data spec assess 2

Jodie

Instructions: You will see a data-driven chart on the "chart" tab of the Excel file. The motivation for creating this type of "Sankey diagram" chart is summarized in the PDF article (for reference only). The assignment is to see if you can find an approach to recreating the same chart in the Excel file, but using ggplot2. An example of such a plot is shown in the file 'example-plot-R.png'. Please do your best to match the excel file output in terms of ribbon color, ribbon opacity, text color, text location, and the size of the white-space surrounding the colored shapes in the plot.

Please submit you answer in the form of an RMarkdown document, or other literate programming (https://en.wikipedia.org/wiki/Literate_programming) form (Jupiter Notebook EG).

Strongly Encouraged: ï Include versions of the sankey diagram with and without the horizontal and vertical white space in the plot. (With and without the white space separating shapes of the same color from one another, and with and without the white space separating shapes of different colors from one another.)

For Bonus Points: ï Create a function that takes in datasets and plots the sankey diagram. Make the function able to output plot versions with and without the vertical and horizontal white space. ï Submit your answer to us by creating a package (http://r-pkgs.had.co.nz/). Include the dataset you use to create the sankey diagram in the package, and include your writeup as a package vignette. Upload it to github so that we can install it from R.

2

```r
library("readxl")
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3

##
## 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
```

```r
data_2 <- read_excel("Stacked_column_stroke_blacks_noAF.xlsx", sheet="Data")
#data_2

cleaned_2 <- data_2[1:5, ] # only want to see the
cleaned_2
```

```
## # A tibble: 5 x 6
##   `Risk Factors for Stroke in Blacks` `1990`         `1995` `2000` `2005` `2010`
##   <chr>                               <chr>           <dbl>  <dbl>  <dbl>  <dbl>
## 1 Obesity                             1E-3            0.013  0.043  0.077  0.115
## 2 Diabetes                            0.35899999999~  0.316  0.26   0.187  0.092
## 3 Smoking                             0.17100000000~  0.156  0.142  0.128  0.116
## 4 Hypercholesterolemia                0.161           0.104  0.045  0.001  0.001
## 5 Hypertension                        0.65400000000~  0.633  0.602  0.561  0.509
```

```
#cleaned_2

# Arrange entries based on descending order of the first column
#cleaned_2 <- arrange(cleaned_2, desc(`1990`))

#cleaned_2[[2]]
#typeof(cleaned_2[[1]]) # 1990 column is a character, need to convert to double
cleaned_2$`1990` <- as.numeric(cleaned_2$`1990`) # 1990 column is a character, need to convert to doubl

# look at the attached notes to see why it is necessary to rearrange the df entries based on the first
# Arrange entries based on descending order of the first column
cleaned_2 <- cleaned_2 %>% arrange(desc(`1990`))
cleaned_2
```

```
## # A tibble: 5 x 6
##   `Risk Factors for Stroke in Blacks` `1990` `1995` `2000` `2005` `2010`
##   <chr>                                <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Hypertension                         0.654  0.633  0.602  0.561  0.509
## 2 Diabetes                             0.359  0.316  0.26   0.187  0.092
## 3 Smoking                              0.171  0.156  0.142  0.128  0.116
## 4 Hypercholesterolemia                 0.161  0.104  0.045  0.001  0.001
## 5 Obesity                              0.001  0.013  0.043  0.077  0.115
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

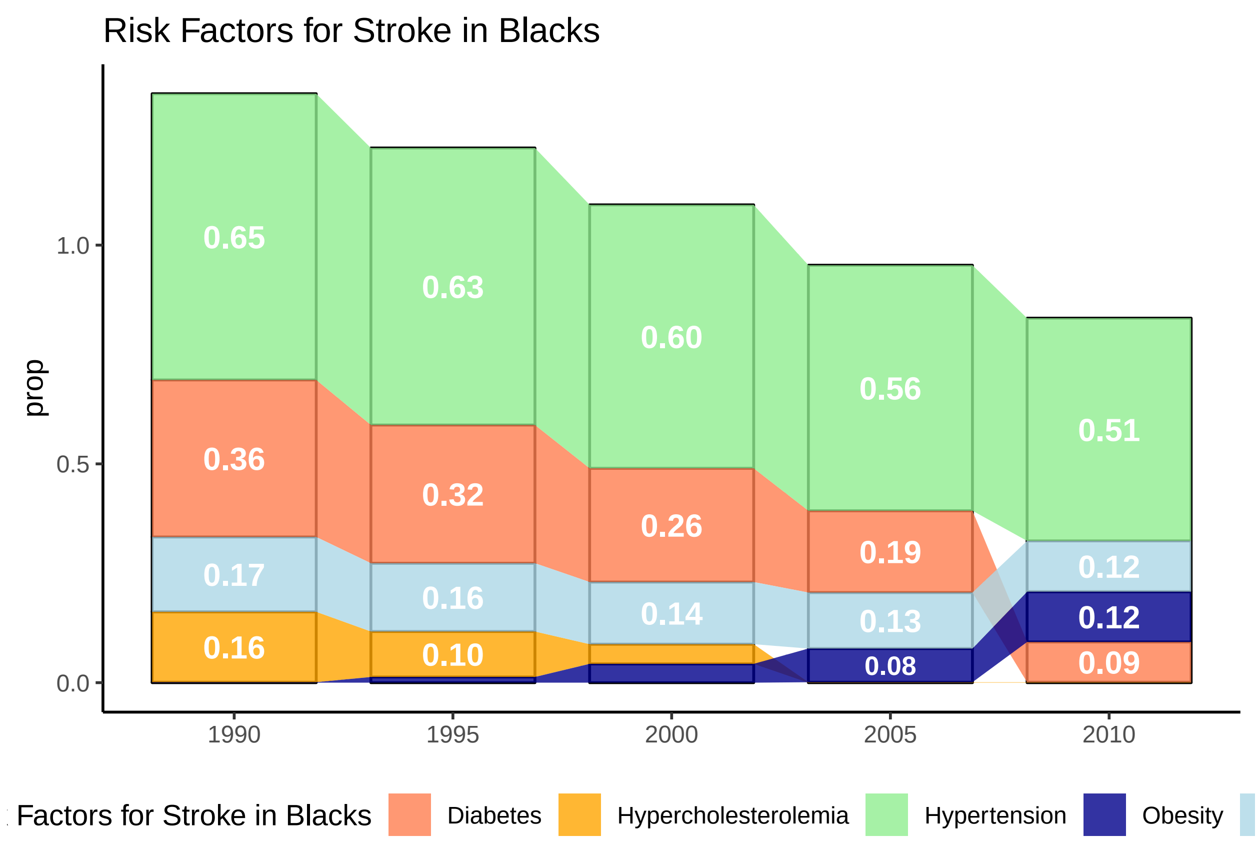
```
library(ggsankey)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(ggalluvial)
#cleaned_2[, -1] <- format(round(data[, -1], 2), nsmall = 2)

