

Spatial analysis without a GIS

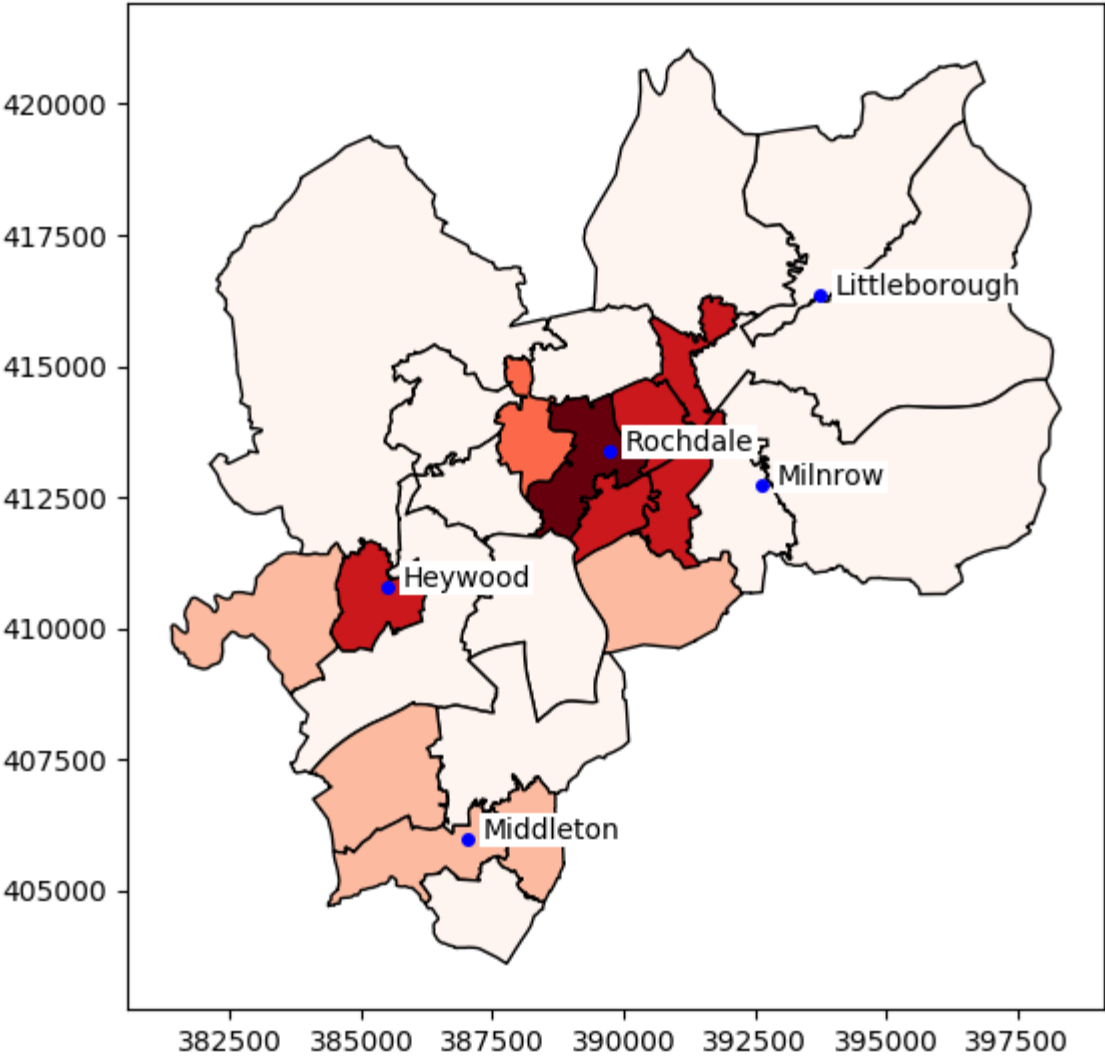


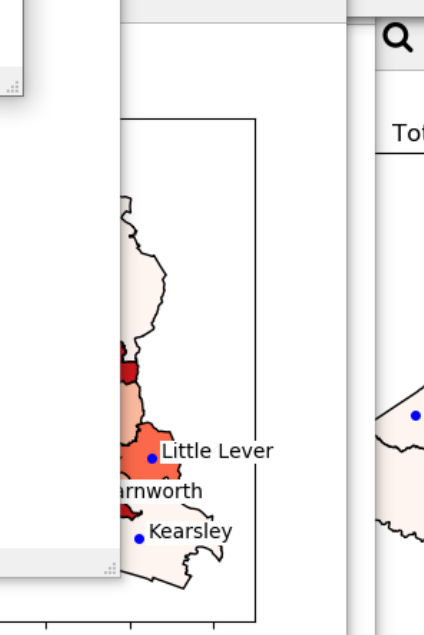
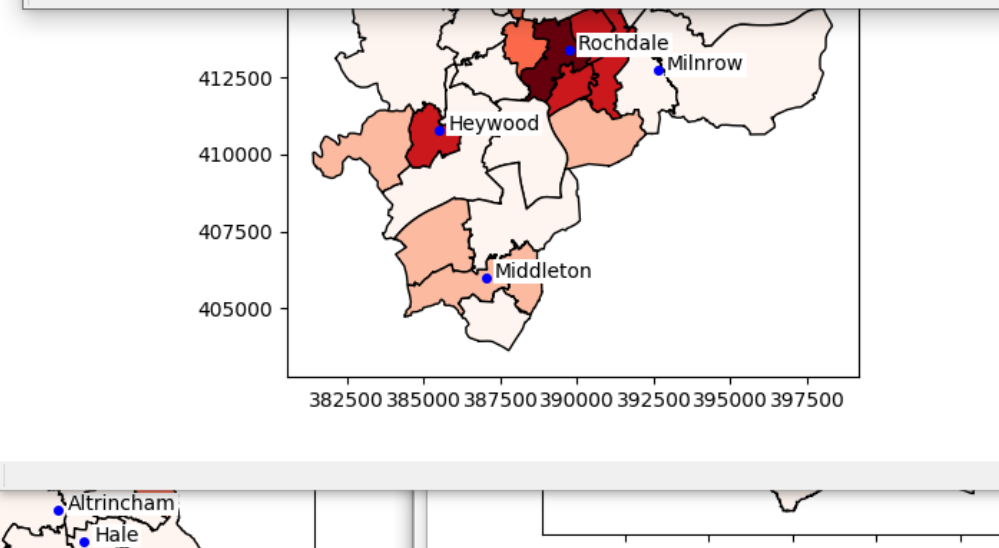
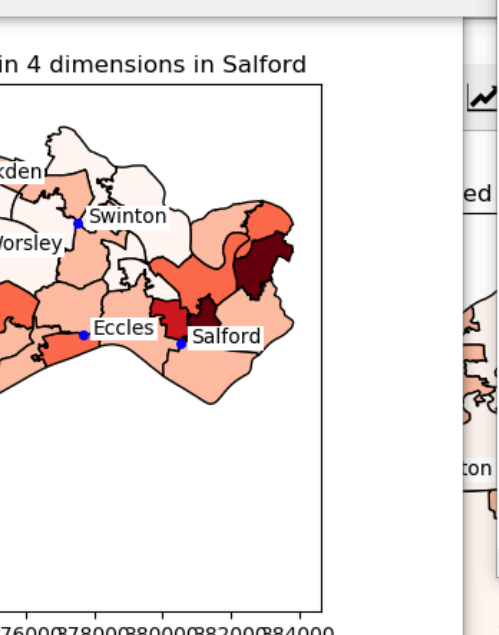
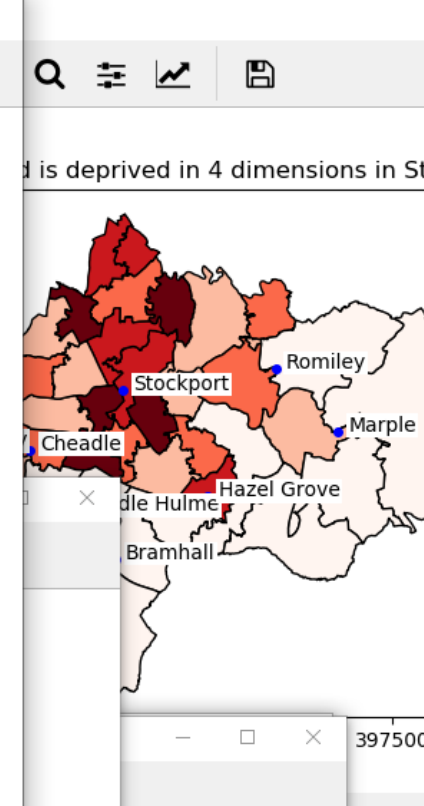
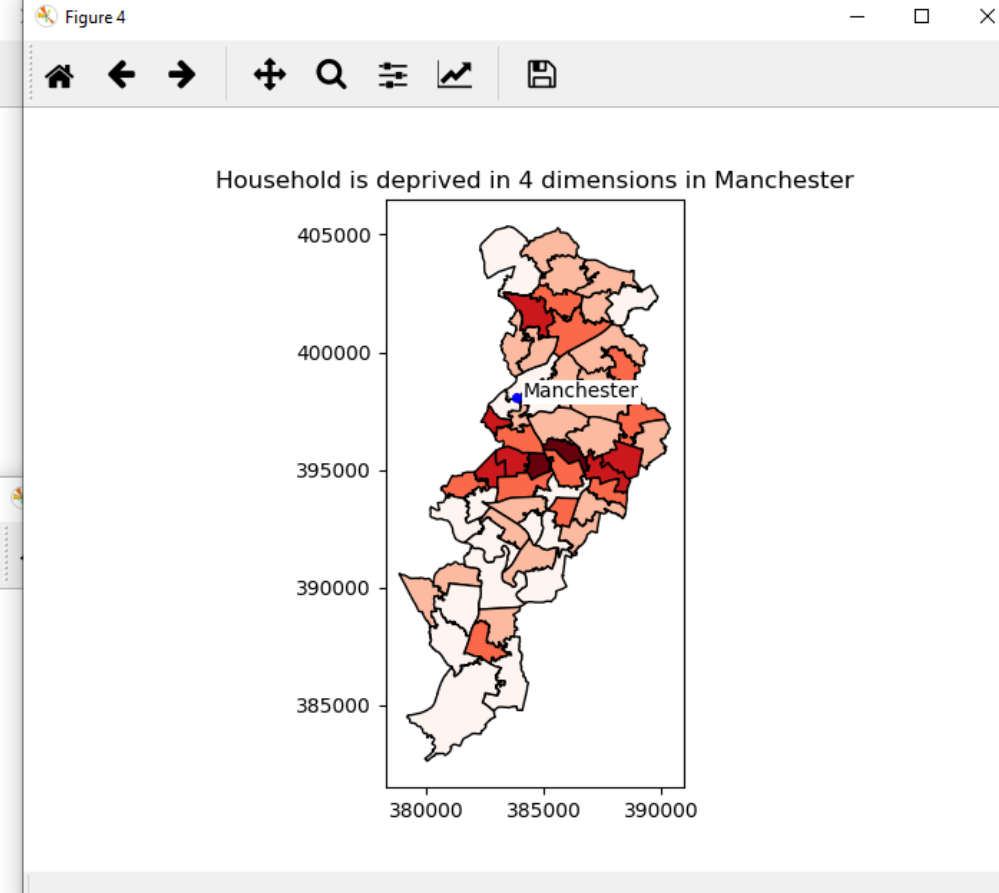
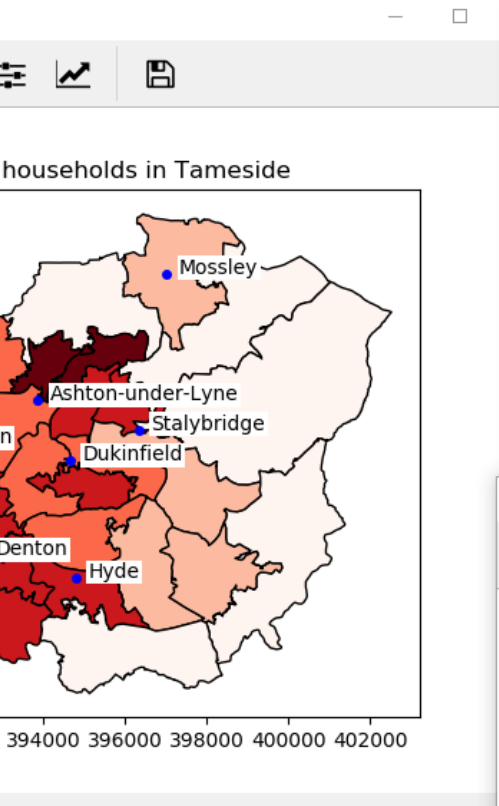
