Victoria Road Analytics Project Report

A. Contribution

Ran Lu s3583185 100%

B. Links to My Project

Crash Stats

https://www.data.vic.gov.au/data/dataset/crash-stats-data-extract

Traffic Volumes

http://vicroadsopendata.vicroadsmaps.opendata.arcgis.com/datasets/147696bb47544a209e0a5e79e165d1b0_0

Amazon S3 for EMR results with viewing access

https://console.aws.amazon.com/s3/home?region=us-west-2#&bucket=s3583185-assignment2&prefix=

Google API Server

https://apis-explorer.appspot.com/apis-explorer/?base=https://s3583185-myapi.appspot.com/_ah/api#p/

Google API Client

http://s3583185-my-client.appspot.com

Google Sheets for Accidents Records

https://docs.google.com/spreadsheets/d/1jlwh4vckh2Tuz_E3z96nXhXOa_OaGtegwJnoY5uf_8I/edit?usp=sharing

User Webpage for BigQuery Results

https://datastudio.google.com/open/0ByLdmFPGXlUFbmJDdFhiR0pjT0E

User Webpage for BigQuery Results Using Google API

http://s3583185-bigquery.appspot.com

C. Project Summary

The aim of this project is to collect and analyse data from multiple websites including Melbourne public transport and VicRoads to build a user-friendly website. After the Google BigQuery process, it will show users 6 features such as accidents types and average vehicles count in a month in a road, more details about features will be covered in the following introduction section. All datasets are collected in the websites, processed in Amazon EMR and imported into Google Cloud Storage. GCS also contains accidents records from both server and user API. The results are in Google BigQuery and finally illustrate a webpage with all information analysis. If further development is made, it can be used in traffic industry.

D. Introduction

i. What are the motivations behind your idea?

This project can be used to analyse traffic volumes in each street/road/freeway, therefore the average waiting time of traffic lights can be changed, pedestrians doesn’t have to wait for a long time if there’s no traffic congestion. The analysis can also reveal age, time and areas of most accidents for the police to handle them more efficiently.

ii. What it does?

The Victoria Roads Analytics can use complex traffic volumes and accidents datasets to output 6 features: accidents types, average vehicles count in a month in a road, average car flows in freeways in a month, dangerous areas in Victoria, high-frequency accidents hours in a day and age groups causing most accidents. One additional feature is to select different types like roads, freeways or streets. There is also an accident recording system for both server and client to insert records easily by using Google APIs.

iii. Why it is required?

With the fast development of cities, more individuals decide to buy cars and use them for daily lives, the personal safety of each person is vital, which means traffic analysis system is necessary for informing high-frequency accidents areas, avoiding traffic jam and spend less time on the road to reduce pollution. As the expansion of cities, the government also need to manage traffic conditions and reduce accidents, therefore monitoring them and make efforts accordingly is important.

iv. How it can be used as real-life application?

The server and client APIs can be used by anyone to manually import accidents records. The webpage can be viewed on everywhere with Internet access, as well as used for presentation.

v. The advantages/positive/new things of your application.

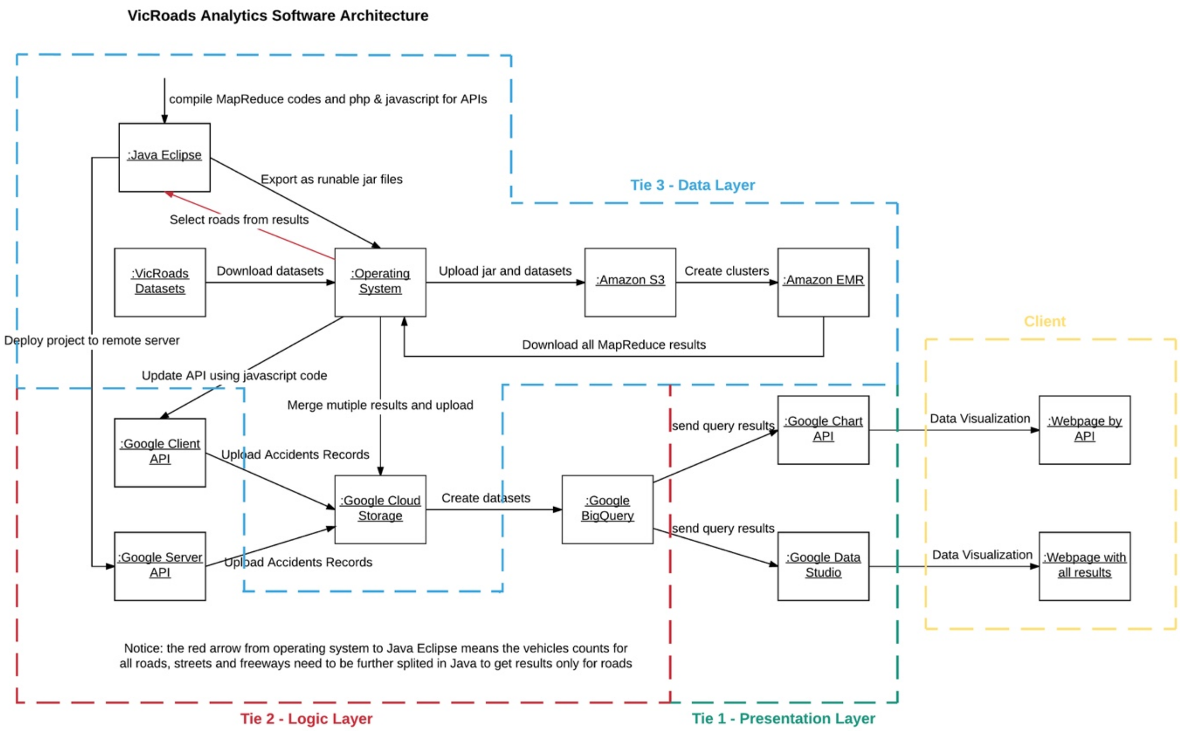
One of the most important advantages of my application is the outputs of all data are being visualized instead of showing database search results which can only being view by professional people, the webpage/client API is also easy to use for clients.

