Configuration Manual

MSc Research Project
Programme Name

Celine Moran Lee
Student ID: 19175248

School of Computing
National College of Ireland

Supervisor: Dr. Paul Stynes, Dr. Pramod Pathek

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Celine Moran Lee		
Student ID:	19175248		
Programme:	MSCDAD-A	**Year:**	2020/2021
Module:	Research Project		
Lecturer:	Dr. Paul Stynes & Dr. Pramod Pathak		
Submission Due Date:	23rd September 2021		
Project Title:	Configuration Manual		
Word Count:	2481 … **Page Count: 30**		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………………………………………………………………………………………………………

Date: 16th August 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Celine Moran Lee
Student ID: 19175248

1 Introduction

This configuration manual is to replicate the research from the report "A Novel Framework for Automated External Defibrillator Deployment (FAEDD) in Identified High Risk Residential Areas". Within the document is a description of the hardware and software used for the research; the list of data sets used within the research; the processing of data and implementation of the experiments. Where appropriate screenshots of code and output are included.

2 Hardware and Software

The hardware which all experiments were implemented on is listed in Table 1 and all the software used for the experiments is in Table 2.

Table 1: Hardware used for experiments

System	Specification
Processor	Intel(R) Core (TM) i5-4210U CPU @ 1.70GHz 2.40 GHz
(RAM)	16.0 GB (15.9 GB usable)
System type	64-bit Windows Operating System, x64-based processor

Table 2: Software Used with corresponding libraries

Software	Library	Version
SPSS	NA	27
GeoDa	NA	1.18
Microsoft Office	Excel, Word, Powerpoint	2107
R Studio	Tmap	2.3-1
	Spdep	1.1-8
	Maptools	0.9-8
	Leaflet	2.0.4.1
	Rgdal	1.4-7
	Rgeos	0.5-2
	Raster	3.0-7
	Dplyr	1.0.6
	Htmltools	0.4.0
	Ggplot2	3.2.1
	Tidyverse	1.3.0

SpatialEpi	1.2.3
Car	3.0-6
Ape	5.5
CARBayes	5.2.3
Shapefiles	0.7
Sp	1.3-1
Sf	0.8-0
Purrr	0.3-4
Matrixcalc	1.0-5
INLA	21.02.23
INLABMA	0.1-11
Geosphere	1.5-10
Reshape2	1.4.3
Maxcovr*	0.1.3.9200
GGally	1.5.0
RColorBrewer	1.1-2
Lattice	0.20-38
SemiPar	1.0-4.2
Mapview	2.10.0
Mapedit	0.6.0
Ggmap	3.0.0
Magrittr	1.5
GISTools	0.7-4
Shp2graph	0-5

3 Datasets

The data needed to rerun this experiment is zipped together within a folder and numbered in order of use. All the files within this folder must be kept together to run the experiments, see Figure 1. Where appropriate within the code where a CSV file is saved and used in another piece of software e.g., R Studio to SPSS, the output of SPSS will be saved within the folder.

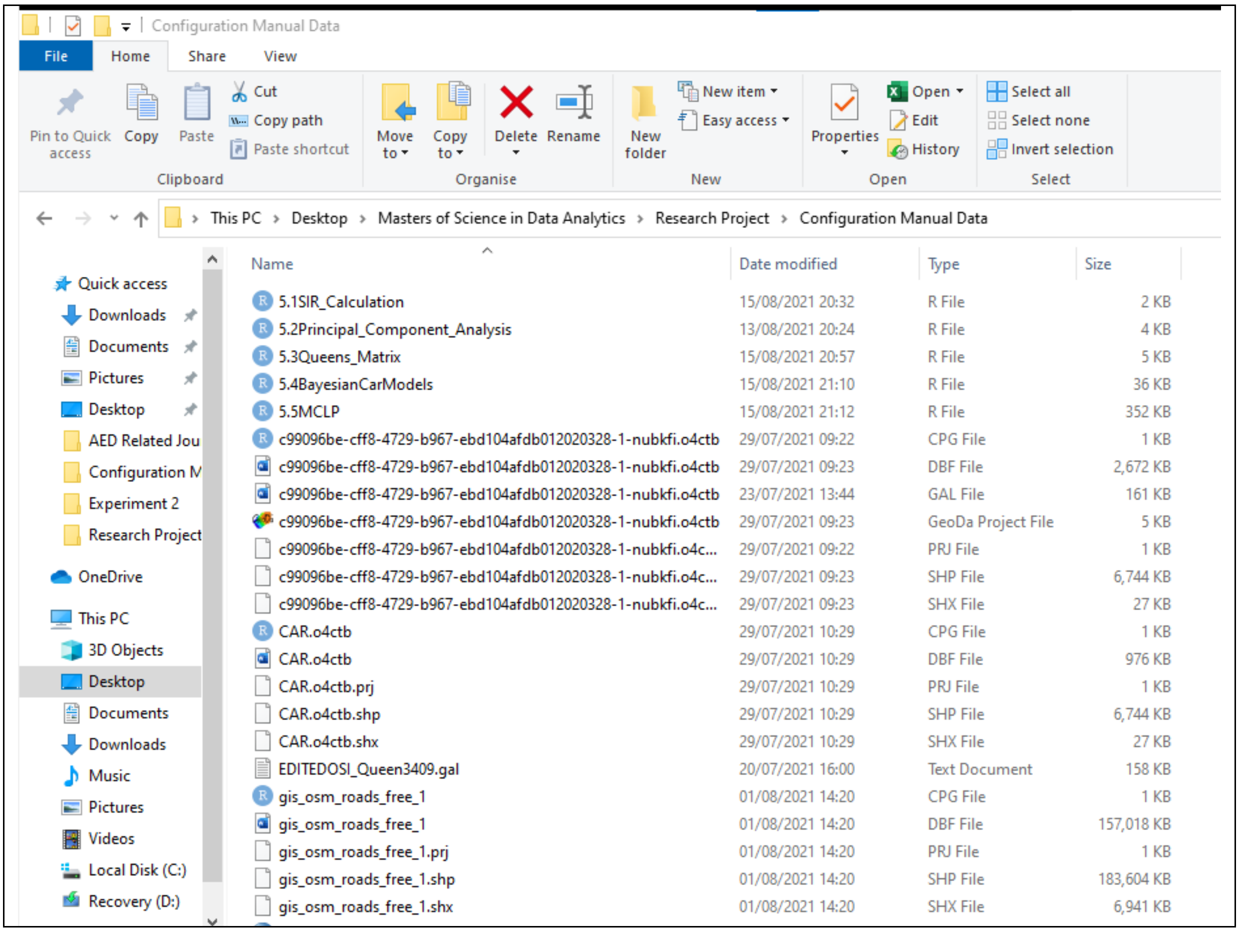


Figure 1: Image illustrating project files to run this code. Ensure these files are saved to directory.

4 Pre-processing Data

Upon downloading the data, the data was cleaned and organised either manually by saving into a CSV or through programming R. A description of each dataset is described below with specific actions to pre-process the data.

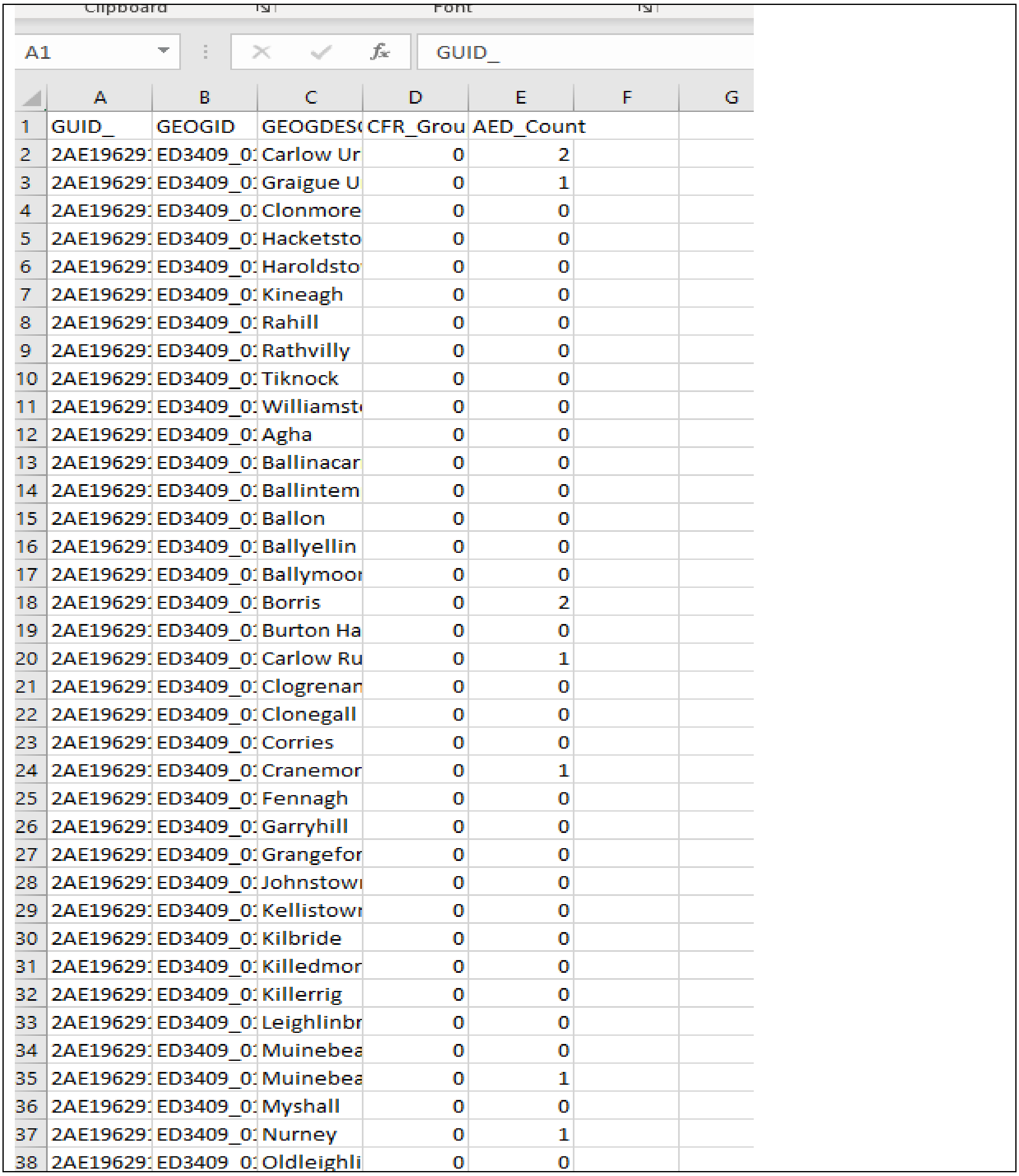
4.1 AED Data

Within the zipped data folder there are two datasets 1) the raw data as received by the HSE entitled "RAWAED CSO Small Areas Jan2021_SMORAN30MAR2021" see Figure 2 and 2) the structured labelled data entitled "AED_CFR_GRoup_Final.csv" see Figure 3. The original dataset has four columns with ID, Small Area Code, CSO Code and CSO Ed Name. This data had to be manually checked against the CSO GeoHive website https://census2016.geohive.ie/datasets/electoral-divisions-cso-generalised-20m/explore?location=53.082175%2C-6.355948%2C9.97&showTable=true to code each of the AEDs to the relevant electoral division. The new coded data was saved to the second file where the GUID, GEOGID and GEOGDESC codes for the electoral division were included.

B1363 | 217086001/217117001

	A	B	C	D	E
1	ID	Small Area	CSO	CSO ED Name	
2	SA1	17008003	1017	BORRIS	
3	SA2	17008004	1017	BORRIS	
4	SA3	17010042	1019	CARLOW RURAL (PT.)	
5	SA4	17011010	1001	CARLOW URBAN	
6	SA5	17011011	1001	CARLOW URBAN	
7	SA6	17017004	1023	CRANEMORE	
8	SA7	17034011	1034	MUINEBEAG URBAN	
9	SA8	17036002	1036	NURNEY	
10	SA9	17051002	1048	TULLOW URBAN	
11	SA10	17051004	1048	TULLOW URBAN	
12	SA11	17051005	1048	TULLOW URBAN	
13	SA12	17051008	1048	TULLOW URBAN	
14	SA13	17054003	1002	GRAIGUE URBAN	
15	SA14	27082003	32071	LARAH SOUTH	
16	SA15	27083003	32026	SWANLINBAR	
17	SA16	37010001	16066	BALLYNACALLY	
18	SA17	37012001	16049	BALLYSTEEN	
19	SA18	37016001	16140	CAHER	
20	SA19	37026005	16026	CLAREABBEY (PART IN R.D.)	
21	SA20	37027001	16027	CLENAGH	
22	SA21	37027002	16027	CLENAGH	
23	SA22	37042005	16019	CORROFIN	
24	SA23	37045003	16028	CRUSHEEN	
25	SA24	37051001	16010	DRUMCREEHY	
26	SA25	37051003	16010	DRUMCREEHY	
27	SA26	37057027	16032	ENNIS RURAL (PART IN R.D.)	
28	SA27	37057036	16032	ENNIS RURAL (PART IN R.D.)	
29	SA28	37057052	16032	ENNIS RURAL (PART IN R.D.)	
30	SA29	37057056	16032	ENNIS RURAL (PART IN R.D.)	
31	SA30	37057068	16032	ENNIS RURAL (PART IN R.D.)	
32	SA31	37058006	16002	ENNIS URBAN NO. 2	
33	SA32	37059013	16054	ENNISTIMON	
34	SA33	37077007	16088	KILKEE	
35	SA34	37077009	16088	KILKEE	
36	SA35	37079004	16071	KILLADYSERT	
37	SA36	37080003	16134	KILLALOE	

Figure 2: Raw data of AED locations to small area code.



	A	B	C	D	E	F	G
1	GUID_	GEOGID	GEOGDES(	CFR_Grou	AED_Count		
2	2AE19629:	ED3409_0:	Carlow Ur	0	2		
3	2AE19629:	ED3409_0:	Graigue U	0	1		
4	2AE19629:	ED3409_0:	Clonmore	0	0		
5	2AE19629:	ED3409_0:	Hacketsto	0	0		
6	2AE19629:	ED3409_0:	Haroldsto	0	0		
7	2AE19629:	ED3409_0:	Kineagh	0	0		
8	2AE19629:	ED3409_0:	Rahill	0	0		
9	2AE19629:	ED3409_0:	Rathvilly	0	0		
10	2AE19629:	ED3409_0:	Tiknock	0	0		
11	2AE19629:	ED3409_0:	Williamst	0	0		
12	2AE19629:	ED3409_0:	Agha	0	0		
13	2AE19629:	ED3409_0:	Ballinacar	0	0		
14	2AE19629:	ED3409_0:	Ballintem	0	0		
15	2AE19629:	ED3409_0:	Ballon	0	0		
16	2AE19629:	ED3409_0:	Ballyellin	0	0		
17	2AE19629:	ED3409_0:	Ballymoor	0	0		
18	2AE19629:	ED3409_0:	Borris	0	2		
19	2AE19629:	ED3409_0:	Burton Ha	0	0		
20	2AE19629:	ED3409_0:	Carlow Ru	0	1		
21	2AE19629:	ED3409_0:	Clogrenan	0	0		
22	2AE19629:	ED3409_0:	Clonegall	0	0		
23	2AE19629:	ED3409_0:	Corries	0	0		
24	2AE19629:	ED3409_0:	Cranemor	0	1		
25	2AE19629:	ED3409_0:	Fennagh	0	0		
26	2AE19629:	ED3409_0:	Garryhill	0	0		
27	2AE19629:	ED3409_0:	Grangefor	0	0		
28	2AE19629:	ED3409_0:	Johnstowı	0	0		
29	2AE19629:	ED3409_0:	Kellistowr	0	0		
30	2AE19629:	ED3409_0:	Kilbride	0	0		
31	2AE19629:	ED3409_0:	Killedmor	0	0		
32	2AE19629:	ED3409_0:	Killerrig	0	0		
33	2AE19629:	ED3409_0:	Leighlinbr	0	0		
34	2AE19629:	ED3409_0:	Muinebea	0	0		
35	2AE19629:	ED3409_0:	Muinebea	0	1		
36	2AE19629:	ED3409_0:	Myshall	0	0		
37	2AE19629:	ED3409_0:	Nurney	0	1		
38	2AE19629:	ED3409_0:	Oldleighli	0	0		

Figure 3: Manually tagged data to electoral division

4.2 CFR Data

Within the zipped data folder there are two datasets 1) the raw data as received by the HSE entitled "RAWCopy of CFRs JAN 2021" see Figure 4 and 2) the structured labelled data entitled "AED_CFR_GRoup_Final.csv" see Figure 3. This had to be manually checked against the CSO GeoHive website to code every electoral division to the correct AED location. This dataset required attention to detail as there was no corresponding CSO code to the census, so each name had to be checked manually against the GeoHive website.

