ETH zürich

The Roadmap for Mobility Behavior Analysis: From Surveys to Travel Choices and Accessibility

Dr. Anna S. Reiffer
Transportation and Mobility Planning
Institute for Transport Planning and Systems

IVT Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

tmp Transportation and Mobility Planning

1

The Basics

2

Interdisciplinarity of Transport Research

3

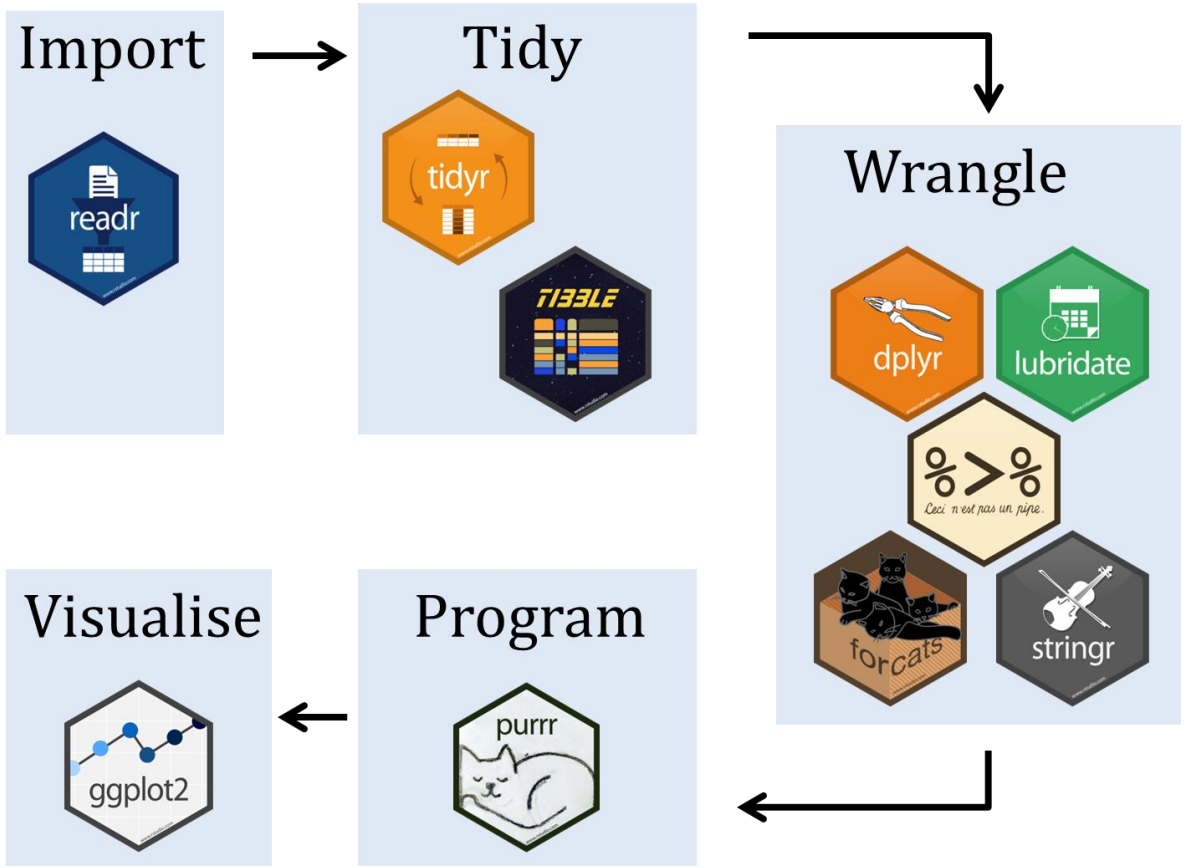
Choice Modeling

4

OpenStreetMap & Routing

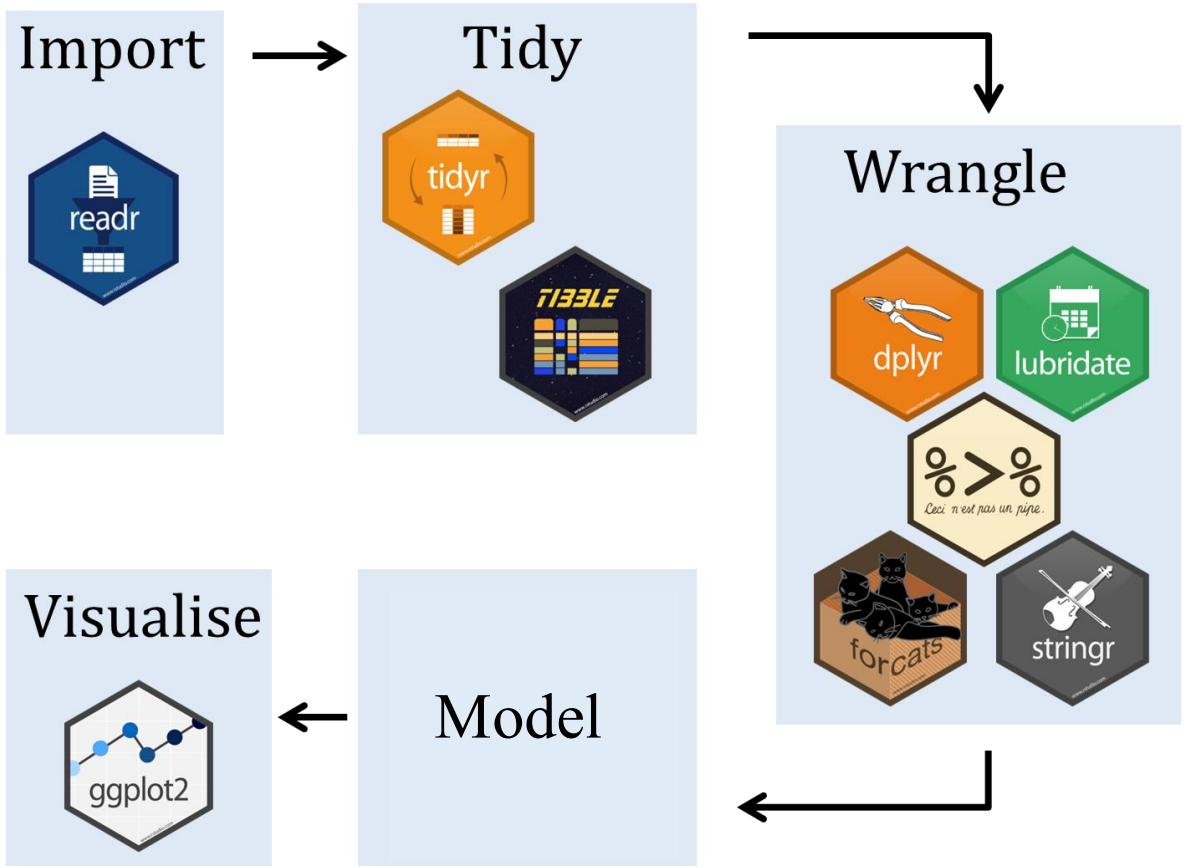
The Basics

The Basics



