

# qgraph example

Justin

2024-01-24

## Load packages

```
library(qgraph)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## 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 dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
##
##   set_names
##
## The following object is masked from 'package:tidyr':
##
##   extract
```

```
library(patchwork)
```

## create first random network with 12 nodes and random edges and weights

```
data1 <- tibble::tribble(
  ~Source, ~Target, ~Weight,
  "Node_1", "Node_1", 0,
```

```
"Node_1", "Node_2", -0.210315659,
"Node_1", "Node_3", 0.247123625,
"Node_1", "Node_4", 0.022205467,
"Node_1", "Node_5", 0.426595225,
"Node_1", "Node_6", 0,
"Node_1", "Node_7", 0,
"Node_1", "Node_8", 0.014395898,
"Node_1", "Node_9", 0,
"Node_1", "Node_10", 0.30805752,
"Node_1", "Node_11", 0,
"Node_1", "Node_12", -0.235996971,
"Node_2", "Node_1", 0.076964418,
"Node_2", "Node_2", 0,
"Node_2", "Node_3", 0,
"Node_2", "Node_4", -0.082147755,
"Node_2", "Node_5", 0,
"Node_2", "Node_6", 0,
"Node_2", "Node_7", -0.517367563,
"Node_2", "Node_8", 0,
"Node_2", "Node_9", 0,
"Node_2", "Node_10", 0.251413943,
"Node_2", "Node_11", 0,
"Node_2", "Node_12", -0.342190115,
"Node_3", "Node_1", 0.265160837,
"Node_3", "Node_2", 0.180934396,
"Node_3", "Node_3", 0,
"Node_3", "Node_4", 0,
"Node_3", "Node_5", 0,
"Node_3", "Node_6", 0,
"Node_3", "Node_7", 0.63793984,
"Node_3", "Node_8", 0,
"Node_3", "Node_9", 0,
"Node_3", "Node_10", 0,
"Node_3", "Node_11", 0,
"Node_3", "Node_12", -0.305990619,
"Node_4", "Node_1", 0,
"Node_4", "Node_2", 0,
"Node_4", "Node_3", 0.083301502,
"Node_4", "Node_4", 0,
"Node_4", "Node_5", 0,
