Belong Data Engineer Coding Exercise

REPORT

Melbourne City Pedestrian Counting Analysis

—— by Sunny Miao Sun

Introduction

The purpose of this project is to look into the Melbourne City Pedestrian Counting Data with sensor locations and get the top 10 locations with the most pedestrian count by day and by month.

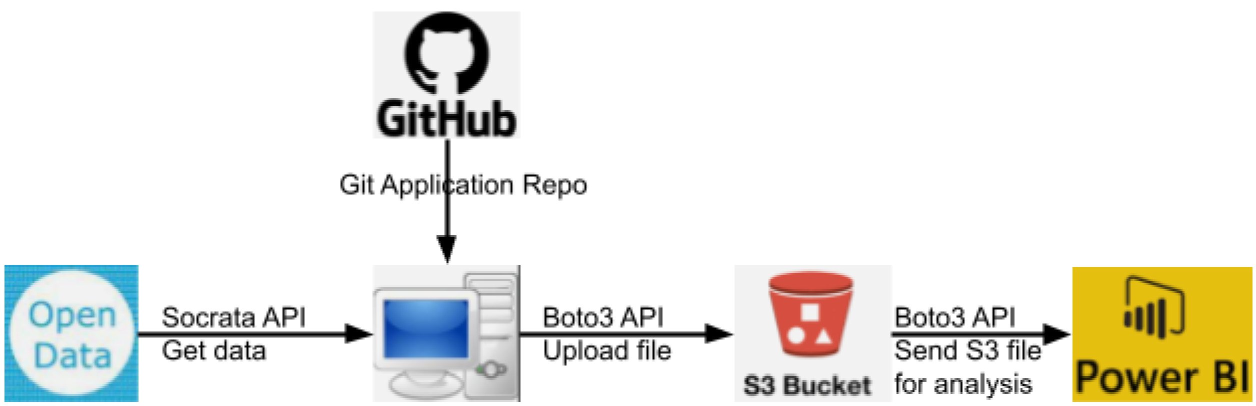
Public Git Repo location: https://github.com/aussunny/Mel_Ped

Requirement:

1. Documentation about the approach, architecture.
2. Script for data analysis.
3. Instruction on how to run the script on the AWS or local environment.
4. Submit a GitHub repo
5. Load the results data into S3 in CSV format for future querying.

Application Architecture

The architecture of this application is designed as below:



Data Wrangling

There are 2 pieces of data required for this project listed as below:

1. Pedestrian Counting System - Monthly (counts per hour)

Link:

https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-Monthly-counts-per-hour/b2ak-trbp

Columns in this Dataset

Column Name	Description	Type
ID	Unique reading ID	Number #
Date_Time	Date and time of reading (dd/mm/yyyy hh:mm:ss)	Date & Time
Year	Year of reading	Number #
Month	Month of year (January, February, ...)	Plain Text T
Mdate	Day of year (1, 2, 3, ..., 31)	Number #
Day	Day of week (Monday, Tuesday, ..., Sunday)	Plain Text T
Time	Time of day (0 = midnight-1am, 1 = 1am-2am, 2 = 2am-3am...	Number #
Sensor_ID	Sensor ID	Number #
Sensor_Name	Sensor name	Plain Text T
Hourly_Counts	Total hourly sensor readings (count of pedestrians)	Number #

2. Pedestrian Counting System - Sensor Locations

Link:

https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-Sensor-Locations/h57g-5234

Columns in this Dataset

Column Name	Description	Type
sensor_id	The unique ID of the sensor	Number
sensor_description	A description of where the sensor is located	Plain Text
sensor_name	The name of the sensor	Plain Text
installation_date	When the sensor was installed	Date & Time
status	Status of each sensor (A = Active I = Inactive R = Removed)	Plain Text
note	Further infomation on changes to sensor status or location	Plain Text
direction_1	Direction 1 of the sensor reading	Plain Text
direction_2	Direction 2 of the sensor reading	Plain Text
latitude	Latitude of each sensor	Number
longitude	Longitude of each sensor	Number
location	The latitude and longitude of each sensor	Location

The two pieces of data above are programmatically acquired from the City of Melbourne Open Data API powered by Socrata. Registration is required to get the API Token for downloading dataset. The Dataset is directly transferred into pandas dataframe for the following steps.

