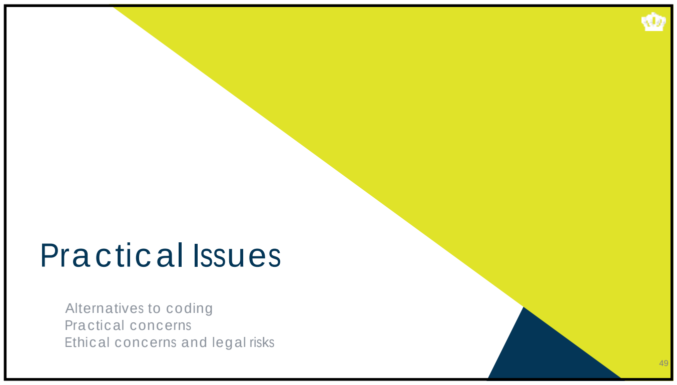
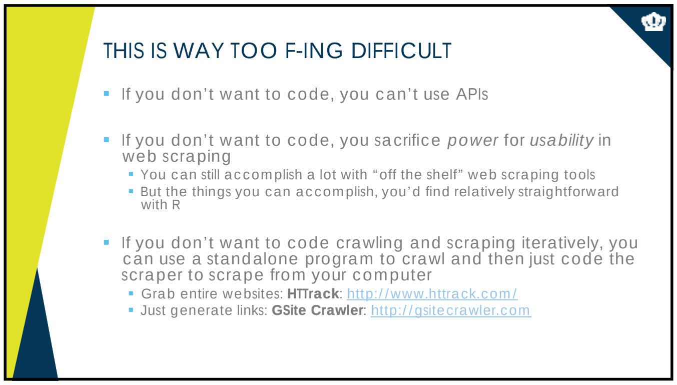


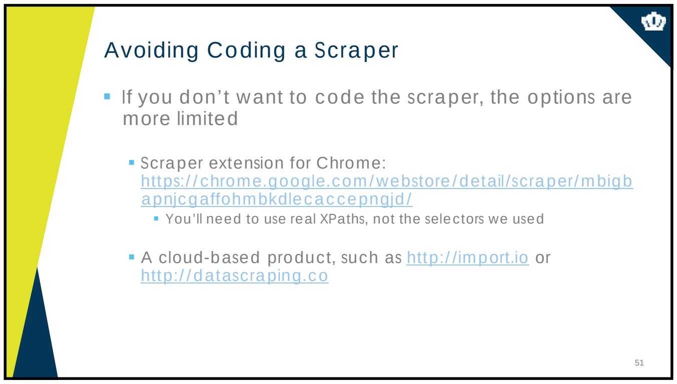
Practical Issues

Alternatives to coding
Practical concerns
Ethical concerns and legal risks



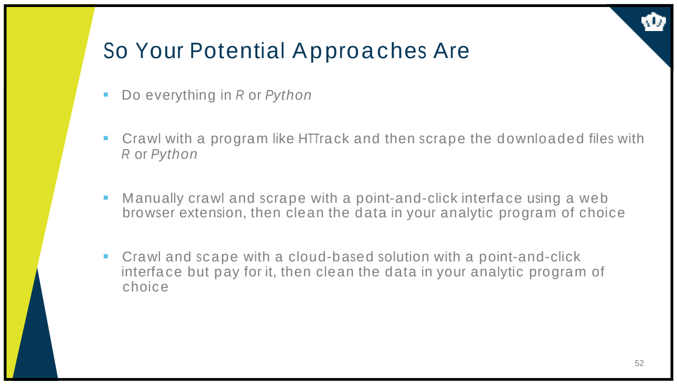
THIS IS WAY TOO F-ING DIFFICULT

- If you don’t want to code, you can’t use APIs
- If you don’t want to code, you sacrifice *power* for *usability* in web scraping
  - You can still accomplish a lot with “off the shelf” web scraping tools
  - But the things you can accomplish, you’d find relatively straightforward with R
- If you don’t want to code crawling and scraping iteratively, you can use a standalone program to crawl and then just code the scraper to scrape from your computer
  - Grab entire websites: **HTTrack**: http://www.httrack.com/
  - Just generate links: **GSite Crawler**: http://gsitecrawler.com



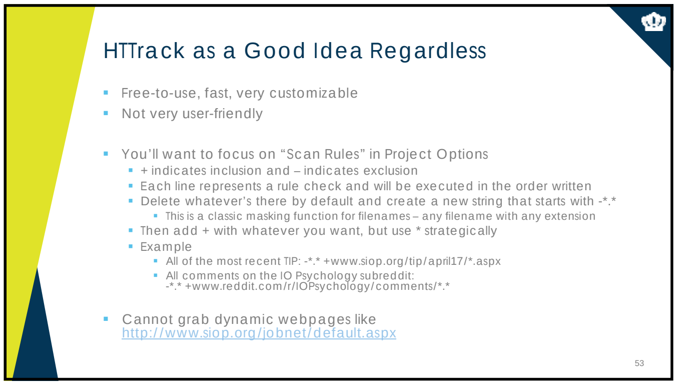
Avoiding Coding a Scraper

- If you don’t want to code the scraper, the options are more limited
  - Scraper extension for Chrome: https://chrome.google.com/webstore/detail/scraper/mbigbapnjcgaffohmbkdlecaccepngjd/
    - You’ll need to use real XPaths, not the selectors we used
  - A cloud-based product, such as http://import.io or http://datascraping.co



