String Charter 2: Visual Transport Schedules
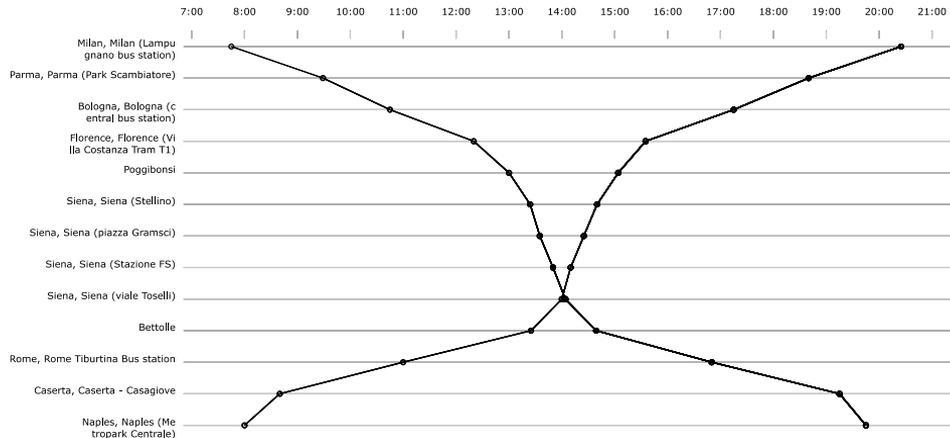
Information Visualisation 2023
Group 2
Inge Gsellmann, Michael Hebesberger, Danijela Lazarevic
28.06.23

String Charts

- Time & distance chart

- Illustrate movement of trains between stations over time.

- Visualise operational problems (e.g. delays, train bunching, or gaps in service).

Source: SVG generated by Inge Gsellmann, from Flixbus dataset.

Source: https://pvibien.com/stringline.htm

History

- Pioneered and developed by Charles Ibry.

- Ibry managed traffic on the route from Paris to Rouen.

- First stringline chart published in 1847.

Source gallica.bnf.fr / Bibliothèque nationale de France

General Transit Feed Specification (GTFS)

- Base is Static with Realtime as extension.

- 6-13 CSV files (as txt) in a zip archive.

- Show **routes** as string chart, with **stop_times** as data values.

- Data from **trips** for data linking.

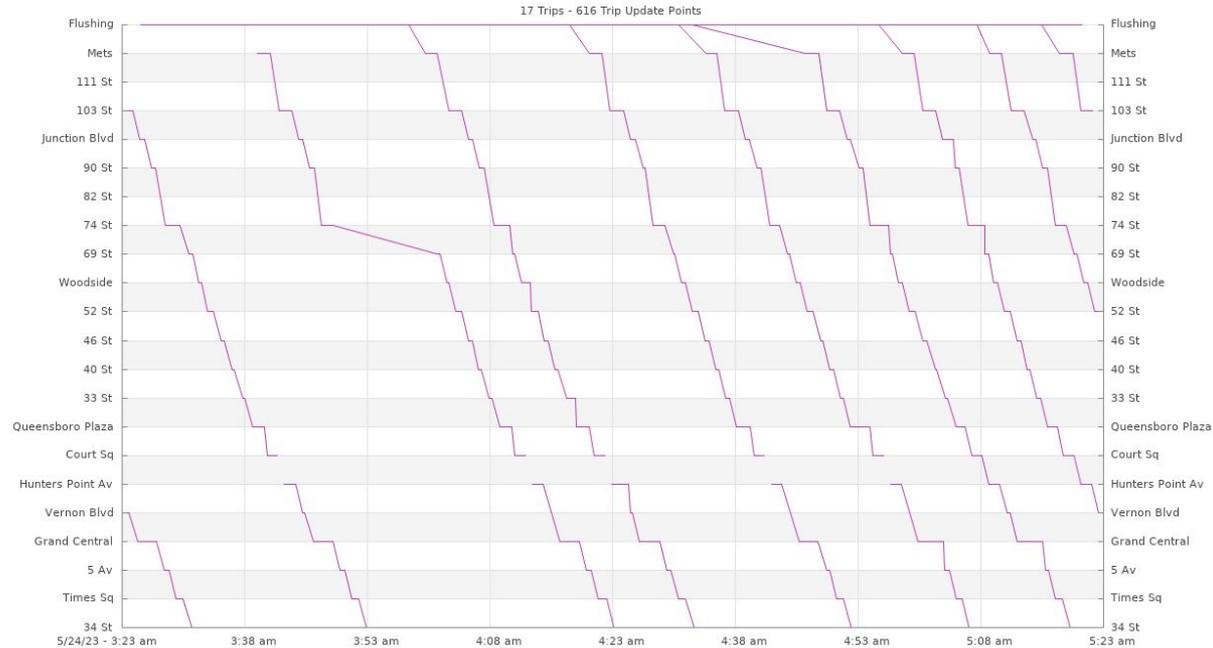- Additional information as needed from **stops**, **calendar** etc.
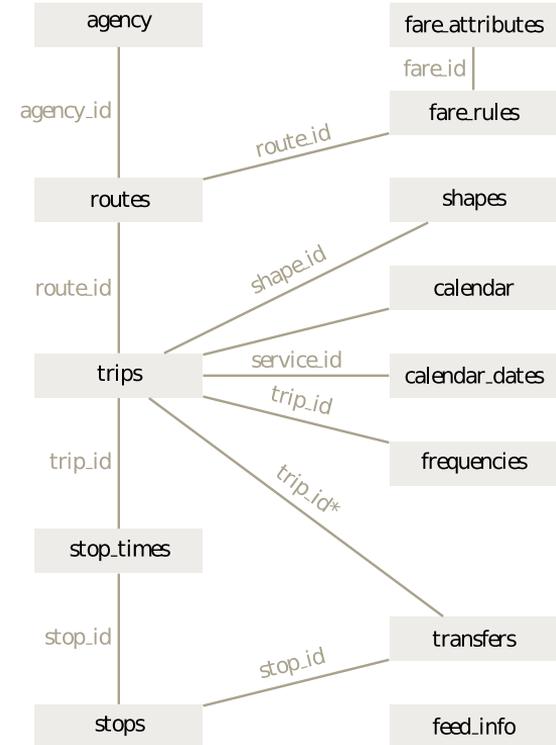
Observable Marey Trains

