

Projeto PIB

Rafael Emerick e Hugo Gravatá

2025-02-02

Bibliotecas, diretório e abrindo arquivos

```
# Biblioteca  
library(tidyverse)
```

```
## Warning: pacote 'tidyverse' foi compilado no R versão 4.4.3
```

```
## Warning: pacote 'ggplot2' foi compilado no R versão 4.4.3
```

```
## Warning: pacote 'tibble' foi compilado no R versão 4.4.3
```

```
## Warning: pacote 'dplyr' foi compilado no R versão 4.4.3
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —  
## ✓ dplyr      1.1.4      ✓ readr      2.1.5  
## ✓ forcats   1.0.0      ✓ stringr    1.5.1  
## ✓ ggplot2    3.5.2      ✓ tibble     3.3.0  
## ✓ lubridate 1.9.4      ✓ tidyr      1.3.1  
## ✓ purrr     1.0.2  
## — Conflicts — tidyverse_conflicts() —  
## ✗ dplyr::filter() masks stats::filter()  
## ✗ dplyr::lag()     masks stats::lag()  
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be  
come errors
```

```
library(geobr)
```

```
## Carregando namespace exigido: sf
```

```
library(descr)  
options(scipen = 999) #tirando notação científica  
  
# Mudando diretório  
getwd()
```

```
## [1] "C:/Users/aemer/OneDrive/Documentos/Faculdade/Econdata Projetos/Projeto PIB"
```

```
setwd("C:\\Users\\aemer\\OneDrive\\Documentos\\Faculdade\\Econdata Projetos\\Projeto PIB")
```

```
# Leitura de dados
```

```
PIB_Municipio = read.csv("base_projeto_pib_municipio.csv")
```

```
str(PIB_Municipio)
```

```
## 'data.frame': 105756 obs. of 17 variables:
```

```
## $ id_municipio : int 1100023 1100023 1100023 1100023 1100023 1100023 1100023 1100023 1100023 1100023 ...
```

```
## $ sigla_uf : chr "RO" "RO" "RO" "RO" ...
```

```
## $ ano : int 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
```

```
...
```

```
## $ pib : num 449592816 539636214 657193231 749021187 790696634
```

```
...
```

```
## $ impostos_liquidos : num 42665972 53927649 68381216 85363362 10042072 ...
```

```
## $ va : num 406926845 485708564 588812015 663657826 690275914
```

```
...
```

```
## $ va_agropecuaria : num 28590178 47307744 46852784 46367075 42170266 ...
```

```
## $ va_industria : num 99922936 107718377 141387012 139521712 15251806
```

```
...
```

```
## $ va_servicos : num 161851549 201088747 251591284 298604412 29184008
```

```
...
```

```
## $ va_adespss : num 116562182 129593695 148980934 179164627 203747507
```

```
...
```

```
## $ ideb_ensino_publico_fundamental: num NA NA NA 3.5 NA 3.2 NA 3.5 NA 4 ...
```

```
## $ expectativa_vida_2010 : num 73.4 73.4 73.4 73.4 73.4 ...
```

```
## $ indice_gini_2010 : num 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53
```

```
...
```

```
## $ renda_pc_2010 : num 690 690 690 690 690 ...
```

```
## $ indice_escolaridade_2010 : num 0.514 0.514 0.514 0.514 0.514 0.514 0.514 0.514 0.514 0.514 ...
```

```
## $ indice_frequencia_escolar_2010 : num 0.648 0.648 0.648 0.648 0.648 0.648 0.648 0.648 0.648 0.648 ...
```

```
## $ idhm_2010 : num 0.702 0.702 0.702 0.702 0.702 0.702 0.702 0.702 0.702 0.702 ...
```

```
PIB_UF = read.csv("base_projeto_pib_uf.csv")
```

```
str(PIB_UF)
```

```
## 'data.frame':    513 obs. of  17 variables:
## $ ano              : int  2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
...
## $ id_uf            : int  11 11 11 11 11 11 11 11 11 11 ...
## $ sigla_uf         : chr  "RO" "RO" "RO" "RO" ...
## $ pib              : num  7467629886 9425010486 11004641436 12511821181 130
54713344 ...
## $ impostos_liquidos : num  839731192 1108434936 1288806654 1476144194 161380
9974 ...
## $ va               : num  6627898698 8316575548 9715834778 11035676990 1144
0903374 ...
## $ va_agropecuaria  : num  715526872 1278658831 1288515348 1342222120 123800
6193 ...
## $ va_industria     : num  1191090432 1216605061 1674933817 1887932121 22106
92147 ...
## $ va_servicos      : num  2484579193 3376727040 3986529419 4603783904 43205
26746 ...
## $ va_adespss       : num  2236702207 2444584625 2765856199 3201738843 36716
78293 ...
## $ ideb_ensino_publico_fundamental: num  NA NA NA 3.1 NA 3.3 NA 3.5 NA 3.6 ...
## $ expectativa_vida_2010          : num  73 73 73 73 73 ...
## $ indice_gini_2010              : num  0.56 0.56 0.56 0.56 0.56 0.56 0.56 0.56 0.56 0.56
...
## $ renda_pc_2010                 : num  671 671 671 671 671 ...
## $ indice_escolaridade_2010      : num  0.48 0.48 0.48 0.48 0.48 0.48 0.48 0.48 0.48 0.48
...
## $ indice_frequencia_escolar_2010 : num  0.633 0.633 0.633 0.633 0.633 0.633 0.633 0.633 0.633
0.633 0.633 ...
## $ idhm_2010                    : num  0.69 0.69 0.69 0.69 0.69 0.69 0.69 0.69 0.69 0.69
...
```