"Node_4", "Node_6", 0,
"Node_4", "Node_7", 0,
"Node_4", "Node_8", 0,
"Node_4", "Node_9", 0.690517186,
"Node_4", "Node_10", 0,
"Node_4", "Node_11", 0,
"Node_4", "Node_12", -0.684508393,
"Node_5", "Node_1", 0,
"Node_5", "Node_2", 0.543840058,
"Node_5", "Node_3", -0.121560909,
"Node_5", "Node_4", -0.175864965,
"Node_5", "Node_5", 0,
"Node_5", "Node_6", 0,
```

```
"Node_5", "Node_7", -0.463780268,
"Node_5", "Node_8", -0.475590735,
"Node_5", "Node_9", 0.375771206,
"Node_5", "Node_10", 0,
"Node_5", "Node_11", 0,
"Node_5", "Node_12", -0.002247814,
"Node_6", "Node_1", 0,
"Node_6", "Node_2", 0,
"Node_6", "Node_3", 0,
"Node_6", "Node_4", -0.564665078,
"Node_6", "Node_5", 0,
"Node_6", "Node_6", 0,
"Node_6", "Node_7", 0,
"Node_6", "Node_8", 0,
"Node_6", "Node_9", 0,
"Node_6", "Node_10", 0.499460219,
"Node_6", "Node_11", 0,
"Node_6", "Node_12", 0,
"Node_7", "Node_1", 0,
"Node_7", "Node_2", 0.294942024,
"Node_7", "Node_3", -0.340622497,
"Node_7", "Node_4", -0.207074673,
"Node_7", "Node_5", 0,
"Node_7", "Node_6", 0,
"Node_7", "Node_7", 0,
"Node_7", "Node_8", 0,
"Node_7", "Node_9", 0,
"Node_7", "Node_10", 0,
"Node_7", "Node_11", -0.222163634,
"Node_7", "Node_12", 0,
"Node_8", "Node_1", 0,
"Node_8", "Node_2", 0,
"Node_8", "Node_3", 0.270231561,
"Node_8", "Node_4", 0,
"Node_8", "Node_5", 0,
"Node_8", "Node_6", 0.301135361,
"Node_8", "Node_7", 0,
"Node_8", "Node_8", 0,
"Node_8", "Node_9", 0.168820177,
"Node_8", "Node_10", 0,
"Node_8", "Node_11", -0.110135317,
"Node_8", "Node_12", 0,
"Node_9", "Node_1", -0.268904503,
"Node_9", "Node_2", 0,
"Node_9", "Node_3", -0.29478998,
"Node_9", "Node_4", 0,
"Node_9", "Node_5", 0,
