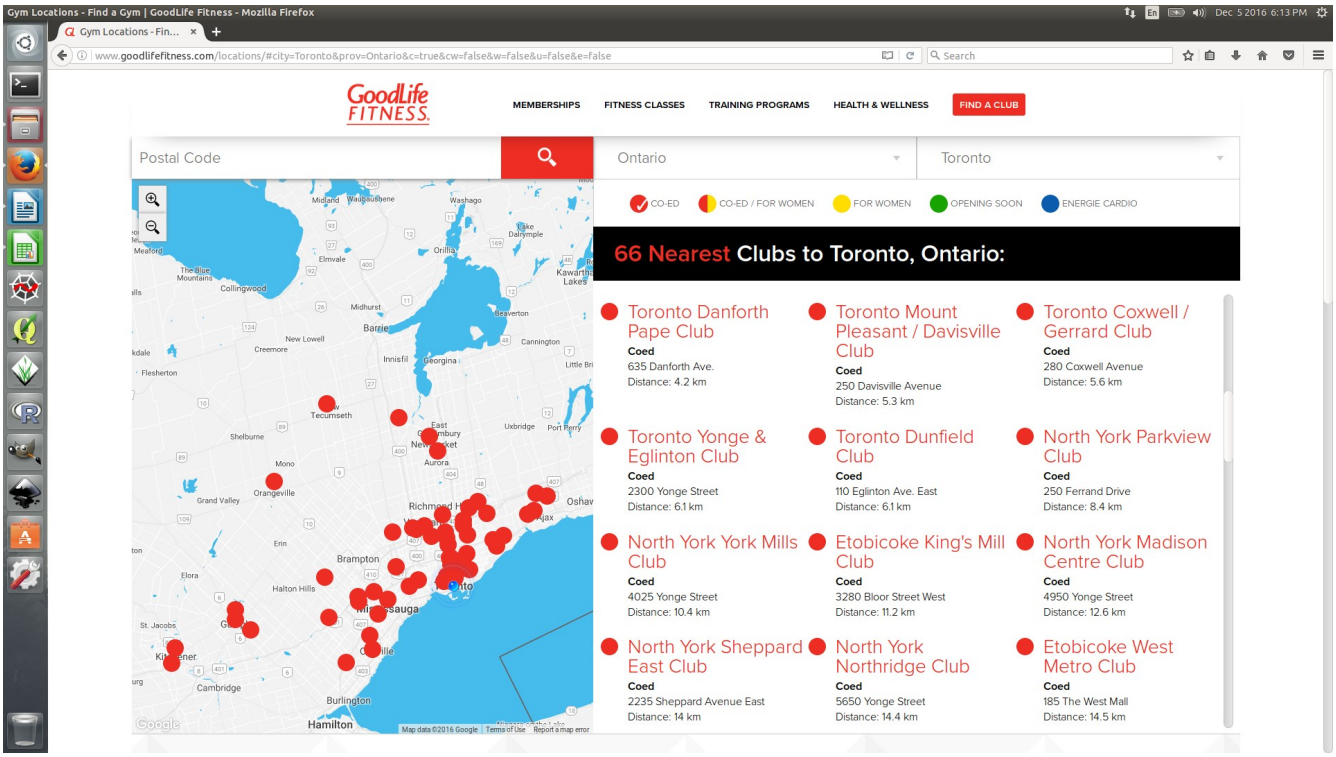
The Huff Model in QGIS: A Goodlife Fitness Case Study

There is interest in developing an open source resource or tool to solve Huff Model problems. It has been suggested that the capabilities of QGIS (geographic information software) could be extended with the use of a plugin written in the Python programming language. To assist in the understanding of the tasks required for such a project, a hypothetical case study is used. This approach allows for the creation of a dataset that can be employed to explore a project of this nature. Goodlife Fitness is the focus of the case study and their co-ed gyms in the Toronto census metropolitan area (CMA) have been used to explore the existing capabilities of standard spreadsheet applications and a standard installation of QGIS 2.14-Essen. Additional capabilities that would be required of a new plugin are also considered.

Step 1: Getting Goodlife Fitness Data

To make use of the Huff Model with QGIS, data is required. Depending on the particular situation, data may not be readily available and may need to be collected before the Huff Model may be used. To start exploring Goodlife Fitness, data has been manually collected from the Goodlife Fitness web site (http://www.goodlifefitness.com/), presented below, and entered/pasted into a spreadsheet.



Data collected for each co-ed club located in the Toronto CMA includes: location name, street addresss, city, province, and postal code. An additional field has been created to hold an arbitrary unique identifier “GFID”. The values for GFID are simply “GF” plus a number starting at 1001. This is done to force the values in this field to be both unique and of string data type. This is important later in QGIS. The spreadsheet has been constructed as presented in the screen capture below.

	A	B	C	D	E	F
1	GFID	Name	Address	City	Province	PostalCode
2	GF1001	Toronto Bell Trinity Centre	483 Bay Street	Toronto	Ontario	M5G2C9
3	GF1002	Toronto 137 Yonge Street	137 Yonge Street	Toronto	Ontario	M5C1W6
4	GF1003	Toronto Yonge & Dundas Square	319 Yonge Street	Toronto	Ontario	M5B2C3
5	GF1004	Toronto Plaza	100 Yonge Street	Toronto	Ontario	M5C2W1
6	GF1005	Toronto McCaul/Queen St. West	21 McCaul Street	Toronto	Ontario	M5T1V7
7	GF1006	Toronto Wellington West	111 Wellington St. West	Toronto	Ontario	M5J2S6
8	GF1007	Toronto Richmond John	267 Richmond Street West	Toronto	Ontario	M5V3M6
9	GF1008	Toronto Union Station	7 Station Street	Toronto	Ontario	M5J1C3
10	GF1009	Toronto Richmond/Bathurst	555 Richmond Street West	Toronto	Ontario	M5V3B1
11	GF1010	Toronto Manulife Centre	210 - 55 Bloor St. W.	Toronto	Ontario	M4W1A5
12	GF1011	Toronto Bloor Yorkville	80 Bloor Street West	Toronto	Ontario	M5S2V1
13	GF1012	Toronto Bloor Park	8 Park Road	Toronto	Ontario	M4W3G8
14	GF1013	Toronto College Street	533 College Street	Toronto	Ontario	M6G1A8
15	GF1014	Toronto King/Liberty	85 Hanna Ave. Suite 200	Toronto	Ontario	M6K3S3

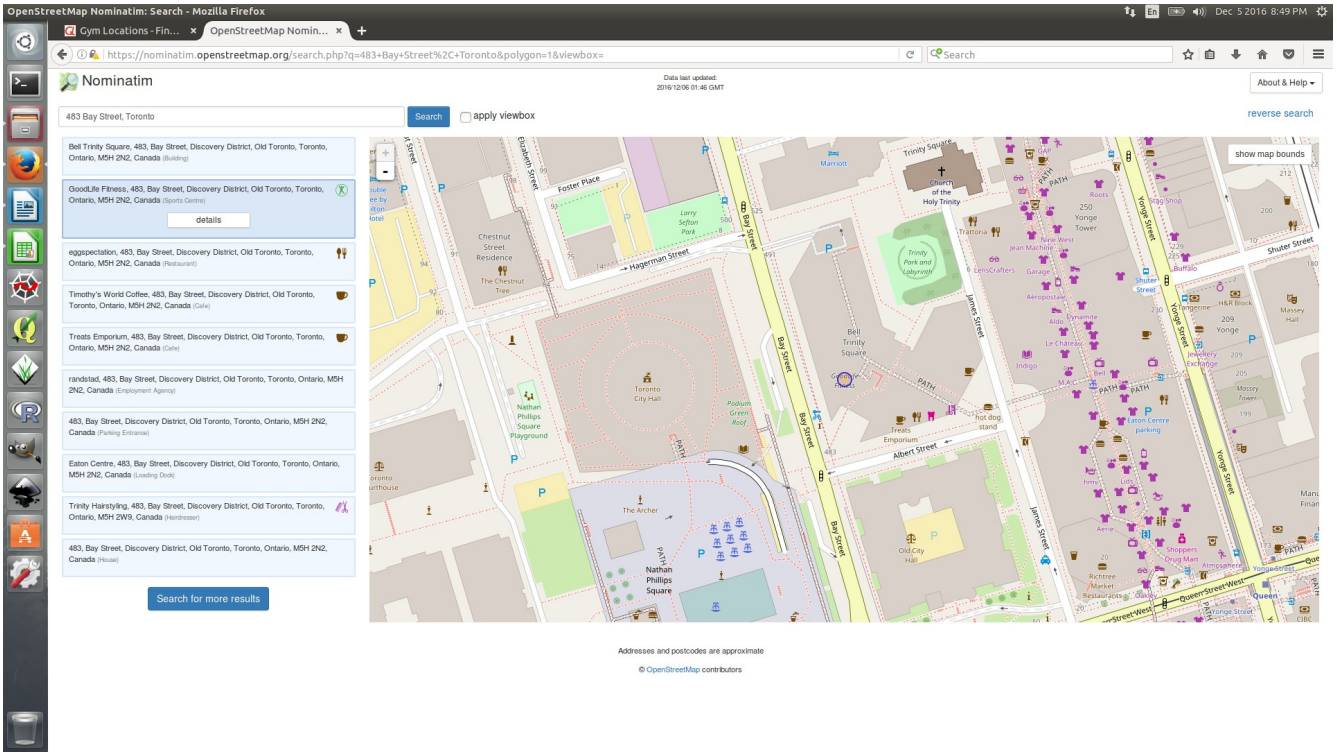
A total of 60 locations have been recorded for Goodlife Fitness locations within the Toronto CMA. Six locations presented on the Goodlife website, towards Kitchener, are outside the Toronto CMA and have not been captured.

It should also be noted that Goodlife Fitness has not been consistent with the format of their addresses. In some cases units or suites come after the street name and in others they come before. Abbreviations are also used inconsistently. This formatting may create issues if the process of identifying the associated latitude and longitude for each address is automated. The use of the old borough names and some punctuation used may also be problematic and may need to be corrected.

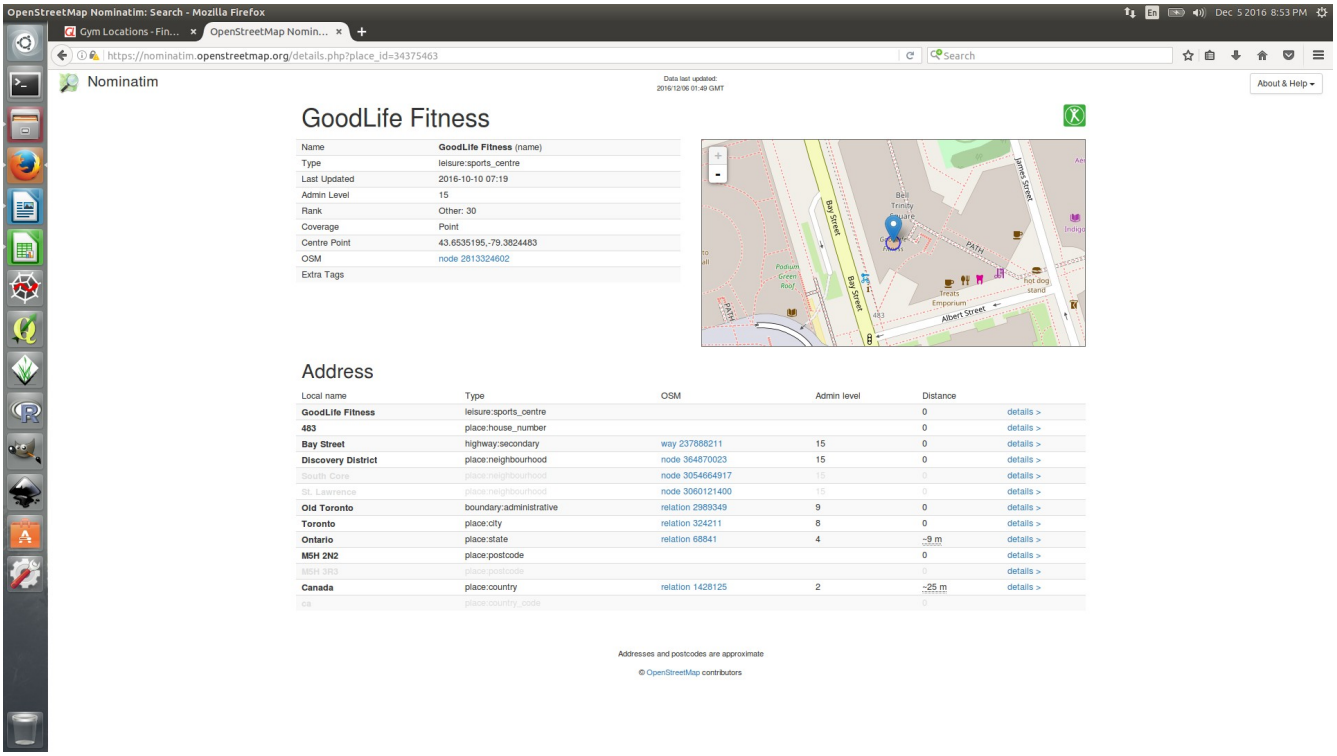
Step 2: Getting Latitude and Longitude Values for Goodlife Fitness Addresses

Geocoding addresses has recently become more challenging as services that have been freely available in the past have evolved into fee-based services. To find an open source, free, and efficient method of geocoding addresses is a problem all by itself. As the number of addresses in projects using the Huff Model are relatively small, the latitude and longitude for each address have been manually sourced and added to the current spreadsheet. Two new colums were created, “Lat” and “Lon”, and then the cells in these columns were populated with the appropriate values.

To obtain the latitude and longitude coordinates for each address, OpenStreetMap's Nominatim geocoding service was used. See: https://nominatim.openstreetmap.org/ . Searching for the first address “483 Bay Street, Toronto” found the first Goodlife Fitness location. See the screen capture below.

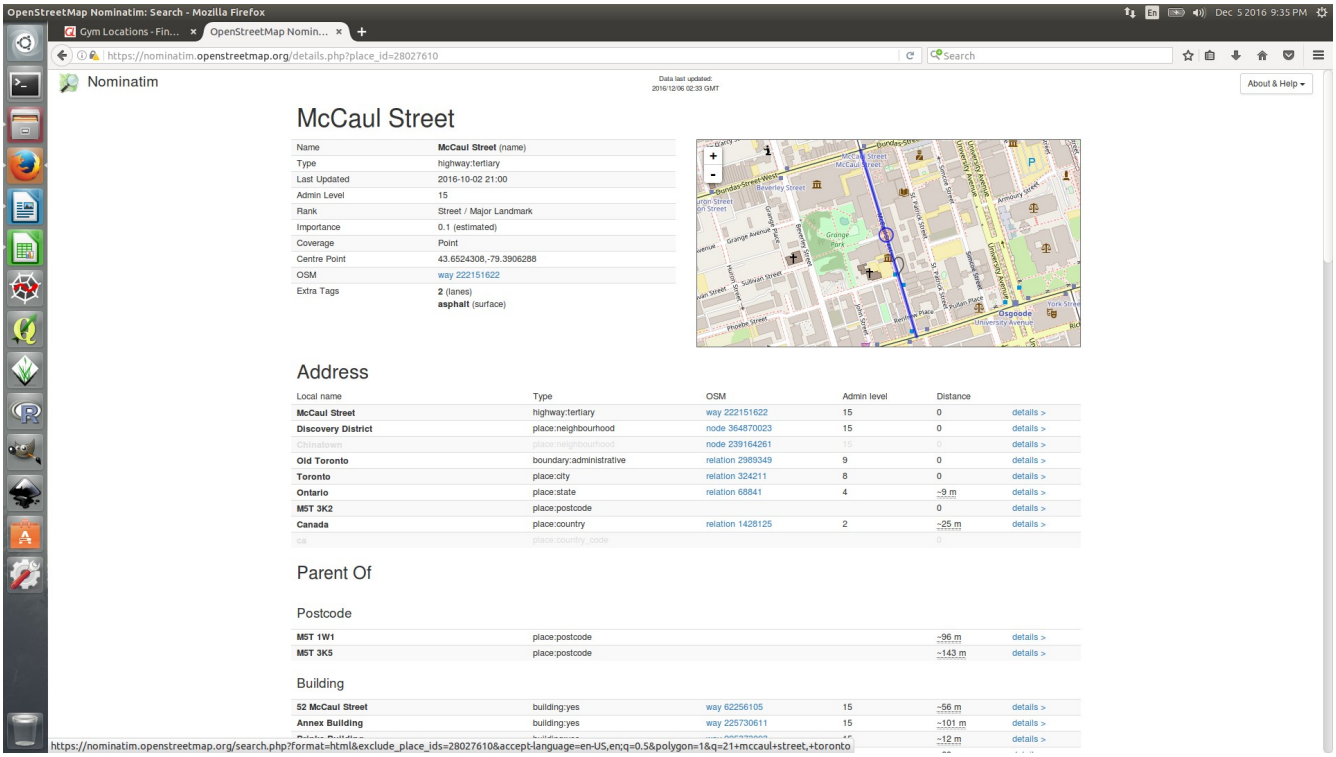


Clicking on the “details” button for the Goodlife Fitness result presents a second page with location details including the latitude and longitude. See the next screen capture.



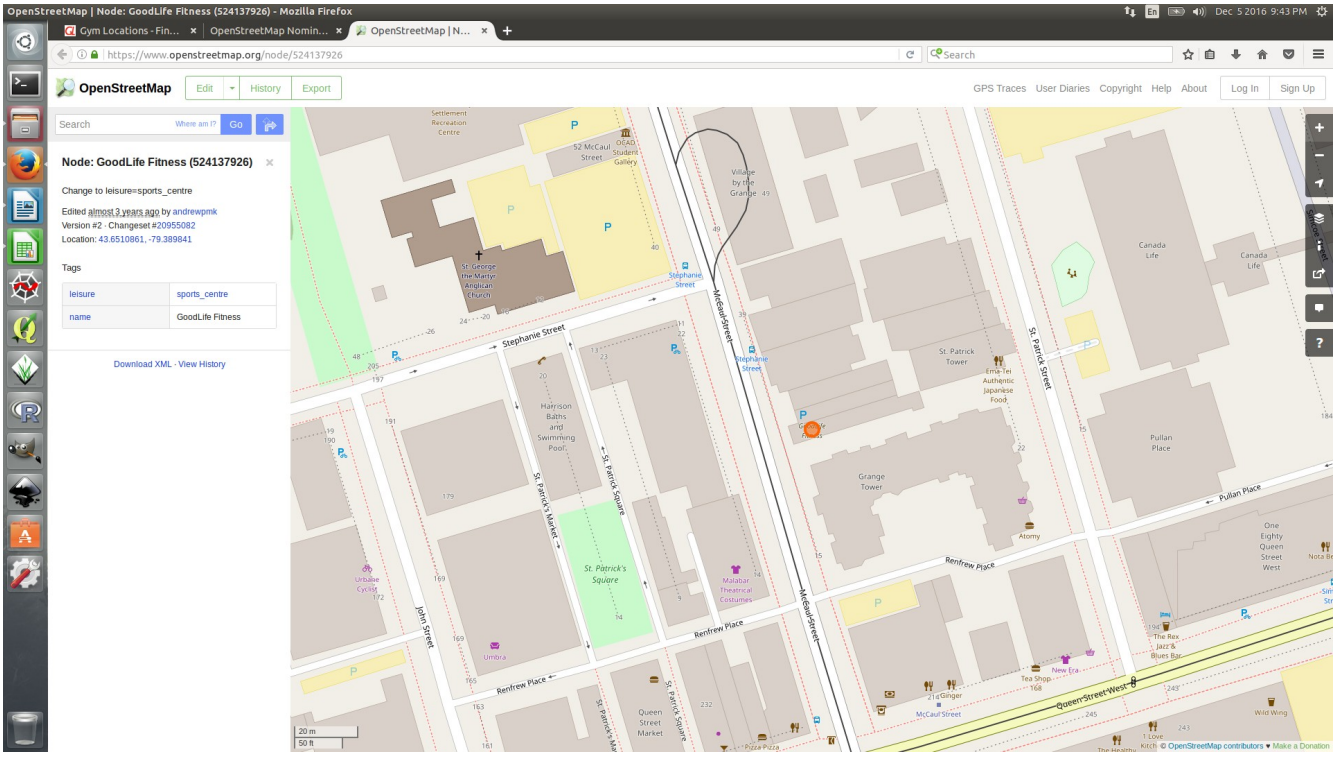
It should be noted that Nominatim seems to get the postal codes wrong and using postal codes in

queries may not be useful; a street address and city seems to work best. The “Centre Point” item in the upper table presents the latitude and longitude values. While the example in the screen capture above is pretty accurate, not all are. Some results use a centre point of the line segment for the street that the point of interest is located on. See the screen capture below for an example of how the location on McCaul Street in Toronto is presented in this way. Note the blue line between Queen Street West and Dundas Street West. The circle in the middle of the line represents the centre point offered for this address.



