Anton’s scraping magic

1. Load libraries
2. Regex\_list.R

Regular expressions are ways to recognize patterns in text.

Not a universal language, but very similar.

Line 10 in regex\_list.R – here’s an example of how I found the patterns I wanted to find.

* Finding ways to recognize the event name and the score associated with that event

1. Get\_event\_funcs.R

All the functions he’ll be using in the larger file.

Get scores associated with an event

And if they’re part of a multi, grabbing them in reference to a multi event.

You’ll see this in action.

Switch is kind of like case when then

1. Meets\_one\_athlete.R

Step 1, get url for an athlete

Html elements – only one title, which happens to have the name of the athlete in it.

Line 20-21, throw away the stuff before the name, and the stuff after the name.

* Positive look behind. Grab everything that comes after space | space
* Grab everything that comes before space hyphen space

In the source course, each table represents a meet, but the formatting works best if you specifically look for new line characters and tab characters

* \*howd you know that the formatting works best in this way
* All of the text representing the first meet has title, date, event name, points score, places, sub events, and scores
* Loop through this, and use all of the regular expressions to isolate where the event name starts and grab all the information after that, and then figure out where the score is.

Browser function allows you to step through functions, super helpful when debugging.

* Detect multi is whether the event was either a dec hep or pent. Does one appear? Then, which one appeared?
* Create an empty vector with the name of the multi event.

Now that we know it’s a dec, we need to grab all the events within that meet.

We know the events within a dec, so we can parse for each of those one by one using the regex functions, hitting the html text.

Once you grab a score, chop it from the text, for the next iteration of the search.

If its not part of a multi, then grab the event name and the performance.

\*\* this code doesn’t grab both prelim and finals because it nulls out the regex name for an event once its grabbed … this is okay because finals are always listed first in the text so that’s what we’re grabbing