ECON6027 project (Group 4)

2024-10-13

```r
library(aspace, quietly = T)
library(sf, quietly = T)
library(tmap, quietly = T)
library(spatstat, quietly = T)
library(ggplot2, quietly = T)
library(dplyr, quietly = T)
library(geojsonsf, quietly = T)
library(geojsonR, quietly = T)
```

We start by loading the shapefile of Singapore planning areas from data.gov.sg and the geojson file of MRT and LRT exits from data.gov.sg.

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

```
## Reading layer `MP14_PLNG_AREA_WEB_PL' from data source
##   `/Users/ch0002techvc/Downloads/EC627 Spatial Econometrics & Data Analysis/Project/MP14_PLNG_AREA_WE
##   using driver `ESRI Shapefile'
## Simple feature collection with 55 features and 12 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 2667.538 ymin: 15748.72 xmax: 56396.44 ymax: 50256.33
## Projected CRS: SVY21
```

```r
sg_regions = st_transform(sg_regions, crs=3414)
table(st_is_valid(sg_regions)) # there are 5 false entries
```

```
##
## FALSE  TRUE
##     8    47
```

```r
sg_regions = st_make_valid(sg_regions)
table(st_is_valid(sg_regions)) # now, all 55 entries are valid
```

```
##
## TRUE
##   55
```

```r
mrt_lrt_exits = geojson_sf("LTAMRTStationExitGEOJSON.geojson")
mrt_lrt_exits_sf = st_transform(mrt_lrt_exits, crs=3414)
table(st_is_valid(mrt_lrt_exits_sf)) # all entries are valid
```

```
##
## TRUE
##  563
```

```r
st_crs(mrt_lrt_exits_sf)$proj4string
```

```
## [1] "+proj=tmerc +lat_0=1.36666666666667 +lon_0=103.833333333333 +k=1 +x_0=28001.642 +y_0=38744.572 +
```

```r
isTRUE(all.equal(st_crs(sg_regions), st_crs(mrt_lrt_exits_sf))) #TRUE, we can proceed
```

```
## [1] TRUE
```

```r
table(is.na(mrt_lrt_exits_sf$geometry)) # no station exits with empty coordinates
```

```
##
## FALSE
##   563
```

```r
sg_map_bg = tm_shape(sg_regions) +
  tm_polygons(alpha=0.5, border.col="black", legend.show=F) +
  tm_fill(col = "white") +
  tm_scale_bar(position = c("right", "top"), width = 0.25, text.size = 0.4) +
  tm_compass(position = c("left", "top"), size =0.8, text.size = 0.7)
```

1. Spatial descriptive summary measures

Computations using the aspace package functions can only be carried out on cartesian coordinates given in matrix form. Thus, we will extract the coordinates of MRT and LRT exits.

```r
mrt_coords = st_coordinates(mrt_lrt_exits_sf)
mrt_coords = st_zm(mrt_coords, drop = T, what = "ZM")
class(mrt_coords)
```

```
## [1] "matrix" "array"
```

Measures of central tendency: We compute the mean and the median and assign them to a list object for later use.

```r
mean_mrt = calc_mnc(id=1, points=mrt_coords)
mean_mrt$LOCATIONS
```

```
##      id        x        y
## [1,]  1 28838.12 35702.44
```

```r
median_mrt = calc_mdc(id=2, points=mrt_coords)
median_mrt$LOCATIONS
```

```
##      id        x        y
## [1,]  2 29485.44 35021.24
```

In the following chunk, we compute the standard deviation distance and ellipse of mrt and lrt exits in Singapore.

