ATAC buses analysis

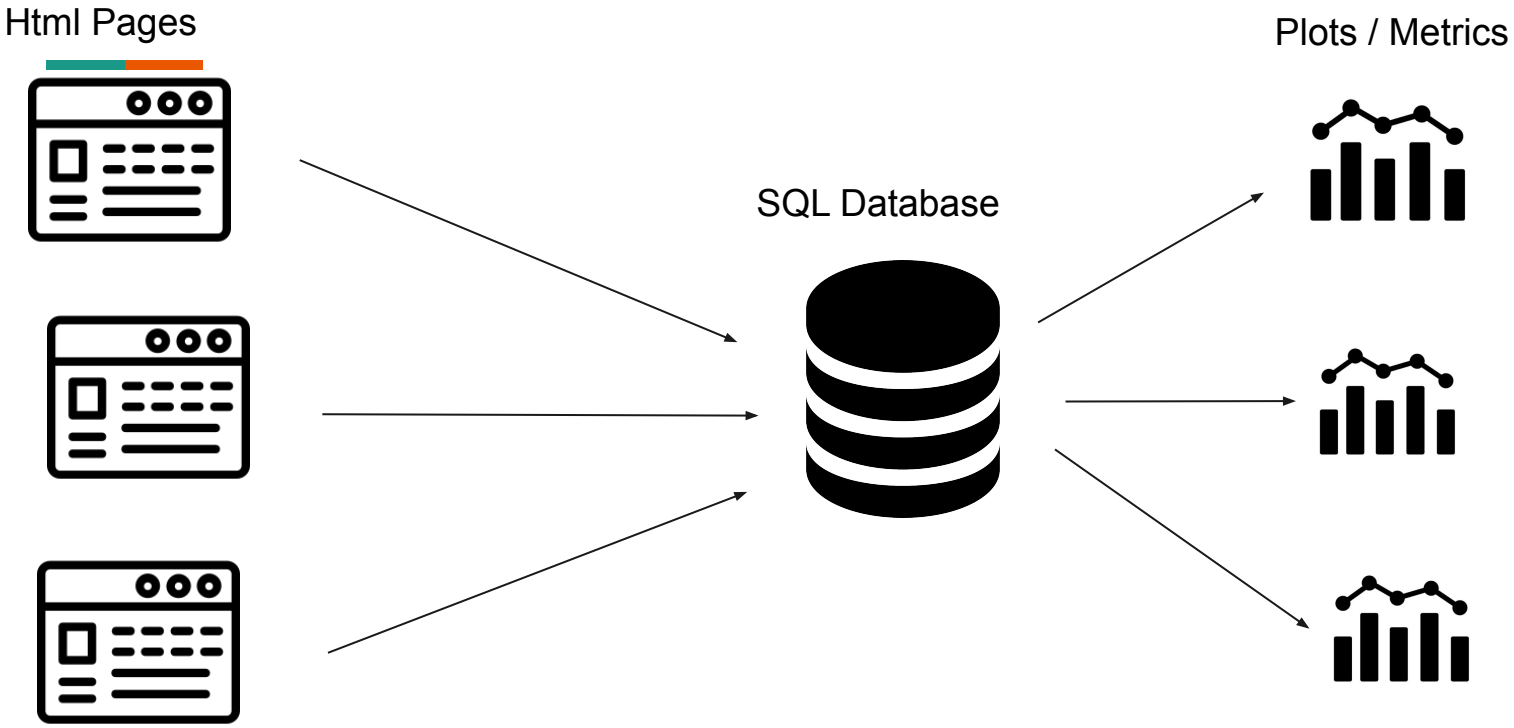
From 20 march to 30 june

Diego Sonaglia, Ibtissam Lachhab

The project:

- Dataset
  - Around 20 Gb of (compressed) html pages crawled from atac website each 15 minutes
- Goal
  - Find Out if Atac's buses lateness is related to traffic
- Approach:
  - Parse raw data into a compact SQL database and then use it to retrieve metrics
- Problems:
  - Locate Useful data in the html pages
  - Data Size / Parsing Time
  - Data Granularity
  - Data Noise
  - Finding a meaningful metric

Approach



Locate data

Daily scheduled departures

20 di martedì: 00 19 39
21 di martedì: 00
5: 20 38 56
6: 15 34 52
7: 06 20 35 50
8: 04 19 33 49
9: 05 21 37 54
10: 11 34
11: 02 29 57
12: 25 52
13: 20 49
14: 18 49
15: 19 50
16: 17 43
17: 10 36 58
18: 19 40
19: 01 21 41
20: 00 19 39
21: 00

Altri giorni

» Martedì 31 marzo
» Mercoledì 1 aprile
» Giovedì 2 aprile
» Venerdì 3 aprile
» Sabato 4 aprile
» Domenica 5 aprile
» Lunedì 6 aprile

Sequence of stops for a route

* Nomentana/G. B. Vico
* Nomentana/Monte Livata
* Nomentana/Monte Bianco
* Nomentana/Poggio Fiorito
* Nomentana/Palombarese
* Nomentana/F.lli Maristi
* Nomentana/De La Riva
* Nomentana/Dante Da Maiano
* Nomentana/Catacombe
* Nomentana/Podere S. Alessandro
* Nomentana/Cesarina
* Nomentana/G.r.a.
* Nomentana/Scuola Rurale
* Nomentana/Spaducci
* Nomentana/Tosatti
* Casal Boccone/Negri A.
* Ojetti/Casal Boccone
* Ojetti/Aleramo
* Ojetti/Pugliese
* Ojetti/Primoli
* Ojetti/Tosatti
* Jonio/Talenti
* Jonio/Bandello
* Stelvio
* Adamello
* Carnaro
* Monte Baldo
* Tirreno/Sempione
* Tirreno/Isole Eolie
* Conca D'oro (MB1)
* Conca D'oro (MB1)
* Conca D'oro (MB1)

Bus Position

* Nomentana/Monte Livata
* Nomentana/Monte Bianco
B Nomentana/Poggio Fiorito
* Nomentana/Palombarese
* Nomentana/F.lli Maristi
* Nomentana/De La Riva
* Nomentana/Dante Da Maiano

Traffic State

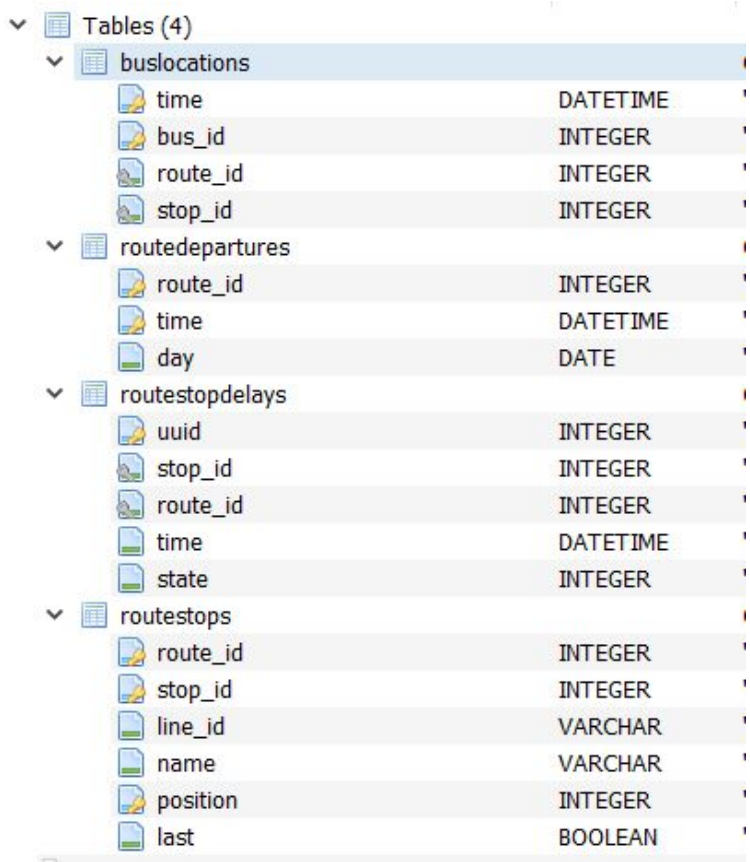
* Nomentana/Monte Livata
* Nomentana/Monte Bianco
B Nomentana/Poggio Fiorito
* Nomentana/Palombarese
* Nomentana/F.lli Maristi
* Nomentana/De La Riva
* Nomentana/Dante Da Maiano

