Homework and Labs

Travel Demand Analysis

April 7, 2014

Contents

1. Introduction and Overview

This first unit has three parts. The first requires you to run a base scenario with the Wasatch Front travel demand model, and to retrieve some results from it. The second part is a TRB-related summary assignment. The final part is a statistical analysis to demonstrate your understanding of linear regression models.

1.1 Running the Wasatch Front Model

The purpose of this lab is to show you how to run the Wasatch Front Travel Model and how to look at the model output in various forms. We are using the Wasatch Front model for several reasons. Principally, your instructor is reasonably familiar with it from his previous professional experience.

The purpose of this document is to show you how to run the Wasatch Front Travel Model and how to look at the model output in various forms. We are using the Wasatch Front model for several reasons. Principally, your instructor is reasonably familiar with it from his previous professional experience.

But beyond this, there are good reasons to use the Wasatch Front model. First, the model is much cleaner than others available to this course, which will help you find scripts and become familiar with the model procedures. Second, Salt Lake City is a major metropolitan area with diverse transportation challenges, and yet is significantly smaller than comparable cities. We can get serious answers with less computing time than had we chosen Seattle, Atlanta, or another city. Finally, the Wasatch Front model uses Citilabs' CUBE travel demand modeling software, which is also the implementation used in Atlanta. Knowing how to operate the Wasatch Front model will shorten the learning curve on the Atlanta model, should you be employed locally in your career.

The first section of this document explains the model directory in some detail, and can be used as a reference throughout the course. The second section is a laboratory activity that will guide you through setting up and executing a model run.

1.1.1 The Model Folders

We will use model version 7.0 in this course, which was turned over from the consultant (Resource Systems Group, Inc.) to the MPO (Wasatch Front Regional Council) in Spring 2011. You will need to download three files from T-Square and save them to a common directory on your system:

`ModelDocumentation.pdf` This is the report prepared by the consultants with information on model calibration, data sources, and model types. We will use it frequently in this class.

`InputData_V7.zip` This compression file contains the socioeconomic data, the highway networks, and the transit line files for all of the base and horizon year default analyses (20.2 MB, 67.6

MB unzipped).

BlankModel_090112.zip This compression file contains all of the scripts necessary to run the model in a given scenario (35 MB, 128 MB unzipped).

Input Data Folder

This folder, shown expanded in Figure 1.1 contains all of the inputs to the model. You'll eventually become intimately familiar with the contents of this directory, but you should know what's here initially. There can be some confusion here, because there is also an _Inputs/ folder in the model directory. The InputData_V7/ directory contains all the potential input files which you may include in an analysis. The _Inputs/ folder is where the model looks for input files relevant to the scenario you are running at the time. This organization scheme is new in version 7, and allows you to run multiple related scenarios with the same inputs.

The MasterNet/ folder contains the master highway network, MASTER_MMDDYY.net file. This file contains information on all the links and nodes in the network, and should never be changed. If your analysis requires that you edit the highway network (for instance, to change the number of lanes on a street in a given year), you should make a copy of the MASTER_X.net file and explicitly change its name before you change the file in any way.[1] This way, you will always know what the base conditions are as defined by the MPO. There are a number of backup copies of the master highway network, should you ignore the above wisdom and need to replace the master. The files with the .VPR extension contain viewing preferences for the CUBE viewer.

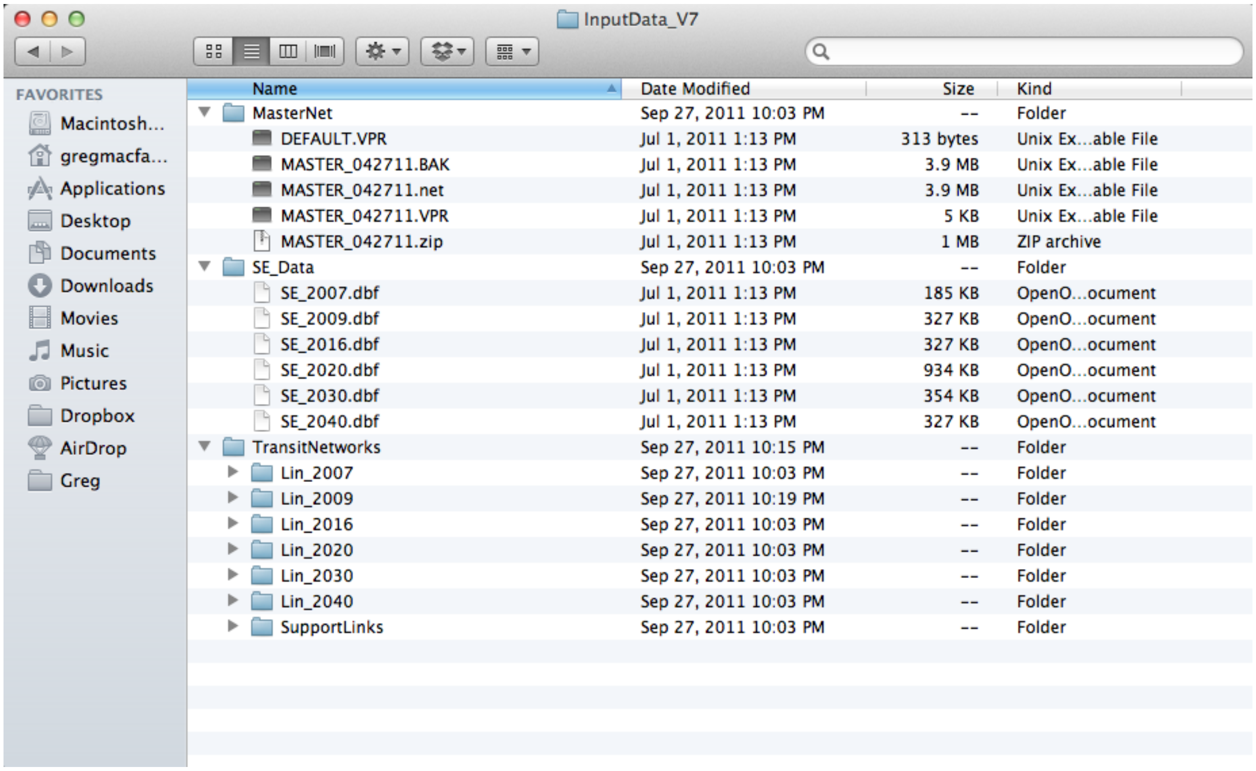
Inside SE_Data/ are database files storing the socioeconomic (SE) information for each zone in the named year. SE_2020.dbf contains the households, income, employment, and population characteristics of all TAZ's in 2020, for instance. The numbers in these files are set by the economists working for the Governor's Office of Planning and Budget working with the MPO, and the files should not be changed.[2] The set of available SE_YYYY.dbf files dictates which years you can model. While you could conceivably interpolate the data into a 2022 model year, the justifications for doing so are shaky.

The TransitNetworks/ folder contains subfolders representing the base transit service plan in the described years. The Lin_2007/ file, intuitively, contains the Utah Transit Authority transit network as it existed in 2007. Within each Lin_YYYY/ folder are a number of files with the YYarea.lin file name format. These files contain the transit routes in a given area and year. The naming is typically intuitive (if not consistent), with 07rail.lin containing the rail services in 2007. The X.link files hold special links that are used only by transit services and are not part of the highway network (rail links, for instance). There is also a spreadsheet to calculate the rail speed given the distance between stations, and a number of scripts to test if the transit network is complete.[3] There are also shapefiles of the transit network so that you can display them in GIS software.

[1] In case this is not clear, DO NOT CHANGE THE MASTER HIGHWAY NETWORK! If you must change the network, change a different file with a different name. It is almost impossible to check if a highway network file has been changed.

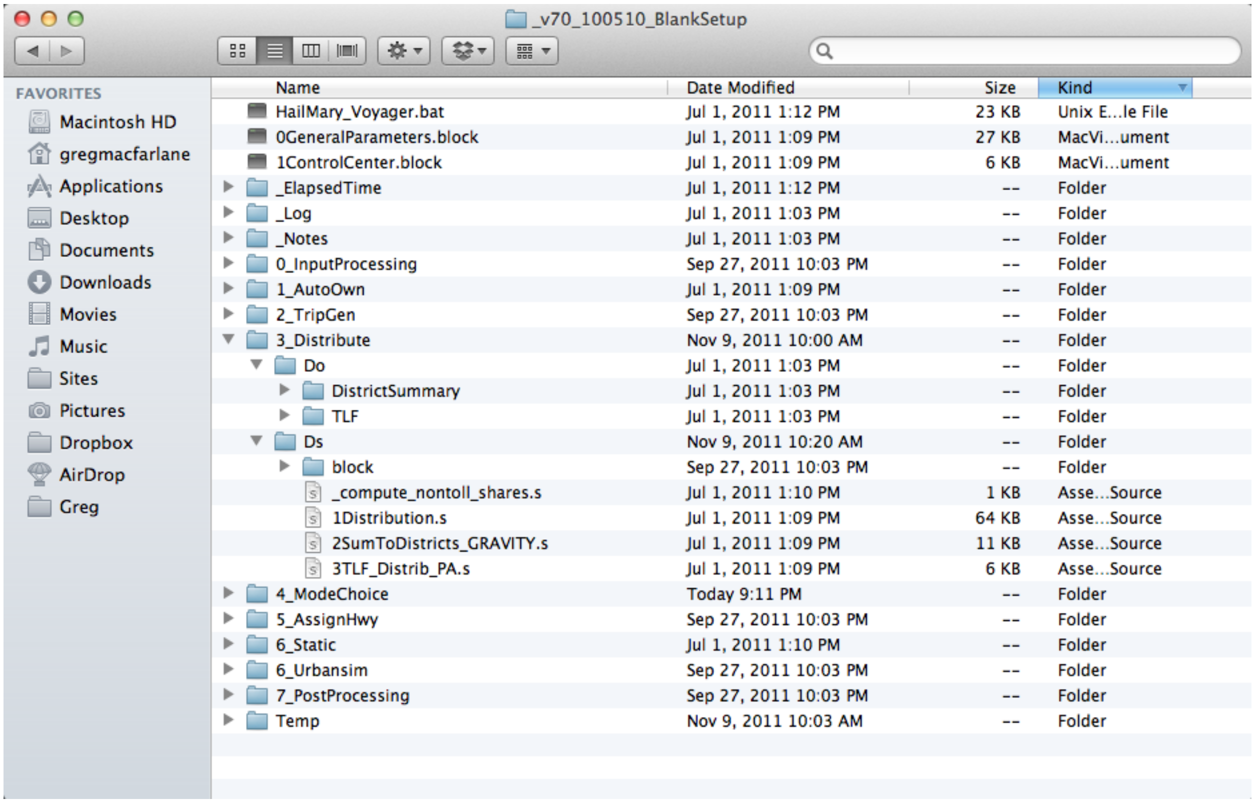
[2] The motivation not to change these numbers is even stronger than with the highway network. While you may conceivably want to see what would happen if you build a new interchange, the practice of changing the results of complicated economic models that you do not understand is questionable at best.

