ITMD 413/513 Final Project Specs & Rubric

Rubric (point breakdown)

|  |  |
| --- | --- |
| Documentation / Organization (modules,etc) / Table of Contents | 50 |
| Source file import (json, csv, etc.) | 25 |
| Presentation( vid./output snapshots) | 25 |
| Source code / Error traps, Security | 100 |
| **Total** | **200** |

Objective:

To read in data over the web, plot and show data analysis results.

Intelligence gathering:

I. Download given files (data.py, plots.py) within your Final Project folder in BB.

These files can be used to help guide you through your project details. Place your files in a desired workspace (directory).

Include any installs you may need for this project.

Ex.

pip install requests

pip install matplotlib

if nec.

pip install pandas (this usually installs numpy as well)

II. Run your plots.py file and observe console output and graphic.

III. File tweaks (data.py):

Modify your line that follows to include your home addresses long and lat as well as a long and lat some miles close to your address in your district area. This will help build a strong data set to work some analyses and graph.

boxurl = 'within\_box(location, 41.975121, -87.791649, 41.978260, -87.763931)'

Each pair in the above function represents a lat/long set of an address. To find coordinates to put in visit the website – <https://www.latlong.net>.

Type in your home address to find the coordinates to place in the function above and another desired location.

You can always tweak columns for your data frame to observe and report on your desired results from the line below.

df = pd.DataFrame(

text, columns=['date', 'block', 'primary\_type', 'description'])

IV. Choose at least three graphs to show differing analysis results.

Examples:

Top 5 various crimes by count within your locations (like a horizontal bar chart).

Ratio of arrests to non-arrest (like a pie chart).

Isolate a particular crime by dates within a range (bar or line chart).

Etc.

For each graph make sure to include a snapshot of it, labeled and described appropriately. Description includes details of your graph. What is it about, how is data represented, etc?

Make sure to have each of your graphs titled, x,y axes labled, and any legends depicting your data.

Ex.

TITLE

A screenshot of a cell phone

Description automatically generated

LEGEND

Labels x,y

For credit

Turn in all your py, ipynb files, data files (csv, json, db, pickled) and a doc file of your snapshots.

Have a title page consisting of a Table of Contents that includes a description your project, snapshot inclusions, a summary and any site references. If you include a login, include first hand your credentials after the title page.

Grads:

Include a GUI Login with security features. After a successful login, include a GUI for user interaction. GUI interaction can include a drop down list to select what graph to display. Other choices can include option buttons and/or check boxes to allow user to nit-pick what data that want selected from your data set. Ex. Crime stats for Burglaries and if any arrests where made. Display can be to the console as well showing detailed data (ex. Noted locations of incidents) or even to a label within the interface.

-[Video](https://www.techsmith.com/jing-tool.html) demoing your work. Include link in your submission.

Extra credit (all):

- GUI’s (tkinter -under grads)

- Webframework, i.e., Django or Flask.

- Menu driven system if your operating thru console interface

- Intuitive charting, chart comparisons – ex. Scatter vs. line charts. Side by side horizontal

depictions or vertical top-down depictions.

- Stats on relevant data. Mean, std dev, variances, counts, averages, correlations, etc.

- Sqlite

- Jupyter notebook <https://notebooks.azure.com/>

- Serializing/Deserializing data sets (example compare last month data with this months data)

- Machine Learning/AI/DL/NLP

\*Include any snapshots to represent your Extra Credit labors, labeled accordingly.

Key references:

Project: Focus

<https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2>

Pandas

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>

<https://www.dataquest.io/blog/pandas-cheat-sheet/>

<https://python-graph-gallery.com/7-custom-barplot-layout/>

Matplotlib

<https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.figure.html>

<https://matplotlib.org/3.1.1/gallery/lines_bars_and_markers/barh.html>

Jupyter

<https://notebooks.azure.com/jpapa/projects/machinelearning>

<https://stackoverflow.com/questions/48757025/trying-to-extract-json-data-from-an-url-into-jupyter>

Heat Maps for correlations – seaborn module

<https://www.kaggle.com/itsbitan/mall-customer-segment>

Other:

<https://hackernoon.com/chicago-crime-mapping-magic-of-data-science-and-python-f2ecad74a597>

<https://www.kaggle.com/djonafegnem/chicago-crime-data-analysis/data>

Project alternatives:

<https://towardsdatascience.com/sentiment-analysis-with-python-part-1-5ce197074184>

API alternative references:

<https://www.kaggle.com/orgesleka/imdbmovies>

<https://openweathermap.org/api>

Other key refs:

JSON - <https://codebeautify.org/jsonviewer>

Chart types - <https://matplotlib.org/tutorials/introductory/sample_plots.html>

Pandas usages

print(df.head())

print(df.head(n=100))

print(df) # full set

print(df['date'])

print(df.columns[0][:])

print(df.describe()) # possible stats

Other:

print(type(data))

# make json data into dict object

data = {k:[v] for k, v in text[0].items()}

for k,v in data.items():

print(k, ":", v)

# iterating the columns

for col in df.columns:

print(col) # print column name

# Yields a tuple of index label and series for each row in the dataframe

for (index\_label, row\_series) in data.itertuples():

if index\_label == 'BURGLARY':

print(row\_series)