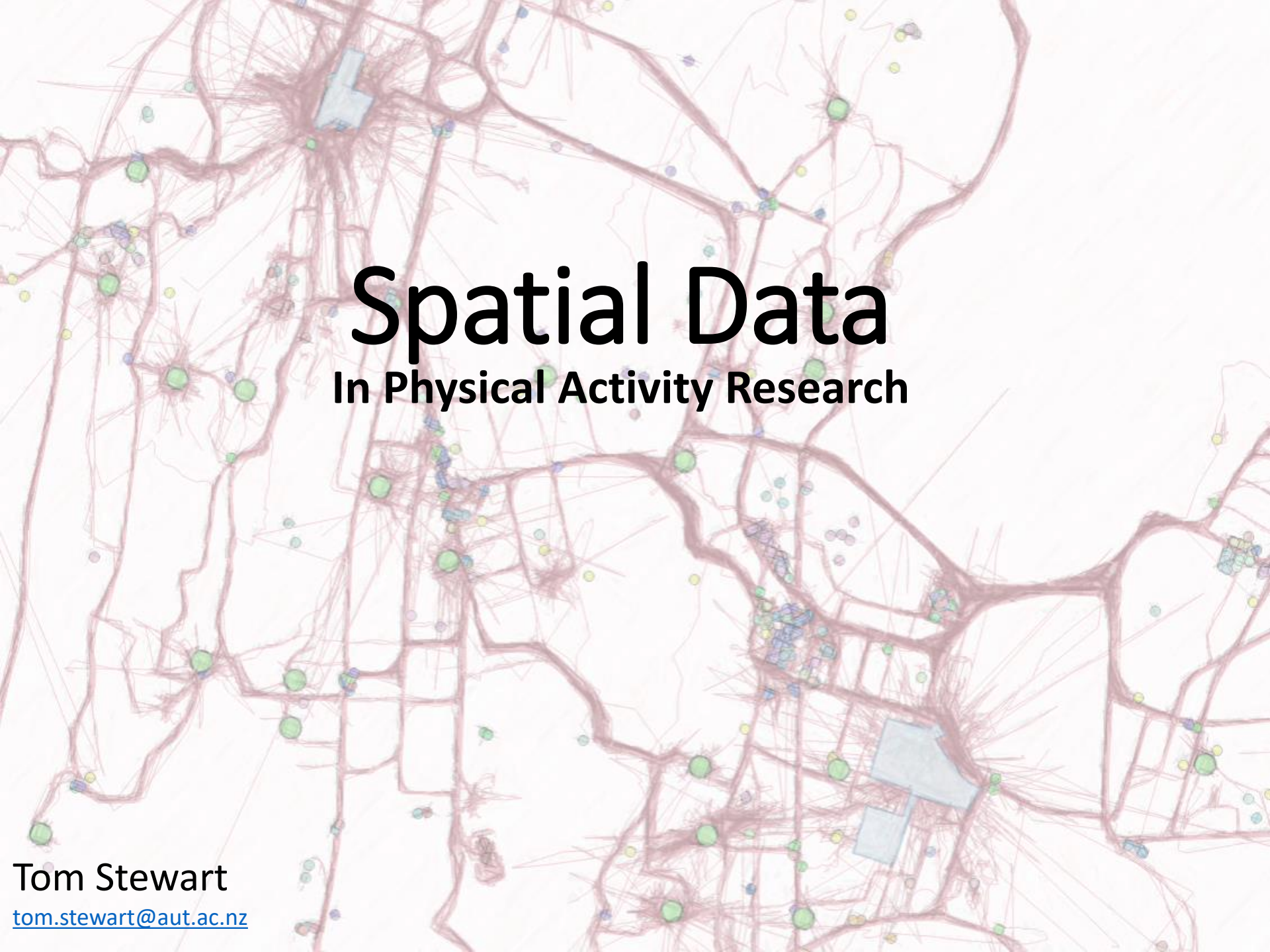


Spatial Data

In Physical Activity Research

Tom Stewart

tom.stewart@aut.ac.nz

Plan

- Why spatial data?
- Spatial data types
- Brief spatial data principles
- A bit of code ☺

Physical activity context

Occupational

Household

Leisure

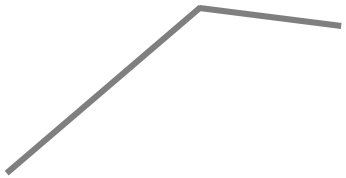
Transportation

- Research questions
- Interventions
- Policy

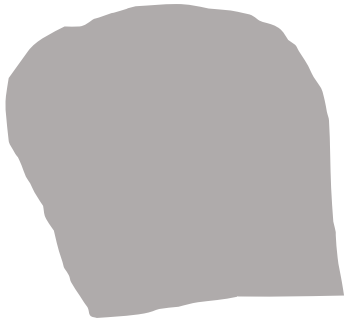
Spatial data (vector)

