

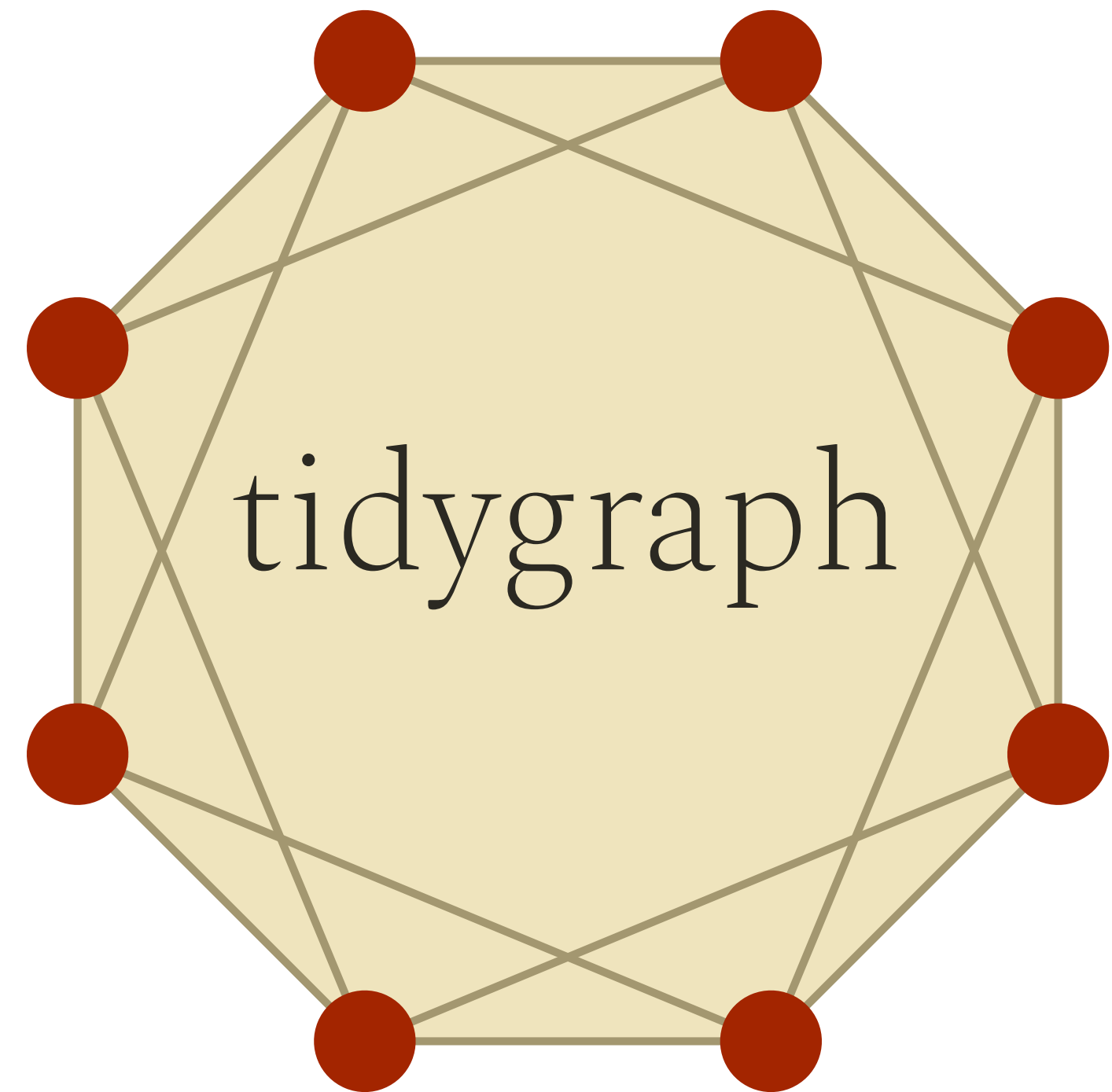
tidygraph

Tidy Geospatial Networks in R

Thomas Lin Pedersen — June 16th 2020

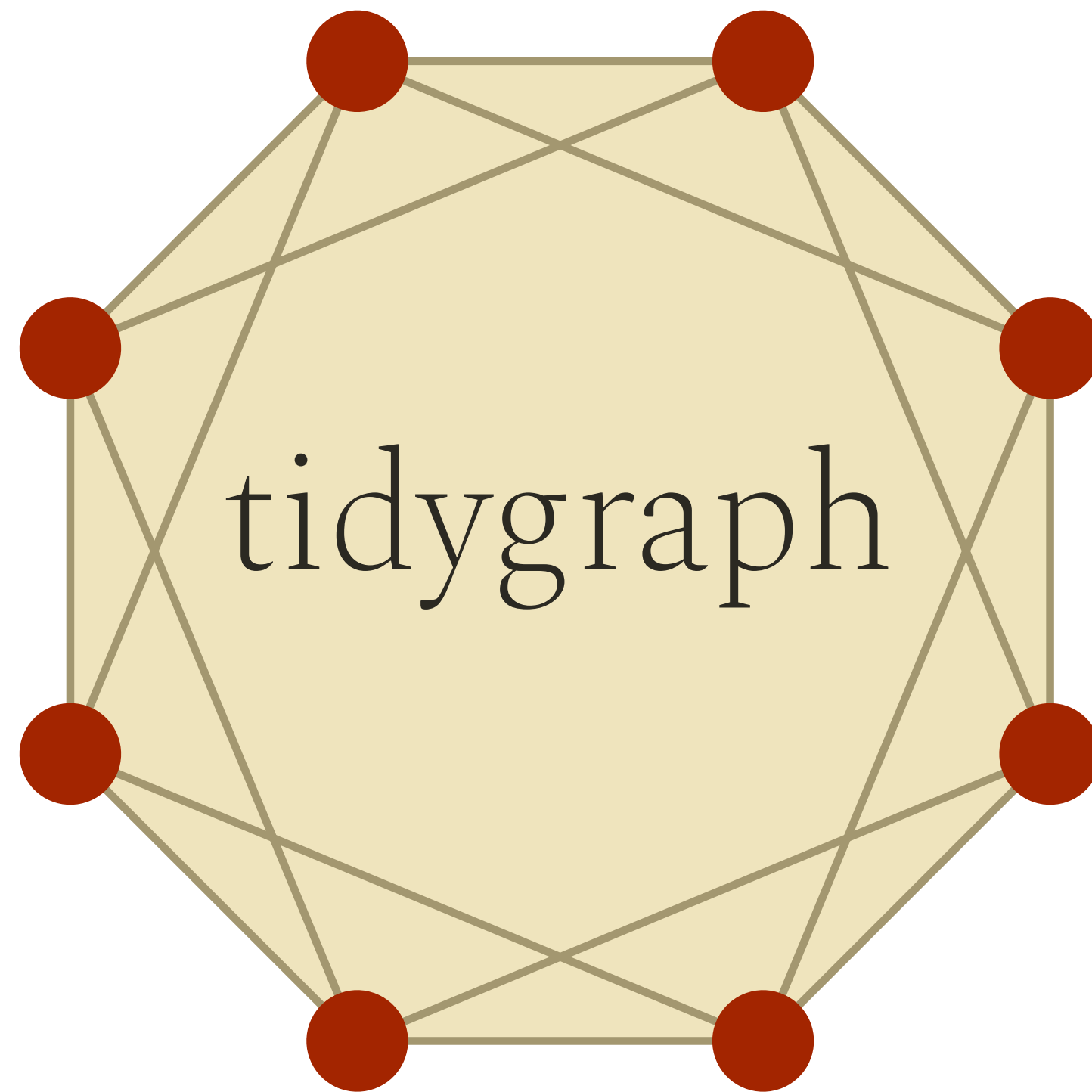
Why?

- ggraph needed a good companion
- Make network data accessible
- Make network analysis accessible



Philosophy

Manipulation



Tight
coupling

Visualisation



Philosophy

- API over performance
- Embrace tidy
- Predictable API
- Predictable output
- `mutate()` friendly functions

Key Concepts

`activate()`

- A `tbl_graph` is two tidy tables
- `activate()` decides which one you operate on
- `%N>%` and `%E>%` are pipe shortcuts

Key Concepts

Algorithms

- Meant for use within dplyr verbs
- Knows the graph and nodes/edges being worked on
- Will always return a vector that can be used in mutate

Node and edge centrality (dbl)

- **centrality_pagerank()**

Node and edge types (lgl)

- **node_is_leaf()**

- **edge_is_loop()**

Node pair measures (dbl)

- **node_distance_[to|from]()**

- **node_similarity_with()**

Node measures (dbl)

- **node_constraint()**

Communities (int)

- **group_walktrap()**

Ranking (int)

- **node_rank_hclust()**

Key Concepts

Algorithms — spur of the moment thought

- `st_filter()` vs `filter()`
with predicate

```
graph %>% filter(  
  node_inside(rect)  
)
```

- Other spatial predicates
- `edge_intersects()`
- `edge_crosses()`
- `node_close_to()`
- ...

