

# Bigrams Processing with Amazon S3 & EMR

Khai Nguyen

[khainguyen@temple.edu](mailto:khainguyen@temple.edu)

CIS 4517 Data Intensive and Cloud Computing

### (50 points) Part 1: MapReduce

Count bigrams: Take the word count from the previous problem and extend it to count bigrams. bigrams are sequences of two consecutive words. (Recall our lecture on shingles.) Don't worry about doing anything fancy in terms of tokenization. You can use Java's StringTokenizer.

Deliverables (on Pride and Prejudice):

#### 1. (10 points) How many unique bigrams are there?

View number of words OR number of lines since each line is designated for 1 word:

```
$ find /c /v "" "/output/combined.txt"
```

```
C:\Users\KhaiNguyen\Documents\CS_4517\Project\Project_5\BigramCount\output> find /c /v "" "/combined.txt"
----- ./COMBINED.TXT: 53092
C:\Users\KhaiNguyen\Documents\CS_4517\Project\Project_5\BigramCount\output>
```

Give 3 examples of such bigrams.

Grabbing data starting at a random line, for example line 20330

```
C:\Users\KhaiNguyen\Documents\CS_4517\Project\Project_5\BigramCount\output>more +20330 combined.txt
Her_&_mother's 2
Her_&_mother 3
her_&_mother 51
her_&_Mr 5
her_&_Mrs 2
her_&_much 1
her_&_must 1
her_&_My 1
her_&_name 2
her_&_natural 1
her_&_nature 3
her_&_nearest 1
her_&_neighbours' 1
her_&_neighbours 2
her_&_nephew's 1
her_&_nephew 5
```

## 2. (5 points) List the top ten most frequent bigrams and their counts.

Using pandas, we can import our recently generated output file, “combined.txt” into a dataframe. Then we can sort by the number of appearances of the bigrams. **Here, 2 bigrams are connected with a “\_&\_” string.**

```
In [44]: import pandas as pd
import os
file_path = os.path.join("./BigramCount", "output", "combined.txt")
price_and_prejudice_df = pd.read_csv(file_path, sep="\s+", header = None)

price_and_prejudice_df.columns = ["Bigram", "Num_appearance"]

top_ten_bigrams = price_and_prejudice_df.sort_values(by="Num_appearance", ascending=False).head(10)

top_ten_bigrams
# top_ten_bigrams.sum()
```

```
Out[44]:
```

	Bigram	Num_appearance
32521	of_&_the	458
46451	to_&_be	420
23289	in_&_the	383
22315	I_&_am	299
32172	of_&_her	252
47034	to_&_the	241
29362	Mr_&_Darcy	228
32179	of_&_his	215
18296	had_&_been	195
22408	I_&_have	181

## 3. (10 points) What fraction of all bigrams occurrences does the top ten bigrams account for? That is, what is the cumulative frequency of the top ten bigrams?

```
In [56]: cumulative_frequency_top_ten = float(top_ten_bigrams["Num_appearance"].sum())/\
float(price_and_prejudice_df["Num_appearance"].sum())

print("Cumulative frequency of top ten is {:.4f}\n".format(cumulative_frequency_top_ten))

Cumulative frequency of top ten is 0.0248
```

## 4. (5 points) How many bigrams appear only once?

```
In [72]: appear_once = price_and_prejudice_df[price_and_prejudice_df["Num_appearance"]==1]
print("Number of bigrams that appear once " + str(len(appear_once)))

Number of bigrams that appear once 39297
```

Give 3 examples of bigrams that appear only once.

```
In [73]: appear_once = price_and_prejudice_df[price_and_prejudice_df["Num_appearance"]==1]
appear_once.tail(10)
```

```
Out[73]:
```

	Bigram	Num_appearance
53080	yourself_&_when	1
53082	yourself_&_your	1
53083	yourselves_&_Do	1
53084	yourselves_&_so	1
53086	youth_&_as	1
53087	youth_&_health	1
53088	youth_&_the	1
53089	youth_&_there	1
53090	youth_&_was	1
53091	youths_&_as	1

**5. (10 points) What are the five most frequent words following the word "light"? What is the frequency of observing each word?**

From below, we can tell that the 5 most frequent words are *in, A, and, as, began*

```
In [80]: # What are the five most frequent words following the word "light"? What is the frequency of observing each word?
bigram_with_light = price_and_prejudice_df[price_and_prejudice_df["Bigram"].str.match('light_&')]
bigram_with_light.sort_values(by="Num_appearance", ascending=False).head(100)
```

```
Out[80]:
```

	Bigram	Num_appearance
25826	light_&_in	2
25817	light_&_A	1
25818	light_&_and	1
25819	light_&_as	1
25820	light_&_began	1
25821	light_&_does	1
25822	light_&_have	1
25823	light_&_He	1
25824	light_&_I	1
25825	light_&_importance	1
25827	light_&_it	1
25828	light_&_might	1
25829	light_&_nor	1
25830	light_&_seemed	1

