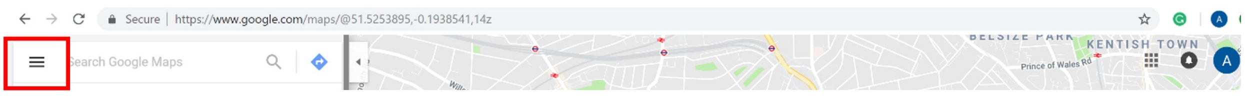
Practical/Assessment Week 7

17 November 2018

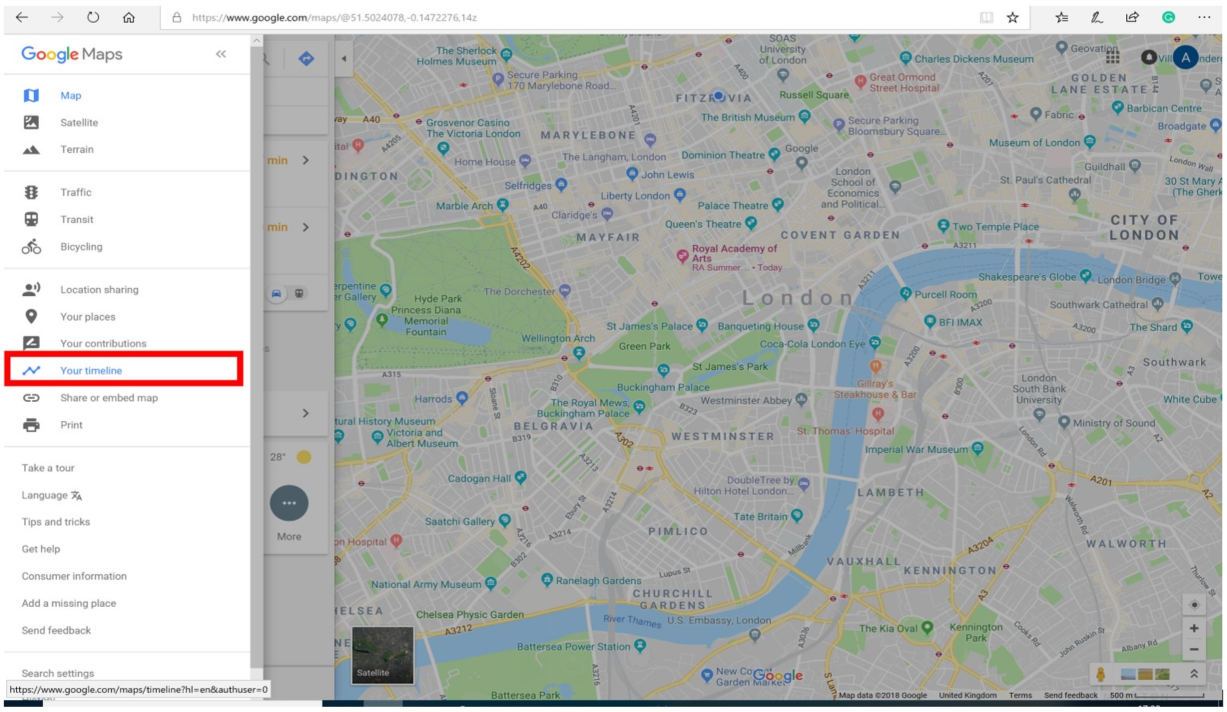
This practical is basically a spatial challenge, in which you will be putting into practice the various data preparation and handling, topographic and thematic mapping skills you have learned over the last few weeks to carry out a piece of spatial analysis in ArcGIS or R (the choice is yours!). We would like to give you some hints and instructions but quite vague ones as this is actually a part of your assessment. So when you finish this practical and answered the six questions we set, you write a 600-word text to explain how you approach these challenges and come up with those answer and submit it. However, you are not all on your own as we give you some instructions and hints here. This practical requires a few datasets, either available on Moodle or you can download/get access to them using the links and instructions we give here. So let's start!

Task - Getting A/Your Treasure Hunt Trace!

