|  |  |  |
| --- | --- | --- |
|  | PTV VISUM Beispielbeschreibung | |
|  | | |
| EIN AKTIVITÄTEN-BASIERTES MODELL | |  |
|  | |  |
|  | |  |

# Kurzbeschreibung

Das Beispiel dient als Grundlage beim Aufbau eines aktivitäten-basierten Modells in PTV Visum. Es wird der Modell-Aufbau sowie die Modell-Verwendung erläutert.

# Voraussetzungen

Visum Module: ABM

# Ziel

Contact: ptvvision@ptv.de

Die Modellierung von aktivitäten-basierten Modellen in Visum unterscheidet sich von anderen Nachfragemodelltypen. Visum bietet für die Entscheidungsmodelle keine exakt vordefinierten Verfahren mit leicht zu bedienenden komfortablen Nutzerdialogen. Stattdessen erfolgt die Verfahrensausführung mit Hilfe von Python-Skripten, die Modellparameter werden in benutzerdefinierten Tabellen gehalten.

Während der Komfort etwas leidet, eröffnet sich dadurch die Möglichkeit, sehr individuell die eigenen Vorstellungen umzusetzen. Dies ist vor allem deshalb wichtig, weil es nach derzeitigem Stand noch keine einheitlich anerkannte Modellstruktur für ABM gibt.

Das vorliegende Beispiel stellt ein vollständiges ABM dar. Es ist nicht notwendig, die Modellstruktur anzupassen. Das Beispiel kann ohne Programmierkenntnisse auf die eigene Situation übertragen werden. Dabei müssen insbesondere die Python-Skripte weder verstanden noch angepasst werden.

Insofern ist dieses Beispiel weit mehr als ein gewöhnliches Beispiel: es erklärt nicht nur die Verwendung der Funktionalitäten, sondern es dient als Vorlage für ein eigenes Modell. Ferner wird in dieser Beschreibung auf die Modellstruktur und alle Teilmodelle detailliert eingegangen.

Das Modell stellt eine Erweiterung des mit Visum 2020 veröffentlichten Modells dar. Dieses war aus einer Zusammenarbeit der PTV AG und der Schweizer Bundesbahn (SBB) hervorgegangen.

Das vorliegende Dokument umfasst die Bereiche:

* Anleitung zur Berechnung des Beispiels
* Aufbau eines eigenen Modells
* Beschreibung der Modellstruktur und der Teilmodelle
* Erläuterungen zum Programm-Code

# Modell-Vorbereitung

Ein aktivitäten-basiertes Nachfragemodell modelliert das Mobilitätsverhalten für jede Person der Bevölkerung individuell. Dafür wurde für das vorliegende Beispiel eine synthetische Bevölkerung geschaffen, die in ihren Eigenschaften der tatsächlichen Bevölkerung ähnlich ist. Diese Bevölkerungsdaten sowie passende Strukturdaten sind dem beiliegenden Beispiel beigelegt und werden im Verfahrensablauf hinzugelesen. Das Modell enthält bereits alle IV- und ÖV-Netzdaten. Falls Sie ein eigenes ABM aufbauen wollen, müssen alle Daten zunächst erzeugt und dann nach Visum importiert werden. Informationen zu diesem Thema finden Sie im Abschnitt „Aufbau eines eigenen Modells“.

Je nach Grad der bevorzugten räumlichen Auflösung können Bevölkerungs- und Strukturdaten auf Bezirksebene oder beliebig höher aufgelöst modelliert werden. Im vorliegenden Beispiel besitzen die Daten eine sehr hohe Auflösung und werden durch Standorte repräsentiert.

# Modell-Überblick

Ein ABM erzeugt für jede Person der Bevölkerung einen kompletten Tagesplan mit allen Aktivitäten, deren Start- und Endzeitpunkten, den zugehörigen Wegen und deren Modi. Die Modellberechnung ist in mehrere Teile unterteilt.

Zunächst werden alle notwendigen Kenngrößen berechnet, die später in Entscheidungsmodellen verwendet werden. Diese Berechnungen sind im Verfahrensablauf definiert. Sie unterscheiden sich nicht wesentlich von denen, die in herkömmlichen makroskopischen Nachfragemodellen zu finden sind. Neben den typischen makroskopischen Kenngrößen (also Matrizen auf Bezirksebene) sind dies auch Daten, die im späteren Verlauf als Grundlage für Kurzwegsuchen dienen. Dazu gehören Kenngrößenmatrizen auf der Ebene von Haltestellenbereichen sowie IV-Umlegungen mitsamt dem Speichern der Umlegungsergebnisse.

Danach werden mehrere Skripte ausgeführt, die die Mobilitätsentscheidungen aller einzelnen Personen individuell simulieren:

* Langfristige Entscheidungen zum Arbeits- und/oder Ausbildungs-/Schulplatz
* Erzeugung der Wegeketten
  + Anzahl der Touren, Zwischentouren und Zwischenhalte
  + Wahl der Aktivitäten, deren Startzeiten und Dauern
* Ziel- und Moduswahl

Am Ende werden die Trips zu Nachfragematrizen zusammengefasst und umgelegt.

Das vorliegende Beispielmodell kann in zwei Varianten berechnet werden: die Erzeugung der Wegeketten kann modell- oder datenbasiert erfolgen. Bei der modellbasierten Variante wird die Bildung der individuellen Wegeketten auf Grundlage von Entscheidungsmodellen simuliert. Bei der datenbasierten Variante werden die Wegeketten einer Haushaltsbefragung entnommen. Die beiden Varianten werden weiter unten näher kommentiert.

Nach der Simulation einer jeden Entscheidung werden die Ergebnisse im Datenmodell hinterlegt. So wird z.B. je gewählter Arbeitstour eine solche erzeugt und mit der jeweiligen Person verknüpft. Durch diese Speicherung der Zwischenergebnisse ist es möglich, die Entscheidungsmodelle einzeln hintereinander zu berechnen und die Zwischenergebnisse jeweils zu validieren.

Das vorliegende Beispiel ist in vielerlei Hinsicht disaggregiert:

* Wohn- und Aktivitätenstandorte sind durch das Visum-Objekt „Standort“ mit x- und y-Koordinate repräsentiert.
* Die Zielwahl basiert in einem ersten Schritt auf herkömmlichen Verkehrsbezirken. In einem zweiten Schritt werden die Trips auf die Aktivitätenstandorte innerhalb des Bezirks verteilt, wobei die Aufteilung proportional zum Anziehungspotential verläuft.
* Die Moduswahl wird für die konkreten Start- und Ziel-Standorte durchgeführt. Sie erfolgt auf Grundlage der widerstandsärmsten Wege zwischen den (aktiven) Knoten, die den Standorten am nächsten liegen.
* Das Modell ist in 8 Zeitbereiche (Analysezeitintervalle) von ‚early morning‘ bis ‚night‘ unterteilt. Kenngrößen werden je Zeitbereich berechnet und gehen in die Ziel- und Moduswahl ein.
* Alle Entscheidungen werden für jede Person individuell simuliert. Dabei können Personeneigenschaften direkt in die Wahlmodelle einfließen. Die Wahlmodelle werden jeweils für Personengruppen definiert. Die Gruppendefinitionen müssen nicht über die gesamte Modellberechnung konstant bleiben, sondern können sich von Teilmodell zu Teilmodell unterscheiden. So gibt es z.B. ein Wahlmodell für die ganze Bevölkerung zur Wahl der Tour-Startzeit. Bei der Moduswahl hingegen wird bei zwei unterschiedlichen Wahlmodellen zwischen Schülern und Erwachsenen unterschieden.

# Modellaufbau und Berechnung

Im Folgenden wird beschrieben, wie für das Beispielmodell die Berechnung durchgeführt und nachvollzogen werden kann. Dabei wird auch darauf eingegangen, wie Sie die Eingangs- und Ergebnisdaten visualisieren können. Die genaue Modellspezifikation und ihre Kodierung finden Sie weiter unten.

1. Setzen Sie sämtliche Projektverzeichnisse auf den Beispielordner.
2. Öffnen Sie die Versionsdatei Halle\_ABM.ver.

Das Modell enthält bereits die Verkehrsnetze, allerdings noch keine Nachfragedaten. Laden Sie diese zunächst dem Modell hinzu:

1. Laden Sie zunächst mit 1\_UDA.net die benutzerdefinierten Attribute.
2. Laden Sie mit 2\_Model\_specifications.net die Modellspezifikationen, also Nutzendefinitionen etc.
3. Laden Sie mit 3\_Base\_Demand\_Model.dmd das ABM-Nachfragemodell und dessen Aktivitäten, dann mit 4\_Base\_Demand\_Model\_Data.att notwendige Daten dazu.
4. Laden Sie mit 5\_Structural\_Data.dmd die Strukturdaten, also Standorte und Aktivitätsstandorte.

Im nächsten Schritt wird die synthetische Bevölkerung mit einem Skript eingelesen. Dafür werden vom Skript drei Daten-Dateien erwartet:

* HHI.dmd mit den Originaldaten von Haushalten und Personen sowie deren Wegeketten aus einer Haushaltsbefragung
* SynPopHH.csv, SynPopPerson.csv mit den daraus generierten synthetischen Haushalten sowie Personen

Falls die Daten für ein eigenes Modell erstellt werden, sollte man die Struktur der Datendateien beibehalten, damit das gleiche Import-Skript verwendet werden kann. Eine Beschreibung der Dateien finden Sie weiter unten im Abschnitt „Aufbau eines eigenen Modells“.

1. Starten Sie die beiden Verfahren im Verfahrensablauf aus der Gruppe Preparation. Die dabei erzeugt Warnung können Sie ignorieren.

Das Modell enthält nun sämtliche Eingangsdaten.

1. Sie können sich jetzt die Modellspezifikationen sowie Bevölkerungs- und Strukturdaten in Listen oder auch im Netzeditor anschauen. Öffnen Sie dazu die Gesamtlayoutdatei Data.lay.

In der Liste **Personen** sind alle Personen des Modells aufgeführt. Ihre individuellen Eigenschaften wie Alter und Beschäftigungskategorie sind in entsprechenden benutzerdefinierten Attributen hinterlegt.

In der Liste **Aktivitätsstandorte** sind alle Paare „Aktivität x Standort“ aufgeführt. Aktivitätsstandorte sind Standorte, an denen die zugehörige Aktivität ausgeführt werden kann. Das Anziehungspotential hinsichtlich der Aktivität, also beispielsweise die Anzahl der Grundschulplätze im Fall der Aktivität „Education primary Pupil“ (Bildung - Grundschule), ist im Attribut „Anziehungspotential“ gespeichert.

Wählen Sie einen oder mehrere Datensätze in den Listen aus, werden diese im Netzeditor hervorgehoben.

1. Wechseln Sie zur benutzerdefinierten Tabelle **Model Steps**.

Die Tabelle **Model Steps** führt alle Modellschritte auf. Sie startet bei den langfristigen Entscheidungen (Spalte „ChoiceModel“ = „PrimLoc“) und endet mit der Ziel- und Moduswahl. In der Spalte „Specification“ ist jeweils der Name der Tabelle enthalten, welche die Modellparameter für den jeweiligen Modellschritt enthält.

