCRAPING, CLEANING, ANALYSIS#######################

#import relevant libraries
import numpy as np
import pandas as pd
import plotly.express as px
import requests
from bs4 import BeautifulSoup

#Define a scraping function to take a url link value
def onThree(scrapeurl):
    myPage_ = requests.get("https://www.on3.com/nil/rankings/player/nil-100/")
    soup = BeautifulSoup(myPage_.text)

    #Scrape Names and append to dataframe
    oo= soup.find_all('a', class_="MuiTypography-root MuiLink-root MuiLink-underlineNone NilPlayerRankingItem_name__nzSp9 MuiTypography-h5 MuiTypography-colorPrimary")
    oo=list(oo)

    ood=[]
    for o in oo:
        oo2=o.text
        ood.append(oo2)
    ood=pd.DataFrame(ood, columns=['Name'])
```

ON3 Top 100 Scraping Cont.

```
#Scrape the text/string number of followers
    for o in oo:
        uu= soup.find_all(’p’, class_="MuiTypography-root NilPlayerRankingItem_followersNumber__ifWQr MuiTypography-body1 MuiTypography-colorTextPrimary")
        uu=list(uu)
    uud=[]
    for u in uu:
        try:
            uu2=u.text
            uud.append(uu2)
        except:
            uud.append("blank")
    uud=pd.DataFrame(uud, columns=[’Followers’])
    #Scape the String of NIL Valuation
    vv= soup.find_all(’p’, class_="MuiTypography-root NilPlayerRankingItem_valuationCurrency__oSkvo MuiTypography-body1 MuiTypography-colorTextPrimary")
    vvd=[]
    for v in vv:
        vvs=v.text
        vvd.append(vvs)
    vvd=pd.DataFrame(vvd, columns=[’Valuation’])
    #Join the datasets
    full_data=ood.join([uud,vvd])
    #No further pages so just return
    return full_data
```

Data Snapshot



```
collectNames('https://nilcollegeathletes.com/athletes')
```

	Name	Univeristy	Sport	Sponsors
0	Jashon Hubbard	The Ohio State University	Wrestling	614 Chiropractic ...
1	Jonathan Shuskey	Christian Brothers University	Golf	Barstool Sports ...
2	Collin Gillespie	Villanova University	Basketball	Barstool Sports ...
3	Buddy Boeheim	Syracuse University	Basketball	Enduraphin ...
4	Armando Bacot	University of North Carolina at Chapel Hill	Basketball	Jimmy's Seafood
...	...	...	...	...

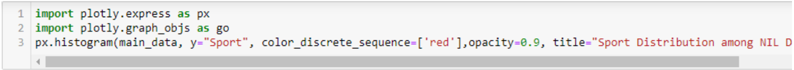
	Name	Followers	Valuation
0	Bronny James	12.8M	$7.4M
1	Livvy Dunne	11.3M	$3.5M
2	Arch Manning	255K	$3.2M
3	Caleb Williams	277K	$2.7M
4	Travis Hunter	1.3M	$1.7M
...	...	...	...
95	Hailey Van Lith	837K	$520K
96	Devin Leary	37K	$519K
97	Flory Bidunga	11.6K	$519K
98	Aaron Bradshaw	8.4K	$516K
99	Dwight McGlothern	31K	$513K

ON3 Top 100 Cleaning

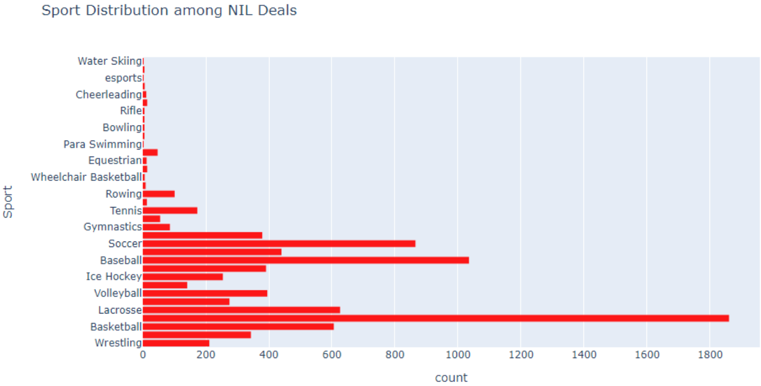
```
##Begin Extraction using defined function
tophundred=onThree("https://www.on3.com/nil/rankings/player/nil-100/")
tophundred.to_csv("on3top100.csv")
############################################CLEANING BELOW########################################
#Valuation has "$", so remove to allow for quantitative analysis
tophundred['Valuation']=tophundred['Valuation'].str.replace('$','')
###Values for thousands, millions etc are as "5K"; write function to convert to numeric values "5,000"
def value_change(num):
    if num[-1:]=='K':
        return float(num[:-1]) * 10**3
    elif num[-1:]=='M':
        return float(num[:-1]) * 10**6
    elif num[-1:]=='B':
        return float(num[:-1]) * 10**9
    else:
        num=float(num)
#Use value_change function to apply it to both followers and valuation columns without replacing original data
tophundred['Followers_total']=tophundred['Followers'].apply(value_change)
tophundred['Valuation_total']=tophundred['Valuation'].apply(value_change)
#Generate Rank variable based on the index
tophundred['Rank']=tophundred.index +1
```

Analysis

NIL Deals by Sport

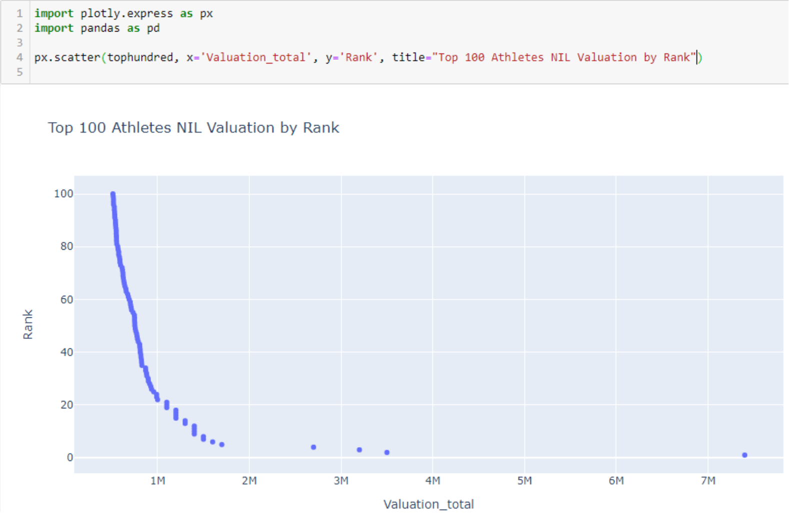