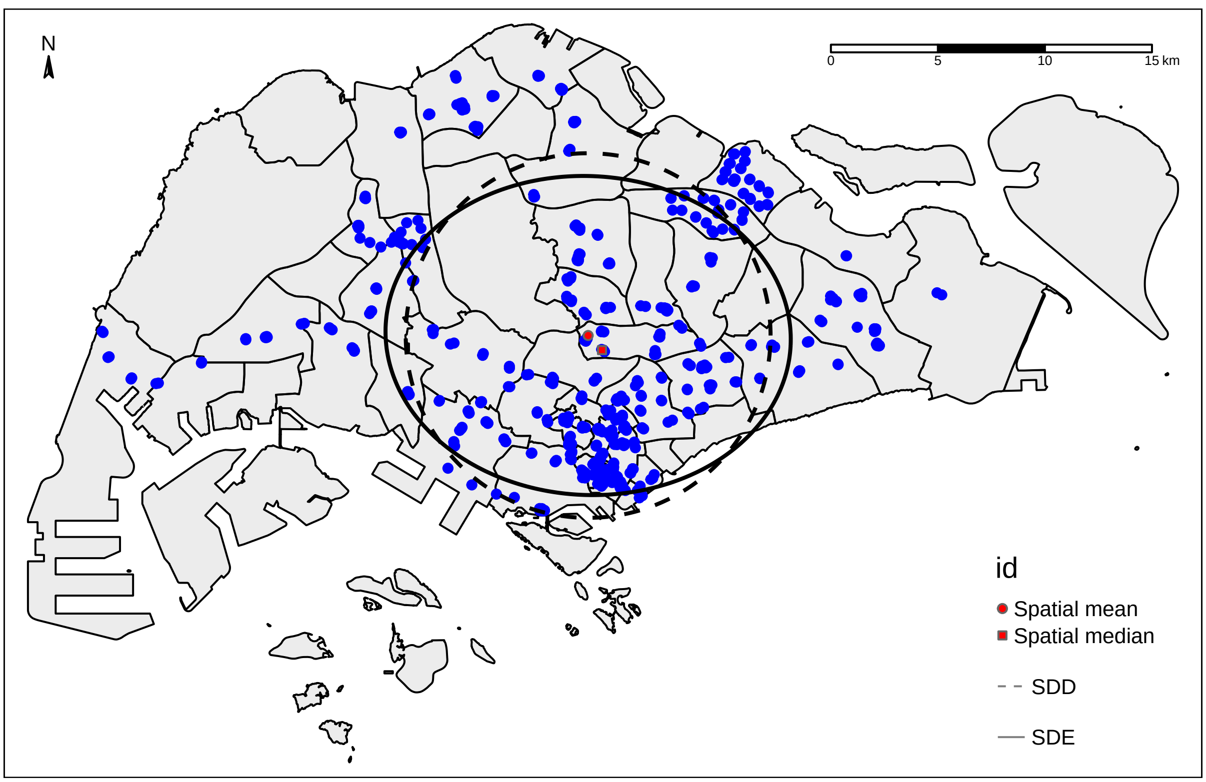
```r
sdd_mrt = calc_sdd(id=3, calccentre = T, weighted=F, points = mrt_coords)
sde_mrt = calc_sde(id=4, calccentre=T, weighted=F, points = mrt_coords)

mrt_sdd_line = sdd_mrt$LOCATIONS %>%
  st_as_sf(coords = c("x", "y"), crs=st_crs(mrt_lrt_exits_sf)) %>%
  st_combine() %>%
  st_cast("LINESTRING")
mrt_std_dist = st_sf(sdd_mrt$ATTRIBUTES, geom=st_geometry(mrt_sdd_line))

mrt_sde_line = sde_mrt$LOCATIONS %>%
  st_as_sf(coords = c("x", "y"), crs=st_crs(mrt_lrt_exits_sf)) %>%
  st_combine() %>%
  st_cast("LINESTRING")
mrt_std_ellps = st_sf(sde_mrt$ATTRIBUTES, geom=st_geometry(mrt_sde_line))
```

```r
mean_mrt$LOCATIONS; median_mrt$LOCATIONS #identify mean and median location, ID 1 is mean and ID 2 is m
```

```
##      id        x        y
## [1,]  1 28838.12 35702.44

##      id        x        y
## [1,]  2 29485.44 35021.24
```