A better result may be offered by scrolling down the page. If the Goodlife Fitness location has been recorded by OpenStreetMap, it may have a more accurate location. See the next screen capture slice.

This Goodlife Fitness location is listed and additional information is available by clicking the blue “node” link. This opens a page on openstreetmap.org with more accurate location information. See the following screen capture.



With the addition of the latitude and longitude values, the spreadsheet evolved as presented in the screen capture below.

	A	B	C	D	E	F	G	H
1	GFID	Name	Address	City	Province	PostalCode	Lat	Lon
2	GF1001	Toronto Bell Trinity Centre	483 Bay Street	Toronto	Ontario	M5G2C9	43.6535195	-79.3824483
3	GF1002	Toronto 137 Yonge Street	137 Yonge Street	Toronto	Ontario	M5C1W6	43.6511	-79.3784479
4	GF1003	Toronto Yonge & Dundas Square	319 Yonge Street	Toronto	Ontario	M5B2C3	43.6579071	-79.3815849
5	GF1004	Toronto Plaza	100 Yonge Street	Toronto	Ontario	M5C2W1	43.64924	-79.3780769
6	GF1005	Toronto McCaul/Queen St. West	21 McCaul Street	Toronto	Ontario	M5T1V7	43.6510861	-79.389841
7	GF1006	Toronto Wellington West	111 Wellington St. West	Toronto	Ontario	M5J2S6	43.6464153	-79.3831754
8	GF1007	Toronto Richmond John	267 Richmond Street West	Toronto	Ontario	M5V3M6	43.648791	-79.3918819
9	GF1008	Toronto Union Station	7 Station Street	Toronto	Ontario	M5J1C3	43.6445604	-79.3835105
10	GF1009	Toronto Richmond/Bathurst	555 Richmond Street West	Toronto	Ontario	M5V3B1	43.646697	-79.4020449
11	GF1010	Toronto Manulife Centre	210 - 55 Bloor St. W.	Toronto	Ontario	M4W1A5	43.6696112	-79.3886006
12	GF1011	Toronto Bloor Yorkville	80 Bloor Street West	Toronto	Ontario	M5S2V1	43.66992405	-79.3901913276
13	GF1012	Toronto Bloor Park	8 Park Road	Toronto	Ontario	M4W3G8	43.6714601	-79.3847340999
14	GF1013	Toronto College Street	533 College Street	Toronto	Ontario	M6G1A8	43.655568	-79.4111385
15	GF1014	Toronto King/Liberty	85 Hanna Ave. Suite 200	Toronto	Ontario	M6K3S3	43.6401067	-79.4197758
16	GF1015	Toronto St. Clair/Yonge	12 St. Clair Avenue East	Toronto	Ontario	M4T1L7	43.6886189	-79.3936835
17	GF1016	Toronto Danforth Pape	635 Danforth Ave.	Toronto	Ontario	M4K1R2	43.6786807	-79.3453936
18	GF1017	Toronto Mount Pleasant/Davisville	250 Davisville Avenue	Toronto	Ontario	M4S2L9	43.700477	-79.3865509
19	GF1018	Toronto Coxwell/Gerrard	280 Coxwell Avenue	Toronto	Ontario	M4L3B6	43.673324	-79.319625
20	GF1019	Toronto Yonge & Eglinton	2300 Yonge Street , Yonge Eglinton Centre	Toronto	Ontario	M4P1E4	43.7067505	-79.399362

With all the location data collected, only the “size” data used by the Huff Model formula needed to be provided for and this was done next.

Step 3: Generating Data to Represent a Goodlife Fitness Location's Size

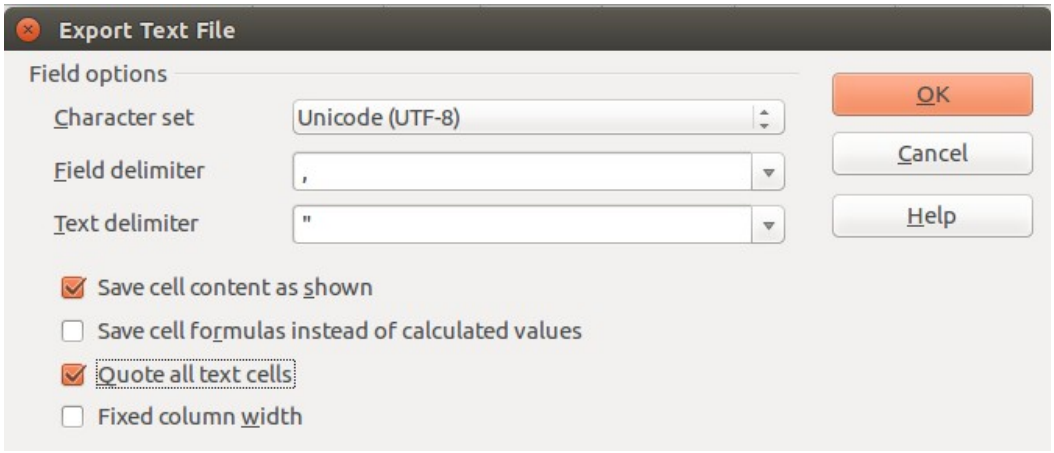
Without having actual data for each Goodlife Fitness location, data needed to be randomly generated for the purposes of this case study. To provide guidance to this task, a Toronto Star article was leveraged (https://www.thestar.com/business/2015/01/20/fitness-chain-interested-in-target-properties.html). This article states that typical locations are about 24,000 square feet. With this in mind club sizes have been randomly generated by a spreadsheet formula to be within 15,000 and

35,000 square feet (some are larger). The resulting formula for LibreOffice Calc is “RANDBETWEEN(15000,35000)”. This new data appears as the last column “SizeSqFt” in the spreadsheet. The completed spreadsheet is presented in the screen capture below.

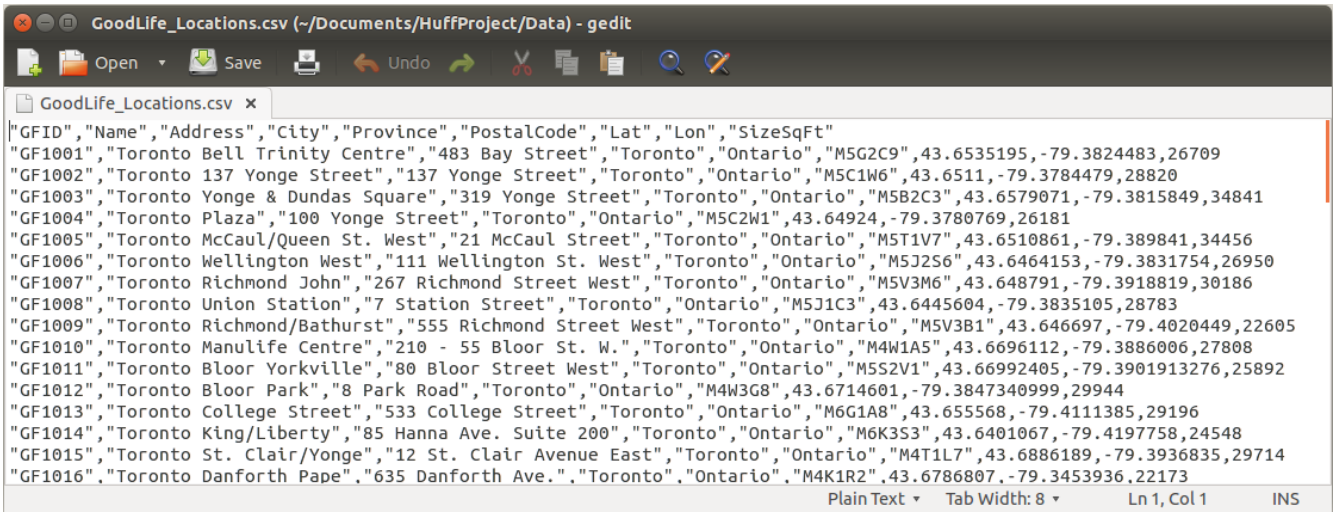
	A	B	C	D	E	F	G	H	I
1	GFID	Name	Address	City	Province	PostalCode	Lat	Lon	SizeSqFt
2	GF1001	Toronto Bell Trinity Centre	483 Bay Street	Toronto	Ontario	M5G2C9	43.6535195	-79.3824483	26709
3	GF1002	Toronto 137 Yonge Street	137 Yonge Street	Toronto	Ontario	M5C1W6	43.6511	-79.3784479	28820
4	GF1003	Toronto Yonge & Dundas Square	319 Yonge Street	Toronto	Ontario	M5B2C3	43.6579071	-79.3815849	34841
5	GF1004	Toronto Plaza	100 Yonge Street	Toronto	Ontario	M5C2W1	43.64924	-79.3780769	26181
6	GF1005	Toronto McCaul/Queen St. West	21 McCaul Street	Toronto	Ontario	M5T1V7	43.6510861	-79.389841	34456
7	GF1006	Toronto Wellington West	111 Wellington St. West	Toronto	Ontario	M5J2S6	43.6464153	-79.3831754	26950
8	GF1007	Toronto Richmond John	267 Richmond Street West	Toronto	Ontario	M5V3M6	43.648791	-79.3918819	30186
9	GF1008	Toronto Union Station	7 Station Street	Toronto	Ontario	M5J1C3	43.6445604	-79.3835105	28783
10	GF1009	Toronto Richmond/Bathurst	555 Richmond Street West	Toronto	Ontario	M5V3B1	43.646697	-79.4020449	22605
11	GF1010	Toronto Manulife Centre	210 - 55 Bloor St. W.	Toronto	Ontario	M4W1A5	43.6696112	-79.3886006	27808
12	GF1011	Toronto Bloor Yorkville	80 Bloor Street West	Toronto	Ontario	M5S2V1	43.66992405	-79.3901913276	25892
13	GF1012	Toronto Bloor Park	8 Park Road	Toronto	Ontario	M4W3G8	43.6714601	-79.3847340999	29944
14	GF1013	Toronto College Street	533 College Street	Toronto	Ontario	M6G1A8	43.655568	-79.4111385	29196
15	GF1014	Toronto King/Liberty	85 Hanna Ave. Suite 200	Toronto	Ontario	M6K3S3	43.6401067	-79.4197758	24548
16	GF1015	Toronto St. Clair/Yonge	12 St. Clair Avenue East	Toronto	Ontario	M4T1L7	43.6886189	-79.3936835	29714
17	GF1016	Toronto Danforth Pape	635 Danforth Ave.	Toronto	Ontario	M4K1R2	43.6786807	-79.3453936	22173
18	GF1017	Toronto Mount Pleasant/Davisville	250 Davisville Avenue	Toronto	Ontario	M4S2L9	43.700477	-79.3865509	31587
19	GF1018	Toronto Coxwell/Gerrard	280 Coxwell Avenue	Toronto	Ontario	M4L3B6	43.673324	-79.319625	25485
20	GF1019	Toronto Yonge & Eglinton	2300 Yonge Street , Yonge Eglinton Centre	Toronto	Ontario	M4P1E4	43.7067505	-79.399362	26724
21	GF1020	Toronto Dunfield	110 Eglinton Ave. East	Toronto	Ontario	M4P1A6	43.7075676	-79.3952713	21273
22	GF1021	North York Parkview	250 Ferrand Drive	Toronto	Ontario	M3C3G8	43.7195101	-79.3319053	30079
23	GF1022	North York York Mills	4025 Yonge Street	North York	Ontario	M2P2E3	43.7445107	-79.4060282	32926
24	GF1023	Etobicoke King's Mill	3280 Bloor Street West , Centre Tower 2nd Floor, Suite Mez 1	Etobicoke	Ontario	M8X2X3	43.6357244	-79.5622346	25042
25	GF1024	North York Madison Centre	4950 Yonge Street	North York	Ontario	M2N6K1	43.7648919	-79.4119615	23681

Step 4: Creating a Comma-Separated Values (.csv) File for the Goodlife Fitness Data

Data that has been prepared for the Goodlife Fitness locations needed to be put into a format that can be imported into QGIS. A common file format that can be used for this task is the comma-separated values (.csv) format that reduces the data to a plain-text file. This was achieved by using the “Save As...” command and saving the file as a different file type. Note that some additional specifcations are usually required; a delimiter needs to be specified (the “comma” in “comma-separated values”) and quotes (“”) to indicate text values. All text should also be in quotes. Some of the data in the addresses has multiple parts and is already separated by commas and this ensures it is compartmentalized properly.



The result of the file export can be viewed in a common text editor and the new file is presented below. Note the quotation marks on the text fields and that the first row includes the column header values.

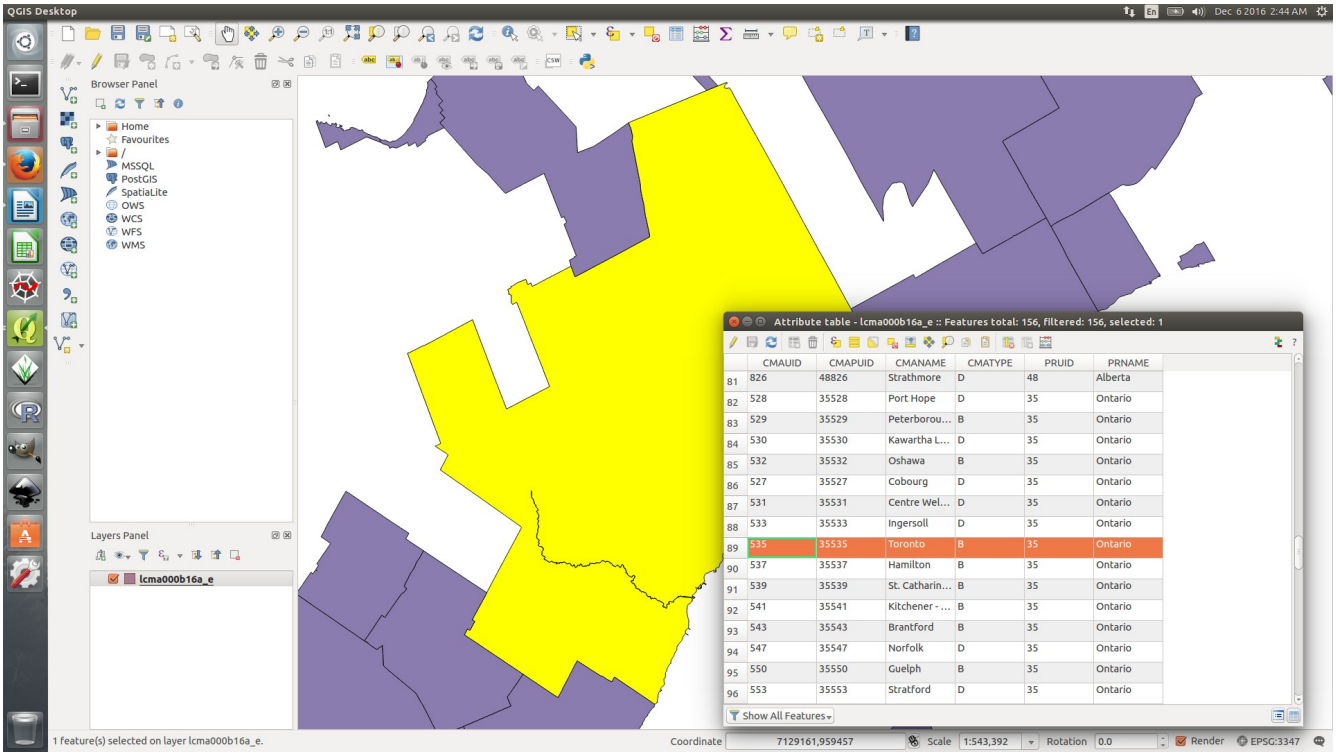


Step 5: Collecting Additional Data Files from Statistics Canada

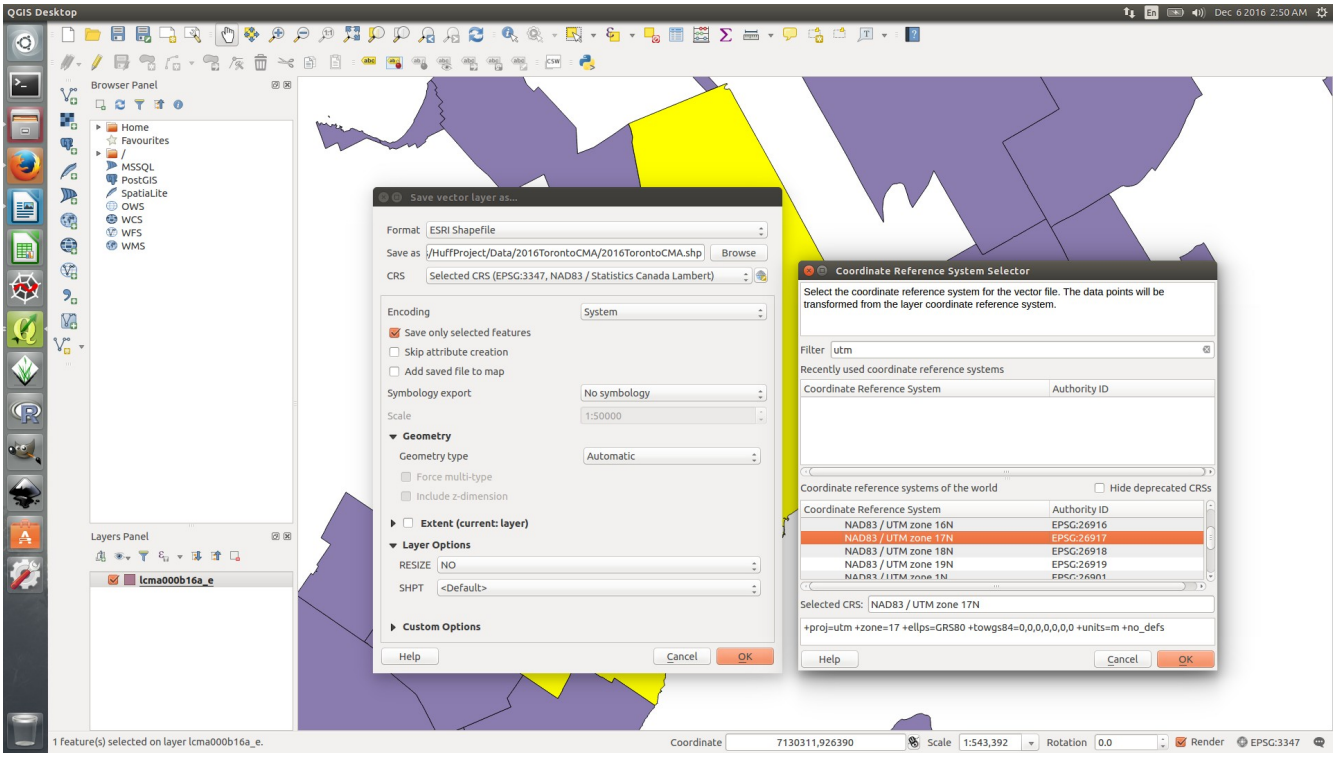
In addition to data for Goodlife Fitness locations, data was also required for the study area. In this case, the Toronto CMA has been specified as the study area and census tracts have been used for the regions for which Huff probabilities are calculated. Geographic boundary files are required and they are available from Statistics Canada. 2016 boundary files are available at this location: https://www12.statcan.gc.ca/census-recensement/2011/geo/bound-limit/bound-limit-2016-eng.cfm . The data files provided by Statistics Canada include regions for all of Canada. To work with only the Toronto CMA (the study area), the data of interest needed to be extracted. This was done in QGIS and is described below.

