

12 — Relational Data (2)

Kieran Healy

January 23, 2024

Relational Data (pt 2)

Load the packages, as always

```
library(here)      # manage file paths  
library(socviz)    # data and some useful functions  
library(tidyverse) # your friend and mine
```

Specialty packages

```
library(tidygraph) # tidy management of relational data
library(ggraph)    # geoms for drawing graphs

#remotes::install_github("kjhealy/kjhnet")
library(kjhnet)    # some network datasets
```

The Iliad

Who is the most central figure in *The Iliad*?

We'll use a "dataset" from *The Iliad* to explore some ideas about ways of measuring centrality and how they express different concepts of social status.

Thanks to Gabriel Rossman (UCLA) for the data and the topic idea. This lecture follows his exposition.

Degree centrality

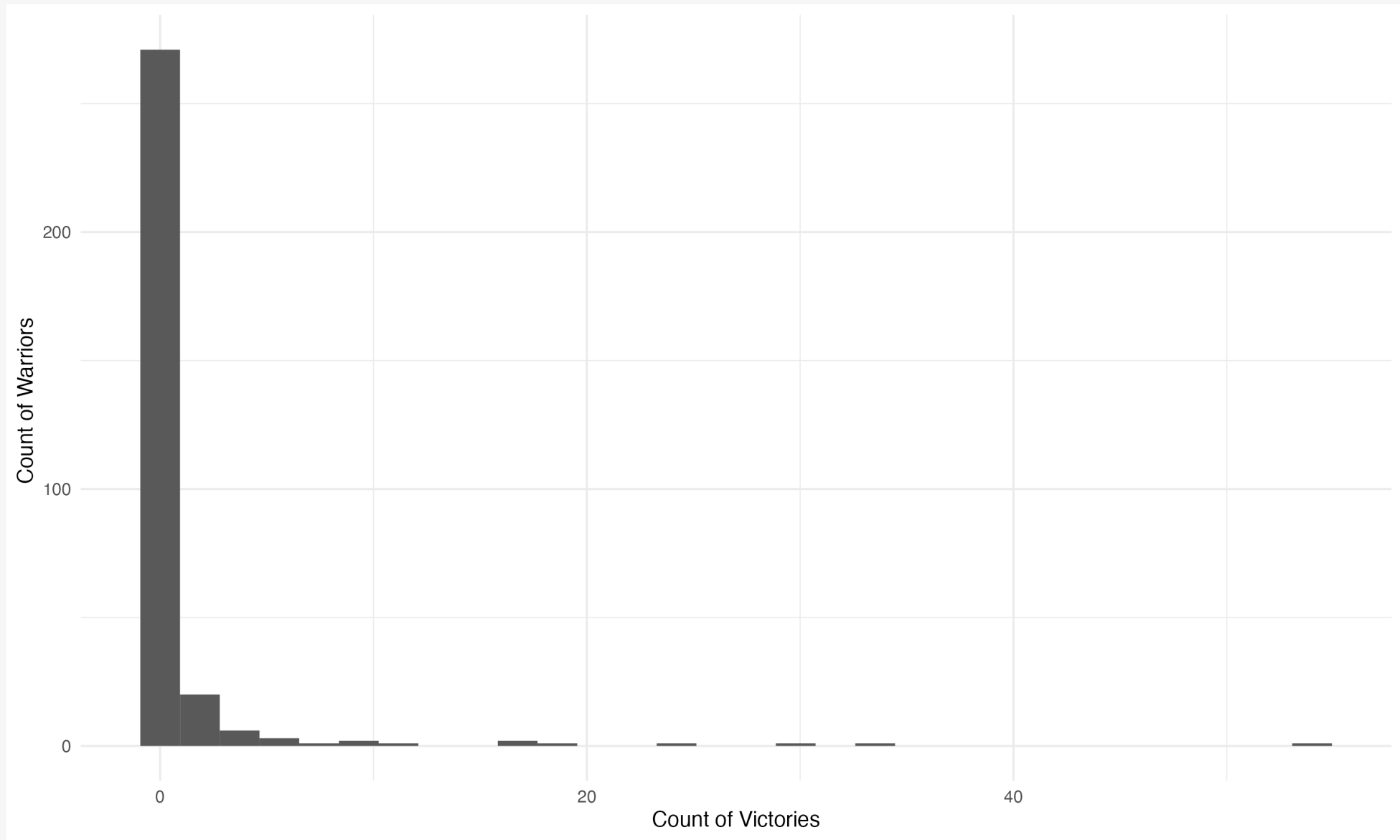
```
il_tidy
```

```
# A tbl_graph: 311 nodes and 292 edges
#
# A directed multigraph with 23 components
#
# Node Data: 311 × 3 (active)
  id name          affil
  <int> <chr>        <chr>
1     1 Antilochus  A
2     2 Agenor     T
3     3 Telamonian Ajax A
4     4 Antiphus   T
5     5 Odysseus   A
6     6 Peirous    T
7     7 ThoasAndraemon A
8     8 Diomedes   A
9     9 Agamemnon  A
10    10 Idomeneus A
# i 301 more rows
#
```