"Node_9", "Node_6", 0.2979087,
"Node_9", "Node_7", -0.185017143,
"Node_9", "Node_8", 0,
"Node_9", "Node_9", 0,
"Node_9", "Node_10", 0,
"Node_9", "Node_11", 0,
```

```

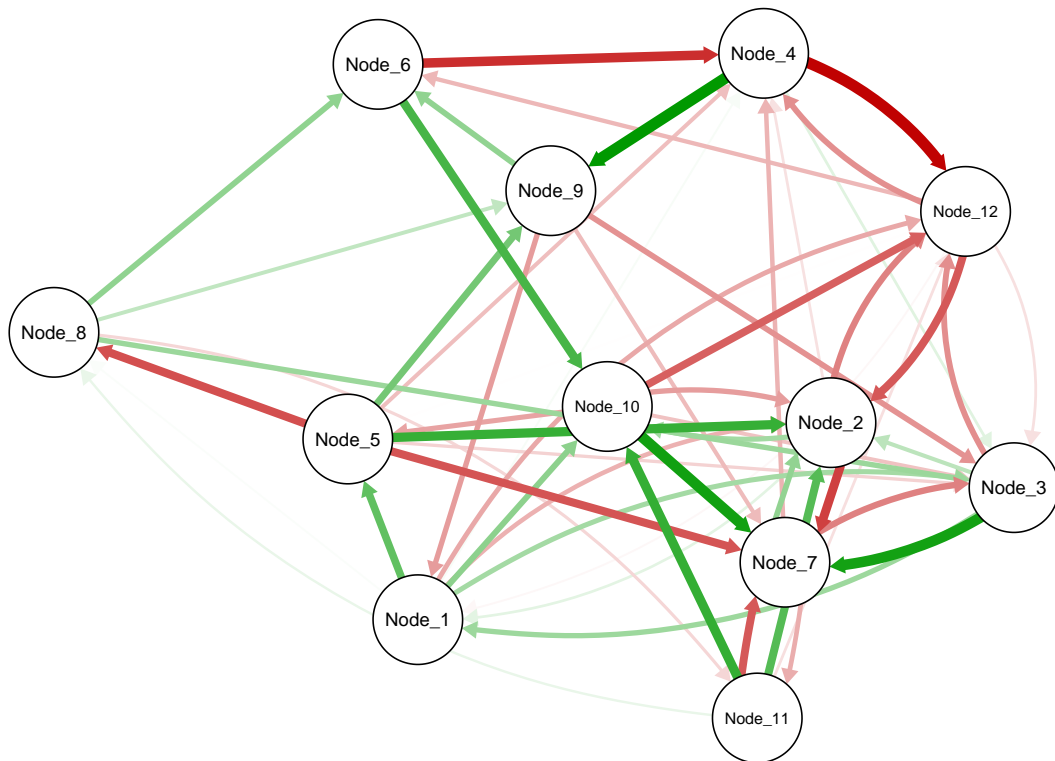
"Node_9", "Node_12",          0,
"Node_10", "Node_1",          0,
"Node_10", "Node_2", -0.265543141,
"Node_10", "Node_3", -0.164960935,
"Node_10", "Node_4",          0,
"Node_10", "Node_5", -0.250023819,
"Node_10", "Node_6",          0,
"Node_10", "Node_7",  0.643813719,
"Node_10", "Node_8",          0,
"Node_10", "Node_9",          0,
"Node_10", "Node_10",         0,
"Node_10", "Node_11",         0,
"Node_10", "Node_12", -0.429969182,
"Node_11", "Node_1",          0,
"Node_11", "Node_2",  0.459833879,
"Node_11", "Node_3",          0,
"Node_11", "Node_4",          0,
"Node_11", "Node_5",          0,
"Node_11", "Node_6",          0,
"Node_11", "Node_7", -0.448953994,
"Node_11", "Node_8",  0.059551164,
"Node_11", "Node_9",          0,
"Node_11", "Node_10",  0.549151382,
"Node_11", "Node_11",         0,
"Node_11", "Node_12", -0.085707379,
"Node_12", "Node_1", -0.021706911,
"Node_12", "Node_2", -0.451544089,
"Node_12", "Node_3", -0.083799623,
"Node_12", "Node_4", -0.302562749,
"Node_12", "Node_5",          0,
"Node_12", "Node_6", -0.193892627,
"Node_12", "Node_7",          0,
"Node_12", "Node_8",          0,
"Node_12", "Node_9",          0,
"Node_12", "Node_10",         0,
"Node_12", "Node_11",         0,
"Node_12", "Node_12",         0
)

## turn nodes into factors

data1 %<>% mutate_at(vars(Source:Target), ~as.factor(.x))

## Create qgraph object with spring layout
q1_spring<- qgraph(data1,layout="spring")