Step 6: Extracting Regional Data from Statistics Canada Shapefiles

In QGIS, the CMA boundary file has been opened and the Toronto CMA has been selected in the attribute table as presented in the first screen capture below. The selected CMA has then been exported as a new shapefile as shown in the second screen capture. The process also allowed the new file to be saved with a more appropriate coordinate reference system for the study area.

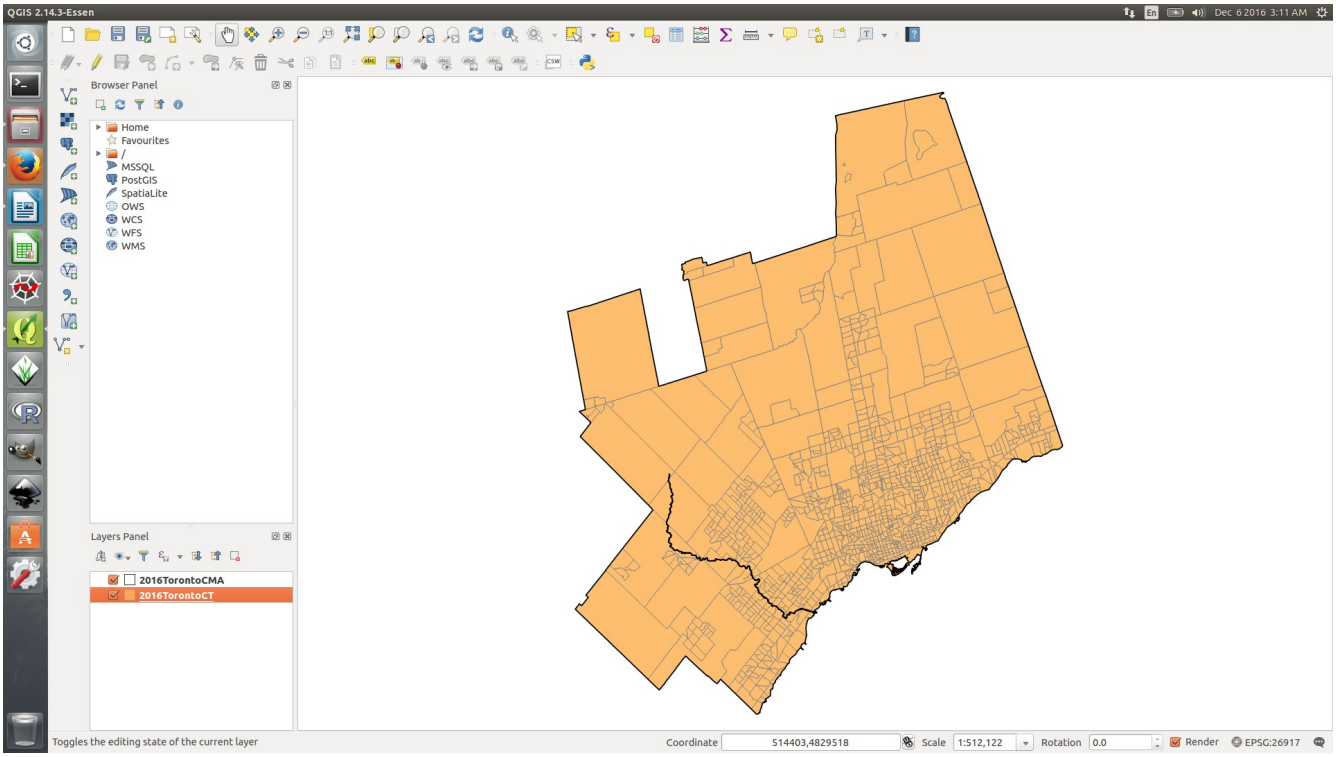


The export process appears in the next screen capture.



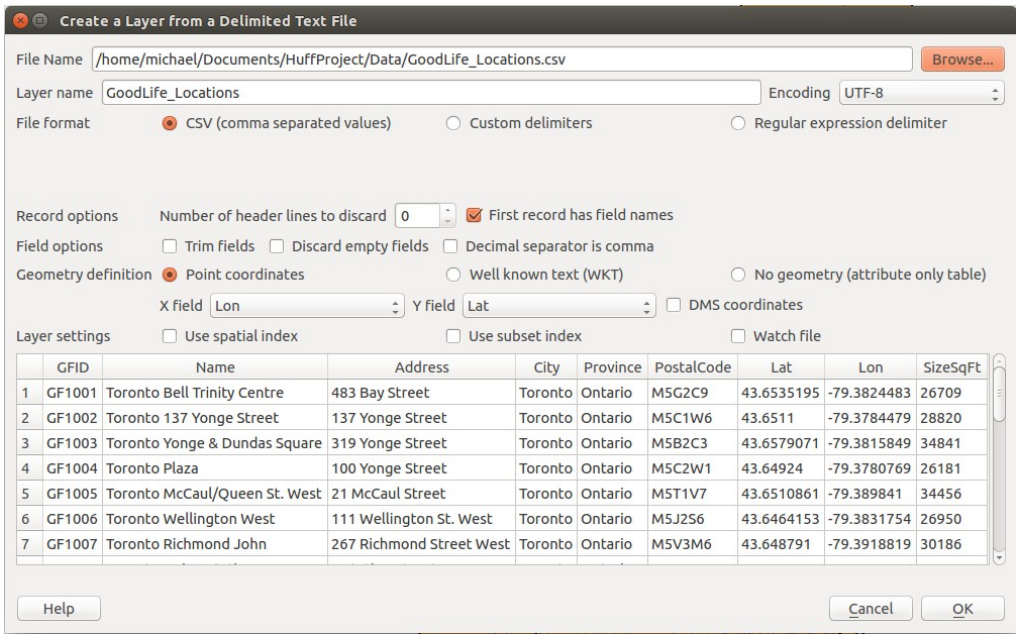
The process was repeated for the census tract files.

The result is two new sets of data files that match the study area. The census tract files will do the "work" while the CMA files have been used to create a presentation boundary. Screen capture below.



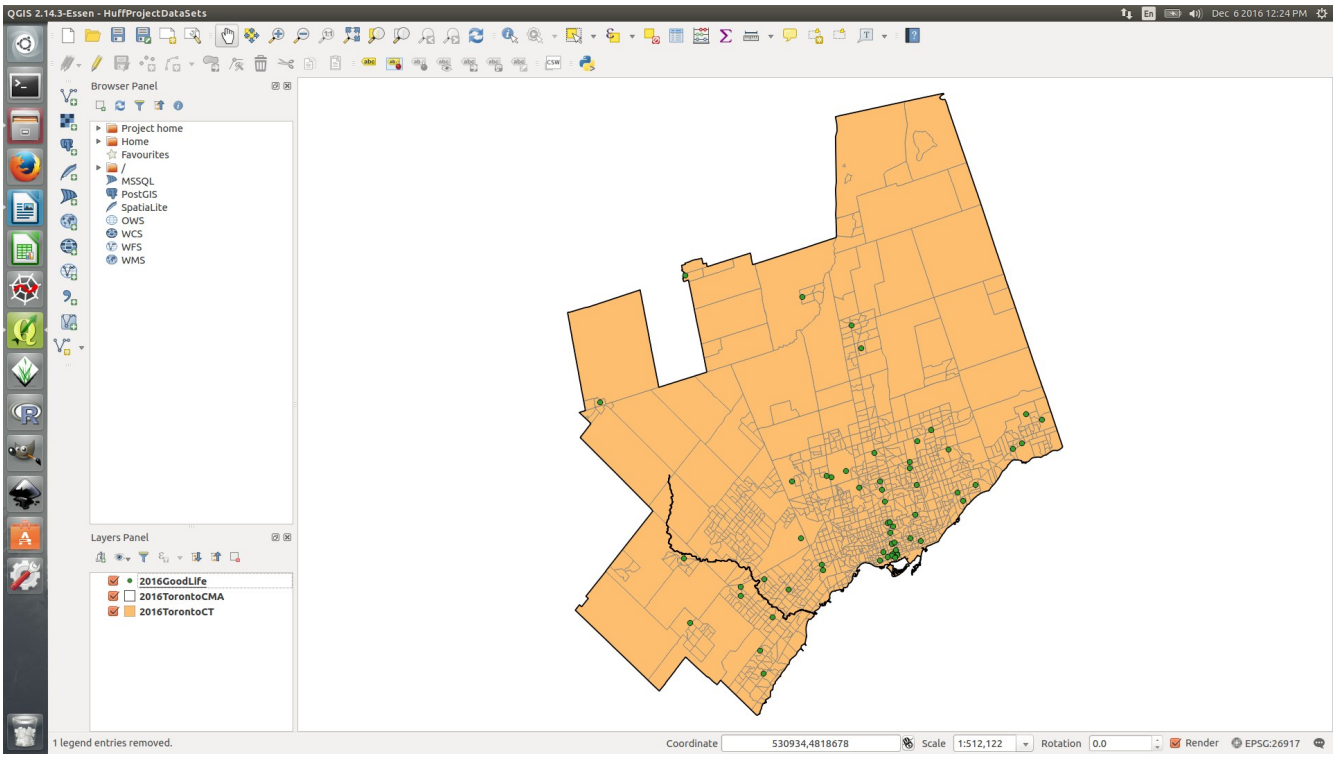
Step 7: Creating the Point Layer for Goodlife Fitness Locations

QGIS provides functionality that makes adding data stored in comma-separated value (.csv) files easy and straightforward. The next screen capture illustrates this with the required import settings.



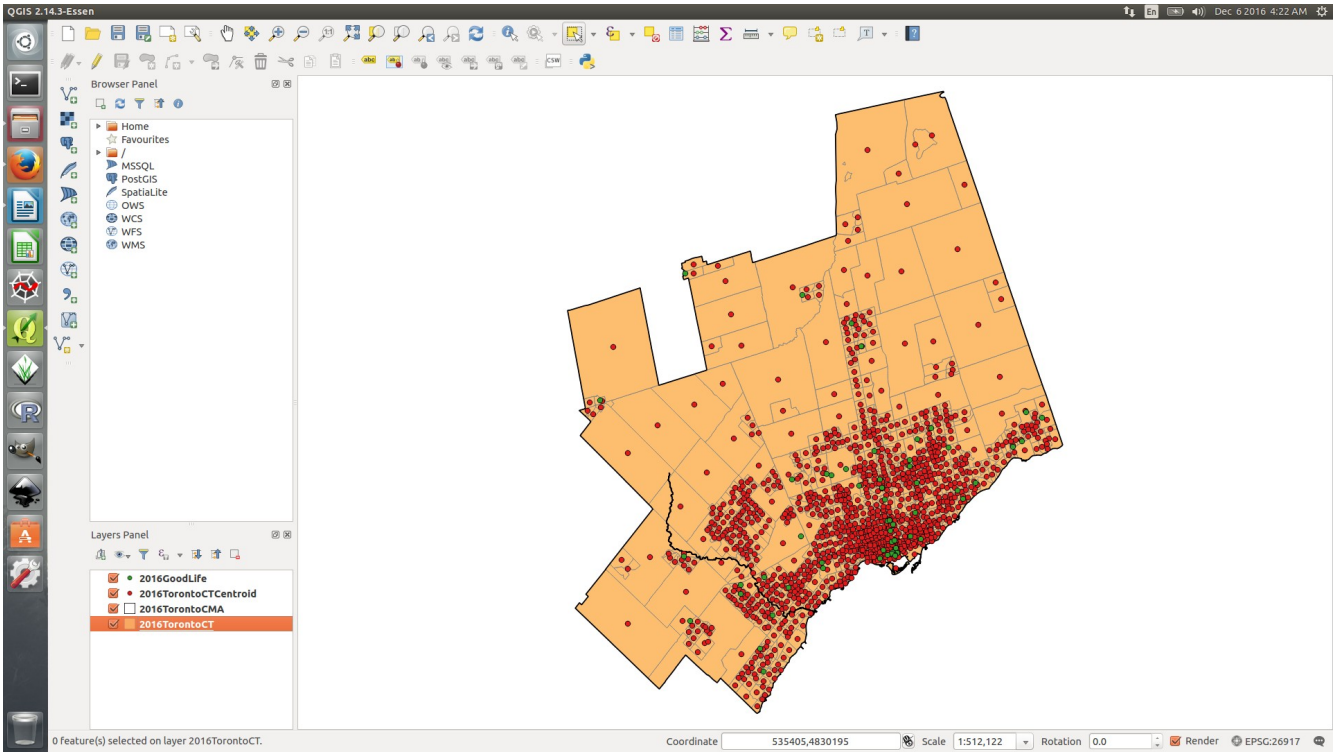
	GFID	Name	Address	City	Province	PostalCode	Lat	Lon	SizeSqFt
1	GF1001	Toronto Bell Trinity Centre	483 Bay Street	Toronto	Ontario	M5G2C9	43.6535195	-79.3824483	26709
2	GF1002	Toronto 137 Yonge Street	137 Yonge Street	Toronto	Ontario	M5C1W6	43.6511	-79.3784479	28820
3	GF1003	Toronto Yonge & Dundas Square	319 Yonge Street	Toronto	Ontario	M5B2C3	43.6579071	-79.3815849	34841
4	GF1004	Toronto Plaza	100 Yonge Street	Toronto	Ontario	M5C2W1	43.64924	-79.3780769	26181
5	GF1005	Toronto McCaul/Queen St. West	21 McCaul Street	Toronto	Ontario	M5T1V7	43.6510861	-79.389841	34456
6	GF1006	Toronto Wellington West	111 Wellington St. West	Toronto	Ontario	M5J2S6	43.6464153	-79.3831754	26950
7	GF1007	Toronto Richmond John	267 Richmond Street West	Toronto	Ontario	M5V3M6	43.648791	-79.3918819	30186

A second screen completes the process and allows for the specification of the coordinate reference system to be used for the data import. This should be “WGS 84”. Once QGIS has created the point layer, it needs to be saved. As part of the step of saving the new file, the coordinate reference system has been changed to match the other data files. This process has been used to import the Goodlife Fitness location data from the comma-separated values (.csv) file created in Step 4. The three data layers are presented together in a new project in QGIS in the screen capture below.



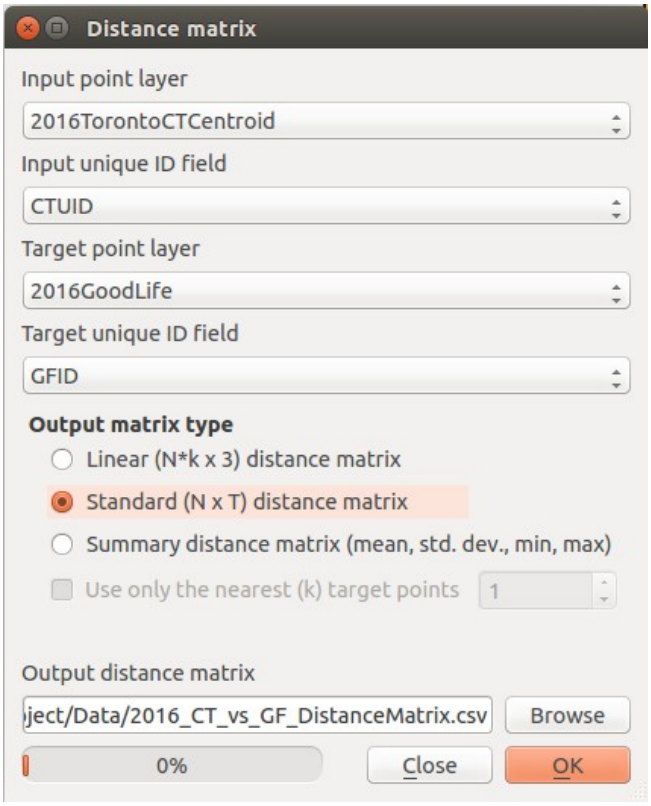
Step 8: Creating Centroids from the Census Tract Layer

The Huff Model requires that distances are calculated between each region (consumers) and each target location (centres). In this case study, census tracts are the regions and Goodlife Fitness locations are the target locations. To specify a distance as required, a single point needs to be used for each census tract and the census tract's centroid has been selected for this purpose. QGIS can prepare new data files with centroids from the polygons of an input file, see the Vector>Geometry Tools>Polygon Centroids menu. With the new centroid file, distances have been calculated between Goodlife Fitness locations and census tract centroids (shown below in Step 9). The new centroid layer has been added to the project and is presented in the screen capture below.



Step 9: Creating a Distance Matrix

To make use of the Huff Model formula, the distance between every census tract centroid and each Goodlife Fitness location is required. QGIS provides functionality to obtain these distances with fTool's Distance Matrix tool located at Vector>Analysis Tools>Distance Matrix. This tool was used to create a new table with census tracts as rows and Goodlife Fitness locations as columns. The census tract identifier (CTUID) is used to identify each row while the GFID field is used to identify clubs in each column. The output of this tool is saved in a comma-separated values (.csv) text file at a location of the user's choice. The screen capture below presents the settings required. Note that the census tract shapefile should be specified first (the rows) and the Goodlife Fitness locations should be specified second (the columns). The output matrix type should be set to “Standard (N x T) distance matrix” as this will provide one row for each census tract and one column for each Goodlife Fitness location.



The resulting file can be viewed in a spreadsheet application and appears below. Note that QGIS doesn't provide the census tracts (rows) or Goodlife Fitness locations (columns) in a paticular order. The distances provided are in the same units as the QGIS project; in this case, it is metres.