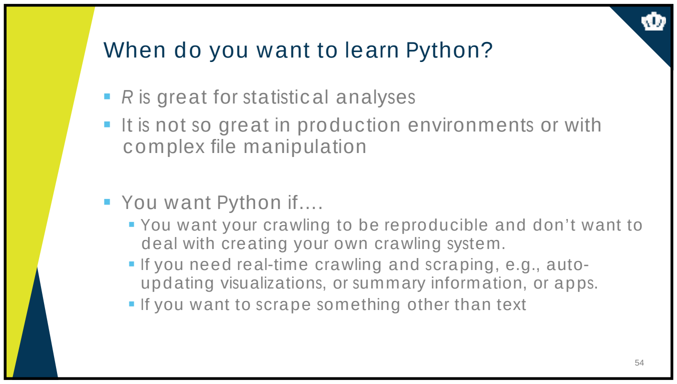
So Your Potential Approaches Are

- Do everything in *R* or *Python*
- Crawl with a program like HTTrack and then scrape the downloaded files with *R* or *Python*
- Manually crawl and scrape with a point-and-click interface using a web browser extension, then clean the data in your analytic program of choice
- Crawl and scape with a cloud-based solution with a point-and-click interface but pay for it, then clean the data in your analytic program of choice



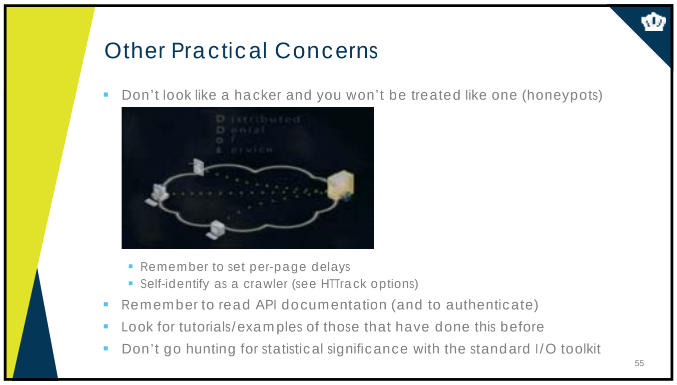
HTTrack as a Good Idea Regardless

- Free-to-use, fast, very customizable
- Not very user-friendly
- You'll want to focus on "Scan Rules" in Project Options
  - + indicates inclusion and – indicates exclusion
  - Each line represents a rule check and will be executed in the order written
  - Delete whatever's there by default and create a new string that starts with -*.*
    - This is a classic masking function for filenames – any filename with any extension
  - Then add + with whatever you want, but use * strategically
  - Example
    - All of the most recent TIP: -*.* +www.siop.org/tip/april17/*.aspx
    - All comments on the IO Psychology subreddit: -*.* +www.reddit.com/r/IOPsychology/comments/*.*
- Cannot grab dynamic webpages like http://www.siop.org/jobnet/default.aspx



When do you want to learn Python?

- *R* is great for statistical analyses
- It is not so great in production environments or with complex file manipulation
- You want Python if….
  - You want your crawling to be reproducible and don't want to deal with creating your own crawling system.
  - If you need real-time crawling and scraping, e.g., auto-updating visualizations, or summary information, or apps.
  - If you want to scrape something other than text

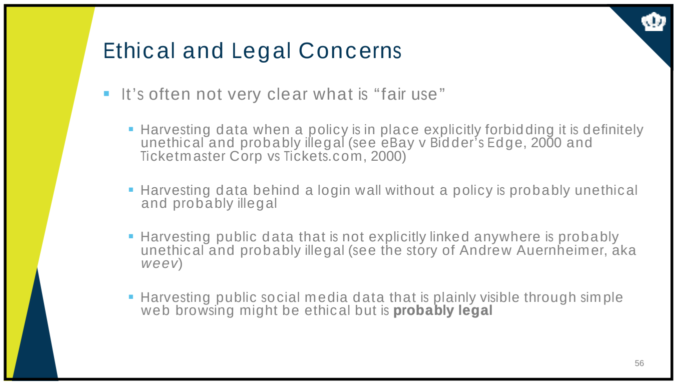


Other Practical Concerns

- Don’t look like a hacker and you won’t be treated like one (honeypots)

D istributed
D enial
o f
S ervice

  - Remember to set per-page delays
  - Self-identify as a crawler (see HTTrack options)
- Remember to read API documentation (and to authenticate)
- Look for tutorials/examples of those that have done this before
- Don’t go hunting for statistical significance with the standard I/O toolkit



Ethical and Legal Concerns

- It’s often not very clear what is “fair use”
  - Harvesting data when a policy is in place explicitly forbidding it is definitely unethical and probably illegal (see eBay v Bidder’s Edge, 2000 and Ticketmaster Corp vs Tickets.com, 2000)
  - Harvesting data behind a login wall without a policy is probably unethical and probably illegal
  - Harvesting public data that is not explicitly linked anywhere is probably unethical and probably illegal (see the story of Andrew Auernheimer, aka *weev*)
  - Harvesting public social media data that is plainly visible through simple web browsing might be ethical but is **probably legal**