MC1

Ng Jin Yao

2025-05-14

|  |
| --- |
| Important |
| You are required to install the R packages above, if necessary, before continue to the next step Check packages to ensure they are installed |

pacman::p\_load(tidyverse, jsonlite,   
 SmartEDA, tidygraph,   
 ggraph)  
  
#tidygraph to build graph object, to work tgt with ggraph  
#ggplot no need to call out again as its in tidyverse  
  
#SmartEDA library for exploratory data analysis, good to have it to check data

## Importing Knowledge Graph Data

kg <- fromJSON("data/MC1\_graph.json")

### Inspect Structure

str(kg,max.level = 1)

List of 5  
 $ directed : logi TRUE  
 $ multigraph: logi TRUE  
 $ graph :List of 2  
 $ nodes :'data.frame': 17412 obs. of 10 variables:  
 $ links :'data.frame': 37857 obs. of 4 variables:

#structure enable to check structure of the data  
#good to have this line to check structues of data

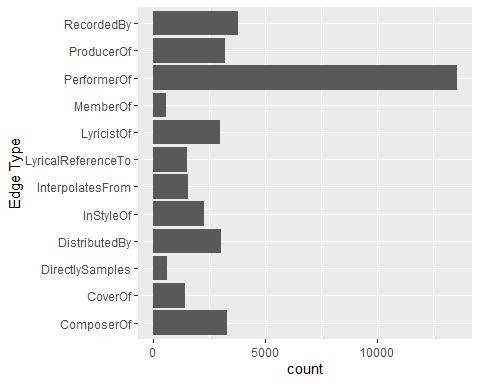
|  |
| --- |
| Note |
| directed - tells you if its a directed graph  nodes and links are in dataframe- can be read  Can see data types etc from Environment tab of R-studio - look at the structure    Can open and see the data table |

### Extract and Inspect

nodes\_tbl <- as\_tibble(kg$nodes) #kg$ know that pull nodes out  
edges\_tbl <- as\_tibble(kg$links)

## Initial EDA

ggplot(data = edges\_tbl,  
 aes(y= `Edge Type`)) +  
 geom\_bar()



##want to know how many classes available for edge type  
## field names always use ``, not ''

ggplot

function (data = NULL, mapping = aes(), ..., environment = parent.frame())   
{  
 UseMethod("ggplot")  
}  
<bytecode: 0x00000165175900b0>  
<environment: namespace:ggplot2>

## Creating Knowledge Graph

This is

### Step 1: Mappying from node id to row index

avoid multiple id that have 0 0 but actually they have missing value

to make sure they have unique id code

To reassign ID

depends on scenario, may need to subset data first before extracting - depends on use case

id\_map <- tibble(id = nodes\_tbl$id, index = seq\_len(nrow(nodes\_tbl)))

### Step 2: Map source and target IDs to row indices

edges\_tbl <- edges\_tbl %>%  
 left\_join(id\_map,by = c("source" = "id"))%>%  
 rename(from =index) %>%  
 left\_join(id\_map, by = c("target" = "id"))%>%  
 rename(to = index)  
## see data table original source is 0 now become from 1

### Step 3: Filter out any unmatched (invalid) edges

edges\_tbl <- edges\_tbl %>%  
 filter(!is.na(from), !is.na(to))

### Step 4: Creating the graph

graph <- tbl\_graph(nodes = nodes\_tbl,  
 edges = edges\_tbl,  
 directed = kg$directed)  
#if its directed, true, if not false. BUt in this case there is data to show its directed = true so can reference the data from the data

## Visualising the knowledge graph

set.seed(1234)  
#once fix this, everytime run random is the same number

### Visualising the Whole Graph

ggraph(graph, layout = "fr") +  
 geom\_edge\_link(alpha = 0.3,  
 colour = "gray") +  
 geom\_node\_point(aes(color = `Node Type`),  
 size = 4) +  
 geom\_node\_text(aes(label = name),  
 repel = TRUE,  
 size =2.5) +  
 theme\_void()

