Appendix

3.0.1 Data Cleaning part 1

See “data-need-clean.json”

3.0.2 Data cleaning part 2

Used SQL to clean null values.

```
-- get bus stops that without any route
SELECT
	BS.*,
	subRS.route_count
FROM
	busstop BS
	left outer join (select bus_stop_id, count(*) route_count from subroutestop group by
bus_stop_id) subRS on BS.id = subRS.bus_stop_id
WHERE
	subRS.route_count is null;
-- get routes that without any bus stop
SELECT
	subR.*,
	subRS.stop_count
FROM
	bussubroute subR
	left outer join (select sub_route_id, count(*) stop_count from subroutestop group by
sub_route_id) subRS on subR.id = subRS.sub_route_id
WHERE
	subRS.stop_count is null;
```

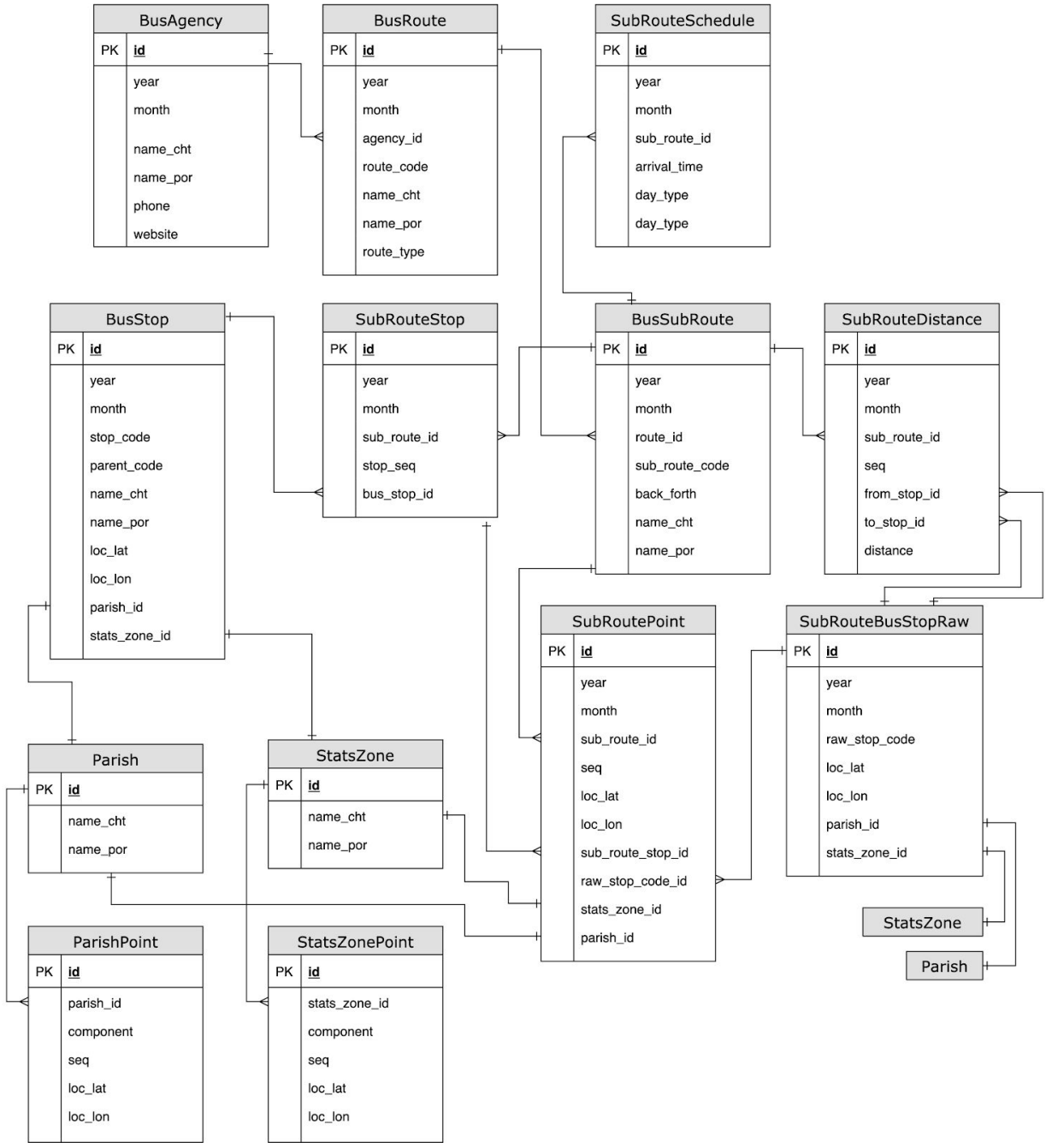
3.0.3 Data cleaning part 3

Python codes, snippet from “processer.py”, shown in “data-need-clean-3.txt”.

```
def get_bus_stop(year, month, busStop):
    originBusStopCode = copy.copy(busStop['code'])
    count = 0
    busStopRec = None

    busStop['code'] = re.sub(r'/(.+)', '', busStop['code'])
    busStop['code'] = re.sub(r'/(.+)', '', busStop['code'])
    while (busStopRec == None) and count < 3:
        try:
            busStopRec = BusStop.select().where(
```

```
            (BusStop.year == year) &
            (BusStop.month == month) &
            (BusStop.stop_code == busStop['code'])
        ).get()
    except DoesNotExist:
        busStopRec = None
    if busStopRec == None:
        try:
            busStopRec = BusStop.select().where(
                (BusStop.year == year) &
                (BusStop.month == month) &
                (BusStop.parent_code == busStop['code'])
            ).get()
        except DoesNotExist:
            busStopRec = None
    if busStopRec == None:
