Canopy Exploration

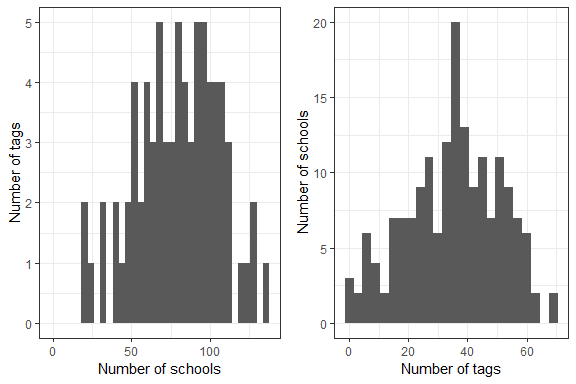
Gregor Thomas

12/12/2019

This is a working document for exploring the Canopy data. It contains incpomplete thoughts and likely mistakes.

TODO a little bit of data checking and summary stats. This will need to be fleshed out more.

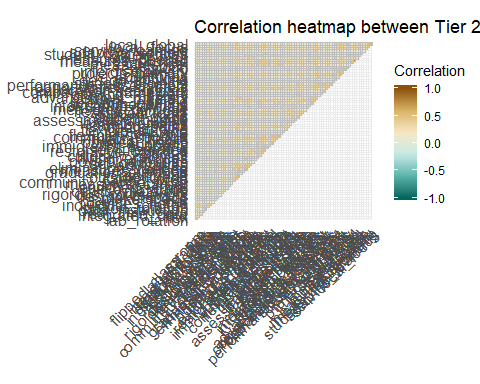
cl\_sch = conf\_long %>% group\_by(school\_id) %>%  
 summarize(n\_tag = sum(value))  
cl\_tag = conf\_long %>% group\_by(t2) %>%  
 summarize(n\_sch = sum(value))  
  
gridExtra::grid.arrange(  
 ggplot(cl\_tag, aes(x = n\_sch)) +   
 geom\_histogram(binwidth = 4) +  
 labs(x = "Number of schools", y = "Number of tags") +  
 expand\_limits(x = 0),  
 ggplot(cl\_sch, aes(x = n\_tag)) +  
 geom\_histogram(binwidth = 3) +   
 labs(x = "Number of tags", y = "Number of schools"),  
 nrow = 1  
)



## Clustering T2 Tags

Let’s look at a heatmap of correlations between the Tier 2 tags. The rows and columns are ordered to cluster similar correlations together, hence the more blue on the left side, and higher correlations on the right.

ggcorrplot(conf\_cor, hc.order = T, type = "upper") +  
 scale\_fill\_distiller(type = "div", limits = c(-1, 1), expand = c(0, 0)) +  
 labs(title = "Correlation heatmap between Tier 2 Tags",  
 fill = "Correlation")

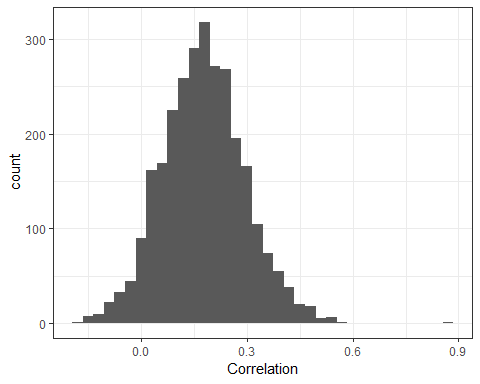


theme(axis.text=element\_blank())

## List of 1  
## $ axis.text: list()  
## ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
## - attr(\*, "class")= chr [1:2] "theme" "gg"  
## - attr(\*, "complete")= logi FALSE  
## - attr(\*, "validate")= logi TRUE

Looking at the distribution of correlations, we see more-or-less a bell curve, with a mean of 0.18, and one outlier pair, measures\_sel and measures\_climate, with an extremely high correlation of 0.87. This appears to be a robust outlier, with 99 schools confirming measures\_sel, 96 confirmint measures\_climate, and 92 of those overlapping.

cor\_df = conf\_cor %>%  
 as.table %>%  
 as.data.frame %>%  
 arrange(desc(Freq)) %>%  
 rename(Correlation = Freq) %>%  
 filter(as.integer(Var1) < as.integer(Var2))  
  
  
ggplot(cor\_df, aes(x = Correlation)) +  
 geom\_histogram(binwidth = 0.03)



knitr::kable(rbind(  
 head(cor\_df),  
 tail(cor\_df)  
), caption = "Top 6 and bottom 6 correlated tags")

Top 6 and bottom 6 correlated tags

|  |  |  |  |
| --- | --- | --- | --- |
|  | Var1 | Var2 | Correlation |
| 1 | measures\_sel | measures\_climate | 0.8713228 |
| 2 | exhibitions | portfolios | 0.5720029 |
| 3 | interdisciplinary | real\_world | 0.5472472 |
| 4 | local\_focus | local\_global | 0.5456812 |
| 5 | poverty\_supports | hiring\_practices | 0.5381816 |
| 6 | graduation\_supports | measures\_college | 0.5375568 |
| 2845 | station\_rotation | advisories | -0.1424051 |
| 2846 | flipped\_classroom | advisories | -0.1433583 |
| 2847 | enriched\_virtual | student\_goals | -0.1448130 |
| 2848 | enriched\_virtual | sel\_plan | -0.1487137 |
| 2849 | station\_rotation | design\_margins | -0.1587455 |
| 2850 | flipped\_classroom | flexible\_schedule | -0.1658566 |