Source: AJ Smit, Amieroh Abrahams & Robert W Schlegel **Intro R Workshop: Data Manipulation, Analysis, and Visualisation** (https://ajsmit.github.io/Intro_R_Official/index.html)

The Basics



Source: AJ Smit, Amieroh Abrahams & Robert W Schlegel **Intro R Workshop: Data Manipulation, Analysis, and Visualisation** (https://ajsmit.github.io/Intro_R_Official/index.html)

The Basics

Household travel survey data

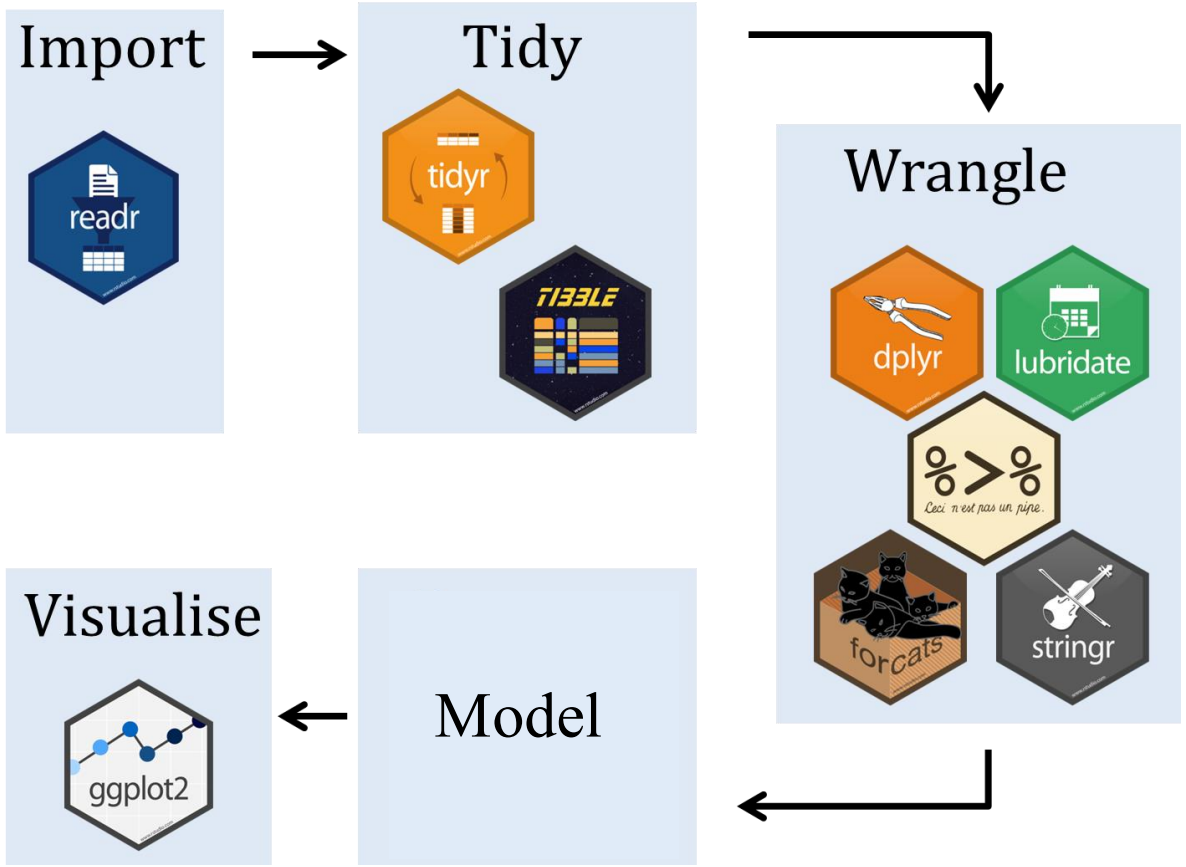
Household data
- household size
- Number of children
- Household income
- Number of cars
…