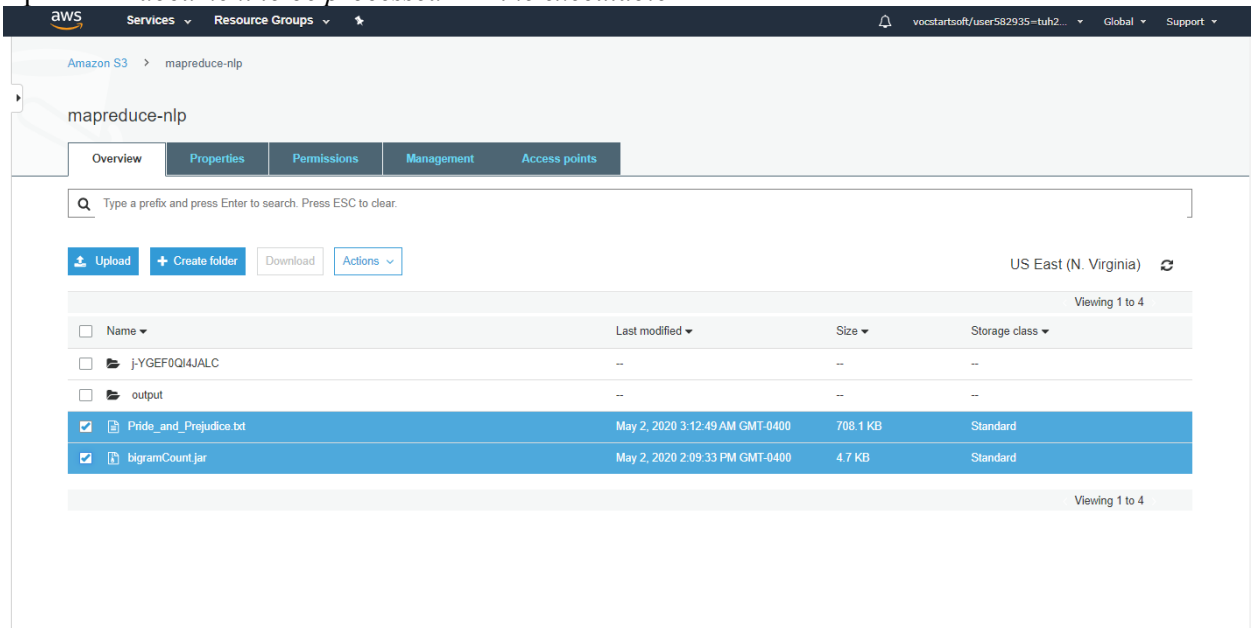
**6. (10 points) Running in the cloud.**

**a. Setup S3**

The screenshot shows the AWS Management Console interface for Amazon S3. The left sidebar contains navigation links for Buckets, Batch Operations, and Access analyzer for S3. The main content area displays the 'Buckets (2)' section. A search bar is present above a table listing the buckets. The table has columns for Name, Region, Access, and Bucket created. Two buckets are listed: 'image-processing-khai-nguyen' and 'mapreduce-nlp'. The 'mapreduce-nlp' bucket is highlighted, indicating it is the selected item. Above the table, there are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. A notification banner at the top of the console area states: 'We're gradually updating the design of the Amazon S3 console. You will notice some updated screens as we improve the performance and user interface. To help us improve the experience, give feedback on the recent updates.'

Name	Region	Access	Bucket created
image-processing-khai-nguyen	US East (N. Virginia) us-east-1	Only authorized users of this account	2020-03-27T02:42:08.000Z
mapreduce-nlp	US East (N. Virginia) us-east-1	Not public	2020-05-02T07:11:38.000Z

