ECON6027 4b

SPATIAL DESCRIPTIVE SUMMARY MEASURES: APPLICATION

Packages you need

sf

aspace

- https://cran.r-project.org/web/packages/aspace/index.html

Load and clean data

```
> sg_regions =
st_read("MP14_PLNG_AREA_WEB_PL.shp")

> table(st_is_valid(sg_regions))

FALSE  TRUE

    8    47


> sg_regions = st_make_valid(sg_regions);
table(st_is_valid(sg_regions))

TRUE

  55
```

Load and clean data

```
> SLC = st_read("SPF_PSLC.shp");
table(st_is_valid(SLC))


TRUE
  48
```

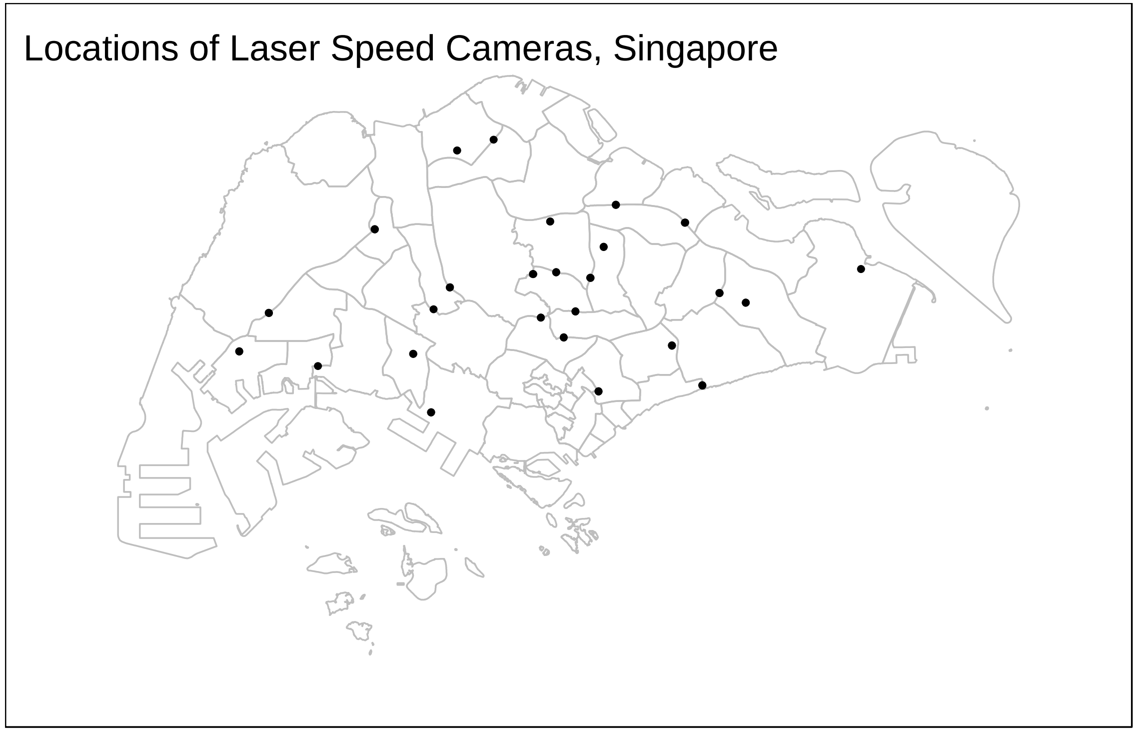
CRS Matters

```
> st_crs(sg_regions)$proj

[1] "tmerc"

> st_crs(SLC)$proj

[1] "tmerc"

> isTRUE(all.equal(st_crs(sg_regions),
st_crs(SLC)))

[1] TRUE
```

Both are projected with the same CRS. So, we are good to go!

Initial maps

```
> (sg_map =
tm_shape(sg_regions) +
tm_polygons(col="REGION_N",
alpha=0.5, border.col =
"grey", legend.show = F))

> (LC_locations = sg_map +
tm_shape(SLC) +
tm_dots(size=0.1) +
tm_layout(title="Locations
of Laser Speed Cameras,
Singapore", inner.margins =
0.1))
```



Descriptive Measures

Extract coordinates

Extract coordinates to compute descriptive measures:

```
> LC_coords = st_coordinates(SLC)

> class(LC_coords)

[1] "matrix"
```

Spatial measures of central tendency

1. Spatial mean

```
> mean_centre(id=1, filename="LC_mean.txt", points=LC_coords)
  id CENTRE.x CENTRE.y
1  1  27775.2 37274.24
```

Output of mean_centre: "meanatt" "meanloc" "r.mean"

2. spatial median

```
> median_centre(id=2, filename="LC_median.txt", points= LC_coords)
  id median.x median.y
1  2  28788.8  37013.8
```

