Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of the Environment, Transport, Energy and Communications (DETEC)

Bundesamt für Raumentwicklung ARE
Sektion Grundlagen

Documentation Plausibility check

Mobility and Transport Microcensus (MTMC) 2021

Datum:	March 2023
Written by:	Fundamental Policy Questions Section, ARE

Table of Contents

List of Tables

List of Figures

1 Introduction

This document describes the plausibility check of the Mobility and Transport Microcensus (MTMC) data 2021. The MTMC is a statistical survey of the travel behaviour of the Swiss population. It is conducted every five years by the Federal Statistical Office (FSO) and the Federal Office for Spatial Development (ARE). This document describes the plausibility checks performed by the ARE, mainly regarding the spatial data. Further checks are performed by the FSO.

The document deals with the different phases of the plausibility check and the selection of the threshold values and factors used. The thresholds and methods used in 2015 are taken over as far as possible, so that the differences found between 2015 and 2021 are due to differences in the data and not in the methodology.

The aim of the plausibility checks is to find wrong or inconsistent data in order to obtain an unbiased and proper final data set. In this sense, the data available to researchers, cantons, practitioners and other federal offices (after signing a data protection contract) are not fully raw data; the data have been made plausible and cleaned. This document describes the different steps of the plausibility checks and serves as documentation. At the same time, it creates transparency about the possible adjustments. Therefore, the two main target groups are the ARE itself and researchers interested in the details of the performed checks.

Hereafter, we describe the MTMC data. We use the open-source software R to perform the plausibility checks. The steps of the plausibility checks are described here, in order to explain the main logic and show the used threshold values and factors. The detailed working steps can be found in the corresponding R scripts. References are indicated in each section. This document has therefore to be considered in addition to the relevant R scripts. The R scripts as well as the additionally used input data can be requested from befragung@are.admin.ch.

1.1 Division of tasks between FSO and ARE

In Figure 1 the areas for which the ARE is primarily responsible are marked in blue. The areas for which the FSO is primarily responsible are highlighted in orange, and the common areas are streamlined.

2 The MTMC

The MTMC contains questions about:

- the socioeconomic characteristics of households and individuals
- mobility tools (vehicles and public transport season tickets)
- daily mobility (trips on a given reference day)
- occasional journeys (day trips and trips with overnight stays)
- preferences for transport policies in Switzerland (see Danalet et al. (2022)).

A short version of the questionnaire can be found here.

2.1 MTMC 2020 and 2021

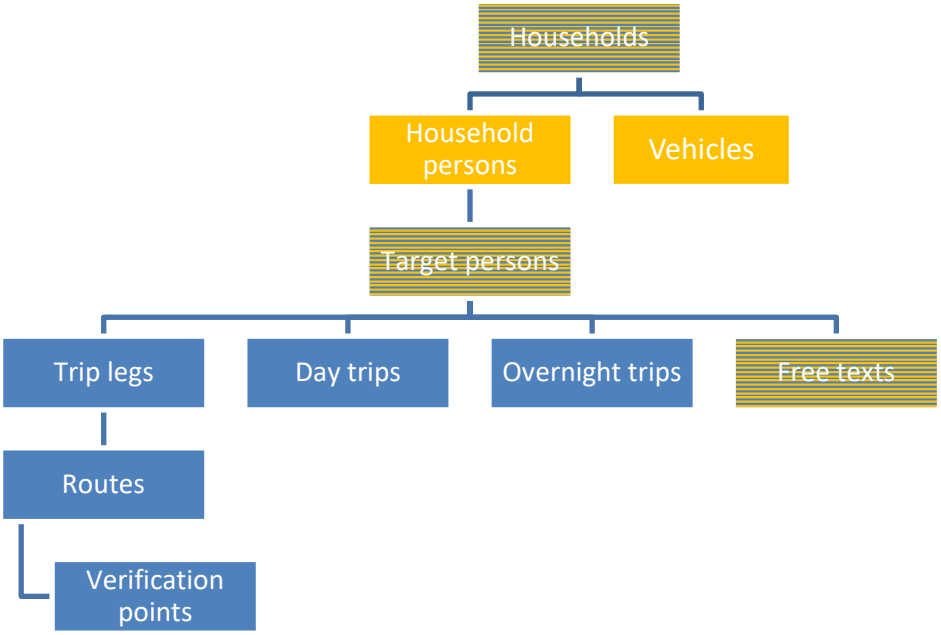
The MTMC survey had been planned for 2020. At the beginning of that year, the interviews could take place as planned. However, when public life largely came to a standstill in March 2020 in the wake of the first wave of the Covid pandemic and public transport services were severely restricted, the survey had to be cancelled and postponed for a year.

A special analysis (without weighting) of the microcensus data from early 2020 and early 2021 was carried out. At the start of 2021, just under a year after the beginning of the pandemic and roughly in the period of the second lockdown, on average almost a third fewer kilometres were covered per person than immediately before the first wave of infections. A particularly sharp decline was seen in public transport distances with a reduction of 52%. Motorised individual transport fell considerably less at -27%. Distances covered on foot and by bike saw hardly any difference. See also Effects of the Covid 19 pandemic on mobility behaviour.

2.2 Data structure MTMC

This section describes the MTMC data. The following diagram shows the simplified data structure. The structure also serves as the basis for the subdivision of the plausibility check. The individual data sets are briefly described below. Since the variable names are originally in German we also provide the German titles here. The data of the Modul 3 (Attitudes towards transport policy) is not treated in this document.

Figure 1: Data structure MTMC



Households (Haushalte): The top level of the data structure is formed by the households. The households file contains information concerning an entire household, e.g., number of persons in the household, number of vehicles or geographical location. Each household is assigned a unique identification number (hhnr).

Household persons (Haushaltspersonen): The information on the persons of the household (e.g., age, sex, possession of driving license) is recorded in a separate file. Each household person can be clearly assigned to a household via the household number (hhnr). The household persons can be clearly identified by the combination of household number and household person number (hhnr + hpnr).

Target persons (Zielpersonen): Each household is assigned one target person who is questioned about their daily mobility (trips on a given reference day), occasional journeys (day trips and trips with overnight stays) and attitudes towards transport policy in Switzerland. The target person file contains information related to the target person, such as marital status, labour market status, ownership of public transport subscriptions, etc. Each target person is uniquely assigned to their household via an identification variable (hhnr). Via this variable hhnr and the variable from the household persons file, a

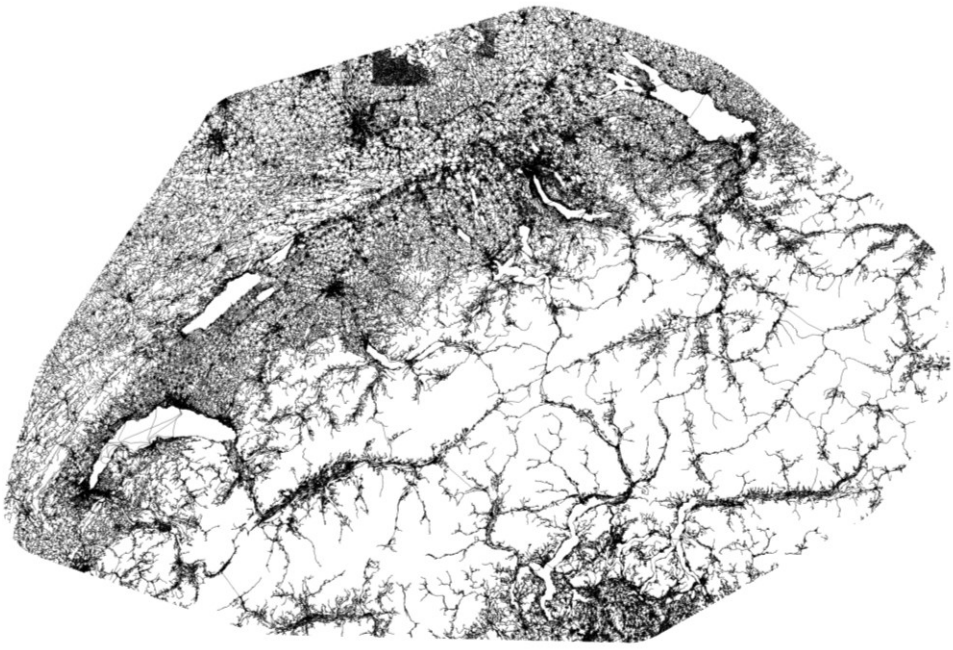
unique assignment to a household person is possible (hpnr, the target person is always the household person with hpnr=1).

Vehicles (Fahrzeuge): The vehicles file contains questions on all cars (vehicle type=1) and motorbikes (vehicle type=2) of a household, e.g., year of introduction, engine capacity, speedometer reading, etc. Each vehicle receives a unique assignment to a household person. Each vehicle receives a unique identification number (hhnr + fznum). If a household has several cars (or motorbikes), these are numbered consecutively in the variable fznum. The cars and motorbikes can be uniquely assigned to a household via the household number (hhnr).