[3] If there is an error in your transit network, the model will crash early rather than print nonsense.



Name	Date Modified	Size	Kind
▼ MasterNet	Sep 27, 2011 10:03 PM	--	Folder
DEFAULT.VPR	Jul 1, 2011 1:13 PM	313 bytes	Unix Ex...able File
MASTER_042711.BAK	Jul 1, 2011 1:13 PM	3.9 MB	Unix Ex...able File
MASTER_042711.net	Jul 1, 2011 1:13 PM	3.9 MB	Unix Ex...able File
MASTER_042711.VPR	Jul 1, 2011 1:13 PM	5 KB	Unix Ex...able File
MASTER_042711.zip	Jul 1, 2011 1:13 PM	1 MB	ZIP archive
▼ SE_Data	Sep 27, 2011 10:03 PM	--	Folder
SE_2007.dbf	Jul 1, 2011 1:13 PM	185 KB	OpenO...ocument
SE_2009.dbf	Jul 1, 2011 1:13 PM	327 KB	OpenO...ocument
SE_2016.dbf	Jul 1, 2011 1:13 PM	327 KB	OpenO...ocument
SE_2020.dbf	Jul 1, 2011 1:13 PM	934 KB	OpenO...ocument
SE_2030.dbf	Jul 1, 2011 1:13 PM	354 KB	OpenO...ocument
SE_2040.dbf	Jul 1, 2011 1:13 PM	327 KB	OpenO...ocument
▼ TransitNetworks	Sep 27, 2011 10:15 PM	--	Folder
▶ Lin_2007	Sep 27, 2011 10:03 PM	--	Folder
▶ Lin_2009	Sep 27, 2011 10:19 PM	--	Folder
▶ Lin_2016	Sep 27, 2011 10:03 PM	--	Folder
▶ Lin_2020	Sep 27, 2011 10:03 PM	--	Folder
▶ Lin_2030	Sep 27, 2011 10:03 PM	--	Folder
▶ Lin_2040	Sep 27, 2011 10:03 PM	--	Folder
▶ SupportLinks	Sep 27, 2011 10:03 PM	--	Folder

Figure 1.1: The Inputs directory, with a view into first-level subfolders.



Name	Date Modified	Size	Kind
HailMary_Voyager.bat	Jul 1, 2011 1:12 PM	23 KB	Unix E...le File
0GeneralParameters.block	Jul 1, 2011 1:09 PM	27 KB	MacVi...ument
1ControlCenter.block	Jul 1, 2011 1:09 PM	6 KB	MacVi...ument
▶ _ElapsedTime	Jul 1, 2011 1:12 PM	--	Folder
▶ _Log	Jul 1, 2011 1:03 PM	--	Folder
▶ _Notes	Jul 1, 2011 1:03 PM	--	Folder
▶ 0_InputProcessing	Sep 27, 2011 10:03 PM	--	Folder
▶ 1_AutoOwn	Jul 1, 2011 1:09 PM	--	Folder
▶ 2_TripGen	Sep 27, 2011 10:03 PM	--	Folder
▼ 3_Distribute	Nov 9, 2011 10:00 AM	--	Folder
▼ Do	Jul 1, 2011 1:03 PM	--	Folder
▶ DistrictSummary	Jul 1, 2011 1:03 PM	--	Folder
▶ TLF	Jul 1, 2011 1:03 PM	--	Folder
▼ Ds	Nov 9, 2011 10:20 AM	--	Folder
▶ block	Sep 27, 2011 10:03 PM	--	Folder
_compute_nontoll_shares.s	Jul 1, 2011 1:10 PM	1 KB	Asse...Source
1Distribution.s	Jul 1, 2011 1:09 PM	64 KB	Asse...Source
2SumToDistricts_GRAVITY.s	Jul 1, 2011 1:09 PM	11 KB	Asse...Source
3TLF_Distrib_PA.s	Jul 1, 2011 1:09 PM	6 KB	Asse...Source
▶ 4_ModeChoice	Today 9:11 PM	--	Folder
▶ 5_AssignHwy	Sep 27, 2011 10:03 PM	--	Folder
▶ 6_Static	Jul 1, 2011 1:10 PM	--	Folder
▶ 6_Urbansim	Sep 27, 2011 10:03 PM	--	Folder
▶ 7_PostProcessing	Sep 27, 2011 10:03 PM	--	Folder
▶ Temp	Nov 9, 2011 10:03 AM	--	Folder

Figure 1.2: The Model directory, with a view into the trip distribution model subfolders.

The Model

This figure, shown expanded in Figure 1.2, contains all the scripts that comprise the model and is where you will find the model outputs. As you can see, the directory is clearly labeled with the various model components numbered in execution order.

Each model component folder contains two subfolders. As you can see in Figure 1.2, the subfolders for the `3_Dist/` component are `Do` and `Ds`; these respectively contain the outputs from and the scripts for the trip distribution models. This naming convention holds through the directory. The script files in each `Xs` folder are numbered in execution order, and are given the `.s` file extension. The name of each file describes what the script does, at least as well as twenty some-odd characters can.

While I could somewhat exhaustively cover the contents of the inputs directory in this document, the model directory will essentially be the rest of the course. But there are three files that need to be explained in detail now.

0GeneralParameters.block This is the General Parameters block file, which contains most of the global variables that are used by the model scripts. The organization of this file is not ideal, but it's not too bad either. The first 150 lines or so contain path variables, so the scripts can say `TransitDir` instead of `../../Inputs/Transit/TransitDirectory/`, for instance. There are also convenient lists of the aggregation districts and the special generators.

More interesting stuff begins around line 200, where you can set variables for transit fares, tolls, speed penalties, disutilities, and other stuff. You will play with some of these numbers throughout the semester, but it's best to leave them alone if you don't know what you're doing yet.

1ControlCenter.block The Control Center block file is something that you will become familiar with quickly, because this is where you specify which model inputs you want to use for a given run. You also give your scenario a name (`RID`). Each time you run a scenario, you will update the path to the base folder, and change some input file. There are detailed instructions on how to do this in Section 1.1.2

HailMary_Voyager.bat This is a Windows batch script that issues the proper commands to run the model. If you double-click on the file, it will open a Windows command prompt and invite you to start the model. If you right-click on the file, you can instead open it in a text editor, and see which scripts it calls.

You could run the model by running each of the thirty-or-so scripts in turn, but the batch file is much easier. Whoever wrote the original HailMary file was clearly a football fan. If one of your model scripts crashes, you will get a friendly message saying your star running back has had his leg destroyed by a 300-point lineman (or something similar). If the model runs successfully, you will get the following triumphant message:

```
!!!!!!! Touch down!  You win!  All scripts appear to have
!!!!!!! run correctly! But remember, that's no guarantee
!!!!!!! that your data is any good. Check the numbers!

!!!!!!! So long, until we meet again on the field of battle!
```

If your model crashes, you can make a copy of the `HailMary_Voyager.bat` file, delete the commands up to where you crashed, and then run the model from there after you fixed whatever was wrong.

Model Output

As the model computes, it will write files into the output folders within the model directory, and will prefix the output files with the `RID` run name variable you set in `1ControlCenter.block` This is handy because you can run two scenarios in the same folder, and you will be able to distinguish the output files by their `RID`. There are far too many output files to discuss them in a document of this scope, but I'll introduce some of the most important ones here.

Log Files Right in the root directory, the model prints out two log files, `RID_log.txt` and `RID_log2.txt`. These files give some high-level statistics on your scenario, such as the fields given in Table 1.1. On the whole these numbers will not change much between scenarios in the same year, but a quick look at the numbers can show you if there were serious problems (like no transit riders). You can also use these numbers in reports showing things like "Reducing transit fares by $0.25 reduces predicted region VMT by 3 percent." [4]

[4] This is entirely hypothetical, but would be an interesting analysis for a final project.

Table 1.1: 2009 Base Scenario - Selected Metrics

Measurement	Value
Total Trips	10,163,525
Total Vehicle Miles Traveled	46,363,848
Total Hours of Vehicle Delay	74,828
Home-based Work Trips (Motorized)	1,227,074
All Trips (Non-Motorized)	826,604
All Transit Trips	100,109
Average HBW Auto Occupancy	1.171

Boardings Most of my work at the Utah Transit Authority was with transit riders (obviously). The Boardings files, in `4_ModeChoice/Mo/Boardings/` show just about everything you want to know about the transit system. The `RID_2_Route.dbf` file, for instance, contains the peak and off-peak ridership of all the transit routes, and whether the riders arrived by car or walk. The `RID_2_OD_Station.dbf` file gives much of the same information, but broken down by station, so you can look at which segments of a transit route are under performing.

This brings up the distinction between PA (Production/Attraction) and OD (Origin/Destination) numbers. In the model, any trip end at a home is considered a "production," whether it is the origin or the destination. This is sometimes confusing, but it simplifies the mathematics substantially: instead of creating two different home-based work trip matrices for the morning and evening commutes, the one can just be the transpose of the other. At any rate you will eventually become familiar with the distinction.

Highway Net This is perhaps the most important output file of the travel demand model: `5_AssignHwy/Ao/UnloadedNetPrefix_4pd.managed.net`. This file contains the volume by period for every link in the network. I confess that I don't entirely understand the distinctions between every file in this part of the model, but you are welcome to explore. One issue- while the transit output files are visible in Excel, R, etc, the highway outputs require CUBE to visualize. Sorry.

Now that you know what the model is and how to run it, let's give you some practice.

1.1.2 Lab Exercise

This lab shows you how to run a base year scenario in the Wasatch Front Travel Model.

1. **Directory Setup:** Download the three compressed files from the site on T-Square, and save them to a folder available to you locally. This could be your local hard disk, an external hard drive, or the space on your Institute account. When finished, the full model will be approximately 6 GB, so ensure you have the required space in whatever directory you use. Unzip the `InputData_V7.zip` file into this folder.

2. **Folder Setup:** Create a new folder in this directory called `Base2009/`. Unzip the `BlankModel_090112.zip` file into this subfolder. You will now have two subfolders in `Base2009/`, `_Inputs` and `blank_model`. Make a copy of the second folder and change its name to `Run_mmddyy`, using today's date in the folder name.