        busStop['code'] = re.sub(r'\_[^_]+$', '', busStop['code'])
    count += 1
if busStopRec == None:
    try:
        busStopCode = re.findall(r'([^\_]+\_).+', busStop['code'])
        busStopCode = busStopCode if len(busStopCode) > 0 else busStop['code'] + '_'

        busStopRec = BusStop.select().where(
            (BusStop.year == year) &
            (BusStop.month == month) & (
                (BusStop.stop_code.startswith(busStopCode)) |
                (BusStop.parent_code.startswith(busStopCode))
            )
        ).get()
    except DoesNotExist:
        busStopRec = None
if (not busStopRec == None) and (not originBusStopCode == busStopRec.stop_code):
    return originBusStopCode, busStopRec
return None, busStopRec
```

4.0.1 Database Structure diagram



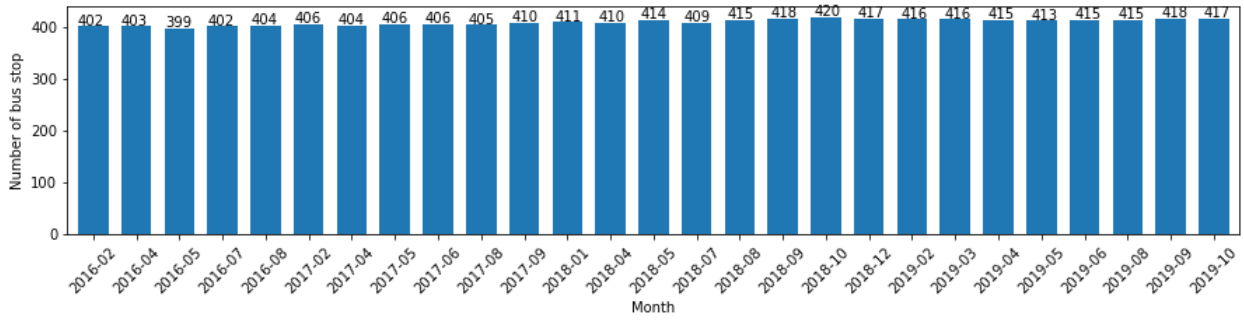
4.1.1 Number of routes on Bus Agencies

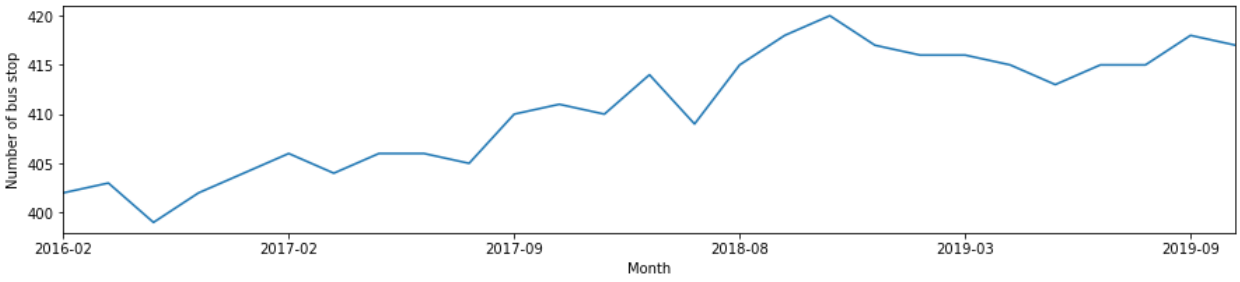
Output from “simple_analyze_and_visualization_using_python.ipynb”, also shown in jupyter notebook.

```
Year: 2016
     Agency  Number of routes
Rank
1       新時代                31
2       新福利                27
3        澳巴                 20
************************************************

Year: 2017
     Agency  Number of routes
Rank
1       新時代                33
2       新福利                29
3        澳巴                 20
************************************************

Year: 2018
     Agency  Number of routes
Rank
1        澳巴                 55
2       新福利                29
************************************************

Year: 2019
     Agency  Number of routes
Rank
1        澳巴                 55
2       新福利                30
************************************************
```

