STAR Analysis

Tomáš Omasta, Lukáš Frána

13th April 2022

Introduction

We are students at the FEE of the CTU in Prague. Currently, we attend an Open informatics study program with a specialization in Data Science. This document is a part of semestral work for the Visualization course. It describes the STAR analysis performed on open data published by Prague public transport.

Table of content

Dataset

We use Prague public transport data downloaded from its Open Data website. The time window of presented data spans from 30th March 2022 to 9th April 2022. The dataset contains various files (mostly CSV with a colon as a separator) on which the visualization itself will be performed.

agency.txt

Contains a table of agencies. For each row, there is **agency_id** (unique integer ID), **agency_name** (string), **agency_url** (string) to the agency's website, **timezone** (string) in which the agency's timetables are described, its **language** (string), and telephone number **agency_phone** (string).

calendar_dates.txt

Columns are **service_id** (integer ID of related service), **date** (in format YYYYMMDD), and **exception_type** (categorical integer).

calendar.txt

Each row stores time schedule of a service and contains **service_id** (unique integer ID), **monday**, **tuesday**, **wednesday**, **thursday**, **friday**, **saturday**, **sunday** (binary value represents if service is operated on that specific day, 1 - yes, 0 - no), **start_date** (in format YYYYMMDD), **end_date** (in format YYYYMMDD).

feed_info.txt

Standardized info about publisher of .gtfs time tables **feed_publisher_name** (string), **feed_publisher_url** (string), **feed_lang** (string), **feed_start_date** (in format YYYYMMDD), **feed_end_date** (in format YYYYMMDD), **feed_version** (integer).

JR_GTFS_Exp.log

This is the only file that does not hold CSV (or any kind of tabular) data. It is a simple log of unexpected events.

routes.txt

Contains **route_id** (unique string ID), **agency_id** (integer ID of transporting agency), **route_short_name** (unique integer indicating route), **route_long_name** (full string name of

the starting and the ending station), **route_type** (categorical integer), **route_color** (categorical integer, may be empty), **route_text_color** (string), **route_url** (URL of the route on Prague public transportation website).

shapes.txt

File contains shapes of public transport lines with **shape_id** (is similar for multiple rows as each row describes one GPS location, foreign key, integer ID), **shape_pt_lat** (GPS latitude, float), **shape_pt_lon** (GPS latitude, float), **shape_pt_sequence** (position of GPS point in the current shape sequence).

stop_times.txt

Columns in the file are **trip_id** (grouping integer, one specific trip a day), **arrival_time** (in format h:mm:ss), **departure_time** (in format h:mm:ss), **stop_id** (string ID of the stop), **stop_sequence** (integer indicating order of the stops), **pickup_type** (integer), **drop_off_type** (integer), **stop_headsign** (string, may be empty).

stops.txt

Describes individual stop stations. Contains **stop_id** (integer), **stop_name** (string), **stop_lat** (GPS latitude, float), **stop_lon** (GPS latitude, float), **location_type** (int), **parent_station** (parent station holds multiple stop stations - for example to multiple directions, string), **wheelchair_boarding** (binary), **stop_desc** (int), **stop_url** (URL of the stop station on Prague public transportation website with time table, string).

trips.txt

Lists all trips as a whole with **route_id** (specifying which route that trip was operating, string), **service_id** (according to which time-schedule trip was operated, integer), **trip_id** (number of trips done for a specific route, integer), **trip_headsign** (terminal station written on vehicle's display, string), **shape_id** (through which points vehicle drove, integer), **wheelchair_accessible** (binary), **block_id** (integer), **direction_id** (binary), **bikes_allowed** (binary), **exceptional** (integer).

Motivation

Having the dataset described we set out to describe the motivation of the final project.

We aim to deliver a reactive dashboard focused on Prague's public transportation system. Such a dashboard will suit as an intuitive way to sell a convincing story to users. With the help of this dashboard, we answer questions that may arise with the visualizations. For visualization, we are going to use D3.js framework

D3.js library has been chosen to help us visualize answers to some of these questions. It is a data-driven library that is widely used in web browsers for graphs. It has proven to be a solid basis for this task. The implementation will take shape as a website dashboard and run entirely from the browser.