E. Related Work

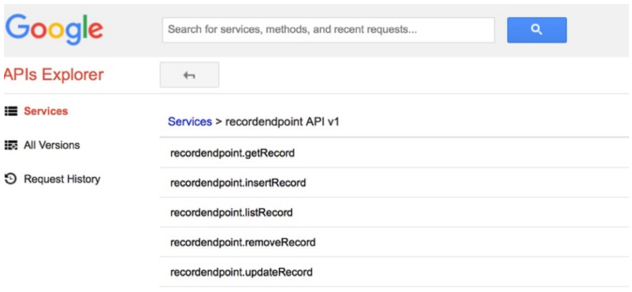
Since most results of my work are processed by Amazon EMR and the outputs are visualized, there's some charts online to show data analysis of traffic, I have found the API on VicRoads with the following URL

https://services2.arcgis.com/18ajPSI0b3ppsmMt/arcgis/rest/services/Traffic_Volume/FeatureServer/0/query?outFields=*&where=1%3D1
It's more like server side API which takes queries and output results, although a little complex for people without the knowledge of Databases.

F. Software Design/Architecture

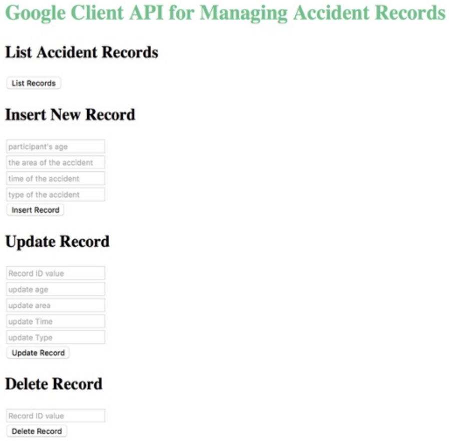

VicRoads Analytics Software Architecture
compile MapReduce codes and php & javascript for APIs
:Java Eclipse
Export as runable jar files
Select roads from results
Tie 3 - Data Layer
:VicRoads Datasets
Download datasets
:Operating System
Upload jar and datasets
:Amazon S3
Create clusters
:Amazon EMR
Client
Deploy project to remote server
Download all MapReduce results
Update API using javascript code
Merge mutiple results and upload
:Google Client API
Upload Accidents Records
:Google Cloud Storage
Create datasets
:Google BigQuery
send query results
:Google Chart API
Data Visualization
:Webpage by API
:Google Server API
Upload Accidents Records
send query results
:Google Data Studio
Data Visualization
:Webpage with all results
Notice: the red arrow from operating system to Java Eclipse means the vehicles counts for all roads, streets and freeways need to be further splited in Java to get results only for roads
Tie 2 - Logic Layer
Tie 1 - Presentation Layer

The raw datasets used are traffic volumes from VicRoads, Accidents from data.vic. Since these data contain multiple attributes, they need to be reduced by clusters in Amazon Elastic MapReduce to get precisely one attribute each time, and adds them up. Amazon S3 is the only storage for raw datasets and MapReduce results. Both endpoint API for server and client is used to insert accidents records.



Services > recordendpoint API v1

recordendpoint.getRecord
recordendpoint.insertRecord
recordendpoint.listRecord
recordendpoint.removeRecord
recordendpoint.updateRecord


Google Client API for Managing Accident Records

List Accident Records

Insert New Record

Update Record

Delete Record

Google BigQuery is another dataset for this project, it contains all information showed on final webpages. Google Chart API and Google Data Studio are used for retrieving data by queries from BigQuery, for Google Data Studio, it collects data by queries.

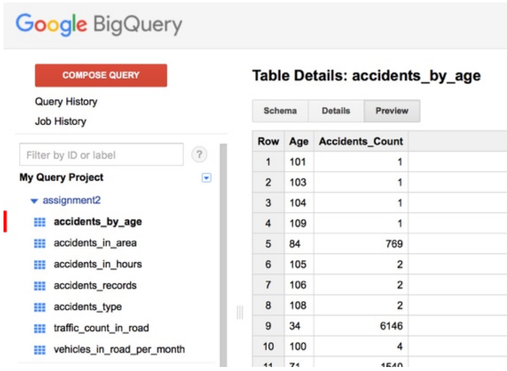
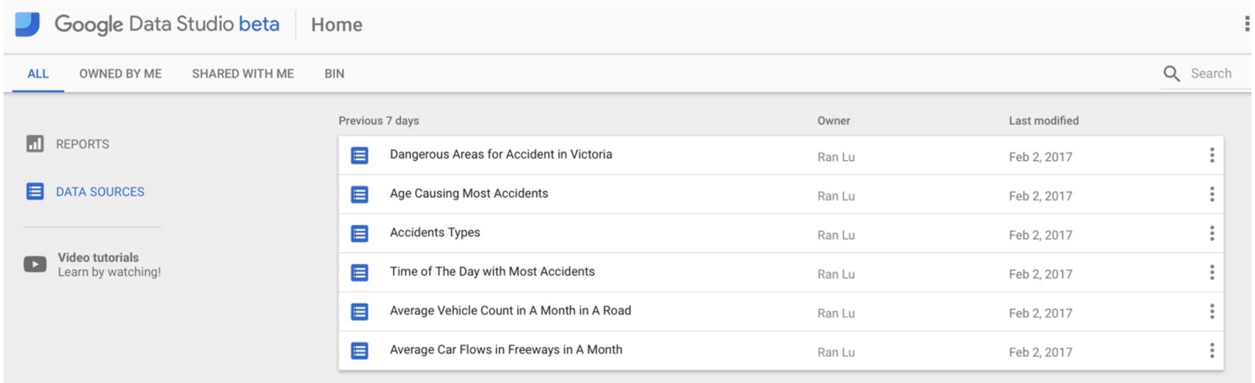
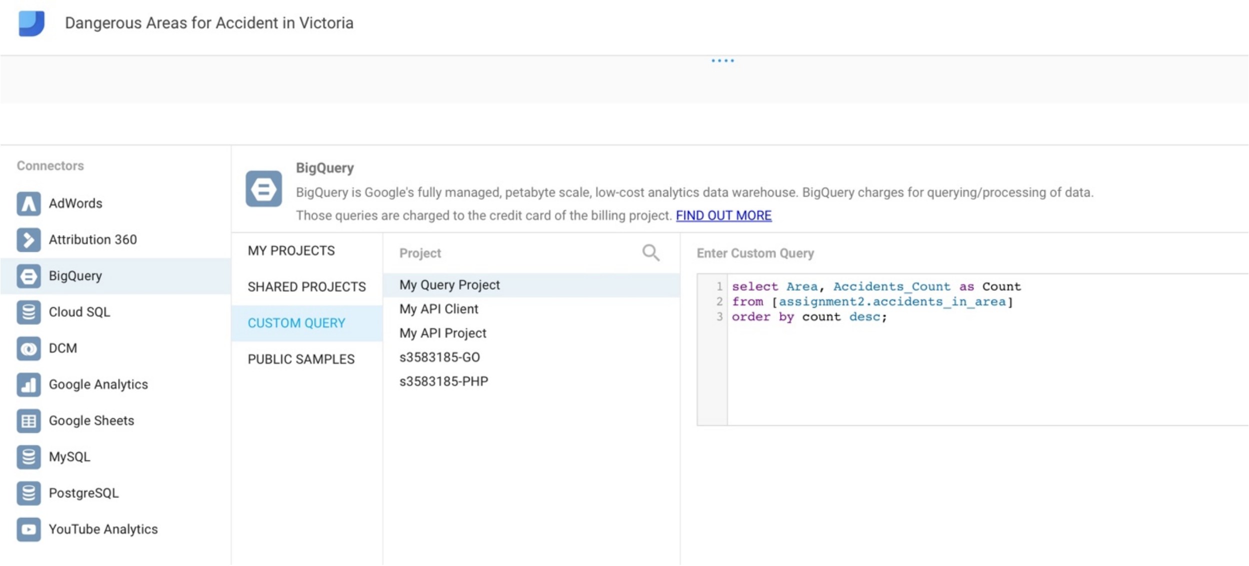