- Interactive chart

- Uses baby bullets for train stops.
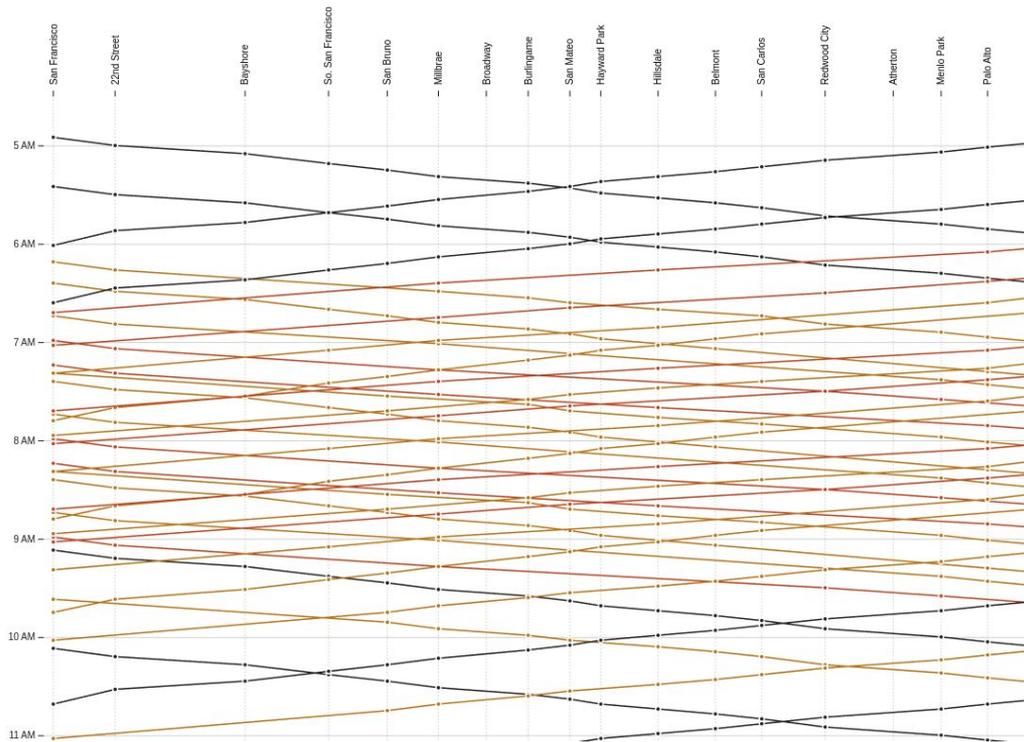
Vibien • NYC Subway Stringlines

