Department of
Computer Science

Data Collection

Erick Gomez Nieto, PhD
emgomez@ucsp.edu.pe

Data collection is the process of collecting and evaluating information or data from multiple sources to find answers to research problems, answer questions, evaluate outcomes, and forecast trends and probabilities. It is an essential phase in all types of research, analysis, and decision-making, including that done in the social sciences, business, and healthcare.

Accurate data collection is necessary to make informed business decisions, ensure quality assurance, and keep research integrity.

Source: *https://www.simplilearn.com/what-is-data-collection-article*

Before an analyst begins collecting data, they must answer three questions first:

- What's the goal or purpose of this research?
- What kinds of data are they planning on gathering?
- What methods and procedures will be used to collect, store, and process the information?

Additionally, we can break up data into qualitative and quantitative types. Qualitative data covers descriptions such as color, size, quality, and appearance. Quantitative data, unsurprisingly, deals with numbers, such as statistics, poll numbers, percentages, etc.

Traditional ways to get data ...

- Using dataset from open repositories
- Web Scraping
- Using APIs

Open Repositories

kaggle

Google Dataset Search Beta

Open Data REPOSITORY

NYC OPEN DATA

OPEN DATA

opendata .swiss

DATA HUB

Web Scraping

Web Scraping (also termed Screen Scraping, Web Data Extraction, Web Harvesting etc.) is a technique employed to *extract large amounts of data from websites* whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format.

https://www.webharvy.com/articles/what-is-web-scraping.html

Data displayed by most websites can only be viewed using a web browser. They do not offer the functionality to save a copy of this **data for personal use**. The only option then is to manually copy and paste the data - a very tedious job which can take many hours or sometimes days to complete. **Web Scraping** is the technique of automating this process, so that instead of manually copying the data from websites, the Web Scraping software will perform the same task within a fraction of the time.

Web scraping tools

Data/Page formats on the web

- HTML, HTML5 (<!DOCTYPE html>)

```
<div align="center">
        <table width="952" border="0" cellspacing="0" cellpadding="0">
                <tr>
                        <td width="1" bgcolor="cccccc">
                                <img src="../images/transparent.gif" widt
                        </td>
                        <td width="950" background="../images/main_back.p

                                <table width="950" border="0" cellspacing
                                        <tr>
```

Data/Page formats on the web

- HTML, HTML (<!DOCTYPE html>)
- data formats: XML, JSON
- PDF
- APIs
- other languages on the web; css, java, php, asp.net…

BeautifulSoup

```
pip install beautifulsoup4
```

- General purpose, robust, works with broken tags
- Parses html and xml, including fixing asymmetric tags, etc.
- Returns unicode text strings
- Alternatives; lxml (also parses html), Scrapy
- Faster alternatives: ElementTree, SGMLParser (custom)