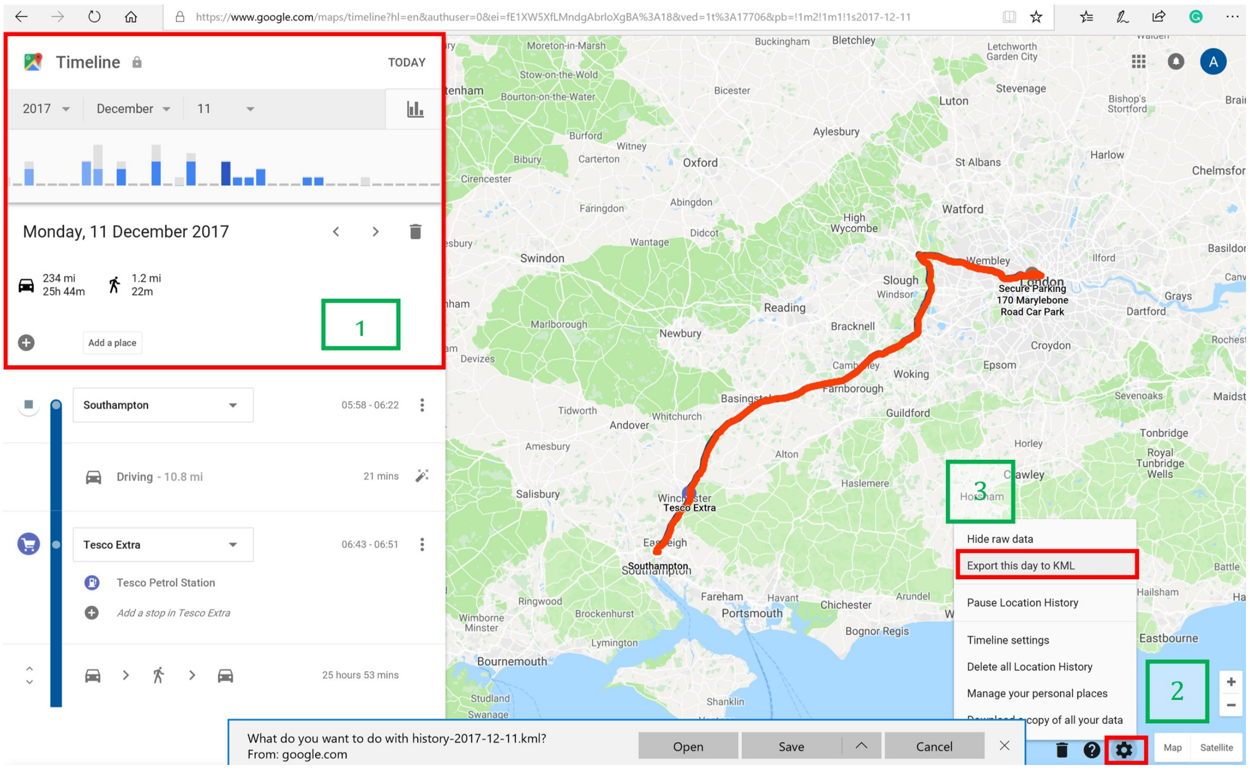
You all remember the CASA Treasure Hunt, right? Check your photos on Twitter: https://twitter.com/search?q=%23CASAtreasurehunt&src=typd (https://twitter.com/search?q=%23CASAtreasurehunt&src=typd) We gave an instruction on how to turn on your mobile location and asked you to record your traces of movement. Some of you downloaded your KML file in the first session and used it for your practical. But don't panic if you don't have your route recorded or simply forgot how to download it. We can use some of the last year's group traces of movement which is quite similar to yours. I also explain here how to download your traces from your Google Map if you had turned on your phone location capabilities. If you log in to your Gmail account and then go to Google Map, there is a menu icon:



If you click that you see a list, one of which is called Your timeline. Google used to track you unless you turn off your location history and this scary game Google played helps us to do this practical. we are here to make most of it!



If you click on Your timeline, you can find the date you went for treasure hunt, click on setting icon and then export this day to KML and then convert it to any format you wish to use (geojson, shp, etc.)