Trip legs (Etappen): The trip legs represent the smallest unit of the daily mobility. They have a minimum length of 25 meters and are covered by a single means of transport, which includes walking. If the means of transport is changed, a new trip leg begins. Changes of location within buildings do not constitute trip legs. Each trip leg has a unique identification number (hhnr + etnr). As the route choice is recorded geographically in the MTMC 2021, the trip leg data record contains diverse route information for each trip leg. For public transport trips, additional information about the course is included (HAFAS-ID), which allows conclusions to be drawn about the stop, train type (now also included as a variable) and departure times.

Routes (Routen): Each mobile target person covers a route between the starting point and destination of a trip leg with a means of transport on a transport system. If a trip leg could be successfully routed in the MTMC 2021, the route file contains the associated geometry data in the file format of an ESRI shapefile. In the case of a trip leg by car or motorbike (motorized individual transport) and road-bound public transport, trip leg routes are based on the road network of the company TomTom. In the case of rail-bound public transport, the routing is based on the rail network of the Swiss passenger transport model developed by the Transport Modelling Unit of DETEC (VM-UVEK). Each route is clearly assigned to a stage (hhnr + etnr). Thus, an unambiguous connection with the household and the target person is also possible.
Routing is done for all trips on the territories of Switzerland and Liechtenstein. In order to be able to route trips starting and ending in Switzerland, but going through the border in-between, the road network has been extended to a convex hull of Switzerland, to which a band of 20 km was added (see Figure 2).

Figure 2: Routing zone on the road: convex hull of Switzerland and 20 km around



Verification points (Verifikationspunkte): As part of the route verification of the MTMC 2021, at least two verification points were requested on the route in the case of motorized individual transport with a distance of more than 3 kilometres. The file “Verification Points” contains information on these passed locations for each verified trip leg. In addition, for cross-border trip legs, the file contains the details of the border crossing, including geocoding.

Segments (Segmente): In addition to the MTMC 2021 routes file, the segments file contains all TomTom segment identification numbers of the routing for motorised private transport (for successfully routed trip legs). The file allows the attributes contained in TomTom (road type, curvature, etc.) to be added to the road-based routing. Each segment has a unique identification number (hhnr + etnr + segnr), which allows it to be assigned to the household, the destination and the trip leg. In 2021 this data is not delivered to those who order the detailed data at FSO.

Day trips (Tagesreisen): 30% of the target persons are asked about their daily trips in the additional module 1a "daily trips" (number of daily trips, purpose of daily trips, choice of means of transport, etc.). The number of daily trips per person is recorded in a reference period (14 days), whereby detailed information is recorded for a maximum of three randomly selected daily trips. The daily trips are clearly identifiable via an identification number (hhnr + trenr) and can be clearly assigned to a target person via the household number (hhnr).

Trips with overnight stays (Reisenmueb): The additional module 1b "Trips with overnight stays", which 30% of the target persons answer, contains questions on travel behaviour (purpose of the trips, choice of means of transport, etc.). As with the daily trips, the number of trips with overnight stays is recorded in a reference period (here 4 months), whereby a maximum of three randomly selected trips per target person are recorded in detail. Each trip is identified by a unique identification number (hhnr + renr) and can be clearly assigned to a target person via the household number (hhnr).

Free texts (Freitexte): At the end of the survey the target persons have the option to leave remarks and also the interviewers can address open points or issues. The free texts are analysed manually and based on that it is decided whether manual changes of the data have to be performed. Section 5 goes into more detail on these manual adjustments.

2.3 Routing during the phone interviews

Since the MTMC 2010, the routes of the trip legs on the reference day are collected as geodata. Based on the information given on the phone (transport mode, start and arrival location of the trip leg, start and arrival time of the trip leg), a route is computed, verified on the phone with the person and, if needed, corrected.

For trip legs on foot, there is generally no verification. If it is a round trip (same departure and arrival location, mostly home, e.g., going for a walk), the interviewer asks for the most distant point and defines it by clicking on the map or by selecting a predefined geolocation in a list (e.g., an address).

For trip legs by bike, the most attractive and the fastest routes are computed based on travel time and slope. The interviewer picks the one corresponding the most to the actually performed route. Then, the interviewer asks for two verification points on the route. If a verification point described by the interviewee is not already on the route, the route will be computed again to include the verification point.

Quite similarly, for trips by cars and motorbikes, the fastest and shortest routes are computed, the interviewee chooses between the two routes and gets asked for two verification points along the route. These two verification points are only asked for routes longer than three kilometers.

For trips by public transport, all trip legs are routed using the official time table.

Table 1 summarizes the above described.

Table 1: Details Routing

	Walking + RT Bikes	**Bikes** (no RT)	**MIV** (cars, motorbikes)	**ÖV (Train, Bus)**
Trip Legs	Round trips (RT)	All		All
Distance	>= 3 km	>= 0 km	>= 3 km	>= 0 km
Verification	Walking: 1 verification point, no routing Bike: Routing to the furthest point	2 verifiable points on the route		According to "courses" in the time windows (stop points = Verification points)
In the case of motorized private transport (MIV) and road-bound public transport, the recording of routes was based on the road network of TomTom. In the case of rail-bound public transport, the route selection was recorded using the rail network of the National Passenger Transport Model (NPVM).				

2.4 Online data-cleaning

The live routing allows for a real-time check of the distances recorded directly in the interview. In case of implausible data, the interviewers have the possibility to inquire directly. In this way, incorrect information can be significantly reduced.

All trip legs by car, motorcycle, bus, train, streetcar and bicycle are routed during the interview. In the following cases, an estimate on the distance travelled is additionally requested from the respondents:

- for all trip legs by bicycle and on foot
- for trip legs by car, motorcycle, bus, train, streetcar, an estimated distance is requested if:
  - the routing did not work,
  - the routed distance is less than 3km (for motorized private transport),
  - the departure or arrival address is not precise,
  - the trip leg crosses the border,
  - the trip leg is a round trip,
  - the routed distance is not plausible.

In the last case, plausibility is defined by computing the detour that the respondent did in comparison with the distance as the crow flies: if the routed distance is too large in comparison to the distance as the crow flies, it is defined as not plausible enough to avoid directly asking the respondent about the distance. More precisely, if the routed distance d_{routed} is in the range:

$$d_{asthecrowflies} < d_{routed} < d_{asthecrowflies} * df_{mode},$$

where $d_{asthecrowflies}$ is the distance as the crow flies (great-circle distance) and df_{mode} is the detour factor, the routed distance is considered plausible and the estimated distance is not asked.

Detours taken can be justified by many factors: natural obstacles, the desire to take a less congested or faster route or the desire to choose a more pleasant route (e.g., sightseeing, tourism). By asking the respondent to estimate the distance, we do not prevent the possibility to make such detours. We only check if the transport mode and the departure and arrival time of the trip leg are coherent with such a detour.

3 Systematic plausibility checks

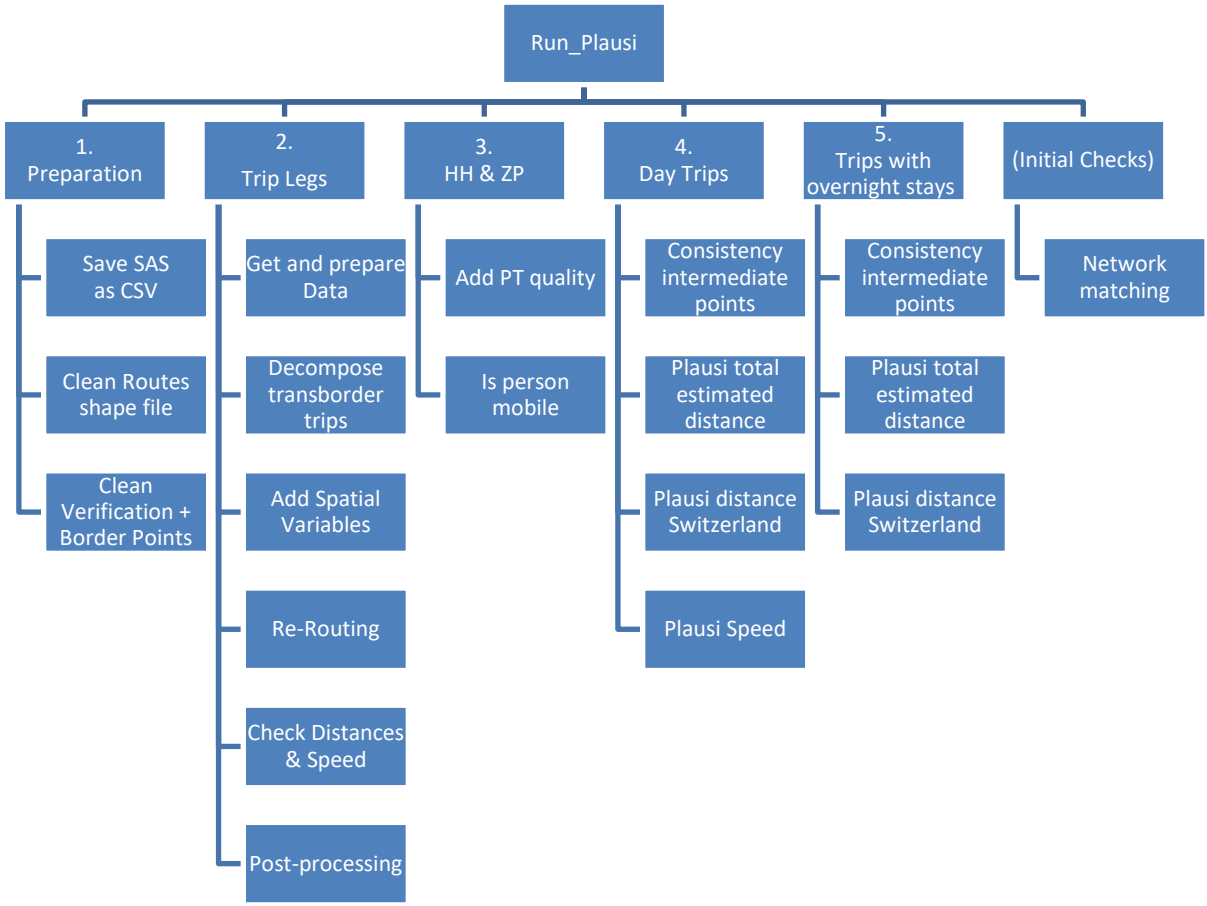
We divide the plausibility checks into systematic and manual checks. Hereafter, we present the systematic part. Given the size of the survey we can't manually check all the details and have to automatize part of the process. The systematic plausibility checks are applied to all data and rely on threshold values and factors. We define limits within which we believe that data are consistent and plausible. If values are beyond the chosen limits an initial correction is applied automatically. After correction we validate the data again, observations remaining implausible are checked manually.