- **Display**

  additional variables.
  - Dwell Time
  - Travel Time
  - Run Time
  - Trip IDs
  - Greyscale colour
  - Headways

- **Real-Time Schedule**

17 Trips - 616 Trip Update Points

GTFS-To-Chart

- JavaScript library to create stringline charts.
- Creates stringline charts:
    - stations across x-axis, spaced to scale.
    - time on y-axis.
- Each line of the chart represents a vehicle.
- Problems:
    - Routes that sometimes make different stops.
    - Routes where the way back is different.
    - Circular routes.

Our Project

- Cross-platform
- Interactive
  - Baby bullets
  - Highlight selected strings.
- Accessible and intuitive design
  - Responsive to different devices.
  - Non-overlapping labels.
- Export as SVG

Tauri

- Framework to build cross-platform apps using web technologies:

  - Windows, macOS and Linux.

- Compatible with almost every frontend stack (e.g. Html/CSS/JavaScript, SvelteKit, Qwik, Next.js, Vite, etc.).

- Uses Rust for the backend (plan on supporting Go, Nim, Python, Csharp, etc. in the future).

- Custom web renderer (more lightweight and performant than Chromium)

- Open Source

- Secure and sandboxed

Tauri Setup

All new versions of:

- IDE: Visual Studio Code 1.78.2
- VS Code Extensions:
  - Cargo 0.2.3
  - CodeLLDB 1.9.2
  - rust-analyzer 0.3.1541
  - Tauri 0.2.6
- Node.js 18.16.0
- npm 9.6.7
- Rust 1.62
- Typescript 4.8.2
- Jszip 3.10.1
- Papaparse 5.4.1

Create template: https://github.com/tauri-apps/create-tauri-app

ÖBB Timetables

- ÖBB Personenverkehr

- All routes operated by ÖBB, not just within Austria.

- License: Creative Commons Namensnennung 4.0 International

- GTFS format

- Provides mandatory tables plus calendar_dates.

GTFS Parsing

- In TypeScript.
- No overhead with separate python script.
- Open GTFS zip and read them as CSV:
  - using JSZip and paparse libraries.
- Parsing trips, stop times and stop names and link them together.

GTFS Parsing

- Route:
  - route_id
  - name
- Trips:
  - trip_id
  - route_id
- StopTimes:
  - trip_id
  - stop_id
  - arrival_time
- Stop:
  - stop_id
  - stop_name