Dr Nicholas Gould
nickgould@live.co.uk

www.ondemandmapping.org.uk/igis

December 2019

Household is deprived in 4 dimensions in Rochdale







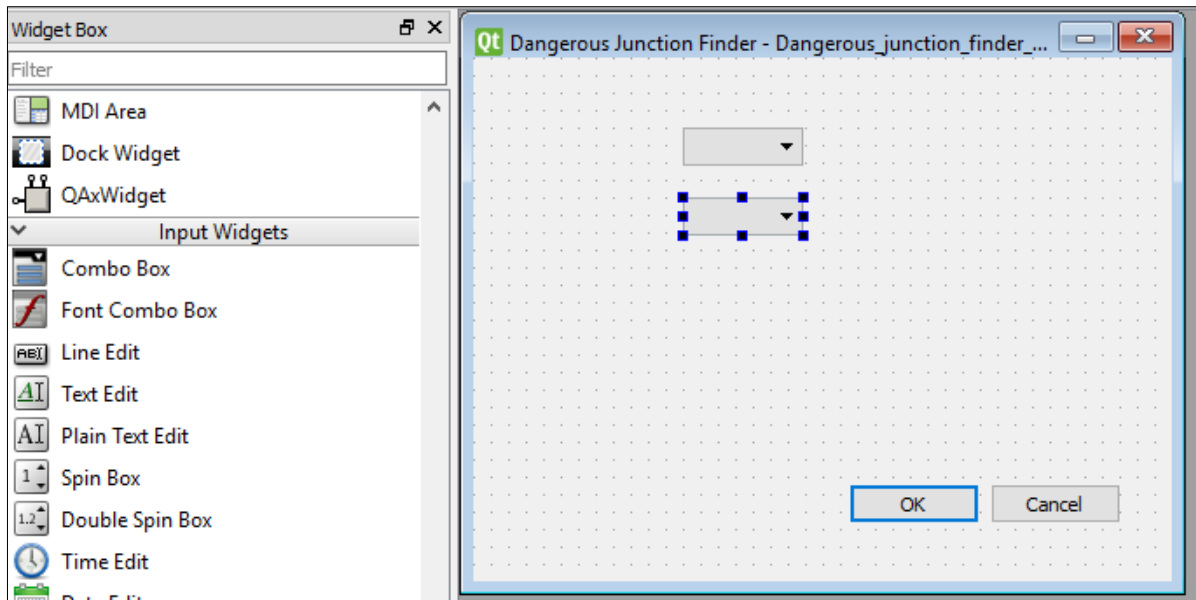
Scripting/Programming



- Within GIS
 - Python scripts
 - Python “plugins” – ArcGIS, QGIS
 - Versions!
- Standalone
 - Python
 - R
 - Python or R?