Locate data 2

Route Id

Bus Id

Sqlite Database



Data Size / Parsing Time

Problem

- The dataset is quite heavy, extracting , moving or deleting the pages can take hours !
- Parsing the whole dataset can take days and one error may force to restart from the beginning

Oursolution:

- Parse one bus line at a time, directly reading from the archive without extracting it with **winrar** library and parsing the html pages with **beautifulsoup4**
- Parsing all the days from one line takes 20-30 minutes this way
- Allows for short parsing / analysis / validation cycles

Data Granularity

- As we can see 15 minutes is a long time, a bus can perform up to 10/20 stops in that time
- Buses appears on the first stop of the line several times before it actually departs
- Possible solution:
  - Interpolation
  - Using metrics that are meaningful with the granularity level we have

	line_id	route_id	bus_id	position	time(time)
1	88	53432	3376	0	07:03:11
2	88	53432	3376	14	07:18:19
3	88	53432	3376	20	07:26:02
4	88	53432	3376	29	07:33:27
5	88	53432	3376	42	07:40:47
6	88	53432	3376	0	08:33:07
7	88	53432	3376	0	08:40:35
8	88	53432	3376	0	08:48:00
9	88	53432	3376	4	08:55:42
10	88	53432	3376	23	09:10:36
11	88	53432	3376	31	09:18:16
12	88	53432	3376	42	09:25:53
13	88	53432	3376	0	10:26:42
14	88	53432	3376	0	10:34:12
15	88	53432	3376	0	10:41:38
16	88	53432	3376	8	10:49:18
17	88	53432	3376	16	10:56:48
18	88	53432	3376	25	11:04:13

Possible metric 1 (Average Lateness from scheduled arrival)

$$M_{1,\,day} = \sum_{s \in stop(day)} \frac{|time(s) - closest\_sched\_time(s)|}{|\ stops(day)\ |}$$

- Idea: find the average difference between scheduled arrival time and actual arrival time
- Problems:
  - Finding a valid arrival schedule for each stop and each day
    - Atac doesn't seem to provide one
    - Third party may derive it from other metrics
  - Data only show when bus is close to a stop, even if it has not actually stopped
  - Data granularity is 15 minutes

- Appears to be fragile

Possible metric 2 (End to end average time)

$$M_{2,\,day} = \sum_{r \in runs(day)} \frac{arrival\_time(r) - departure\_time(r)}{|\,runs(day)\,|}$$

- End to End time:
  - Time required to travel from one end to the other of a route,
- Problems:
  - We need to find a bus on both ends
  - Dataset is somehow unbalanced under this point of view

	line_id	route_id	position	count(*)	count(distinct(date(buslocations.time)))
1	88	53432	0	3036	53
2	88	53432	1	48	32
3	88	53432	2	73	39
4	88	53432	3	110	43
5	88	53432	4	211	51
6	88	53432	5	178	53
7	88	53432	6	204	50
8	88	53432	7	306	53
37	88	53432	36	67	41
38	88	53432	37	369	53
39	88	53432	38	113	43
40	88	53432	39	91	40
41	88	53432	40	229	52
42	88	53432	41	54	31
43	88	53432	42	109	45
44	88	53432	43	202	48

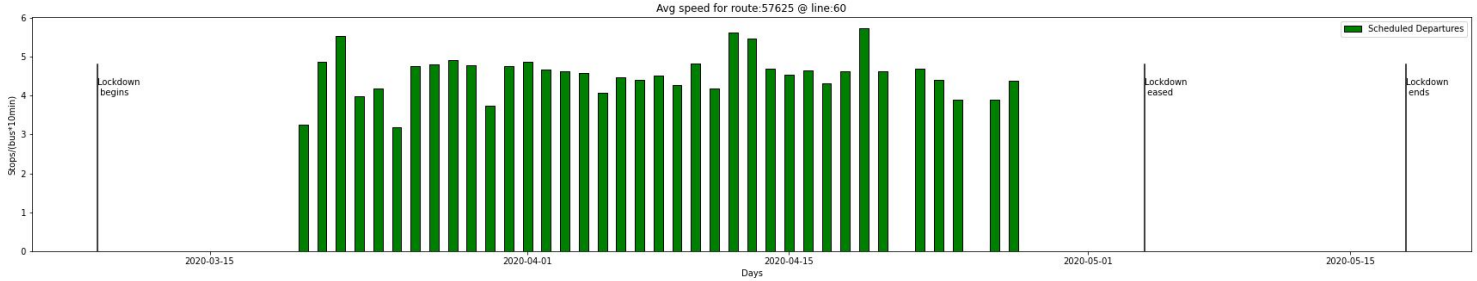
Possible metric 2b (Average Speed stops/10min)

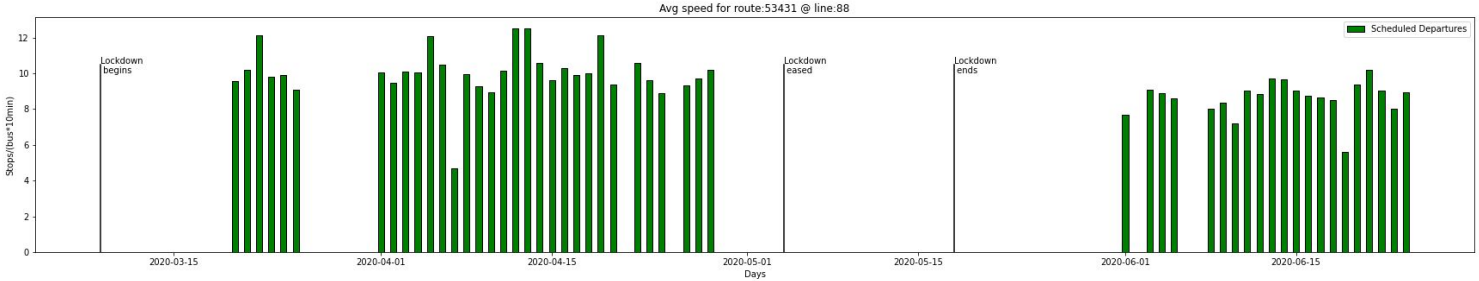
$$M_{2b,\,day} = \sum_{bus \in buses(day)} \sum_{s1,s2 \in succ\_stops(day,\,bus)} \frac{stop\_position(s2) - stop\_position(s1)}{time(s2) - time(s1)}$$

