Make\_Abortions\_Move\_Again

Emil

26/11/2020

library(tidyverse)

## -- Attaching packages ----------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.3 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts -------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(gganimate) # Her loades en pakke til at kunne lave visualiseringer  
library(gifski) # Her loades en pakke, så vi kan vise visualiseringer som bevæger sig, længere nede.  
library(ggplot2)  
library(shiny)

## Warning: package 'shiny' was built under R version 4.0.3

library(plotly)

## Warning: package 'plotly' was built under R version 4.0.3

##   
## 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

Okay, now that we have the tools we need from the library, we will download the data we would like to visualize

finaldflink <- "https://raw.githubusercontent.com/Digital-Methods-HASS/au557916\_Thomsen\_Emil/main/final\_df.csv"  
finaldf05\_18link <- "https://raw.githubusercontent.com/Digital-Methods-HASS/au557916\_Thomsen\_Emil/main/final\_df05\_18.csv"  
  
final\_df <- read\_csv(url(finaldflink))

## Parsed with column specification:  
## cols(  
## Municipalities = col\_character(),  
## Year = col\_double(),  
## allAges = col\_double(),  
## age15\_19 = col\_double(),  
## age20\_24 = col\_double(),  
## age25\_29 = col\_double(),  
## age30\_34 = col\_double(),  
## age35\_39 = col\_double(),  
## age40\_44 = col\_double(),  
## age45\_49 = col\_double(),  
## av\_Income = col\_double()  
## )

final\_df05\_18 <- read\_csv(url(finaldf05\_18link))

## Parsed with column specification:  
## cols(  
## Municipalities = col\_character(),  
## Year = col\_double(),  
## allAges = col\_double(),  
## age15\_19 = col\_double(),  
## age20\_24 = col\_double(),  
## age25\_29 = col\_double(),  
## age30\_34 = col\_double(),  
## age35\_39 = col\_double(),  
## age40\_44 = col\_double(),  
## age45\_49 = col\_double(),  
## av\_Income = col\_double(),  
## av\_age = col\_double()  
## )

final\_df

## # A tibble: 2,304 x 11  
## Municipalities Year allAges age15\_19 age20\_24 age25\_29 age30\_34 age35\_39  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Albertslund 1995 0.466 0.0647 0.0987 0.136 0.0885 0.0544  
## 2 Albertslund 1996 0.565 0.0993 0.151 0.103 0.130 0.0582  
## 3 Albertslund 1997 0.492 0.0679 0.126 0.0747 0.146 0.0577  
## 4 Albertslund 1998 0.500 0.0850 0.116 0.0918 0.129 0.0646  
## 5 Albertslund 1999 0.457 0.0375 0.130 0.0989 0.0921 0.0750  
## 6 Albertslund 2000 0.358 0.0614 0.0750 0.0784 0.0852 0.0443  
## 7 Albertslund 2001 0.404 0.0514 0.103 0.103 0.0753 0.0616  
## 8 Albertslund 2002 0.409 0.0790 0.113 0.0721 0.0515 0.0755  
## 9 Albertslund 2003 0.506 0.0520 0.128 0.118 0.101 0.0867  
## 10 Albertslund 2004 0.456 0.123 0.119 0.0877 0.0702 0.0491  
## # ... with 2,294 more rows, and 3 more variables: age40\_44 <dbl>,  
## # age45\_49 <dbl>, av\_Income <dbl>

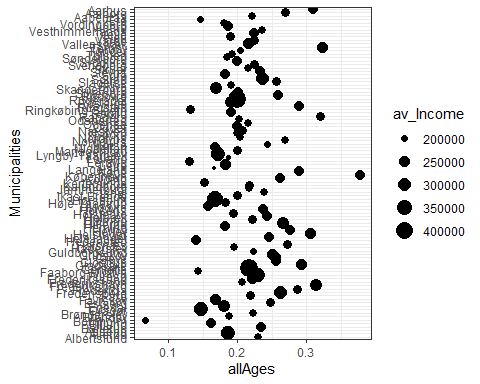
head(final\_df,26)