	A	B	C
1		Kilcogy	Cavan
2		Ballyjamesduff	Cavan
3		Cavan	Cavan
4		Shercock	Cavan
5		Kildysart	Clare
6		Rathcormac	Cork
7		Carrigtwohill	Cork
8		Duhallow	Cork
9		Liscarroll	Cork
0		Charleville	Cork
1		Buttevant	Cork
2		Doneraile	Cork
3	ID	Mallow	Cork
4	CFR1	Blarney	Cork
5	CFR2	Ballincollig	Cork
6	CFR3	Ballygarvan	Cork
7	CFR4	Carrigaline	Cork
8	CFR5	Crosshaven	Cork
9	CFR6	Whitegate	Cork
0	CFR7	Passage West	Cork
1	CFR8	Glanmire	Cork
2	CFR9	Castlemartyr	Cork
3	CFR10	Youghal	Cork
4	CFR11	Fermoy	Cork
5	CFR12	Falcarragh	Donegal
6	CFR13	Gweedore	Donegal
7	CFR14	Ballyshannon	Donegal
8	CFR15	Bundoran	Donegal
9	CFR16	Creevy	Donegal
0	CFR17	Ardara	Donegal
1	CFR18	Mountcharles	Donegal
2	CFR19	Milford	Donegal
3	CFR20	Tory Island	Donegal

Figure 4: Raw CFR dataset with no census codes.

4.3 Census 2016 & OSI Data

Both files were downloaded directly from the CSO website using the 20m version. The raw census 2016 data is in an excel entitled "RAWCENSUS_SAPS2016_ED3409" see Figure 5.

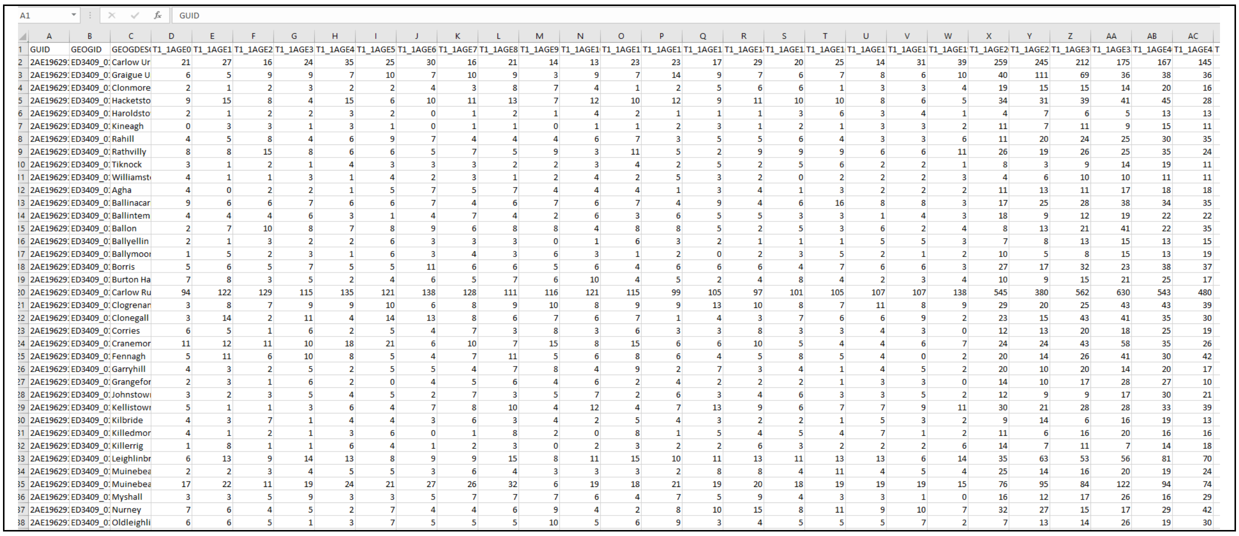
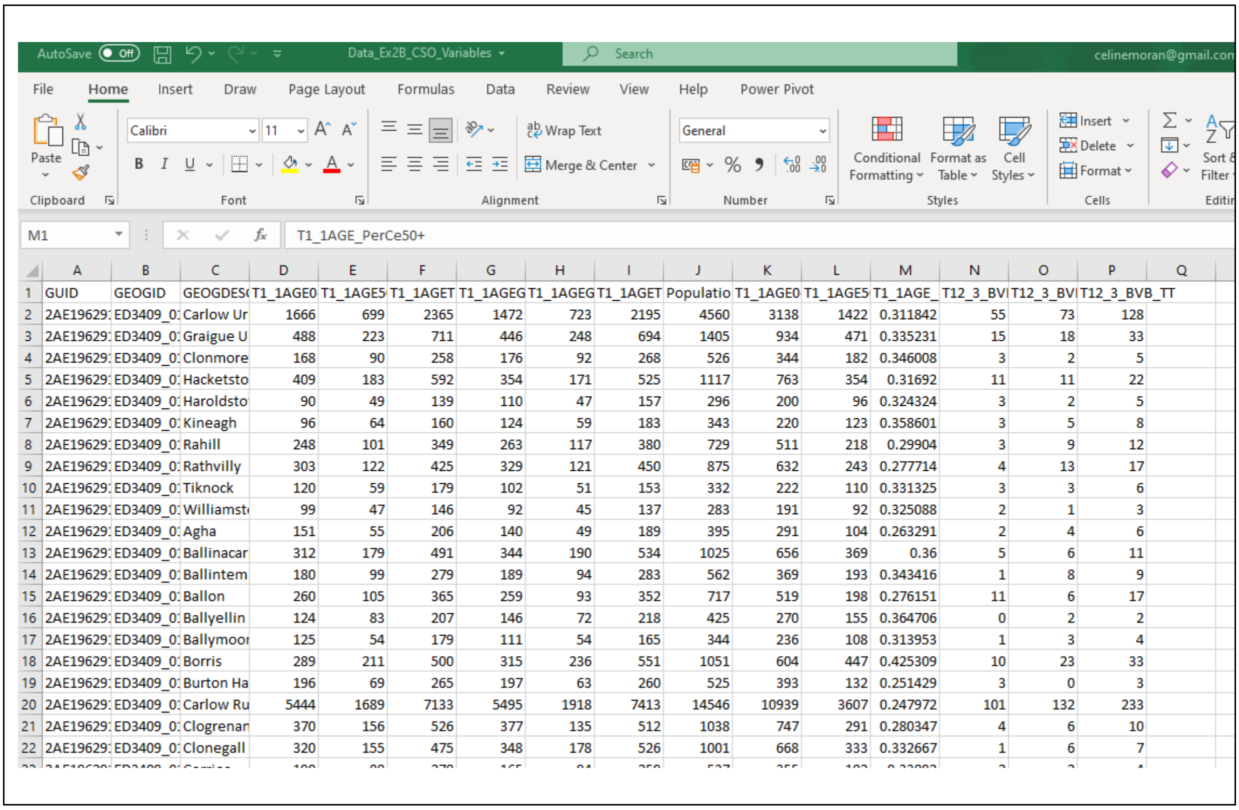


Figure 5: Raw Census 2016 data from the CSO website

The census data was edited in Excel and resaved as a CSV. The age columns for Male, Female and Total were summed by years 0-49 and 50+. There was a new variable created called Age50percent which accounts for the total percentage population of each electoral division above the age of 50. This CSV was entitled "Data_Ex2B_CSO_Variables.csv" Figure 6.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	GUID	GEOGID	GEOGDES(	T1_1AGE0	T1_1AGE5	T1_1AGET	T1_1AGEG	T1_1AGEG	T1_1AGET	Populatio	T1_1AGE0	T1_1AGE5	T1_1AGE_	T12_3_BVI	T12_3_BVI	T12_3_BVB_TT
2	2AE19629	ED3409_0	Carlow Ur	1666	699	2365	1472	723	2195	4560	3138	1422	0.311842	55	73	128
3	2AE19629	ED3409_0	Graigue U	488	223	711	446	248	694	1405	934	471	0.335231	15	18	33
4	2AE19629	ED3409_0	Clonmore	168	90	258	176	92	268	526	344	182	0.346008	3	2	5
5	2AE19629	ED3409_0	Hacketsto	409	183	592	354	171	525	1117	763	354	0.31692	11	11	22
6	2AE19629	ED3409_0	Haroldsto	90	49	139	110	47	157	296	200	96	0.324324	3	2	5
7	2AE19629	ED3409_0	Kineagh	96	64	160	124	59	183	343	220	123	0.358601	3	5	8
8	2AE19629	ED3409_0	Rahill	248	101	349	263	117	380	729	511	218	0.29904	3	9	12
9	2AE19629	ED3409_0	Rathvilly	303	122	425	329	121	450	875	632	243	0.277714	4	13	17
10	2AE19629	ED3409_0	Tiknock	120	59	179	102	51	153	332	222	110	0.331325	3	3	6
11	2AE19629	ED3409_0	Williamst	99	47	146	92	45	137	283	191	92	0.325088	2	1	3
12	2AE19629	ED3409_0	Agha	151	55	206	140	49	189	395	291	104	0.263291	2	4	6
13	2AE19629	ED3409_0	Ballinacar	312	179	491	344	190	534	1025	656	369	0.36	5	6	11
14	2AE19629	ED3409_0	Ballintem	180	99	279	189	94	283	562	369	193	0.343416	1	8	9
15	2AE19629	ED3409_0	Ballon	260	105	365	259	93	352	717	519	198	0.276151	11	6	17
16	2AE19629	ED3409_0	Ballyellin	124	83	207	146	72	218	425	270	155	0.364706	0	2	2
17	2AE19629	ED3409_0	Ballymoor	125	54	179	111	54	165	344	236	108	0.313953	1	3	4
18	2AE19629	ED3409_0	Borris	289	211	500	315	236	551	1051	604	447	0.425309	10	23	33
19	2AE19629	ED3409_0	Burton Ha	196	69	265	197	63	260	525	393	132	0.251429	3	0	3
20	2AE19629	ED3409_0	Carlow Ru	5444	1689	7133	5495	1918	7413	14546	10939	3607	0.247972	101	132	233
21	2AE19629	ED3409_0	Clogrenan	370	156	526	377	135	512	1038	747	291	0.280347	4	6	10
22	2AE19629	ED3409_0	Clonegall	320	155	475	348	178	526	1001	668	333	0.332667	1	6	7

Figure 6: CSO edited data in excel

4.4 Irish Road Network

This was downloaded as a shapefile directly from the website http://download.geofabrik.de/europe/ireland-and-northern-ireland.html and entitled "gis_osm_roads_free_1". Figure 7-9 illustrates the road network derived from the code described in section 5.5. This dataset crashed the computer several times due to computational issues. Ensure there is enough memory to run the code before loading in the road network data. See R Code entitled "MemoryCode.r" to increase memory limit.

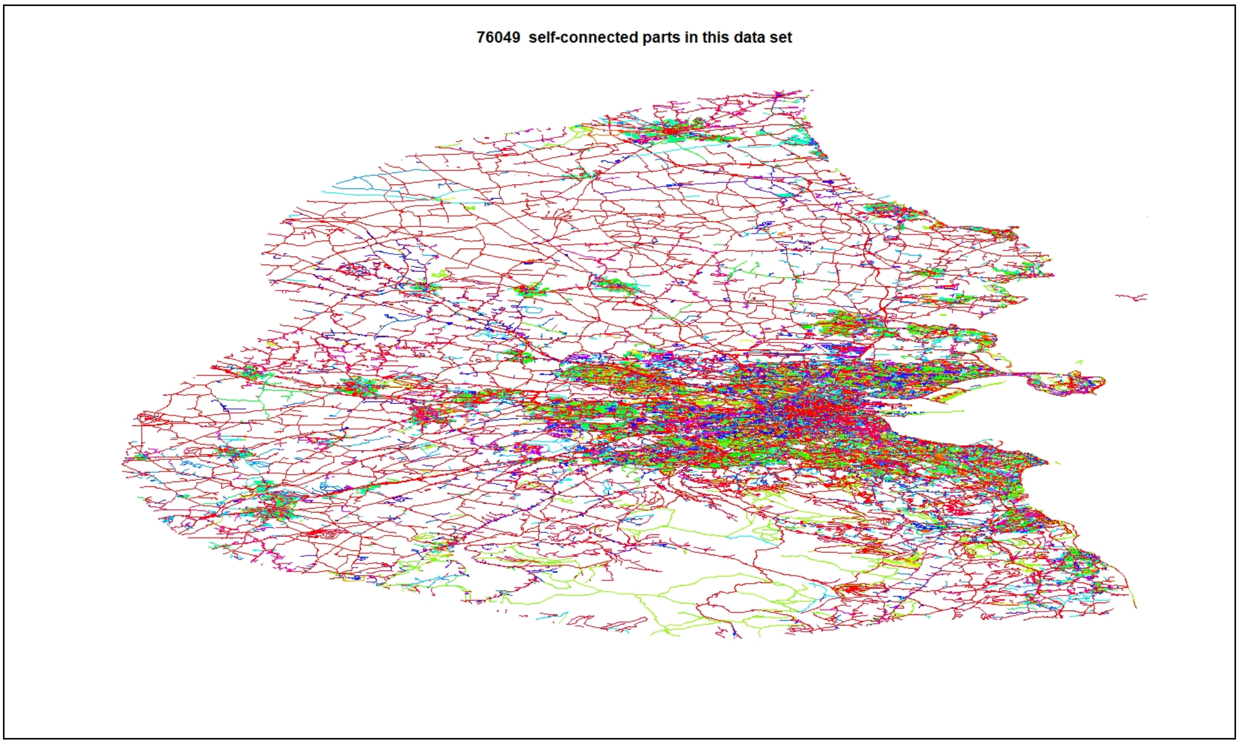


Figure 7: Various Road network levels in Dublin County and city

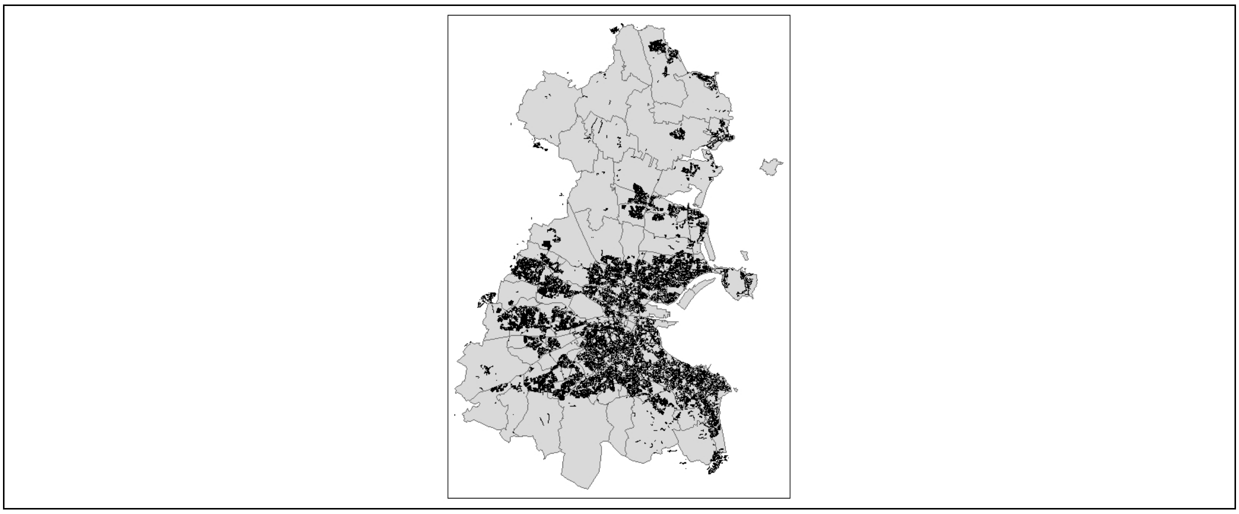


Figure 8: Dublin residential road network against electoral divisions.

