Removing a activity :

In order to demonstrate these changes, we show how to remove an activity called "NewType"
The changes required in the config file is a single line as shown in red below:

Step – 1 : Changes in the config file

In the config file `data/simulation.xml`, we add an activity type as shown in green below

```
<activity_types>
 <activity_type name="Work" withinday_mode_choice="wdmw" num_tours="ntw" tour_mode_dest="tmdw"
tour_mode="tmw" tour_time_of_day="ttdw" type="1" logsum_table_column="work"/>
 <activity_type name="Education" withinday_mode_choice="wdme" num_tours="nte" tour_mode_dest=""
tour_mode="tme" tour_time_of_day="ttde" type="2" logsum_table_column="education"/>
 <activity_type name="Shop" withinday_mode_choice="wdmso" num_tours="nts" tour_mode_dest="tmds"
tour_mode="" tour_time_of_day="ttdo" type="3" logsum_table_column="shop"/>
 <activity_type name="Other" withinday_mode_choice="wdmso" num_tours="nto" tour_mode_dest="tmdo"
tour_mode="" tour_time_of_day="ttdo" type="3" logsum_table_column="other"/>
 <activity_type name="NewType" withinday_mode_choice="wdmso" num_tours="nto" tour_mode_dest="tmdo"
tour_mode="" tour_time_of_day="ttdo" type="3" logsum_table_column="newtype"/>
</activity_types>
```


While the activity names are made configurable, the main types of activities are NOT. The
types of activities as specified in the config file simrun_Midterm.xml must be the same as
the of activities as hardcoded in the `dev/Basic/shared/StopType.hpp`, as shown below :

```
StopType NULL_STOP = -1; StopType
WORK_ACTIVITY_TYPE = 1;
StopType EDUCATION_ACTIVITY_TYPE = 2;
StopType OTHER_ACTIVITY_TYPE = 3; StopType
WORK_BASED_SUBTOUR = 5 ;
```

Step - 2: Changes in lua files:

dps.lua:

   i) The activity type map should be updated: The new key for "NewType" is removed as
   shown in red below:

   local activity_types = { [**"Work"**] = 1, [**"Education"**] = 2, [**"Shop"**] = 3,

                                   [**"Others"**] = 4, ["NewType"] = 5 }

ii) Inside function compute_utilities(), delete the new variable for the logsum of the "NewType", shown in red.

```
local otherlogsum = params:activity_logsum(activity_types.Others)

local otherlogsum = params:activity_logsum(activity_types.Others)

local newTypeLogsum = params:activity_logsum(activity_types.NewType)
```

iii) The choice set should be altered. For example the choice set needs to be reduced d to include just 4 activities as shown in red below:

```
--choiceset local
choice = {
            {0,0,0,0,0},
            {1,0,0,0,0},
            {0,1,0,0,0},
            {0,0,1,0,0},
            {0,0,0,1,0},
            {0,0,0,0,1},
            {1,0,1,0,0},
            {1,0,0,1,0},
            {0,1,1,0,0},
            {0,1,0,1,0}


}
```

===
dpt.lua:

Changes same as dps.lua

===

dpb.lua:

i) The activity type map should be updated: The key for "NewType" is deleted as shown in red below:

```
local activity_types = { ["Work"] = 1, ["Education"] = 2, ["Shop"] = 3,

                          ["Others"] = 4, ["NewType"] = 5 }
```

ii) Inside function compute_utilities(), the variable for the logsum of the "NewType" activity, should be deleted as shown in red.

```
local otherlogsum = params:activity_logsum(activity_types.Others)

local otherlogsum = params:activity_logsum(activity_types.Others)

local newTypeLogsum = params:activity_logsum(activity_types.NewType)
```

==

isg.lua

i) Updating the choice vector. The choice 6 should be deleted as shown in red --choice set

--1 for work; 2 for edu; 3 for shopping; 4 for other ;5 for quit 5 for NewType; 6 for quit

```lua
local choice = { 1,
    2,
    3,
    4,
    5,
    6

}
```

ii) Inside `compute_utlities function()`: The lines in red to be deleted and the lines in green should be added as shown below:

utility[4] = beta_cons_other + beta_work_tour_dummy_O * 1 ....

utility[5] = beta_cons_other+beta_work_tour_dummy_O * 1......

utility[6] = beta_cons_other+beta_work_tour_dummy_O * 1......

Earlier, The utility[6] represented "quit". But after removing an activity_type, utility[5] represents for "quit".

utility[4] = beta_cons_other + beta_work_tour_dummy_O * 1 ....

utility[5] = beta_cons_other+beta_work_tour_dummy_O * 1......

3) Inside the `compute_availabilities()` function, the range of the for loop shoud be changed :

for i = 1, 6 **do**

for i = 1, 5 do

==

Step- 3: Changes in the logsum table

The logsum table should have (N+4) columns, where N is the number of Activity types allowed. For example when we have a total of 4 allowed activities:

Work,Education,Shopping,Other,and NewType , the create table query of the logsum table should look as shown below:

```
CREATE TABLE demand.preday_logsum_test_new_activity_aditi
(
  person_id bigint NOT NULL,
  work double precision,
  education double precision,
  shop double precision, other
  double precision, newtype
  double precision, dp_tour
  double precision, dp_stop
  double precision,
  CONSTRAINT preday_logsum_test_new_activity_aditi_pkey PRIMARY KEY (person_id)
)
```

Important Consideration: The order of the Activity names must be in accordance with the order specified in the simrun_MidTerm.xml config file