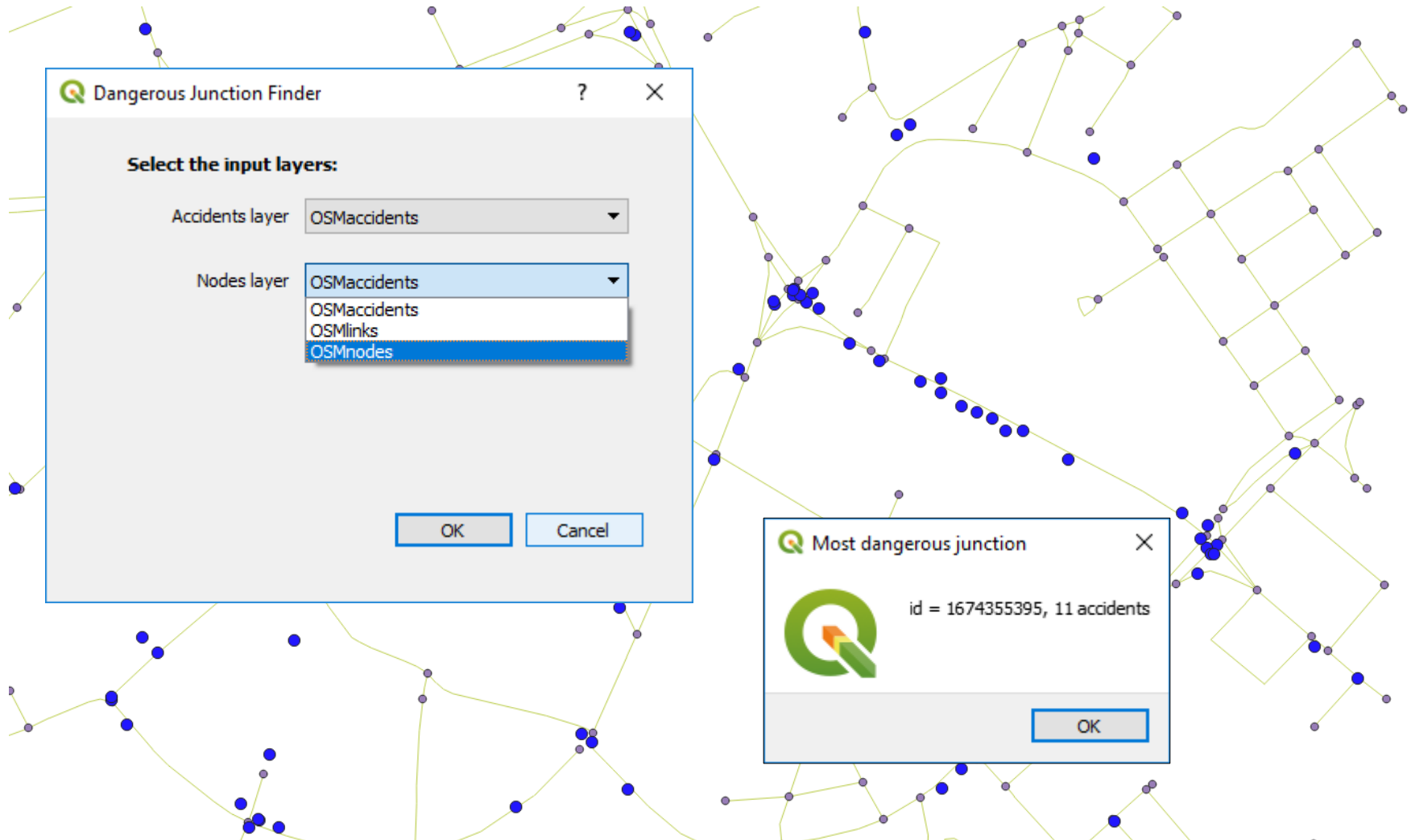
QGIS plugins

- *Plugin Builder* tool
 - Creates necessary files
- Design interface with Qt Designer



- Write script with editor

QGIS plugins



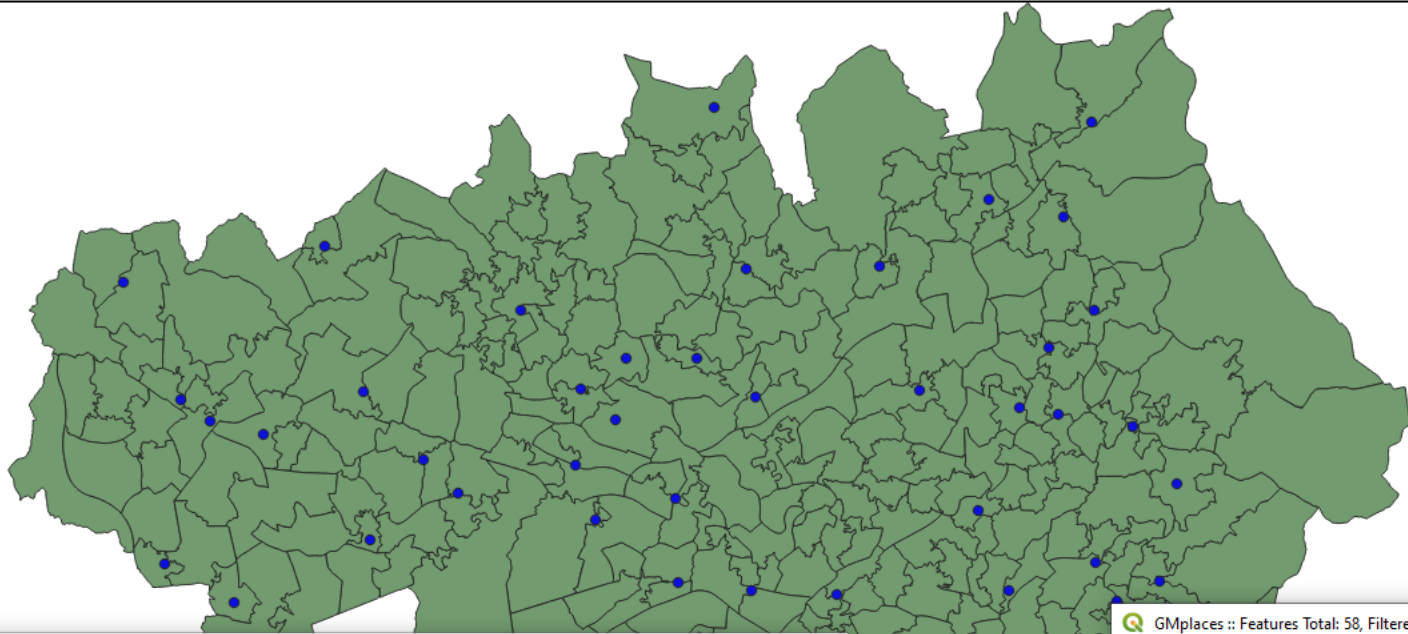
Python

- Scripting language
- Efficient
- Large code-base of libraries esp. *geospatial*
- Simple

```
def run(self):
    """Run method that performs all the real work"""
    self.dlg.accidentsLayer.clear()
    self.dlg.nodesLayer.clear()
    layers = self.iface.mapCanvas().layers()
    layer_list = []
    for layer in layers:
        layer_list.append(layer.name())
    self.dlg.accidentsLayer.addItems(layer_list)
    self.dlg.nodesLayer.addItems(layer_list)
    # show the dialog
    self.dlg.show()
    # Run the dialog event loop
    result = self.dlg.exec_()
    # See if OK was pressed
    if result:
        for layer in layers:
            if layer.name() == self.dlg.accidentsLayer.currentText():
                accident_layer = layer
            if layer.name() == self.dlg.nodesLayer.currentText():
                node_layer = layer
        QgsMessageLog.logMessage("Number of accidents:" + str(accident_layer.featureCount()), "Dangerous junctions", level=0)
        junction_threshold = self.dlg.junctionThreshold.text() #metres
        nodes = {}
        for accident in accident_layer.getFeatures():
            nearest_neighbour_ID = 0
            nearest_neighbour_distance = 1000000
            for node in node_layer.getFeatures():
                dist = accident.geometry().distance(node.geometry())
                if dist < nearest_neighbour_distance:
                    nearest_neighbour_distance = dist
                    nearest_neighbour_ID = node['osmid']
            if nearest_neighbour_distance < int(junction_threshold):
                #this accident is near to a junction
                if not nearest_neighbour_ID in nodes:
```



