

INLS 560  
Programming for  
Information Professionals

# Visualizing Web Data



UNC  
SCHOOL OF INFORMATION  
AND LIBRARY SCIENCE

Joan Boone

*jpboone@email.unc.edu*

**Part 1: Overview**

Part 2: Creating a Google Pie Chart using Python

Part 3: Charting Web Data

Part 4: Examples

Part 5: SSL certificate error

# Visualizing JSON data with Google Charts

One Thing At A Time 2023-03-03  
Country Music

HOPE 2023-04-07  
Hip-Hop/Rap Music

NEVER ENOUGH 2023-04-07  
R&B/Soul Music

Sremm 4 Life 2023-04-07  
Hip-Hop/Rap Music

PORTALS (Deluxe) 2023-03-31  
Alternative Music Pop

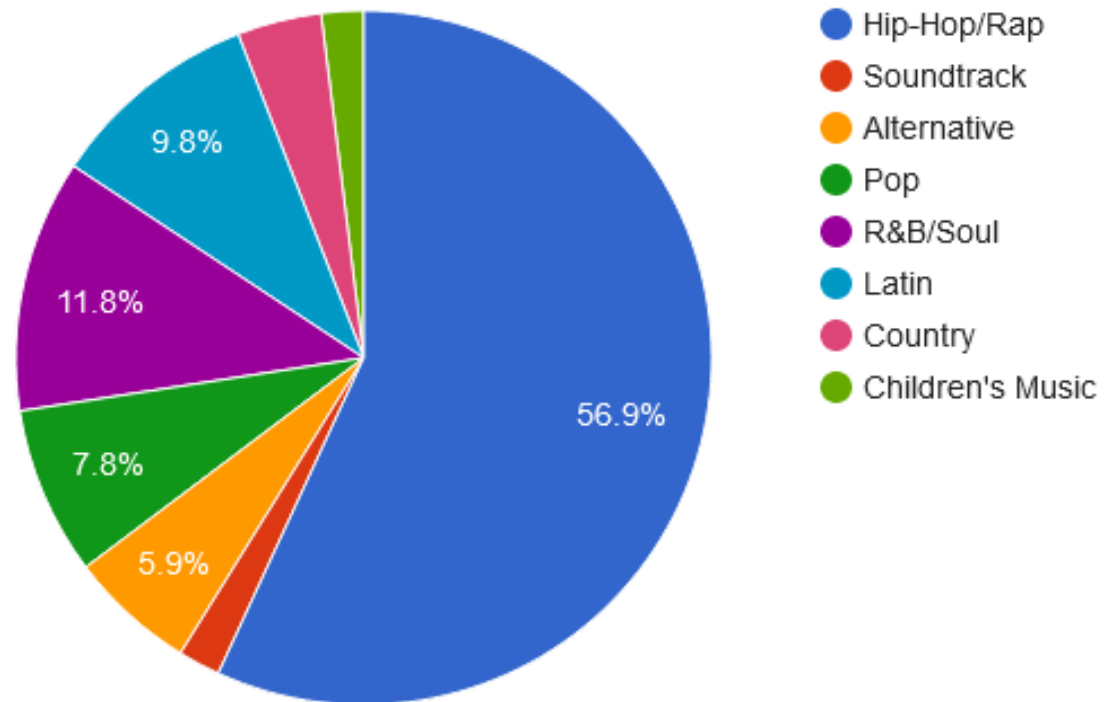
Her Loss 2022-11-04  
Hip-Hop/Rap Music

Gettin' Old 2023-03-24  
Country Music

SOS 2022-12-09  
R&B/Soul Music

. . .

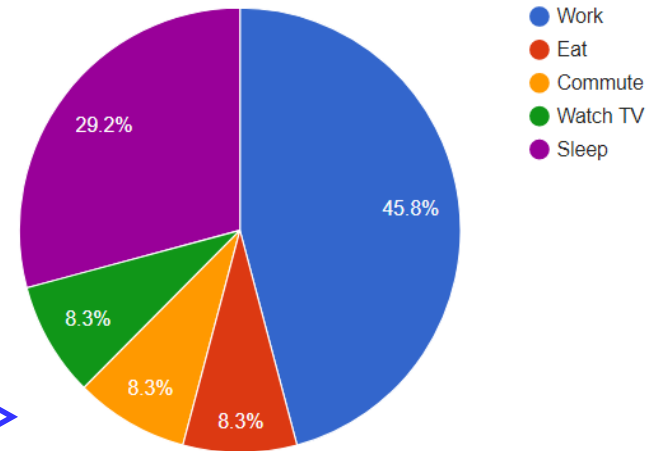
Apple Music: Top 50 Most Played Albums by Genre