Degree centrality

```
out ← il_tidy ▷  
  activate(nodes) ▷  
  mutate(centrality = centrality_degree()) ▷  
  as_tibble() ▷  
  ggplot(mapping = aes(x = centrality)) +  
  geom_histogram() +  
  labs(x = "Count of Victories", y = "Count of Warriors ") +  
  theme_minimal() +  
  theme(legend.position = "top")
```

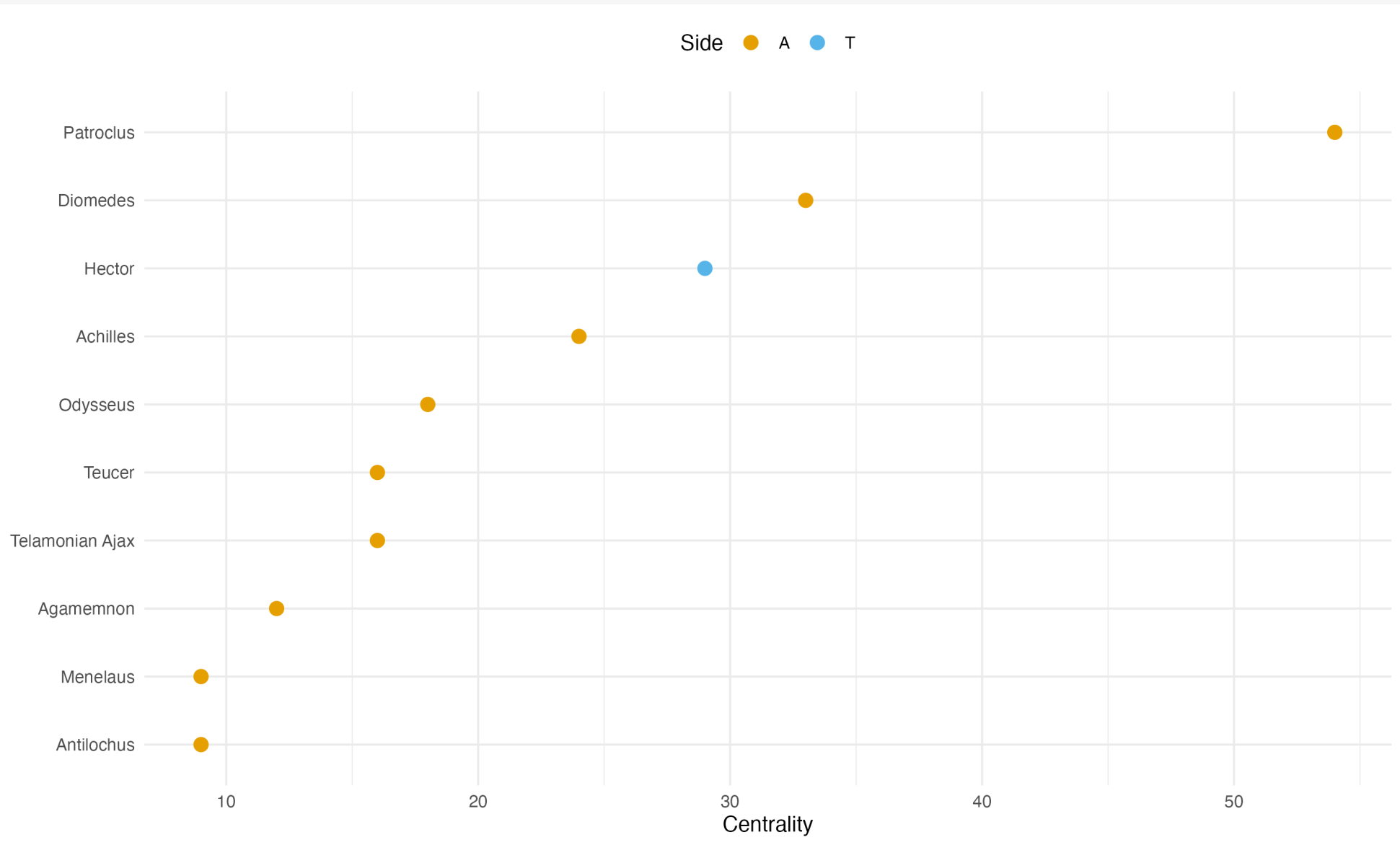
```
print(out)
```



Degree Centrality (with affiliation)

```
out ← il_tidy ▷
  activate(nodes) ▷
  mutate(centrality = centrality_degree()) ▷
  as_tibble() ▷
  arrange(desc(centrality)) ▷
  top_n(10, wt = centrality) ▷
  ggplot(mapping = aes(x = centrality,
                       y = reorder(name, centrality),
                       color = affil)) +
  geom_point(size = 3) +
  labs(x = "Centrality", y = NULL, color = "Side") +
  theme_minimal() +
  theme(legend.position = "top")
```

```
print(out)
```