Upload the *document to be processed* and *the executable*



Amazon S3 > mapreduce-nlp

mapreduce-nlp

Overview Properties Permissions Management Access points

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions

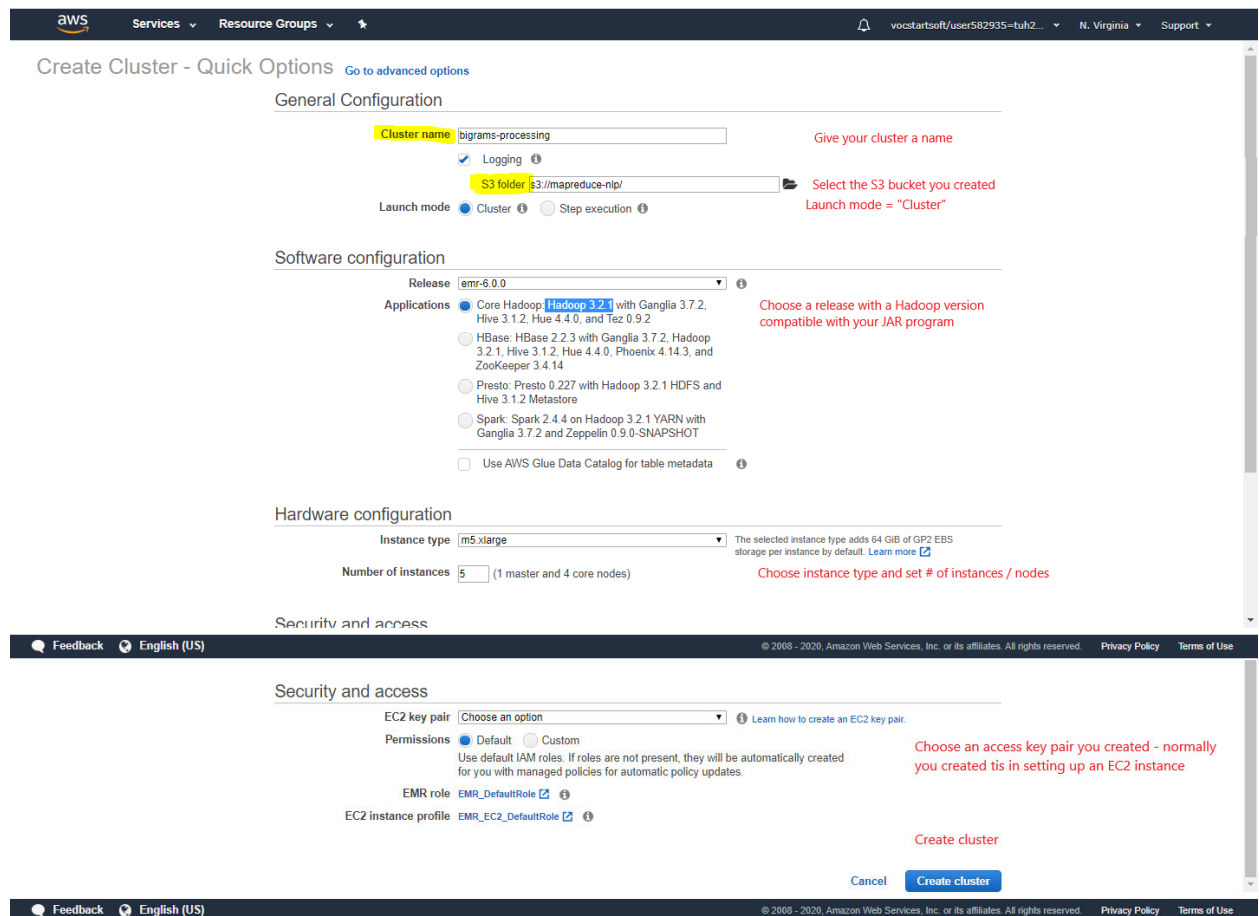
US East (N. Virginia)

Viewing 1 to 4

<input type="checkbox"/>	Name	Last modified	Size	Storage class
<input type="checkbox"/>	j-YGEF0QI4JALC	--	--	--
<input type="checkbox"/>	output	--	--	--
<input checked="" type="checkbox"/>	Pride_and_Prejudice.txt	May 2, 2020 3:12:49 AM GMT-0400	708.1 KB	Standard
<input checked="" type="checkbox"/>	bigramCount.jar	May 2, 2020 2:09:33 PM GMT-0400	4.7 KB	Standard

Viewing 1 to 4

## b. Setup EMR - Amazon Elastic MapReduce



aws Services Resource Groups

Create Cluster - Quick Options [Go to advanced options](#)

General Configuration

Cluster name  Give your cluster a name

☒ Logging

S3 folder  Select the S3 bucket you created

Launch mode ☒ Cluster ☐ Step execution Launch mode = "Cluster"

Software configuration

Release  Choose a release with a Hadoop version compatible with your JAR program

Applications ☒ Core Hadoop ☐ HBase ☐ Presto ☐ Spark ☐ Use AWS Glue Data Catalog for table metadata

Hardware configuration

Instance type  The selected instance type adds 64 GiB of GP2 EBS storage per instance by default. [Learn more](#)

Number of instances  (1 master and 4 core nodes) Choose instance type and set # of instances / nodes

Security and access

EC2 key pair  [Learn how to create an EC2 key pair](#)

Permissions ☒ Default ☐ Custom Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