Match by household-id

Person data
- Age
- Gender
- Employment
- Mobility tools
…

Match by person-id

Trip data
- Start & End (coord.)
- Start & End (time)
- Mode
…



Source: AJ Smit, Amieroh Abrahams & Robert W Schlegel **Intro R Workshop: Data Manipulation, Analysis, and Visualisation** (https://ajsmit.github.io/Intro_R_Official/index.html)

The Interdisciplinarity of Transport Research

The Interdisciplinarity of Transport Research

- Transport is an inherently interdisciplinary field

- We borrow methods from various other disciplines, including geography and geosciences, environmental sciences, sociology, physics & computer science.

The Interdisciplinarity of Transport Research

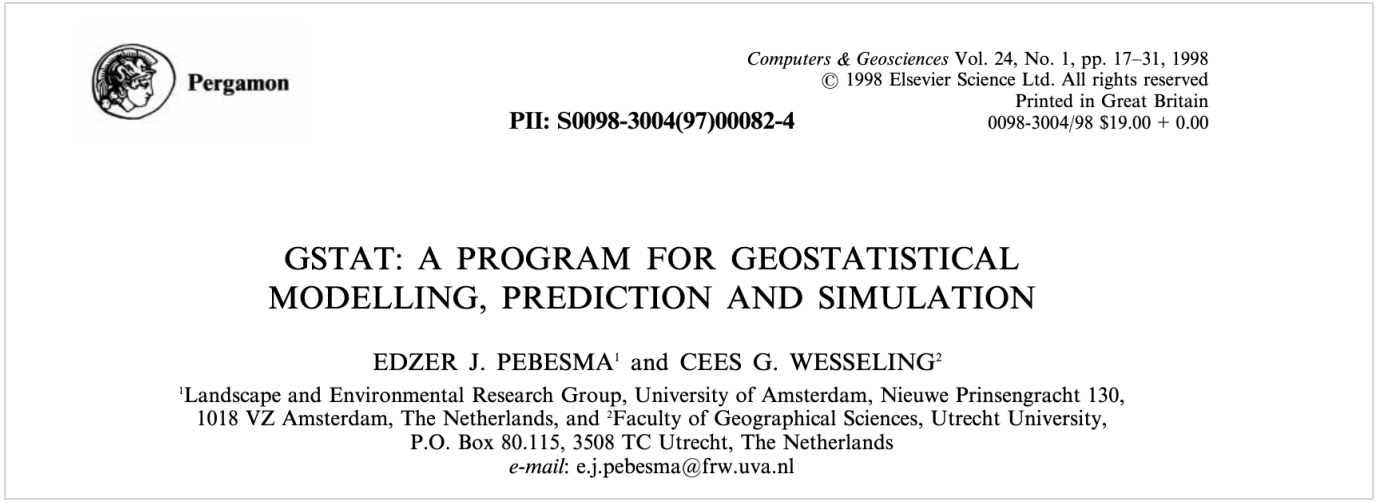
Example: Impact of Extreme Heat on Intra-Person Changes in Walking and Cycling Behavior

- **Data**
  - Travel Behavior: Longitudinal GPS-based travel diary (TimeUse+) beginning 17 July 2022.
  - Temperature: Spatial temperature distribution through kriging.
- **Exposures**
  - Tropical night: daily minimum temperature ≥ 20 °C
  - Heat stress: maximum daily temperature ≥ 35 °C
- **Outcomes**
  - Number of walking and cycling trips
  - Distance (km per day) traveled for walking and cycling

