sfnetworks

igraph + sf

網絡資料分析與模式

2021.05.21

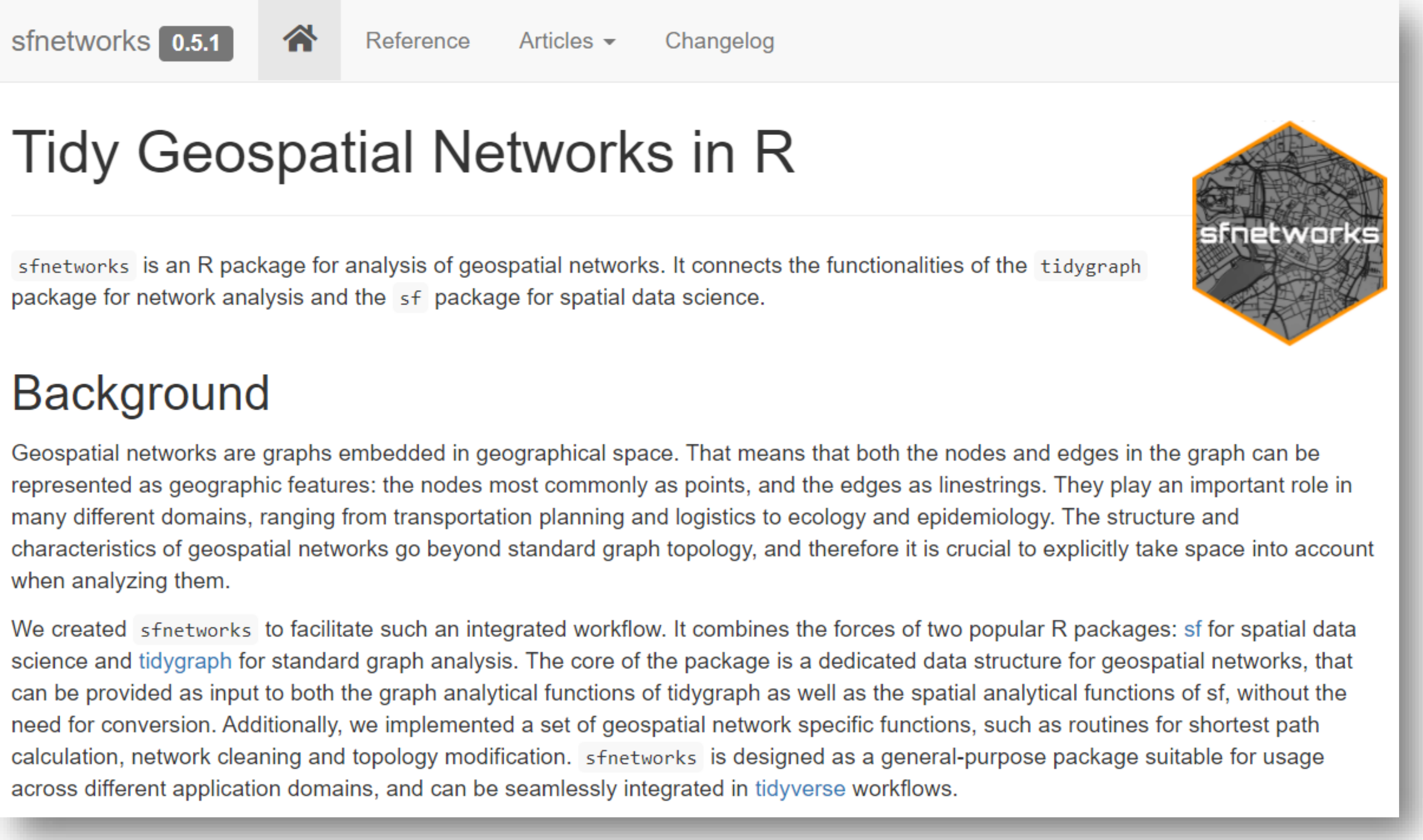
杜承軒

Contents

1. 初步了解 sf 格式，並用 tmap 套件畫地圖。
2. shp → sf → stnetwork，介紹 stnetwork 的格式。
3. 利用 stnetwork 格式，計算台北市路網的betweenness與繪製地圖。
4. 利用 igraph 的 community structure，計算路網的分群結構與繪製地圖。
5. 利用 igraph 的 distance 函數，產生 node 之間的(最短)距離矩陣。
6. 利用 igraph 的 shortest_paths 函數，找出最短路徑的 vertex & edges，並繪製地圖。

sfnetworks

整合 sf + igraph 功能



Tidy Geospatial Networks in R

`sfnetworks` is an R package for analysis of geospatial networks. It connects the functionalities of the `tidygraph` package for network analysis and the `sf` package for spatial data science.

Background

Geospatial networks are graphs embedded in geographical space. That means that both the nodes and edges in the graph can be represented as geographic features: the nodes most commonly as points, and the edges as linestrings. They play an important role in many different domains, ranging from transportation planning and logistics to ecology and epidemiology. The structure and characteristics of geospatial networks go beyond standard graph topology, and therefore it is crucial to explicitly take space into account when analyzing them.

We created `sfnetworks` to facilitate such an integrated workflow. It combines the forces of two popular R packages: sf for spatial data science and tidygraph for standard graph analysis. The core of the package is a dedicated data structure for geospatial networks, that can be provided as input to both the graph analytical functions of tidygraph as well as the spatial analytical functions of sf, without the need for conversion. Additionally, we implemented a set of geospatial network specific functions, such as routines for shortest path calculation, network cleaning and topology modification. `sfnetworks` is designed as a general-purpose package suitable for usage across different application domains, and can be seamlessly integrated in tidyverse workflows.

https://luukvdmeer.github.io/sfnetworks

認識 sf 格式 & tmap

讀取 sf 資料 → 透過 tmap 畫地圖

Loading Data

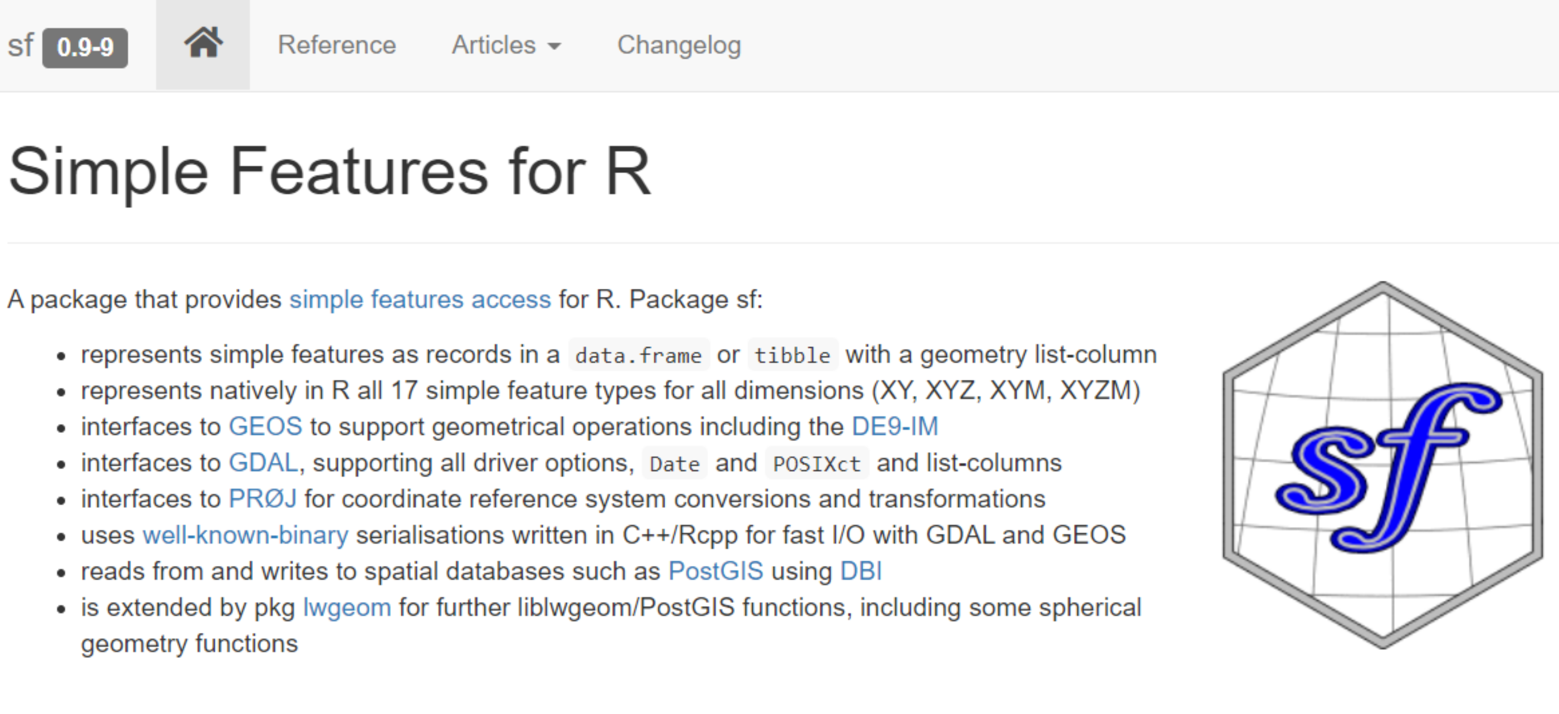
```
> library(sfnetworks)
> library(sf)
> library(tmap)
> library(igraph)
> library(tidygraph)
> library(tidyverse)
```

→ 如尚未安裝 請執行 install.packages("sfnetworks")

```
> load("TPE_Road.Rdata")
> class(road_sf)
[1] "sf"         "data.frame"
```

sf

A **simple feature** is defined by the OpenGIS Abstract specification to have both **spatial and non-spatial attributes**. Spatial attributes are geometry valued, and simple features are based on 2D geometry with linear interpolation between vertices.