Key Concepts

Morphing

- Sometimes a different “view” of the graph is needed
 - morphing is the act of temporarily changing the graph topology
 - changes done while morphed is carried over
- `to_complement()`
 - `to_minimum_spanning_tree()`
 - `to_subgraph()`
 - `to_bfs_tree()`
 - `to_components()`
 - `to_linegraph()`
 - `to_local_neighborhood()`
 - `to_undirected()`
 - ...

Example: Iron Mans closest enemy

```
> marvel_graph %>%
+   activate(edges) %>%
+   morph(to_subgraph, class == 'Ally') %>%
+   activate(nodes) %>%
+   mutate(friend_dist = bfs_dist(id == 'Iron_Man', mode = 'all')) %>%
+   unmorph() %>%
+   filter(id %in% Enemies[[which(id == 'Iron_Man')]]) %>%
+   arrange(friend_dist)
Subsetting by edges
# A tbl_graph: 25 nodes and 161 edges
#
# A bipartite multigraph with 2 components
#
# Node Data: 25 x 9 (active)
  id            characters  type Affiliation  Family  Allies  Enemies Appearance friend_dist
  <chr>         <chr>      <chr>  <list>      <list>  <list>  <list>  <list>      <int>
1   Winter_Soldier Winter Soldier Both  <chr [3]> <chr [1]> <chr [17]> <chr [14]> <chr [9]>      2
2     Loki         Loki    Both  <chr [2]> <chr [5]> <chr [9]>  <chr [23]> <chr [11]>      2
3   Ant-Man       Ant-Man  Hero  <chr [2]> <chr [1]> <chr [6]>  <chr [9]>  <chr [6]>      2
4     List       List    Villain <chr [2]> <chr [1]> <chr [9]>  <chr [10]> <chr [3]>      2
5     Stern      Stern    Villain <chr [2]> <chr [1]> <chr [1]>  <chr [1]>  <chr [2]>      2
6 Leviathan_(Creature) Leviathan (Creature) Villain <chr [1]> <chr [1]> <chr [1]>  <chr [1]>  <chr [6]>      3
# ... with 19 more rows
#
# Edge Data: 161 x 4
  from  to weight  class
  <int> <int> <dbl>  <chr>
1     1   1     3 Affiliation
2     1   4     1 Affiliation
3     1  17     1 Affiliation
# ... with 158 more rows
```

Example: Pillars of the Franchise

```
> marvel_graph %>%
+   activate(edges) %>%
+   filter(class == 'Appearance') %>%
+   convert(to_components, .clean = TRUE) %>%
+   activate(nodes) %>%
+   morph(to_split, group_infomap()) %>%
+   mutate(group_degree = centrality_degree(mode = 'all', loops = FALSE)) %>%
+   filter(group_degree == max(group_degree)) %>%
+   crystallise() %>%
+   pull(graph) %>%
+   map(. %>% pull(characters))
Subsetting by nodes
[[1]]
[1] "Hulk"      "Nick Fury" "Iron Man"

[[2]]
[1] "Claire Temple" "Daredevil"    "Jessica Jones" "Luke Cage"    "Karen Page"    "Madame Gao"

[[3]]
[1] "Drax the Destroyer" "Gamora"        "Groot"        "Nebula"        "Rocket Raccoon" "Star-Lord"
[7] "Thanos"

[[4]]
[1] "Happy Sam Sawyer" "Daniel Sousa"    "Junior Juniper" "Jason Wilkes"    "Edwin Jarvis"    "Pinky Pinkerton" "Calvin Chadwick"
[8] "Dottie Underwood" "Rufus Hunt"      "Fyodor"        "Vernon Masters" "Leet Brannis"    "Spider Raymond"  "Jerome Zandow"
[15] "Ernst Mueller"    "Sasha Demidov"   "Wilhelm Keitel" "Arnim Zola"      "Whitney Frost"   "Joseph Manfredi" "Johann Fennhoff"
[22] "Otto Mink"

. . .
```

Future

- I always wanted support for spatial and temporal networks
- One of them I no longer need to think about 🎉
- (still thinking about a minimal class as well)
- Provide build in support for sfnetwork in ggraph for visualisation
- changes in tidygraph to support sfnetwork