The Interdisciplinarity of Transport Research

Example: Impact of Extreme Heat on Intra-Person Changes in Walking and Cycling Behavior

- **Data**
  - Travel Behavior: Longitudinal GPS-based travel diary (TimeUse+) beginning 17 July 2022.
  - **Temperature: Spatial temperature distribution through kriging.**



Pergamon

Computers & Geosciences Vol. 24, No. 1, pp. 17–31, 1998

Printed in Great Britain
PII: S0098-3004(97)00082-4
0098-3004/98 $19.00 + 0.00

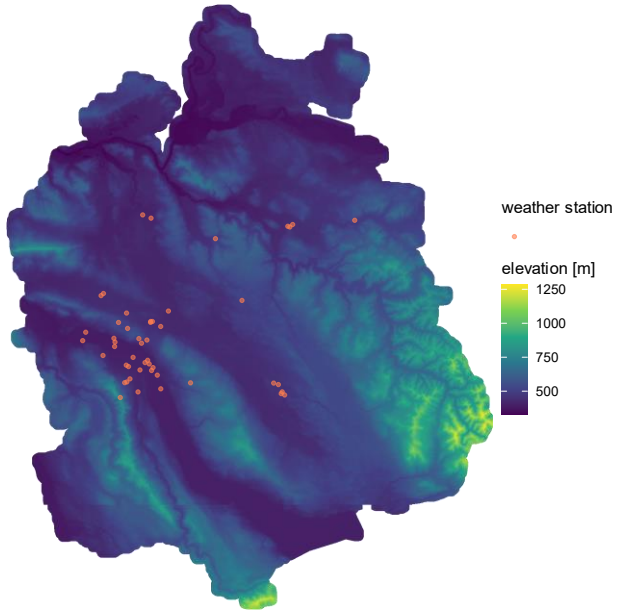
GSTAT: A PROGRAM FOR GEOSTATISTICAL MODELLING, PREDICTION AND SIMULATION

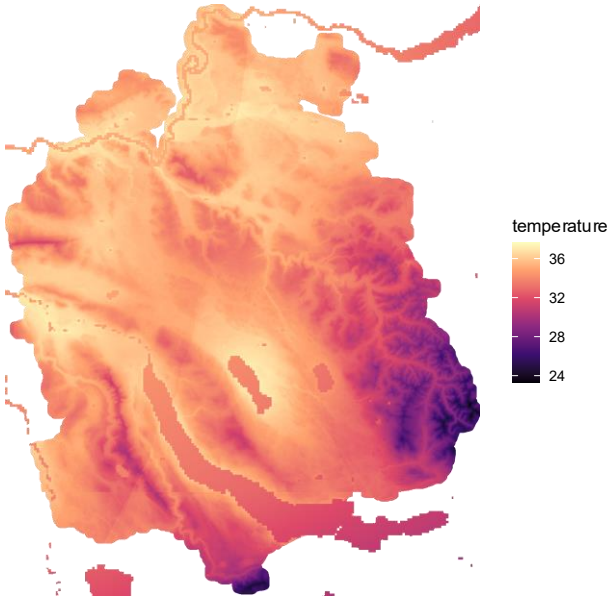
EDZER J. PEBESMA[1] and CEES G. WESSELING[2]
[1]Landscape and Environmental Research Group, University of Amsterdam, Nieuwe Prinsengracht 130, 1018 VZ Amsterdam, The Netherlands, and [2]Faculty of Geographical Sciences, Utrecht University, P.O. Box 80.115, 3508 TC Utrecht, The Netherlands
e-mail: e.j.pebesma@frw.uva.nl

The Interdisciplinarity of Transport Research

Example: Impact of Extreme Heat on Intra-Person Changes in Walking and Cycling Behavior