```
import plotly.express as px
import plotly.graph_objs as go
px.histogram(main_data, y="Sport", color_discrete_sequence=['red'],opacity=0.9, title="Sport Distribution among NIL D
```



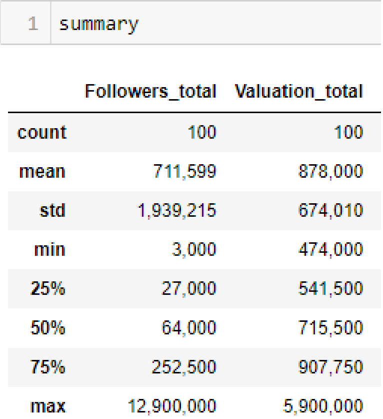
Football has the most deals by far; Barstool made up over 88% of NIL deals

On3 Top 100 Athletes Valuation



```
import plotly.express as px
import pandas as pd

px.scatter(tophundred, x='Valuation_total', y='Rank', title="Top 100 Athletes NIL Valuation by Rank")

```

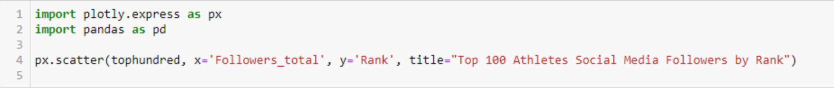
On3 Top 100 Athletes Summary



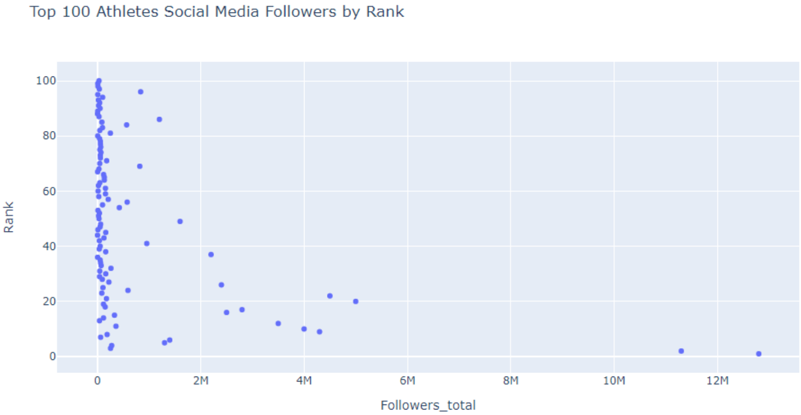
```
summary
```

	Followers_total	Valuation_total
count	100	100
mean	711,599	878,000
std	1,939,215	674,010
min	3,000	474,000
25%	27,000	541,500
50%	64,000	715,500
75%	252,500	907,750
max	12,900,000	5,900,000

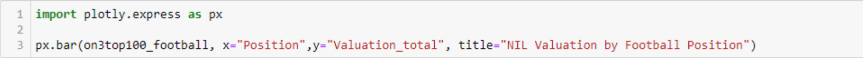
On3 Top 100 Athletes Followers



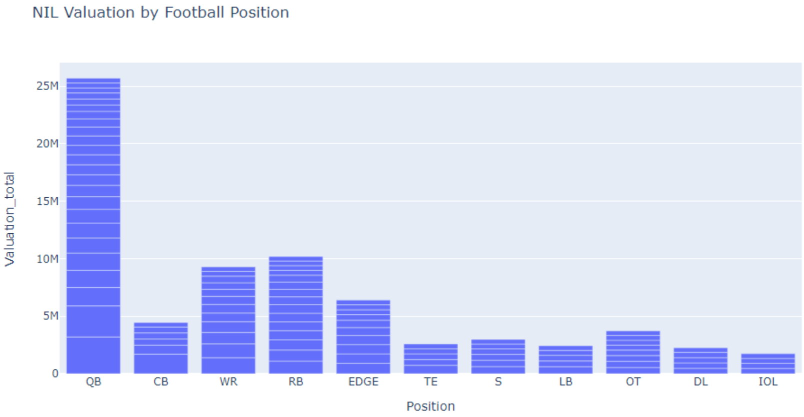
```
import plotly.express as px
import pandas as pd

