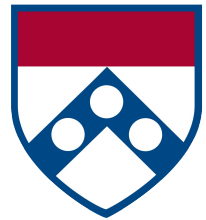
Wharton PhD Tech Camp

Session 5

Alex Miller

Ph.D. Student, Information Systems

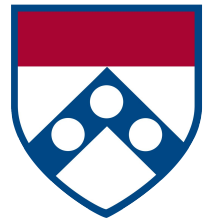
Wharton, OID



Goals for today

- Intro
- Non-scraping
- PyQuery Basics
  - Scraping Walkthrough
- Things to be aware of when scraping
- Lab Exercise

Web Scraping Intro



Scraping 101

- What is scraping?
  - Downloading data from the web
- Why scrape?
  - Data can’t be easily accessed through other means, but it is accessible through the web
- What type of data can be scraped?
  - Almost anything!
  - Anything you can see in your browser can be programmatically accessed with web scraping

Scraping Terminology

- Crawling
  - Systematically downloading every page (or a specific subset of pages) on a website
  - Mostly useful when you need to be exhaustive in your search
- Scraping
  - Downloading the data
- Parsing
  - Turning your scraped HTML into plain-text or specific features

Simple Example

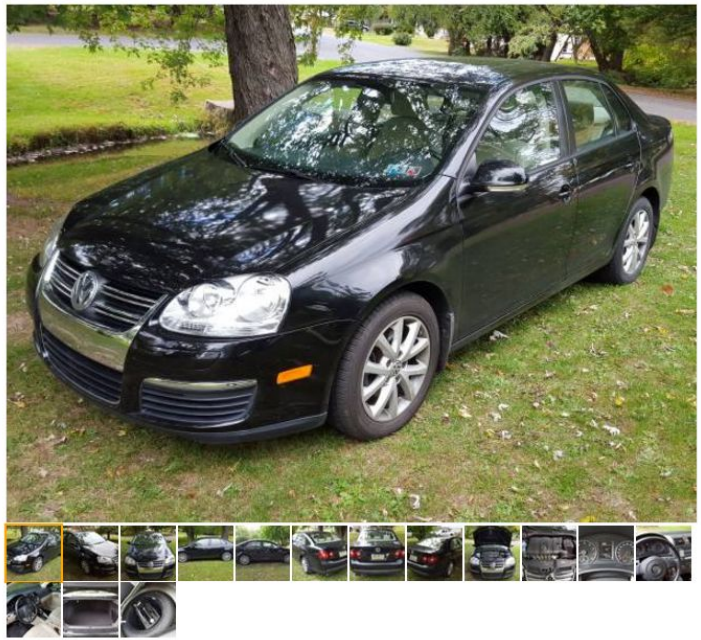
- Craigslist
  - I wanted to get the market price of used Volkswagen cars
  - Was interested in studying the effect of the emissions scandal (2015)
    - This required fairly recent prices; incident had only happened months earlier
    - Historical data on car prices wouldn’t be helpful

Data Scraped

Unstructured Data

Structured Data

2010 Volkswagen Jetta SE 35k Miles - $6900 (Trevose)



2010 Volkswagen Jetta

VIN: 3VWRZ7AJ0AM162557
condition: excellent
fuel: gas
odometer: 35319
paint color: black
title status: rebuilt
transmission: automatic

- safety tips
- prohibited items
- product recalls
- avoiding scams

2010 Volkswagen Jetta with 35,319 miles on it for sale. The car comes with leather and heated seats, sunroof, alloy wheels, Bluetooth phone support and much more. The car is in great shape with no issues whatsoever. For any questions about the vehicle or if you would like to come see it please call me at