Note that we can save the dataset to the S3 bucket and use the S3 Select to retrieve the data via SQL (with selected columns or criteria). Due to the limit of time, I will just use the dataset directly and manipulate it with python.

Data Assessment and Cleaning

In the data assessment stage, I have looked into the dataset, check the data type for each column and drop any null or duplicated records. The first 5 rows of the pedestrian counting dataset are shown below.

```
count_df.head()
```

	date_time	day	hourly_counts	id	mdate	month	sensor_id	sensor_name	time	year
0	2019-11-01T17:00:00.000	Friday	300	2887628	1	November	34	Flinders St-Spark La	17	2019
1	2019-11-01T17:00:00.000	Friday	604	2887629	1	November	39	Alfred Place	17	2019
2	2019-11-01T17:00:00.000	Friday	216	2887630	1	November	37	Lygon St (East)	17	2019
3	2019-11-01T17:00:00.000	Friday	627	2887631	1	November	40	Lonsdale St-Spring St (West)	17	2019
4	2019-11-01T17:00:00.000	Friday	774	2887632	1	November	36	Queen St (West)	17	2019

For the pedestrians counting dataset, Only the columns of "hourly_counts", "month", "year" , "sensor_id" are retained; the original "date_time" columns are trimmed into a string of YYYY-MM-DD format as "Day" as shown below.

```
c_df.head()
```

	Hourly_Counts	Month	Sensor_ID	Year	Day
33060	921	May	12	2009	2009-05-01
38878	4093	May	4	2009	2009-05-01
38877	2525	May	2	2009	2009-05-01
38876	3676	May	1	2009	2009-05-01
38875	701	May	18	2009	2009-05-01

Data Analysis

To find the top 10 locations with the most pedestrian count per day, I calculated the sum of pedestrian count for each day and each sensor and retained the top 10 records with the most pedestrian count for each day. Then I combined the location table with the output day count dataset over the Sensor_ID. The final output for the Top 10 Locations with Most Dedestrian Daily Count is shown below.

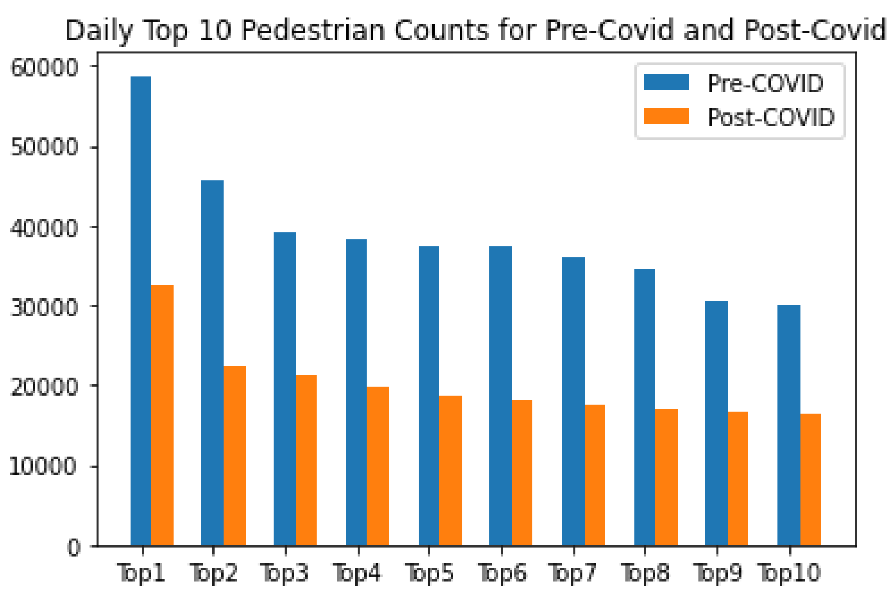
:

	Day	Daily_Top10	Sensor_ID	Daily_Counts	Location	Sensor_Description	Latitude	Longitude
0	2009-05-01	0	4	45185	{'latitude': '-37.81487988', 'longitude': '144...	Town Hall (West)	-37.81487988	144.9660878
1	2009-05-01	1	1	36869	{'latitude': '-37.8134944', 'longitude': '144....	Bourke Street Mall (North)	-37.8134944	144.96515324
2	2009-05-01	2	6	29015	{'latitude': '-37.81911704', 'longitude': '144...	Flinders Street Station Underpass	-37.81911704	144.96558256
3	2009-05-01	3	2	27587	{'latitude': '-37.81380667', 'longitude': '144...	Bourke Street Mall (South)	-37.81380667	144.96516719
4	2009-05-01	4	5	25590	{'latitude': '-37.81874249', 'longitude': '144...	Princes Bridge	-37.81874249	144.96787656

Similar steps have been done to get the Top 10 Locations with Most Dedestrian Monthly Count, the head of the dataset is shown below.

	Year	Month	Monthly_Top10	Sensor_ID	Monthly_Counts	Location	Sensor_Description	Latitude	Longitude
0	2009	August	0	4	1050461	{'latitude': '-37.81487988', 'longitude': '144...	Town Hall (West)	-37.81487988	144.9660878
1	2009	August	1	2	847853	{'latitude': '-37.81380667', 'longitude': '144...	Bourke Street Mall (South)	-37.81380667	144.96516719
2	2009	August	2	3	827432	{'latitude': '-37.81101523', 'longitude': '144...	Melbourne Central	-37.81101523	144.96429485
3	2009	August	3	6	721539	{'latitude': '-37.81911704', 'longitude': '144...	Flinders Street Station Underpass	-37.81911704	144.96558256
4	2009	August	4	5	700272	{'latitude': '-37.81874249', 'longitude': '144...	Princes Bridge	-37.81874249	144.96787656

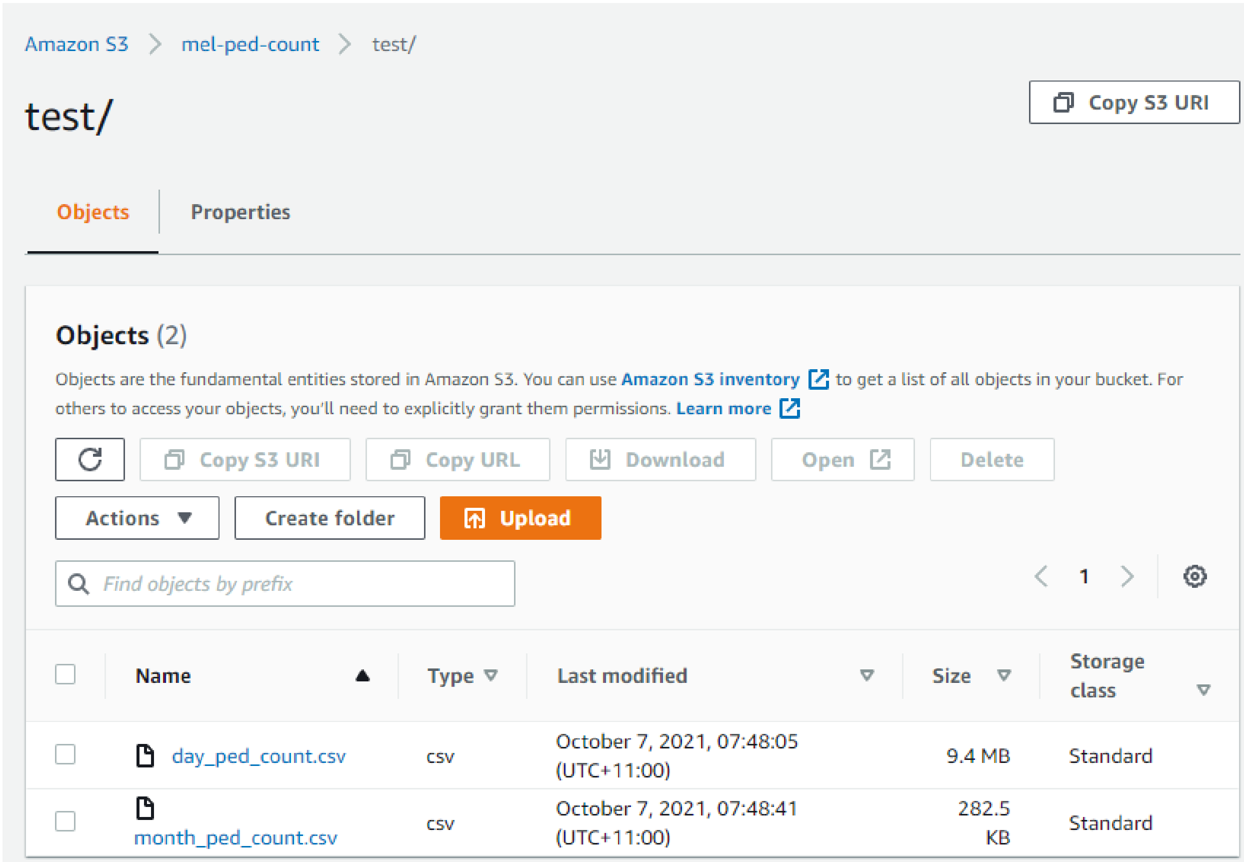
To further investigate the data and get some extra insights, I've sliced the daily count data into pre-covid set and post-covid set and calculated the mean counting for each sensor over time. I found that after covid, the pedestrians inside Melbourne city nearly halved. There is a lot to dig into. For example, we could easily look into the top 10 locations to get more insights on the pedestrian movement routine change during Covid. Due to time limitations, I just did the below exploration.