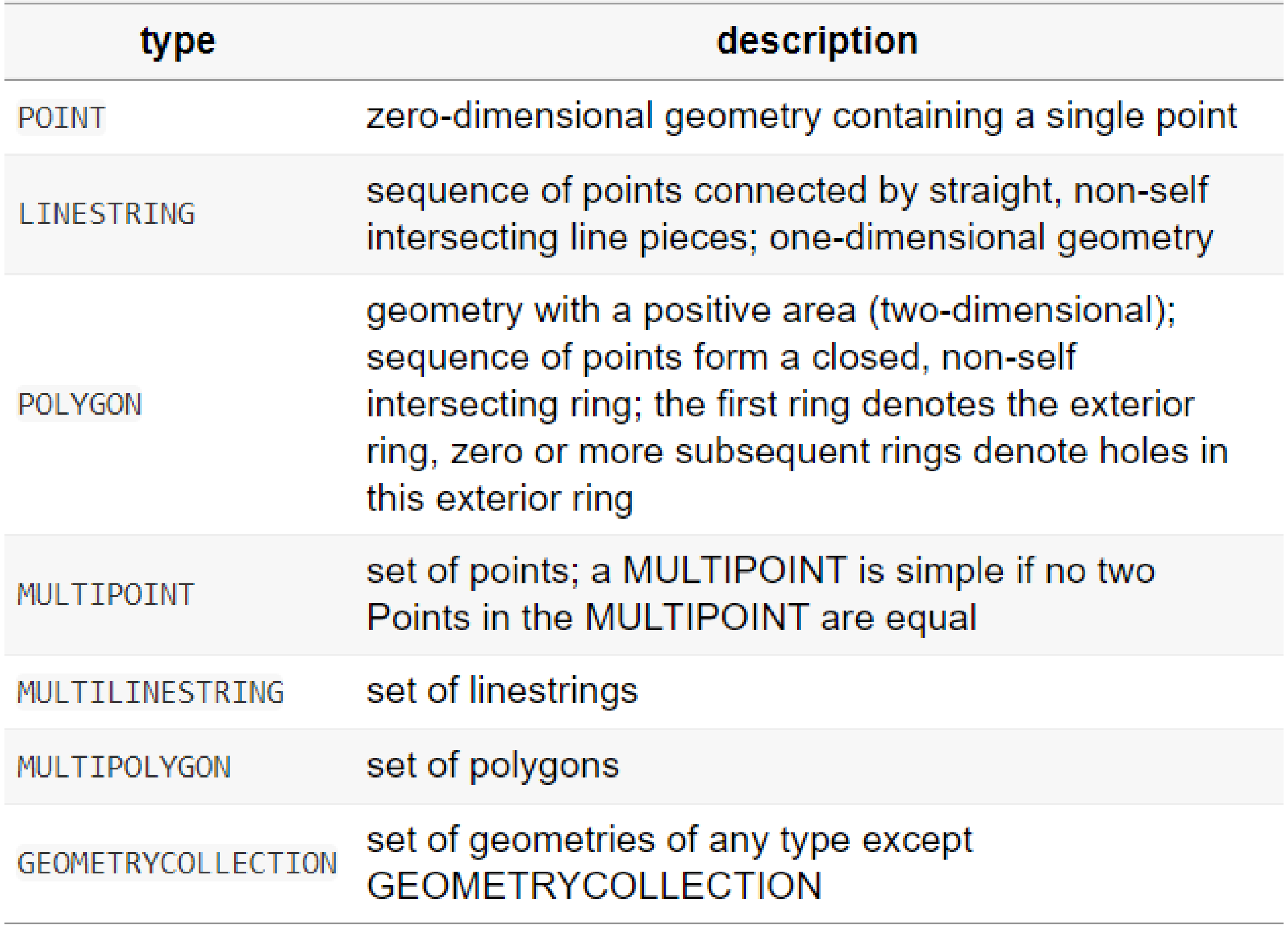
Simple Features for R

A package that provides simple features access for R. Package sf:

- represents simple features as records in a `data.frame` or `tibble` with a geometry list-column
- represents natively in R all 17 simple feature types for all dimensions (XY, XYZ, XYM, XYZM)
- interfaces to GEOS to support geometrical operations including the DE9-IM
- interfaces to GDAL, supporting all driver options, `Date` and `POSIXct` and list-columns
- interfaces to PRØJ for coordinate reference system conversions and transformations
- uses well-known-binary serialisations written in C++/Rcpp for fast I/O with GDAL and GEOS
- reads from and writes to spatial databases such as PostGIS using DBI
- is extended by pkg lwgeom for further liblwgeom/PostGIS functions, including some spherical geometry functions

https://r-spatial.github.io/sf/

sf - geometry types

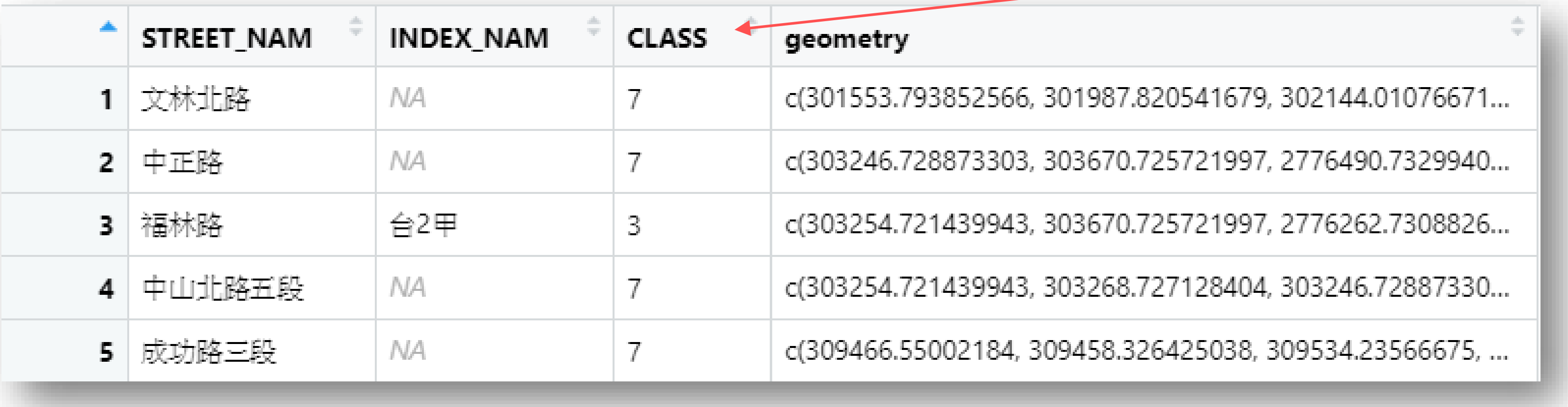


type	description
`POINT`	zero-dimensional geometry containing a single point
`LINESTRING`	sequence of points connected by straight, non-self intersecting line pieces; one-dimensional geometry
`POLYGON`	geometry with a positive area (two-dimensional); sequence of points form a closed, non-self intersecting ring; the first ring denotes the exterior ring, zero or more subsequent rings denote holes in this exterior ring
`MULTIPOINT`	set of points; a MULTIPOINT is simple if no two Points in the MULTIPOINT are equal
`MULTILINESTRING`	set of linestrings
`MULTIPOLYGON`	set of polygons
`GEOMETRYCOLLECTION`	set of geometries of any type except GEOMETRYCOLLECTION

sf object

```
> road_sf
Simple feature collection with 932 features and 3 fields
Geometry type: LINESTRING
Dimension:     XY
Bounding box:  xmin: 296597.4 ymin: 2761987 xmax: 312903.7 ymax: 2786095
Projected CRS: TWD97 / TM2 zone 121
First 10 features:
     STREET_NAM INDEX_NAM CLASS                       geometry
1        文林北路      <NA>     7 LINESTRING (301553.8 277814...
2          中正路      <NA>     7 LINESTRING (303246.7 277649...
3          福林路       台2甲     3 LINESTRING (303254.7 277626...
4    中山北路五段      <NA>     7 LINESTRING (303254.7 277626...
5      成功路三段      <NA>     7 LINESTRING (309466.6 277471...
```

road_sf$CLASS



	STREET_NAM	INDEX_NAM	CLASS	geometry
1	文林北路	NA	7	c(301553.793852566, 301987.820541679, 302144.01076671...
2	中正路	NA	7	c(303246.728873303, 303670.725721997, 2776490.7329940...
3	福林路	台2甲	3	c(303254.721439943, 303670.725721997, 2776262.7308826...
4	中山北路五段	NA	7	c(303254.721439943, 303268.727128404, 303246.72887330...
5	成功路三段	NA	7	c(309466.55002184, 309458.326425038, 309534.23566675, ...

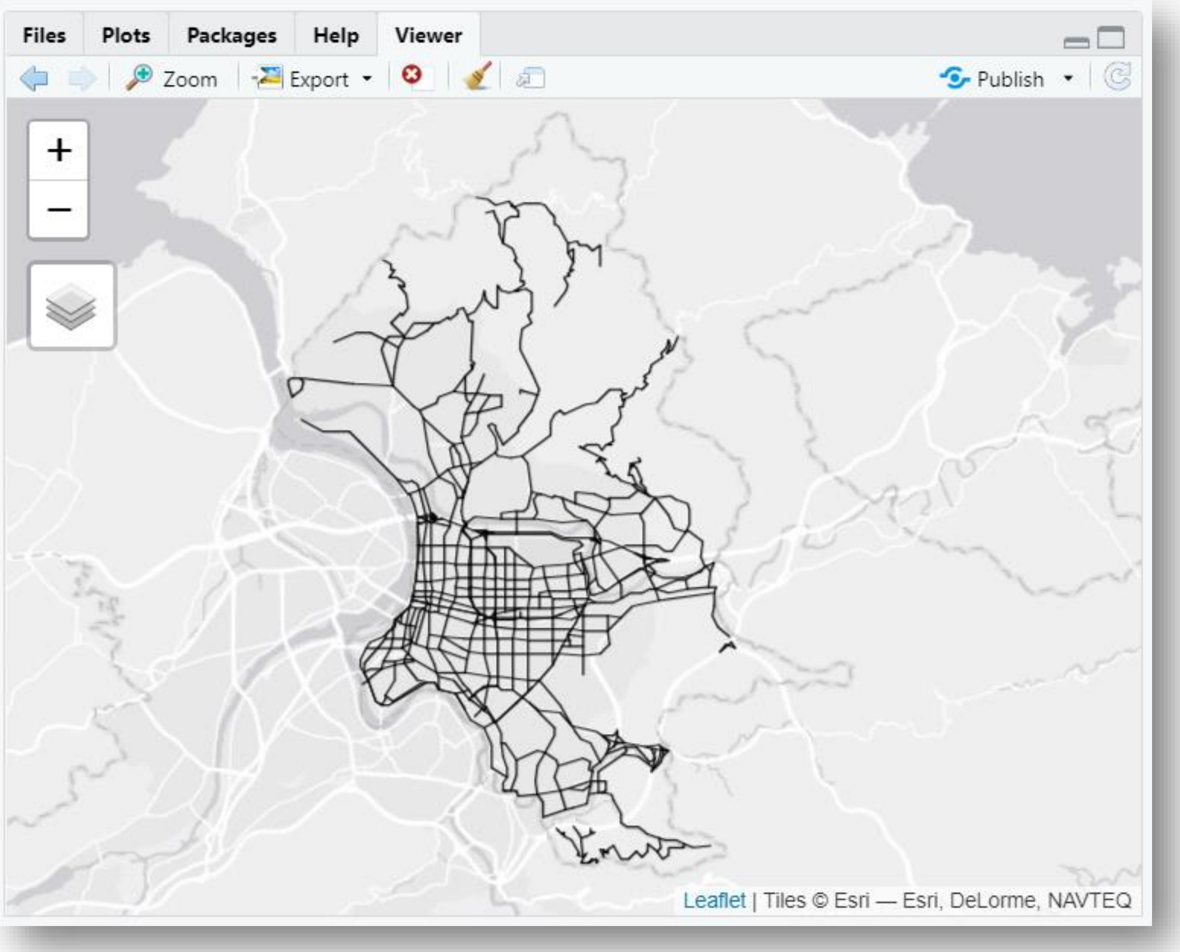
tmap - sf visualization

```
> tmap_mode("view") #interactive viewing
> map = tm_shape(road_sf) + tm_lines()
> map
```