3. **Inputs Setup:** Open the `Directory/Base2009/_Inputs/` subfolder. In a different window, open the `Directory/InputData_V7/` subfolder. Copy the following files from `InputData_V7` to `Run_mmddyy/_Inputs`:

   - `SE_Data/SE_2009.dbf` ⟶ `2_SEData/`
   - `SE_Data/SE_2007.dbf` ⟶ `2_SEData/`
   - `MasterNet/MASTER_042711.net` ⟶ `3_Highway/`
   - `TransitNetworks/Lin_2009/` ⟶ `4_Transit/`

4. **Run Setup:** Open the `1ControlCenter.block` file in a text editor or in CUBE. Change the following variables to the appropriate values:

   - `UserName = 'Your Name'`
   - `UserCompany = 'Georgia Tech'`
   - `RID = 'Base2009'`
   - `ParentDir = 'Directory\Base2009\Run_mmddyy\'`
   - `DemographicYear = 2009`
   - `SEFile = 'SE_2009.dbf'`
   - `BaseYear = 2009`
   - `SEFileBY = 'SE_2009.dbf'`
   - `NetworkYear = 2009`
   - `pnr_field = PNR09`
   - `LNfield = 'LN09'`
   - `FTfield = 'FT09'`
   - `UnloadedNetPrefix = 'Base2009'`
   - `MLin = 'Lin_2009\'`
   - `IXXIFile = '2009_IXXI.dbf'`
   - `XXFile = '2009_XX.mtx'`

5. **Hail Mary:**[5] Double-click on the `HailMary_Voyager.bat` executable file. A Windows command prompt window will open, welcoming you to the travel demand model, and inviting you press any key to continue. I usually press the “0” button on the numeric pad because I’m superstitious, but push whichever key you feel comfortable with. The model will begin crunching away, and will pretty regularly print output to the command window.

6. **Wait:** If the model is going to crash, it will usually do so within the first ten or fifteen minutes. Once you see that the model is into the Trip Generation scripts, you should be good. Go eat lunch, take a nap, write a blog post for the ITE website, or all three, and maybe some other things too. A standard desktop computer will take three to four hours to complete a model run. When I ran the model professionally, I usually started several model runs when I left for the evening, and could rely on the outputs being ready when I got to work the next day.

[5] Each of the above steps you could do on any computer. This one will require a computer with CUBE installed.

Table 1.2: 2009 Base Scenario - Requested Metrics

Measurement	Value	Source File
Total lane miles		Log
Total freeway VMT		Log
Transit share of college trips		Log
Work trip share of all trips		Log
Daily LRT riders		Boardings
Peak CRT riders/revenue mile		Boardings
Daily passengers who walk to route M830		Boardings
PM volume on I-15 S near 3300 South		Highway Net
AM V/C on 500 S near Rice-Eccles Stadium		Highway Net

Assignment Deliverables As your deliverable, please complete the information in Table 1.2. Fill in the empty cells, and change the generic file name to the explicit, complete file name where you obtained your number. Some of the requested items require some simple arithmetic, but no substantial calculations. Google Maps may be helpful for identifying locations in Salt Lake City.

1.2 TRB Assignment

Each year at TRB there are a number of presentations on topics relevant to this course. The purpose of this lab assignment is to acquaint you with this researchers and those who perform it.

If you are attending TRB You should attend a presentation by one of the following researchers (or at least a paper on which they are a coauthor):

- Chandra Bhat, University of Texas
- Ram Pendyala, Arizona State University
- Paul Waddell, University of California - Berkeley
- Joan Walker, University of California - Berkeley
- Eric Miller, University of Toronto
- Stephane Hess, University of Leeds
- Abolfazl Mohammadian, University of Illinois - Chicago
- Harry Timmermans, Technische Universiteit Eindhoven
- Kai Axhausen, Eidgenössische Technische Hochschule (ETH) Zürich

You may also attend session 877 or 862, *Doctoral Research in Transportation Modeling.*[6]

If you are not attending TRB Find a paper published in *Transportation Research Record* authored by one of the researchers listed above.

Deliverable Write a short but sufficient summary of the research. Discuss the motivation for the research, the analysis method, and the conclusions. Also discuss what you learned about travel demand analysis from this activity.

[6] The session is at the Hilton on Thursday morning; I will be presenting my dissertation defense.

1.3 Regression Models

There are two parts to this lab. In the first part you will estimate a linear regression model from actual survey data to give you practice using the regression and diagnotistic functions included in R. In the second part, you will derive your model parameter estimates algebraically and numerically so as to demonstrate understanding of regression mathematics.

1.3.1 Estimating a Regression Model

For this assignment we will use data from the 2009 National Household Travel Survey. I've already done some preparatory work to clean the data for you; you can find my cleaned data on T-square. There may be additional steps you should take to create the best model.

```r
load("VehicleData.Rdata")
```

Build a linear model to predict the annual miles driven on the vehicles in the dataset. Use `BESTMILE` as the response variable, as oppposed to `ANNMILES`. The `BESTMILE` is an estimated variable, including imputations for respondents who declined to disclose their mileage, or for those whose stated values did not make sense. You may want to consider executing a logarithmic transformation on this variable and some of your independent variables.

Select a model that gives the best fit. You must include at least one categorical variable in your final model. Categorical variables are defined in R by using the `factor()` function. For example, you can define categorical levels for the presence of rail transit in the city:

```r
Vehicles$MSARAIL <- factor(Vehicles$MSACAT, levels = c(4, 1, 2, 3), labels = c(" Not in MSA",
    " >1 M w/ rail", " >1 M w/o rail", " <1 M"))
```

You may want to consider using variables such as the following:[7]

- Cost of gasoline
- Vehicle fuel efficiency
- Size of metropolitan area
- Household or employment density of block group.
- Household income, size, and lifecycle stage.
- Whether the metropolitan area has rail transit service.
- Whether the vehicle is driven primarily by the householder.

Some R commands you might find useful (type `?command` for a help file):

- `stem()` - prepare stemplot
- `summary()` - give summary statistics

[7] Many of these measurements have multiple variables. Please consult the codebook at the NHTS website.

- `plot()` - create plot; for useful tricks, type `?plot.lm`
- `lm()` - estimate a linear model
- `load()` - import an Rdata object file into a data frame
- `require()` - load an accessory package
- `install.packages()` - download and install an accessory package
- `cor()` - correlation among variables
- `cov()` - variance / covariance among variables

As an example, here is a simple model:

```
Vehicles$lBESTMILE <- ifelse(Vehicles$BESTMILE > 1, log(Vehicles$BESTMILE),
    0)
model1.lm <- lm(lBESTMILE ~ MSARAIL + EPATMPG, data = Vehicles)
summary(model1.lm)

##
## Call:
## lm(formula = lBESTMILE ~ MSARAIL + EPATMPG, data = Vehicles)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.274 -0.384  0.253  0.725  4.622
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              9.358497   0.008493 1101.92  < 2e-16 ***
## MSARAIL >1 M w/ rail     0.045326   0.007757    5.84  5.1e-09 ***
## MSARAIL >1 M w/o rail    0.038952   0.006592    5.91  3.4e-09 ***
## MSARAIL <1 M            -0.024589   0.006614   -3.72    2e-04 ***
## EPATMPG                 -0.020211   0.000268  -75.34  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.3 on 301425 degrees of freedom
##   (7733 observations deleted due to missingness)
## Multiple R-squared:  0.0188,Adjusted R-squared:  0.0188
## F-statistic: 1.45e+03 on 4 and 301425 DF,  p-value: <2e-16
```

1.3.2 Regression Mathematics

There are many different ways to estimate regression parameters β for the linear regression model $y = X\beta + \epsilon$. This is the "true" model, whereas the estimated model is $y = X\hat{\beta} + u$, as β is unknown.

The two most common estimation methods are *least squares* and *maximum likelihood*; indeed the estimates obtained by the two methods are mathematically identical for linear regresion.

Least Squares

This method is focused on finding the estimates of β, $\hat{\beta}$ that minimize the sum of squared residuals, $SSR(\hat{\beta}) = \sum_{i=1}^{n}(y_i - x_i\beta)^2$. The minimum of a function occurs where the derivative is zero (as a first order condition):

$$\begin{aligned} \frac{\partial SSR(\hat{\beta})}{\hat{\beta}} &= \sum_{i=1}^{n} -2(y_i - x_i\hat{\beta})x_i \\ 0 &= \sum_{i=1}^{n} x_i'(y_i - x_i\hat{\beta}) \end{aligned}$$

Replacing the sum with matrix operations gives $0 = X'(y - X\hat{\beta})$. This allows us to easily solve for the least-squares estimates of $\hat{\beta}$:

$$\begin{aligned} X'y =& X'X\hat{\beta} \\ \hat{\beta} =& (X'X)^{-1}X'y \end{aligned}$$

If the true data generating process is $X\beta + \epsilon$, the expected value of the estimator is

$$\begin{aligned} \hat{\beta} &= (X'X)^{-1}X'(X\beta + \epsilon) \\ \hat{\beta} &= (X'X)^{-1}X'X\beta + (X'X)^{-1}X'\epsilon \\ \hat{\beta} &= I\beta + (X'X)^{-1}X'\epsilon \qquad (1.1) \\ \mathrm{E}\,(\hat{\beta}|X) &= \mathrm{E}\,(\beta|X) + (X'X)^{-1}X'\mathrm{E}\,(\epsilon|X) \end{aligned}$$

If $\mathrm{E}\,(\epsilon|X) = 0$, then $\mathrm{E}\,(\hat{\beta}|X) = \beta + 0$, and $\hat{\beta}$ is an unbiased estimator of β. The variance of the estimator can be had by taking the variance of Equation 1.1.

$$\begin{aligned} \mathrm{Var}\,(\hat{\beta}|X) &= \mathrm{Var}\,(I\beta + (X'X)^{-1}X'\epsilon) \\ &= 0 + \mathrm{Var}\,((X'X)^{-1}X'\epsilon|X) \\ &= (X'X)^{-1}X'\Sigma_\epsilon X(X'X)^{-1} \qquad (1.2) \end{aligned}$$

Where Σ_ϵ is the variance-covariance matrix of ϵ. If we assume that ϵ is distributed identically and independently with zero mean and variance σ^2, this implies

$$\begin{aligned} \Sigma_\epsilon &= \mathrm{E}\,[(\epsilon - \mathrm{E}\,(\epsilon))(\epsilon - \mathrm{E}\,(\epsilon))^T] \\ &= \mathrm{E}\,[\epsilon\epsilon^T] \\ &= \begin{pmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{pmatrix} \\ &= \sigma^2 I \qquad (1.3) \end{aligned}$$

Because $\mathrm{E}\left(\epsilon_i \epsilon_i\right) = \sigma^2$ and $\mathrm{E}\left(\epsilon_i \epsilon_j\right) = 0$. Placing this result into Equation 1.2 gives the variance of the estimator as

$$\begin{aligned}
\mathrm{Var}\left(\hat{\beta}|X\right) &= (X'X)^{-1} X' \sigma^2 I X (X'X)^{-1} \\
&= \sigma^2 (X'X)^{-1} X' X (X'X)^{-1} \\
&= \sigma^2 (X'X)^{-1} \qquad (1.4)
\end{aligned}$$

This result is used to test hypothesis statistics against assumed distributions.[8]