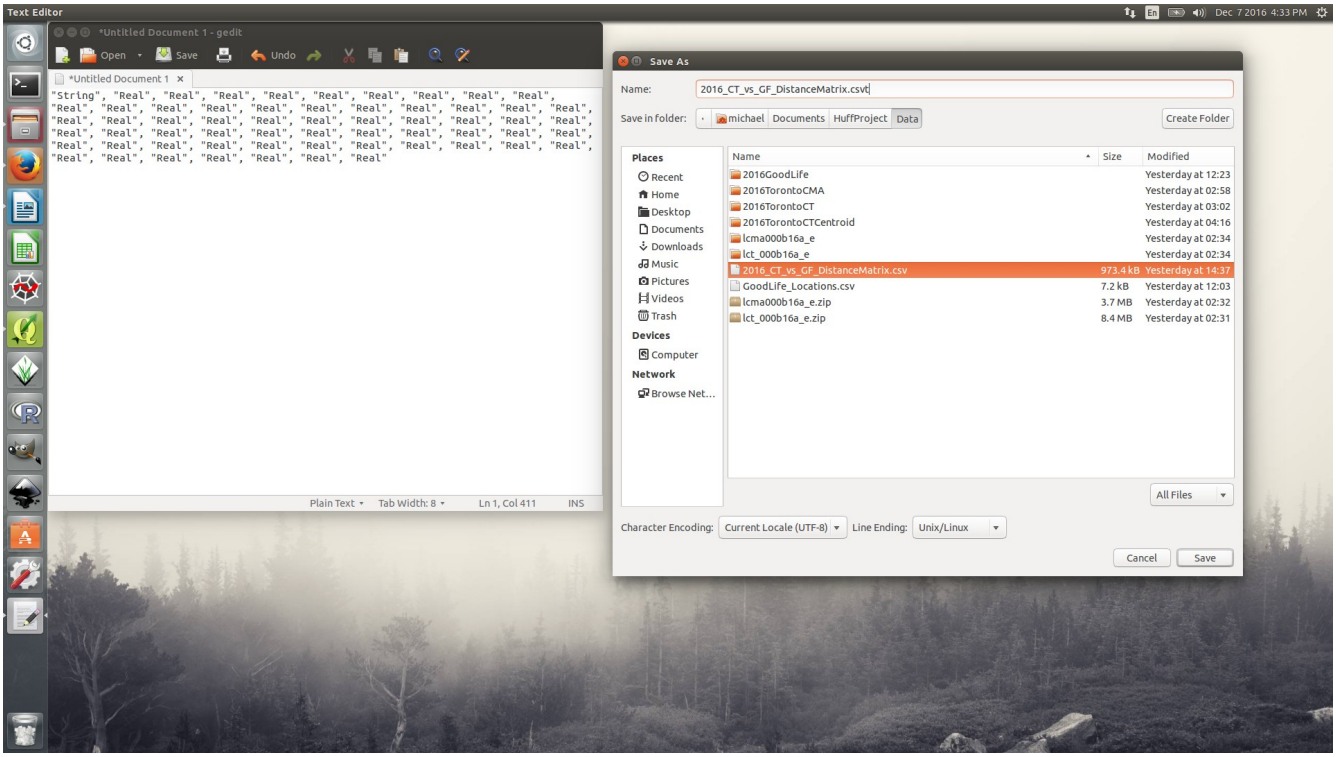
	A	B	C	D	E	F	G	H	I	J	K	L
1	ID	GF1036	GF1040	GF1032	GF1034	GF1029	GF1026	GF1043	GF1028	GF1024	GF1035	GF1037
2	5350422.06	5418.90750607	8904.11493305	9850.53168984	10478.0331223	10860.3616695	10920.3761762	11111.9241094	12529.5907209	12646.4412446	13192.9131612	13561.4842626
3	5350420.13	1602.71204166	7670.58669507	6847.71446462	7986.78028882	7552.34420507	6860.37419036	10803.0144463	8973.2143402	8577.24674527	11049.736381	11639.3187381
4	5350422.03	7427.46154397	7391.45214072	9601.93001677	13227.0569086	10822.3607435	12486.9061407	8831.69758934	14796.2395059	14134.0399042	16057.3938041	16471.5520944
5	5350422.04	6521.83028554	7189.73676291	9032.19966606	12358.5529577	10211.5961015	11616.4000734	8979.44987317	13890.1312535	13274.8198453	15216.0422098	15648.3587783
6	5350422.05	6124.34129279	7915.00701115	9418.44889271	11684.194556	10538.3072546	11420.7725442	9858.39262532	13435.9015739	13112.9476158	14482.4362877	14885.6610153
7	5350424.07	8249.10267913	12153.8605319	13332.9942032	11739.946268	14330.627433	13796.7980523	13944.2181553	14634.143113	15534.0206156	13938.9664324	14089.8907427
8	5350424.08	7233.26093146	11600.2287558	12524.7376126	10796.2666931	13480.381983	12773.6381708	13604.8304084	13616.4984332	14510.2312909	13083.6668846	13274.4683552
9	5350451.07	22384.5373694	23698.7418847	26343.8020514	25207.5099632	27547.8252394	27945.1810479	23705.2924448	28603.2857524	29683.2669428	26674.4698314	26519.5358622
10	5350451.06	23635.8098809	24814.2838844	27525.2063141	26457.1727911	28738.1026627	29197.1403783	24724.7234789	29862.2525516	30935.2030371	27900.781288	27735.6943744
11	5350516.31	36613.5529276	43673.0586986	39766.7216943	30724.669881	38834.5188042	33036.2725418	47138.7799851	29403.2971071	32067.7956708	28263.4040526	28146.4800812
12	5350516.32	39960.0858813	47675.8077222	44016.3728873	33651.0224907	43221.0237536	37122.6209638	51179.186473	33133.1933542	36401.3290769	30842.5307673	30528.1228895
13	5350520.07	31677.6866198	38221.4552817	34198.0462615	26227.3751554	33192.8957057	27646.0447727	41639.7000998	24329.1502196	26530.7239652	24123.5722551	24174.4532963
14	5350520.08	32710.6754162	39270.1846473	35246.5188104	27230.1458918	34240.3459952	28693.7214753	42688.465799	25366.0242465	27579.5189511	25094.8790261	25130.5168574
15	5350804.10	27195.4167378	18820.9529265	20456.5670213	33082.1807356	20665.1049757	26866.3890742	16410.7526894	31185.2096831	26959.5163159	36094.662771	36905.3912979
16	5350804.11	28757.2677687	20287.5928204	22101.9261273	34723.3056869	22355.9289711	28584.4578849	17717.7854797	32906.8804181	28712.5841161	37753.1701033	38555.2513616
17	5350450.05	22809.882708	22917.9529804	25962.1375853	26366.7233436	27226.8539953	28343.9679788	22467.3392092	29465.8275893	30071.5794745	28126.9163351	28065.149817
18	5350450.06	23385.0089452	22749.421672	26022.1775728	27333.5480701	27315.7798716	28862.7231718	22000.3022254	30248.4086392	30574.386168	29246.1645892	29235.3396715
19	5350573.06	32661.6040032	41195.8138261	38256.8226275	26118.1405067	37809.3368849	31526.5401555	44613.3139938	27203.1932428	31405.6932715	23002.3862609	22379.2844891
20	5350573.07	32190.879316	40651.4382198	37606.4805496	25635.5930653	37113.9056414	30818.1960559	44092.8935616	26503.9934391	30623.957463	22537.6853624	21958.5075037
21	5350576.16	28314.8569094	36989.5686038	34411.0597002	21900.9090949	34122.4328569	27964.4834427	40314.3425748	23686.5566754	28120.1489358	18798.6237471	18058.9515665
22	5350576.17	27697.0653258	36377.0218749	33825.2266287	21297.5467394	33548.9061235	27406.2216128	39694.7187063	23137.7534071	27585.1355186	18200.5313438	17452.9212624
23	5350585.09	24362.5308209	32328.6306948	31459.2799696	20120.8343912	31789.2537633	27087.8824312	34939.7727935	23952.4650174	28142.5266009	18223.1068039	17345.007072
24	5350585.10	22946.8328596	31086.48291	30027.7017221	18433.7027259	30302.3568097	25433.1641815	33805.1791624	22184.6490041	26434.9149939	16417.8208702	15530.083679
25	5350586.01	33105.3929326	41739.8362585	39823.9482428	27251.6986248	39776.1737624	33982.6611977	44811.9560904	29938.4921266	34474.5801159	24388.8164733	23520.3758838

Step 10: Joining the Distance Matrix with a Shapefile

To make use of the distance matrix created in Step 9, the new file was joined with a census tract boundary file for the Toronto CMA. This was done to make the data more accessible to a Python-based QGIS script. QGIS provides functionality for joining non-spatial data files to shapefiles, however; it's

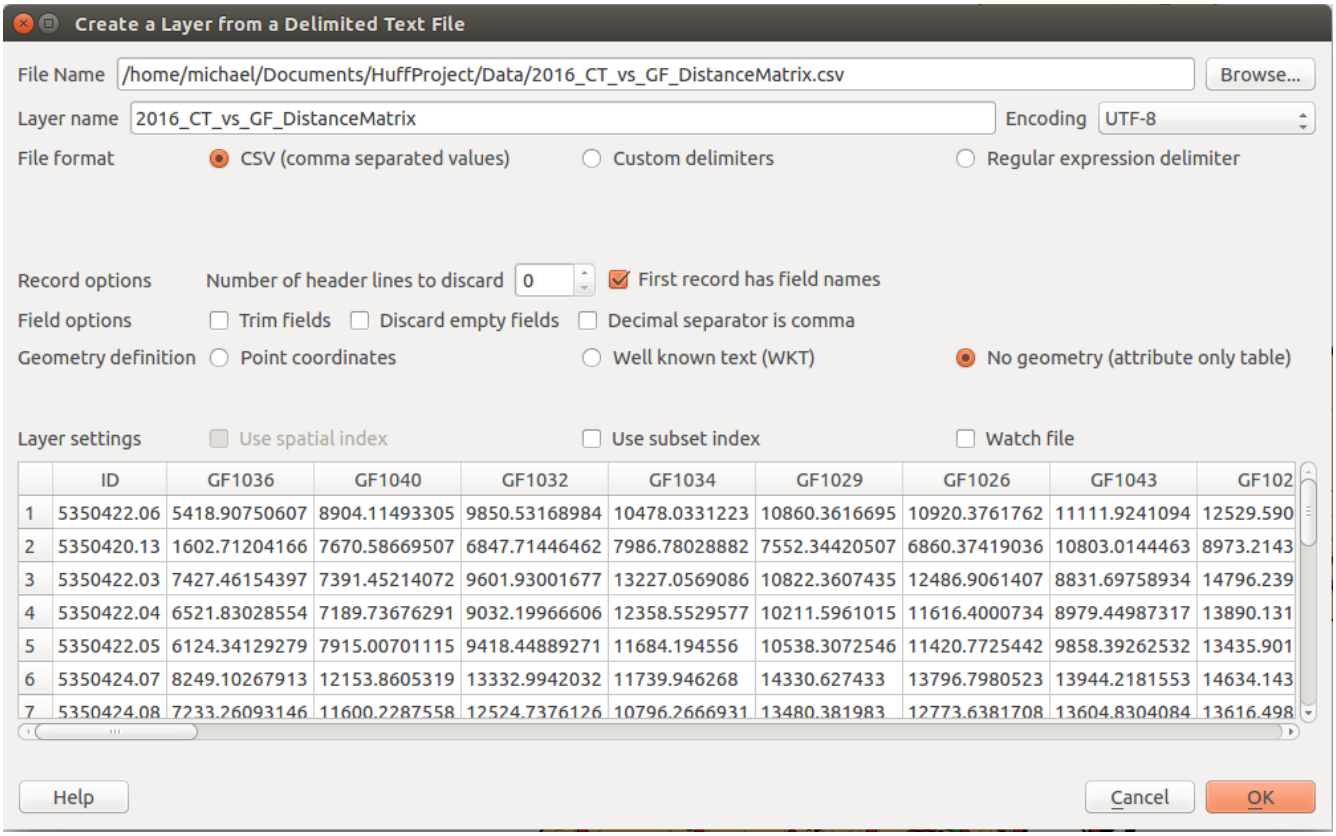
not as straightforward as in other products on the market. When QGIS imports a comma-separated values (.csv) file as a data table for a join, all values are treated as text by default. To specify the correct data types for each column, a second “helper” .csvt file needs to be created and saved in the same location as the primary .csv file. A .csvt file is simply a plain-text file created in a text editor with a comma-separated list of values for the data types to be used for each field in the paired .csv file. The .csvt file requires the same name as the primary file.

To create a .csvt file for the previously generated distance matrix file, it first needs to be recognized that there are 61 fields. The first field was defined as a text (string) field for the ID column (which is the CTUID field from the census tract shapefile) and the remaining 60 fields were defined as numeric (real) as they are the distance values and the ability to perform mathematical operations on them was required. The completed .csvt file is presented in the screen capture below.



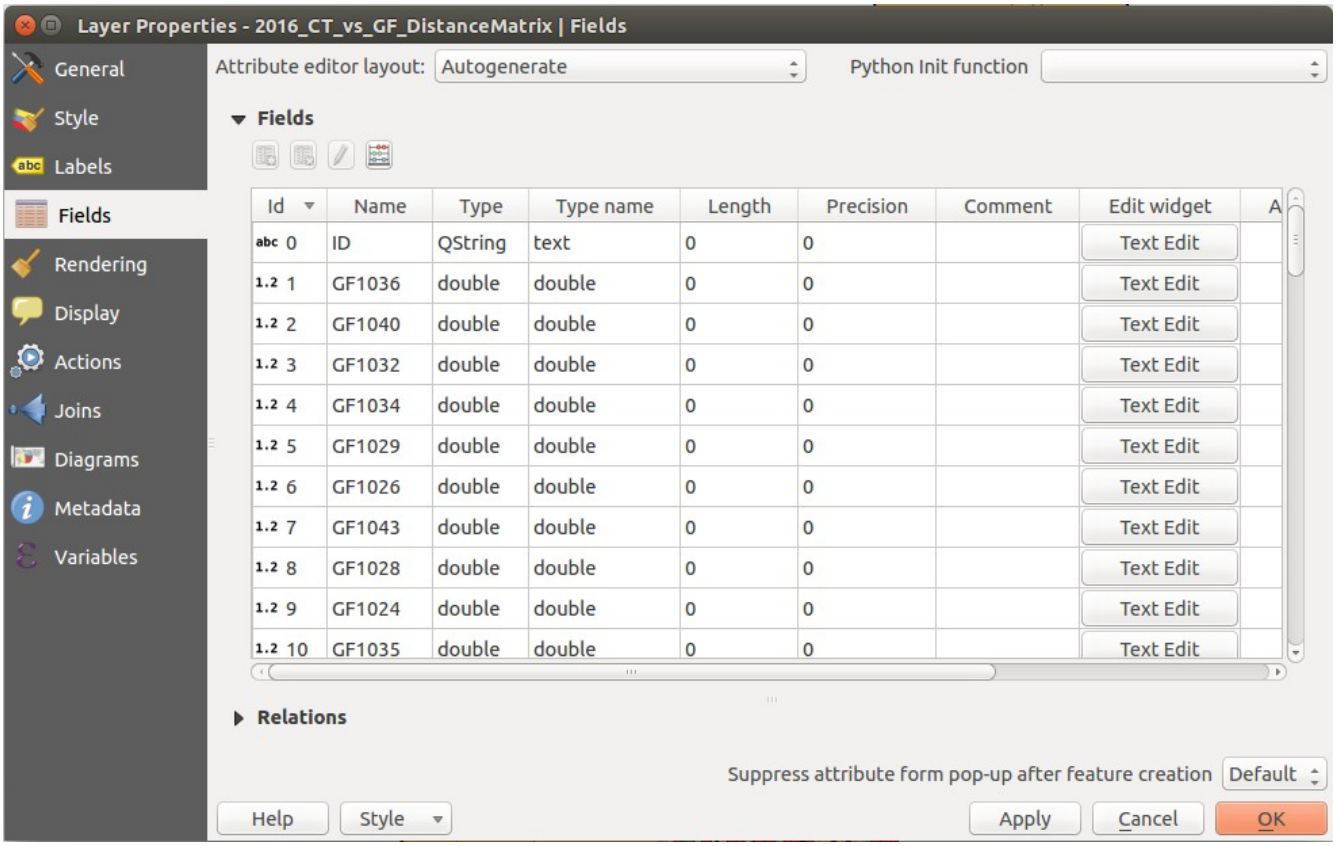
Windows users should also be careful when working with text files as programs like Notepad can sneak an extra, hidden extension (.txt) on the end of files and this will create problems.

With the two text files prepared, the join process was completed in QGIS. From the Layers menu, selecting Add Layer>Add Delimited Text Layer presents the import screen. This functionality was used earlier in Step 7, but this time the settings are a bit different. See the next screen capture below for the settings that were used for this task.



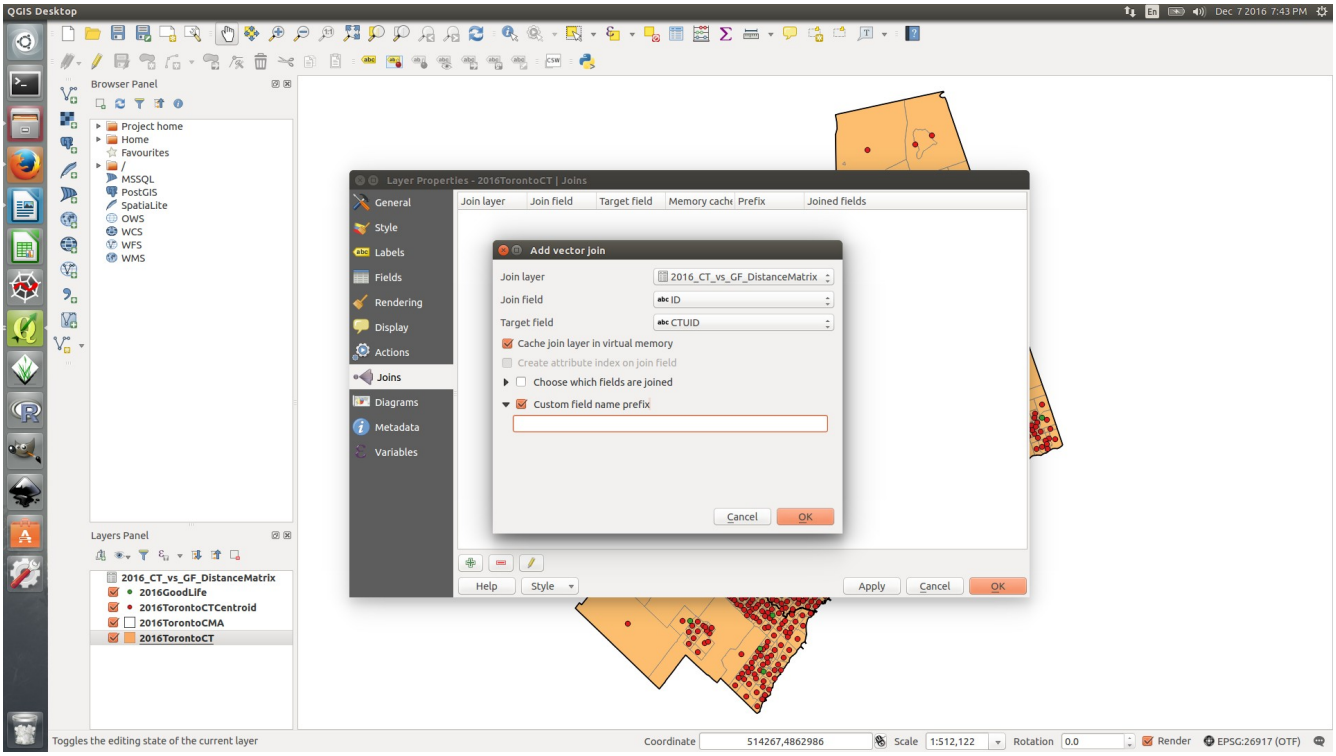
Note that the key difference here is that “No geometry (attribute only table)” needed to be selected as this data is non-geographic.