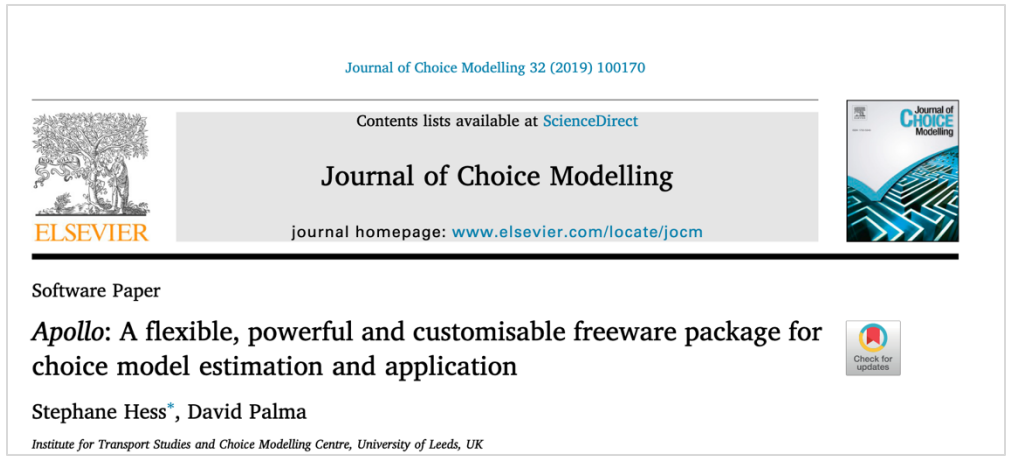
- **Data**
  - Travel Behavior: Longitudinal GPS-based travel diary (TimeUse+) beginning 17 July 2022.
  - **Temperature: Spatial temperature distribution through kriging.**





Choice Modeling

Choice Modeling



Journal of Choice Modelling 32 (2019) 100170

Contents lists available at ScienceDirect

Journal of Choice Modelling

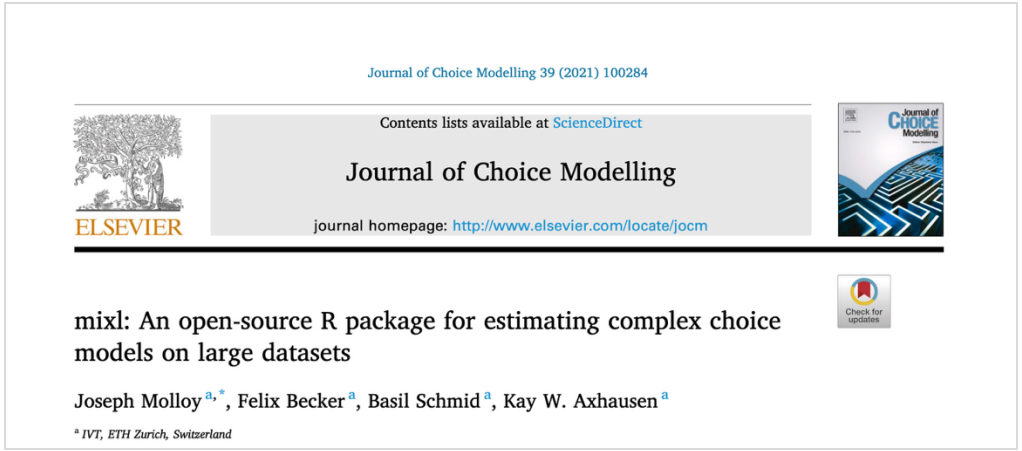
journal homepage: www.elsevier.com/locate/jocm

Software Paper

Apollo: A flexible, powerful and customisable freeware package for choice model estimation and application

Stephane Hess*, David Palma

Institute for Transport Studies and Choice Modelling Centre, University of Leeds, UK



Journal of Choice Modelling 39 (2021) 100284

Contents lists available at ScienceDirect

Journal of Choice Modelling

journal homepage: http://www.elsevier.com/locate/jocm

mixl: An open-source R package for estimating complex choice models on large datasets

Joseph Molloy[a,*], Felix Becker[a], Basil Schmid[a], Kay W. Axhausen[a]

[a] *IVT, ETH Zurich, Switzerland*

Choice Modeling using Apollo

More info on the package here: https://apollochoicemodelling.com/

- Given their own and mode characteristics, which mode will individuals choose?

	ID	av_car	av_bus	av_air	av_rail	time_car	cost_car	time_bus	cost_bus	access_bus	time_air	cost_air	access_air	service_air	time_rail	cost_rail	access_rail	female	income	choice
1	1	0	0	1	1	0	0	0	0	0	50	80	55	0	140	55	5	0	46705	4
2	1	0	0	1	1	0	0	0	0	0	70	80	45	0	170	45	20	0	46705	4
17	2	1	1	0	1	345	40	345	30	15	0	0	0	0	130	55	20	1	50123	2
18	2	1	1	0	1	345	45	345	25	20	0	0	0	0	130	45	10	1	50123	2
33	3	0	0	1	1	0	0	0	0	0	90	65	40	0	130	35	20	1	67589	4
34	3	0	0	1	1	0	0	0	0	0	60	110	40	0	170	35	15	1	67589	4

