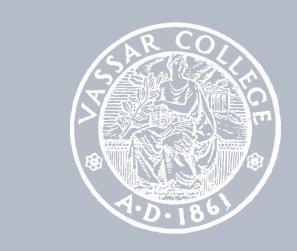
CMPU 100 · Programming with Data

Working with Tabular Data

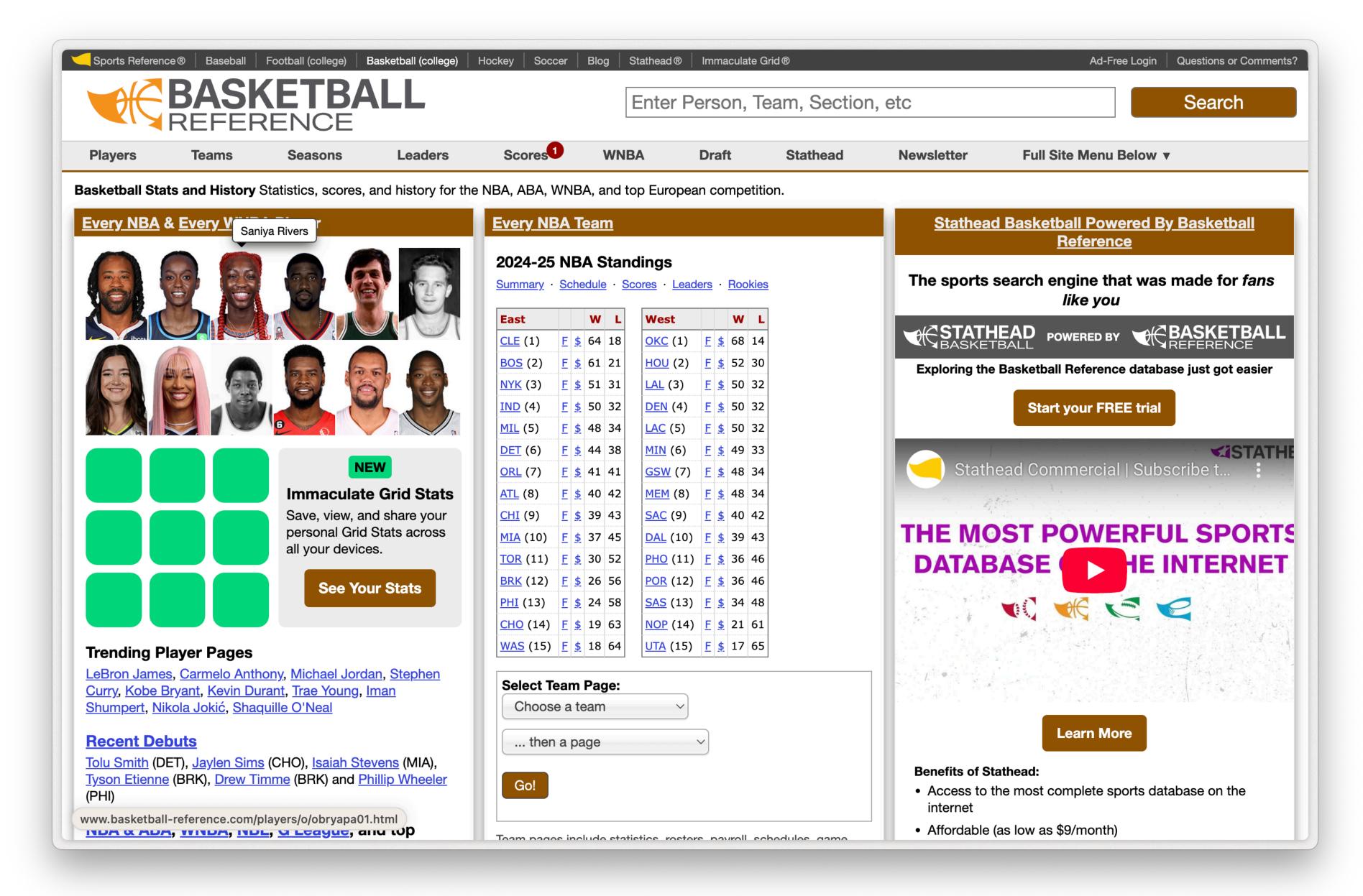
Class 7

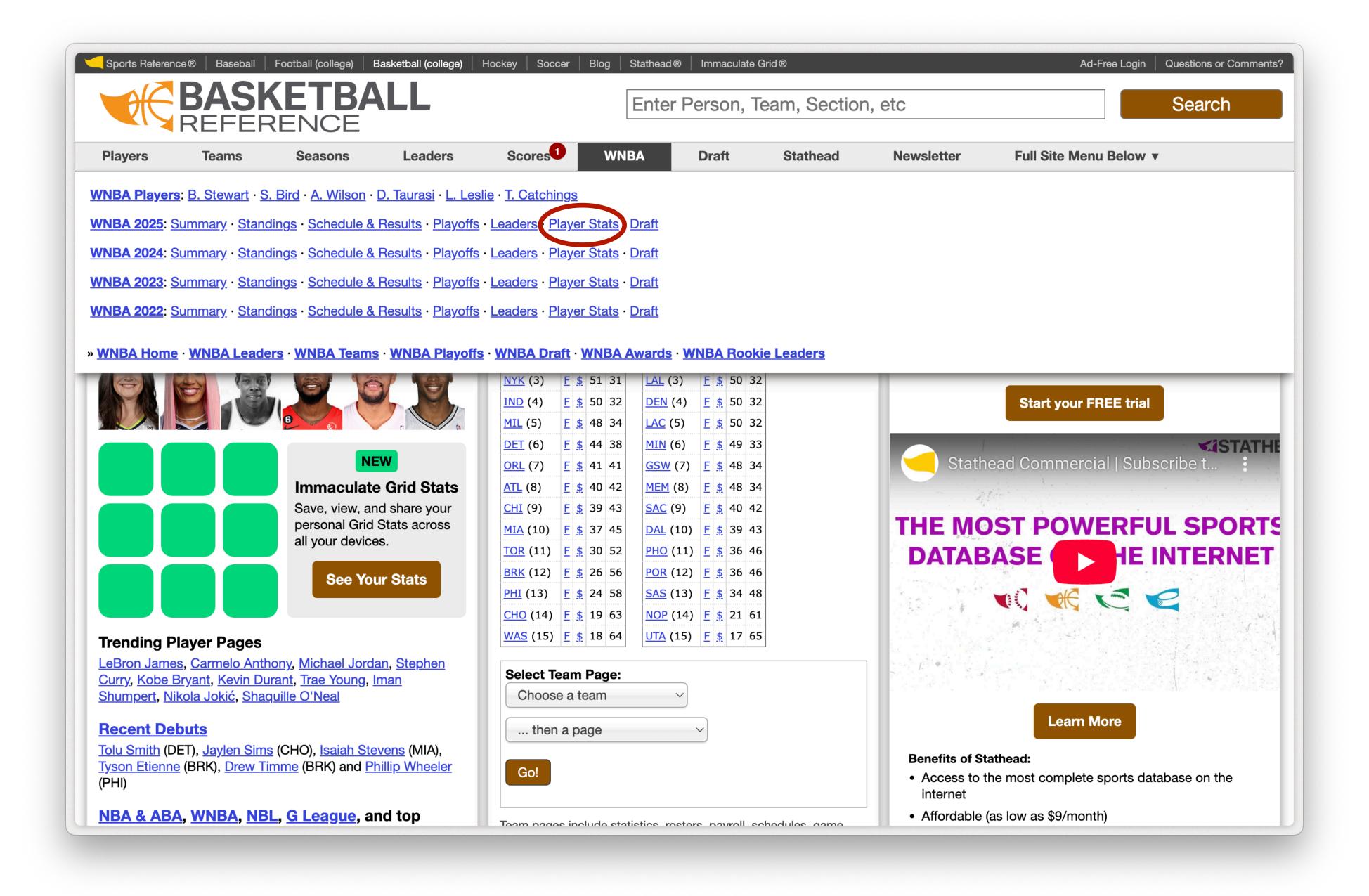


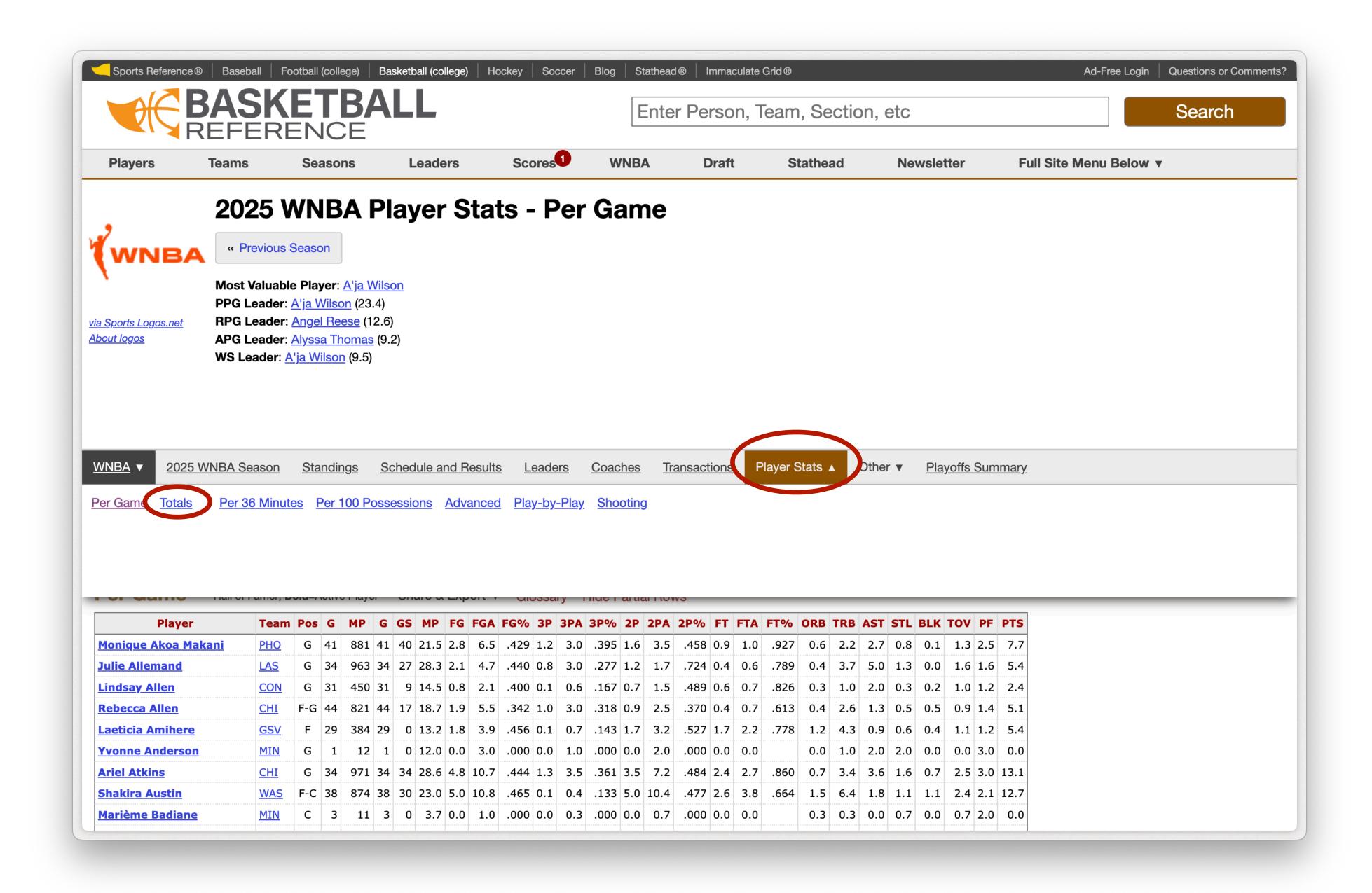
Loading tabular data

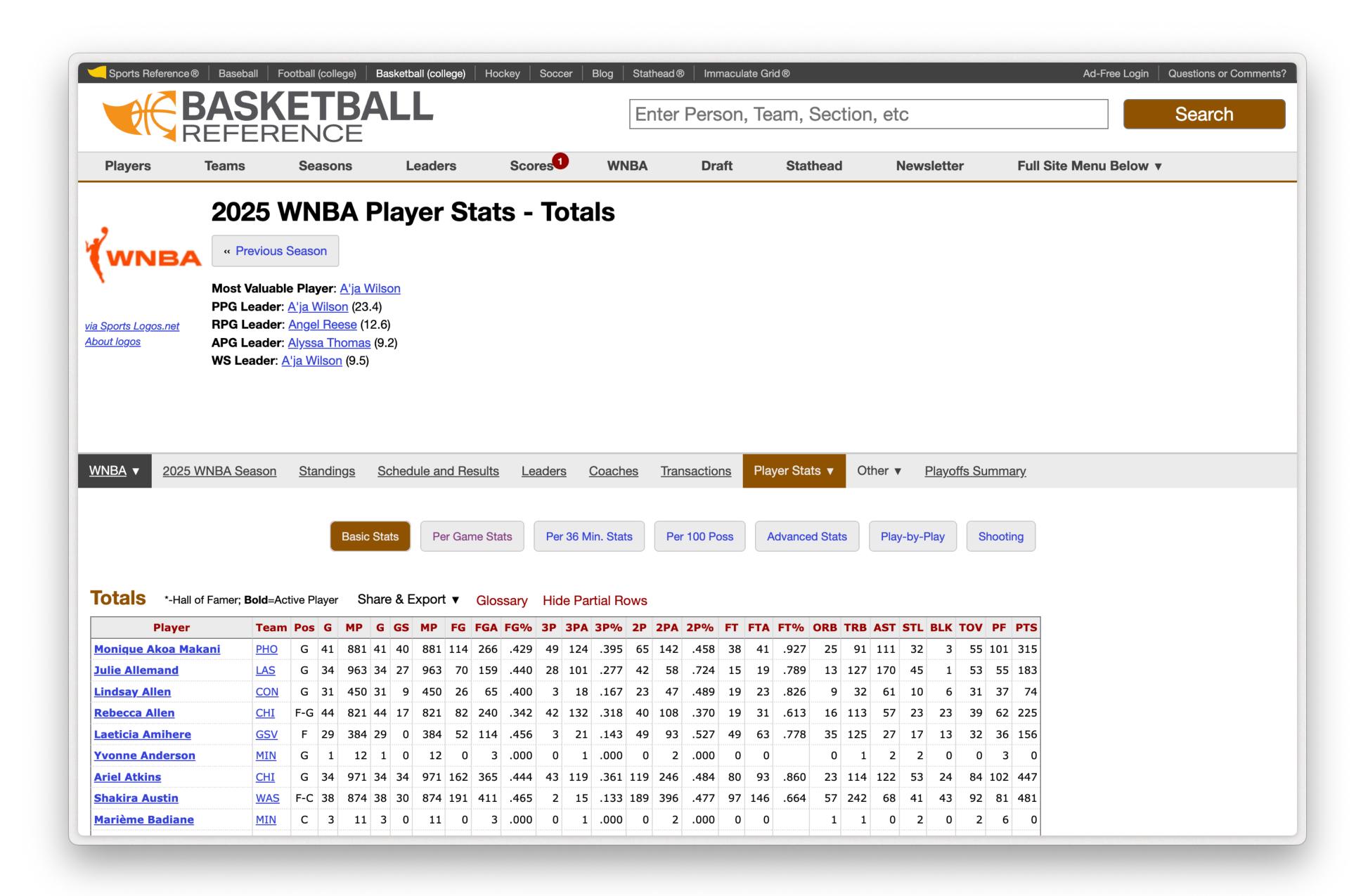


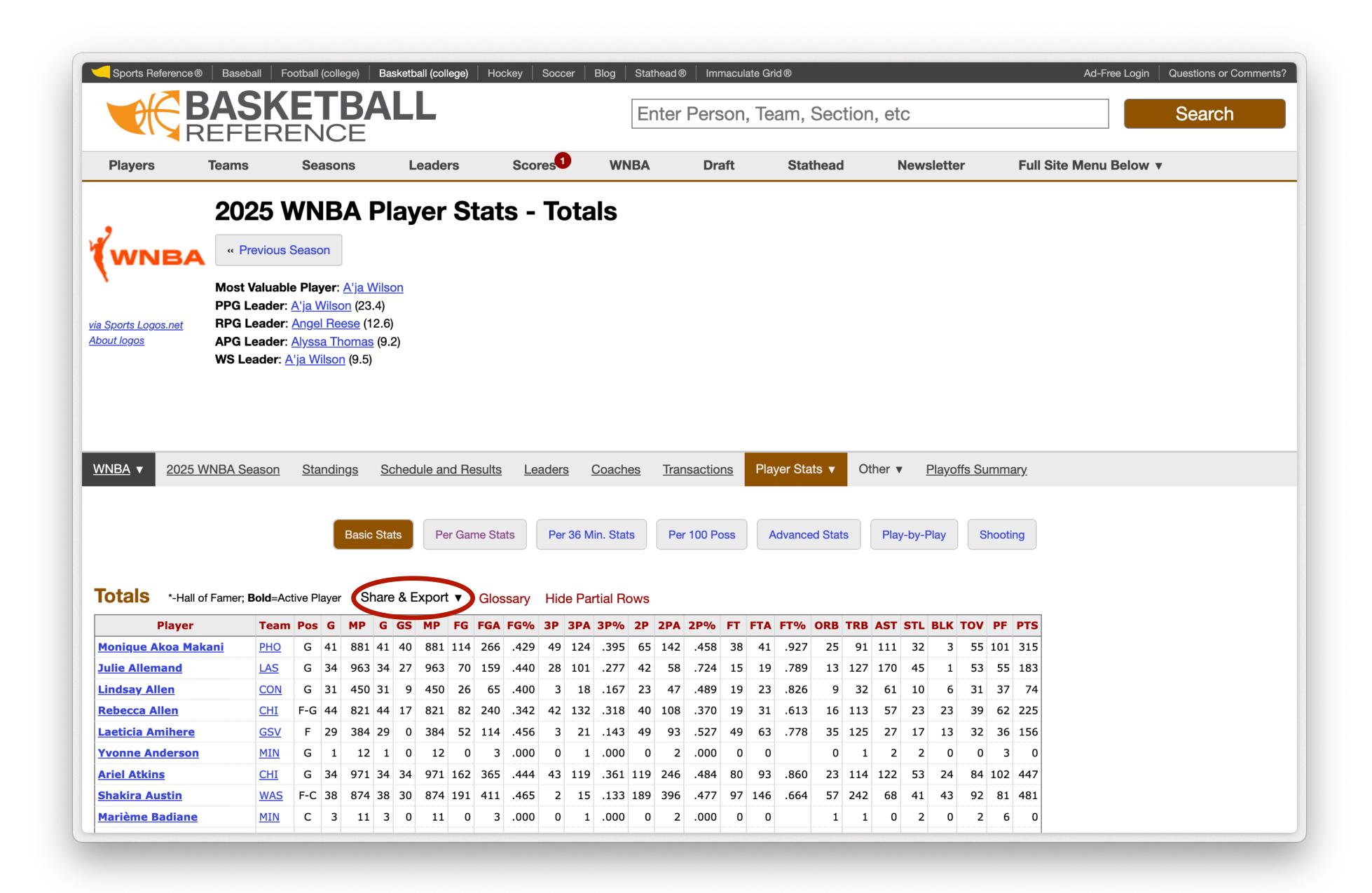


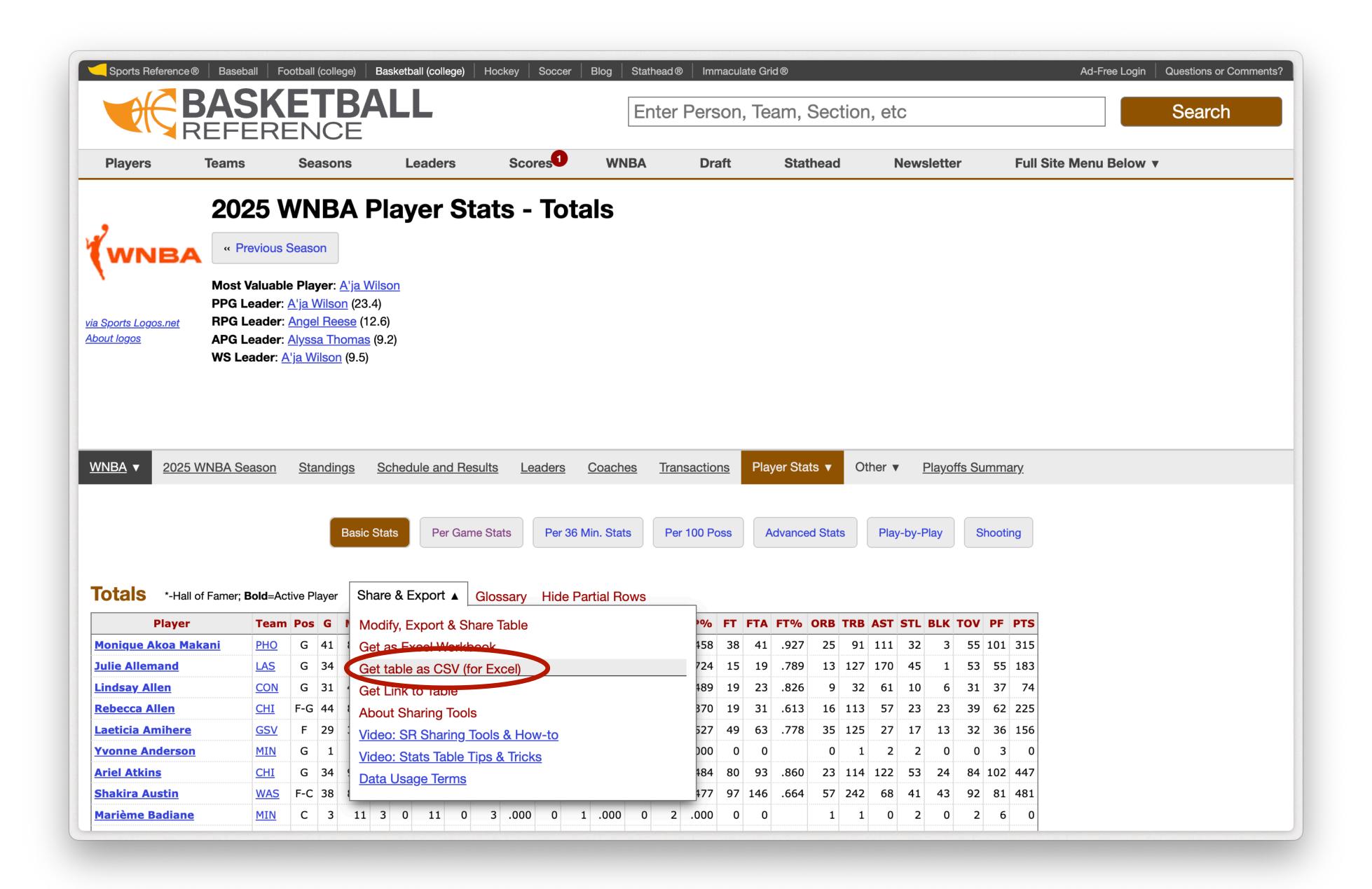