Für den ersten Modellschritt ist dort der Name „PrimLocStu“ hinterlegt.

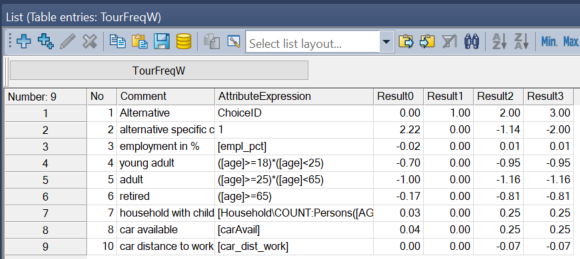
1. Wechseln Sie zur benutzerdefinierten Tabelle PrimLocStu.

In dieser Tabelle finden Sie die Definition des Nutzens uod für die Zielwahl des Schulorts d vom Wohnort o aus. Für Schüler von weiterführenden Schulen ist sie z.B. definiert durch:

uod = 0.8 • Matrix(LogSumNoCarAv, AM)od

Die Matrix(LogSumNoCarAv, AM) wurde bereits bei der Kenngrößenberechnung berechnet. Sie stellt den Nutzen (= negativer Widerstand) einer Ortsveränderung in Form einer Logsum über alle Verkehrsmittel dar. Die Verkehrsmittelwiderstände werden für die Morgenspitze (AM) berechnet.

1. Wechseln Sie zur Tabelle TourFreqW.



Diese Liste enthält die Parameter des Wahlmodells für die Anzahl der Arbeitstouren. In der Spalte „Attribute Expression“ sind die Personen-Attribute (xi) aufgeführt, die zum Nutzen der einzelnen Wahlalternativen beitragen. In den weiteren Spalten stehen die entsprechenden Parameter (βA,i), die zur jeweiligen Alternative (A) und zum Personen-Attribut xi gehören. Dabei ist die Alternative selbst (also die Zahlen 0, 1, 2, 3) in der Zeile mit Attribute Expression = ChoiceID definiert (siehe erste Zeile in der Liste).

Der Nutzen uA einer Alternative A berechnet sich wie folgt:

uA = Σi βA,i • xi (Σi bedeutet hier Summe über i)

Konkret bedeutet dies zum Beispiel für die Alternative „0 Arbeitstouren“

u0 = 2.22

- 0.02 • [empl\_pct]

- 0.70 • [age\_18\_u25]

- 1.00 • [age\_25\_u65]

- 0.17 • [age\_65plus]

+ 0.03 • [has\_child]

+ 0.04 • [carAvail]

An den Parametern βA,i lässt sich ablesen, welchen Einfluss die Attribute auf den Nutzen der Wahlalternativen haben. So verringert beispielsweise ein junges Alter (18 – 25 Jahre) den Nutzen der Alternative „0 Arbeitstouren“ um 0.7. Dagegen erhöht das Vorhandensein von Kindern den Nutzen der Alternative „2 Arbeitstouren“ um 0.25.

Die Parameter zu den einzelnen Attributen werden in der Regel auf Basis einer Mobilitätsbefragung geschätzt. Im vorliegenden Modell wurden die Parameter zu Demonstrationszwecken willkürlich gesetzt.

Bei den Attributen xi handelt es sich im vorliegenden Fall immer um benutzerdefinierte Attribute von Personen. Es ist aber auch möglich, abgeleitete Attribute zu verwenden. Die Schreibweise für solche abgeleiteten Attribute ist z.B. in Zeile 7 demonstriert. Außerdem können die Attributausdrücke auch gewisse Funktionen enthalten (siehe z.B. in Zeile 4 und 5). Die Syntax der Ausdrücke ist identisch mit der der normalen Formeln in Visum.

Das hier beschriebene Wahlmodell ist ein sogenanntes nutzenbasiertes Logit-Modell, ein für aktivitäten-basierte Modelle typisches Modell (weiter unten referenziert als „Standardwahlmodell“). Für die Wahlwahrscheinlichkeiten pA der Alternative A gilt

pA = exp (uA ) / Σi exp (uA,i ).

Das Modell enthält bereits für alle Personen deren Touren, Trips und Aktivitätenausübungen aus den im vorigen Schritt eingelesenen Befragungsdaten. Es fehlen, analog zu üblichen 4-Stufen-Modellen, lediglich die Ziel- und Moduswahlen. Sie können sich die Daten in den entsprechenden Listen anschauen.

Die Daten wurden für die sogenannte datenbasierte ABM-Variante eingelesen. Dabei wird die Bildung der Wegeketten nicht durch Entscheidungsmodelle simuliert, sondern direkt den Ergebnissen einer Befragung entnommen. Für den Fall, dass die modellbasierte ABM-Variante gerechnet werden soll, aktivieren Sie das Skript-Verfahren run\_init\_schedules.py in der Gruppe ABM und löschen damit alle Ergebnisdaten.

In den folgenden Gruppen werden Kenngrößenmatrizen berechnet, die für die Ziel- und Moduswahlmodelle benötigt werden. Dazu werden die jeweiligen Nachfragesegmente vorher umgelegt. Für die Nachfragesegmente Walk und Bike spielt die Nachfrage zwar keine Rolle, da sie entsprechend dem Bestweg umgelegt werden. Allerdings sollen alle Anbindungen entsprechend ihrem Gewicht verwendet werden, weshalb eine Umlegung einer Dummy-Nachfrage notwendig ist.

1. Starten Sie alle Verfahren der Gruppe initialize demand matrices, damit die Nachfragematrizen initialisiert werden.
2. Starten Sie die Verfahren der Gruppen PT Skims, PrT Skims without Car.

Für die Berechnung der Pkw-Kenngrößen wird je Analysezeitperiode eine statische Umlegung gerechnet. Dafür wird in der Gruppe **Car Skims** je Zeitintervall eine Umlegung und darauf basierend Kenngrößen sowie LogSum-Matrizen für die Wahl der langfristigen Entscheidungen berechnet. Die Nachfrage je Zeitintervall ist zunächst immer identisch. Erst wenn man die gesamte Nachfrageschleife mehrfach durchläuft und die Pkw-Nachfrage dabei aktualisiert wird, wirken sich die unterschiedlichen Nachfragen der unterschiedlichen Zeitintervalle auf die Umlegungsergebnisse aus. Am Ende der Gruppe werden für spätere Kurzwegsuchen Widerstand und Fahrzeit an allen Wege-Objekten gespeichert.

1. Starten Sie die Verfahren der Gruppe Car Skims.

Nun sind alle Daten und Kenngrößen vorhanden und die Berechnung des ABM kann gestartet werden.

1. Starten Sie das erste Verfahren der Gruppe ABM.

Dieses Verfahren prüft, ob die Modellspezifikationen Tippfehler enthalten.

1. Starten Sie das zweite Verfahren der Gruppe ABM.

Der geöffnete Filter aktiviert alle Knoten, die als Anbindung der Standorte an das Netz geeignet sind. Standorte sind nicht explizit an das Netz angebunden, sondern immer über den nächstgelegenen aktiven Knoten. Deshalb muss darauf geachtet werden, dass alle Knoten, die sich nicht für eine Anbindung eignen, durch den Filter deaktiviert werden.

1. Für den Fall, dass die modellbasierte ABM-Variante gerechnet werden soll, starten Sie das dritte Verfahren run\_init\_schedules.py und löschen damit alle Ergebnisdaten.

Im nächsten Verfahren werden die langfristigen Entscheidungen simuliert, also die Ziele der Pflichtaktivitäten wie Arbeiten oder Schule.

1. Starten Sie das nächste Verfahren der Gruppe ABM.

Falls die Aktivitätenketten modellbasiert erzeugt werden sollen, starten Sie auch das nächste Verfahren (run\_tripchain\_choice.py).

1. Im letzten Schritt des ABM starten Sie die Ziel- und Moduswahl (Skript run\_destination\_and\_mode\_choice.py).
2. Nun wird die Nachfrage aus den Trips in Nachfragematrizen geschrieben, um diese dann wieder umlegen zu können. Starten Sie dazu das nächste Verfahren (Skript run\_aggregate\_trip\_volumes.py), welches die Nachfrage aus den Trips je NSeg und Analysezeitintervall aggregiert und das Ergebnis in die entsprechenden Nachfragematrizen schreibt (diese müssen neben passendem NSeg und den Attributen FromTime sowie ToTime den Namen „assignment“ haben).

Falls die Nachfrage mit einer Nachfrageschleife berechnet werden soll, so dass sich die Pkw-Fahrtzeiten jeweils aktualisieren, aktivieren und starten Sie das letzte Verfahren der Gruppe ABM (Rücksprung).

Nach Abschluss der Nachfrageberechnung können die Ergebnisse analysiert werden, siehe dazu den folgenden Abschnitt. Um die Ergebnisse im Zusammenhang mit Umlegungen untersuchen zu können (z.B. Spinnenberechnungen), soll die Nachfrage beispielhaft für die Morgenspitze umgelegt werden. Außerdem werden die Touren in Wege-Abfolgen konvertiert.

1. Starten Sie die ersten beiden Verfahren aus der Gruppe Assignment.

Anschließend können die Nachfragesegmente umgelegt werden.

1. Starten Sie dazu die beiden Umlegungen aus der Gruppe Assignment.

## Ergebnisse

Im Folgenden werden die Ergebnisse der datenbasierten ABM-Variante nach einmaligem Durchlauf dargestellt, also ohne Aktivierung des Rücksprungs.

Wechseln Sie abwechselnd in die Listen Touren und Wegeabfolgen.

Die Tour-Liste zeigt alle Touren, die während der Modellberechnung erstellt wurden. Wählen Sie einzelne davon aus, werden diese im Netzeditor angezeigt.   
Die Wegeabfolgen-Liste zeigt die zugehörigen Wegeabfolgen, bei denen im Netz-Editor zusätzlich die entsprechenden Umlegungswege angezeigt werden.

Weitere Ergebnisse finden Sie u.a. in den Listen:

Trips

Aktivitätsausübungen

Halle\_ABM\_Result.ver ist eine Versionsdatei mit einer vollständigen Berechnung des ABM.

# Aufbau eines eigenen Modells

Das vorliegende Beispiel dient, anders als andere Beispiele, nicht nur der Darstellung bestimmter Features, sondern auch direkt als Grundlage für ein eigenes ABM. Für den Aufbau eines ABM ist es also sinnvoll, mit dem Beispiel zu starten und es geeignet anzupassen.

Im Folgenden wird auf die anzupassenden Bausteine genauer eingegangen.

## Netzmodell, Bezirkseinteilung, Nachfragemodelltyp

Zunächst müssen die IV- und ÖV-Netzmodelle, die Bezirkseinteilung sowie die Anbindungen ausgetauscht werden. Dabei werden auch Verkehrssysteme, Modi und Nachfragesegmente ausgetauscht bzw. angepasst. Dieser Vorgang unterscheidet sich nicht vom Aufbau eines klassischen makroskopischen Nachfragemodells.

