Data Mining
STIB Network Reliability Analysis

Yang Liu 000473773
You Xu 000558502
Tianheng Zhou 000559189

Content

- Methods
  - Identify Peak-Hours
  - Calculate Punctuality
  - Calculate Regularity
- Data Manipulation
- Results and Performance
- Analysis and assumptions
- Limitations
- Interactive Dashboard

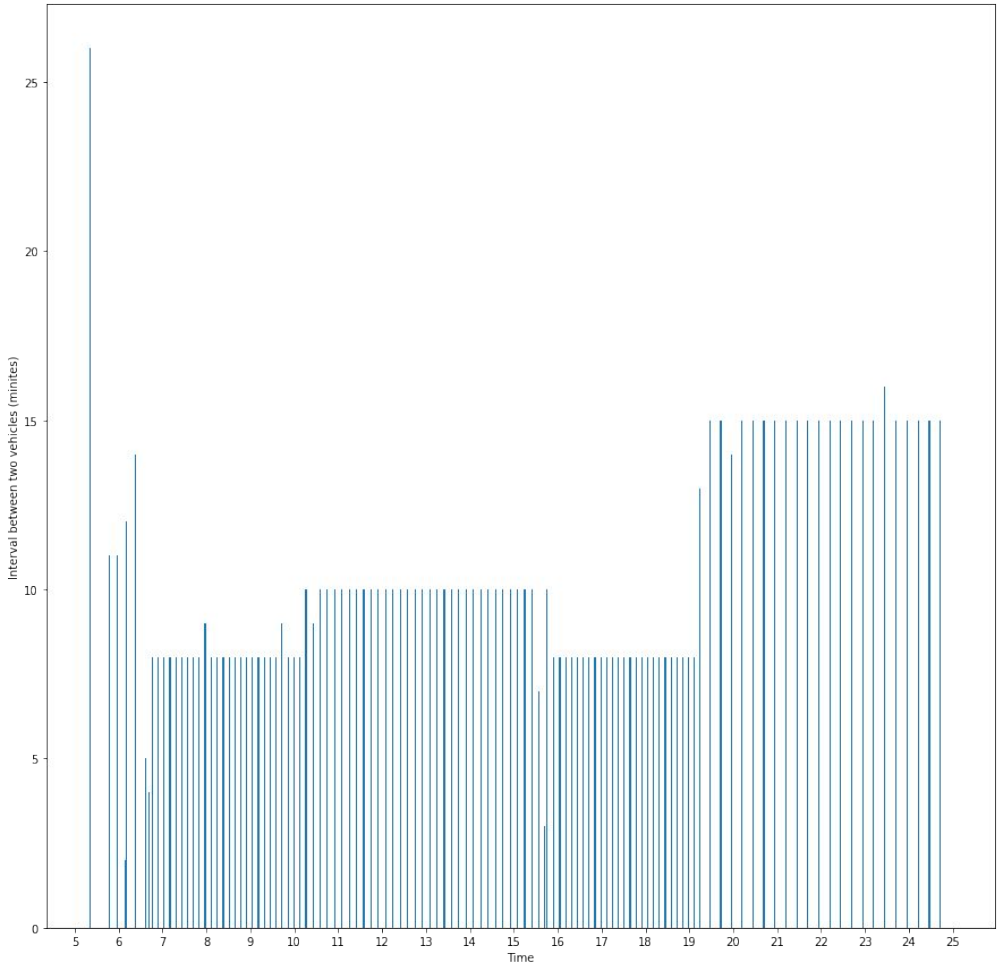
Identify Peak Hours

Step 1: Find clusters

Step 2: Assign them into punctual/regular zone

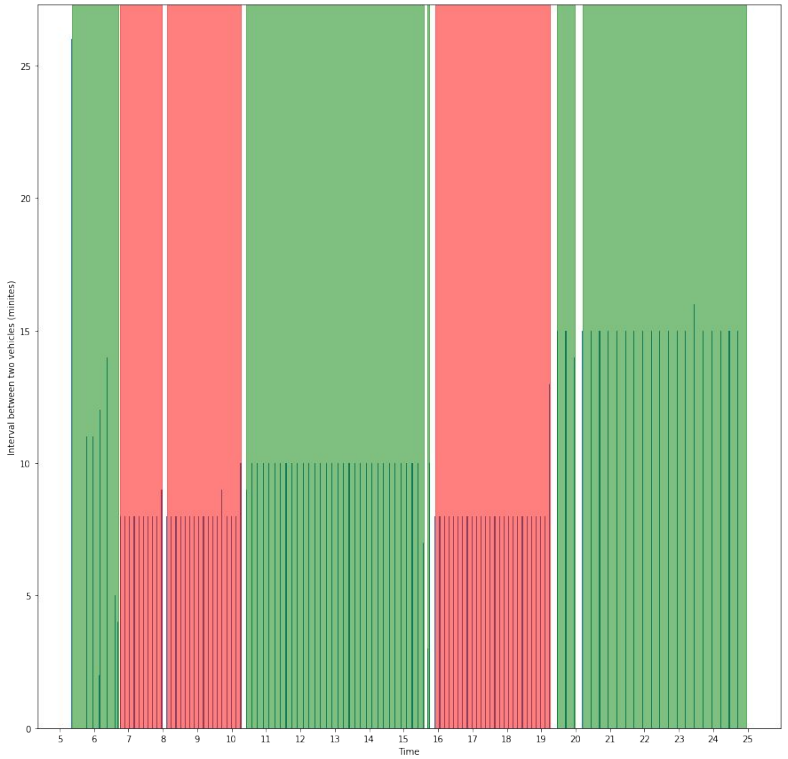
Step 1: Find clusters

1. Try to cluster 4 elements in each time.
2. If the **cluster's standard deviation** > 0.3 : go forward one element
3. When the cluster's standard deviation <=0.3 : try to add more elements into the cluster while remaining the standard deviation <=0.3
4. In the end, cluster all the single elements together.



Step 2: Assign them into punctual/regular zone

Threshold = Median



QoS: Punctuality

Date changing time: **4 am**

For a specific selection **line, direction, date, stop:**

- Get schedule list from schedule **files join and selection**
- Select **schedule time** within peak hours
- Select **actual arrival time** within peak hours
- Calculate **on-time rate** for this line with this direction on this date at this stop

Exceptions: if no off-peak hours, leave on-time rate as empty.

Actual Arrival Time

Time	Distance From Point
10:00:30	0
10:01:00	0
10:01:30	36
10:02:00	350
10:05:30	50
10:06:00	245

Step 1: Select rows with <= 200 distance

Step 2: Delete duplicate cars

- Detect if two neighbor rows have time different within 15 - 45s
- If so, keep only the first row

Special case: two cars following each other

- Do step 2 twice

Actual Arrival Time

Time	Distance From Point
10:00:30	0
10:01:00	0
10:01:30	36
~~10:02:00~~	~~350~~
10:05:30	50
~~10:06:00~~	~~245~~

Step 1: Select rows with <= 200 distance

Step 2: Delete duplicate cars

- Detect if two neighbor rows have time different within 15 - 45s