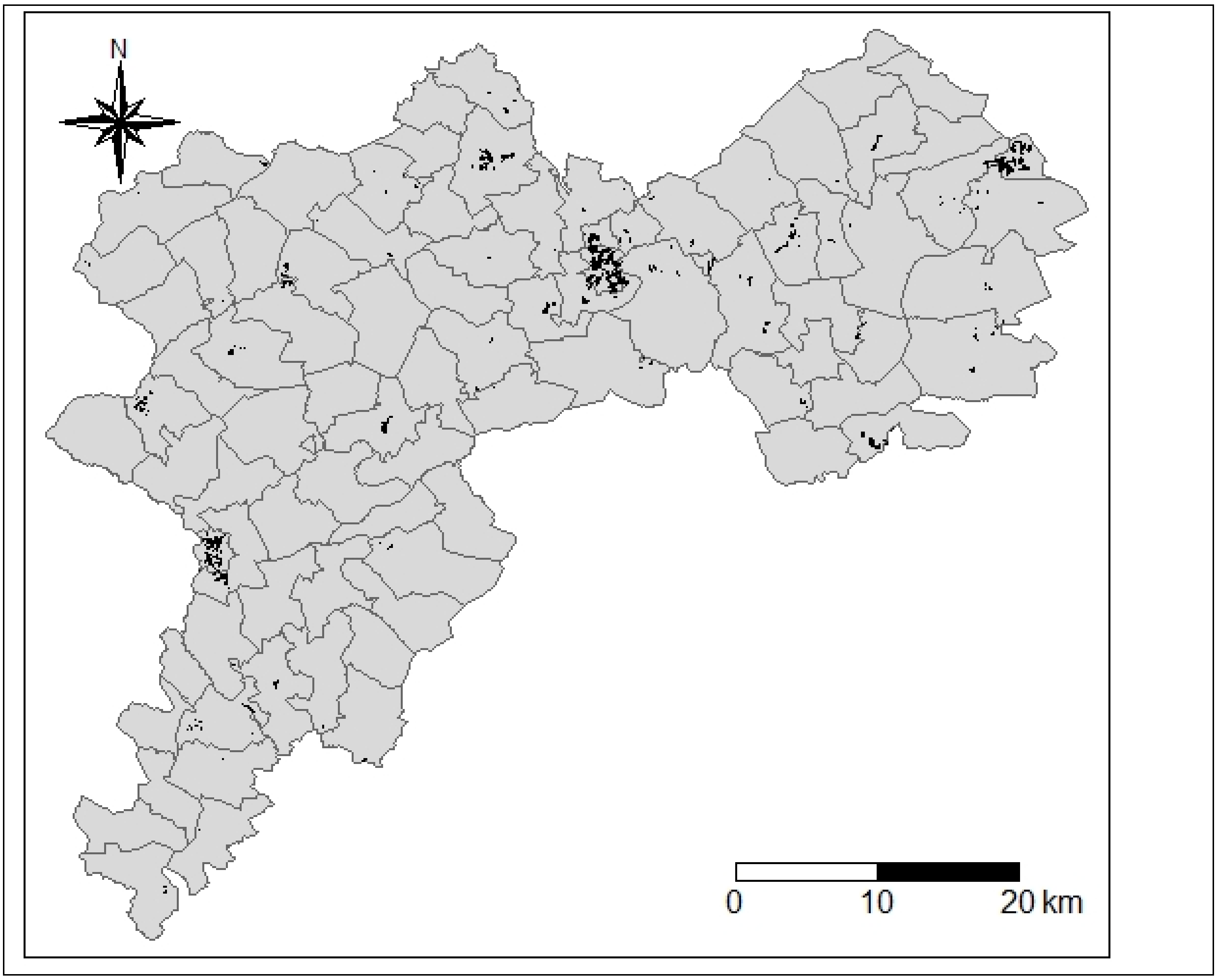


Figure 9: Electoral Divisions against residential road network in County Offaly

5 Experiment Implementation

5.1 SIR Calculation

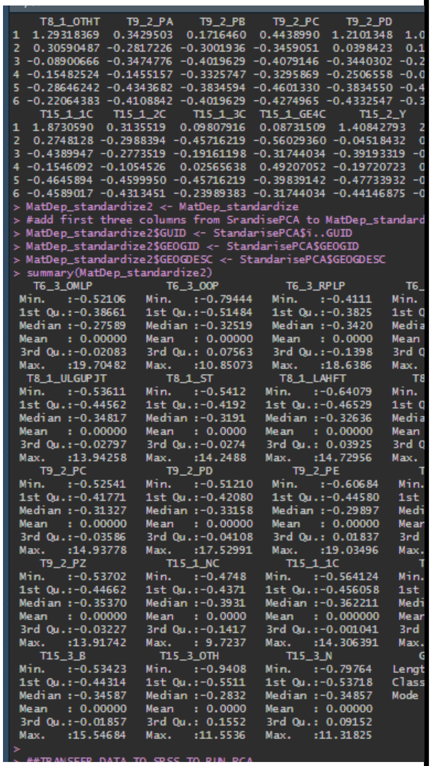
The Standardised Incidence Ratio (SIR) was calculated in R Studio using the CSV file "Data_Ex2B_CSO_Variables.csv" and was saved to the data frame "Data_SIR" with two files saved to CSV 1) "SIR_ABOVE1_Exp2B.csv" and 2) "Data_SIR_healthNoFair.csv". The variable Age50plus was also created within this code and the summary function was used to apply the statistics to table 1 in the report Figure 10.

```
Console  Terminal  Jobs
~/
> #inspect dataframe
> summary(Exp2B)
     GUID              GEOGID             GEOGDESC          T1_1AGE0.49M    T1_1AGE50._M      T1_1AGETM       T1_1AGEGE0.49F    T1_1AGEGE_50.F
 Length:3409        Length:3409        Length:3409        Min.   :   14   Min.   :  12.0   Min.   :   30.0   Min.   :   12.0   Min.   :  10.0
 Class :character   Class :character   Class :character   1st Qu.:  113   1st Qu.:  65.0   1st Qu.:  178.0   1st Qu.:  107.0   1st Qu.:  61.0
 Mode  :character   Mode  :character   Mode  :character   Median :  206   Median : 109.0   Median :  320.0   Median :  201.0   Median : 106.0
                                                          Mean   :  486   Mean   : 204.6   Mean   :  690.7   Mean   :  486.5   Mean   : 219.7
                                                          3rd Qu.:  465   3rd Qu.: 223.0   3rd Qu.:  706.0   3rd Qu.:  471.0   3rd Qu.: 231.0
                                                          Max.   :16279   Max.   :2916.0   Max.   :19195.0   Max.   :16509.0   Max.   :3190.0

   T1_1AGETF       PopulationTotal T1_1AGE0.49TT     T1_1AGE50.TT    T1_1AGE_PerCe50.   T12_3_BVB_M      T12_3_BVB_F      T12_3_BVB_TT      Age50plus
 Min.   :   31.0   Min.   :   66   Min.   :   34.0   Min.   :  22.0   Min.   :0.08443   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   [0-1):   2
 1st Qu.:  168.0   1st Qu.:  349   1st Qu.:  221.0   1st Qu.: 126.0   1st Qu.:0.30330   1st Qu.:  2.00   1st Qu.:  2.00   1st Qu.:  4.00   [1-2):  85
 Median :  311.0   Median :  630   Median :  409.0   Median : 214.0   Median :0.34876   Median :  4.00   Median :  4.00   Median :  8.00   [2-3): 724
 Mean   :  706.2   Mean   : 1397   Mean   :  972.5   Mean   : 424.3   Mean   :0.34798   Mean   : 10.71   Mean   : 11.72   Mean   : 22.42   [3-4):1855
 3rd Qu.:  730.0   3rd Qu.: 1443   3rd Qu.:  937.0   3rd Qu.: 458.0   3rd Qu.:0.39286   3rd Qu.: 10.00   3rd Qu.: 12.00   3rd Qu.: 22.00   [4-5): 673
 Max.   :19699.0   Max.   :38894   Max.   :32788.0   Max.   :6106.0   Max.   :0.61147   Max.   :190.00   Max.   :216.00   Max.   :395.00   [5-6):  67
                                                                                                                                           [6-7):   3
>
> #OBSERVED CASES
> d <- aggregate(x = Exp2B$T12_3_BVB_TT, by = list(GUID = Exp2B$GUID),
+                FUN = sum)
> names(d) <- c("GUID", "Y")
> library(SpatialEpi)
> population <- Exp2B$PopulationTotal
> selfbadhealth <- Exp2B$T12_3_BVB_TT
> n.strata <- 7
> Exp2B$E <- expected(population, selfbadhealth, n.strata)
>
> Data_SIR <- merge(Exp2B, d, by.x = "GUID", by.y = "GUID")
> Data_SIR$SIR_TTBH <- Data_SIR$Y/Data_SIR$E
> summary(Data_SIR)
     GUID              GEOGID             GEOGDESC          T1_1AGE0.49M    T1_1AGE50._M      T1_1AGETM       T1_1AGEGE0.49F    T1_1AGEGE_50.F
 Length:3409        Length:3409        Length:3409        Min.   :   14   Min.   :  12.0   Min.   :   30.0   Min.   :   12.0   Min.   :  10.0
 Class :character   Class :character   Class :character   1st Qu.:  113   1st Qu.:  65.0   1st Qu.:  178.0   1st Qu.:  107.0   1st Qu.:  61.0
 Mode  :character   Mode  :character   Mode  :character   Median :  206   Median : 109.0   Median :  320.0   Median :  201.0   Median : 106.0
                                                          Mean   :  486   Mean   : 204.6   Mean   :  690.7   Mean   :  486.5   Mean   : 219.7
                                                          3rd Qu.:  465   3rd Qu.: 223.0   3rd Qu.:  706.0   3rd Qu.:  471.0   3rd Qu.: 231.0
                                                          Max.   :16279   Max.   :2916.0   Max.   :19195.0   Max.   :16509.0   Max.   :3190.0

   T1_1AGETF       PopulationTotal T1_1AGE0.49TT     T1_1AGE50.TT    T1_1AGE_PerCe50.   T12_3_BVB_M      T12_3_BVB_F      T12_3_BVB_TT      Age50plus
 Min.   :   31.0   Min.   :   66   Min.   :   34.0   Min.   :  22.0   Min.   :0.08443   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   [0-1):   2
 1st Qu.:  168.0   1st Qu.:  349   1st Qu.:  221.0   1st Qu.: 126.0   1st Qu.:0.30330   1st Qu.:  2.00   1st Qu.:  2.00   1st Qu.:  4.00   [1-2):  85
 Median :  311.0   Median :  630   Median :  409.0   Median : 214.0   Median :0.34876   Median :  4.00   Median :  4.00   Median :  8.00   [2-3): 724
 Mean   :  706.2   Mean   : 1397   Mean   :  972.5   Mean   : 424.3   Mean   :0.34798   Mean   : 10.71   Mean   : 11.72   Mean   : 22.42   [3-4):1855
 3rd Qu.:  730.0   3rd Qu.: 1443   3rd Qu.:  937.0   3rd Qu.: 458.0   3rd Qu.:0.39286   3rd Qu.: 10.00   3rd Qu.: 12.00   3rd Qu.: 22.00   [4-5): 673
 Max.   :19699.0   Max.   :38894   Max.   :32788.0   Max.   :6106.0   Max.   :0.61147   Max.   :190.00   Max.   :216.00   Max.   :395.00   [5-6):  67
                                                                                                                                           [6-7):   3
       E                 Y              SIR_TTBH
 Min.   :  21.15   Min.   :  0.00   Min.   :0.00000
 1st Qu.:  60.45   1st Qu.:  4.00   1st Qu.:0.03108
 Median :  90.33   Median :  8.00   Median :0.08682
 Mean   : 156.95   Mean   : 22.42   Mean   :0.26874
 3rd Qu.: 182.86   3rd Qu.: 22.00   3rd Qu.:0.24123
 Max.   :1220.29   Max.   :395.00   Max.   :8.73161
```

Figure 10: SIR Code results and creation of Age50plus variable

5.2 Principal Component Analysis

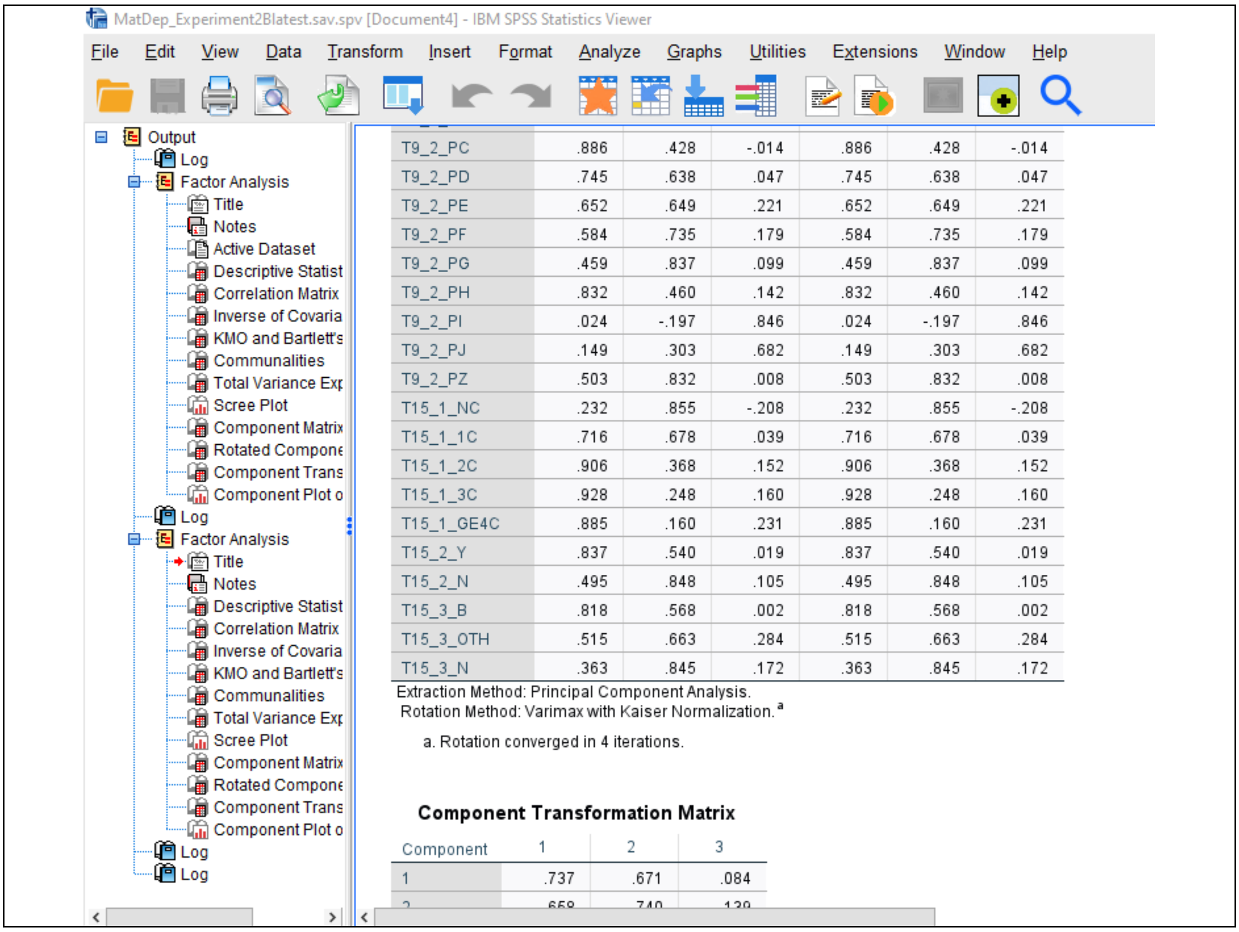
The columns were selected from the census excel and saved into a CSV entitled "MaterialDeprivation_ResearchProject.csv" to upload to R Studio to apply standardisation and to clean the data to remove punctuation. Standardization was applied using the base package in R. Once the data was standardised a CSV was created entitled "MatDep_standardize2.csv" to upload to SPSS Figure 11.



```
    T8_1_OTHT   T9_2_PA    T9_2_PB    T9_2_PC    T9_2_PD
1  1.29318369  0.3429503  0.1716460  0.4438990  1.2101348  1.0
2  0.30590487 -0.2817226 -0.3001936 -0.3459051  0.0398423  0.1
3 -0.08900666 -0.3474776 -0.4019629 -0.4079146 -0.3440302 -0.2
4 -0.15482524 -0.1455157 -0.3325747 -0.3295869 -0.2506558 -0.0
5 -0.28646242 -0.4343682 -0.3834594 -0.4601330 -0.3834550 -0.4
6 -0.22064383 -0.4108842 -0.4019629 -0.4274965 -0.4332547 -0.3
    T15_1_1C   T15_1_2C    T15_1_3C  T15_1_GE4C     T15_2_Y
1  1.8730590  0.3135519  0.09807916  0.08731509  1.40842793  2
2  0.2748128 -0.2988394 -0.45716219 -0.56029360 -0.04518432  0
3 -0.4389947 -0.2773519 -0.19161198 -0.31744034 -0.39193319 -0
4 -0.1546092 -0.1054526  0.02565638  0.49207052 -0.19720723  0
5 -0.4645894 -0.4599950 -0.45716219 -0.39839142 -0.47733932 -0
6 -0.4589017 -0.4313451 -0.23989383 -0.31744034 -0.44146875 -0
> MatDep_standardize2 <- MatDep_standardize
> #add first three columns from SrandisePCA to MatDep_standard
> MatDep_standardize2$GUID <- StandarisePCA$ï..GUID
> MatDep_standardize2$GEOGID <- StandarisePCA$GEOGID
> MatDep_standardize2$GEOGDESC <- StandarisePCA$GEOGDESC
> summary(MatDep_standardize2)
   T6_3_OMLP          T6_3_OOP          T6_3_RPLP        T6_
 Min.   :-0.52106   Min.   :-0.79444   Min.   :-0.4111   Min.
 1st Qu.:-0.38661   1st Qu.:-0.51484   1st Qu.:-0.3825   1st Q
 Median :-0.27589   Median :-0.32519   Median :-0.3420   Media
 Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean
 3rd Qu.:-0.02083   3rd Qu.: 0.07563   3rd Qu.:-0.1398   3rd Q
 Max.   :19.70482   Max.   :10.85073   Max.   :18.6386   Max.
  T8_1_ULGUPJT        T8_1_ST          T8_1_LAHFT         T8
 Min.   :-0.53611   Min.   :-0.5412   Min.   :-0.64079   Min.
 1st Qu.:-0.44562   1st Qu.:-0.4192   1st Qu.:-0.46529   1st Q
 Median :-0.34817   Median :-0.3191   Median :-0.32636   Media
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000   Mean
 3rd Qu.:-0.02797   3rd Qu.:-0.0274   3rd Qu.: 0.03925   3rd Q
 Max.   :13.94258   Max.   :14.2488   Max.   :14.72956   Max.
    T9_2_PC            T9_2_PD            T9_2_PE           T
 Min.   :-0.52541   Min.   :-0.51210   Min.   :-0.60684   Min.
 1st Qu.:-0.41771   1st Qu.:-0.42080   1st Qu.:-0.44580   1st
 Median :-0.31327   Median :-0.33158   Median :-0.29897   Medi
 Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000   Mear
 3rd Qu.:-0.03586   3rd Qu.:-0.04108   3rd Qu.: 0.01837   3rd
 Max.   :14.93778   Max.   :17.52991   Max.   :19.03496   Max.
    T9_2_PZ           T15_1_NC          T15_1_1C            T
 Min.   :-0.53702   Min.   :-0.4748   Min.   :-0.564124   Min.
 1st Qu.:-0.44662   1st Qu.:-0.4371   1st Qu.:-0.456058   1st
 Median :-0.35370   Median :-0.3931   Median :-0.362211   Medi
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.000000   Mear
 3rd Qu.:-0.03227   3rd Qu.:-0.1417   3rd Qu.:-0.001041   3rd
 Max.   :13.91742   Max.   : 9.7237   Max.   :14.306391   Max.
    T15_3_B          T15_3_OTH          T15_3_N            G
 Min.   :-0.53423   Min.   :-0.9408   Min.   :-0.79764   Lengt
 1st Qu.:-0.44314   1st Qu.:-0.5511   1st Qu.:-0.53718   Class
 Median :-0.34587   Median :-0.2832   Median :-0.34857   Mode
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000
 3rd Qu.:-0.01857   3rd Qu.: 0.1552   3rd Qu.: 0.09152
 Max.   :15.54684   Max.   :11.5536   Max.   :11.31825
>
```

