Tutorial - Network data

Ariane Ducellier

2023-12-05

Load R packages.

```
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

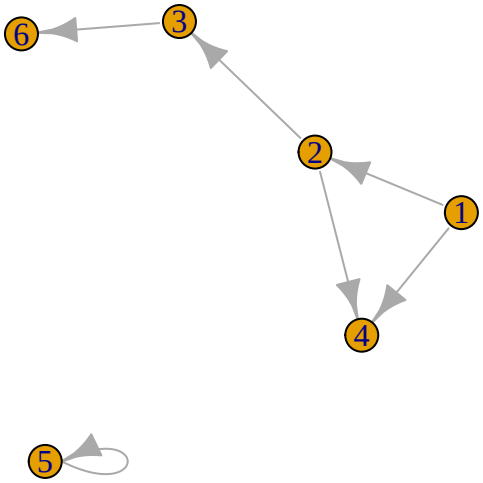
```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
```

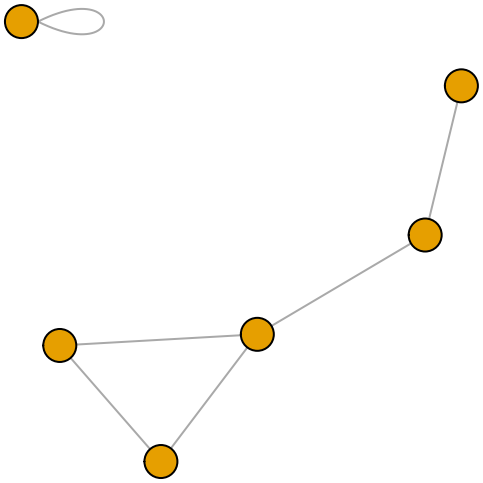
```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::%--%()      masks igraph::%--%()
## x dplyr::as_data_frame() masks tibble::as_data_frame(), igraph::as_data_frame()
## x purrr::compose()       masks igraph::compose()
## x tidyr::crossing()      masks igraph::crossing()
## x dplyr::filter()        masks stats::filter()
## x dplyr::lag()           masks stats::lag()
## x purrr::simplify()      masks igraph::simplify()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Simple synthetic graph

```
gd <- graph(c(1,2, 2,3, 2,4, 1,4, 5,5, 3,6))
plot(gd)
```




```
gu <- graph(c(1,2, 2,3, 2,4, 1,4, 5,5, 3,6), directed=FALSE)
plot(gu, vertex.label=NA)
```



A more complex graph dataset

```
refugeedata <- read_delim("../data/dataset-unhcr-refugees-data-sudan-south-sudan-2016.csv", "," )
```

```
## Rows: 180 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Country or territory of asylum or residence, ISO_A3_residence, Coun...
## dbl (2): Refugees_assisted, Total_refugees
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
edges <- refugeedata %>%
  drop_na() %>%
  select("ISO_A3_origin",
         "ISO_A3_residence",
         "Total_refugees") %>%
  rename("from"="ISO_A3_origin",
         "to"="ISO_A3_residence",
```

```
         "weight"="Total_refugees") %>%
  slice_max(n=15, order_by=weight)
```

```
nodesFrom <- edges %>%
  select("from") %>%
  distinct() %>%
  rename("country"="from")

nodesTo <- edges %>%
  select("to") %>%
  distinct() %>%
  rename("country"="to")

nodes <- bind_rows(nodesFrom, nodesTo) %>%
  distinct()
```

```
# Generate ID's for all nodes
nodes$ID <- seq.int(nrow(nodes))

# Create the network object
net <- graph_from_data_frame(d=edges, vertices=nodes, directed=TRUE)
```

```
par(mar=c(0, 0, 0, 0))

V(net)$size <- log(strength(net)) + 4
E(net)$width <- log(edges$weight / 350)

edgesSSD <- incident(net, V(net)[name=="SSD"], mode="out")
edgesSDN <- incident(net, V(net)[name=="SDN"], mode="out")
ecol <- rep("gray", ecount(net))
ecol[edgesSSD] <- "orange"
ecol[edgesSDN] <- "gold"

vcol <- rep("gray", vcount(net))
vcol[V(net)$name=="SSD"] <- "orange"
vcol[V(net)$name=="SDN"] <- "gold"

l <- layout_with_lgl(net, maxiter=93)

plot(net,
     main="Major flows of refugees to and from Sudan and South Sudan in 2016",
     sub="Source: UNHCR, 2016",
     layout=l,
     edge.color=ecol,
     edge.curved=.25,
     edge.arrow.size=log(E(net)$weight)/6,
     edge.label=E(net)$weight,
     edge.label.color="black",
     edge.label.cex=.7,
     vertex.color=vcol,
     vertex.label.color="black",
     vertex.label.cex=log(strength(net))/12)
```

Major flows of refugees to and from Sudan and South Sudan in 2016