UNIVERSITY OF GUELPH
School of Computer Science

CIS*6180/DATA*6300	**Analysis of Big Data**	**Winter 2023**
Total Marks: 10	**Assignment# 2**	**Sohail Habib**

Assignment date: Feb 03, 2023
Assignment due date: Feb 17, 2023, 3 :00 pm

Overview

The purpose of this assignment is to get you introduce the data structures and modeling. You will be reading and processing variety of data types using python.

Please use CourseLink for all communications. Ask a private question if necessary.

What to submit?

You need to provide screen shots, spreadsheet, Jupyter notebook both in PDF and ipbny formats. Zip all the files. The zipped file should be named as first-name_last-name.zip.

Note: The time should be visible in all the screenshots

Exploring the Relational Data Model of CSV

In this section you will explore the relational data model using a spreadsheet software and python.

The student VM has Libre Office installed, you can use either Libre Office Calc, Excel or any other spreadsheet software. Libre Office Calc can be started from terminal using “localc” command.

CSV Import and basic filtering using spreadsheet [0.25 point]:

Step 1. Import the census.csv file into spreadsheet.

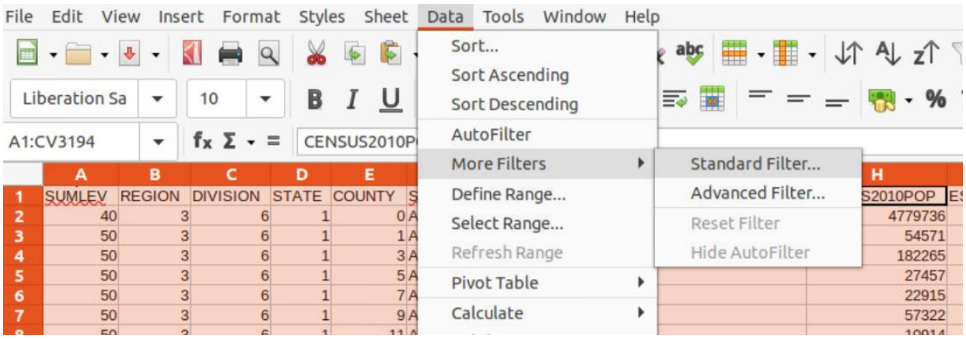
Use the provided census.csv file for this part.

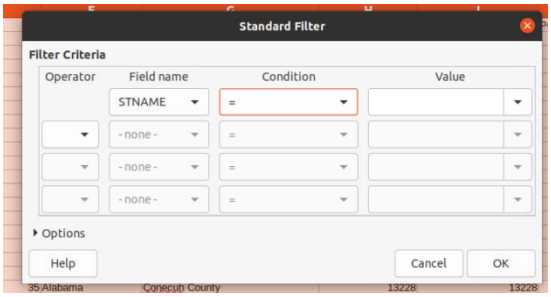
Instructions for importing csv into Libre Office Calc are in the link below.

https://help.libreoffice.org/6.1/en-US/text/scalc/guide/csv_files.html?DbPAR=CALC

Step.2 Filter rows in the data

A select operation on the data can be performed to show only the counties in the state of New York containing more than one million people. To perform the select operation, you can use the standard filter from the data tab, screenshots provided below. Provide the screenshots for filter and the result

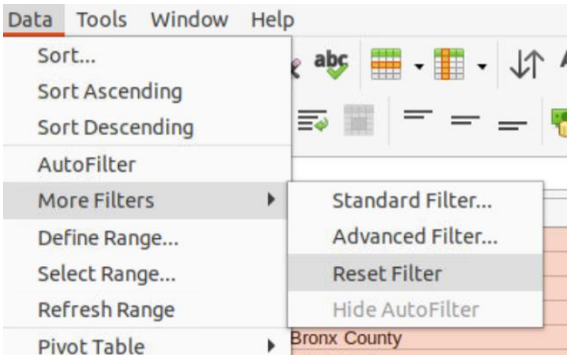




Aggregation using spread sheet [0.25 point]:

Step. 3 Reset the filter and add a new sheet

Filters can be reset from the data tab.