- If so, keep only the first row

Special case: two cars following each other

- Do step 2 twice

Actual Arrival Time

Time	Distance From Point
10:00:30	0
~~10:01:00~~	~~0~~
~~10:01:30~~	~~36~~
~~10:02:00~~	~~350~~
10:05:30	50
~~10:06:00~~	~~245~~

Step 1: Select rows with <= 200 distance

Step 2: Delete duplicate cars

- Detect if two neighbor rows have time different within 15 - 45s
- If so, keep only the first row

Special case: two cars following each other

- Do step 2 twice

Actual Arrival Time

Time	Distance From Point
10:01:00	0
10:01:00	23
~~10:01:30~~	~~45~~
10:01:30	99
~~10:02:30~~	~~68~~
10:02:30	150

Step 1: Select rows with <= 200 distance

Step 2: Delete duplicate cars

- Detect if two neighbor rows have time different within 15 - 45s
- If so, keep only the first row

Special case: two cars following each other

- Do step 2 twice

Actual Arrival Time

Time	Distance From Point
10:01:00	0
10:01:00	23
~~10:01:30~~	~~45~~
~~10:01:30~~	~~99~~
~~10:02:30~~	~~68~~
~~10:02:30~~	~~150~~

Step 1: Select rows with <= 200 distance

Step 2: Delete duplicate cars

- Detect if two neighbor rows have time different within 15 - 45s
- If so, keep only the first row

Special case: two cars following each other

- Do step 2 twice

Actual Arrival Time - Adjustment

	Time	Distance From Point
	10:01:00	0
10:00:45	10:01:00	23

If distance is 0: keep original time

If distance >0: Time - 15s

Inaccuracy Range: +/- 15s

Loss track of some cars is possible

On-Time Rate

Schedule	Actual
	10:28:00
10:30:00	10:29:05
	10:30:50
	10:32:00

On-time: car arrive within +/- 1 min of schedule time

On-time rate: on-time count / schedule count
Range: [0,1]

#Actual cars / # Schedule cars
Range: >0
Regardless of peak/off-peak hour

QoS: Regularity

Schedule Waiting Time (SWT):

In Schedule timeline, For each regular zone, for each vehicle in this zone:

Step1: Calculate the time interval between two neighbored vehicles (There is always at least one value.)

