

## Lista 5: Teoria de Vis

João Vitor Freitas Sousa

05/05/2020

### **Problema 1: The Economist**

**Resposta:**

### **Problema 2: Minard**

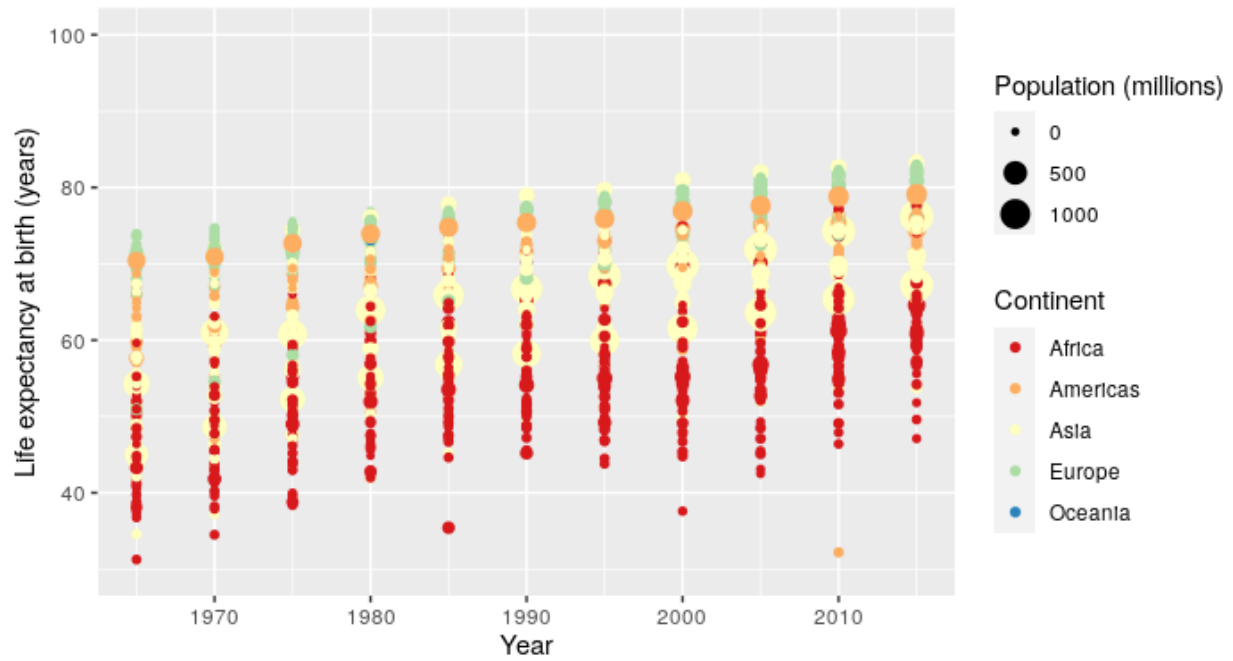
**Resposta:**

Pronto!

