

# Improving your Visualizations

## Dicas para melhorar as visualizações

Ainda utilizaremos a base do National Center for Health Statistics sobre tendências de mortalidade nos Estados Unidos ao longo dos anos que pode ser encontrada clicando aqui.

Importando o dataset

```
base <-  
  read.csv("https://data.cdc.gov/api/views/w9j2-ggv5/rows.csv?accessType=DOWNLOAD")
```

```
glimpse(base)
```

```
## Rows: 1,071  
## Columns: 5  
## $ Year      <int> 1900, 1901, 1902, 1903, 1904, 1905, 1906, 1907, 190~  
## $ Race      <chr> "All Races", "All Races", "All Races", "All Races", ~  
## $ Sex       <chr> "Both Sexes", "Both Sexes", "Both Sexes", "Both Sex~  
## $ Avg_Life_Expect <dbl> 47.3, 49.1, 51.5, 50.5, 47.6, 48.7, 48.7, 47.6, 51.~  
## $ Age_Adj_Death_Rate <dbl> 2518.0, 2473.1, 2301.3, 2379.0, 2502.5, 2423.7, 239~
```

## Visualizando diferentes informações no mesmo gráfico

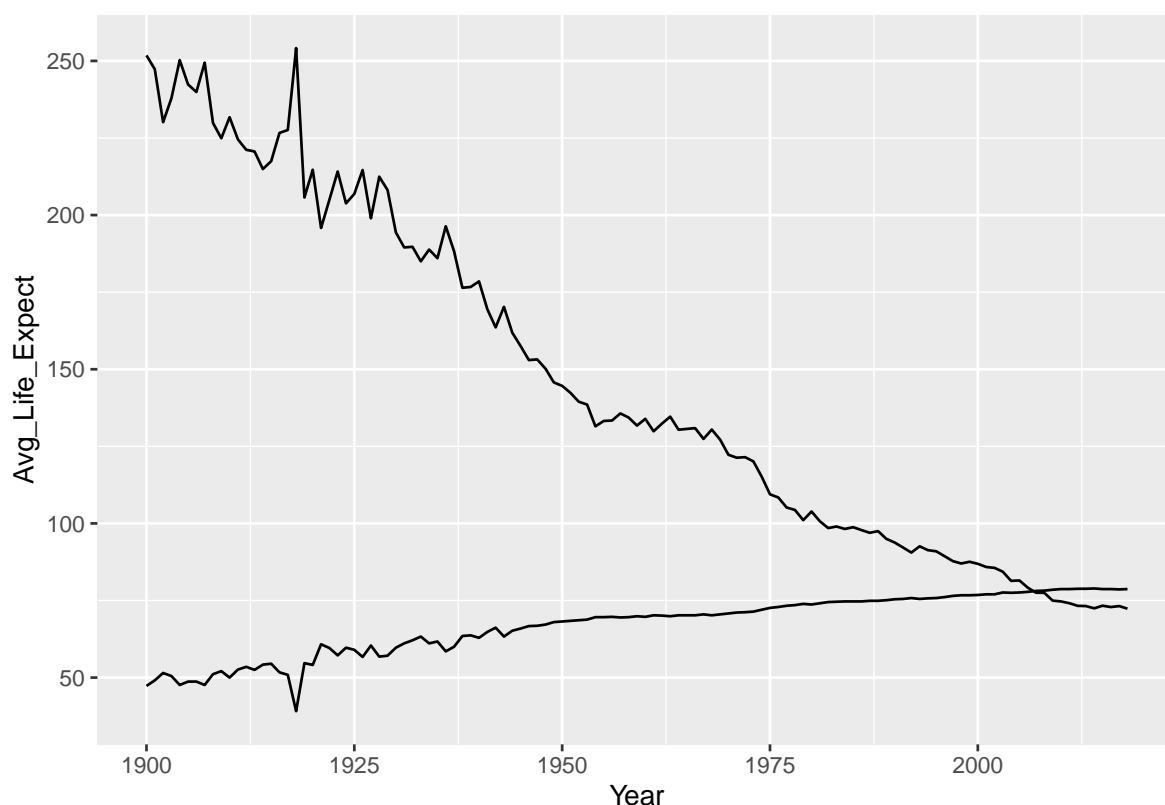
No último exercício, visualizamos a mesma informação, mas para públicos diferentes, nós verificamos a expectativa de vida para diferentes raças e gêneros. E caso a ideia seja ver duas informações diferentes como a expectativa de vida e a Taxa de Mortalidade?