```



Create second

network with the same number of nodes, but random edges and weights.

```
data2<- tibble::tribble(
  ~Source, ~Target, ~Weight,
  "Node_1", "Node_1", 0,
  "Node_1", "Node_2", 0,
  "Node_1", "Node_3", 0,
  "Node_1", "Node_4", 0,
  "Node_1", "Node_5", 0,
  "Node_1", "Node_6", 0,
  "Node_1", "Node_7", 0,
  "Node_1", "Node_8", 0,
  "Node_1", "Node_9", 0,
  "Node_1", "Node_10", 0.239790353,
  "Node_1", "Node_11", 0.218355791,
  "Node_1", "Node_12", 0.521025401,
  "Node_2", "Node_1", 0,
  "Node_2", "Node_2", 0,
  "Node_2", "Node_3", -0.78820502,
  "Node_2", "Node_4", 0,
  "Node_2", "Node_5", 0,
  "Node_2", "Node_6", 0,
  "Node_2", "Node_7", 0,
  "Node_2", "Node_8", 0.223410459,
  "Node_2", "Node_9", 0,
  "Node_2", "Node_10", 0,
  "Node_2", "Node_11", 0.119252125,
  "Node_2", "Node_12", 0,
  "Node_3", "Node_1", 0,
  "Node_3", "Node_2", 0,
```

```
"Node_3", "Node_3", 0,
"Node_3", "Node_4", 0,
"Node_3", "Node_5", 0,
"Node_3", "Node_6", 0,
"Node_3", "Node_7", 0,
"Node_3", "Node_8", -0.029698157,
"Node_3", "Node_9", 0,
"Node_3", "Node_10", 0,
"Node_3", "Node_11", 0.097025854,
"Node_3", "Node_12", -0.215934924,
"Node_4", "Node_1", 0,
"Node_4", "Node_2", -0.139744398,
"Node_4", "Node_3", -0.146255431,
"Node_4", "Node_4", 0,
"Node_4", "Node_5", 0.446772118,
"Node_4", "Node_6", 0.325309673,
"Node_4", "Node_7", -0.217882354,
"Node_4", "Node_8", 0,
"Node_4", "Node_9", 0.38677796,
"Node_4", "Node_10", 0.507699655,
"Node_4", "Node_11", 0.489642845,
"Node_4", "Node_12", 0.256835078,
"Node_5", "Node_1", 0,
"Node_5", "Node_2", -0.551449712,
"Node_5", "Node_3", 0,
"Node_5", "Node_4", 0,
"Node_5", "Node_5", 0,
"Node_5", "Node_6", -0.323225734,
"Node_5", "Node_7", -0.442932039,
"Node_5", "Node_8", -0.035572494,
"Node_5", "Node_9", 0,
"Node_5", "Node_10", 0,
"Node_5", "Node_11", 0,
"Node_5", "Node_12", 0,
"Node_6", "Node_1", 0,
"Node_6", "Node_2", 0,
"Node_6", "Node_3", 0,
"Node_6", "Node_4", 0,
"Node_6", "Node_5", 0,
"Node_6", "Node_6", 0,
"Node_6", "Node_7", -0.307602418,
"Node_6", "Node_8", 0,
"Node_6", "Node_9", 0,
"Node_6", "Node_10", 0,
"Node_6", "Node_11", -1.152466944,
"Node_6", "Node_12", 0.179426393,
"Node_7", "Node_1", 0,
"Node_7", "Node_2", 0,
"Node_7", "Node_3", 0,
"Node_7", "Node_4", -0.380578994,