3.1 Structure of the plausibility checks

We structure the checks according to the input data. The following figure gives an overview. The boxes roughly correspond to single R scripts. The here provided explanations should support the understanding of the code.

For more or less each function or script we hereafter provide some additional information. The name of the function or script is always indicated at the end of each section. Together, script and documentation should ease the understanding and application of the code.

Figure 3: Structure of plausibility checks



3.2 Preparation of the data

In a first step, we check the raw data and prepare them for further work.

3.2.1 Save data

We receive the data from the FSO in SAS format continuously over the survey year. First, we convert the data to CSV format.

➔ "save_sas_files_as_csv"

3.2.2 Routes: Clean the shape file

Some trip legs are routed on a network. This section describes how we identify the implausible routes. In particular we remove the straight lines between two points. Straight lines do not correspond to a route, even if the corresponding trip leg might be plausible. In a first step, we load the routed trip legs as geodata. After that we include the manual corrections.

Next, we remove the straight lines and routes with very few information about the routes. We use the number of points per route and the length as relevant criteria. We remove routes with only two points and all routes with less than 20 points, if the length is bigger than 5km.

In the end all routes outside the convex hull (see Figure 2) are removed. We save the cleaned route geodata for later plausibility checks as an intermediate result. After the whole data cleaning process, we clean the shape file again (in case there have been adaptions) and save it as the final plausible file.

➔ run_route_as_geodata

3.2.3 Verification points: adding border points if missing

This data set contains the border points and the intermediate points for the trip legs generated during the interviews. All these geodata are also checked and corrected by the ARE. In case the routing distance is not used as the final distance “rdist” in the trip leg table (etappen), we delete the data from the intermediate points in the dataset “routvfy”, but this is done in a later stage, see chapter 3.3.6.2.

First, we load the route verification and border points (routvfy) received from LINK institute and insert the manual corrections. After that, we add the missing border points. For this we use the routed trip legs and identify the crossing points (intersections) with the Swiss border (including LIE) and add the corresponding information about street name, PLZ, city name and BFS number.

Next, we remove close border points. We remove points computed as crossing the border that are more than 2km from an official border point and similar border points which are less than 250m from each other. Last, we apply post-processing manual corrections and safe the data as an intermediate result.

➔ run_verification_and_border_points

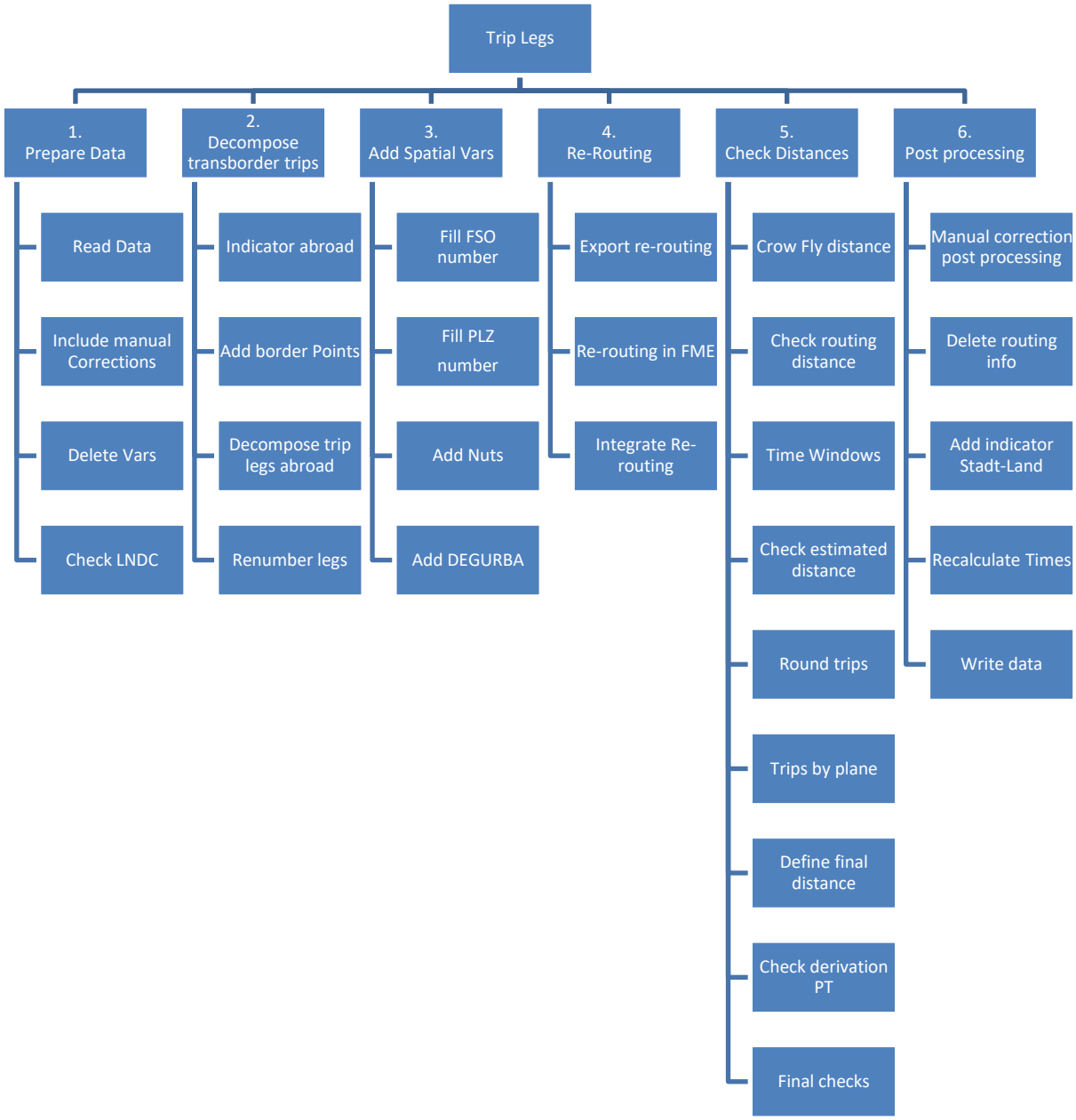
3.3 Trip legs

The trip leg dataset contains all trip legs carried out by all respondents on their reference day. **This is the main part of the plausibility check performed by the ARE. The plausible distance of each trip leg is the single most important output of this check and crucial for further analyses.** That is also why most of the checks share the aim of validating the distance of the trip legs.

As mentioned in the introduction, the plausibility checks base on what was done for the MTMC in 2010 and 2015. Changes in the data due to changes in methodology can therefore be limited. Figure 4 gives an overview of the performed checks.

Here we use the term “trip legs” to translate the German term “Etappen”. Since there is no exact English translation, we use this term. The concept of trips legs is explained in chapter 3.2.3 of the MTMC.

Figure 4: Structure plausibility check of trip legs



3.3.1 Prepare data

Some preparation work is necessary to start the plausibility check of the trip legs.

3.3.1.1 Read data

First, we load the SAS trip leg data (etappen). Small adaptions are necessary. We replace the NA-values of the variable “schaetzdist” by -99, reformat the times and coordinates and delete the variable “stadt-land-2012” for start, end and domicile. This latter variable is not available in all data deliveries and will be re-added at the end of the plausibility check of the trip legs (see chapter 3.3.6.4).

