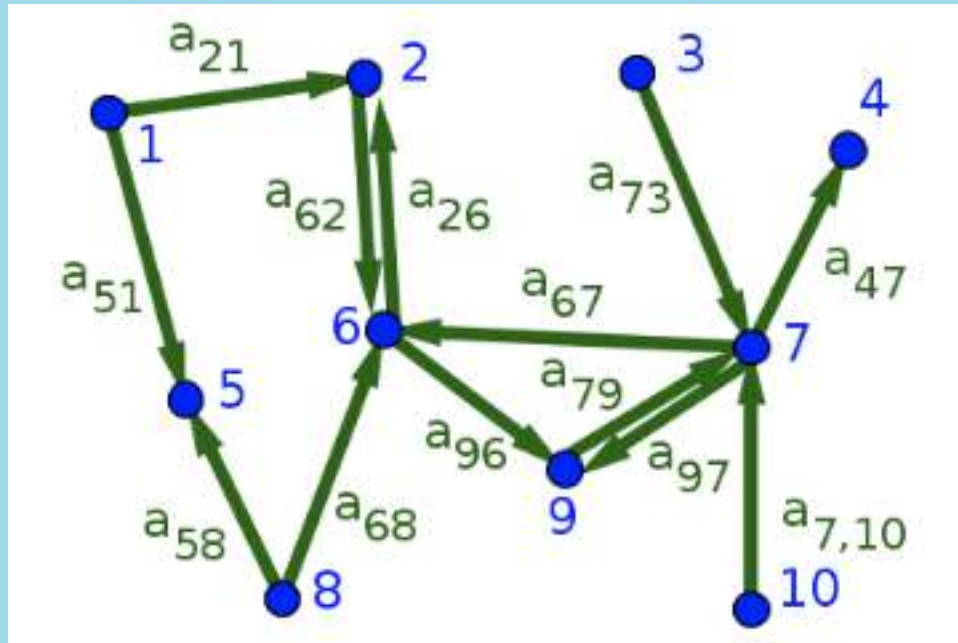


# Movement and Networks II

HES 505 Fall 2023: Session 26

Matt Williamson

# Describing networks for analysis



$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Common measures

- Graph-level: density, diameter, distance
- Component-level: density, distribution
- Node-level: centrality, degree-distribution

# Common questions

- What are the shortest paths across the network?
- Where are the most important locations for maintaining the network?
- How does the loss of a node alter the subsequent configuration of the network?
- How do we translate typical movement paths into network structures?

# An example

# Sage Grouse in the West

Reading layer `GRSG\_2015\_USFWS\_StatusReview\_PACs' from data source

`/Users/mattwilliamson/Websites/isdrfall23/slides/data/GRSG\_2015\_USFWS\_StatusRe'  
using driver `ESRI Shapefile'

