Midterm Exam

Name: JN Date: October 16, 2022 Subject/Professor: INST377/Sigalo

In [25]:
```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

import json
import sqlite3
from pandas.io.json import json_normalize
```

Start off by reading all csv and JSON EXCLUDING nobel_prize.json. Will be included later for it's specific question

In [84]:
```
cv_cases = pd.read_csv('cases_coronavirus.csv')
cv = pd.read_csv('covid19.csv')
de = pd.read_csv('dewiki_pageviews.csv')
en = pd.read_csv('enwiki_pageviews.csv')
tweets = open("tweets.json")
twe = json.load(tweets)
```

Question 8

In [45]:
```
avg_age = cv[['gender', 'age']].groupby(['gender']).mean()
avg_age = avg_age.reset_index()
avg_age[1:2]
```

Out[45]:

	gender	age
1	male	49.847689

Question 9

In [44]:
```
country = cv[['country','id']].groupby(['country']).count()
country = country.reset_index()
country.columns = ['Country', '# Of Cases']
country[7:8]
```

Out[44]:

	Country	# Of Cases
7	Canada	12

Question 10

In [42]:
```
GermanPageViews = de[['Coronavirus', 'NovelCoronavirus']]
GermanPageViews = GermanPageViews[['Coronavirus', 'NovelCoronavirus']].max()
GermanPageViews
```

Out[42]:
```
Coronavirus         290130
NovelCoronavirus     56280
dtype: int64
```

Question 11 (Joining) & Question 12 (How many rows)

In [51]:
```
df = pd.merge(en, de, how = 'inner', on = ['Date'])
df.shape[0]
```

Out[51]:
```
21
```

Question 13

In [57]:
```
twe['id']
```

Out[57]:
```
1292450854042308609
```

Question 14 & 15

In [83]:
```
tweets_df = json_normalize(twe['user'])
tweets_df
```

```
C:\Users\X\AppData\Local\Temp\ipykernel_8472\147137382.py:1: FutureWarning: pandas.i
o.json.json_normalize is deprecated, use pandas.json_normalize instead.
  tweets_df = json_normalize(twe['user'])
```

Out[83]:

	id	id_str	name	screen_name	location	url	description	translator_type	protecte
0	624248526	624248526	Azalia	a_degollado	Texas, USA	None	21 • TXARNG •	none	Fals

1 rows × 39 columns

Question 16 (Python Data Type) & Question 17 (Normalization)

In [113…
```
f = open("nobel_prizes.json")
prizes = json.load(f)
prizes_df = json_normalize(prizes, record_path="laureates", meta=["year","category"])
prizes_df
```

```
C:\Users\X\AppData\Local\Temp\ipykernel_8472\271951965.py:3: FutureWarning: pandas.io.json.json_normalize is deprecated, use pandas.json_normalize instead.
  prizes_df = json_normalize(prizes, record_path="laureates", meta=["year","category"])
```

Out[113]:

	id	firstname	surname	motivation	share	year	category
0	976	John	Goodenough	"for the development of lithium-ion batteries"	3	2019	chemistry
1	977	M. Stanley	Whittingham	"for the development of lithium-ion batteries"	3	2019	chemistry
2	978	Akira	Yoshino	"for the development of lithium-ion batteries"	3	2019	chemistry
3	982	Abhijit	Banerjee	"for their experimental approach to alleviatin...	3	2019	economics
4	983	Esther	Duflo	"for their experimental approach to alleviatin...	3	2019	economics
...	...	...	...	...	...	...	...
494	103	Ben R.	Mottelson	"for the discovery of the connection between c...	3	1975	physics
495	104	James	Rainwater	"for the discovery of the connection between c...	3	1975	physics
496	406	David	Baltimore	"for their discoveries concerning the interact...	3	1975	medicine
497	407	Renato	Dulbecco	"for their discoveries concerning the interact...	3	1975	medicine
498	408	Howard M.	Temin	"for their discoveries concerning the interact...	3	1975	medicine

499 rows × 7 columns

Question 18

In [98]:
```
prizes_sl = prizes_df[['category', 'id']].groupby(['category']).count()
prizes_sl = prizes_sl.reset_index()
prizes_sl.columns = ['Categories', '# of Winners']
prizes_sl.max()
```

Out[98]:
```
Categories      physics
# of Winners       112
dtype: object
```

Question 19

In [105…
```
ndf = prizes_df[["year","category","id"]].groupby(["year","category"]).agg("count")
ndf = ndf.reset_index()
ndf.columns = ['Year', 'Category', '# of Winners']
ndf
```

Out[105]:

	Year	Category	# of Winners
0	1975	chemistry	2
1	1975	economics	2
2	1975	literature	1
3	1975	medicine	3
4	1975	peace	1
…	…	…	…
265	2019	economics	3
266	2019	literature	1
267	2019	medicine	3
268	2019	peace	1
269	2019	physics	3

270 rows × 3 columns

Question 20

- Has two code queries below due to slight confusion (per year or just in general)

In [106…
```
prizes_sl.min()
```

Out[106]:
```
Categories      chemistry
# of Winners            45
dtype: object
```

In [108…
```
ndf.min()
```

Out[108]:
```
Year                 1975
Category        chemistry
# of Winners            1
dtype: object
```

In [ ]:
```

```