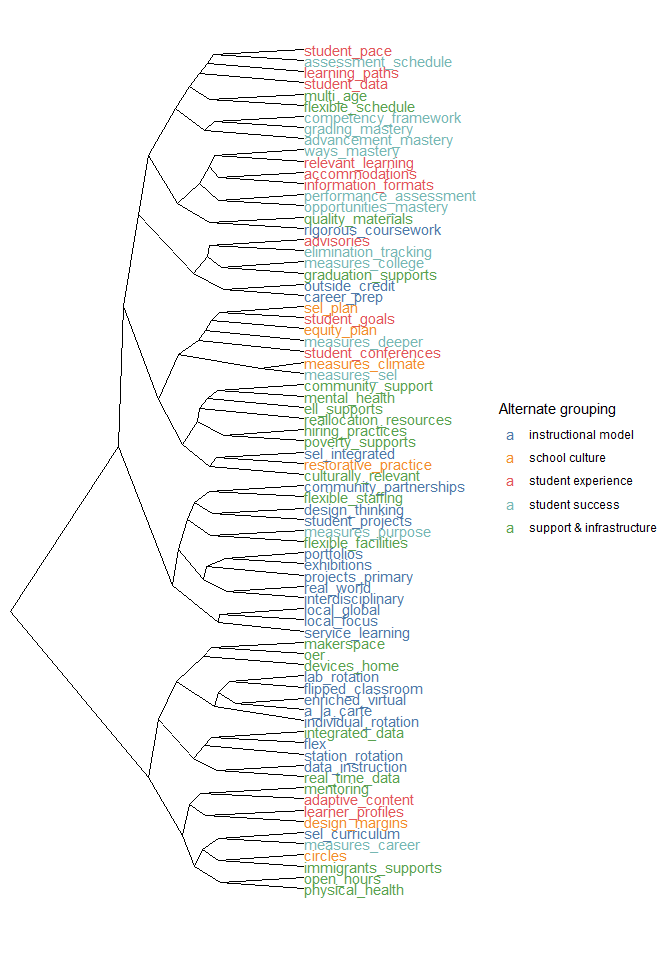
## Clustering

We’ll try a few clustering methods to explore the nuances they each give us and make sure our results are robust. First, hierarchical agglomerative clustering, then exploratory factor analysis. We will also look at the data as a network graph with weighted edges, and use community detection (as if it was a social network).

For now, we’ll focus on the confirmed data.

### Hierarchical clustering

ggplot() +  
 geom\_segment(data = segment(gg\_hc), aes(x, y, xend = xend, yend = yend)) +  
 geom\_text(data = hc\_labs, aes(x, y, label = label, color = t1\_alt), hjust = 0, size = 4) +  
 coord\_flip(clip = "off") +  
 scale\_color\_tableau(name = "Alternate grouping") +  
 scale\_y\_reverse(expand = expand\_scale(mult = c(0, 0.5), add = c(0, 2))) +  
 theme\_dendro() +  
 theme(panel.border = element\_blank())

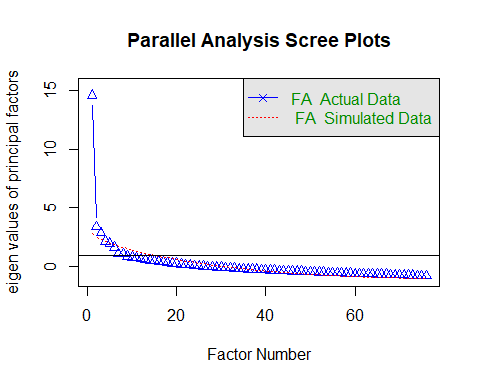


The Ward distance metric seems pretty good - might be worth trying out some others (definitely Jaccard, maybe Manhattan). Euclidean didn’t seem as sensible.

### Exploratory Factor Analysis

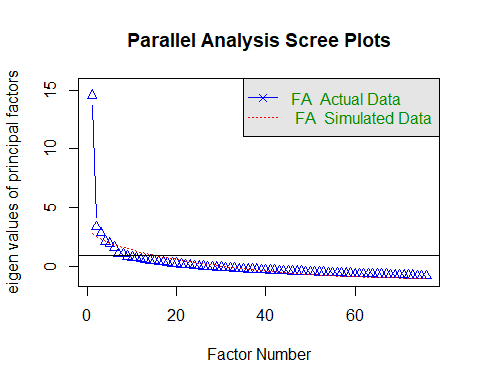
With EFA, we first need to run diagnostics to select the optimal number of clusters.

fa\_pa = fa.parallel(conf\_cor, fm = "pa", fa = "fa", n.obs = nrow(conf\_cor))



## Parallel analysis suggests that the number of factors = 4 and the number of components = NA

fa\_mr = fa.parallel(conf\_cor, fm = "minres", fa = "fa", n.obs = nrow(conf\_cor))



## Parallel analysis suggests that the number of factors = 4 and the number of components = NA

The principal axis method and minimum residual methods recommend 4 clusters, but 5 and 6 clusters are also fairly close.

TODO - actual factor analysis.

### Community Detection

TODO