We can use R to calculate these estimates "by hand." These are precisely the calculations that the `lm()` command performs, but it's good to know what's happening inside your software.

```
X <- model.matrix(~lBESTMILE + MSARAIL + EPATMPG, data = Vehicles)
y <- X[, 2]
X <- X[, -2]
# Parameter Estimates
XTX <- solve(t(X) %*% X)
Beta <- XTX %*% t(X) %*% y
# Mean squared error
e <- y - X %*% Beta
s <- mean(e^2)
# Standard errors
SE <- sqrt(diag(s * XTX))
# output table
cbind(Beta, SE, Beta/SE)

##                                       SE
## (Intercept)            9.35850 0.0084928 1101.927
## MSARAIL >1 M w/ rail   0.04533 0.0077566    5.844
## MSARAIL >1 M w/o rail  0.03895 0.0065915    5.909
## MSARAIL <1 M          -0.02459 0.0066142   -3.718
## EPATMPG               -0.02021 0.0002682  -75.343
```

Maximum Likelihood

Not every model has an algebraic solution. Even if there is such a solution, we can find the values of $\hat{\beta}$ that maximizes the *likelihood* of the model. If we assume that some dependent variable y is distributed normally

$$y_i = \mathcal{N}(X_i \beta, \sigma^2) \tag{1.5}$$

with the mean equal to the regression line $X\beta$ and variance of σ^2 the likelihood function is

$$\mathcal{L} = \prod_{i=1}^{n} y_i = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - X_i\beta)^2}{2\sigma^2}\right) \tag{1.6}$$

[8] Making the additional assumption that the mean squared error of the OLS residuals is an estimate of σ^2.

where we use the normal density function. It's easier to use the log of the likelihood function is

$$\log(\mathcal{L}) = \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \Big(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(y_i - X_i\beta)^2 \Big)$$

$$\log(\mathcal{L}) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(Y - X\beta)'(Y - X\beta)$$

We can code this as a function in R with $\theta = (\beta, \sigma^2)$.

```
linear.lik <- function(theta, y, X) {
    n <- nrow(X)
    k <- ncol(X)
    beta <- theta[1:k]
    sigma2 <- theta[k + 1]
    e <- y - X %*% beta
    logl <- -0.5 * n * log(2 * pi) - 0.5 * n * log(sigma2) - ((t(e) %*% e)/(2 *
        sigma2))
    return(-logl)
}
```

We can then find parameters that maximize this function with R's built-in optimization commands. Note that we need to provide an initial value for each coefficient, the intercept *and* the mean squared error.

```
linear.MLE <- nlm(f = linear.lik, p = c(1, 1, 1, 1, 1, 1), hessian = TRUE, y = y,
    X = X)
par <- linear.MLE$estimate
se <- sqrt(diag(solve(linear.MLE$hessian)))
t <- par/se
pval <- 2 * (1 - pt(abs(t), nrow(X) - ncol(X)))
cbind(par, se, t, pval)

##            par        se        t      pval
## [1,]  9.35856 0.0084929 1101.934 0.000e+00
## [2,]  0.04535 0.0077566    5.847 5.017e-09
## [3,]  0.03897 0.0065915    5.913 3.367e-09
## [4,] -0.02457 0.0066142   -3.714 2.038e-04
## [5,] -0.02021 0.0002682  -75.355 0.000e+00
## [6,]  1.68625 0.0043444  388.141 0.000e+00
```

1.3.3 Assignment Deliverables

Write a technical memorandum describing your analysis. Describe how you selected your preferred model, and include a diagnostic plot demonstrating that your model is not subject to severe heteroskedasticity or influential observations. Discuss in detail the implications of your model for vehicle miles traveled. What are the most influential factors? Are there factors that are statistically significant but practically unimportant?

Include an R script file that contains code necessary to estimate your preferred model by least-squares and maximum likelihood.

2. Trip Generation

There are two weeks to this unit. In the first week, you will build linear regression and cross-classification models to predict household trips. In the second week, you will use the models you develop in the Wasatch Front Travel Demand Model.

2.1 Model Generation

2.1.1 Linear Regression Models

In the first part of this assigment, you will build a linear model to predict home-based work trips, and a cross-classification model to predict three different types of trips. You will then compare these two models to see which variables significantly predict different types of trips. We will use the `TripGen.Rdata` file which was created from the NHTS.

```
library(gdata)
load("./TripGen.Rdata")
```

Build a linear regression model to predict the number of household work trips. Include a model with just the number of workers (`WRKCOUNT`), and one with just the number of automobiles (`HHVEHCNT`), and one with severable variables that you find interesting and significant. Ensure to treat categorical variables appropriately (`factor()`), and eliminate missing information from your model variables (`unknownToNA`). Provide a residual plot of your final model.

Build a linear regression model that incorporates the household income alone, `HHFAMINC`. You should code this variable so that it represents actual dollar values, in thousands of dollars. A mechanism for doing this might be

```
missing.values <- c(-9, -8, -7)
TripGen$HHFAMINC <- unknownToNA(TripGen$HHFAMINC, missing.values)
TripGen$HHINCOME <- ifelse(TripGen$HHFAMINC == 18, 100, ifelse(TripGen$HHFAMINC ==
    17, 90, (TripGen$HHFAMINC * 5 - 2.5)))
```

After this model, include income in a model controlling for other variables such as the number of workers in a household. What happens to the sign of the income paramater estimate? How do you explain this?

2.1.2 Cross-classification Models

In this portion of the assignment, you will build cross-classification models to predict household trips. A cross classification model simply assigns an expected value to a household, conditioned on one or more variables. Because it may not be appropriate to estimate mean values for classes that are unlikely to be represented, you should cap the number of vehicles or adults that you estimate categories for.

```
TripGen$adultclass <- ifelse(TripGen$NUMADLT > 6, 6, TripGen$NUMADLT)
TripGen$vehclass <- ifelse(TripGen$HHVEHCNT > 3, 3, TripGen$HHVEHCNT)
```

Calculate the average number of HBO trips per household, conditioned on the household size and number of household vehicles. Present your results in a table. A good way to do this is with the `aggregate()` function:

```
hboaverage <- aggregate(TripGen$HBOTrips, by = list(TripGen$adultclass, TripGen$vehclass),
    FUN = mean)
```

Provide similar tables for HBW and NHB trips. Are these numbers reasonable? Submit a memorandum outlining your work. Attach a script containing the R commands you used in your analysis.

2.2 Model Implementation

Replace the trip generation rates in the Wasatch Front Travel Demand Model with the rates you calculated in Section 2.1.2. The script necessary is `1TripGen.s`; you should copy this file and save the original as `1TripGen(ORIGINAL).s`. The rates are discussed on page 36 of the model documentation. You should notice that HBO and NHB trips are divided into further subcategories,

$$HBO = \text{hbotp} + \text{hbscp} + \text{hbshp} + \text{hbpbp}$$

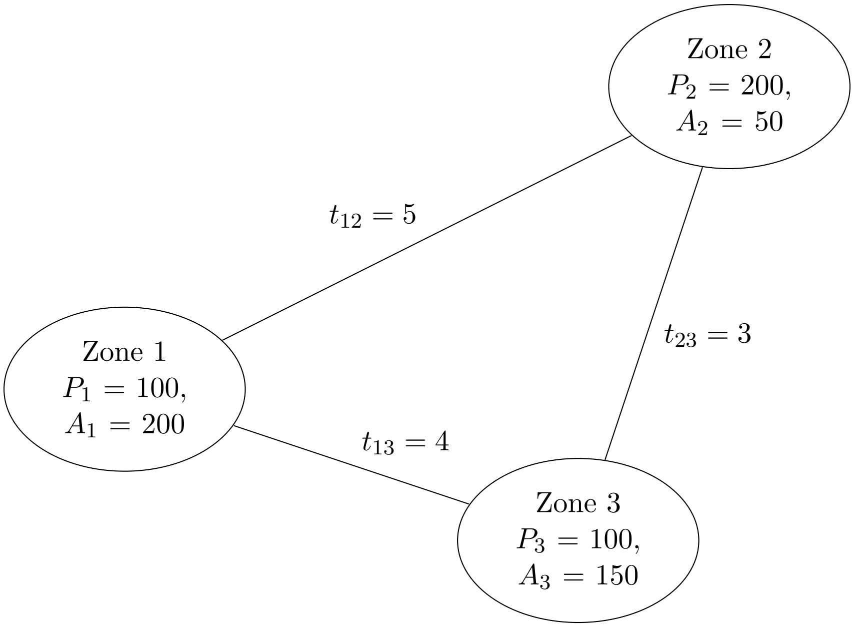
Because we did not calculate each individual HBO trip sub-type, you should develop a weighting system to distribute your estimates across the sub-types. Any method is appropriate, but clearly document what you do.

Run a 2009 scenario with the altered trip generation rates, and compare the principal log files with the base model you computed in Lab **??**. Discuss the linearity of changes to the trip generation rates; that is, does a 1% decrease in the average generation rate correspond to a 1% decrease in region trips? Why or why not?

Comment on the adequacy of the National Household Travel Survey as a replacement for locally collected surveys. Submit your analysis in a technical memorandum.

3. Trip Distribution

3.1 Gravity Models



The figure above[1] presents a simple three zone system, the link travel times for this system (for internal trips, $t_{ii} = 2$ globally), and the zonal productions and attractions. Assume a gravity model of the form

$$T_{ij} = \frac{P_i A_j^*(t_{ij})^{-b}}{\sum_{j'} A_{j'}^*(t_{ij})^{-b}} \tag{3.1}$$

where A_j^* is a "modified attraction term" defined by the algorithm shown in Figure 3.1. This algorithm ensures that the predicted trips to a given zone equal the true zonal attractions A_j.

Use Matlab or a similar programming software to write a program that computes the O-D flows for this system using the algorithm (with $\varepsilon = 1$). Find the value of b (to a single decimal place) which provides the "best fit" with the observed O-D matrix in Table 3.1. In your memo reporting the results of this analysis, discuss how you determined which b value gave the "best fit."

[1] This problem is adapted from "Urban Transportation Planning" by Michael D. Meyer and Eric J. Miller.

