Project Milestone II Figure 3B 2nd Graph

Ryan Thackston

4/8/2021

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(LICORS)
```

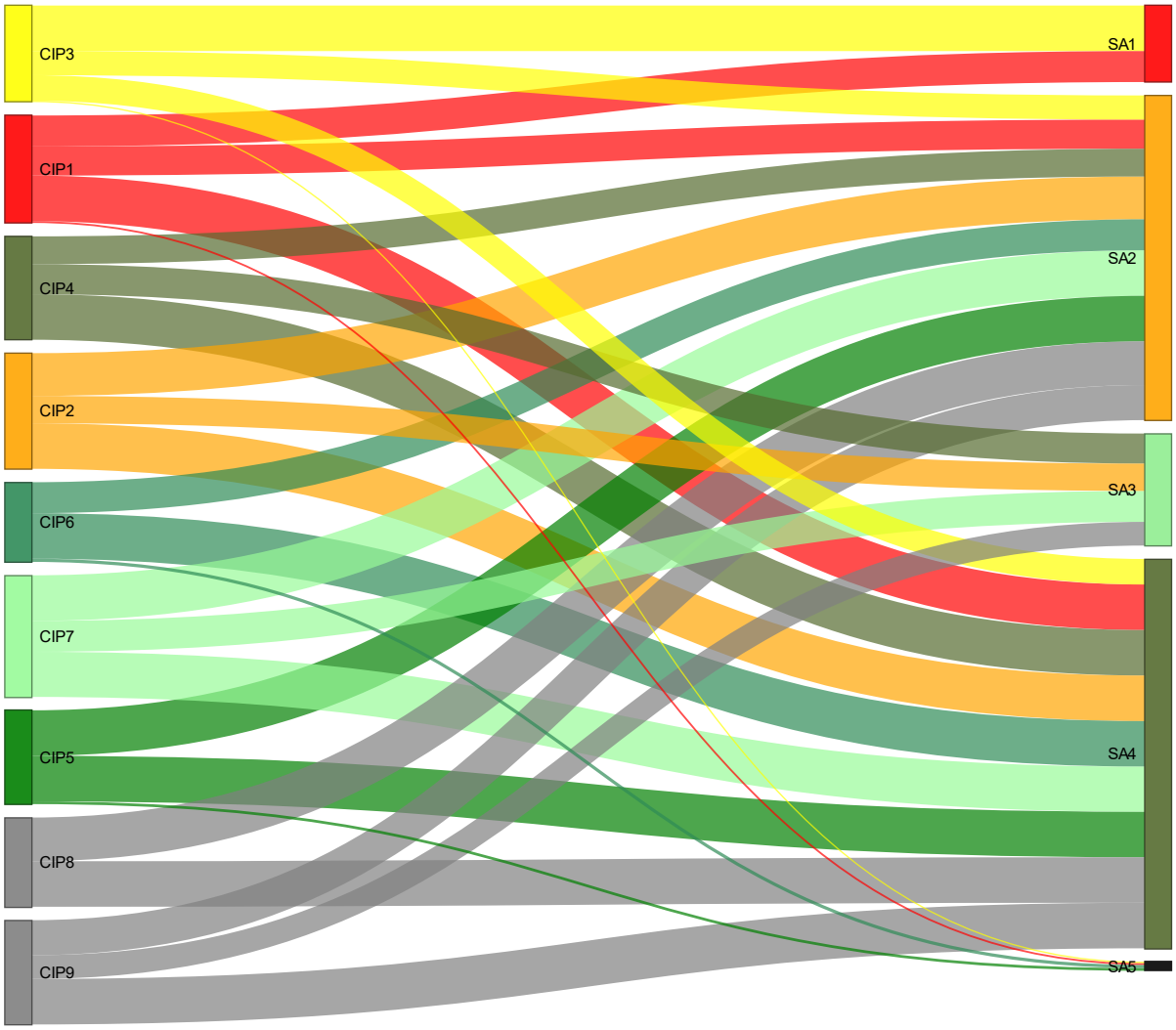
```
## Warning: package 'LICORS' was built under R version 4.0.5
```

```
library(scales)
library(networkD3)

setwd("G:\\COSC_6323_Statistics_for_Researchers\\Project")
data_csv<-read.csv("ArticleLevel-RegData-ALLSA_Xc_1_NData_655386_LONGXCIP2.csv")


# filter out years 2008 to 2018
year_2008_2018<-filter(data_csv, Yp >= 2008 & Yp <= 2018)

# Filter out IRegionRefined, unneccessary
IRegionRefinedp<-filter(year_2008_2018, IRegionRefinedp > 0 & IRegionRefinedp < 4)

# year_2008_2018

# Filter out where both NEUROLONGXSAp & NEUROLONGXCIPp == 0
df_mono = year_2008_2018 %>% filter(NEUROLONGXSAp == 0 & NEUROLONGXCIPp == 0)


mono_mat = matrix(0L, nrow = 9, ncol = 6)
# mono matrix
for(i in 1:nrow(df_mono)){
  row = df_mono[i,]
  vSA = c(row$SA1, row$SA2, row$SA3, row$SA4, row$SA5, row$SA6)
  vCIP = c(row$CIP3, row$CIP1, row$CIP4, row$CIP2, row$CIP6, row$CIP7, row$CIP5, ro
w$CIP8, row$CIP9)
  vSA = round(vSA / sum(vSA),2)
  for(k in which(vCIP > 0)){
    for(j in 1:6){
      mono_mat[[k,j]] = mono_mat[[k,j]] + vSA[j]
    }