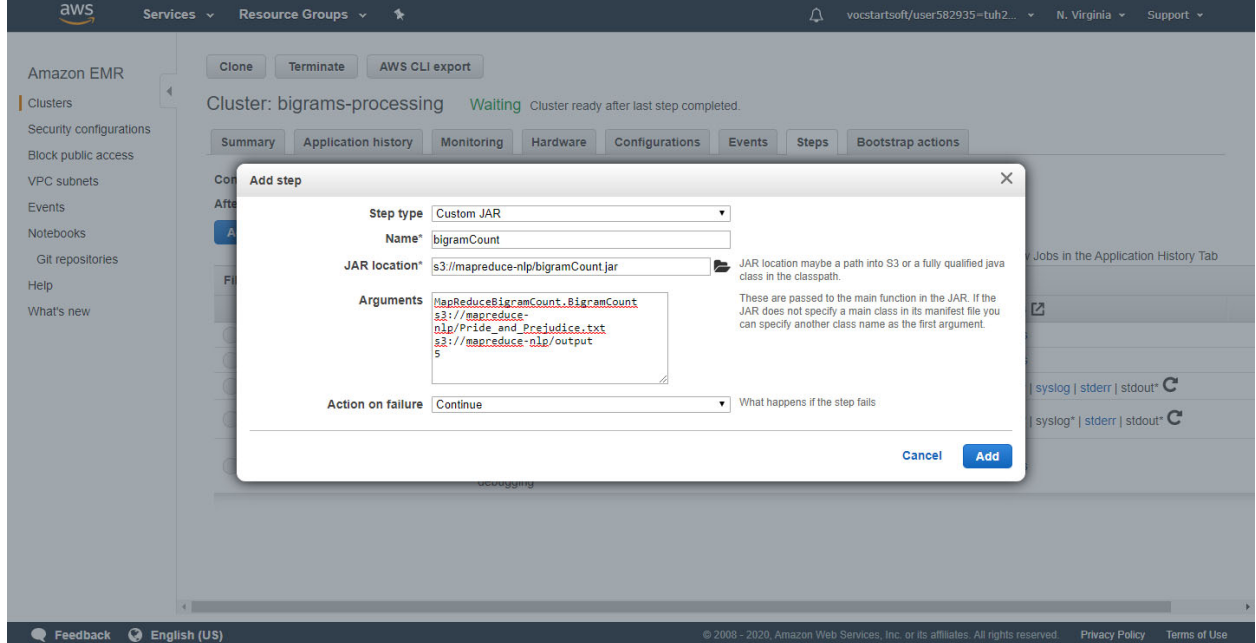
EMR role  [Learn more](#)

EC2 instance profile  [Learn more](#)

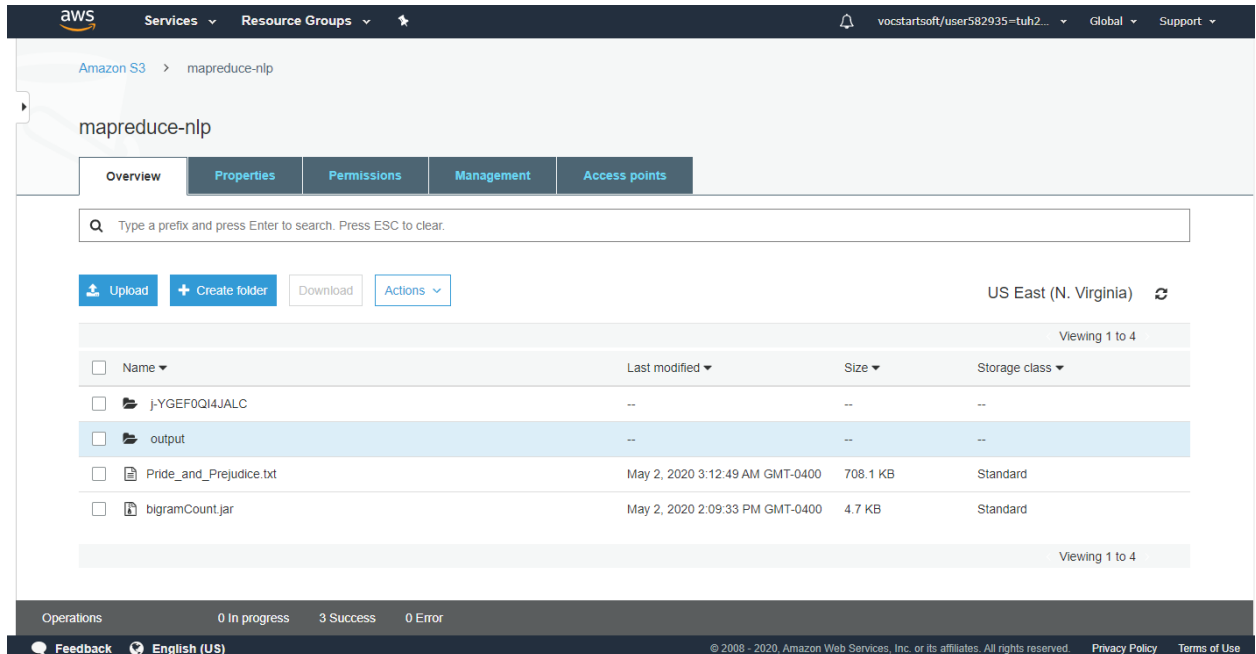
Create cluster

Cancel Create cluster

c. Create a MapReduce job OR Step in the EMR cluster



d. Go back to the S3 console to check the result



Amazon S3 > mapreduce-nlp > output

mapreduce-nlp

Overview

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions

US East (N. Virginia)

Viewing 1 to 6

Name	Last modified	Size	Storage class
_SUCCESS	May 2, 2020 2:11:29 PM GMT-0400	0 B	Standard
part-r-00000	May 2, 2020 2:11:28 PM GMT-0400	166.1 KB	Standard
part-r-00001	May 2, 2020 2:11:28 PM GMT-0400	170.2 KB	Standard
part-r-00002	May 2, 2020 2:11:28 PM GMT-0400	167.4 KB	Standard
part-r-00003	May 2, 2020 2:11:28 PM GMT-0400	166.1 KB	Standard
part-r-00004	May 2, 2020 2:11:26 PM GMT-0400	166.1 KB	Standard