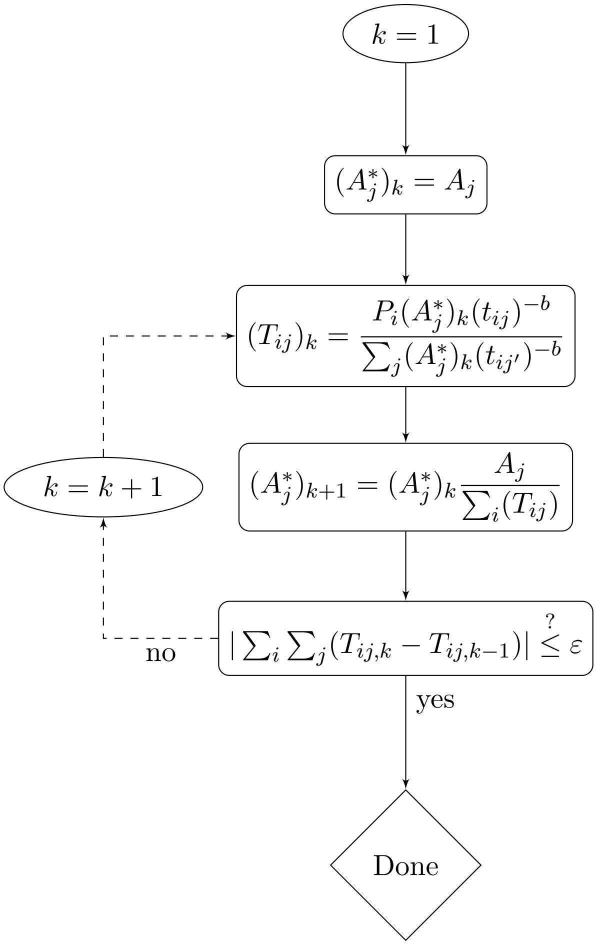

$k = 1$
$(A_j^*)_k = A_j$
$(T_{ij})_k = \frac{P_i (A_j^*)_k (t_{ij})^{-b}}{\sum_j (A_j^*)_k (t_{ij'})^{-b}}$
$k = k + 1$
$(A_j^*)_{k+1} = (A_j^*)_k \frac{A_j}{\sum_i (T_{ij})}$
$|\sum_i \sum_j (T_{ij,k} - T_{ij,k-1})| \overset{?}{\leq} \varepsilon$
no
yes
Done

Figure 3.1: Algorithm to match A^* and A

Table 3.1: Observed Origin-Destination flows for Problem 3.1.

i	1	2	3
1	80	5	15
2	80	40	80
3	40	5	55

3.1.1 Network Improvements

The region represented in Problem 6.1 has begun improvements to the link between zones 1 and 2 that will reduce the travel time from 5 to 3. Using your program and the value of b estimated above, determine the effect of this improvement on the predicted trip distribution matrix.

3.2 Lab

The distribution models in the Wasatch Front model are not really good to experiment with, because the really interesting steps are done with a destination choice model, something we have not really covered in detail. So instead of fiddling with the gravity model, we will use this week to show you how to get more interesting data from Cube.

3.2.1 Viper Files

Cube makes it possible to visually inspect the attributes of a highway network, by adjusting the color and thickness of lines based on rules you give it. These rules can be saved as a "viper" file, with a `.vpr` extension. When you open a highway network in Cube, the software will try to match a viper file from the current directory. So if you open `foo.net` Cube will try to find `foo.vpr` in that directory. If it can't find it, it will try and find `default.vpr`. If that fails, it will just do a default color scheme, with blue links and grey TAZ connectors.[2]

From the Cube open file dialog, you can open either a highway network or a viper file. If you open the viper file first, you can then view any highway network using the layer manager. So you can develop a viper file with very complex specifications and then use it for all of the model runs in a project.

To set the viper specifications, open a highway network file. Any network will do, although you will typically use either the input highway network or a loaded network post-assignment. If you have no other `.vpr` files in your directory, you will get the Cube default, as shown in Figure 3.2.

The "Home" tab on the Cube menu bar contains several commands for displaying the highway network and editing viper files. If you click on the linetypes and colors on the "Post Link" box, you will get a dialog like the one shown in Figure 3.3. You can change these definitions and save them as group types. Clicking on the "Color" drop down tab in the same box will allow you to quickly switch between color definitions. So you could have a color definition for 2010 and 2030 functional types, or AM and PM volumes. Similar tools are available for displaying node attributes. An example is shown in Figure 3.4.

The Wasatch Front travel demand model has a number of out-of-the box viper files; unfortunately many of them have not been updated from model version 6, and so require fields that are no longer in the model. You can save the viper definitions you create by selecting `File -> Save As -> Project As`.

3.2.2 Desire Lines

Many of the most important output files from the travel demand model are trip matrices, showing where people travel to and from. Unfortunately, these are extremely large files that are almost impossible for a human to read. So Cube has tools to display these matrices on top of a highway network.

To do this, follow the steps outlined below.

1. Open a trip distribution matrix. Perhaps the most comprehensive is the `AllTripsX100_pkok.mtx` file which has all trip types broken out by mode.

[2] Note that the Cube default is not necessarily the same as what you have in a default viper file.

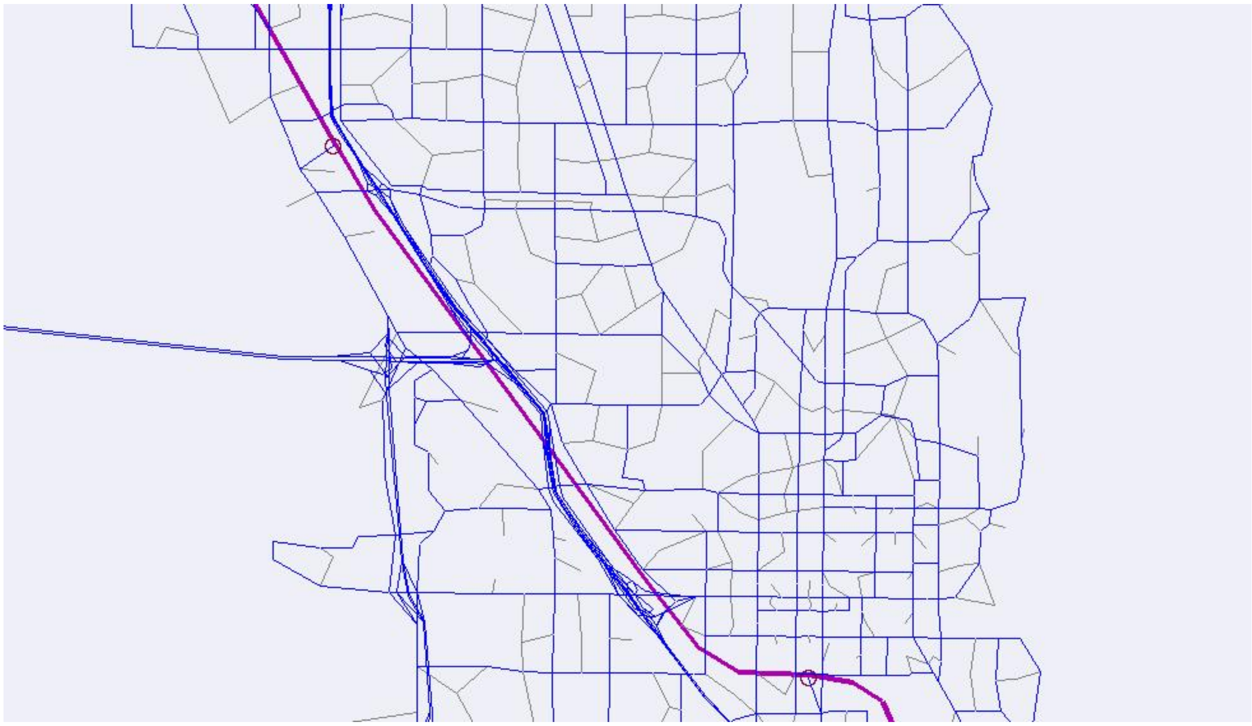


Figure 3.2: Provo viewed with no viper specifications.

2. From your highway network click on “Link to Matrix...” on the “Analysis” tab. You will see a list of all your open matrix files. Select the matrix you wish to analyze, click “Add,” and notice that your matrix has been labeled “M1=...” Exit the dialog.

3. The “Desire Lines ...” command on the “Analysis” tab has now been activated. Click this button to bring up the desire lines toolbar.

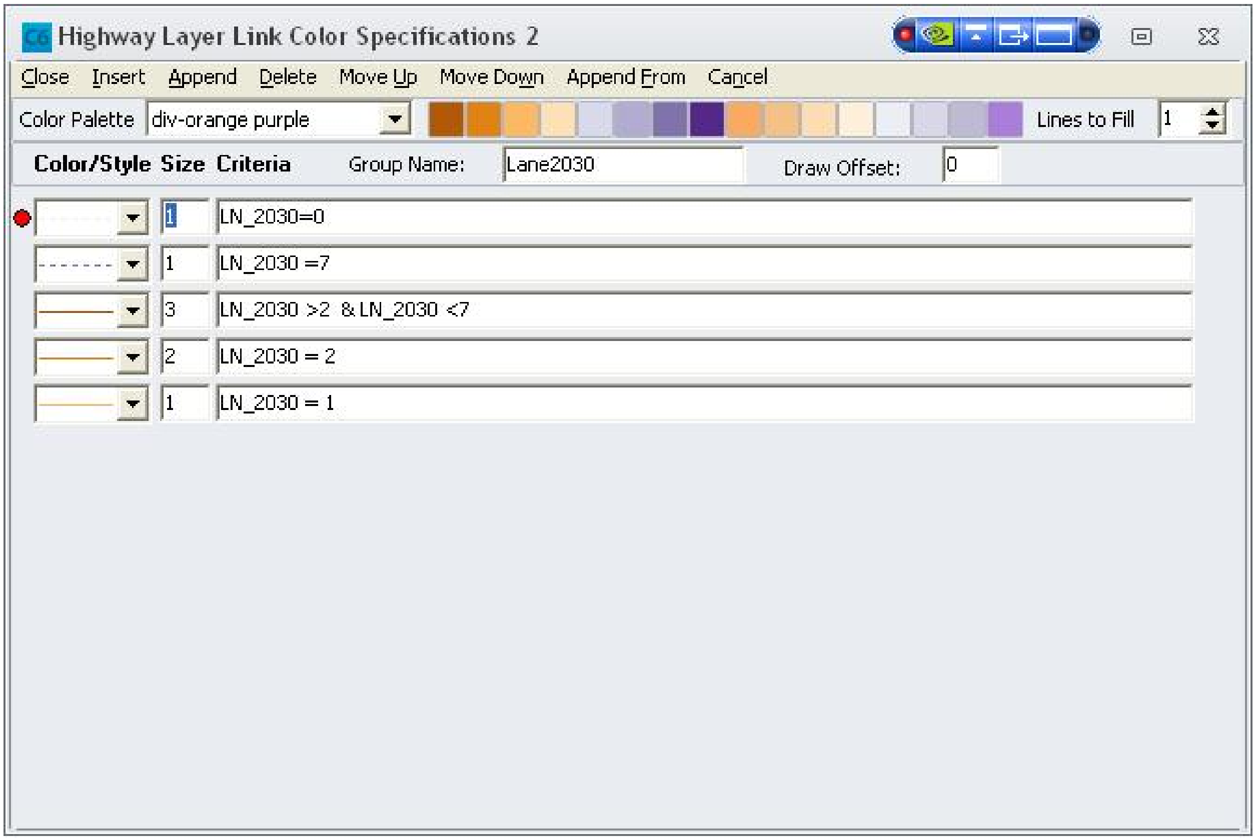
4. In the “Matrix Table(s)” field, enter `M1.Tn` where `n` is the number of the table you want to visualize. If we wanted to see all auto trips on the `AllTripsX100_pkok.mtx` matrix, for instance, we would use `M1.T4`.