4.1.2 Number of bus stop in month

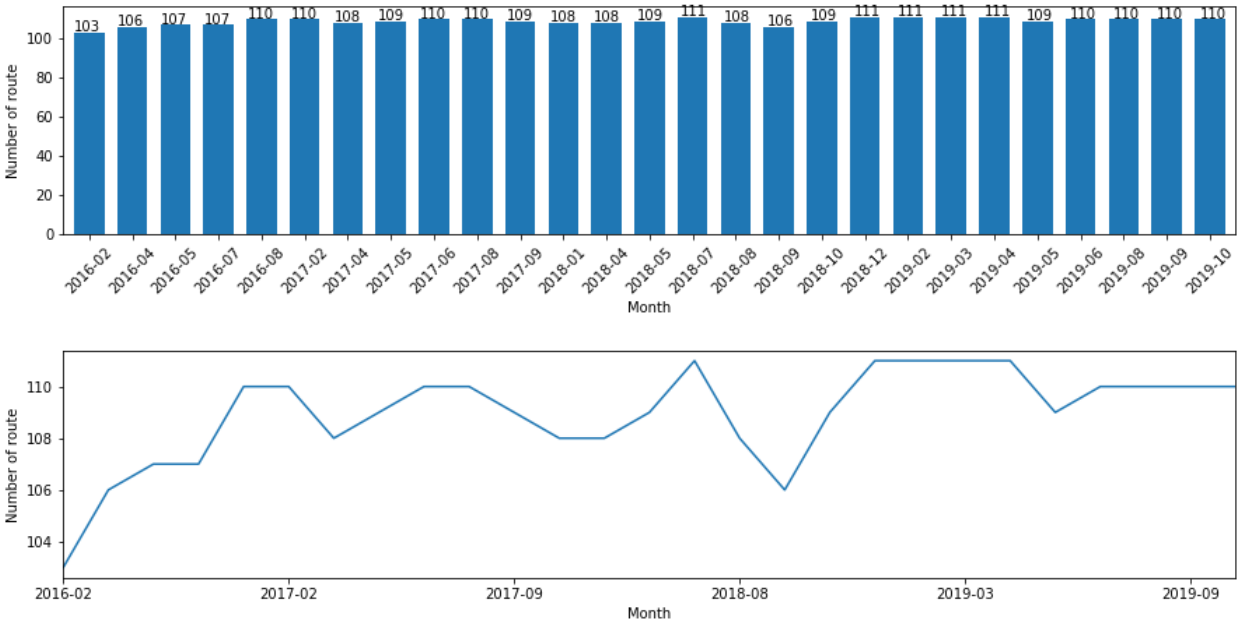
Output from “simple_analyze_and_visualization_using_python.ipynb”, also shown in jupyter notebook.





