Sunburst Plot of NCF Alumn data

Data Viz 2022 class

2/1/2022

Reading dataset in Long Format:

```
library(readr)
```

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

```
mypath <- 'https://raw.githubusercontent.com/bklingen/DataViz2022/main/Data/NCAlumnIndustry2Long.csv'
df <- read_csv(file=mypath)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   ID = col_double(),
##   GRAD_YEAR = col_double(),
##   AOC_num = col_double(),
##   AOC = col_character(),
##   DIVISION = col_character(),
##   Industry = col_character(),
##   Industry2 = col_character(),
##   Position = col_character(),
##   Org_Name = col_character()
## )
```

Let's see how many students fall in each of the `Industry2` categories:

```
library(dplyr)
```

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

```
##
## 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
```

```
df %>% count(Industry2, sort=TRUE)
```

```
## # A tibble: 7 x 2
##   Industry2                          n
##   <chr>                          <int>
## 1 Education                        378
## 2 Government and Social Services   303
```

```
## 3 Business, Finance, and Retail    250
## 4 STEM                             211
## 5 Health and Medicine              161
## 6 Arts and Media                   144
## 7 Other                             29
```

Note that this double counts students who have double or triple majors (AOC's). E.g., a student with two AOCs is counted twice.

Now, let's see how within an `Industry2` code, e.g., `Government and Social Services`, the Industries (variable `Industry`) are distributed:

```
df %>%
  filter(Industry2 == "Government and Social Services") %>%
  count(Industry, sort=TRUE)
```

```
## # A tibble: 16 x 2
##    Industry                    n
##    <chr>                   <int>
##  1 Law                        98
##  2 Non Profit                 92
##  3 Government                 61
##  4 City/County/State          11
##  5 Nonprofit                  11
##  6 Religion                    8
##  7 Community Services          5
##  8 Attorney                    4
##  9 Development                 3
## 10 International Relations     2
## 11 Military                    2
## 12 Public Relations            2
## 13 Defense & Space             1
## 14 Economic Development        1
## 15 Environmental Law           1
## 16 Philanthropy                1
```

I would say we take some top percentage of the Industries within each Industry2 label, which we plot as the "outer" ring in the sunburst chart. Here is one idea to go about this, but it may be wrong. Also, I dropped the "Other" category for `Industry2`:

```
library(forcats)
```

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

```
df1 = df %>%
  filter(Industry2 != 'Other') %>%
  group_by(Industry2) %>%
  mutate(Industry2.1 = fct_lump_prop(Industry, 0.05)) %>% count(Industry2.1)

df1
```

```
## # A tibble: 34 x 3
## # Groups:   Industry2 [6]
##    Industry2                    Industry2.1              n
##    <chr>                        <fct>                <int>
##  1 Arts and Media               Arts                    25
##  2 Arts and Media               Arts & Entertainment    10
##  3 Arts and Media               Entertainment           19
```

```
##  4 Arts and Media                Film                    8
##  5 Arts and Media                Media                  33
##  6 Arts and Media                Music                  10
##  7 Arts and Media                Publishing             12
##  8 Arts and Media                Other                  27
##  9 Business, Finance, and Retail Other                  75
## 10 Business, Finance, and Retail Banking                13
## # ... with 24 more rows
```

Wit hthis infomration in `df1`, I hope you can create the Sunburst plot. Look up the help for sunburst plot in plotyly. I think the third example might be helpful, where they define `ids`, but I'm not certain. (But we would need `values`, in the example everything seems to have equal weight.)

```
library(plotly)
```

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

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```