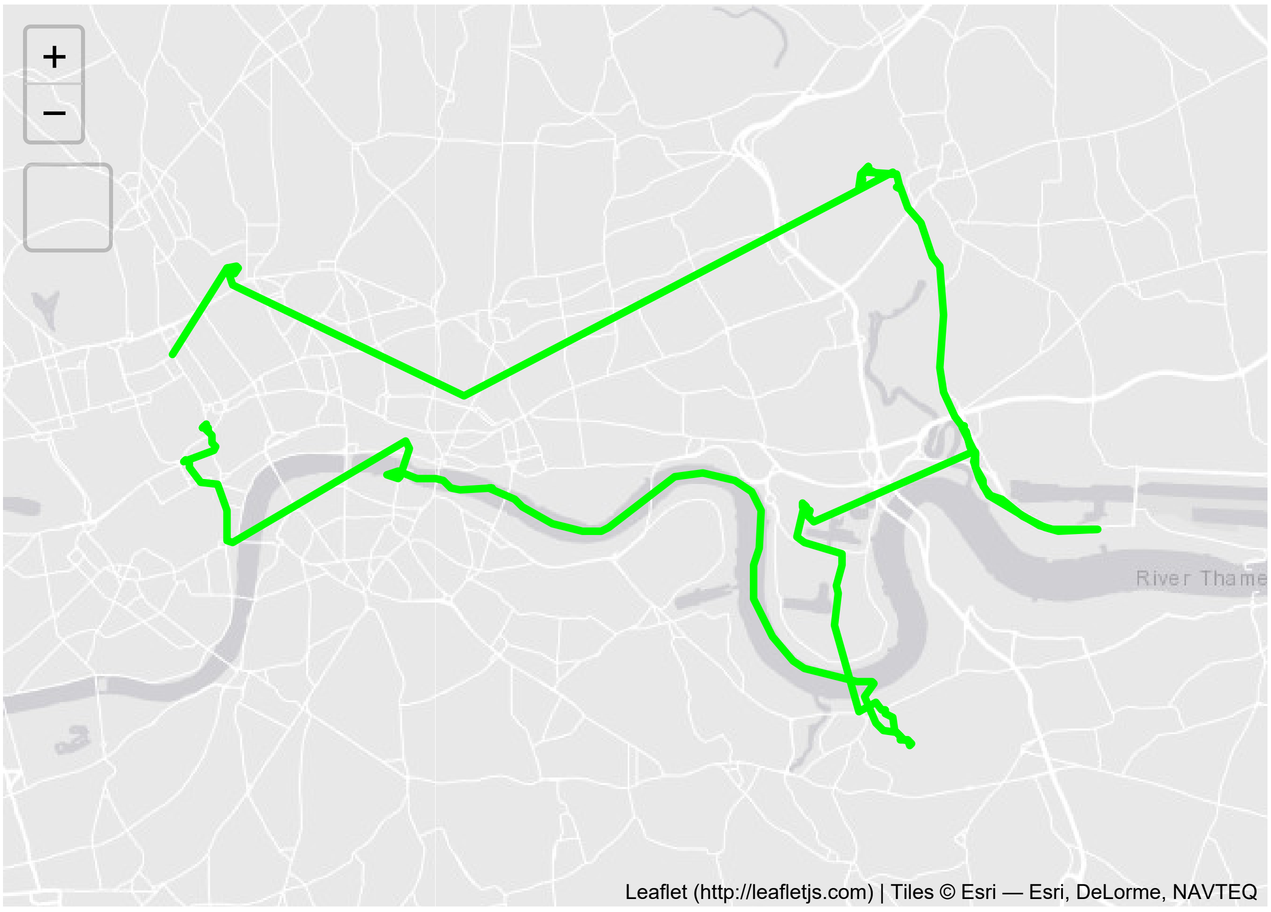
However, if for any reason you do not have your trace recorded, you can use the last year's winning group trace as it should be on the quite the same route as you went. Here's where they went:

Team 7’s Route in 2016

```r
library(tmap)
library(geojsonio)

#Here's their data:
hunt <- geojson_read("https://www.dropbox.com/s/wa2ip35tcmt93g3/Team7.geojson?raw=1", wha
t = "sp")

#And here's where they went...
tmap_mode("view")
tm_shape(hunt) +
  tm_lines(col = "green", lwd = 4)
```


+
−
River Thame
Leaflet (http://leafletjs.com) | Tiles © Esri — Esri, DeLorme, NAVTEQ

I would like you to analyse the route using a variety of spatial analysis functions available in either R or ArcGIS

Questions to Answer

Ok, using ‘a’ trajectory (either yours or last year’s winning group’s) we would like you to analyse the route using a variety of spatial analysis functions available in either R or ArcGIS and answer these 6 questions:

1. How far did you travel (according to the trace they recorded We all can calculate google travel distance using routing service so do not do this.