Da das Modell zeitlich disaggregiert ist, müssen die Analysezeitintervalle an die eigenen Bedürfnisse angepasst bzw. erweitert werden. Dabei müssen alle Analysezeitintervalle überschneidungsfrei den gesamten Tag abdecken. Ein Kalender wird nicht unterstützt.

Es muss ein Nachfragemodell vom Typ ABM angelegt werden. Einzige Attribute dieses Nachfragemodelltyps sind die Aktivitäten, von denen genau eine als Heimataktivität gekennzeichnet sein muss (diese muss den Code „H“ besitzen).

Die für das ABM notwendige Attributierung der Nachfrageobjekte wird weiter unten in der Modellspezifikation im Bereich Attributierung der Nachfrageobjekte beschrieben.

## Synthetische Bevölkerung und Strukturdaten

Die Bevölkerung wird in einem ABM als eine Menge konkreter Personen abgebildet, synthetische Bevölkerung genannt. Genauso wie bei makroskopischen Modellen ist die Abbildung der Bevölkerung auch bei ABM keine triviale Angelegenheit. Ähnlich wie bei makroskopischen Modellen müssen die Haushalte so ausgewählt werden, dass die Randverteilungen möglichst genau eingehalten werden. Für diese Datenaufbereitung stellt Visum keine spezielle Funktionalität zur Verfügung (gleiches gilt übrigens auch für den makroskopischen Fall).

Es gibt allerdings zahlreiche Software, die genau für diesen Fall geschrieben wurde. Wir empfehlen, die Software von PopulationSim[[1]](#footnote-2) zu verwenden. Sie ist eine offene Plattform, kann also im Rahmen ihrer Lizenz frei verwendet werden. PopulationSim ist sehr gut dokumentiert, leicht anzuwenden und ideal für diesen Anwendungsfall zugeschnitten.

Grundlage der synthetischen Bevölkerung ist, wie auch meistens bei makroskopischen Nachfragemodellen, eine Haushaltsbefragung (die nicht unbedingt im Modellraum durchgeführt wurde) und diverse Randverteilungen (Schulplätze, sozialversicherungspflichtige Beschäftigungsverhältnisse je Gemeinde, …). Bei klassischen Nachfragemodellen werden daraus je Bezirk die Anzahl Personen je verhaltenshomogener Personengruppen geschätzt, dazu die zugehörigen Erzeugungsraten je Aktivitätenpaar oder Wegekette. Entsprechend werden für ein ABM konkrete Haushalte für sogenannte Mikro-Zonen erzeugt. Die Haushalte sind dabei Kopien der Haushalte aus der Haushaltsbefragung inklusive der zugehörigen Personen.

Im Verfahrensablauf der Beispielversion ist das Einlesen der synthetischen Bevölkerung als Skript umgesetzt. Dabei werden drei Dateien erwartet:

* HHI.dmd
* SynPopHH.csv
* SynPopPerson.csv

Die Datei HHI.dmd enthält Daten aus einer Haushaltsbefragung:

* Die Tabellen Locations und ActivityLocations erzeugen ein Dummy-Objekt, das aus technischen Gründen vorhanden sein muss. Beide Tabellen werden nach Einlesen wieder gelöscht. Es muss lediglich darauf geachtet werden, dass die Nummer der Location nicht schon im Modell vergeben ist.
* Die Tabelle Households enthält
  + eine HH-Nummer (sie spielt im weiteren Verlauf keine Rolle)
  + den Haushaltsschlüssel (er ist immer identisch (H,1), da der genaue Ort in der Datei SynPopHH.csv definiert wird)
  + die originale Haushalts-ID, welche die Verbindung zur entsprechenden ID in der Datei SynPopHH.csv herstellt
  + sowie beliebig viele weitere Haushaltsattribute (in diesem Fall mit NOCars die Anzahl der Pkw).
* Die Tabelle Persons enthält
  + die Personen- und Haushaltsnummer (diese spielen im weiteren Verlauf keine Rolle)
  + die originale Personen-ID, welche die Verbindung zur entsprechenden ID in der Datei SynPopPerson.csv herstellt
  + sowie beliebig viele weitere Personenattribute, die in Wahlmodellen eine Rolle spielen
* Die Tabelle Schedules (also die Tagespläne) enthält lediglich die zugehörige Personennummer sowie die Nummer des Tagesplans. Da es je Person immer nur einen Tagesplan gibt, ist die Nummer konstant 1.
* Die Tabelle Tours, die nur die zugehörigen Nummern des Tagesplans und der Person enthält sowie die Nummer der Tour selbst.
* Die Tabelle Activity executions mit den folgenden Attributen:
  + die zugehörigen Personen- und Tagesplannummer sowie Index der Aktivitätenausübung; die Indizes entsprechen der Reihenfolge der Aktivitätenausführungen
  + Code der Aktivitätenausübung; jede Tour beginnt und endet mit einer Home-Aktivitätenausübung (Code „H“); aufeinanderfolgende Touren eines Tagesplans teilen sich eine Home-Aktivitätsausübung
  + das boolesche Attribut IsMajorActivity: 1, falls die Aktivitätenausübung die Hauptaktivitätenausübung der Tour darstellt (dies ist die zentrale Aktivitätenausübung einer Tour, welche wesentlich die Ziel-, Modus- und Startzeitwahl aller Aktivitätenausübungen der Tour beeinflusst); jede Tour muss eine solche Hauptaktivitätenausübung besitzen. Eine zweite Hauptaktivitätenausübung darf es nur bei Subtouren geben und auch nur dann, wenn sie den gleichen Typ besitzt (siehe dazu weiter unten).
  + Dauer der Aktivitätenausführung: jede Aktivitätenausübung muss eine Dauer besitzen
  + Startzeit der Aktivitätenausführung: Die Hauptaktivitätenausübung muss eine Startzeit besitzen. Weitere Startzeiten sind erlaubt, werden aber im Verlauf der ABM-Berechnung überschrieben.
  + Das boolesche Attribut IsPartOfSubtour: 1, falls die Aktivitätenausübung Teil einer Subtour ist. Subtouren dürfen nur bei primären Touren vorkommen (maximal eine). Primäre Touren sind Touren, deren Hauptaktivität eine Pflichtaktivität (Schule, Arbeit) darstellt.   
    Sind in einer Tour zwei Hauptaktivitäten identischen Typs vorhanden, bilden beide zusammen mit allen dazwischen liegenden Aktivitätenausübungen eine Subtour (IsPartOfSubtour = True bei allen zugehörigen Aktivitätenausübungen). Damit ist sichergestellt, dass die Subtour an der gleichen Stelle endet, an der sie begonnen hat. Es kann vorkommen, dass in Erhebungsdaten in einer Tour mehr als zwei Arbeitsaktivitäten vorkommen. In diesem Fall werden nur die erste und letzte als Hauptaktivitätenausübung markiert (ISMAJORACTIVITY = True).
  + die Standortnummer, die immer konstant 1 ist (der Standort wird erst im Zielwahlmodell bestimmt)
* Die Tabelle Trips
  + die zugehörigen Personen-, Tagesplan- und Tournummer sowie Index der Aktivitätenausübung; die Indizes entsprechen der Reihenfolge der Trips
  + die Indizes der vorherigen und nachfolgenden Aktivitätenausübungen (ein Trip liegt immer zwischen zwei Aktivitätenausübungen)

Die beiden SynPop-Dateien entsprechen dem Output einer Software, die eine synthetische Bevölkerung erzeugt. SynPopHH.csv enthält neben der Haushalts-ID die Standort-Nummer und die originale Haushalts-ID. Letztere wird dafür verwendet, eine Kopie des Original-Haushalts aus der Haushaltsbefragung an den entsprechenden Standort zu setzen. SynPopPerson.csv ist ähnlich aufgebaut, sie enthält neben der Personen- und Haushalts-ID die originale Personen-ID, mit deren Hilfe eine Kopie der originalen Person aus der Haushaltsbefragung erzeugt wird.

**Strukturdaten**

Strukturdaten werden im ABM als Aktivitätenstandorte abgebildet, also als Orte, an denen eine bestimmte Aktivität ausgeübt werden kann. Ein Aktivitätenstandort besteht somit aus einem Standort und einer Aktivität. Das Potential, also z.B. die Anzahl der Arbeitsplätze oder die Verkaufsfläche, wird im Attribut „Anziehungspotential“ gespeichert.

Die Daten müssen dabei nicht unbedingt in mikroskopischer Auflösung vorliegen. Insbesondere können unscharf vorliegende Strukturdaten analog makroskopischen Modellen unscharf modelliert werden: ein Standort kann sowohl für ein Gebäude als auch für eine Fläche stehen. Je genauer die Strukturdaten allerdings lokalisiert werden, desto genauer werden die Ergebnisse.

Im oben beschriebenen Beispielvorgehen wird der Aufbau eines Modells mit Beispieldaten durchgeführt.

## Erzeugung

Die Erzeugung im ABM entspricht im Wesentlichen dem ersten Schritt eines klassischen 4-Stufen-Modells. Das Ergebnis der Erzeugung sind für jede Person die zugehörigen Wegeketten, also Touren, Trips und Aktivitätenausübungen, dazu die Startzeit der Hauptaktivität, allerdings alles noch ohne konkrete Ziele, ohne Modi und ohne Fahrzeiten. Die Erzeugung bei ABM kann, analog zu klassischen Modellen, datenbasiert erfolgen. Bei den meisten Modellierern von ABM ist allerdings die modellbasierte Erzeugung üblicher. Im Folgenden gehen wir auf beide Ansätze ein.

**Datenbasierte Erzeugung**

Grundlage der Erzeugung ist die Haushaltsbefragung, die schon bei der Erzeugung der synthetischen Bevölkerung verwendet wurde. Bei klassischen Nachfragemodellen werden daraus die zugehörigen Erzeugungsraten je Aktivitätenpaar oder Wegekette bestimmt. Entsprechend werden für ein ABM den Personen ihre ursprüngliche Wegekette zugespielt. Das Wirkungsmodell des ABM bestimmt dann nur noch die Ziel- und Moduswahl, vergleichbar einem makroskopischen 4-Stufen-Modell.

Im Verfahrensablauf der Beispielversion ist das Zuspielen der Wegeketten zu den synthetischen Personen als Skript umgesetzt, Beispieldateien liegen dem Beispiel bei. Wenn die datenbasierte Erzeugung verwendet wird, müssen diese Erzeugungsdaten wieder gelöscht werden (siehe auch in der Beispielbeschreibung).

Bei der Erzeugung der dmd-Datei, welche die Wegeketten aus der Haushaltsbefragung enthält, sollte man sich hinsichtlich des Formats an die mitgelieferten Beispieldaten halten (das Format wird weiter oben bei der Beschreibung der synthetischen Bevölkerung beschrieben).