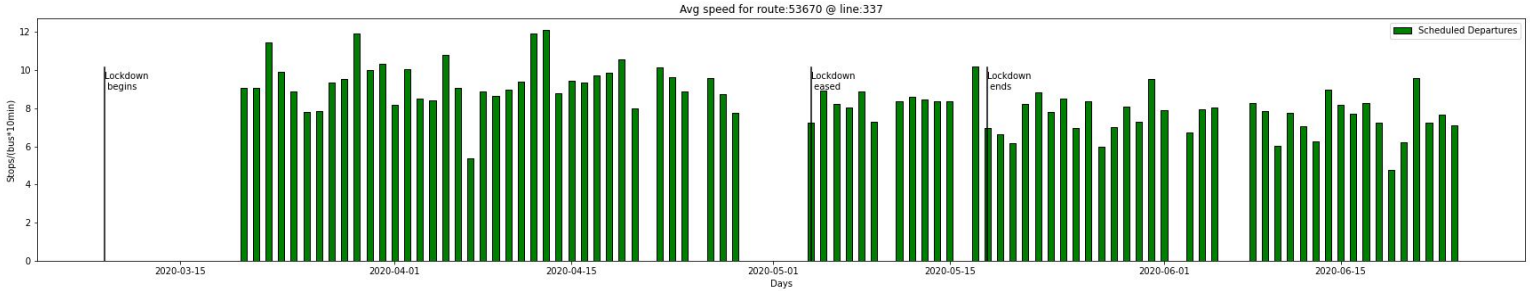
- Avg speed measured in stops
  - Basic Idea: a bus has to perform **all the stops** of a certain line after each departure, so we can basically take each successive stops and measure the average speed in each second (or each 10 minutes)

	line_id	route_id	bus_id	position	time(time)	
1	88	53432	3376	0	07:03:11	s1
2	88	53432	3376	14	07:18:19	s2
3	88	53432	3376	20	07:26:02	s3
4	88	53432	3376	29	07:33:27	...
5	88	53432	3376	42	07:40:47	
6	88	53432	3376	0	08:33:07	
7	88	53432	3376	0	08:40:35	
8	88	53432	3376	0	08:48:00	
9	88	53432	3376	4	08:55:42	
10	88	53432	3376	23	09:10:36	
11	88	53432	3376	31	09:18:16	
12	88	53432	3376	42	09:25:53	
13	88	53432	3376	0	10:26:42	
14	88	53432	3376	0	10:34:12	
15	88	53432	3376	0	10:41:38	
16	88	53432	3376	8	10:49:18	
17	88	53432	3376	16	10:56:48	
18	88	53432	3376	25	11:04:13	

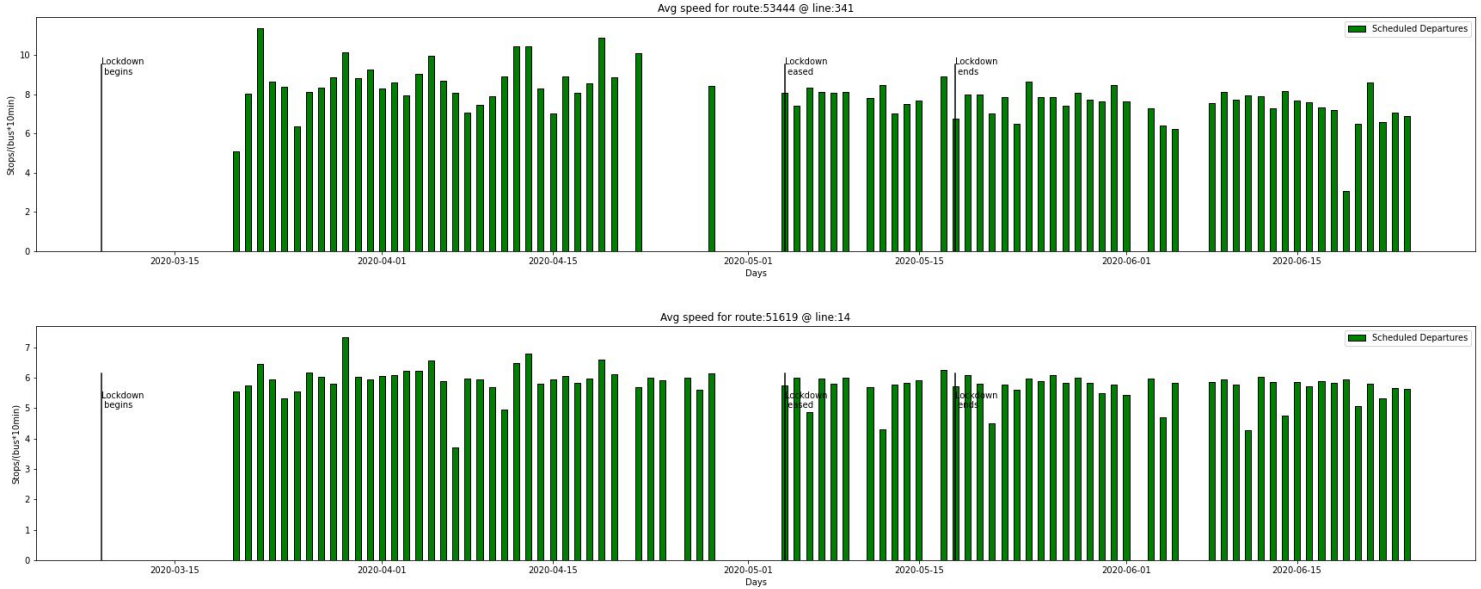
Plots for 2b (Average Speed stops/10min)







Plots for 2b (Average Speed stops/10min)

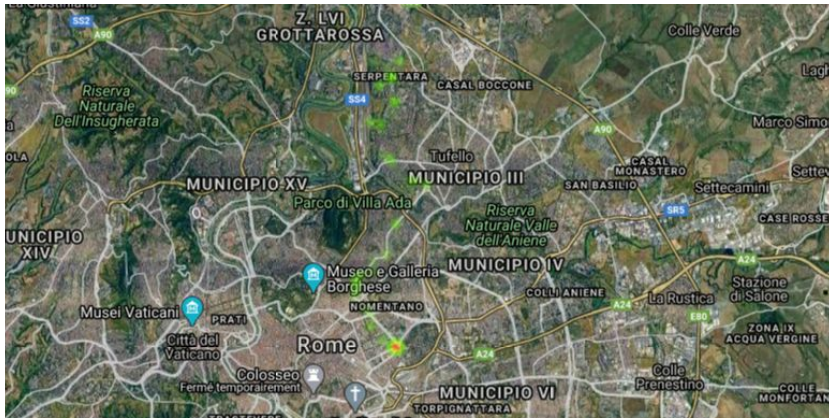


- Observations:
  - Routes can change over time
  - Some of them are shown as unavailable, others as not existing, for other there are just no pages for a certain day
  - Doesn't seem to be meaningful