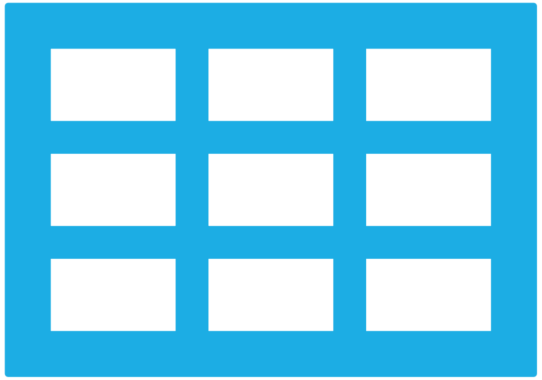
Output of median_centre: "medianatt" "medianloc” "r.median"

Spatial measures of dispersion

3. Standard distance

4. standard deviational ellipse

3. Standard Distance



```
> calc_sdd(id=3, filename = "LC_sdd.txt", calccentre = T, weighted=F, points = LC_coords)
```

Output of calc_sdd: "r.SDD" "sddatt" "sddloc"

Inspect the object "sddloc"

```
> head(sddloc)
```

It is a set of locations that make up the standard distance disc. We need to transform this to sf for mapping purposes.

Cast “sddloc” data frame to a sf “LINESTRING”

```
> class(sddloc)
```

[1] "data.frame”

Convert the data.frame into a linestring sf object.

```
> (LC_sdd_line = sddloc %>% st_as_sf(coords = c("x", "y"), crs=CRS)
%>% st_combine() %>% st_cast("LINESTRING"))
```

*%>% in the command is called a pipe operator. It is useful when you want to combine many commands where the output of the preceding command is the input of the other.

The linestring is then converted to a sf object.

> (LC_std_dist = st_sf(sddatt, geom=st_geometry(LC_sdd_line)))

Digression: pipe operator

The following code chunk…

```
> (LC_sdd_line = st_as_sf(sddloc, coords = c("x", "y"), crs=CRS))

> (LC_sdd_line = st_combine(LC_sdd_line))

> (LC_sdd_line = st_cast(LC_sdd_line,"LINESTRING"))
```

… can be combined using the pipe operator as…

```
> (LC_sdd_line = sddloc %>% st_as_sf(coords = c("x", "y"),
crs=CRS) %>% st_combine() %>% st_cast("LINESTRING"))
```

4. Standard Deviational Ellipse

```
> calc_sde(id=4, filename = "LC_sde.txt", calccentre = T, weighted=F, points= LC_coords)

> class(LC_sde)
```

Output of calc_sde: "r.SDE" "sdeatt" "sdeloc"

Cast “sdeloc” to a “LINESTRING”

```
> (LC_sde_line = sdeloc %>% st_as_sf(coords = c("x", "y"), crs=CRS) %>% st_combine() %>% st_cast("LINESTRING"))

> (LC_std_ellps = st_sf(sdeatt, geom=st_geometry(LC_sde_line)))
```

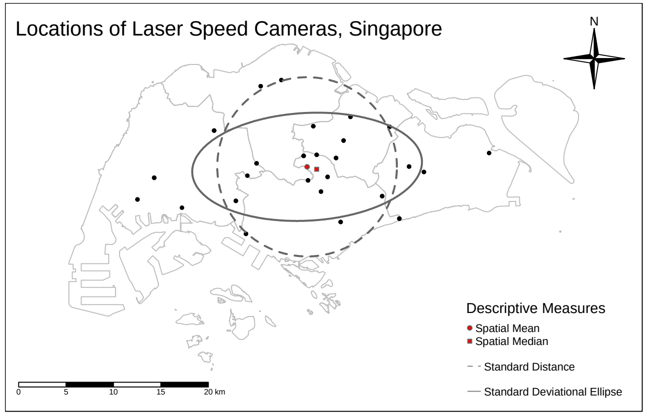
One more thing…

Convert mean-median locations to sf points (to mark in the map)

```
> mean_median = rbind(meanloc,medianloc) %>% data.frame() %>% st_as_sf(coords=c("x","y"), crs=CRS) # rbind is used to stack different objects

> mean_median$id = c("Spatial Mean","Spatial Median") # assign names

> mean_median # sf point object containing the mean and the median locations
```

Final plot

```
> LC_locations
+ tm_shape(LC_sde_line) + tm_lines(lwd=2)
+ tm_shape(LC_sdd_line) + tm_lines(lwd=2,
lty=2)
+ tm_shape(mean_median) + tm_dots(size=0.1,
shape="id", col="red",
title.shape="Descriptive Measures")
+ tm_add_legend(type="line", labels="Standard
Distance", lty=2)
+ tm_add_legend(type="line", labels="Standard
Deviational Ellipse", lty=1)
```