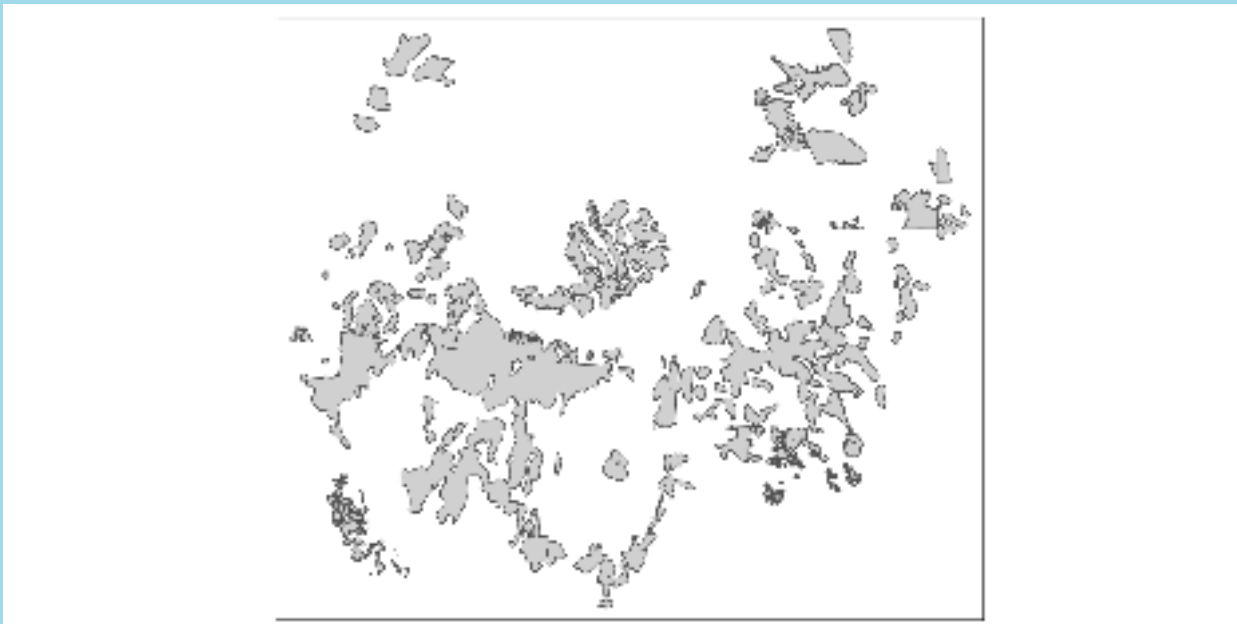
Simple feature collection with 301 features and 9 fields

Geometry type: POLYGON

Dimension: XY

Bounding box: xmin: -1361708 ymin: 381165.6 xmax: 147308.4 ymax: 1661769

Projected CRS: GRSG Range Wide



# The Data

```
1 head(sg.pacs[,1:9])
```

Simple feature collection with 6 features and 9 fields

Geometry type: POLYGON

Dimension: XY

Bounding box: xmin: -1183048 ymin: 446911.4 xmax: -1096618 ymax: 472038.5

Projected CRS: GRSG Range Wide

	OBJECTID	UniqueID	Population	MgmtZone	ID_Name	Acres	Shape_Leng
1	1	401	Bi-State	MZ3	401-Bi-State-MZ3	444.2940	5691.176
2	2	362	Bi-State	MZ3	362-Bi-State-MZ3	3553.0707	25207.100
3	3	400	Bi-State	MZ3	400-Bi-State-MZ3	494.7130	5958.419
4	4	399	Bi-State	MZ3	399-Bi-State-MZ3	995.5216	10911.550
5	5	398	Bi-State	MZ3	398-Bi-State-MZ3	395.0896	7904.861
6	6	361	Bi-State	MZ3	361-Bi-State-MZ3	8252.1309	60989.117

	Shape_Area	Bi_State	geometry
1	1797994	Yes	POLYGON ((-1181624 457935.6...
2	14378767	Yes	POLYGON ((-1102151 456210.8...
3	2222222	Yes	POLYGON ((-1107222 456250.1...

# Preparing the adjacency matrix

```
1 sg.pacs.cent <- sg.pacs %>%  
2   filter(., MgmtZone == "MZ3") %>%  
3   st_centroid(sg.pacs, of_largest_polygon = TRUE)
```



# Preparing the Adjacency Matrix

```
1 sg.pacs.dist <- st_distance(sg.pac  
2  
3 threshold <- units::as_units(50, "  
4  
5 adj.mtx <- sg.pacs.dist < threshol
```

	1	2	3	4	5
1	TRUE	FALSE	FALSE	FALSE	FALSE
2	FALSE	TRUE	TRUE	TRUE	TRUE
3	FALSE	TRUE	TRUE	TRUE	TRUE
4	FALSE	TRUE	TRUE	TRUE	TRUE
5	FALSE	TRUE	TRUE	TRUE	TRUE