Objetivo do PIB

1)

```

### 1)

# Filtrar dados para apenas ano inicial e final
PibDelta = PIB_UF %>%
  filter(ano %in% c(2002, 2020))

# Filtrar dados de PIB e VA
PibImpacto = PibDelta %>%
  select(ano, id_uf, sigla_uf, pib, impostos_liquidos, va, va_agropecuaria, va_industria, va_servicos, va_adespss)

# Calcular impacto das variáveis em %
PibImpacto$PctImposto = round((PibDelta$impostos_liquidos / PibDelta$pib) * 100, 2)
PibImpacto$PctAgropecuaria = round((PibDelta$va_agropecuaria / PibDelta$pib) * 100, 2)
PibImpacto$PctIndustria = round((PibDelta$va_industria / PibDelta$pib) * 100, 2)
PibImpacto$PctServicos = round((PibDelta$va_servicos / PibDelta$pib) * 100, 2)
PibImpacto$PctAdespss = round((PibDelta$va_adespss / PibDelta$pib) * 100, 2)

# Filtrar apenas as %
PibImpacto = PibImpacto %>%
  select(ano, sigla_uf, PctAgropecuaria, PctImposto, PctIndustria, PctServicos, PctAdespss)

# Separar os anos para melhor visualização
PibImpacto2002 = PibImpacto %>%
  filter(ano == 2002)

PibImpacto2020 = PibImpacto %>%
  filter(ano == 2020)

## Plot

# 2002
# Criar um mapeamento de estados para regiões
regioes = data.frame(
  sigla_uf = c("RO", "AC", "AM", "RR", "PA", "AP", "TO",
               "MA", "PI", "CE", "RN", "PB", "PE", "AL", "SE", "BA",
               "MG", "ES", "RJ", "SP",
               "PR", "SC", "RS",
               "MS", "MT", "GO", "DF"),
  regioao = c(rep("Norte", 7), rep("Nordeste", 9),
               rep("Sudeste", 4), rep("Sul", 3),
               rep("Centro-Oeste", 4))
)

# Adicionar a coluna de região
PibImpacto2002 = PibImpacto2002 %>%
  left_join(regioes, by = "sigla_uf")

# Transformar os dados para formato longo (Sem isso não empilhava os dados)
data_long = PibImpacto2002 %>%