2. How many TfL station did your route pass within 100 metres distance?

3. How many points did you score based on treasure hunt locations they managed to get within 300 metres of? Download the csv file of the location and scores from Moodle.

4. Which Wards did you pass through that had the (a) lowest and (b) the highest rates of Male Life Expectancy?

5. Taking the average of all Wards that you passed through, what was the average life expectancy at birth for babies born in those wards along the whole route? This can be both Male and Female life expectancies.

6. Is there any spatial patterns for CASA Treasure Hunt locations or are they randomly distributed?

How to tackle the challenge

1. "Choose which side of the force you are on (Dark Side or Light Side)", Adam says!
2. Download your data **Your trajectory of movement as explained above or last year's winning teams** I also put these file on Moodle but you can get them from Adam's dropbox

GeoJSON - https://www.dropbox.com/s/wa2ip35tcmt93g3/Team7.geojson?raw=1 (https://www.dropbox.com/s/wa2ip35tcmt93g3/Team7.geojson?raw=1)

Shapefile - https://www.dropbox.com/s/chzovfrvw9qsrsi/Team7.zip?dl=0 (https://www.dropbox.com/s/chzovfrvw9qsrsi/Team7.zip?dl=0)

The location of Tube / Rail Stations in London

These can be obtained from here: https://www.doogal.co.uk/london_stations.php (https://www.doogal.co.uk/london_stations.php)

These can be obtained from here: https://www.doogal.co.uk/london_stations.php (https://www.doogal.co.uk/london_stations.php)

```
#You can have this for free, R fans. Reading XML can be a pain in R, you will need the 'L
ayer' information, which is contained in the <name> tag in a KML file...
#install.packages("rgdal")
library(rgdal)
tubestations <- readOGR("https://www.doogal.co.uk/LondonStationsKML.ashx", "London statio
ns with zone information")
```

```
## OGR data source with driver: KML
## Source: "https://www.doogal.co.uk/LondonStationsKML.ashx", layer: "London stations wit
h zone information"
## with 641 features
## It has 2 fields
```

The treasure hunt locations

A raw list with the points can be downloaded from here: https://www.dropbox.com/s/v66l4cx7aia9jlo/huntLocations.csv?raw=1 (https://www.dropbox.com/s/v66l4cx7aia9jlo/huntLocations.csv?raw=1)

You will notice that the raw list doesn't contain any spatial information at all. You will need to add this. Now, you can do it yourself, however below is quite a cool way for doing it automatically using the Google Places API

R Code for GeoCoding a list of places with no spatial reference:

1. First read a list of treasure hunt locations and their points scores from my dropbox

```
library(tidyverse)

huntaddresses <- read_csv("https://www.dropbox.com/s/v66l4cx7aia9jlo/huntLocations.csv?ra
w=1")
```