"Node_7", "Node_5", 0,
"Node_7", "Node_6", 0,
"Node_7", "Node_7", 0,
```

```
"Node_7", "Node_8", 0,
"Node_7", "Node_9", -0.164635895,
"Node_7", "Node_10", -0.689027156,
"Node_7", "Node_11", 0,
"Node_7", "Node_12", -0.088476796,
"Node_8", "Node_1", 0.179650535,
"Node_8", "Node_2", 0.747478465,
"Node_8", "Node_3", 0.226601162,
"Node_8", "Node_4", 0.653520634,
"Node_8", "Node_5", -0.268224203,
"Node_8", "Node_6", 0.550722155,
"Node_8", "Node_7", 0,
"Node_8", "Node_8", 0,
"Node_8", "Node_9", 0.553526399,
"Node_8", "Node_10", -0.217222506,
"Node_8", "Node_11", -0.35334156,
"Node_8", "Node_12", -0.503552256,
"Node_9", "Node_1", 0.316299526,
"Node_9", "Node_2", 0,
"Node_9", "Node_3", -0.082455343,
"Node_9", "Node_4", -0.317537574,
"Node_9", "Node_5", -0.05285038,
"Node_9", "Node_6", 0.299459209,
"Node_9", "Node_7", 0,
"Node_9", "Node_8", -0.346076585,
"Node_9", "Node_9", 0,
"Node_9", "Node_10", 0,
"Node_9", "Node_11", 0,
"Node_9", "Node_12", 0,
"Node_10", "Node_1", 0,
"Node_10", "Node_2", -0.142912355,
"Node_10", "Node_3", 0,
"Node_10", "Node_4", 0,
"Node_10", "Node_5", 0,
"Node_10", "Node_6", 0,
"Node_10", "Node_7", -0.358022017,
"Node_10", "Node_8", 0.148487876,
"Node_10", "Node_9", 0.150659375,
"Node_10", "Node_10", 0,
"Node_10", "Node_11", -0.71075785,
"Node_10", "Node_12", 0,
"Node_11", "Node_1", 0.690094563,
"Node_11", "Node_2", -0.19767747,
"Node_11", "Node_3", 0.426450588,
"Node_11", "Node_4", 0.373505703,
"Node_11", "Node_5", 0,
"Node_11", "Node_6", 0,
"Node_11", "Node_7", 0.173924444,
"Node_11", "Node_8", 0,
"Node_11", "Node_9", 0,
"Node_11", "Node_10", 0.226449072,
"Node_11", "Node_11", 0,
"Node_11", "Node_12", 0,
```

```

"Node_12", "Node_1", -0.347366611,
"Node_12", "Node_2", 0.139538891,
"Node_12", "Node_3", 0,
"Node_12", "Node_4", 0,
"Node_12", "Node_5", 0,
"Node_12", "Node_6", 0.182930175,
"Node_12", "Node_7", 0,
"Node_12", "Node_8", -0.532286002,
"Node_12", "Node_9", -0.045497183,
"Node_12", "Node_10", 0,
"Node_12", "Node_11", 0.123527054,
"Node_12", "Node_12", 0
)

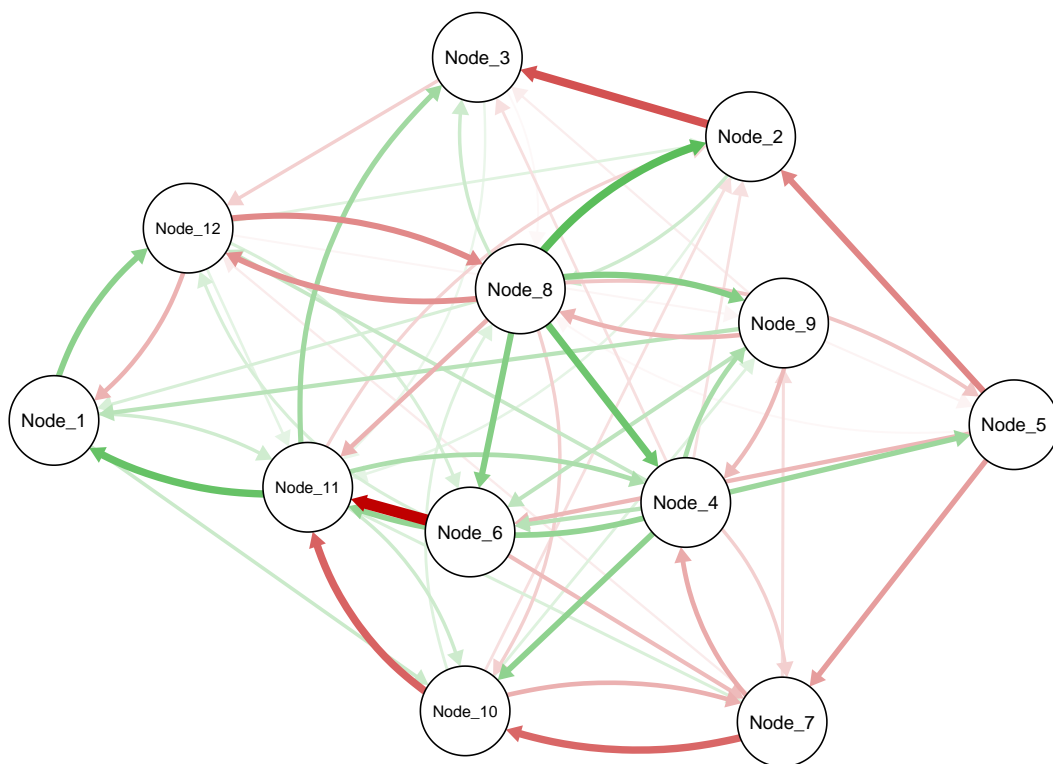
```

```
## turn nodes into factors
```

```
data2 %<>% mutate_at(vars(Source:Target), ~as.factor(.x))
```

```
## save the spring layout
```

```
q2_spring<-qgraph(data2,layout="spring")
```



