21 Dec 2018

# Web scraping NBA data into Stata

- Author: Kevin Crow, Senior Software Developer   
- https://blog.stata.com/2018/10/10/web-scraping-nba-data-into-stata/

This is a replication of a post from Stata blog to test MarkDoc package. Unfortunately, the example was not replicable on Stata 15.1 and Windows 10. Though, this post is based on a user-written program.

Since our intern, Chris Hassell, finished **nfl2stata** earlier than expected, he went ahead and created another command to web scrape https://stats.nba.com for data on the NBA. The command is **nba2stata.** To install the command type

**. net install http://www.stata.com/users/kcrow/nba2stata, replace**  
 checking nba2stata consistency and verifying not already installed...  
 all files already exist and are up to date.

When Chris first wrote the command, I knew I wanted to look at how the three-point shot has changed the way the game is played. For example, I can find the best three-point shooter from last season.

**. nba2stata playerstats \_all, season(2017) seasontype(reg) stat(season) clear**  
 Processing x/548 requests  
  
 .........x.........x.........x.........x.........50  
 .........x.........x.........x.........x.........100  
 .........x.........x.........x.........x.........150  
 .........x.........x.........x.........x.........200  
 .........x.........x.........x.........x.........250  
 .........x.........x.........x.........x.........300  
 .........x.........x.........x.........x.........350  
 .........x.........x.........x.........x.........400  
 .........x.........x.........x.........x.........450  
 .........x.........x.........x.........x.........500  
 .........x.........x.........x.........x........  
 664 observation(s) loaded

**. gsort -threepointfieldgoalsmade**  
  
 **. list playername teamname threepointfieldgoalsmade in 1/10**  
  
 +----------------------------------------------------+  
 | playername teamname three~de |  
 |----------------------------------------------------|  
 1. | James Harden Houston Rockets 265 |  
 2. | Paul George Oklahoma City Thunder 244 |  
 3. | Kyle Lowry Toronto Raptors 238 |  
 4. | Kemba Walker Charlotte Hornets 231 |  
 5. | Klay Thompson Golden State Warriors 229 |  
 |----------------------------------------------------|  
 6. | Wayne Ellington Miami Heat 227 |  
 7. | Damian Lillard Portland Trailblazers 227 |  
 8. | Eric Gordon Houston Rockets 218 |  
 9. | Stephen Curry Golden State Warriors 212 |  
 10. | Joe Ingles Utah Jazz 204 |  
 +----------------------------------------------------+

Or I can check a player’s regular-season three-point percentage for the last five years.

**. nba2stata playerstat "Dirk", stat(season) seasontype(reg) clear**  
 28 observation(s) loaded  
  
 **. gsort -playerage**   
  
 **. list playername playerage threepointfieldgoalpercentage in 1/5**  
  
 +-------------------------------------+  
 | playername playe~ge three~ge |  
 |-------------------------------------|  
 1. | Dirk Nowitzki 40 .409 |  
 2. | Dirk Nowitzki 40 .25 |  
 3. | Dirk Nowitzki 39 .378 |  
 4. | Dirk Nowitzki 38 .368 |  
 5. | Dirk Nowitzki 37 .38 |  
 +-------------------------------------+

Or I can see how three-point percentage affects your favorite team’s chance of winning.

**. nba2stata teamstats "HOU", season(2017) stat(game) seasontype(reg) clear**  
 82 observation(s) loaded  
  
 **. keep if threepointfieldgoalpercentage > .35**  
 (35 observations deleted)  
  
 **. tab winloss**  
  
 Win / loss | Freq. Percent Cum.  
 ------------+-----------------------------------  
 L | 4 8.51 8.51  
 W | 43 91.49 100.00  
 ------------+-----------------------------------  
 Total | 47 100.00

**nba2stata** is great if you are planning on doing pro basketball analysis. Although this command looks identical to **nfl2stata,** it is not. The command works quite differently.

## Web scraping JSON

In our last blog post, we talked about web scraping the https://www.nfl.com and extracting the data from the HTML pages. The NBA data are different. You can access the data via JSON objects from https://stats.nba.com. JSON is a lightweight data format. This data format is easy to parse; therefore, we don’t have a scrape command for these data. We scrape and load these data on the fly.

The NBA’s copyright is similar to that of the NFL; you can use a personal copy of the data on your own personal computer. If you “use, display or publish” anything using these data, you must include “a prominent attribution to http://www.nba.com“. Another difference is that the NBA data stored on http://stats.nba.com can go as far back as the 1960s, depending on the team.

### Command

There are only four subcommands to nba2stata, though we could have developed more. Chris had to go back to school.

- To scrape player statistics data into Stata, use

nba2stata playerstats name\_pattern [, playerstats\_options]

- To scrape player profile data into Stata, use

nba2stata playerprofile name\_pattern [, playerprofile\_options]

- To scrape team statistics data into Stata, use

nba2stata teamstats team\_adv [, teamstats\_options]

- To scrape team roster data into Stata, use

nba2stata teamroster team\_adv [, teamroster\_options]

Just like with nfl2stata, you will need to use Stata commands like collapse, gsort, and merge to generate the statistics, sort the data, and merge two or more NBA datasets together to examine the data.

