

Visualizing Relationship Between Variables

Visualizando múltiplas informações num único gráfico

Com o objetivo de explorar a relação entre as informações, vamos novamente utilizar 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.

Bibliotecas

```
library(readr)
library(dplyr)
library(ggplot2)
```

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, 19~
## $ Race                <chr> "All Races", "All Races", "All Races", ~
## $ Sex                 <chr> "Both Sexes", "Both Sexes", "Both Sexe~
## $ Average.Life.Expectancy..Years. <dbl> 47.3, 49.1, 51.5, 50.5, 47.6, 48.7, 48~
## $ Age.adjusted.Death.Rate <dbl> 2518.0, 2473.1, 2301.3, 2379.0, 2502.5~
```

Campo	Significado
Year	Ano de Nascimento
Race	A raça da população avaliada

	Campo	Significado
	Sex	Sexo da população avaliada
Average.Life.Expectancy..Years.		A expectativa de vida em anos dado o ano do nascimento
Age.adjusted.Death.Rate		Taxa de mortalidade ajustada por idade de pessoas nascidas em um determinado ano

Padronizando nome dos campos

```
names(base)[4] <- "Avg_Life_Expect"
names(base)[5] <- "Age_Adj_Death_Rate"

names(base)
```

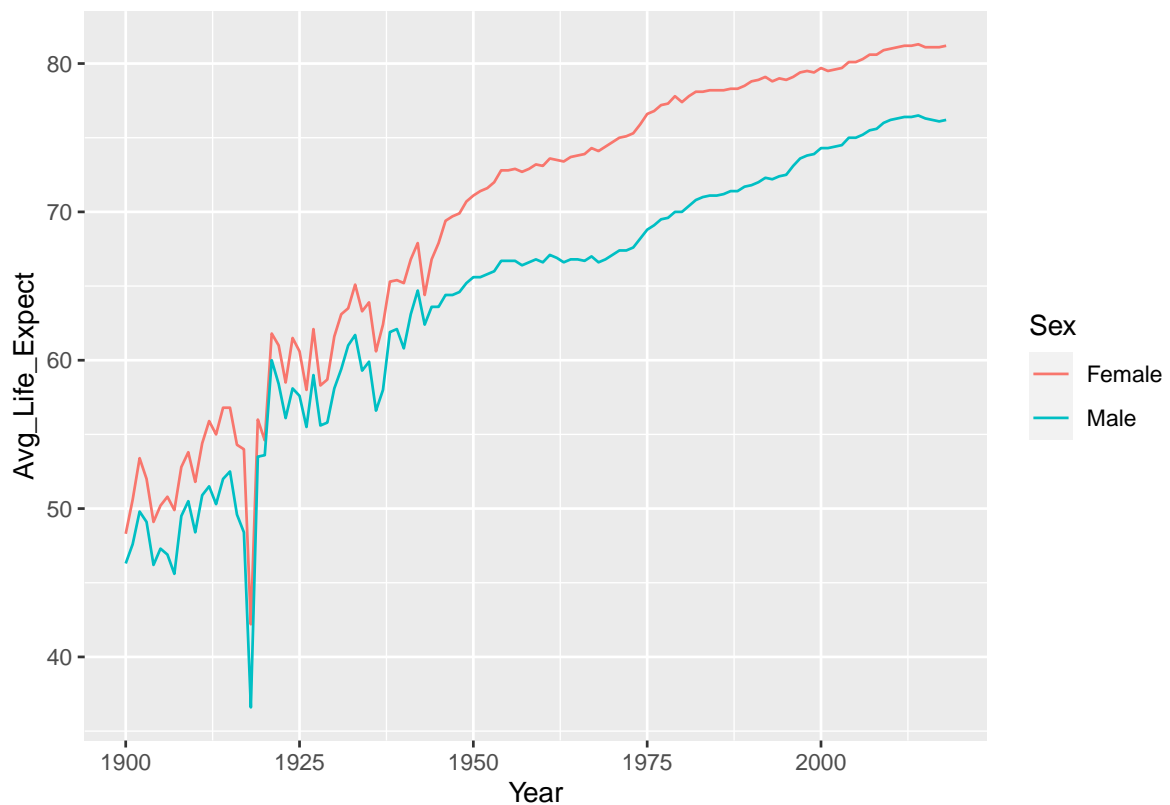
```
## [1] "Year"          "Race"          "Sex"
## [4] "Avg_Life_Expect" "Age_Adj_Death_Rate"
```

Diferenciando por cor

Acima vimos que a base possui informações da média de anos de vida a cada ano de acordo com gênero e raça. No primeiro exercício fizemos gráficos sem distinguir essas informações, já aqui queremos compara-las.