Table Details: accidents_by_age

Row	Age	Accidents_Count
1	101	1
2	103	1
3	104	1
4	109	1
5	84	769
6	105	2
7	106	2
8	108	2
9	34	6146
10	100	4
11	71	1540



Previous 7 days	Owner	Last modified
Dangerous Areas for Accident in Victoria	Ran Lu	Feb 2, 2017
Age Causing Most Accidents	Ran Lu	Feb 2, 2017
Accidents Types	Ran Lu	Feb 2, 2017
Time of The Day with Most Accidents	Ran Lu	Feb 2, 2017
Average Vehicle Count in A Month in A Road	Ran Lu	Feb 2, 2017
Average Car Flows in Freeways in A Month	Ran Lu	Feb 2, 2017



Dangerous Areas for Accident in Victoria

BigQuery

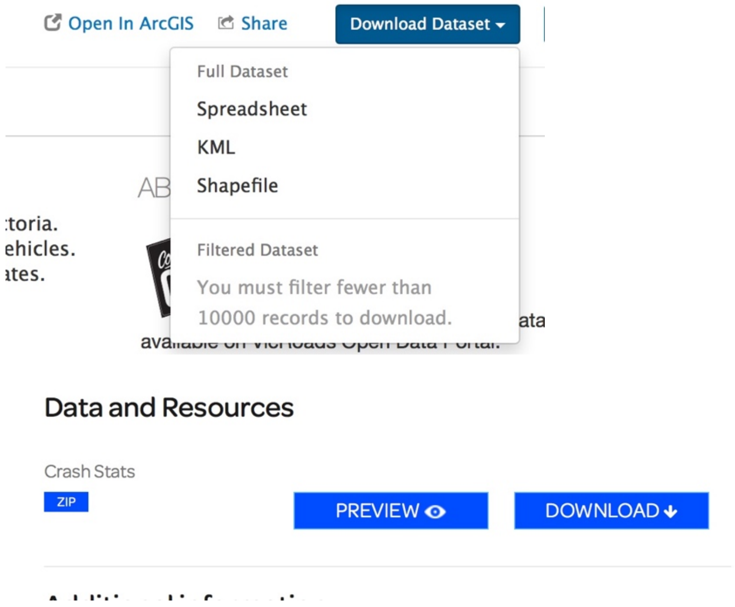
BigQuery is Google's fully managed, petabyte scale, low-cost analytics data warehouse. BigQuery charges for querying/processing of data. Those queries are charged to the credit card of the billing project. FIND OUT MORE

```
select Area, Accidents_Count as Count
from [assignment2.accidents_in_area]
order by count desc;
```

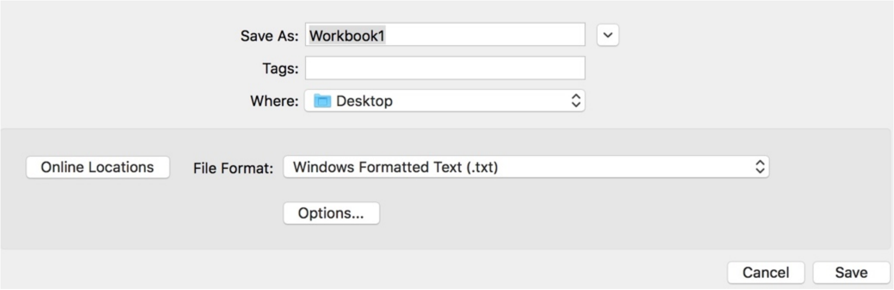
G. Implementation

For convenience, all data, jar and code can be found in related folders.

Download all datasets from VicRoads and data.vic. For the VicRoads dataset, download as spreadsheet.



The csv files need to be changed into txt, so open them with applications like Microsoft Excel, and save as txt.