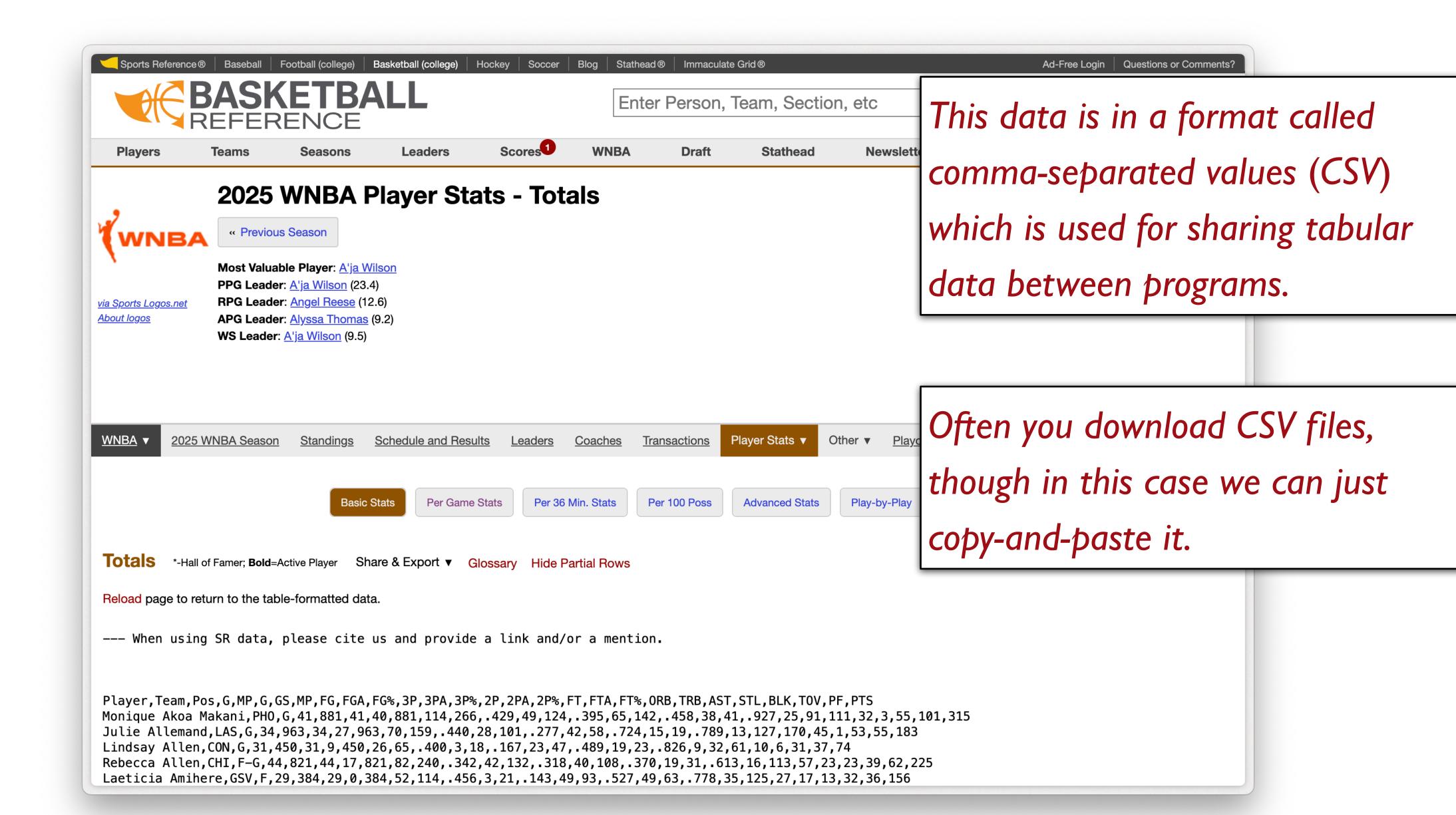




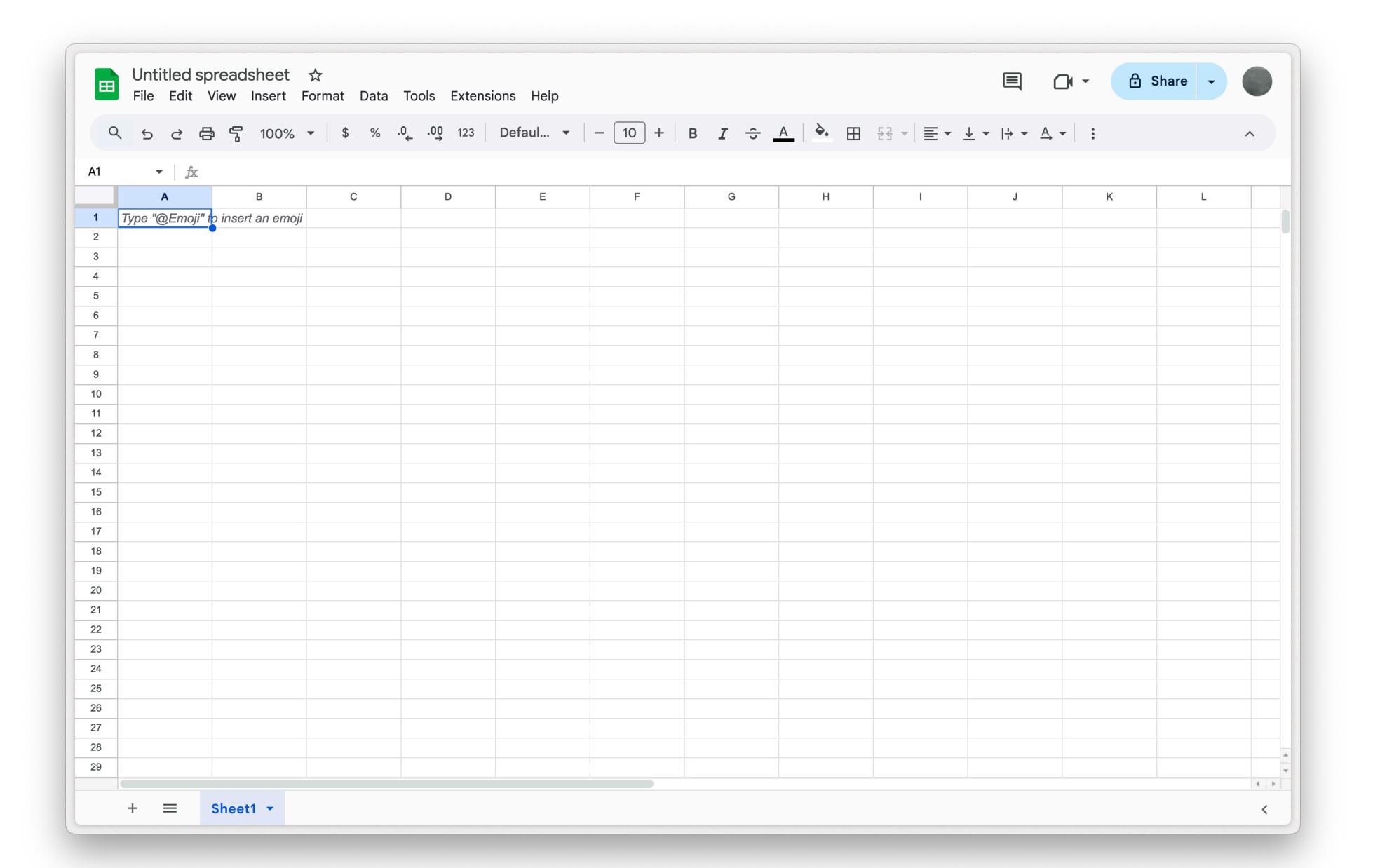


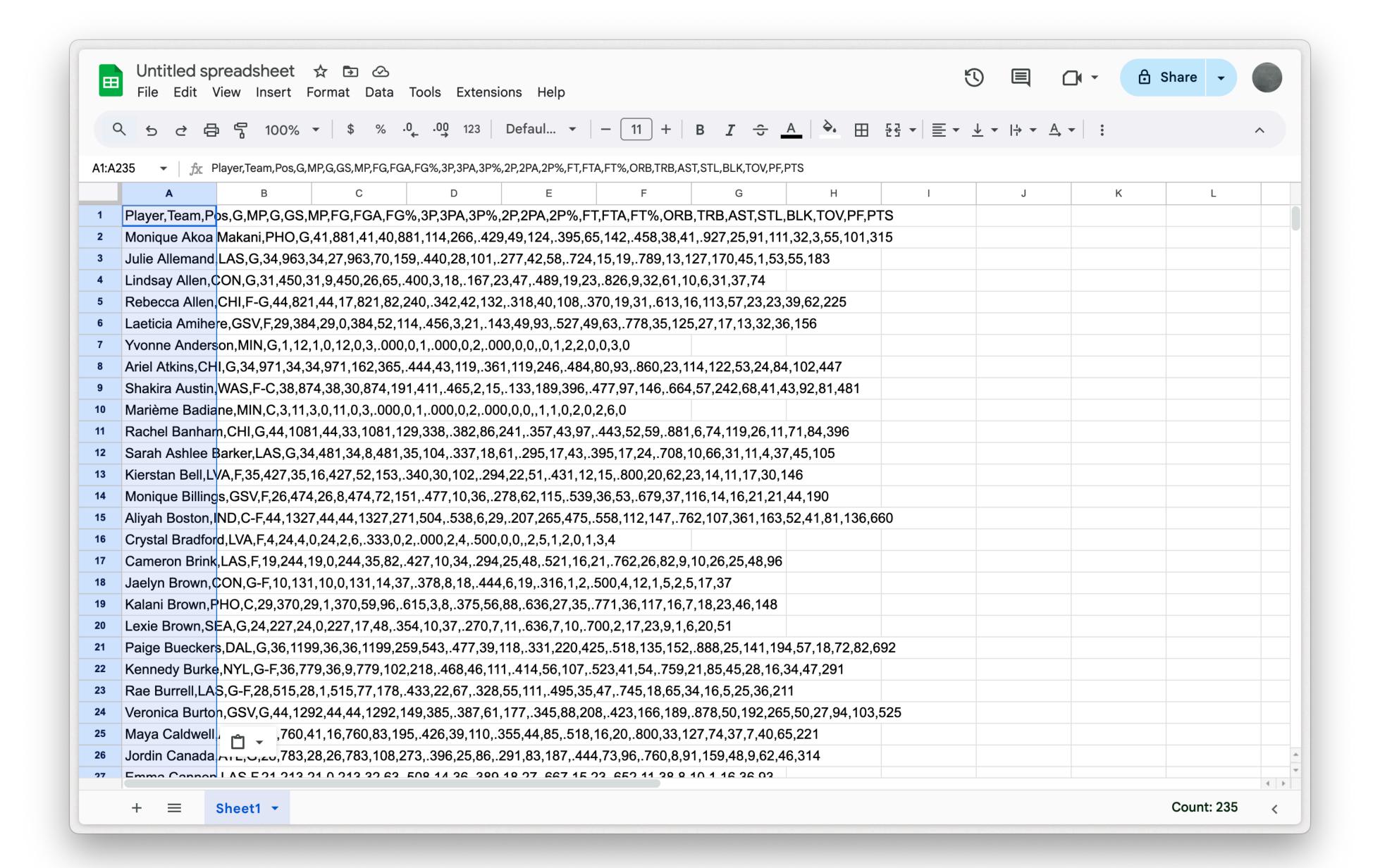


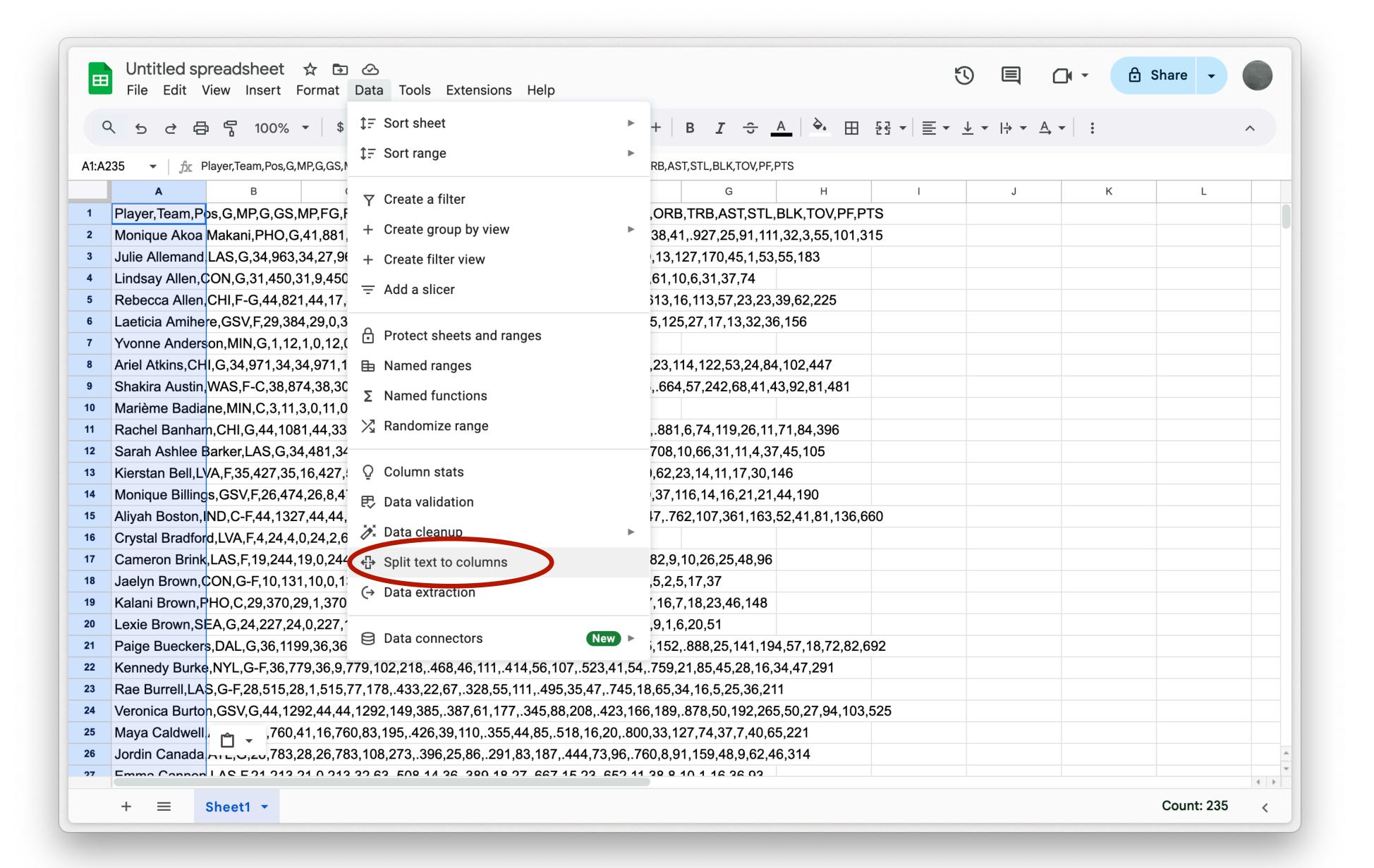


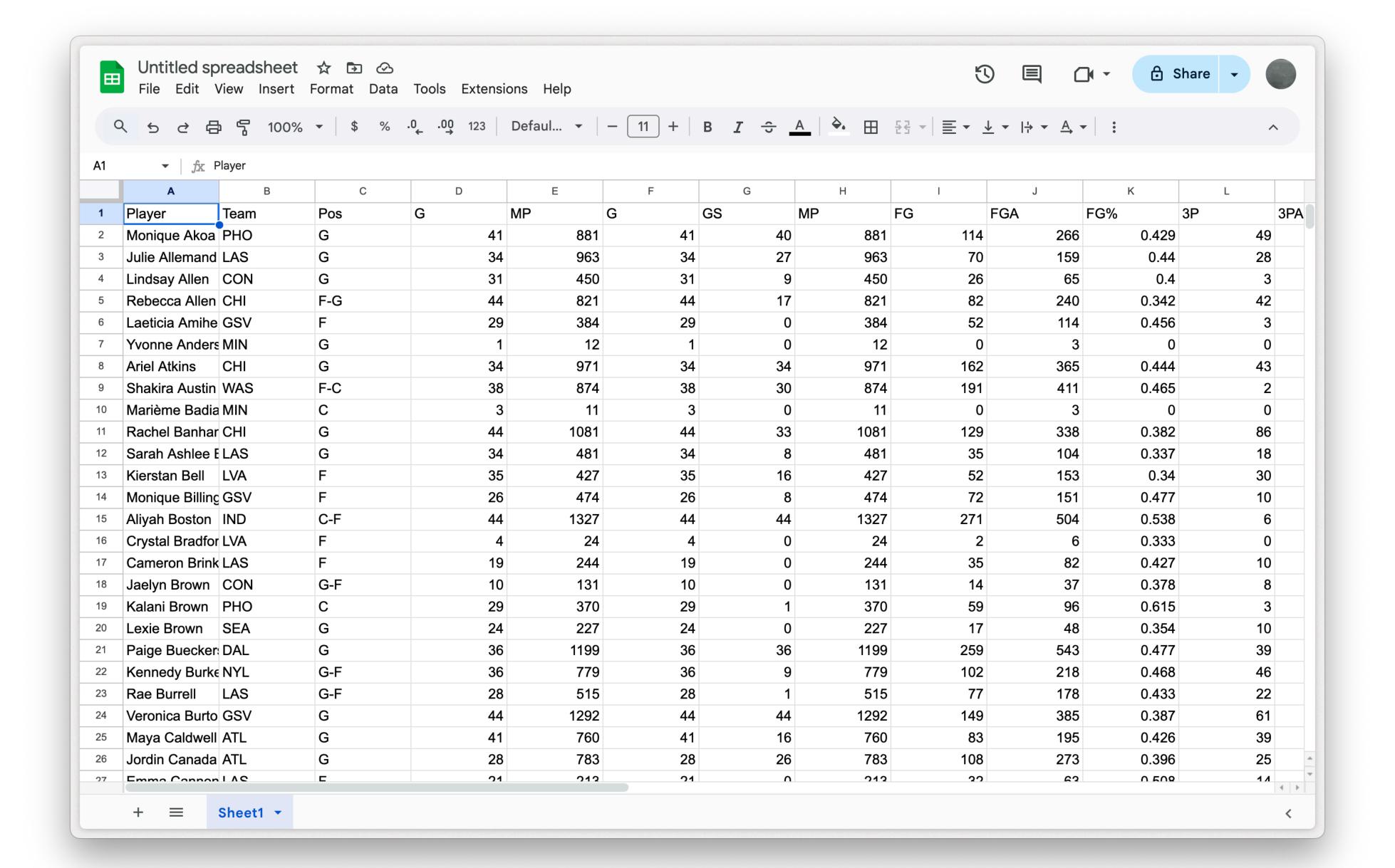


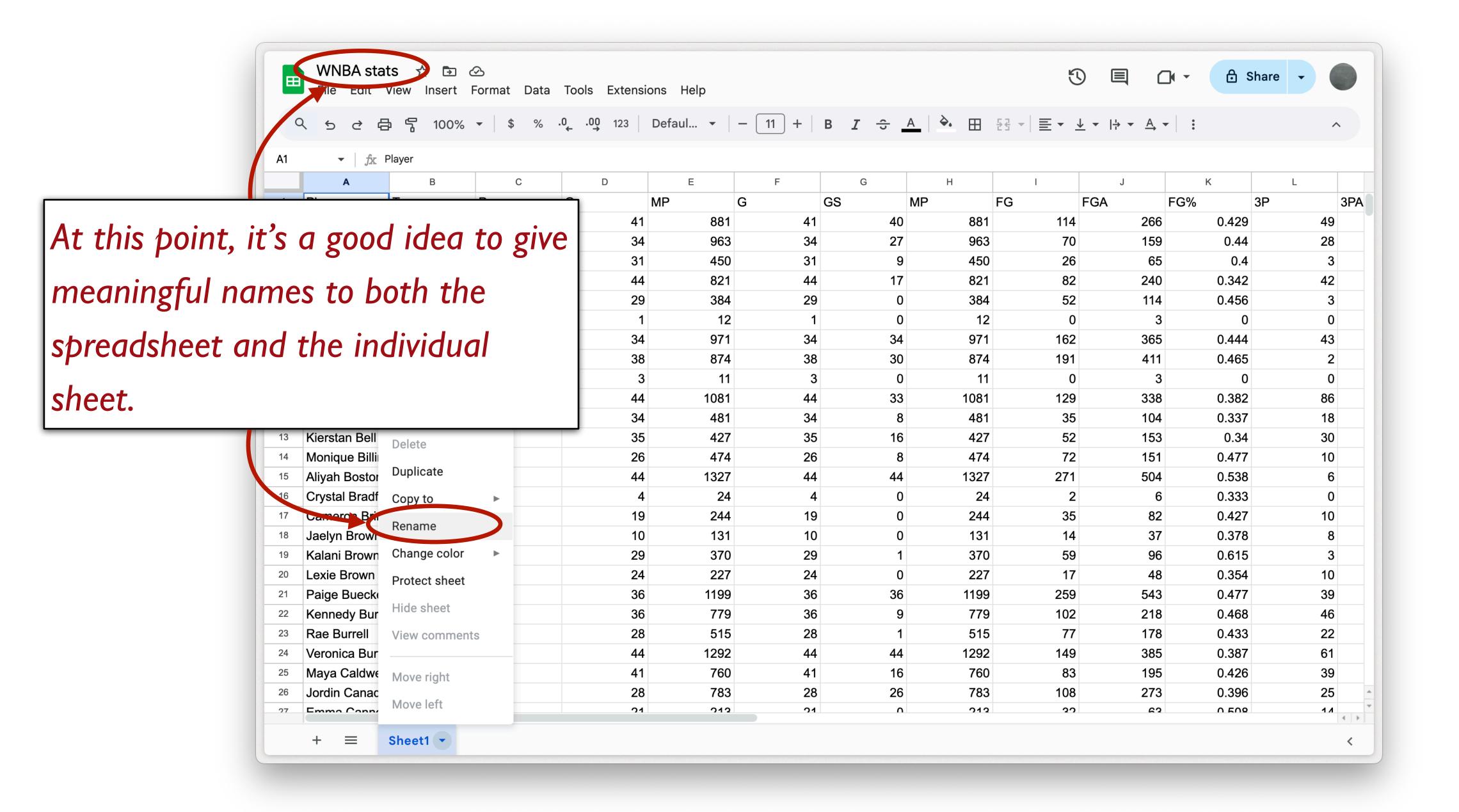
Step 1: Make a spreadsheet





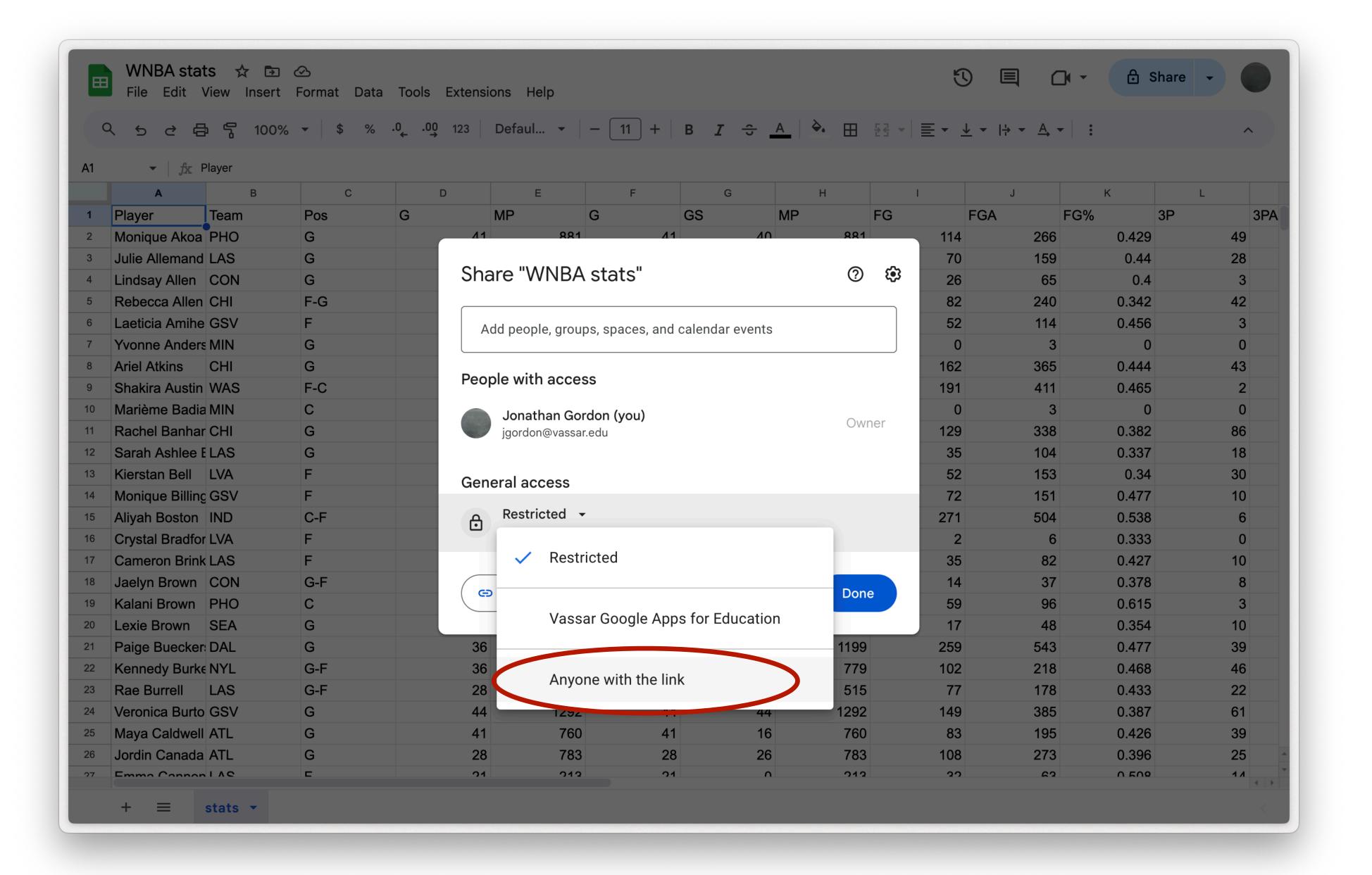






Step 1: Make a spreadsheet