First of all, we present questions that may or may not be answered in the dashboard. Then we research visualization techniques that may be of use for visualizing the answer for these questions. In the end we discuss our choice of visualization techniques, used mappings, transformations etc.

Be advised, amongst our main goals is to visualize overused lines and stations (business analytics use case), thus we explicitly do **not** use geographical positions.

1. What can we tell about types of transportation?
2. Is the ratio of transportation vehicles the same throughout the day?
3. How many lines exist in our dataset?
4. What are some extremes in the data? (the longest/shortest route, the most frequent line, the most used agency)
5. What options has a person tied to a wheelchair?
6. What is the ratio of the wheelchair-friendly stop stations and vehicles?
7. Is there any repeating behavior in time series data?
8. How frequent are unexpected events?
9. Do unexpected events correlate with traffic peaks?
10. Which of the lines are most used?
11. Is there a correlation between working hours and the frequency of lines?
12. How to display data of one particular stop station? Is there a pattern between weekdays and weekends transportation peaks?
13. How are the stop stations connected to each other?
14. When is public transport used the most?
15. Which stop stations are most critical for functioning public transport?

As our dataset contains several thousands stations which implicitly introduces problems with overcrowded visualizations, aggregations, performance issues, etc., we decided to limit our research only to a selected subset of them.

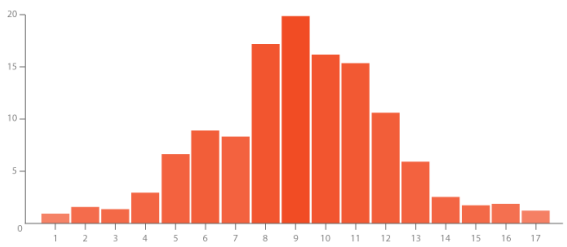
In addition to this, the dataset itself has over 100 MB, so it is not recommended to load into the browser's memory. The solution to this problem is to aggregate the data with preprocessing. It will greatly reduce the overall size and speedup both visualization and interaction.

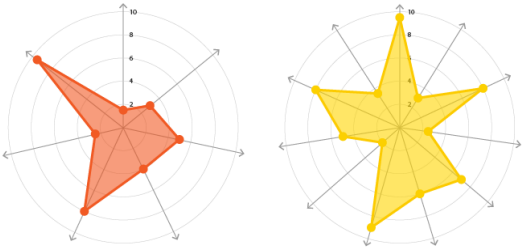
Research of possible visualizations

To enhance our understanding of possible approaches we list multiple visualization methodologies below. Each of them contains a simple description in line with images corresponding to specific strategies it represents. Based on the following research, we set to choose visualization techniques which we use for answering some of previously listed questions in the final project.

Comparisons (bar chart, radar chart)

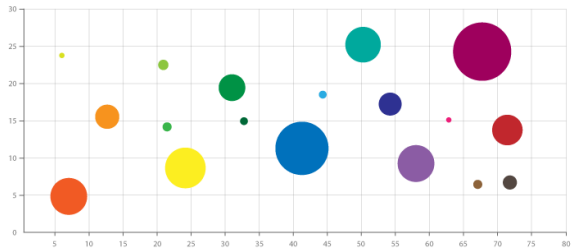
Visualization methods that help show the differences or similarities between values.





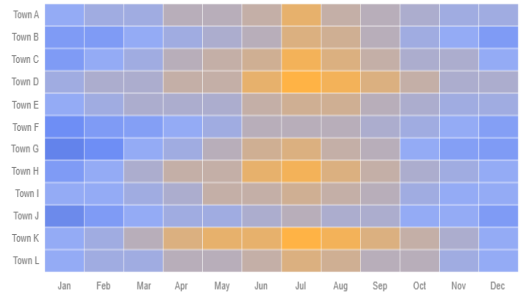
Proportions (bubble chart, treemap)

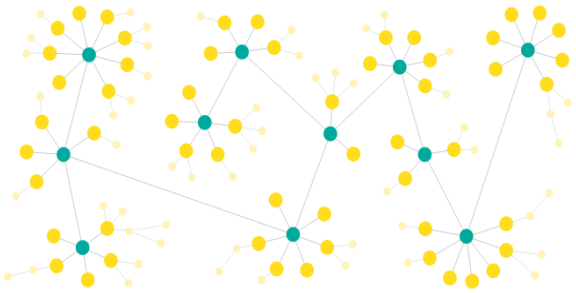
Visualization methods that use size or area to show differences or similarities between values or for parts to a whole.