Figure 11: SIR Code results and creation of Age50plus variable

Within SPSS PCA was applied to the data relying on the Eigenvalues to determine the components and then rerun a second time specifying two components. The method relying on the Eigenvalues was used for the report. See SPSS files entitled "MatDep_Experiment2blatest.sav.spv" and MatDep_Experiment2B.sav[DataSet1]. See screenshot in Figure 12 and 13. The output of PCA was saved into a CSV and entitled "SPSS_MatDepOutputforR" Figure 14.



T9_2_PC	.886	.428	-.014	.886	.428	-.014
T9_2_PD	.745	.638	.047	.745	.638	.047
T9_2_PE	.652	.649	.221	.652	.649	.221
T9_2_PF	.584	.735	.179	.584	.735	.179
T9_2_PG	.459	.837	.099	.459	.837	.099
T9_2_PH	.832	.460	.142	.832	.460	.142
T9_2_PI	.024	-.197	.846	.024	-.197	.846
T9_2_PJ	.149	.303	.682	.149	.303	.682
T9_2_PZ	.503	.832	.008	.503	.832	.008
T15_1_NC	.232	.855	-.208	.232	.855	-.208
T15_1_1C	.716	.678	.039	.716	.678	.039
T15_1_2C	.906	.368	.152	.906	.368	.152
T15_1_3C	.928	.248	.160	.928	.248	.160
T15_1_GE4C	.885	.160	.231	.885	.160	.231
T15_2_Y	.837	.540	.019	.837	.540	.019
T15_2_N	.495	.848	.105	.495	.848	.105
T15_3_B	.818	.568	.002	.818	.568	.002
T15_3_OTH	.515	.663	.284	.515	.663	.284
T15_3_N	.363	.845	.172	.363	.845	.172

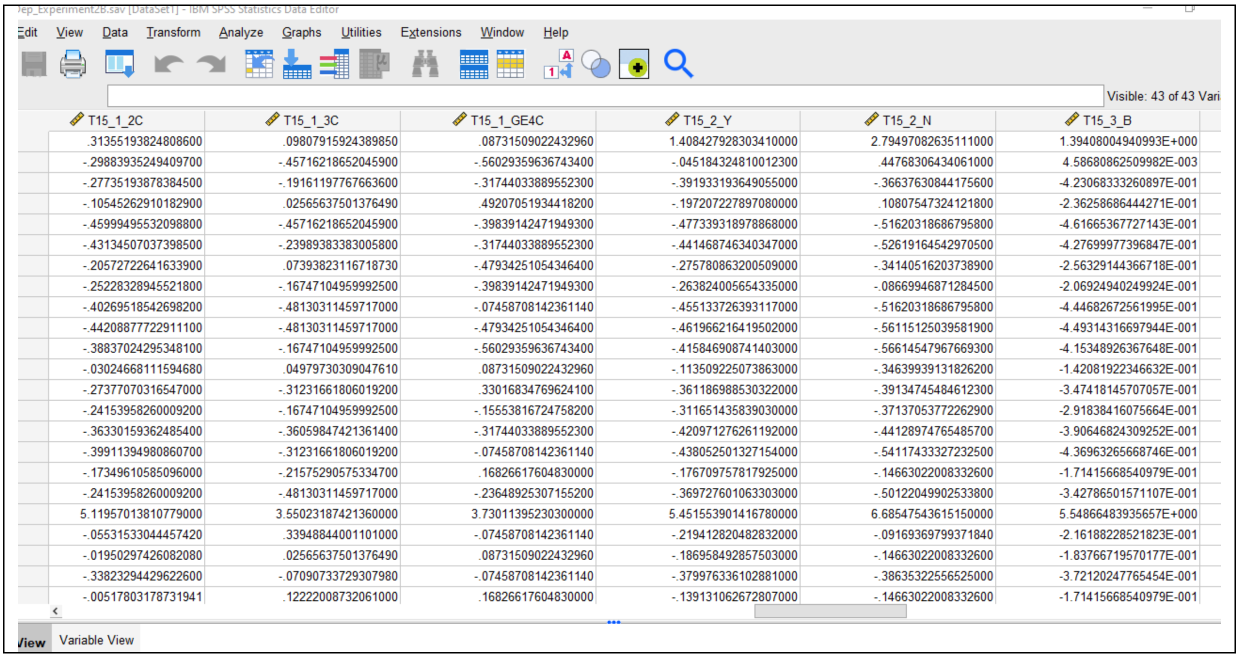
Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.[a]

a. Rotation converged in 4 iterations.

Component Transformation Matrix

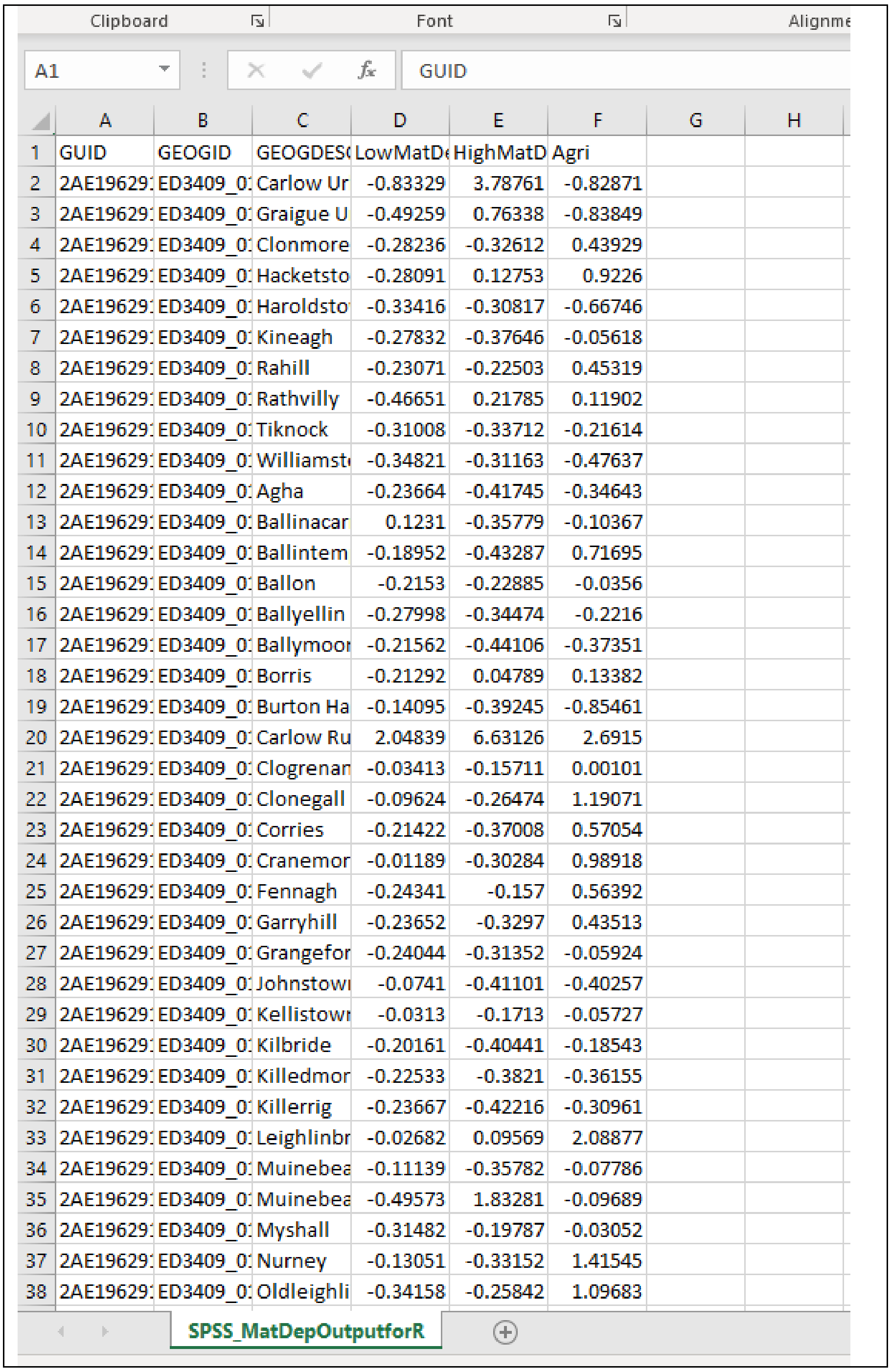
Component	1	2	3
1	.737	.671	.084

Figure 12: Output of PCA in SPSS



T15_1_2C	T15_1_3C	T15_1_GE4C	T15_2_Y	T15_2_N	T15_3_B
.31355193824808600	.09807915924389850	.08731509022432960	1.40842792830341000	2.79497082635111000	1.39408004940993E+000
-.29883935249409700	-.45716218652045900	-.56029359636743400	-.04518432481001230	.44768306434061000	4.58680862509982E-003
-.27735193878384500	-.19161197767663600	-.31744033889552300	-.39193319364905500	-.36637630844175600	-4.23068333260897E-001
-.10545262910182900	.02565637501376490	.49207051934418200	-.19720722789708000	.10807547324121800	-2.36258686444271E-001
-.45999495532098800	-.45716218652045900	-.39839142471949300	-.47733931897886800	-.51620318686795800	-4.61665367727143E-001
-.43134507037398500	-.23989383383005800	-.31744033889552300	-.44146874634034700	-.52619164542970500	-4.27699977396847E-001
-.20572722641633900	.07393823116718730	-.47934251054346400	-.27578086320050900	-.34140516203738900	-2.56329144366718E-001
-.25228328945521800	-.16747104959992500	-.39839142471949300	-.26382400565433500	-.08669946871284500	-2.06924940249924E-001
-.40269518542698200	-.48130311459717000	-.07458708142361140	-.45513372639311700	-.51620318686795800	-4.44682672561995E-001
-.44208877722911100	-.48130311459717000	-.47934251054346400	-.46196621641950200	-.56115125039581900	-4.49314316697944E-001
-.38837024295348100	-.16747104959992500	-.56029359636743400	-.41584690874140300	-.56614547967669300	-4.15348926367648E-001
-.03024668111594680	.04979730309047610	.08731509022432960	-.11350922507386300	-.34639939131826200	-1.42081922346632E-001
-.27377070316547000	-.31231661806019200	.33016834769624100	-.36118698853032200	-.39134745484612300	-3.47418145707057E-001
-.24153958260009200	-.16747104959992500	-.15553816724758200	-.31165143583903000	-.37137053772262900	-2.91838416075664E-001
-.36330159362485400	-.36059847421361400	-.31744033889552300	-.42097127626119200	-.44128974765485700	-3.90646824309252E-001
-.39911394980860700	-.31231661806019200	-.07458708142361140	-.43805250132715400	-.54117433327232500	-4.36963265668746E-001
-.17349610585096000	-.21575290575334700	.16826617604830000	-.17670975781792500	-.14663022008332600	-1.71415668540979E-001
-.24153958260009200	-.48130311459717000	-.23648925307155200	-.36972760106330300	-.50122049902533800	-3.42786501571107E-001
5.11957013810779000	3.55023187421360000	3.73011395230300000	5.45155390141678000	6.68547543615150000	5.54866483935657E+000
-.05531533044457420	.33948844001101000	-.07458708142361140	-.21941282048283200	-.09169369799371840	-2.16188228521823E-001
-.01950297426082080	.02565637501376490	.08731509022432960	-.18695849285750300	-.14663022008332600	-1.83766719570177E-001
-.33823294429622600	-.07090733729307980	-.07458708142361140	-.37997633610288100	-.38635322556525000	-3.72120247765454E-001
-.00517803178731941	.12222008732061000	.16826617604830000	-.13913106267280700	-.14663022008332600	-1.71415668540979E-001

Figure 13: Output of R of standardized data



Clipboard Font Alignme

A1 GUID

	A	B	C	D	E	F	G	H
1	GUID	GEOGID	GEOGDES(	LowMatD(	HighMatD	Agri		
2	2AE19629	ED3409_0	Carlow Ur	-0.83329	3.78761	-0.82871		
3	2AE19629	ED3409_0	Graigue U	-0.49259	0.76338	-0.83849		
4	2AE19629	ED3409_0	Clonmore	-0.28236	-0.32612	0.43929		
5	2AE19629	ED3409_0	Hacketsto	-0.28091	0.12753	0.9226		
6	2AE19629	ED3409_0	Haroldsto	-0.33416	-0.30817	-0.66746		
7	2AE19629	ED3409_0	Kineagh	-0.27832	-0.37646	-0.05618		
8	2AE19629	ED3409_0	Rahill	-0.23071	-0.22503	0.45319		
9	2AE19629	ED3409_0	Rathvilly	-0.46651	0.21785	0.11902		
10	2AE19629	ED3409_0	Tiknock	-0.31008	-0.33712	-0.21614		
11	2AE19629	ED3409_0	Williamst	-0.34821	-0.31163	-0.47637		
12	2AE19629	ED3409_0	Agha	-0.23664	-0.41745	-0.34643		
13	2AE19629	ED3409_0	Ballinacar	0.1231	-0.35779	-0.10367		
14	2AE19629	ED3409_0	Ballintem	-0.18952	-0.43287	0.71695		
15	2AE19629	ED3409_0	Ballon	-0.2153	-0.22885	-0.0356		
16	2AE19629	ED3409_0	Ballyellin	-0.27998	-0.34474	-0.2216		
17	2AE19629	ED3409_0	Ballymoor	-0.21562	-0.44106	-0.37351		
18	2AE19629	ED3409_0	Borris	-0.21292	0.04789	0.13382		
19	2AE19629	ED3409_0	Burton Ha	-0.14095	-0.39245	-0.85461		
20	2AE19629	ED3409_0	Carlow Ru	2.04839	6.63126	2.6915		
21	2AE19629	ED3409_0	Clogrenan	-0.03413	-0.15711	0.00101		
22	2AE19629	ED3409_0	Clonegall	-0.09624	-0.26474	1.19071		
23	2AE19629	ED3409_0	Corries	-0.21422	-0.37008	0.57054		
24	2AE19629	ED3409_0	Cranemor	-0.01189	-0.30284	0.98918		
25	2AE19629	ED3409_0	Fennagh	-0.24341	-0.157	0.56392		
26	2AE19629	ED3409_0	Garryhill	-0.23652	-0.3297	0.43513		
27	2AE19629	ED3409_0	Grangefor	-0.24044	-0.31352	-0.05924		
28	2AE19629	ED3409_0	Johnstowi	-0.0741	-0.41101	-0.40257		
29	2AE19629	ED3409_0	Kellistowi	-0.0313	-0.1713	-0.05727		
30	2AE19629	ED3409_0	Kilbride	-0.20161	-0.40441	-0.18543		
31	2AE19629	ED3409_0	Killedmor	-0.22533	-0.3821	-0.36155		
32	2AE19629	ED3409_0	Killerrig	-0.23667	-0.42216	-0.30961		
33	2AE19629	ED3409_0	Leighlinbr	-0.02682	0.09569	2.08877		
34	2AE19629	ED3409_0	Muinebea	-0.11139	-0.35782	-0.07786		
35	2AE19629	ED3409_0	Muinebea	-0.49573	1.83281	-0.09689		
36	2AE19629	ED3409_0	Myshall	-0.31482	-0.19787	-0.03052		
37	2AE19629	ED3409_0	Nurney	-0.13051	-0.33152	1.41545		
38	2AE19629	ED3409_0	Oldleighli	-0.34158	-0.25842	1.09683		

SPSS_MatDepOutputforR

Figure 14: Material Deprivation Scores from SPSS in CSV to upload to R

5.3 Queen's Matrix

R code was created to apply the Queen's Matrix however some islands and electoral divisions were omitted totalling 100. These had to be manually connected in GeoDa by importing the shapefiles ("c99096be-cff8-4729-b967-ebd104afdb012020328-1-nubkfi.o4ctb") for the electoral divisions and the data created in R ("TotalCARData.csv") to merge in Geoda and output as a shapefile Figure 15.

