Homework 4

In this homework, you are required to use the data from previous homework to do some visualization with d3.js.

Part 1

In this part you are going to answer some basic problems of D3 first. Then you should modified a sample code to get a simple bar chart.

Problem 1 (35 pt)

1. Answer these questions in simple words. (These ideas will help you to finish the following problems). (4*5pt)

   1.1 What's the difference between SVG Coordinate Space and Mathematical / Graph Coordinate Space?

   1.2. What is `enter()` and `exit()` in d3.js?

   1.3 What is `transform` and `translate` in SVG?

   1.4 Try to understand the idea of anonymous function and its use in d3.js. If there is a list `a = [a,c,b,d,e]`, what is the return value of this anonymous function: `a.map(function(d,i) {return i+5})` (It should be a list)

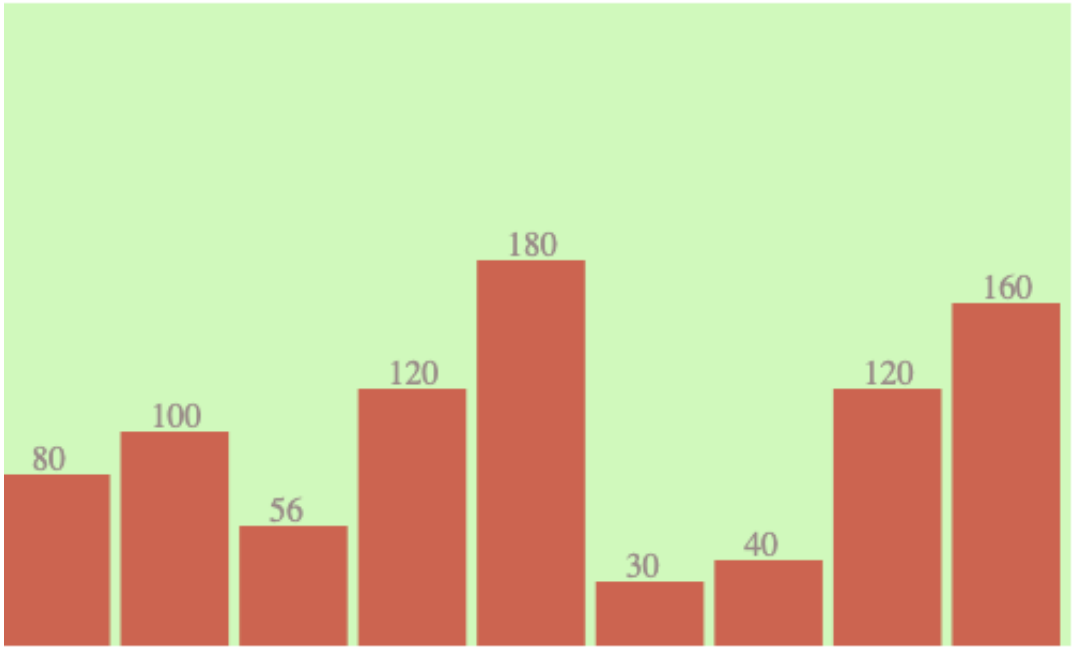
2. Modify the sample code to get the same figure as below: (15pt)
   - You **must** have the same width and paddings of 5px as this given bar-chart.

     Hint: As the svg_width and bar_padding is given, you can divide the width equally to get the width of bar.
   - The label **must** locate 2px above the middle of each bar.
   - You **must** use `transform` to display the bars instead of the attr(x),(y) in sample code.

     You can use attr(x),(y) to display the text, but it is recommend to use transform to do this too.
   - You should write the javascript in a single file (`.js`), separated with the structure file (`.html`).(You can get this idea from problem 2,3)

- Hint: add another elements "text" to display the labels.
- (you can use any color you like).

Submission requirement: Your answer should include both screenshots of your codes and the output bar-chart.



Part 2

After finishing problem 1, you will get some basic ideas of d3.js. Now have a look at the *big_data_tutorial_hw4_part2* tutorial to get a brief idea of Django. Then you are required to query the data you got from hw3 and use django to build a simple web application for visualization.

You can either do this on you computer or create a VM (compute engine) on GCP or even use App Engine to do this. However, the tutorial is for localhost and there is no additional tutorials for finishing this homework on Google cloud, you should search for how to do this on cloud by yourself if you want.

Problem 2 (30 pt)

In this problem you are required to process the data you got from HW3, create a django project and finish the missing code to draw a dashboard.

