**PASAJEROS VBZ**

**Fuente de los datos:** [**https://data.stadt-zuerich.ch/dataset/vbz\_fahrgastzahlen\_ogd**](https://data.stadt-zuerich.ch/dataset/vbz_fahrgastzahlen_ogd)

[**https://github.com/VerkehrsbetriebeZuerich/ogd\_examples\_python?tab=readme-ov-file**](https://github.com/VerkehrsbetriebeZuerich/ogd_examples_python?tab=readme-ov-file)

Calidad de los datos: valores calculados

Objetivo: Help you getting started with the análisis of travel time data or Passenger data of Verkehrsbetriebe Zürich (VBZ), the public transport operation in the Swiss city of Zurich. The datasets are published as Open Government Data (OGD)

Use cases:

Passengers per line/stop

Script

Example\_passengerdata.py

In order to perform the example análisis you need to download a .csv-file that contains Passenger data (“Reisende.csv”) as well as three matching tables from the Open Data Portal. There you can also find additional descriptions (in German only)

Description of tables:

* REISENDE.csv: Main table, contains information about the number of passengers etc
* LINIE: Matching table, contains information about line numbers etc
* TAGTYP.csv: Matching table, contains information about the validation of timetables etc
* HALTESTELLEN.csv: Matching table, contains information about stops names etc
* GEGAESSGROESSE.csv: Matching table, contains information about vehicle capacity

Output: dataframe con las siguientes columnas: Linien\_id, Linienname, Linienname\_Fahrgastauskunft, pax\_per\_year

Punctuality per line

example\_traveltimedata.py

To perform the example análisis you need to download a .csv-file containing travel time as well as two matching tables from the Open Data Portal. You’ll also find addiotional descriptions there

* Fahrzeiten\_SOLL\_IST\_20200809\_20200815.csv: Main table, contains actual travel time raw data (each file contains one week of data)
* Haltepunkt.csv: Matching table, contains information about the GPS sition of each stop point
* Haltestelle.csv: Matching table, contains information about the full stop names.

Output

Percentage of the punctuality per line for 2019 based on the input table above as shown in the example scrip (data frame “punctuality”)

According to the punctuality definition of VBZ, a ride is considered on time (punctual) when the actual arrival time at the stop does not exceed the scheduled arrival time by more tan 2 minutes (otherwise defined as “delayed”) or the actual departure at a stop does not happen more tan 1 minute earlier tan the scheduled departure (otherwise defined as “too early”)

**1. Passenger Metrics**

• **Daily/Hourly Passenger Counts**: Visualize the number of passengers per route, line, or station over time.

• **Peak Hours**: Identify peak travel times across different lines and days (e.g., weekdays vs weekends).

• **Boarding/Alighting Data**: Highlight stations with the highest boardings and alightings.

• **Occupancy Rates**: Calculate and display occupancy rates of vehicles based on seating capacity (SITZPLAETZE).

**2. Route and Line Performance**

• **Line Utilization**: Compare lines based on passenger counts and utilization percentages.

• **Distance and Efficiency**: Evaluate the average distance traveled by passengers per line or station and compare it to capacity.

• **Bottlenecks**: Identify bottlenecks in the network using passenger flow data at stations or sections.

**3. Operational Metrics**

• **On-Time Performance**: Track adherence to planned schedules using Plan\_Fahrt\_Id and compare it with actual performance.

• **Route Frequencies**: Display the frequency of routes and identify gaps or areas for service improvement.

• **Vehicle Capacity Utilization**: Analyze utilization rates for different vehicle configurations (KAP\_1m2, KAP\_2m2, etc.).

**4. Spatial Analysis**

• **Heatmaps**: Create spatial heatmaps to visualize boarding/alighting counts across the network.

• **Station Accessibility**: Rank stations by usage and proximity to critical areas (e.g., downtown, suburbs).

• **Network Coverage**: Show the geographic spread of the network and areas with low accessibility.

**5. Temporal Trends**

• **Seasonal Usage Patterns**: Compare usage data across seasons or holidays (e.g., weekdays, weekends, special events).

• **Year-over-Year Comparisons**: Highlight changes in passenger counts, route performance, and operational efficiency over different years.

**6. Sustainability Metrics**

• **CO2 Emissions**: Estimate emissions per line or route based on passenger numbers and vehicle types.

• **Energy Efficiency**: Display energy consumption per passenger-km.

**7. Customizable Filters**

• Include interactive filters for users to drill down into:

• **Specific lines (**Linien\_Id**)**

• **Time periods (peak vs non-peak hours)**

• **Station-level metrics (**Haltestellen\_Id**)**

• **Vehicle capacities**

**8. Anomalies and Alerts**

• Highlight anomalies in passenger flow, such as unexpected spikes or drops.

• Provide alerts for routes that consistently operate under or over capacity.

**Tools and Technologies**

• **Frontend Dashboard**: Use tools like Tableau, Power BI, or Python Dash for interactive visualization.

• **Backend Analysis**: Use Python or R for data cleaning and metric calculations.

• **Database Integration**: Connect the schema (likely SQL-based) to your dashboard for real-time updates.

**Example Dashboard Layout**

1. **Overview Page**: Total passengers, total routes, and key metrics at a glance.

2. **Performance Page**: Detailed operational metrics for each route and line.

3. **Spatial Page**: Interactive maps with route and station-level data.

4. **Trends Page**: Year-over-year or seasonal performance comparisons.

5. **Sustainability Page**: Emissions and energy use insights.