The imported data was now available in the Layers Panel and right-clicking on it provided access to the properties. The “Fields” tab confirmed that the data had been imported correctly with the desired data types. The “Fields” tab of the Properties screen is shown in the next screen capture below.

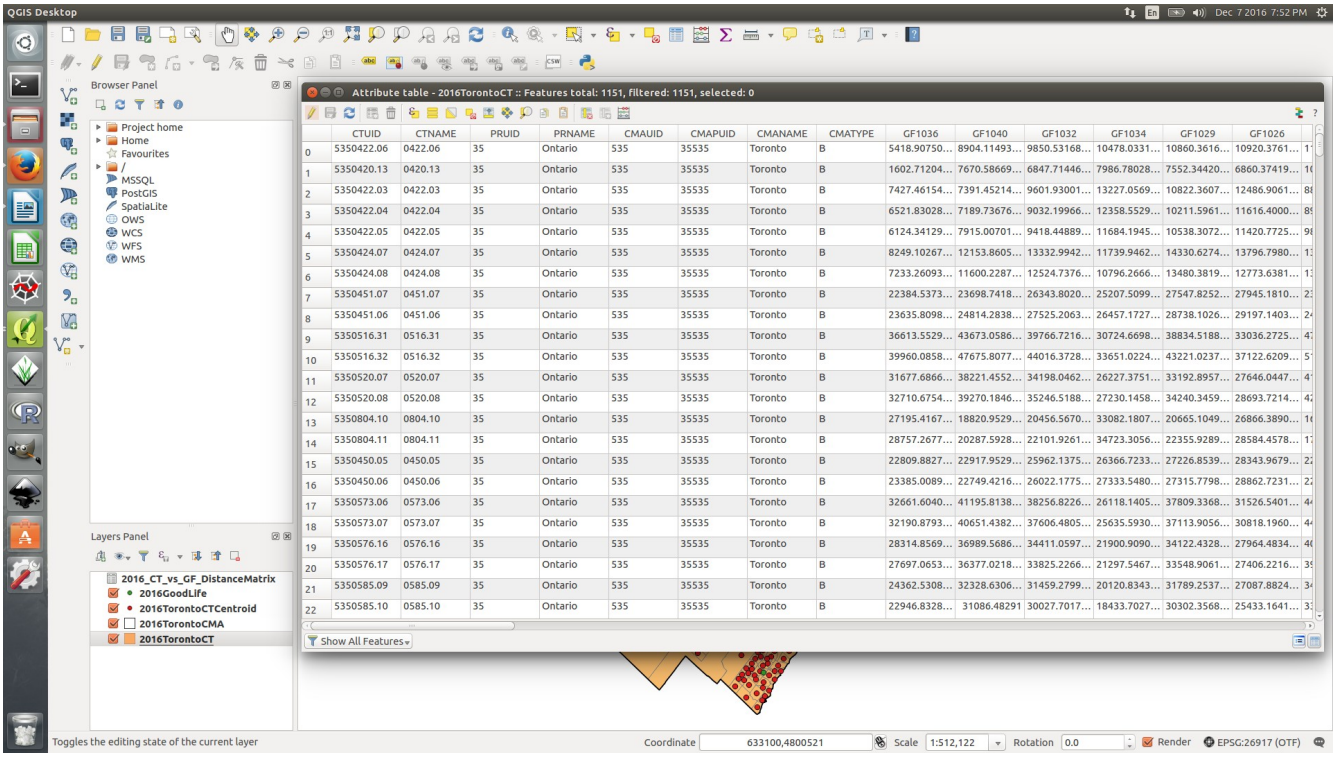


Swtiching to the properties for the census tract shapefile and using the “Joins” tab allowed for the completion of joining the distance matrix data to the census tract boundary file. The “+” button at the bottom of the “Join” tab adds a new join relationship. The following screen capture presents the parameter settings. The first two fields specify the imported distance matrix file and join field (key field) “ID” that was used to join the original census file to the distance matrix data. The third field is the key field from the census tract shapefile. Lastly, “Custom field name prefix” needs to be checked and the field should be cleared (QGIS tries to add a prefix!). It is desired to keep the original field names as they are. Within this file all fields are still unique and the original names were needed later when being used by a Python script. The join is now listed on the “Joins” tab.

The second screen capture below shows the updated attribute table that has resulted from the join process. To complete the process, the modified census tract shapefile was saved with a new name so that the original remains available for other applications.



The updated attribute table appears below.



The Huff Model

The Huff Model allows questions to be asked about market areas. The formula considers a consumer at a location “i” and a centre (often a shopping centre) at location “j”. The consumer is often represented as the population of a region such as a census tract. The centre is usually thought of as a shopping centre but could also be a city, attraction, or other type of destination. The formula is concerned with identifying the probability that consumer i will shop or travel to centre j. Influencing the formula are the distance “d” between the consumer and the centre and some measure of utility associated with the centre. Utility usually takes the form of size “s” and may measure square feet, number of employees, number of stores, or number of courses. Temperature could also be used – warmer weather attracts tourists. As with other gravity models, a greater size will draw greater attention from consumer i.

The Huff Model from the Consumer's Perspective

$$P_{ij} = \frac{\frac{S_j}{d_{ij}^b}}{\sum_j \left(\frac{S_j}{d_{ij}^b} \right)}$$

The Huff Model is typically viewed from the perspective of the consumer. The question may be asked: “What is the probabilty that consumer i will shop at centre j?”. To answer this question, a study area is considered and a probability that consumer i will shop at any centre j in the study area is calculated. If there are five centres, then five probabilities will be calculated for consumer i and the sum of the five probabilites will equal 1.

When considering a map that would present the study area of the consumers (census tracts) and centres (shopping centres), only one centre's set of probabilities would be displayed. If there are five centres, the map would show each census tract's probability for just the one shopping centre that is being considered. If a map of probabilities is produced for each of the five centres, then the five values for a census tract could be read across the five maps and they will sum to 1. In contrast, the sum of the values for all the consumers (census tracts) on the individual map will not equal 1.

The consumer perspective of the Huff Model is defined by the j under the Sigma in the formula above. If a grid is established to calculate the probabilities, consumers (census tracts) can be put into rows and centres (shopping centres) can be put into columns. The j under the Sigma indicates that the denominator of the Huff Model formula will only include the sum of the values for one row's (consumer) data. If the denominator represents all of one consumer's spending or attention, then the numerator represents that portion assigned to a given centre j.

The consumer perspective is important when considering the possible applications. It is often desirable to understand how consumers allocate their resources to the market. This perspective of the Huff Model allows for the likely distribution of consumer resources (disposable income, time, etc.) to be predicted. This estimate may be compared with reality and performance by a given centre may be questioned. If the centre is attracting less than their expected share of consumer resources, then action may be required. If the centre is attracting more than their share of resources, then they are likely doing well.

“What if” scenarios may also be completed. For example, what would be the impact of adding or removing or changing a centre in the study area? These types of experiments can be completed with relative ease.

A Python Script for the Huff Model (RU_Spatial_-_Huff_model.py)

A Python script has been written for the Huff Model as presented above. The code for the script is presented below.

```
##Consumer_Layer_with_Distance_Matrix=vector
##Consumer_Layer_ID_Field=field Consumer_Layer_with_Distance_Matrix
##Centre_Layer=vector
##Centre_Layer_ID_Field=field Centre_Layer
##Centre_Layer_Size_Field=field Centre_Layer
##Output_Layer=output file

# Script: RU Spatial - Huff Model
# Author: Michael Morrish
# Date: December 9, 2016
#
# This script takes in two input shapefiles and three field specifications and
# produces Huff model probabilities.

# Imports.
from qgis.core import *
from PyQt4.QtCore import *

# Get the layers.
lyrConsumer = processing.getObject(Consumer_Layer_with_Distance_Matrix)
lyrCentre = processing.getObject(Centre_Layer)

# Get the fields.
fldConsumerID_index = lyrConsumer.fieldNameIndex(Consumer_Layer_ID_Field)
fldCentreID_index = lyrCentre.fieldNameIndex(Centre_Layer_ID_Field)
fldCentreSize_index = lyrCentre.fieldNameIndex(Centre_Layer_Size_Field)

# Need to prepare output layer and add new field.
# New field is "Hi" plus the ID of the Centre.
# Loop through each Centre to construct new field names.
lyrOutput = processing.getObject(Output_Layer)
provider = lyrOutput.dataProvider()

for centreFeature in lyrCentre.getFeatures():

    # Capture value of fldCentreID_index (current ID).
    currentCentreID = centreFeature[fldCentreID_index]

    # Add and name the field.
    new_field_name = 'Hi' + currentCentreID
    provider.addAttributes([QgsField(new_field_name, QVariant.Double)])
    lyrOutput.updateFields()

# Loop through each Consumer feature.
for consumerFeature in lyrConsumer.getFeatures():
```

```
# Capture value of fldConsumerID_index (current ID).
currentConsumerID = consumerFeature[fldConsumerID_index]

# Create a total field for the sumJ of Sj/dij values for use in the nested loop.
sumJ_sjdivdij = 0.0

# Huff Formula: [(sj/dij)/(SUMj(sj/dij))] for a given consumer i and centre j.
# sumJ_sjdivdij is the denominator of this formula and is first loop below.
# Second loop below calculates numerator and completes Huff calc for a given ij.

# Loop through each Centre feature to calculate a consumer's sumJ_sjdivdij.
# This is the Huff formula denominator.
for centreFeature in lyrCentre.getFeatures():

    # Capture value of fldCentreID_index (current ID).
    currentCentreID = centreFeature[fldCentreID_index]

    # Capture value of fldCentreSize_index (current Centre Size).
    currentCentreSize = centreFeature[fldCentreSize_index]

    # Capture distance value for this Centre and this Consumer.
    # currentCentreID should match to field name in attrib table.
    currentDistance = consumerFeature[currentCentreID]

    # Calculate Centre Size / Distance (Sj/dij)
    sjdivdij = currentCentreSize / currentDistance

    # Add new Sj/dij to sumJ_sjdivdij.
    sumJ_sjdivdij = sumJ_sjdivdij + sjdivdij

# Loop through each Centre a second time to calculate Huff proportion.
for centreFeature in lyrCentre.getFeatures():

    # Capture value of fldCentreID_index (current ID).
    currentCentreID = centreFeature[fldCentreID_index]

    # Capture value of fldCentreSize_index (current Centre Size).
    currentCentreSize = centreFeature[fldCentreSize_index]

    # Capture distance value for this Centre and this Consumer.
    # currentCentreID should match to field name in attrib table.
    currentDistance = consumerFeature[currentCentreID]

    # Calculate Centre Size / Distance (Sj/dij)
    sjdivdij = currentCentreSize / currentDistance

    # Complete the Huff formula calculation.
    calcHuffI = sjdivdij / sumJ_sjdivdij

    # Set the value of the new Hi field for the current Consumer and Centre.
    lyrOutput.startEditing()
    current_Huff_field = 'Hi' + currentCentreID
    Huff_field_index = lyrOutput.fieldNameIndex(current_Huff_field)
    lyrOutput.changeAttributeValue(consumerFeature.id(), Huff_field_index,calcHuffI)
```

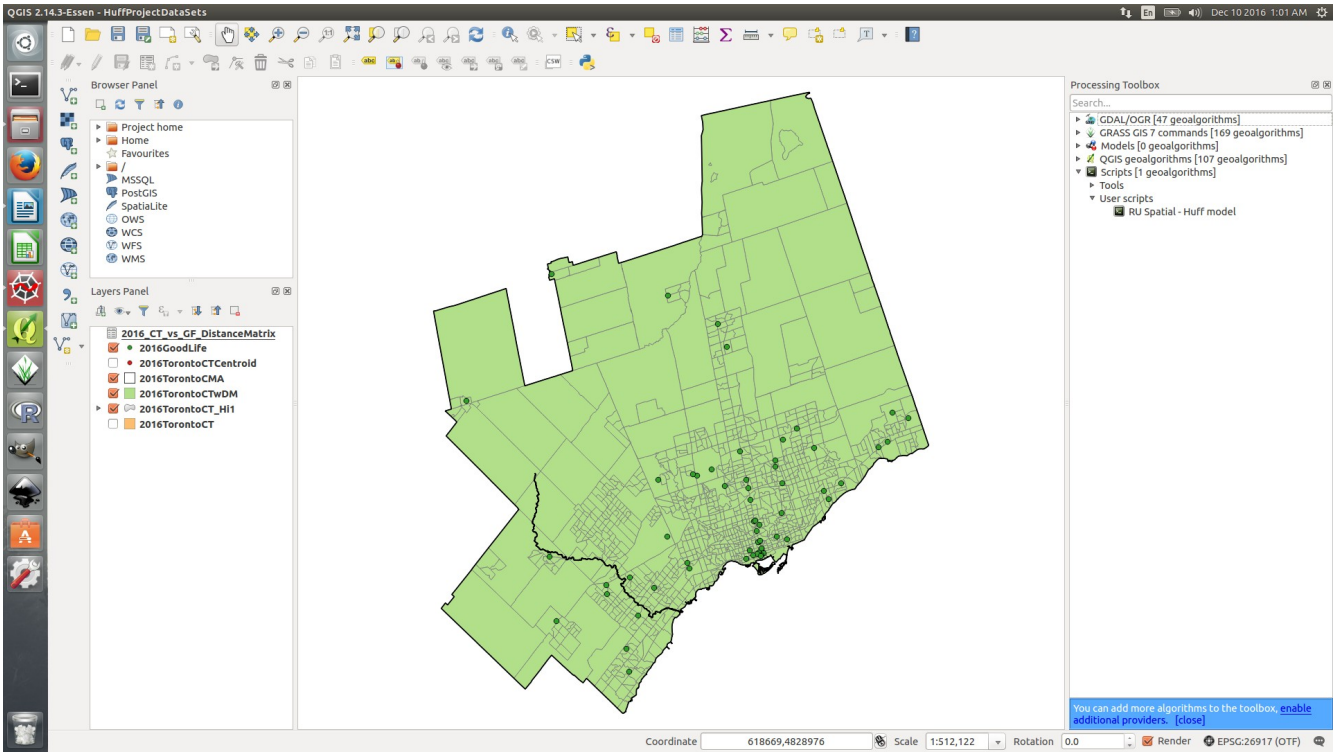
```
lyrOutput.commitChanges()
```

Step 11: Using the Huff Model Python Script in QGIS

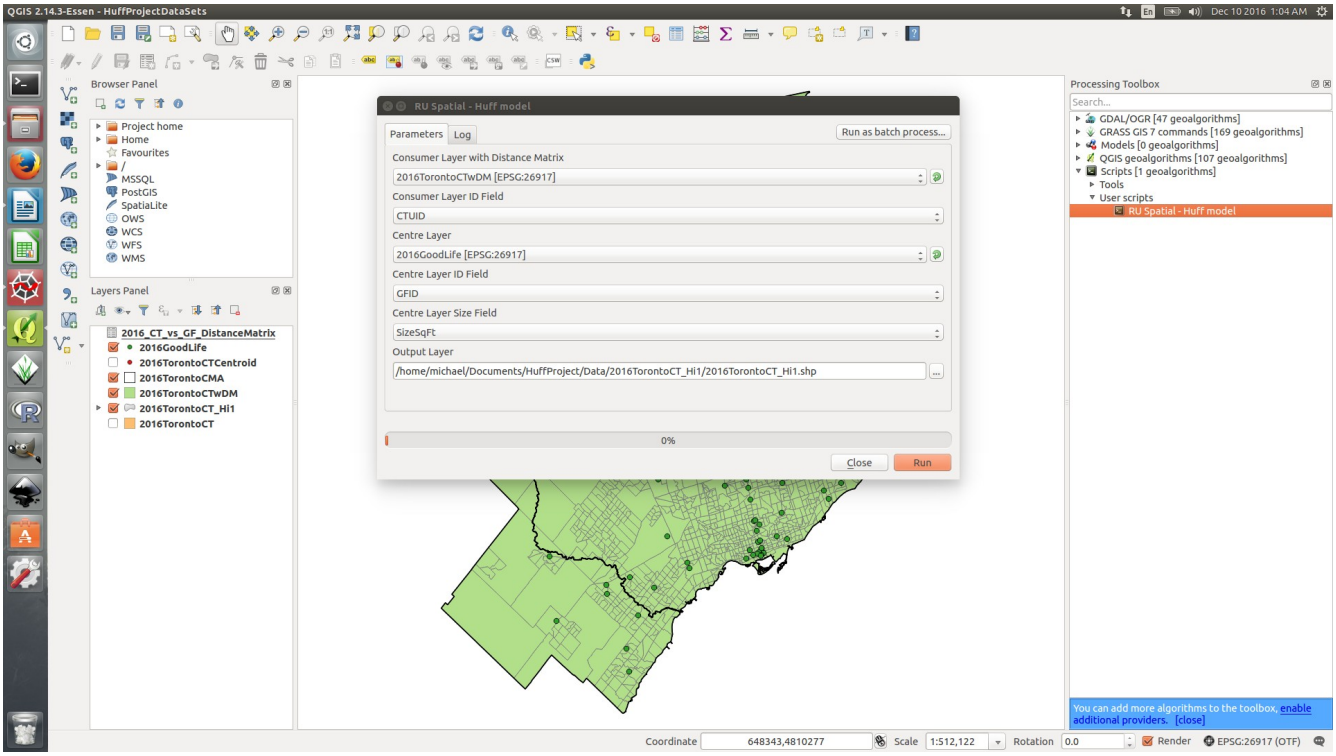
Using the Huff Model Python script in QGIS is straightforward. The output of the first ten steps above provides for most of the requirements to make use of the script. A few additonal steps are also required and these are discussed below.

An output file is needed for the script. The output file is actually an existing file that is modified by the script – fields are added to it along with new data. It is the user's choice as to what they provide for this file. The last shapefile used in Step 10 will be an input file for the script. This has both the census tract data from Statistics Canada and the distance matrix data joined to it. This may be used as the output file to keep everything in one place. It should be backed up before use, if that is the choice made. Alternatively, the clutter of additional distance matrix data may be unwanted. A fresh copy of the original census tract boundary file could also be used and this would be close to the minimum required – the CTUID field is needed for this case study. A fresh copy of the empty boundary files have been saved with a new name for use here.