- Step 1: Data pocessing: (5 pt)

  Process the *wordcount* table to required format below(You use any method to do this, such as create a new table):

Schema Details Preview

Row	time	ai	data	good	movie	spark
1	2019-10-23 02:33:20 UTC	10	0	17	205	1
2	2019-10-23 02:38:20 UTC	10	0	4	213	6
3	2019-10-23 02:36:20 UTC	13	1	13	192	9
4	2019-10-23 02:32:20 UTC	15	1	10	233	4
5	2019-10-23 02:37:20 UTC	6	1	10	202	5
6	2019-10-23 02:29:20 UTC	14	2	9	210	4
7	2019-10-23 02:34:20 UTC	10	2	11	207	4

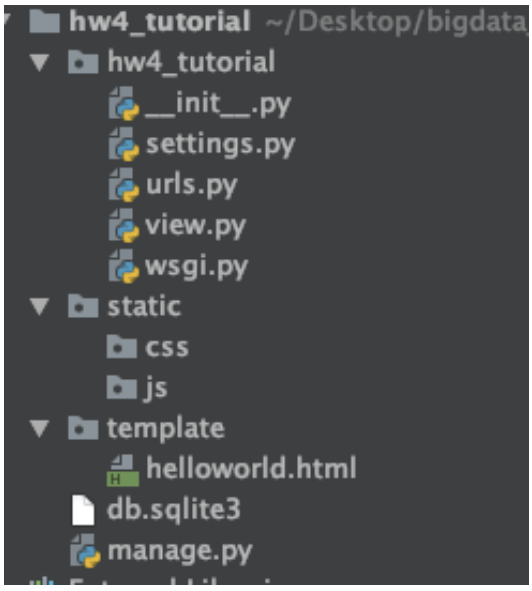
It must contain 6 columns: time and the 5 words; more than 8 rows. You should combine the data which have the same time, and fill the missing value with 0.

You can use any way to process the data, no matter Pandas or SQL.

Submission requirement: screenshot of your code of data processing and the preview of the table in BigQuery like above.

- Step 2: Create Django project. (5pt)

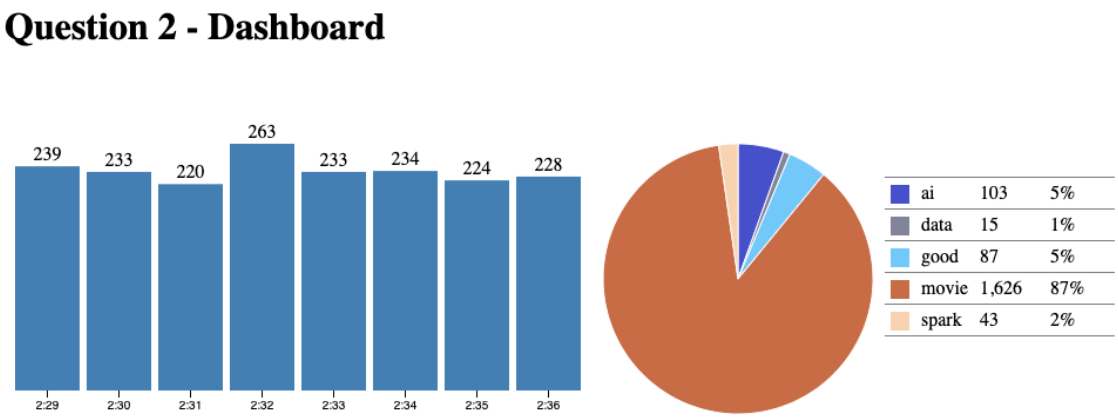
  Follow the tutorial *big_data_tutorial_hw4_part2* in `files-tutorials` of canvas and create a Django project. If you create a project named `hw4_tutorial`, the Directory Structure should looks like below:



Submission requirement: screenshot of Directory Structure of your project and a screenshot to show the *helloworld* page.

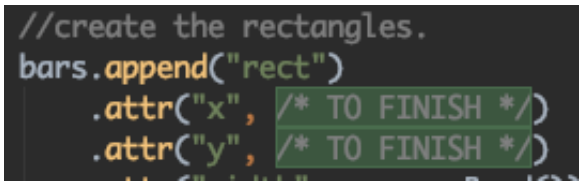
- Step 3: Finish the code (20pt)

Replace/ Copy and pase the content in `view.py` / `urls.py` to the same files in your project. Put the 3 files: `dashboard.html`, `dashboard.css` and `dashboard.js` to the corresponding documents. Modified the code to draw a dashboard like below:



- In `view.py`, you need to finish a SQL to query the data, the data is limited to 8 rows. (1pt) Then you should modified the data to the required format. (5pt)
- There are 8 blanks in `dashboard.js` to finish. (8*1pt)
- Result (6pt)

Submission requirement: screenshot of your code (Please only capture the part of code that you are required to finish, like below) and the output result.