  }

}
print(mono_mat)
```

```
##           [,1]     [,2]     [,3]     [,4]    [,5]   [,6]
##  [1,]  7294.03  3911.82  3005.30  4152.91  360.43 131.57
##  [2,] 21556.94 20333.56 12605.68 31666.00 1714.39 470.61
##  [3,]  1371.51  2588.36  2785.61  4095.13  407.94 194.06
##  [4,]  3493.89 10541.45  6908.68 11152.84 1368.91 484.09
##  [5,]  4619.69  8034.68  5299.87 11628.55 1292.40 462.60
##  [6,]  1699.08  3959.05  2720.97  3932.20  353.55 104.45
##  [7,]  7460.56 12556.45  8254.49 28062.07 2417.49 682.51
##  [8,]   910.27  1531.91   922.62  1590.40  684.88 279.12
##  [9,]   631.49  1753.99  1325.08  2154.90  868.40 416.60
```

```
m = mono_mat
for(i in 1:9){
  row = mono_mat[i,]
  # m[i,] = sapply(row, function(X) {(X - min(row))/(max(row)-min(row))})
  m[i,] = rescale(row, to=c(0,1))

}
#mm_b = apply(m, 2, function(x) {ifelse(x > 0.5, round(x,2), 0)})
mm_b = apply(m, 2, function(x) {ifelse(x > 0, round(x,2), 0)})
mm = rescale(mm_b, to=c(0,0.02))


# nodes = data.frame("name" = c("CIP3", "CIP1", "CIP4", "CIP2", "CIP6", "CIP7", "CIP
5", "CIP8", "CIP9", "SA1", "SA2", "SA3", "SA4", "SA5", "SA6"))
nodes = data.frame("name" = c("CIP3", "CIP1", "CIP4", "CIP2", "CIP6", "CIP7", "CIP5",
"CIP8", "CIP9", "SA1", "SA2", "SA3", "SA4", "SA5"))

links = as.data.frame(matrix(c(0,9,  mm[1,1],
                               0,10, mm[1,2],
                               0,12, mm[1,4],
                               0,13, mm[1,5],
                               1,9,  mm[2,1],
                               1,10, mm[2,2],
                               1,12, mm[2,4],
                               1,13, mm[2,5],
                               2,10, mm[3,2],
                               2,11, mm[3,3],
                               2,12, mm[3,4],
                               3,10, mm[4,2],
                               3,11, mm[4,3],
                               3,12, mm[4,4],
                               4,10, mm[5,2],
                               4,12, mm[5,4],
                               4,13, mm[5,5],
                               5,10, mm[6,2],
                               5,11, mm[6,3],
                               5,12, mm[6,4],
                               6,10, mm[6,2],
                               6,12, mm[7,4],
                               6,13, mm[7,5],
                               7,10, mm[8,2],
                               7,12, mm[8,4],
                               8,10, mm[9,2],
                               8,11, mm[9,3],
                               8,12, mm[9,4]
                               ), byrow = TRUE, ncol = 3))
names(links) = c("source", "target", "value")
links$group <- as.factor(c("type_0","type_0","type_0","type_0", "type_1","type_1","ty
pe_1","type_1","type_2", "type_2","type_2","type_3","type_3", "type_3","type_4","type
_4","type_4","type_5","type_5","type_5","type_6","type_6","type_6","type_7","type_
7","type_8","type_8","type_8"))
```

```
# node_color <- 'd3.scaleOrdinal() .domain(["CIP3", "CIP1", "CIP4", "CIP2", "CIP6", "CIP7", "CIP5", "CIP8", "CIP9", "SA1", "SA2", "SA3", "SA4", "SA5", "SA6", "type_0", "type_1", "type_2", "type_3", "type_4", "type_5", "type_6", "type_7", "type_8", "type_12"]) .range(["yellow", "red", "darkolivegreen" , "orange", "seagreen", "palegreen", "green", "gray", "gray", "red", "orange", "lightgreen", "darkolivegreen", "black", "gray", "yellow", "red", "darkolivegreen" , "orange", "seagreen", "palegreen", "green", "gray", "gray","white"])'

node_color <- 'd3.scaleOrdinal() .domain(["CIP3", "CIP1", "CIP4", "CIP2", "CIP6", "CIP7", "CIP5", "CIP8", "CIP9", "SA1", "SA2", "SA3", "SA4", "SA5", "SA6", "type_0", "type_1", "type_2", "type_3", "type_4", "type_5", "type_6", "type_7", "type_8", "type_12"]) .range(["yellow", "red", "darkolivegreen" , "orange", "seagreen", "palegreen", "green", "gray", "gray", "red", "orange", "lightgreen", "darkolivegreen", "black", "gray", "yellow", "red", "darkolivegreen" , "orange", "seagreen", "palegreen", "green", "gray", "gray","white"])'


p = sankeyNetwork(Links = links,
                  Nodes = nodes,
                  Source = "source",
                  Target = "target",
                  Value = "value",
                  NodeID = "name",
                  fontSize= 12,
                  nodeWidth = 20,
                  height = 800,
                  width = "100%",
                  colourScale=node_color,
                  LinkGroup="group",
                  iterations = 0,
                  nodePadding=10)

p
```



Mono-domain Articles