## *DFG RiD: Attribute Description of Data Table*

| Attribute | Unit | Type | Description |
|---|---|---|---|
| ARS | - | text | Official municipality code (ARS) assigned to the trip. Documentation: https://www.destatis.de/EN/Themes/Countries-Regions/Regional-Statistics/OnlineListMunicipalities/_inhalt.html, please note: this has to be a 12 Character value. If not, a leading zero is missing due to import |
| gemeindename_kurz | - | text | Municipality name according to German ARS code, please use UTF-8 encoding, to get the right name deisplayed |
| trajectory_id | - | UUID | ID of the cycling trip (trip ID) |
| device_id | - | UUID | ID of the recording device |
| points_count | count | Int | Number of GPS points of the trip |
| mode_type | - | Int | Mode of transport (2: sport cycling, 3: cycling) |
| start_time | seconds | Int | Start time of the trip (UNIX timestamp) |
| end_time | seconds | Int | End time of the trip (UNIX timestamp) |
| duration | seconds | Int | Trip duration |
| start_lat | decimal degrees | double | Starting Latitude |
| start_lon | decimal degrees | double | Starting Latitude |
| end_lat | decimal degrees | double | Ending Latitude |
| end_lon | decimal degrees | double | Ending Longitude |
| distance | meters | double | Travelled distance |
| air_distance | meters | double | Euclidean distance between trip origin and destination |
| detour_factor | – | double | Detour factor of the trip *=distance/air_distance* |
| speed_avg | m/s | double | Average speed of the trip |
| speed_v50 | m/s | double | 50th percentile (median) of speed values |
| speed_v85 | m/s | double | 85th percentile of speed values |
| waiting_events_count | count | Int | Number of waiting events during the trip |
| waiting_events_total_duration | seconds | Int | Total waiting time of all waiting events |
| waiting_events_tl_count | count | Int | Number of waiting events at signalized intersections |
| waiting_events_tl_total_duration | seconds | Int | Total waiting time at signalized intersections |
| waiting_events_ratio | % | Double | Ratio of total waiting time to trip duration (waiting_events_total_duration / duration) |
| crossed_junctions_count | count | Int | Number of crossed signalized intersections |
| accelerations_pos_count | count | Int | Number of positive acceleration events |
| accelerations_pos_total_time | seconds | Int | Total duration of positive acceleration events |
| accelerations_pos_a50 | $m/s^2$ | Double | 50th percentile of positive acceleration values |
| accelerations_pos_a90 | $m/s^2$ | Double | 90th percentile of positive acceleration values |
| accelerations_pos_a95 | $m/s^2$ | Double | 95th percentile of positive acceleration values |

| | | | |
|---|---|---|---|
| accelerations_neg_count | count | Int | Number of negative acceleration events |
| accelerations_neg_total_time | seconds | Int | Total duration of negative acceleration events |
| accelerations_neg_a50 | $m/s^2$ | Double | 50th percentile of negative acceleration values |
| accelerations_neg_a90 | $m/s^2$ | Double | 90th percentile of negative acceleration values |
| accelerations_neg_a95 | $m/s^2$ | Double | 95th percentile of negative acceleration values |
| gender | text | Char | Gender (based on device ID) |
| age | years | Int | Age (based on device ID) |

**For details regarding data pre-processing please see**:

Lißner, S., Huber, S. Facing the needs for clean bicycle data – a bicycle-specific approach of GPS data processing. *Eur. Transp. Res. Rev.* **13**, 8 (2021). https://doi.org/10.1186/s12544-020-00462-2

## Specification of Waiting Events

**Waiting time** represents the actual delay occurring at intersections. It is detected based on a movement state classification and a speed threshold. A stop is defined when the smoothed speed falls below 0.5 km/h. Consecutive GPS points with the status "stop" are grouped into a waiting event.

The waiting time $t_w$ in general is calculated as:

$$t_w = \sum_T \begin{cases} 0\ s, if\ v_s \geq v_t \\ t_T, if\ v_s \leq v_t \end{cases}$$

With $v_s$ denoting the vehicle speed in the given second and $v_t$ denotes the threshold. To account for inaccuracies in GPS trajectories, consecutive waiting events that are separated by no more than 10 seconds and 40 meters are merged into a single waiting event. The duration of a waiting event is defined as the time difference between the first and the last GPS point with the status "stop".

The dataset reports (i) the total duration of all waiting events and (ii) the total duration of waiting events occurring specifically at intersections. Waiting events are assigned to intersections if the corresponding GPS points fall within spatially defined intersection areas. The delineation of these areas is described in the following section.

The purpose of defining intersection areas spatially is to distinguish between waiting events occurring at intersections and those occurring outside of them. The identified areas include both road intersections and pedestrian and cyclist crossing facilities. The road network within the study areas was derived from data provided by OpenStreetMap and includes all roads accessible to motorized traffic. Buffer zones are generated around intersections and crossings to represent cyclist waiting areas, including an additional margin to account for positional uncertainty. An example can be seen in Fig. 1. Grade-separated intersections are excluded.

*Fig 1: Map Example of Intersection Areas (Map Data: OPENSTREETMAP CONTRIBUTORS, 2025)*

**Documentation – Intersection Layer**

The intersection buffers were generated in QGIS using the following procedure:

1. *Load the road network from OpenStreetMap.*
2. *Dissolve the network.*
3. *Split lines using the network itself.*
4. *Create buffers (18 m; square; beveled) from the resulting segments.*
5. *Intersect the buffers with themselves.*
6. *Remove polygons resulting from self-intersections.*
7. *Calculate compactness using:*

$$compactness = \frac{perimeter^2}{4\pi \times area}$$

8. *Remove polygons with a compactness value below 1.6.*
9. *Dissolve while keeping disjoint features separate.*
10. *Create buffers (100 m; round).*
11. *Apply a negative buffer (−100 m; round).*
12. *Merge with crossing features that do not intersect polygons from step 11.*
13. *Compute indicators and counts for tram lines, bus lines, traffic signals, and crossings.*

14. *Create buffers for signalized intersections (15 m; round).*
15. *Assign overlapping areas from step 14 to the larger polygon.*
16. *Perform spatial joins (aggregation) between step 15 and at-grade intersections to compute the number of at-grade intersections per polygon.*
17. *Remove intersections where all indicators equal zero (traffic signals, crossings, and intersection count).*

**Documentation – At-grade Intersections**

1. *Extract specific vertices (start and end points) from the intersection layer (step 3).*
2. *Dissolve while keeping disjoint features separate → resulting in all intersection points in the network.*
3. *Extract vertices from the original network (step 1).*
4. *Compute the difference between steps 2 and 3 → grade-separated intersections.*
5. *Compute the difference between steps 2 and 4 → at-grade intersections.*