DMA Assignment 1 - Data Pr

Data transformations are useful for preparing a dataset for this process involves generating features from the dataset For this lab, we will be using a Yelp reviews dataset. Each r along with the features of the review (the reviewer, the revie convert this reviews dataset into a reviewers dataset by cre

The submission for this assignment should be done *indivic* groups of 2.

Environment Setup

Run this cell to setup your environment.

```
# Importing libraries
import pandas as pd
import math
import numpy as np
print('Libraries Imported')

#DOWNLOADING DATASET
!wget http://people.ischool.berkeley.edu/~zp/cc
!unzip yelp_reviews.zip
print('Dataset Downloaded: yelp_reviews.csv')
df=pd.read_csv('yelp_reviews.csv')
print(df.head())

print('Setup Complete')
```

```
Libraries Imported
--2019-02-06 05:56:00--  http://people.
Resolving people.ischool.berkeley.edu (
Connecting to people.ischool.berkeley.e
HTTP request sent, awaiting response...
Length: 376638166 (359M) [text/csv]
Saving to: 'yelp_reviews.csv.1'

yelp_reviews.csv.1  100%[==============

2019-02-06 05:58:09 (2.79 MB/s) - 'yelp

unzip:  cannot find or open yelp_revie
```

Q1: What was the highest number of reviews for any one b

- For this task, we will need to group the reviews datas data for each business, which is what we need. This
  - `yelp_businesses = yelp_dataset.grc`
  - The .size() counts the number of instances fo number of reviews as each instance in this da
  - The following command will sort this list, afte value: `sorted_yelp_businesses = yelp_businesses.sort(ascending=Fal`
  - This approach allows you to see the data stru approach to getting the max would be to use t `max(yelp_businesses)`

```
1              0              0
```

```
# YOUR CODE HERE
df.groupby('business_id').size().sort_values(as
```

```
4128
```

Answer:4128

Q2: What was the average number of reviews for a busine

```
# YOUR CODE HERE

df.groupby("business_id").size().mean()
```

```
12.63413902163123
```

Answer:12.63413902163123

Q3: What is the average number of reviews per reviewer?

```
# YOUR CODE HERE
```

```
df.groupby("user_id").size().mean()
```

```
3.188511934933203
```

Answer:3.188511934933203

Q4: What is the average number of cool votes per reviewe

```
# YOUR CODE HERE

df[["cool_votes","user_id"]].groupby("user_id"
```

```
cool_votes    1.241728
dtype: float64
```

Answer: 1.241728

Q5: What is the average number of funny votes per review

```
# YOUR CODE HERE

df[["funny_votes","user_id"]].groupby("user_id'
```

```
funny_votes    1.101265
dtype: float64
```

Answer:1.101265

Q6: What is the average number of useful votes per reviev

```
# YOUR CODE HERE

df[["useful_votes","user_id"]].groupby("user_ic
```

```
useful_votes    2.484476
dtype: float64
```

Answer:2.484476

Q7: What is the average of the log of the number of reviev

```
# YOUR CODE HERE

np.log(df.groupby("user_id").size()).mean()
```

```
0.7681766036714884
```

Answer:0.7681766036714884

Q8: What is the average of the log of the number of cool v

```
# YOUR CODE HERE

np.log(df[["cool_votes","user_id"]].groupby("us
```

```
/usr/local/lib/python2.7/dist-packages/

cool_votes   -inf
dtype: float64
```

Answer:-inf

Q9: What is the average of the log of the number of funny

```
# YOUR CODE HERE


np.log(df[["funny_votes","user_id"]].groupby("u
```

```
/usr/local/lib/python2.7/dist-packages/
  This is separate from the ipykernel p
funny_votes   -inf
dtype: float64
```