➔ load_trip_data

3.3.1.2 Include manual corrections

In the next step we include the manual corrections of the trip data coming from the free texts. This ensures that the plausibility checks are also performed to the manual corrections too.

➔ manual_corrections_trip_legs

3.3.1.3 Delete not needed variables

We delete the variables which are not needed for further analysis:

- GIS_Etap_Zeit_min
- HAF_lDist

➔ delete_variables

3.3.1.4 Check LNDC

We check if all stages have a valid country code and unify the names and codes for Switzerland and Liechtenstein.

- LNDC 8100 = Schweiz
- LNDC 8222 = Liechtenstein

➔ check_LNDC

3.3.2 Decompose transborder trip legs

For the final analysis we have to be able to distinguish the distances made on Swiss ground from those abroad. The raw data includes distance, start and end point of the whole trip leg. Trip legs crossing the border have therefore to be split at the border point(s) (CH incl. LIE). If the trip takes place within the convex hull the data contain the repartition of kilometres abroad and in Switzerland.

This section thus does not only describe a pure plausibility check, but also how we split the trips crossing the border. If a trip leg crosses the border once we receive 2 trip legs, if twice 3 trips legs and so forth. For cases, where the trip ends or starts directly at the border, one trip leg gets subtracted. Trips crossing borders other than the Swiss (incl. LIE) are not relevant for this plausibility check. Hereafter the detailed steps of the decomposition of the trip legs are explained.

➔ decompose_transborder_trip_legs

3.3.2.1 Add indicator abroad

The trip legs are classified as follows:

Table 2: Indicator abroad

Indicator_abroad	Start	End
0	CH or LIE	CH or LIE
1	Abroad	CH or LIE
2	CH or LIE	Abroad
3	Abroad	Abroad

Trip legs carried out on the territory of Liechtenstein or the two foreign enclaves located in Switzerland, Campione d'Italia (Italy) and Büsingen am Hochrhein (Germany) are considered in exactly the same way as those carried out on Swiss soil only. Trips by plane are treated in a separate analysis, but are considered here as being entirely in Switzerland (Cat. 0), since we don't want to split those routes at the border points, for more details see section 3.3.5.9.

➔ add_indicator_abroad

3.3.2.2 Add existing border points

We add the existing border points from the verification and border point's dataset (routvfy) to the trip legs.

➔ add_existing_border_points

3.3.2.3 Add missing border points manually

Some border points are missing and get added manually. The missing border points come from trips which are crossing the border more than once and are identified in section 3.3.2.6.

➔ add_missing_border_points_manually

3.3.2.4 Add missing border points

Beside the manual adding of border points, we check if there is information about the border point in the Variable G1. If a trip has different start and end country, we conclude that there has to be at least one border point. If that is not the case, we calculate the border point by intersection the route with the Swiss border for the trip legs with a valid routing.

➔ add_missing_border_points

3.3.2.5 Decompose trip legs crossing the border once

This section describes the decomposition of trip legs crossing the border once. The aim is to receive two distinct trip legs - one entirely in Switzerland and the other entirely abroad, split at the border point.

First, we calculate the number of border points (function "get_max_nb_border_points"), add the information about the convex hull and then duplicate all trip legs crossing the border once based on the variable "indicator_abroad". We receive two identical data frames and mark one copy as abroad and one as in Switzerland.

Next, we replace the information about the start or end point of the trip leg - depending on the direction. For trips from Switzerland going abroad we replace the information of the destination in the copy marked as in Switzerland and replace the information about the origin in the copy marked as abroad. For trips legs from abroad to Switzerland vice versa. We also change the purpose for trip legs going to the border in the variable “f52900”. The variable purpose also effects other variables (e.g., type of leisure activity at destination for trip legs with purpose leisure). We change those values accordingly, for details consult the R code.

The values for distance are calculated using the ratio of the share of the trip in Switzerland and the whole trip. We distinguish between trips legs with (→ define_proportion_in_out_CH_with_routing) and without routing (→ define_proportion_in_out_CH_without_routing). Trips legs with routing are within the convex hull and thus, the proportion of the kilometres made in Switzerland is contained in the data. We can calculate the ratio directly.
For trips without routing, the distances of the parts made in Switzerland are calculated with help of the interzonal distance matrix of the VM_UVEK. If a trip leg is not within the VM_UVEK distance matrix we use the crow fly distance to calculate the ratio (→ add_ratio_using_crow_fly_distance).

Based on the calculated ratios we update the distance, duration and times of departure and arrival. For the trip legs abroad without routing (outside convex hull), the distances from the routing as well as the distances travelled on each type of road are set to 0. For the trip legs with routing and within the convex hull, we decompose the variable of distance by type of road proportionally to the distance in CH and abroad. This process should however be reconsidered for future implementations.

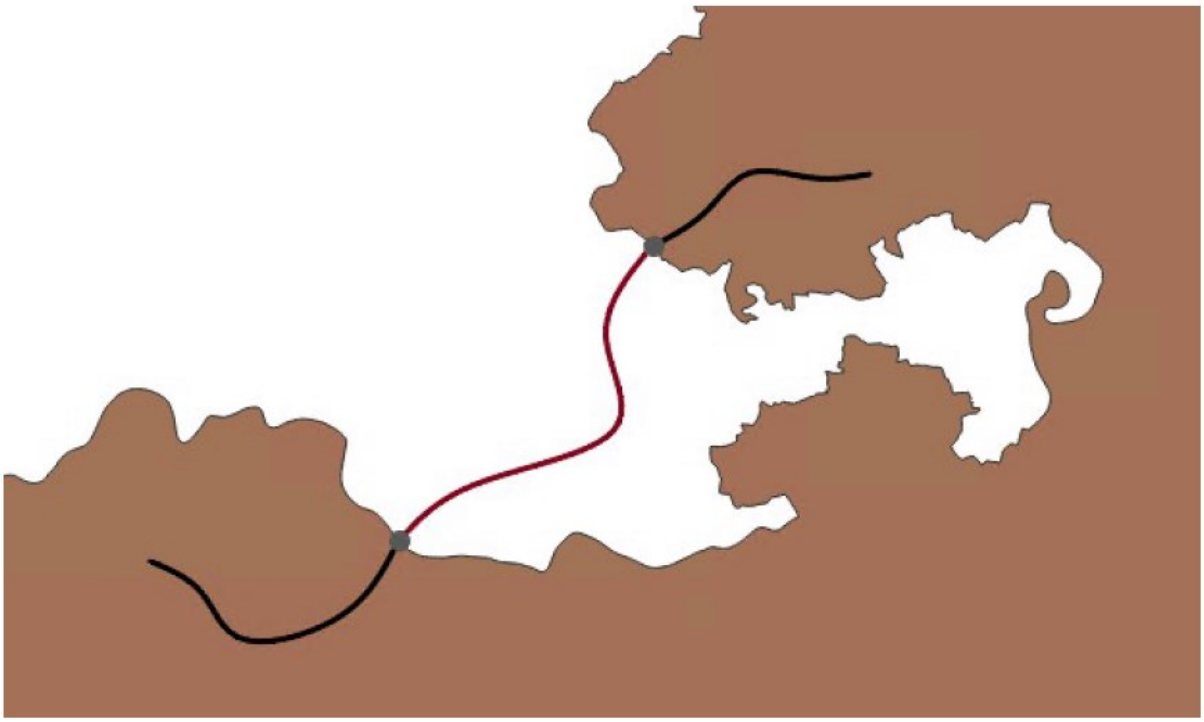
After that we add the information, we have about the border points to the splitted trip legs, we define the GIS calculation model (Transborder trip leg abroad (not in the convex hull), Transborder trip leg abroad (in the convex hull), Transborder trip leg in Switzerland), we change information about the shape files for trips abroad and outside the convex hull (Grenzetappe Ausland (not in convex hull)) update the country info because border points are often coded as being in Switzerland. For the part abroad, we assume that start LND is equal end LND for the trips going to the neighbor countries. After that the decomposed trip legs get re-added to the original data.

➔ decompose_trips_crossing_border_once

3.3.2.6 Decompose trip legs crossing the border more than once

For trips legs crossing the border more than once we cannot simply calculate the distance abroad as difference between the total distance and the distance in Switzerland, as in section 3.3.2.5. We instead cut the routes each time they are crossing the border (CH incl. LIE). Said that, this only applies to trip legs with a valid routing. Trips without routing are treated as in section 3.3.2.5. Thus, assuming just one border point.

Figure 5: Decomposition of trip leg crossing the border twice



In a first step we calculate the number of border points and load the needed data for the decomposition of the trip legs. These include:

- Border points
- Routes
- Swiss border incl. LIE

Next, we check whether a trip leg starts or ends at the border. Those trip legs are treated separately. Normally for a given number of border points we receive one more trip leg after the decomposition. E.g., for a trip leg with two border points we get 3 trip legs. In the case of trips starting at (or ending at) the border the number of trip legs corresponds to the number of border points. We apply a buffer of 5 meters to identify these cases.