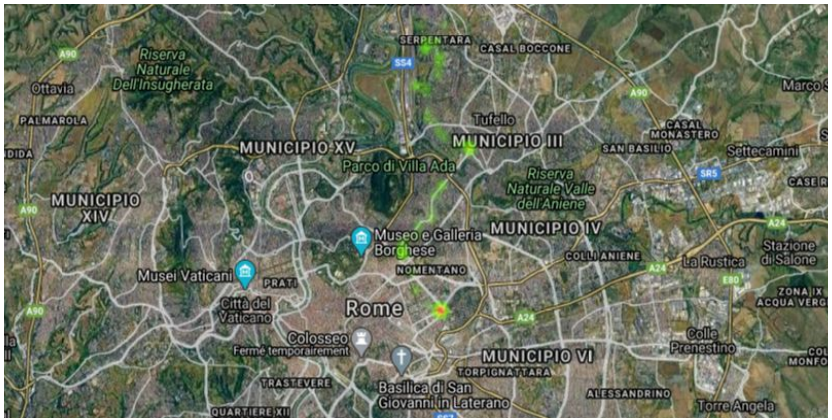
Possible metric 3 (Bus Locations Heatmap)

Heatmap of the route 51619 line 14

During lockdown

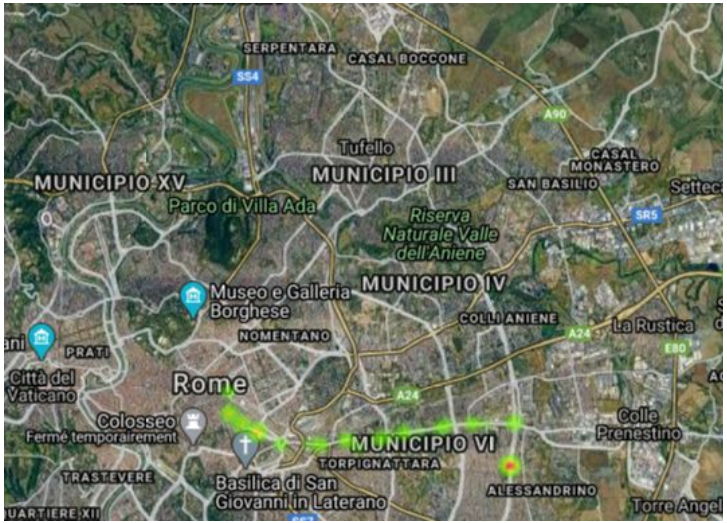


Line 14 after lockdown

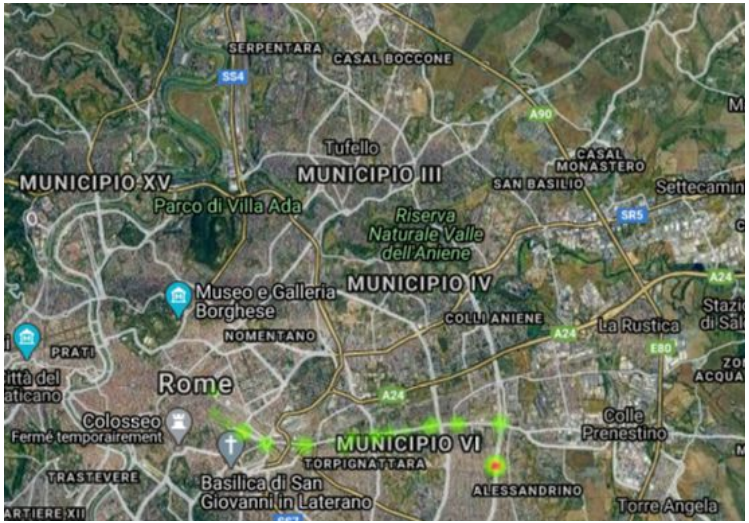


Heatmap of the route 51619 line 14

During Lockdown

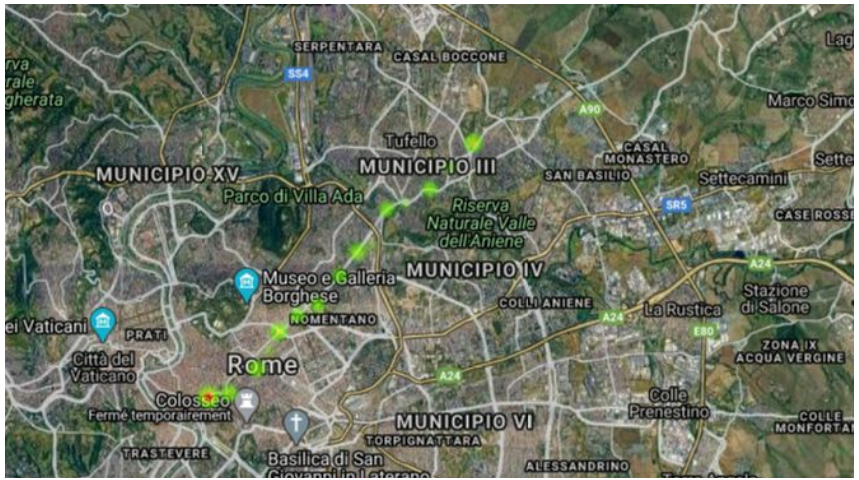


After Lockdown

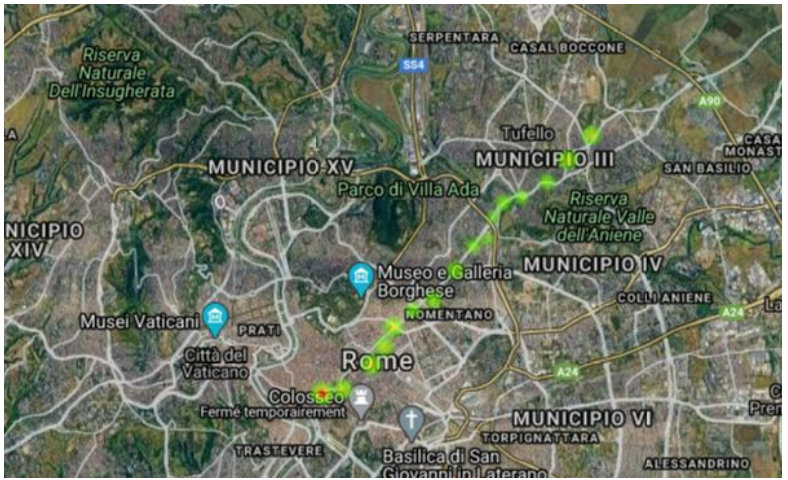


Heatmap of the route 57625 line 60

During Lockdown

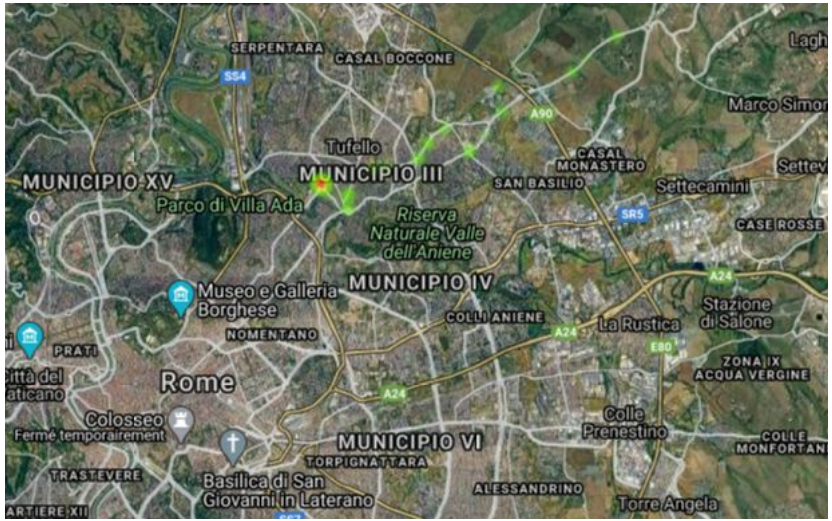