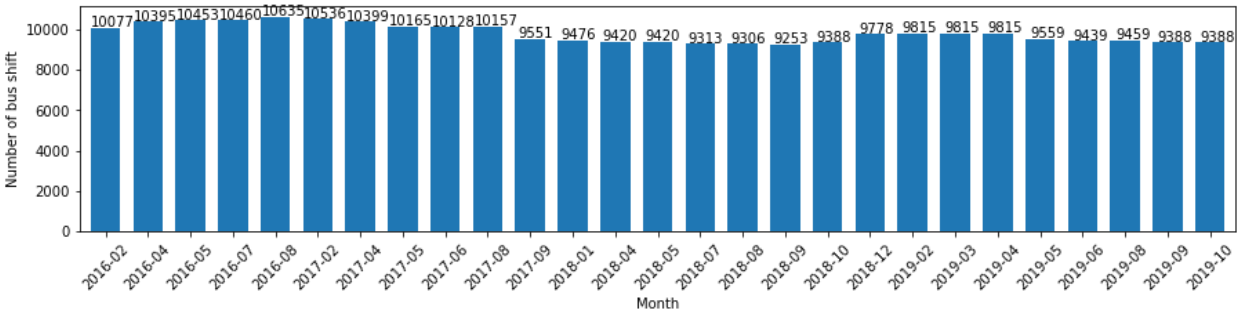
4.1.3 Number of bus routes in month

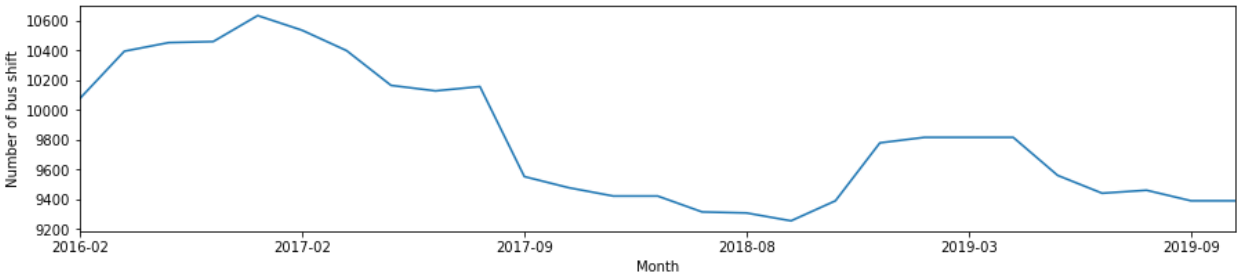
Output from “simple_analyze_and_visualization_using_python.ipynb”, also shown in jupyter notebook.



4.1.4 Number of bus shift in weekday

Output from “simple_analyze_and_visualization_using_python.ipynb”, also shown in jupyter notebook.


Number of bus shift
Month
10077
10395
10453
10460
10635
10536
10399
10165
10128
10157
9551
9476
9420
9420
9313
9306
9253
9388
9778
9815
9815
9815
9559
9439
9459
9388
9388
2016-02
2016-04
2016-05
2016-07
2016-08
2017-02
2017-04
2017-05
2017-06
2017-08
2017-09
2018-01
2018-04
2018-05
2018-07
2018-08
2018-09
2018-10
2018-12
2019-02
2019-03
2019-04
2019-05
2019-06
2019-08
2019-09
2019-10


Number of bus shift
Month
2016-02
2017-02
2017-09
2018-08
2019-03
2019-09

4.1.5 Number of bus stops on different Parish

Output from “simple_analyze_and_visualization_using_python.ipynb”, also shown in jupyter notebook.

4.1.6 Top 10 bus stop that pass by most no. of routes

Output from “simple_analyze_and_visualization_using_python.ipynb”, also shown in jupyter notebook.

```
In 2016-02:
        Bus Stop  Number of routes
Rank
1          亞馬喇前地                65
2           關閘總站                37
3           媽閣總站                23
4           外港碼頭                22
5          石排灣總站                22
6           葡京酒店                20
7          蘇利安圓形地                20
8          路氹邊檢大樓                18
9           台山街市                18
10    水坑尾/公共行政大樓                17
************************************************
...
...
In 2019-10:
         Bus Stop  Number of routes
Rank
1           亞馬喇前地                69
2            葡京酒店                22
3            媽閣總站                22
4          蘇利安圓形地                22
5            關閘總站                21
6            外港碼頭                21
7     提督馬路/雅廉訪臨時站                19
8            湖畔大廈                19
9           百利寶花園                18
10           台山街市                18
************************************************
```

4.1.7 Top 10 longest distance of bus route

Output from “simple_analyze_and_visualization_using_python.ipynb”, also shown in jupyter notebook.

```
In 2016-02:
     Route Code  Total Distance
Rank
1            26       48.884649
2           21A       43.199310
3            25       40.143834
4           26A       40.083776
```