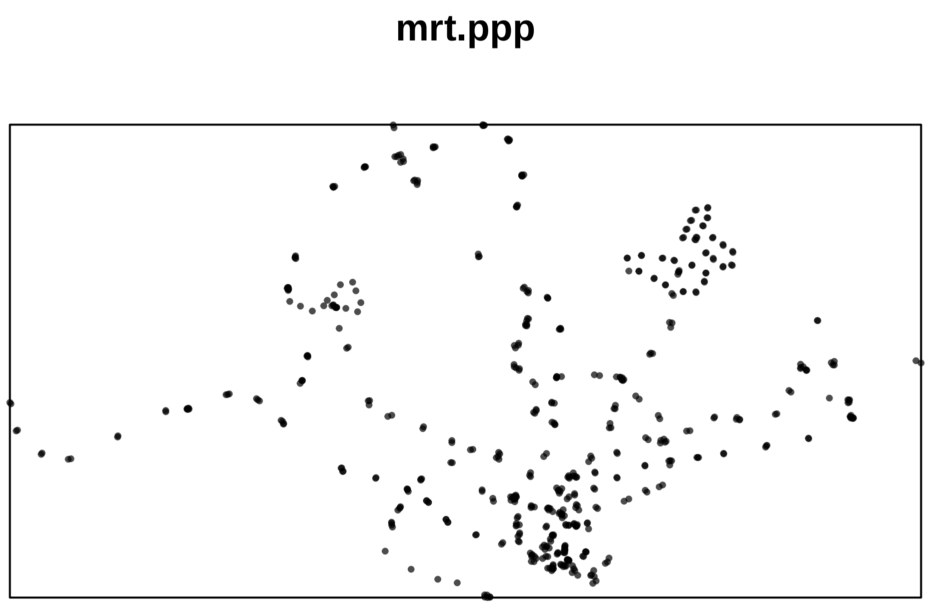
```r
mean_median = rbind(mean_mrt$LOCATIONS, median_mrt$LOCATIONS) %>% data.frame() %>%
  st_as_sf(coords=c("x","y"), crs=st_crs(mrt_lrt_exits_sf))
mean_median$id = c("Spatial mean","Spatial median")

sg_map_bg +
  tm_shape(mrt_lrt_exits_sf) + tm_dots(size=0.1, col= "blue") +
  tm_shape(mrt_std_ellps) + tm_lines(lwd=2) +
  tm_shape(mrt_std_dist) + tm_lines(lwd=2, lty=2) +
  tm_shape(mean_median) + tm_dots(size=0.1, shape="id", col="red") +
  tm_add_legend(type="line", labels="SDD", lty=2, ) +
  tm_add_legend(type="line", labels="SDE", lty=1)
```

```
## Warning: One tm layer group has duplicated layer types, which are omitted.
## To draw multiple layers of the same type, use multiple layer groups (i.e.
## specify tm_shape prior to each of them).
```



2. Quadrat count analysis

```r
mrt.ppp = as.ppp(mrt_lrt_exits_sf$geometry) #transform sf object to ppp
plot(mrt.ppp, pch=20, size=0.5)
```




```r
st_bbox(mrt_lrt_exits_sf)
```

```
##      xmin      ymin      xmax      ymax
##  6134.086 27499.697 45356.362 47865.923
```

```r
sg = st_union(sg_regions)
sg.owin = as.owin(sg)
class(sg.owin)
```

```
## [1] "owin"
```

```r
Window(mrt.ppp) = sg.owin
mrt.ppp # 563 points, no points out of the boundary
```

```
## Planar point pattern: 563 points
## window: polygonal boundary
## enclosing rectangle: [2667.54, 56396.44] x [15748.72, 50256.33] units
```

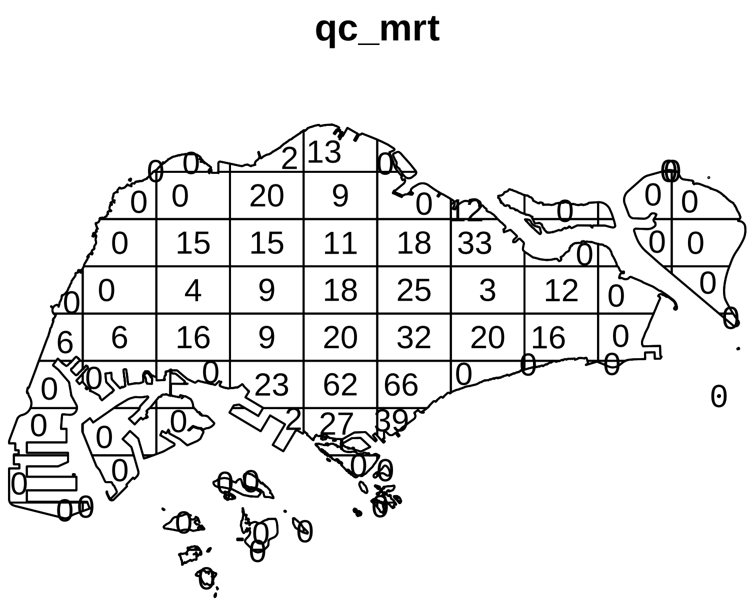
```r
table(duplicated(mrt.ppp)) # no duplicated entries, we can proceed
```

```
##
## FALSE
##   563
```

```r
qc_mrt = quadratcount(mrt.ppp, nx=10, ny=10)
class(qc_mrt)
```

```
## [1] "quadratcount" "table"
```

```r
plot(qc_mrt)
```