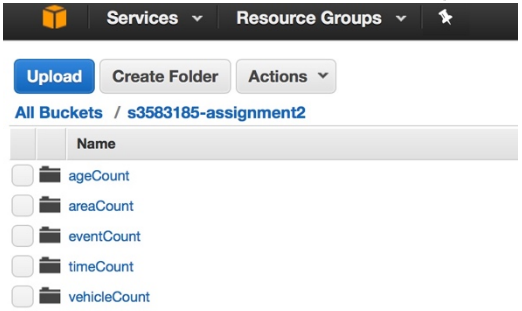

Save As: Workbook1

Tags:

Where: Desktop

File Format: Windows Formatted Text (.txt)

In Amazon S3, create a bucket named sxxxxxx-assignment2, all following id's will use s3583185 instead. And create 5 folders using the names as following:

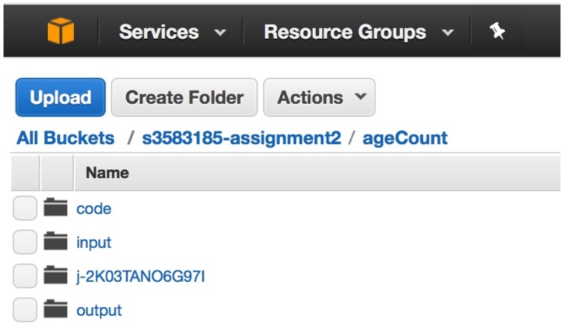


All Buckets / s3583185-assignment2

Name

- ageCount
- areaCount
- eventCount
- timeCount
- vehicleCount

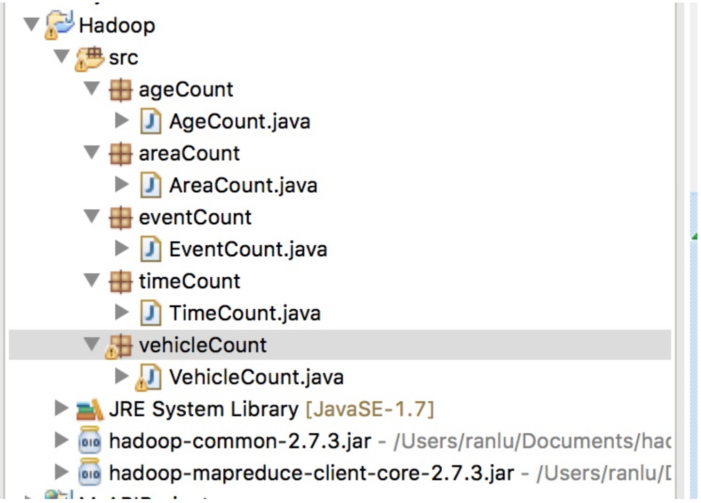
In each folder, create two folders named code and input.



All Buckets / s3583185-assignment2 / ageCount

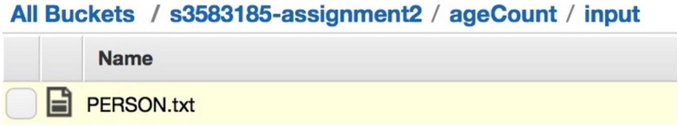
Name
code
input
j-2K03TANO6G97I
output

Following the labs process, create a Hadoop project in Java Eclipse, and use the codes provided to export 5 jar files such as time.jar, vehicle.jar etc.



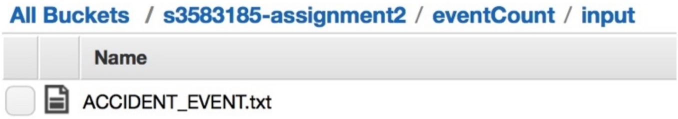
```
▼ Hadoop
  ▼ src
    ▼ ageCount
      ▶ AgeCount.java
    ▼ areaCount
      ▶ AreaCount.java
    ▼ eventCount
      ▶ EventCount.java
    ▼ timeCount
      ▶ TimeCount.java
    ▼ vehicleCount
      ▶ VehicleCount.java
  ▶ JRE System Library [JavaSE-1.7]
  ▶ hadoop-common-2.7.3.jar - /Users/ranlu/Documents/hac
  ▶ hadoop-mapreduce-client-core-2.7.3.jar - /Users/ranlu/[
```

Upload jar files to code folder in Amazon S3 bucket separately, and upload these files into input folder, which are those downloaded from websites before.



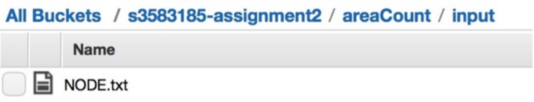
All Buckets / s3583185-assignment2 / ageCount / input

Name
PERSON.txt



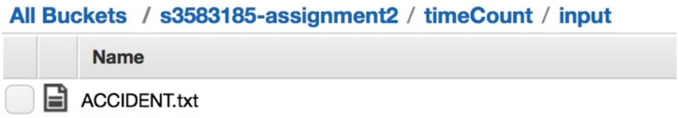
All Buckets / s3583185-assignment2 / eventCount / input

Name
ACCIDENT_EVENT.txt



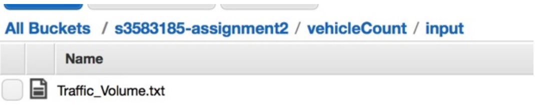
All Buckets / s3583185-assignment2 / areaCount / input

Name
NODE.txt



All Buckets / s3583185-assignment2 / timeCount / input

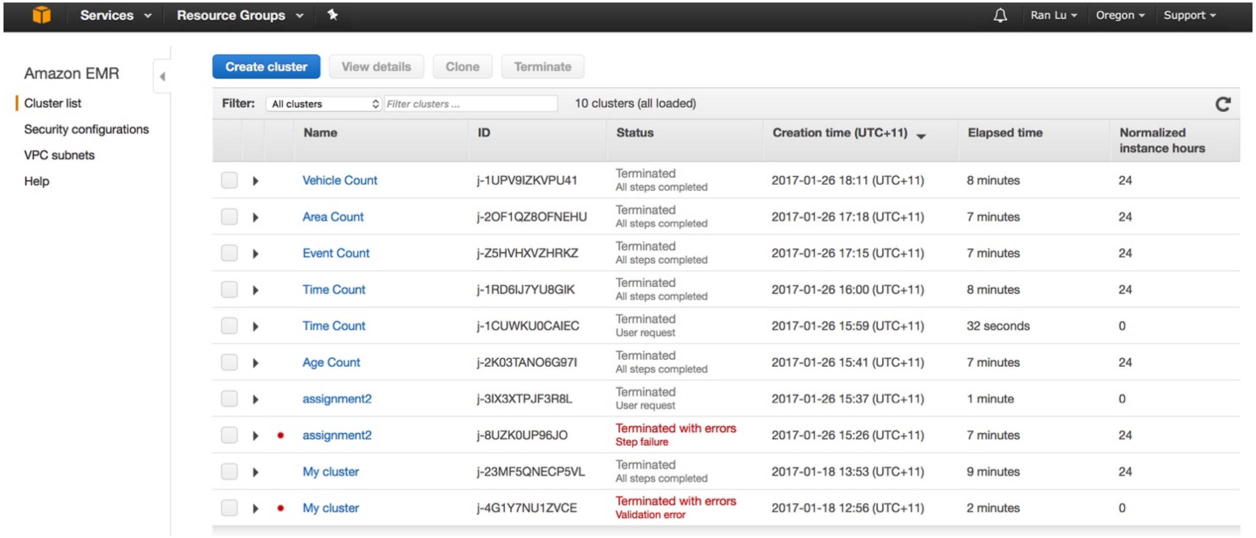
Name
ACCIDENT.txt



All Buckets / s3583185-assignment2 / vehicleCount / input

Name
Traffic_Volume.txt

Create clusters following the steps in lab to make MapReduce process.