Der Vorteil der datenbasierten Erzeugung ist der hohe Grad an Konsistenz der Wegekettenelemente, was mit einer modellbasierten Erzeugung nur sehr schwer zu erreichen ist. Insbesondere das Zusammenspiel von Aktivitätenfolgen und deren Dauern sowie hintereinander ausgeführte Touren passen automatisch zusammen und sind per Definition realistisch.

**Modellbasierte Erzeugung**

Bei der modellbasierten Erzeugung werden alle Elemente hintereinander simuliert: zunächst die Anzahl der Touren, dann die Anzahl der Aktivitäten je Tour, der Typ einer jeden Aktivität und so weiter. Das Erzeugungsmodell wird durch das Skript run\_tripchain\_choice.py durchgeführt. Weitere Einzelheiten finden Sie in der Modellspezifikation weiter unten.

Der wesentliche Unterschied zur datenbasierten Erzeugung liegt in der Kalibrierung: während die datenbasierte Erzeugung per Definition exakt mit den beobachteten Daten übereinstimmt, muss dies bei der modellbasierten Erzeugung erst durch Kalibrierung sichergestellt werden. Eine Kalibrierung erfolgt entlang der Kette der Teilmodelle, wobei jedes Teilmodell für sich kalibriert wird. Dabei werden jeweils Modellergebnisse den Beobachtungen gegenübergestellt und Modellparameter so angepasst, dass sich beide nur noch geringfügig unterscheiden.

Eine solche Modellkalibrierung kann mitunter sehr langwierig sein. Jedes Teilmodell besitzt sehr viele Einflussfaktoren, deren Wirkungen oft nicht gut abgeschätzt werden können. Während die Teilmodelle im Erzeugungsmodell hintereinander und im Wesentlichen unabhängig voneinander agieren, ist die Wirklichkeit deutlich komplexer. Bei der Entscheidung für einen bestimmten Tagesplan hängen in Wirklichkeit alle Teilentscheidungen, also z.B. Anzahl der Touren, Länge und Dauer der Aktivitätenausübungen, miteinander zusammen.

Ein Vorteil der modellbasierten Erzeugung liegt in seiner Reaktivität auf Änderungen. Wenn sich verkehrliche oder strukturelle Veränderungen ergeben, kann ein Erzeugungsmodell darauf geeignet reagieren. Ein datenbasiertes Erzeugungsmodell kann dies nicht in diesem Maße, es ist im Wesentlichen konstant.

Ein weiterer Vorteil ist die meistens höhere Variabilität der Wegeketten. Eine Datenbasierte Erzeugung kann nur die Wegeketten reproduzieren, die in einer Befragung erhoben wurden. Sind die Stichprobenzahlen gering, ist auch die Variabilität der erhobenen Mobilität entsprechend gering. Dies kann zu parallelen gleichförmigen Bewegungsmustern von sehr vielen Personen führen und im Extremfall zu unrealistischen Ergebnissen. Allerdings sind solche Ergebnisse eher dann problematisch, wenn sie als Eingangsdaten für eine Mobilitätssimulation dienen: Nachfrage, die sich in der Realität über einen gewissen Zeitraum gleichmäßig verteilt, tritt in der Simulation nun punktförmig zu einem Zeitpunkt auf.

Bei der Modellbasierten Erzeugung ist zu beachten, dass sich die Wahrscheinlichkeitsverteilungen der Startzeitpunkte von Aktivitätenausübungen auf Hauptaktivitäten beziehen und nicht auf alle Aktivitätenausübungen.

**Vergleich der beiden Erzeugungsmodelle**

Die Auswirkungen von Strukturänderungen auf die Zusammensetzung von Wegeketten und deren zeitlicher Lage sind vermutlich eher gering. Für die meisten Szenarien ist vermutlich auch gar nicht klar, in welcher Art und welchem Maße Änderungen zu erwarten sind. Angenommen, die Fahrzeiten würden um 10% steigen: welche Änderungen der Erzeugung müsste dies nach sich ziehen und in welchem Maße? Die Hinwege könnten früher, die Rückwege später begonnen werden, die Aktivitätenausübungen könnten kürzer werden, Aktivitäten könnten ausgelassen werden, Touren könnten zusammengefasst werden. Alle Reaktionen scheinen plausibel, aber für keine gibt es eine Datengrundlage und somit kein kalibriertes Modell.

Auf der anderen Seite können auch mit einem datenbasierten Modell Reaktionen auf Strukturänderungen modelliert werden, und zwar in der Art, wie es makroskopische Modelle schon immer tun: die Auswirkungen werden in einem externen Modell bestimmt und dann durch Verschiebung der Anteile der Personengruppen implementiert. Erwartet man z.B. eine Verringerung der Pkw-Verfügbarkeit, so verringert man in einem makroskopischen Modell den Anteil an Personen mit Pkw-Verfügbarkeit. Bei einem ABM kann dies genauso erfolgen: die Erzeugung der synthetischen Bevölkerung erfolgt auf Grundlage von Randverteilungen, also z.B. auch dem Anteil an Personen mit Pkw-Verfügbarkeit. Ändert man diesen Anteil auf Grundlage einer externen Einschätzung und erzeugt eine neue synthetische Bevölkerung, so kann diese für ein entsprechendes Zukunftsszenario dienen[[2]](#footnote-3).

**Zeitlich überschneidende Touren**

Bei der modellbasierten Erzeugung können Touren entstehen, die in der Realität wegen zeitlicher Überschneidungen nicht durchführbar wären: die erste Tour endet später als die zweite beginnt[[3]](#footnote-4). Daneben können auch Aktivitätenausübungen zu sehr unwahrscheinlichen Zeiten entstehen. Man könnte das leicht ändern, indem man solche Touren löscht. Die Teilmodelle müssten dann hinsichtlich der übrigen Touren entsprechend kalibriert werden, damit wieder die korrekten Randverteilungen erreicht werden.

Allerdings ist dieser Aufwand in den meisten Fällen nicht nötig: Das Ziel eines ABM ist nicht die Realisierbarkeit aller Touren, sondern die verbesserte Reaktivität auf strukturelle oder verkehrliche Änderungen. Eine Anpassung des Erzeugungsmodells ist also nur dann notwendig, wenn Szenarien betrachtet werden sollen, bei denen sich die Bedingungen für gültige Touren verändern. Dies wäre z.B. der Fall, wenn Auswirkungen von veränderten Öffnungszeiten auf die Erzeugung untersucht werden sollen. Ein anderer Fall wäre die Verringerung der Mobilität wegen längerer Fahrzeiten. Da solche Fragen aktuell eher selten allein mit Hilfe von Verkehrsmodellen beantwortet werden, ist diese Einschränkung vermutlich eher unbedeutend.

Im vorliegenden Modell wurde also darauf verzichtet, Touren zeitlich überschneidungsfrei zu modellieren. Dadurch erscheinen die Ergebnisse leider nicht mehr so anschaulich wie vielleicht erwartet. Insbesondere sind die Start- und Endzeiten der Heim-Aktivitätsausübungen zwischen zwei Touren nicht konsistent: sie passen immer nur zu einer Tour.

## Verhaltensdaten

Verhaltensdaten sind im Wesentlichen Modellparameter (darunter auch die Ganglinien, welche die Verteilung der Startzeiten von Hauptaktivitäten darstellen). Für den Fall der Modus- und Zielwahl können diese von makroskopischen Modellen übernommen werden. Die Kalibrierungskonstanten müssen dabei allerdings neu geschätzt werden.

Die Kalibrierung erfolgt analog zu makroskopischen Modellen: kalibriert werden Modal Split und Reiseweitenverteilung[[4]](#footnote-5). Da die mikroskopische Ergebnisstruktur in der Regel nicht im Vordergrund der Untersuchungen steht, erfolgt die Kalibrierung auf der Ebene gewöhnlicher Bezirke.

Falls neue Einflussgrößen eingeführt werden, müssen die zugehörigen Parameter natürlich neu geschätzt werden. Eine solche Schätzung kann häufig auf Grundlage einer Mobilitätsbefragung durchgeführt werden: der Parameter wird so lange angepasst, bis die Modellergebnisse den beobachteten gleichen.

# Spezifikation des aktivitäten-basierten Nachfragemodells

Die Implementierung des Modells unterscheidet zwei Bereiche der Modellspezifikation. Die Architektur der einzelnen Wahlmodelle und die Reihenfolge, in der sie simuliert werden, sind durch die konkrete Implementierung im Python-Code definiert.

Die in den Wahlmodellen verwendeten Attribute und Parameter befinden sich dagegen im Wesentlichen in benutzerdefinierten Tabellen. Die Datenhaltung in benutzerdefinierten Tabellen ermöglicht eine einfache Übersicht über die Modellspezifikationen und erleichtert deren Erweiterung oder Anpassung, falls dies z.B. im Zuge einer Modell-Kalibrierung nötig sein sollte.

Die folgende Modellbeschreibung ist als Nachschlagewerk gedacht. Für eine Modellerstellung ist es nicht notwendig, sämtliche Details im Voraus zu kennen.

## Attributierung der Nachfrage-Objekte

Neben den benutzerdefinierten Tabellen gibt es noch weitere Attribute an Nachfrageobjekten, die bei der Modellspezifikation eine Rolle spielen:

* Modi
  + ID: Anhand der ID wird der Modus identifiziert; dies ist z.B. wichtig beim Moduswahlmodell. Modi mit IDs, die in keinem Wahlmodell als Alternative vorkommen, sind irrelevant.
  + Austauschbar: Legt fest, ob der Modus innerhalb einer Tour austauschbar ist oder nicht (in der Regel sind z.B. ÖV-Modi und Fuß austauschbar, Pkw nicht).
  + Rank: Die austauschbaren Modi müssen einen eindeutigen Rang besitzen. Wird eine Tour mit unterschiedlichen austauschbaren Modi durchgeführt, erhält die Tour als Hauptmodus denjenigen mit dem höchsten Rang.
* Aktivitäten
  + ID: Anhand der ID wird die Aktivität identifiziert (siehe hierzu z.B. in den ersten Zeilen der Tabelle Model Steps die Spalte „AddData“). Aktivitäten mit IDs, die in keinem Wahlmodell als Alternative vorkommen, sind irrelevant.
  + IstHeimat: Genau eine Aktivität mit Code „H“ muss eine Heimataktivität sein.
* Standorte
  + Das Standorte-Attribut „Bezirks-Nummer“ spielt keine Rolle. Die Zuordnung eines Standortes zu einem Bezirk erfolgt über die Koordinaten des Standortes und das Bezirkspolygon. Der Modellierer muss also darauf achten, dass die Bezirkspolygone überschneidungsfrei sind und dass alle Standorte innerhalb eines Bezirkspolygons liegen.
* Aktivitätenstandorte
  + Anziehungspotential: Das Potential eines Aktivitätenstandorts. Es wird bei der Zielwahl verwendet.
* Haushalte  
  In einigen Wahlmodellen wird das Attribut NOCars (number of cars), also die Anzahl an Pkw im Haushalt, verwendet. Falls die Modellspezifikation geändert wird, wird dieses Attribut ggf. nicht mehr benötigt.
* Personen  
  Folgende Attribute werden in den aktuell verwendeten Wahlmodellen verwendet. Falls die Modellspezifikation geändert wird, werden diese Attribute ggf. nicht mehr benötigt.
  + age (Alter)
  + car availability: wahr, wenn die Person einen Pkw zur Verfügung hat
  + driving license (Pkw-Führerschein)
  + employment percent: Beschäftigungsgrad; 100 für Vollzeitbeschäftigte, entsprechend weniger für Teilzeitbeschäftigte
  + is primary pupil, is secondary pupil, is apprentice, is student (Schüler einer Grundschule, Schüler einer weiterführenden Schule, Auszubildende/r, Universitätsstudent): jeweils 1, falls die Eigenschaft für die Person zutrifft.
* Ganglinien
  + Für die modellbasierte Erzeugung werden Ganglinien je Aktivität benötigt. Das Gewicht einer Zeitperiode entspricht dabei der Wahrscheinlichkeit, dass die zugehörige Aktivität als Hauptaktivität einer Tour innerhalb der Zeitperiode beginnt. Die Zuordnung der Ganglinie zur zugehörigen Aktivität erfolgt im entsprechenden Teil-Modell der Startzeitwahl (siehe dazu auch die benutzerdefinierte Tabelle StartTime).