# Using Google Charts

- Many types of charts: [Gallery](#)
- How it works (using JavaScript)
  - Load some Google Charts libraries
  - Define the data for your chart
  - Customize it (optional)
  - Create a chart object to display in a `<div>`
- Example: [simple Pie Chart](#)
- The most common way to use Google Charts is with JavaScript embedded in a web page
  - JSON is the data format used to populate charts with data
  - Google has open-sourced [a Python library](#) that creates JSON objects for consumption by its charts

My Daily Activities



Part 1: Overview

**Part 2: Creating a Google Pie Chart using Python**

Part 3: Charting Web Data

Part 4: Examples

Part 5: SSL certificate error

# Using Google's Python Library to Create a Pie Chart

## Python program

- Defines the **description** for data
- Defines the **data** to be charted
- Converts the **data** to JSON format

```
# Create the schema (data columns and types)
description = [{"Genre", "string"},
               {"Frequency", "number"}]
```

```
# Create the data
data = [{"Hip-Hop/Rap", 29},
        {"R&B/Soul", 6},
        {"Latin", 5},
        {"Pop", 4}],
        {"Alternative", 3},
        . . . ]
```

*"hard-coded" in the program*

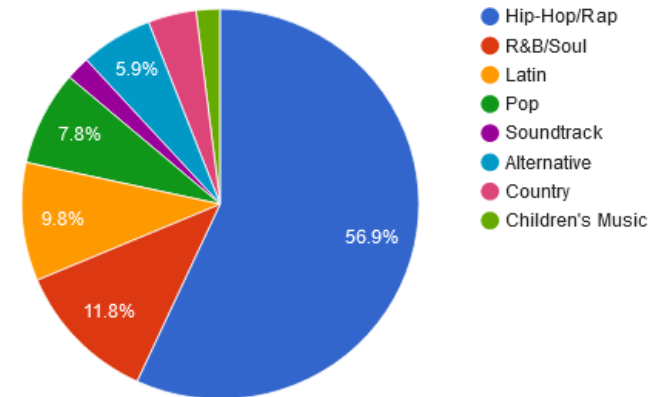
```
# Create a DataTable object
data_table = gviz_api.DataTable(description)
data_table.LoadData(data)
```

```
# Convert to JSON
json_text = data_table.ToJson()
```

Google Charts  
**DataTable** in  
JSON format

Google Charts Visualization program

Apple Music: Top 50 Most Played Albums by Genre



genre\_piechart\_basic.py creates → genre\_piechart\_basic.html *Slide 6*

# Install the Google Python Library, **gviz.api**

...

```
# Create a DataTable object  
data_table = gviz_api.DataTable(description)  
data_table.LoadData(data)
```

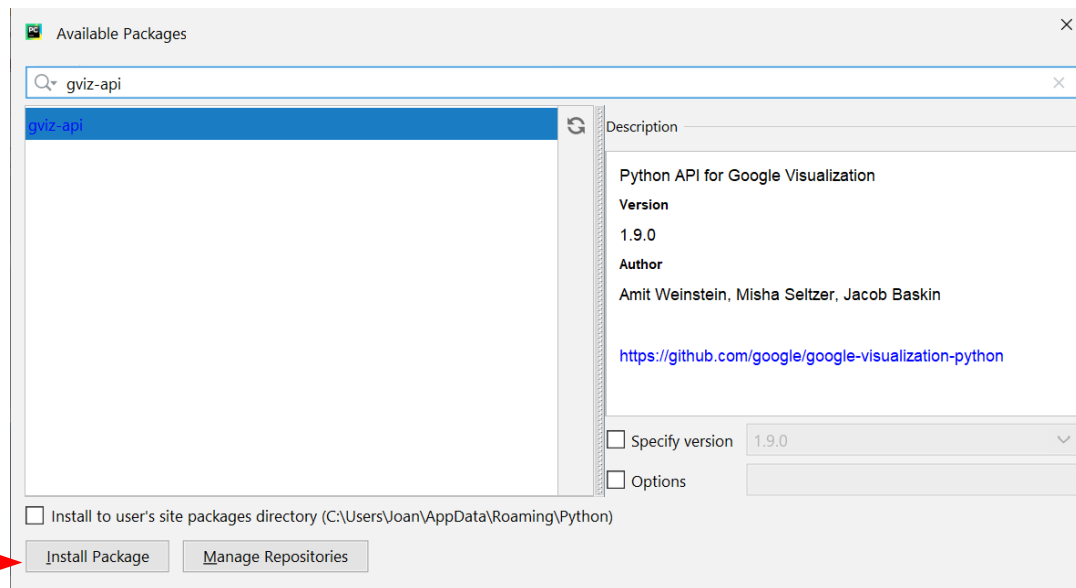
...