Viewing 1 to 6

Operations 0 In progress 3 Success 0 Error

## e. Download the files to your machine

aws Services Resource Groups

Amazon S3 > mapreduce-nlp > output

mapreduce-nlp

Overview

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions






US East (N. Virginia)

Viewing 1 to 6

Name	Last modified	Size	Storage class
_SUCCESS	May 2, 2020 2:11:29 PM GMT-0400	0 B	Standard
<input checked="" type="checkbox"/> part-r-00000	May 2, 2020 2:11:28 PM GMT-0400	166.1 KB	Standard
<input checked="" type="checkbox"/> part-r-00001	May 2, 2020 2:11:28 PM GMT-0400	170.2 KB	Standard
<input checked="" type="checkbox"/> part-r-00002	May 2, 2020 2:11:28 PM GMT-0400	167.4 KB	Standard
<input checked="" type="checkbox"/> part-r-00003	May 2, 2020 2:11:28 PM GMT-0400	166.1 KB	Standard
<input checked="" type="checkbox"/> part-r-00004	May 2, 2020 2:11:26 PM GMT-0400	166.1 KB	Standard

Viewing 1 to 6

On machine:

Name	Date modified	Type	Size
 part-r-00000	5/2/2020 2:22 PM	File	167 KB
 part-r-00001	5/2/2020 2:22 PM	File	171 KB
 part-r-00002	5/2/2020 2:22 PM	File	168 KB
 part-r-00003	5/2/2020 2:22 PM	File	167 KB
 part-r-00004	5/2/2020 2:22 PM	File	167 KB

**f. Combine and sort**

\$ copy part-r-\* temp combined.txt

```
C:\Users\KhaiNguyen\Documents\CS_4517\Project\Project_5\BigramCount\output_EMR>copy part-r-* temp_combined.txt
part-r-00000
part-r-00001
part-r-00002
part-r-00003
part-r-00004
1 file(s) copied.
```

\$ sort temp combined.txt > combined.txt

```
C:\Users\KhaiNguyen\Documents\CS_4517\Project\Project_5\BigramCount\output_EMR>sort temp_combined.txt > combined.txt
C:\Users\KhaiNguyen\Documents\CS_4517\Project\Project_5\BigramCount\output_EMR>
```

**g. Check the result**

The screenshot shows a file explorer window on the left with a list of files: combined, part-r-00000, part-r-00001, part-r-00002, part-r-00003, part-r-00004, and temp\_combined. The 'combined' file is selected. To the right, a Notepad window titled 'combined - Notepad' displays the sorted contents of the file. The text is sorted alphabetically and includes counts for each line. The status bar at the bottom of the Notepad window shows 'Ln 1, Col 1', '100%', 'Windows (CRLF)', and 'UTF-8'.

File Name	Content
combined	a_&_tête 4
part-r-00000	a_&_bad 2
part-r-00001	a_&_ball 12
part-r-00002	a_&_beautiful 3
part-r-00003	a_&_beauty 1
part-r-00004	a_&_beginning 2
temp_combined	a_&_belief 2
	a_&_better 3
	a_&_billiard 1
	a_&_bit 2
	a_&_black 1
	a_&_blemish 1
	a_&_blessing 1
	a_&_blue 1
	a_&_blush 1
	a_&_book 4
	a_&_bottle 1
	a_&_bow 3
	a_&_bowl 1
	a_&_breach 1
	a_&_broken 1
	a_&_brother 5
	A_&_By 1
	a_&_call 2
	a_&_calm 1
	a_&_camp 1
	a_&_capital 1
	A_&_carat 1
	a_&_carriage 2
	a_&_case 5
	a_&_cause 5



## (50 points) Part 2: Pandas

For this problem use the data crawled from your Project 2.

### 1. (5 points) Describe the that you use to solve this problem. (Hint: get it from your report.)

BeautifulSoup4 : <https://pypi.org/project/beautifulsoup4/>

python tika : <https://pypi.org/project/tika/>

Pandas: <https://pypi.org/project/pandas/>

IDE: VSCode and Jupyter Notebook

### 2. (5 points) Import the Pandas package.

```
$ pip3 install pandas
```

### 3. (5 points) Load data into separate Data Frames. (All of you have collected multiple files. Work with at least 2 files for this problem.)

- Use csv functions to load data from csv files if your data is in csv.
- Use json functions to load data from json documents if your data is in json.

```
In [25]: # a) Import the Pandas package.
import pandas as pd
import os

# b) Load data into separate DataFrames. Work with at least 2 files for this problem.
territory_data_1 = pd.read_csv(os.path.join("./csv", "2020-04-11_Switzerland.csv"))
territory_data_2 = pd.read_csv(os.path.join("./csv", "US", "2020-04-11_California.csv"))
```

### 4. (5 points) Check the data-type of each of column by outputting the dtypes attribute of your DataFrame.

```
In [17]: territory_data_1["Grocery & pharmacy"].dtype
```

```
Out[17]: dtype('O')
```

```
In [18]: territory_data_1.dtypes
```

```
Out[18]: Territory          object
Retail & recreation      object
Grocery & pharmacy       object
Parks                    object
Transit stations         object
Workplace                object
Residential              object
dtype: object
```

5. (5 points) Show an example of sorting one of your DataFrames by a column. Give the top-15 entries in descending order.