Results storing

I created an S3 bucket name: s3://mel-ped-count for file storing. I use the python package boto3 to upload the CSV files.

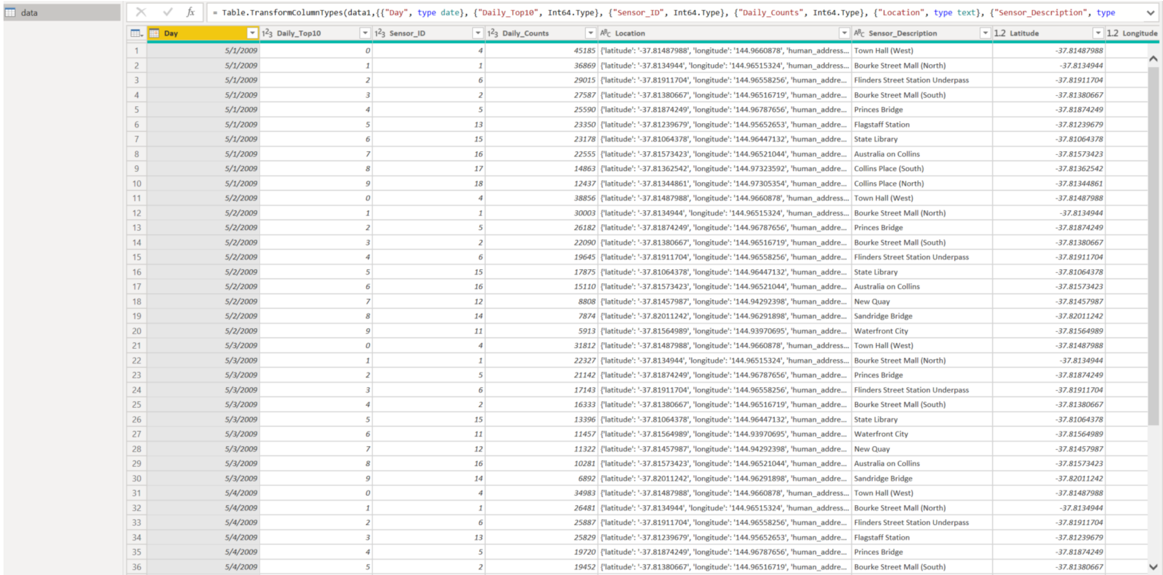
```
> aws s3 mb s3://mel-ped-count
```



Data Store

Amazon S3 Select works on objects stored in CSV, JSON, or Apache Parquet format. Given this, the output CSV file saved in S3 can be queried directly via SQL without configuring a traditional database. But I assume we are expecting a different approach here.

