Playing around with the Inside Out dataset

Adhavan Mohana Sivaraj

Contents

What are my intentions?

- To create a heat map of sorts of conversations that take place in the movie Inside Out.
- What will this allow me to do?
  - - To comment on the frequency/duration of certain interactions inside and outside Riley's mind.
  - - The thought then arises that Joy is obviously the most frequently portrayed character. So why do it at all?

Beyond simply visualising the heatmap of interactions is to pick on character tells.

For example: My friends tell me that Anger is stereotypically portrayed as the dominant character in the Dad's mind. And mother has her own stereotypical dominant emotion. Is this true? What other characteristic tells exist that the creators have weaved into the story wittingly or unwittingly?

Data collection

```
InsideOutEdges <- read.csv("./data/edges_emotions.csv",na.strings=c("","NA")) %>% drop_na(Target) %>% di
InsideOutNodes <- read.csv("./data/nodes_emotions.csv")

#head(InsideOutNodes,2)
#head(InsideOutEdges,2)
```

Find the Google Sheets spreadsheet here - The dataset is incomplete. It is only about 20 minutes into the movie.

Creating categories for types of interactions

- Interactions can be categorised at first sight as simply being directed and not directed
- If directed, there is a clear enough target. Does Joy address Sadness directly? Does Dad talk to Riley?
- If not directed, who is the intended recipient? That is where categorising becomes muddled. I've gotten over it temporarily by categorising any un-directed statements made by the emotions in the presence of all the emotions as 'to all emotions'. Should I create individual fields for each emotion as the target instead of saying 'to all emotions'?