```
base_sex <- base %>% filter(Sex %in% c("Female",
                                         "Male"),
                           Race == "All Races")

base_sex %>%
  ggplot(aes(x=Year, y=Avg_Life_Expect, color= Sex)) +
  geom_line()
```



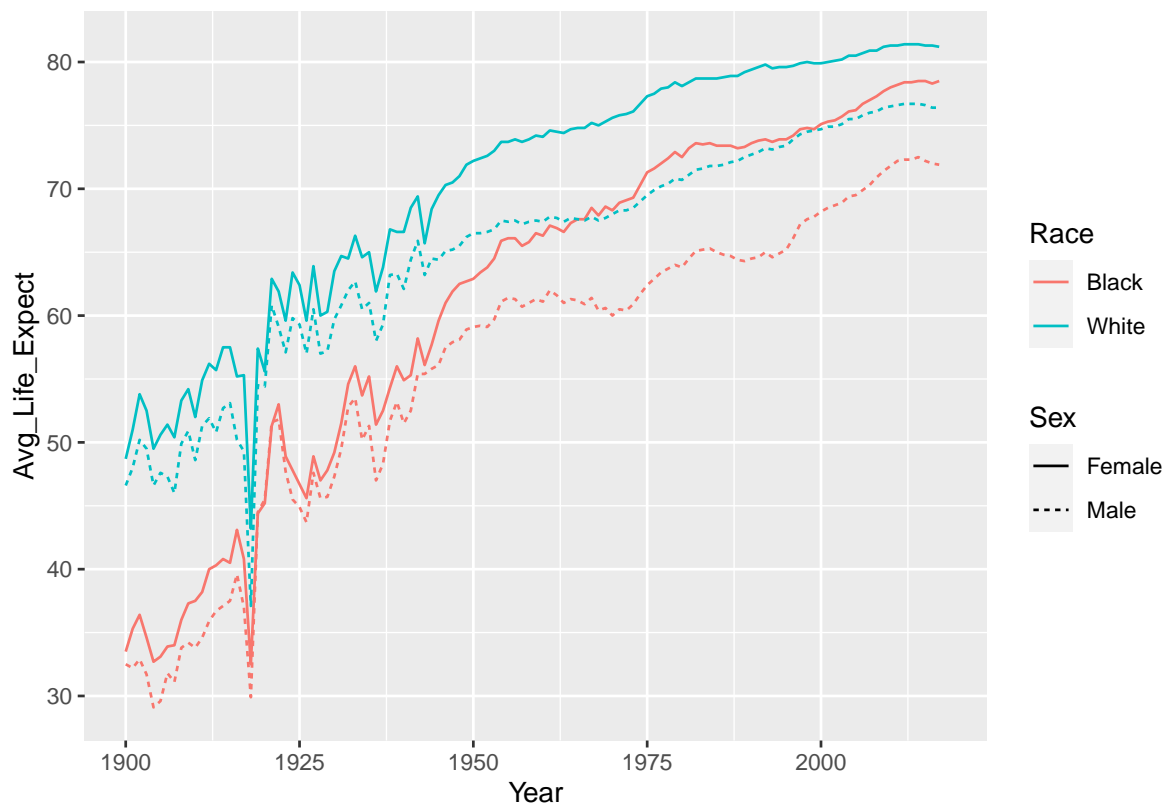
Diferenciando por tipo de linha

Aqui utilizei o elemento cor para diferenciar a Raça, enquanto usei tipos de linhas distintos para o Gênero.