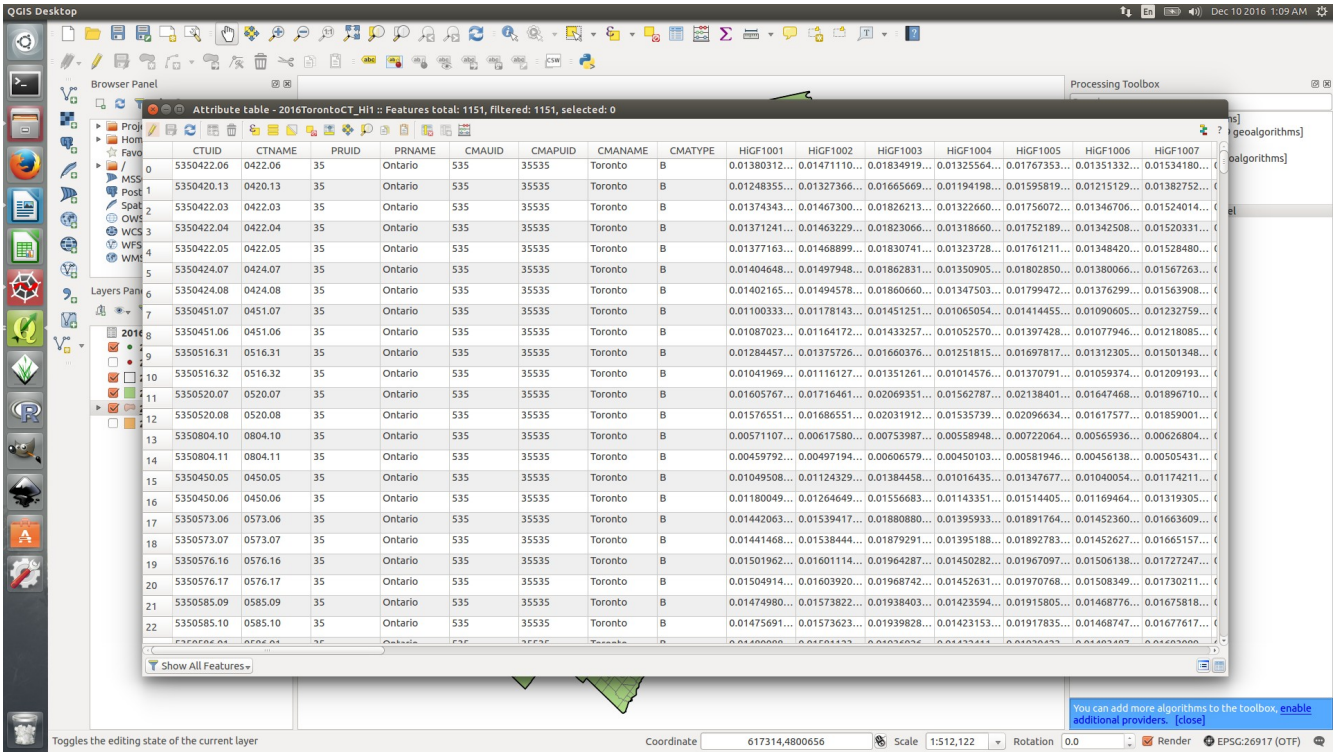
To make use of the Python script, it also needs to be added to the appropriate folder on the user's computer so that it is recognized by QGIS. Once there, it can be added to a toolbox (and may be recognized automatically) in the Processing Toolbox panel of QGIS. To execute the script, right-click on it and click “execute”. See the screen capture below.



Once executed, the script starts off by requesting several parameters to be specified by the user. See the next screen capture below.

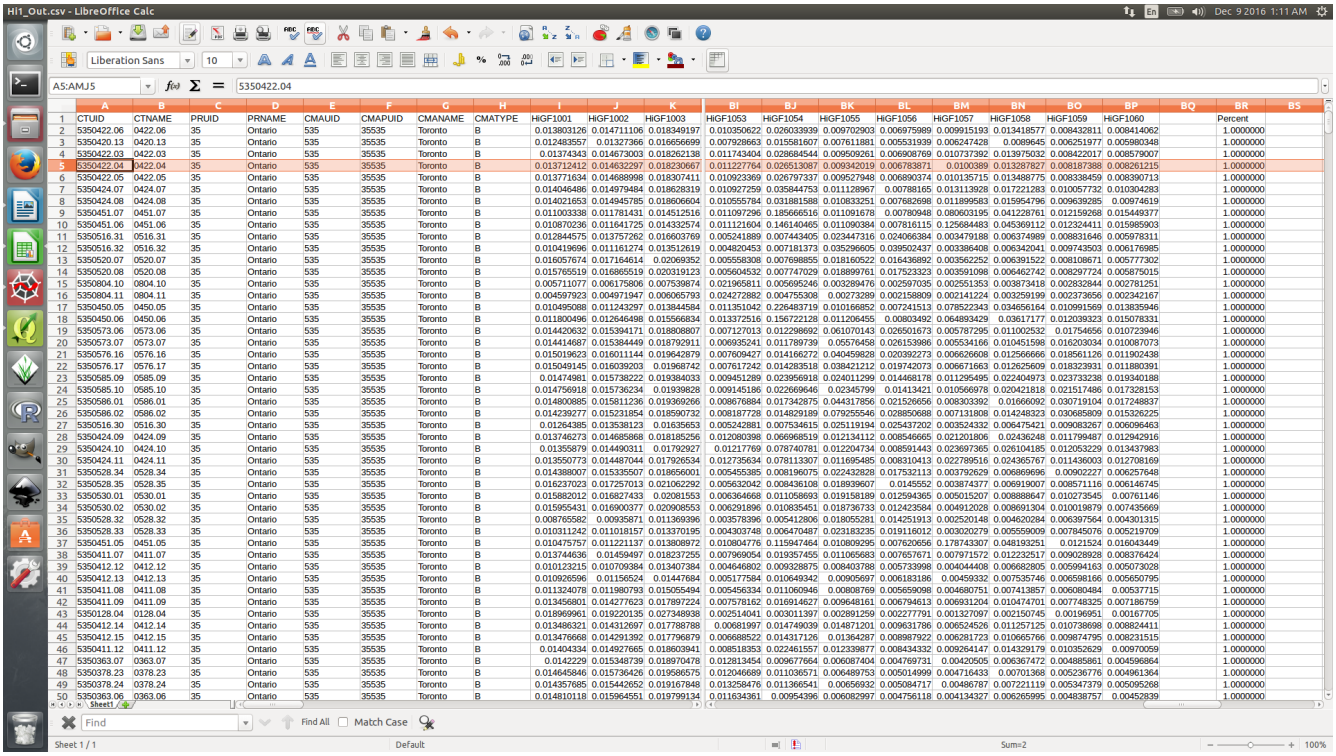


The first request is for a shapefile that contains the consumer information along with the distances between each consumer and each centre. The shapefile created in Step 10 with the distance matrix attached provides for this need. Second, the ID field for the consumer layer needs to be specified. In this case, it is “CTUID”. The next request is for the point layer for the centres. A shapefile with the Goodlife Fitness locations along with the size parameter is specified. This file was created in Steps 1-4 and 7. Two parameters are specified next to identify the ID and Size fields for the specified centre layer. It should be noted that if more than one size varable is available, the desired one could be selected here and the script could be run multiple times to explore different cases. Lastly, the script needs to know where to put the results. The clean file prepared above has been specified. Different output files can be prepared and specfied for different cases. Running the script for this case study takes a minute or two and creates sixty new fields and writes sixty-thousand pieces of data to the file. The attributes table for the results file is presented in the screen capture below.



Each of the sixty new fields have been added. The prefix “Hi” is used in combination with the existing field names (which are the centre IDs). The “H” is for “Huff” and the “i” is for “consumer”. Caution needs to be taken with the choice of field names and their length. Shapefiles have limits on their table field names of less than 10 characters. With two for the prefix, this leaves 8 characters for the original name. The field names are used as keys by the script and it is important that they remain as expected (and not abbreviated by QGIS to correct for length limitations). This is more likely to be an issue with identifiers in the data for centres as these are the ones that end up as field names.

To review some of the characteristics of the data output for the script, an export file has been created and a new screen shot of a spreadsheet is presented below.

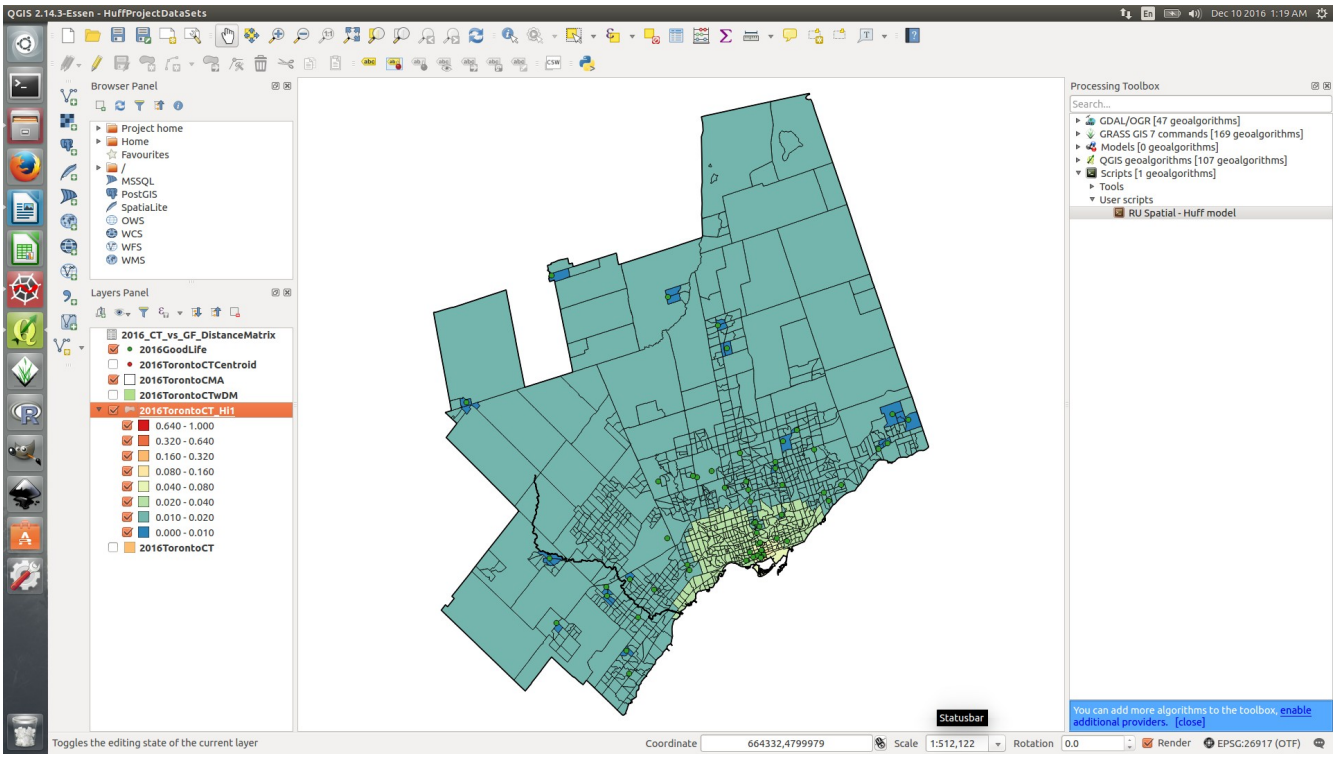


As discussed earlier, when considering the probabilties generated for the Huff Model, it is expected that the sum of the probabilities for a consumer shopping at all centres will sum to 1. Adding a totals column to the spreadsheet and summing the probabilities of a census tract across centres shows that this is true and that the script is performing as expected.

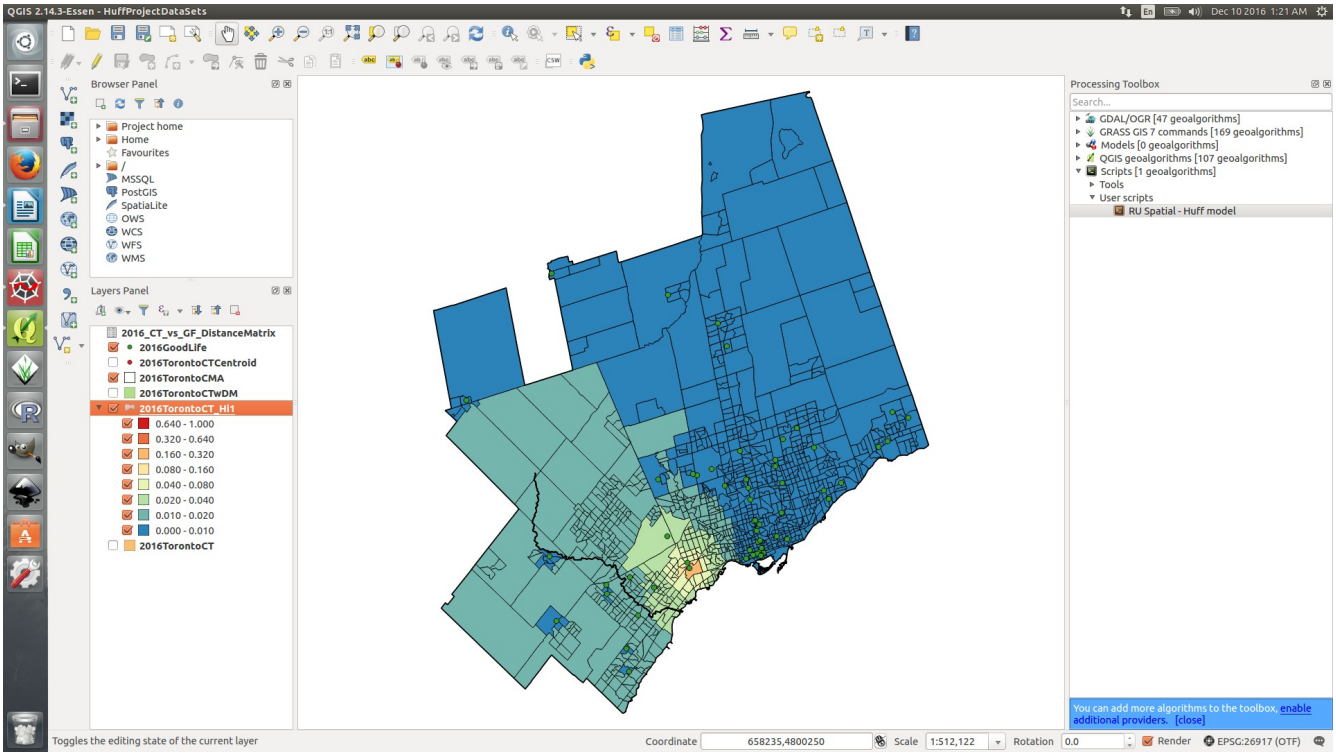
Visualizing Huff Model Output with Maps

A sample of maps for the Huff Model probabilities produced are presented below in screen captures. Each map uses the same scale to ensure each is interpreted the same way. Eight classes are used with blues covering the lower values and reds covering the higher values. Having a consistent scale for all sixty maps is a challenge as the probability values presented cover such a wide range of values and the detail can easily be lost if not enough classes are used. To keep the number of classes managable, a non-linear scale has been applied. Two to the power of n divided by 100 has been used to set the upper bounds of each class. That is: 0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 100 (100 should be 128, but the highest possible value for a probability is 100). This scale seems to work for this data. See the screen captures below for centres identified by GF1002, GF1027, GF1035, GF1051, and GF1054 respectively.

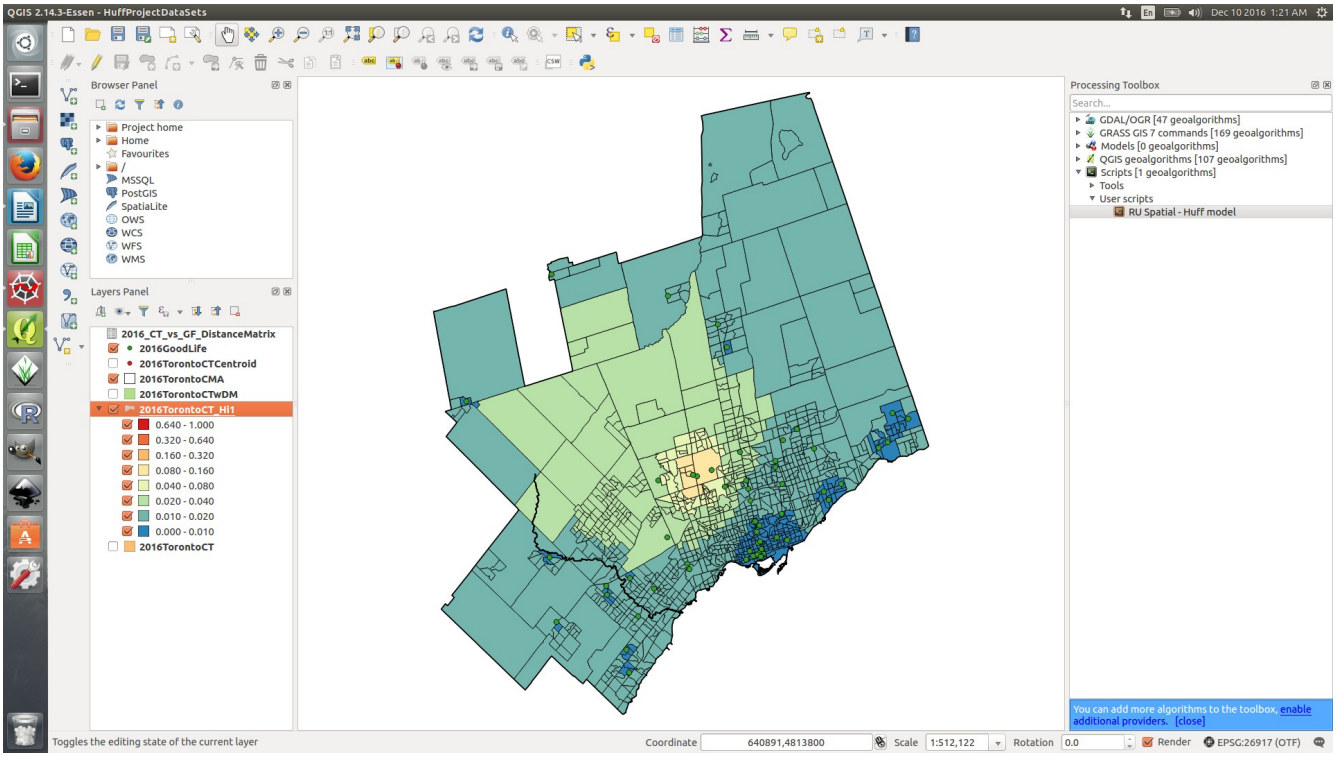
GF1002:



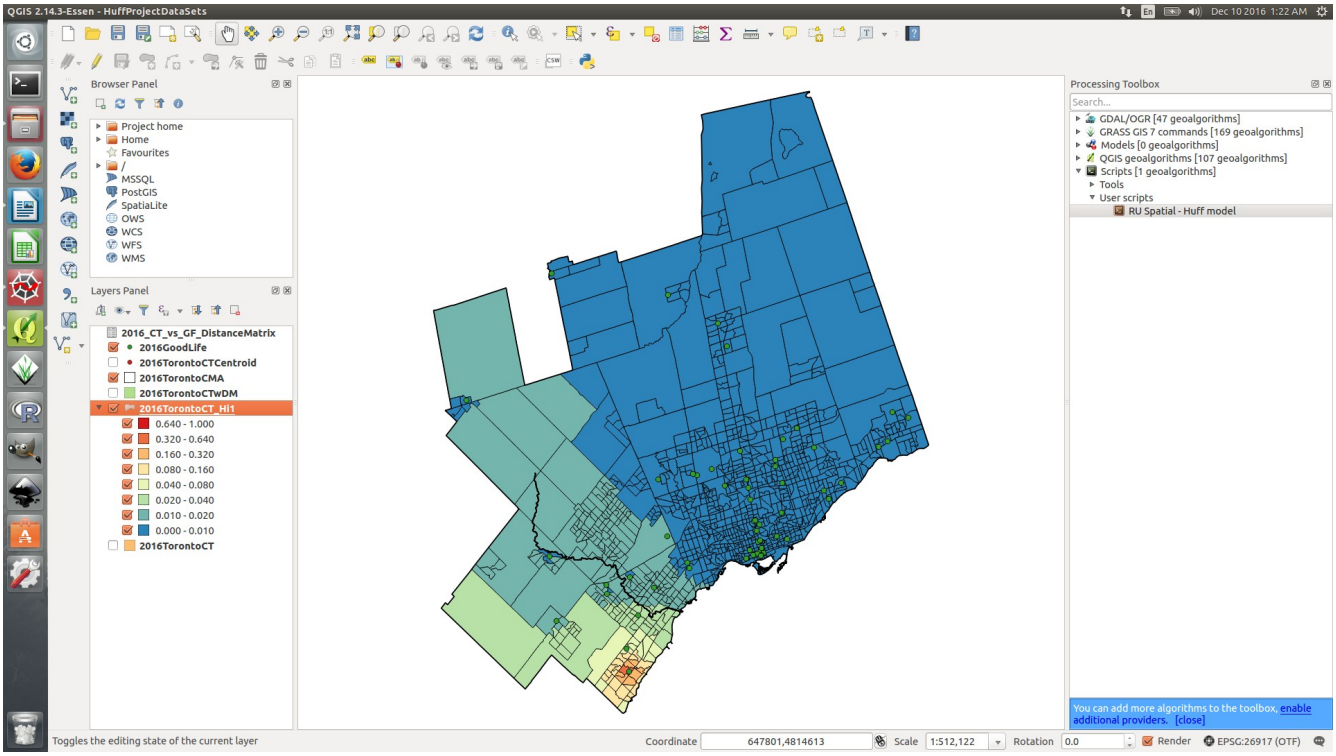
GF1027:



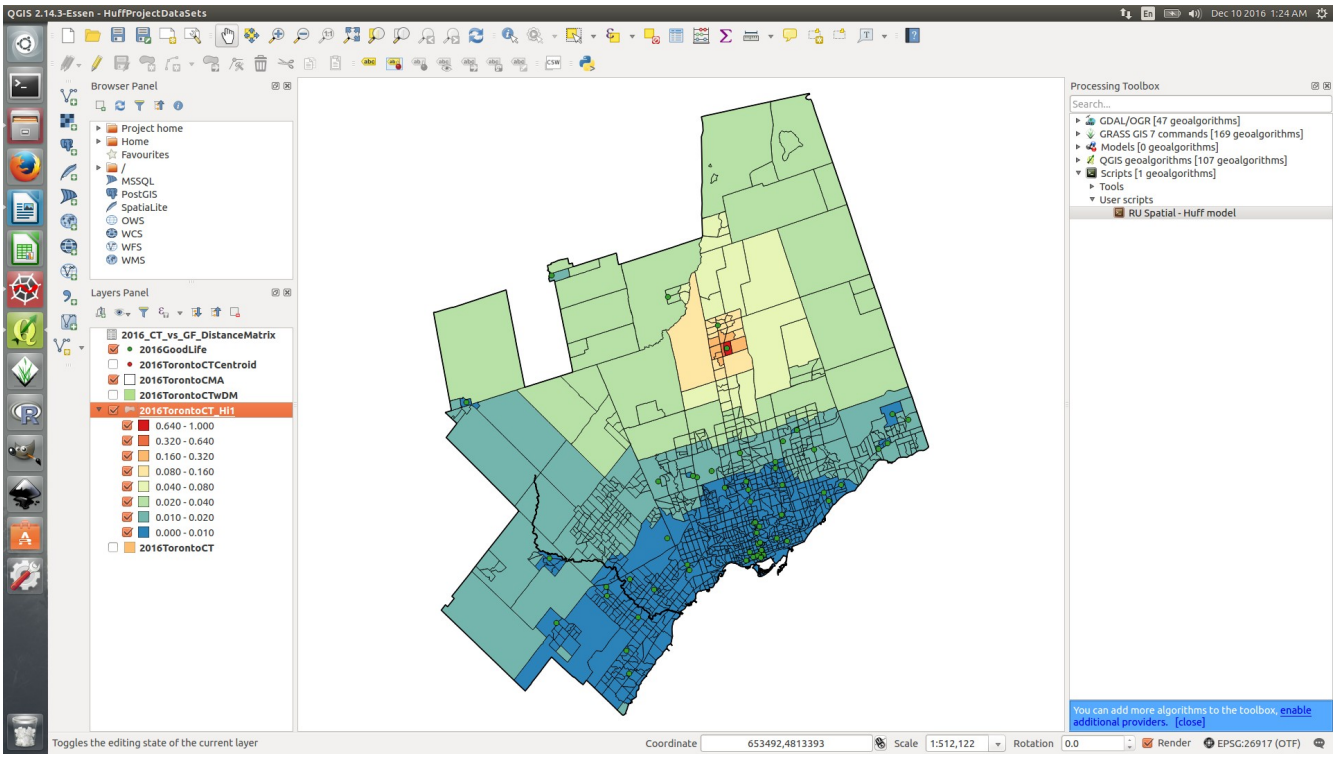
GF1035:



GF1051:



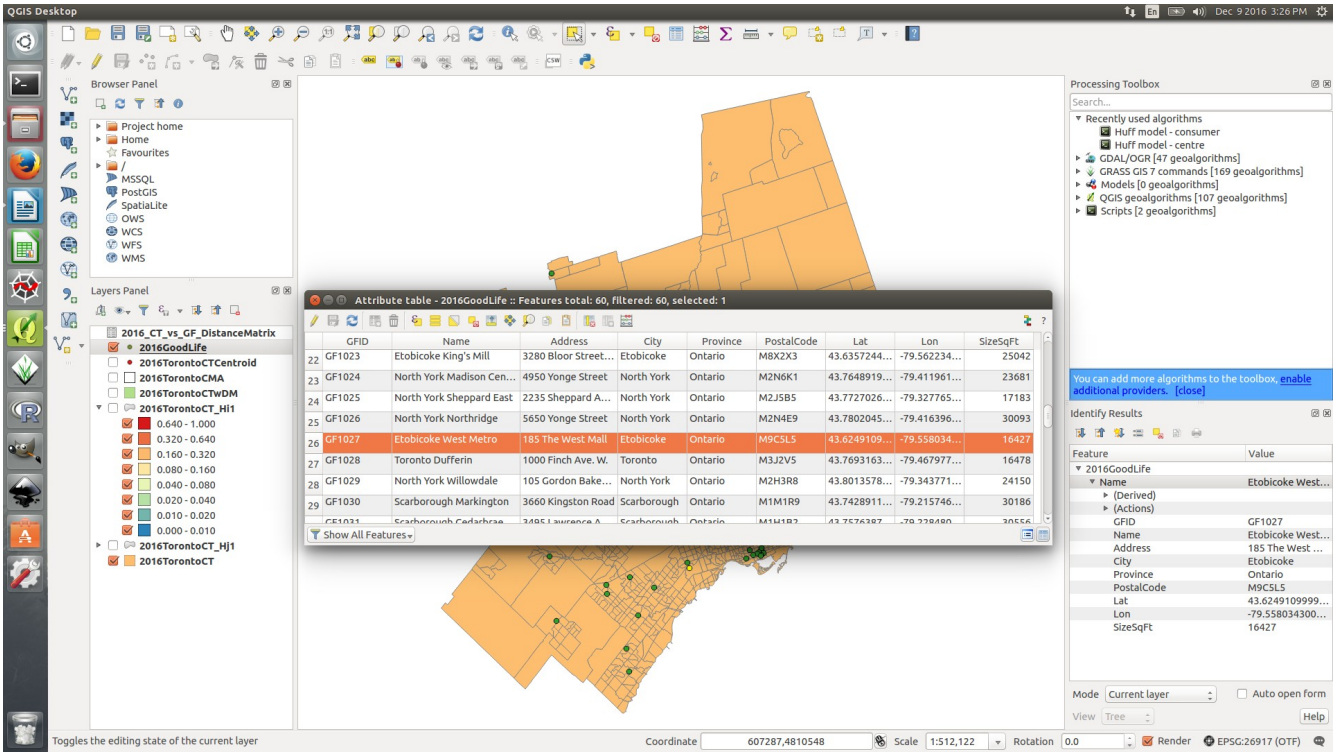
GF1054:



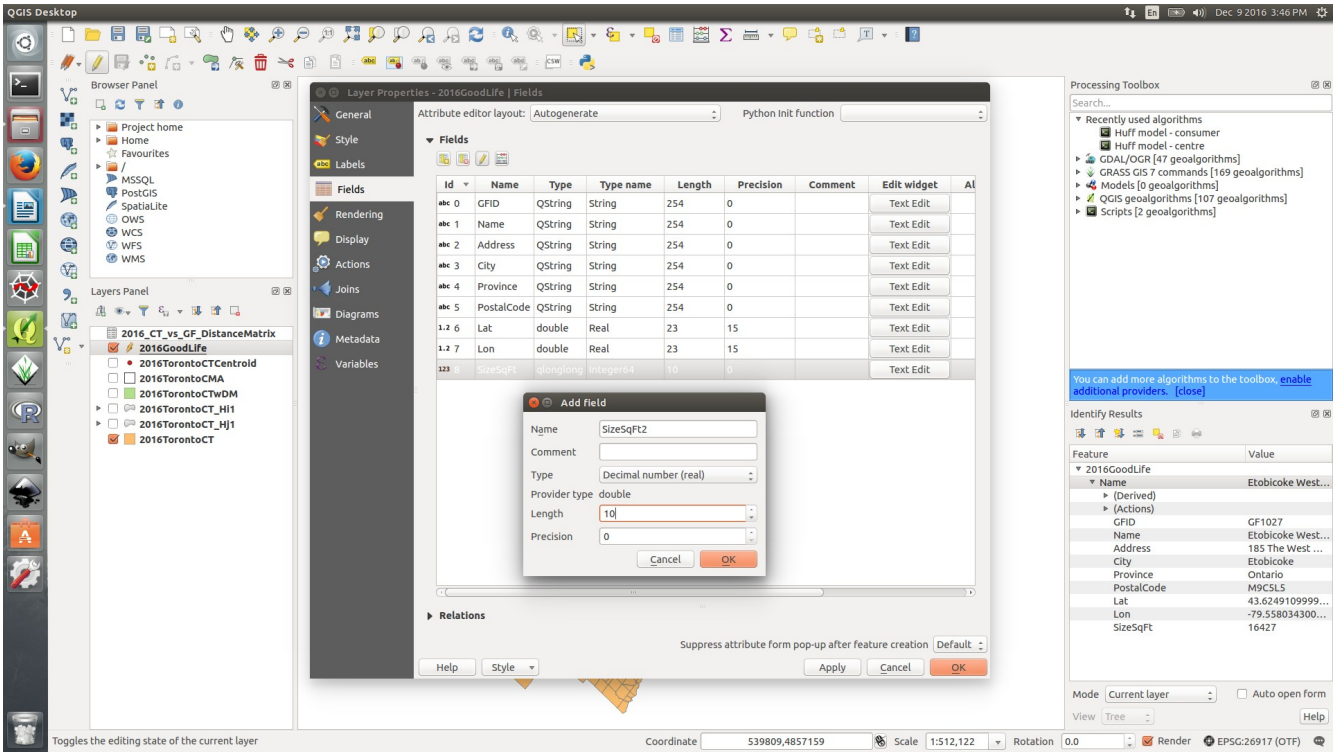
Step 12: Adding Utility Variables for "What-if" Scenario Exploration

Let's assume that Goodlife Fitness has decided to invest in the Etobicoke West Metro club. This is GF1027 (see the maps above) and is shown higlighted in yellow in the next screen capture. Head Office wants to know what the impact will be on their other locations if they convert this location into a 250,000 sqft mega club. This is explored with the new Python script in QGIS.

To consider this problem, not much more is required of the data that is aready in place. The consumer shapefile prepared in Step 10 does not need to change as it holds the census tract boundaries and the distance matrix. With all the clubs remaining in their existing locations, this file is still valid. If new clubs are opened or existing clubs are closed, then this file would need to be recreated to reflect the change. The second input for the Python scripts, the centres shapefile, required editing; this is the data for the Goodlife Fitness locations. As the original locations are being considered in place, this file can be modified from the current form with the editor in QGIS. If locations need to be added, removed, or relocated; then the file could either be recreated from a .csv file or edited with "heads-up" digitizing techniques. To complete the "what if", a second size column has been added to the file with the same values for all clubs except for GF1027. This location has been set to a new value of 250,000. This editing process is reviewed below.

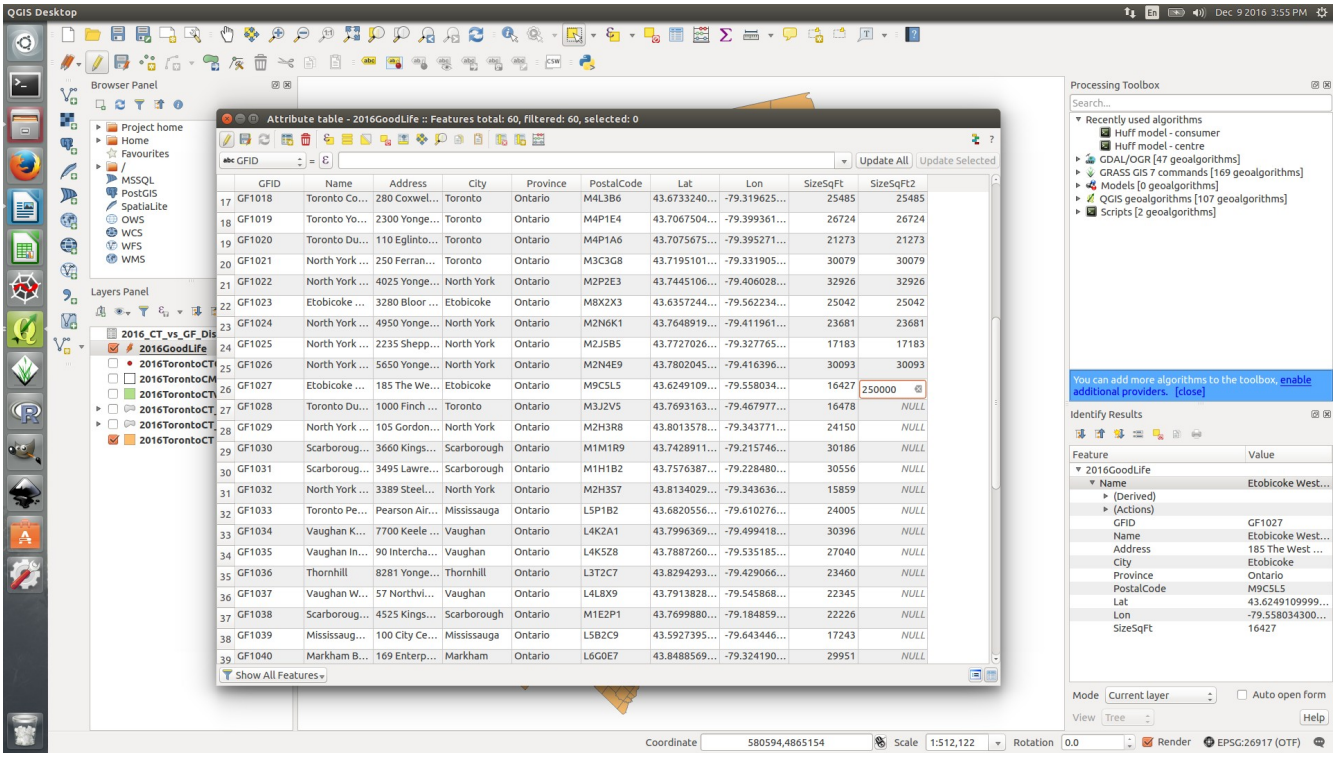


In the layer properties for the Goodlife Fitness locations, a new field is added on the “Fields” tab. See the next screen shot. The field type and length are matched to the exiting SizeSqFt field.



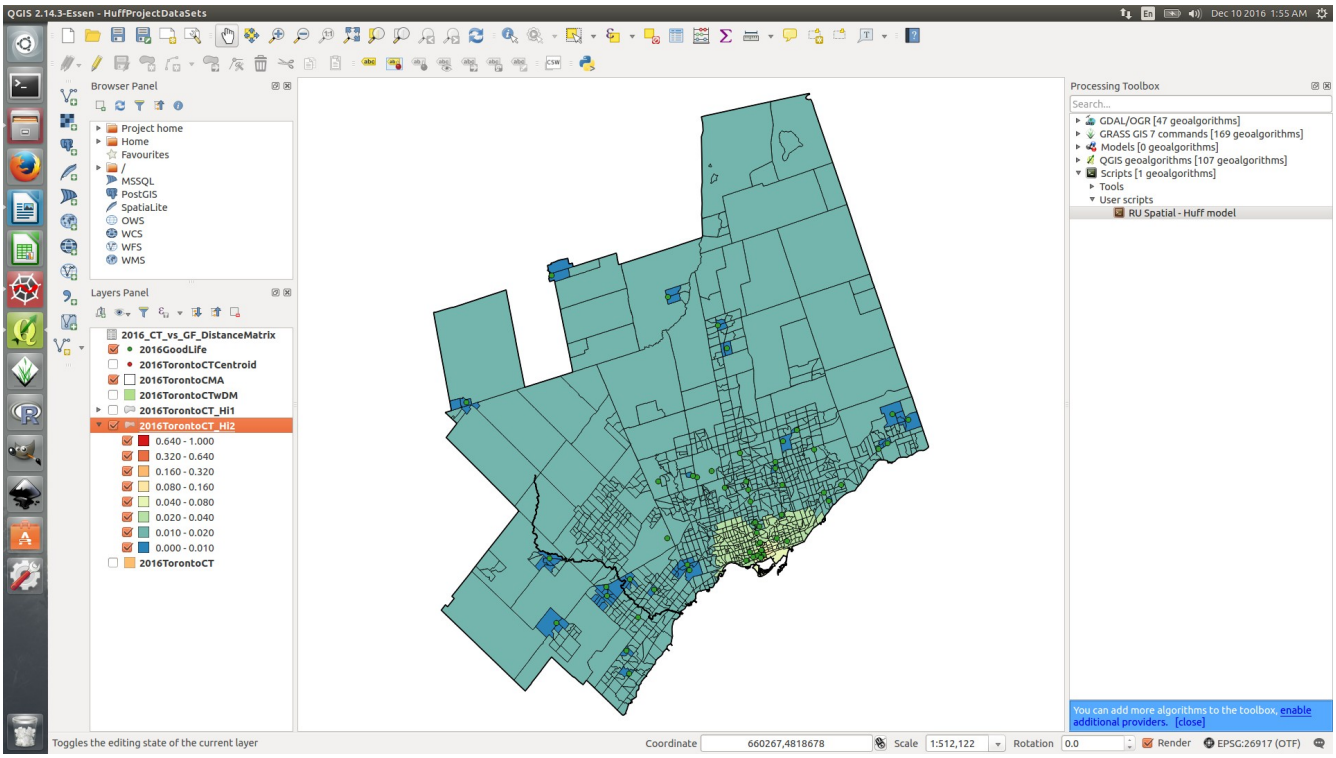
Using the editor in the attribute table, the values in the original field are re-keyed into the new column.