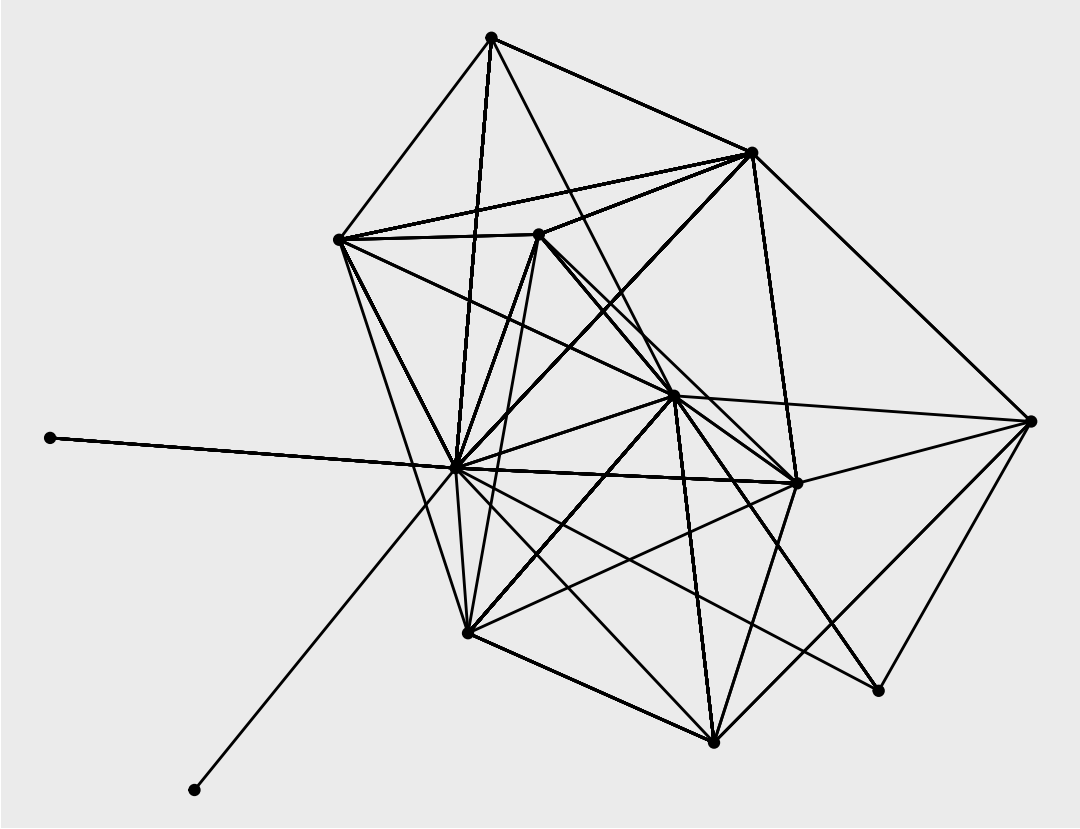
Workflow

- I initially started by using the Closed Captions file to determine and categorise individual dialogue. Learned within a week that it was painstakingly slow.
- Was able to find the transcript later, but transcripts don't come with the time codes and so filling in the duration of each duration is going to be tough.
- I then removed the blank lines in the transcript and was able to automate filling in the source fields by using a RegEx pattern in excel
- I wish there was a way to fill in the targets for directed statements, but no. It will have to be manually entered in.
- Is there a better way to go about this?

Creating the graph

```
#i<-tbl_graph(nodes=InsideOutNodes, edges = InsideOutEdges, directed = TRUE)
i<-graph_from_data_frame(InsideOutEdges, vertices =InsideOutNodes, directed = T) %>% as_tbl_graph()
```

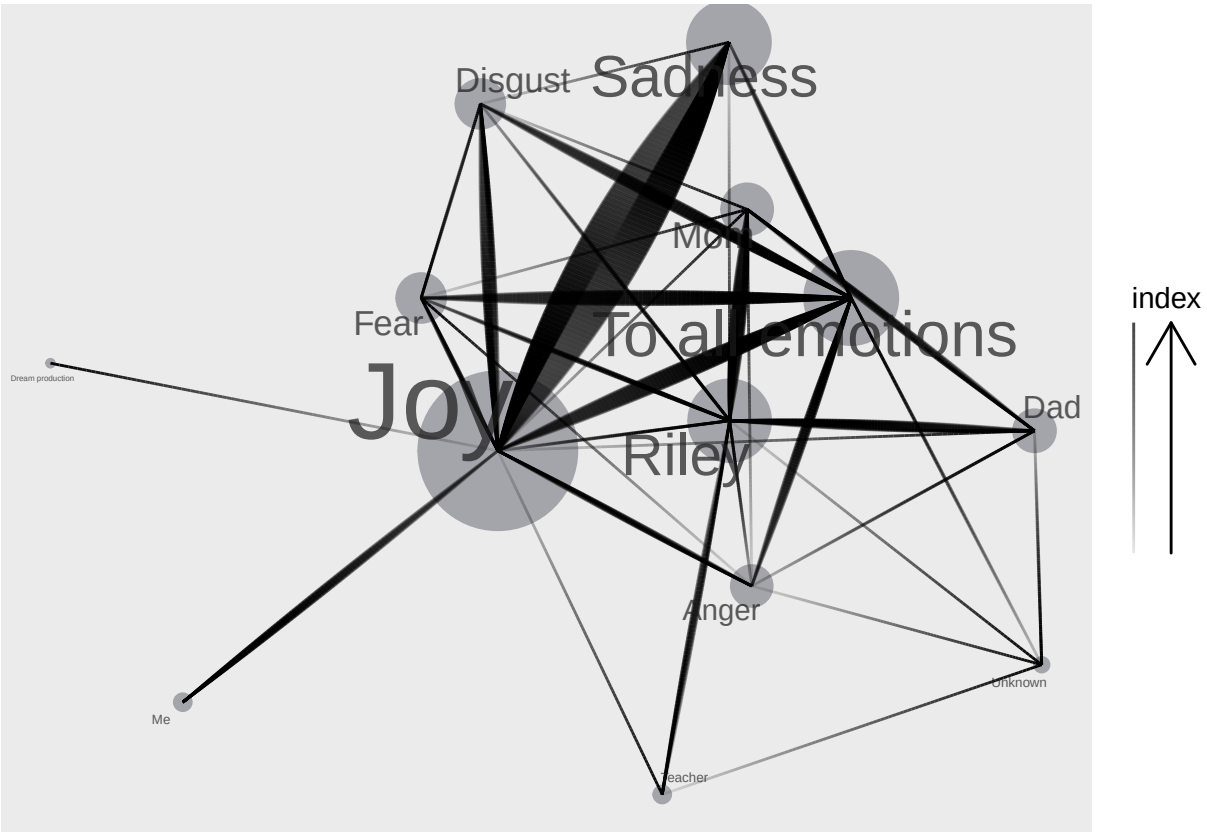
```
autograph(i)
```



Visualising

- The plot is too dense at this point to arrive at any sort of conclusion. And the dataset is incomplete! What happens when the dataset is complete? Will a circle layout help?