Para começar vamos colocar a Taxa de Mortalidade na mesma escala da expectativa de vida para não distorcer o gráfico.

```
base$scaled_age_adj_death_rate <- base$Age_Adj_Death_Rate/10
```

Para o eixo x (horizontal) usamos a variável tempo, mas queremos conferir a variação de duas informações distintas ao longo desse tempo, ou seja, temos duas variáveis para y. Nesse caso basta chamarmos a função `geom_line` duas vezes e teremos duas linhas distintas.

```
base %>%  
  filter(Sex == "Both Sexes",  
         Race == "All Races") %>%  
  ggplot(aes(x=Year))+  
  geom_line(aes(y=Avg_Life_Expect))+  
  geom_line(aes(y=scaled_age_adj_death_rate))
```



O problema da visualização acima é que sem legenda, é impossível diferenciar o que cada linha representa.

Dessa forma uma alternativa é transformar a estrutura dos dados para conseguir utilizar as técnicas do exercício anterior. Vamos “tombar” os dados das colunas e transformar em linhas, e assim teremos uma coluna que vai indicar o tipo do dados se é a Expectativa de Vida ou a Taxa de Mortalidade, e uma coluna com os valores. Vai se comportar da mesma forma que a coluna de Raça e Gênero.

Para realizar essa reestruturação do dataframe, utilizamos o `pivot`. Existem 2 tipos de `pivot`:

- Wider - Um Dataset wide significa que tem muitas colunas, portanto usamos essa função para transformar linhas em colunas
- longer - Um Dataset longer significa que tem muitas linhas, portanto usamos essa função para transformar colunas em linhas

E assim o pivot longer é o que resolve nosso problema.

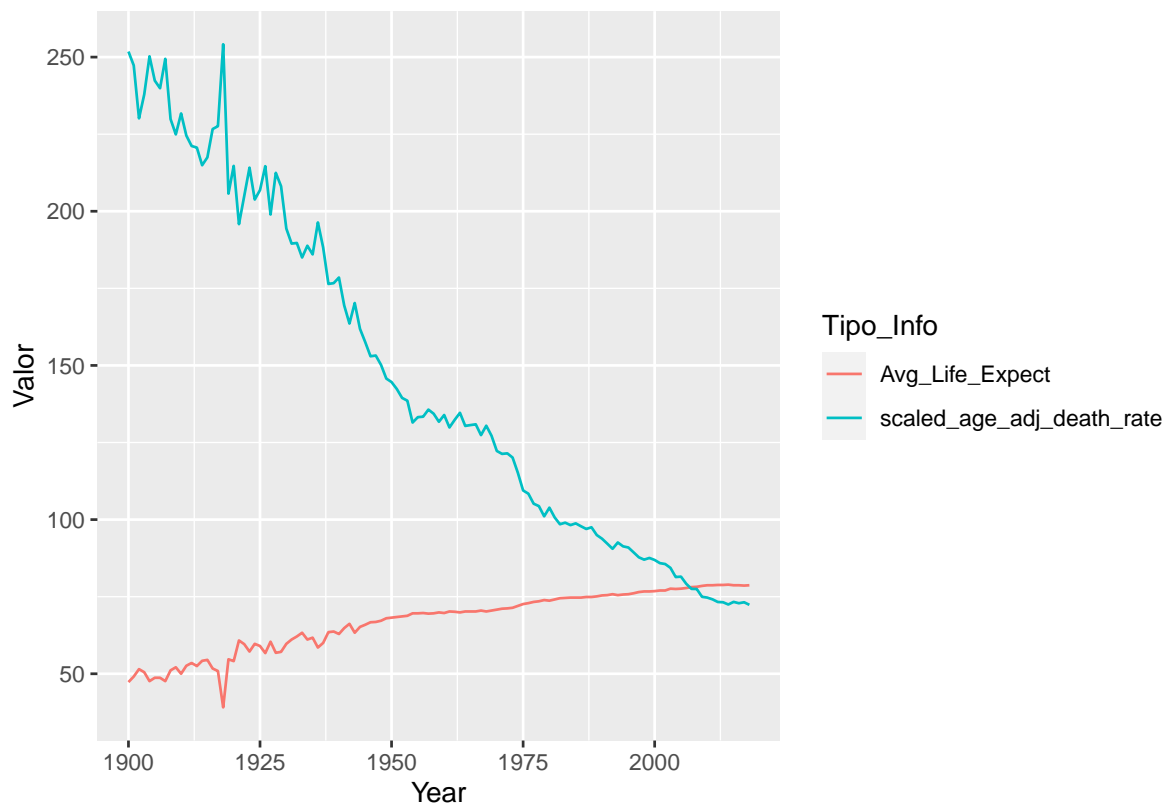
```
pivot <- base %>%
  pivot_longer(c("Avg_Life_Expect", "scaled_age_adj_death_rate"),
               names_to = "Tipo_Info",
               values_to = "Valor") %>%
  select(-Age_Adj_Death_Rate)

pivot %>% head()
```

```
## # A tibble: 6 x 5
##   Year Race      Sex      Tipo_Info      Valor
##   <int> <chr>    <chr>    <chr>      <dbl>
## 1  1900 All Races Both Sexes Avg_Life_Expect      47.3
## 2  1900 All Races Both Sexes scaled_age_adj_death_rate 252.
## 3  1901 All Races Both Sexes Avg_Life_Expect      49.1
## 4  1901 All Races Both Sexes scaled_age_adj_death_rate 247.
## 5  1902 All Races Both Sexes Avg_Life_Expect      51.5
## 6  1902 All Races Both Sexes scaled_age_adj_death_rate 230.
```