Choice Modeling using Apollo

More info on the package here: https://apollochoicemodelling.com/

- Given their own and mode characteristics, which mode will individuals choose?

```r
library(apollo)
# ################################################################# #
#### LOAD LIBRARY AND DEFINE CORE SETTINGS                       ####
# ################################################################# #

### Initialise code
apollo_initialise()

### Set core controls
apollo_control = list(
  modelDescr      = "Simple MNL model on mode choice RP data",
  indivID         = "ID",
  outputDirectory = "output"
)

# ################################################################# #
#### LOAD DATA AND APPLY ANY TRANSFORMATIONS                     ####
# ################################################################# #

### Loading data from package
### if data is to be loaded from a file (e.g. called data.csv),
### the code would be: database = read.csv("data.csv",header=TRUE)
database = apollo_modeChoiceData
### for data dictionary, use ?apollo_modeChoiceData

### Use only RP data
database = subset(database,database$RP==1)
```

Choice Modeling using Apollo

More info on the package here: https://apollochoicemodelling.com/

- Given their own and mode characteristics, which mode will individuals choose?

```
29  # ################################################################# #
30  #### DEFINE MODEL PARAMETERS                                     ####
31  # ################################################################# #
32
33  ### Vector of parameters, including any that are kept fixed in estimation
34  apollo_beta=c(asc_car   = 0,
35                asc_bus   = 0,
36                asc_air   = 0,
37                asc_rail  = 0,
38                b_tt_car  = 0,
39                b_tt_bus  = 0,
40                b_tt_air  = 0,
41                b_tt_rail = 0,
42                b_access  = 0,
43                b_cost    = 0)
44
45  ### Vector with names (in quotes) of parameters to be kept fixed at their start
46  apollo_fixed = c("asc_car")
47
48  # ################################################################# #
49  #### GROUP AND VALIDATE INPUTS                                   ####
50  # ################################################################# #
51
52  apollo_inputs = apollo_validateInputs()
```

Choice Modeling using Apollo

More info on the package here: https://apollochoicemodelling.com/

- Given their own and mode characteristics, which mode will individuals choose?

```r
# ################################################################# #
#### DEFINE MODEL AND LIKELIHOOD FUNCTION                        ####
# ################################################################# #

apollo_probabilities=function(apollo_beta, apollo_inputs, functionality="estimate"){

  ### Attach inputs and detach after function exit
  apollo_attach(apollo_beta, apollo_inputs)
  on.exit(apollo_detach(apollo_beta, apollo_inputs))

  ### Create list of probabilities P
  P = list()

  ### List of utilities: these must use the same names as in mnl_settings, order is irrelevant
  V = list()
  V[["car"]]  = asc_car  + b_tt_car  * time_car                          + b_cost * cost_car
  V[["bus"]]  = asc_bus  + b_tt_bus  * time_bus  + b_access * access_bus  + b_cost * cost_bus
  V[["air"]]  = asc_air  + b_tt_air  * time_air  + b_access * access_air  + b_cost * cost_air
  V[["rail"]] = asc_rail + b_tt_rail * time_rail + b_access * access_rail + b_cost * cost_rail

  ### Define settings for MNL model component
  mnl_settings = list(
    alternatives  = c(car=1, bus=2, air=3, rail=4),
    avail         = list(car=av_car, bus=av_bus, air=av_air, rail=av_rail),
    choiceVar     = choice,
    utilities     = V
  )

  ### Compute probabilities using MNL model
  P[["model"]] = apollo_mnl(mnl_settings, functionality)

  ### Take product across observation for same individual
  P = apollo_panelProd(P, apollo_inputs, functionality)

  ### Prepare and return outputs of function
  P = apollo_prepareProb(P, apollo_inputs, functionality)
  return(P)
}

# ################################################################# #
#### MODEL ESTIMATION                                            ####
# ################################################################# #

model = apollo_estimate(apollo_beta, apollo_fixed, apollo_probabilities, apollo_inputs)
```

- Given their own and mode characteristics, which mode will individuals choose?

Model Results:

Estimates:

	Estimate	s.e.	t.rat.(0)	Rob.s.e.	Rob.t.rat.(0)
asc_car	0.000000	NA	NA	NA	NA
asc_bus	0.474676	1.018672	0.4660	0.990600	0.4792
asc_air	1.629072	0.827435	1.9688	0.817513	1.9927
asc_rail	0.944534	0.803551	1.1755	0.795160	1.1879
b_tt_car	-0.003646	0.001555	-2.3455	0.001568	-2.3251
b_tt_bus	-0.008846	0.002624	-3.3711	0.002616	-3.3815
b_tt_air	-0.020685	0.006599	-3.1346	0.006424	-3.2202
b_tt_rail	-0.011239	0.004386	-2.5626	0.004468	-2.5157
b_access	-0.011466	0.006430	-1.7832	0.006297	-1.8210
b_cost	-0.033947	0.003295	-10.3035	0.003180	-10.6755

