Project Milestone II Figure 3B 3rd 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)

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

year_2008_2018<-filter(data_csv, Yp >= 2009 & Yp <= 2018)

IRegionRefinedp<-filter(year_2008_2018, IRegionRefinedp > 0 & IRegionRefinedp < 4)


df_mono = year_2008_2018 %>% filter(NEUROLONGXSAp == 0 & NEUROLONGXCIPp == 0)
df_XD = year_2008_2018 %>% filter(NEUROLONGXSAp == 1 & NEUROLONGXCIPp == 1)


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)
```

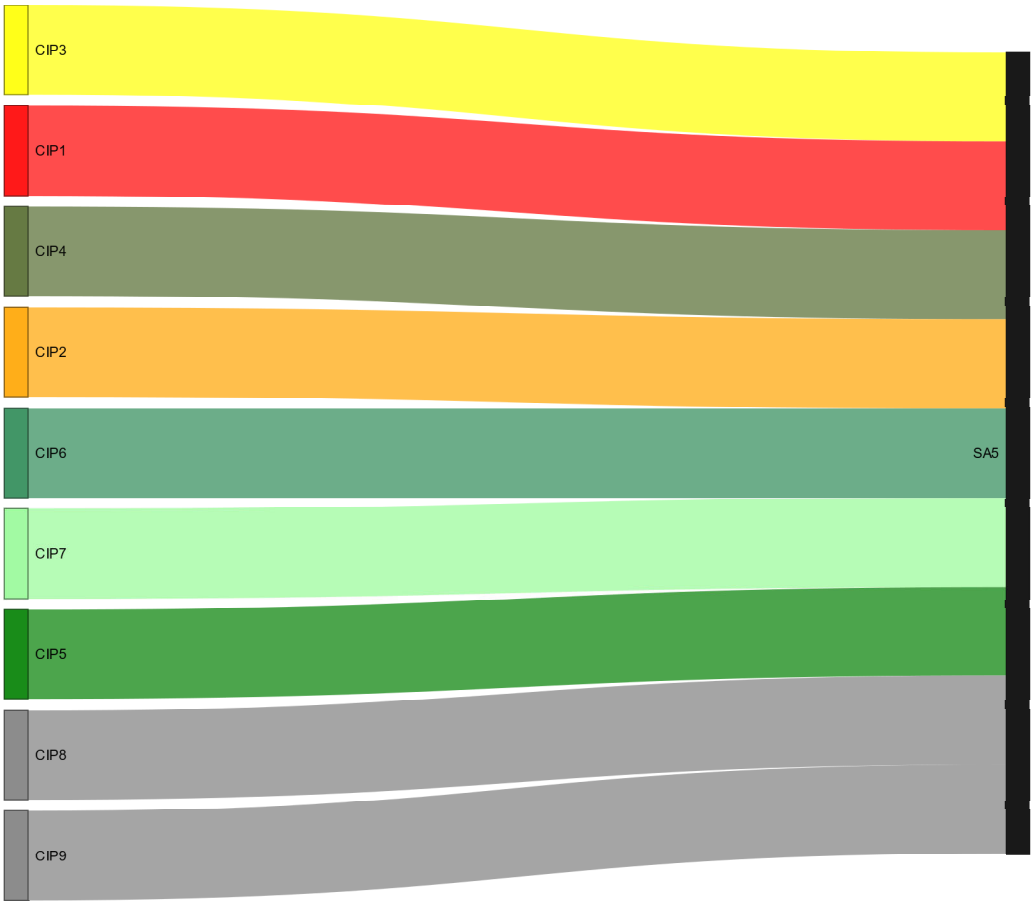
```
##           [,1]     [,2]     [,3]     [,4]    [,5]   [,6]
##  [1,]  6787.17  3618.41  2759.66  3857.22  327.66 119.34
##  [2,] 19934.66 18673.17 11479.54 29265.15 1590.53 432.47
##  [3,]  1275.54  2339.73  2539.63  3785.82  383.36 183.64
##  [4,]  3282.75  9598.81  6297.16 10326.54 1269.26 454.74
##  [5,]  4267.80  7378.05  4826.99 10700.34 1188.72 419.28
##  [6,]  1560.95  3585.80  2471.23  3617.39  329.15  96.85
##  [7,]  6939.61 11540.33  7540.50 25982.80 2206.71 618.29
##  [8,]   856.00  1388.11   811.67  1451.48  624.57 249.43
##  [9,]   573.83  1527.55  1151.15  1889.12  762.46 378.54
```