## # A tibble: 26 x 11  
## Municipalities Year allAges age15\_19 age20\_24 age25\_29 age30\_34 age35\_39  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Albertslund 1995 0.466 0.0647 0.0987 0.136 0.0885 0.0544  
## 2 Albertslund 1996 0.565 0.0993 0.151 0.103 0.130 0.0582  
## 3 Albertslund 1997 0.492 0.0679 0.126 0.0747 0.146 0.0577  
## 4 Albertslund 1998 0.500 0.0850 0.116 0.0918 0.129 0.0646  
## 5 Albertslund 1999 0.457 0.0375 0.130 0.0989 0.0921 0.0750  
## 6 Albertslund 2000 0.358 0.0614 0.0750 0.0784 0.0852 0.0443  
## 7 Albertslund 2001 0.404 0.0514 0.103 0.103 0.0753 0.0616  
## 8 Albertslund 2002 0.409 0.0790 0.113 0.0721 0.0515 0.0755  
## 9 Albertslund 2003 0.506 0.0520 0.128 0.118 0.101 0.0867  
## 10 Albertslund 2004 0.456 0.123 0.119 0.0877 0.0702 0.0491  
## # ... with 16 more rows, and 3 more variables: age40\_44 <dbl>, age45\_49 <dbl>,  
## # av\_Income <dbl>

head(final\_df05\_18, 26)

## # A tibble: 26 x 12  
## Municipalities Year allAges age15\_19 age20\_24 age25\_29 age30\_34 age35\_39  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Albertslund 2005 0.419 0.0746 0.110 0.0959 0.0533 0.0426  
## 2 Albertslund 2006 0.445 0.0826 0.118 0.0718 0.0790 0.0646  
## 3 Albertslund 2007 0.529 0.0942 0.170 0.109 0.0761 0.0580  
## 4 Albertslund 2008 0.514 0.0870 0.156 0.0797 0.101 0.0543  
## 5 Albertslund 2009 0.444 0.0794 0.123 0.115 0.0686 0.0541  
## 6 Albertslund 2010 0.361 0.0613 0.0649 0.0793 0.112 0.0288  
## 7 Albertslund 2011 0.450 0.0755 0.165 0.0504 0.101 0.0360  
## 8 Albertslund 2012 0.384 0.0538 0.0825 0.0969 0.0790 0.0538  
## 9 Albertslund 2013 0.381 0.0575 0.0934 0.0575 0.0647 0.0647  
## 10 Albertslund 2014 0.339 0.0361 0.0938 0.123 0.0613 0.0216  
## # ... with 16 more rows, and 4 more variables: age40\_44 <dbl>, age45\_49 <dbl>,  
## # av\_Income <dbl>, av\_age <dbl>

theme\_set(theme\_bw()) # set theme to white background for better visibility  
  
ggplot(subset(final\_df, Year == 2018), aes(allAges, Municipalities, size = av\_Income)) +   
 geom\_point()



ui <- fluidPage(  
 plotlyOutput("distPlot")  
)  
  
server <- function(input, output) {  
 output$distPlot <- renderPlotly({  
 qplot(allAges, Year, size = av\_Income, color = Municipalities, data = final\_df) +   
 geom\_point()  
 })  
}  
  
shinyApp(ui = ui, server = server)

Shiny applications not supported in static R Markdown documents

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
ui <- fluidPage(  
 plotlyOutput("distPlot")  
)  
  
server <- function(input, output) {  
 output$distPlot <- renderPlotly({  
 ggplot(subset(final\_df05\_18, Year == 2018), aes(x = allAges, y = av\_age, size = av\_Income, color = Municipalities)) +   
 labs(title = "scatter plot", x = "Abortions pr capita", y = "Averige age")+  
 geom\_point()  
 })  
}  
  
shinyApp(ui = ui, server = server)

Shiny applications not supported in static R Markdown documents

#Cool visualisation that shows averidge abortions per capita on the x-axis and the average age in the different municipalities on the y-axis  
  
  
  
  
  
  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
ui <- fluidPage(  
 plotlyOutput("distPlot")  
)  
  
server <- function(input, output) {  
 output$distPlot <- renderPlotly({  
 ggplot(subset(final\_df, Year == 2018), aes(x=Municipalities, y=allAges, color = av\_Income)) +  
 geom\_bar(stat="identity")  
 })  
}  
  
shinyApp(ui = ui, server = server)