Point

Line

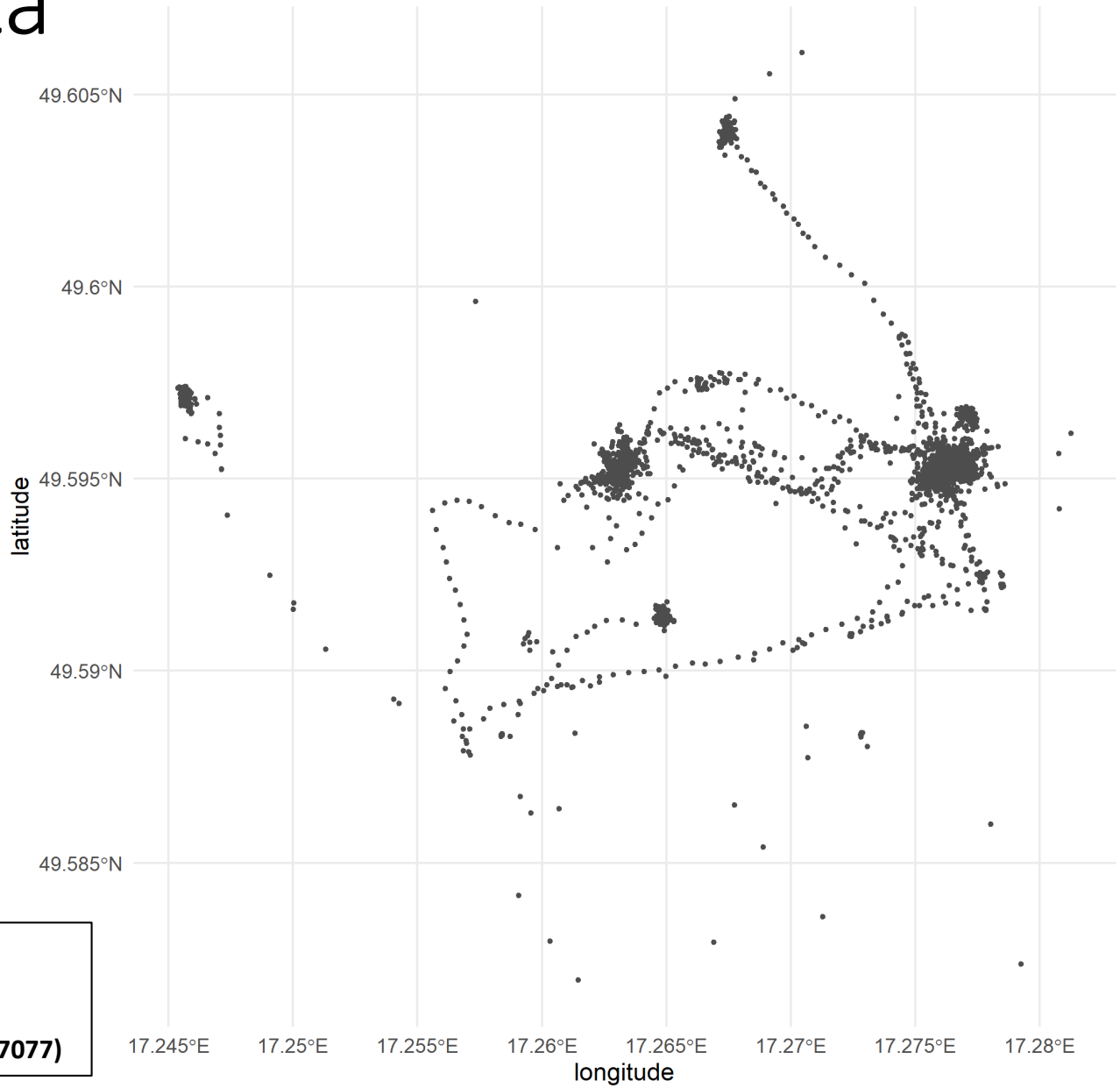


Polygon



Spatial data

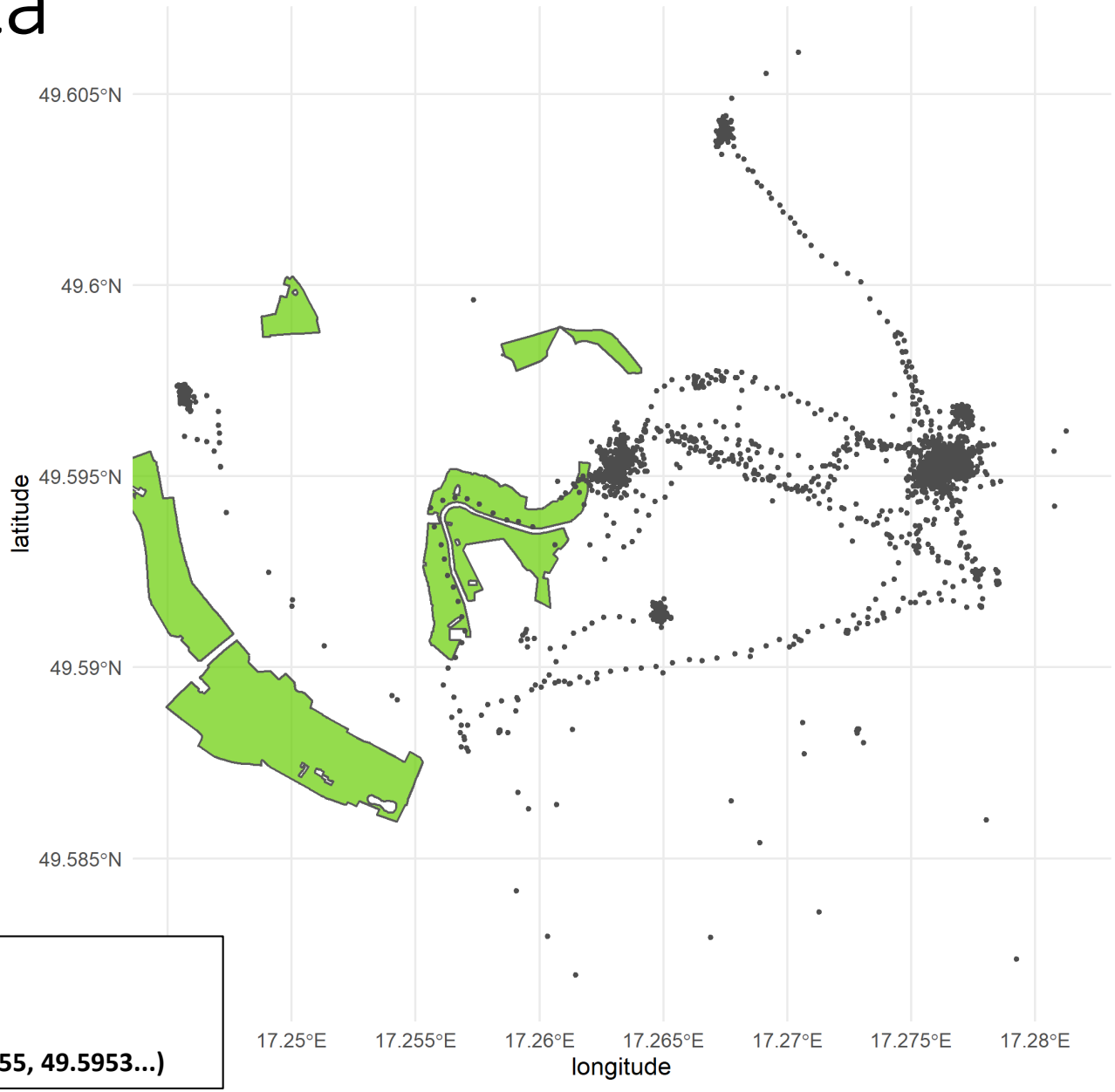
- Individual-level




```
# A tibble: 1 x 2
  datetime            geometry
  <dttm>              <POINT [°]>
  2014-05-26 11:51:00 (17.29771, 49.57077)
```