Python example: mapping census data



GM_MSOA_deprivation_2011 :: Features Total: 346, Filtered: 346, Selected: 0

	name	code	F996	F997	F998	F999	F1000
22	Oldham 029	E02001126	3141	823	1076	907	305
23	Manchester 055	E02006912	3885	1422	1898	469	92
24	Oldham 028	E02001125	3351	1295	1174	702	171
25	Manchester 059	E02006916	3274	1264	1287	495	204
26	Oldham 032	E02001129	2622	846	854	682	223
27	Rochdale 004	E02001135	3052	682	981	945	407
28	Rochdale 005	E02001136	3667	1533	1218	720	186

GMplaces :: Features Total: 58, Filtered: 58, Selected: 0

	NAME1	TYPE	LOCAL_TYPE
1	Wigan	populatedPlace	Town
2	Little Lever	populatedPlace	Town
3	Ince-in-Makerfi...	populatedPlace	Town
4	Standish	populatedPlace	Town
5	Bolton	populatedPlace	Town
6	Swinton	populatedPlace	Town
7	Tyldesley	populatedPlace	Town
8	Leigh	populatedPlace	Town

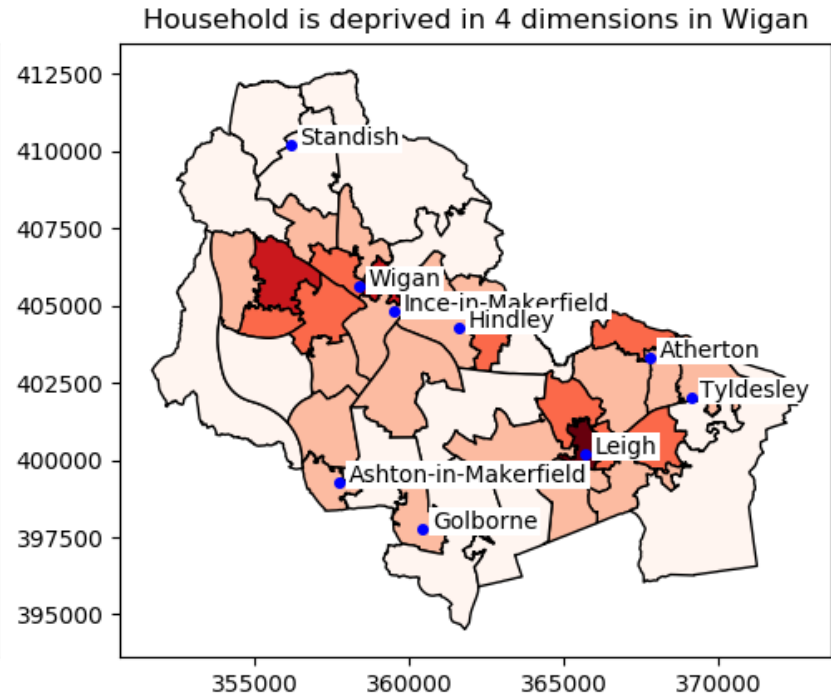
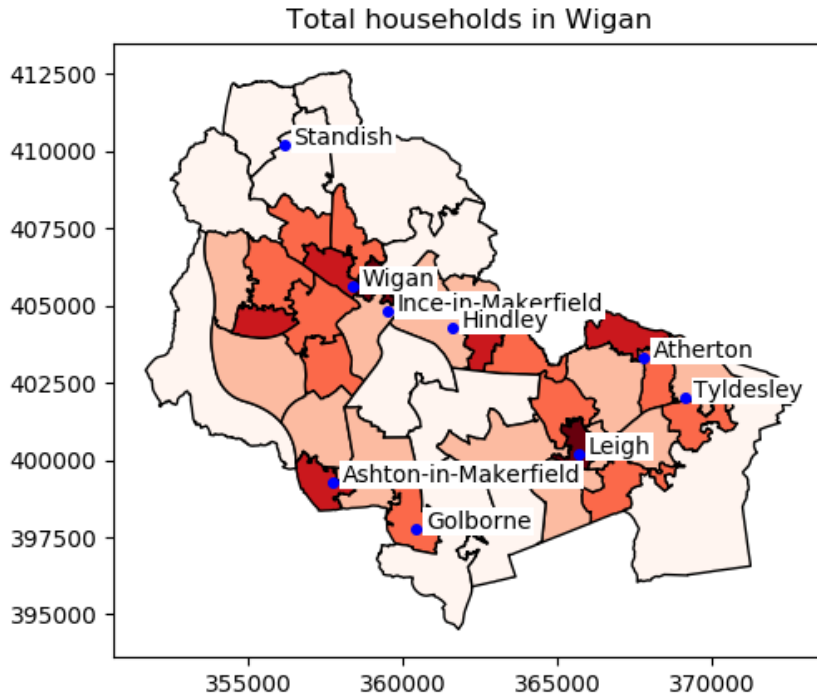
Python example: mapping census data

- Aim: for each district in Greater Manchester
 - Produce a map of household density at MSOA level
 - Produce a map of deprivation density at MSOA level
 - Add place names for context

Python example: mapping census data

- For each MSOA:
 - Calculate area for each MSOA from geometry
 - Calculate household density & deprivation density
 - Extract district name from MSOA name
- For each district (Manchester, Wigan etc.):
 - Extract data for district
 - Extract place names for district (point in polygon)
 - For each census statistic
 - Map density + place names + title
 - Save as PNG format

Python example: mapping census data



Key libraries – matplotlib and geopandas

Anaconda

- Open source distribution for Python and R
 - Focus on data-science
 - Including spatial tools and editor - Spyder
- Package management
 - Handles *dependencies* better than PIP
- Navigator
- www.anaconda.com/download/



Installed Channels Update index... Search Packages

Name

- fiona
- flask
- folium
- freetype
- freexl
- gdal

Update Packages X

2 packages will be installed

	Name	Unlink	Link	Channel
1	geopandas	0.3.0	0.6.1	defaults
2	*pandas	0.22.0	0.23.4	defaults