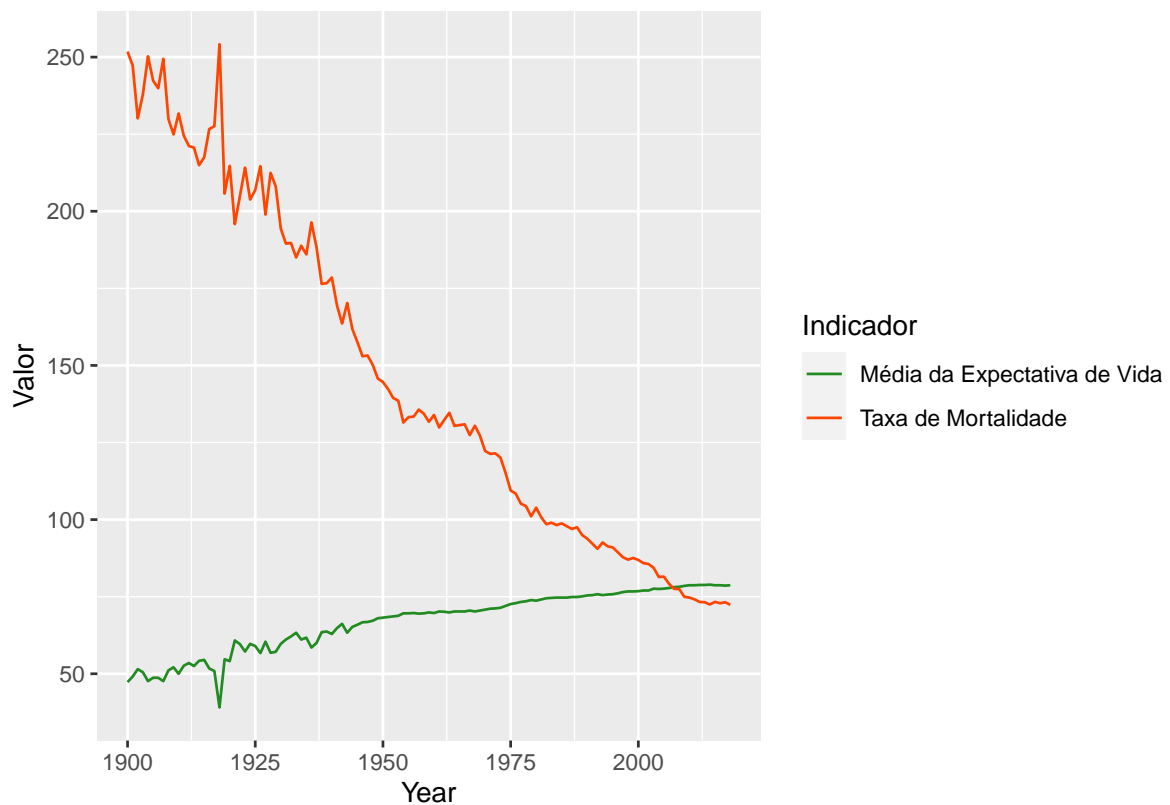
Agora sim, podemos separar as informações por cor ou tipo de linha

```
pivot %>%
  filter(Sex == "Both Sexes",
         Race == "All Races") %>%
  ggplot(aes(x=Year, y=Valor, color=Tipo_Info))+
  geom_line()
```