Chi2 (goodness of fit) test

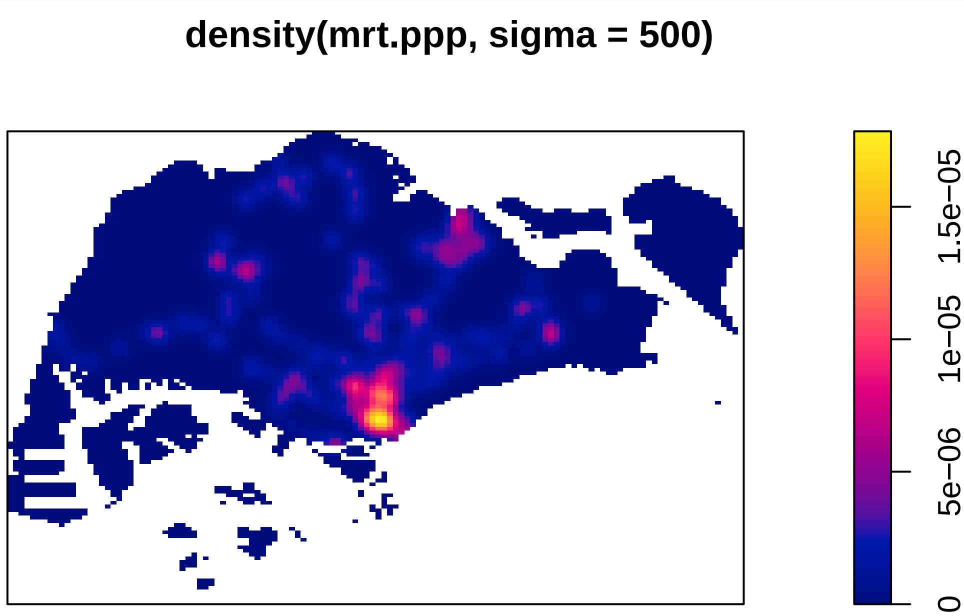
```
quad.test_mrt = quadrat.test(mrt.ppp, nx=10, ny=10)
```

```
## Warning: Some expected counts are small; chi^2 approximation may be
## inaccurate
```

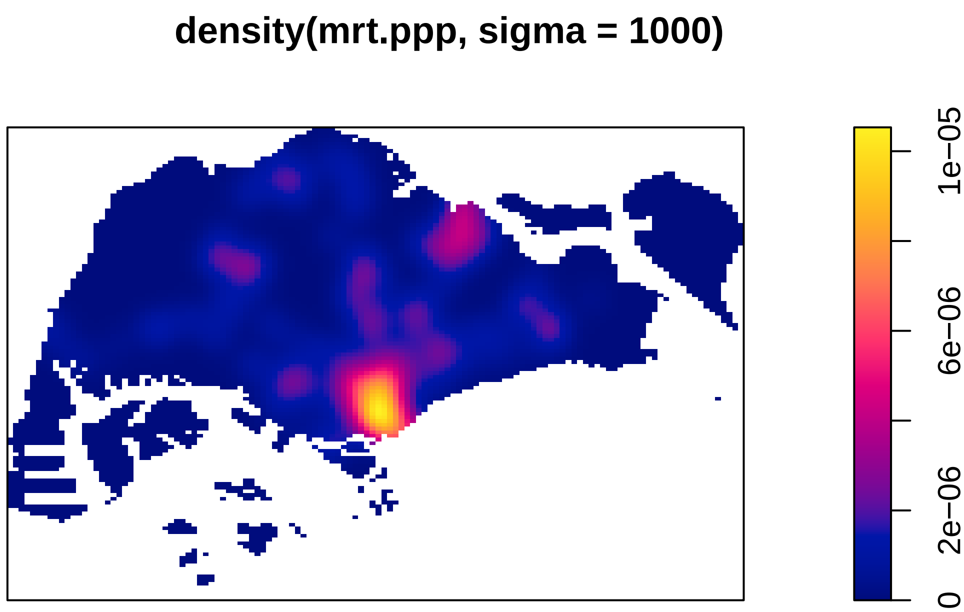
Test outcome: p-value < 2.2e-16, reject the null at 5% level.