```
base_sex_race <- base %>% filter(Sex %in% c("Female",
                                             "Male"),
                                Race %in% c("Black",
                                             "White"))

base_sex_race %>%
  ggplot(aes(x=Year, y=Avg_Life_Expect, color= Race, lty=Sex)) +
  geom_line()
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
```



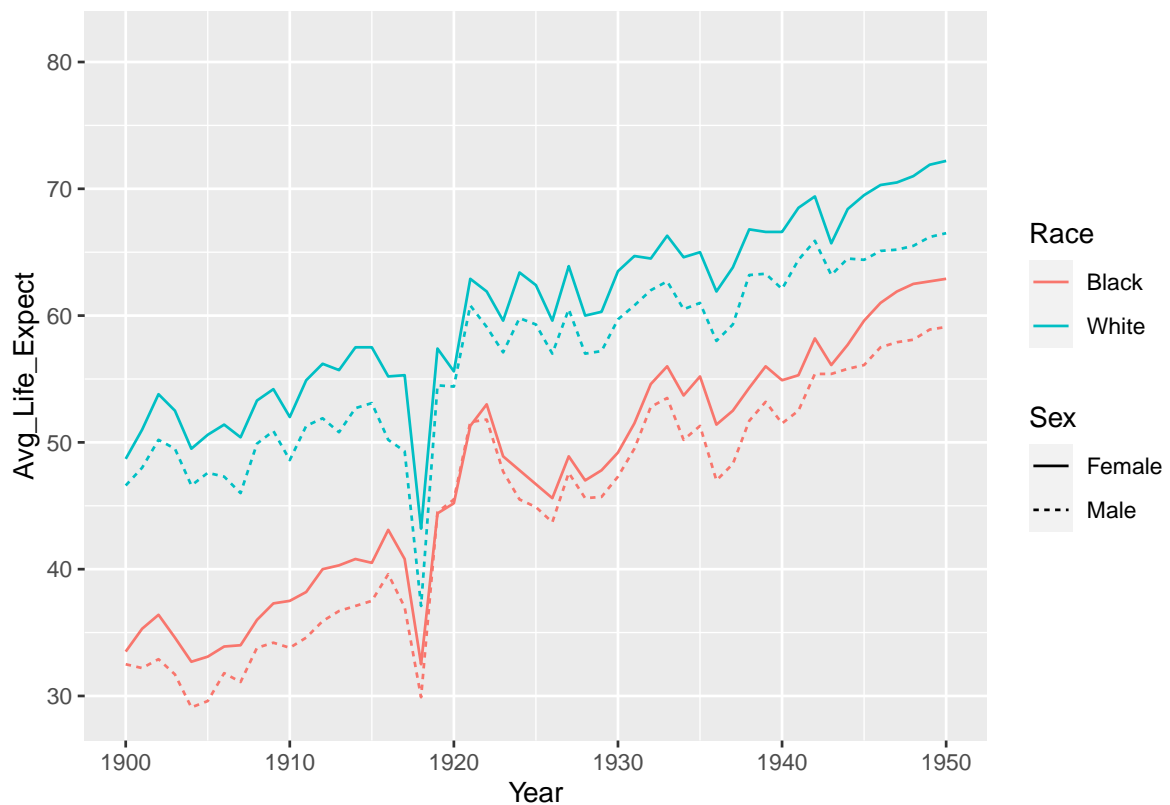
Dando zoom no gráfico

Ou em outras palavras, alterando os limites das extremidades. A visualização do gráfico por volta de 1920 ficou um pouco ruim, e seria legal “dar um zoom” para entender melhor. Uma alternativa é filtrar esse período no tempo e criar um gráfico a partir de um novo objeto.

No entanto uma alternativa é adicionar uma camada ao gráfico que determina os limites dos eixos.

```
base_sex_race %>%
  ggplot(aes(x=Year,y=Avg_Life_Expect, color= Race, lty=Sex)) +
  geom_line() +
  xlim(1900,1950)
```