BeautifulSoup

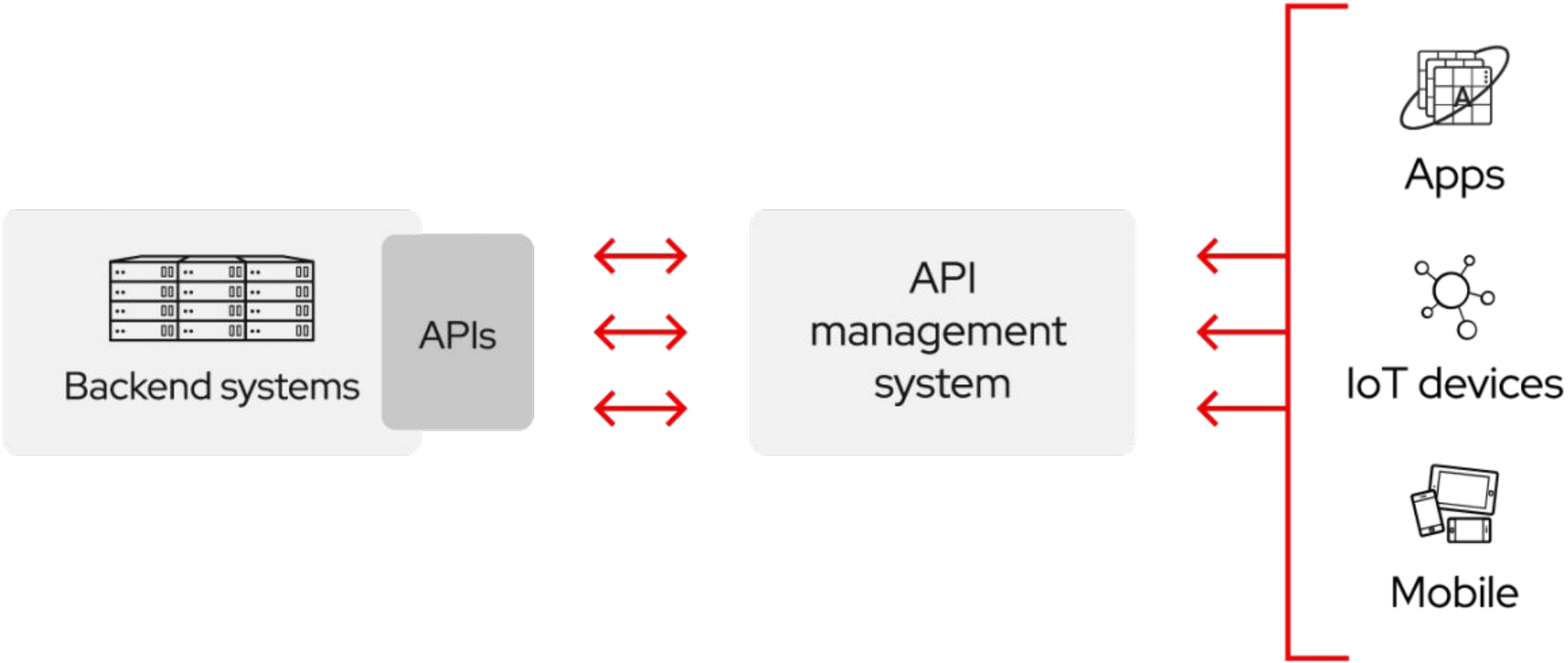
Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

APIs

An API is a set of definitions and protocols for building and integrating application software. API stands for application programming interface.

APIs let your product or service communicate with other products and services without having to know how they're implemented. This can simplify app development, saving time and money. When you're designing new tools and products—or managing existing ones—APIs give you flexibility; simplify design, administration, and use; and provide opportunities for innovation.

https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces



https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces

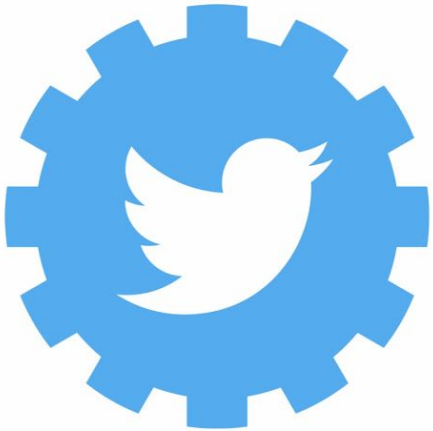
THE WORLD BANK

Google APIs

aws

arXiv.org

Bing



OSMnx

https://osmnx.readthedocs.io/en/stable/index.html

OSMnx is a Python package to retrieve, model, analyze, and visualize street networks from OpenStreetMap. Users can download and model walkable, drivable, or bikeable urban networks with a single line of Python code, and then easily analyze and visualize them. You can just as easily download and work with amenities/points of interest, building footprints, elevation data, street bearings/orientations, and network routing.