3. Kernel density estimation

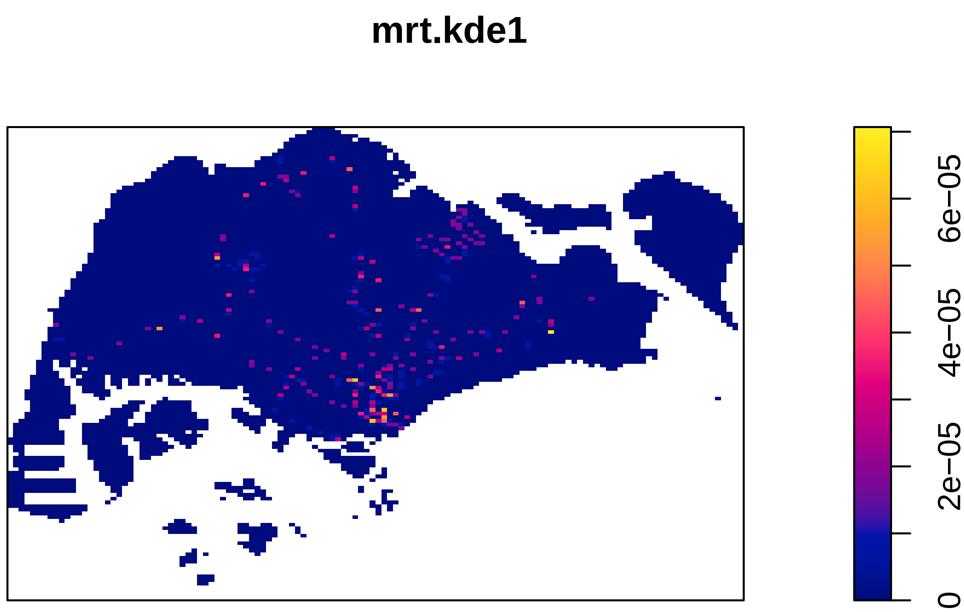
```
plot(density(mrt.ppp, sigma=500))
```




```
plot(density(mrt.ppp, sigma=1000))
```

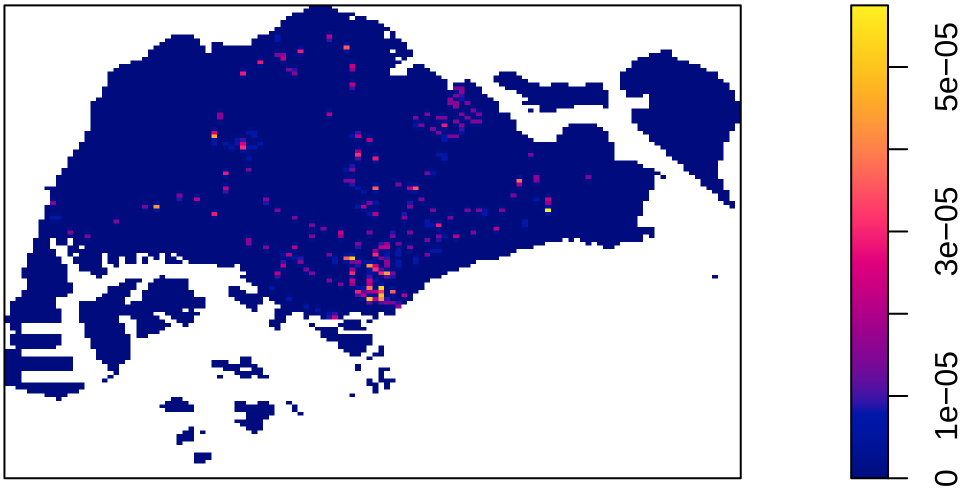



```r
mrt.bw1 = bw.diggle(mrt.ppp)
mrt.kde1 = density(mrt.ppp, sigma=mrt.bw1)
plot(mrt.kde1)
```