P.S.

quick tmap

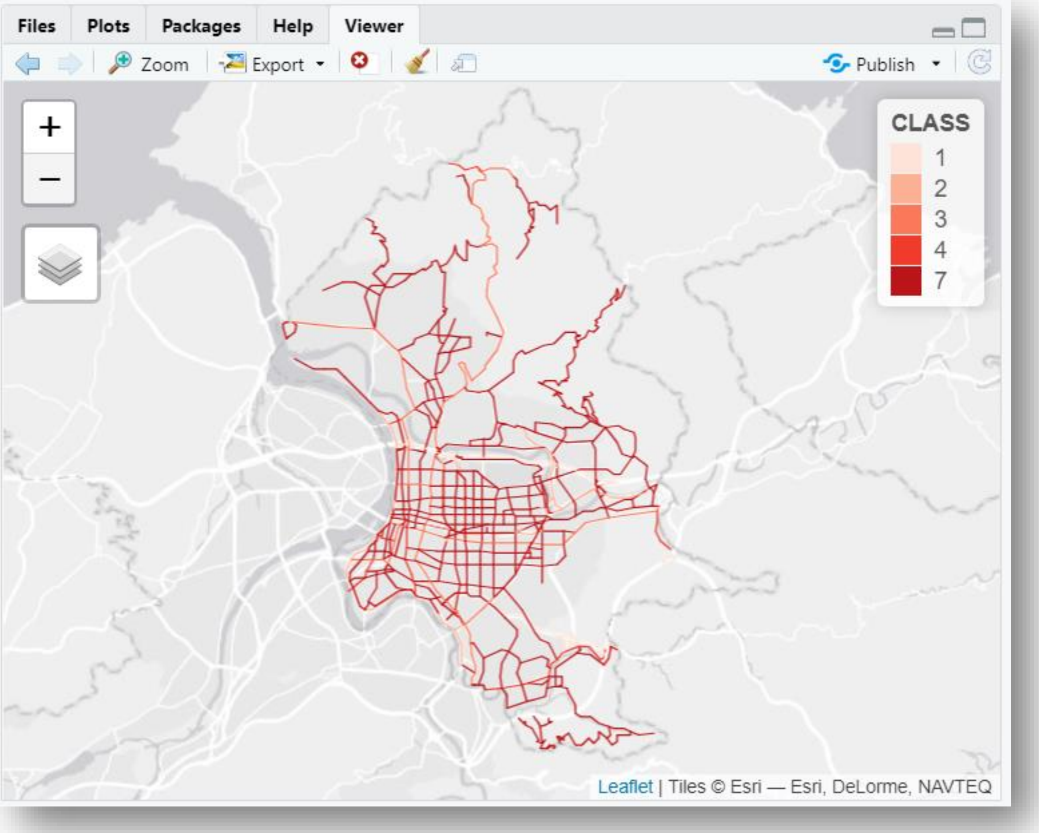
```
> qtm(road_sf)
```



tmap format

```
tm_shape (sf) +  tm_polygons ("欄位", 圖層美觀設計) #面資料
                 tm_borders   #面資料
                 tm_lines     #線資料
                 tm_dots      #點資料
                 tm_symbols   #點資料
```

```
> tm_shape(road_sf) +
    tm_lines("CLASS",palette = "Reds")
```



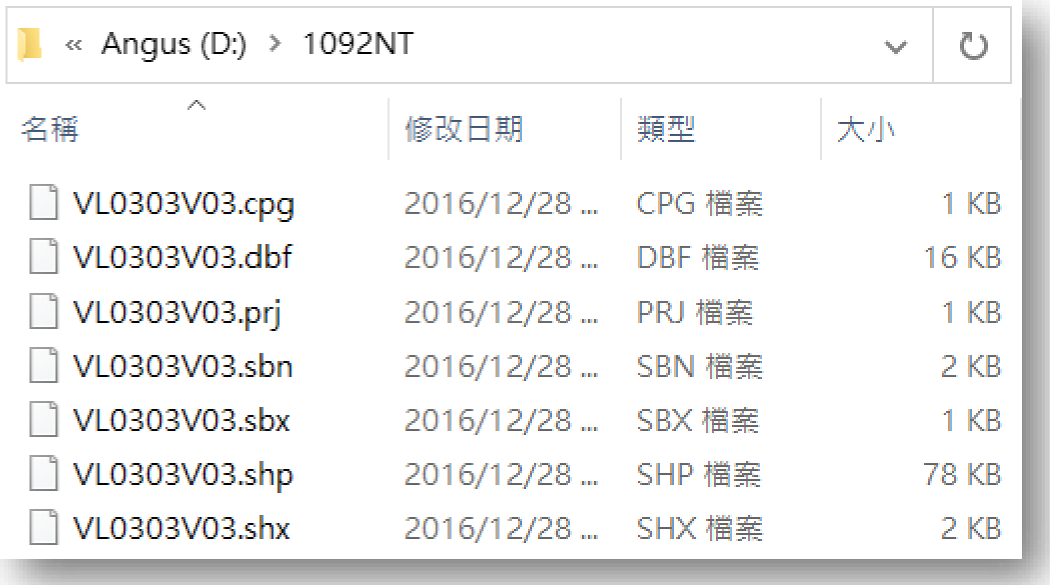
tmap: get started!
https://cran.r-project.org/web/packages/tmap/vignettes/tmap-getstarted.html

Read Shapefile

GIS-T交通網路地理資訊倉儲系統
https://gist.motc.gov.tw/gist_web/MapDataService/Retrieval

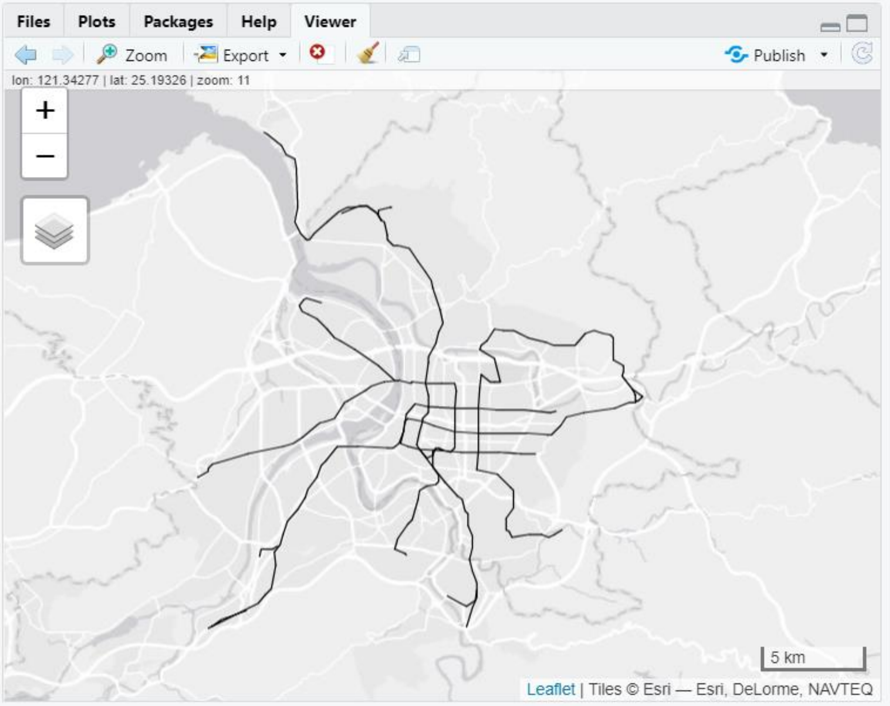
查詢「台北捷運路線」 > 下載shp > 解壓縮

```
> MRT = st_read("D:/1092NT/VL0303V03.shp")
> qtm(MRT)
```



« Angus (D:) › 1092NT

名稱	修改日期	類型	大小
VL0303V03.cpg	2016/12/28 ...	CPG 檔案	1 KB
VL0303V03.dbf	2016/12/28 ...	DBF 檔案	16 KB
VL0303V03.prj	2016/12/28 ...	PRJ 檔案	1 KB
VL0303V03.sbn	2016/12/28 ...	SBN 檔案	2 KB
VL0303V03.sbx	2016/12/28 ...	SBX 檔案	1 KB
VL0303V03.shp	2016/12/28 ...	SHP 檔案	78 KB
VL0303V03.shx	2016/12/28 ...	SHX 檔案	2 KB


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
lon: 121.34277 | lat: 25.19326 | zoom: 11
+
−
5 km
Leaflet | Tiles © Esri — Esri, DeLorme, NAVTEQ