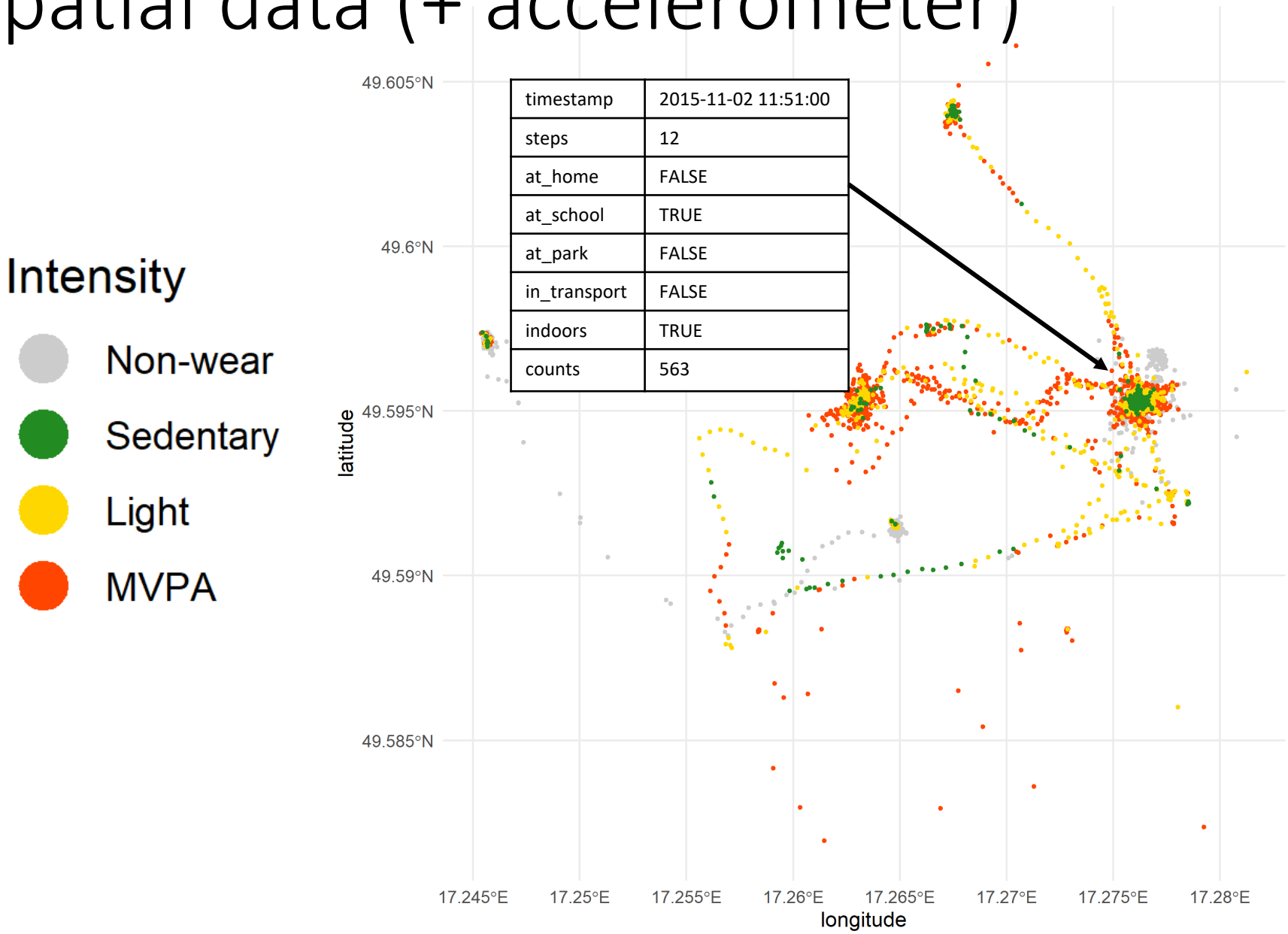
Spatial data

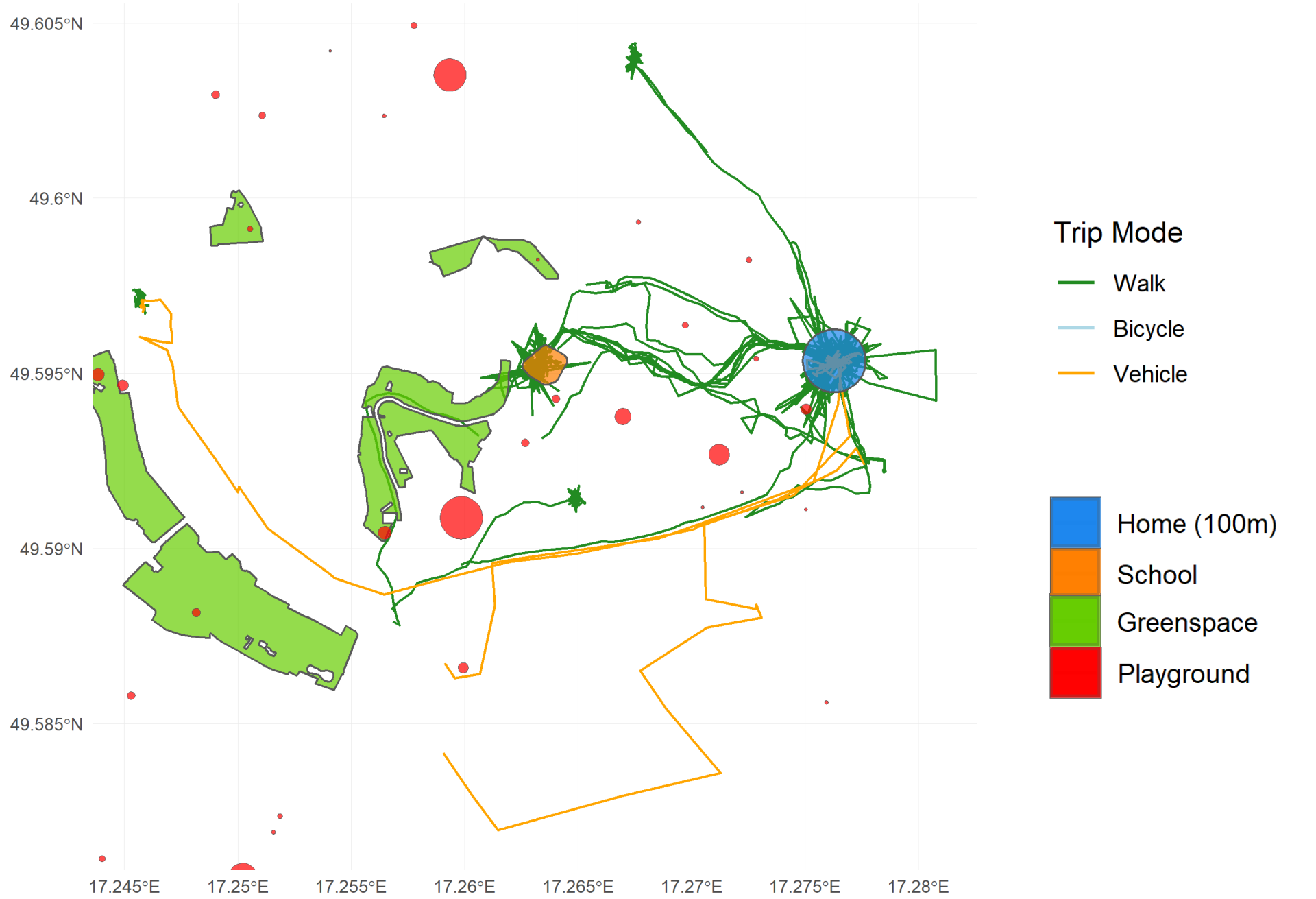
- Macro-level

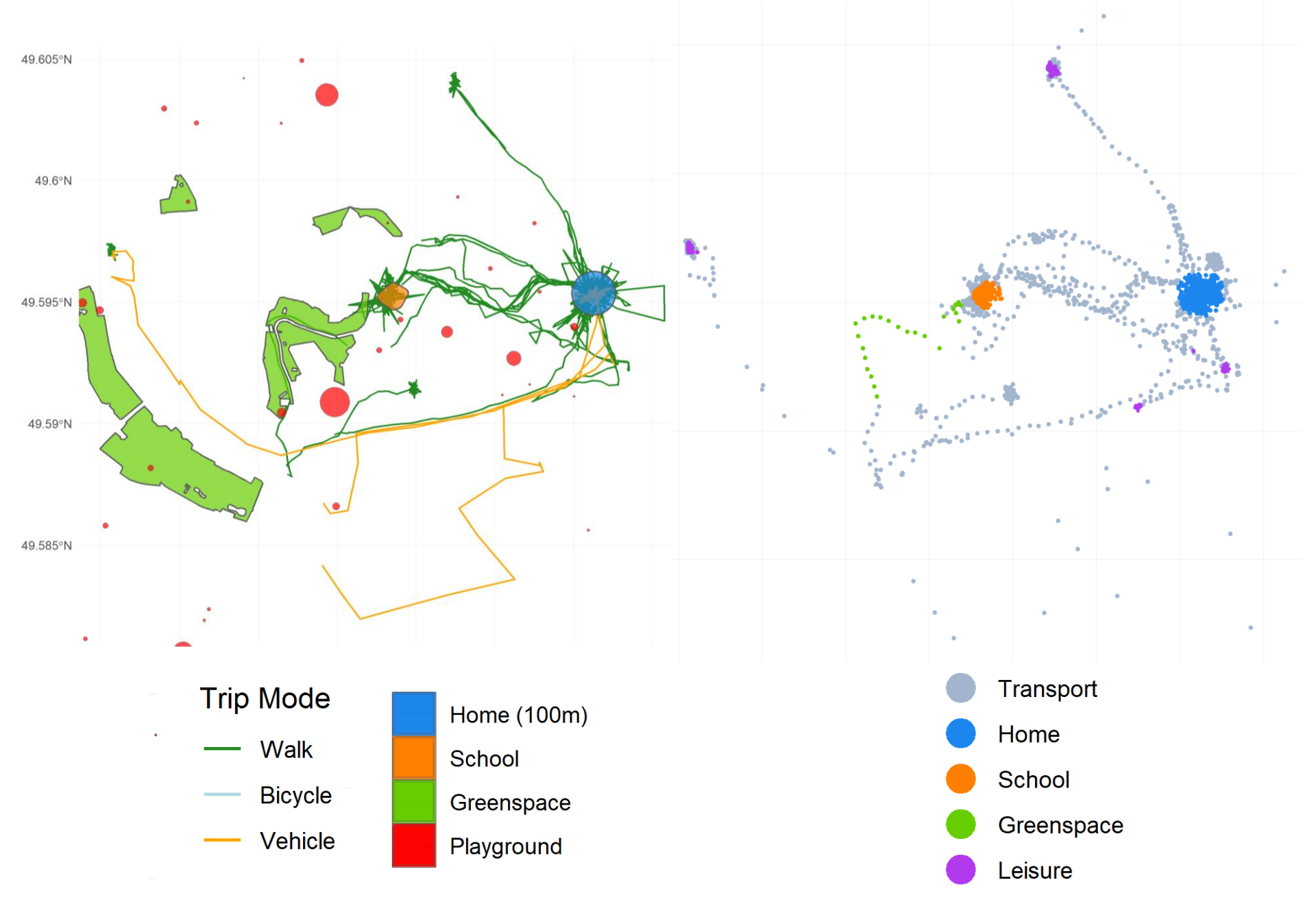



```
# A tibble: 1 x 2
  ID        geometry
  <dbl>     <MULTIPOLYGON [°]>
  1         (17.24452, 49.59539; 17.24455, 49.5953...)
```

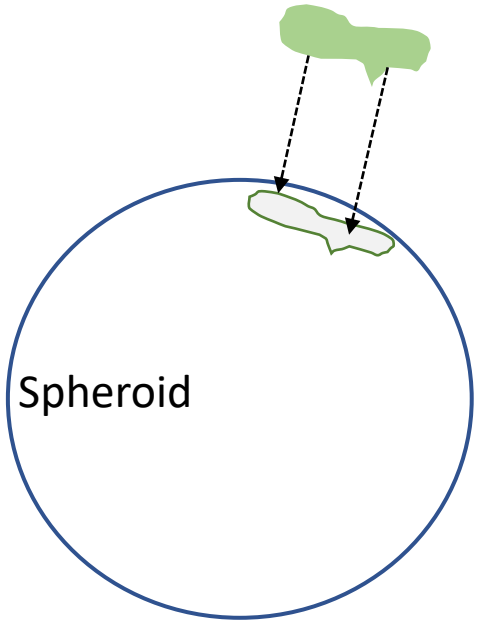
Spatial data (+ accelerometer)