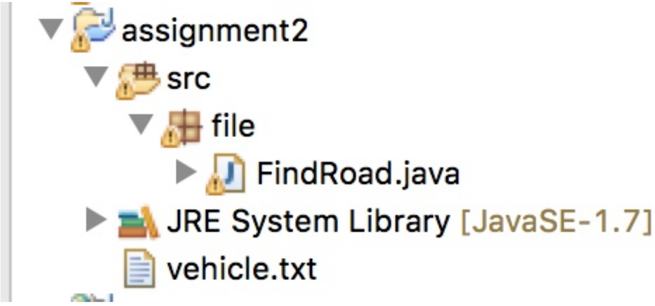
Filter: All clusters | Filter clusters ... | 10 clusters (all loaded)

Name	ID	Status	Creation time (UTC+11)	Elapsed time	Normalized instance hours
Vehicle Count	j-1UPV9IZKVPU41	Terminated All steps completed	2017-01-26 18:11 (UTC+11)	8 minutes	24
Area Count	j-2OF1QZ8OFNEHU	Terminated All steps completed	2017-01-26 17:18 (UTC+11)	7 minutes	24
Event Count	j-Z5HVHXVZHRKZ	Terminated All steps completed	2017-01-26 17:15 (UTC+11)	7 minutes	24
Time Count	j-1RD6IJ7YU8GIK	Terminated All steps completed	2017-01-26 16:00 (UTC+11)	8 minutes	24
Time Count	j-1CUWKU0CAIEC	Terminated User request	2017-01-26 15:59 (UTC+11)	32 seconds	0
Age Count	j-2K03TANO6G97I	Terminated All steps completed	2017-01-26 15:41 (UTC+11)	7 minutes	24
assignment2	j-3IX3XTPJF3R8L	Terminated User request	2017-01-26 15:37 (UTC+11)	1 minute	0
● assignment2	j-8UZK0UP96JO	Terminated with errors Step failure	2017-01-26 15:26 (UTC+11)	7 minutes	24
My cluster	j-23MF5QNECP5VL	Terminated All steps completed	2017-01-18 13:53 (UTC+11)	9 minutes	24
● My cluster	j-4G1Y7NU1ZVCE	Terminated with errors Validation error	2017-01-18 12:56 (UTC+11)	2 minutes	0

Download all results from Amazon S3 output folders, save to Hadoop folder in local machine, notice the results need to be saved in different folders like time, vehicles etc. for merge, and using the following command in command line to merge them:

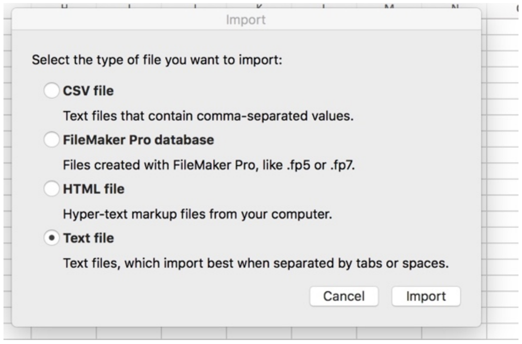
```
hadoop fs –getmerge <downloadlocations>/ /output/output.txt
```

This step will merge files like part-r-00000 and create 5 txt files, rename them by age, area, event, time and vehicle. Create a new project in Eclipse, copy vehicle.txt and FindRoad.java into this project, and run as java application, it will create a file named road.txt in java workspace, which only contains vehicle count for each road.



```
▼ assignment2
    ▼ src
        ▼ file
            ▶ FindRoad.java
    ▶ JRE System Library [JavaSE-1.7]
      vehicle.txt
```

Open Microsoft Excel, select File -> Import, and choose the last one, then select the merged txt files.



Import

Select the type of file you want to import:

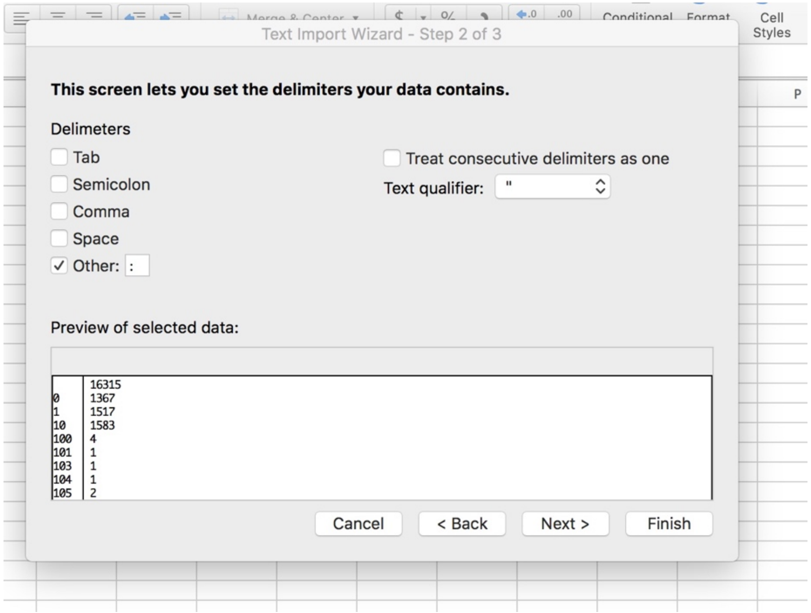
○ **CSV file**
Text files that contain comma-separated values.

○ **FileMaker Pro database**
Files created with FileMaker Pro, like .fp5 or .fp7.