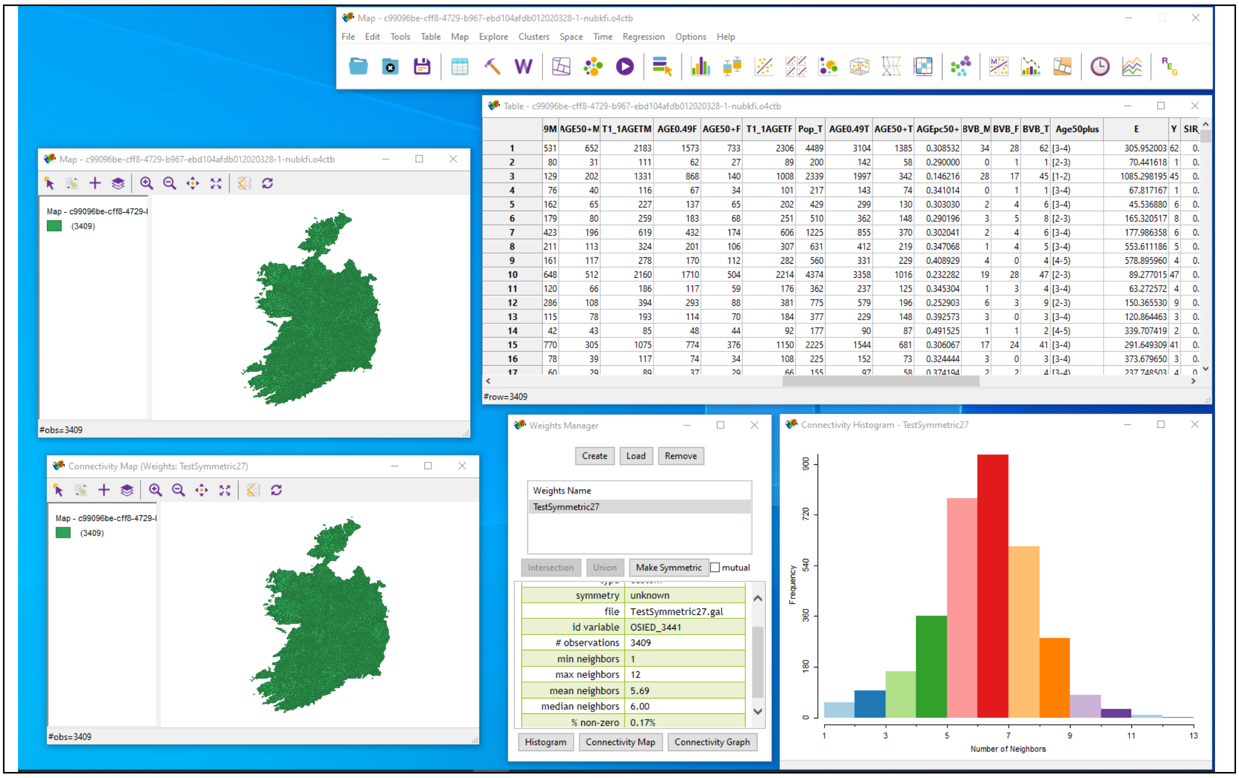
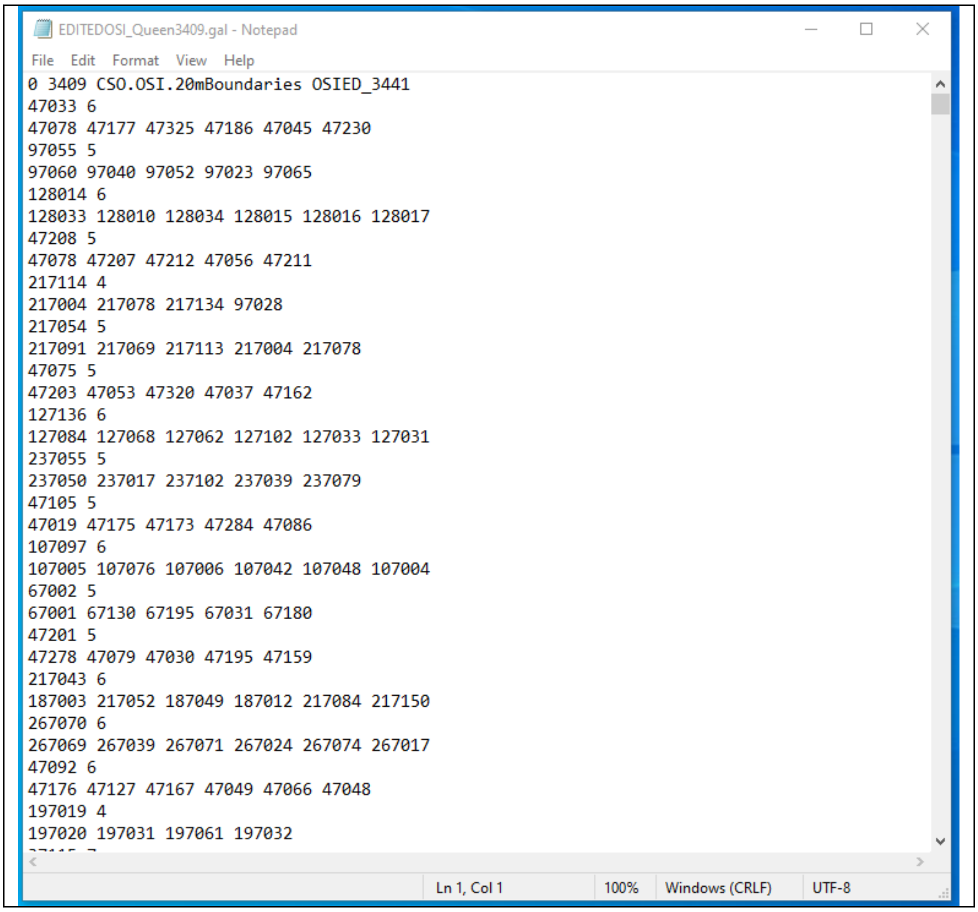


Figure 15: Merged data in GeoDa including creation of Queen's Matrix

The Queen's matrix was created with this information and the EDOSI codes were manually tagged to all 100 missing electoral divisions. The first row represents the electoral division to be manually tagged to a neighbour, the second number is the number of neighbours and the next line is the codes for each neighbour Figure 16 & 16a. This text file (EDIEDOSI_Queen3409.gal) was then uploaded to GeoDa and symmetry applied to connect the neighbours together. This was saved as a Gal file named "TestSummetric27" Figure 14. The data was then all saved together as a shapefile under the same name. NB the current files in the ZIP folder has the GeoDa data saved onto the file. However, Geoda is very simple to use, it is recommended to download the software.



EDITEDOSI_Queen3409.gal - Notepad

File Edit Format View Help

```
0 3409 CSO.OSI.20mBoundaries OSIED_3441
47033 6
47078 47177 47325 47186 47045 47230
97055 5
97060 97040 97052 97023 97065
128014 6
128033 128010 128034 128015 128016 128017
47208 5
47078 47207 47212 47056 47211
217114 4
217004 217078 217134 97028
217054 5
217091 217069 217113 217004 217078
47075 5
47203 47053 47320 47037 47162
127136 6
127084 127068 127062 127102 127033 127031
237055 5
237050 237017 237102 237039 237079
47105 5
47019 47175 47173 47284 47086
107097 6
107005 107076 107006 107042 107048 107004
67002 5
67001 67130 67195 67031 67180
47201 5
47278 47079 47030 47195 47159
217043 6
187003 217052 187049 187012 217084 217150
267070 6
267069 267039 267071 267024 267074 267017
47092 6
47176 47127 47167 47049 47066 47048
197019 4
197020 197031 197061 197032
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

Figure 16: Raw CSO codes for Queen's matrix

Figure 16a: Queen's matrix neighbour connections on map of Ireland

Within Geoda each of the variables intended for use in the Bayesian CAR models were tested for Auto Spatial Correlation using the Moran's I univariate statistic. The results are illustrated in figure 17. The final dataset in Geoda was saved to as a shapefile under the name “c99096be-cff8-4729-b967-ebd104afdb012020328-1-nubkfi.o4ctb” and imported into R.

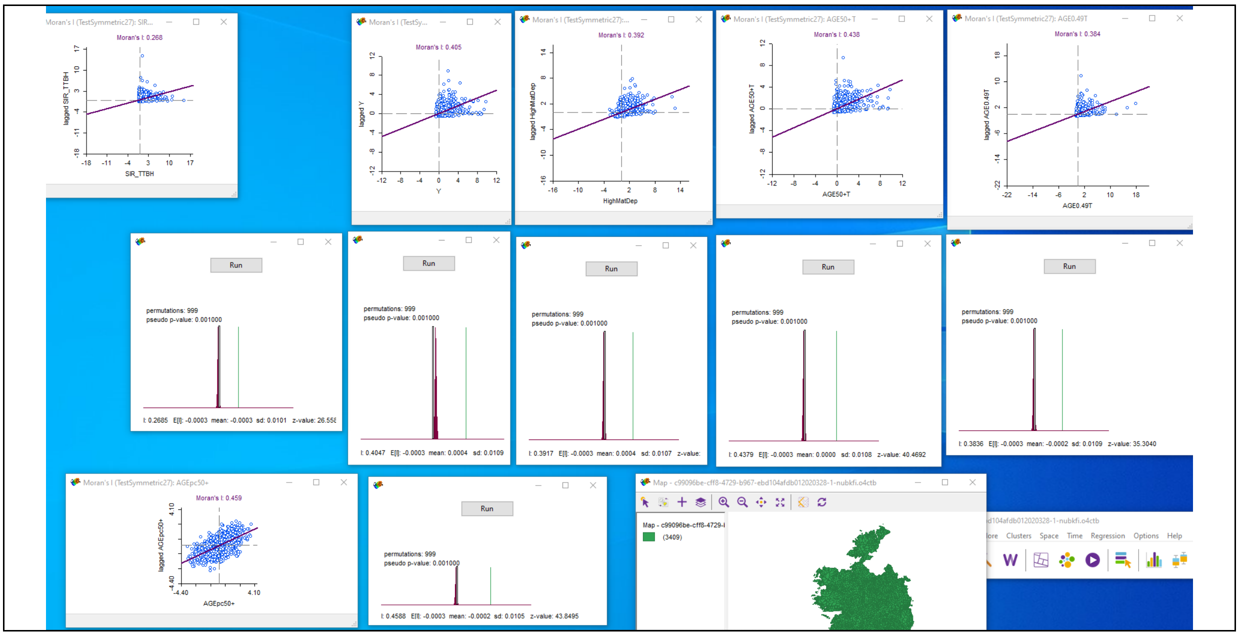


Figure 17: Moran’s I statistic for each variable.

At the end of the R script 5.3 Queens Matrix, the CART model was created. See image below of code and results Figure 18.

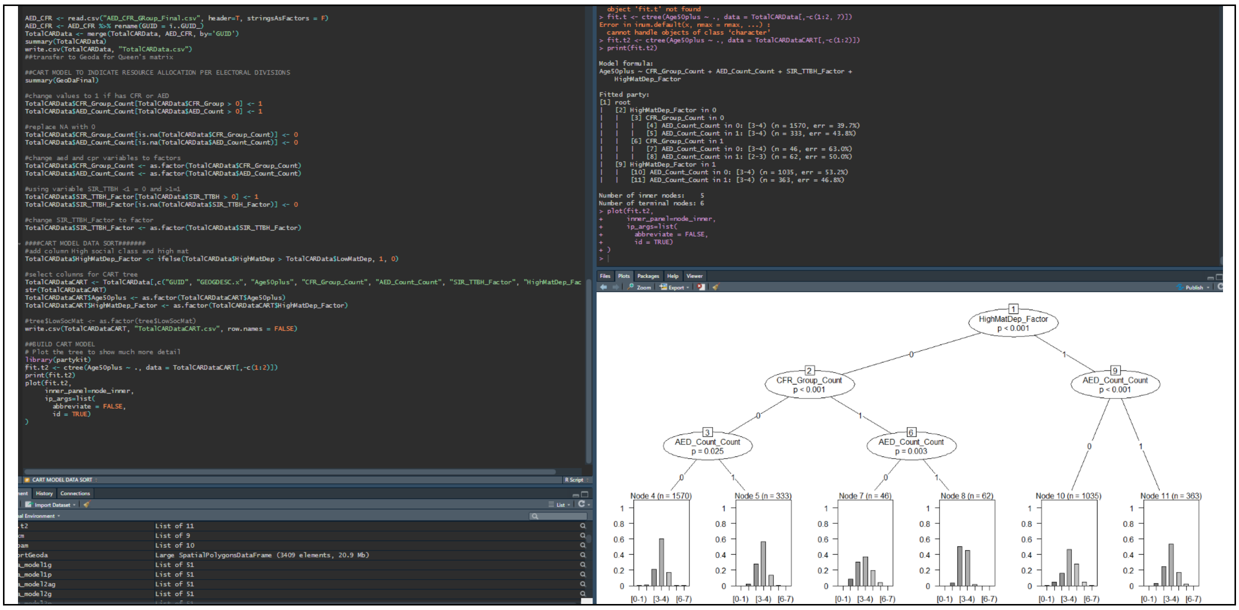


Figure 18: CART model indicating relationship between age, material deprivation, AED and CFR groups per electoral division.

5.4 Bayesian Car Models

The GeoDa dataset was imported into R and the name was converted to bvb.sp. In Figure 19 the import is illustrated as well as the import of the GAL file with the neighbour matrix and the libraries to run the Bayesian CAR models. The final line of code in Figure 19 is converting the longitude and latitude coordinates to Ireland.

```r
##Bayesian Models
##resource https://cancerqld.blob.core.windows.net/content/docs/Investigation-of-Bayesian-spatial-models.pdf
##https://cran.r-project.org/web/packages/CARBayes/vignettes/CARBayes.pdf
importGeoda <- readOGR(dsn=".", layer = "c99096be-cff8-4729-b967-ebd104afdb012020328-1-rubkfi.o4ctb")
summary(importGeoda)
importGeoda$Age50plus <- as.factor(importGeoda$Age50plus)
sum(importGeoda$T12_3_BVB_TT) #76435
sum(importGeoda$PopulationTotal) #4761865
bvb.sp <- importGeoda
bvb.sp@data
bvb.sp@data <- bvb.sp@data %>% rename(long = x, lat = y_1)
##step 2 - import gal file with summetric neighbours into r
library(spdep)
queen.nb <- read.gal("TestSymmetric27.gal", override.id = TRUE)
lw <- nb2listw(queen.nb, style="W", zero.policy=TRUE)
lw$weights[1]
Inc.lag<- lag.listw(lw, GeoDaFinal$T12_3_BVB_TT)

# step 3 - run moran's test - done in Geoda

##step 4 - run car in different formats
# taken from page 19 onwards https://cran.r-project.org/web/packages/CARBayes/vignettes/CARBayes.pdf
library(CARBayes)
W <- nb2mat(queen.nb, style="B")

##step 5 - figure out maps for results
##testing CARBayes Package from https://cran.r-project.org/web/packages/CARBayes/vignettes/CARBayes.pdf
library(purrr)
library(matrixcalc)
library(CARBayes)
library(CARBayesdata)
library(sp)
#create data frame with columns needed for CAR analysis - do total first
library(leaflet)
library(rgdal)
library(rgeos)
library(raster)
library(dplyr)
library(htmltools)

bvb.sp <- spTransform(bvb.sp, CRS("+proj=longlat +datum=WGS84 +no_defs")) #add long lat coordinates
```

Figure 19: Import of Geoda data, Queen's matrix and conversion to matrix, libraries for implementing Bayesian CAR models and conversion of longitude and latitude coordinates.

Within this code there were map related to SIR were created but weren't used in the report due to space Figure 20.