Answer: -inf

Q10: What is the average of the log of the number of usefu

```
# YOUR CODE HERE


np.log(df[["useful_votes","user_id"]].groupby('
```

```
/usr/local/lib/python2.7/dist-packages/
  after removing the cwd from sys.path.
useful_votes   -inf
dtype: float64
```

Answer:-inf

Q11: Find the average of the percentage of total cool vote

```
# YOUR CODE HERE
df["total"]=df["cool_votes"]+df["useful_votes"]
df[["total","user_id"]].groupby("user_id").sum(
a=df[["cool_votes","user_id"]].groupby("user_id
b=df[["total","user_id"]].groupby("user_id").su
np.mean(a["cool_votes"]/b["total"])
```

```
0.1927040524186044
```

Answer:19.27040524186044%

Q12: Find the average of the percentage of total funny vot

```
# YOUR CODE HERE

df["total"]=df["cool_votes"]+df["useful_votes"]
df[["total","user_id"]].groupby("user_id").sum(
a=df[["funny_votes","user_id"]].groupby("user_i
b=df[["total","user_id"]].groupby("user_id").su
np.mean(a["funny_votes"]/b["total"])
```

```
0.18256423845801606
```

Answer:18.256423845801606%

Q13: Find the average of the percentage of total useful vo

```
# YOUR CODE HERE

df["total"]=df["cool_votes"]+df["useful_votes"]
df[["total","user_id"]].groupby("user_id").sum(
a=df[["useful_votes","user_id"]].groupby("user_
b=df[["total","user_id"]].groupby("user_id").su
np.mean(a["useful_votes"]/b["total"])*100
```

```
62.47317091233795
```

Answer:62.47317091233795%

Q14: Average review text length (in non-space characters)

```
# YOUR CODE HERE
q=0
for i in df["text"]:
  q+=len(i) - i.count(' ')
q*1.0/len(df["text"])
```

```
499.18610090393645
```

Answer:499.18610090393645

Q15: Year in which the reviewer wrote the most reviews. C
subtract the minimum possible year (2005) from each so t
etc.

```
# YOUR CODE HERE
p=[]
import re
for i in df["date"]:
  p+=re.findall("[\d]{4}",i)
df["year"]=p
max_year=df.groupby(["user_id","year"]).size()
[int(i)-2005 for i in max_year.values]
```

```
        10,
        5,
        8,
        10,
        10,
        8,
        10,
        11,
        6,
        8,
        9,
        8,
        7,
        11,
        11,
        10,
        10,
        9,
        7
```

Answer:

[6, 9, 3, 11, 10, 11, 9, 11, 10, 9, 9, 9, 10, 10, 2, 6, 8, 9, 8, 7, 7, 6
8, 7, 11, 11, 10, 10, 9, 7, 9, 8, 10, 10, 11, 9, 9, 8, 9, 10, 5, 6, 10,
10, 9, 10, 8, 7, 4, 10, 11, 6, 9, 10, 10, 10, 10, 9, 11, 10, 4, 11, 9,
11, 9, 11, 10, 11, 10, 10, 10, 9, 5, 11, 7, 5, 8, 8, 5, 10, 0, 11, 10,
9, 8, 7, 9, 8, 9, 8, 5, 11, 10, 11, 11, 10, 10, 8, 9, 6, 9, 10, 10, 7, 8
10, 9, 11, 7, 8, 11, 9, 10, 11, 11, 7, 8, 7, 10, 8, 5, 5, 6, 10, 11, 7,
10, 11, 10, 10, 5, 4, 10, 6, 10, 7, 10, 7, 9, 11, 6, 9, 8, 8, 11, 11, 1
11, 11, 8, 6, 9, 9, 11, 11, 8, 11, 10, 11, 11, 10, 8, 10, 6, 11, 10,
8, 9, 8, 11, 10, 10, 7, 9, 5, 11, 9, 8, 7, 9, 5, 11, 10, 10, 5, 11, 9, 1
7, 10, 11, 10, 8, 8, 8, 3, 10, 9, 9, 8, 10, 9, 7, 8, 10, 7, 11, 10, 7, 1
11, 10, 10, 10, 4, 11, 8, 10, 10, 8, 11, 11, 8, 10, 7, 10, 11, 11, 1
6, 10, 10, 6, 7, 8, 8, 10, 11, 10, 11, 10, 11, 8, 10, 8, 6, 10, 9, 11,
11, 8, 11, 9, 10, 5, 6, 11, 10, 11, 10, 11, 7, 7, 10, 11, 9, 7, 9, 11,
9, 10, 10, 7, 10, 11, 11, 9, 2, 11, 8, 10, 11, 6, 9, 10, 8, 10, 8, 11,
10, 11, 7, 11, 10, 5, 7, 11, 11, 8, 10, 9, 11, 4, 10, 9, 11, 10, 10,
10, 2, 11, 9, 10, 9, 5, 9, 10, 11, 11, 9, 11, 10, 11, 10, 8, 11, 10,
3, 4, 9, 6, 10, 5, 10, 7, 10, 8, 3, 2, 11, 11, 9, 8, 9, 9, 9, 6, 10, 11,
9, 10, 11, 10, 10, 11, 9, 4, 10, 5, 8, 10, 6, 5, 10, 9, 9, 10, 11, 10,
10, 7, 7, 10, 11, 9, 6, 9, 7, 11, 10, 10, 5, 3, 10, 9, 5, 9, 10, 9, 10,
8, 9, 9, 5, 9, 11, 11, 9, 9, 10, 9, 11, 11, 9, 11, 7, 8, 9, 5, 4, 9, 11,
10, 9, 11, 10, 7, 8, 10, 7, 10, 9, 10, 11, 8, 7, 9, 8, 11, 11, 9, 10, 1
9, 9, 11, 7, 10, 11, 11, 10, 9, 8, 8, 9, 11, 10, 11, 11, 7, 6, 11, 9, 1
11, 10, 9, 4, 10, 8, 11, 10, 9, 8, 10, 6, 11, 10, 4, 4, 9, 10, 8, 10, 9
3, 11, 10, 6, 9, 9, 8, 6, 6, 7, 10, 6, 9, 8, 9, 8, 11, 6, 7, 10, 3, 10, 1
7, 2, 10, 9, 10, 9, 10, 9, 10, 8, 11, 10, 11, 10, 2, 11, 7, 9, 7, 9, 8,
7, 11, 11, 8, 11, 9, 10, 9, 9, 10, 9, 8, 7, 10, 6, 8, 7, 11, 6, 8, 10, 9
10, 6, 9, 10, 8, 10, 6, 10, 10, 7, 3, 11, 10, 9, 11, 4, 6, 7, 10, 10, 8
10, 8, 11, 10, 5, 9, 11, 10, 4, 9, 7, 8, 8, 8, 9, 10, 11, 10, 9, 10, 8,
10, 10, 6, 11, 9, 5, 8, 7, 11, 7, 8, 8, 7, 9, 10, ...]

Q16: Come up with a new feature. This may be derived fro
the name *my_new_feature.* **Dsiplay head() of this new featu**

```
# YOUR CODE HERE

def percentage(x):
  return sum([1 for i in x if (i > 3)])*1.0/le
percentage_rating=df.groupby("business_id")["st
my_new_feature =percentage_rating["stars"].appl
my_new_feature.head()
```

```
business_id
--5jkZ3-nUPZxUvtcbr8Uw    0.920000
--AKjxBmhm9DWrh-e0hTOw    1.000000
--BlvDO_RG2yElKu9XA1_g    0.900000
--Ol5mVSMaW8ExtmWRUmKA    1.000000
--Y_2lDOtVDioX5bwF6GIw    0.166667
Name: stars, dtype: float64
```

Answer: My new feature is the percentage of goo
considering 4 and 5 stars as good rating.