○ **HTML file**
Hyper-text markup files from your computer.

● **Text file**
Text files, which import best when separated by tabs or spaces.

Keep everything as default in the next screen, click next, and modify settings as the following:



Text Import Wizard - Step 2 of 3

This screen lets you set the delimiters your data contains.

Delimeters

☐ Tab
☐ Semicolon
☐ Comma
☐ Space
☑ Other: :

☐ Treat consecutive delimiters as one

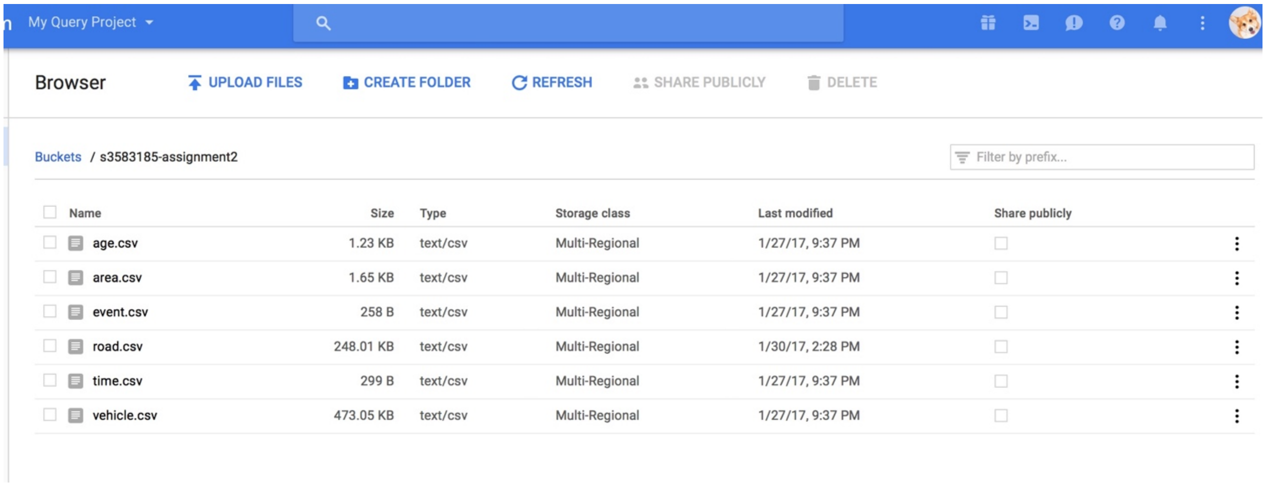
Text qualifier: "

Preview of selected data:

```
     16315
0    1367
1    1517
10   1583
100  4
101  1
103  1
104  1
105  2
```

This step will import all txt files into Excel, save the 5 merged files and road.txt as CSV files.

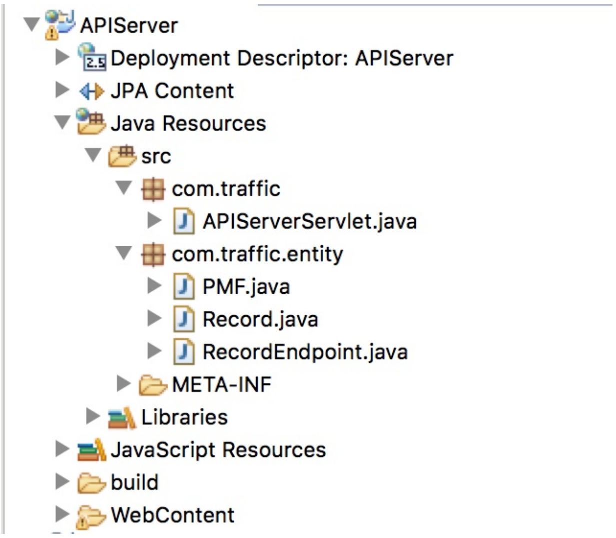
Create a project named My Query Project in Google Cloud Platform, and create a bucket in GCS named s3583185-bigquery, upload csv files to it. Create another two projects named s3583185-myapi and s3583185-my-client.



Buckets / s3583185-assignment2

Name	Size	Type	Storage class	Last modified	Share publicly
age.csv	1.23 KB	text/csv	Multi-Regional	1/27/17, 9:37 PM	☐
area.csv	1.65 KB	text/csv	Multi-Regional	1/27/17, 9:37 PM	☐
event.csv	258 B	text/csv	Multi-Regional	1/27/17, 9:37 PM	☐
road.csv	248.01 KB	text/csv	Multi-Regional	1/30/17, 2:28 PM	☐
time.csv	299 B	text/csv	Multi-Regional	1/27/17, 9:37 PM	☐
vehicle.csv	473.05 KB	text/csv	Multi-Regional	1/27/17, 9:37 PM	☐

Create a APIServer Dynamic web project in Java Eclipse, choose project id as s3583185-myapi, and a package named com.traffic.entity, copy Record.java into this package.



```
▼ APIServer
    ▶ Deployment Descriptor: APIServer
    ▶ JPA Content
    ▼ Java Resources
        ▼ src
            ▼ com.traffic
                ▶ APIServerServlet.java
            ▼ com.traffic.entity
                ▶ PMF.java
                ▶ Record.java
                ▶ RecordEndpoint.java
            ▶ META-INF
        ▶ Libraries
    ▶ JavaScript Resources
    ▶ build
    ▶ WebContent
```

Select Record.java, right click and select Google App Engine WTP -> Generate endpoint class, and follow the steps in labs to finally get PMF.java and RecordEndpoint.java.
Change java compiler to version 1.7, right click on the project name then select Google App Engine WTP -> Deploy project to Remote Server.