* indicates the package is a dependency of a selected package

ImportError: PyS

'e' keyword



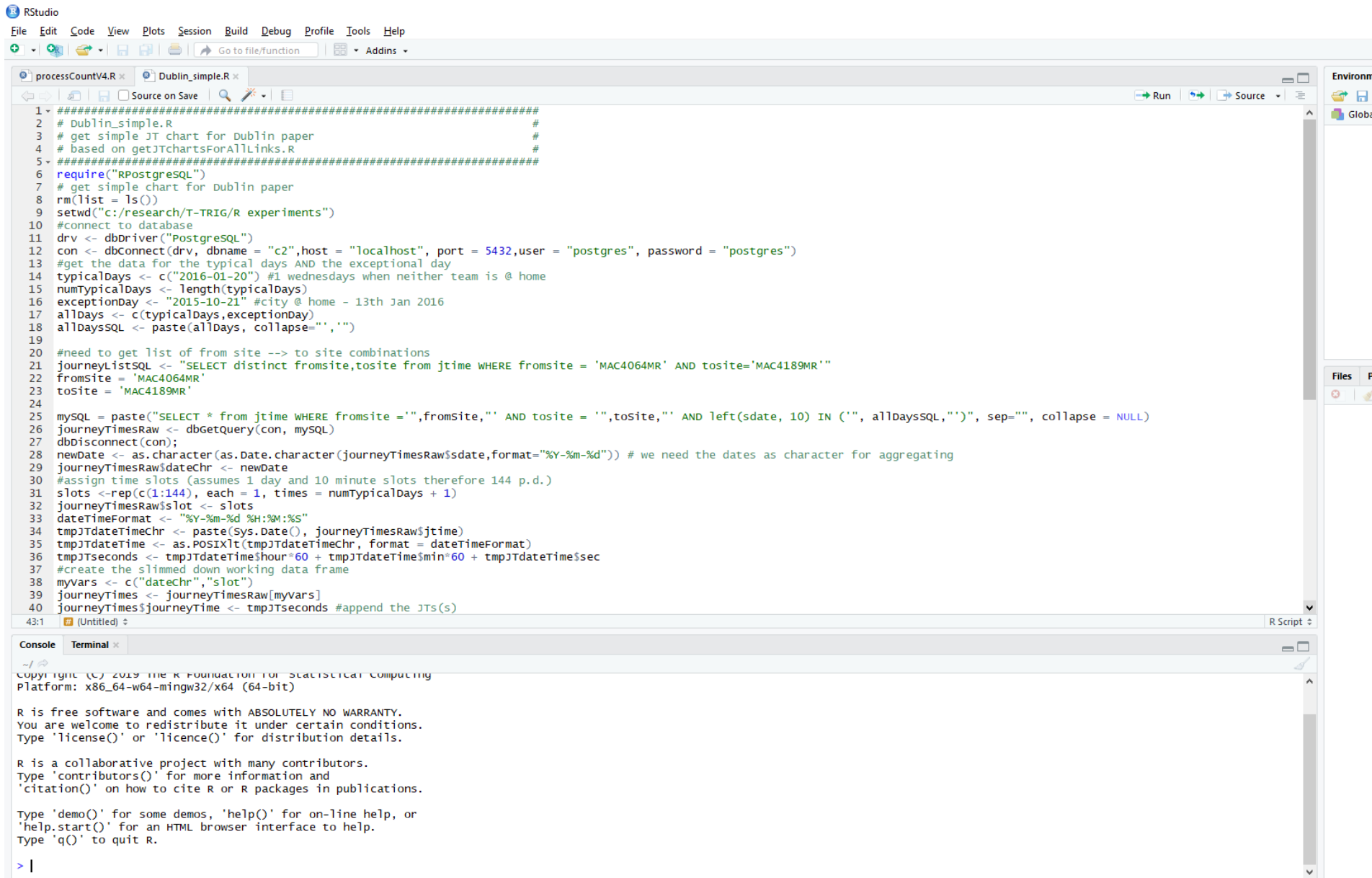


R



- Language and environment for statistics
- Scripting
- R Studio
 - IDE for R

R Studio



The image shows the R Studio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu bar is a toolbar with icons for file operations and a search bar. The main editor window displays a script named 'Dublin_simple.R' with the following R code:

```
1 #####
2 # Dublin_simple.R
3 # get simple JT chart for Dublin paper
4 # based on getJTchartsForAllLinks.R
5 #####
6 require("RPostgreSQL")
7 # get simple chart for Dublin paper
8 rm(list = ls())
9 setwd("c:/research/T-TRIG/R experiments")
10 #connect to database
11 drv <- dbDriver("PostgreSQL")
12 con <- dbConnect(drv, dbname = "c2", host = "localhost", port = 5432, user = "postgres", password = "postgres")
13 #get the data for the typical days AND the exceptional day
14 typicalDays <- c("2016-01-20") #1 wednesdays when neither team is @ home
15 numTypicalDays <- length(typicalDays)
16 exceptionDay <- "2015-10-21" #city @ home - 13th Jan 2016
17 allDays <- c(typicalDays, exceptionDay)
18 allDaysSQL <- paste(allDays, collapse=" ", "")
19
20 #need to get list of from site --> to site combinations
21 journeyListSQL <- "SELECT distinct fromsite, tosite from jtime WHERE fromsite = 'MAC4064MR' AND tosite='MAC4189MR'"
22 fromSite = 'MAC4064MR'
23 toSite = 'MAC4189MR'
24
25 mySQL = paste("SELECT * from jtime WHERE fromsite = '", fromSite, "' AND tosite = '", toSite, "' AND left(sdate, 10) IN ('", allDaysSQL, "')", sep=" ", collapse = NULL)
26 journeyTimesRaw <- dbGetQuery(con, mySQL)
27 dbDisconnect(con);
28 newDate <- as.character(as.Date.character(journeyTimesRaw$sdate, format="%Y-%m-%d")) # we need the dates as character for aggregating
29 journeyTimesRaw$dateChr <- newDate
30 #assign time slots (assumes 1 day and 10 minute slots therefore 144 p.d.)
31 slots <- rep(c(1:144), each = 1, times = numTypicalDays + 1)
32 journeyTimesRaw$slot <- slots
33 dateTimeFormat <- "%Y-%m-%d %H:%M:%S"
34 tmpJTdateTimeChr <- paste(Sys.Date(), journeyTimesRaw$jtime)
35 tmpJTdateTime <- as.POSIXlt(tmpJTdateTimeChr, format = dateTimeFormat)
36 tmpJTseconds <- tmpJTdateTime$hour*60 + tmpJTdateTime$min*60 + tmpJTdateTime$sec
37 #create the slimmed down working data frame
38 myVars <- c("dateChr", "slot")
39 journeyTimes <- journeyTimesRaw[myVars]
40 journeyTimes$journeyTime <- tmpJTseconds #append the JTs(s)
43:1 [Untitled] R Script
```