Here we sort by the “Parks” column

```
In [31]: territory_data_1.sort_values(by="Parks", ascending=False).head(15)
```

Out[31]:

	Territory	Retail & recreation	Grocery & pharmacy	Parks	Transit stations	Workplace	Residential
17	Schaffhausen	-63%	+30%	Not enough data for this date	-34%	-39%	+2%
2	Appenzell_Ausserrhoden	-62%	+13%	Not enough data for this date	-21%	-33%	Not
3	Appenzell_Innerrhoden	-80%	Not enough data for this date	Not enough data for this date	Not enough data for this date	-84%	Not
23	Uri	-74%	+0%	Not enough data for this date	-54%	-48%	Not
7	Canton_of_Zug	-68%	+46%	Not enough data for this date	-40%	-44%	+22%
10	Glarus	-87%	-6%	Not enough data for this date	+5%	-35%	Not
12	Jura	-82%	+12%	Not enough data for this date	-34%	-40%	+14%
15	Nidwalden	-62%	+0%	-5%	-50%	-46%	Not
16	Obwalden	-83%	-4%	-49%	-40%	-40%	Not
24	Valais	-85%	-8%	-35%	-60%	-52%	+18%
11	Grisons	-85%	-15%	-35%	-65%	-49%	+14%
14	Neuchâtel	-70%	+8%	-18%	-37%	-47%	+23%
22	Ticino	-83%	-21%	-18%	-50%	-50%	+20%
1	Aargau	-74%	+19%	+99%	-32%	-36%	+15%
26	Zurich	-78%	+20%	+96%	-55%	-43%	+15%

6. (10 point) Give an example of using filtering.

a. Give an example for vertical filtering/slicing where you select a subset of the columns. This corresponds to a projection in a SELECT statement.

```
In [57]: # Filtering
```

```
# a) Give an example for vertical filtering/slicing where you select a subset of the
# columns. This corresponds to a projection in a SELECT statement.
vertical_slice_1 = territory_data_1[['Transit stations', 'Workplace']]
vertical_slice_1.head(15)
```

Out[57]:

	Transit stations	Workplace
0	-47%	-41%
1	-32%	-36%
2	-21%	-33%
3	Not enough data for this date	-84%
4	-63%	-37%
5	-33%	-37%
6	-50%	-36%
7	-40%	-44%
8	-37%	-41%
9	-74%	-49%
10	+5%	-35%
11	-65%	-49%
12	-34%	-40%
13	-47%	-39%
14	-37%	-47%

b. Give an example for **horizontal filtering/slicing** where you select a subset of the rows in your DataFramer according to some criteria. This corresponds to WHERE in SELECT statement.

```
In [58]: # b) Give an example for vertical filtering/slicing where you select a subset of the
# rows in your DataFrame according to some criteria. This corresponds to WHERE
# in SELECT statement
numbers = ["-15%", "+30%"]

horizontal_slice_1 = territory_data_1[territory_data_1["Grocery & pharmacy"].isin(numbers)]
horizontal_slice_1.head()
```

```
Out[58]:
```

	Territory	Retail & recreation	Grocery & pharmacy	Parks	Transit stations	Workplace	Residential
11	Grisons	-85%	-15%	-35%	-85%	-49%	+14%
17	Schaffhausen	-83%	+30%	Not enough data for this date	-34%	-39%	+2%

7. (10 points) Show an example where you **merge two DataFrames**.

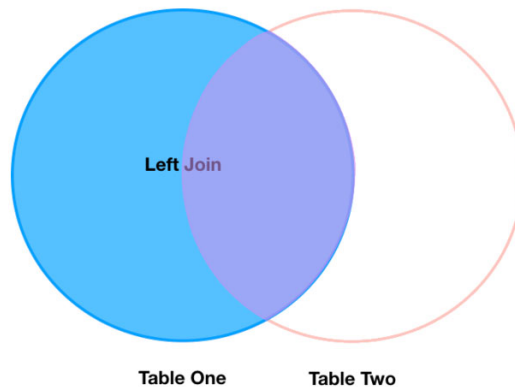
Perform a concat()

```
In [64]: # 7) Merge 2 dataFrames
concat_territory_1 = pd.concat([vertical_slice_1, horizontal_slice_1], ignore_index=True)
concat_territory_1.tail(15)
```

```
Out[64]:
```