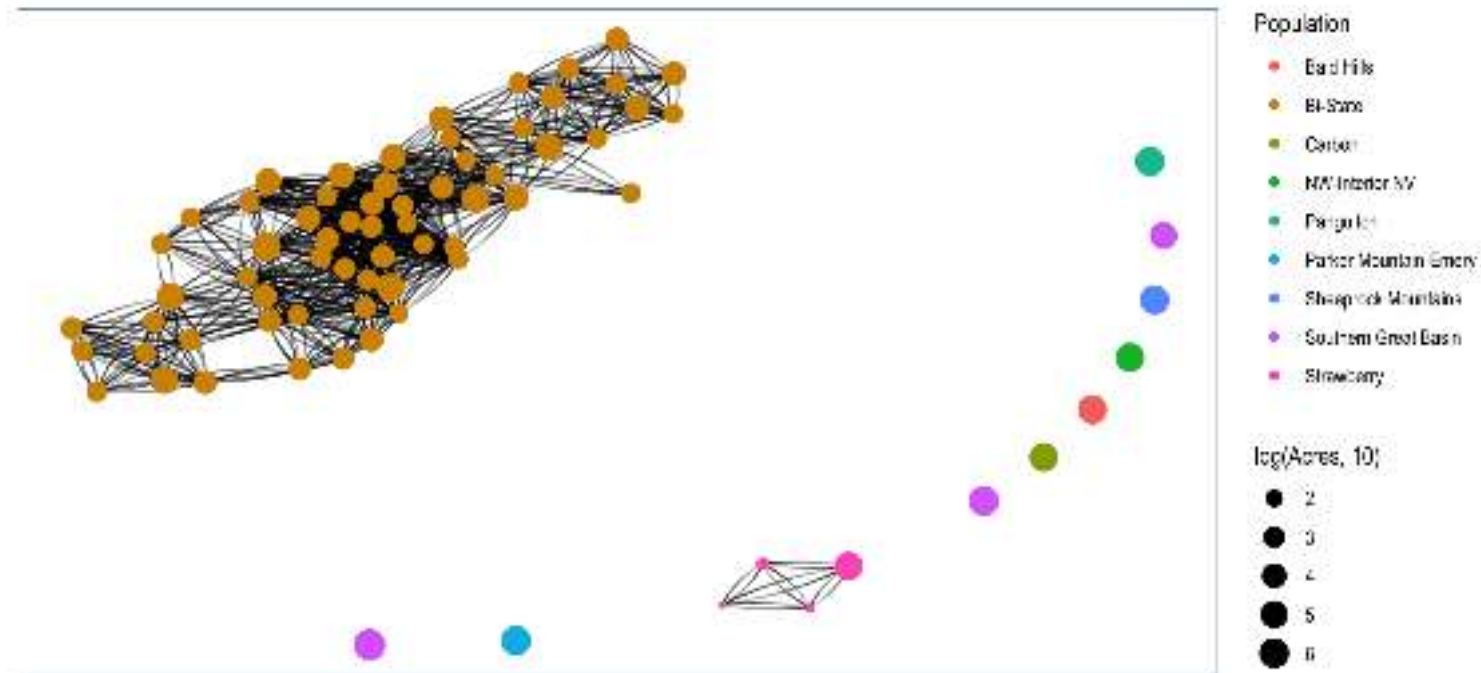
# Preparing the Adjacency Matrix

```
1 adj.mtx <- adj.mtx *1
2 diag(adj.mtx) <- 0
3
4 dimnames(adj.mtx) <- list(
5   sg.pacs.cent$UniqueID,
6   sg.pacs.cent$UniqueID)
```

	401	362	400	399	398
401	0	0	0	0	0
362	0	0	1	1	1
400	0	1	0	1	1
399	0	1	1	0	1
398	0	1	1	1	0

# Graphing your network

```
1 sg.graph <- igraph::graph_from_adjacency_matrix(adj.mtx) %>%  
2   as_tbl_graph(directed = FALSE,  
3               node_key = "UniqueID") %>%  
4   left_join(., sg.pacs.cent,  
5             by = c("name" = "UniqueID"))
```