Relationships (heatmap, network diagram)

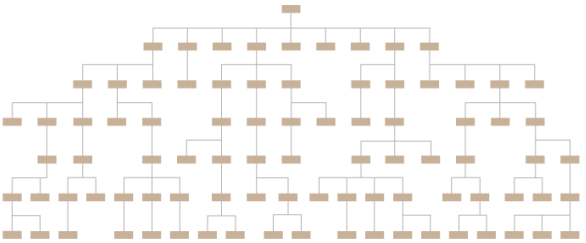
Relationships: Visualization methods that show relationships and connections between the data or show correlations between two or more variables.





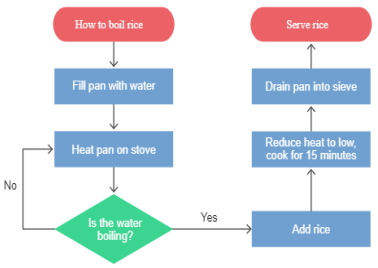
Hierarchy (sunburst diagram, tree diagram)

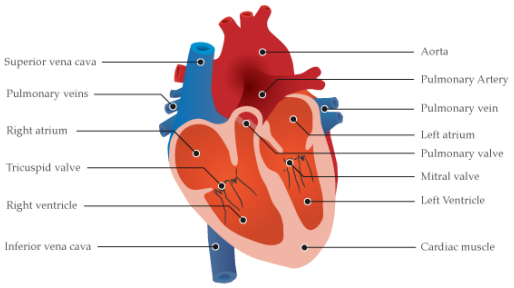
Visualization methods that show how data or objects are ranked and ordered together in an organization or system.



Concepts (flow chart, illustration diagram)

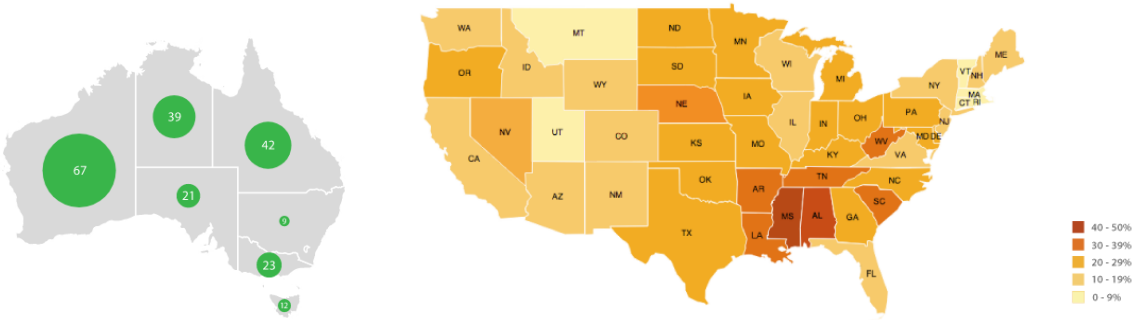
Visualisation methods that help explain and show ideas or concepts.





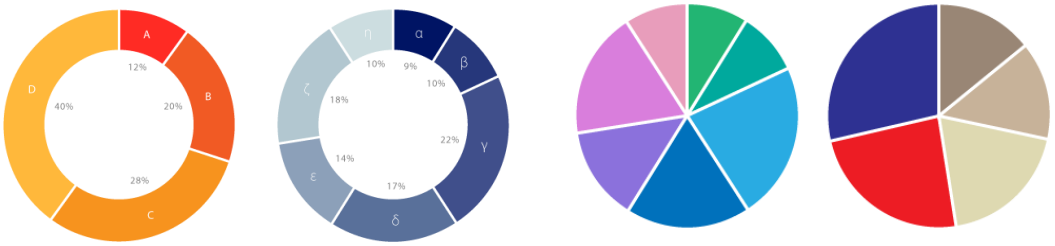
Location (bubble map, choropleth map)

Visualization methods that show data over geographical regions.



