Assignment 4 Job Postings

STA141B Fall 2020

Due: December 2, 5pm

The task here is to scrape job postings related to the search terms statistician, data scientist, data analyst, etc. from one or more Web sites so that you can then address questions such as

- how much experience is needed?
- what are required skills?
- what types of companies are these jobs in (e.g., large, established companies, startups)?
- which industries are the jobs in?
- what's the salary distribution?
- where are the jobs located?
- how does salary relate to location?
- how many posts are there on each site for different queries?

The job posting Web sites include

- cybercoders.com
- indeed.com
- monster.com
- careerbuilder.com

You are to collect the job posts programmatically.

For each post, extract information for

- required skills
- preferred skills
- salary, if available
- education level required or preferred
- degree fields/subjects mentioned
- location (city, state)
- full, part-time, hourly, contract, or short-term
- the free form text describing the position

Collect this information for the queries + statistician, + data scientist, + data analyst + and other terms that interest you

How many postings are available for each query on each site?

Summarize these data and the differences across the title queries/descriptions.

Approach

A common approach to scraping data that come from queries that may span multiple pages of results is to

- make the initial query and parse the resulting HTML document
- get the list of nodes for each job posting by finding the identifying HTML pattern and XPath query
- process each job posting node, both the information in the result page and the full job posting that may be in a related link
  - the short summary and the full posting may have related by slightly different information or some information may be easier to access in one version.
- get the link to the next page of the results
- loop over these next pages, processing the job postings on each page, appending to them to those from earlier pages.

So there is an inherent loop and the i-th iteration depends on the previous iteration as we have to find the next page.

Useful Packages and Functions

Packages: XML, xml2, RCurl, curl, rvest, ...

Functions from RCurl: getForm(), getURLContent(). You can also use download.files(), readLines() as you do not need to use cookies or login information for the session.

Functions from XML:

- htmlParse() for parsing an HTML document into a tree
- xmlName(), xmlAttrs, xmlGetAttr() for manipulating HTML/XML nodes
- getNodeSet(), xpathSApply()/xpathApply() for performing XPath queries on a (sub) tree
- getRelativeURL() for combining one relative URL and expanding it to a full URL relative to some initial URL.

XPath Information

- https://devhints.io/xpath
- https://www.softwaretestinghelp.com/xpath-writing-cheat-sheet-tutorial-examples/
- Chapters 2 and 3 of XPath and XPointer book. John E. Simpson, O'Reilly.
- XML and Web Technologies for Data Sciences with R, Nolan and Temple Lang. Chapters 3, 4 and 5.