After Lockdown

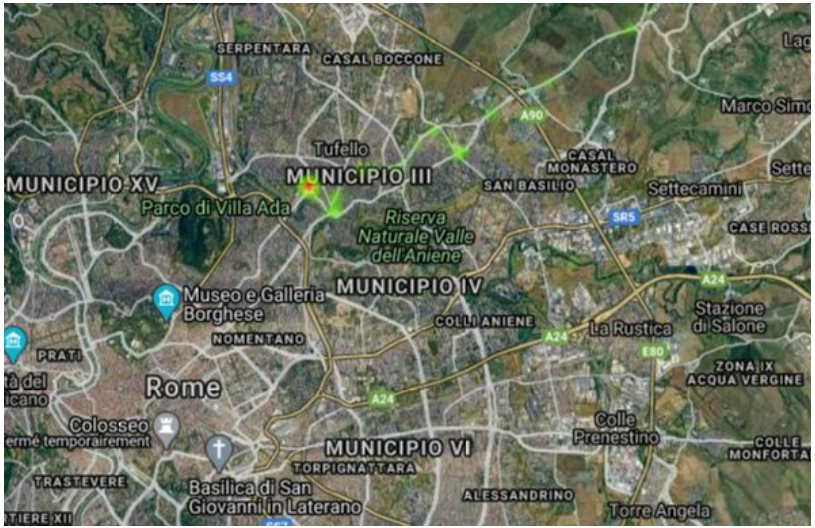


Heatmap of the route 53670 line 337

During Lockdown

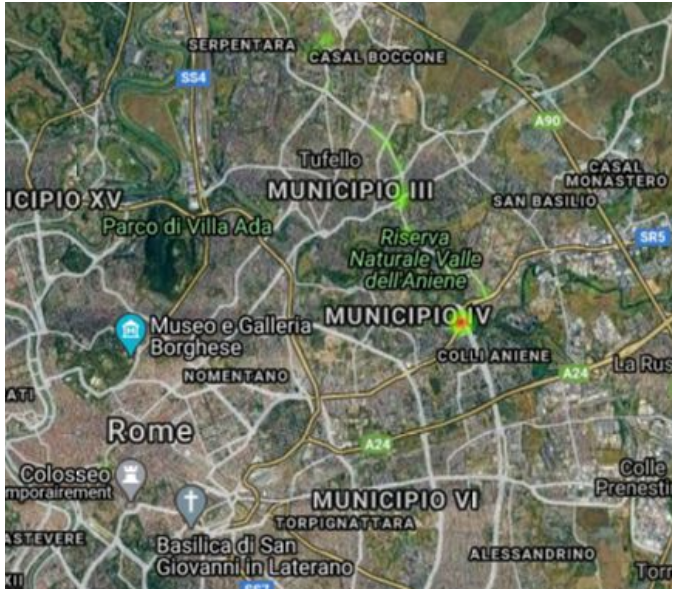


After Lockdown

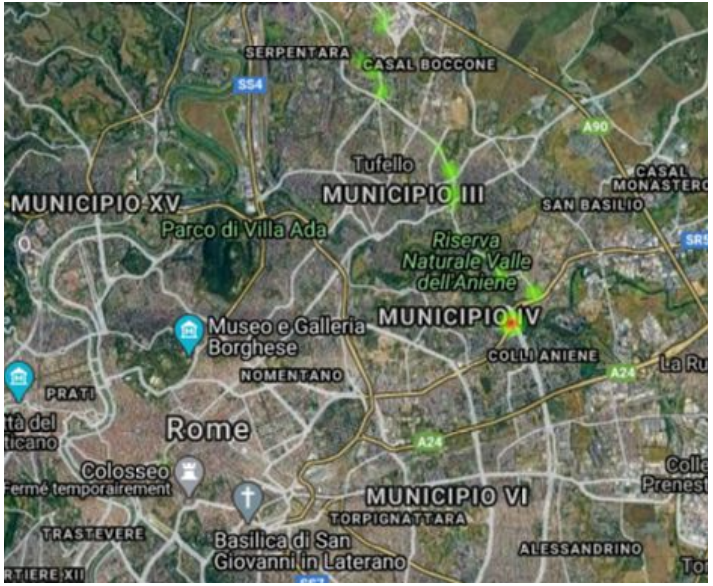


Heatmap of the route 53444 line 341

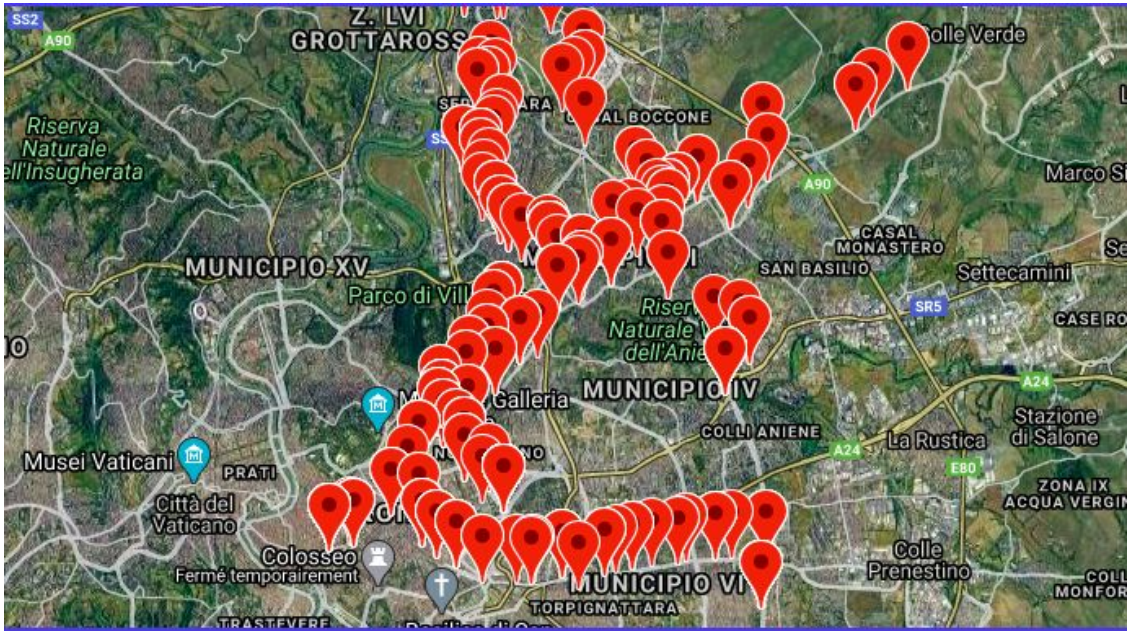
During Lockdown



After Lockdown



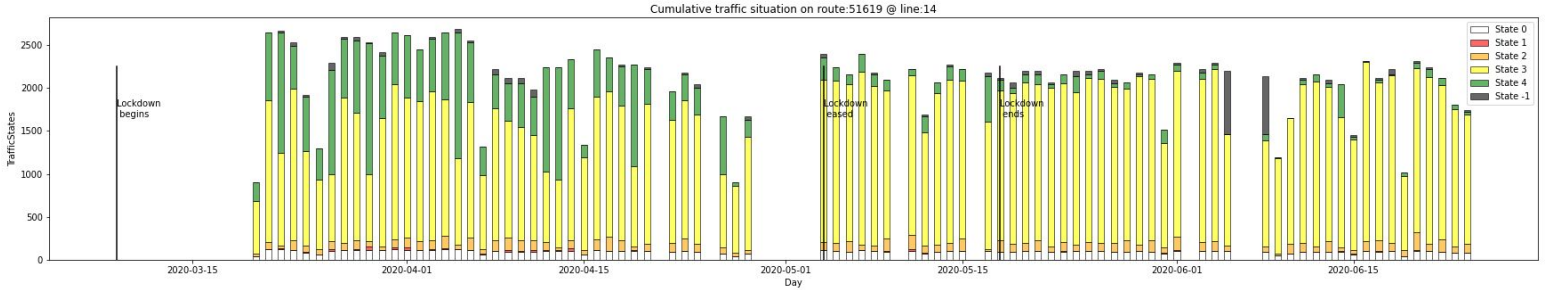
Routestops used to create the heatmaps

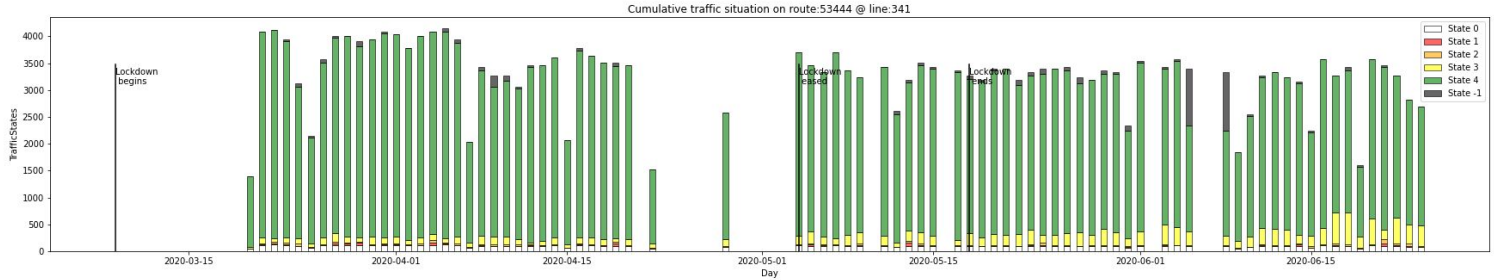


