Q1. Tan Chapter 2, Problem 8. Discuss why a document-term matrix is an example of a data set that has asymmetric discrete or asymmetric continuous features. [10 pts]

Answer:

The value of the i,jth entry of a document-term matrix is the number of times the jth term occurs in the ith document. The data was processed using standard techniques to remove common words, to adjust for the different frequencies with which terms appear and to adjust for the different lengths of documents.

Most documents contain only a small fraction of all the possible terms, and hence, zero entries are not very meaningful, either in describing or comparing documents. Thus, a document-term matrix has asymmetric discrete features. If we apply a TF/IDF normalization to terms and normalize the documents to have an L2 norm of 1, then this creates a term-document matrix with continuous features. However, the features are still asymmetric because these transformations do not create non-zero entries for any entries that were previously 0, and thus, zero entries are still not very meaningful.

Q2. Tan Chapter 2, Problem 12. Distinguish between noise and outliers. Be sure to consider the following questions. [10 pts]

• Is noise ever interesting or desirable? Outliers?

Answer: i. By definition, noise is never desirable. It distorts the original attribute values.
ii. Outliers can be interesting and desirable. They can be legitimate values and even finding outliers can be the task of some data mining problems.

• Can noise objects be outliers?

Answer: Random distortion of the data is often responsible for outliers. Noise can make the data look more randomized and unusual. Therefore, noise objects can be outliers.

• Are noise objects always outliers?

Answer: Noisy data can appear as normal data. Random distortion can result in an object or value much like a normal one. Hence, noise objects are not always outliers.

• Are outliers always noise objects?

Answer: Often outliers merely represent a class of objects that are different from normal objects. Outliers can be legitimate data objects that appear to not belong in the data set. Those outliers would typically not classify as noise objects. Hence, outliers are not always noise objects.

• Can noise make a typical value into an unusual one, or vice versa?

Answer: Noise can make a typical value into an unusual one and vice versa. The source of noise in data can randomly make some values appear as unusual or some outliers as typical data objects. Hence, noise can make a typical value into an unusual one and vice versa.

Q3. Implement a notebook on Kaggle to explore this dataset. This dataset lists the number of antibiotic resistance genes (AMR), and the presence or absence of the CRISPR-Cas systems in the genomes included in the file. Report what you have learned by including the html output from your notebook in the PDF file you are going to submit. What to check? The distribution of the two variables (AMR, CRISPR-Cas), and if there is any correlation between the two variables. [25 pts]

First, we will import all the required libraries.

In [1]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

Then we will read the data from the csv and get the gist of the dataset.

In [2]:
```
amr_data = pd.read_csv("Efaecium_AMRC.csv")

# Gist of the dataset
amr_data.head()
```

Out[2]:

	genome_ID	CRISPR_Cas	AMR
0	GCA_010120755.1_ASM1012075v1	0	8
1	GCA_001720945.1_ASM172094v1	0	21
2	GCA_009697285.1_ASM969728v1	0	13
3	GCA_900639535.1_EI202_hybrid_assembly	0	11
4	GCA_002007625.1_ASM200762v1	0	18

In [3]:
```
amr_data
```

Out[3]:

	genome_ID	CRISPR_Cas	AMR
0	GCA_010120755.1_ASM1012075v1	0	8
1	GCA_001720945.1_ASM172094v1	0	21
2	GCA_009697285.1_ASM969728v1	0	13
3	GCA_900639535.1_EI202_hybrid_assembly	0	11
4	GCA_002007625.1_ASM200762v1	0	18
...	...	...	...
2218	GCA_001058635.1_ASM105863v1	0	24
2219	GCA_900148625.1_Hp_24-1_05	0	16
2220	GCA_000981965.1_ASM98196v1	0	0
2221	GCA_002156235.1_ASM215623v1	0	10
2222	GCA_900080195.1_isolate_3	0	18

2223 rows × 3 columns

We will describe the datatypes of all the columns present.

In [20]:
```
amr_data.dtypes
```

Out[20]:
```
genome_ID     object
CRISPR_Cas     int64
AMR            int64
dtype: object
```

We can see that the column "CRISPR_Cas" is a binary attribute with only 2 values: 0 and 1. All columns have nominal attributes in them.

We'll use the describe function in pandas to find the mean, minimum, maximum, etc. of each column.