Exercise A

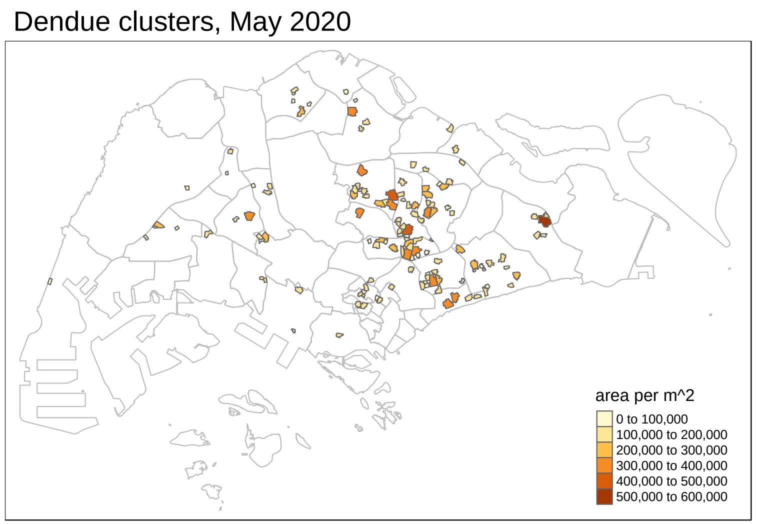
Add all the different kinds of speed cameras and redo the exercise.

Hint: rbind()

Note: all your function output files in the R environment such as r.SDE", "sdeatt", "sdeloc" will be overwritten each time you run the function. So, remember to assign the objects if you are running a multi layered analysis.

Weighted descriptive summary measures

Load

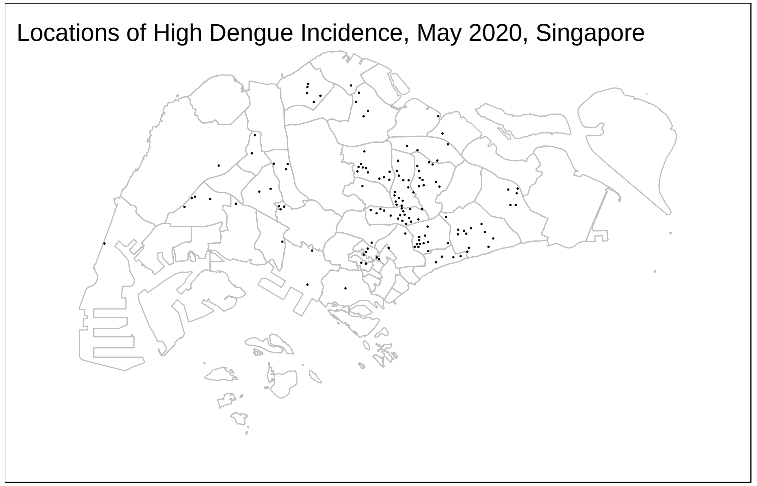



```
> dengue = st_read("dengue-clusters-may20.geojson")
%>% st_transform(crs=st_crs(sg_regions))

> table(st_is_valid(dengue))

> dengue = st_make_valid(dengue)
```

need to project data for st_centroid to work accurately (the original CRS is WGS 84)

Add area attribute (dengue cluster size) to use as weights

```
> dengue$area = st_area(dengue)

> sg_map + tm_shape(dengue) + tm_polygons(col="area")
+ tm_layout(main.title = "Dendue clusters, May 2020")
```



Convert clusters to points

```
> dengue_spots = st_centroid(dengue)

> dengue_coords =
st_coordinates(dengue_spots)


> (dengue_map = sg_map +
tm_shape(dengue_spots) + tm_dots()
+tm_layout(title="Locations of High
Dengue Incidence, May 2020,
Singapore", inner.margins = 0.1))
```

Assign weights

We will use the size of the cluster as weights.

```
> (dengue_area = st_area(dengue))
> class(dengue_area)
```

[1] "units"

The weight object required by aspace must be “unit” free. However, ”`dengue_area`” is in square meters which we will remove now. (if your weights are in frequencies, you don’t have a problem.)

```
> dengue_area = units::drop_units(st_area(dengue))
> class(dengue_area)
```

[1] "numeric"

Weighted descriptive summaries

```
> mean_centre(id=1,
filename="dengue_centre.txt", weighted = T,
weights=dengue_area, points = dengue_coords)
```

```
> calc_sdd(id=2, filename = "SDD_Dengue.txt",
calccentre = T, weighted = T, weights =
dengue_area, points = dengue_coords)
```