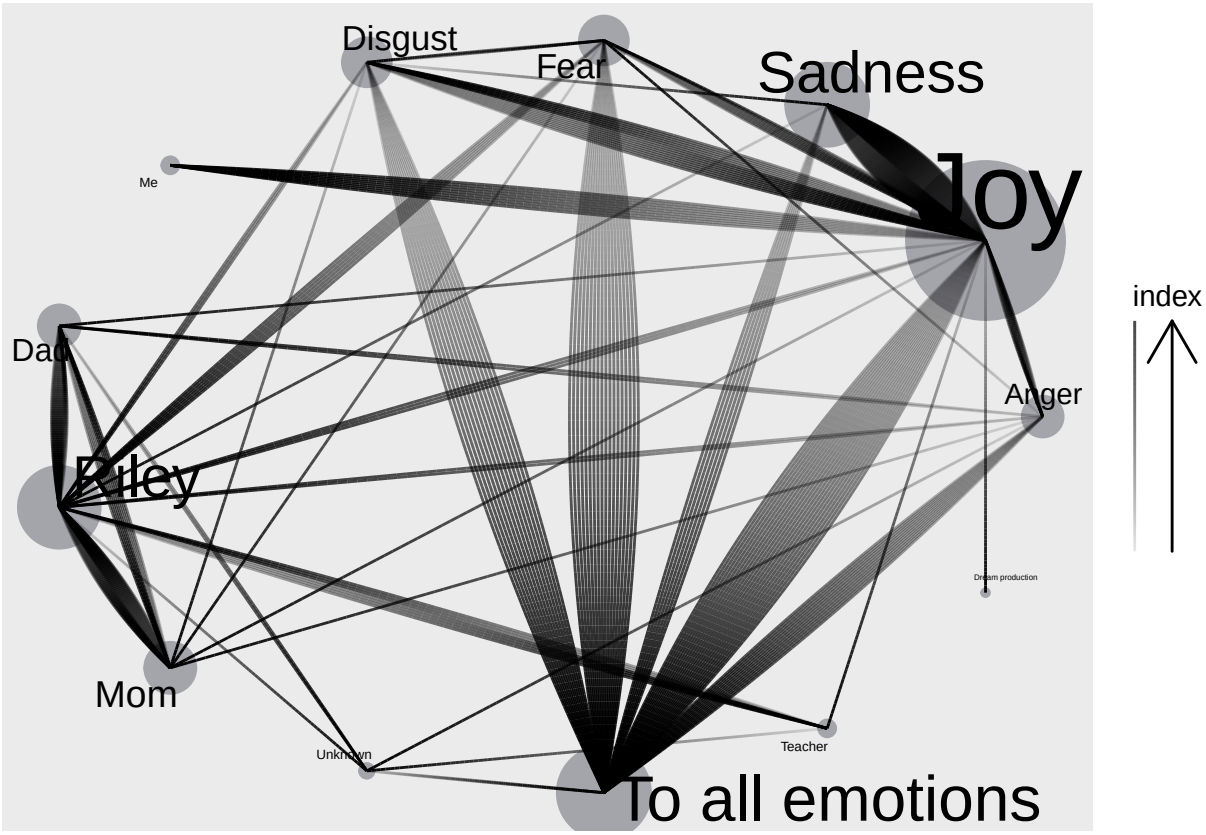
```
j<-ggraph(graph=i, layout="kk") +
  geom_node_point(size = 1 + 0.2*degree(i), color = rgb(0,0,.1,.3)) +

  # geom_edge_density(n=3) +

  geom_edge_fan(aes(alpha = stat(index)), strength=.5) +

  guides(edge_alpha = guide_edge_direction()) +

  geom_node_text(aes(label = name), size = 1 + 0.1*degree(i), color=rgb(.1,.1,.1,.7), repel = T)
j
```




```
# what is stat? refer https://nkha149.github.io/stat385-sp2020/files/notes/html/graphics-p3.html
# https://stackoverflow.com/questions/38775661/what-is-the-difference-between-geoms-and-stats-in-ggplot
# Each point along the line has a numeric value associated with it giving theposition along the path, a
```

```
j<-ggraph(graph=i, layout="circle") +

  geom_node_point(size = 1 + 0.2*degree(i), color =rgb(0,0,.1,.3)) +
```

```
  # geom_edge_density(n=3)+
  geom_edge_fan(color="black", aes(alpha = stat(index)), strength=.95) +
  guides(edge_alpha = guide_edge_direction()) +
  geom_node_text(aes(label = name), size = 1 + 0.1*degree(i), color="black", repel = T)
j
```