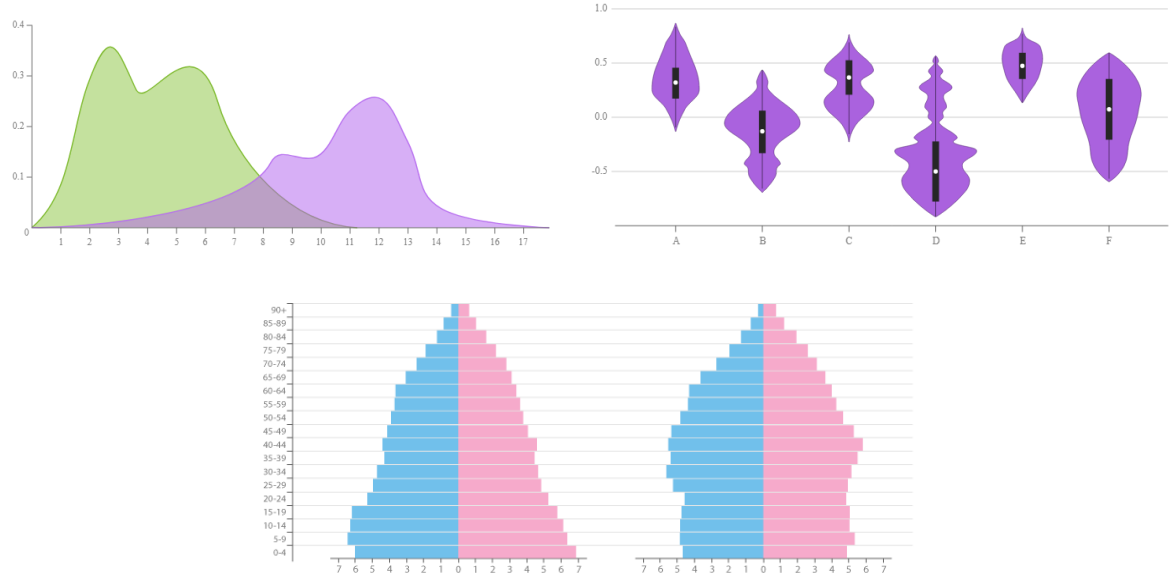
Part-to-a-whole (donut chart, pie chart)

Visualization methods that show part (or parts) of a variable to its total. Often used to show how something is divided up.



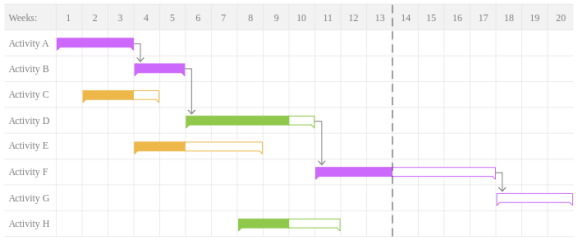
Distribution (density plot, violin plot, pyramid plot)

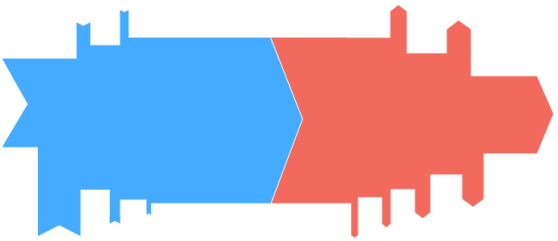
Visualization methods that display frequency, how data spread out over an interval or is grouped.



Processes & methods (gantt chart, sankey diagram)

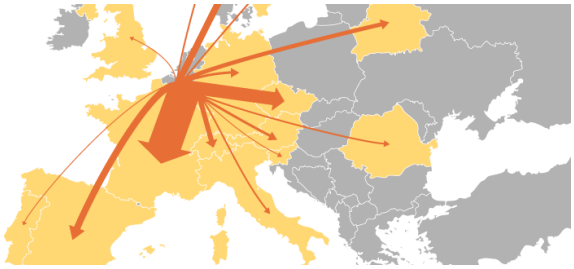
Visualization methods that help explain processes or methods.