Before

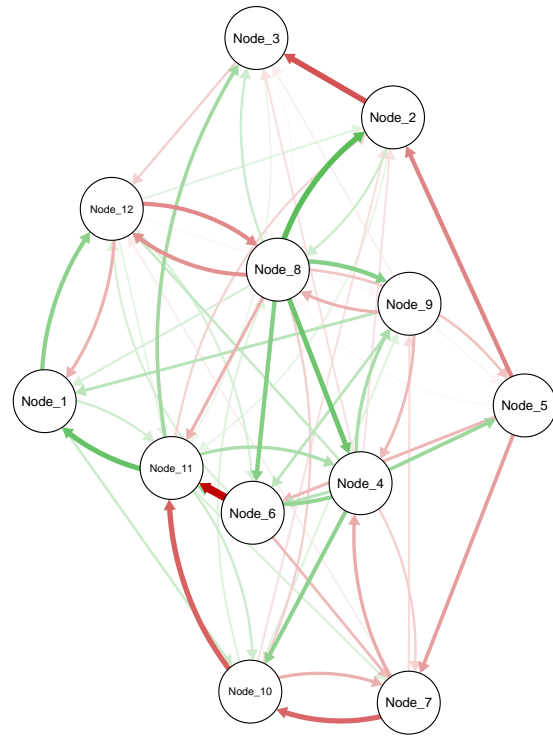
In this instance, each graph is generated using the FR algorithm resulting in two separate layouts.

```

par(mfrow=c(1,2))
qgraph(q1_spring)
qgraph(q2_spring)

```



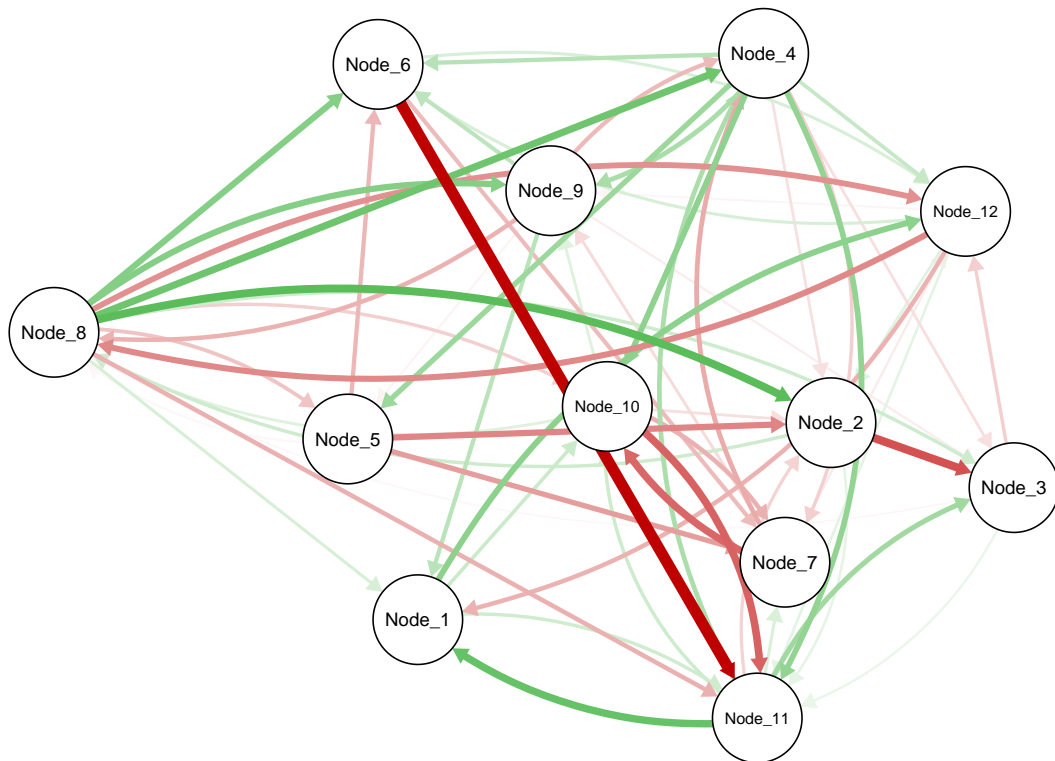


After

What you need to do is extract the layout from the first network, and assign it as the layout for the second graph. This way the second graph is not being algorithmically generated.

```
## save layout from first plot
q1_lay<- q1_spring$layout

## assign layout to second plot
q3<- qgraph(data2,layout = q1_lay)
```



```

par(mfrow=c(1,2))
qgraph(q1_spring)
qgraph(q3)

```