Play and Invent

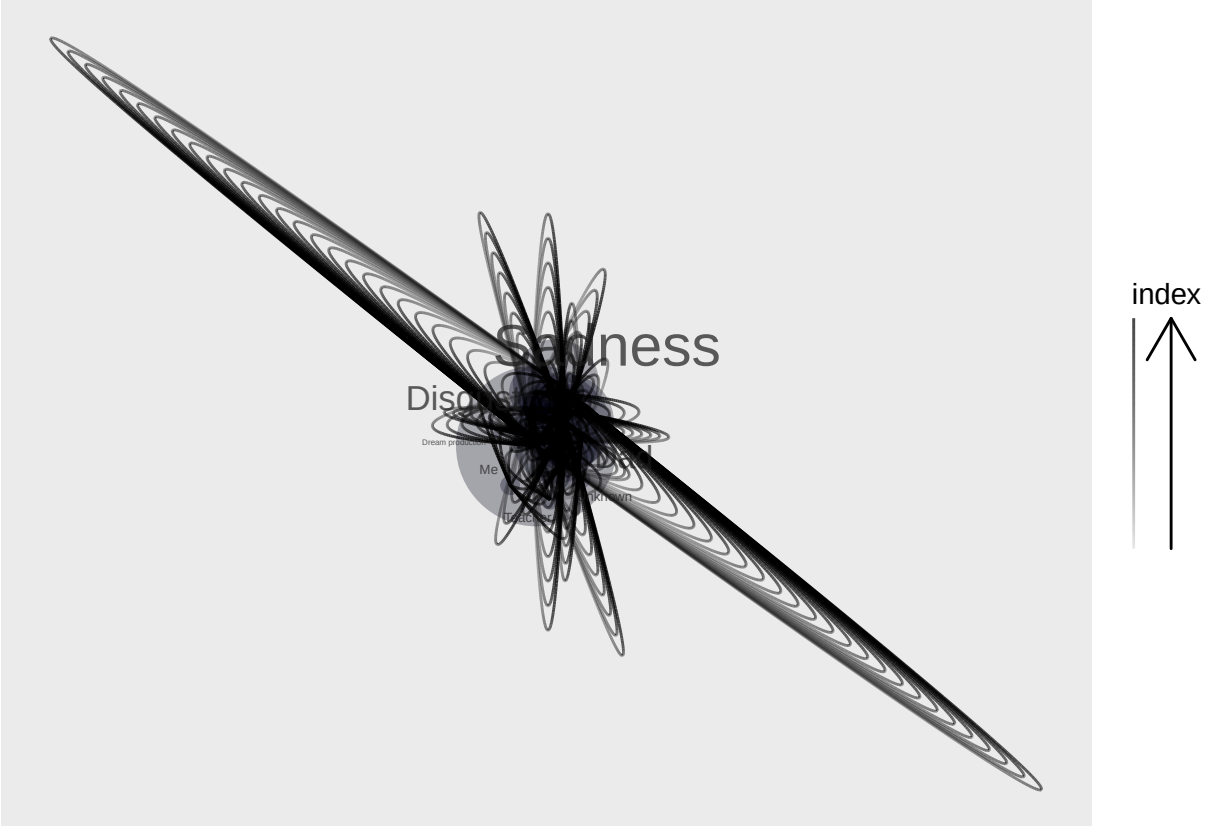
The absurd

The following contain the same data with exaggerated parameters to make the data intentionally unreadable and abstract as an exploratory exercise for my Play and Invent unit S2C2.