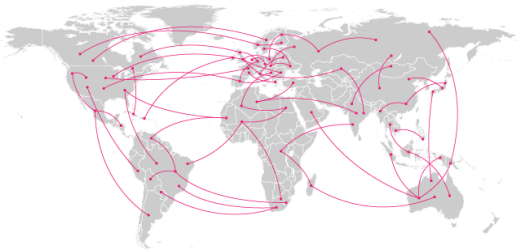




Movement or flow (flow map, connection map)

Visualization methods that are useful for showing movement data or the flow of data.



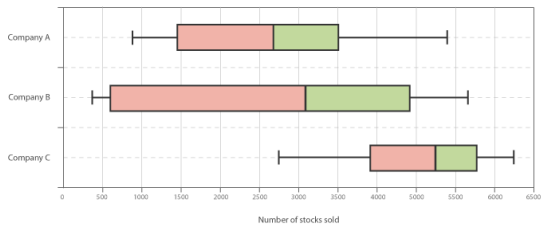


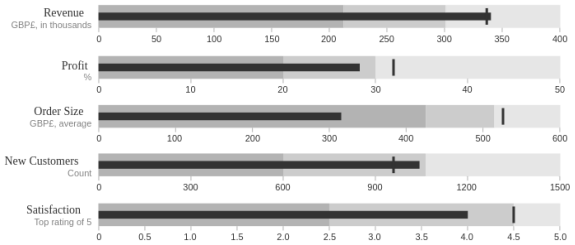
Patterns

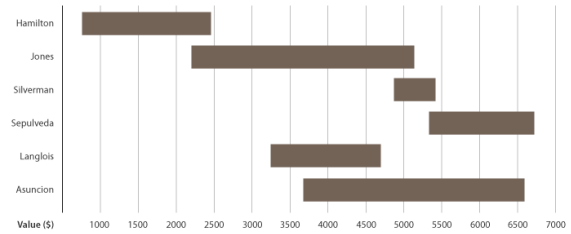
Visualization methods that can reveal forms or patterns in the data to give it meaning.

Range (box plot, bullet graph, span chart)

Visualization methods that display the variations between upper and lower limits on a scale.

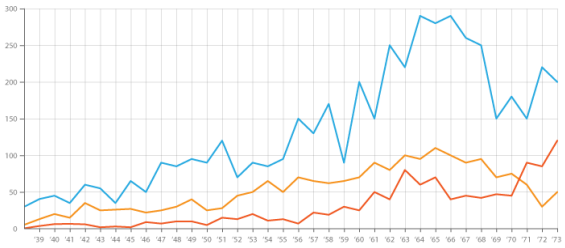


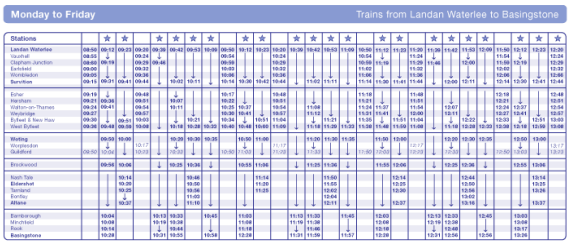




Data over time (line graph, time table)

Visualization methods that show data over a time period to display as a way to find trends or changes over time. Show the sequencing of events over time periods.





Analyzing text (word cloud)

Visualization methods that reveal patterns and insights from a body of text.

Reference tool (stem and leaf plot)

Visualization methods that can be used as a referencing tool to easily look-up individual data points.

Northbound	Hour	Southbound
45 20 03	5	05 48
55 49 32 20 13 01	6	02 23 35 57
58 53 49 44 38 32 25 19 13 08 02	7	00 07 16 20 26 30 37 46 52 59
59 57 54 50 47 44 39 35 31 28 24 21 18 14 09 05 00	8	01 08 12 17 21 29 31 35 39 44 49 53 58
52 48 44 39 34 29 23 18 12 05	9	03 10 18 27 32 37 45 51 58
53 47 41 37 32 27 22 15 07	10	00 07 14 21 30 39 48 57
55 49 35 29 23 16 08 01	11	06 11 19 27 34 41 50 59
56 48 44 39 32 27 21 14 05	12	02 15 30 45 57
50 45 35 30 25 20 15 05	13	03 10 18 23 29 37 45 56
52 43 32 24 12 03	14	00 09 18 27 39 48 57
58 44 31 26 15 06	15	01 17 29 41 55
56 40 30 22 11	16	10 25 38 50
55 41 32 23 14 01	17	00 20 34 53
58 49 42 36 28 22 16 09	18	05 14 21 29 37 45 56
57 51 46 39 33 28 23 17 13 08 02	19	02 09 14 19 23 27 32 36 40 44 48 53 57
52 43 30 21 15 06	20	09 17 26 34 40 49 55
45 30 16 03	21	10 20 30 40 50
50 30 10	22	15 35 55