# Evaluating Network Metrics

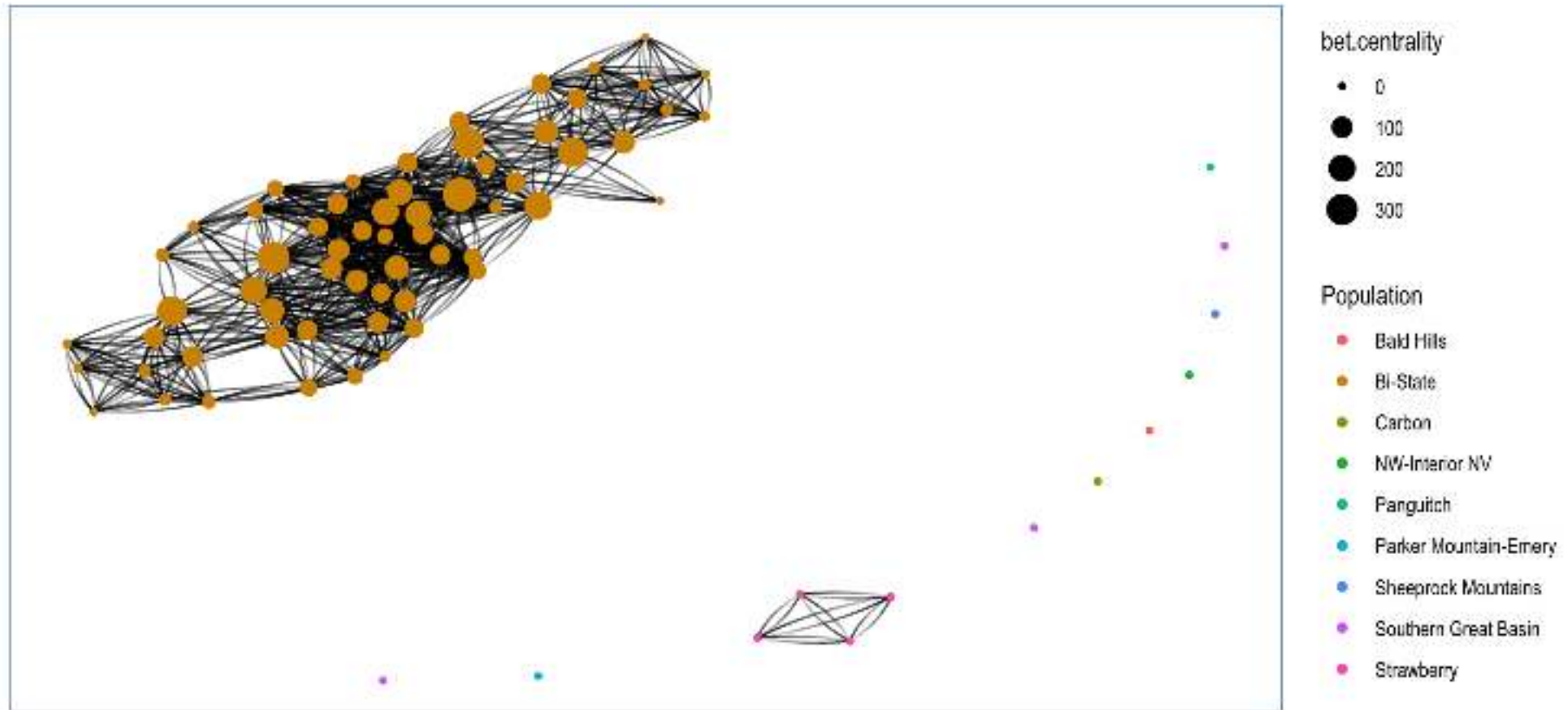
# Common Metrics

- **Betweenness Centrality:** the sum of all of the shortest paths that flow through that node.
- **Degree** reflects the number of connections a node has to other nodes.
- **Component size:** the number of nodes in a group that is connected to each other, but disconnected from the rest of the graph
- **Degree Distribution:** Describes the general connectedness of all the nodes in a network; vulnerability