```r
library(leaflet) #create map
colours <- colorNumeric(palette = "YlOrRd", domain = bvb.sp@data$SIR_TTBH)
SIR_map2 <- leaflet(data=bvb.sp) %>%
  addTiles() %>%
  addPolygons(fillColor = ~colours(SIR_TTBH), color="", weight=1, fillOpacity = 0.7) %>%
  addLegend(pal = colours, values = bvb.sp@data$SIR_TTBH, opacity = 1, title="SIR BVB Health") %>%
  addScaleBar(position="bottomleft")
SIR_map2

#if bymmatdep is >1 high <1 low
bvb.sp@data$SIR_TTBHCat <- ifelse(bvb.sp@data$SIR_TTBH > 1, "high", "low")
library(leaflet) #create map
pal <- colorFactor(palette = c("grey", "red"), levels = c("low", "high"))
SIR_TTBH_HL <- leaflet(bvb.sp) %>% addPolygons(stroke = FALSE, smoothFactor = 0.2, fillOpacity = 1,
              color = ~pal(SIR_TTBHCat))
SIR_TTBH_HL

#queens matrix - use file created from GeoDa
library(spdep)
queen.nb <- read.gal("TestSymmetric27.gal", override.id = TRUE)
W <- nb2mat(queen.nb, style="B")

#DISSIMILARITY AT LOCAL LEVEL BY lee and mitchell 2012 - fits localised model in poisson or gaussian
#measure of high mat deprivation
dep <- bvb.sp@data$HighMatDep
Z.dep <- as.matrix(dist(dep, diag=TRUE, upper=TRUE))
formula <- Y ~ offset(log(E))
depchain1 <- S.CARdissimilarity(formula=formula, data=bvb.sp@data, family="poisson", W=W, Z=list(Z.dep=Z.dep), W.binary=TRUE, burnin=10000, n.s

print(depchain1)
border.locations <- depchain1$localised.structure$W.posterior
bvb.sp@data$risk <- depchain1$fitted.values / bvb.sp@data$E
boundary.final <- highlight.borders(border.locations=border.locations, spdata=bvb.sp)
colours <- colorNumeric(palette = "YlOrRd", domain = bvb.sp@data$risk)
CARdissmap3 <- leaflet(data=bvb.sp) %>%
  addTiles() %>%
  addPolygons(fillColor = ~colours(risk), color="", weight=1,
              fillOpacity = 0.7) %>%
  addLegend(pal = colours, values = bvb.sp@data$risk, opacity = 1,
            title="Risk") %>%
  addCircles(lng = ~boundary.final$X, lat = ~boundary.final$Y, weight = 1,
             radius = 2) %>%
  addScaleBar(position="bottomleft")
CARdissmap3

##try gaussian
depchain1_gauss <- S.CARdissimilarity(formula=formula, data=bvb.sp@data, family="gaussian", W=W, Z=list(Z.dep=Z.dep),
                               W.binary=TRUE, burnin=10000, n.sample=40000, thin=20)

print(depchain1_gauss)
border.locations_gauss <- depchain1_gauss$localised.structure$W.posterior
```

Figure 20: Code illustration of maps and some models that weren't used.

There was a total of 40 Bayesian models applied to the dataset. Figure 21 illustrates the breakdown of the variables for the model and the set up of code per type of mode. Each group of models is labelled, and seed set to replicate results. There is also a function for this MCMC Poisson model to calculate the posterior values and the risk probability.

```r
library(GGally)
ggpairs(data = bvb.sp@data, columns = c(18:19, 22:27))

#import queen's matrix manually linked in GeoDa
queen.nb <- read.gal("TestSymmetric27.gal", override.id = TRUE)
W.ed <- nb2mat(queen.nb, style = "B")
W.ed.rs <- nb2mat(queen.nb, style = "W")

###Create models for various covariates
##BYM ref Besag et al, 1991.  Gaussian likelihood modelwith an identify link function or Poisson likelihood model with a log link function
#set parameters first
M.burnin <- 10000
M <- 40000
bvb.sp@data$logAGE0.49T <- log(bvb.sp@data$AGE0.49T)
bvb.sp@data$logAGE50.T <- log(bvb.sp@data$AGE50.T)

model1g <- Y ~ 1 + offset(log(E))
model2g <- Y ~ 1 + offset(log(E)) + logAGE0.49T + logAGE50.T
model3g <- Y ~ 1 + offset(log(E)) + logAGE0.49T + logAGE50.T + HighMatDep
model4g <- Y ~ 1 + offset(log(E)) + logAGE0.49T + logAGE50.T + AGEpc50.
model5g <- Y ~ 1 + offset(log(E)) + logAGE0.49T + logAGE50.T + HighMatDep + AGEpc50.
model6g <- Y ~ 1 + offset(log(E)) + HighMatDep
model7g <- Y ~ 1 + offset(log(E)) + AGEpc50.
model8g <- Y ~ 1 + offset(log(E)) + HighMatDep + AGEpc50.

##MCMC BYM POISSON
set.seed(1234)
mcmcbym_model1g <- S.CARbym(formula = model1g, data=bvb.sp@data, family = "poisson", W=W.ed, burnin = M.burnin, n.sample = M, verbose = FALSE)
mcmcbym_model2g <- S.CARbym(formula = model2g, data=bvb.sp@data, family = "poisson", W=W.ed, burnin = M.burnin, n.sample = M, verbose = FALSE)
mcmcbym_model3g <- S.CARbym(formula = model3g, data=bvb.sp@data, family = "poisson", W=W.ed, burnin = M.burnin, n.sample = M, verbose = FALSE)
mcmcbym_model4g <- S.CARbym(formula = model4g, data=bvb.sp@data, family = "poisson", W=W.ed, burnin = M.burnin, n.sample = M, verbose = FALSE)
mcmcbym_model5g <- S.CARbym(formula = model5g, data=bvb.sp@data, family = "poisson", W=W.ed, burnin = M.burnin, n.sample = M, verbose = FALSE)
mcmcbym_model6g <- S.CARbym(formula = model6g, data=bvb.sp@data, family = "poisson", W=W.ed, burnin = M.burnin, n.sample = M, verbose = FALSE)
mcmcbym_model7g <- S.CARbym(formula = model7g, data=bvb.sp@data, family = "poisson", W=W.ed, burnin = M.burnin, n.sample = M, verbose = FALSE)
mcmcbym_model8g <- S.CARbym(formula = model8g, data=bvb.sp@data, family = "poisson", W=W.ed, burnin = M.burnin, n.sample = M, verbose = FALSE)

summary(mcmcbym_model1g)

bvb.sp$mcmcbym_model4g_fit <- mcmcbym_model4g$fitted.values
bvb.sp@data$
# Compute posterior SIR
y.fit <- mcmcbym_model4g$samples$fitted
SIR <- t(t(y.fit) / bvb.sp@data$E)

# Add summary statistics of the posterior SIR to map
bvb.sp@data$mcmcbym_model4g_SIR.median <- apply(SIR, 2, median) #denotes the releative risk of smooth sir
min(bvb.sp@data$mcmcbym_model4g_SIR.median)
max(bvb.sp@data$mcmcbym_model4g_SIR.median)

mcmcbym_model4g_SIR.025 <- apply(SIR, 2, quantile, 0.025)
mcmcbym_model4g_SIR.975 <- apply(SIR, 2, quantile, 0.975)
bvb.sp@data$mcmcbym_model4g_PP <- apply(SIR, 2, function(x) length(which(x > 1))) / M #thiis is the posterior probability of the relative risk
min(bvb.sp@data$mcmcbym_model4g_PP)
max(bvb.sp@data$mcmcbym_model4g_PP)

bvb.sp@data$mcmcbym_model4g_PPCat <- ifelse(bvb.sp@data$mcmcbym_model4g_PP > 0.5, "high", "low")
bvb.sp@data$mcmcbym_model4g_SIR.medianCat <- ifelse(bvb.sp@data$mcmcbym_model4g_SIR.median > 1, "high", "low")

library(leaflet) #create map
```

Figure 21: Code illustrating set up for the models and BYM MCMC Poisson model.

Figure 22 illustrates the results from the MCMC BYM Poisson model.

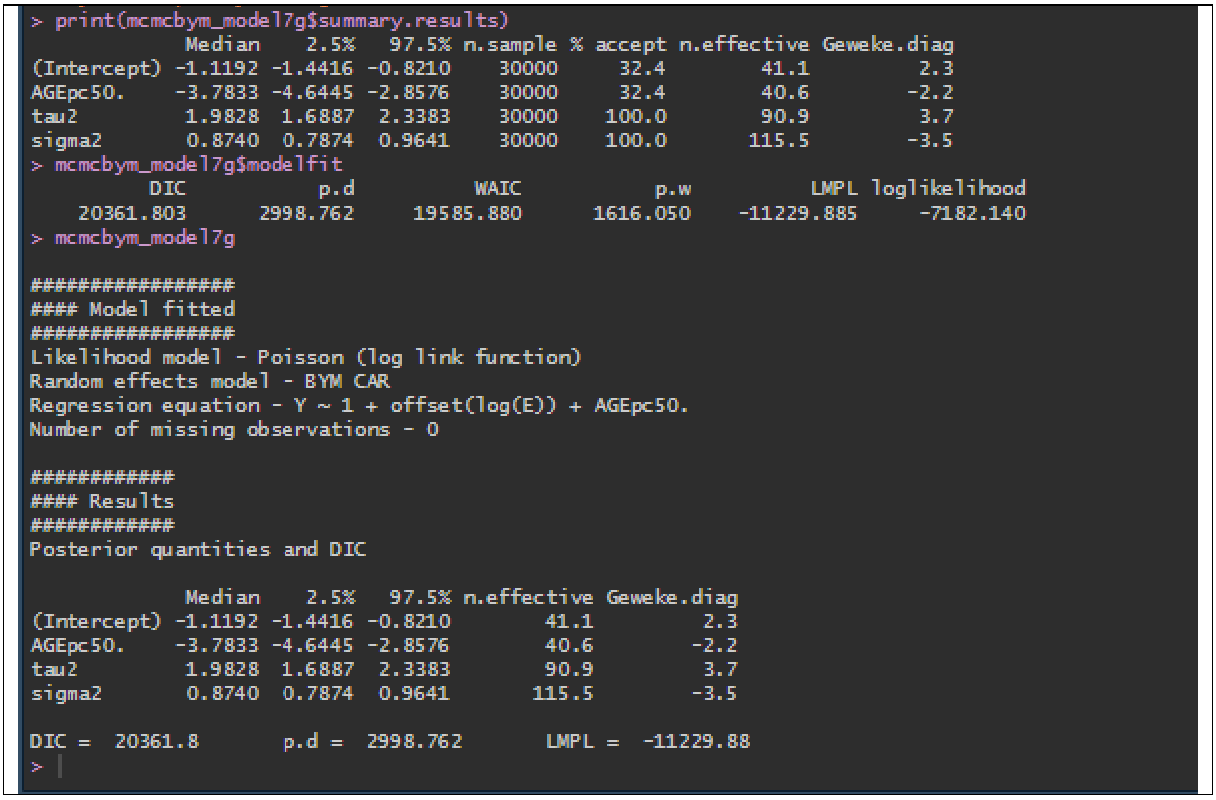


Figure 22: Results of BYM MCMC Poisson Model.

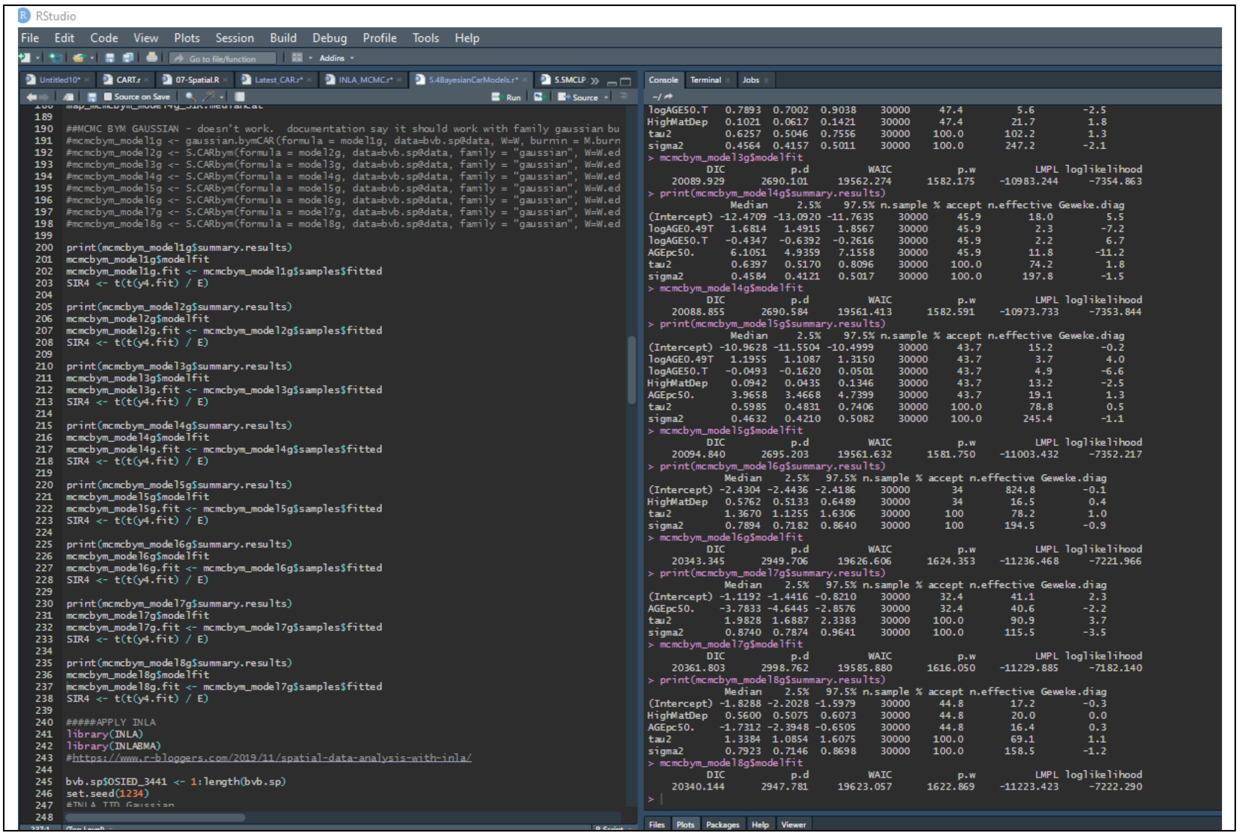


Figure 23: Code and summary results of MCMC BYM Poisson model.

Post MCMC BYM Poisson model, the INLA models were implemented. See figure 24 for code of IID model and BYM model with gaussian. There is also code to create a map of the posterior values.