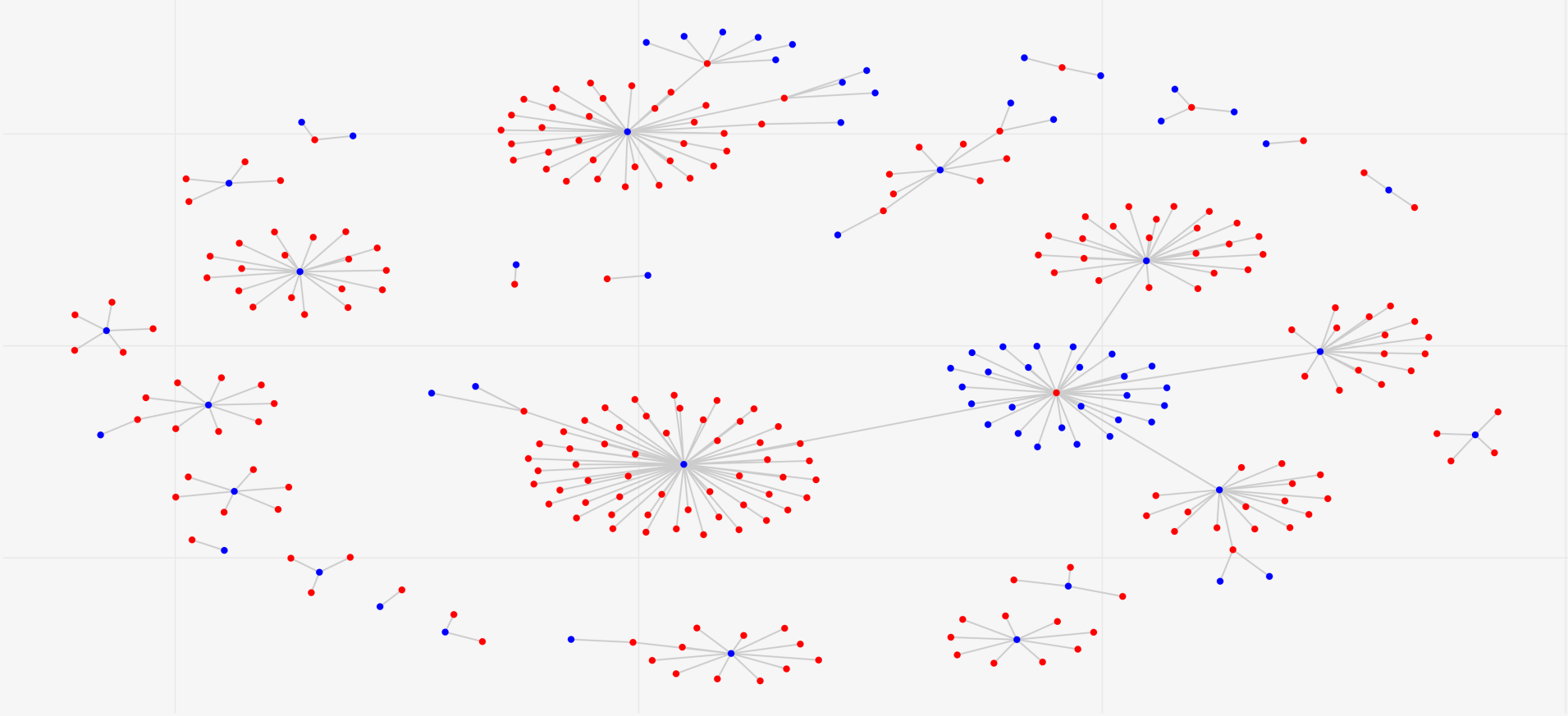


Graph representation

```
out ← il_tidy ▷
  activate(nodes) ▷
  ggraph(layout = "fr") +
  geom_edge_link(color = "gray80") +
  geom_node_point(aes(color = affil)) +
  scale_color_manual(values = c("blue", "red"),
                     labels = c("Athenian", "Trojan")) +
  guides(color = guide_legend(title = "Side")) +
  labs(title = "Violence in The Iliad") +
  theme(plot.title = element_text(size = rel(3)))
```