	Transit stations	Workplace	Territory	Retail & recreation	Grocery & pharmacy	Parks	Residential
14	-37%	-47%	NaN	NaN	NaN	NaN	NaN
15	-50%	-48%	NaN	NaN	NaN	NaN	NaN
16	-40%	-40%	NaN	NaN	NaN	NaN	NaN
17	-34%	-39%	NaN	NaN	NaN	NaN	NaN
18	-28%	-38%	NaN	NaN	NaN	NaN	NaN
19	-42%	-35%	NaN	NaN	NaN	NaN	NaN
20	-25%	-38%	NaN	NaN	NaN	NaN	NaN
21	-29%	-24%	NaN	NaN	NaN	NaN	NaN
22	-50%	-50%	NaN	NaN	NaN	NaN	NaN
23	-54%	-48%	NaN	NaN	NaN	NaN	NaN
24	-80%	-52%	NaN	NaN	NaN	NaN	NaN
25	-49%	-48%	NaN	NaN	NaN	NaN	NaN
26	-55%	-43%	NaN	NaN	NaN	NaN	NaN
27	-85%	-49%	Grisons	-85%	-15%	-35%	+14%
28	-34%	-39%	Schaffhausen	-83%	+30%	Not enough data for this date	+2%

Perform a join(), left\_join() in particular:



```

In [91]: Texas_df = pd.read_csv(os.path.join("./csv", "US", "2020-04-11_Texas.csv"))
Cali_df = pd.read_csv(os.path.join("./csv", "US", "2020-04-11_California.csv"))

merged_inner = pd.merge(left=Texas_df, right=Cali_df, left_on='Grocery & pharmacy', right_on='Grocery & pharmacy')

In [98]: #Texas_df.head()

In [99]: #Cali_df.tail()

In [97]: merged_left = pd.merge(left=Texas_df, right=Cali_df, left_on='Grocery & pharmacy', right_on='Grocery & pharmacy', how='left')

# The LEFT JOIN produces a complete set of records from DataFrame A (Left DataFrame), with the matching records (where available)
# here, Texas, California, Humboldt County(CA) and Sonoma County (CA) have Grocery & pharmacy value of -14%
# same for Angelina County(TX), Solano County(CA), Stanislaus County(CA), Sutter County(CA) with value of -4%—
merged_left.head(15)

```

Out[97]:

	Territory_x	Retail & recreation_x	Grocery & pharmacy	Parks_x	Transit stations_x	Workplace_x	Residential_x	Territory_y	Retail & recreation_y	Parks_y	Transit stations_y	Workplace_y
0	Texas	-45%	-14%	-54%	-49%	-38%	+16%	California	-48%	-48%	-53%	-40%
1	Texas	-45%	-14%	-54%	-49%	-38%	+16%	Humboldt County	-45%	-18%	-55%	-35%
2	Texas	-45%	-14%	-54%	-49%	-38%	+16%	Sonoma County	-50%	-59%	-49%	-41%
3	Anderson County	-35%	-2%	Not enough data for this date	Not enough data for this date	-24%	+16%	Kern County	-39%	-24%	-37%	-25%
4	Anderson County	-35%	-2%	Not enough data for this date	Not enough data for this date	-24%	+16%	Tulare County	-36%	-47%	-32%	-25%
5	Andrews County	-45%	-40%	Not enough data for this date	-13%	-36%	Not	NaN	NaN	NaN	NaN	NaN
6	Angelina County	-46%	-4%	-20%	-38%	-29%	+13%	Solano County	-43%	-18%	-39%	-35%
7	Angelina County	-46%	-4%	-20%	-38%	-29%	+13%	Stanislaus County	-42%	-15%	-27%	-32%
8	Angelina County	-46%	-4%	-20%	-38%	-29%	+13%	Sutter County	-37%	-19%	-20%	-32%
9	Aransas County	-45%	-31%	-57%	Not enough data for this date	-27%	+10%	NaN	NaN	NaN	NaN	NaN

8. (5 points) Export the merged DataFrame to
- A csv file if your input data is in json documents.
  - A **json file** if your input data is in csv files.

```

In [109]: merged_left.to_json(os.path.join("./json_test", "2020-04-11_Texas_and_Cali.json"))