Step 2: Load the spreadsheet as a table



```
from datascience import *
```

```
url = "https://docs.google.com/spreadsheets/d/
1Zmd79B0w0z6Uuov17j0hsmA-nmY2PNatxuWTe1QsSSk/export?
format=csv"
```

stats = Table.read_table(url)

We replaced the edit... part of the URL with export?format=csv

This is a lot of columns!

Player	Team	Pos	G	MP	G.1	GS	MP.1	FG	FGA	FG%	3P	3PA	3 P %	2P	2PA	2 P %	FT	FTA	FT%	ORB	TRB	AST	STL	BLK	TOV	PF	PTS
Monique Akoa Makani	РНО	G	41	881	41	40	881	114	266	0.429	49	124	0.395	65	142	0.458	38	41	0.927	25	91	111	32	3	55	101	315
Julie Allemand	LAS	G	34	963	34	27	963	70	159	0.44	28	101	0.277	42	58	0.724	15	19	0.789	13	127	170	45	1	53	55	183
Lindsay Allen	CON	G	31	450	31	9	450	26	65	0.4	3	18	0.167	23	47	0.489	19	23	0.826	9	32	61	10	6	31	37	74
Rebecca Allen	CHI	F-G	44	821	44	17	821	82	240	0.342	42	132	0.318	40	108	0.37	19	31	0.613	16	113	57	23	23	39	62	225
Laeticia Amihere	GSV	F	29	384	29	0	384	52	114	0.456	3	21	0.143	49	93	0.527	49	63	0.778	35	125	27	17	13	32	36	156
Yvonne Anderson	MIN	G	1	12	1	0	12	0	3	0	0	1	0	0	2	0	0	0	nan	0	1	2	2	0	0	3	0
Ariel Atkins	CHI	G	34	971	34	34	971	162	365	0.444	43	119	0.361	119	246	0.484	80	93	0.86	23	114	122	53	24	84	102	447
Shakira Austin	WAS	F-C	38	874	38	30	874	191	411	0.465	2	15	0.133	189	396	0.477	97	146	0.664	57	242	68	41	43	92	81	481
Marième Badiane	MIN	С	3	11	3	0	11	0	3	0	0	1	0	0	2	0	0	0	nan	1	1	0	2	0	2	6	0
Rachel Banham	CHI	G	44	1081	44	33	1081	129	338	0.382	86	241	0.357	43	97	0.443	52	59	0.881	6	74	119	26	11	71	84	396