Violence in The Iliad

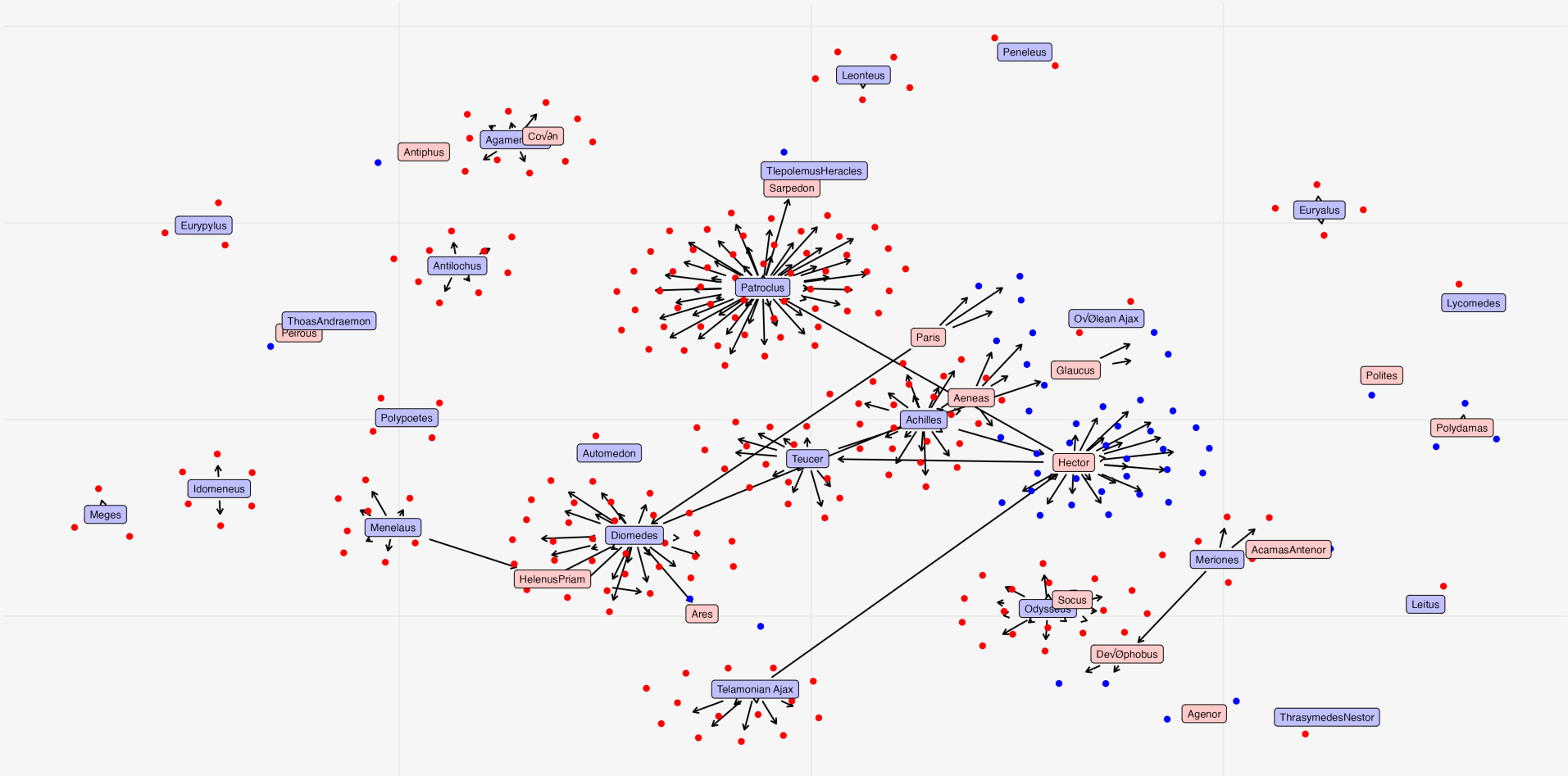
Side • Athenian • Trojan



... with labels

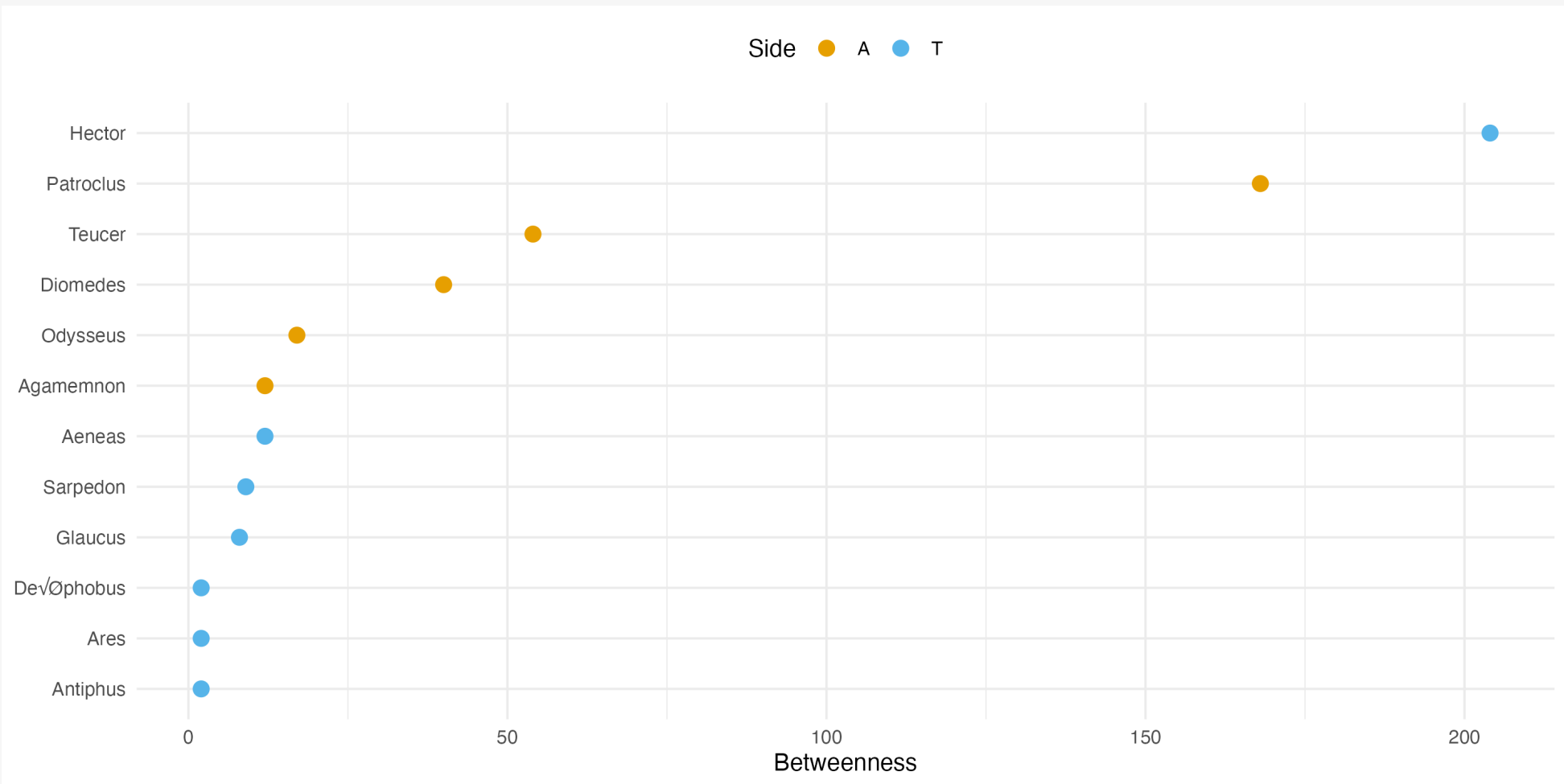
```
label_colors ← c(prismatic::clr_lighten("blue", 0.7),
                 prismatic::clr_lighten("red", 0.7))
out ← il_tidy ▷
  activate(nodes) ▷
  mutate(centrality = centrality_degree(mode = "out")) ▷
  ggraph(layout = "graphopt") +
  geom_edge_link(aes(start_cap = label_rect(node1.name),
                    end_cap = label_rect(node2.name)),
                arrow = arrow(length = unit(1.5, 'mm')))) +
  geom_node_point(aes(color = affil)) +
  scale_color_manual(values = c("blue", "red"),
                    labels = c("Athenian", "Trojan")) +
  guides(color = "none", fill = "none") +
  geom_node_label(aes(filter = centrality > 0,
                    label = name, fill = affil),
                size = rel(2.5)) +
  scale_fill_manual(values = label_colors) +
  labs(title = "Violence in The Iliad") +
  theme(plot.title = element_text(size = rel(3)))
```