**Windows:** In PyCharm, select the File menu item, then Settings...

**Mac:** Select PyCharm menu item, then Preferences...

For both Windows and Mac:

- On the left hand side, select Project: PyCharm\_Projects (or whatever you called your folder)
- Select Project Interpreter, and you will see a list of packages already installed
- Click '+' (top of pane on Windows, bottom of pane on Mac) to display 'Available Packages'.
- Search for **gviz.api**, select, and click 'Install Package' button



Part 1: Overview

Part 2: Creating a Google Pie Chart using Python

**Part 3: Charting Web Data**

Part 4: Examples

Part 5: SSL certificate error



# Charting Web Data

## Python program

- Parses input and selects data for charting
- Defines the **description** for data
- Defines the **data** to be charted
- Converts the **data** to JSON format

JSON format

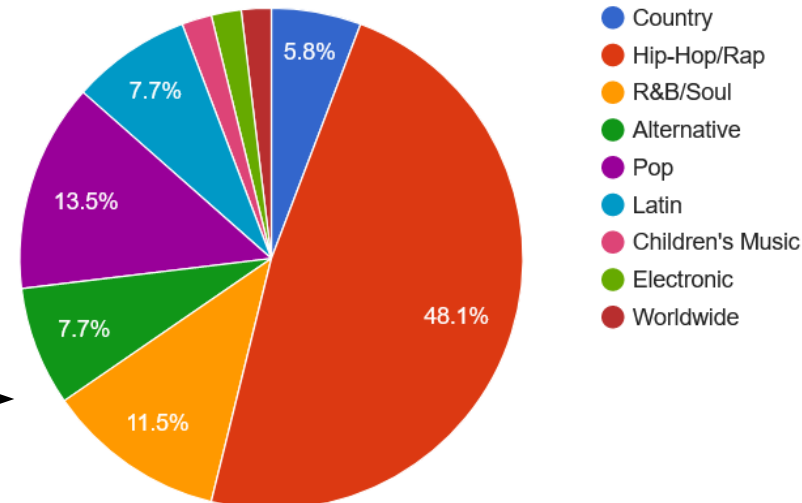
Web Data

<https://rss.applemarketingtools.com/api/v2/us/music/most-played/50/albums.json>

Google Charts  
**DataTable** in  
JSON format

Google Charts Visualization program

Apple Music: Top 50 Most Played Albums by Genre



genre\_piechart\_url.py creates

genre\_piechart\_url.html

Slide 9

# Web Data

Web data is often 'open data' that is free and publicly available so that anyone can access and use it without restrictions.

## W3C Recommendation: [Data on the Web Best Practices](#)

- As data becomes more ubiquitous, and data sets become larger and more complex, processing by computers becomes ever more crucial.
- Data becomes useful when it has been processed and transformed into information.
- [Best Practice for Data Formats](#)
  - Make data available in a machine-readable standardized data format that is easily parseable including but not limited to CSV, XML, HDF5, JSON and RDF serialization syntaxes like RDF/XML, JSON-LD.