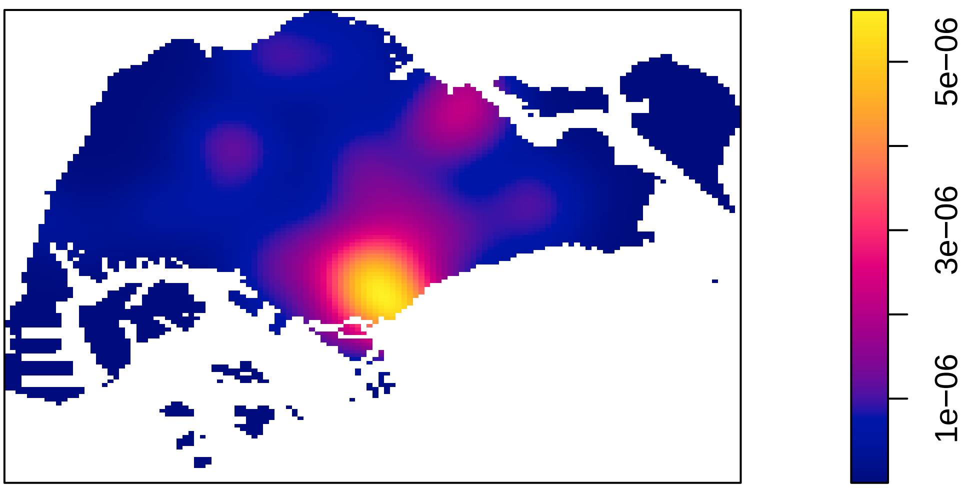

```r
mrt.bw2 = bw.ppl(mrt.ppp)
mrt.kde2 = density(mrt.ppp, sigma=mrt.bw2)
plot(mrt.kde2)
```

mrt.kde2

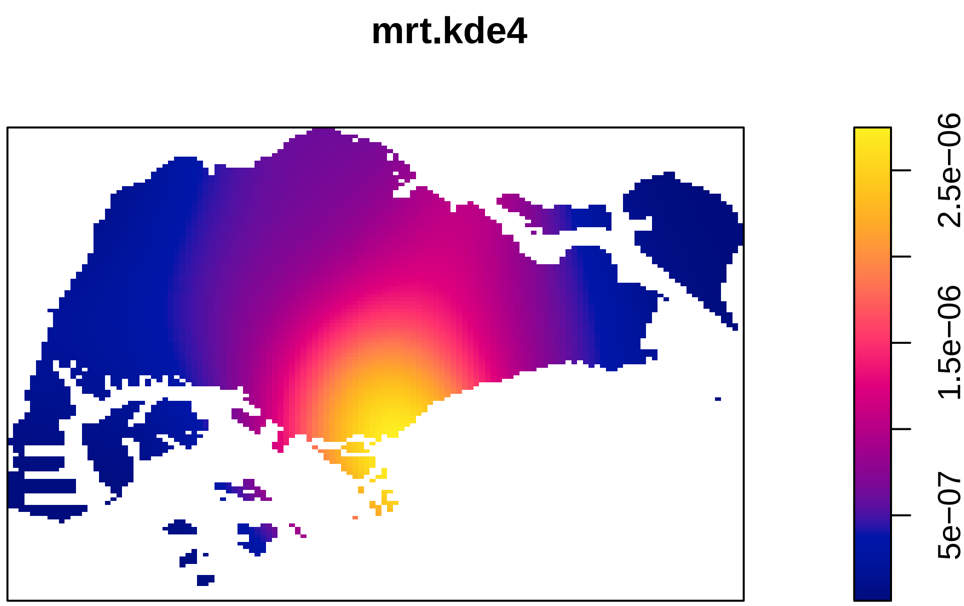



```r
mrt.bw3 = bw.scott(mrt.ppp)
mrt.kde3 = density(mrt.ppp, sigma=mrt.bw3)
plot(mrt.kde3)
```