px.scatter(tophundred, x='Followers_total', y='Rank', title="Top 100 Athletes Social Media Followers by Rank")

```



On3 Top 100 Football Positions

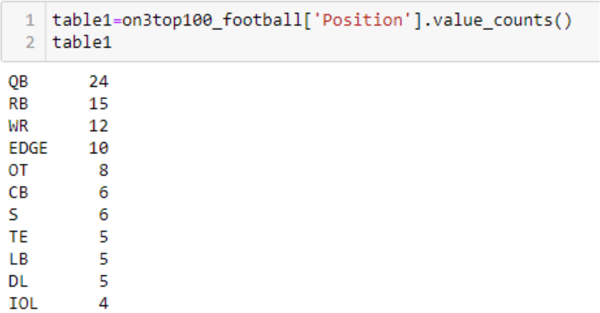


```
import plotly.express as px

px.bar(on3top100_football, x="Position",y="Valuation_total", title="NIL Valuation by Football Position")
```



NIL Valuation by Football Position

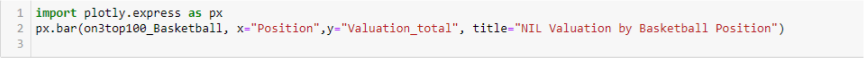
On3 Top 100 Football Positions Cont.



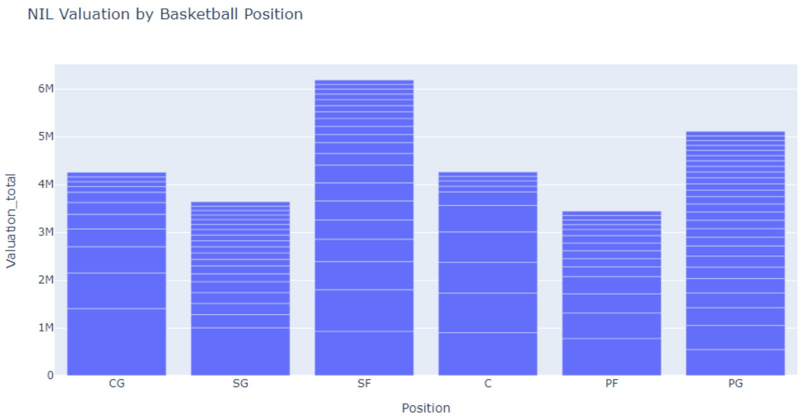
```
table1=on3top100_football['Position'].value_counts()
table1
```

```
QB      24
RB      15
WR      12
EDGE    10
OT       8
CB       6
S        6
TE       5
LB       5
DL       5
IOL      4
```

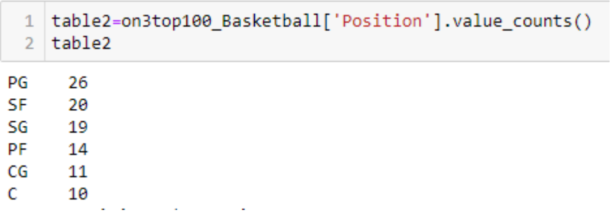
On3 Top 100 Basketball Positions



```
import plotly.express as px
px.bar(on3top100_Basketball, x="Position",y="Valuation_total", title="NIL Valuation by Basketball Position")

```



On3 Top 100 Basketball Positions Cont.



```
table2=on3top100_Basketball['Position'].value_counts()
table2
```

```
PG    26
SF    20
SG    19
PF    14
CG    11
C     10
```

Other

Not Working

- Twitter links proved difficult to scrape from NIL
- Values on NIL website were not readily available
- Load More Button is not very compatible with Beautiful Soup
- University and Individual Names using hyphens or apostrophes were unable to be recognized without correction

Further Ideas

More Quantitative Values

- difficult to do thorough analysis without more than just values
- could look to pull further athlete data from roster websites or NCAA database to match with NIL valuation deals

This is a consistently updating list of deals so only pulling new values would be the most helpful moving forward (especially for On3)