Step 1: Make a spreadsheet

Step 2: Load the spreadsheet as a table

Step 3: Rethink that table

Player	Team	Pos	G	MP	G.1	GS	MP.1	FG	FGA	FG%	3P	3PA	3 P %	2P	2PA	2 P %	FT	FTA	FT%	ORB	TRB	AST	STL	BLK	TOV	PF	PTS
Monique Akoa Makani	РНО	G	41	881	41	40	881	114	266	0.429	49	124	0.395	65	142	0.458	38	41	0.927	25	91	111	32	3	55	101	315
Julie Allemand	LAS	G	34	963	34	27	963	70	159	0.44	28	101	0.277	42	58	0.724	15	19	0.789	13	127	170	45	1	53	55	183
Lindsay Allen	CON	G	31	450	31	9	450	26	65	0.4	3	18	0.167	23	47	0.489	19	23	0.826	9	32	61	10	6	31	37	74
Rebecca Allen	СНІ	F-G	44	821	44	17	821	82	240	0.342	42	132	0.318	40	108	0.37	19	31	0.613	16	113	57	23	23	39	62	225
Laeticia Amihere	GSV	F	29	384	29	0	384	52	114	0.456	3	21	0.143	49	93	0.527	49	63	0.778	35	125	27	17	13	32	36	156
Yvonne Anderson	MIN	G	1	12	1	0	12	0	3	0	0	1	0	0	2	0	0	0	nan	0	1	2	2	0	0	3	0
Ariel Atkins	СНІ	G	34	971	34	34	971	162	365	0.444	43	119	0.361	119	246	0.484	80	93	0.86	23	114	122	53	24	84	102	447
Shakira Austin	WAS	F-C	38	874	38	30	874	191	411	0.465	2	15	0.133	189	396	0.477	97	146	0.664	57	242	68	41	43	92	81	481
Marième Badiane	MIN	С	3	11	3	0	11	0	3	0	0	1	0	0	2	0	0	0	nan	1	1	0	2	0	2	6	0
Rachel Banham	СНІ	G	44	1081	44	33	1081	129	338	0.382	86	241	0.357	43	97	0.443	52	59	0.881	6	74	119	26	11	71	84	396

```
stats = stats.select(
    "Player", "Team", "Pos", "G", "PTS"
)
```

stats

Player	Team	Pos	G	PTS
Monique Akoa Makani	PHO	G	41	315
Julie Allemand	LAS	G	34	183
Lindsay Allen	CON	G	31	74
Rebecca Allen	CHI	F-G	44	225
Laeticia Amihere	GSV	F	29	156
Yvonne Anderson	MIN	G	1	0
Ariel Atkins	CHI	G	34	447
Shakira Austin	WAS	F-C	38	481
Marième Badiane	MIN	С	3	0
Rachel Banham	CHI	G	44	396

Player	Team	Pos	Games	Pts
Monique Akoa Makani	PHO	G	41	315
Julie Allemand	LAS	G	34	183
Lindsay Allen	CON	G	31	74
Rebecca Allen	CHI	F-G	44	225
Laeticia Amihere	GSV	F	29	156
Yvonne Anderson	MIN	G	1	0
Ariel Atkins	CHI	G	34	447
Shakira Austin	WAS	F-C	38	481
Marième Badiane	MIN	С	3	0
Rachel Banham	CHI	G	44	396

stats

Player	Team	Pos	Games	Pts
Monique Akoa Makani	PHO	G	41	315
Julie Allemand	LAS	G	34	183
Lindsay Allen	CON	G	31	74
Rebecca Allen	CHI	F-G	44	225
Laeticia Amihere	GSV	F	29	156
Yvonne Anderson	MIN	G	1	0
Ariel Atkins	CHI	G	34	447
Shakira Austin	WAS	F-C	38	481
Marième Badiane	MIN	С	3	0
Rachel Banham	CHI	G	44	396

Where are we?

Player	Team	Pos	Games	Pts
Monique Akoa Makani	РНО	G	41	315
Julie Allemand	LAS	G	34	183
Lindsay Allen	CON	G	31	74
Rebecca Allen	CHI	F-G	44	225
Laeticia Amihere	GSV	F	29	156
Yvonne Anderson	MIN	G	1	0
Ariel Atkins	CHI	G	34	447
Shakira Austin	WAS	F-C	38	481
Marième Badiane	MIN	С	3	0
Rachel Banham	CHI	G	44	396

... (224 rows omitted)

Rows!

Player	Team	Pos	Games	Pts
Monique Akoa Makani	PHO	G	41	315
Julie Allemand	LAS	G	34	183
Lindsay Allen	CON	G	31	74
Rebecca Allen	CHI	F-G	44	225
Laeticia Amihere	GSV	F	29	156
Yvonne Anderson	MIN	G	1	0
Ariel Atkins	CHI	G	34	447
Shakira Austin	WAS	F-C	38	481
Marième Badiane	MIN	С	3	0
Rachel Banham	CHI	G	44	396

How do I get just this row from stats?

n	Player	Team	Pos	Games	Pts
0	Monique Akoa Makani	PHO	G	41	315
1	Julie Allemand	LAS	G	34	183
2	Lindsay Allen	CON	G	31	74
3	Rebecca Allen	CHI	F-G	44	225
4	Laeticia Amihere	GSV	F	29	156
5	Yvonne Anderson	MIN	G	1	0
6	Ariel Atkins	CHI	G	34	447
7	Shakira Austin	WAS	F-C	38	481
8	Marième Badiane	MIN	С	3	0
9	Rachel Banham	CHI	G	44	396
	(224 rows omitted)				

How do I get just this row from stats?

n	Player	Team	Pos	Games	Pts
0	Monique Akoa Makani	РНО	G	41	315
1	Julie Allemand	LAS	G	34	183
2	Lindsay Allen	CON	G	31	74
3	Rebecca Allen	CHI	F-G	44	225
4	Laeticia Amihere	GSV	F	29	156
5	Yvonne Anderson	MIN	G	1	0
6	Ariel Atkins	CHI	G	34	447
7	Shakira Austin	WAS	F-C	38	481
8	Marième Badiane	MIN	С	3	0
9	Rachel Banham	CHI	G	44	396
	(224 rows omitted)				

How do I get just this row from stats?

stats.rows[3]

Player	Team	Pos	Games	Pts
Monique Akoa Makani	РНО	G	41	315
Julie Allemand	LAS	G	34	183
Lindsay Allen	CON	G	31	74
Rebecca Allen	CHI	F-G	44	225
Laeticia Amihere	GSV	F	29	156
Yvonne Anderson	MIN	G	1	0
Ariel Atkins	CHI	G	34	447
Shakira Austin	WAS	F-C	38	481
Marième Badiane	MIN	С	3	0
Rachel Banham	CHI	G	44	396

How do I get just the rows for players who are guards?

Team Pos Games Pts Player Monique Akoa Makani PHO G 41 315 LAS Julie Allemand G 34 183 CON Lindsay Allen G 31 74 Rebecca Allen CHI F-G 44 225 Laeticia Amihere GSV F 29 156 G Yvonne Anderson MIN Ariel Atkins CHI G 34 447 Shakira Austin WAS F-C 38 481

How do I get just the rows for players who are guards?

stats =

stats.where("Pos", are.containing("G"))

Table.where Predicates

All of the predicates described below can be negated by preceding the name with not_. For example, we can find values *not* equal to a specific value using are.not_equal_to(value).

are.equal_to()

tbl.where(column, are.equal_to(value))

Filter leaves rows only where the value in column is equal to value.

[166]: trips.where("Duration", are.equal_to(519))

[166]:

Trip I	D Dui	ration	Start Date	Start Station	Start Terminal	End Date	End Station	End Terminal	Bike #	Subscriber Type	Zip Code
72335	62	519	4/13/2015 17:04	Howard at 2nd	63	4/13/2015 17:12	San Francisco Caltrain (Townsend at 4th)	70	629	Subscriber	94061
82497	'9	519	6/27/2015 15:02	Japantown	9	6/27/2015 15:11	San Jose City Hall	10	660	Customer	nil
43994	-6	519	9/5/2014 12:38	Yerba Buena Center of the Arts (3rd @ Howard)	68	9/5/2014 12:47	Civic Center BART (7th at Market)	72	452	Subscriber	94105
78826	i 1	519	6/1/2015 9:21	Powell at Post (Union Square)	71	6/1/2015 9:30	Steuart at Market	74	575	Subscriber	94108
56047	'9	519	11/28/2014 14:20	South Van Ness at Market	66	11/28/2014 14:29	Powell at Post (Union Square)	71	609	Subscriber	94108
65379	7	519	2/23/2015 8:47	Market at 10th	67	2/23/2015 8:55	Yerba Buena Center of the Arts (3rd @ Howard)	68	546	Subscriber	94102
88713	4	519	8/12/2015 17:29	Civic Center BART (7th at Market)	72	8/12/2015 17:38	Mechanics Plaza (Market at Battery)	75	313	Subscriber	94103
48222	.5	519	10/3/2014 16:41	Spear at Folsom	49	10/3/2014 16:50	Broadway St at Battery St	82	209	Subscriber	94133

```
def is_guard(pos: str) -> bool:
    """Return True if the player is any kind of guard,
    e.g., "G" (guard) or "C-G" (center guard).
    11 11 11
    return "G" in pos
assert is_guard("G") == True
assert is_guard("C-G") == True
assert is_guard("F") == False
```

stats.where("Pos", is_guard)

Player	Team	Pos	Games	Pts
Monique Akoa Makani	PHO	G	41	315
Julie Allemand	LAS	G	34	183
Lindsay Allen	CON	G	31	74
Rebecca Allen	CHI	F-G	44	225
Laeticia Amihere	GSV	F	29	156
Yvonne Anderson	MIN	G	1	0
Ariel Atkins	CHI	G	34	447
Shakira Austin	WAS	F-C	38	481
Marième Badiane	MIN	С	3	0
Rachel Banham	CHI	G	44	396

... (224 rows omitted)

Team	Pos	Games	Pts
PHO	G	41	315
LAS	G	34	183
CON	G	31	74
СНІ	F-G	44	225
GSV	F	29	156
MIN	G	1	0
СНІ	G	34	447
WAS	F-C	38	481
MIN	С	3	0
СНІ	G	44	396
	PHO LAS CON GSV MIN CHI WAS MIN	PHO G LAS G CON G GSV F MIN G WAS F-C MIN C	PHO G 41 LAS G 34 CON G 31 CHI F-G 44 GSV F 29 MIN G 1 CHI G 34 WAS F-C 38 MIN C 3

What about columns?