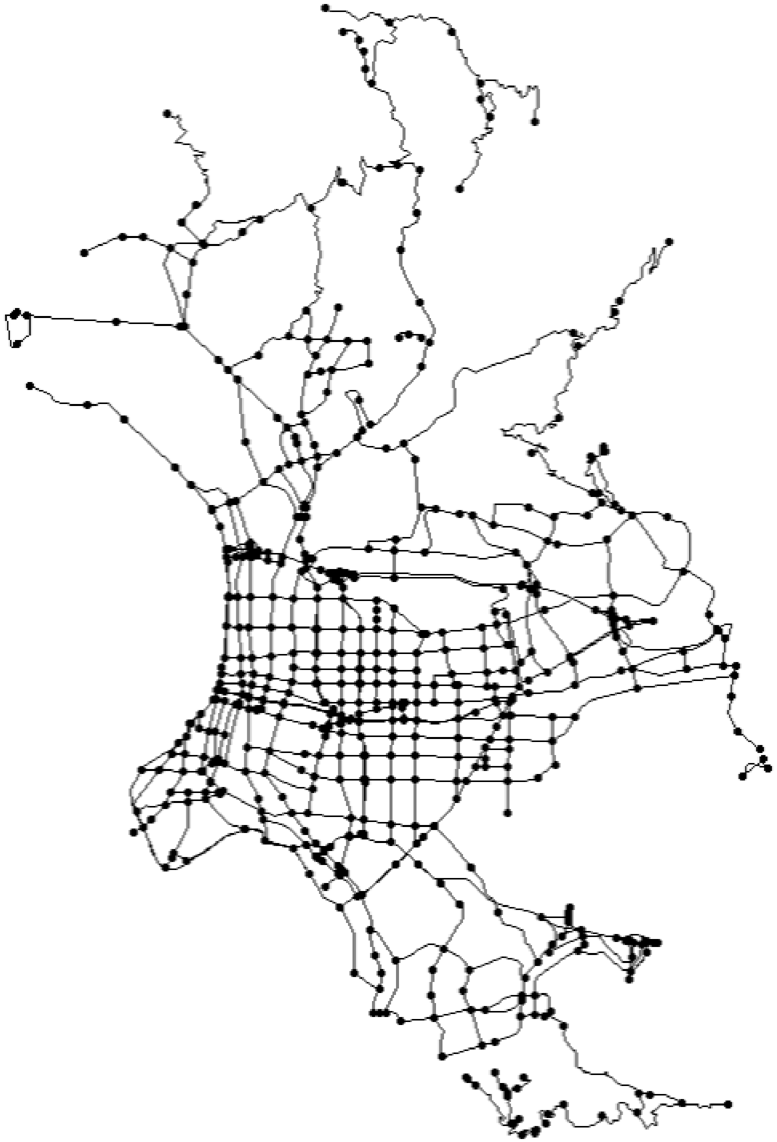
sf → sfnetwork

格式轉換 → 線圖資轉換成網絡資料

sfnetwork object

- sf → sfnetwork

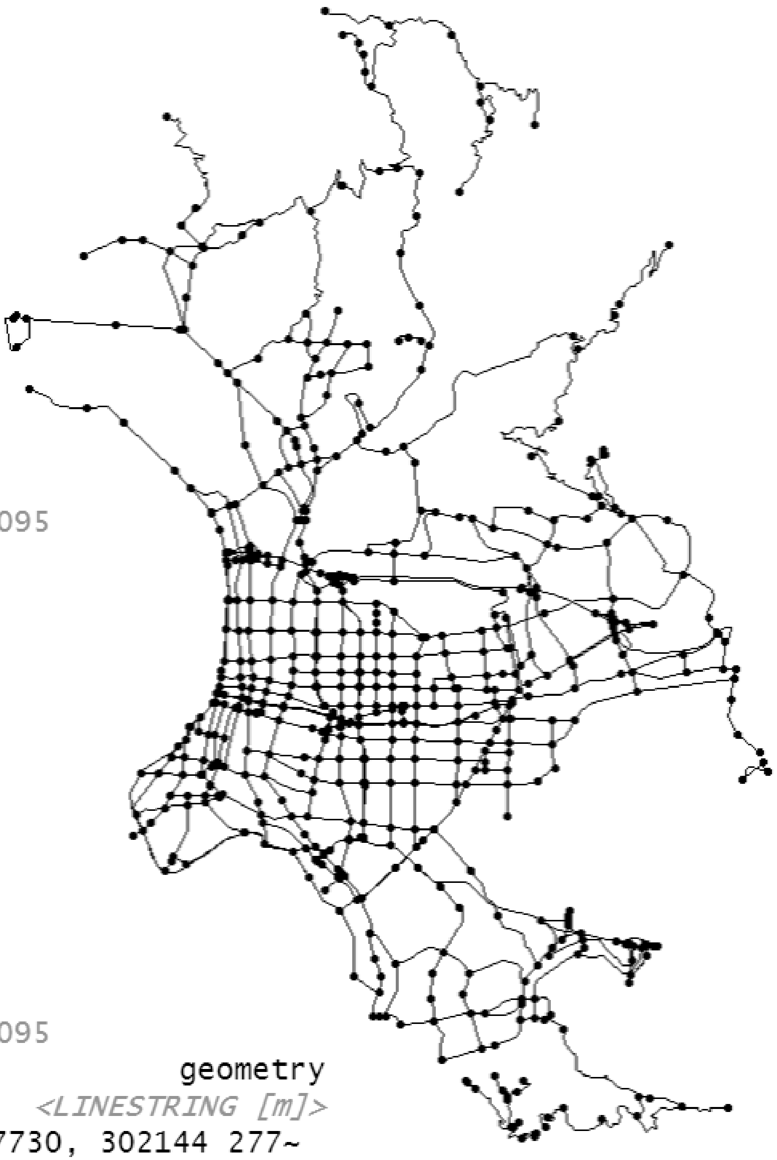
```
> road_sfnet = as_sfnetwork(road_sf,
                            directed = FALSE)

> class(road_sfnet)
[1] "sfnetwork" "tbl_graph" "igraph"

> plot(road_sfnet)
```



By default, the created network is a directed network. If you want to create an **undirected network**, set **directed = FALSE**.

sfnetwork object

```
> road_sfnet
# A sfnetwork with 618 nodes and 932 edges
#
# CRS:  EPSG:3826
#
# A directed multigraph with 3 components with spatially explicit edges
#
# Node Data:     618 x 1 (active)
# Geometry type: POINT
# Dimension:     XY
# Bounding box:  xmin: 296761.5 ymin: 2761987 xmax: 312903.7 ymax: 2786095
            geometry
         <POINT [m]>
1 (301553.8 2778143)
2   (302412 2777319)
3 (303246.7 2776491)
4 (303670.7 2776568)
5 (303254.7 2776263)
6 (309466.6 2774716)
# ... with 612 more rows
#
# Edge Data:     932 x 6
# Geometry type: LINESTRING
# Dimension:     XY
# Bounding box:  xmin: 296597.4 ymin: 2761987 xmax: 312903.7 ymax: 2786095
   from    to STREET_NAM INDEX_NAM CLASS                                       geometry
  <int> <int> <chr>      <chr>     <chr>                              <LINESTRING [m]>
1     1     2 文林北路    NA            7     (301553.8 2778143, 301987.8 2777730, 302144 277~
2     3     4 中正路      NA            7                  (303246.7 2776491, 303670.7 2776568)
3     5     4 福林路      台2甲         3                  (303254.7 2776263, 303670.7 2776568)
# ... with 929 more rows
```



計算路網betweenness

計算台北市路網 node/edge betweenness → 繪製地圖

Node's Betweenness

針對nodes這個屬性

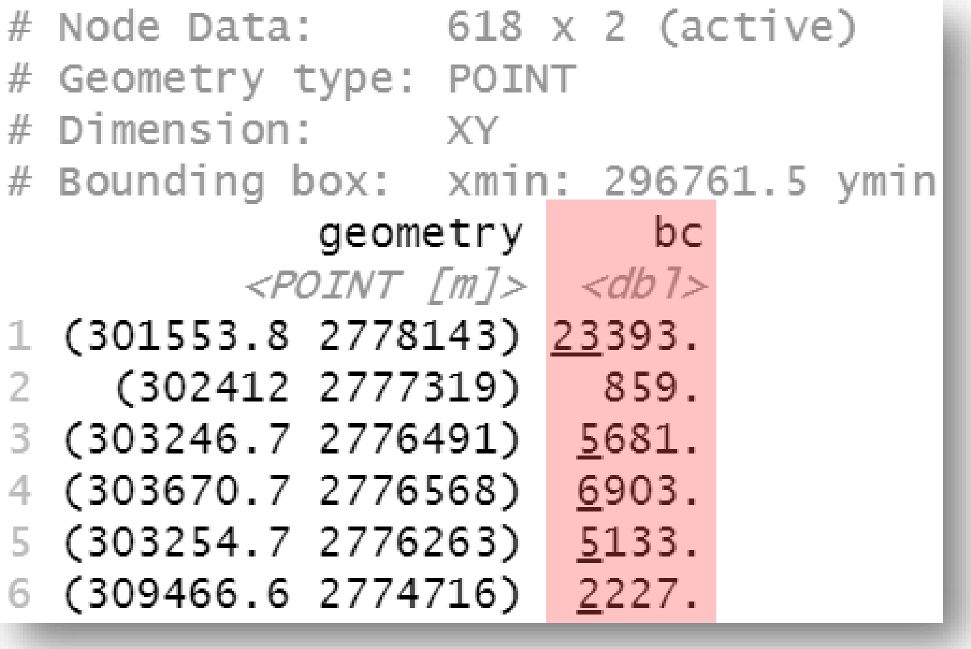
```
> road_sfnet = road_sfnet %>% activate("nodes") %>%
                                   mutate(  bc  = centrality_betweenness())
```

新增一個欄位

欄位名稱為bc

透過centrality_betweenness函數來計算

- 在 Node Data 新增一欄 bc
  數值為節點betweenness中心性



```
# Node Data:     618 x 2 (active)
# Geometry type: POINT
# Dimension:     XY
# Bounding box:  xmin: 296761.5 ymin
             geometry     bc
          <POINT [m]>  <dbl>
1 (301553.8 2778143) 23393.
2   (302412 2777319)   859.
3 (303246.7 2776491)  5681.
4 (303670.7 2776568)  6903.
5 (303254.7 2776263)  5133.
6 (309466.6 2774716)  2227.
```

Node's Betweenness

轉換成 sf 格式 → 透過 tmap 繪圖

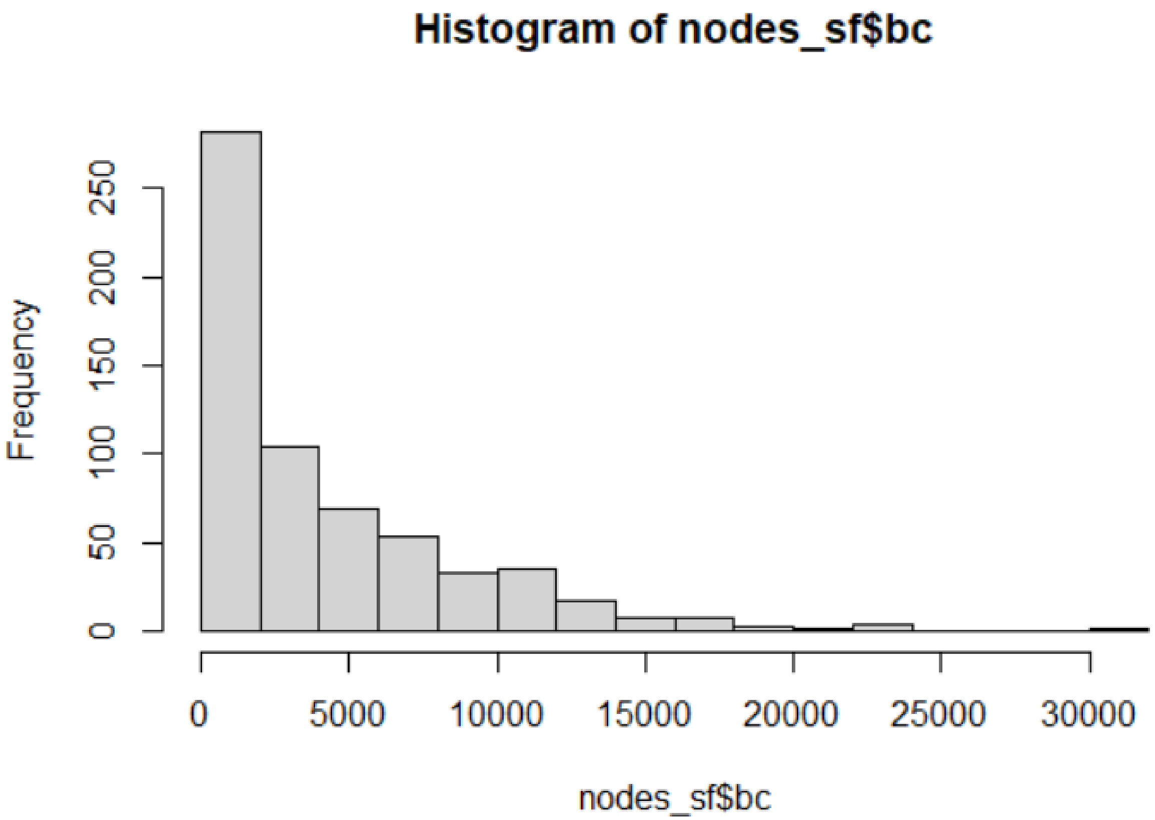
```
> nodes_sf = st_as_sf(road_sfnet, "nodes")
```