Create a folder named api_client, and copy these two files:

app.yaml   index.html

Use the Command Line Tool, run the following command:

```
appcfg.py update –A s3583185-my-client api_client/
```

Now both API server and API client are running, you can insert accidents records, all datasets will be stored in Google Datastore in My API Project.
https://apis-explorer.appspot.com/apis-explorer/?base=https://s3583185-myapi.appspot.com/_ah/api#p/
http://s3583185-my-client.appspot.com

Create the following datasets in BigQuery in My Query Project, select the original files from storage in the same project, the schema of each table are
Age, Accidents_Count
Area, Accidents_Count
Hours_In_24, Accidents_Count
Type, Count
Road, Accidents_Count
Road, Vehicles_Per_Month

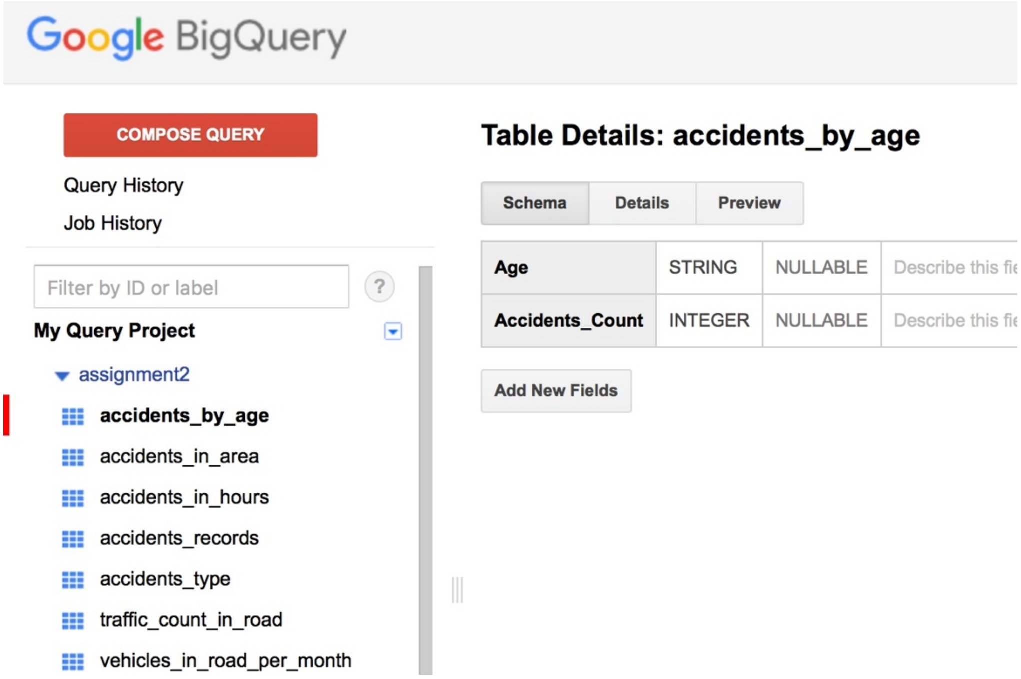


Table Details: accidents_by_age

Age	STRING	NULLABLE	Describe this fie
Accidents_Count	INTEGER	NULLABLE	Describe this fie

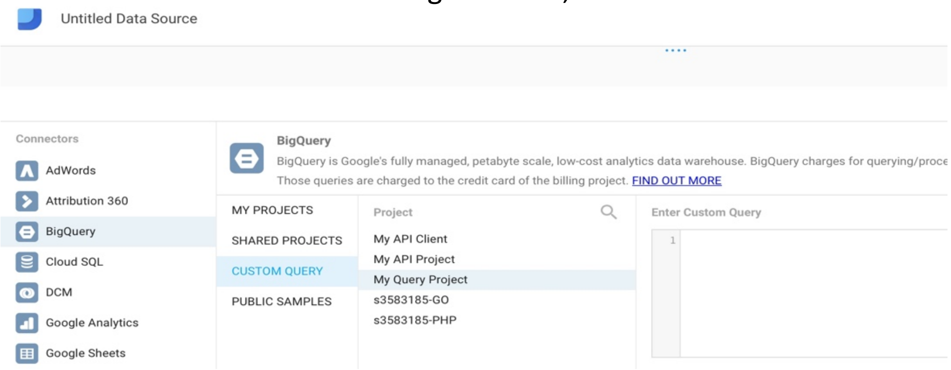
Notice accidents_records table need to be transferred from Google Datastore backup in My API Project, here is the link about this: https://cloud.google.com/bigquery/loading-data-cloud-datastore

Now copy the folder Visualization to your cloud workspace, run the following command:
appcfg.py update –A s3583185-bigquery Visualization/

The results of all data analysis will be shown here first:
http://s3583185-bigquery.appspot.com

Open Google Data Studio in this link:
https://datastudio.google.com/#/org//navigation/datasources

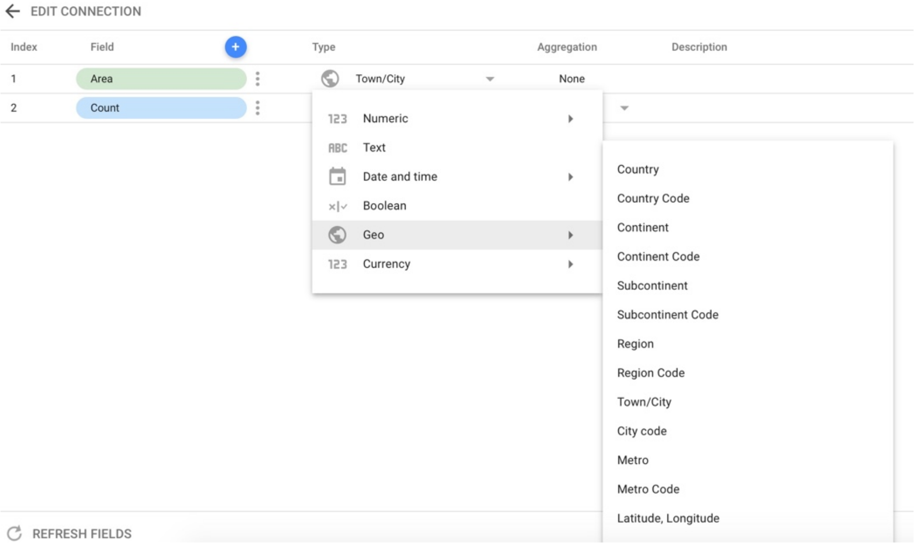
Click on add button on bottom-right corner, select data sources as following:



BigQuery

BigQuery is Google's fully managed, petabyte scale, low-cost analytics data warehouse. BigQuery charges for querying/proce
Those queries are charged to the credit card of the billing project. FIND OUT MORE

Use query provided, and click connect, keep everything as default, name sources accordingly, click buttons on top-left to return. Notice for areas, the attribute need to be town/city.