Step 4: In the new sheet use the SUMIFS and AVERAGEIF functions to calculate sum and average population of counties in the state of New York containing more than one million people. Provide screenshot of the formulas and the results. Documentation of SUMIFS and AVERAGEIF function are in the link below.

https://help.libreoffice.org/latest/is/text/scalc/01/func_sumifs.html

https://help.libreoffice.org/latest/lo/text/scalc/01/func_averageif.html

P.S. If you are using any other spreadsheet software use the equivalent functions.

CSV Import and basic filtering using Python [0.5 point]:

For this part do all your work in a Jupyter notebook and printout the notebook as PDF. You need to submit both the notebook and the PDF.

Step 1. Import the census.csv file into python using pandas.
Use the provided census.csv file for this part.

You can use the Pandas library to read the csv files used previously imported in the spreadsheets.

The syntax of reading a csv using pandas is below.

```
# Syntax of read_csv()
pandas.read_csv(filepath_or_buffer, sep=NoDefault.no_default, delimiter=None,
header='infer', names=NoDefault.no_default, index_col=None, usecols=None,
squeeze=None, prefix=NoDefault.no_default, mangle_dupe_cols=True, dtype=None,
engine=None, converters=None, true_values=None, false_values=None,
skipinitialspace=False, skiprows=None, skipfooter=0, nrows=None, na_values=None,
keep_default_na=True, na_filter=True, verbose=False, skip_blank_lines=True,
parse_dates=None, infer_datetime_format=False, keep_date_col=False,
date_parser=None, dayfirst=False, cache_dates=True, iterator=False,
chunksize=None, compression='infer', thousands=None, decimal='.',
lineterminator=None, quotechar='"', quoting=0, doublequote=True, escapechar=None,
comment=None, encoding=None, encoding_errors='strict', dialect=None,
error_bad_lines=None, warn_bad_lines=None, on_bad_lines=None,
delim_whitespace=False, low_memory=True, memory_map=False, float_precision=None,
storage_options=None)
```

Documentation for the csv function is in the link below.

https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html

Step 2. Import the census.csv file into python using pandas.

A select operation on the data can be performed to show only the counties in the state of New York containing more than one million people. This can be achieved using filtering and querying. A short example is shown below.

```
# Importing pandas
import pandas as pd

record = {

    'Name': ['Ankit', 'Amit', 'John', 'Bob', 'Alice', 'Shahryar'],
    'Age': [21, 19, 20, 18, 17, 21],
    'Stream': ['Math', 'Commerce', 'Science', 'Math', 'Math', 'Science'],
    'Percentage': [88, 92, 95, 70, 65, 78]}

# Create a dataframe
dataframe = pd.DataFrame(record, columns=['Name', 'Age', 'Stream', 'Percentage'])

print("Given Dataframe :\n", dataframe)

# Selecting rows based on condition
rslt_df_filter_1 = dataframe[dataframe["Percentage"] > 80]
rslt_df_filter_2 = dataframe[((dataframe["Percentage"] >= 60) &
(dataframe["Stream"] == 'Math') & (dataframe['Age'] > 17))]
rslt_df_query_1 = dataframe.query("Percentage > 80")
rslt_df_query_2 = dataframe.query("Percentage >= 60 and Stream == 'Math' and Age
> 17")

print(f'\nResult dataframe via filtering:\n', rslt_df_filter_1)

print('--' * 50)

print(f'\nResult dataframe via filtering:\n', rslt_df_filter_2)

print('--' * 50)

print(f'\nResult dataframe via query 1:\n', rslt_df_query_1)

print('--' * 50)

print(f'\nResult dataframe via query 2:\n', rslt_df_query_2)
```

Documentation for basic operations, filtering and querying are in the links below.

https://pandas.pydata.org/docs/user_guide/basics.html

https://pandas.pydata.org/docs/user_guide/indexing.html

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.query.html

Step 3. Calculate sum and average population of counties in the state of New York containing more than one million people

Step 4. Print all the datatypes with column names.
Documentation on pandas data type is in the link below.