觀察數值分布

```
> hist(nodes_sf$bc)
```

將bc數值除以10000

```
> nodes_sf$bc = nodes_sf$bc/10000
```

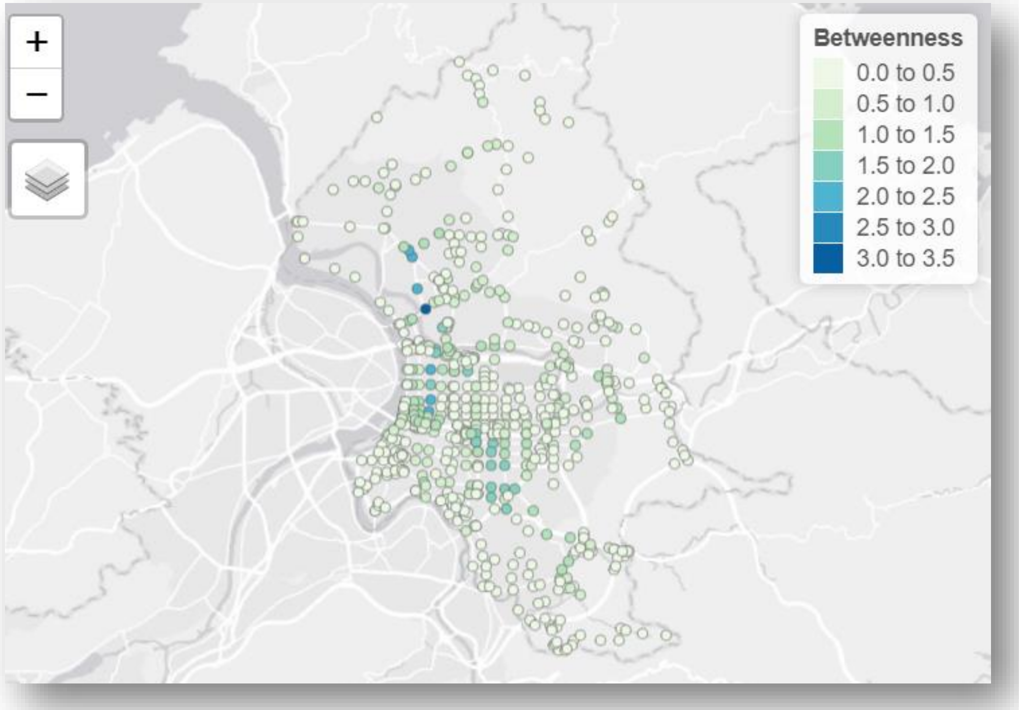


Node's Betweenness

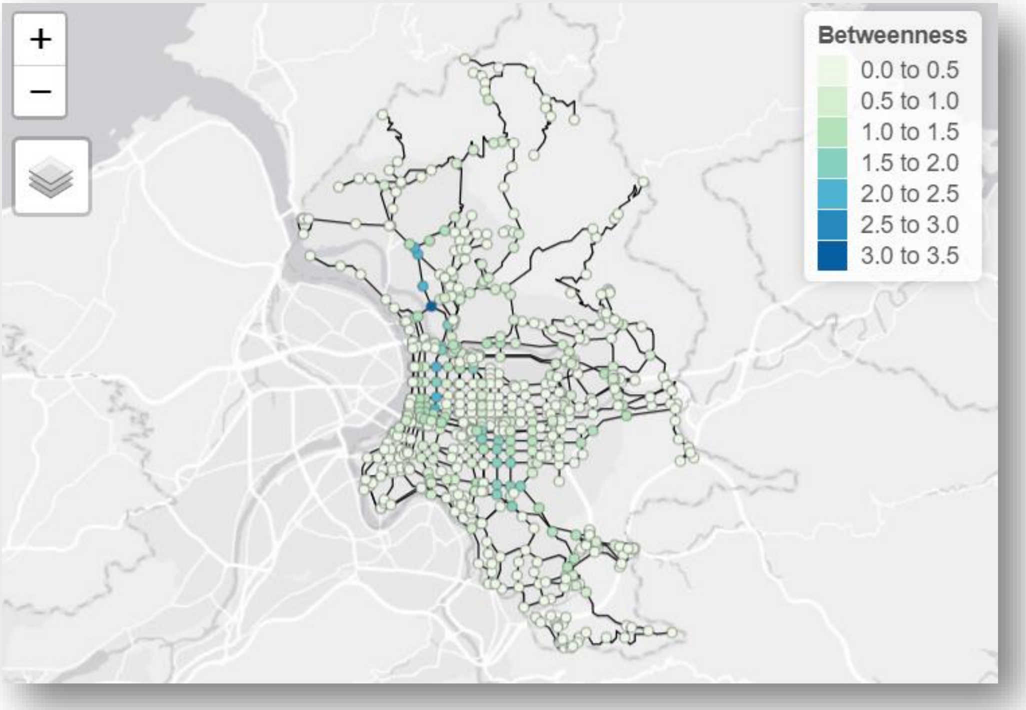
```
> nodes_lyr = tm_shape(nodes_sf) +
                     tm_dots("bc", palette = "GnBu",  title = "Betweenness")
```

針對bc欄位著色　色彩選擇　圖例標題

```
> nodes_lyr
```



```
> map + nodes_lyr #疊圖
```



Edge's Betweenness

針對edges這個屬性

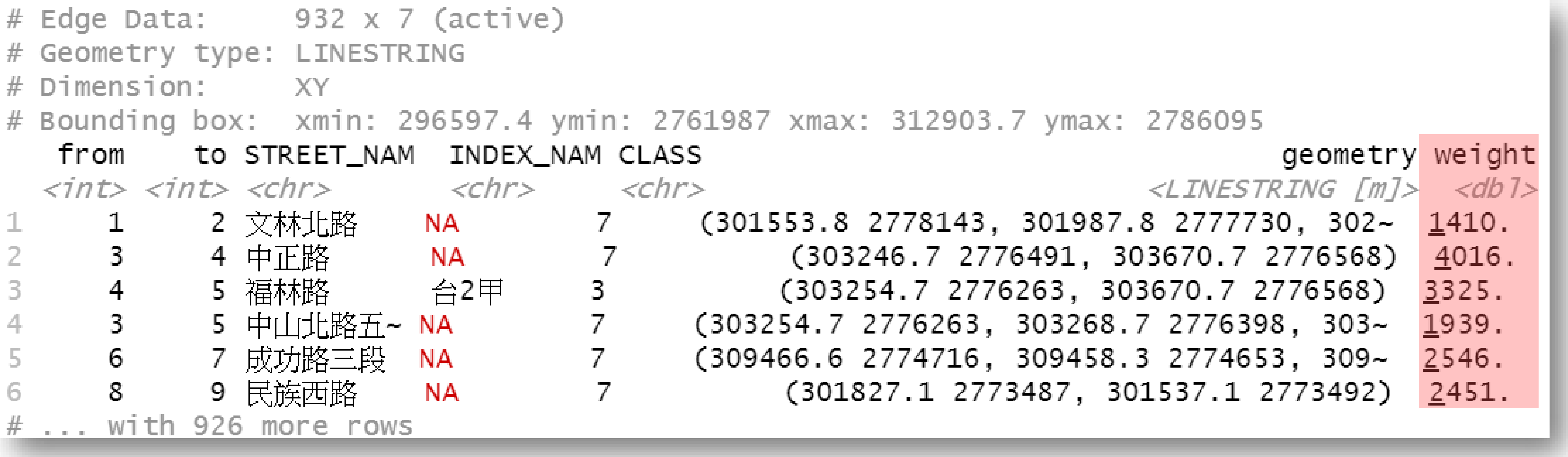
```
> road_sfnet = road_sfnet %>% activate("edges") %>%
                               mutate(weight= centrality_edge_betweenness())
```

新增欄位

欄位名稱為weight

透過centrality_edge_betweenness函數來計算

- 在 Edge Data 新增一欄 weight
  數值為edge betweenness中心性



```
# Edge Data:     932 x 7 (active)
# Geometry type: LINESTRING
# Dimension:     XY
# Bounding box:  xmin: 296597.4 ymin: 2761987 xmax: 312903.7 ymax: 2786095
   from    to STREET_NAM   INDEX_NAM CLASS                                   geometry weight
  <int> <int> <chr>        <chr>     <chr>                           <LINESTRING [m]>  <dbl>
1     1     2 文林北路     NA        7      (301553.8 2778143, 301987.8 2777730, 302~  1410.
2     3     4 中正路       NA        7              (303246.7 2776491, 303670.7 2776568)  4016.
3     4     5 福林路       台2甲     3              (303254.7 2776263, 303670.7 2776568)  3325.
4     3     5 中山北路五~  NA        7      (303254.7 2776263, 303268.7 2776398, 303~  1939.
5     6     7 成功路三段   NA        7      (309466.6 2774716, 309458.3 2774653, 309~  2546.
6     8     9 民族西路     NA        7              (301827.1 2773487, 301537.1 2773492)  2451.
# ... with 926 more rows
```