Para melhorar ainda mais a clareza das informações, vamos estiliza-lo

```
pivot %>%
  filter(Sex == "Both Sexes",
         Race == "All Races") %>%
  ggplot(aes(x=Year, y=Valor, color=Tipo_Info))+
  geom_line()+
  scale_color_manual(
    name="Indicador", #Título das legendas
    values = c("forestgreen", "orangered"), #cores
    breaks = c("Avg_Life_Expect",
               "scaled_age_adj_death_rate"), #colunas utilizadas
    labels = c("Média da Expectativa de Vida",
               "Taxa de Mortalidade") # legenda a exibir
  )
```

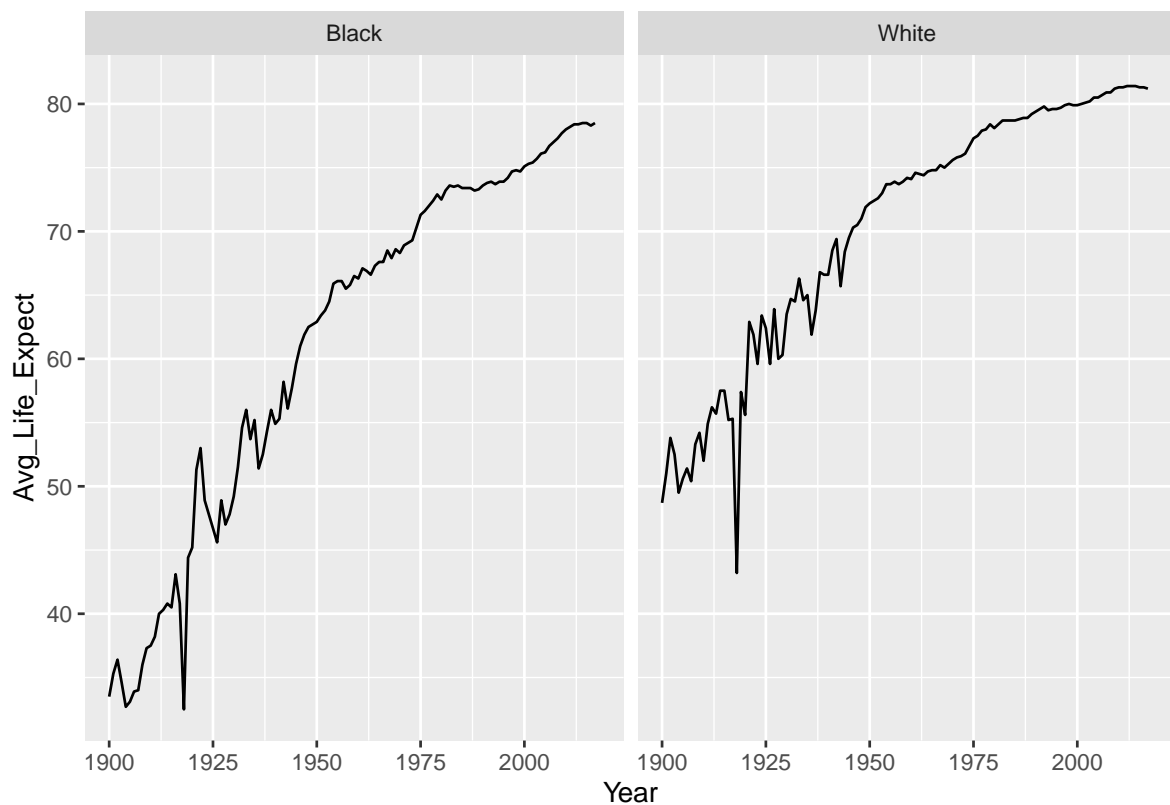


## Comparando na mesma imagem em gráficos distintos

Nem sempre fará sentido colocar diferentes informações dentro de um mesmo gráfico, apesar da necessidade de visualizar na mesma figura. Sendo assim podemos adicionar uma camada ao gráfico que divide automaticamente em 2 gráficos.

```
base %>%
  filter(Sex == "Female",
         Race != "All Races") %>%
  ggplot(aes(x=Year, y=Avg_Life_Expect))+
  geom_line()+
  facet_wrap(vars(Race))
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



Para mais de 2 gráficos, uma boa opção é a função `face_grid`

```
base %>%
  filter(Sex != "Both Sexes",
         Race != "All Races") %>%
  ggplot(aes(x=Year, y=Avg_Life_Expect))+
  geom_line()+
  facet_grid(
    rows = vars(Sex),
    cols = vars(Race)
  )
```

## Warning: Removed 1 row(s) containing missing values (geom\_path).