Die folgenden Attribute werden entweder direkt einer Mobilitätsbefragung entnommen (datenbasierte Erzeugung) oder automatisch bei der Modellausführung hinzugefügt (modellbasiertes Erzeugungsmodell).

* Aktivitätenausübungen
  + Index: Ein eindeutiger Index (innerhalb eines Tagesplans)
  + Startzeit: Jede Ausübung einer Hauptaktivität (zentrale Aktivität einer Wegekette, siehe auch unten) benötigt eine Startzeit.   
    Die Startzeiten aller anderen Aktivitätenausübungen ergeben sich aus der Startzeit der Hauptaktivität und werden ggf. überschrieben.
  + Dauer
  + Aktivitäten-Code
  + IsMajorActivity: Jede Tour besitzt in der Regel genau eine Hauptaktivitätenausübung. Diese ist die zentrale Aktivitätenausübung und bestimmt wesentlich die Ziel- und Moduswahl aller Aktivitätenausübungen. Falls die Hauptaktivitätenausübung primär ist (also eine Pflichtaktivität), ergibt die zugehörige Zielwahl immer die entsprechende langfristige Entscheidung.  
    Primäre Touren können (maximal) eine Subtour besitzen, in dem Fall gibt es zwei Hauptaktivitätenausübungen identischen Typs. Sie bilden jeweils den Start und das Ende der Subtour (in diesem Fall muss auch das Attribut IsPartOfSubtour entsprechend gesetzt werden). Mehr als zwei Hauptaktivitätenausübungen darf eine Tour nicht besitzen.
  + IsPartOfSubtour: Eine Subtour ist eine Zwischentour, die beim Arbeitsplatz beginnt und dorthin wieder zurückkehrt, z.B. W-L-W innerhalb der Wegekette H-W-L-W-H. Klassisches Beispiel ist das Mittagessen.
* Trips
  + Index: Definiert die Reihenfolge der Trips innerhalb der Tour.
  + VonAktausübungIndex, NachAktausübungIndex: Start und Ziel des Trips.

## Model Steps

Eine zentrale Rolle kommt der Tabelle Model Steps zu, die für jedes Wahlmodell (Spalte „ChoiceModel“) Nachfrageschichten mit der dazugehörigen Modellspezifikation verbindet. In der Liste werden alle Modellschritte aufgelistet, die während eines Programmlaufs durchgeführt werden. Die dabei verwendete Reihenfolge hat keinen Einfluss auf die Reihenfolge der Teilmodelle, welche durch den Python-Code vorgegeben ist. Falls die datenbasierte Variante gerechnet werden soll, haben die Zeilen mit den Nummern 201 – 701 keine Bedeutung.

In jeder Zeile definiert die Spalte „ChoiceModel“, zu welchem Teilmodell die jeweilige Zeile gehört.

Die Nachfrageschicht wird durch die Spalte „Filter“ definiert: Nur solche Datensätze werden für die aktuelle Simulation ausgewählt, bei denen die angegebene Bedingung erfüllt ist. Die im Filter angegebenen Attribute beziehen sich dabei auf unterschiedliche Bezugsobjekte, abhängig vom jeweiligen Wahlmodell. So ist z.B. das Bezugsobjekt bei den langfristigen Entscheidungen „Person“, während es bei der Wahl der Anzahl der Zwischenziele „Tour“ ist. Die unten folgende Tabelle im Abschnitt Modellparameter liefert eine Übersicht zu den Bezugsobjekten bei den jeweiligen Wahlmodellen.

Die Spalte „Specification“ enthält den Namen der Tabelle, die die Modellparameter enthält (s.u.).

In manchen Fällen ist ein weiterer Parameter zur Modellspezifikation notwendig, der dann in der Spalte „AddData“ hinterlegt ist. Dies ist der Fall bei den ersten 11 Zeilen. Sie geben jeweils die ID der Aktivität an, für die die Wahl durchgeführt wird. Die Aktivitäten-ID ist in der Aktivitäten-Liste definiert. So bedeutet z.B. der Wert 5 in der ersten Zeile, dass die Standort-Wahl für die Aktivität E1 (Education primary Pupil) durchgeführt wird.

## Modellparameter

Die anderen Tabellen enthalten Modellattribute und die zugehörigen Parameter (siehe folgende Tabelle). Die Strukturen der Listen unterscheiden sich und hängen mit dem jeweils implementierten Wahlmodell zusammen. Die genaue Bedeutung der jeweiligen Einträge in den Listen ergibt sich aus dem Programm-Code.

**Tabellen für die modellbasierte und datenbasierte Variante:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tabelle | Wahl | Bezugsobjekt | Modell |
| PrimLocStu | Wahl der Schule/des Ausbildungsplatzes | Person | Zielwahlmodell |
| PrimLocAdult | Wahl des Arbeitsplatzes | Person | Zielwahlmodell |
| DestMajorStu | Zielwahl für die Hauptaktivität für Schüler und Studenten | Aktivitäts-ausübung am Ziel des Trips | Zielwahlmodell |
| DestMajorAdult | Zielwahl für die Hauptaktivität für Erwachsene | Aktivitäts-ausübung am Ziel des Trips | Zielwahlmodell |
| ModeMajorStu | Moduswahl für den Trip zur Hauptaktivität für Schüler und Studenten | Hauptaktivitäts-ausübung | Standardwahlmodell |
| ModeMajorAdult | Moduswahl für den Trip zur Hauptaktivität für Erwachsene | Hauptaktivitäts-ausübung | Standardwahlmodell |
| DestMinorStu | Zielwahl für Nebenaktivitäten für Schüler und Studenten | Aktivitäts-ausübung am Ziel des Trips | Zielwahlmodell |
| DestMinorAdult | Zielwahl für Nebenaktivitäten für Erwachsene | Aktivitäts-ausübung am Ziel des Trips | Zielwahlmodell |
| ModeMinorStu | Moduswahl für Trips zu Nebenaktivitäten für Schüler und Studenten | Aktivitäts-ausübung am Start oder Ziel des Trips | Standardwahlmodell |
| ModeMinorAdult | Moduswahl für Trips zu Nebenaktivitäten für Erwachsene | Aktivitäts-ausübung am Start oder Ziel des Trips | Standardwahlmodell |

**Tabellen für die modellbasierte Variante:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tabelle | Wahl | Bezugsobjekt | Modell |
| TourFreqE | Anzahl der Ausbildungstouren | Person | Standardwahlmodell |
| TourFreqW | Anzahl der Arbeitstouren | Person | Standardwahlmodell |
| TourFreqO | Anzahl sonstiger Touren | Person | Standardwahlmodell |
| StopFreqInbW | Anzahl Zwischenziele einer Arbeitstour zwischen Arbeitsplatz und Wohnort | Tour | Standardwahlmodell |
| StopFreqInbE | Anzahl Zwischenziele einer Ausbildungstour zwischen Schule/Ausbildungs-platz und Wohnort | Tour | Standardwahlmodell |
| StopFreqInbO | Anzahl Zwischenziele einer sonstigen Tour zwischen Hauptaktivität und Wohnort | Tour | Standardwahlmodell |
| StopFreqOutbW, StopFreqOutbE, StopFreqOutbO | Anzahl Zwischenziele einer Tour zwischen Wohnort und Hauptaktivität | Tour | Standardwahlmodell |
| SubFreqW | Anzahl Zwischentouren einer Arbeitstour | Tour | Standardwahlmodell |
| SubFreqE | Anzahl Zwischentouren einer Ausbildungstour | Tour | Standardwahlmodell |
| ActType | Typ einer Aktivität | Aktivitäts-ausübung | Standardwahlmodell |
| ActDur | Dauer einer Aktivitätsausübung | Aktivitäts-ausübung | Stetiges Wahlmodell |
| StartTime | Wahl der Startzeit einer Tour | Aktivitäts-ausübung der Hauptaktivität | Komplexes Wahlmodell (siehe Python-Script-Code) |

## Teil-Modelle

Im Folgenden werden alle Teil-Modelle des ABM, also die im Python-Script implementierten Wahlmodelle, inhaltlich beschrieben. Zur besseren Abgrenzung verwenden wir die folgenden Begriffsdefinitionen:

* primär: eine Aktivität wird als „primär“ bezeichnet, wenn es sich um eine Pflicht-Aktivität (in diesem Modell Arbeit oder Ausbildung) handelt. Die Pflichtaktivität wird immer am gleichen Standort ausgeführt: es ist derjenige, der bei der Wahl der langfristigen Entscheidungen gewählt wurde. Eine Tour wird „primär“ genannt, wenn sie eine Pflicht-Aktivität enthält.
* sekundär: Aktivitäten, Touren und Standorte werden „sekundär“ genannt, wenn es sich nicht um primäre handelt.
* Haupt-: eine Aktivität ist eine Hauptaktivität, wenn es die bestimmende oder zentrale Aktivität einer Tour ist. Das Hauptziel und der Hauptmodus einer Tour wird auf Grundlage des Trips vom Heimatstandort zum Standort der Hauptaktivität bestimmt. Die Zwischenziele (der Nebenaktivitäten) reihen sich zwischen diesen beiden Standorten ein.

Im Falle einer primären Tour ist die zugehörige primäre Aktivität immer auch die Hauptaktivität.

* Neben-: eine Aktivität wird als Nebenaktivität bezeichnet, wenn sie keine Hauptaktivität ist.

Jeder Modellschritt bezieht sich auf ein bestimmtes Visum-Objekt. So bezieht sich z.B. die Wahl der Standorte der primären Aktivitäten auf die Person. Das Ergebnis wird also in ein Personenattribut geschrieben. Die Filterdefinitionen (in der benutzerdefinierten Tabelle Model Steps) sowie die Nutzendefinitionen (in der jeweiligen Modellspezifikationstabelle) beziehen sich jeweils auf dieses Objekt, d.h. die dort angegebenen Attribute sind Attribute dieses Objekts (bei dem eben angesprochenen Beispiel handelt es sich also immer um Personenattribute).