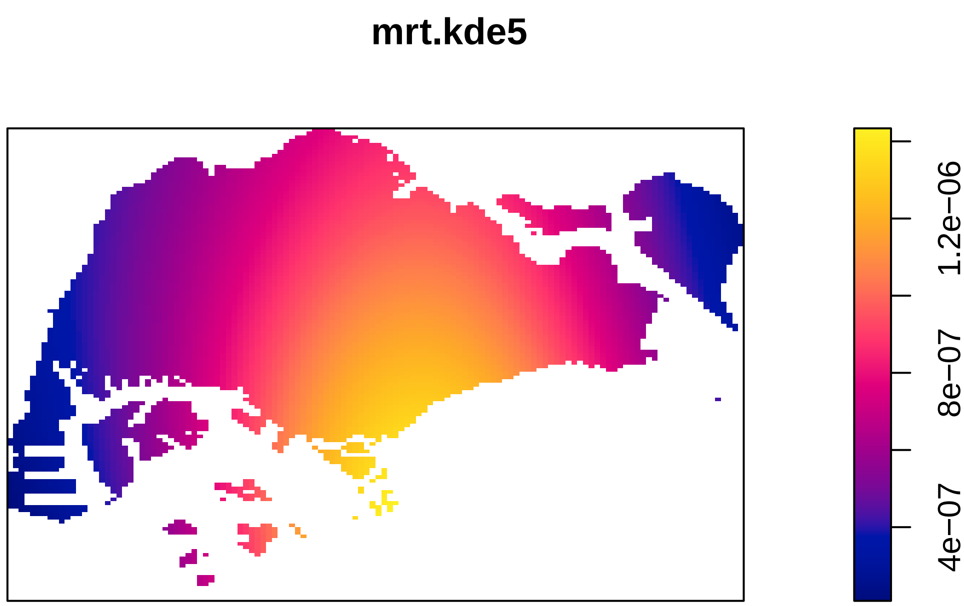
mrt.kde3




```r
mrt.bw4 = bw.CvL(mrt.ppp)
mrt.kde4 = density(mrt.ppp, sigma=mrt.bw4)
plot(mrt.kde4)
```




```
mrt.bw5 = bw.frac(mrt.ppp)
mrt.kde5 = density(mrt.ppp, sigma=mrt.bw5)
plot(mrt.kde5)
```



4. Nearest neighbour distance analysis

```
clarkevans(mrt.ppp, correction="none") #calculated index values less than 1 (r=0.1399807)
```

```
## [1] 0.1399807
```

```
clarkevans.test(mrt.ppp, correction="none", alternative="less")
```

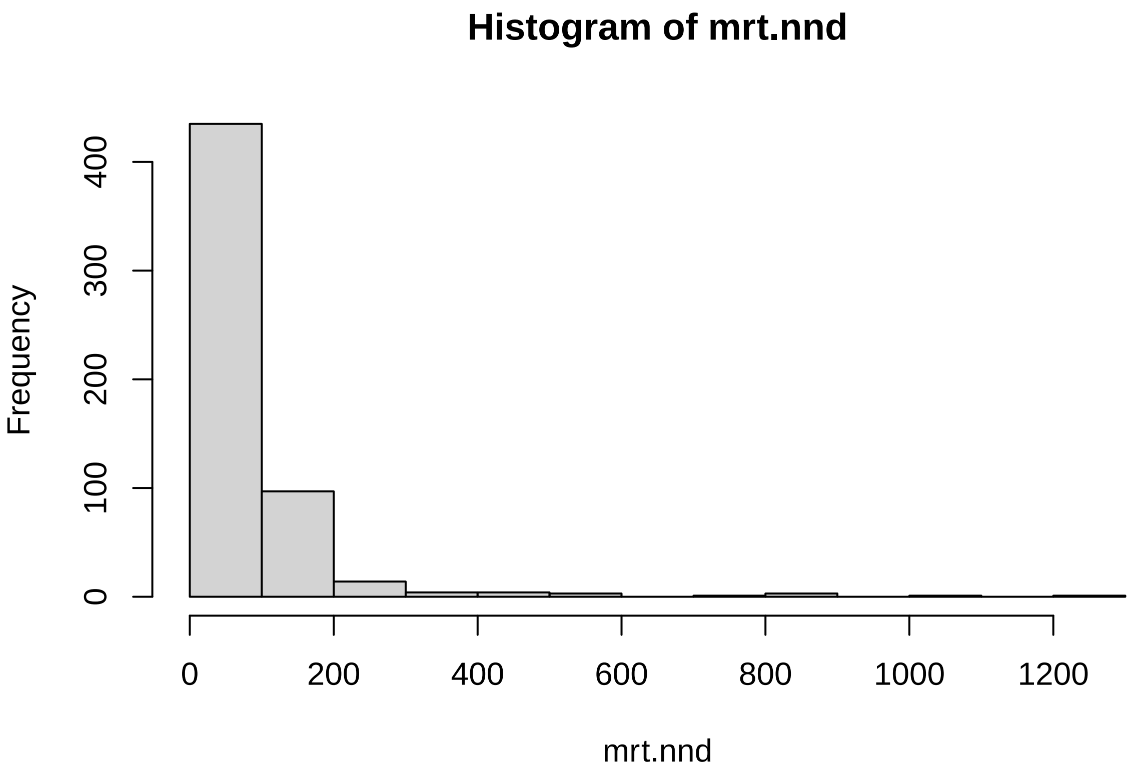
```
##
##  Clark-Evans test
##  No edge correction
##  Z-test
```

```
##
## data:  mrt.ppp
## R = 0.13998, p-value < 2.2e-16
## alternative hypothesis: clustered (R < 1)
```

```
mrt.nnd = nndist(mrt.ppp, k=1)
summary(mrt.nnd)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    3.474   32.250   51.796   82.484   91.776 1227.241
```

```
hist(mrt.nnd)
```




```
class(mrt.nnd)
```

```
## [1] "numeric"
```

```
mrt.nnd = as.matrix(mrt.nnd)
```