In [4]:
```
amr_data.describe(include='all')
```

Out[4]:

	genome_ID	CRISPR_Cas	AMR
count	2223	2223.000000	2223.000000
unique	2223	NaN	NaN
top	GCA_002945895.1_ASM294589v1	NaN	NaN
freq	1	NaN	NaN
mean	NaN	0.024741	10.330184
std	NaN	0.155371	6.661470
min	NaN	0.000000	0.000000
25%	NaN	0.000000	3.000000
50%	NaN	0.000000	12.000000
75%	NaN	0.000000	16.000000
max	NaN	1.000000	31.000000

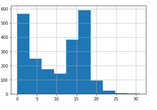
In [5]:
```
amr_data.columns = ['genome_ID', 'CRISPR_Cas', 'AMR']
```

Distribution

First, we plot a histogram that shows the distribution of the attribute values.

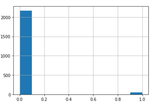
In [6]:
```
#distribution
amr_data['AMR'].hist(bins=10)
```

Out[6]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f58e836588>
```



In [7]:
```
#distribution
amr_data['CRISPR_Cas'].hist(bins=10)
```

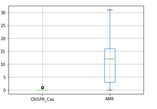
Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f59098b048>
```



We will also show the distribution of values for each attribute using boxplot

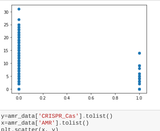
In [8]:
```
#distribution
amr_data.boxplot()
```

Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f590a1d348>
```

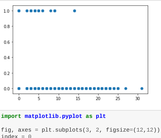


For each pair of attributes, we can use a scatter plot to visualize their joint distribution.

In [9]:
```
x=amr_data['CRISPR_Cas'].tolist()
y=amr_data['AMR'].tolist()
plt.scatter(x, y)
plt.show()
```

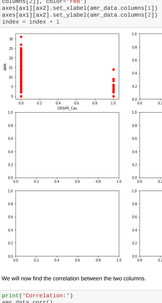


In [10]:
```
y=amr_data['CRISPR_Cas'].tolist()
x=amr_data['AMR'].tolist()
plt.scatter(x, y)
plt.show()
```



In [11]:
```
import matplotlib.pyplot as plt