  pivot_longer(cols = starts_with("Pct"),
               names_to = "Componente",
               values_to = "Percentual") %>%
  mutate(Componente = recode(Componente,

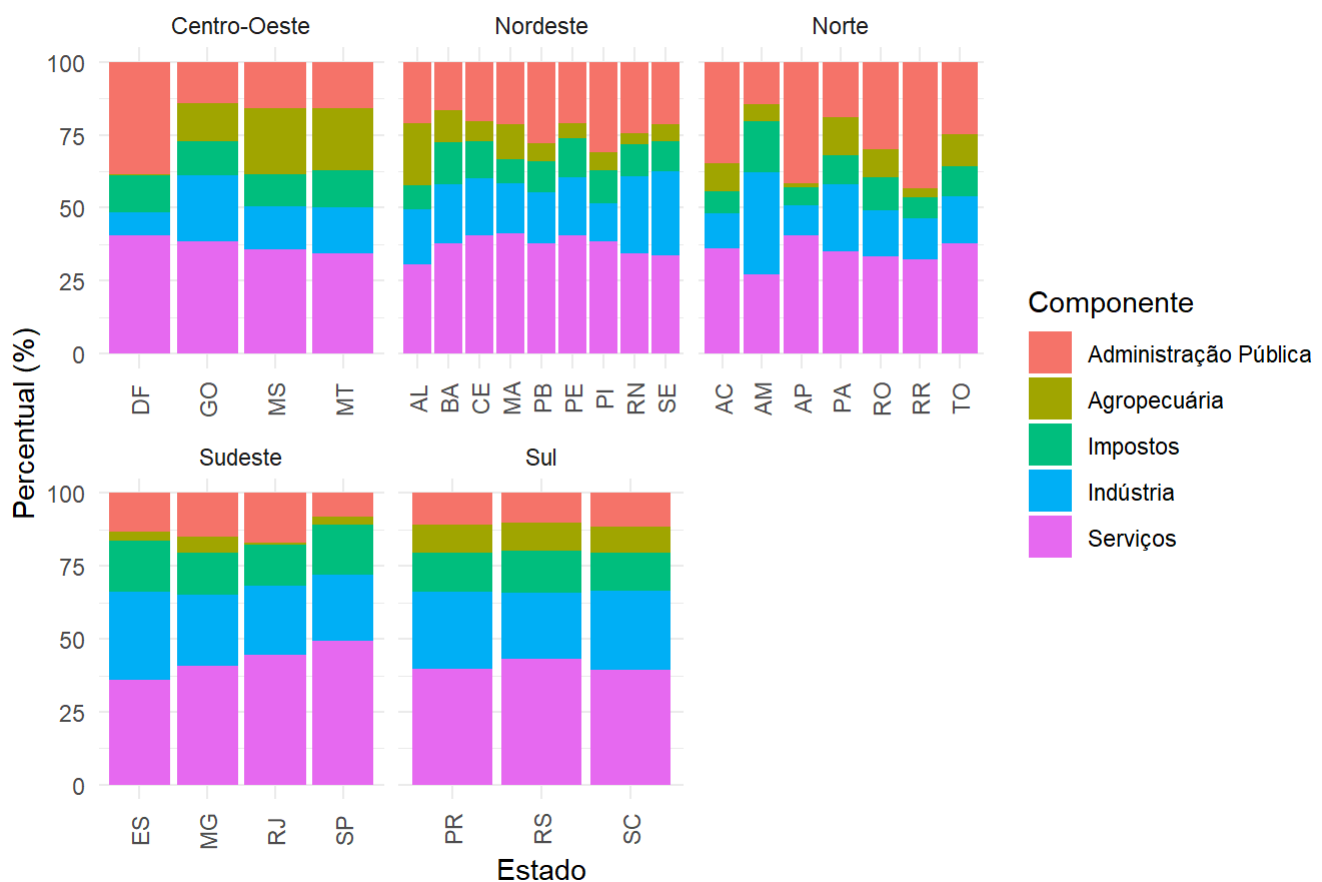
```

```
PctAgropecuaria = "Agropecuária",
PctImposto = "Impostos",
PctIndustria = "Indústria",
PctServicos = "Serviços",
PctAdespss = "Administração Pública"))
```

```
# Criar o gráfico
```

```
ggplot(data_long, aes(x = sigla_uf, y = Percentual, fill = Componente)) +
  geom_bar(stat = "identity", position = "stack") +
  facet_wrap(~ regioao, scales = "free_x", nrow = 2) +
  labs(title = "Porcentagem dos Componentes do PIB por Estado 2002",
       x = "Estado",
       y = "Percentual (%)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```

Porcentagem dos Componentes do PIB por Estado 2002



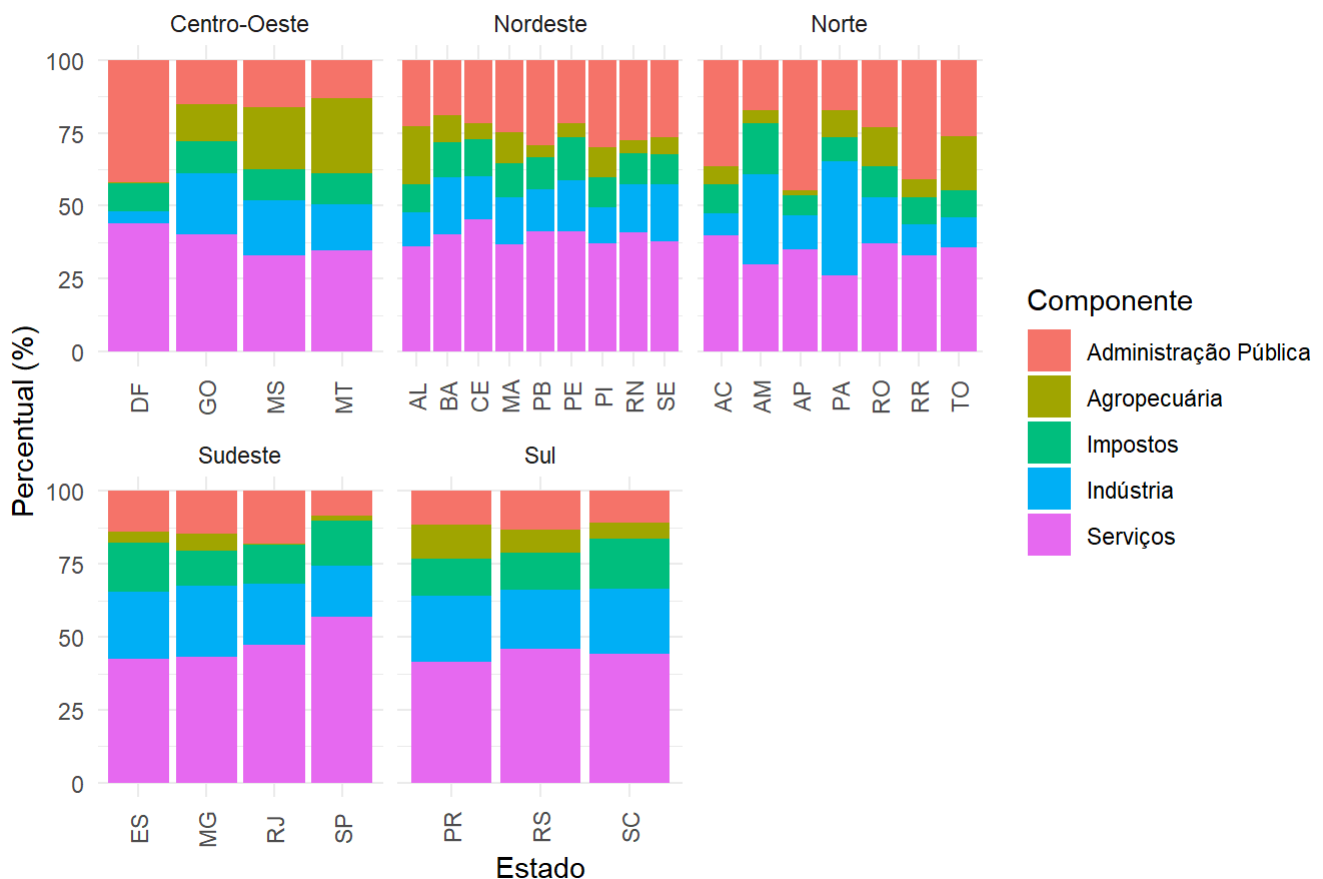
```
# 2020