#### Step 1:Filter edges to only “Memberof”

mining for those “Memberof”

graph\_memberof <- graph %>%  
 activate(edges) %>%  
 filter (`Edge Type` == "MemberOf")  
  
##filter off the edges but the nodes are still there, still a large set of data 17412 as the source

#### Step 2: Extract only connected nodes (ie used in these edges)

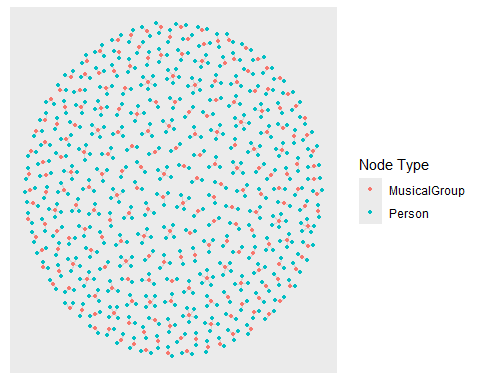
##additional code to remove and only used connected nodes, those that are useful  
used\_nodes\_indices <- graph\_memberof %>%  
 activate(edges)%>%  
 as\_tibble() %>%  
 select(from, to) %>%  
 unlist() %>%  
 unique()

#### Step 3: keep only those node

graph\_memberof <- graph\_memberof %>%  
 activate(nodes) %>%  
 mutate(row\_id = row\_number()) %>%  
 filter(row\_id %in% used\_nodes\_indices) %>%  
 select(-row\_id) # optional cleanup

#### plot the sub graph

ggraph(graph\_memberof,  
 layout = "fr") +  
 geom\_edge\_link(alpha= 0.5,  
 colour = "gray") +  
 geom\_node\_point(aes(color = `Node Type`),  
 size = 1)+  
 geom\_node\_text(aes(label = name),  
 repel = TRUE,  
 size=2.5)



theme\_void()