EDIT CONNECTION

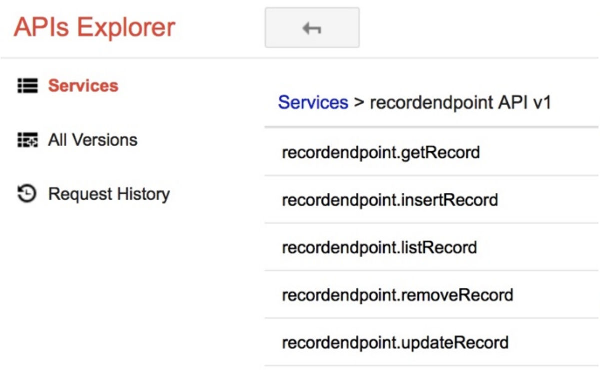
Index	Field	Type	Aggregation	Description
1	Area	Town/City	None	
2	Count			

```
select Area, Accidents_Count as Count
from [assignment2.accidents_in_area]
order by count desc;

select Age, Accidents_Count as Count
from [assignment2.accidents_by_age]
order by Count DESC;

select Type, Accidents_Count as Count
from [assignment2.accidents_type]
order by count desc;

select Hours_In_24 as Hour, Accidents_Count as Count
from [assignment2.accidents_in_hours]
order by count desc;

select Road, Count
from [assignment2.traffic_count_in_road]
order by count desc;

select Road, Vehicles_Per_Month as Count
from [assignment2.vehicles_in_road_per_month]
order by count desc;
```

Now create a new report in data studio, you can insert any types of charts, and select data sources on the right, the final report can be viewed here: https://datastudio.google.com/u/0/#/org//reporting/0ByLdmFPGXlUFbmJDdFhiR0pjT0E/page/kK3B

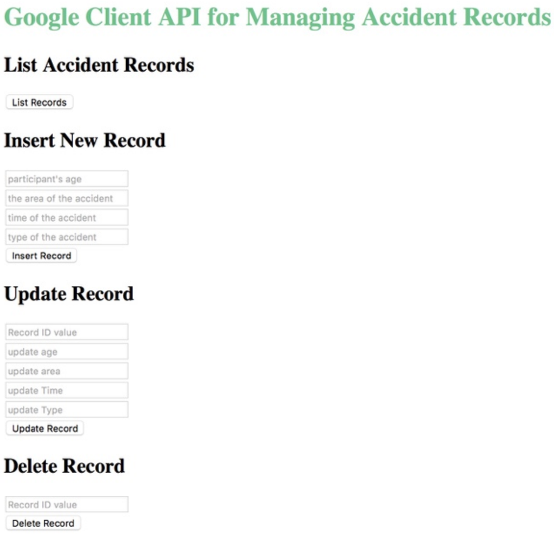
H. User Manual

Both server and client APIs can be used for accidents records.



Services > recordendpoint API v1

- recordendpoint.getRecord
- recordendpoint.insertRecord
- recordendpoint.listRecord
- recordendpoint.removeRecord
- recordendpoint.updateRecord



Google Client API for Managing Accident Records

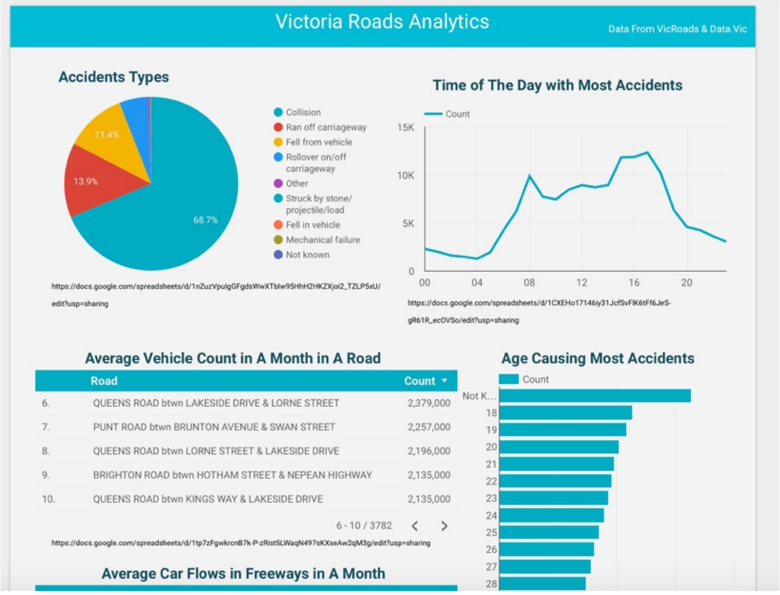
List Accident Records

Insert New Record

Update Record

Delete Record

For update and delete records, user need to provide record id, which is not necessary for insertRecord. On the webpage, user can click any analysis charts to view counts, and choose to look for top 5, top 10 and continuing results etc.



Victoria Roads Analytics

Data From VicRoads & Data.Vic

Accidents Types

https://docs.google.com/spreadsheets/d/1nZuzVpulgGFgdsWwXTblw95HhH2HKZXjoi2_TZLP5xU/edit?usp=sharing

Time of The Day with Most Accidents

https://docs.google.com/spreadsheets/d/1CXEHo17146iy31JcfSvFIK6tFf6JeS-gR61R_ecOVSo/edit?usp=sharing

Average Vehicle Count in A Month in A Road

	Road	Count ▾
6.	QUEENS ROAD btwn LAKESIDE DRIVE & LORNE STREET	2,379,000
7.	PUNT ROAD btwn BRUNTON AVENUE & SWAN STREET	2,257,000
8.	QUEENS ROAD btwn LORNE STREET & LAKESIDE DRIVE	2,196,000
9.	BRIGHTON ROAD btwn HOTHAM STREET & NEPEAN HIGHWAY	2,135,000
10.	QUEENS ROAD btwn KINGS WAY & LAKESIDE DRIVE	2,135,000

6 - 10 / 3782 ‹ ›

https://docs.google.com/spreadsheets/d/1tp7zFgwkrcnB7k-P-zRist5LWaqN497sKXseAw2qM3g/edit?usp=sharing

Age Causing Most Accidents

Average Car Flows in Freeways in A Month