cleaned2_long <- cleaned_2 %>% # convert dataframe to long format so that
                              # we have columns for risk factor, year, prop (numerical values)
  pivot_longer(c(`1990`, `1995`, `2000`, `2005`, `2010`),
    names_to = "year",
    values_to = "prop"
  )
```

```r
plot <- ggplot(data=cleaned2_long,
       aes(x = factor(year), # make year into categorical variables
           y = prop,
           alluvium = `Risk Factors for Stroke in Blacks`,
           node = factor(year),
           stratum = `Risk Factors for Stroke in Blacks`,
           label = prop))

plot +
  geom_stratum(aes(stratum = `Risk Factors for Stroke in Blacks`), decreasing = F, width = 3/4) +
  geom_alluvium(aes(fill = `Risk Factors for Stroke in Blacks`),
                decreasing = F,
                alpha = 0.8,  # to set opacity
                width = 3/4,
                knot.prop = T,
                curve_type = "linear",
                stratum_spacing = 0.5) + # stratumspacing to adjust the spacing between the ribbons?
  ggfittext::geom_fit_text(stat = "stratum", width = 3/4, decreasing = FALSE,# min.size to adjust the m
                           aes(label = sprintf("%.2f", prop)), color="white", fontface="bold") + # spri

  ggtitle("Risk Factors for Stroke in Blacks") +
  scale_fill_manual(values = c("Hypertension" = "lightgreen",  # change the color of the stratum
                               "Diabetes" = "coral",
                               "Smoking" = "lightblue",
                               "Hypercholesterolemia" = "orange",
                               "Obesity" = "darkblue")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab(NULL) +
  theme_classic() +
  theme(legend.position = "bottom")
```

```
## Warning in geom_alluvium(aes(fill = `Risk Factors for Stroke in Blacks`), :
## Ignoring unknown parameters: `stratum_spacing`
```



I’m having trouble adjusting the spacing between the ribbons labelling the ribbons with risk factors.