```
xd_mat = matrix(0L, nrow = 9, ncol = 6)
# xd matrix
for(i in 1:nrow(df_XD)){
  row = df_XD[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){
      xd_mat[[k,j]] = xd_mat[[k,j]] + vSA[j]
    }

  }

}
print(xd_mat)
```

```
##         [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
##  [1,] 107.03 103.99 103.94 129.29 270.55  35.11
##  [2,] 311.98 391.72 308.33 562.26 956.83 121.64
##  [3,] 244.03 316.47 287.85 583.69 802.55  94.18
##  [4,]  47.05  73.74  63.37 108.79 183.66  21.71
##  [5,]  66.22  79.42  67.69 130.16 234.92  25.65
##  [6,]  18.37  17.28  24.72  30.24  60.68   5.49
##  [7,] 135.00 300.13 204.11 442.86 710.68 106.56
##  [8,] 191.13 346.64 209.12 321.25 766.94 141.68
##  [9,]   7.35  11.50  17.04  21.38  40.63   4.84
```

```
m = mono_mat
for(i in 1:9){
  row = mono_mat[i,]
  m[i,] = rescale(row, to=c(0,1))

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


x = xd_mat
for(i in 1:9){
  row = xd_mat[i,]
  x[i,] = sapply(row, function(X) {(X - min(row))/(max(row)-min(row))})
}
XD_b = apply(x, 2, function(x) {ifelse(x > 0.5, round(x,2), 0)})
XD = rescale(XD_b, to=c(0,0.02))




## Diff between Mono and XD
diff_x_m = XD_b - mm_b



## keeping only positive(+) values
diff_x_n = apply(diff_x_m, 2, function(x) {ifelse(x > 0, round(x,2), 0)})
diff_x_mm = rescale(diff_x_n, to=c(0,0.02))

nodes = data.frame("name" = c("CIP3", "CIP1", "CIP4", "CIP2", "CIP6", "CIP7", "CIP5",
"CIP8", "CIP9", "", "", "", "", "SA5", ""))

links = as.data.frame(matrix(c(0,13,  diff_x_mm[1,5],
                               1,13,  diff_x_mm[2,5],
                               2,13,  diff_x_mm[3,5],
                               3,13,  diff_x_mm[4,5],
                               4,13,  diff_x_mm[5,5],
                               5,13,  diff_x_mm[6,5],
                               6,13,  diff_x_mm[7,5],
                               7,13,  diff_x_mm[8,5],
                               8,13,  diff_x_mm[9,5]
), byrow = TRUE, ncol = 3))
names(links) = c("source", "target", "value")
links$group <- as.factor(c("type_0","type_1","type_2","type_3","type_4","type_5","typ
e_6","type_7","type_8"))
node_color <- 'd3.scaleOrdinal() .domain(["CIP3", "CIP1", "CIP4", "CIP2", "CIP6", "CI
P7", "CIP5", "CIP8", "CIP9", "SA1", "SA2", "SA3", "SA4", "SA5", "SA6", "type_0", "typ
e_1", "type_2", "type_3", "type_4", "type_5", "type_6", "type_7", "type_8", "type_1
2"]) .range(["yellow", "red", "darkolivegreen" , "orange", "seagreen", "palegreen", "
green", "gray", "gray", "red", "orange", "lightgreen", "darkolivegreen", "black", "gr
```

```
ay", "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
```



Difference