Edge's Betweenness

轉換成 sf 格式 → 透過 tmap 繪圖

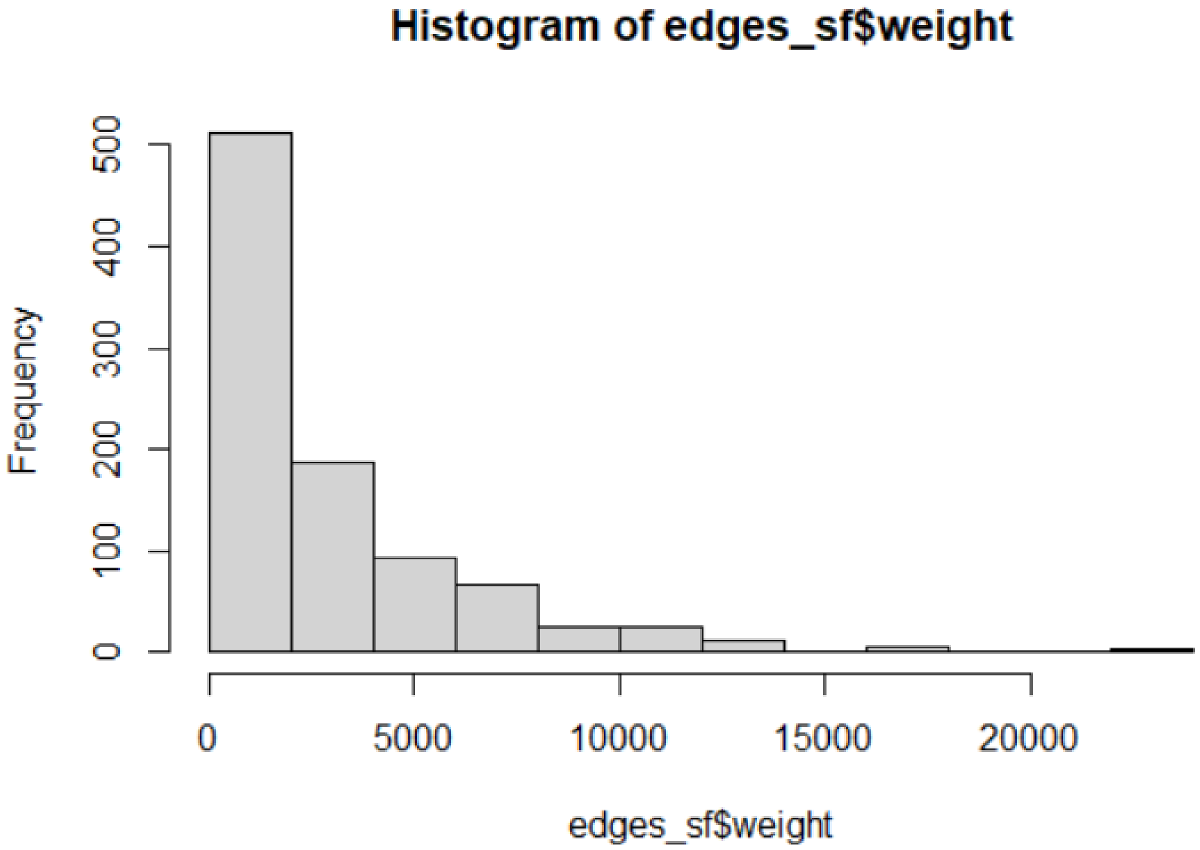
```
> edges_sf = st_as_sf(road_sfnet, "edges")
```

觀察數值分布

```
> hist(edges_sf$weight)
```

將bc數值除以10000

```
> edges_sf$weight =
     edges_sf$weight/10000
```

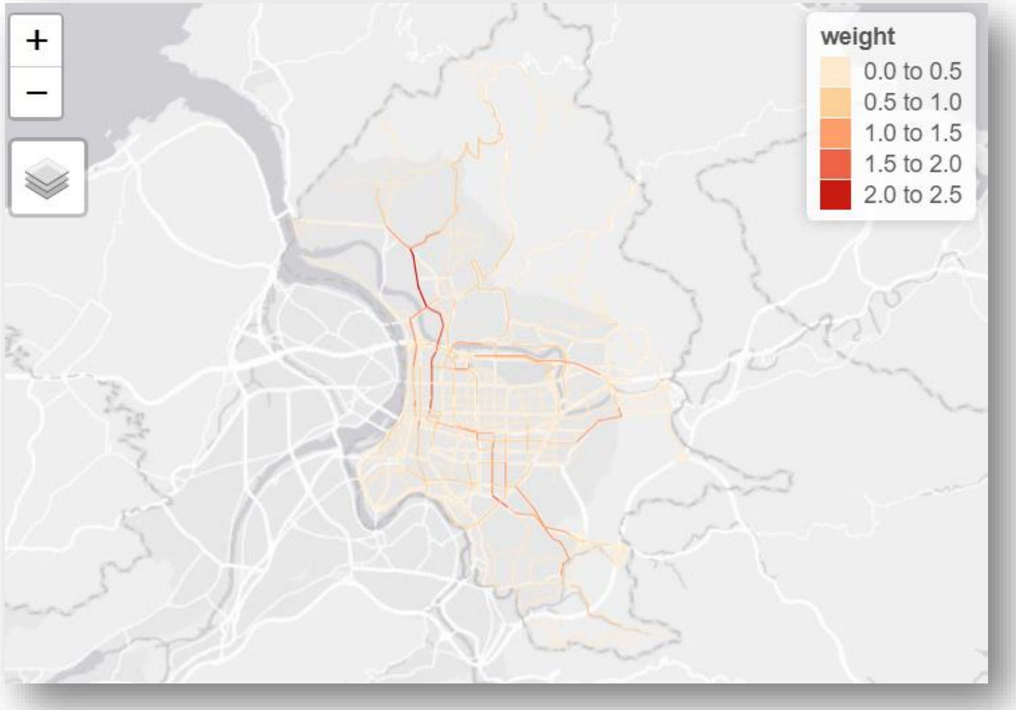


Edge's Betweenness

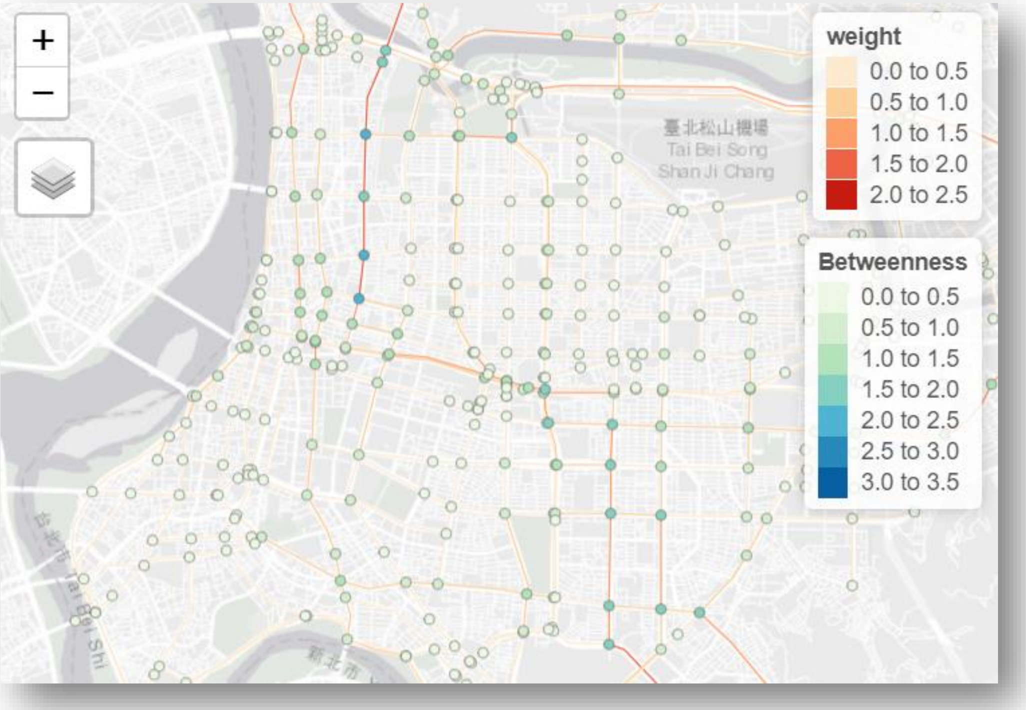
```
> edges_lyr = tm_shape(edges_sf) +
                 tm_lines("weight", palette = "OrRd")
```