Die Überschriften der folgenden Abschnitte enthalten in Klammern jeweils den Namen des zugehörigen Wahlmodells aus der Tabelle Model Steps.

**Teilmodelle für die modellbasierte und datenbasierte Variante**

Langfristige Entscheidungen: Standorte der primären Aktivitäten (PrimLoc)

Zu jeder Person und jeder von dieser Person wählbaren primären Aktivität wird ein zugehöriger Standort gewählt. Diese Wahlen entsprechen den langfristigen Entscheidungen zu Arbeits- und Ausbildungsplatz. Der Unterschied zu anderen Wahlentscheidungen ist, dass sie bei Szenarienberechnungen häufig nicht noch einmal simuliert werden, da sich diese Entscheidungen im Normalfall nicht aufgrund von kleineren Änderungen der verkehrlichen Situation verändern.

Die Wahl erfolgt je Nachfrageschicht auf Grundlage von Erreichbarkeiten und Zielpotential, und zwar über alle Nachfrageschichten hinweg zielgekoppelt. Sie entspricht damit einer typischen Zielwahl in makroskopischen Modellen.

Die Erreichbarkeit ergibt sich aus einer LogSum-Matrix, die sich aus den Fahrzeiten aller verfügbaren Verkehrsmittel zusammensetzt (siehe auch oben unter Punkt 4 im Abschnitt „Modellberechnung“). Die ID der Aktivität, für welche der Standort gewählt wird, ist in der Spalte „AddData“ der Tabelle Model Steps definiert.

Die Wahl der langfristigen Entscheidung findet, wie alle Zielwahlen in diesem Modell, in zwei Schritten statt: zunächst wird eine makroskopische Wahl auf Basis von Bezirken durchgeführt, die dann im zweiten Schritt auf Standorte diskretisiert wird. Das Zielpotential entstammt in beiden Fällen den Zielpotentialen der Aktivitätenstandorte, wobei beim ersten Schritt über alle im Bezirkspolygon befindlichen Standorte aggregiert wird.

Die Spezifikation des Modells befindet sich in den Tabellen „PrimLocStu“ und „PrimLocAdult“, das Bezugsobjekt ist Person. Sie besteht je Bevölkerungssegment nur aus einer Zeile. Der Nutzen eines Ziels definiert sich entsprechend als

u = Parameter \* Matrix().

Bei Bedarf können an dieser Stelle weitere Matrix- und Attribut-Ausdrücke hinzugefügt werden, die im Wahlmodell zu einem Gesamtnutzen addiert würden. Das Wahlmodell ist ein auf den Nutzen basierendes Logit-Modell.

Zielwahl für Hauptaktivitäten (DestMajor)

Zu jeder Tour wird das Ziel der Hauptaktivität (= Hauptziel) gewählt. Im Falle von primären Touren ist die Wahl trivial: es ist jeweils der schon zu Beginn definierte primäre Standort (siehe langfristige Entscheidungen). In den anderen Fällen erfolgt die Zielwahl ähnlich wie die Wahl der primären Standorte. Es wird allerdings keine Zielkopplung gerechnet, da diese schon bei den langfristigen Entscheidungen berücksichtigt wurde.

Die Spezifikation des Modells befindet sich in den Tabellen „DestMajorStu“ und „DestMajorAdult“. Sie sind ähnlich aufgebaut wie beim Wahlmodell der Standorte der primären Aktivitäten (PrimLoc). Das Bezugsobjekt ist jedoch die Aktivitätsausübung.

* Attribute expression: Skalare sowie Personenfilter, falls Nutzenelemente nur für Teil-Populationen gelten sollen
* MatrixExpr: Falls neben der LogSumme weitere Kenngrößen, wie z.B. Kalibrierungsmatrizen, verwendet werden sollen.
* ModeChoiceTable: Dieser Eintrag bezieht sich auf die LogSumme, die auf Grundlage der angegebenen Moduswahl-Tabelle berechnet wird. Da die Zielwahl auf Bezirksmatrizen, die Moduswahl jedoch auf Kurzwegsuchen basiert, muss bei der Moduswahl noch angegeben werden, welche Matrizen den Kurzwegsuchen entsprechen und diese bei der Zielwahl ersetzen. Dies erfolgt in der Spalte „DestinationImpedance“ in der Spezifikationstabelle der Moduswahl. Falls es sich um zeitlich variierende Matrizen handelt, muss dies in der Matrixdefinition durch den Zusatz   
   „[FromTime] = CONTEXT[FromTime] & [ToTime] = CONTEXT[ToTime])“   
  definiert sein.

Zur Nutzenberechnung werden je Zeile alle Komponenten miteinander multipliziert, also

Attribute expression \* MatrixExpr \* LogSumme(ModeChoiceTable).

Der Nutzen ergibt sich dann aus der Addition aller Zeilenergebnisse.

Die Modellparameter der Ziel- und Moduswahl sind miteinander verschränkt, da die Nutzendefinitionen der Moduswahl als LogSumme in die Zielwahl eingehen. Deshalb wird die Zielwahl auch durch die Tabellen „ModeMajorStu“ und „ModeMajorAdult“ definiert.

Moduswahl für Hauptaktivitäten (ModeMajor)

Zu jeder Tour wird der Hauptmodus der Tour bestimmt. Er wird auf Grundlage der Reisezeiten (hin und zurück) zwischen Wohnort und Hauptziel gewählt (auch, wenn es die Wege „Wohnort à Hauptziel“ bzw. „Hauptziel à Wohnort“ wegen eventueller Zwischenhalte gar nicht gibt). Ist der gewählte Modus ein austauschbarer Modus, so wird der Hauptmodus noch nicht konkret festgelegt, sondern als „austauschbar“ definiert. Die konkrete Wahl des Modus erfolgt dann in einem späteren Schritt.

Die Spezifikation des Modells befindet sich in den Tabellen „ModeMajorStu“ und „ModeMajorAdult“. Sie ist sehr ähnlich aufgebaut wie die Listen in den vorigen Abschnitten. Folgende Parameter sind zu setzen:

* Attribute expression: Skalare sowie Personenfilter, falls Nutzenelemente nur für Teil-Populationen gelten sollen
* MatrixExpr: Falls der Widerstand aus den Kurzwegen auf Matrixebene korrigiert werden soll
* DestinationImpedance: Dieser Eintrag bezieht sich auf die Bildung der LogSumme im Zielwahlmodell.
* ModeImpedance: Hier sind die Einträge PrTShortestPathImpedance(), PuTShortestPathImpedance() sowie DirectDistance() möglich.
  + Das Argument der Funktion PrTShortestPathImpedance() ist der Name des Attributs, welches den Widerstand enthält. Es muss für alle Objekte definiert sein, die Teil des IV-Widerstands sind, also für Strecken, Knoten, (Ober-)Abbieger, Anbindungen sowie Verkehrsbereiche. In der Beispielversion werden diese Attribute im Verfahrensablauf erzeugt und belegt. Falls es sich um zeitlich variierende Attribute handelt, wird der entsprechende Zeitpunkt bei der Moduswahl automatisch korrekt berücksichtigt.
  + Das Argument der Funktion PuTShortestPathImpedance() ist der Name der entsprechenden Nachfrageschicht. Die ÖV-Kurzwegsuche basiert auf Haltestellenbereichsmatrizen des Widerstands (Code IPD) sowie der Reisezeit (Code RIT und ADT). Dabei werden auch das NSeg und das Zeitintervall entsprechend berücksichtigt. Der Fußweg zum Haltestellenbereich wird im Netz bezüglich des im globalen Attribut WalkPrTSys hinterlegten VSys berechnet (dieses wird in der benutzerdefinierten Tabelle Global attributes definiert). Die Gewichtung des Fußwegs relativ zum Widerstand aus der Haltestellenbereichsmatrix ist im globalen Attribut WalkTimeImpedanceFactor definiert, die maximale Gehzeit im Attribut MaxWalkTimeInMinutesForPutShortestPathSearch.
  + DirectDistance() besitzt kein Argument. Die Umrechnung der Luftliniendistanz in Gehzeit erfolgt über das globale Attribut DirectDistanceSpeed aus der Tabelle Global Attributes.
* Bike, Car, etc.: Parameter für die jeweiligen Modi. Sie werden identisch wie bei makroskopischen Modellen gewählt.

Zur Nutzenberechnung werden alle Parameter einer Zeile miteinander multipliziert und über alle Zeilen addiert.

Ziel- und Moduswahl für Nebenaktivitäten

Die Ziel- und Moduswahlen für Nebenaktivitäten erfolgen gemeinsam. Nacheinander werden Ziele und die zugehörigen Modi der Wege zu bzw. von den Zielen bestimmt. Dabei wird mit dem ersten Ziel vor der Hauptaktivität begonnen, und dann nacheinander aller Aktivitätenausübungen behandelt, bis die Home-Aktivität erreicht ist. Danach wird mit dem ersten Ziel nach der Hauptaktivität begonnen und wieder gerechnet, bis die Home-Aktivität erreicht ist.

Die Filter, die in den Model Steps zur Ziel- und Moduswahl für Nebenaktivitäten definiert sind, dürfen sich nicht überschneiden und müssen zusammen genommen alle Aktivitätenausübungen abdecken.

Zielwahl für Nebenaktivitäten (DestMinor)

Zu jeder Aktivität, die bisher noch keinen Standort besitzt, wird dieser gewählt. Die Wahl erfolgt ähnlich der Zielwahl für Hauptaktivitäten, allerdings unter Verwendung der Rubberbanding-Methode.

Bei der Rubberbanding-Methode werden die Zwischenziele zwischen Heim- und Hauptaktivität sukzessive eingefügt. Dabei basiert die Wahl auf der Summe der Fahrzeiten zwischen dem neuen Ziel und den jeweils vorangegangenen und nachfolgenden Zielen. Wird also z.B. ein Ziel zwischen die Heimat- und Hauptaktivität eingefügt, so basiert die Wahl dieses Zieles auf der Summe der Fahrzeiten zwischen Heimataktivität und neuem Ziel sowie zwischen neuem Ziel und Hauptaktivität. Bei der Wahl der Zwischenziele werden also immer die Fahrten zum und vom Zwischenziel berücksichtigt (und tendenziell möglichst minimal gewählt).

Die Spezifikation des Modells ist identisch mit der Spezifikation der Zielwahl der Hauptaktivitäten (DestMajor).

Moduswahl für Nebenaktivitäten (ModeMinor)

Zu jeder Aktivitätenausübung wird der Modus des zugehörigen Trips gewählt. Entsprechend der oben angesprochenen Abarbeitungsreihenfolge ist abwechselnd der Trip zur oder von der Aktivitätenausübung gemeint.

Die zur Verfügung stehenden Modi sind der Hauptmodus und bei Zwischentouren zusätzlich die austauschbaren Modi.

Im Anschluss an die Moduswahl wird allen Trips ihre Fahrzeit aus den Kurzwegsuchen zugeordnet.