#### Examples

One thing I’m always curious about is which college teams produce the most NBA players. This is easy to find out using **nba2stata, collapse,** and **gsort.**

**. nba2stata playerprofile "\_all", clear**  
 Processing x/4351 requests  
  
 .........x.........x.........x.........x.........50  
 .........x.........x.........x.........x.........100  
 .........x.........x.........x.........x.........150  
 .........x.........x.........x.........x.........200  
 .........x.........x.........x.........x.........250  
 .........x.........x.........x.........x.........300  
 .........x.........x.........x.........x.........350  
 .........x.........x.........x.........x.........400  
 .........x.........x.........x.........x.........450  
 .........x.........x.........x.........x.........500  
 .........x.........x.........x.........x.........550  
 .........x.........x.........x.........x.........600  
 .........x.........x.........x.........x.........650  
 .........x.........x.........x.........x.........700  
 .........x.........x.........x.........x.........750  
 .........x.........x..java.lang.reflect.InvocationTargetException  
 at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
 at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)  
 at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)  
 at java.lang.reflect.Method.invoke(Method.java:498)  
 at com.stata.Javacall.load(Javacall.java:130)  
 at com.stata.Javacall.load(Javacall.java:90)  
 Caused by: java.lang.NumberFormatException: For input string: "N"  
 at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)  
 at java.lang.Integer.parseInt(Integer.java:580)  
 at java.lang.Integer.parseInt(Integer.java:615)  
 at com.google.gson.JsonPrimitive.getAsInt(JsonPrimitive.java:264)  
 at com.stata.chassell.deserializers.PlayerProfileRowSetDeserializer.deserialize(PlayerProfileRowSetDeserializer.java:43)  
 at com.stata.chassell.deserializers.PlayerProfileRowSetDeserializer.deserialize(PlayerProfileRowSetDeserializer.java:1)  
 at com.google.gson.internal.bind.TreeTypeAdapter.read(TreeTypeAdapter.java:69)  
 at com.google.gson.internal.bind.TypeAdapterRuntimeTypeWrapper.read(TypeAdapterRuntimeTypeWrapper.java:41)  
 at com.google.gson.internal.bind.ArrayTypeAdapter.read(ArrayTypeAdapter.java:72)  
 at com.google.gson.Gson.fromJson(Gson.java:927)  
 at com.google.gson.Gson.fromJson(Gson.java:994)  
 at com.google.gson.internal.bind.TreeTypeAdapter.deserialize(TreeTypeAdapter.java:162)  
 at com.stata.chassell.deserializers.PlayerProfileResultSetDeserializer.deserialize(PlayerProfileResultSetDeserializer.java:24)  
 at com.stata.chassell.deserializers.PlayerProfileResultSetDeserializer.deserialize(PlayerProfileResultSetDeserializer.java:1)  
 at com.google.gson.internal.bind.TreeTypeAdapter.read(TreeTypeAdapter.java:69)  
 at com.google.gson.internal.bind.TypeAdapterRuntimeTypeWrapper.read(TypeAdapterRuntimeTypeWrapper.java:41)  
 at com.google.gson.internal.bind.ArrayTypeAdapter.read(ArrayTypeAdapter.java:72)  
 at com.google.gson.Gson.fromJson(Gson.java:927)  
 at com.google.gson.Gson.fromJson(Gson.java:994)  
 at com.google.gson.internal.bind.TreeTypeAdapter.deserialize(TreeTypeAdapter.java:162)  
 at com.stata.chassell.deserializers.PlayerProfileResponseDeserializer.deserialize(PlayerProfileResponseDeserializer.java:20)  
 at com.stata.chassell.deserializers.PlayerProfileResponseDeserializer.deserialize(PlayerProfileResponseDeserializer.java:1)  
 at com.google.gson.internal.bind.TreeTypeAdapter.read(TreeTypeAdapter.java:69)  
 at com.google.gson.Gson.fromJson(Gson.java:927)  
 at com.google.gson.Gson.fromJson(Gson.java:892)  
 at com.google.gson.Gson.fromJson(Gson.java:841)  
 at com.google.gson.Gson.fromJson(Gson.java:813)  
 at com.stata.chassell.NBA2Stata.getPlayerProfile(NBA2Stata.java:1781)  
 at com.stata.chassell.NBA2Stata.lambda$8(NBA2Stata.java:2729)  
 at java.util.stream.ForEachOps.accept(ForEachOps.java:184)  
 at java.util.stream.ReferencePipeline$2$1.accept(ReferencePipeline.java:175)  
 at java.util.ArrayList.forEachRemaining(ArrayList.java:1382)  
 at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:481)  
 at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:471)  
 at java.util.stream.ForEachOps.evaluateSequential(ForEachOps.java:151)  
 at java.util.stream.ForEachOps.evaluateSequential(ForEachOps.java:174)  
 at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)  
 at java.util.stream.ReferencePipeline.forEach(ReferencePipeline.java:418)  
 at com.stata.chassell.NBA2Stata.doCommand(NBA2Stata.java:2726)  
 ... 6 more  
 r(5100);  
  
 end of do-file  
 r(5100);  
  
 end of do-file  
  
 r(5100);