igraph_data_scientists

Cathy Atkinson

28.08.16

Demo code for the igraph package for network data visualisation using the academic paths of a sample of data scientist colleagues.

Entering the data

A dataframe is created where each row is an edge; the two columns are the vertices the edge runs between. The first vertex in the path is an individual datascientist; the other vertices are qualifications. The last vertex in the path for each person is "Data Scientist".

u- is an undergraduate qualification

p- is a postgraduate qualification

dr- is a PhD

```
head(ds_quals)
```

```
##      [,1]             [,2]
## [1,] "A"              "u-geography"
## [2,] "u-geography"    "p-data science"
## [3,] "p-data science" "DATA \nSCIENTIST"
## [4,] "B"              "u-maths/stats"
## [5,] "u-maths/stats"  "dr-physics"
## [6,] "dr-physics"     "DATA \nSCIENTIST"
```

The igraph package

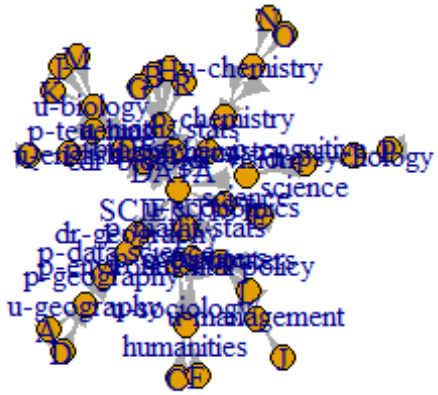
The igraph package can create the graph from data in a range of formats. In this example the igraph package creates the network from the edgelist:

```
ds_net <- graph_from_edgelist(ds_quals, directed=TRUE)
```

In this example the graph is 'directed'; each edge runs from the vertex in the left-hand column of the dataframe to the vertex in the right-hand column of the dataframe.

Basic plot

```
plot(ds_net)
```

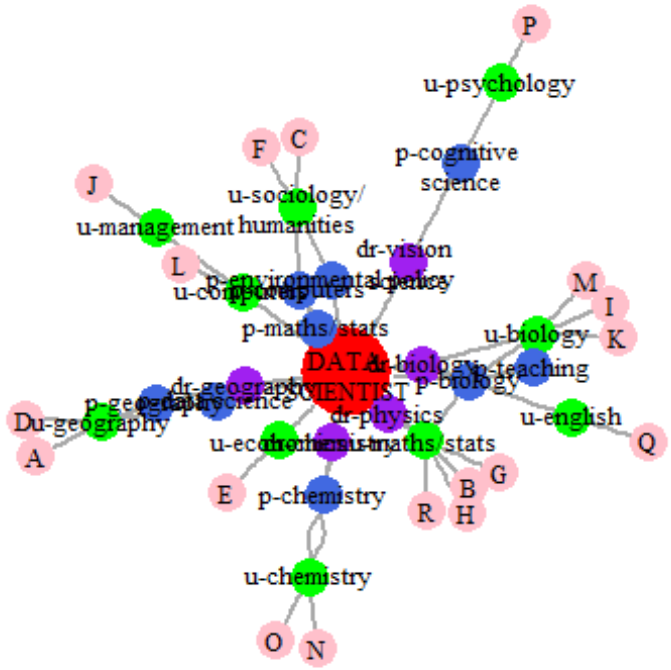


Improve the appearance of the plot

The appearance of the plot can be improved by specifying parameters within the plot command.

For example, use a better layout (Fruchterman-Reingold), sort out the colours, remove the arrows, reduce the margins around the plot etc.