5. Set the scale to some number that makes sense. Cube will create a line from i to j of one pixel width for each “scale.” That is, if we set the scale at 5, then every 5 trips from i to j will add another pixel width. This number will usually require some adjustment. Remember that in the `AllTripsX100` matrix, a scale of 100 means one trip.

6. Set the “Org Exp” and “Dest Exp” zones to some value representing the type of map you want. I typically look at one destination zone and all origin zones in some geography.

7. Push “Display.”

Figure 3.5 shows desire lines for all auto trips to Brigham Young University with a scale of 30. Viewing transit trips is simply an issue of changing the table definition.



Color Palette: div-orange purple

Lines to Fill: 1

Group Name: Lane2030

Draw Offset: 0

Color/Style	Size	Criteria
	1	LN_2030=0
- - - - - - -	1	LN_2030 =7
———	3	LN_2030 >2 & LN_2030 <7
———	2	LN_2030 = 2
———	1	LN_2030 = 1

Figure 3.3: The link specifications dialog box.

3.2.3 GIS Integration

Cube is very much integrated with ArcGIS[3], and it is possible to use ArcGIS's capabilities to produce truly beautiful data visualizations.

Cube has capabilities to create, edit, and view shapefiles within Cube itself, but I usually find it less frustrating to just use ArcGIS some other GIS engine.[4] At any rate, you will need to create a *personal geodatabase* to store the geospatial information related to your projects. This is just a repository to conveniently store binary or ASCII files (like the `.NET` and `.lin`) files in a way that can be projected in GIS software. To do this, follow the steps below:

1. Open the Data Manager in Cube by clicking on the cylinder-shaped button at the top of the menu bar.

2. Right-click in the Data Manager window, and select ”New Geodatabase”

3. Select where you would like the file stores, and how you want it named. You may want a geodatabase for each project, or for each scenario, depending on how much you are going to use it. Do not use spaces in the name.

[3] In fact, you cannot buy Cube without also buying a license for ArcGIS, a fact that keeps Cube Windows-exclusive for now.

[4] I'm a big proponent of open source software, and Mac OS, so I use QGIS.

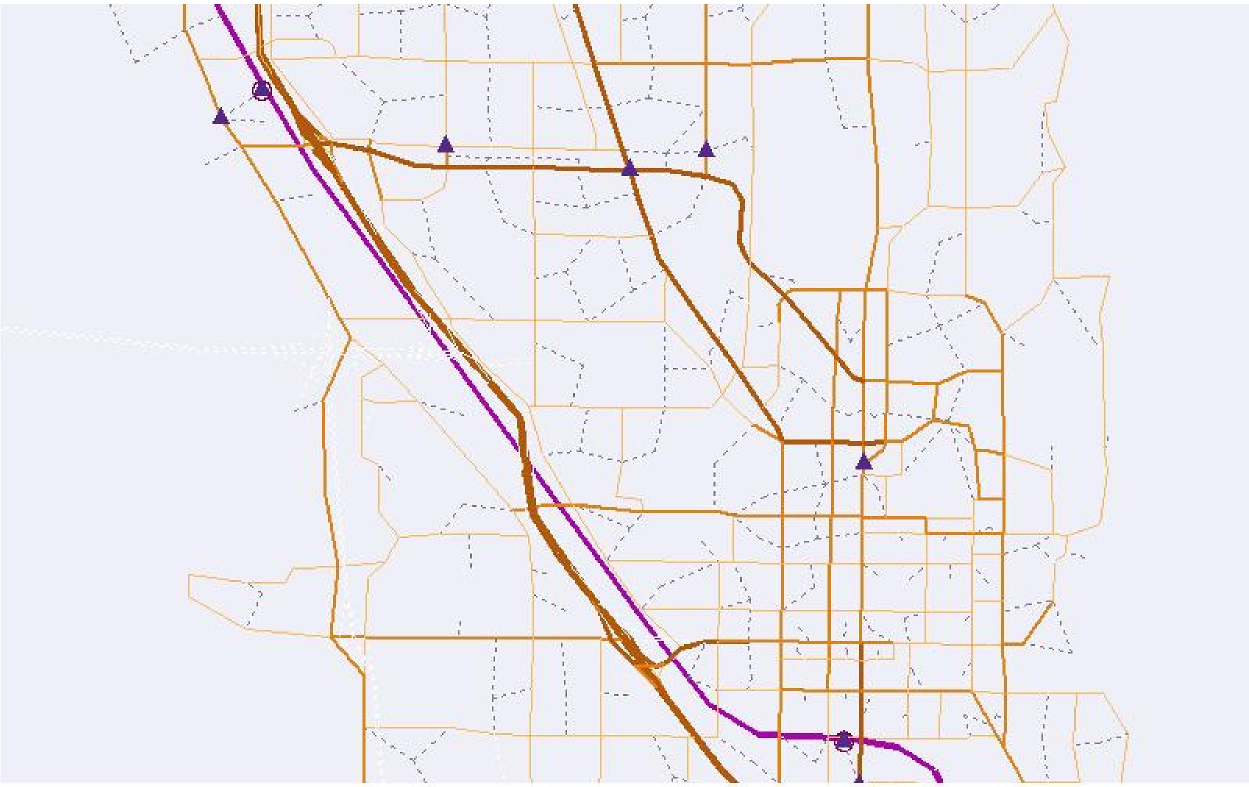


Figure 3.4: Provo viewed with definitions showing 2030 lanes and park and ride lots.

4. The Create Data dialog will come up instantly. Set the spatial reference to projected coordinate system 1983 NAD UTM Zone 12N.
5. You can add network and line files to this geodatabase using the Import/Export Data dialog. These networks can now be projected in GIS software, and you can add other non-Cube shapefiles to your maps.

When you have transit line files mapped in ArcGIS, you can join information on the route ridership using the output dbf files in the mode choice directory, though some data manipulation may be needed.

It is possible to add desire line information to a geodatabase map, but that is something I will have to update later.

3.2.4 Homework

1. Viper File Create a viper file that shows peak period volume/capacity levels of service as colors. Use the level of service definitions given in Table 3.2. In your memo, include a screenshot of your Base 2009 scenario (zoomed in on some interesting geography) with your definitions. Attach your `los.vpr` file to the assignment in T-square.

2. Desire Lines Build desire line maps (screenshots are fine) showing non-motorized trips to the University of Utah and Brigham Young University. Build two more maps showing walk-access

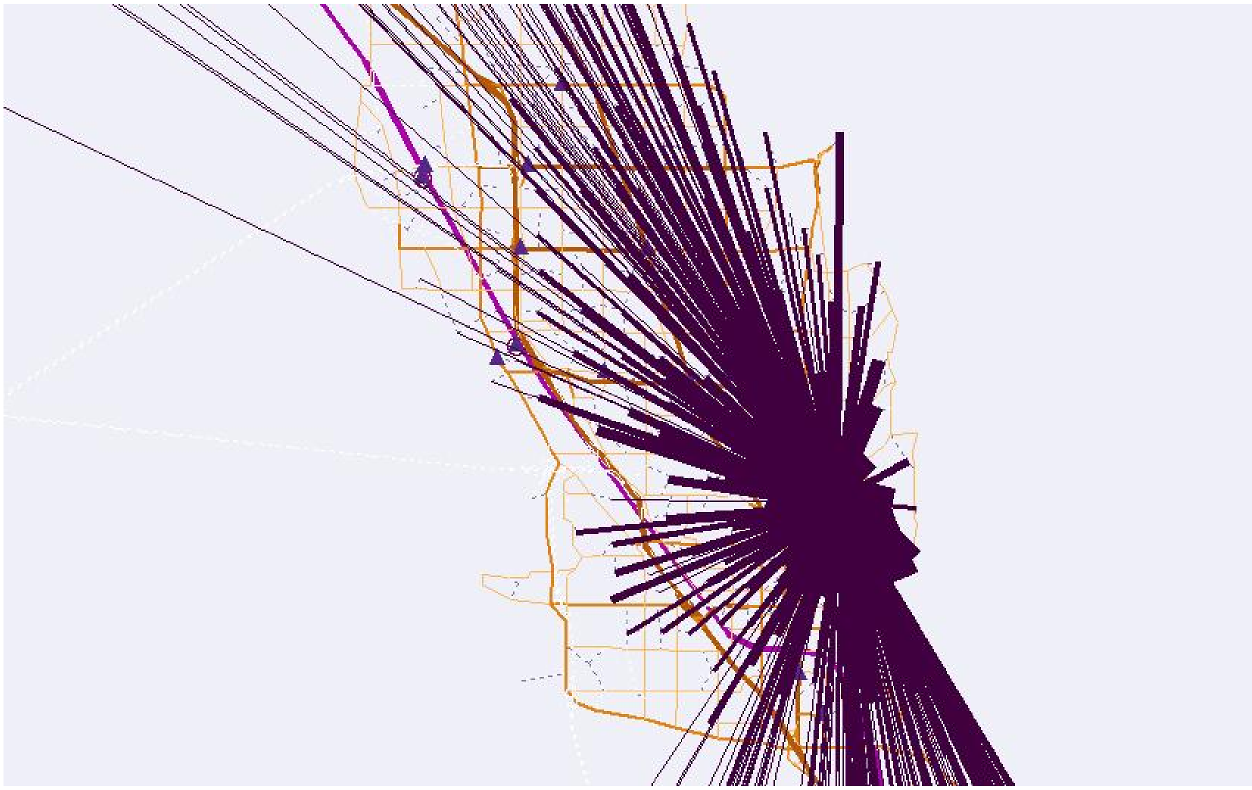


Figure 3.5: Desire lines for automobile trips to Brigham Young University

transit trips to both schools. Comment on the distribution of trips to the two schools.

3. Data Visualization Create a personal geodatabase in your directory. Add the highway network and the 2030 rail files. Create a map using GIS software that shows the investment per rider on each segment of the rail network in the 2030 base scenario (use an estimated $7 per revenue mile). Comment on what you see.

Table 3.2: Level-of-Service Definitions

LOS	V/C	Color
A	≤ 0.34	Blue
B	0.35 - 0.54	Light Blue
C	0.55 - 0.77	Green
D	0.78 - 0.93	Yellow
E	0.94 - 0.99	Orange
F	≥ 1.00	Red

4. Mode Choice

There are two parts to this unit's lab. The first part involves estimating multinomial logit models from actual data, and the second demonstrates the application of mode choice models in the Wasatch Front Travel Demand model.