Putting it all together

We chose to use a network diagram and line plot, because we believe they offer the means to answer a lot of our questions while still being easy to quickly understand.

Furthermore, if deemed necessary, both plots may be enhanced with zoom and / or pan techniques. This would greatly improve user experience.

Line connections (aka network diagram)

This network diagram type graph will show stop stations as nodes and edges as a line connection between them. First, we identify the most frequent stop station and from it we perform breadth-first lookup of its neighbors. We can continue until we have all stations in our diagram. However, the total count of the stations is too much for our use case, so we will limit ourselves to trams or subways only. Furthermore, if there exists a station that has only two neighbors, we can omit that station too, because there will still exist a connection between aforementioned two stations. These improvements would greatly reduce the number of nodes.

The node diameter will be bigger the more connection with other stations it has [questions 3, 15]. The edges between two stations will be thicker for connections that have more different lines between them (it could happen that two trams are sharing part of their trips, so the shared parts will have thicker lines) [10, 12].

The type of transportation makes sense only when showing more than one type. We can make the edge connecting two nodes in parallel colors. For example, the edge has 2 trams and 1 bus, so the line with thickness of 3 will have 2 pixels in tram color and 1 color in bus color [1].

We can merge multiple platforms into stations. We can also add to the diagram the direction of the connection, because some of the stations do not exist in both directions. This could be done by creating an arrow instead of a line. The graph will then be directed.

In addition to this, the color of the edge could indicate if the connection is wheelchair accessible or not (green and black respectively). The same would apply to stations. [5, 6]

One can easily incorporate zoom and mouse interaction to have a better overview of certain areas. After the zoom in, the previously hidden stations with only two neighbors will be shown again. The nodes can be dragged and reordered. (We explicitly do not use geographic location for our case).

Because the number of the stations could still be significantly big, we will provide a search box for users to identify searched stations. That station could be marked as selected and we can show information about it (name, which lines it connects and so on). We can expect the same behavior when we click on the station's node [12].

The purpose of this graph is **not** to represent an actual map of the Prague public transport but rather to show relationships between individual stations and how they connect to themselves.

Line plot

In order to identify and analyze transportation patterns in time, we selected a line plot as the one visualization technique we are going to use.

Starting with a simple line visualizing density of transport connection [questions 7, 11 and 14]. We may build up with connection type recognition [2], and / or reporting of traffic accidents in time [8, 9].

In the simplest transportation density line graph. We plot time on a discrete scale with a predefined time window (for example 15 minutes) on the x-axis. On the y-axis, we plot the density of transportation as the count of connections in a given time window. We either choose a line or a line with a filled area below - with a single color. During the 11 day period there will be hundreds of time windows, thus we do not map points to circles, but rather interpolate the line. For plotting the ratio of connection types (bus / subway / tram) we use three different colors and values stacked on each other. The traffic issues may be mapped to red circles plotted onto the line on top.

We discussed using either a) parallel line plot with multiple days plotted under each other and b) serial line plot with time windows spanning the whole 10 day period.

In the end we chose the serial one.

Conclusion

In this report, we discussed our task and the data it contains. Furthermore, we determined questions that we may answer. Then we researched various visualization techniques. Finally we chose visualization techniques that we may use, mapping and other implementation details.

Following this report, we are going to create a responsive web application with a dashboard illustrating answers to our questions.

References

https://yali.state.gov/use-the-star-analysis-method-as-an-interview-technique/

https://opendata.praha.eu/dataset/dpp-jizdni-rady

https://opendata.iprpraha.cz/DPP/JR/jrdata.zip

https://datavizcatalogue.com/index.html