# Adicionar a coluna de região
PibImpacto2020 = PibImpacto2020 %>%
  left_join(regioes, by = "sigla_uf")

# Transformar os dados para formato Longo (Sem isso não empilhava os dados)
data_long2 = PibImpacto2020 %>%
  pivot_longer(cols = starts_with("Pct"),
               names_to = "Componente",
               values_to = "Percentual") %>%
  mutate(Componente = recode(Componente,
                             PctAgroPecuararia = "Agropecuária",
                             PctImposto = "Impostos",
                             PctIndustria = "Indústria",
                             PctServicos = "Serviços",
                             PctAdespss = "Administração Pública"))

# Criar o gráfico
ggplot(data_long2, aes(x = sigla_uf, y = Percentual, fill = Componente)) +
  geom_bar(stat = "identity", position = "stack") +
  facet_wrap(~ regioao, scales = "free_x", nrow = 2) +
  labs(title = "Porcentagem dos Componentes do PIB por Estado 2020",
       x = "Estado",
       y = "Percentual (%)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```

Porcentagem dos Componentes do PIB por Estado 2020



```
### 2)

## Calcular PIB total

# 2002
Pib2002 = PibDelta %>%
  filter(ano == 2002)

Pib2002 = Pib2002 %>%
  mutate(PibTotal = sum(Pib2002$pib))

# 2020
Pib2020 = PibDelta %>%
  filter(ano == 2020)

Pib2020 = Pib2020 %>%
  mutate(PibTotal = sum(Pib2020$pib))

## Calcular impacto das variáveis em %

# 2002
Pib2002$PctPIB = round((Pib2002$pib / Pib2002$PibTotal) * 100, 2)

# 2020
Pib2020$PctPIB = round((Pib2020$pib / Pib2020$PibTotal) * 100, 2)

## Filtrar apenas o que interessa

Pib2002 = Pib2002 %>%
  select(ano, sigla_uf, pib, PctPIB)

Pib2020 = Pib2020 %>%
  select(ano, sigla_uf, pib, PctPIB)

## Mapa de calor do Brasil

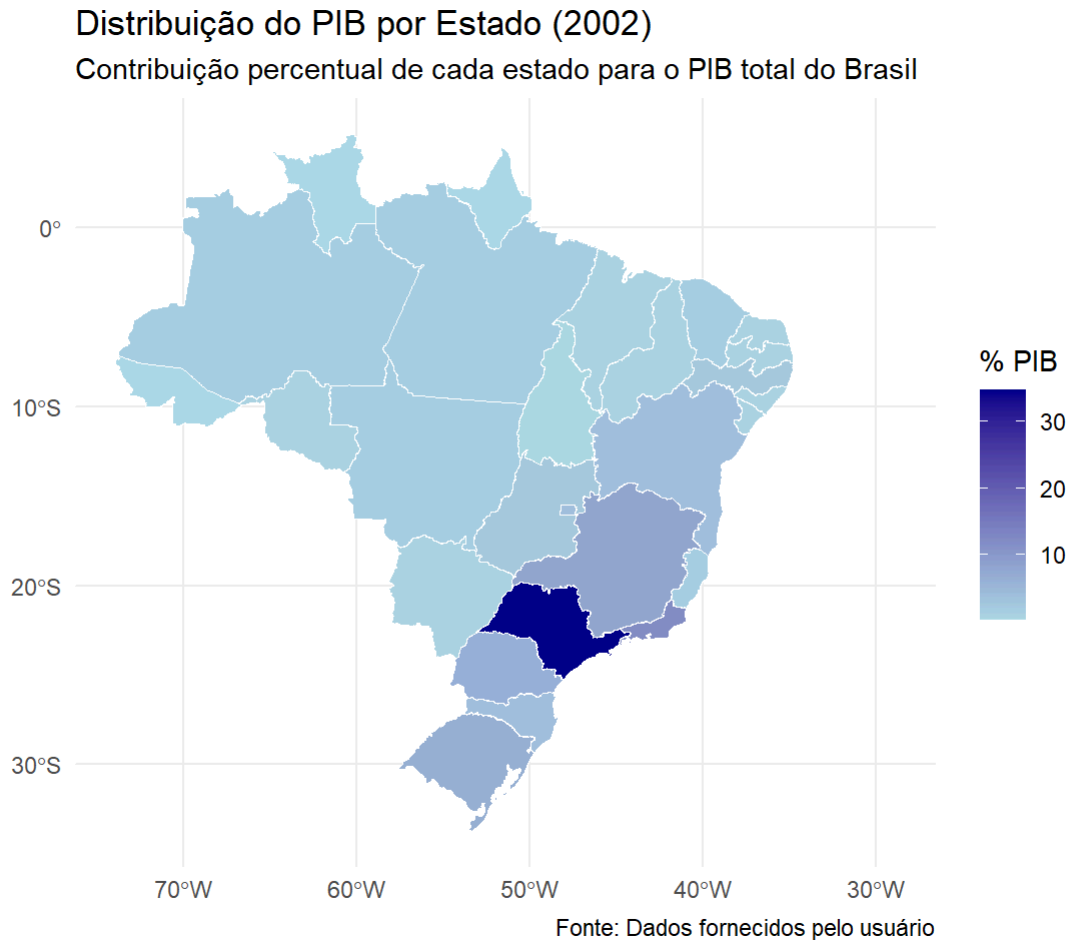
# 2002

# Baixar o mapa dos estados brasileiros
MapaEstados = read_state(code_state = "all", year = 2020)
```

```
## Using year/date 2020
```

```
# Unir os dados do PIB com o mapa
MapaPib = MapaEstados %>%
  left_join(Pib2002, by = c("abbrev_state" = "sigla_uf"))

# Criar o mapa de calor
ggplot(MapaPib) +
  geom_sf(aes(fill = PctPIB), color = "white") +
  scale_fill_gradient(low = "lightblue", high = "darkblue", name = "% PIB") +
  theme_minimal() +
  labs(title = "Distribuição do PIB por Estado (2002)",
        subtitle = "Contribuição percentual de cada estado para o PIB total do Brasil",
        caption = "Fonte: Dados fornecidos pelo usuário")
```