5. K-function

6. Real-world analysis

Patching in population density by planning areas from singstat.gov.sg. An area of consideration could be whether there are enough MRT/LRT in each planning area for residents of Singapore. This can be seen by patching a layer of population density onto the Singapore planning area map with the MRT/LRT exit points.

```
population_pa = read.csv("residentpopulation_2024.csv")
population_sf = inner_join(population_pa, sg_regions)
```

```
## Joining with `by = join_by(PLN_AREA_N)`
```

```
population_sf = st_as_sf(population_sf)
population_sf= st_transform(population_sf, 3414)
```

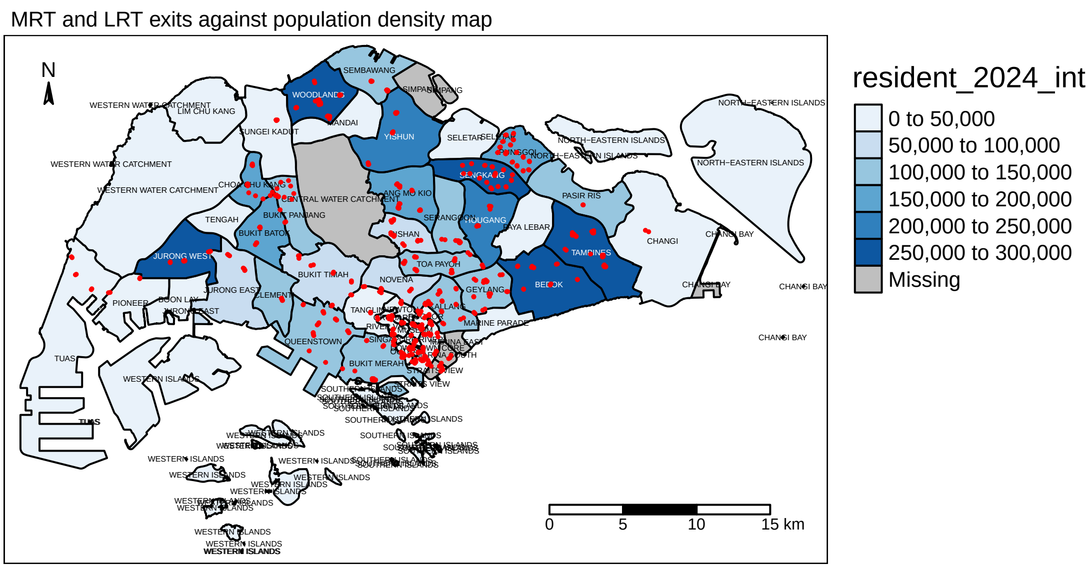
```r
isTRUE(all.equal(st_crs(sg_regions), st_crs(population_sf))) # TRUE
```

```
## [1] TRUE
```

```r
population_sf$resident_2024_int = as.numeric(gsub(",", "", population_sf$resident_2024))
```

```
## Warning: NAs introduced by coercion
```

```r
population_density =
  tm_shape(population_sf) +
  tm_borders(col="black", lwd=1, lty=1) +
  tm_fill("resident_2024_int", lty=3, palette = "Blues") +
  tm_scale_bar(position = c("right", "bottom")) +
  tm_compass(position = c("left", "top"), size =0.8, text.size = 0.7) +
  tm_layout(main.title = "Population density map", legend.outside = T) + tm_text("PLN_AREA_N", size = 0

mrt = population_density + tm_shape(mrt_lrt_exits_sf) +
  tm_dots(col = "red", size=0.01) +
  tm_layout(main.title = "MRT and LRT exits against population density map", legend.outside = T, main.t
mrt
```



From the map, we can see that there is a clustering of MRT/LRT stations at the central region with areas such as Newton, Orchard, River Valley, Downtown Core, and Outram despite the low population density.