**Teilmodelle für die modellbasierte Variante**

Wahl der Anzahl an primären Touren (TourFreqPrim)

Zu jeder Person wird für jede von dieser Person wählbaren primären Aktivitäten eine zugehörige Anzahl an Touren gewählt. Die ID der Aktivität, für welche die Anzahl der Touren gewählt wird, ist in der Spalte „AddData“ der Tabelle Model Steps definiert.

Bei der Definition der Bevölkerungssegmente (also der Filter in der Tabelle Model Steps) muss darauf geachtet werden, dass diese mit den entsprechenden Definitionen in der Wahl der Standorte der primären Aktivitäten (PrimLoc) kompatibel sind. Wird für eine Person die Anzahl an primären Touren gewählt, muss bereits ein Standort für die entsprechende Aktivität gewählt worden sein. Ist dies nicht so, bricht das Python-Programm ab und es erscheint eine Fehlermeldung.

Die Spezifikation des Modells befindet sich in den Tabellen „TourFreqE“ und „TourFreqW“, das Bezugsobjekt ist Person. Die Spalten Result0, Result1, Result2 und Result3 enthalten die Nutzenparameter zu den jeweiligen Alternativen. Beachten Sie, dass der Spaltenname keine Auswirkung auf das Modell hat. Der Wert der Alternative ist in der ersten Zeile (mit Attribute expression = ChoiceID) definiert.

Der Nutzen einer Alternative ergibt sich als Summe der Produkte aus den Attributausdrücken und ihren zugehörigen Parametern. Bei Bedarf können an dieser Stelle weitere Attribut-Ausdrücke hinzugefügt werden. Falls eine weitere Alternative hinzugefügt werden soll, muss dazu ein weiteres benutzerdefiniertes Attribut eingefügt werden (in diesem Fall würde sich ein BDA mit Namen „4“ anbieten).

Das Wahlmodell ist das oben unter Punkt 7 im Abschnitt „Modellberechnung“ angesprochene Standardwahlmodell für diskrete Entscheidungen mit wenigen Alternativen.

Wahl der Anzahl an sekundären Touren (TourFreqSec)

Zu jeder Person wird die Anzahl an sekundären Touren gewählt.

Die Spezifikation des Modells befindet sich in der Tabelle „TourFreqO“. Sie ist identisch aufgebaut wie die Listen im vorigen Abschnitt. Das Bezugsobjekt ist die Person.

Das Wahlmodell ist identisch zum obigen Standardwahlmodell.

Wahl der Anzahl an Zwischenhalten (TourStopFreq)

Zu jeder Tour wird die Anzahl zusätzlicher ein- und ausgehender Aufenthalte gewählt (inbound and outbound stops). Ausgehende Aufenthalte entsprechen dabei Zwischenaktivitäten, die auf dem Weg zur Hauptaktivität liegen.

Das Wahlmodell entspricht wieder dem Standardwahlmodell.

Die Spezifikation des Modells befindet sich in den Tabellen „StopFreqInbE“, „StopFreqInbO“, „StopFreqInbW“, „StopFreqOutbE“, „StopFreqOutbO“ und „StopFreqOutbW“. Sie sind identisch aufgebaut wie die Listen in den vorigen Abschnitten. Das Bezugsobjekt ist die Tour, das Wahlmodell entspricht wieder dem Standardwahlmodell.

Wahl der Anzahl an Zwischentouren (SubtourFreq)

Zu jeder primären Tour wird entschieden, ob es eine Zwischentour gibt und wenn ja, wieviel Zwischenhalte diese hat. Eine Zwischentour ist eine Tour, die bei dem Standort startet und endet, an dem die primäre Aktivität ausgeführt wird. Sie hat ein oder zwei Aufenthalte, nach denen der Weg wieder zurück zum Ausgangstandort führt. Zwischentouren können nur bei primären Touren auftreten.

Die Spezifikation des Modells befindet sich in den Tabellen „SubFreqE“ und „SubFreqW“. Sie sind identisch aufgebaut wie die Listen in den vorigen Abschnitten. Das Bezugsobjekt ist die Tour, das Wahlmodell entspricht wieder dem Standardwahlmodell.

Beim Wahlmodell stehen als Alternativen 0, 1, oder 2 zur Verfügung. Die Alternative 0 steht für „keine Zwischentour“, 1 steht für „Zwischentour mit einem Aufenthalt“, 2 steht für „Zwischentour mit 2 Aufenthalten“.

Wahl des Aktivitätentyps (ActType)

Zu jedem Aufenthalt wird die zugehörige Aktivität gewählt. Da primäre Aktivitäten nur als Hauptaktivitäten von primären Touren auftauchen, können an dieser Stelle nur sekundäre Aktivitäten gewählt werden. Insbesondere werden hier auch die Hauptaktivitäten (also die zentralen Aktivitäten) von sekundären Touren (also von Touren, die keine primäre Aktivität besitzen) festgelegt.

Die Spezifikation des Modells befindet sich in der Tabelle „ActType“. Sie ist identisch aufgebaut wie die Listen in den vorigen Abschnitten. Es ist zu beachten, dass die Alternativausprägung (also der Aktivitäten-Typ) nicht durch die Spaltenüberschrift (also den Namen des benutzerdefinierten Attributs) definiert wird. Die Ausprägungen der zu wählenden Alternativen befindet sich in der Zeile mit Attribute expression = ChoiceID. Hier stehen die zu den Aktivitäten-Typen gehörenden IDs. Die IDs sind in der Aktivitätenliste definiert.

Das Bezugsobjekt ist die Aktivitätsausübung, das Wahlmodell entspricht wieder dem Standardwahlmodell.

Wahl der Dauer der Aktivitätsausübungen (ActDur)

Zu jeder Aktivität wird die Dauer ihrer Ausübung gewählt. Das Wahlmodell unterscheidet sich von allen anderen bisherigen diskreten Wahlmodellen. Zu jeder Nachfrageschicht gehört eine geeignete Verteilungsfunktion mitsamt bestimmter Parameter, auf deren Grundlage dann die Dauer der Aktivitätsausübungen gewählt wird.

Die Spezifikation des Modells befindet sich in der Tabelle „ActDur“, das Bezugsobjekt ist die Aktivitätsausübung. Die Listen sind etwas anders aufgebaut als die vorherigen Listen. Das Attribut ‚Attribute expression‘ fungiert hier wie ein Filter: die dort angegebene Bedingung definiert die Nachfrageschicht. Die weiteren Parameter definieren die Verteilungsfunktion für die Dauer der Aktivitätenausübung.

Die Modelle für Aktivitäten aus Zwischentouren und sonstigen Aktivitäten unterscheiden sich. Die Dauern für Zwischentouren sind im Mittel kürzer.

Es stehen drei verschiedene Verteilungsfunktionen für die Aktivitätendauern zur Verfügung: Weibull, Normal und LogNormal.

Wahl der Tageszeit (StartTime)

Bei der Tageszeitwahl wird je Tour nur die Startzeit der Hauptaktivitätsausübung gewählt. Alle anderen Zeiten ergeben sich daraus automatisch, da sowohl die Dauern der Aktivitätsausübungen als auch die Fahrzeiten (aus den entsprechenden Fahrzeitenmatrizen) bekannt sind. Die Startzeiten der Aktivitätsausübungen werden so gewählt, dass über alle Aktivitätsausübungen aggregiert die zur Aktivität gehörige Ganglinie möglichst gut getroffen wird.

Das im Skript-Beispiel implementierte Modell zur Wahl der Tageszeit ist ein eher einfaches Modell und entspricht nicht, im Gegensatz zu den meisten anderen hier eingesetzten Teil-Modellen, einer nutzenbasierten Wahl. Es reproduziert so gut wie möglich die fest vorgegebene Verteilung des Beginns einer Aktivitätsausübung und ist damit nicht sensitiv gegenüber Angebotsänderungen. Die Startzeiten verschiedener Touren werden nicht auf Überlappungsfreiheit überprüft. Es kann somit vorkommen, dass Touren einer Person parallel ablaufen.

Für viele Analysen ist ein solches Modell völlig ausreichend. Falls Sie allerdings Modifikationen wie beispielsweise sich zeitlich ändernde Angebote analysieren möchten, wäre ein anderes Wahlmodell der Tageszeit notwendig.

Die Spezifikation des Modells befindet sich in der Tabelle „StartTime“, das Bezugsobjekt ist die Aktivitätsausübung der Hauptaktivität einer Tour. Die Liste ist ähnlich der vorigen aufgebaut. Das Attribut ‚Attribute expression‘ fungiert wieder als Filter zur Definition der Nachfragesegmente. Das Attribut ‚TimeSeriesNo‘ enthält die Nummer der Standard-Ganglinie. Die Ganglinien findet man im Menu Nachfrage > Nachfragedaten im Register Standardganglinien.

# Tipps und Tricks

Die folgende Liste ist aus unseren Erfahrungen in diversen Pilotprojekten hervorgegangen. Wird das vorliegende Visum-Beispiel als Ausgangspunkt für die Entwicklung eines eigenen ABM verwendet, so sollten Sie die folgenden Punkte beachten:

* Im Allgemeinen werden Sie andere Personenattribute verwenden als im Beispiel vorliegen. Das ist problemlos möglich, solange Sie darauf achten, dass die entsprechenden Attribute existieren und dass die Filter in der Tabelle Model Steps sowie die Nutzendefinitionen in den Tabellen der Modellparameter angepasst werden. Es hat sich dabei als nützlich erwiesen, bei Anpassungen der Nutzendefinitionen bzw. Filter gleichzeitig die entsprechende Liste geöffnet zu halten, welche die Nutzen-Attribute enthält. Wird also z.B. die Nutzenfunktion zur Anzahl an primären Touren (TourFreqPrim) angepasst, sollte gleichzeitig die Personenliste geöffnet sein, damit bei der Wahl der Personenattribute keine Schreibfehler entstehen.
* Falls einer Nutzendefinition ein weiterer Term hinzugefügt werden soll, muss der entsprechenden benutzerdefinierten Tabelle eine Zeile hinzugefügt werden. In diese Zeile kann dann der zusätzliche Nutzenterm geschrieben werden.
* Falls bei einer Modellspezifikation eine weitere Alternative hinzugefügt werden soll, muss dazu ein weiteres benutzerdefiniertes Attribut der zugehörigen Tabelle eingefügt werden. Der Name des Attributes sollte sich aus Gründen der Übersichtlichkeit an der Ausprägung der neuen Alternative orientieren. In der Tabellenzeile, deren Eintrag bei Attribute expression „ChoiceID“ ist, muss die Ausprägung der neuen Alternative angegeben werden.
* Beim vorliegenden Beispiel gibt es für die meisten Wahlmodelle maximal zwei Modellparametersätze: einen für Erwachsene und einen für Schüler. Falls bei einem Wahlmodell ein weiterer Parametersatz angelegt werden soll (z.B. für Rentner), muss dazu eine neue benutzerdefinierte Tabelle hinzugefügt werden. Diese Tabelle muss dann in der Tabelle Model Steps mit dem Teil-Modell sowie der entsprechenden Nachfrageschicht verbunden werden.
* Während des Modellierungsprozesses ist es sinnvoll, das ABM Schritt für Schritt durchzurechnen und die Ergebnisse jeden Schritts zu prüfen, bevor der nächste Schritt berechnet wird. Sie können dies sehr einfach realisieren, indem Sie im Python-Skript die schon getätigten Schritte auskommentieren. Dazu können Sie in der Datei abm.py drei doppelte Anführungszeichen als Start und Ende der auskommentierten Bereiche einfügen. Soll also z.B. nur der Schritt TourFreq kalibriert werden, so würde man im Skript abm.py die Funktion tripchain\_choice wie folgt anpassen:

def tripchain\_choice(self):

...

logging.info('--- tour frequency choice ---')

self.tour\_frequency\_choice()

"""

logging.info('--- stop frequency choice ---')

self.stop\_frequency\_choice()

logging.info('--- subtour choice ---')

self.subtour\_choice()

...