Problem 3 (35 pt)

In this problem you are required to process the data from HW2 and upload it to BigQuery, then also finish the missing code to draw a dashboard.

- Step 1: Data pocessing: (10 pt)

  In HW2 Question 2.2, you are required to provide a list of top 10 clusters . The result is like below:

```
+------------+-----+
|   component|count|
+------------+-----+
|           0|48860|
|         161|   66|
| 42949673000|   31|
|103079215141|   25|
| 34359738423|   19|
| 17179869446|   16|
|         231|   13|
|146028888124|    6|
| 51539607798|    5|
|618475290697|    4|
|670014898346|    3|
```

- Here, you are required to collect the nodes in component **103079215141**, which have 25 nodes and save it as a table in Bigquery. (You can save it as `.csv` first and create a table based on the data) (4 pt)
- Then, you should get the *edges* (source, target) whose source is the nodes you get above. Then re-label them from 0 to 24 (mapping the 25 nodes to 0~24) and save it into another table in Bigquery. (6 pt)

  Submission requirement: screenshot of your code and the 2 table in BigQuery.

nodes

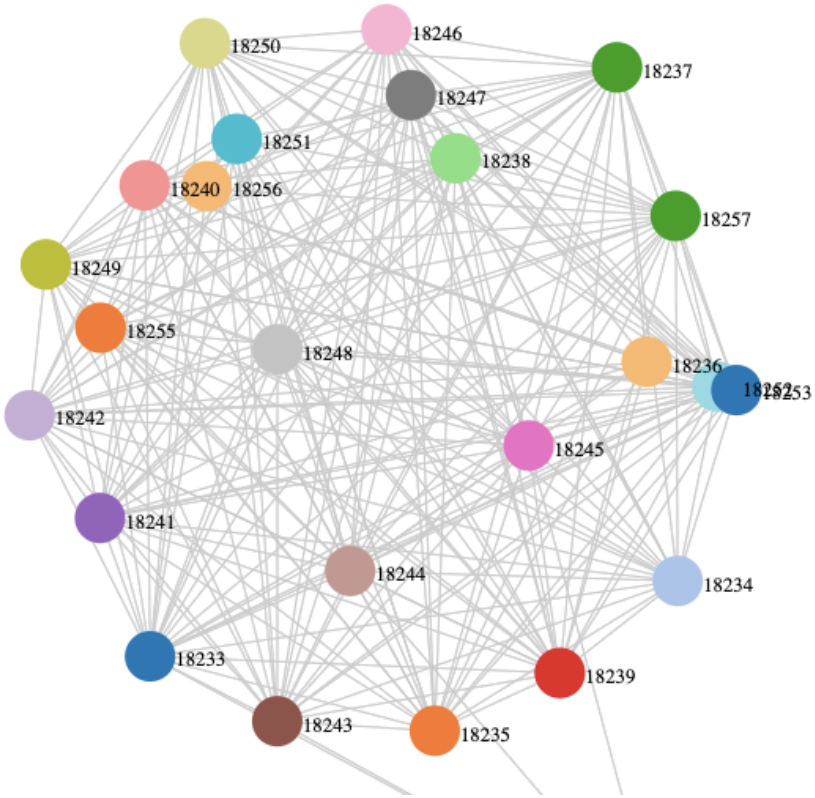
Schema Details Preview

Row	node
1	18233
2	18234
3	18235
4	18236

Schema Details Preview

Row	source	target
1	0	1
2	0	1
3	0	2
4	0	2
5	0	3

- Step 2: Finish the code (25 pt)
  - Modified the code to draw a figure like below:



- In `view.py`, you need to finish 2 SQL to query the 2 table to get data. (2*1 pt) Then modified them to the required format. Notice that there may be some deplicates of edges' data and you are required to remove them. The number of unique edges are 452. (5 pt)

  There is a typo of the format, please have a look at the correction in Piazza

- Finish the blanks in `connection.js` . (12*1 pt)
- Result (6 pt)

  Submission requirement: screenshot of your code and the output result.