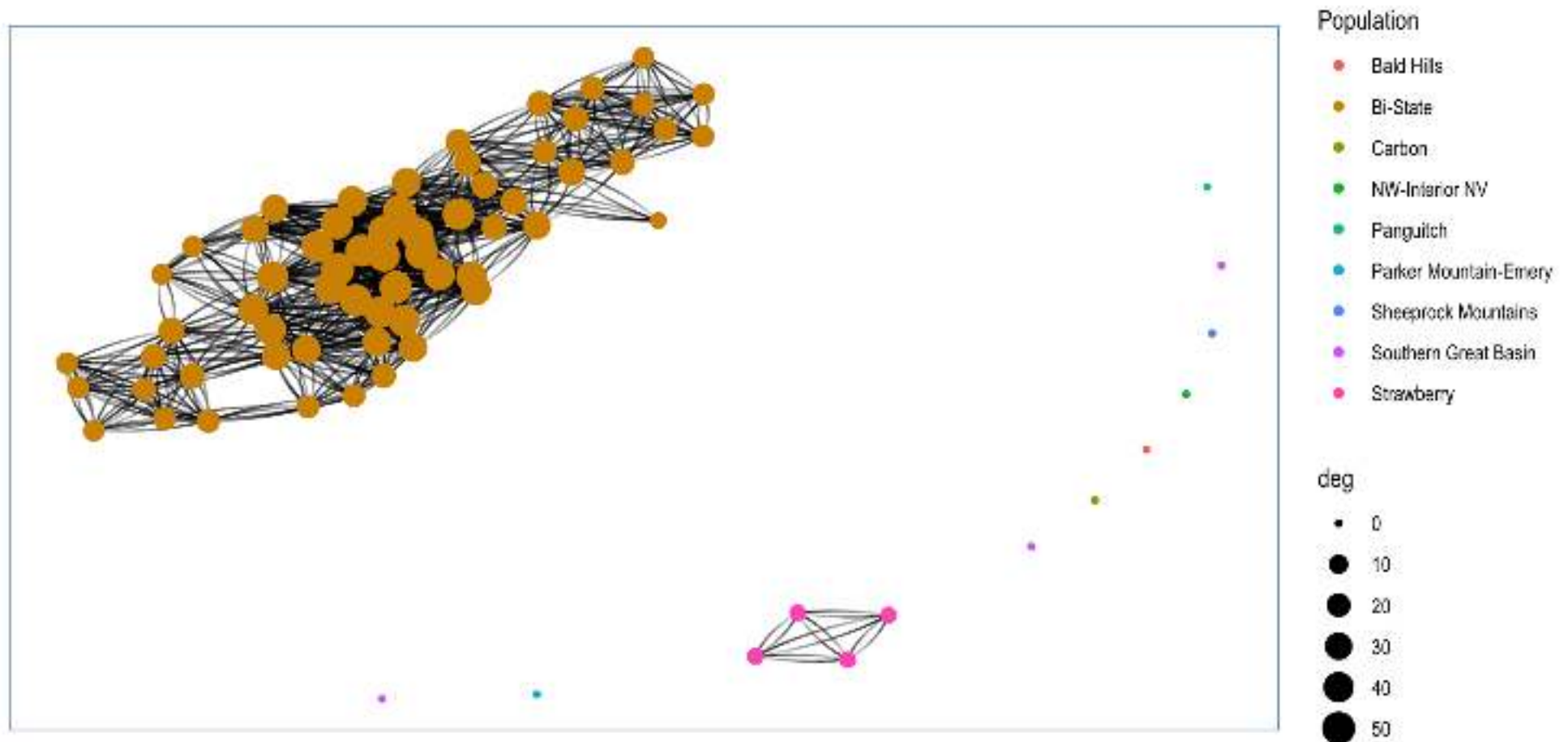
# Estimating metrics

```
1 sg.graph.mets <- sg.graph %>%  
2   activate(nodes) %>%  
3   mutate(., bet.centralitity = centrality_betweenness(),  
4           deg = degree(.))
```

# Estimating metrics: Betweenness



# Estimating metrics: Degree





# Estimating metrics: Component Size

```
1 comps <- components(sg.graph)
2 comps$csizes
```

```
[1] 61  1  1  1  1  1  4  1  1  1  1
```