- Basically routes were chosen among the ones that we actually know to validate our findings

Possible metric 4 (Daily cumulative traffic state)

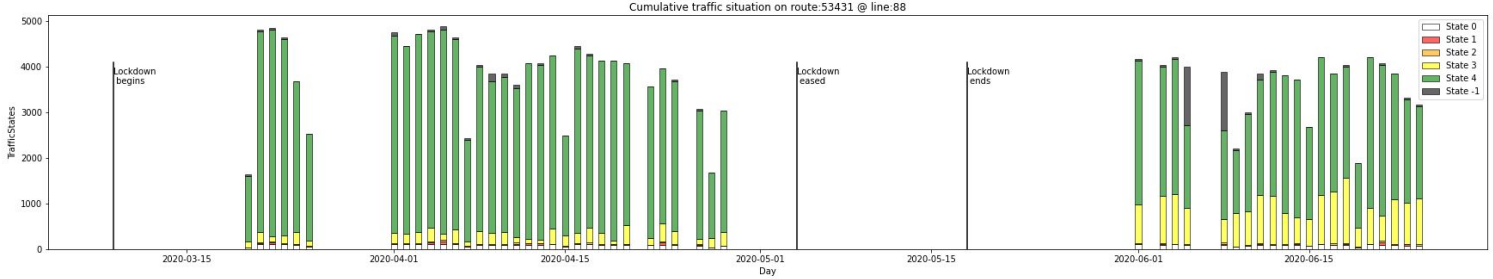
- Idea: using the state indicator from the html pages to retrieve the traffic conditions

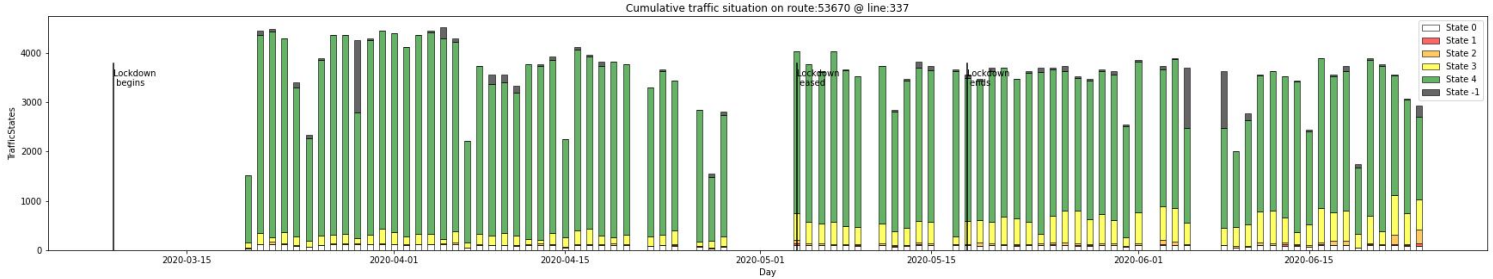




- Observations:
  - This metric clearly shows the impact that lockdown had on traffic, and how much it impacted more the central lines compared to the peripheric ones

Possible metric 4 (Daily cumulative traffic state)