```
## Warning: Removed 272 row(s) containing missing values (geom_path).
```

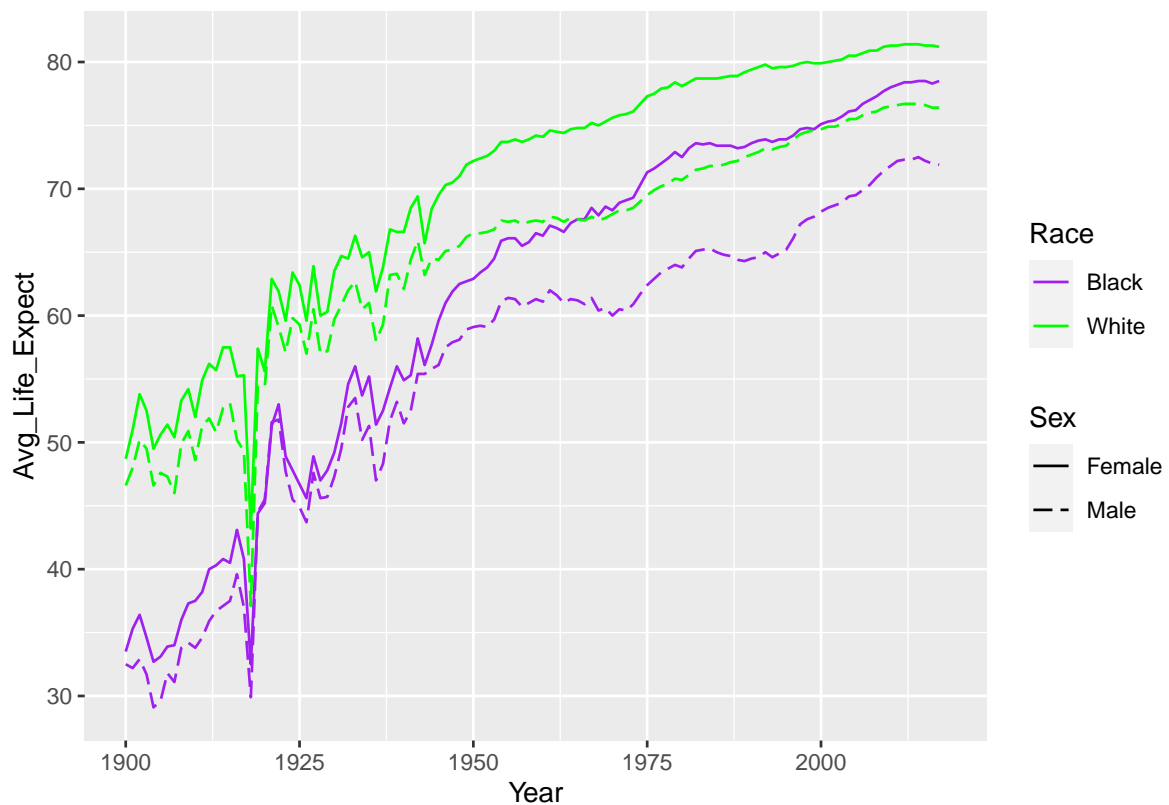


Melhorando a estilização do gráfico

As cores e tipos de linhas default podem não ser a melhor escolha, por isso é possível adicionar uma camada no gráfico para atribuir o tema que achar melhor.

```
base_sex_race %>%
  ggplot(aes(x=Year, y=Avg_Life_Expect, color= Race, lty=Sex)) +
  geom_line() +
  scale_color_manual(values = c("purple", "green")) +
  scale_linetype_manual(values = c(1,5))
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
```



Temos 7 opções de linhas:

0. blank
1. solid
2. dashed
3. dotted
4. dotdash
5. longdash
6. twodash

E para saber as cores disponíveis basta executar a função abaixo

```
colors() %>% head(50)
```

```
## [1] "white" "aliceblue" "antiquewhite" "antiquewhite1"
## [5] "antiquewhite2" "antiquewhite3" "antiquewhite4" "aquamarine"
## [9] "aquamarine1" "aquamarine2" "aquamarine3" "aquamarine4"
## [13] "azure" "azure1" "azure2" "azure3"
## [17] "azure4" "beige" "bisque" "bisque1"
## [21] "bisque2" "bisque3" "bisque4" "black"
## [25] "blanchedalmond" "blue" "blue1" "blue2"
## [29] "blue3" "blue4" "blueviolet" "brown"
```

```
## [33] "brown1"          "brown2"          "brown3"          "brown4"
## [37] "burlywood"       "burlywood1"      "burlywood2"      "burlywood3"
## [41] "burlywood4"      "cadetblue"       "cadetblue1"      "cadetblue2"
## [45] "cadetblue3"      "cadetblue4"      "chartreuse"       "chartreuse1"
## [49] "chartreuse2"     "chartreuse3"
```

Scatter Plot

Nem sempre queremos conectar os dados através linhas, muitas vezes estamos tentando descobrir se existe de fato relação entre as informações, a linha preenche um gap que nem sabemos se existe.

O scatter plot é a melhor alternativa, pois literalmente marca os pontos no gráfico respeitando as coordenadas x e y, e assim através dos pontos apenas conseguimos observar se há uma tendência, se traça uma “linha” só pelo acúmulo de pontos.

```
base_sex %>%
  ggplot(aes(x=Year,y=Avg_Life_Expect, color=Sex)) +
  geom_point()
```