Violence in The Iliad



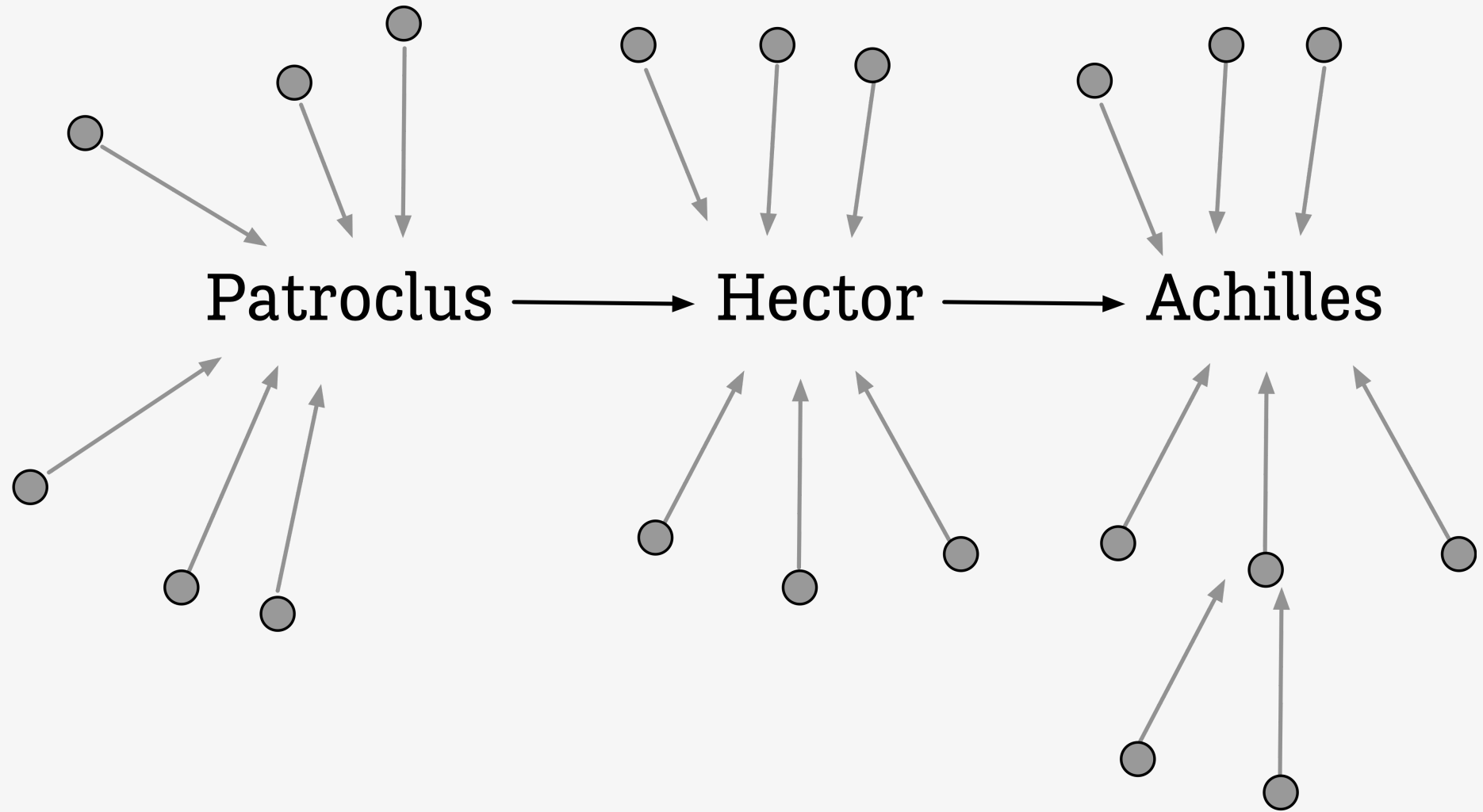
Betweenness centrality

```
out ← il_tidy ▷
  activate(nodes) ▷
  mutate(centrality = centrality_degree(),
         betweenness = centrality_betweenness()) ▷
  as_tibble() ▷
  arrange(desc(betweenness)) ▷
  top_n(10, wt = betweenness) ▷
  ggplot(mapping = aes(x = betweenness,
                      y = reorder(name,
                                   betweenness), color = affil)) +
  geom_point(size = 3) +
  labs(x = "Betweenness", y = NULL, color = "Side") +
  theme_minimal() +
  theme(legend.position = "top")
```





Sarcophagus



Someone → is killed by

Alpha centrality

```
out ← il_tidy ▷
  activate(edges) ▷
  filter(act = "kills") ▷
  reroute(from = to, to = from) ▷
  activate(nodes) ▷
  mutate(alpha = centrality_alpha()) ▷
  as_tibble() ▷
  arrange(desc(alpha)) ▷
  top_n(10, wt = alpha) ▷
  ggplot(mapping = aes(x = alpha,
                      y = reorder(name, alpha),
                      color = affil)) +
  geom_point(size = 3) +
  labs(x = "Alpha Centrality", y = NULL, color = "Side") +
  theme_minimal() +
  theme(legend.position = "top")
```