# Many Sources of Web Data...

## a few places to start looking

- [Data.gov](#) (DATA > JSON Format)
- github: [Awesome JSON Datasets](#)
- [HealthData.gov](#)
- [NYC OpenData](#)
- [Hawaii Open Data](#) (Datasets > JSON Format)
- [Kaggle Datasets](#)
- [Data Science for Social Good](#): Best Sources for Free Open Data

Part 1: Overview

Part 2: Creating a Google Pie Chart using Python

Part 3: Charting Web Data

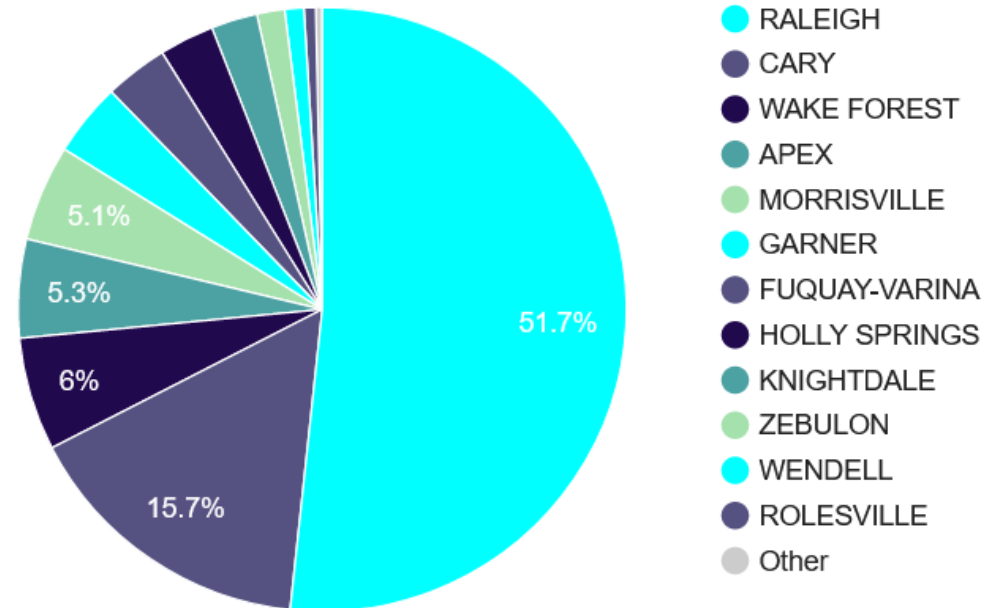
**Part 4: Examples**

Part 5: SSL certificate error

# Location of Wake County Restaurants

```
[
  {
    "objectid": 2308689,
    "hsisid": "04092017248",
    "name": "CELLAR 55",
    "address1": "1351 E BROAD ST",
    "address2": null,
    "city": "FUQUAY-VARINA",
    "state": "NC",
    "postalcode": "27526",
    "phonenumber": "(919) 446-1156",
    "restaurantopendate": "2016-06-15T06:00",
    "facilitytype": "Restaurant",
    "permitid": 19376,
    "x": -78.77604319,
    "y": 35.59548799,
    "geocodestatus": "M",
    "geo_shape": {
      "type": "Feature",
      "geometry": {
        "coordinates": [
          -78.77604582921708,
          35.59549502181785
        ],
        "type": "Point"
      },
      "properties": {}
    },
    "geo_point_2d": {
      "lon": -78.77604582921708,
      "lat": 35.59549502181785
    }
  },
  ...
]
```