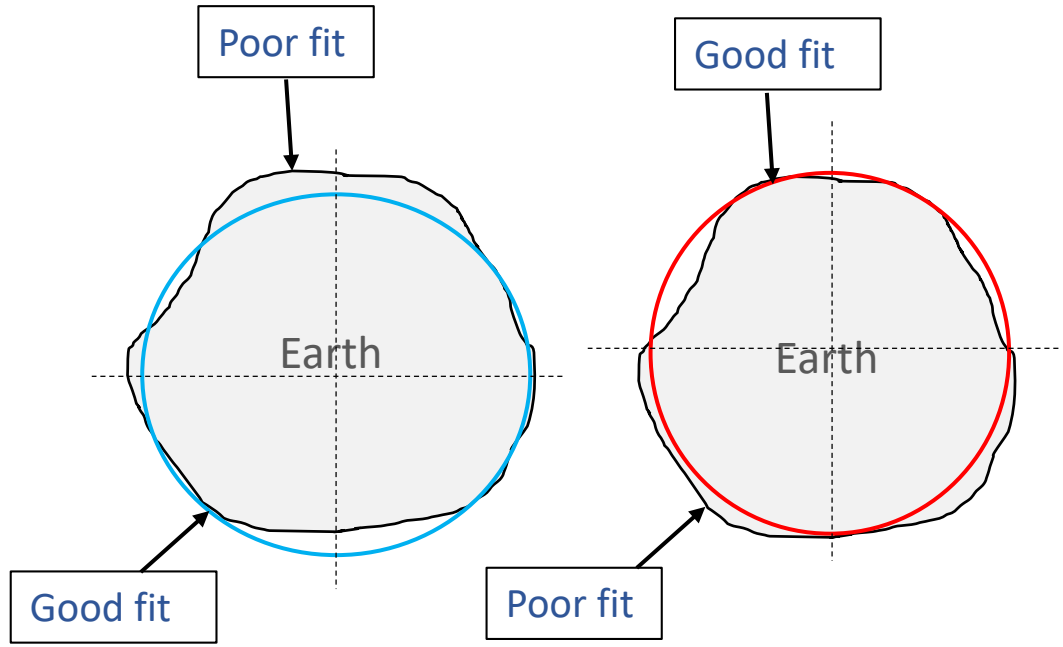



49.605°N
49.6°N
49.595°N
49.59°N
49.585°N
17.245°E
17.25°E
17.255°E
17.26°E
17.265°E
17.27°E
17.275°E
17.28°E
Trip Mode
Walk
Bicycle
Vehicle
Home (100m)
School
Greenspace
Playground


49.605°N
49.6°N
49.595°N
49.59°N
49.585°N
Trip Mode
Walk
Bicycle
Vehicle
Home (100m)
School
Greenspace
Playground
Transport
Home
School
Greenspace
Leisure

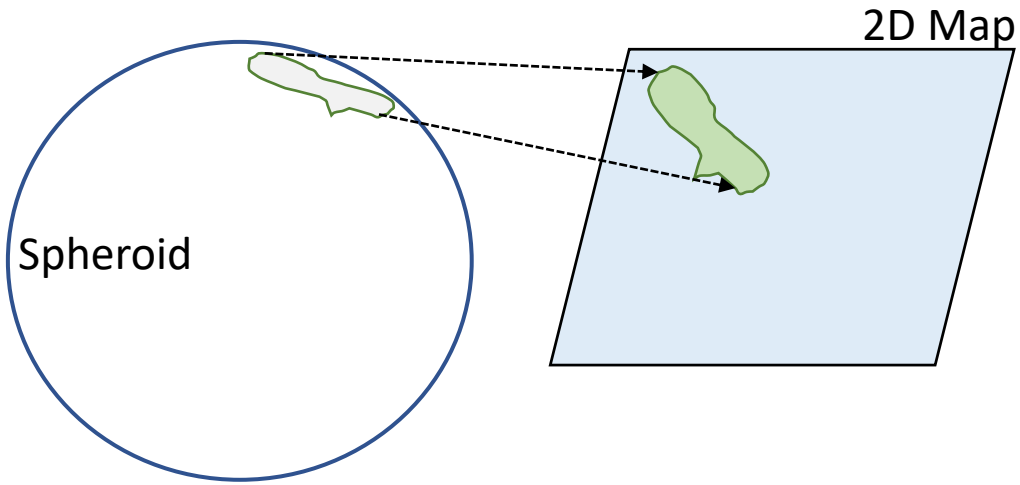
Geographic Coordinate Systems





The Global Positioning System
WGS 84 (ESPG:4326)