Step2: Use the following formula to calculate the Schedule Waiting Time for this regular zone

$$SWT = \frac{\sum Scheduled\ headways^2}{2 \times \sum Scheduled\ headways}$$

QoS: Regularity

Actual Waiting Time (AWT):

In Actual timeline, for each regular zone:

$$AWT = \frac{\sum Actual\ headways^2}{2 \times \sum Actual\ headways}$$

Step1: Find vehicles that are inside this regular zone.

Step2:

- If no vehicles -> length of this zone as one interval value (A reasonable estimation).
- If one vehicle -> the interval between this vehicle and the first vehicle in the next punctuality zone as one interval value.
- If more than one vehicle -> calculate intervals as normal.

Step3: Use the formula to calculate the Actual Waiting Time for this regular zone.

QoS: Regularity

Excess Waiting Time (EWT):

For each regular zone:

Step1: Find the correspond SWT value and AWT value in this zone.

Step2: Use the following formula to calculate the EWT.

$$EWT = AWT - SWT$$

$$Weighted\ Excess\ Waiting\ Time = \frac{\sum Excess\ Waiting\ Time\ in\ this\ regular\ zone \times Number\ of\ Scheduled\ cars\ in\ this\ regular\ zone}{\sum Number\ of\ Scheduled\ cars\ in\ each\ regular\ zone}$$

Data Manipulation Overview

Transform vehiclePosition.json → Join schedule tables → Get Peak/Off-Peak Hours

Columns:

- Time
- LineID
- DirectionID
- DistanceFromPoint
- PointID

19328710 rows for combined 13 JSON files (full data cube), using JSON_to_CSV.ipynb in our Github repo.

Tables:

- Trips
- Calendar
- Stop_times

For calendar: Split date range to corresponding rows with specific date

Combination:

- Route
- Direction
- Date
- Stop

Only one schedule car: skip

Off-peak hour example:
[[0, 5], [60, 90]]

Total rows: 260,864

Data Manipulation Overview

Calculate Punctuality/Regularity → Join results → Analysis and Visualization

Select date within range: [20210908, 20210920]

If no actual car on the whole day: skip

Lines with “T” and “N”: skip due to lack of data

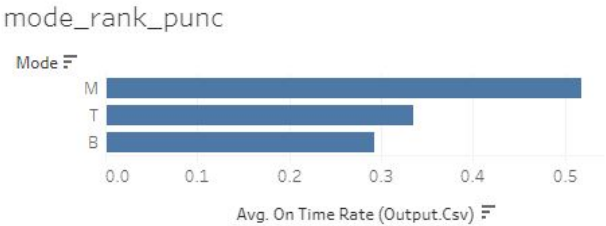
Total rows: 44,528

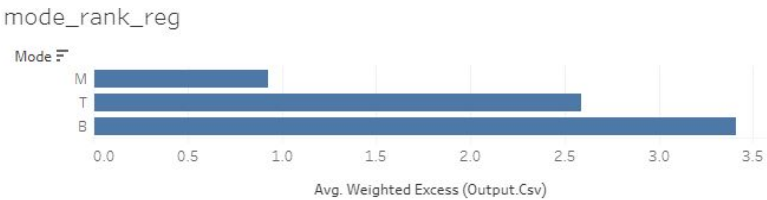
Translate route_id to actual name of the line

Join two map tables with the result

Performance Overview

- Means of transportation: Metro > Tram > Bus





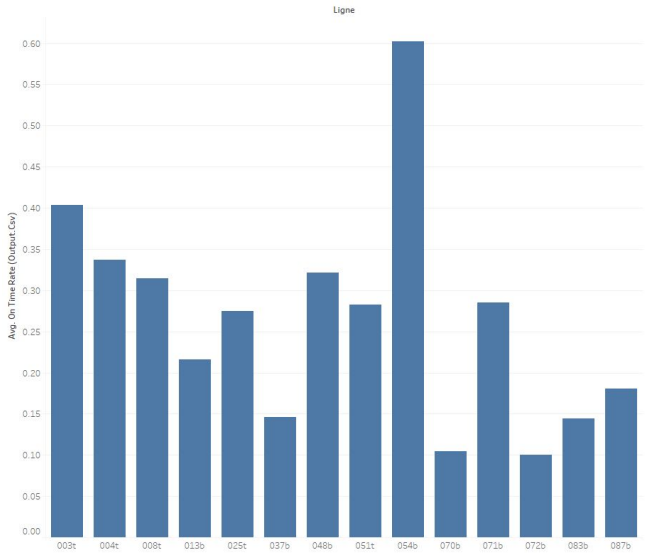
Analysis On-time rate / Excess waiting time