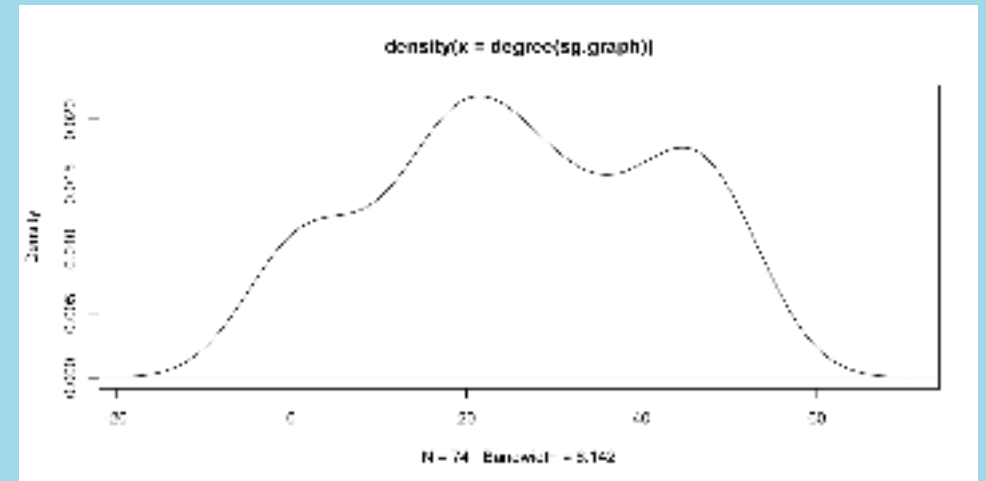
```
1 comps$membership
```

```
401 362 400 399 398 361 360 397 359 396 395 358 353 394 393 392 357 354 355
352
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
1
351 350 391 390 389 388 356 387 386 385 349 384 383 382 381 380 379 378 377
376
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
1
375 374 239 347 345 373 372 348 371 346 370 369 368 344 367 366 365 364 343
242
  1   1   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
3
342 363 341 238 237 241 234 232 235 236 233 244 243 240
  1   1   1   4   5   6   7   8   7   7   7   9  10  11
```

# Estimating metrics: Degree distribution

```
[1] 26.10811
```

```
[1] 16.13943
```



# Experimenting

# Setting up the data

```
1 thresholds <- as_units(c(1, 10, 25, 50, 100, 150, 250, 500), "km")
2 thresh.df <- data.frame(thresh.dist = rep(NA, length(thresholds)),
3                          mean.between = rep(NA, length(thresholds)),
4                          ncomps = rep(NA, length(thresholds)),
5                          mean.deg = rep(NA, length(thresholds)))
6 sg.pacs.cent <- sg.pacs %>%
7   st_centroid(sg.pacs, of_largest_polygon = TRUE)
8 sg.pacs.dist <- st_distance(sg.pacs.cent)
```

# Iterating

```
1 for (i in 1:length(thresholds)){
2   adj.mtx <- sg.pacs.dist < thresholds[i]
3   adj.mtx <- adj.mtx * 1
4   diag(adj.mtx) <- 0
5   dimnames(adj.mtx) <- list(sg.pacs.cent$UniqueID, sg.pacs.cent$UniqueID)
6
7   sg.graph <- igraph::graph_from_adjacency_matrix(adj.mtx) %>%
8   as_tbl_graph(directed = FALSE,
9               node_key = "UniqueID") %>%
10  left_join(., sg.pacs.cent,
11            by = c("name" = "UniqueID"))
12  thresh.df$thresh.dist[i] <- thresholds[i]
13  thresh.df$mean.between[i] <- mean(betweenness(sg.graph, directed = FALSE))
14  thresh.df$ncomps[i] <- length(components(sg.graph)$csize)
15  thresh.df$mean.deg[i] <- mean(degree(sg.graph))
16 }
```

# Collecting Data

```
1 thresh.long <- thresh.df %>%  
2   tidyr::pivot_longer(!thresh.dist,  
3                       names_to = "metric",  
4                       values_to = "estimate")  
5  
6 ggplot(data = thresh.long, aes(x= thresh.dist, y = estimate)) +  
7   geom_line()+  
8   facet_wrap(vars(metric)) +  
9   ggtitle("Whole Network")
```