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dtypes.html

Step 5. Print the number of empty values in each column.
The documentation for finding empty values is in the link below.

https://pandas.pydata.org/docs/user_guide/basics.html

Exploring the relational model [1.0 point]:

Q1: Which state has Box Butte County

Q2: What is the population of Box Butte County the CENSUS2010POP

Q3: What County in the state of Montana has the smallest estimated population?

Q4: Plot a histogram of CENSUS2010POP for counties with population grater than one million in state of California.
Documentation for plotting histogram using pandas is available at the link below.

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.hist.html

Exploring the Semi-structured Data Model of JSON data

Use the provided twitter.json file for this part.

JSON Import and exploration printing [1 point]

Step 1. Import twitter.json into Python.

Provided ex.json file has the following contents. The code snippet that follows uses this file as an example of reading JSON and iterating over its content. This example shows reading a multiline JSON with every other line is empty.

```
{"emp_name": "Bob" , "email": "bob@mail.com", "role": "tester"}

{"emp_name": "Alice" , "email": "bob@mail.com", "role": "developer"}

{"emp_name": "Martin" , "email": "martin@mail.com", "role": "manager"}

{"emp_name": "Cyndy" , "email": "cyndy@mail.com", "role": "developer"}
```

```
import json

# JSON string
a = '{"name": "Bob", "languages": "English"}'

# Deserializes into dict and returns dict.
y = json.loads(a)

print(f"JSON string = {y}")
print()

data = []
i = 0
# Reading from file
with open('ex.json', 'r') as f:
    for line in f:
        i = i + 1
        if i % 2 == 1:  # skip every other line since it is empty
            data.append(json.loads(line))

# Iterating through the json list
for line in data:
    print(line['emp_name'], line['email'])
```

Pandas also have the capability to read JSON file, documentation for pandas read_json is available in the link below.

https://pandas.pydata.org/docs/reference/api/pandas.read_json.html

Step 2: Print the schema of the twitter.json
The sample output is shown below

```
contributors
truncated
text
in_reply_to_status_id
id
favorite_count
source
retweeted
coordinates
timestamp_ms
entities
 ....symbols
 ....media
 ....hashtags
 ....user_mentions
 ....trends
 ....urls
in_reply_to_screen_name
```

Step 3: Given a tweet, what path would you enter to obtain a count of the number of friends for a user?

Step 4: Which of the following fields are nested within the 'extended_entities' field

List and count the distinct locations of the users in the provided JSON file? [1 point]

Exploring Sensory Data

You will be streaming weather data from a WebSocket hosted by UCSD. The code snippet below shows how to connect to a web socket and print its data.

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('rtd.hpwren.ucsd.edu', 12020))
for i in range(0, 60):
    d = s.recv(1024)
    p = d.decode('utf-8').split('\t', 2)
    print(p)
    print(('--' * 50))
    print(i, p[2])
s.close()
```

The legend for the DataStream elements also shown below.