... (224 rows omitted)

Player	Team	Pos	Games	Pts
Monique Akoa Makani	PHO	G	41	315
Julie Allemand	LAS	G	34	183
Lindsay Allen	CON	G	31	74
Rebecca Allen	CHI	F-G	44	225
Laeticia Amihere	GSV	F	29	156
Yvonne Anderson	MIN	G	1	0
Ariel Atkins	CHI	G	34	447
Shakira Austin	WAS	F-C	38	481

How do I get just the points column?

stats =

```
stats["Pts"]

→ array([228, 133, 17, 532, ...])
```

```
stats["Pts"]

→ array([228, 133, 17, 532, ...])

The data type isn't Column; it's an array!
```

Exercise: Who scores the most points per game?

See notebook for example.

Changing a column

So, we've seen that we can build a new column based on the values in each row, but what if we just want to change an existing column?

Player	Team	Pos	Games	Pts
Monique Akoa Makani	РНО	G	41	315
Julie Allemand	LAS	G	34	183
Lindsay Allen	CON	G	31	74
Rebecca Allen	CHI	F-G	44	225
Laeticia Amihere	GSV	F	29	156
Yvonne Anderson	MIN	G	1	0
Ariel Atkins	CHI	G	34	447
Shakira Austin	WAS	F-C	38	481
Marième Badiane	MIN	С	3	0
Rachel Banham	CHI	G	44	396

A fake WNBA fan like me can't remember what these team abbreviations stand for.

Let's fill in the actual team names.

... (224 rows omitted)

What are the team names?











WNBA Home Scores Schedule Standings Stats Teams Odds WNBA Playoffs More v

WNBA Teams

Eastern Conference



Atlanta Dream

Statistics | Schedule | Roster | Tickets



Chicago Sky

Statistics | Schedule | Roster | Tickets



Connecticut Sun

Statistics | Schedule | Roster | Tickets



Indiana Fever

Statistics | Schedule | Roster | Tickets



New York Liberty

Statistics | Schedule | Roster | Tickets



Washington Mystics

Statistics | Schedule | Roster | Tickets

Western Conference



Dallas Wings

Statistics | Schedule | Roster | Tickets



Golden State Valkyries

Statistics | Schedule | Roster | Tickets



Las Vegas Aces

Statistics | Schedule | Roster | Tickets



Los Angeles Sparks

Statistics | Schedule | Roster | Tickets



Minnesota Lynx

Statistics | Schedule | Roster | Tickets



Phoenix Mercury

Statistics | Schedule | Roster | Tickets



Seattle Storm

Statistics | Schedule | Roster | Tickets

DAL

GSV

LVA

LAS

MIN

PHO

SEA

```
def team_name(abbr: str) -> str:
    """Return the name of the team with the given
    abbreviation
    """
    ...

assert team_name("NYL") == "New York Liberty"
assert team_name("CHI") == "Chicago Sky"
```

```
def team_name(abbr: str) -> str:
    """Return the name of the team with the given
    abbreviation
    11 11 11
    if abbr == "DAL": return "Dallas Wings"
    elif abbr == "LVA": return "Las Vegas Aces"
    . . .
assert team_name("NYL") == "New York Liberty"
assert team_name("CHI") == "Chicago Sky"
```

```
def team_name(abbr: str) -> str:
    """Return the name of the team with the given
    abbreviation
    11 11 11
    if abbr == "DAL": return "Dallas Wings"
    elif abbr == "LVA": return "Las Vegas Aces"
assert team_name("NYL") == "New York Liberty"
assert team_name("CHI") == "Chicago Sky"
```

This will work, but remember what we said when we introduced tables for looking up population: We want to separate data from computation.

```
teams = Table().with_columns(
    "Abbr",
    ["DAL", "GSV", ...],
    "Name",
    ["Dallas Wings", "Golden State Valkyries", ...]
)
```

Advantage: This makes it easy to add new teams or more information about these teams, in a central place.

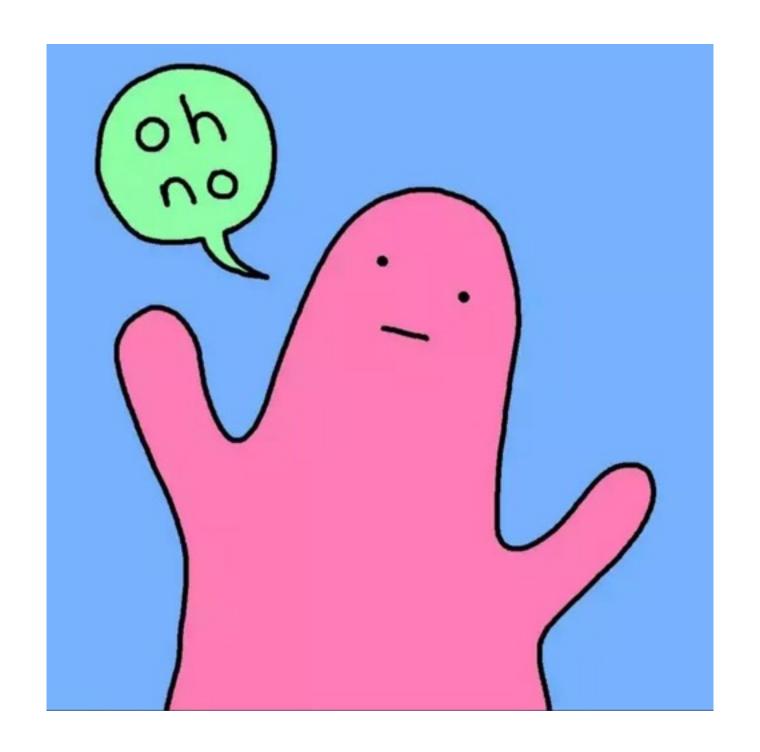
```
def team_name(abbr: str) -> str:
    """Return the name of the team with the given abbreviation"""
    # 1. Get the row with abbreviation `abbr`
    # 2. Return the value in the `name` column

assert team_name("NYL") == "New York Liberty"
assert team_name("CHI") == "Chicago Sky"
...
```

```
teams = ...
def team_name(abbr: str) -> str:
    """Return the name of the team with the given abbreviation"""
   matches = teams.where("Abbr", abbr)
    team = matches.rows[0]
   # 2. Return the value in the `name` column
assert team_name("NYL") == "New York Liberty"
assert team_name("CHI") == "Chicago Sky"
```

```
teams = ...
def team_name(abbr: str) -> str:
    """Return the name of the team with the given abbreviation"""
   matches = teams.where("Abbr", abbr)
    team = matches.rows[0]
    return team.name
assert team_name("NYL") == "New York Liberty"
assert team_name("CHI") == "Chicago Sky"
```

IndexError: index 0 is out of bounds for axis 0 with size 0



```
def team_name(abbr: str) -> str:
    """Return the name of the team with the given abbreviation"""

matches = teams.where("Abbr", abbr)
    team = matches.rows[0]
    return team.name

assert team_name("NYL") == "New York Liberty"
assert team_name("CHI") == "Chicago Sky"
O is too big
```

0 is too big? That means there were no matching rows! An abbreviation not in our table – what is it?

```
teams = ...
def team_name(abbr: str) -> str:
    """Return the name of the team with the given abbreviation"""
   if abbr == "TOT":
        return "Total"
   matches = teams.where("Abbr", abbr)
    team = matches.rows[0]
    return team.name
assert team_name("NYL") == "New York Liberty"
assert team_name("CHI") == "Chicago Sky"
```

```
teams = ...
def team_name(abbr: str) -> str:
    """Return the name of the team with the given abbreviation"""
   if abbr == "TOT":
        return "Total"
   matches = teams.where("Abbr", abbr)
    team = matches.rows[0]
    return team.name
assert team_name("NYL") == "New York Liberty"
assert team_name("CHI") == "Chicago Sky"
stats = stats.with_columns(
    "Team",
    stats.apply(team_name, "Team")
```





Acknowledgments

This class incorporates material from:

Greg Daniels & Michael Schur, Parks and Recreation

Kathi Fisler and colleagues, Brown University

Data 6, University of California, Berkeley (CC BY-NC-SA)

basketball-reference.com

Back to the Future Part II