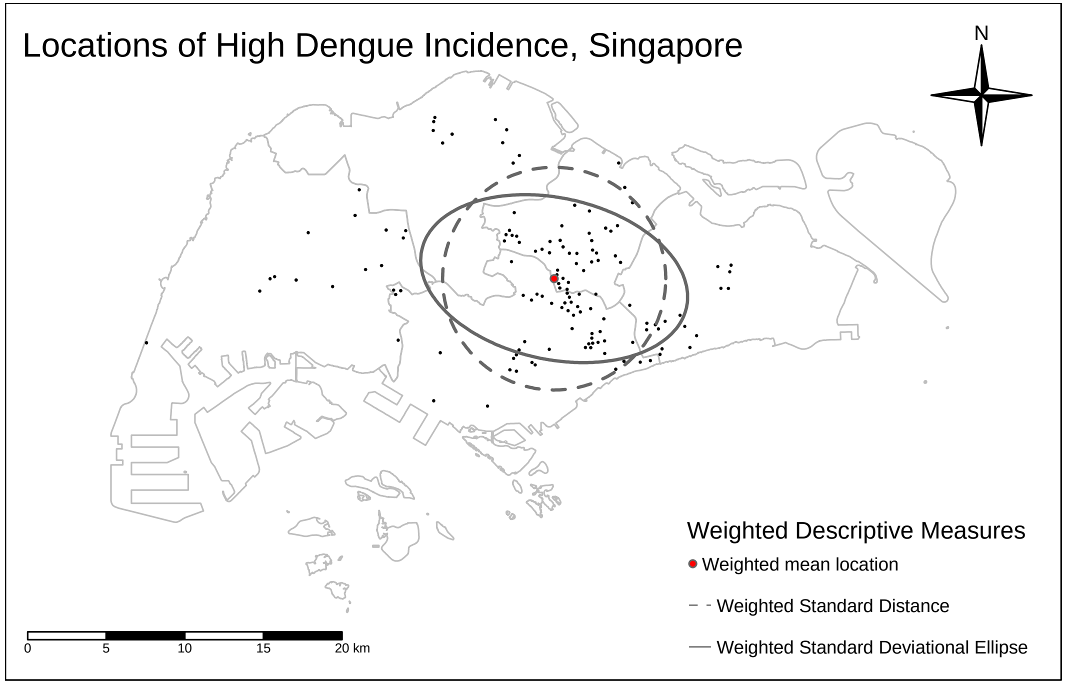
```
> calc_sde(id=3, filename = "SDE_Dengue.txt",
calccentre = T, weighted = T, weights =
dengue_area, points = dengue_coords)
```

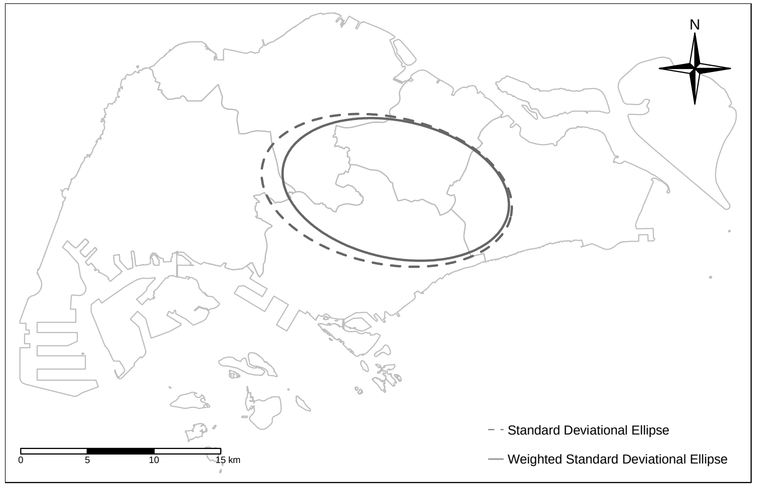
(notice this would overwrite all outputs from previous usage of these functions.)

Exercise B

Follow the remaining steps above to create a map like the one on the right.

Comment on the directional pattern of dengue clusters in Singapore in May 2020.





Exercise C

Redo the exercise without weights and compare the two standard deviational ellipses.

Take home points

Conduct spatial descriptive summary analysis using the "aspace" package.

Interpret the standard deviational ellipse and other spatial descriptive summary measures.

Important R functions

mean_centre()

median_centre()

calc_sdd()

calc_sde()

convert.to.shapefile()

write.shapefile()

References

Spatial Analysis by Tonny Oyana, 2nd edition, (2016), Chapter 3.

https://cran.r-project.org/web/packages/aspace/aspace.pdf

https://pro.arcgis.com/en/pro-app/tool-reference/spatial-statistics/h-how-directional-distribution-standard-deviationa.htm