針對weight欄位著色

```
> edges_lyr
```



```
> edges_lyr + nodes_lyr
```

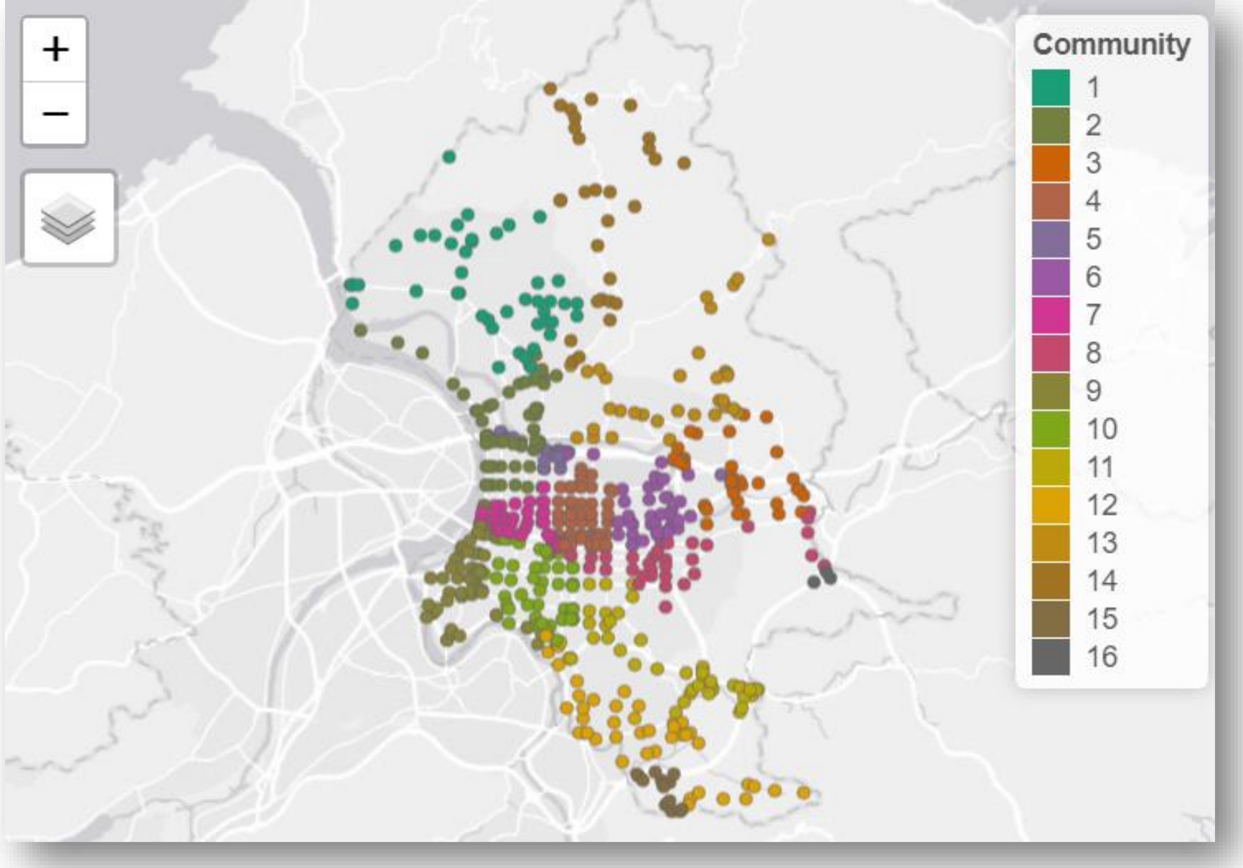


計算路網分群結構

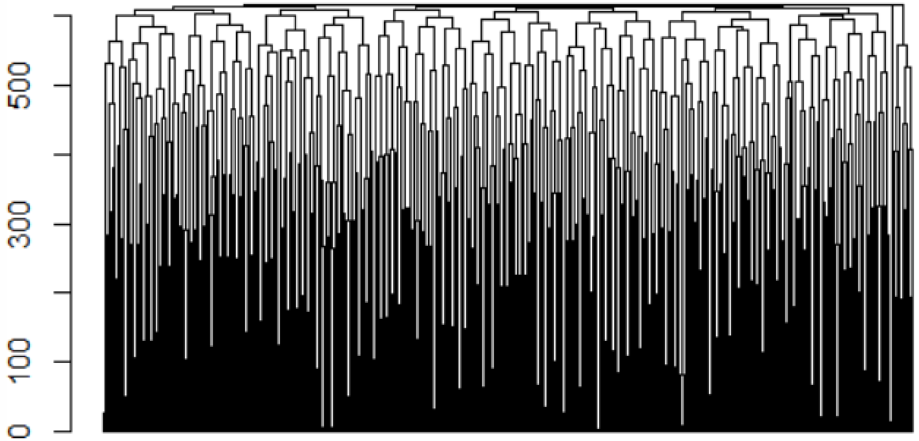
計算台北市路網 community structure → 繪製地圖

Community

```
> comm_eb = road_sfnet %>% activate("edges") %>%
                                   cluster_edge_betweenness
> nodes_sf$member = as.factor(comm_eb$membership)  #GN method
> tm_shape(nodes_sf)+tm_dots("member",palette="Dark2",title="Community")
```



```
> comm_dend = as.dendrogram(comm_eb)
> plot(comm_dend)
```



計算距離矩陣

計算兩兩節點的距離矩陣

Distance Matrix

- 計算每個連結的長度

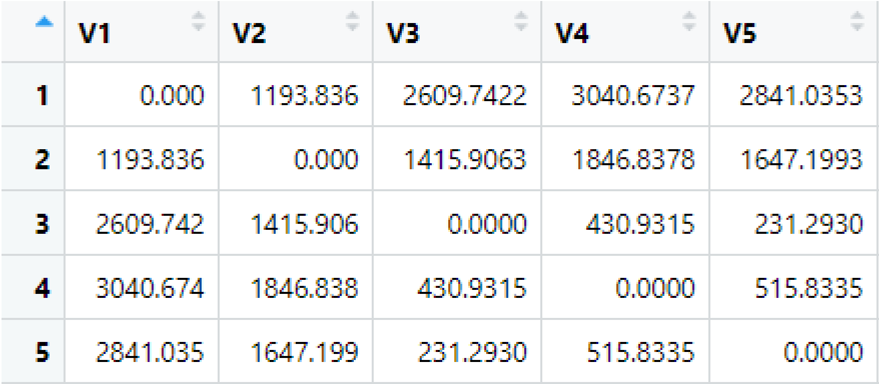
```
> road_sfnet = road_sfnet %>% activate("edges") %>%
                                 mutate(length = edge_length())
```

- 計算距離矩陣

```
> dist_matrix =  distances(road_sfnet,
               weights = road_sfnet %>% activate(edges) %>% pull(length))
```

把「length欄位」抽取出來，根據「length」加權

- dist_matrix (618x618)



	V1	V2	V3	V4	V5
1	0.000	1193.836	2609.7422	3040.6737	2841.0353
2	1193.836	0.000	1415.9063	1846.8378	1647.1993
3	2609.742	1415.906	0.0000	430.9315	231.2930
4	3040.674	1846.838	430.9315	0.0000	515.8335
5	2841.035	1647.199	231.2930	515.8335	0.0000

最短路徑分析

shortest_paths 函數 → 找出最短路徑 vertex & edges → 繪製地圖

Index / From / To

※目標：從Node200走到Node500的最短距離

- index 新增nodeID

```
> road_sfnet = road_sfnet %>% activate("nodes") %>%
                         mutate(nodeID = c(1:618))
```

- from

```
> from_node = road_sfnet %>%  activate(nodes) %>%
                         filter(nodeID == 200) %>%  pull(nodeID)
```

篩選出nodeID為200的節點　　　　把「nodeID欄位」抽取出來

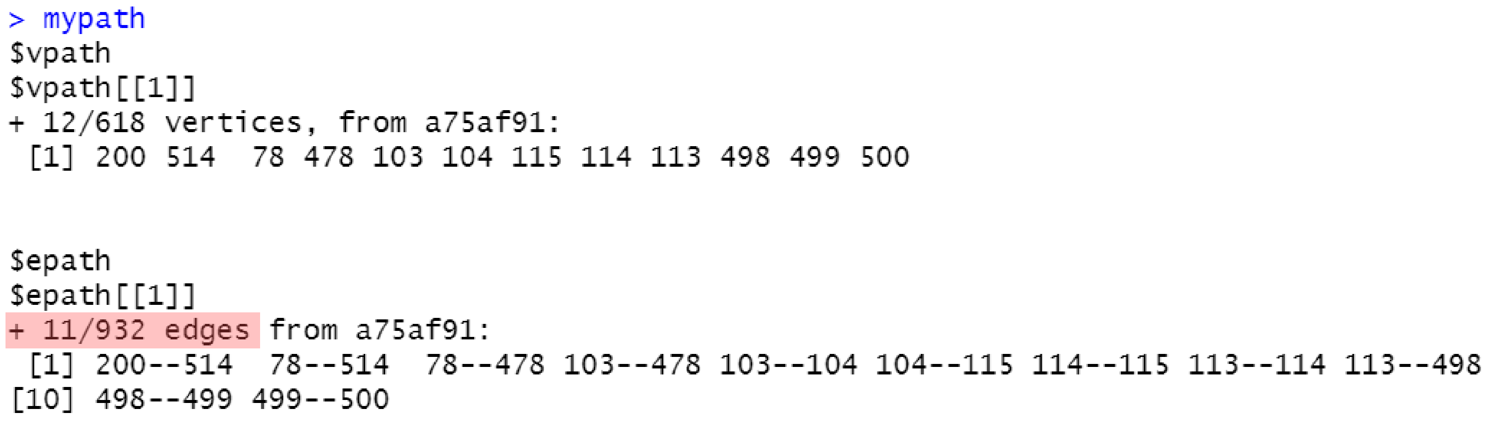
```
# from_node = 200
```

- to

```
> to_node = road_sfnet %>%  activate(nodes) %>%
                         filter(nodeID == 500) %>%  pull(nodeID)
# to_node = 500
```

Shortest Path

```
> mypath = shortest_paths(
    graph = road_sfnet,
    from = from_node, #200
    to = to_node, #500
    output = 'both',
    weights = road_sfnet %>% activate(edges) %>% pull(length)
  )
```



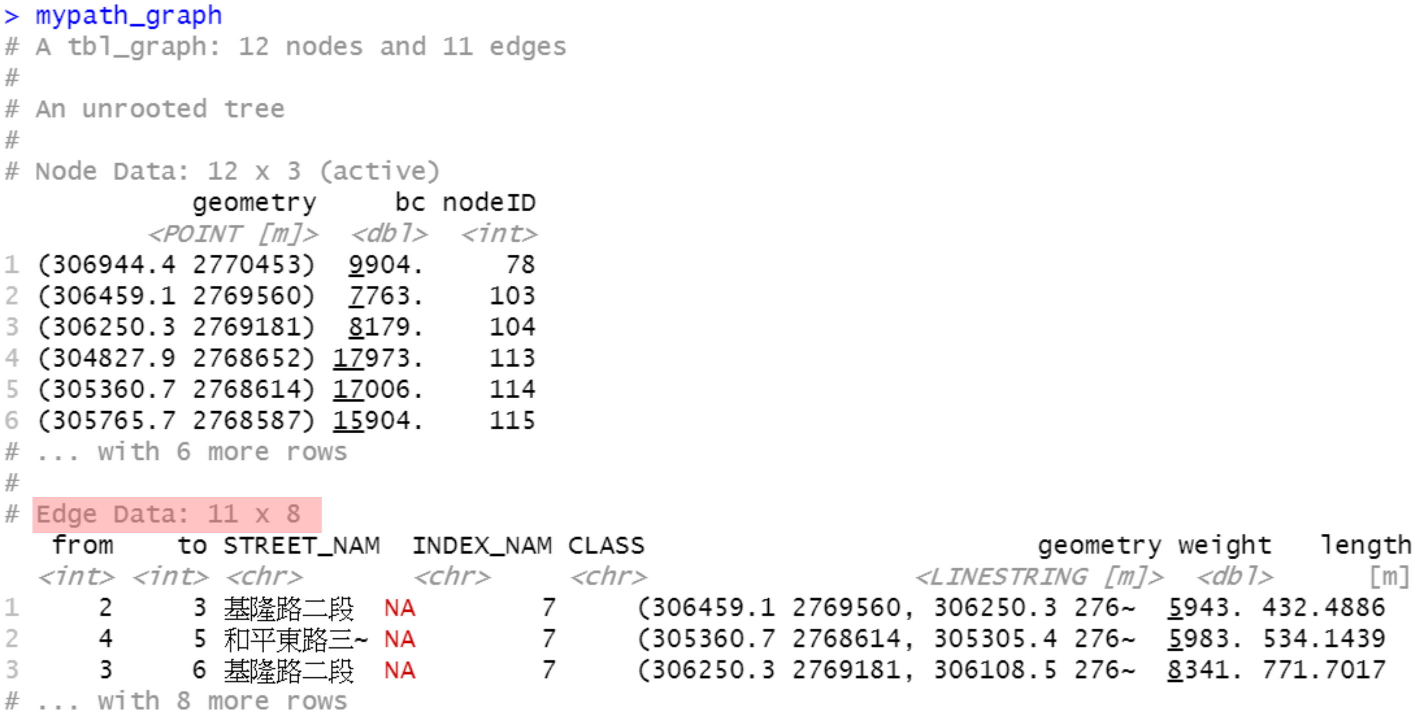
```
> mypath
$vpath
$vpath[[1]]
+ 12/618 vertices, from a75af91:
 [1] 200 514  78 478 103 104 115 114 113 498 499 500


$epath
$epath[[1]]
+ 11/932 edges from a75af91:
 [1] 200--514  78--514  78--478 103--478 103--104 104--115 114--115 113--114 113--498
[10] 498--499 499--500
```

200 → 514 → 78 → 478 → 103 → 104 → 115 → 114 → 113 → 498 → 499 → 500

Shortest Path

- 繪製最短距離

```
> mypath_graph = road_sfnet %>%
        subgraph.edges(eids = mypath$epath %>% unlist()) %>%
        as_tbl_graph()
> class(mypath_graph)
[1] "tbl_graph" "igraph"
```