- 1. Point :

  Compare all lines on a given stop, if all bad, then the stop is a problematic stop. If certains lines are much worse than others, analyse these lines.

- 2. Line:

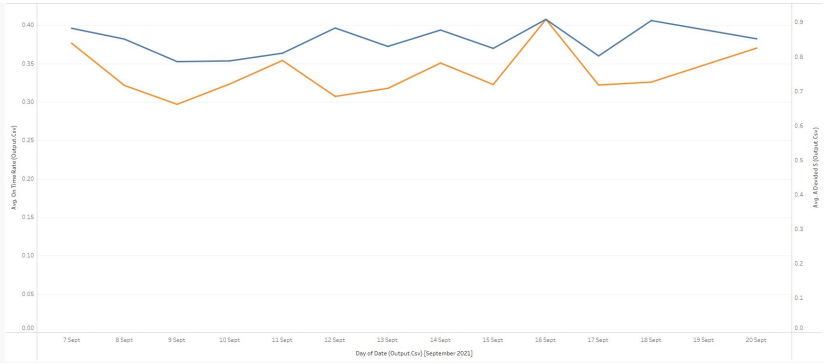
  See next slides



For stop “Albert”, on-time-rate of all lines which pass through this stop are plotted

Assumption 1: Due to the integrity (missing vehicles)

Integrity = Actual # of vehicles / Scheduled # of vehicles



Variation of punctuality and regularity comparing to the integrity of the line on different dates.

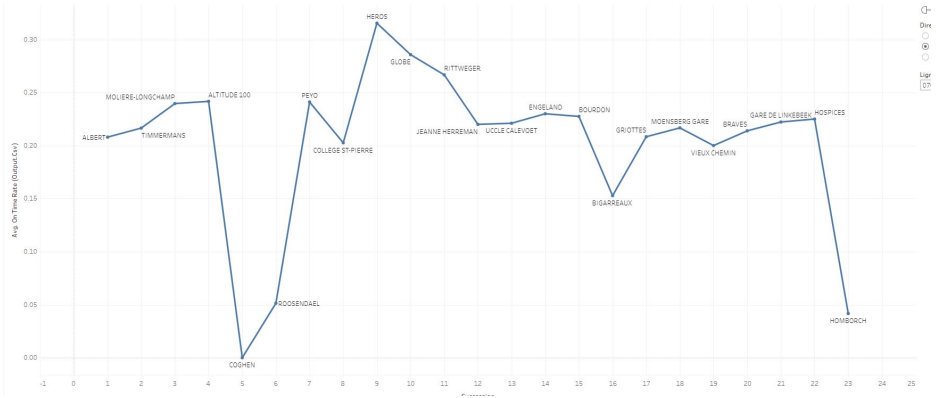
Orange: punctuality of line 7

Blue: integrity of line 7

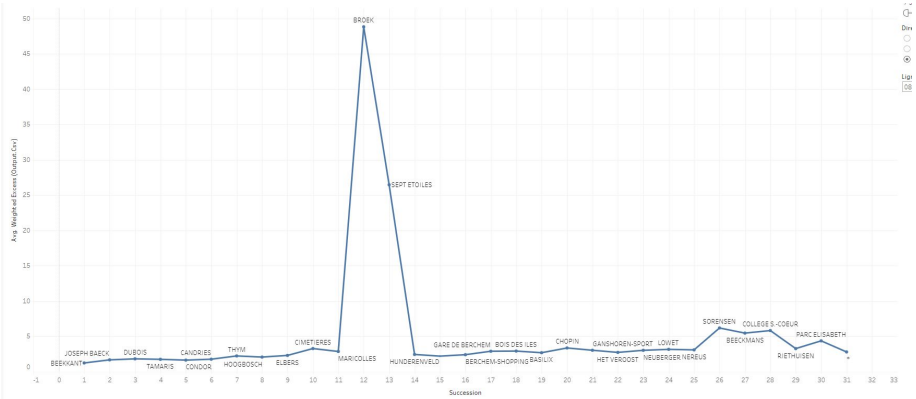
Red: regularity of line 61

Blue: integrity of line 61

Assumption 2: Due to a particular stop (maintenance)