Projected Coordinate Systems



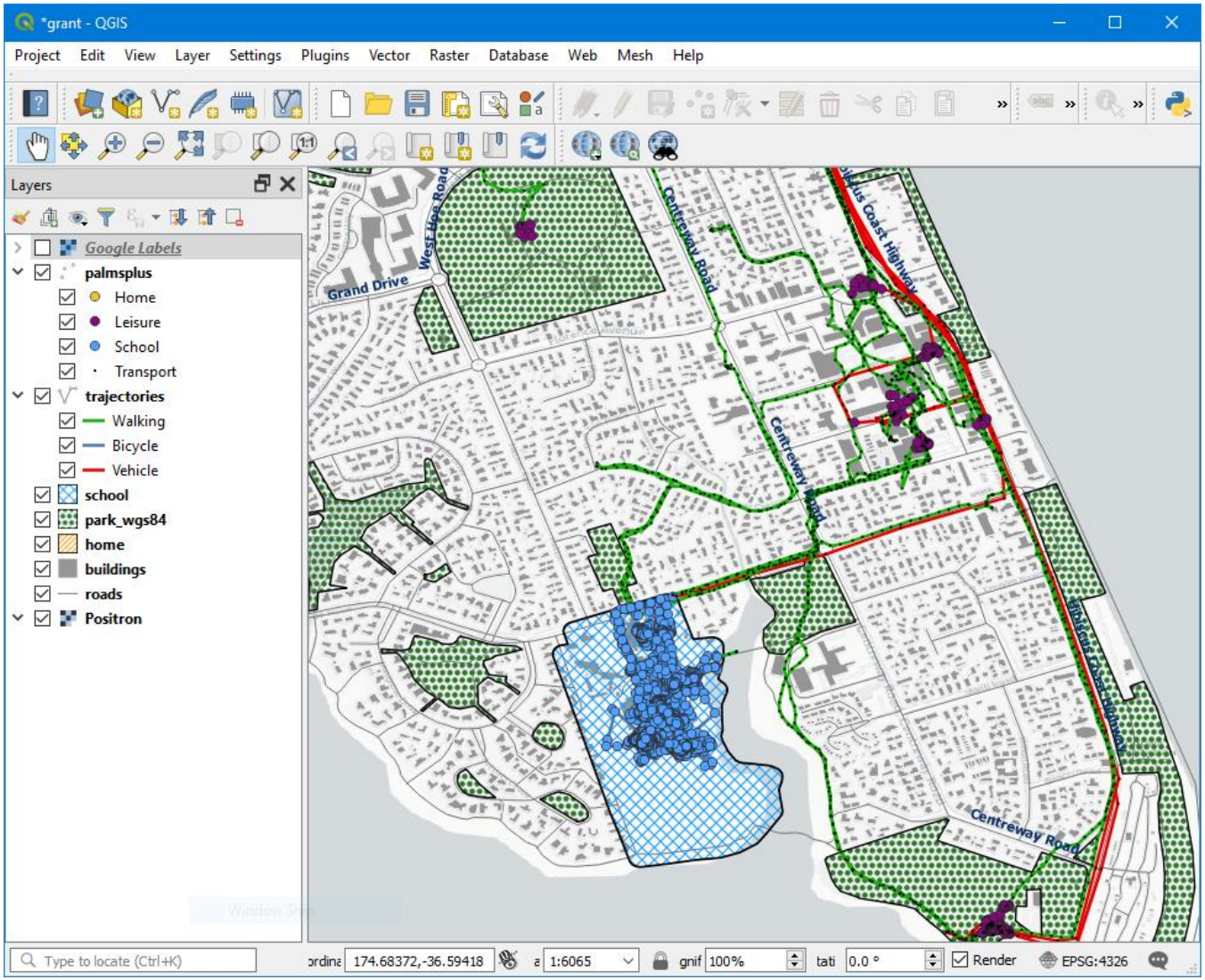
New Zealand

NZGD2000 (ESPG:2193)

Software

ArcGIS

QGIS



R packages for spatial data

- sp
- sf (https://github.com/r-spatial/sf)

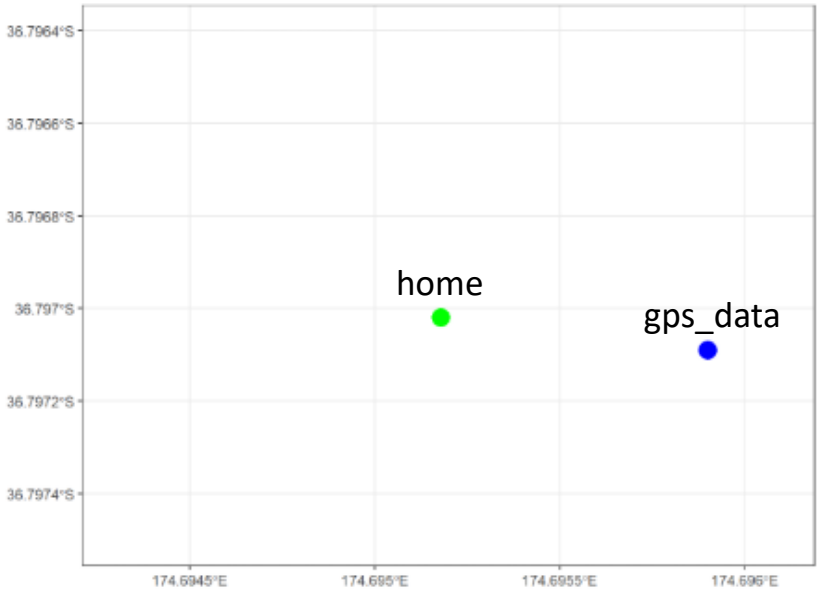
- Mapview

- RQGIS

- rpostgis

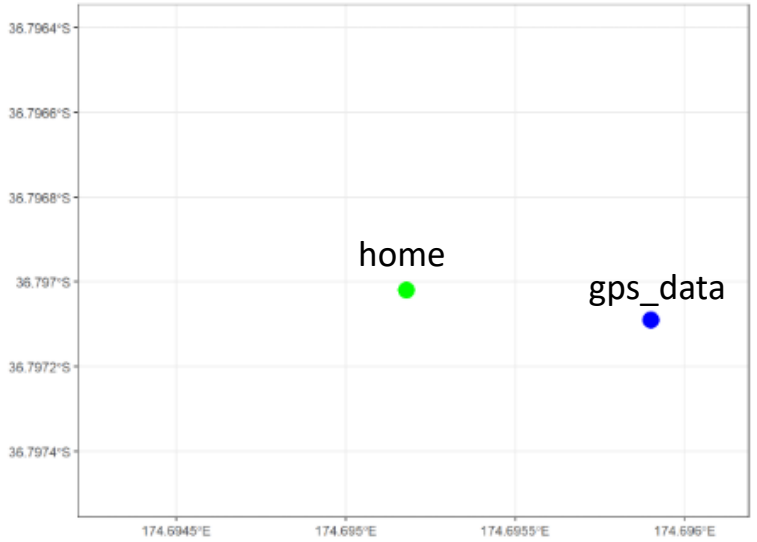
```r
# Read GPS data (single point)
gps_data <- read_csv('test.csv') %>%
  st_as_sf(coords = c('lon', 'lat'), crs = 4326)
```

```r
# Read in a shapefile (single point)
home <- read_sf('home.shp')
```