"""

Bitte bedenken Sie dabei, dass die ersten Zeilen (bis einschließlich self.initialize\_data\_and\_config()) nicht auskommentiert werden dürfen.

* Während der Entwicklung des Modells kann es immer wieder notwendig sein, das Modell zu debuggen, also Breakpoints zu setzen und so Zwischenstände des Skriptablaufs zu analysieren. Dieser Vorgang kann erheblich erleichtert werden, wenn dazu eine Entwicklungsumgebung wie Visual Studio Code verwendet wird. Die Installation ist in den meisten Fällen recht unkompliziert (siehe dazu auch die Hinweise weiter unten).
* Da im Verlauf der Modellierung häufig wiederholte Modellberechnungen notwendig sind, ist es ratsam, zunächst nur eine geringe Menge an Personen zu modellieren. Dadurch kann die Rechenzeit während des Modellaufbaus gering gehalten werden.

# Code-Überblick

Dieser Abschnitt gibt einen kurzen Überblick zum Skript-Code. Er dient dem Einstieg und der besseren Orientierung im Code. Es handelt sich allerdings nicht um eine vollständige Spezifikation des Codes. Um genauere Informationen zur Umsetzung des Modells zu erhalten, sollten Sie sich den Code direkt anschauen.

Das Skript ist in Python geschrieben und kann mit den üblichen Editoren geöffnet und bearbeitet werden.

## Überblick

Das ABM-Modell ist in mehrere Skripte aufgeteilt, die normalerweise aus dem Verfahrensablauf heraus gestartet werden. Zu Debugging-Zwecken kann es auch von einer Entwicklungsumgebung heraus gestartet werden.

Das Modul abm enthält die Klasse ABM, die die eigentliche Berechnung des ABM-Modells beinhaltet. Die Einzelschritte sind im Untermodul src.core implementiert. Gemeinsam genutzte Funktionen des Wahlmodells befinden sich in src.choice\_engine, src.location\_choice\_engine sowie src.mode\_choice\_engine. Das Modul src.config ermöglicht das Lesen der Konfiguration aus den Tabellen der Versionsdatei. Weitere nützliche Funktionen sind in src.utilities zu finden.

## Vorbereitung

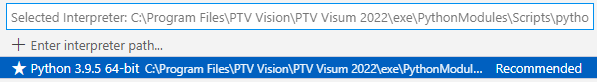
Wenn das Modell debuggt werden soll, ist eine Python-Entwicklungsumgebung empfehlenswert. Unterstützt werden unter anderem PyCharm Community Edition (JetBrains) und Visual Studio Code (Microsoft). Ebenfalls ist es möglich, das Skript aus einer Konsole heraus aufzurufen.

Alle benötigten Python-Bibliotheken werden durch das Visum-Setup bereits mitinstalliert. Die Ausführung muss mit Python 3 erfolgen. Python 2 wird nicht unterstützt.

Mit dem Visum-Setup wird für Python 3 eine virtuelle Umgebung (venv) installiert, in die zusätzliche Pakete, die sich im Unterordner Exe\Python37Modules des Visum-Installationspfades befinden, eingebunden sind. Bei Ausführung des Skriptes muss diese virtuelle Umgebung genutzt werden.

In Visual Studio Code kann die verwendete Umgebung beispielsweise in der linken unteren Fensterecke eingestellt werden.





Um die ABM-Berechnung auszuführen, muss lediglich die Datei run\_abm.py ausgeführt werden. Die dort definierte Methode run\_abm enthält mehrere Berechnungsoptionen, von denen die geeignete aktiv gesetzt werden muss. Wird das Programm nicht von Visum aus gestartet, ist darauf zu achten, dass die virtuelle Umgebung von Visum genutzt wird.

Wird das Programm nicht von Visum aus gestartet, wird zunächst ein Visum gestartet und die im Python-Code angegebene Versionsdatei (Halle\_ABM.ver) geladen. Andernfalls wird die laufende Visum-Instanz automatisch verwendet und das Skript mit der aktuell geöffneten Versionsdatei ausgeführt.

Beachten Sie bitte, dass eine Berechnung des ABM innerhalb einer Entwicklungsumgebung nicht identisch ist mit einer regulären Berechnung durch den Verfahrensablauf. Im Verfahrensablauf werden zwischen einzelnen Skript-Berechnungen diverse weitere Visum-Verfahren durchgeführt. So müssen z.B. zunächst alle Verfahrensschritte des Verfahrensablaufs durchgeführt werden, die vor der ABM-Gruppe stehen, bevor das Skript run\_abm.py von einer Entwicklungsumgebung heraus gestartet wird.

Es ist auch möglich, Einzelschritte des ABM-Skriptes separat auszuführen. Dazu müssen im Skript abm.py in der Methode run\_full\_abm lediglich solche Verfahren auskommentiert werden, die nicht durchgeführt werden sollen. Es ist zu beachten, dass die Methode ABM.initialize\_data\_and\_config() nicht auskommentiert werden darf.

## Logging

Über das Modul logging werden Debug-Ausgaben auf der Konsole ausgegeben. Wird das Skript innerhalb von Visum ausgeführt, so werden die Debug-Ausgaben in die Datei Messages.txt geschrieben.

## Verwendung von numpy und dask

Die Module numpy und dask dienet dem effizienten Umgang mit großen numerischen Datenstrukturen (Arrays/Vektoren, Matrizen und n-dimensionale Vektoren). Im ABM-Skript wird an vielen Stellen Gebrauch davon gemacht. So arbeitet unter anderem die gesamte Choice-Engine mit numpy-Datenstrukturen.

## Lesen der Konfigurationstabellen

Die Konfigurationstabellen werden über das src.config-Modul in der Methode ABM.initialize\_data\_and\_config() eingelesen und stehen während der Skript-Ausführung als Member (ABM.config) zur Verfügung.

Die Konfiguration eines einzelnen Verfahrensschritts wird mittels self.config.load\_choice\_para(‘[Name des Schrittes]‘) verfügbar gemacht und enthält eine Liste von Segmenten, die in diesem Schritt betrachtet werden (siehe Tabelle **Model Steps**).

## Durchführung einer einfachen Diskreten Wahl

In vielen ABM-Schritten wird eine klassische diskrete Wahl (discrete choice) durchgeführt. Diese wird segmentweise durchgeführt, wobei die Segmente in der Regel eine Partition (disjunkte Unterteilung) der Objekt-Menge darstellen. Als Objekt-Menge treten die Menge aller Personen, aller Aktivitätsausübungen sowie aller Touren und aller Trips auf.

In der diskreten Wahl gibt es häufig mehrere Aspekte, die die Wahl beeinflussen. Diese werden in mehreren Zeilen einer Wahl-Tabelle dargestellt. Die einzelnen Aspekte werden mit einem festen Gewicht Beta sowie einem variablen Gewicht multipliziert, das sich aus der Auswertung eines Formelausdrucks AttrExpr ergibt. Dieser Formelausdruck kann Eigenschaften des einzelnen Objekts berücksichtigen, beispielsweise das Alter einer Person.

Die diskrete Wahl wird durch einen Aufruf von choice\_engine.run\_simple\_choice(Objekt-Menge, Segment) durchgeführt. Dabei enthält das Segment-Objekt, das aus den Tabellen befüllt wurde, folgende Daten:

* Filter: Filter-Ausdruck, nach dem die Objekt-Menge zunächst gefiltert wird.
* Choices: Die diskrete Menge der Wahl-Möglichkeiten, in der Regel 0..n oder 1..n
* AttrExpr: Ein Formelausdruck pro Nutzenkomponente der Wahl. Dieser Ausdruck wird für jedes Objekt individuell ausgewertet und mit dem konstanten Gewicht multipliziert.
* Beta: Ein festes Gewicht pro Aspekt der Wahl.
* ResAttr: Ergebnis-Attribut, in das die gewählten Werte geschrieben werden.

## Effiziente Datenübertragung über COM

Bei der Übertragung größerer Datenmengen aus Visum an das Python-Skript und umgekehrt sind mehrere Dinge zu beachten.

* Für das effiziente Lesen und Schreiben der Daten sollten die Methoden VPH.GetMulti() und VPH.SetMulti() bzw. für mehrere Attribute Container.GetMultipleAttributes() und Container.SetMultipleAttributes() verwendet werden (VPH steht für das Modul VisumPy.helpers, das mit Visum ausgeliefert wird, Container für einen beliebigen Objekt-Container, bspw. Visum.Net.Persons).
* Beim Lesen und Schreiben sehr großer Datenmengen soll   
   visum\_utilities.SetMulti(container, attribute, values, active\_only=False, chunk\_size=abm\_settings.chunk\_size\_trips) verwendet werden. Werden für sehr viele Objekte die Daten auf einmal geholt, so kann das die maximale Übertragungsgröße von Pywin32 übersteigen.
* Wenn eine Objekt-Menge gefiltert wird, sollte die Methode Container.GetFilteredSet() verwendet werden, die direkt einen Container mit der gefilterten Objekt-Menge zurückliefert. Diese ist gegenüber der Methode Container.FilteredBy() zu bevorzugen, die erst bei der konkreten Verwendung der gefilterten Objekt-Menge ausgewertet wird, dann jedoch bei jedem einzelnen Aufruf auf dieser Menge erneut.

Diese effiziente Übertragung ist im ABM-Skript in jedem Schritt implementiert.

1. PopulationSim ist Teil des ActivitySim-Projekts und in Zusammenarbeit mit dem Oregon DOT (Department of Transportation) entstanden. Siehe auch https://activitysim.github.io/populationsim/ [↑](#footnote-ref-2)
2. Die Software PopulationSim bietet auch Funktionen an, die eine Evolution der Bevölkerung simulieren. [↑](#footnote-ref-3)
3. In sehr geringem Maße ist dies auch bei der datenbasierten Erzeugung möglich. [↑](#footnote-ref-4)
4. Falls die Erzeugung modellbasiert erfolgt, muss diese auch kalibriert werden. [↑](#footnote-ref-5)