List of 136  
 $ line : list()  
 ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
 $ rect : list()  
 ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
 $ text :List of 11  
 ..$ family : chr ""  
 ..$ face : chr "plain"  
 ..$ colour : chr "black"  
 ..$ size : num 11  
 ..$ hjust : num 0.5  
 ..$ vjust : num 0.5  
 ..$ angle : num 0  
 ..$ lineheight : num 0.9  
 ..$ margin : 'margin' num [1:4] 0points 0points 0points 0points  
 .. ..- attr(\*, "unit")= int 8  
 ..$ debug : logi FALSE  
 ..$ inherit.blank: logi TRUE  
 ..- attr(\*, "class")= chr [1:2] "element\_text" "element"  
 $ title : NULL  
 $ aspect.ratio : NULL  
 $ axis.title : list()  
 ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
 $ axis.title.x : NULL  
 $ axis.title.x.top : NULL  
 $ axis.title.x.bottom : NULL  
 $ axis.title.y : NULL  
 $ axis.title.y.left : NULL  
 $ axis.title.y.right : NULL  
 $ axis.text : list()  
 ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
 $ axis.text.x : NULL  
 $ axis.text.x.top : NULL  
 $ axis.text.x.bottom : NULL  
 $ axis.text.y : NULL  
 $ axis.text.y.left : NULL  
 $ axis.text.y.right : NULL  
 $ axis.text.theta : NULL  
 $ axis.text.r : NULL  
 $ axis.ticks : NULL  
 $ axis.ticks.x : NULL  
 $ axis.ticks.x.top : NULL  
 $ axis.ticks.x.bottom : NULL  
 $ axis.ticks.y : NULL  
 $ axis.ticks.y.left : NULL  
 $ axis.ticks.y.right : NULL  
 $ axis.ticks.theta : NULL  
 $ axis.ticks.r : NULL  
 $ axis.minor.ticks.x.top : NULL  
 $ axis.minor.ticks.x.bottom : NULL  
 $ axis.minor.ticks.y.left : NULL  
 $ axis.minor.ticks.y.right : NULL  
 $ axis.minor.ticks.theta : NULL  
 $ axis.minor.ticks.r : NULL  
 $ axis.ticks.length : 'simpleUnit' num 0points  
 ..- attr(\*, "unit")= int 8  
 $ axis.ticks.length.x : NULL  
 $ axis.ticks.length.x.top : NULL  
 $ axis.ticks.length.x.bottom : NULL  
 $ axis.ticks.length.y : NULL  
 $ axis.ticks.length.y.left : NULL  
 $ axis.ticks.length.y.right : NULL  
 $ axis.ticks.length.theta : NULL  
 $ axis.ticks.length.r : NULL  
 $ axis.minor.ticks.length : 'simpleUnit' num 0points  
 ..- attr(\*, "unit")= int 8  
 $ axis.minor.ticks.length.x : NULL  
 $ axis.minor.ticks.length.x.top : NULL  
 $ axis.minor.ticks.length.x.bottom: NULL  
 $ axis.minor.ticks.length.y : NULL  
 $ axis.minor.ticks.length.y.left : NULL  
 $ axis.minor.ticks.length.y.right : NULL  
 $ axis.minor.ticks.length.theta : NULL  
 $ axis.minor.ticks.length.r : NULL  
 $ axis.line : NULL  
 $ axis.line.x : NULL  
 $ axis.line.x.top : NULL  
 $ axis.line.x.bottom : NULL  
 $ axis.line.y : NULL  
 $ axis.line.y.left : NULL  
 $ axis.line.y.right : NULL  
 $ axis.line.theta : NULL  
 $ axis.line.r : NULL  
 $ legend.background : NULL  
 $ legend.margin : NULL  
 $ legend.spacing : NULL  
 $ legend.spacing.x : NULL  
 $ legend.spacing.y : NULL  
 $ legend.key : NULL  
 $ legend.key.size : 'simpleUnit' num 1.2lines  
 ..- attr(\*, "unit")= int 3  
 $ legend.key.height : NULL  
 $ legend.key.width : NULL  
 $ legend.key.spacing : 'simpleUnit' num 5.5points  
 ..- attr(\*, "unit")= int 8  
 $ legend.key.spacing.x : NULL  
 $ legend.key.spacing.y : NULL  
 $ legend.frame : NULL  
 $ legend.ticks : NULL  
 $ legend.ticks.length : 'rel' num 0.2  
 $ legend.axis.line : NULL  
 $ legend.text :List of 11  
 ..$ family : NULL  
 ..$ face : NULL  
 ..$ colour : NULL  
 ..$ size : 'rel' num 0.8  
 ..$ hjust : NULL  
 ..$ vjust : NULL  
 ..$ angle : NULL  
 ..$ lineheight : NULL  
 ..$ margin : NULL  
 ..$ debug : NULL  
 ..$ inherit.blank: logi TRUE  
 ..- attr(\*, "class")= chr [1:2] "element\_text" "element"  
 $ legend.text.position : NULL  
 $ legend.title :List of 11  
 ..$ family : NULL  
 ..$ face : NULL  
 ..$ colour : NULL  
 ..$ size : NULL  
 ..$ hjust : num 0  
 ..$ vjust : NULL  
 ..$ angle : NULL  
 ..$ lineheight : NULL  
 ..$ margin : NULL  
 ..$ debug : NULL  
 ..$ inherit.blank: logi TRUE  
 ..- attr(\*, "class")= chr [1:2] "element\_text" "element"  
 $ legend.title.position : NULL  
 $ legend.position : chr "right"  
 $ legend.position.inside : NULL  
 $ legend.direction : NULL  
 $ legend.byrow : NULL  
 $ legend.justification : NULL  
 $ legend.justification.top : NULL  
 $ legend.justification.bottom : NULL  
 $ legend.justification.left : NULL  
 $ legend.justification.right : NULL  
 $ legend.justification.inside : NULL  
 $ legend.location : NULL  
 $ legend.box : NULL  
 $ legend.box.just : NULL  
 $ legend.box.margin : NULL  
 $ legend.box.background : NULL  
 $ legend.box.spacing : NULL  
 [list output truncated]  
 - attr(\*, "class")= chr [1:2] "theme" "gg"  
 - attr(\*, "complete")= logi TRUE  
 - attr(\*, "validate")= logi TRUE