2. Now geocode them using the Google Maps API and a neat bit of code plundered from here: https://gist.github.com/josecarlosgonz/6417633 (https://gist.github.com/josecarlosgonz/6417633)

```r
#code here lifted directly from - https://gist.github.com/josecarlosgonz/6417633

#library the required packages
install.packages("RCurl")
install.packages("RJSONIO")
install.packages("plyr")
install.packages("tidyverse")
library(RCurl)
library(RJSONIO)
library(plyr)
library(tidyverse)

#highlight this whole block and create this function to access the Google Places API
url <- function(address, return.call = "json", sensor = "false") {
  root <- "http://maps.google.com/maps/api/geocode/"
  u <- paste(root, return.call, "?address=", address, "&sensor=", sensor, sep = "")
  return(URLencode(u))
}

#highlight this whole block and create this function to geocode some places just from a r
andom list of treasure hunt locations
geoCode <- function(address,verbose=FALSE) {
  if(verbose) cat(address,"\n")
  u <- url(address)
  doc <- getURL(u)
  x <- fromJSON(doc,simplify = FALSE)
  if(x$status=="OK") {
    lat <- x$results[[1]]$geometry$location$lat
    lng <- x$results[[1]]$geometry$location$lng
    location_type  <- x$results[[1]]$geometry$location_type
    formatted_address  <- x$results[[1]]$formatted_address
    return(c(lat, lng, location_type, formatted_address))
    Sys.sleep(0.5)
  } else {
    return(c(NA,NA,NA, NA))
  }
}

#now use the geoCode() function (which calls the URL function) to geocode our list of pla
ces

#for loop to cycle through every treasure hunt location
i=1
for(i in 1:nrow(huntaddresses)){
  # Every nine records, pause 3 seconds so that the API doesn't kick us off...
  if(i %% 9 == 0) Sys.sleep(3)
  #now create a temporary list of useful elements
  tempdf <- as.list(geoCode(huntaddresses[i,1]))
  #and write these back into our dataframe
  huntaddresses[i,3] <- tempdf[1]
  huntaddresses[i,4] <- tempdf[2]
  huntaddresses[i,5] <- tempdf[4]
}

# rename the columns
```

```
names(huntaddresses)  <- c("Location","Points","lat","lon","GoogleAddress")
head(huntaddresses)

#write a new .csv file to your working directory
write_csv(huntaddresses, "huntaddresses.csv")
```

3. Now, for whatever reason, sometimes the API doesn't return all of the addresses (seems to depend on the time of day you run the query!) If you have not managed to get all of the addresses from the first run of this code, then you can download my effort from here:

https://www.dropbox.com/s/2cbu2ux9ddy9c0l/huntaddresses.csv?dl=0 (https://www.dropbox.com/s/2cbu2ux9ddy9c0l/huntaddresses.csv?dl=0)

```
huntaddresses <- read.csv("https://www.dropbox.com/s/2cbu2ux9ddy9c0l/huntaddresses.csv?ra
w=1")
```

3 - Tips for Success

a) Tips for the (**D**)[ArcGIS] Side

Tools you might need to use: *When Adam says might, He means almost certainly!*

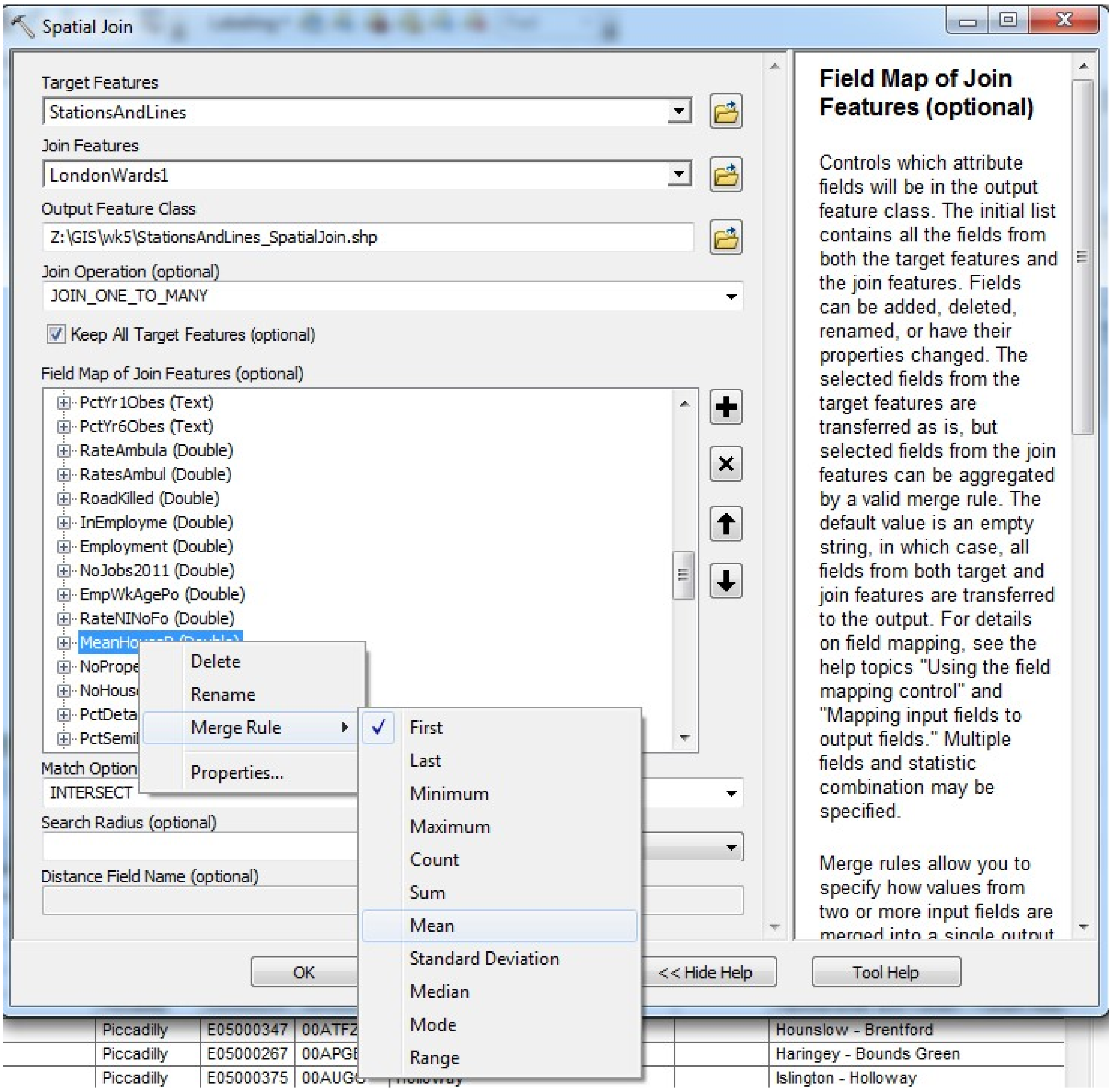
- Editor toolbar Spatial Join (one-to-many and one-to-one)
- KML to Layer
- Buffer
- Repair Geometry
- Intersect
- Dissolve
- Project
- Select by Attribute