```

File Edit View Language

JSON

```
1 {"Territory_x":
{"0":"Texas","1":"Texas","2":"Texas","3":"Anderson County","4":"Anderson County","5":"Andrews County","6":"Angelina County","7":"Angelina_C
ounty","8":"Angelina County","9":"Aransas County","10":"Archer County","11":"Atascosa County","12":"Atascosa County","13":"Atascosa County"
,"14":"Austin County","15":"Bailey County","16":"Bandaera County","17":"Bastrop County","18":"Bee County","19":"Bell County","20":"Bell Coun
ty","21":"Bell County","22":"Bexar County","23":"Bexar County","24":"Blanco County","25":"Bosque County","26":"Bowie County","27":"Bowie Co
unt y","28":"Bowie County","29":"Brazoria County","30":"Brazos County","31":"Brewster County","32":"Brooks County","33":"Brooks County","34"
:"Brown County","35":"Burleson County","36":"Burleson County","37":"Burnet County","38":"Burnet County","39":"Burnet County","40":"Caldwell
_County","41":"Calhoun County","42":"Callahan County","43":"Cameron County","44":"Camp County","45":"Camp County","46":"Carson County","47"
:"Cass County","48":"Cass County","49":"Cass County","50":"Castro County","51":"Chambers County","52":"Chambers County","53":"Cherokee Coun
ty","54":"Childress County","55":"Clay County","56":"Coke County","57":"Coleman County","58":"Coleman County","59":"Collin County","60":"Co
lorado County","61":"Colorado County","62":"Comal County","63":"Comanche County","64":"Cooke County","65":"Coryell County","66":"Crane Coun
ty","67":"Crockett County","68":"Crosby County","69":"Culberson County","70":"Dallam County","71":"Dallas County","72":"Dallas County","73"
:"Dallas County","74":"Dawson County","75":"Dawson County","76":"Dewitt County","77":"Deaf Smith County","78":"Deaf Smith County","79":"Dea
f Smith County","80":"Delta County","81":"Denton County","82":"Dimmit County","83":"Donley County","84":"Duval County","85":"Eastland Count
y","86":"Eastland County","87":"Eastland County","88":"Ector County","89":"Ector County","90":"El Paso County","91":"El Paso County","92":"E
l Paso County","93":"Ellis County","94":"Ellis County","95":"Ellis County","96":"Erath County","97":"Falls County","98":"Falls County","99"
:"Fannin County","100":"Fannin County","101":"Fayette County","102":"Floyd County","103":"Fort Bend County","104":"Fort Bend County","105"
:"Franklin County","106":"Franklin County","107":"Freestone County","108":"Freestone County","109":"Frio County","110":"Gaines County","111"
:"Galveston County","112":"Galveston County","113":"Garza County","114":"Gillespie County","115":"Goliad County","116":"Gonzales County","1
17":"Gray County","118":"Grayson County","119":"Grayson County","120":"Gregg County","121":"Gregg County","122":"Grimes County","123":"Gua
dalupe County","124":"Guadalupe County","125":"Guadalupe County","126":"Hale County","127":"Hall County","128":"Hamilton County","129":"Har
deman County","130":"Hardin County","131":"Harris County","132":"Harris County","133":"Harrison County","134":"Hartley County","135":"Haske
ll County","136":"Hays County","137":"Henderson County","138":"Hidalgo County","139":"Hill County","140":"Hockley County","141":"Hood Count
y","142":"Hopkins County","143":"Houston County","144":"Howard County","145":"Hunt County","146":"Hunt County","147":"Hunt County","148":"H
utchinson County","149":"Jack County","150":"Jackson County","151":"Jasper County","152":"Jefferson County","153":"Jim Hogg County","154":"J
im Wells County","155":"Jim Wells County","156":"Johnson County","157":"Johnson County","158":"Johnson County","159":"Jones County","160"
:"Jones County","161":"Karnes County","162":"Kaufman County","163":"Kaufman County","164":"Kaufman County","165":"Kendall County","166":"Ken
dall County","167":"Kerr County","168":"Kerr County","169":"Kerr County","170":"Kimble County","171":"Kleberg County","172":"La Salle Count
y","173":"La Salle County","174":"Lamar County","175":"Lamb County","176":"Lamb County","177":"Lampasas County","178":"Lavaca County","179"
:"Lee County","180":"Leon County","181":"Liberty County","182":"Limestone County","183":"Live Oak County","184":"Llano County","185":"Lovin
g County","186":"Lubbock County","187":"Lubbock County","188":"Lubbock County","189":"Lynn County","190":"Madison County","191":"Marion Cou
nty","192":"Marion County","193":"Marion County","194":"Martin County","195":"Martin County","196":"Martin County","197":"Matagorda County"
,"198":"Matagorda County","199":"Haverick County","200":"McCulloch County","201":"McLennan County","202":"McLennan County","203":"McLennan
County","204":"McMullen County","205":"Medina County","206":"Midland County","207":"Milam County","208":"Mitchell County","209":"Montague_C
ounty","210":"Montgomery County","211":"Moore County","212":"Morris County","213":"Morris County","214":"Nacogdoches County","215":"Nacogdo
ches County","216":"Nacogdoches County","217":"Navarro County","218":"Newton County","219":"Newton County","220":"Nolan County","221":"Nola
n County","222":"Nolan County","223":"Nueces County","224":"Nueces County","225":"Ochiltree County","226":"Ochiltree County","227":"Ochiltr
ee County","228":"Oldham County","229":"Orange County","230":"Palo Pinto County","231":"Palo Pinto County","232":"Palo Pinto County","233"
:"Panola County","234":"Panola County","235":"Parker County","236":"Parmer County","237":"Parmer County","238":"Parmer County","239":"Pecos_
County","240":"Polk County","241":"Potter County","242":"Potter County","243":"Potter County","244":"Presidio County","245":"Rains County"
,"246":"Randall County","247":"Randall County","248":"Randall County","249":"Reagan County","250":"Real County","251":"Red River County","25
2":"Reeves County","253":"Reeves County","254":"Refugio County","255":"Robertson County","256":"Rockwall County","257":"Rockwall County","25
8":"Runnels County","259":"Rusk County","260":"Sabine County","261":"Sabine County","262":"San Augustine County","263":"San Augustine Coun
ty","264":"San Augustine County","265":"San Jacinto County","266":"San Jacinto County","267":"San Patricio County","268":"San Patricio Coun
```