show contact info

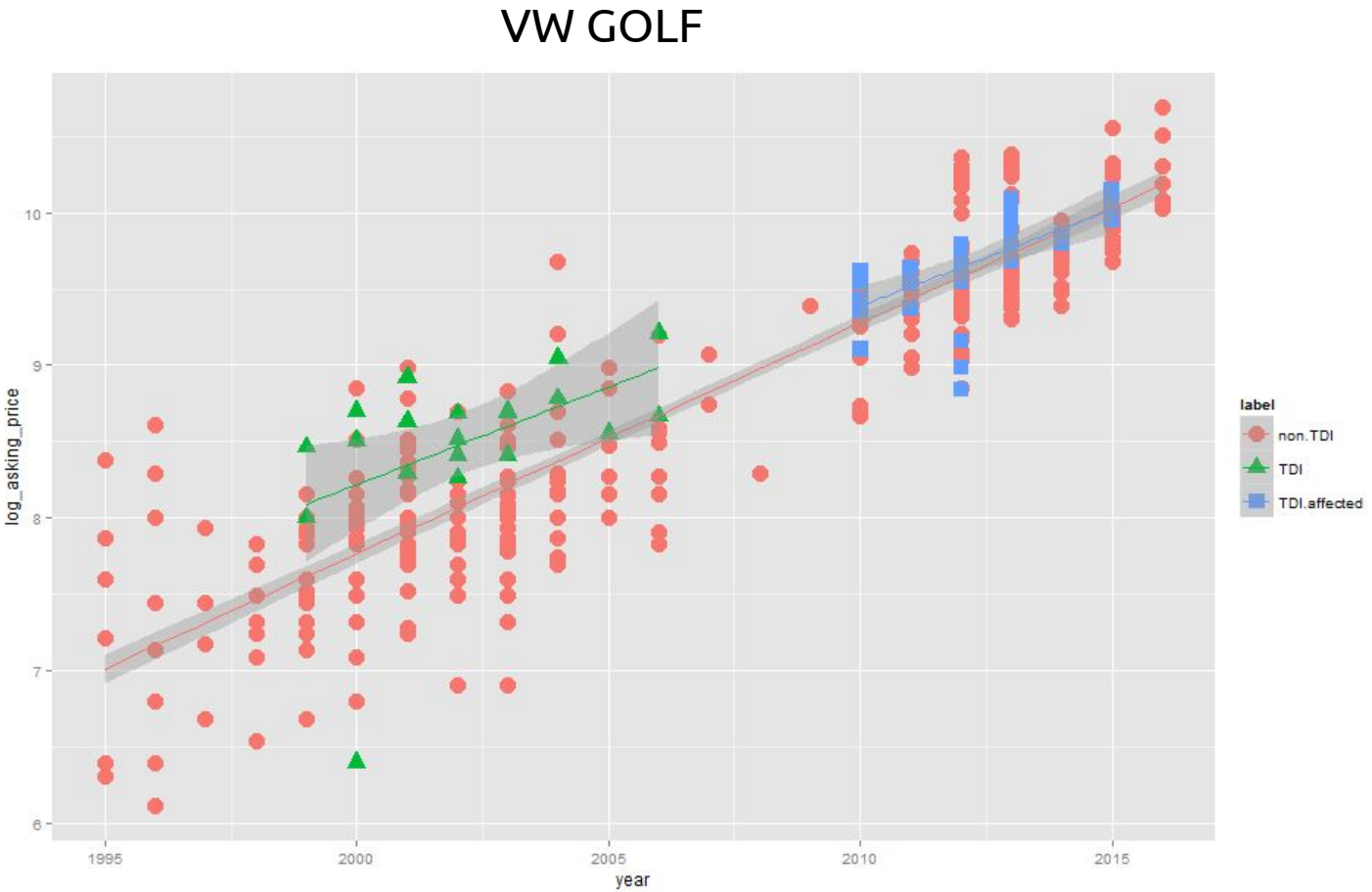
This car is located in Trevose, PA 19053. If the ad is still up, the car is still available.

asking_price (DV)
year
model
fuel_type
scandalized (IV)
city
luxury_model
dealer
drive_type
mileage
transmission
body_type
cylinders
condition
title_status
timestamp
last_updated
days_since_scandal_posted
days_since_scandal_updated
weeks_since_scandal_posted
weeks_since_scandal_updated

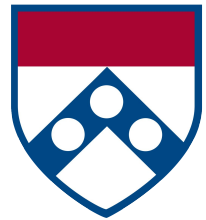
What was the result?

- Scraped over 13,000 Craigslist listings for used Volkswagens
- 11 metro areas
    - Philadelphia, Boston, Miami, Minneapolis, Houston, Dallas, Chicago, Seattle, Portland, San Francisco, Los Angeles
- Posting dates vary from 10/9/2015 to 11/9/2015
- Cleaned data yielded 8,634 listings

Outcome


VW GOLF
log_asking_price
10
9
8
7
6
1995
2000
2005
2010
2015
year
label
non.TDI
TDI
TDI.affected

Non-scraping



Non-scraping?