Things you will need to consider:

1. Think carefully about choosing the appropriate ward boundaries to match your data as there will be several candidates for you to choose from. Dont use generalised or clipped boundaries and select the appropriate time period

2. Your data does not contain any information for the wards in the City of London so you will need to either edit your ward boundaries (using the editor toolbar. see the appendix of this document) and merge all of the wards in the city of London into a new City of London object. While your editing session is open, you will also be able to edit the attribute table so that the code for the City of London matches the code in your data.

3. In order to obtain variable summaries, you will probably need to make use of the Summarize facility when right-clicking a field in an attribute table
4. If you cant find a tool, use the search facility in Arc
5. When you carry out a spatial join, think about whether you should use a one-to-one or one-to-many join. You will probably need to use both in this task.
6. When you carry out a spatial join, you can also specify a merge rule by right-clicking on the field you are interested in. The default is First which is not that good for most joins a Mean and Sum will probably be more useful in this task! It might also be useful to not join all fields, but only those of use, deleting others with the A button.


Spatial Join

Target Features: StationsAndLines

Join Features: LondonWards1

Output Feature Class: Z:\GIS\wk5\StationsAndLines_SpatialJoin.shp

Join Operation (optional): JOIN_ONE_TO_MANY

☑ Keep All Target Features (optional)

Field Map of Join Features (optional)

```
PctYr10bes (Text)
PctYr60bes (Text)
RateAmbula (Double)
RatesAmbul (Double)
RoadKilled (Double)
InEmployme (Double)
Employment (Double)
NoJobs2011 (Double)
EmpWkAgePo (Double)
RateNINoFo (Double)
MeanHouseP (Double)
NoPrope
NoHous
PctDeta
PctSemi
```

Match Option: INTERSECT

Search Radius (optional)

Distance Field Name (optional)

Field Map of Join Features (optional)

Controls which attribute fields will be in the output feature class. The initial list contains all the fields from both the target features and the join features. Fields can be added, deleted, renamed, or have their properties changed. The selected fields from the target features are transferred as is, but selected fields from the join features can be aggregated by a valid merge rule. The default value is an empty string, in which case, all fields from both target and join features are transferred to the output. For details on field mapping, see the help topics "Using the field mapping control" and "Mapping input fields to output fields." Multiple fields and statistic combination may be specified.