To split the trip legs crossing the border more than once we use the software FME. We get back the length and number of the splitted legs and check whether the number of trip legs from splitting corresponds to the number coming from verification and border point data. The differences are automatically identified but have to be corrected manually. To ease that work, a shape file with the problematic routes is generated. There are several possible sources of error. However, the following are particularly worthy of attention: The number or order of the boundary points in “routvfy” is wrong, or the number or order of the split routes from FME is wrong. Special attention should be paid to the case when a route is identical to the border over a longer distance. It may be that in this case many border points are created (see also Figure 11).

Next, we create a data frame with the right number of copies per trip leg crossing the border several times. That means that for a trip leg crossing the border twice 3 copies or sub-legs are created. After that we start to change the relevant variables of those copies. We start with the first sub-leg and replace the information of the destination by the information about the border point. For the last sub-leg we replace the information of the origin. For the middle legs we replace the information about both, origin and destination.

As for trip legs crossing the border once, we also change the purpose for trip legs going to the border in the variable “f52900”. The variable purpose also effects other variables (e.g., type of leisure activity

at destination for trip legs with purpose leisure). We change those values accordingly, for details consult the R code.

Next, we calculate the routing and estimated distance “GIS_Etap_Dist_km” and “schaetzdist_plausibel” as the ratio of length of sub-leg to total length. This “ratio_length” we also apply to travel time. We calculate the travel time for each sub-leg by multiplying the length ratio with the total travel time. Then we start to add those times to the start time and replace the variable for travel time “e_dauer” by this new value. This approach is based on the assumption that there is no systematic difference for speed on the separate sub-legs (which of course is not necessarily true).

The variable "number of kilometres by type of road" (km_str) has generally two different definitions:

- For trips in a convex hull around Switzerland + a 20 km buffer around Switzerland: km for the whole trip
- For trips from Switzerland to further than the convex hull: km only in Switzerland

Since all trips crossing the border more than once have a routing, we calculate the kilometres per type of road proportionally.

Before re-merging the data with the trip legs, we replace the information about the GIS routing model and define the part made in Switzerland and the one abroad via the variable “LNDC”. The parts in Switzerland are coded as “8100” – Switzerland and the ones abroad as unknown “-99”. Finally, we delete all temporary variables and merge the trip legs crossing the border more than once with the other trip legs.

➔ decompose_trips_crossing_border_more_than_once

3.3.2.7 Delete variables after decomposition of trip legs

Since all trip legs crossing the border are now split and only trip legs purely in Switzerland (incl. LIE) or abroad exist we can delete the variables which contained the relevant information:

- GIS_km_CH
- GIS_km_Au
- GIS_km_CH

➔ delete_variables_after_decomposing_transborder_trip_legs

3.3.2.8 Re-numbering of trip legs

Finally, in the process of decomposing the trip legs crossing the border, a new numbering of the trip legs is introduced and is saved in the variable “etnr_NEW”. It takes into account the separation of the border legs and the effects of the manual corrections

➔ renumbering_trip_legs

3.3.3 Compute spatial variables

In this part we check whether the data on the FSO commune number (Section 3.3.3.1), the PLZ number (Section 3.3.3.2)3.3.3.2, the Nuts3 regions (Section 3.3.3.3) and the Degree of Urbanization DEGURBA (Section 3.3.3.4) are complete. We also check if both coordinates are zero, or if the coordinates are missing completely (Section 3.3.3.5).

3.3.3.1 FSO commune number

Different from 2015, missing FSO commune numbers are completed based on the coordinates for points with sufficient accuracy (variable QAL = 1, 2 or 3) in Switzerland. For that we use the file “Gemeindegrenzen”.

We still check whether there are any differences between the original data and the FSO commune number identified by the coordinates and check the differences manually. The code stops if the numbers to not correspond.

➔ fill_commune_number

3.3.3.2 PLZ Number (postcode)

In 2015, the postcodes were recalculated. In 2021 it was decided not to recalculate them because:

- there is no evidence that LINK delivered wrong data,
- the data are not crucial for the analyses and
- the geodata on postcodes are updated frequently.

We do however still check whether there are any differences between the LINK data and the PLZ numbers from the Swiss cadastral information system (based on Gemeindestand 2021). Differences are checked manually and in case of doubt the recoded values are taken. The code stops if the PLZ numbers are still deviating after the checks. The “lapply-Warning” is ok and we want that NA are generated.

➔ fill_plz_number

3.3.3.3 Add Nuts

We replace the NUTS 3 region data by the most up to date data based on the X-Y coordinates. One option, which is currently not applied, is to use the “giscor” package (https://gis.stackexchange.com/a/404388/6432). The R code however already contains the necessary adaptions.

Nuts3 is available for start and end points of the trip leg and for the place of living of the person (S_, Z_ and W_ - W stands for “Wohnort”, “place of living” in German). We recode the variable Nuts3 for all observation, except observations with NA values in their coordinates.

➔ add_NUTS

3.3.3.4 Add DEGURBA

We replace the codes of the degree of urbanization (DEGURBA) by the up-to-date data. DEGURBA is available for start and end points of the trip leg and for the place of living of the person (S_, Z_ and W_). We recode the variable DEGURBA for all observation, except observations with NA values.

➔ add_DEGURBA

3.3.3.5 Detect null coordinates

We check if X and Y coordinates of the starting or arrival points (variables S_X, S_Y, Z_X and Z_Y – Z stands for “Ziel”, “arrival” in German) are missing or both zero or almost zero (also known as “Null Island”). It might happen that there is a bug, instead of coding a missing value (NA or -99). If null coordinates are detected, the R code raises an error and the coordinates are manually corrected, taking the

context into account and the most accurate information available (centre of the municipality, canton, Nuts3 zones or country).

➔ detect_null_coordinates

3.3.4 Re-Routing

Free text can contain information about wrong or unreasonable routing. During the handling of those manual plausibility checks trip legs can be marked for re-routing. The aim of this process is not to replace the original routing in the shape-file, but to verify the distance. Thus, we replace the "wrong" routing distance by the re-calculated distance. This process is not handled directly in R, but in the software FME and the special routing-plug-in for the GIS-software MapInfo, called RouteFinder.

3.3.4.1 Export re-routing

In a first step we filter all data marked for re-routing and export separate files in csv format for:

- Street modes
- Train

For those modes of transport we have separate networks.

➔ export_re_routing

3.3.4.2 Re-routing in FME

As mentioned in the introduction of this section, the re-routing is not directly performed in R. For a detailed description of the re-routing in FME and RouteFinder.

3.3.4.3 Integrate re-routing

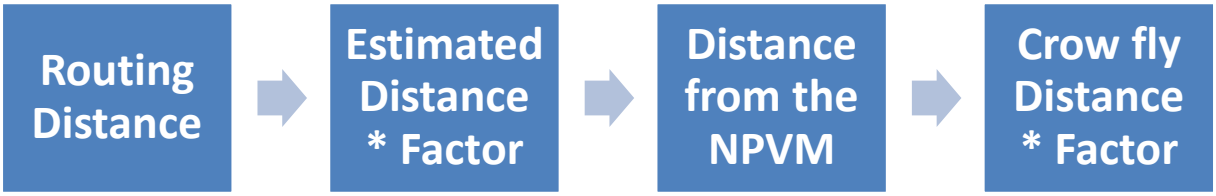
The results from the re-routing get finally re-merged with original data. Since we lose the information about how many kilometres have been made on each type of road "km_Str0 – km_Str8" we replace the information by "-99". We also delete the "mark" for re-routing and set the value of the variable "nachrout" to "2", which means: got re-routed.

➔ integrate_re_routing

3.3.5 Check distances

The aim of this section is to check whether the distances of the trip legs are plausible. Which is the most important part of the plausibility check. It is not exclusively a question of checking the plausibility, but of defining the correct distance per trip leg as precise as possible. If no plausible distance is available from the routing or the estimation, we calculate alternative distance measures. In the following we describe how we calculate these measures and in which case which measure is used (see Figure 6 and especially Section 3.3.5.9).

Figure 6: Defining final distance rdist



3.3.5.1 Compute crow fly distance

First, we recalculate of the variable “ldist”. This provides a consistent measure as the crow fly distance also takes into account the manual corrections made earlier.

We use the Haversine formula for the calculation of the crow fly distance. For the constant “Earth radius” R we use a value of 6371 km. δ is the latitude (in radians) and λ is the longitude (in radians).

Figure 7: Great circle distance by Haversine formula

$$D = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\delta' - \delta}{2}\right) + \cos\delta \cdot \cos\delta' \cdot \sin^2\left(\frac{\lambda' - \lambda}{2}\right)}\right)$$

➔ compute_crow_fly_distance (in utils_trip_legs)

3.3.5.2 Check routing distance

Next, we check whether the GIS routing distance is plausible.

First, we compare it to the newly calculated crow fly distance and identify the cases where the crow fly distance is smaller or equal the routing distance for the trip legs with a routing distance and save the info in the variable “plausible_routing_distance”.