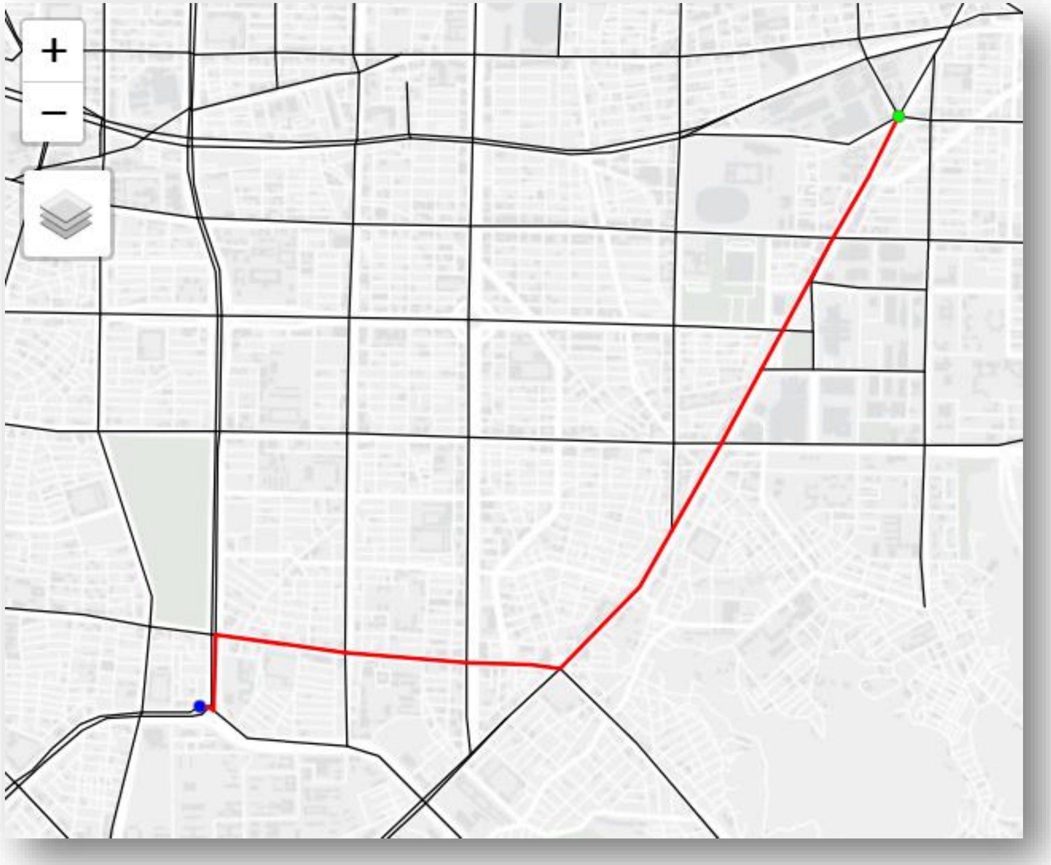


```
> mypath_graph
# A tbl_graph: 12 nodes and 11 edges
#
# An unrooted tree
#
# Node Data: 12 x 3 (active)
             geometry     bc nodeID
          <POINT [m]>  <dbl>  <int>
1 (306944.4 2770453)  9904.     78
2 (306459.1 2769560)  7763.    103
3 (306250.3 2769181)  8179.    104
4 (304827.9 2768652) 17973.    113
5 (305360.7 2768614) 17006.    114
6 (305765.7 2768587) 15904.    115
# ... with 6 more rows
#
# Edge Data: 11 x 8
   from    to STREET_NAM   INDEX_NAM CLASS                          geometry weight   length
  <int> <int> <chr>        <chr>     <chr>                <LINESTRING [m]>  <dbl>      [m]
1     2     3 基隆路二段   NA        7     (306459.1 2769560, 306250.3 276~  5943. 432.4886
2     4     5 和平東路三~  NA        7     (305360.7 2768614, 305305.4 276~  5983. 534.1439
3     3     6 基隆路二段   NA        7     (306250.3 2769181, 306108.5 276~  8341. 771.7017
# ... with 8 more rows
```

Shortest Path

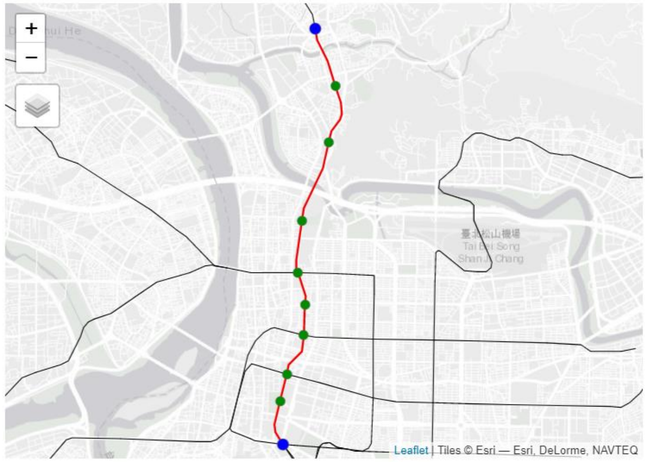
- 轉換成sf格式 → 繪圖

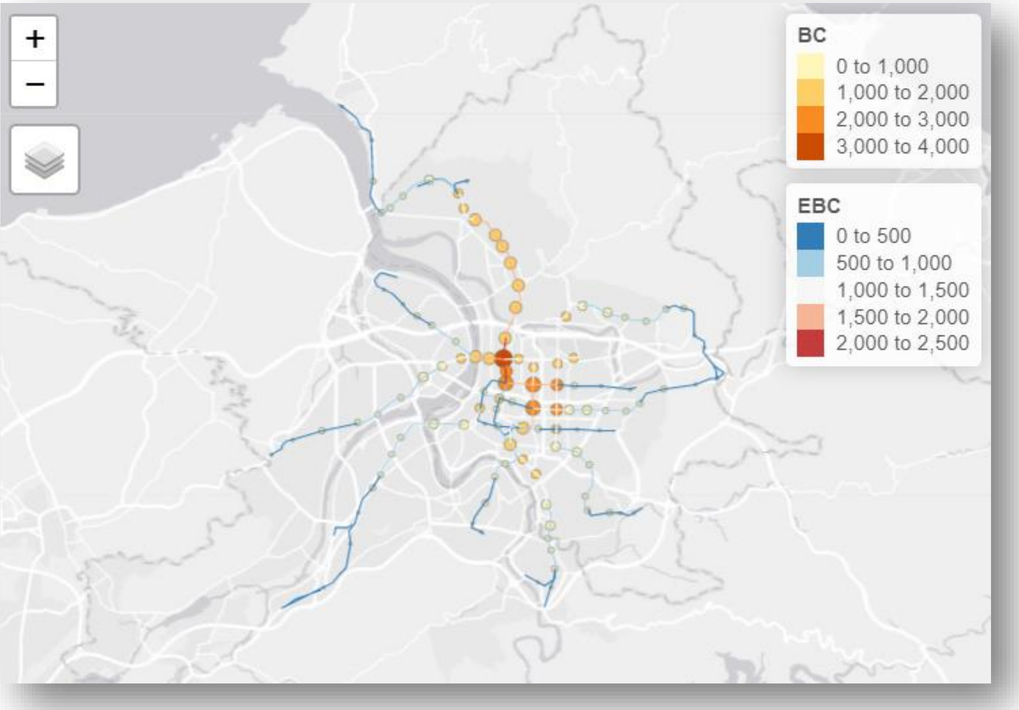
```
> epath_sf = mypath_graph %>% activate(edges) %>% as_tibble() %>% st_as_sf()
> mypath_lyr = tm_shape(epath_sf) + tm_lines(col="red", lwd=2)

> from_node_sf = road_sfnet %>%
        activate(nodes) %>%
        filter(nodeID == 200) %>%
        as_tibble() %>% st_as_sf()
> from_node_lyr = tm_shape(from_node_sf) +
         tm_dots(col="green") #to_node一樣方法

> map + mypath_lyr +
        from_node_lyr + to_node_lyr
```

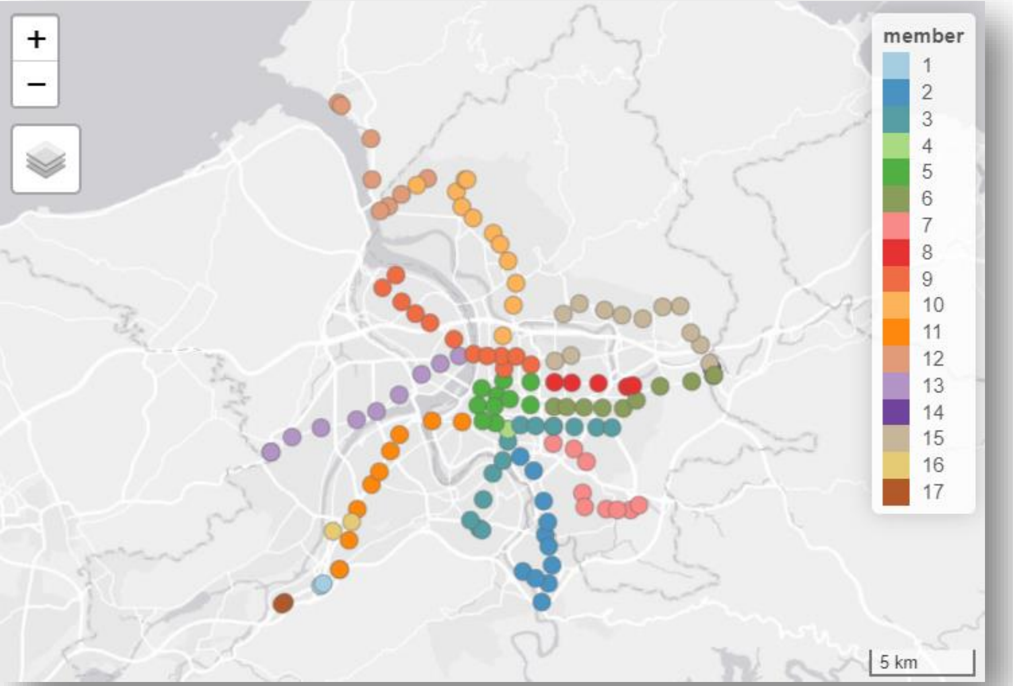


Self-Practice

- 讀取GIS-T交通網路地理資訊倉儲系統的台北捷運路線
- 計算介數中心性 (Betweenness Centrality)
- 進行分群


+
−
臺北松山機場
Tai Pei Song Shan Ji Chang
Leaflet | Tiles © Esri — Esri, DeLorme, NAVTEQ


+
−
BC
0 to 1,000
1,000 to 2,000
2,000 to 3,000
3,000 to 4,000
EBC
0 to 500
500 to 1,000
1,000 to 1,500
1,500 to 2,000
2,000 to 2,500


+
−
member
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
5 km