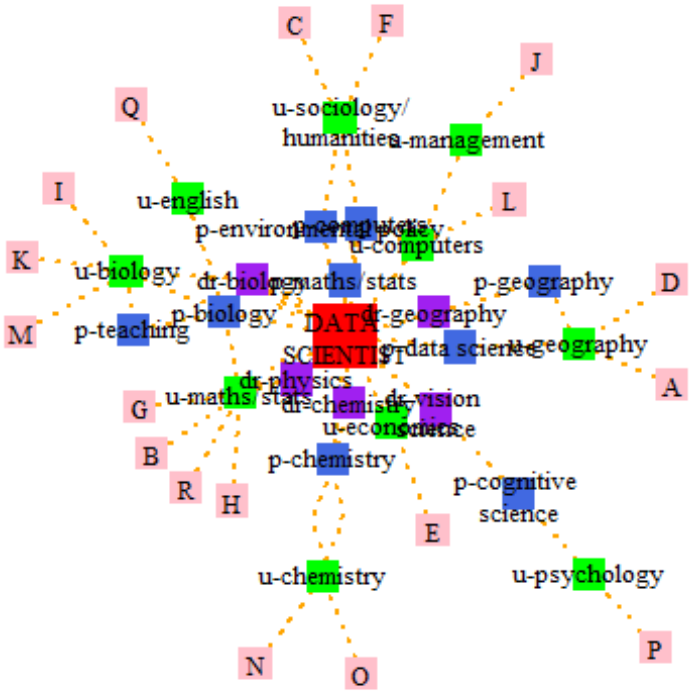
```
par(mar=c(0, 0, 0, 0))
plot(ds_net,
      layout=layout_with_fr,
      vertex.size=ifelse(V(ds_net)$name=="DATA \nSCIENTIST", 30, 12),
      vertex.color=ifelse(V(ds_net)$name=="DATA \nSCIENTIST", "red",
                          ifelse(str_sub(V(ds_net)$name, 1, 3) == "dr-",
"purple",
                                 ifelse(str_sub(V(ds_net)$name, 1, 2) == "p-
", "royalblue",
                                        ifelse(str_sub(V(ds_net)$name, 1, 2)
== "u-", "green", "pink")))),
      vertex.frame.color=NA,
      vertex.label.color="black",
      vertex.label.cex=0.8,
      edge.color="darkgrey",
      edge.width=2,
      edge.arrow.size=0
      )
```



Alternative plot

This is another example of a revised plot. A different layout algorithm is used, and the shape of the vertices and style of the edges is changed.

```
par(mar=c(0, 0, 0, 0))
plot(ds_net,
      layout=layout_components,
      vertex.shape="square",
      vertex.size=ifelse(V(ds_net)$name=="DATA \nSCIENTIST", 20, 10),
      vertex.color=ifelse(V(ds_net)$name=="DATA \nSCIENTIST", "red",
                          ifelse(str_sub(V(ds_net)$name, 1, 3) == "dr-",
"purple",
                                 ifelse(str_sub(V(ds_net)$name, 1, 2) == "p-
", "royalblue",
                                        ifelse(str_sub(V(ds_net)$name, 1, 2)
== "u-", "green", "pink")))),
      vertex.frame.color=NA,
      vertex.label.color="black",
      vertex.label.cex=0.8,
      edge.color="orange",
      edge.width=2,
      edge.arrow.size=0,
      edge.lty=3
      )
```



Arrange the layout by hand

If none of the standard igraph plot layouts is quite how you would like it to look you can use the interative plot, tkplot:

```
tkplot(ds_net)
```

In the tkplot windown that opens you can arrange the nodes freely by simply clickly and dragging the ones you want to move to new positions.

You can then capture the coordinates of the vertices in your preferred layout using:

```
coord <- tk_coords(1)
```

You can then use these coordinates in the standard plot and incorporate all the other aesthetic changes (as above)

```
plot(ds_net,
     layout=coord)
```

NB. If you want to use the coordinates in a future R session it is worth saving them.

```
write.csv(coord, "coord.csv", row.names=FALSE)
```

Analyse the network

The nature of this demo data doesn't lend itself to a lot of analysis of the graph. However, you can find the maximum number of edges linked to a node (aka degree)

```
max(degree(ds_net))
```

```
## [1] 18
```

Get the name of the node with the highest degree

```
V(ds_net)$name[degree(ds_net)==max(degree(ds_net))]
```

```
## [1] "DATA \nSCIENTIST"
```

Find properties of an individual vertex. E.g. how many edges connect to the vertex for undergraduate Geography?

```
degree(ds_net)[V(ds_net)$name=="u-geography"]
```

```
## u-geography
##           4
```

However, this figure is both edges going into and out of the vertex. In this case to find the number of people who had studied undergraduate Geography the measure needed is the 'in' degree:

```
degree(ds_net, mode="in")[V(ds_net)$name=="u-geography"]
```

```
## u-geography
##           2
```

For more complex graphs analysis could include measures of vertex centrality, shortest paths, degree distribution, community detection and many pther functions igraph includes.