There are many different kinds of data stores. I choose Power BI as a data store as it is versatile and easy to use. I load the S3 file to the Power BI for future investigation via a python script. You will need to configure the AWS CLI in order to use the script in the git repo. Note that I could provision an AWS RDS instance, due to the time limit I use Power BI.



data

= Table.TransformColumnTypes(data1,{{"Day", type date}, {"Daily_Top10", Int64.Type}, {"Sensor_ID", Int64.Type}, {"Daily_Counts", Int64.Type}, {"Location", type text}, {"Sensor_Description", type

	Day	Daily_Top10	Sensor_ID	Daily_Counts	Location	Sensor_Description	Latitude	Longitude
1	5/1/2009	0	4	45185	{'latitude': '-37.81487988', 'longitude': '144.9660878', 'human_address...	Town Hall (West)	-37.81487988	
2	5/1/2009	1	1	36869	{'latitude': '-37.8134944', 'longitude': '144.96515324', 'human_address...	Bourke Street Mall (North)	-37.8134944	
3	5/1/2009	2	6	29015	{'latitude': '-37.81911704', 'longitude': '144.96558256', 'human_addre...	Flinders Street Station Underpass	-37.81911704	
4	5/1/2009	3	2	27587	{'latitude': '-37.81380667', 'longitude': '144.96516719', 'human_addre...	Bourke Street Mall (South)	-37.81380667	
5	5/1/2009	4	5	25590	{'latitude': '-37.81874249', 'longitude': '144.96787656', 'human_addre...	Princes Bridge	-37.81874249	
6	5/1/2009	5	13	23350	{'latitude': '-37.81239679', 'longitude': '144.95652653', 'human_addre...	Flagstaff Station	-37.81239679	
7	5/1/2009	6	15	23178	{'latitude': '-37.81064378', 'longitude': '144.96447132', 'human_addre...	State Library	-37.81064378	
8	5/1/2009	7	16	22555	{'latitude': '-37.81573423', 'longitude': '144.96521044', 'human_addre...	Australia on Collins	-37.81573423	
9	5/1/2009	8	17	14863	{'latitude': '-37.81362542', 'longitude': '144.97323592', 'human_addre...	Collins Place (South)	-37.81362542	
10	5/1/2009	9	18	12437	{'latitude': '-37.81344861', 'longitude': '144.97305354', 'human_addre...	Collins Place (North)	-37.81344861	
11	5/2/2009	0	4	38856	{'latitude': '-37.81487988', 'longitude': '144.9660878', 'human_address...	Town Hall (West)	-37.81487988	
12	5/2/2009	1	1	30003	{'latitude': '-37.8134944', 'longitude': '144.96515324', 'human_address...	Bourke Street Mall (North)	-37.8134944	
13	5/2/2009	2	5	26182	{'latitude': '-37.81874249', 'longitude': '144.96787656', 'human_addre...	Princes Bridge	-37.81874249	
14	5/2/2009	3	2	22090	{'latitude': '-37.81380667', 'longitude': '144.96516719', 'human_addre...	Bourke Street Mall (South)	-37.81380667	
15	5/2/2009	4	6	19645	{'latitude': '-37.81911704', 'longitude': '144.96558256', 'human_addre...	Flinders Street Station Underpass	-37.81911704	
16	5/2/2009	5	15	17875	{'latitude': '-37.81064378', 'longitude': '144.96447132', 'human_addre...	State Library	-37.81064378	
17	5/2/2009	6	16	15110	{'latitude': '-37.81573423', 'longitude': '144.96521044', 'human_addre...	Australia on Collins	-37.81573423	
18	5/2/2009	7	12	8808	{'latitude': '-37.81457987', 'longitude': '144.94292398', 'human_addre...	New Quay	-37.81457987	
19	5/2/2009	8	14	7874	{'latitude': '-37.82011242', 'longitude': '144.96291898', 'human_addre...	Sandridge Bridge	-37.82011242	
20	5/2/2009	9	11	5913	{'latitude': '-37.81564989', 'longitude': '144.93970695', 'human_addre...	Waterfront City	-37.81564989	
21	5/3/2009	0	4	31812	{'latitude': '-37.81487988', 'longitude': '144.9660878', 'human_address...	Town Hall (West)	-37.81487988	