fig, axes = plt.subplots(3, 2, figsize=(12,12))
index = 0
ax1 = int(index/2)
ax2 = index % 2
axes[ax1][ax2].scatter(amr_data[amr_data.columns[1]], amr_data[amr_data.
columns[2]], color='red')
axes[ax1][ax2].set_xlabel(amr_data.columns[1])
axes[ax1][ax2].set_ylabel(amr_data.columns[2])
index = index + 1
```



We will now find the correlation between the two columns.

In [29]:
```
print('Correlation:')
amr_data.corr()
```

Out[29]:
```
Correlation:
```

	CRISPR_Cas	AMR
CRISPR_Cas	1.000000	-0.156173
AMR	-0.156173	1.000000

The correlation matrix shows the following information

x-values y-values

x-values 1.00000 -0.156173

y-values -0.156173 1.00000

The negative value of correlation coefficient indicates how much the 2 variables move in opposite directions. An increase in CRISPR_Cas means a decrease in AMR and vice versa.

Q4. Learn about Omicron using Google Trend. Write a brief summary including four highlights of what you have learned. [25 pts]. (Solved in PDF)

Q5. Write a summary for this paper: COVID-19 or Flu? Discriminative Knowledge Discovery of COVID-19 Symptoms from Google Trends Data. [30 pts]

• The length of your summary is about one page.

• State main ideas: problem that the paper tries to address, what data was used, and what was the method that was applied/developed to solve the problem?

• Add your personal opinion. Do you like the paper or not? Why? How do you think about the paper?

• Write the summary in your own words; don't copy and paste.

(Solved in PDF)

4. Learn about Omicron using Google Trend. Write a brief summary including four highlights of what you have learned. [25 pts].

Looking at the Omicron Google Trends page, we can see many interesting things. We can see the interest in the search term “omicron” in the past 12 months.

1. We can see that there was virtually no interest in Omicron for at least 10 of the last 12 months. Then in the week of November 21-27, when the first case of Omicron was discovered, the search interest index for “omicron” rose from less than 1 to 15. The following week, as omicron variant made headlines, the search interest index jumped sharply from 15 to 76. After that sharp jump, there was a decline in the search interest index due to the media not reporting much on the Omicron variant. However, the week before Christmas, as there was an uptick of cases, the search interest index grew from 38 to 86 in the week of December 19-25 i.e. in Christmas. The search term “Omicron” reached maximum search interest value:100, on December 26-January 1, when there was a massive spike in the number of cases of the Omicron variant. The search interest has since dipped in the subsequent weeks that followed.

2. The Google Trends page also allows us to view the interest in the search term “omicron” according to subregion or states in the USA. We can see that the top 5 regions most interested in the search term are the District of Columbia with a maximum search interest index of 100, followed by Hawaii, New Jersey, California, and Connecticut. The state with the least interest in “omicron” is North Dakota with 40 search interest index. From the map given on the page, it can be seen that the coastal states, both the East Coast and West Coast, have more search interest in “Omicron” than states in the middle like Indiana, Iowa. The states in the north have more interest than the states in the south when it comes to “omicron” search term on Google. The metro with the most interest in “omicron” is Glendive MT. The city with the most interest in “omicron” is San Francisco.

3. The related topics being searched with omicron are also shown on the Trends page. The most trending topic being searched along with “omicron” is “Variant”. “Symptom”, “Infection”, “Symptoms of COVID-19” and “Virus” round out the current rising topics. When it comes to top topics that have a high search interest index along with “omicron”, “Signs and Symptoms” have the highest value of 100 followed by “omicron”, “Coronavirus Disease 2019” and “Variant”. Since many people search on Google to quickly get an idea about symptoms related to COVID, it makes sense that “Signs and Symptoms” have such a high search interest index along with “omicron”.

4. Also, we can see the related queries to the “omicron” search term. Currently, the most popular related query is “omicron variant”. The related query with the highest search interest index of 100 is “omicron symptoms”, which makes sense as many people would quickly Google “omicron symptoms” to get a quick overview of the symptoms that come with Omicron.

COVID-19 or Flu? Discriminative Knowledge Discovery of COVID-19 Symptoms from Google Trends Data

The paper explores methods to extract knowledge about symptoms related to COVID-19 from Google Trends data from different periods of time, including before the COVID-19 pandemic started. One of the proposed methods is a novel nonnegative discriminative analysis (DNA) to extract the unique information of one dataset relative to another dataset. The paper also shows that numerical tests corroborated the success of the proposed approaches to discover the three unique COVID-19 symptoms w.r.t. flu including ageusia, shortness of breath, and anosmia. By comparing the Google Trends data during the COVID-19 pandemic and the flu, the paper says that we would be able to distinguish the features/symptoms associated with the novel spreading COVID-19 and Flu. The datasets that they have used for this task are the Google Trends 2019 and the Google Trends 2020 datasets i.e. one dataset before the pandemic and one dataset after the pandemic. To discover which symptoms are more discriminative, the paper proposes DNA, a novel non-negative discriminative principal component analysis. The paper tries to demonstrate the viability of the method in extracting symptoms that are discriminative of COVID-19 with respect to the Flu.

The proposed methodology is the following. Assume the dataset for Google Trends 2019 and 2018 to be $\{y_i \in R^D\}^n_{i=1}$, containing data for the Flu(background dataset) and assume the dataset for Google Trends 2020 to be $\{x_i \in R^D\}^n_{i=1}$ containing both data for COVID-19 and Flu(target dataset). D denotes the number of searched symptoms and i is time index. The paper proposes to apply discriminative principal component analysis (PCA) and contrastive (c) PCA on both datasets. Discriminative PCA will seek a projection matrix to maximize the ratio of the projected target data variance over that of the background data. The approach searches for subspace vectors, namely the columns of $U \in R^{D\times d}$ with $d \leq D$ by solving

$$\max_{U} \operatorname{Tr}\left[(U^T C_y U)^{-1} U^T C_x U\right]$$

where $C_x := 1/m \sum_{i=1}^{m} (x_i - \mu_x)(x_i - \mu_x)^\top \in R^{D\times D}$ representing the sample covariance of the target data with μ_x denoting the corresponding sample mean; C_y is the sample covariance of the background data. This is a ratio trace maximization problem. (dPCA).

To overcome the challenge of sign ambiguity, the paper proposes the DNA method. We take the input of Nonzero-mean target and background data $\{x_i\}$ (i=1 to m) and $\{y_i\}$ (i=1 to n) and a number of dimensions d. We construct covariance matrices of x_i and y_i to get C_x and C_y. We then perform nonnegative matrix decomposition on $C_y^{-1} C_x$ to obtain the two factorization components W and H, a desired output. To solve WH, we use the Kullback–Leibler (KL) divergence metric. W gives the importance of each symptom. The experimental evaluation uses a subset of the Google Trends data and uses symptoms like shortness of breath, ageusia, and anosmia along with mutual symptoms like cough, diarrhea, etc. We assume d=1. After comparing dPCA, NNMF and DNA, we can observe that DNA outperforms the existing alternatives in terms of higher frequencies of successfully searching for the discriminative symptoms. The paper concludes with the authors opening up the research for further evaluation with nonnegative dPCA and broadening the applications of DNA.

Overall, I really liked the paper. The paper is short but detailed. It has useful graphs to display and compare the results of the methods. The paper's proposed method will help us in distinguishing unique COVID-19 symptoms from the common ones with the Flu.