```
Sn      Wind speed minimum m/s, km/h, mph, knots #,M, K, S, N
Sm      Wind speed average m/s, km/h, mph, knots #,M, K, S, N
Sx      Wind speed maximum m/s, km/h, mph, knots #,M, K, S, N
Dn      Wind direction minimum deg #, D
Dm      Wind direction average deg #, D
Dx      Wind direction maximum deg #, D
Pa      Air pressure hPa, Pa, bar, mmHg, inHg #, H, P, B, M, I
Ta      Air temperature °C, °F #, C, F
Tp      Internal temperature °C, °F #, C, F
Ua      Relative humidity %RH #, P
Rc      Rain accumulation mm, in #, M, I
Rd      Rain duration s #, S
Ri      Rain intensity mm/h, in/h #, M, I
Rp      Rain peak intensity mm/h, in/h #, M, I
Hc      Hail accumulation hits/cm2, hits/in2, hits #, M, I, H
Hd      Hail duration s #, S
Hi      Hail intensity hits/cm2h, hits/in2h, hits/ h #, M, I, H
Hp      Hail peak intensity hits/cm2h, hits/in2h, hits/ h #, M, I, H
Th      Heating temperature °C, °F #, C, F
Vh      Heating voltage V #, N, V, W, F2
Vs      Supply voltage V V
Vr      3.5 V ref. voltage V V
```

Connect to web socket and stream at least fifty samples. [0.5 points]

Gather at least five hundred sensor readings from the stream for air pressure and air temperature [1 point]

You need to parse the incoming stream for timestamps and the sensor readings.
The samples should be stored as a relational table with timestamps as the index. You can use pandas dataframe to store the sensor readings.

Example of one DataStream event is shown below.

b'198.202.124.3\tHPWREN:LP-WXT536:0R1:4:0\t1675285648\t0R1,Dn=000#,Dm=000#,Dx=000#,Sn=0.0#,Sm=99.9#,Sx=0.0#\n\r'

The code snippet below shows how to convert integer timestamp into datetime objects. Documentation for datetime in Python is available at the link below.

https://docs.python.org/3/library/datetime.html

```
from datetime import datetime
datetime.fromtimestamp(1675285648)
```

The code snippet below shows and example of using regex for matching string pattern of two characters followed by digits. Documentation for Python's regex is available at the link below.

https://docs.python.org/3/library/re.html

```
import re
re.match('pattren', 'string_to_match')
re.match('Dn' + '=(\d+\.?\d+).*', 'Dn=000#')
```

Hint: p object in the WebSocket example will have the timestamp and sensor values.

Plot the sensor readings as a line plot. [0.25 points]

Print summary statistics for the sensor values [0.25 points]

Print mean, median, minimum, maximum, range and mode for both sensor readings.

Exploring the Vector Model of an image

Import the image as 3-dimensinal array. **[0.25 points]**

The following code snippet reads image as a 3-D numpy array. Documentation for numpy array indexing can be found in the link below.

https://numpy.org/doc/stable/user/basics.indexing.html

```
! pip install opencv-python
import cv2
import numpy as np
from matplotlib import pyplot as plt

im_array = cv2.imread(f"{data_path / 'Australia.jpg'}")
# Converting BGR to RGB
im_array = cv2.cvtColor(im_array, cv2.COLOR_BGR2RGB)
plt.imshow(im_array)
plt.show()
```

What is the (Red, Green, Blue) pixel value for location 500, 2000? [0.25 points]

Plot the histogram of RGB channels? [0.5 point]

You can use matplotlib or any other python library to generate histogram. Documentation for Matplotlib's histogram can be found in the link below.

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html

Exploring Vector Model of Text

For this part you will use the three-news csv file provided with the assignment.

The code snippet below shows an example of converting text into TFIDF vectors and computing cosine similarity. Higher cosine similarity means that two vectors are similar.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity


d1 = "new york times"
d2 = "new york post"
d3 = "los angeles times"

q = "new new york"
corpus = [d1, d2, d3]

vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(corpus)
Q = vectorizer.transform([q])

print(f"d1 and q: {cosine_similarity(vectorizer.transform([d1]),
vectorizer.transform([q]))}")
print(f"d2 and q: {cosine_similarity(vectorizer.transform([d2]),
vectorizer.transform([q]))}")
print(f"d3 and q: {cosine_similarity(vectorizer.transform([d3]),
vectorizer.transform([q]))}")
```

Import texts and TFIDF vectorize [2 points]

What news file talks about voters more. [0.5 points]

What news file talks about delegates more. [0.5 points]