



CAMBRIDGE
SYSTEMATICS

Think  Forward

GISDK Workshop

Calculating Matrix-Based VMT

presented to

Caltrans District 12

presented by

Cambridge Systematics, Inc.

Sean McAtee

1/20/2017

Approach

- We will work together to create GISDK scripts to compute VMT per Capita for a set of TAZs
 - » Run for either Orange County or a custom set of zones
- We will discuss good coding practice, including
 - » File definition at the beginning
 - » Variable references rather than hard-coded values
 - » Extensive comments
- We will learn specific tools to support working with matrices
- We will introduce errors and practice debugging

1. Setup, Files, and Fields

➤ Create a Macro

- » Macro "Main"
- » EndMacro

➤ Define your Path

- » `pth = "C:\\SCAG Training\\16R16s3_set7_setting\\"`

Note double backslash
& trailing backslash

➤ Define required files

- » `taz_file = pth + "Geography\\TAZ_t1.dbd"`
- » `sed_file = pth + "SED\\Output\\model_sed.bin"`
- » `od_file = pth + "ODTable\\Outputs\\AM_OD.mtx"`
- » `skm_file = pth + "Skims\\Outputs\\SPMATRIX_DA_Tier1.mtx"`

Use relative paths

➤ Define zone set

- » `taz_gry = "Select * Where CNTY = 3"`

GISDK requires
Select * Where ...



2a. Open all required files

➔ Open the TAZ File

Not documented in
TransCAD help!

» `{taz_lyr} = RunMacro("TCB Add DB Layers", taz_file, ,)`

» Alternate:

- `lyr_name = GetDBLayers(taz_file)`
- `taz_lyr = AddLayerToWorkspace(lyr_name, taz_file, lyr_name,)`

Syntax

`actual_name = AddLayerToWorkspace(string layer_name, string db_name, string db_layer_name [, array options])`

Argument	Contents
<code>layer_name</code>	The desired name for the new layer
<code>db_name</code>	Path and file name of the geographic file that contains the layer
<code>db_layer_name</code>	The name of the layer in the geographic file

2b. Open all required files

➤ Work in a copy of the original OD matrix

- » We will change the OD file, but not the skim file
- » `t = SplitPath(od_file)`
- » `od_temp = t[1] + t[2] + "__TEMP__" + t[3] + t[4]`
- » `CopyFile(od_file, od_temp)`

➤ Open the matrices

- » `od_mat = OpenMatrix(od_temp,)`
- » `skm_mat = OpenMatrix(skm_file,)`

3. Compute Total Trips

➤ Compute the total trips

Input Currency:

1. Matrix file
2. Core name
(any valid name will work in this case)
3. Row and column index
(blank uses primary index)

- » `Opts = null`
- » `Opts.Input.[Input Currency] = {od_temp, "DA", , }`
- » `RunMacro("TCB Run Operation", "Matrix Quicksum", Opts)`

➤ Alternate Method:

- » `AddMatrixCore(od_mat, "TOT")`
- » `cur = CreateMatrixCurrencies(od_mat, , ,)`
- » `cur.TOT := cur.DA + ... //or loop through cores`



4a. Compute VMT

- ➔ Add a temporary matrix core to hold Length x Trips
 - » `AddMatrixCore(od_mat, "VMT")`
- ➔ Review files to find a common Index
 - » Same number of records
 - 4109 is the only common set
 - » Exact same IDs
 - Only available are sequential TAZ, not Sequential IDs

Index Name	Type	# Records
Rows	Rows	4192
NodeIDs	Rows & Columns	4192
Ids_truck	Rows & Columns	4192
idinternal	Rows & Columns	4192
InternalTAZ	Rows & Columns	4109
Columns	Columns	4192

Index Name	Type	# Records
Origin	Rows	4484
OnlyZones	Rows & Columns	4109
idinternal	Rows & Columns	4109
Destination	Columns	4484

4b. Compute VMT

➤ Add a VMT Core

- » `AddMatrixCore(od_mat, "VMT")`

➤ Create matrix currencies

- » `vmt_cur = CreateMatrixCurrency(od_mat, "VMT", "InternalTAZ", "InternalTAZ",)`

- » `trip_cur = CreateMatrixCurrency(od_mat, "TOT", "InternalTAZ", "InternalTAZ",)`

- » `len_cur = CreateMatrixCurrency(skm_mat, "Length (Skim)", "OnlyZones", "OnlyZones",)`

➤ Calculate VMT for each cell

- » `vmt_cur := trip_cur * len_cur`

5. Create Matrix Indices

➤ Select Zones

- » `SetLayer(taz_lyr)`
- » `cnt = SelectByQuery("Subarea", "Several", taz_qry,)`

➤ Create New Index

- » `CreateMatrixIndex("Subarea", od_mat, "Both", taz_lyr+"|Subarea", "ID", null)`

Must be the primary index from the matrix file, matched to the TAZ layer



6. Compute To + From VMT

➤ Create Matrix Currencies

- » `to_cur = CreateMatrixCurrency(od_mat, "VMT", "Rows", "Subarea",)`
- » `fr_cur = CreateMatrixCurrency(od_mat, "VMT", "Subarea", "Columns",)`

➤ Get Totals

- » `to_vector = GetMatrixVector(to_cur, {"Marginal", "Row Sum"})`
- » `to_sum = VectorStatistic(to_vector, "Sum",)`
- » **//Repeat for from**

➤ Compute Total

- » `subarea_vmt = to_sum + fr_sum`

7. Compute zone activity

➤ Join SED to the TAZ layer

```
» sed_vw = OpenTable("SED", "FFB", {sed_file}, )  
» join_vw = JoinViews("join", taz_lyr+".ID", sed_vw+".TAZ", )
```

➤ Get total HH and Employment in the selection

➤ Use what you've learned:

➤ Try the following functions:

- `SelectByQuery()`
- `GetDataVector()` //HH and Emp
- `VectorStatistic()` //Get sums

8. Data Output

➤ Simple Way:

» `ShowArray({total_vmt, total_pop, total_emp})`

➤ Write to a File:

» Use the TransCAD help.

- `OpenFile()`
- `WriteLine()`
- `CloseFile()`



Assignment / Next Steps

- This procedure computes VMT for the AM time period.
- Create a loop that will compute and report VMT for each of the SCAG time periods
 - » Report totals for each time period
 - » Report sum over all time periods
- Hints:
 - » Use an array of periods **//pers = {"AM", "PM", ...}**
 - » Extra Credit Method:
 - Replace "AM" with "%PER%" in the OD filename
 - Try the Substitute() function to change %PER% to AM, PM, etc.



Extra Credit Example

```
1 //MyFile = pth + "AM_OD.mtx"
2 MyFile = pth + "%PER%_OD.mtx"
3
4 PerFile = Substitute(MyFile, "%PER%", "AM", ) //AM_OD.mtx
5
6 pers = {"AM", "PM"}
7 for per in pers do
8
9     //Run substitute
10
11     //Run the rest
12
13 end
14
15 //Easier than:
16
17 pers = {"AM", "PM"}
18 od_files = {pth + "AM_OD.mtx", pth + "PM_OD.mtx"}
19
```