Features (1/3)

Download street networks anywhere in the world with a single line of code

Download other infrastructure types, place boundaries, building footprints, and points of interest

Download by city name, polygon, bounding box, or point/address + network distance

Download drivable, walkable, bikeable, or all street networks

Download node elevations and calculate edge grades (inclines)

Features (2/3)

Impute missing speeds and calculate graph edge travel times

Simplify and correct the network's topology to clean-up nodes and consolidate intersections

Fast map-matching of points, routes, or trajectories to nearest graph edges or nodes

Save networks to disk as shapefiles, GeoPackages, and GraphML

Save/load street network to/from a local .osm xml file

Features (3/3)

Conduct topological and spatial analyses to automatically calculate dozens of indicators

Calculate and visualize street bearings and orientations

Calculate and visualize shortest-path routes that minimize distance, travel time, elevation, etc

Visualize street network as a static map or interactive leaflet web map

Visualize travel distance and travel time with isoline and isochrone maps

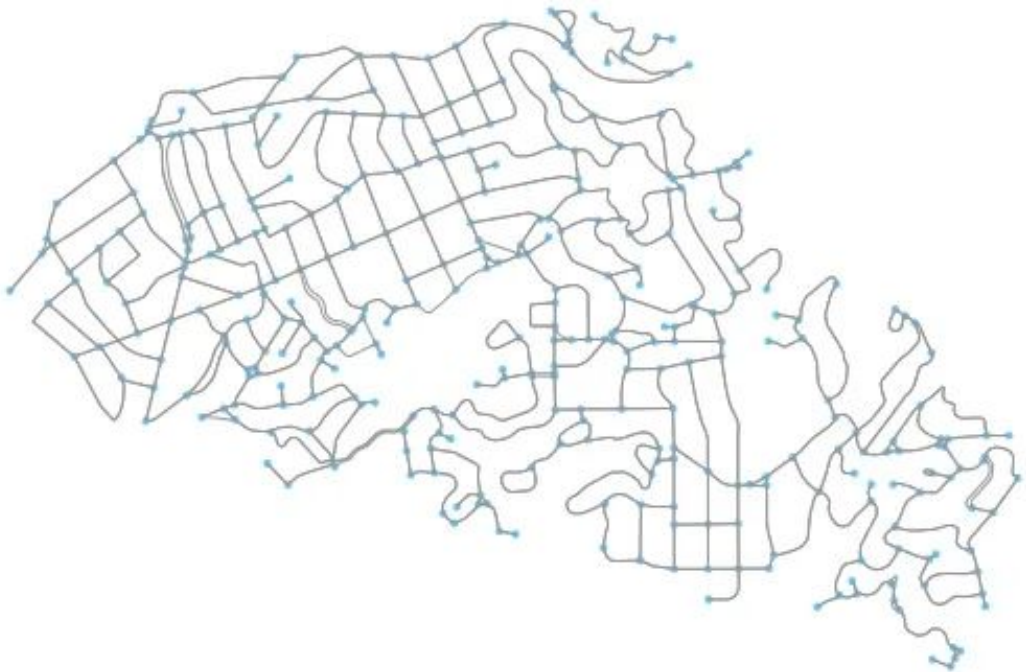
Plot figure-ground diagrams of street networks and/or building footprints

Installation

You can install OSMnx with conda:

```
conda config --prepend channels conda-forge
conda create -n ox --strict-channel-priority osmnx
```

```
import osmnx as ox
%matplotlib inline
G = ox.graph_from_place('Piedmont, California, USA', network_type='drive')
fig, ax = ox.plot_graph(ox.project_graph(G))
```



Twint (*off-air*)

https://osmnx.readthedocs.io/en/stable/index.html

Twint

Twint is an advanced Twitter scraping tool written in Python that allows for scraping Tweets from Twitter profiles without using Twitter's API.

Twint utilizes Twitter's search operators to let you scrape Tweets from specific users, scrape Tweets relating to certain topics, hashtags & trends, or sort out sensitive information from Tweets like e-mail and phone numbers.

Twint also makes special queries to Twitter allowing you to also scrape a Twitter user's followers, Tweets a user has liked, and who they follow without any authentication, API, Selenium, or browser emulation.

Twint Benefits

Some of the benefits of using Twint vs Twitter API:

- Can fetch almost all Tweets (Twitter API limits to last 3200 Tweets only);
- Fast initial setup;
- Can be used anonymously and without Twitter sign up;
- No rate limitations.