Second, we verify the duration of the trip legs by calculating the variable “e_dauer” as the difference between departure- (f51400) and arrival-time (f51100). The duration is then used to calculate the average speed as the ratio between the routing distance (GIS_Etap_Dist_km) and the duration (e_dauer) multiplied by 60 to get kilometres per hour.

To verify the quality of the routing distance we compare the calculated speed with the speed thresholds per mean of transportation given in Table 3 (→ compute_plausible_routing_speed). Trips legs with speeds outside the defined thresholds do get assigned a FALSE value in the variable “plausible_routing_speed”.

Table 3: Speed limits per transport mode

Transport mode	Limit min (km/h)	Limit max (km/h)
Foot	0.2	30
(E-)Bike	0.2	40
Fast E-Bikes, Light Motorbikes,	0.2	60
Motorbike	0.2	200
Car, Taxi, Truck	0.2	200
Bus, Coaches Tram	0.2	150
Train	0.2	400
Boat	0.2	150
Plane	0.2	2000
Cable car	0.2	250
Vehicle-like devices	0.2	200
Others	0.2	250

Third, we create the variable “plausible_routing”. Trip legs with plausible routing, have a plausible routing distance, a plausible routing speed (see first and second) and have an estimated distance and a routing distance which is less than fifty times bigger than the estimated distance.

If there are any trip legs with a crow fly distance bigger than 5 metres and a routing distance bigger than thirty times the crow fly distance, the code stops and lists the problematic cases.

➔ check_routing_distance

3.3.5.3 Identify round trips

We identify round trips by comparing the coordinates of origin and destination. All trip legs which have either identical coordinates or a crow fly distance smaller than 5 metres are categorized as being round trips.

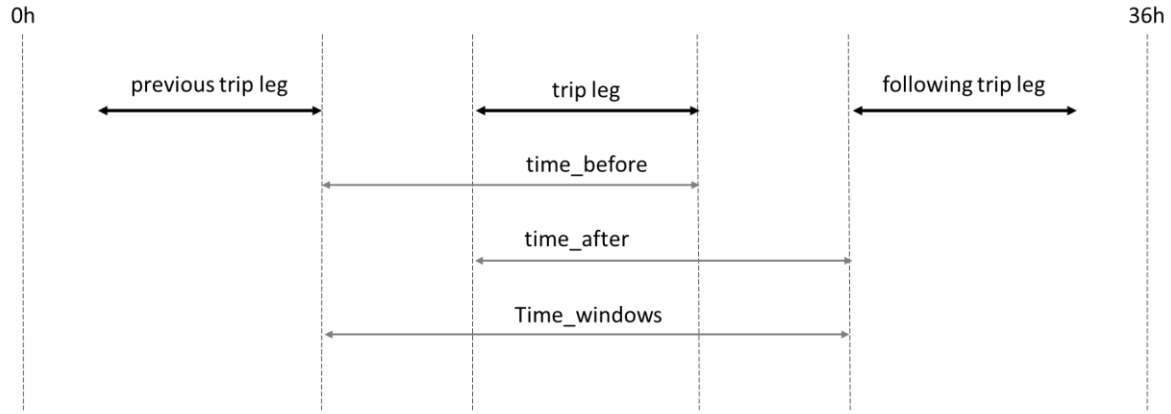
➔ add_variable_round_trips

3.3.5.4 Compute time windows

In later steps we adjust the duration of the trip legs for which we believe the duration is wrong. In order to do that, we first calculate the available time windows before and after the trip leg. We calculate the following time windows, as also shown in Figure 8:

- Time_before: Time interval between the time of arrival of the previous trip leg and the time of arrival of the trip leg.
- Time_after: Time interval between the start of the trip leg and the start of the next leg.
- Time_window: Time window in which the trip leg necessarily took place. This is the interval between the end of the previous trip leg and the beginning of the next.

Figure 8: Time windows concept



For this step it is important that the trip legs are ordered correctly.
If a target person only performs two trips the time windows correspond to the time before if there is only a previous trip leg and to the time after if there is only a following trip leg. In the case where there's only one trip the time window is limited by the start resp. end of the reference day (thus, 2160 minutes or 36 hours)

➔ compute_time_windows

3.3.5.5 Check estimated distances (create comparison distance)

We compare the distance from the national transport model (NPVM) with the routing distance for trips in Switzerland with precise start and end points. We first add the interzonal distance with the NPVM for trip legs which are not round trips, are not made by plane, have an estimated distance and a known mode of transport.

For each start and end point of the trip legs we add the zone numbers of the NPVM and check that all points correspond to a zone. Then we load the distance between zones on the road and merge it with the trip leg data. We do the same to get the public transport distances.

We save the distance from the NPVM in the variable "comparison_distance". If the distance is shorter than 5 km, we replace the comparison distance by the crow fly distance multiplied by a factor depending on the transport mode. The following multiplies are used:

Table 4: Multipliers for crow fly distance

Transport mode	Factor
Boat, Funiculaire	1.1
Walking, Light Motorcycle, Motorcycle, Bus, Tram/Metro, Taxi, Uber-like	1.2
Bicycles, Ebikes, Vehicle-like devices, Others, Car, Truck, Coach, Train	1.3

We define a temporary variable describing the origin of the plausible estimated distance: "schaetzdist_plausibel_source" (in 2015 it was named Ungleich_Distanzen_15):

- 0 = estimated distance was plausible and has not been modified
- 1 = estimated distance was not plausible and has been modified
- 2 = no estimated distance given

Next, we update the variable of the plausible estimated distance "schatzdist_plausibel".

If a trip was done by train, the estimated distance is plausible when it is between 0.8 and 1.7 times the comparison distance. If the estimated distance is not plausible, the comparison distance is used instead. If a trip leg was done by another transport mean, the estimated distance is plausible when it is between 0.8 and 10 times the comparison distance.

Again, if the estimated distance is not plausible, the comparison distance is used instead.

➔ check_estimated_distances

3.3.5.6 Check estimated speed

We also check the distances by comparing the routing distance with the speed calculated on the basis of the estimated distance. The duration is used to calculate the average speed as the ratio between the estimated distance (schaetzdist_plausibel) and the duration (e_dauer) multiplied by 60 to get kilometres per hour (→ compute_speed).

Then we check whether the calculated speed is plausible based on the thresholds in Table 3.

➔ check_estimated_speed

3.3.5.7 Set back estimated distance

If a trip leg has no plausible routing distance and the speed (based on estimated distance) is not plausible, we replace the plausible estimated distance by the original estimated distance. We simultaneously update the temporary variable describing the origin of the plausible estimated distance:

- 0 = estimated distance was plausible and has not been modified
- 1 = estimated distance was not plausible and has been modified
- 2 = no estimated distance given

In an earlier version of the plausibility checks we here recalculated the duration based on average speed per transport mode. We changed that and this step is executed later (see → final_check_rdist).

➔ set_back_estimated_distance

3.3.5.8 Check distances of round trips

The distances for round trips are checked based on the plausibility of speed calculated on the basis of the estimated distance. The procedure corresponds roughly to the one described in the previous sections. We proceed as follows:

1. We identify the round trips
2. We calculate the speed
3. We identify the trip legs with implausible speed
4. We recalculate the estimated distance for trip legs with implausible speed (on the basis of duration and average speed per mode of transportation)
5. We recalculate the duration based on the average speed and the estimated distance (see also Section 3.3.5.9)
6. We adjust the start and end times using the available time windows. After each adaption we re-(re-)calculate the time windows since those could have been changed.

Further details can be found directly in the code.

➔ check_distances_round_trips

3.3.5.9 Check trip legs by plane

We start by identifying the trips by plane and check manually missing distances. All trip leg distances by plane which are shorter than the distance as the crow flies are replaced by the distance as the crow flies multiplied by the factor 1.1.

For the distances which exceed 2 * distance as the crow flies, we also use the distance as the crow flies multiplied by 1.1 instead.

After that, we apply the comparable procedure as for the round trips (steps 4 – 6) and re-calculate the duration based on average speed.

Next, we calculate the speed again. All trip legs by plane with a speed below 100 km/h or faster than 2000 km/h we check manually. For now, we define all trip legs by plane as being 100% abroad if they cross the border.

➔ check_trip_legs_by_plane

3.3.5.10 Define the final distance

This function defines the final distance used for analysis. These values are saved in the variable “rdist”, which already exists, with NA values only. We also add a variable with the source of the final distance (rdist_source”. Generally, we distinguish the following cases (see also Figure 6):

1. If the routing distance is plausible, “rdist” takes the value of the routing distance

2. If the routing distance is not plausible, but the estimated distance is plausible, “rdist” takes the value of the estimated distance multiplied by a factor (depending on trip distance)

3. if the routing distance and the estimated distance are not plausible, “rdist” takes the value of the distance from the NPVM

4. And if there are no NPVM zones for the trip leg the crow fly distance multiplied by a factor is used.

We finally update “rdist” in the original table and test if “rdist” has been defined for all trip legs (that are not pseudo trip legs). The remaining cases we check manually. Hereafter we give some further details about the points two till 4 in the list above.

In the case 2 (routing distance is not plausible, but the estimated distance is plausible) we use the correction factors in Table 5.