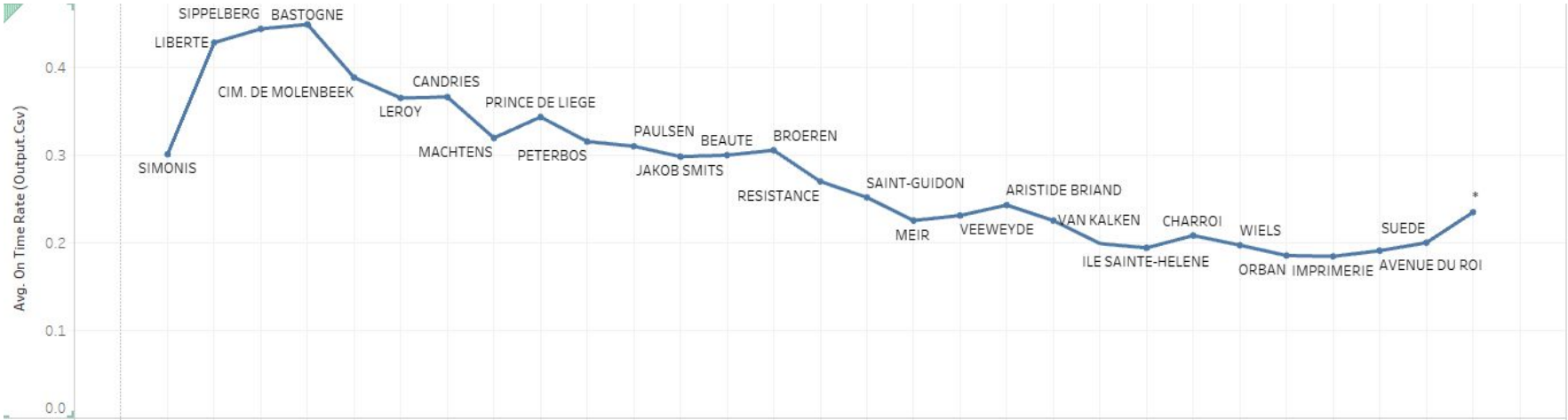


Avg on-time-rate of stops on line 70



Avg excess-waiting-time of stops on line 87

Assumption 3: Due to a difficult segment (propagation of congestion)



Avg on-time-rate of stops on line 49

QoS: Reliability

Reliability of stops or lines:

Step1: Find the rank of its punctuality (on time rate) in **descending** order.

Step2: Find the rank of its regularity (weighted excess waiting time) in **ascending** order.

Step3: Plus these two ranks as the reliability (E.g Brussels Airport Stop is top 4 in punctuality, top 7 in regularity, then 11 is its reliability value). The lower, the better.

Advantage: Avoid deciding weight for punctuality and regularity, as they have different scales in value and distribution.

Limitation

Completeness:

- Some date not consider due to lack of data
- Some lines not consider due to unclear data correspondence
- Peak hour zones not analyzed separately

Accuracy:

- Outliers: no car/substantially less cars, might due to stop closure, accidents, strike etc.
- Inaccurate division of peak/off-peak hours
- Some actual cars are omitted due to imperfect data manipulation, resulting in under-estimated on-time rate
- Estimation of actual waiting time when no car/only one car presents is not perfect, resulting in over-estimated excess waiting time

Interactive dashboard (For maps, please Use the Full Screen mode to watch)

Punctuality Map and Analysis: https://public.tableau.com/views/12170001/Dashboard1

Regularity Map and Analysis: https://public.tableau.com/views/12170002/Dashboard2

Reliability Map and Analysis: https://public.tableau.com/views/12170003/Dashboard3

Assumption1:https://public.tableau.com/app/profile/yangliu4035/viz/12170004/assumption1Ontimerateisduetothemissingbuses

Assumption2/3:
https://public.tableau.com/app/profile/yangliu4035/viz/12170005/Tofindfromwhichstoptheontimerateisaffected

Analysis on differents lines for a given stop:
https://public.tableau.com/app/profile/yangliu4035/viz/12170006/Foragivenstopplotdifferentlines

GitHub Repository

https://github.com/xuyou1999/Data_Mining_F22