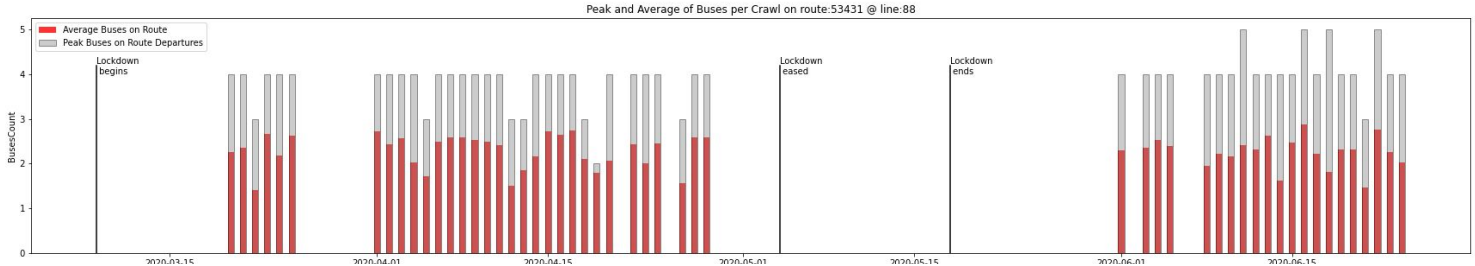


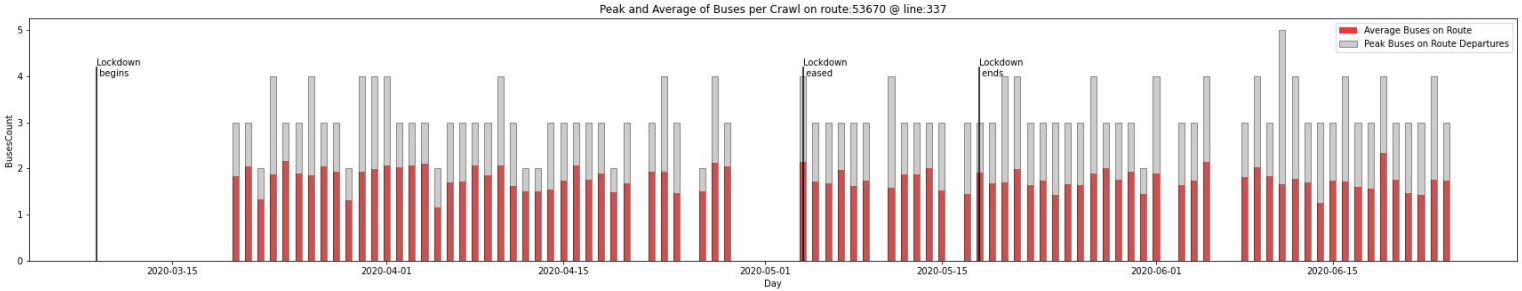


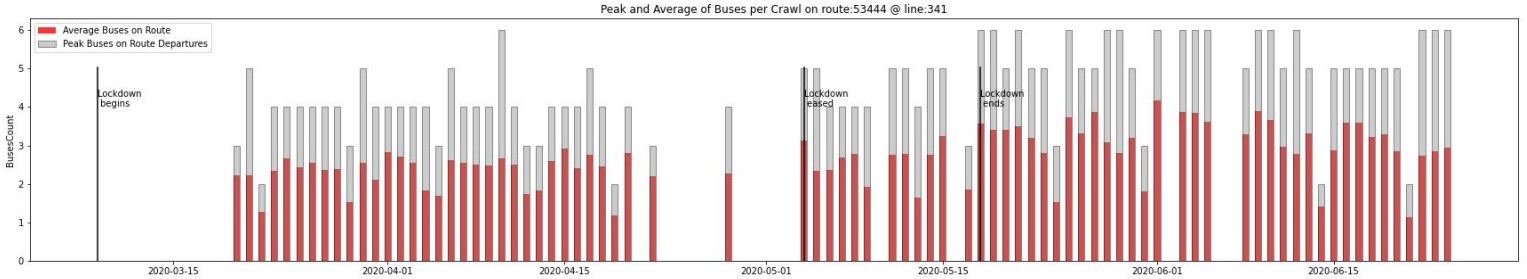
Possible metric 5 (Average number of buses on route)

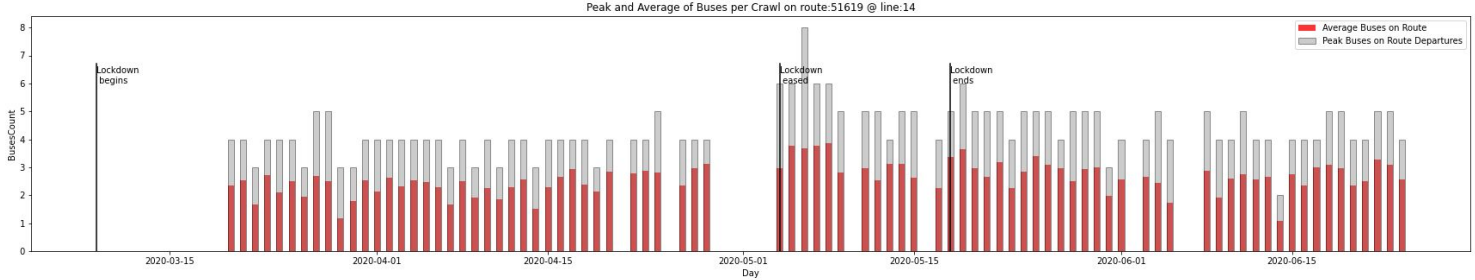
Idea: extract average/maximum number of buses we see each html page to have an esteem of how many buses were assigned to a certain route

$$M_{4,\,day} = \sum_{p \in pages(day)} \frac{|\,buses(p)\,|}{|pages(day)|}$$









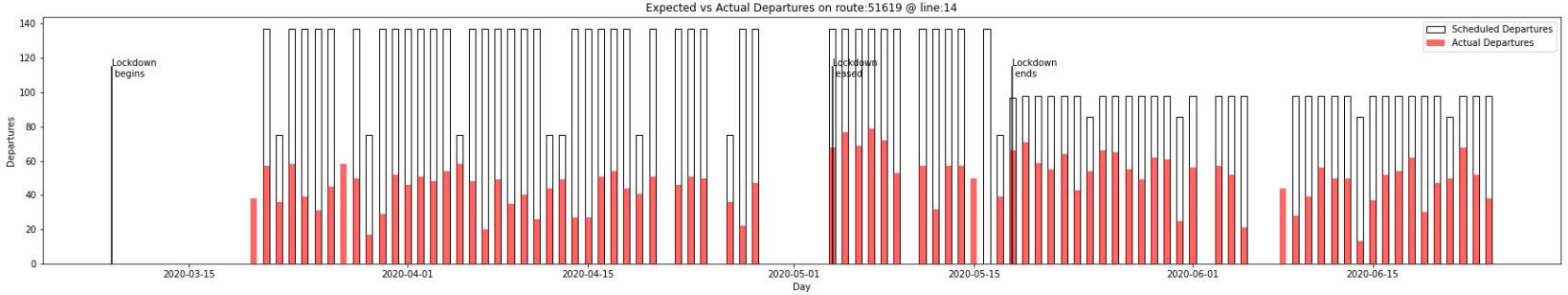
- Observation:
  - The number of buses seen on one line at the same time generally increases after lockdown
- Problem
  - It can due either to the traffic or to an increase in departures, how to find out which one was ?

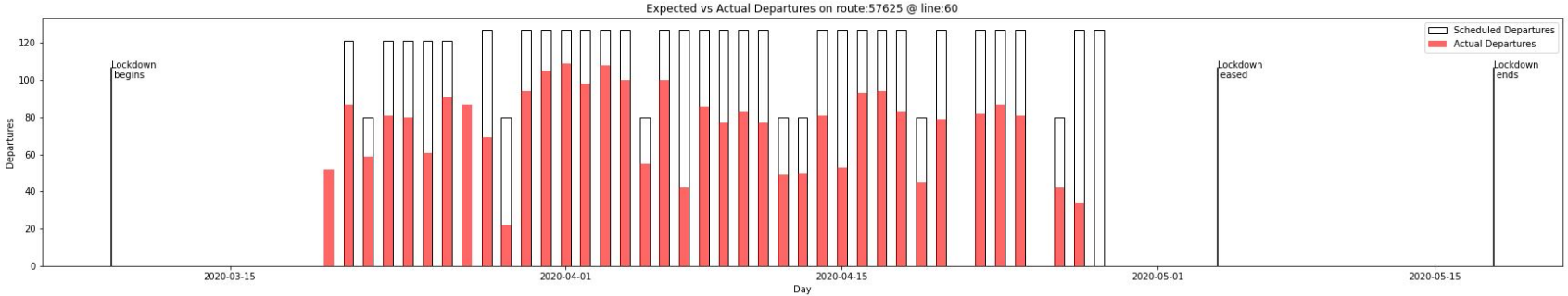
Possible metric 5 (Actual vs Expected Daily Departures)