Table 5: Correction factors for estimated distances

Routing distance (in km)	Factors individual modes	Factors public transport
< 1km	1.01	0.97
>= 1km & < 3km	0.91	0.94
>= 3km & < 10km	0.87	0.88
>= 10km & < 20km	0.89	0.8
>= 20km	0.98	0.94

We use correction factors since it was noticed that the estimation distances were systematically wrong, in comparison with the routing distances. Correction factors were calculated to keep comparability with 2005.

In the case 3 (routing distance and the estimated distance are not plausible) we take the comparison distance as final distance. In the function → check_estimated_distances (see Section 3.3.5.5) we replaced directly the plausible estimated distance for the trip legs with implausible values. Thus, in this function we only add the indicator where the “rdist” comes from.

In the case 4 (no comparison distance available) we take the crow fly distance multiplied with a factor as final distance. We use the factors shown in Table 4.

If final distance (rdist) is smaller than the crow fly distance (ldist), we also use “ldist” * factor instead, except for short trips with bad quality. Further, we replace the distance of all trip legs below 25m meters with “rdist” equal to 25 meters, otherwise they would not fulfil the official definition of a trip leg (they have to be at least 25m).

After that, we check if the final distance is defined for all trip legs. If not, the code stops. We have to check the remaining cases manually. If there is an estimated distance in “f51500” which seems plausible after a manual check, we take this distance (without any correction factor) as final distance.

➔ define_final_distance

3.3.5.11 Derivation of trip legs

For trip legs with public transport and an outward and return trip we assume the distances of these trip pairs to be similar. For rail bound services we check all trips legs with a deviation of more than 20% manually and for the road bound services we use 50% as the threshold value for allowed deviations. All trip legs with a higher deviation we check manually and decide which of the two distances is the “correct” one and impute this value. Use the file which got saved for the comparison purpose in the folder “verifications”.

➔ derivation_trips_pt

3.3.5.12 Final check of rdist

First, we check if “rdist” has been defined for all trip legs (that are not pseudo trip legs) and if there are cases with a crow fly distance smaller than the final distance.

After that, we compute the speed again and check whether now it is plausible. If not, we alter the duration and then the departure and arrival times. Thus, we replace the duration, assuming an average speed per mode whereas the duration cannot be longer than the time window between the previous and the next trip leg (see also Section 3.3.5.4). The duration is then computed based on average speed and rounded to the minute. We use the average speeds in Table 6.

Next, we update start and end times, taking into account the new trip duration and re-recompute speed. If still implausible, we check the identified trip legs manually.

Table 6: Average speed per transport mode

Transport mode	Average Speed* (km/h)
Walking	4
Bicycle, Slow E-Bike, Light Motorbikes	20
Fast E-Bike	30
Small Motorcycle	24
Motorcycle	29
Car	34
Truck, Coach	45
Taxi, Uber-like	30
Bus, Tram	21
Train	63
Boat	17
Plane	600
Other Public Transport (funicular, cable car, chair lift, ski lift, etc.)	16
Vehicle-like devices (Skateboard, trottinette, etc.)	7
Others	17

*Values come from MTMC 2015

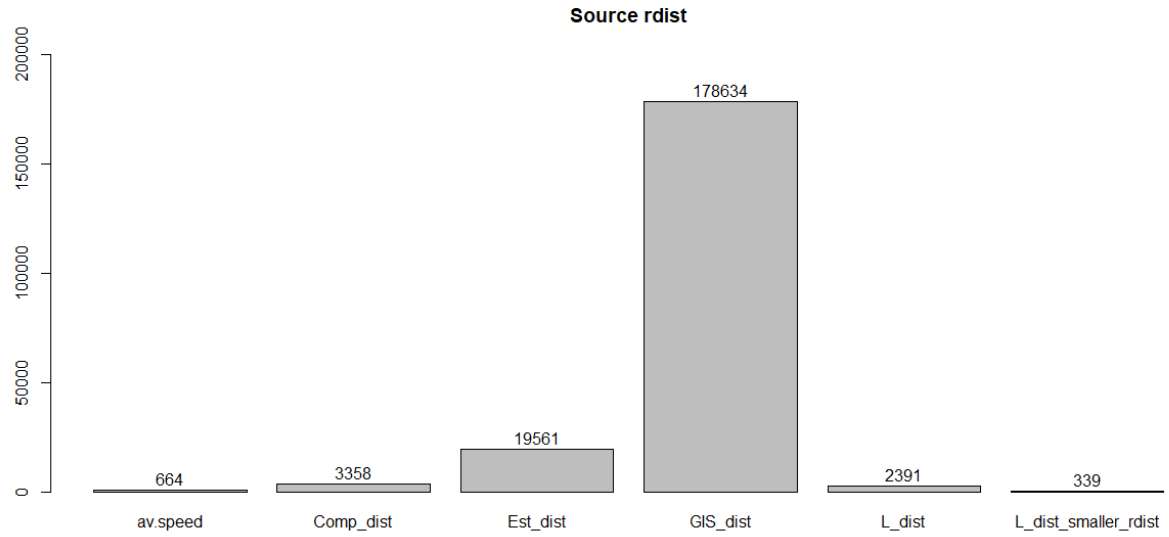
The impact of mayor duration changes (> 30min) is checked manually. We focus on the following aspects:

- Rdist > 8 km for walking trips
- Difference in duration is bigger than 120min for car trips
- Trip legs with one or more border points and more

Big differences in the duration can be a hint that the coordinates are wrong. It is important to check if there is a place with a similar name in the vicinity. If this place can be reached in the originally specified duration, the coordinates (and all other information) should in this case be adjusted manually.

Next, we check whether start and end times of consecutive trip legs do not overlap. And we check the trip legs with an especially high ratio of “rdist” and “ldist”. As well we plot the distribution of “rdist” and the distribution of the different sources of “rdist”. The latter can be found in Figure 9. Measured by the total number, the final distance comes from a plausible routing in a good 87% of all trip legs, and from a plausible estimate of the target in just under 10%. All other sources occur much less frequently.

Figure 9: Distribution of rdist source, 2021



Last, we check the longest trip legs manually in order to identify outliers.

➔ final_check_rdist

3.3.6 Post-processing

3.3.6.1 Manual corrections post processing

Some minor adaptions are performed, concerning locations in lakes, wrong PLZ/BFS numbers. Those overwrite the systematic checks and adaptions and are not checked again.

➔ manual_corrections_post_processing

3.3.6.2 Create final shape file

In the course of the data cleaning process, implausible data is discarded. If the adjustments had an impact on trip legs with routing information and the final distance (rdist) does not match the original routing distance (GIS_Etap_dist_km), we delete the routing information.

This also not only the trip legs dataset (etappen) but also:

- The shape file with the geo-information oabout the routes
- The verification and border point file (routvfy)
- The segments file.

➔ delete_routing_info

3.3.6.3 Recode Road class

This function makes sure that rdist corresponds to the sum of the distances on the different road classes. Differences below 1m are ignored.

➔ recode_km_str

3.3.6.4 Apply Stadt-Land typology

The variable “stadt_land” has not been part of all data deliveries by the FSO and LINK. That’s why we suppress the data at the beginning of the plausibility check and re-add it at the end. The variable “stadt-land” can have three different values (city- intermediate – rural) and is determined per municipality. The variable is available for the origin, destination and the domicile. The typology with “Gemeindestand” 2021 is used.

➔ add_stadt_land

3.3.6.5 Add Travel times to centres

The travel time to the 6 major centres Basel, Bern, Geneva, Lausanne, Lugano and Zurich on the road and public transport network are added. From each person’s place of living (W_), the travel time to the fastest accessible centre is determined. We aggregate the durations in 4 categories.

➔ fill_ISO_dist

3.3.6.6 Recalculate times

During the plausibility check we just worked with the time variables in the unit seconds after midnight. This is easier to handle for arithmetic operations. Finally, we update the relevant variables in the format hh:mm:ss.

➔ recalculate_times_in_hours

Finally: Save the data in the results folder with the ending "_plausi". That's the file which is then sent back to the FSO.

3.4 Households

This data set contains information on place of residence, possible secondary residences, the ownership of cars, motorbikes and bicycles and income. The ARE complements respectively recalculates the public transport quality at the place of residence and updates these values in the data set. For this, we overlay the coordinates and the public transport quality classes and thus identify the corresponding quality class.

➔ fill_public_transport_quality

3.5 Target person data ("Zielpersonen")

This data set contains information about the target person itself, such as: level of education, work status and place, education and availability of mobility tools. In this section we also include the manual corrections indicated in the free text concerning target person.

3.5.1 Add public transport quality

Also, for this data set we recalculate the values for the public transport quality for the education and work place. For more details, see section above.

➔ fill_public_transport_quality

3.5.2 Check whether target person is mobile

If there are no records on trip legs on the reference day a target person was not mobile that day. It is possible that in the free texts a missing trip leg is indicated. A person thus becomes mobile. Also, the opposite is possible. The code updates the target person data accordingly.
It adapts the variables "f50200", "f50600a", "f50600b" and "f50600c" in case of adding the first resp. deleting the last trip leg. For that, we compare the trip legs data before and after the plausi check.