Wake County Restaurant Locations

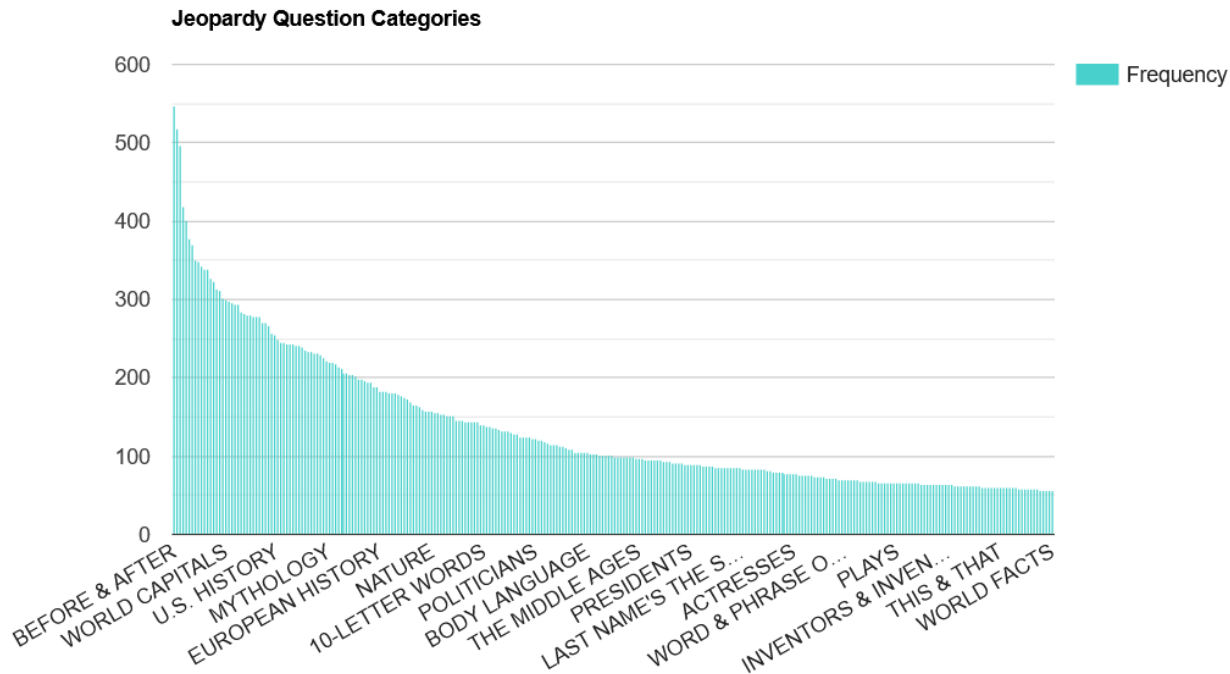


This dataset can be found at [Data.gov](https://data.townofcary.org/api/v2/catalog/datasets/wake-county-restaurants/exports/json) under Organizations → [Town of Cary](https://data.townofcary.org)

# Jeopardy Categories

Frequency of categories for 200K+ Jeopardy questions

```
[
  {
    "category": "HISTORY",
    "air_date": "2004-12-31",
    "question": "'For the last 8 years of his
      life, Galileo was under
      house arrest for espousing
      this man's theory'",
    "value": "$200",
    "answer": "Copernicus",
    "round": "Jeopardy!",
    "show_number": "4680"
  },
  {
    "category": "ESPN's TOP 10",
    "air_date": "2004-12-31",
    "question": "'No. 2: 1912
      football star
      6 MLB seasons'",
    "value": "$200",
    "answer": "Jim Thorpe",
    "round": "Jeopardy!",
    "show_number": "4680"
  },
  ...
]
```



[https://opal.ils.unc.edu/~jpboone/json/jeopardy\\_200K\\_questions.json](https://opal.ils.unc.edu/~jpboone/json/jeopardy_200K_questions.json)

jeopardy\_category\_chart.py, jeopardy\_200K\_category\_chart.html

Part 1: Overview

Part 2: Creating a Google Pie Chart using Python

Part 3: Charting Web Data

Part 4: Examples

**Part 5: SSL certificate error**

# Important!

## When accessing web data...

- Opening a URL is equivalent to sending an HTTP request from a client (your program) to a server that hosts the web resource
- Secure Sockets Layer (SSL) is a security standard that enables encrypted communication between a client (or web browser) and a web server
- Typically, a client verifies a server's certificate, but it is also the case that a server may require a signed certificate from a client. Open data sources vary in whether they require a signed certificate.
- When you run this program, `genre_piechart_from_url.py`, if you see either of these exceptions

```
<urlopen error [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1076)>
```

```
<urlopen error [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get local issuer certificate (_ssl.c:1076)>
```

this means your Python installation does not have the SSL certificates being requested by the web server.



# How to avoid this SSL certificate error (if you encounter it)

Use the Python SSL module

- Add `import ssl` to your program
- Add this statement before calling `urlopen`:

```
ssl._create_default_https_context =  
    ssl._create_unverified_context
```

Examples of how to do this can be found in these programs

`genre_piechart_url.py`

`jeopardy_category_chart.py`

# For Mac users: another possible fix

There is a [stackoverflow question](#) that addresses this issue, and a fix, in the last entry

For novice users, you can go in the Applications folder and expand the Python 3.7 folder. Now first run (or double click) the Install Certificates.command and then Update Shell Profile.command