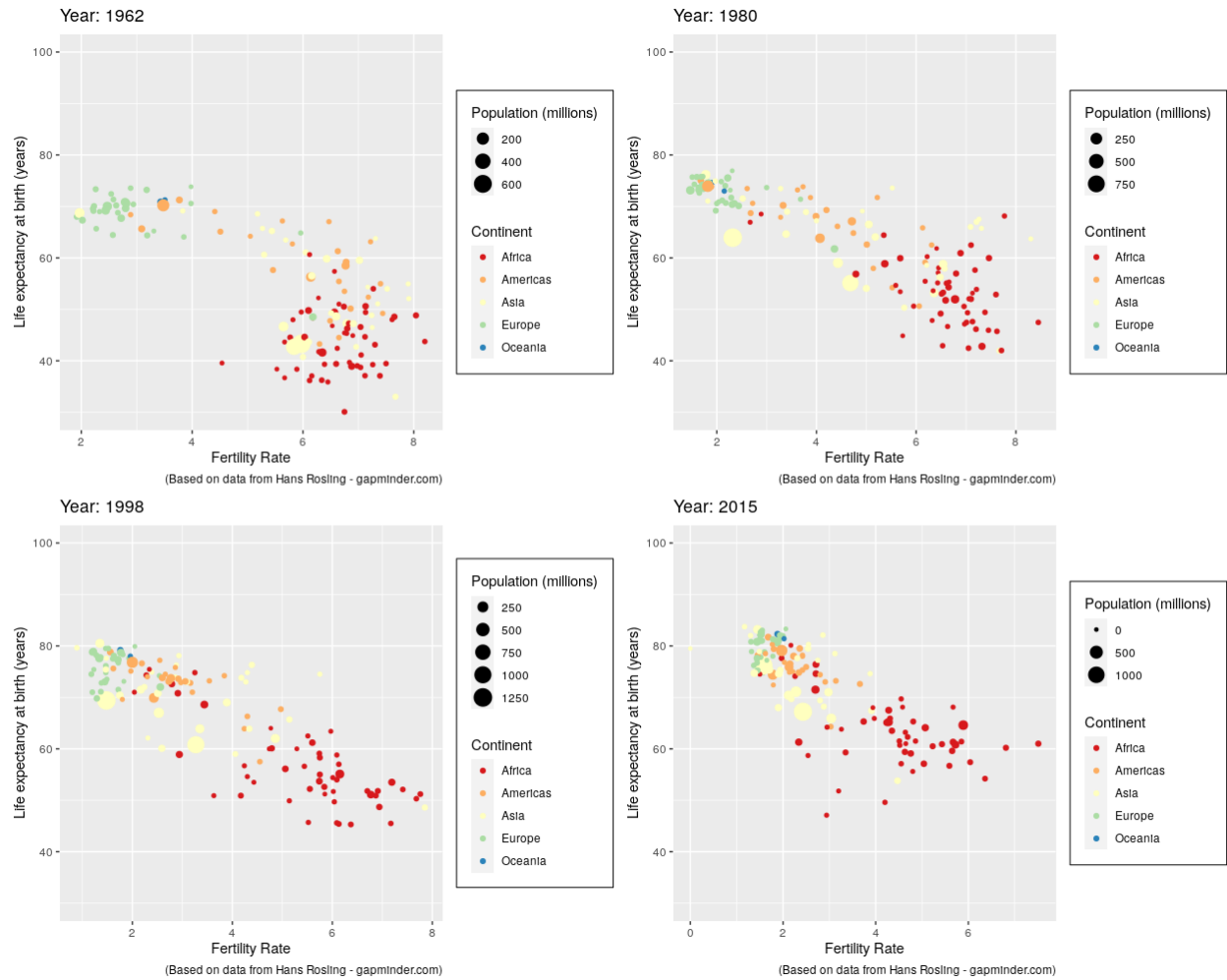
### **Problema 3: Gapminder**

**Resposta:**

O proposto para esse problema foi transformar o gráfico animado em algo estático. Algumas possibilidades foram testadas como, por exemplo, passar para o eixo X a sequência de anos. Soluções como essa foram descartadas por serem extensas, esteticamente desagradáveis ou nada intuitivas. Visto que o objetivo do gráfico original era demonstrar a mudança nos índices com a passagem de anos e a aglomeração de países da mesma região, pegar representações de anos igualmente espaçados se mostra uma boa maneira de chegar ao desejado. A solução apresentada será escolher 4 anos espaçados, recriar o gráfico de cada um desses anos e uni-los em um grid. Segue então uma das soluções descartadas e então o resultado final.



(Based on data from Hans Rosling - gapminder.com)



Código que gera o grid:

1 - Manipulação dos dados:

```
population_xls <- read.xlsx("./3/indicator gapminder population.xlsx",
                           encoding = "UTF-8",stringsAsFactors= F, sheetIndex = 1,
                           as.data.frame = TRUE, header=TRUE)
fertility_xls <- read.xlsx("./3/indicator undata total_fertility.xlsx",
                           encoding = "UTF-8",stringsAsFactors= F, sheetIndex = 1,
                           as.data.frame = TRUE, header=TRUE)
lifeexp_xls <- read.xlsx("./3/indicator life_expectancy_at_birth.xlsx",
                          encoding = "UTF-8",stringsAsFactors= F, sheetIndex = 1,
                          as.data.frame = TRUE, header=TRUE)

# Cria uma variável que será usada para filtrar os dados nas linhas a seguir
myvars <- paste("X", 1962:2015, sep="")
# Gera novos dataframes somente com os anos desejados.
population <- population_xls[c('Total.population',myvars)]
fertility <- fertility_xls[c('Total.fertility.rate',myvars)]
lifeexp <- lifeexp_xls[c('Life.expectancy',myvars)]
# Renomeia a primeira coluna dos dataframes com "Country"
colnames(population)[1] <- "Country"
colnames(fertility)[1] <- "Country"
colnames(lifeexp)[1] <- "Country"
# Remove linhas vazias no final do dataframe
lifeexp <- lifeexp[1:275,]
population <- population[1:275,]
# Transforma os dataframes trazendo os anos como coluna
population_m <- melt(population, id=c("Country"))
lifeexp_m <- melt(lifeexp, id=c("Country"))
fertility_m <- melt(fertility, id=c("Country"))
# Renomeia a coluna que mantém os dados
colnames(population_m)[3] <- "pop"
colnames(lifeexp_m)[3] <- "life"
colnames(fertility_m)[3] <- "fert"
# Unifica os dataframes
mydf <- merge(lifeexp_m, fertility_m, by=c("Country","variable"), header =T)
mydf <- merge(mydf, population_m, by=c("Country","variable"), header =T)
# Usando o gapminder para criar um lista com os continentes de cada país do dataframe
continent <- gapminder %>% group_by(continent, country) %>% distinct(country, continent)
continent <- data.frame(lapply(continent, as.character), stringsAsFactors=FALSE)
colnames(continent)[1] <- "Country"
# Retira os países que não estão presentes na tabela de continentes
mydf_filter <- mydf %>% filter(Country %in% unique(continent$Country))
# E finalmente adiciona a coluna continentes ao dataframe
mydf_filter <- merge(mydf_filter, continent, by=c("Country"), header =T)
#Remove as linhas com valores "N/A", retorna as strings para o formato "character"
#e converte os dados para o formato numérico.
mydf_filter[is.na(mydf_filter)] <- 0
mydf_filter <- data.frame(lapply(mydf_filter, as.character), stringsAsFactors=FALSE)
mydf_filter$variable <- as.integer(as.character(gsub("X","",mydf_filter$variable)))
colnames(mydf_filter)[colnames(mydf_filter)=="variable"] <- "year"
mydf_filter$pop <- round(as.numeric(as.character(mydf_filter$pop))/1000000,1)
mydf_filter$fert <- as.numeric(as.character(mydf_filter$fert))
mydf_filter$life <- as.numeric(as.character(mydf_filter$life))
```

```

ano <- mydf_filter[mydf_filter$year == 1962,]
ano2 <- mydf_filter[mydf_filter$year == 1980,]
ano3 <- mydf_filter[mydf_filter$year == 1998,]
ano4 <- mydf_filter[mydf_filter$year == 2015,]

```

2 - Criação dos gráficos e união.

```

# Criação de um tema global para o ggplot
theme_set(theme_grey() + theme(legend.box.background = element_rect(), legend.box.margin = margin(6, 6, 6, 6)))

p <- ggplot(ano, aes(fert, life, size = pop, color = continent)) +
  labs(x="Fertility Rate", y = "Life expectancy at birth (years)",
       caption = "(Based on data from Hans Rosling - gapminder.com)",
       color = 'Continent', size = "Population (millions)") +
  ylim(30,100) +
  geom_point() +
  scale_color_brewer(type = 'div', palette = 'Spectral') +
  ggtitle("Year: 1962")

p2 <- ggplot(ano2, aes(fert, life, size = pop, color = continent)) +
  labs(x="Fertility Rate", y = "Life expectancy at birth (years)",
       caption = "(Based on data from Hans Rosling - gapminder.com)",
       color = 'Continent', size = "Population (millions)") +
  ylim(30,100) +
  geom_point() +
  scale_color_brewer(type = 'div', palette = 'Spectral') +
  ggtitle("Year: 1980")

p3 <- ggplot(ano3, aes(fert, life, size = pop, color = continent)) +
  labs(x="Fertility Rate", y = "Life expectancy at birth (years)",
       caption = "(Based on data from Hans Rosling - gapminder.com)",
       color = 'Continent', size = "Population (millions)") +
  ylim(30,100) +
  geom_point() +
  scale_color_brewer(type = 'div', palette = 'Spectral') +
  ggtitle("Year: 1998")

p4 <- ggplot(ano4, aes(fert, life, size = pop, color = continent)) +
  labs(x="Fertility Rate", y = "Life expectancy at birth (years)",
       caption = "(Based on data from Hans Rosling - gapminder.com)",
       color = 'Continent', size = "Population (millions)") +
  ylim(30,100) +
  geom_point() +
  scale_color_brewer(type = 'div', palette = 'Spectral') +
  ggtitle("Year: 2015")

#grid.arrange(p, p2, p3, p4, nrow = 2, ncol = 2)
# Juntando os gráficos.
p12 <- cbind(ggplotGrob(p),
             ggplotGrob(p2), size="last")
p34 <- cbind(ggplotGrob(p3),
             ggplotGrob(p4), size="last")

allp <- rbind(p12, p34, size="last")

```

```
# Arrumando novamente a distribuição dos painéis.
panels <- p$layout$t[grep("panel", allp$layout$name)]

allp$heights[panels] <- unit(c(3, 1), "null")

grid::grid.newpage()
grid::grid.draw(allp)
```

## Problema 4: Atirei o pau no gráfico