The bottom panel shows the Terminal window with the following text:

```
~ |
Copyright (c) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

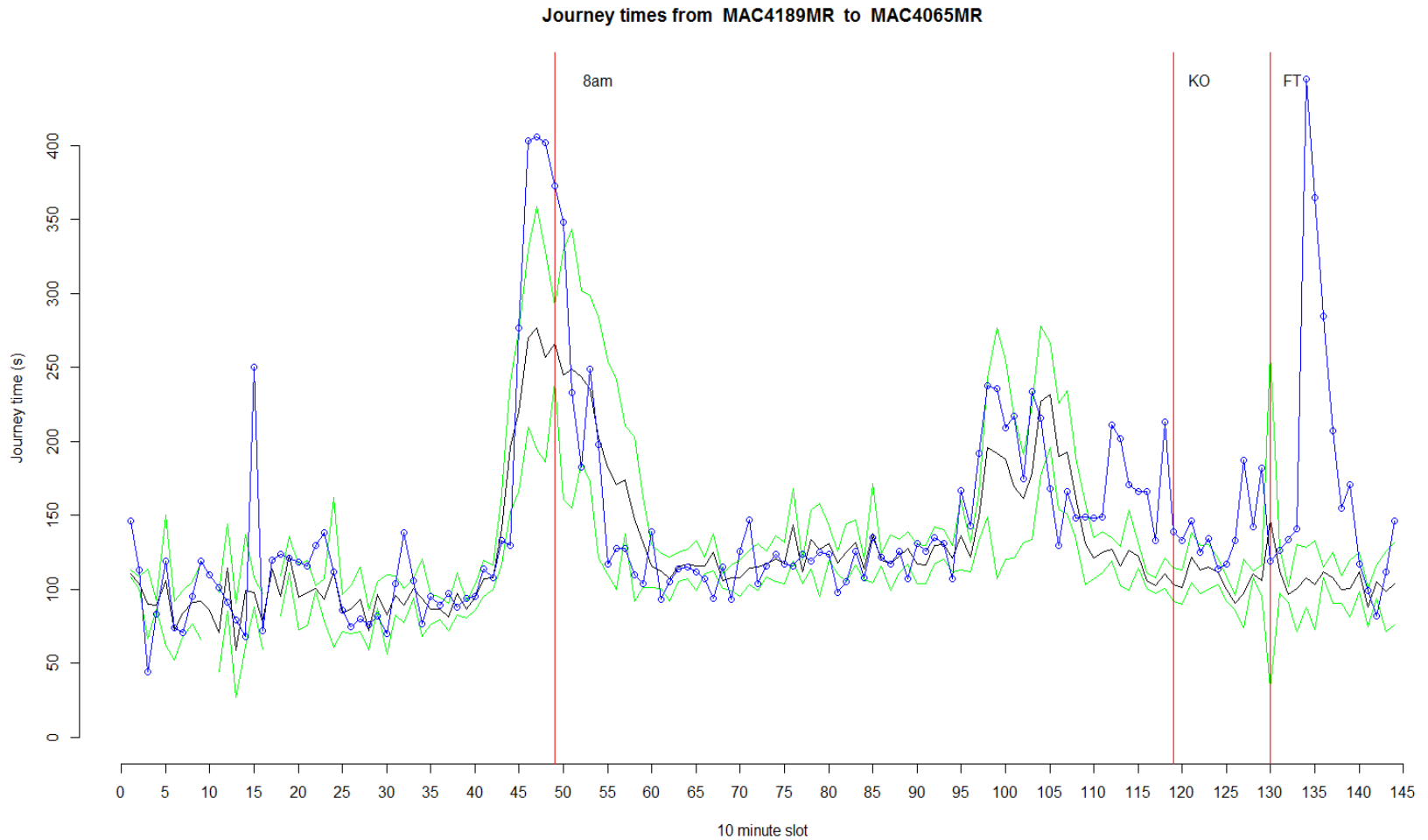
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

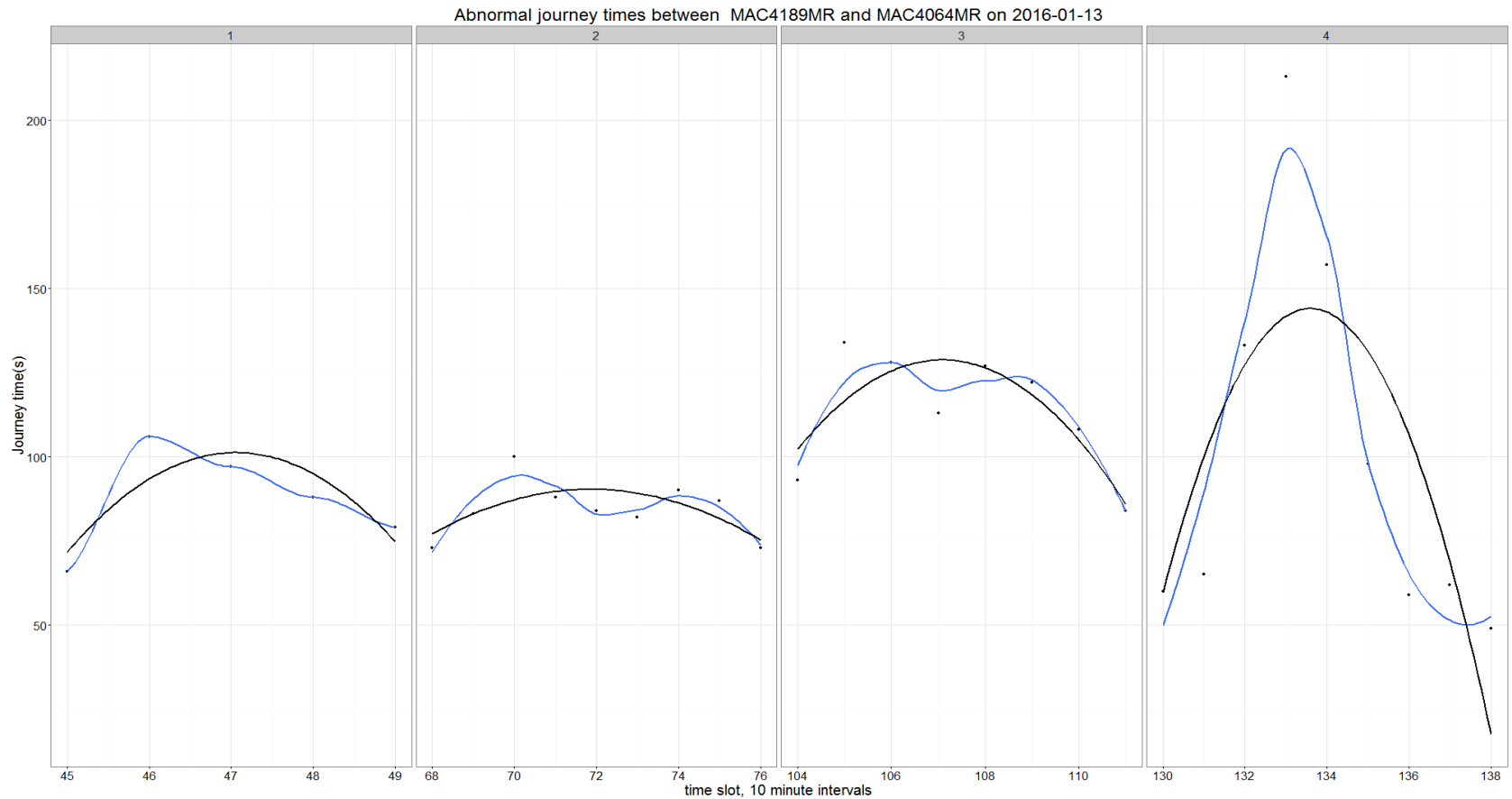
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