Choice Modeling using Apollo

More info on the package here: https://apollochoicemodelling.com/

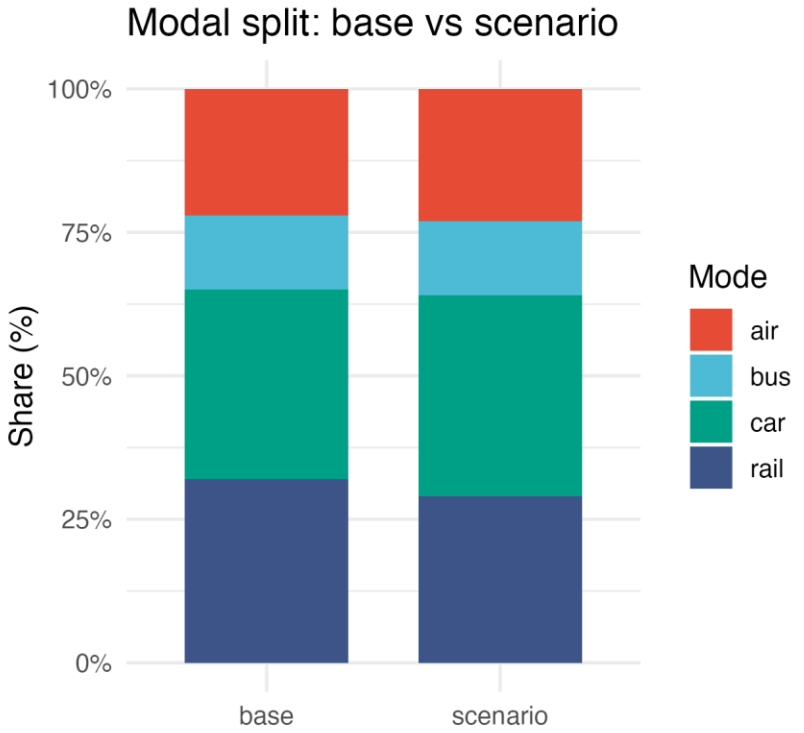
- Model predictions:

```
Running predictions from model using parameter estimates...
Prediction at user provided parameters
             car    bus    air   rail
Aggregate 332.00 126.00 215.00 327.00
Average     0.33   0.13   0.21   0.33
```

- Scenario: Increased rail costs

```
### Now imagine the cost for rail increases by 1%
database$cost_rail = 1.1*database$cost_rail
### Rerun predictions with the new data
apollo_inputs = apollo_validateInputs()
predictions_new = apollo_prediction(model, apollo_probabilities, apollo_inputs)
```

```
> predictions_new = apollo_prediction(model, apollo_probabilities, apollo_inputs)
Running predictions from model using parameter estimates...
Prediction at user provided parameters
             car    bus    air   rail
Aggregate 349.17 133.87 226.19 290.77
Average     0.35   0.13   0.23   0.29
```



OpenStreetMaps and Routing

Downloading OpenStreetMap Data with osmextract

```r
#canton network
zurich_lines = oe_get("Zurich", download_directory = "./data")
#clip network to city boundaries in terminal:
#osmium extract -b 8.47,47.32,8.60,47.42 openstreetmap_fr_zurich-latest.osm.pbf -o zurich_city_clip.osm.pbf

z_lines <- oe_read("./data/zurich_city_clip.osm.pbf")

z_network <- z_lines |>
  filter(highway %in% c("motorway",
                        "primary", "primary_link",
                        "secondary", "secondary_link",
                        "tertiary", "tertiary_link")) |>
  ggplot() +
  annotation_map_tile(type = "cartolight", zoom = 13) +
  geom_sf(color = "navyblue", alpha = .6) +
  theme_void()
```