Route

Trips		

StopTimes		

| Stops		
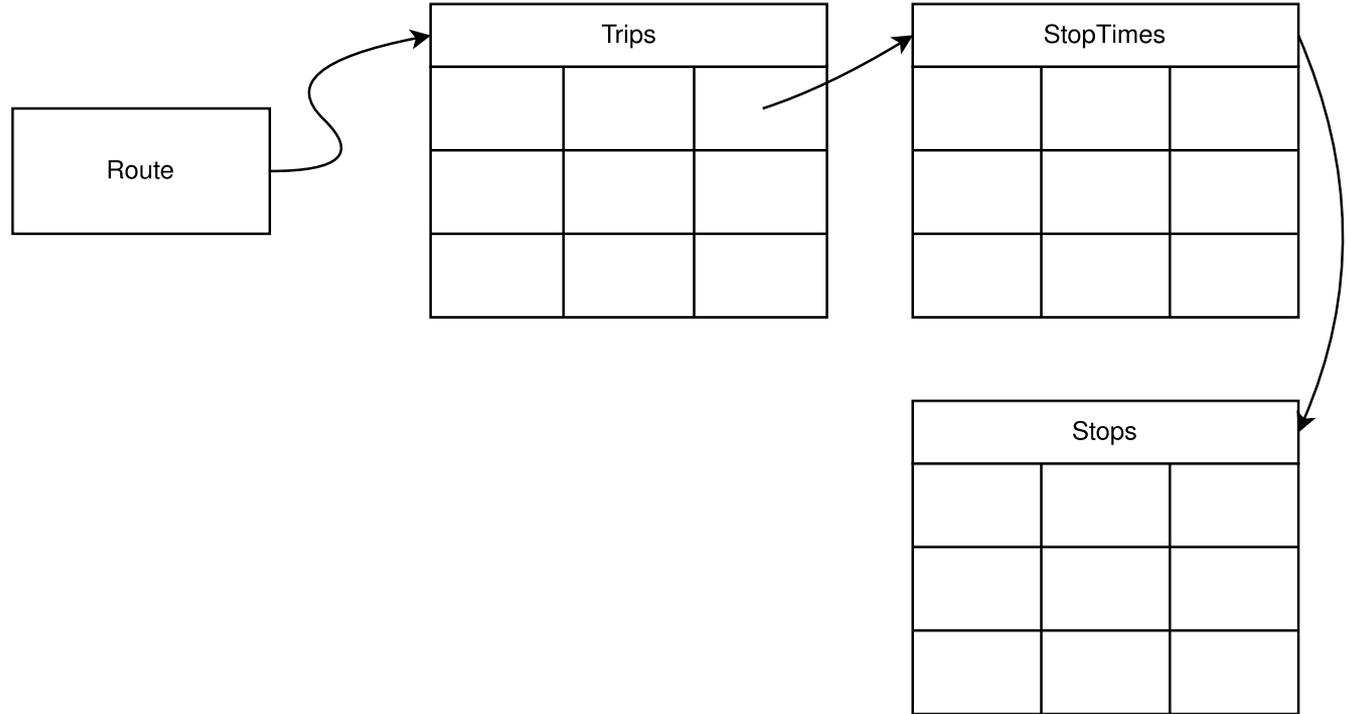
Routes and Trips

- Route describes lines in public transportation systems.

- Trips describe when a specific route occurs.

- Trips can have different stations inside a route.
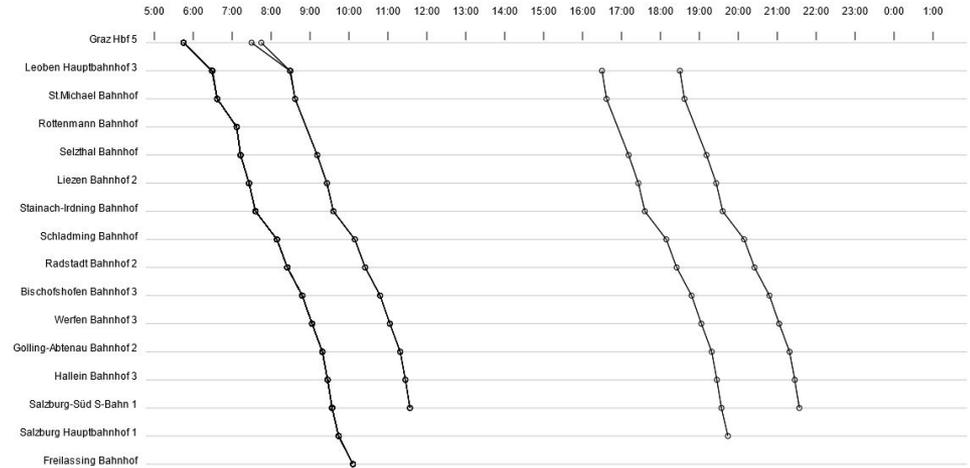
Source: Screenshot captured by Michael Hebesberger.

Drawing the Chart

- Draws chart on an HTML canvas.
- Stops are marked by circles.
- Stops are placed between time labels in proportion to the time.
- Mouseover info on circles show trip, station and exact time.
- Axes can be flipped with a toggle.
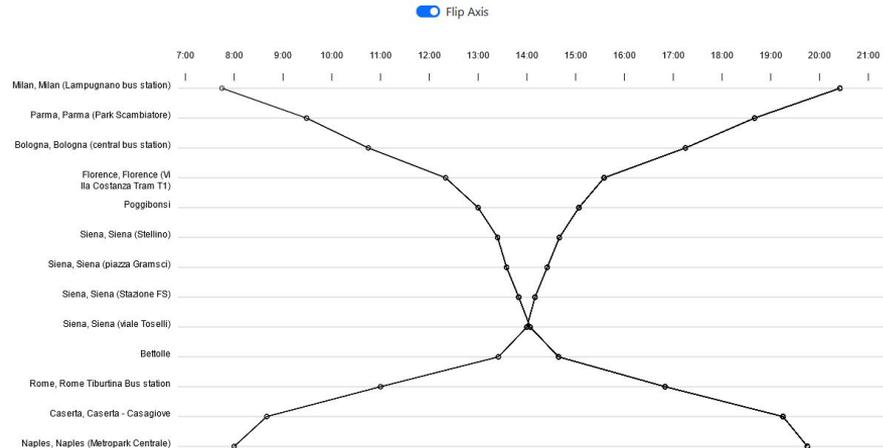
Source: Screenshot captured by Inge Gsellmann.