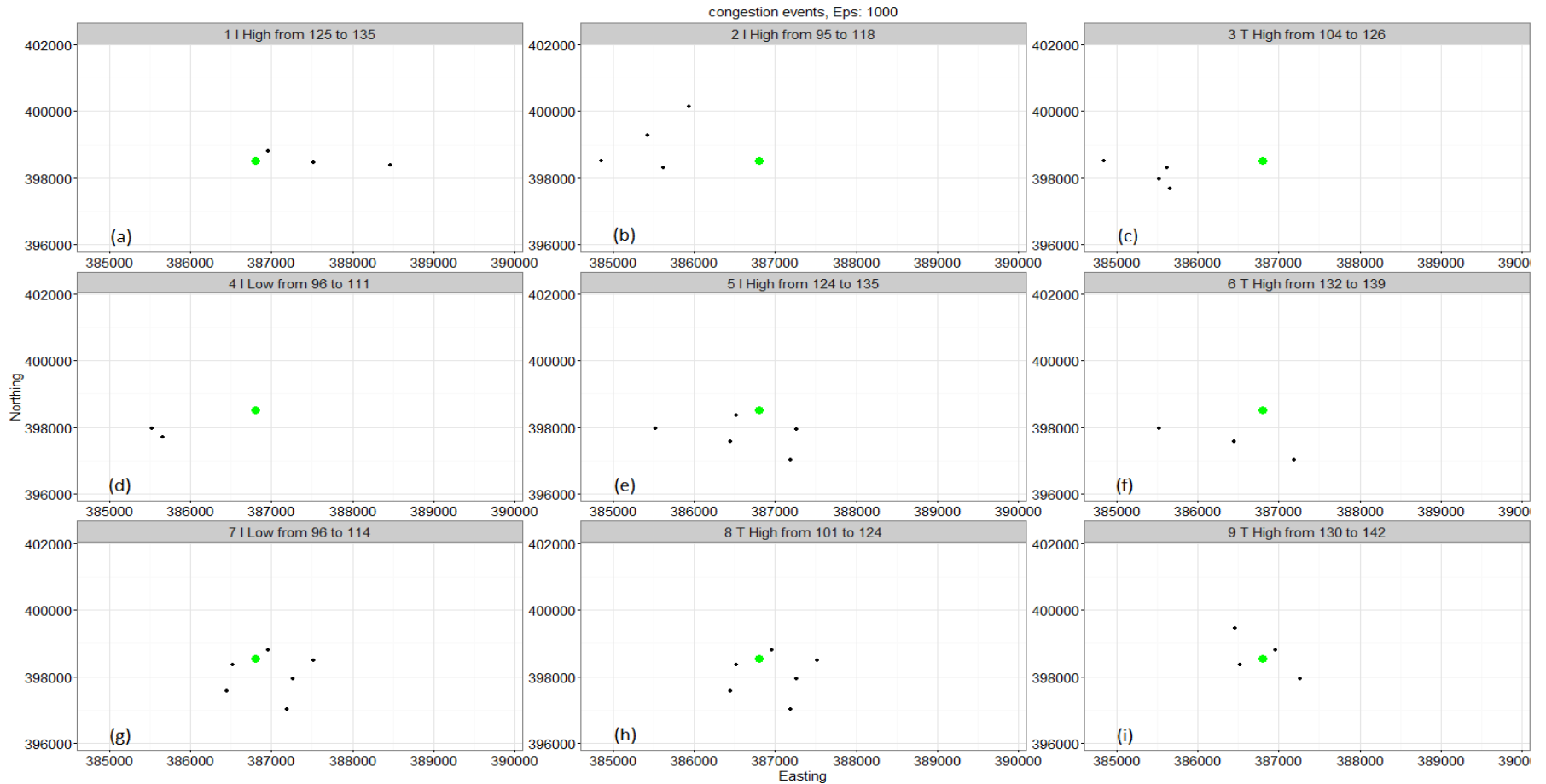
R project – road congestion



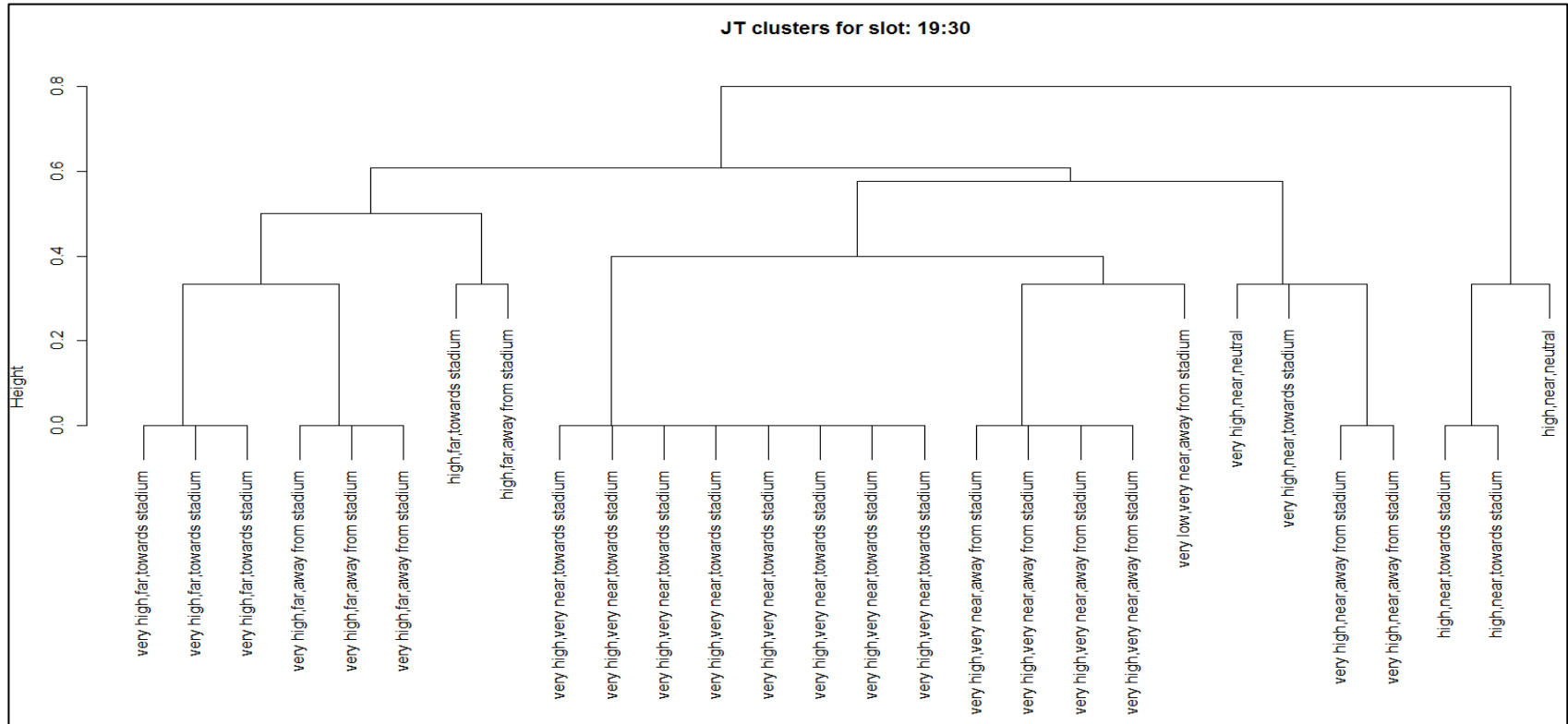
R project – road congestion



R project – road congestion



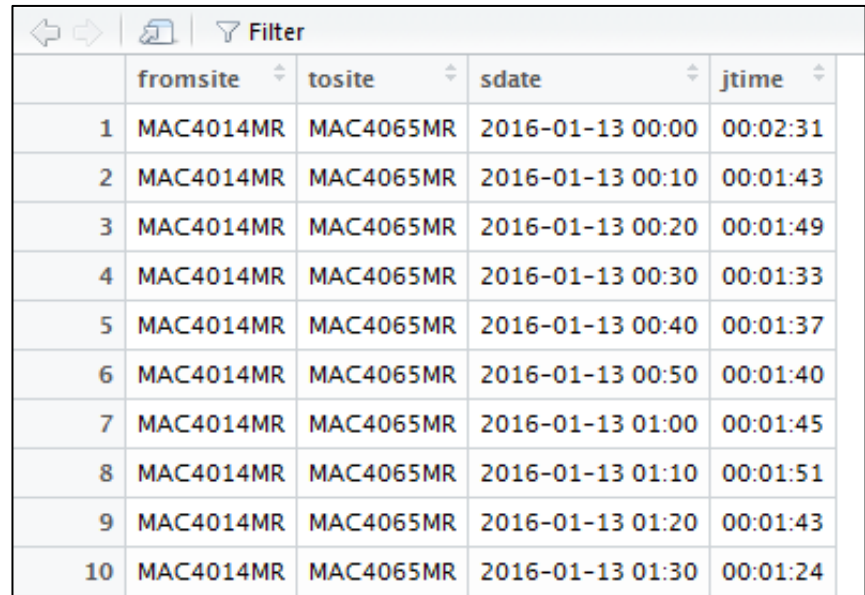
R project – road congestion



Mixing spatial and non-spatial analysis in the same project

R – database connection