There are likely other ways to do this, but for 60 values, this is probably quicker. The next screen capture shows the editing process.

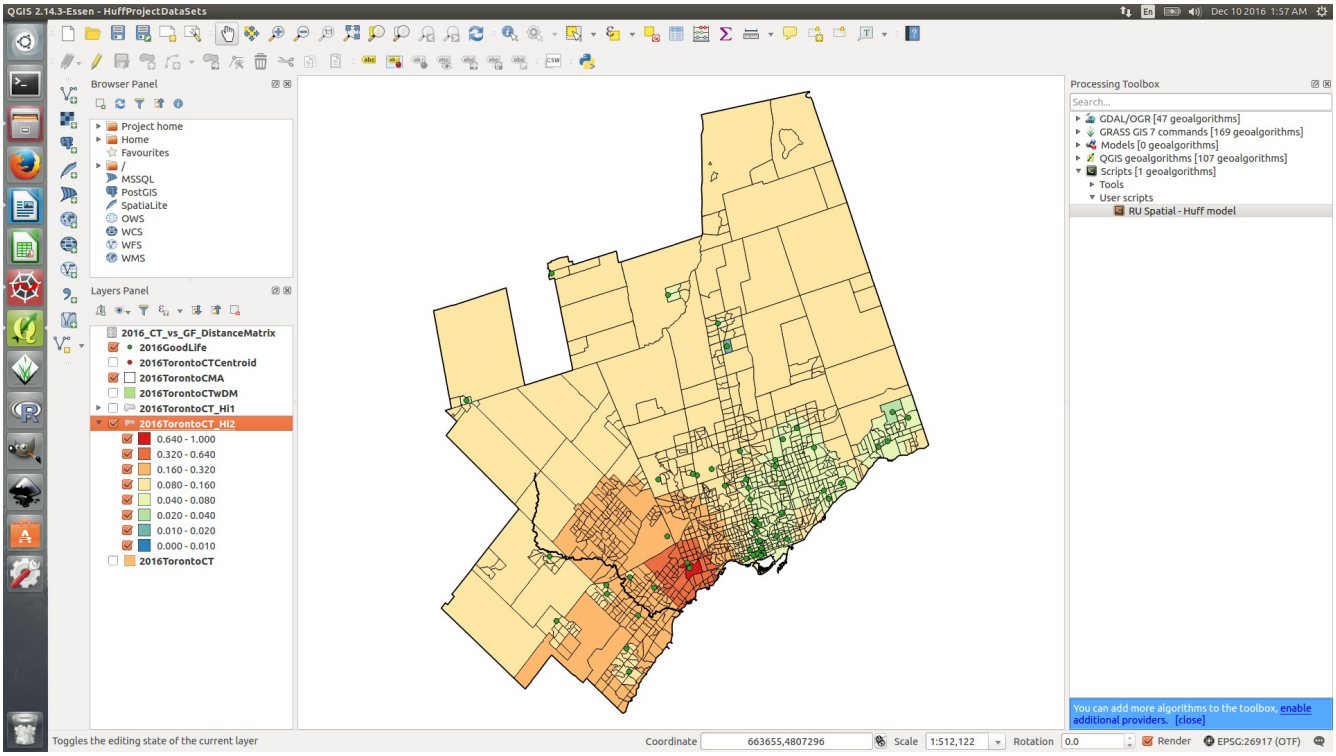


Running the script again with the second size variable and outputing the results to a new output file has been completed. New visualizations have been produced for the same five centres (GF1002, GF1027, GF1035, GF1051, and GF1054). Screen captures start on the next page.

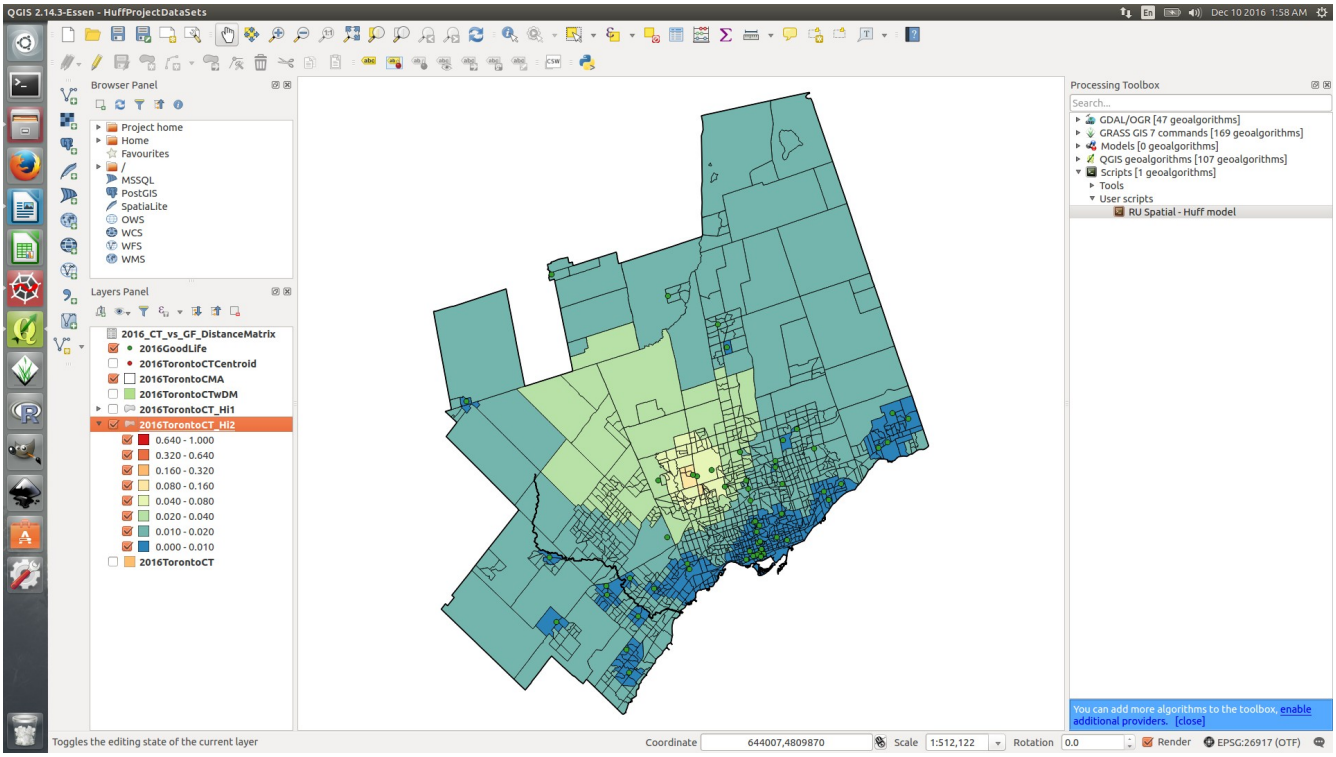
GF1002:



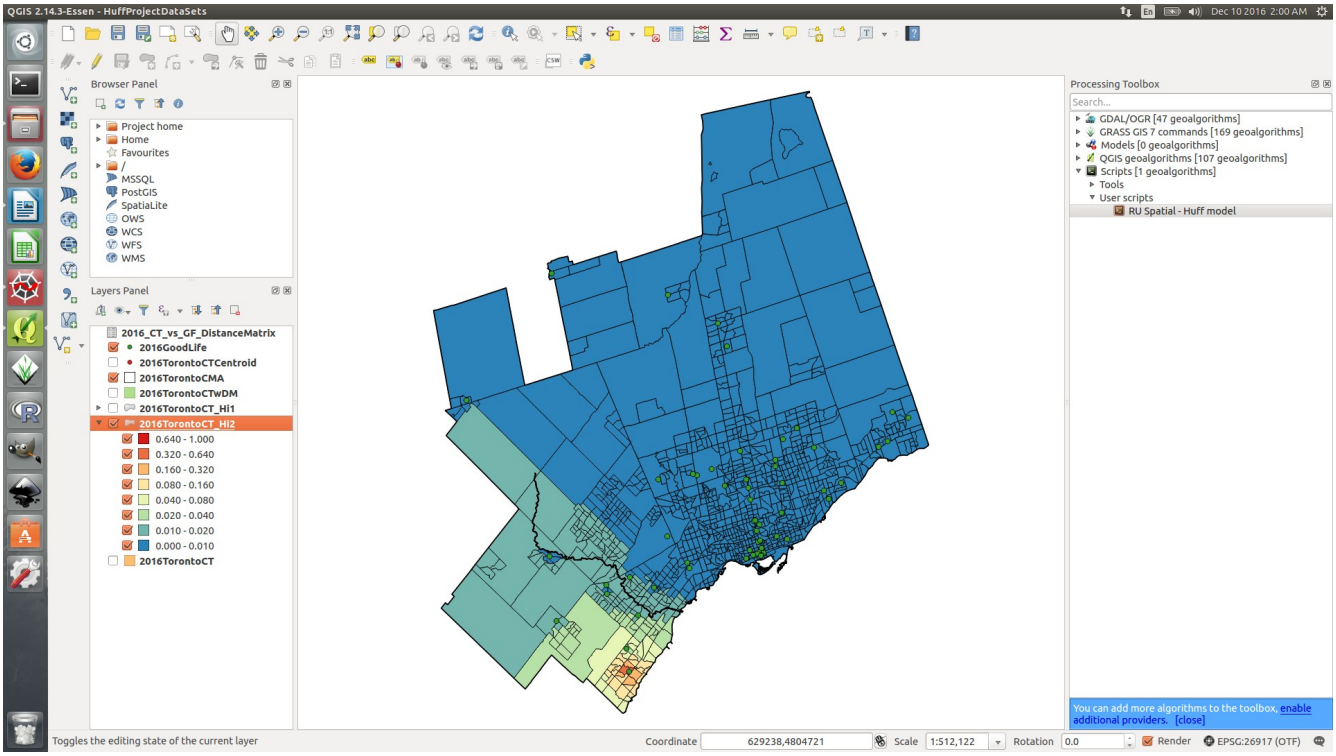
GF1027:



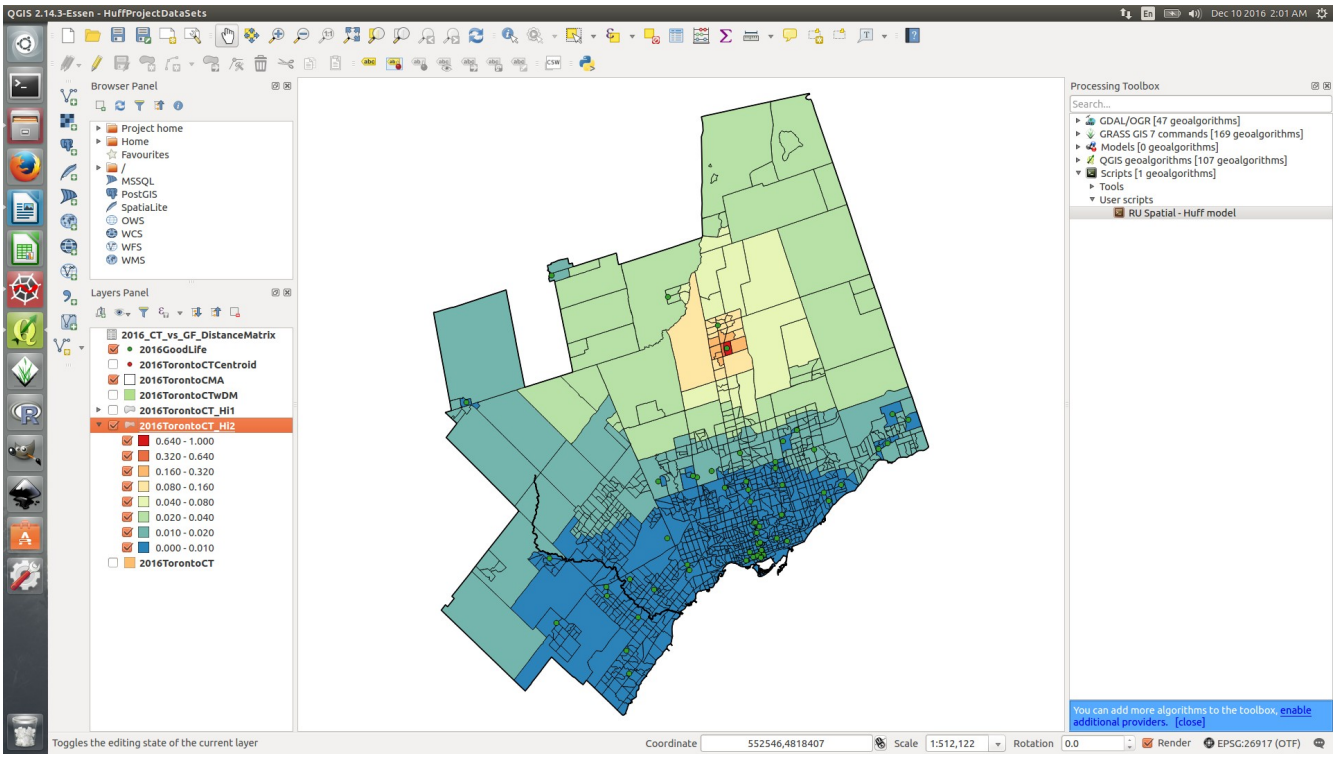
GF1035:



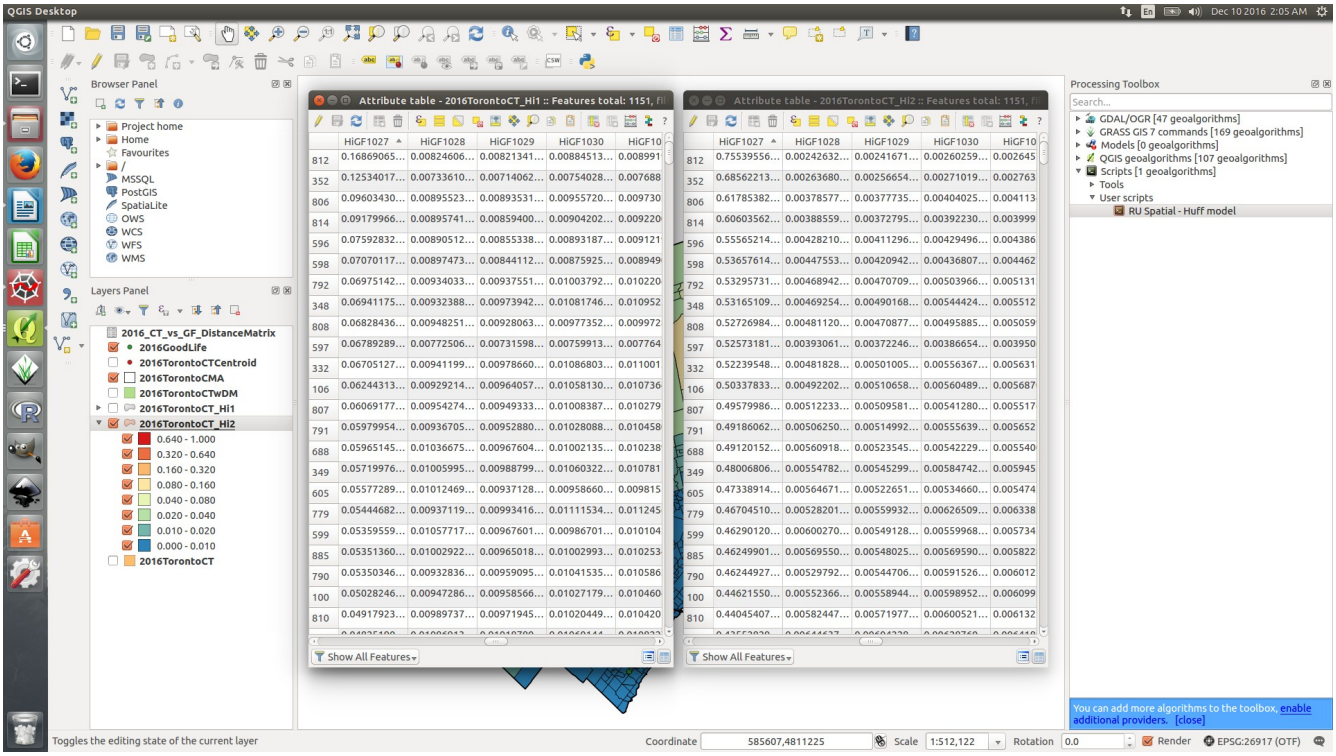
GF1051:



GF1054:



Attribute table values for GF1027:



As can be observed by reviewing the maps generated with the new size data, the mega club (GF1027) will have an impact on both its own status in the market and also on those clubs that surround it. The last screen capture above presents the attribute tables for GF1027 (see the left column in each table). The table on the left is for the original “Size” variable and the one on the right reflects the new 250,000 sqft facility. Scanning the numbers and reading down the GF1027 columns, it can be seen that this location has significantly strengthened it's position in the market. Reading across the rows, and comparing the two tables, it is observed that the improvement in GF1027 is at the expense of the other locations – it is harder for them to attract those close to GF1027 to their own locations. The probabilities for any given consumer (census tract) must sum to 1. If GF1027's share goes up, then the others must go down.

Conclusions

QGIS provides an adequate platform for conducting Huff Model calculations and experiments. The majority of the functionality required for this work is available in a standard installation of QGIS 2.14 Essen and with a standard spreadsheet application. The functionality to complete the Huff Model calcuations themselves are not built-in standard and these have been provided for with the use of a custom-written Python script. All of these capabilities and the operation of the new script have been demonstrated here.

It was initially suggested that a Python plugin be used for the task of providing capabilities related to the Huff Model. This approach was rejected for two reasons. Python plugins are a lot more complex and require the development of a graphical user interface (GUI) and also need to be compiled. There appears to be little in the way of documentation to assist navigating this more complex path to the desired solution. The second reason for the use of a script over a plugin is that it is better suited to the task. Huff Model calculations do not require the complexity of a plugin. The script may also be integrated into the Toolbox or Processing Models (like Esri's Model Builder) to create workflows or automated processes. This may be explored in the future. This is a starting point and the potential exists to develop many Toolbox items under the RU Spatial folder/brand. These may work together in a workflow or be used independly as Toolbox items.

The provided Python script is designed to provide capabilities based on Huff Model as presented in academic textbooks. There are other variations of the Huff Model including an Advanced Huff Model and implementations by Esri in Business Analyst. Results from this work appear to match textbook results. Esri's results seem to match something different.

This work should be considered in the development stage. While the developed script appears to work well, more comprehensive testing should be completed. The script should also be enhanced to “idiot proof” it as much as possible.

Moving forward, this work may be expanded to create additional scripts to provide functionality similar to those found in other implementations. Work may also be completed to simplify or enhance the process with workflow models or other algorithm components existing or to be developed. The review completed here has also focused on generating Huff Model results. Functionality, through the use of scripting, could also be produced to automated the use of the Huff Model results. This could take the form of visualization production or further downstream analysis.