SVG Export

- File is generated and can be downloaded.
- SVG tags generated separately from Canvas Elements for better spacing.
- Proper SVG tags for text, lines, circles.
- Generated file passes W3 Markup Validation for SVGs.
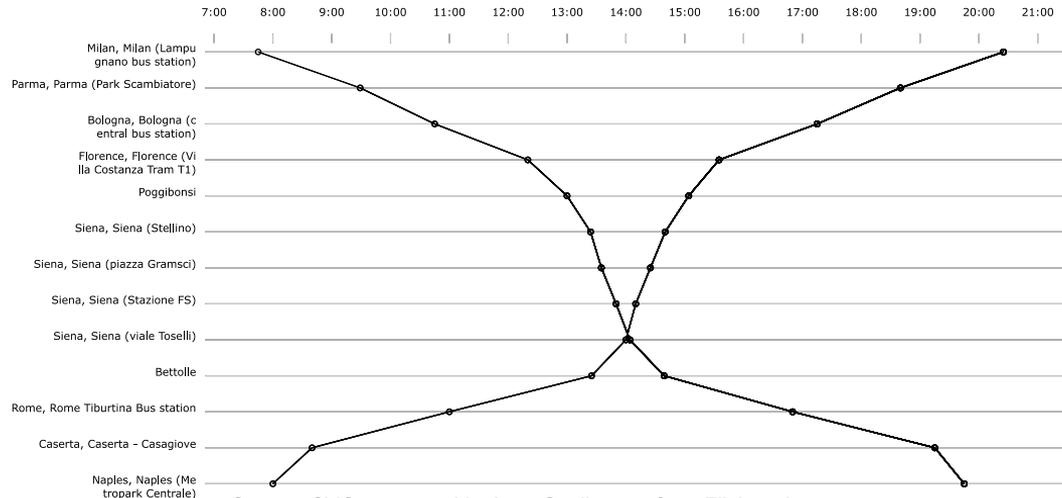
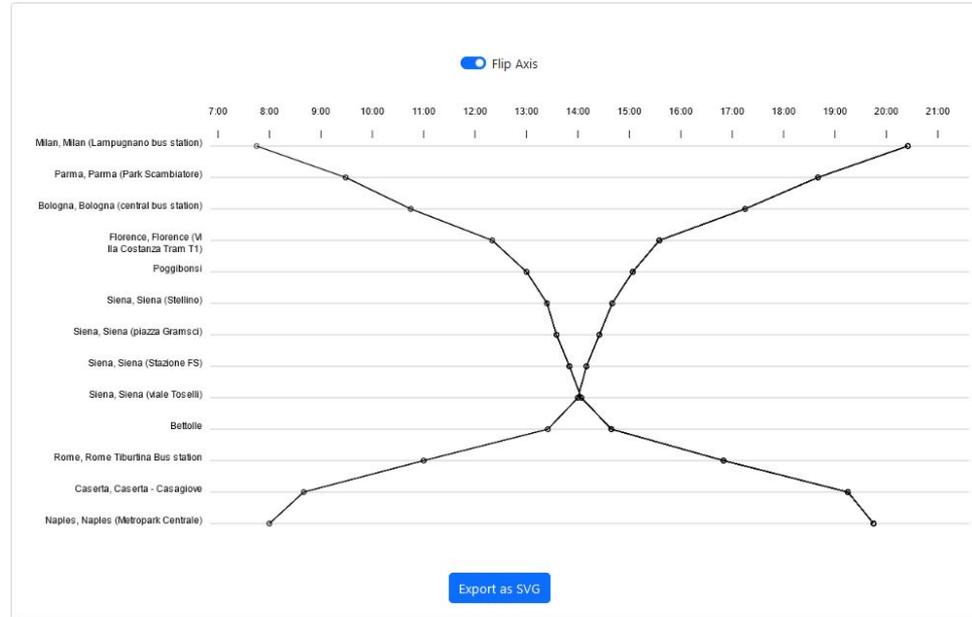Source: SVG generated by Inge Gsellmann, from Flixbus dataset.

String Charter 2: Visual Transport Schedule

https://youtu.be/hlEWq_SxbqI

Questions?