```
require("RPostgreSQL")
drv <- dbDriver("PostgreSQL")
con <- dbConnect(drv, dbname = "c2", host = "localhost",
port = 5432, user = "postgres", password = "postgres")
journeyListSQL <- "SELECT distinct fromsite, tosite from
jtime ORDER BY fromsite, tosite"
journeyList <- dbGetQuery(con, journeyListSQL)
dbDisconnect(con);
```



	fromsite	tosite	sdate	jtime
1	MAC4014MR	MAC4065MR	2016-01-13 00:00	00:02:31
2	MAC4014MR	MAC4065MR	2016-01-13 00:10	00:01:43
3	MAC4014MR	MAC4065MR	2016-01-13 00:20	00:01:49
4	MAC4014MR	MAC4065MR	2016-01-13 00:30	00:01:33
5	MAC4014MR	MAC4065MR	2016-01-13 00:40	00:01:37
6	MAC4014MR	MAC4065MR	2016-01-13 00:50	00:01:40
7	MAC4014MR	MAC4065MR	2016-01-13 01:00	00:01:45
8	MAC4014MR	MAC4065MR	2016-01-13 01:10	00:01:51
9	MAC4014MR	MAC4065MR	2016-01-13 01:20	00:01:43
10	MAC4014MR	MAC4065MR	2016-01-13 01:30	00:01:24

Databases

- MS SQL Server, Oracle, PostgreSQL
 - Spatial extensions
- For serving data
 - With QGIS, Python, R
- For spatial queries
 - Good for mixed attribute, spatial queries, multi-table

PostgreSQL/PostGIS and QGIS

Data Output											
Explain Messages Query History											
	gid [PK] integer	accidentid double precision	year integer	dayname character varying (9)	severity integer	numbercasu integer	date character varying (10)	easting double precision	northing double precision	ward character varying (100)	geom geometry
1	1	11	1996	Monday	3	1	1996/04/01	383811	398681	CITY CENTRE	0101000020346C0...
2	2	21	1996	Monday	3	2	1996/04/01	383806	398697	CITY CENTRE	0101000020346C0...
3	3	33	1994	Monday	3	1	1994/01/17	383723	398467	CITY CENTRE	0101000020346C0...
4	4	63	1995	Monday	3	2	1995/04/17	384094	398862	CITY CENTRE	0101000020346C0...
5	5	124	1995	Monday	2	1	1995/05/01	383923	398274	CITY CENTRE	0101000020346C0...
6	6	138	1994	Monday	3	1	1994/02/07	384200	398426	CITY CENTRE	0101000020346C0...
7	7	142	1994	Monday	2	1	1994/02/07	384388	398320	CITY CENTRE	0101000020346C0...
8	8	146	1997	Monday	3	2	1997/04/14	383789	399116	CHEETHAM	0101000020346C0...
9	9	159	1995	Monday	3	1	1995/05/15	383826	398189	CITY CENTRE	0101000020346C0...
10	10	397	1995	Monday	3	2	1995/07/17	384458	398445	CITY CENTRE	0101000020346C0...
11	11	405	1994	Monday	3	1	1994/04/18	383660	398325	CITY CENTRE	0101000020346C0...
12	12	437	1995	Monday	3	1	1995/07/31	384055	398826	CITY CENTRE	0101000020346C0...
13	13	607	1996	Monday	3	1	1996/05/27	384403	398368	CITY CENTRE	0101000020346C0...
14	14	627	1995	Monday	3	1	1995/10/02	383941	398500	CITY CENTRE	0101000020346C0...
15	15	678	1995	Monday	3	1	1995/10/09	383971	398628	CITY CENTRE	0101000020346C0...
16	16	691	1995	Monday	3	1	1995/10/09	384455	398447	CITY CENTRE	0101000020346C0...
17	17	705	1997	Monday	3	1	1997/06/02	384526	398385	CITY CENTRE	0101000020346C0...
18	18	720	1995	Monday	3	2	1995/10/02	383989	398626	CITY CENTRE	0101000020346C0...
19	19	721	1996	Monday	3	1	1996/06/03	383818	398686	CITY CENTRE	0101000020346C0...
20	20	725	1994	Monday	3	1	1994/07/04	384391	398321	CITY CENTRE	0101000020346C0...
21	21	846	1994	Monday	3	1	1994/08/15	384002	398260	CITY CENTRE	0101000020346C0...

Dataset – imported from Shapefile – stored in table
 Feature geometry – stored in *geometry* column

PostgreSQL/PostGIS and QGIS

Connecting to database from QGIS
Supply credentials and database name

Create a New PostGIS Connection

Connection Information

Name: local

Service:

Host: localhost

Layer > Add Layer > Add PostGIS layers

Connections

local

Connect New Edit Remove Load Save

Schema	Table	Comment	Column	Data Type	Spatial Type	SRID
public	accidents		geom	Geometry	Point	27700
public	osmlinks		geom	Geometry	MultiLineString	27700
public	osmnodes		geom	Geometry	Point	27700
public	raster_columns		extent	Geometry	Select...	Enter...
tiger						

Don't resolve type of unrestricted columns (GEOMETRY)

Only look in the 'public' schema

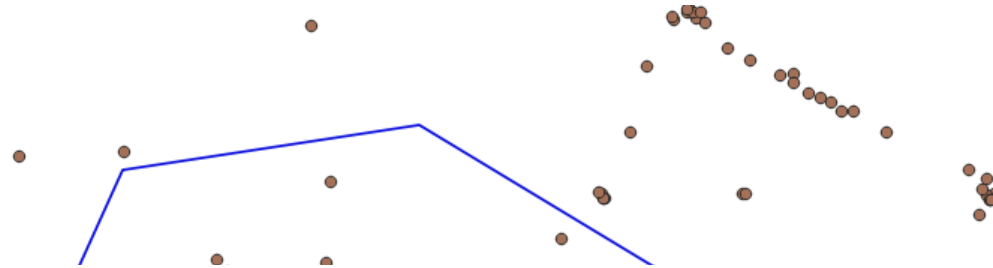
Also list tables with no geometry

Use estimated table metadata

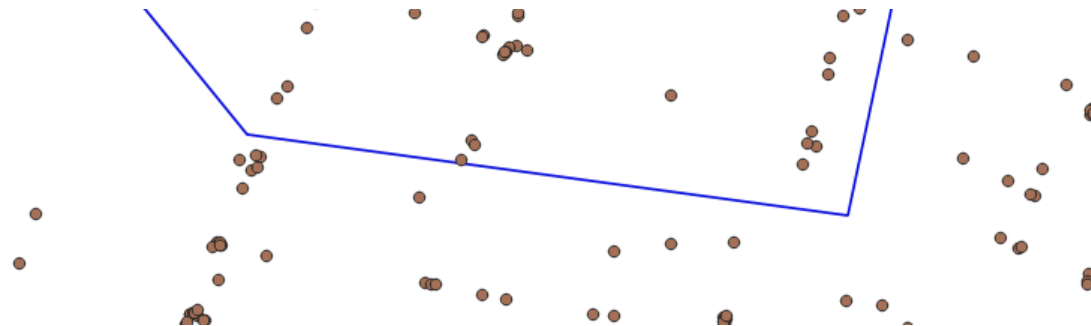
Allow saving/loading QGIS projects in the database

OK Cancel Help

Spatial database queries



```
SELECT * FROM accidents, citycentre WHERE  
accidents.severity = 3 AND  
ST_Within(accidents.geom,citycentre.geom)
```



Can easily combine spatial and attribute queries

Conclusion

- Not a case of *either or*
- Technologies are integrated
 - Query SQL databases in QGIS
 - Create Leaflet maps from R
 - Run Python scripts within QGIS