4.1 Mode Choice Model Estimation

For this week's assignment, you will use data from the 2000 Bay Area Travel Survey to estimate multinomial logit models that predict mode choice for work trips. The data is available on T-Square attached to this assignment. The data are listed as `WorkTrips.Rdata`. We will also need to load the texttttmlogit library package, which contains the tools necessary to estimate multinomial logit models.[1]

```
load("WorkTrips.Rdata")
library(mlogit)
```

Because multinomial logit models are so different from other models, we need to use a special function to coerce the data into a form the model software can use. To do this, there needs to be a unique `HH.Person` identification variable. Also, we should change the alternatives from their number to the name of the mode.

```
# Make new id field
worktrips$IDFIELD <- paste(worktrips$HHID, worktrips$PERID, sep = "-")

# Name alternatives
alternatives <- c(" Drive Alone", " Share 2", " Share 3+", " Transit", " Bike",
    " Walk")
worktrips$ALTNUM <- factor(worktrips$ALTNUM, labels = alternatives)

# make mlogit data frame
logitdata <- mlogit.data(worktrips, choice = "CHOSEN", alt.var = "ALTNUM", chid.var = "IDFIELD"
    shape = "long")

head(logitdata[, 1:8], n = 12)
```

[1] and other discrete choice structures.

```
##                    HHID PERID CASE       ALTNUM NUMALTS CHOSEN  IVTT OVTT
## 2-1. Drive Alone    2     1    1  Drive Alone       5   TRUE 13.38  2.0
## 2-1. Share 2        2     1    1      Share 2       5  FALSE 18.38  2.0
## 2-1. Share 3+       2     1    1     Share 3+       5  FALSE 20.38  2.0
## 2-1. Transit        2     1    1      Transit       5  FALSE 25.90 15.2
## 2-1. Bike           2     1    1         Bike       5  FALSE 40.50  2.0
## 3-1. Drive Alone    3     1    2  Drive Alone       5  FALSE 29.92 10.0
## 3-1. Share 2        3     1    2      Share 2       5  FALSE 34.92 10.0
## 3-1. Share 3+       3     1    2     Share 3+       5  FALSE 21.92 10.0
## 3-1. Transit        3     1    2      Transit       5   TRUE 22.96 14.2
## 3-1. Bike           3     1    2         Bike       5  FALSE 58.95 10.0
## 5-1. Drive Alone    5     1    3  Drive Alone       4   TRUE  8.60  6.0
## 5-1. Share 2        5     1    3      Share 2       4  FALSE 13.60  6.0
```

Now that your data is cleaned and formatted, you can estimate multinomial logit models. To do this, use the `mlogit()` function, in a manner sort of like you would use the `lm()` command. One thing to look out for: the difference between generic and alternative-specific variables.[2]

Generic Variables These are coefficients with a single estimated parameter. That is, the $\hat{\beta}$ coefficient for these variables has the same value in the utility equation for every alternative. These estimates come from variables that vary naturally across the alternatives, like the cost of travel.

Alternative-Specific Variables This type of coefficient has a unique estimate for each alternative. That is, $\hat{\beta}_{DA}$ is different from $\hat{\beta}_{Walk}$. This type of estimate comes from variables that are constant across alternatives, like the distance of the trip.

To specify the model, we use the following construction.

```
fit.mnl <- mlogit(CHOICE ~ Generic | Alt.Specific, data = logitdata)
```

To examine the model output, the standard `summary()` command will produce a coefficients table and a few test statistics. A more robust course in discrete choice analysis would explain these in greater detail, but for our limited purposes you just need to understand that McFadden's R^2 statistic (or ρ^2) is roughly equivalent to that in a linear regression model, and that the t-values associated with the coefficients have the same interpretations. You want to have a high R^2 value and avoid insigificant explanatory variables as much as possible.

4.1.1 Models to Estimate

For this lab, you will need to estimate and interpret four different models on the BATS dataset.

1. **Value of Time** Estimate a model with just the total travel time (`TVTT`) and the cost of the trip (`COST`). These two parameter estimates will allow you to calculate the value of time for the sample population as

$$VOT = \frac{\hat{\beta}_{TVTT}}{100\hat{\beta}_{COST}}$$

[2] This can be confusing for many students; just remember that the difference between generic and alternative-specific is in the coefficients, not the variables.

Report the value of time you calculate. Is this reasonable?

2. **Ratio of Time** Estimate a model with just the out-of-vehicle travel time (`OVTT`) and the in-vehicle travel time (`IVTT`). What is the ratio of these parameters? What does this tell you about how people feel waiting for the bus?

3. **Density** Estimate a model with the residential population density (`RSPOPDEN`) and the workplace employment density (`WKEMPDEN`), controlling for the affordability of the trip (`COSTINC`). Does land use at the origin or the destination of the trip affect the choice problem more?

4. **The Model** The Wasatch Front HBW mode choice model is quite complicated, but the coefficients are given on page 57 of the model documentation (you should read this section). Using variables that you select, try and build a predictive model using data that could be available at the mode choice step in a travel demand model. Try and get a ρ^2 value above 0.25.

When you have estimated your final model, present it as a system of equations, such as the auto ownership models on page 34 of the Wasatch Front model documentation. Present your other three models in a model parameters table.

Submit your analysis as a memorandum in PDF format. Your discussion should consider why certain parameters take certain values and whether your results are reasonable.

4.2 Lab

As you showed in your estimated models, people care much more about the time they spend waiting for the bus than the time actually traveling. This causes a particular problem for transit agencies, because they often have few mechanisms to limit wait time. On-time reliability is important, but is often difficult to implement, and reducing transit vehicle headway can be extraordinarily expensive.

The advent of real-time data, such as that available in the "OneBusAway" software application, provides a way to limit initial wait time on transit services, at least theoretically. The idea is that people will track the bus and only go wait for it when they know it is almost there. The impact of these applications is potentially huge, but may be difficult to forecast.

One way to simulate the effect of real-time passenger information on passenger decision-making is to cap the initial wait time imposed on transit riders. Typically, the initial wait time is half of the transit service headway (a 30 minute bus will have a 15 minute wait time).[3] The model also imposes a minimum wait time of 5 minutes on bus modes and 3 minutes on rail modes.

These parameters are controlled by the `4_ModeChoice/Ms/block/trnb_static.param.block` files. There are two of these files, one each for walk and drive access transit trips. The are called every time a transit skim is created, so any changes will be included in the entire model. There are a number of other things this file does, but the most important for our purposes are the `IWAITMIN` and `XWAITMIN` variables. There is another option in Cube Voyager that goes unused here, `IWAITMAX`. Add the following line to both the walk and drive parameters files:

```
IWAITMAX[4]=6,6,6,4,4,6
```

This basically guarantees that people wait for the bus for exactly five minutes, regardless of its headway.

Run a 2030 scenario with your capped initial wait times. Compare the transit ridership from this scenario with a 2030 Base scenario[4], at both a macroscopic and microscopic scale. Questions you could consider in your analysis:

- Which trip type is most sensitive to changes in initial wait time?
- Are rail lines or bus routes more affected by real-time data?
- Which routes gain the most riders in absolute or percentage terms? What do these routes have in common?

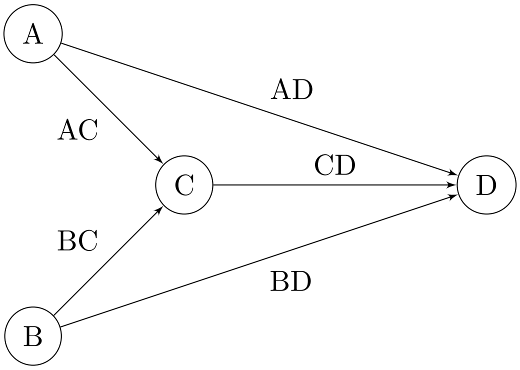
Submit your analysis in a memorandum.

[3] Outlined in the model documentation.

[4] This means that you will need to run a 2030 scenario

5. Route Assignment

You have been given[1] the following network:



7000 vehicles travel from A to D, and 5000 from B to D (node C generates no trips). Link travel time functions are given as

$$\begin{aligned}
t_{AD} &= 20.0 + 0.01V_{AD} \\
t_{AC} &= 10.0 + 0.005V_{AC} \\
t_{CD} &= 12.0 + 0.005V_{CD} \\
t_{BC} &= 7.25 + 0.005V_{BC} \\
t_{BD} &= 20.0 + 0.01V_{BD}
\end{aligned}$$

1. Solve for the user equilibrium (UE) link flows and travel times (HINT: write and solve a set of simultaneous equations that explicitly define the UE conditions). Demonstrate that your solution is the user equilibrium by showing through example that all UE conditions are satisfied.
2. Perform four iterations of All Or Nothing (AON) assignment on the network and O/D volumes. Show the link flows and travel times at the end of each iteration.
3. Perform an incremental assignment, using trip table increments of 25% for each step. Show the link flows and travel times at the end of each incremental assignment.
4. Assign trips using the FHWA assignment heuristic. Show the link flows and travel times for four assignments, and the final average assignment and resulting travel times.

[1] This is adapted from Dr. John Ivan at UConn

5. Compare the three traffic assignment heuristic approaches to the UE assignment and to each other. How do the resulting flow patterns differ (cite specific differences)? Which one comes closest to the UE flows?

6. State in words the general theory underlying each of the heuristic approaches. Which one do you prefer and why? Consider accuracy, ease of computation and the underlying theory.

All of these exercises are doable by hand. For this assignment, you may choose to submit your answers in class on engineering calculation paper (professional quality presentation is still expected). Students who program the solutions (by optimizing to find the UE conditions or implementing a graph algorithm) will earn substantial extra credit.

6. Validation

In our labs thus far, we have treated the outputs of the model as “truth.” This is not really a good assumption; for starters, we know that all of the underlying models have confidence intervals. It would be responsible to extend these confidence intervals to the ridership and volume estimates, but this would triple or quadruple the number of calculations required. Also, the assumptions that we make to simplify the model may introduce bias. For instance, the transfer penalty may inflate the ridership of long bus routes at the expense of shorter ones.

Whenever we start using a newly-assembled travel demand model, we need to *validate* that the numbers coming from it represent reality at some level. If they do not, we can *calibrate* the model to better approximate observed conditions.[1] Data for validation can be obtained from departments of transportation, transit agencies, custom traffic counts, or even the underlying surveys used to construct the model. While we should be careful about circular validation, going back to the NHTS or other surveys can identify any distortions that your assumptions introduced in the intervening steps.

6.1 Lab

Because the most recent year for which data is available is 2007, you will need to run a 2007 base scenario. You may as well get this started now.

6.1.1 Highway Volumes

UDOT maintains records of average annual daily traffic for important roads in the state, and makes maps of them publicly available on its website at `http://www.udot.utah.gov/main/f?p=100:pg:0::::V,T:,528`. Luckily for you, the planners at WFRC and MAG have already included the AADT for the most important links as fields in the highway network file (`AWDT07`, etc.). Analyzing the fit between the model and the UDOT Counts is simply an issue of comparing the two vectors in R.