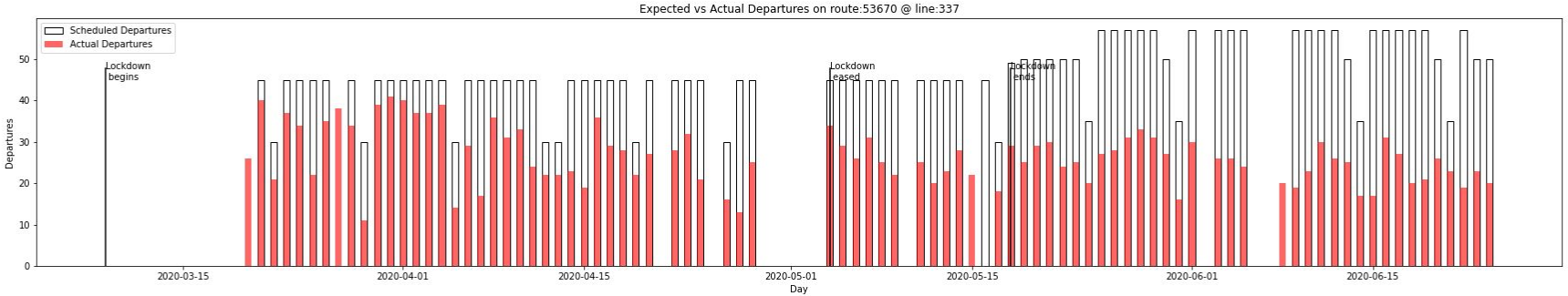
- Does not suffer from granularity problem. (Unless a route is completed in less than 15 minutes)
- Easy to compute (once you figure out)
- Basically we can just scan the ordered list of **positions** and look for “overflows”

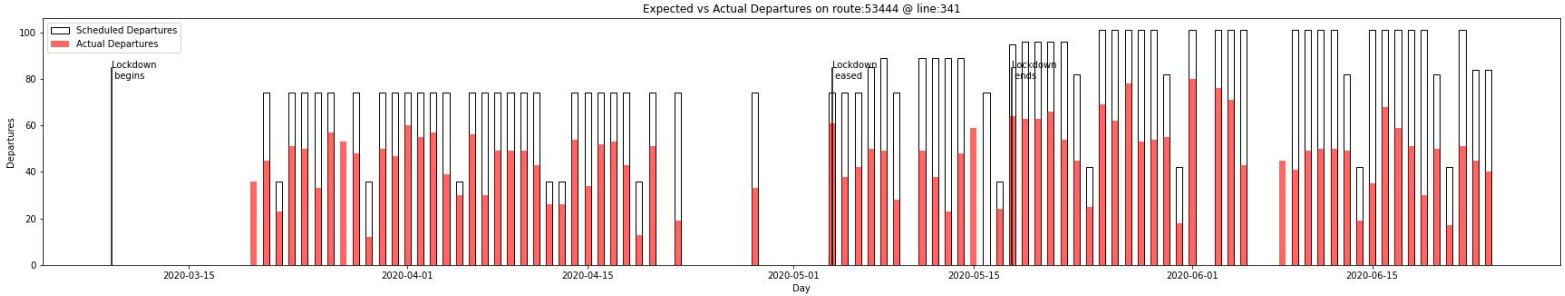
	line_id	route_id	bus_id	position	time(time)
1	88	53432	3376	0	07:03:11
2	88	53432	3376	14	07:18:19
3	88	53432	3376	20	07:26:02
4	88	53432	3376	29	07:33:27
5	88	53432	3376	42	07:40:47
6	88	53432	3376	0	08:33:07
7	88	53432	3376	0	08:40:35
8	88	53432	3376	0	08:48:00
9	88	53432	3376	4	08:55:42
10	88	53432	3376	23	09:10:36
11	88	53432	3376	31	09:18:16
12	88	53432	3376	42	09:25:53
13	88	53432	3376	0	10:26:42
14	88	53432	3376	0	10:34:12
15	88	53432	3376	0	10:41:38
16	88	53432	3376	8	10:49:18
17	88	53432	3376	16	10:56:48
18	88	53432	3376	25	11:04:13

Possible metric 6 (Actual vs Expected Daily Departures)





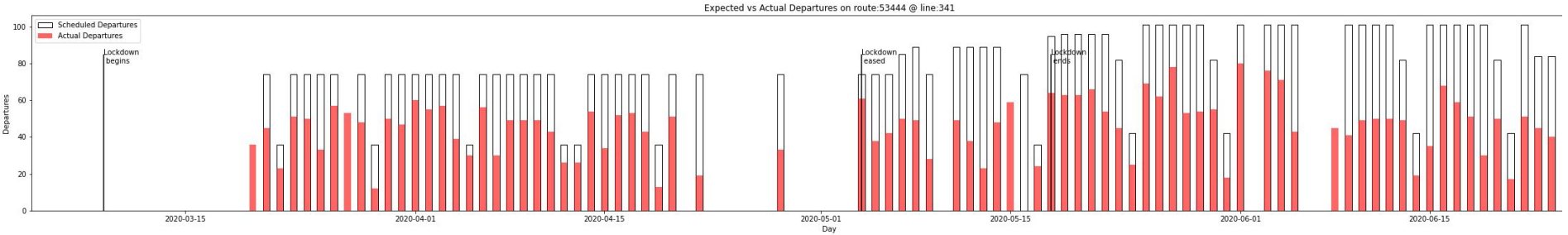




Observations:

- This metric looks extremely interesting !
- Actual and Scheduled departures have significative differences between different routes
- It looks like that promised changes in departures do not reflect real ones

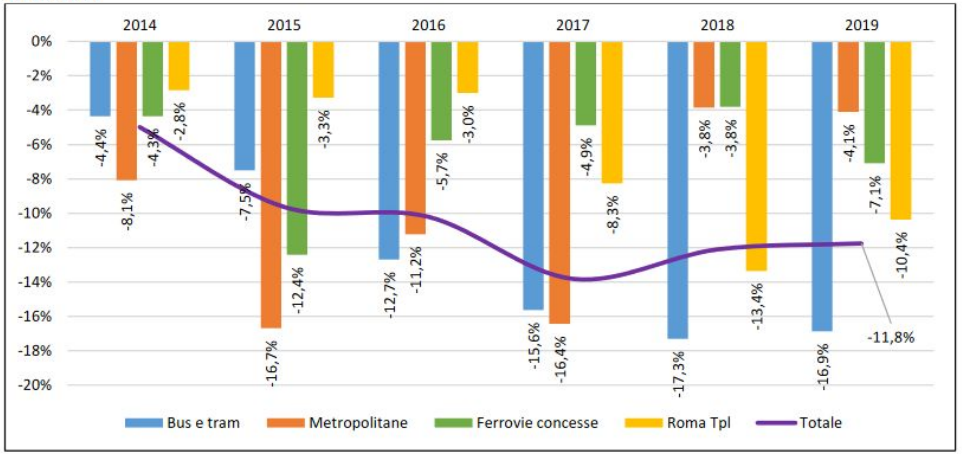
Final considerations



Atac states that they can guarantee more than 80 % of the departures in 2019, but this preliminary analysis show that this numbers may be inflated or that performance worsened this year

Link to reference

Graf. 6 - Differenza tra produzione effettuata e programmata per linea (% di vetture-km). Roma. Anni 2014-2019



Fonte: Elaborazioni Ufficio di Statistica di Roma Capitale su dati RSM / Atac

Further Improvements

- Parse the whole database
- Extract global version of the above metrics
- Modify the scraper to parse the reason of page unavailability
- Investigate more on the difference between scheduled and actual departures

Thanks for the attention