```r
library(INLA)
library(INLABMA)
#https://www.r-bloggers.com/2019/11/spatial-data-analysis-with-inla/

bvb.sp$OSIED_3441 <- 1:length(bvb.sp)
set.seed(1234)
#INLA IID Gaussian
inla_model1g <- inla(update(model1g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model2g <- inla(update(model2g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model3g <- inla(update(model3g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model4g <- inla(update(model4g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model5g <- inla(update(model5g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model6g <- inla(update(model6g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model7g <- inla(update(model7g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model8g <- inla(update(model8g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))

summary(inla_model1g)
summary(inla_model2g)
summary(inla_model3g)
summary(inla_model4g)
summary(inla_model5g)
summary(inla_model6g)
summary(inla_model7g)
summary(inla_model8g)


bvb.sp$inla_model7g_mean <- inla_model7g$summary.fitted.values[, "mean"] #prosterior values
summary(bvb.sp$inla_model7g_mean)
bvb.sp@data$inla_model7g_meanCat <- ifelse(bvb.sp@data$inla_model7g_mean > 100, "high", "low")

library(leaflet) #create map
pal <- colorFactor(palette = c("grey", "tomato"), levels = c("low", "high"))
map_inla_model7g_meanCat <- leaflet(bvb.sp) %>%
  addPolygons(stroke = FALSE, smoothFactor = 0.2, fillOpacity = 1,
              color = ~pal(inla_model7g_meanCat)) %>%
  addLegend(pal = pal, values = bvb.sp@data$inla_model7g_meanCat, opacity = 1,
            title="Posterior Value") %>%
  addScaleBar(position="bottomleft")
map_inla_model7g_meanCat

set.seed(1234)
##INLA BYM GAUSSIAN MODELS
inlabym_model1g <- inla(update(model1g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model2g <- inla(update(model2g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model3g <- inla(update(model3g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model4g <- inla(update(model4g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model5g <- inla(update(model5g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model6g <- inla(update(model6g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model7g <- inla(update(model7g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model8g <- inla(update(model8g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="gaussian", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))

summary(inlabym_model1g)
summary(inlabym_model2g)
summary(inlabym_model3g)
summary(inlabym_model4g)
summary(inlabym_model5g)
summary(inlabym_model6g)
summary(inlabym_model7g)
summary(inlabym_model8g)
```

Figure 24: Code to implement INLA IID and BYM Gaussian models

```
  Name    Model
    OSIED_3441 BYM model

Model hyperparameters:
                                            mean   sd 0.025quant 0.5quant 0.975quant  mode
Precision for the Gaussian observations      Inf  NaN      0.000    0.000        Inf   NaN
Precision for OSIED_3441 (iid component)   0.003 0.00      0.003    0.003      0.003 0.003
Precision for OSIED_3441 (spatial component) Inf  NaN      0.000    0.000        Inf   NaN

Expected number of effective parameters(stdev): 3386.00(2.53)
Number of equivalent replicates : 1.01

Deviance Information Criterion (DIC) ...............: 15319.27
Deviance Information Criterion (DIC, saturated) ....: 6519.94
Effective number of parameters .....................: 3248.81

Watanabe-Akaike information criterion (WAIC) ...: 14443.90
Effective number of parameters .................: 1715.03

Marginal log-Likelihood:  -13984.91
CPO and PIT are computed

Posterior marginals for the linear predictor and
 the fitted values are computed

> summary(inlabym_model6g)

Call:
   c("inla(formula = update(model6g, . ~ . + f(OSIED_3441, model = \"bym\", ", " graph = W.ed)), family = \"gaussian\", data = as.data.frame(bvb.sp@data), ", " E = E,
   control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), ", " control.predictor = list(compute = TRUE))")
Time used:
    Pre = 1.25, Running = 20, Post = 0.633, Total = 21.9
Fixed effects:
              mean    sd 0.025quant 0.5quant 0.975quant   mode kld
(Intercept) 17.727 0.387     16.967   17.727     18.486 17.727   0
HighMatDep  30.612 0.387     29.852   30.612     31.371 30.612   0

Random effects:
  Name    Model
    OSIED_3441 BYM model

Model hyperparameters:
                                               mean     sd 0.025quant 0.5quant 0.975quant   mode
Precision for the Gaussian observations       0.635  0.235      0.249    0.614      1.147  0.556
Precision for OSIED_3441 (iid component)      0.002  0.000      0.002    0.002      0.002  0.002
Precision for OSIED_3441 (spatial component) 68.461 14.951     43.756   66.884    102.287 63.851

Expected number of effective parameters(stdev): 3399.51(4.50)
Number of equivalent replicates : 1.00

Deviance Information Criterion (DIC) ...............: 13779.73
Deviance Information Criterion (DIC, saturated) ....: 6404.20
Effective number of parameters .....................: 3197.22

Watanabe-Akaike information criterion (WAIC) ...: 13080.75
Effective number of parameters .................: 1831.38

Marginal log-Likelihood:  -14917.35
CPO and PIT are computed

Posterior marginals for the linear predictor and
 the fitted values are computed

> summary(inlabym_model7g)

Call:
```

Figure 25: Output of models specifically INLA BYM models 7g and 6g

The next models created were the IID and BYM Poisson models which follows the same pattern of code as the previous models. See figure 26 and 27.

```r
           title="Posterior Value") %>%
  addScaleBar(position="bottomleft")
map_inlabym_model6g_meanCat

set.seed(1234)
##INLA IID POISSON
inla_model1p <- inla(update(model1g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model2p <- inla(update(model2g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model3p <- inla(update(model3g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model4p <- inla(update(model4g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model5p <- inla(update(model5g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model6p <- inla(update(model6g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model7p <- inla(update(model7g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inla_model8p <- inla(update(model8g, . ~. + f(OSIED_3441, model = "iid")), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))

summary(inla_model1p)
summary(inla_model2p)
summary(inla_model3p)
summary(inla_model4p)
summary(inla_model5p)
summary(inla_model6p)
summary(inla_model7p)
summary(inla_model8p)

bvb.sp$inla_model2p_mean <- inla_model2p$summary.fitted.values[, "mean"]
summary(bvb.sp$inla_model2p_mean)
plot(bvb.sp$inla_model2p_mean)
bvb.sp$inla_model2p_meanCat <- ifelse(bvb.sp$inla_model2p_mean > 2, "high", "low")

library(leaflet) #create map
pal <- colorFactor(palette = c("grey", "tomato"), levels = c("low", "high"))
map_inla_model2p_meanCat <- leaflet(bvb.sp) %>%
  addPolygons(stroke = FALSE, smoothFactor = 0.2, fillOpacity = 1,
              color = ~pal(inla_model2p_meanCat)) %>%
  addLegend(pal = pal, values = bvb.sp@data$inla_model2p_meanCat, opacity = 1,
            title="Posterior Value") %>%
  addScaleBar(position="bottomleft")
map_inlabym_model6g_meanCat

set.seed(1234)
##INLA BYM POISSON MODELS
inlabym_model1p <- inla(update(model1g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model2p <- inla(update(model2g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model3p <- inla(update(model3g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model4p <- inla(update(model4g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model5p <- inla(update(model5g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model6p <- inla(update(model6g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model7p <- inla(update(model7g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))
inlabym_model8p <- inla(update(model8g, . ~. + f(OSIED_3441, model = "bym", graph = W.ed)), data = as.data.frame(bvb.sp@data), E = E, family ="poisson", control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE), control.predictor = list(compute = TRUE))

summary(inlabym_model1p)
summary(inlabym_model2p)
summary(inlabym_model3p)
summary(inlabym_model4p)
summary(inlabym_model5p)
summary(inlabym_model6p)
summary(inlabym_model7p)
summary(inlabym_model8p)
```

Figure 26: INLA BYM and IID Poisson Models

```
  Name    Model
    OSIED_3441 BYM model

Model hyperparameters:
                                          mean    sd 0.025quant 0.5quant 0.975quant  mode
Precision for OSIED_3441 (iid component)     0.614 0.027      0.564    0.613      0.670 0.611
Precision for OSIED_3441 (spatial component) 0.457 0.048      0.370    0.454      0.557 0.449

Expected number of effective parameters(stdev): 3132.41(5.72)
Number of equivalent replicates : 1.09

Deviance Information Criterion (DIC) ...............: 20772.91
Deviance Information Criterion (DIC, saturated) ....: 6826.73
Effective number of parameters .....................: 3260.66

Watanabe-Akaike information criterion (WAIC) ...: 20671.53
Effective number of parameters .................: 2158.48

Marginal log-Likelihood:  -13552.17
CPO and PIT are computed

Posterior marginals for the linear predictor and
 the fitted values are computed

> summary(inlabym_model4p)

Call:
   c("inla(formula = update(model4g, . ~ . + f(OSIED_3441, model = \"bym\", ", " graph = W.ed)), family = \"poisson\", data = as.data.frame(bvb.sp@data), ", " E = E, control.compute
   = list(dic = TRUE, waic = TRUE, cpo = TRUE), ", " control.predictor = list(compute = TRUE))")
Time used:
    Pre = 1.21, Running = 87.2, Post = 0.602, Total = 89
Fixed effects:
               mean    sd 0.025quant 0.5quant 0.975quant    mode kld
(Intercept) -15.817 1.841    -19.432  -15.817    -12.205 -15.818   0
logAGE0.49T   0.966 0.795     -0.596    0.966      2.525   0.966   0
logAGE50.T    0.327 0.794     -1.232    0.327      1.885   0.326   0
AGEpc50.      2.406 3.749     -4.958    2.406      9.759   2.408   0

Random effects:
  Name    Model
    OSIED_3441 BYM model

Model hyperparameters:
                                          mean    sd 0.025quant 0.5quant 0.975quant  mode
Precision for OSIED_3441 (iid component)     0.615 0.027      0.564    0.614      0.671 0.612
Precision for OSIED_3441 (spatial component) 0.455 0.047      0.369    0.452      0.555 0.447

Expected number of effective parameters(stdev): 3132.09(5.73)
Number of equivalent replicates : 1.09

Deviance Information Criterion (DIC) ...............: 20772.26
Deviance Information Criterion (DIC, saturated) ....: 6826.08
Effective number of parameters .....................: 3260.46

Watanabe-Akaike information criterion (WAIC) ...: 20670.66
Effective number of parameters .................: 2158.18

Marginal log-Likelihood:  -13547.78
CPO and PIT are computed

Posterior marginals for the linear predictor and
 the fitted values are computed

> summary(inlabym_model5p)

Call:
```

Figure 27: Sample of results from INLA BYM and IID Poisson models

Post model creation, the fitted values were applied to the map of Ireland for each of the 5 models. Figure 28 shows the code in action and Figure 29 the final map. This was not included in the final report due to space.

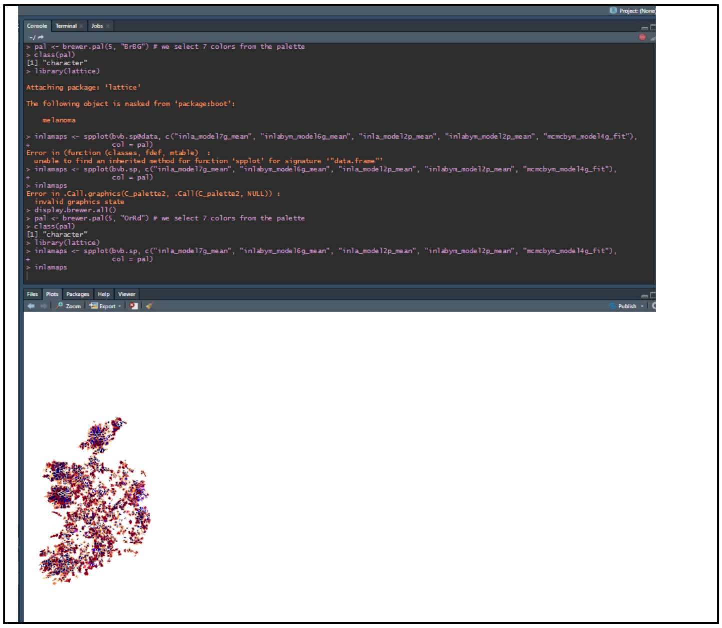


Figure 28: Code for 5 maps being created that illustrate the distribution of Posterior values of models

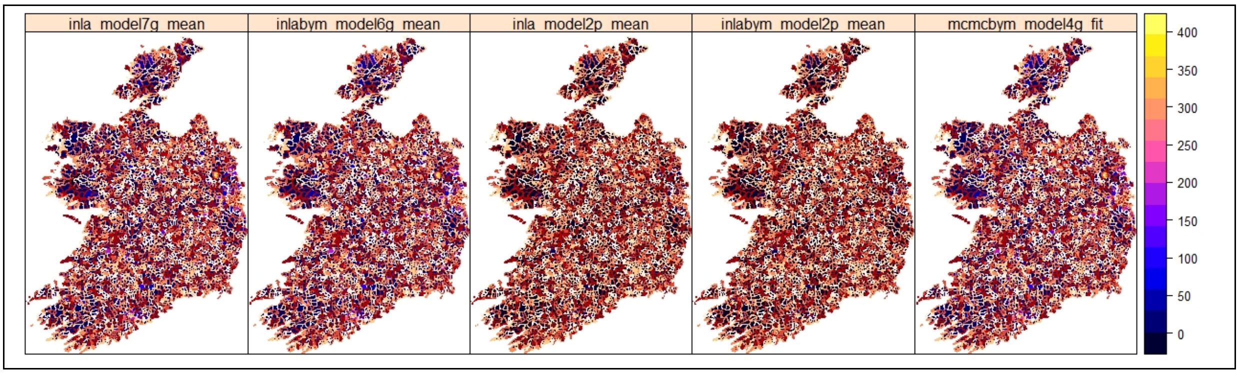


Figure 29: Final maps of electoral division with posterior values of each optimal model.

The relevant data related to cateogrical data that was used to create the maps of the posterior values were saved to the csv file "finalmodels.csv". This information was used to construct table 3 in the document illustrating the 56 common electoral divisions from the models.

5.5 MCLP

A clean shapefile of electoral divisions and the road network was imported into R. Both files were filtered to Offaly with the correct coordinates attached to both. The road network, a spatial lines file, was filtered to the residential road network for Offaly Figure 9 & 30.

```r
##example taken from https://rpubs.com/Billy_Archbold/Billy_Archbold
# Load initial packages
library(sp)
library(rgdal)
library(GISTools)
library(rgeos)
library(rgdal)

##Boundary files
# Read Electoral Divisions counties shapefile into R
County <- readOGR(dsn = '.', layer = "CAR.o4ctb")

####OFFALY
Offaly <- County[County@data$COUNTY == "OFFALY", ]
offaly_proj <- proj4string(Offaly)
Offaly_Buf <- gBuffer(Offaly, width = 100)
# Read in the 'roads' dataset. This takes roughly 5-10 mins
road <- readOGR(dsn = '.', "gis_osm_roads_free_1")
roadTemp <- road
road <- spTransform(roadTemp, offaly_proj)
offalyroad <- gIntersection(Offaly_Buf, road, byid = TRUE) #intersection
offalytmp <- names(offalyroad)
offalytmp <- strsplit(offalytmp, " ")
offalyroad.id <- (sapply(offalytmp, "[[", 2))
offalyroaddf <- data.frame(road[offalyroad.id, ])
rownames(offalyroaddf) <- paste("buffer", rownames(offalyroaddf), sep = " ")
offalyroad.sldf <- SpatialLinesDataFrame(offalyroad, data = offalyroaddf)
levels(offalyroad.sldf@data$fclass)
# There are 26 levels in Road Network, and 32 overall. To plot
# different road types:
offalyresx <- offalyroad.sldf[offalyroad.sldf@data$fclass == "residential", ]
class(offalyresx)
plot(offalyresx, col = "darkred")
title(main = "Offaly Residential Road Segments", font.main = 2, cex.main = 1.5)

#####FINS COORDINATES NORTH, SOUTH, EAST AND WEST TO THE BOUNDARY OF EACH ELECTORAL DIVISION
##find coordinates of n,e,w,s on boundary for each electoral division
cardinal <- function(n, sep){
  ### generate n points separated by sep in the NSEW directions from (0,0)
  delta = seq(sep, length.out=n, by=sep)
  N = cbind(0, delta)
  S = cbind(0, -delta)
  E = cbind(delta, 0)
  W = cbind(-delta, 0)
  rbind(N,S,E,W)
}

addoff <- function(pt, offs){
  ### add the offsets in `offs` to the point in `pt`:
  offs[,1] = offs[,1]+pt[,1]
  offs[,2] = offs[,2]+pt[,2]
  offs
}

#Then the main function that takes an `sf` spatial data frame and a pattern of cardinal points:

nsew <- function(geoms, cps, messages=FALSE){
  allpts = lapply(1:nrow(geoms), function(i){
    if(messages)message("region ",i)
```

Figure 30: Code for road network filter for Offaly with plot for road network and coordinates for North, South, East and West.

The coordinates for potential AED locations were sought using https://gis.stackexchange.com/questions/407535/calculate-distance-from-centroid-to-border-of-spatial-polygon-in-r Figure 31.

```
  offs[,1] = offs[,1]+pt[,1]
  offs[,2] = offs[,2]+pt[,2]
  offs
}

#Then the main function that takes an `sf` spatial data frame and a pattern of cardinal points:

nsew <- function(geoms, cps, messages=FALSE){
  allpts = lapply(1:nrow(geoms), function(i){
    if(messages)message("region ",i)
    geom = st_geometry(geoms)[i]
    pts = addoff(st_coordinates(st_centroid(geom)), cps)
    pts = st_as_sf(data.frame(pts), coords=1:2, crs=st_crs(geoms))
    inside = which(lengths(st_intersects(pts, geom))==1)
    if(length(inside)==0){
      pts = numeric(0)
    }else{
      pts = pts[inside,]
    }
    pts
  })
  allpts
}

sizeneeded <- function(regions, w){
  boxes = data.frame(t(sapply(st_geometry(regions), st_bbox)))
  widths = boxes$xmax - boxes$xmin
  heights = boxes$ymax - boxes$ymin
  maxbox = max(c(widths, heights))
  n = round(1 + (maxbox/w))
  n
}

# read the electoral divisions
library(sf)
ed = st_read("./CAR.o4ctb.shp")
#filter(ed@data$OBJECTID == 1001)
class(ed)
attr(ed, "geometry")
methods(class = "sf")
# how many points max do we need?
ns = sizeneeded(ed,400)
print(ns) # its 61

# create the points
cps = cardinal(ns,400)

# loop over all EDs and find the cardinal points in `cps` that
# centred on the centroid fall in the ED. This may take a few minutes:

pts = nsew(ed, cps)
plot(ed$geometry[1768])
plot(pts[[1768]], add=TRUE)
pts[[1768]]

plot(ed$geometry[80])
axis(1)
axis(2)
pts[[80]]
numeric(0)

library(sf)
library(rgdal)
#create long lat points in spatial list
pts[[1768]] <- st_transform(pts[[1768]], CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))
pts[[1265]] <- st_transform(pts[[1265]], CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))
pts[[2810]] <- st_transform(pts[[2810]], CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))

##OFFALY AT RISK - TULLAMORE URBAN, BIRR URBAN AND EDENDERRY###
offalyatrisked <- do.call(rbind, list(pts[[1265]], pts[[1768]], pts[[2810]]))
```

Figure 31: Code for coordinates to the border of every electoral division.

A function was created to determine the points from the coordinates to the boundary north, south, east and west. These points were then allocated per boundary in a list with point file for each 3409 boundaries. The points were then converted to longitude and latitude coordinates. The ID number (1768) was referenced to the original shapefile to find Birr, the area to test the AED framework. The list was converted to a matrix and the correct names of the variables added (long/lat) with an ID list Figure 31 & 32.