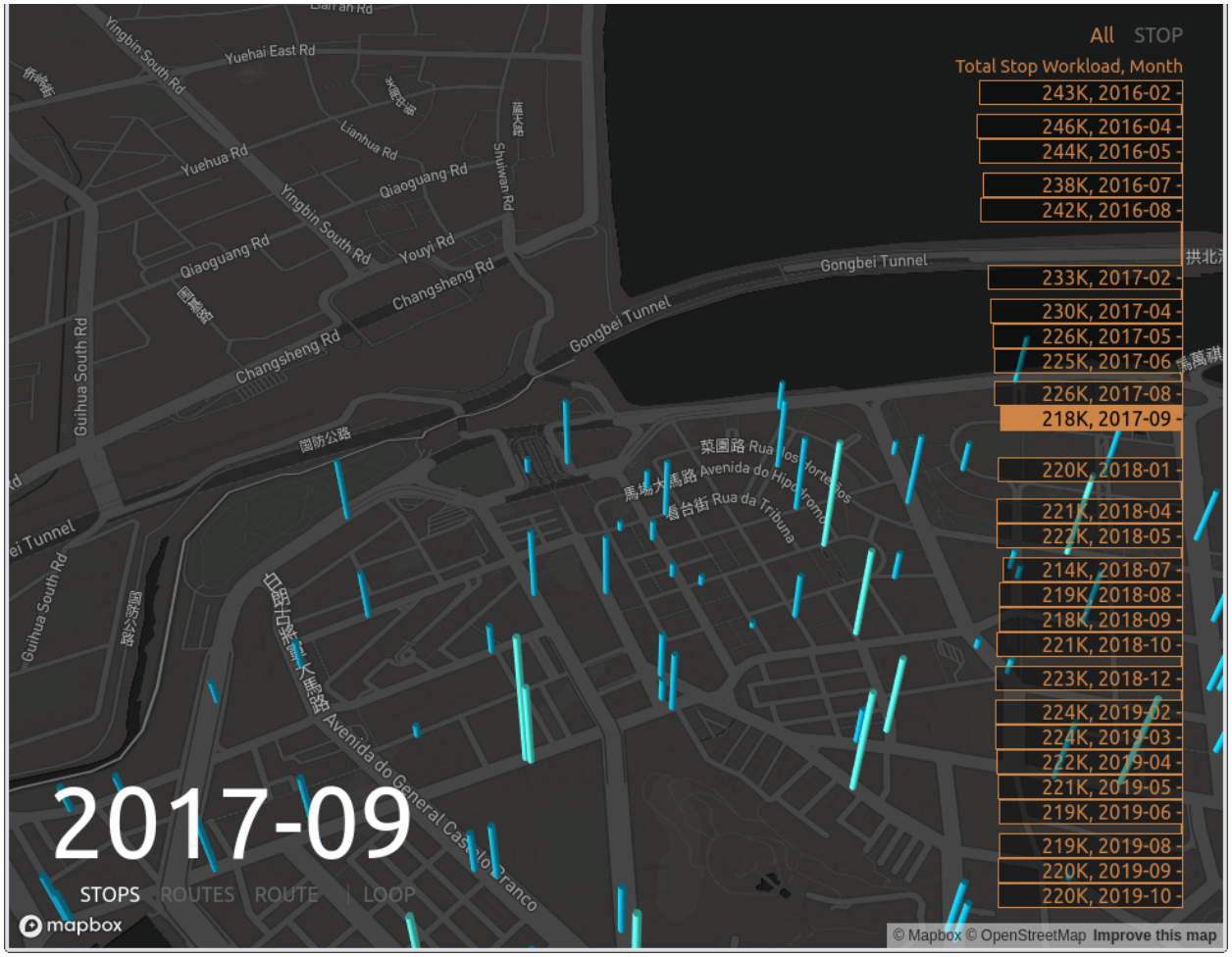
```
5             73       33.265359
6             15       32.686540
7            MT4       29.937282
8             N5       29.867948
9             59       29.711536
10            N2       29.361851
**************************************************
...
...
In 2019-10:
     Route Code  Total Distance
Rank
1             26       47.084130
2            21A       46.886480
3            26A       42.016293
4            MT4       36.977678
5             N2       34.652382
6             15       34.425101
7            51A       33.509624
8             73       33.242311
9             25       32.253031
10            N3       29.655544
**************************************************
```

5.3.1 Visualization border gate before Hato

Screenshot from visualization of border gate terminal before the typhoon Hato.

5.3.2 Visualization closed border gate stop after Hato

Screenshot from visualization of border gate terminal after the typhoon Hato.



5.3.2 Visualization reopened border gate stop after Hato

Screenshot from visualization of border gate terminal after the typhoon Hato, stop reopen.