Merge rules allow you to specify how values from two or more input fields are merged into a single output

Delete
Rename
Merge Rule
Properties...
✓ First
Last
Minimum
Maximum
Count
Sum
Mean
Standard Deviation
Median
Mode
Range
OK
<< Hide Help
Tool Help
Piccadilly | E05000347 | 00ATF2 | Hounslow - Brentford
Piccadilly | E05000267 | 00APGE | Haringey - Bounds Green
Piccadilly | E05000375 | 00AUG | Islington - Holloway

b) Tips for the R fans!

1. Convert all of your spatial objects to `sf` (Simple Features) objects - https://cran.r-project.org/web/packages/sf/ (https://cran.r-project.org/web/packages/sf/) it will make some of the aggregation easier and you will be able to make use of some of the neat `geos_uniary` functions such as `st_buffer()` for drawing buffers or the `geos_binary_pred` functions such as `st_intersects` for detecting where your points interset polygons, for example.

2. Use some of these online guides to help you:

http://strimas.com/r/tidy-sf/ (http://strimas.com/r/tidy-sf/)

https://bookdown.org/robinlovelace/geocompr/ (https://bookdown.org/robinlovelace/geocompr/)

https://cran.r-project.org/web/packages/sf/vignettes/sf1.html (https://cran.r-project.org/web/packages/sf/vignettes/sf1.html)

https://cran.r-project.org/web/packages/sf/vignettes/sf2.html (https://cran.r-project.org/web/packages/sf/vignettes/sf2.html)

https://cran.r-project.org/web/packages/sf/vignettes/sf3.html (https://cran.r-project.org/web/packages/sf/vignettes/sf3.html)

https://cran.r-project.org/web/packages/sf/vignettes/sf4.html
(https://cran.r-project.org/web/packages/sf/vignettes/sf4.html)

Appendix (a) Merging the City of London Wards into a single zone: D(Arc)side GIS Style

1. Start ArcMap and add your 2011 Ward boundary data (downloaded from either the UK Data Service or ONS Geoportal dont get the Census Merged Wards, just Wards E+W) to a new map document. Or the clean set I've put on Moodle…
2. Right-click on your layer and select Data > Export Data. Save your data as a new file geodatabase feature class and give it an appropriate name we are doing this as we will be messing around with boundary polygons and it's good practice not to ruin the original file!
3. Add your newly saved feature class your map. Open up the attribute table and sort the data by the column WD11CDO (right-click the header column and sort ascending). As the borough code for the City of London is 00AA, all of the Wards within will now be sorted to the top of the file.
4. Left-click the small box to the left of the first row this should highlight the first row in blue (and the polygon on the map). Scroll down a few rows until you find the last City of London ward this should be 00AAGB, Walbrook. Hold down shift and click the small box to the left of this row. All rows above should now be highlighted in blue (and this should show on the map).
5. We are now ready to dissolve these boundaries into a single City of London zone. If you have not already done so, activate the Editor toolbar: Customize > Toolbars > Editor. On the Editor toolbar, select Start Editing from the dropdown menu.
6. If you have other layers open in your map, you may have to select the correct layer to edit.
7. In the editor dropdown menu, Merge should now be active. Click Merge. Select the first feature as your merge feature and click OK.
8. You should now see on your map that the ward boundaries have dissolved and in your attribute table, only the first row. Aldersgate, should be left.
9. In your attribute table, Click in the cells for WD11CD, WD11CDO and WD11NM and manually edit the codes so that for this new polygon, WD11CD = E09000001, WD11CDO = 00AA and WD11NM = City of London.
10. We are now ready to join your London data to your new boundaries
11. Add your LondonData.xls layer to your map (not the .csv file this, for some unknown reason might cause problems!). Join the xls data to your newly edited boundary data, keeping only the matching records. Export your new layer as a new geodatabase feature class in a new file geodatabase for this week practical.
12. In the Editor drop down menu, select Save Edits and then Stop Editing.