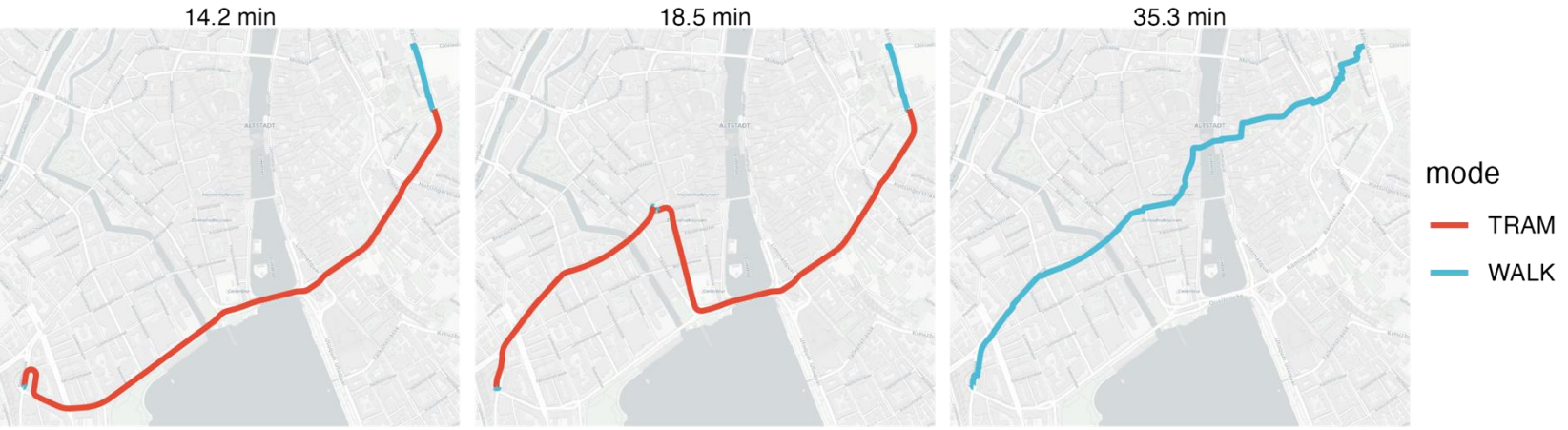
Routing with r5r

- Necessary input:
  - OSM network (.pbf file) → mandatory
  - GTFS data (.zip file) → optional
  - Digital Elevation Model → optional

```r
options(java.parameters = "-Xmx8G")

library(r5r)
library(data.table)


r5r_network <- r5r::build_network(data_path = "./data")

# extract OSM network
street_net <- r5r::street_network_to_sf(r5r_network)

# extract public transport network
transit_net <- r5r::transit_network_to_sf(r5r_network)

poi <- data.frame(
  id = c("UZH", "ETH_poly", "ETH_hoengg", "HB", "BHF_enge"),
  lat = c(47.37472, 47.376174, 47.408259, 47.37753, 47.36422),
  lon = c( 8.54932,  8.546599,  8.507384,  8.54198,  8.53140)
)

# set inputs
origins <- poi[1,]
destinations <- poi[5,]
mode <- c("WALK", "TRANSIT")
max_walk_time <- 60 # minutes
departure_datetime <- as.POSIXct("09-10-2025 21:00:00", format = "%d-%m-%Y %H:%M:%S")

det <- detailed_itineraries(
  r5r_network,
  origins = origins,
  destinations = destinations,
  mode = mode,
  departure_datetime = departure_datetime,
  max_walk_time = max_walk_time,
  shortest_path = FALSE
)
```

- Routing output

```
> det |> st_drop_geometry() |>
+   select(-c(route)) |> print()
   from_id from_lat from_lon    to_id   to_lat to_lon option departure_time total_duration total_distance segment   mode segment_duration  wait distance total_duration_str
    <char>    <num>    <num>   <char>    <num>  <num>  <int>         <char>          <num>          <int>   <int> <char>            <num> <num>    <int>              <char>
1:     UZH 47.37472  8.54932 BHF_enge 47.36422 8.5314      1       21:00:00           35.3           2074       1   WALK             35.3   0.0     2074           35.3 min
2:     UZH 47.37472  8.54932 BHF_enge 47.36422 8.5314      2       21:05:58           14.2           2306       1   WALK              4.3   0.0      250           14.2 min
3:     UZH 47.37472  8.54932 BHF_enge 47.36422 8.5314      2       21:05:58           14.2           2306       2   TRAM              8.3   1.5     2050           14.2 min
4:     UZH 47.37472  8.54932 BHF_enge 47.36422 8.5314      2       21:05:58           14.2           2306       3   WALK              0.1   0.0        6           14.2 min
5:     UZH 47.37472  8.54932 BHF_enge 47.36422 8.5314      3       21:09:44           18.5           2698       1   WALK              4.3   0.0      250           18.5 min
6:     UZH 47.37472  8.54932 BHF_enge 47.36422 8.5314      3       21:09:44           18.5           2698       2   TRAM              7.2   1.6     1525           18.5 min
7:     UZH 47.37472  8.54932 BHF_enge 47.36422 8.5314      3       21:09:44           18.5           2698       3   WALK              0.6   0.0       33           18.5 min
8:     UZH 47.37472  8.54932 BHF_enge 47.36422 8.5314      3       21:09:44           18.5           2698       4   TRAM              3.5   1.1      877           18.5 min
9:     UZH 47.37472  8.54932 BHF_enge 47.36422 8.5314      3       21:09:44           18.5           2698       5   WALK              0.3   0.0       13           18.5 min
```




```
route_plot <- ggplot() +
  annotation_map_tile(type = "cartolight", zoom = 16) +
  geom_sf(data = det, aes(color = mode), linewidth = 1) +
  facet_wrap(. ~ total_duration_str) +
  scale_color_npg() +
  theme_void()
```

Dr. Anna Reiffer
Postdoctoral Researcher
areiffer@ethz.ch

ETH Zürich
Institute for Transport Planning and Systems

www.ivt.baug.ethz.ch