

1.

```
#!/usr/bin/python

import numpy
import requests
import csv
from matplotlib import pyplot as plt
from datetime import datetime

# Data dictionaries to store data
# Originally did it with lists which required less code
# Changed it to dictionaries cause I found the data structure to be better formatted with key pairs and easier to understand
# Putting data in dictionary is nice for any future use as well
dataconfirmed = {}
datadeaths = {}

try:
    # Gets current data from Github
    print("Downloading current Coronavirus data files\n")
    confirmed = requests.get("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv")
    deaths = requests.get("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_global.csv")

    confirmed_decoded = confirmed.content.decode('utf-8')
    deaths_decoded = deaths.content.decode('utf-8')

    # Loads in data from CSVs as lists
    print("Loading in data from current Coronavirus data files\n")
    c = list(csv.reader(confirmed_decoded.splitlines(), delimiter=','))
    d = list(csv.reader(deaths_decoded.splitlines(), delimiter=','))

    # Gets relative death rate and time from data and formats as a dictionary
    # This is done so data is better formatted and easier to read
    print("Getting relative death rate and time from current Coronavirus data\n")
    # Gets dates from start to current so matplotlib can use it to plot
    time = []
    for date in c[0][4:]:
        split = date.split('/')
        dates = datetime(int(split[2]) + 2000, int(split[0]), int(split[1]))
        time.append(dates)

    # Puts data into dictionaries for confirmed and deaths and formats it
    # Puts data into dataconfirmed dictionary
    # If statement for Providence/State if listed
    for row in c[1:]:
        if row[0] == '':
            dataconfirmed[row[1]] = {}
            for date in range(len(row[4:])):
                dataconfirmed[row[1]][c[0][date+4]] = int(row[date+4])
        else:
            dataconfirmed[row[0] + ", " + row[1]] = {}
            for date in range(len(row[4:])):
                dataconfirmed[row[0] + ", " + row[1]][c[0][date+4]] = int(row[date+4])

    # Puts data into datadeaths dictionary
    for row in d[1:]:
        if row[0] == '':
            datadeaths[row[1]] = {}
            for date in range(len(row[4:])):
                datadeaths[row[1]][d[0][date+4]] = int(row[date+4])
        else:
            datadeaths[row[0] + ", " + row[1]] = {}
            for date in range(len(row[4:])):
                datadeaths[row[0] + ", " + row[1]][d[0][date+4]] = int(row[date+4])
```

```
# Gets world relative death rate by summing up total deaths and total confirmed and dividing for each date
deathrate = []
for col in range(len(c[0][4:])):
    totalconfirmed = 0
    totaldeath = 0
    for row in c[1:]:
        totalconfirmed = totalconfirmed + int(row[col + 4])
    for row in d[1:]:
        totaldeath = totaldeath + int(row[col + 4])
    deathrate.append(float(totaldeath)/float(totalconfirmed))

# Plots relative world death rate vs time
plt.plot(time, deathrate)

plt.title("World")
plt.xlabel("Time")
plt.ylabel("Relative Death Rate")

plt.xticks(rotation=45)

plt.show()

# Calculates relative death rate and plots the data for each place (similar to what is done for the world)
print("Plotting relative death rate vs time from current Coronavirus data\n")
deathrate[:] = []
for row in c[1:]:
    if row[0] == '':
        name = row[1]
    else:
        name = row[0] + ", " + row[1]
    for date in c[0][4:]:
        if dataconfirmed[name][date] != 0:
            deathrate.append(float(datadeaths[name][date])/float(dataconfirmed[name][date]))
        else:
            deathrate.append(0)

plt.plot(time, deathrate)

plt.title(name)
plt.xlabel("Time")
plt.ylabel("Relative Death Rate")

plt.xticks(rotation=45)

plt.show()

deathrate[:] = []

# Prints if not able to get data and plot (error handling)
except:
    print("Unable to download files or plot data\n")
    exit()
```

```
2. enscript -T 4 --header='$n %E %*|$%|Spencer Goulette' hw06.txt -o - | ps2pdf - ECE-331-Goulette-Spencer-HW-06.pdf
```