Appendix (b) Merging the City of London Wards into a single zone: R(ebel) Style

```r
library(sf)
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.5.1
```

```r
library(rgdal)
#Download the Ward Boundaries from Moodle
LondonWards <- readOGR("LondonWardsBoundaries/LondonWardsNew.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\AnaBasiri\Teaching\GISModule2018AnaBasiri\Session7\Treasure_Hunt_Task\Lond
onWardsBoundaries\LondonWardsNew.shp", layer: "LondonWardsNew"
## with 649 features
## It has 7 fields
```

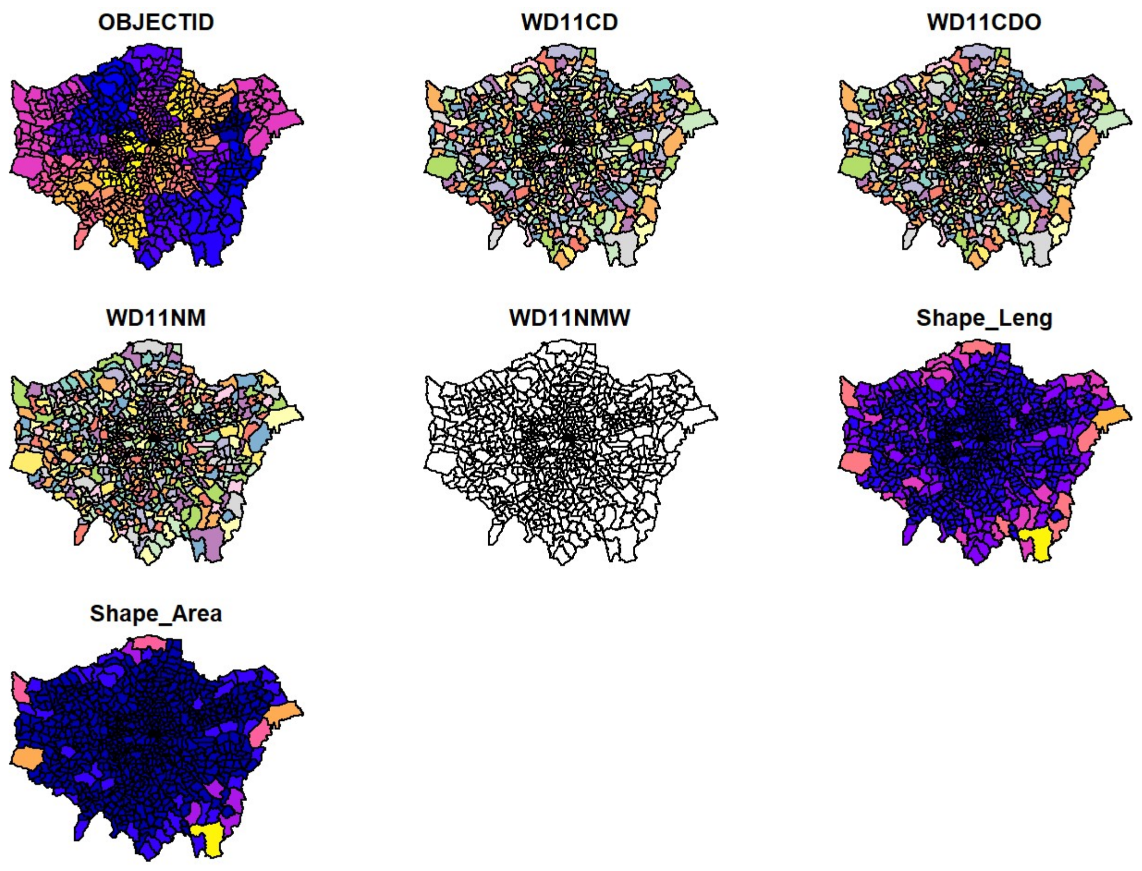
```
LondonWardsSF <- st_as_sf(LondonWards)


#cut the city out - this just happens to be the first 25 rows, usefully!
city <- LondonWardsSF[1:25,]
city$agg  <- 1


#merge all of the boundaries together so we've got a single object
#cityagg <- city %>% group_by(city$agg) %>% summarise()

#disolve the ward boundaries and just leave the first one as the city of London
LondonWards_dis <- aggregate(LondonWardsSF, by = LondonWardsSF, FUN = first)

#merge them back together into a new object
LondonWards_new <- rbind(LondonWards_dis, LondonWardsSF[26:649,])
plot(LondonWards_new)
```



```
#finally, update the codes for the city of London so that they match your
#LondonWards_new[1,2] <- as.character("E09000001")
#LondonWards_new[1,3] <- as.character("00AA")

#LondonWards_new[1,4] <- as.character("City of London")
```