➔ correct_is_mobile_reference_day

3.6 Day trips

This dataset contains the day trips. A day trip means that someone leaves his or her familiar environment for at least three hours (incl. stay) without staying overnight. Day trip departure and arrival coordinates are approximated according to the most accurate information available at the level of the centroids of the municipalities or Nuts3 zones or countries. Up to three stopovers were recorded.

The plausibility checks for the data sets day trips and trips with overnight stays are similar. In this documentation we describe the detailed process for the day trips only. The respondents have to estimate the distance of the whole trip. The estimated distance is compared with a comparison distance, if the estimated distance is implausible, the comparison distance is used instead. After the plausibility checks we re-code the Nuts-Code.

➔ run_daytrips

3.6.1 Preparation

First, we load the day trip and household data, we integrate the manual corrections and check if the coordinates of the start and end points are valid.

The following two variables are mainly relevant (the variable names in brackets are the ones corresponding to the trips with overnight stays):

- f61600 (f71600): is the total distance estimated by the person
- f61700 (f71700): is the distance estimated by the person on Swiss ground

3.6.2 Check estimated distance

We check also for the day trips whether the total estimated distance is plausible. Before that, we however have to handle the cases with missing crow fly distance (or ldist = 0). We distinguish the following cases and proceed as follows:

- Start point (TRS) not equal end point (TRZ) → We calculate ldist again as ldist between start and end without intermediate points
- End point (TRZ) is "correct" and TRS should be home → We replace TRS by the home address
- TRZ is wrong but there is the necessary info in "Grobziel" -> We take TRS home and TRZ "Grobziel"
- If none of the above is the case, we check the remaining cases manually. If no information about the possible origin and destination can be found, we delete the day trip.

The respondents have to estimate the distance of the whole trip (meaning the distance of the outward journey, the return journey and the sum of the distance of the journeys at the destination). The estimated distance is compared with a comparison distance. This comparison distance ($d_{comp)}$ is calculated as:

$$d_{comp} = d_{asthecrowflies} * 2 * 1.1$$

We use the distance as the crow flies between the starting point and the end point if there are no intermediate stops. If there are intermediate stops, the distance as the crow flies is equal to the sum of the distances as the crow flies between the subsequent points. The variables “f60500” and “f60550” indicate whether there are intermediate points for the trip. The comparison distance is to be understood as a lower bound and a conservative measure of the distance made.

- The value used for the lower limit is: $0.7 * d_{comp}$
- For the upper limits we use: $6 * d_{comp}$

If the estimated total distance is outside the limits, it is replaced with d_{comp}. All journeys with implausible distance, on the road, in Switzerland and with no via Points were re-routed on the TomTom network (see also Section 3.3.4).

➔ check_total_distance_TR

3.6.3 Check distance in Switzerland

Next, we check the plausibility of the distance on Swiss territory. For trips that took place entirely on Swiss territory, only the variable “f61600” is recorded. Only trips with at least one of the points abroad have a value in variable “f61700”.
For this check, we calculate the comparison distance (d_{comp}) for part in Switzerland based on the NPVM. As for the decomposition of the trip legs in “in Switzerland” and abroad, we define the ratio CH - abroad (ratio comes from NPVM if start and end in NPVM-zones, otherwise ratio of crow fly distance split at border is taken). Within the following limits we believe that the estimated distance (d_{est}) is plausible.

$0.5 * d_{comp} <= d_{est} <= 1.5 * d_{comp}$

If the estimated distance is not within these limits, we replace it by the comparison distance.

We further check if there are trips with the distance in Switzerland being bigger than the total distance. For those we assume that the start point has been set incorrectly, thus, replace it by the home address and re-perform the above described steps. Day trips with still implausible distances we check manually.

➔ check_distance_CH_TR

3.6.4 Check speed

Comparable to the process for the trip legs, we compute the variable plausible speed on the basis of predefined speed limits (we use the limits in Table 3). If the calculated speed is implausible, we change the duration by multiplying the distance by average speed (see also Table 6). The trips which are still implausible after this correction, we check manually.

➔ check_speed_TR

3.7 Trips with overnight stays

This data set contains trips of more than one day. Trips with overnight stays are trips in which at least one overnight stay is away from home (regardless of distance travelled). Not taken into account are regular (once or several times a week) repetitive trips. As with the table "Day trips", the departure and arrival coordinates of the journey are approximated on the level of the centroids of the municipalities or the Nuts3 zones or the countries according to the most accurate information available.

Since the plausibility checks for the data sets day trips and trips with overnight stays are similar, we describe the whole process for the day trips only. For details, please consult the previous section (see 3.6). Hereafter, we highlight the differences. One difference is that the variables do start with “f7xxx” compared to “f6xxx” in the day trips.

Also, for the day trips the estimated distance is compared with the comparison distance. For the trips with overnight stays we however apply slightly different threshold values:

- The value used for the lower limit is: $0.8 * d_{comp}$
- For the upper limits we use: $4 * d_{comp}$

In contrast to the day trips, there is no need to check the speed limits here, as there is no record on travel times. All other applied values correspond to the ones used for the plausibility checks of the day trips.

4 Manual plausibility checks

Manual plausibility checks and corrections are inevitable and necessary. We distinguish two main types of manual data processing:

- necessary modifications indicated in the free texts and
- filtered observations which are still implausible after the systematic quality control.

4.1 Free texts

At the end of the survey, the target persons can leave remarks and also the interviewers can address open points or issues in the form of free texts. The free texts are analyzed manually and based on that it is decided whether manual changes are necessary. In 2021, all manual correction coming from the free texts have been documented in the function → manual_corrections_trip_legs. E.g., for a new trip leg all available variables had to be added manually. For that we copied an existing observation and changed the necessary fields manually. The documentation directly in R was however cumbersome.

4.2 Manual corrections of implausible data

Wrong coords vs wrong duration: For trip legs remaining implausible after the plausibility checks normally either the duration or the cords are wrong. Thus, those have to be checked manually. Particularly, we have to check whether two cities or villages with similar or equal names have been confused. If the estimated distance and “ldist”
differ a lot and there exists a Village with the “same” name close to the border, it can be assumed, that this closer one has been meant. Common examples are:

St-Julien -> Saint-Julien-en-Genevois
Saint-Louis -> St.Louis
Singen -> Singen (Hohentwiel)
Lindau -> Lindau (Bodensee)
Waldshut -> Waldshut-Tiengen
Volksbourg -> Folgensbourg
Corsier -> Corsier-sur-Vevey
Widnau -> Wittnau
Biel -> Biel/Bienne
Busswil -> Busswil BE
Birmensdorf -> Birmenstorf (is difficult to detect, since not much apart

5 Conclusions and Suggestions

Consistency in the time series is one of the objectives of this process. In order to be able to compare the results, we cannot drastically change the data-cleaning method from one MTMC to the next. It is also necessary to consider the relationship between effort and improvement of data quality.
On the other hand, the MTMC survey method has been continuously updated and each improvement needs a new approach to data-cleaning.

Some challenges were already present in 2015. For example, when recording a trip on a highway, the interviewer might click on the wrong side of the highway. The routing tool then computes the shortest path going through this point, generating a long route including leaving the highway, taking it again in the other direction in order to go through the verification point and then leaving it again and changing direction again (see Figure 10). Interviewers were made aware of this problem at the beginning of the 2020/2021 MTMC in order to limit this undesirable phenomenon.

Figure 10: Wrong verification point

In 2020/2021 the MTMC features for the first time a routing tool allowing routing abroad, in a zone close to the Swiss border (convex hull of Switzerland plus 20-kilometre buffer). This created new challenges. Until 2015, interviewees crossed the border in one specific point, since the question was “where did you cross the border”. In 2020/2021 trip legs with several border crossings are possible. This asks for new geodata-cleaning approaches (see Section 3.3.2). Additionally, due to data imprecisions of roads and the border, some routes following the border generate many border points and many very small segments of routes.

Figure 11: Multiple border points

Geodata-cleaning: Too much or too little?
It is difficult to define the right amount of data-cleaning and it will always be a challenge to automatically identify the wrong observations and to distinguish them from rare and special cases. Although we try to keep the same methodology in the plausibility checks to allow for comparable time series, we try to incorporate new approaches, findings and developments.

Geodata-cleaning is also quality control
Verification of the chosen routes, distances and times directly in the interview are important and allow for follow-up questions. In this way, errors can be efficiently avoided and, at the same time, special and rare cases can be validated by the respondent herself.
The examination of the data during the survey year allows the early detection of possible systematic errors. In this way, problems in the routing tool or in the questionnaire can be identified. Thus, the focus of this documentation, the post-interview plausibility check, forms a final check to find and correct the observations that are still implausible. It will not only improve the final data, but also the way we design the next MTMC in 2025. Finally, comparing aggregated statistics of 2021 with previous MTMCs and with other data sources will be yet another step in the quality control process.