- One more time…
  - The easiest way to scrape the web is to avoid scraping!
- Certain types of data, although not exposed by public API, can still be accessed
- In particular:
  - Tabular data
  - Async data

Extracting tabular data

- If your data is in an HTML <table> tag, there are easier ways to access it besides scraping
  - ImportHTML function in Google Sheets
  - Python Pandas read_html function

Basic Table Example

- Wikipedia table:
  - https://en.wikipedia.org/wiki/List_of_United_States_television_markets
  - For a different project, I needed to map US Census data (given at the county level) to DMA data
  - Nielsen data is private (and expensive)
    - → I'm pretty sure Wharton has access to it, but this is easy enough

Basic Table Example

- Google Sheets Import
- Pandas read_table

Quick Digression: Async Data

- If a website is more “dynamic”, a lot of its data will be downloaded asynchronously
- Downloading dynamic/async content requires **Javascript** to be running in your web client
- This means if you try to download the page using Python’s requests module, you will only see part of the data
  - E.g., https://www.theatlantic.com/science/archive/2017/08/advice-for-eclipse-newbies/536010/#article-comments

Solutions

- Selenium Webdriver
  - “Headless” browser (PhantomJS)
  - Open browser (Firefox is easiest) and programmatically load the content
  - Won’t cover here
- Async API Snooping
  - Companies use their own APIs to access content
  - This often means that you can access the API too

Async Snooping Examples

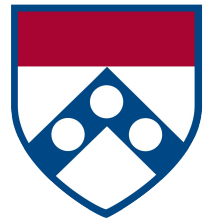
Google Trends

https://trends.google.com/trends/explore?geo=US&q=%2Fm%2F0dl567

The Atlantic/Disqus Comments

https://www.theatlantic.com/science/archive/2017/08/advice-for-eclipse-newbies/536010/#article-comments

Scraping & Basic Parsing



Selecting Text from HTML Docs

- Once you've downloaded your HTML page… then what?
  - Parse it!
- Three ways of using the HTML formatting to get the data you want:
  - XPath
    - → https://www.w3schools.com/xml/xpath_intro.asp
  - BeautifulSoup
    - → https://www.crummy.com/software/BeautifulSoup/bs4/doc/
  - PyQuery
    - → jQuery like selection of objects using CSS selectors
    - → `pip install pyquery`

HTML Basics

- Valid HTML pages are built using a specific set of tags
- The most important/common ones
  - <head>
  - <body>
    - <h1>, <h2>, <h3>: headers
    - : generic holder
    - <p>, : text
    - : lists
      - : individual elements
    - <img>, <table>, <select>: miscellaneous

HTML Basics

- Every tag can have a set of attributes
- The most common ones in HTML are **class**, **id**
  - Also the **href** attribute for <a> tags (links)
- Example:
  - ```

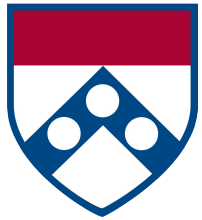
        This is some text

    ```
- IDs are supposed to be unique (no other element on the page should have the same ID)
- Classes need not be unique; an element can have multiple classes
    ```

CSS Selectors

- Way of identifying an element (or set of elements) on a web page (originally used for styling HTML pages)
- Basic rules
  - Elements by type: `div` (any HTML tag, e.g., p, ul, table)
  - classes: `.className`
  - ids: `#idName`
- Combinations:
  - Subsetting elements: `div.className div#idName`
  - Multiple classes: `.firstClass.secondClass`
  - Direct children: `.parent > .child`
  - Descendants: `.parent .descendant`
  - Multiple selectors: `.className, #idName, p`

Conclusion



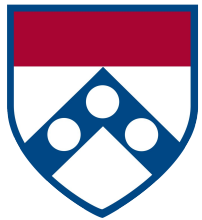
Things to be aware of!

- Many (most!) sites don't like scrapers!
  - Do your best to be respectful
  - There *are* ways around most website blockages
    - → CORS, UserAgent, Even Logins
      - Will cover some in later sessions
  - Legality is kind of up in air right now
    - → https://arstechnica.com/tech-policy/2017/07/linkedin-its-illegal-to-scrape-our-website-without-permission/
- Best practices:
  - Download (i.e., save) HTML offline first, parse later
  - This not only requires fewer requests to the web, but also allows you to parse things you didn't realize were there

Important Takeaways

- Can scrape some things without code
- Sometimes, the HTML return through programmatic access will be different that the HTML you see in “view source” from your browser
  - Good indicator the site is using dynamic content!
- Save your HTML for *offline* parsing
- Learn your CSS selectors

Notebook Walkthrough



Exercise

- Task
  - Take the list of Canadian PGA winners at the Wiki link below and turn it into a list of tuples with the year, first name, and last name as entries in each tuple
    - → https://en.wikipedia.org/wiki/Canadian_PGA_Championship
      - (Use the longer list called “Canadian PGA Championship”)
    - → [(2014, “Alex”, “Miller”), (2013, “Bob”, “Ross”), …]
  - I encourage you to do it in 100% Python for practice
    - → But also try out the Google Sheets method!
- Next class
  - More advanced scraping and parsing