Read more from this pdf click on me

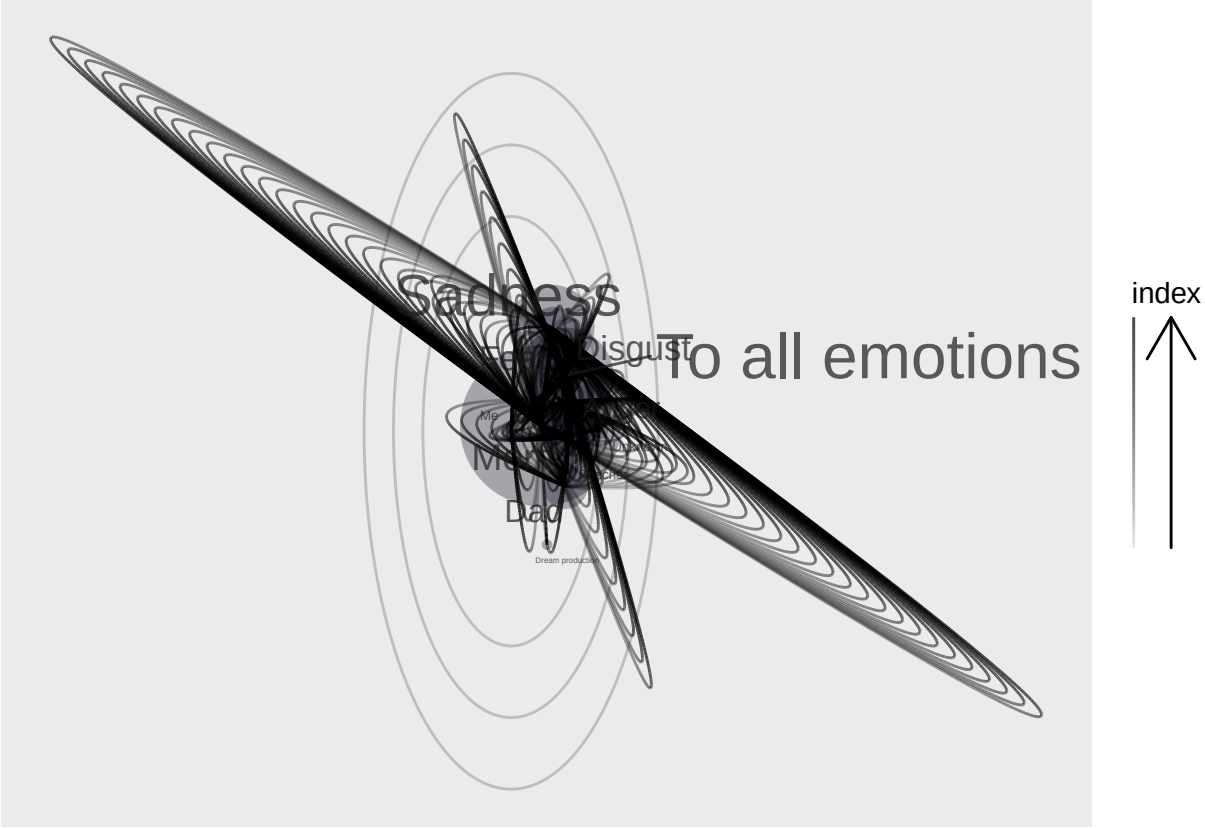
```
j<-ggraph(graph=i, layout="kk")+

  geom_node_point(size = 1 + 0.2*degree(i), color = rgb(0,0,.1,.3)) +

  # geom_edge_density(n=3) +

  geom_edge_fan(aes(alpha = stat(index)), strength=100, linejoin = "mitre") +

  guides(edge_alpha = guide_edge_direction()) +

  geom_node_text(aes(label = name), size = 1 + 0.1*degree(i), color=rgb(.1,.1,.1,.7), repel = T)

j
```

```
## Warning: ggrepel: 6 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



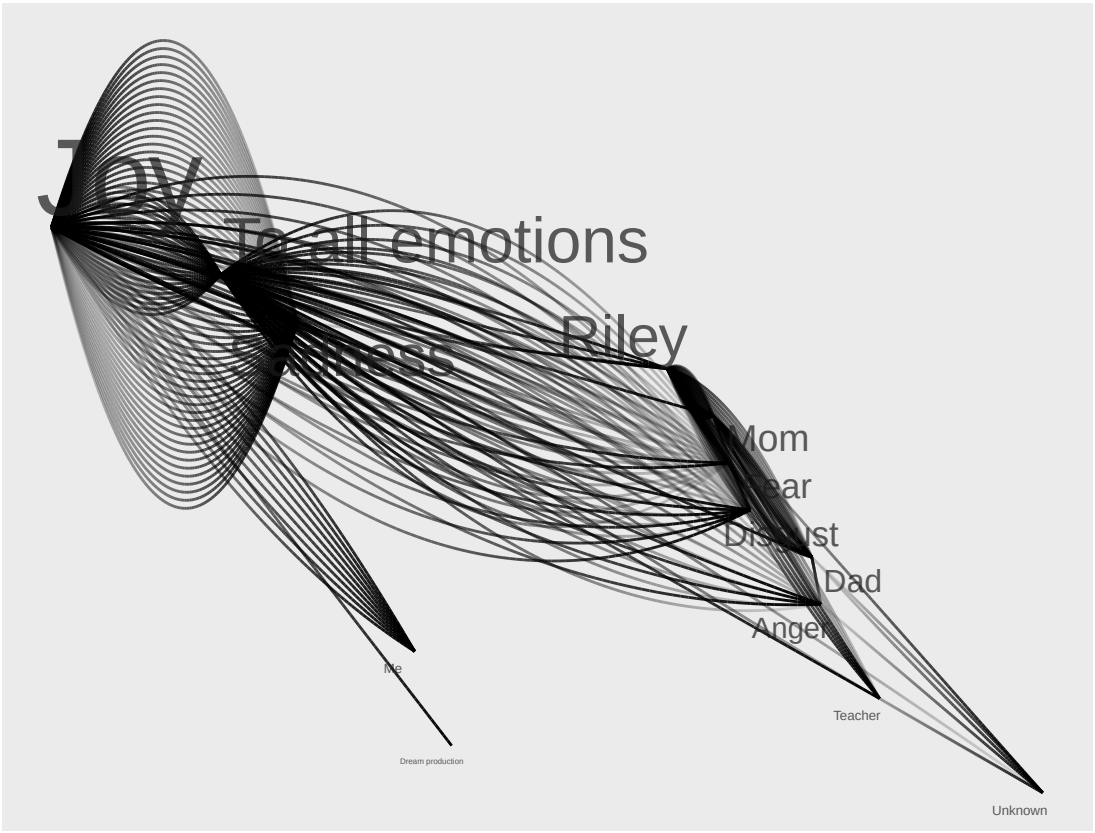
```
j<-ggraph(graph=i,'focus', focus = node_is_center()) +
  ggforce::geom_circle(aes(x0 = 0, y0 = 0, r = r), data.frame(r = 3:5), colour = 'grey') +

  geom_node_point(size = 1 + 0.2*degree(i), color = rgb(0,0,.1,.3)) +

  # geom_edge_density(n=3) +

  geom_edge_fan(aes(alpha = stat(index)), strength=100, linejoin = "mitre") +

  guides(edge_alpha = guide_edge_direction()) +

  geom_node_text(aes(label = name), size = 1 + 0.1*degree(i), color=rgb(.1,.1,.1,.7), repel = T)

j
```

```
## Warning: ggrepel: 2 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
j<-ggraph(graph=i, 'fabric', sort.by = node_rank_fabric()) +

  # geom_edge_density(n=3) +

  geom_edge_fan(aes(alpha = stat(index)), strength=1, linejoin = "mitre") +
```

```
  guides(edge_alpha = guide_edge_direction()) +

  geom_node_text(aes(label = name), size = 1 + 0.1*degree(i), color=rgb(.1,.1,.1,.7), repel = T)

j
```



No Idea how this is being visualised:

```
ggraph(graph=i, 'fabric', sort.by = node_rank_fabric()) +
  geom_node_range(colour = 'grey') +
  geom_edge_span(end_shape = 'square') +
  coord_fixed()
```