The data you need is in the `COMPARE_.....CSV` file. Basically, what we want to do is compare the `ModelAWDT` and `ObsAWDT` fields. Because the UDOT counts aren't available for every link, however, you need to restrict your analyis to links where $AWDT07 > 0$.

Create a plot of AWDT versus the forecasted two-way volume. Include the average trend line. What is the slope of this trend line? What should this slope be? What is the correlation coefficient? Are these good enough for you?

[1] Calibration can introduce its own error, however. Relying on an over-calibrated model weakens the behavioral theory that should govern forecasts.

Separate the data into the functional types listed below, and provide a plot of each type's observed and forecasted volumes. Which type has the best fit, and the most realistic slope? Which type has the worst? What does this say about the accuracy of the model in general?

- Freeways
- Multilane Highways
- Principal Arterials
- Principal Rural Highways

6.2 Transit Ridership

Validating transit ridership is a little more difficult, because UTA is not required to make route-level ridership data available to the public. Also, the data they have is not entirely trustworthy (in my opinion). I have attached a spreadsheet to this assignment showing what UTA uses internally, and what they would probably give to the MPO for validation purposes. I have also included a `BusRoutes.csv` file to spare you the effort of macro-processing the spreadsheet. You should note that the route number needs to be separated from the region prefix in the model.

Create a plot comparing the bus ridership with that predicted in your model. This time, do a thorough analysis on which routes are over or under predicted. Does the transit comparison show any traits that the highway links did not? What factors lead the model to heavily skew a prediction for a transit route? What does this say about the faith you should place in the model outputs?

7. Land Use Models

This lab requires students to examine spatial autoregressive models. In R, this is done with the `spdep` library. Additional plotting commands are available with the `ggmaps` and `ggplot2` libraries.

```
library(spdep)
library(ggmap)
library(apsrtable)
```

The data we will use for this analysis are provided on T-Square in the `HomePrices.Rdata` file. These data are evaluations of home prices in the central Atlanta in 2012, obtained from the Fulton County Tax Assessor's Office. The data are stored in a `SpatialPointsDataFrame` object that already contains projection data. These points are plotted in Figure 7.1.

```
load("./HomePrices.Rdata")
```

The dataset contains the following fields:

`X2012` The home's assessed value in 2012.

`X,Y` The home's coordinates in latitude and longitude.

`INCOME2010` The median income in the home's Census tract according to the 2010 Census.

`PCTWHITE` The percentage of white householders in the home's Census tract, according to the 2010 Census

`NEAREST_ STATION` The nearest MARTA station to the home, by Euclidean distance.[1]

`Acres` The acres of the home's property.

`EffAge` The home's effective age, considering improvements to the structure and plumbing or electrical systems.

`Rmtot` The number of rooms in the home.

`Rmbed` The number of bedrooms in the home.

`Fixbath` The number of bathrooms in the home.

[1] Extra points to someone who can use R to efficiently calculate network distance.

```r
# Plot stamen background map
Atlanta <- get_map(bbox(ClassPointsLL), source='stamen', maptype='toner')
# Create ggmap object to project points onto
AtlantaMap <- ggmap(Atlanta, extent="device")
AtlantaMap + geom_point(data = ClassPointsLL@data,
                        aes(x=X, y=Y, color=NEAREST_STATION), alpha=0.8)
```

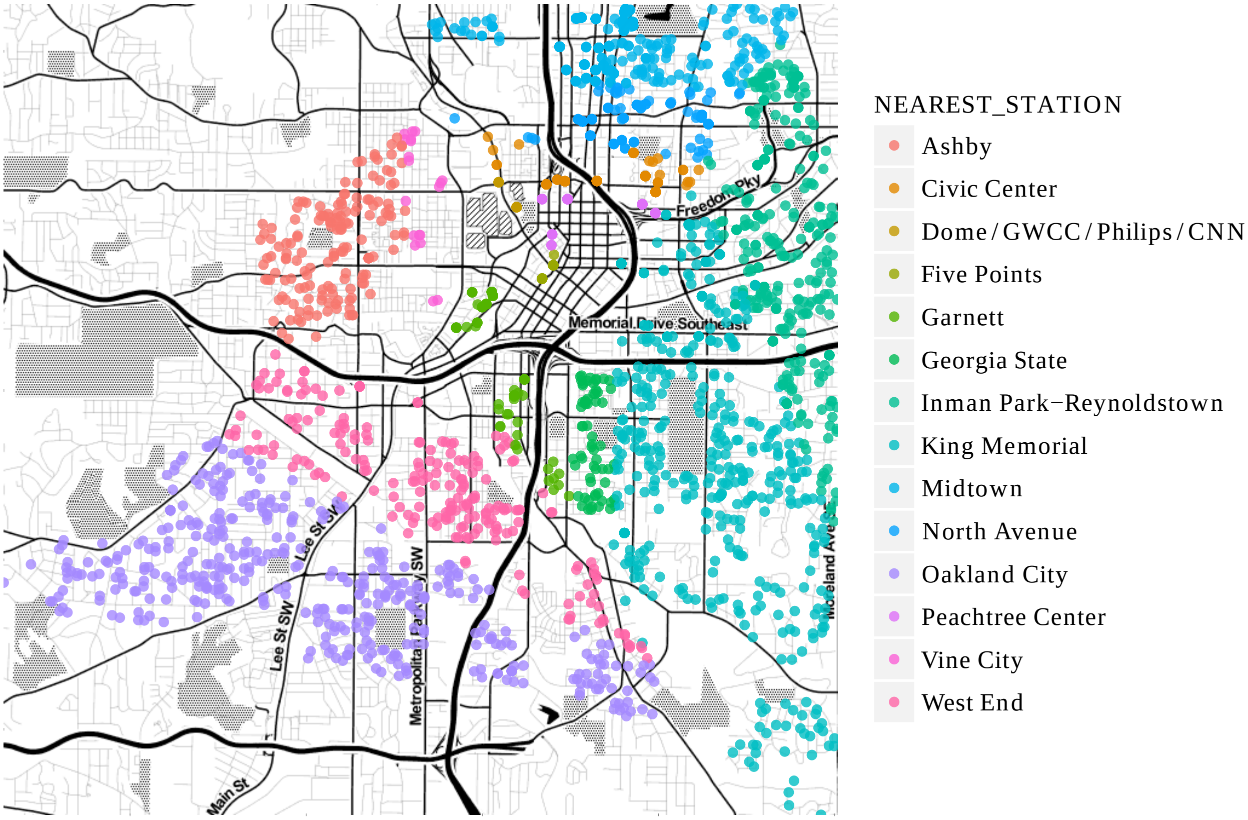


Figure 7.1: Homes in the analysis dataset, by nearest MARTA station.

7.1 Correlation in OLS

An important element of spatial models is a weights matrix, W, that captures the spatial relationship between each point in the dataset. Create a matrix that weights points by the inverse distance between them $(1/d_{ij})$, for the 50 nearest neighbors. Note that because some points are at the same location (condos), you will have to accomodate the fact that d_{ij} could equal zero; also note that the distances are given in kilometers.[2]

```
points.knn <- knn2nb(knearneigh(ClassPointsLL, k=50))

## Warning:  knearneigh:  identical points found

dists  <- lapply(nbdists(points.knn, ClassPointsLL), function(x) x)
dists.inv <- lapply(dists, function(x) 1/(x+0.01))
W <- nb2listw(points.knn, glist=dists.inv, style="W")
```

Run a least squares regression model ($y = X\beta$) to predict the price[3] of a home given its size, acreage, age, and neighborhood income and racial composition. Comment on the implications of this model for home price. Specifically, what features of a house lead to it having a higher value? Test, using the Moran's I autocorrelation statistic,[4] whether the residuals from this model are spatially autocorrelated. Provide a map of the points by OLS residual.

7.2 Autoregressive Models

Estimate SAR, SEM, and SDM versions of your OLS model. These models are

$$\text{SAR} : y = \rho W y + X\beta + \epsilon \tag{7.1}$$

$$\text{SEM} : y = X\beta + u, u = \lambda W u + \epsilon \tag{7.2}$$

$$\text{SDM} : y = \rho W y + X\beta + \gamma W X + \epsilon \tag{7.3}$$

```
# SAR
lagsarlm(formula, listw=W, data = ClassPointsLL@data)
#SEM
errorsarlm(formula, listw=W, data = ClassPointsLL@data)
#SDM
lagsarlm(formula, listw=W, type="mixed", data = ClassPointsLL@data)
```

Which model provides the best statistical fit? Plot the residuals of the SDM model, and comment on their spatial distribution. What happens to the explanatory variables you identified as being statistically significant in your OLS model when you use a spatial autoregressive model? Comment on the implications of this for research and practice.

[2] For details, see `?nbdists`.

[3] You should use a log transform here.

[4] The `spdep` package contains a `moran.test` function

8. Activity-based Models

9. Final Project

For this project, your student group (of three or four students) is to complete an alternatives analysis on a project of your choice. You will need to code the alternatives in the travel demand model, thoroughly analyze the model output, and identify a preferred alternative.

There are a good deal of options for this project. You may feel free to select from the list below, or to develop one on your own. Potential projects include:

- What rail operating plan gives the “best” operating scenario?
- What are the best transit options for connecting Brigham Young University to the FrontRunner commuter rail?
- What is the best transportaion alternative along the 5600 West corridor in West Valley City?
- What are the transportation service consequences of BYU closing Campus Drive and moving UTA bus service to 900 East?
- Are HOV lanes along I-215 justified? In which segments are they most useful?
- What would be the transportation effects should Davis County adopt transit-oriented land use policies?

You are required to submit three deliverables: an initial project identification and scope of work, a technical report, and a professional presentation.

Scope Once you have chosen your group and identified your area of interest, submit a project scope and methodology to me. This one-page document should describe your project’s purpose and your proposed alternatives, and outline the modeling steps and methodologies required. I will arrange a kickoff meeting with your entire group to approve your project and provide help and caution regarding how your alternatives may be coded. Due February 14.

Report Each analysis should include at least five competing alternatives, including the “no-build” alternative. The major component of the project will be to code the alternatives for near-term (2016) and long-term (2030) planning horizons. Your technical report will include the following items:

1. Purpose and Needs statement, describing the transportation problem you seek to solve.
2. Description of each proposed alternative.

3. Discussion of appropriate model-based performance measures to compare modeled alternatives.
4. Results of the alternatives comparison including appropriate tables and graphics.
5. Identification of a preferred alternative.

The report will be due (as a PDF emailed to me) at the final presentation.

Presentation Your group will present the results of your analysis in a professional presentation. This will be held during the final exam period on May 2, 2014. Your presentation should be about 20 minutes long, with an additional ten minutes for questions from the instructors and the class.