Shiny applications not supported in static R Markdown documents

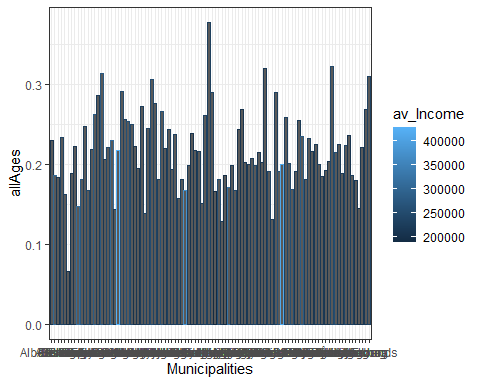
final\_df %>% #We choose to look at the gapminder dataset and add pipes to it, so we can send an output of one fucntion directly to another.  
 filter(Year == 2018) %>% #First we filter the years in tha dataset so vi are only looking at data from 1952  
 select(Municipalities , allAges) %>% #then we select or choose to look at the countries and their gdpPercap, as everything else is irelevant right now  
 arrange(desc(allAges)) #lastly, we arange our data in a decending order so we have the richest nation first.

## # A tibble: 96 x 2  
## Municipalities allAges  
## <chr> <dbl>  
## 1 København 0.378  
## 2 Tårnby 0.323  
## 3 Randers 0.321  
## 4 Frederiksberg 0.314  
## 5 Aarhus 0.310  
## 6 Helsingør 0.306  
## 7 Gladsaxe 0.292  
## 8 Ringsted 0.290  
## 9 Køge 0.290  
## 10 Fredericia 0.286  
## # ... with 86 more rows

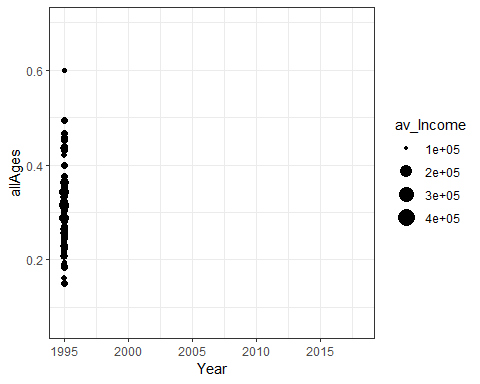
final\_df05\_18 %>% #We choose to look at the gapminder dataset and add pipes to it, so we can send an output of one fucntion directly to another.  
 filter(Year == 2018) %>% #First we filter the years in tha dataset so vi are only looking at data from 1952  
 select(Municipalities , av\_age) %>% #then we select or choose to look at the countries and their gdpPercap, as everything else is irelevant right now  
 arrange((av\_age))

## # A tibble: 96 x 2  
## Municipalities av\_age  
## <chr> <dbl>  
## 1 København 35.9  
## 2 Aarhus 37.7  
## 3 Ishøj 38.8  
## 4 Odense 39.5  
## 5 Frederiksberg 39.6  
## 6 Gladsaxe 39.7  
## 7 Vallensbæk 39.7  
## 8 Albertslund 39.8  
## 9 Aalborg 39.9  
## 10 Hvidovre 40   
## # ... with 86 more rows

søjle <- ggplot(subset(final\_df, Year == 2018), aes(x=Municipalities, y=allAges, color = av\_Income)) +  
 geom\_bar(stat="identity")  
  
søjle



kbh <- ggplot(subset(final\_df, Municipalities == Municipalities), aes( x = Year, y = allAges, size = av\_Income)) +  
 geom\_point() + #geom\_point() gør at vi får scatter plots altså prikker i vores kordinatsystem  
 transition\_time(Year) +  
 #scale\_x\_log10() +  
 transition\_states(Year, transition\_length = 12, state\_length = 1, wrap = TRUE)  
kbh



The code chunkbelow uses our final dataset to show the development in abortions per capita in procent along with the average income in the different municipalities in Denmark.

options(scipen=999) #By adding this line of code we disable the scientific way of writing lables.  
ggplot(final\_df, aes(x = av\_Income, y = allAges, size = av\_Income))+  
 geom\_point() +   
 # scale\_x\_log10() +   
 transition\_time(Year) +   
 labs(title = "{closest\_state}", x = "Averidge Income", y = "Abortions per capita") +   
 transition\_states(Year, transition\_length = 12, state\_length = 1, wrap = TRUE) +  
 theme(axis.text.x = element\_text(size = 12), #here we change the size of the labels.  
 axis.text.y = element\_text(size = 12),  
 text = element\_text(size = 16),  
 plot.title = element\_text(hjust = 0.5))