22	5/3/2009	1	1	22327	{'latitude': '-37.8134944', 'longitude': '144.96515324', 'human_address...	Bourke Street Mall (North)	-37.8134944	
23	5/3/2009	2	5	21142	{'latitude': '-37.81874249', 'longitude': '144.96787656', 'human_addre...	Princes Bridge	-37.81874249	
24	5/3/2009	3	6	17143	{'latitude': '-37.81911704', 'longitude': '144.96558256', 'human_addre...	Flinders Street Station Underpass	-37.81911704	
25	5/3/2009	4	2	16333	{'latitude': '-37.81380667', 'longitude': '144.96516719', 'human_addre...	Bourke Street Mall (South)	-37.81380667	
26	5/3/2009	5	15	13396	{'latitude': '-37.81064378', 'longitude': '144.96447132', 'human_addre...	State Library	-37.81064378	
27	5/3/2009	6	11	11457	{'latitude': '-37.81564989', 'longitude': '144.93970695', 'human_addre...	Waterfront City	-37.81564989	
28	5/3/2009	7	12	11322	{'latitude': '-37.81457987', 'longitude': '144.94292398', 'human_addre...	New Quay	-37.81457987	
29	5/3/2009	8	16	10281	{'latitude': '-37.81573423', 'longitude': '144.96521044', 'human_addre...	Australia on Collins	-37.81573423	
30	5/3/2009	9	14	6892	{'latitude': '-37.82011242', 'longitude': '144.96291898', 'human_addre...	Sandridge Bridge	-37.82011242	
31	5/4/2009	0	4	34983	{'latitude': '-37.81487988', 'longitude': '144.9660878', 'human_address...	Town Hall (West)	-37.81487988	
32	5/4/2009	1	1	26481	{'latitude': '-37.8134944', 'longitude': '144.96515324', 'human_address...	Bourke Street Mall (North)	-37.8134944	
33	5/4/2009	2	6	25887	{'latitude': '-37.81911704', 'longitude': '144.96558256', 'human_addre...	Flinders Street Station Underpass	-37.81911704	
34	5/4/2009	3	13	25829	{'latitude': '-37.81239679', 'longitude': '144.95652653', 'human_addre...	Flagstaff Station	-37.81239679	
35	5/4/2009	4	5	19720	{'latitude': '-37.81874249', 'longitude': '144.96787656', 'human_addre...	Princes Bridge	-37.81874249	
36	5/4/2009	5	2	19452	{'latitude': '-37.81380667', 'longitude': '144.96516719', 'human_addre...	Bourke Street Mall (South)	-37.81380667	

Steps to execute the application

1. Git clone the repo

```
> git clone https://github.com/aussunny/Mel_Ped.git
```

2. Replace the credential placeholder with your own credential for downloading data from Socrata and upload the file to S3 Bucket. (Please also use/create ur own bucket and configure the AWS CLI)
3. Execute the python script, it will do the data analysis, generate the daily count and monthly count CSV files locally and upload them to the S3 bucket under the path "s3://mel-ped-count/test/".

```
> python Mel_Ped\Belong_Code_Test.py
```

4. Upload the S3 file to Power BI via a python script. The script is included in the repo.

Future Improvement

1. The location information does not contain the human address. Next step we could leverage python geo packages(like geopandas) to map the sensor location(latitude

and longitude) into a geomap, plot the data into a heat map, and fill out the human address for each sensor location(like street, state, postcode, etc).

2. We can save the original dataset to the S3 bucket and use the S3 Select to retrieve the data via SQL (for pre-filtering). But for our case, which is a simple and small project, we can directly use the dataframe downloaded from the source.

3. We could provision an ec2 instance instead of a local machine to run the python file and link with the S3 bucket.