```r
pts[[1768]] <- st_transform(pts[[1768]], CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))
pts[[1265]] <- st_transform(pts[[1265]], CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))
pts[[2810]] <- st_transform(pts[[2810]], CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))

##OFFALY AT RISK - TULLAMORE URBAN, BIRR URBAN AND EDENDERRY###
offalyatrisked <- do.call(rbind, list(pts[[1265]], pts[[1768]], pts[[2810]]))
offalyatrisked_df <- data.frame(matrix(unlist(offalyatrisked), nrow = 38, byrow = TRUE, ncol = 2), stringsAsFactors = FALSE)
library(tidyverse)
offalyatrisked_df <- offalyatrisked_df %>% rename(long = X1, lat = X2)
offalyatrisked_df <- offalyatrisked_df[c(2, 1)]
write.csv(offalyatrisked_df, "offalyatrisked_df.csv")
offalyatrisked_df <- st_as_sf(offalyatrisked_df, coords = c("long", "lat"), crs = "+proj=longlat +datum=WGS84 +no_defs")
offalyatrisked_df <- st_transform(offalyatrisked_df, crs = "+proj=longlat +datum=WGS84 +no_defs")

AEDpotential <- tm_shape(offalyatrisked_df) + tm_dots()
class(AEDpotential)
library(tmap)
mapOffaly = tm_shape(Offaly) + tm_polygons()
class(mapOffaly)

map2Offaly = mapOffaly + tm_shape(offalyresx) + tm_lines() + tm_compass(type = "8star", position = c("left", "top")) +
  tm_scale_bar(breaks = c(0, 10, 20), text.size = 1)

map3Offaly = map2Offaly + AEDpotential +   tm_compass(type = "8star", position = c("left", "top")) +
  tm_scale_bar(breaks = c(0, 10, 20), text.size = 1)

map3Offaly2 = map2Offaly + AEDpotential

library(leaflet)
library(leaflet.extras)
library(mapedit)
library(ggmap)
library(mapview)
library(raster)
library(magrittr)
map3Offaly <- leaflet() %>%
  addTiles() %>%
  addPolygons(data = Offaly) #%>%
addPolylines(data = offalyresx)# %>%
addCircles(data = offalyatrisked_df,
           radius = 200,
           stroke = TRUE,
           fill = NULL,
           opacity = 0.8,
           weight = 2,
           color = "green")
map4Offaly = map2Offaly + map3Offaly
class(map4Offaly)

testmapOffaly <- tmap_leaflet(map3Offaly2) %>%
  addDrawToolbar()

??addDrawToolbar

class(Offaly)
tesetmap <- leaflet(Offaly) %>% addPolygons(stroke = FALSE, smoothFactor = 0.2, fillOpacity = 1)

library(mapview)
library(mapedit)
what_we_created <- mapview() %>%
  editMap()

mapview(what_we_created$finished)

#BIRR only
birratrisked <- do.call(rbind, list(pts[[1768]]))
birratrisked_df <- data.frame(matrix(unlist(birratrisked), nrow = 12, byrow = TRUE, ncol = 2), stringsAsFactors = FALSE)
library(tidyverse)
birratrisked_df <- birratrisked_df %>% rename(long = X1, lat = X2)
birratrisked_df <- birratrisked_df[c(2, 1)]
```

Figure 32: Code for coordinates to the border of every electoral division and filtering to Birr

In Figure 33 the road network and electoral divisions were overlaid to find the coordinates on the residential road network which is in highlighted in orange.

```r

library(mapview)
library(mapedit)
what_we_created <- mapview() %>%
  editMap()

mapview(what_we_created$finished)

#BIRR only
birratrisked <- do.call(rbind, list(pts[[1768]]))
birratrisked_df <- data.frame(matrix(unlist(birratrisked), nrow = 12, byrow = TRUE, ncol = 2), stringsAsFactors = FALSE)
library(tidyverse)
birratrisked_df <- birratrisked_df %>% rename(long = X1, lat = X2)
birratrisked_df <- birratrisked_df[c(2, 1)]
birratrisked_df <- st_as_sf(birratrisked_df, coords = c("long", "lat"), crs = "+proj=longlat +datum=WGS84 +no_defs")
birratrisked_df <- st_transform(birratrisked_df, crs = "+proj=longlat +datum=WGS84 +no_defs")


NEWSbirr <- st_as_sfc(birratrisked_df)
plot(birratrisked_df)

test <- st_as_sfc(offalyresx)
plot(test)

test2 <- st_as_sfc(Offaly)
plot(test2)

test3 <- leaflet(test2) %>%
  addTiles() %>%
  addPolylines(data = test) %>%
  addPolygons(data = test2,
              color = '#0FF')



test3 <- mapview(test) %>%
  editMap()

test4 <- mapview(test2) %>%
  editMap()

mapview(test3$finished)
mapviewOutput(test3$finished)
as.data.frame(test5)
print(test5)
write.csv(test5, "test5.csv")
class(test5)

test6 <- mapview(list(test, test2), layer.name = c("roads", "ed")) %>% editMap()
mapview(test6$finished)
test7 <- mapview(test6) + mapview(test5) %>% editMap()

AED_user_long <- c(-7.91372, -7.90494, -7.89343, -7.89890, -7.91596, -7.91625, -7.91372, -7.91273, -7.91110, -7.90809,
                   -7.90582, -7.90432, -7.89951, -7.90260, -7.89951, -7.90509, -7.90724, -7.90719, -7.90895, -7.91162,
                   -7.91428, -7.91200, -7.91063,-7.90147, -7.90216, -7.90332, -7.89748, -7.90718, -7.91345, -7.91212,
                   -7.91431, -7.90941, -7.90173, -7.90336, -7.90658, -7.91277, -7.91105, -7.90774, -7.91053, -7.91315,
                   -7.91551, -7.90998, -7.91144, -7.90212, -7.89954, -7.89988, -7.90023, -7.89766, -7.89971,
                   -7.90366, -7.90568, -7.90676, -7.89769, -7.90018, -7.90125, -7.90396)
AED_user_lat <-  c(53.10603, 53.10452, 53.10648, 53.10125, 53.10331, 53.10249, 53.10285, 53.10213, 53.10226, 53.10187,
                   53.10218, 53.10136, 53.09960, 53.10058, 53.09965, 53.10022, 53.10084, 53.09971, 53.09873, 53.09996,
                   53.09927, 53.09813, 53.09703, 53.09801, 53.09698, 53.09610, 53.09460, 53.09653, 53.09646, 53.09612,
                   53.09399, 53.09319, 53.09338, 53.09219, 53.09204, 53.09224, 53.09108, 53.09064, 53.09005, 53.09006,
                   53.08935, 53.08874, 53.08547, 53.09100, 53.09080, 53.08843, 53.08716, 53.08734, 53.08601,
                   53.08748, 53.08748, 53.08637, 53.10466, 53.10304, 53.08299, 53.08454)

AED_user <- cbind(AED_user_lat, AED_user_long)

```

Figure 33: Code to select coordinates for OHCA location in the residential area

The next step (Figure 34) is to create points in every direction to the border based on the number of 400 metre points to the border from Figure 31.

```r
b <- 270
d <- 400
a <- 6378137
f <- 1/298.257223563
r <- 6378137
Birr_PotentialAEDs$West <- destPoint(p, b, d, a, f)

#3 points west of centroid
plot(birratrisked_df)
numberOfIterations = 3 #Change That as needed
westOutput = list()
westOutput[[1]] = destPoint(p[1,],b[1],d[1],a,f)
for(i in 2:numberOfIterations) {
  westOutput[[i]] = destPoint(westOutput[[i-1]],b[1],d[1])
}

birrwest <- matrix(unlist(westOutput), ncol = 2, byrow = TRUE)

#first point west, then north
p <- cbind(c(-7.912423), c(53.09579))
b <- 360
numberOfIterations = 3 #Change That as needed
n1Output = list()
n1Output[[1]] = destPoint(p[1,],b[1],d[1],a,f)
for(i in 2:numberOfIterations) {
  n1Output[[i]] = destPoint(n1Output[[i-1]],b[1],d[1])
}

birrn1 <- matrix(unlist(n1Output), ncol = 2, byrow = TRUE)

#second point west, then north
p <- cbind(c(-7.918394), c(53.09579))
b <- 360
numberOfIterations = 3 #Change That as needed
n2Output = list()
n2Output[[1]] = destPoint(p[1,],b[1],d[1],a,f)
for(i in 2:numberOfIterations) {
  n2Output[[i]] = destPoint(n2Output[[i-1]],b[1],d[1])
}

birrn2 <- matrix(unlist(n2Output), ncol = 2, byrow = TRUE)

#third point west, then north
p <- cbind(c(-7.924365), c(53.09579))
b <- 360
numberOfIterations = 3 #Change That as needed
n3Output = list()
n3Output[[1]] = destPoint(p[1,],b[1],d[1],a,f)
for(i in 2:numberOfIterations) {
  n3Output[[i]] = destPoint(n3Output[[i-1]],b[1],d[1])
}

birrn3 <- matrix(unlist(n3Output), ncol = 2, byrow = TRUE)

#first point west, then south
p <- cbind(c(-7.912423), c(53.09579))
b <- 180
numberOfIterations = 3 #Change That as needed
s1Output = list()
s1Output[[1]] = destPoint(p[1,],b[1],d[1],a,f)
for(i in 2:numberOfIterations) {
  s1Output[[i]] = destPoint(s1Output[[i-1]],b[1],d[1])
}

birrs1 <- matrix(unlist(s1Output), ncol = 2, byrow = TRUE)

#second point west, then south
p <- cbind(c(-7.918394), c(53.09579))
b <- 180
numberOfIterations = 3 #Change That as needed
s2Output = list()
```

Figure 34: Code to select coordinates for potential AED location to boundary.

The final coordinates were saved as a list with do.call code to identify the longitude and latitude coordinates.

```
birrsouth <- matrix(unlist(southOutput), ncol = 2, byrow = TRUE)

birrlist <- list(birrsouth, birrnorth, birrse3, birrse2, birrse1, birrne3, birrne2, birrne1,  birreast,  birrs3,  birrs2,  birrs1,  birrn3, b
birr_aed_potential <- do.call(rbind, birrlist)
birr_aed_potential_df <- as.data.frame(birr_aed_potential)
birr_aed_potential_df <- birr_aed_potential_df %>% rename(long = V1, lat = V2)
birr_aed_potential_df$ID <- seq.int(nrow(birr_aed_potential_df))
birr_aed_potential_df <- birr_aed_potential_df[, c(2, 1)]
##centroid
birr_centroid <- cbind(Birr_PotentialAEDs$long, Birr_PotentialAEDs$lat)
birr_centroid_df <- as.data.frame(birr_centroid)
birr_centroid_df <- birr_centroid_df %>% rename(long = V1, lat = V2)
birr_centroid_df <- birr_centroid_df[, c(2, 1)]
birr_centroid_df$ID <- seq.int(nrow(birr_centroid_df))
#aed ohca points
AED_user_df <- as.data.frame(AED_user)
AED_user_df <- AED_user_df %>%
  rename(long = AED_user_long, lat = AED_user_lat)
AED_user_df$ID <- seq.int(nrow(AED_user_df))
```

Figure 35: Do.call function to combine all the created coordinate lists in various directions.

The maxcovr package is not on cran in R and will need to be downloaded using the following code from Figure 36. Within this figure is also the distance calculation for the matrix for MCLP,

```
# install.packages("devtools")
devtools::install_github("njtierney/maxcovr")
library(maxcovr)
library(dplyr)
dat_dist <- birr_centroid_df %>% nearest(AED_user_df)
write.csv(dat_dist, "datdist.csv")
min(dat_dist$distance) #95.60513
max(dat_dist$distance) #1472.684
dat_dist_bldg <- AED_user_df %>% nearest(birr_centroid_df)
head(dat_dist_bldg)
centroidusercoverage <- coverage(birr_centroid_df, AED_user_df)
```

Figure 36: Code to install maxcovr library which is not on cran for MCLP

The final part is the application of MCLP which is illustrated in Figure 37 for each of the distances of 100 metres to 400 metres.

```r

library(purrr)
n_add_vec <- c(5, 10, 15, 20, 25, 30)

system.time(
  map_mc_model <- map_df(.x = n_add_vec,
                         .f = ~max_coverage(existing_facility = birr_centroid_df,
                                            proposed_facility = birr_aed_potential_df,
                                            user = AED_user_df,
                                            distance_cutoff = 100,
                                            n_added = .))
)

map_cov_results <- bind_rows(map_mc_model$model_coverage)

system.time(
  map_mc_model15 <- map_df(.x = n_add_vec,
                           .f = ~max_coverage(existing_facility = birr_centroid_df,
                                              proposed_facility = birr_aed_potential_df,
                                              user = AED_user_df,
                                              distance_cutoff = 150,
                                              n_added = .))
)

map_cov_results15 <- bind_rows(map_mc_model15$model_coverage)
summary(map_mc_model15)

system.time(
  map_mc_model2 <- map_df(.x = n_add_vec,
                          .f = ~max_coverage(existing_facility = birr_centroid_df,
                                             proposed_facility = birr_aed_potential_df,
                                             user = AED_user_df,
                                             distance_cutoff = 200,
                                             n_added = .))
)

map_cov_results2 <- bind_rows(map_mc_model2$model_coverage)

system.time(
  map_mc_model3 <- map_df(.x = n_add_vec,
                          .f = ~max_coverage(existing_facility = birr_centroid_df,
                                             proposed_facility = birr_aed_potential_df,
                                             user = AED_user_df,
                                             distance_cutoff = 300,
                                             n_added = .))
)

map_cov_results3 <- bind_rows(map_mc_model3$model_coverage)

system.time(
  map_mc_model4 <- map_df(.x = n_add_vec,
                          .f = ~max_coverage(existing_facility = birr_centroid_df,
                                             proposed_facility = birr_aed_potential_df,
                                             user = AED_user_df,
                                             distance_cutoff = 400,
                                             n_added = .))
)

map_cov_results4 <- bind_rows(map_mc_model4$model_coverage)


library(ggplot2)
bind_rows(map_mc_model$existing_coverage[[1]],
          map_cov_results) %>%
  ggplot(aes(x = factor(n_added),
```

Figure 37: Code to apply MCLP to the datasets created for potential aed locations

```
Console  Terminal ×  Jobs ×
~/
>
> map_cov_results3 <- bind_rows(map_mc_model3$model_coverage)
>
> system.time(
+   map_mc_model4 <- map_df(.x = n_add_vec,
+                           .f = ~max_coverage(existing_facility = birr_centroid_df,
+                                              proposed_facility = birr_aed_potential_df,
+                                              user = AED_user_df,
+                                              distance_cutoff = 400,
+                                              n_added = .))
+ )
   user  system elapsed
   0.67    0.02    0.72
>
> map_cov_results4 <- bind_rows(map_mc_model4$model_coverage)
> map_cov_results3
# A tibble: 6 x 8
  n_added distance_within n_cov pct_cov n_not_cov pct_not_cov dist_avg dist_sd
    <dbl>           <dbl> <int>   <dbl>     <int>       <dbl>    <dbl>   <dbl>
1       5             300    32   0.571        24      0.429      305.   180.
2      10             300    47   0.839         9      0.161      214.   119.
3      15             300    54   0.964         2      0.0357     174.    90.3
4      20             300    56   1             0      0          160.    73.1
5      25             300    56   1             0      0          157.    71.4
6      30             300    56   1             0      0          151.    65.3
Warning message:
`...` is not empty.

We detected these problematic arguments:
* `needs_dots`

These dots only exist to allow future extensions and should be empty.
Did you misspecify an argument?
> map_cov_results4
# A tibble: 6 x 8
  n_added distance_within n_cov pct_cov n_not_cov pct_not_cov dist_avg dist_sd
    <dbl>           <dbl> <int>   <dbl>     <int>       <dbl>    <dbl>   <dbl>
1       5             400    45   0.804        11       0.196     309.   145.
2      10             400    56   1             0       0         230.   100.
3      15             400    56   1             0       0         196.    93.0
4      20             400    56   1             0       0         254.    98.9
5      25             400    56   1             0       0         221.    99.8
6      30             400    56   1             0       0         212.    99.1
Warning message:
`...` is not empty.

We detected these problematic arguments:
* `needs_dots`

These dots only exist to allow future extensions and should be empty.
Did you misspecify an argument?
>
```

Figure 38: Results of MCLP