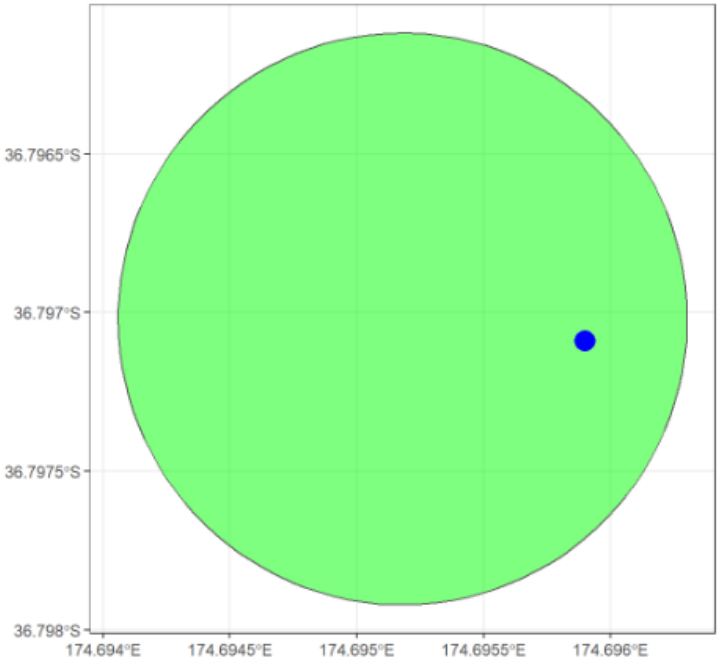
```r
# Read GPS data (single point)
gps_data <- read_csv('test.csv') %>%
  st_as_sf(coords = c('lon', 'lat'), crs = 4326)
```

```r
# Read in a shapefile (single point)
home <- read_sf('home.shp')
```




```r
# Buffer the point by 100 m
home %<>%
  st_transform(2193) %>%
  st_buffer(100) %>%
  st_transform(4326)
```

```r
ggplot() +
  geom_sf(data = home, fill = alpha('green', 0.5)) +
  geom_sf(data = gps_data, color = 'blue', size = 5) +
  theme_bw()
```




```r
# Check if point is inside polygon
st_contains(home, gps_data)
```

```
> [1] TRUE
```

The future

- Physical activity type (not just count-based intensity)
- Spatial patterns of behaviour (e.g. before / after)
- Physical activity with others

- Software to make all of this happen!

Getting started with palmsplusr

2018-01-17

Contents

Loading the PALMS dataset
Building palmsplus
Building days
Building trajectories
Building multimodal trajectories
Saving geometry and results

Loading the PALMS dataset

A PALMS dataset (in csv format) is read in using the `read_palms()` function. This function checks that all required column names are present before converting the csv file to a simple features (spatial) object. If any columns are missing you will receive an error message. For a list of required column names, please see the `read_palms()` help file. This example will use the data built into this package, which has been collected from one participant.

```
library(palmsplusr)

palms <- read_palms(system.file("extdata", "one_participant.csv", package =
"palmsplusr"))
names(palms)
#>  [1] "identifier"          "datetime"            "dow"
#>  [4] "fixtypecode"         "iov"                 "tripnumber"
#>  [7] "triptype"            "tripmot"             "activity"
#> [10] "activityintensity"   "activityboutnumber"  "sedentaryboutnumber"
#> [13] "geometry"
```

This `palms` object contains 13 columns. Notice how the `lon` and `lat` columns that were present in the csv have been replaced by a `geometry` column. This is POINT geometry, as each row in `palms` represents a point. You can plot this data to look at the distribution of points in space. Here two columns are plotted:

```
plot(palms[, c("activity", "tripnumber")])
```

Getting started with palmsplusr

2018-01-17

Loading the PALMS dataset

A PALMS dataset (in csv format) is read in using the `read_palms()` function. This function checks that all required column names are present before converting the csv file to a simple features (spatial) object. If any columns are missing you will receive an error message. For a list of required column names, please see the `read_palms()` help file. This example will use the data built into this package, which has been collected from one participant.

```
library(palmsplusr)

palms <- read_palms(system.file("extdata", "one_participant.csv", package =
"palmsplusr"))
names(palms)
#>  [1] "identifier"          "datetime"            "dow"
#>  [4] "fixtypecode"         "iov"                 "tripnumber"
#>  [7] "triptype"            "tripmot"             "activity"
#> [10] "activityintensity"   "activityboutnumber"  "sedentaryboutnumber"
#> [13] "geometry"
```

This `palms` object contains 13 columns. Notice how the `lon` and `lat` columns that were present in the csv have been replaced by a `geometry` column. This is POINT geometry, as each row in `palms` represents a point. You can plot this data to look at the distribution of points in space. Here two columns are plotted:

```
plot(palms[, c("activity", "tripnumber")])
```

Contents

Loading the PALMS dataset
Building palmsplus
Building days
Building trajectories
Building multimodal trajectories
Saving geometry and results

Other useful links

- https://github.com/r-spatial
- https://geocompr.robinlovelace.net/