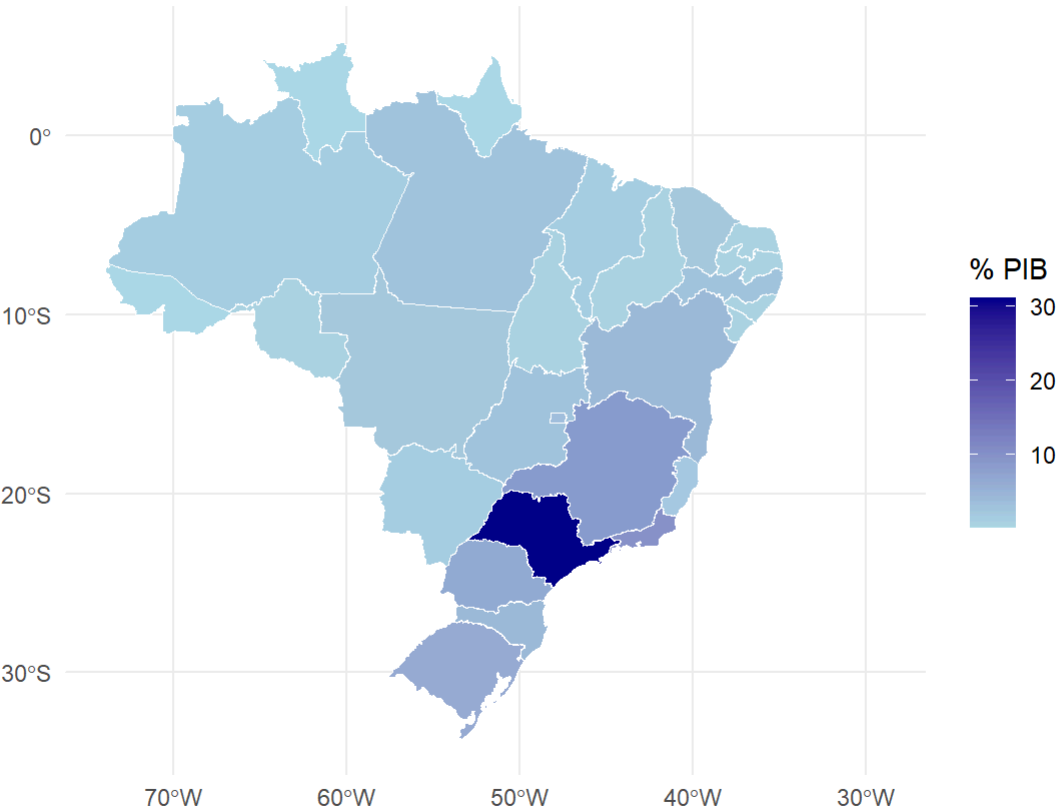


```
# 2020

# Unir os dados do PIB com o mapa
MapaPib2 = MapaEstados %>%
  left_join(Pib2020, by = c("abbrev_state" = "sigla_uf"))

# Criar o mapa de calor
ggplot(MapaPib2) +
  geom_sf(aes(fill = PctPIB), color = "white") +
  scale_fill_gradient(low = "lightblue", high = "darkblue", name = "% PIB") +
  theme_minimal() +
  labs(title = "Distribuição do PIB por Estado (2020)",
        subtitle = "Contribuição percentual de cada estado para o PIB total do Brasil",
        caption = "Fonte: Dados fornecidos pelo usuário")
```


Distribuição do PIB por Estado (2020)
Contribuição percentual de cada estado para o PIB total do Brasil



Fonte: Dados fornecidos pelo usuário

3)

```
### 3)

# Filtrar dados apenas para comparação entre PIB e imposto

PibImposto = PIB_UF %>%
  select(ano, id_uf, sigla_uf, pib, impostos_liquidos, va)

# Analisar diferença de aumento no Va com o do Imposto

PibImposto$DiferençaImpostoVa = PibImposto$va - PibImposto$impostos_liquidos

# Contribuição para o Pib em % de imposto e Va
PibImposto$PctImposto = round((PibImposto$impostos_liquidos / PibImposto$pib) * 100, 2)
PibImposto$PctVa = round((PibImposto$va / PibImposto$pib) * 100, 2)

## Plot

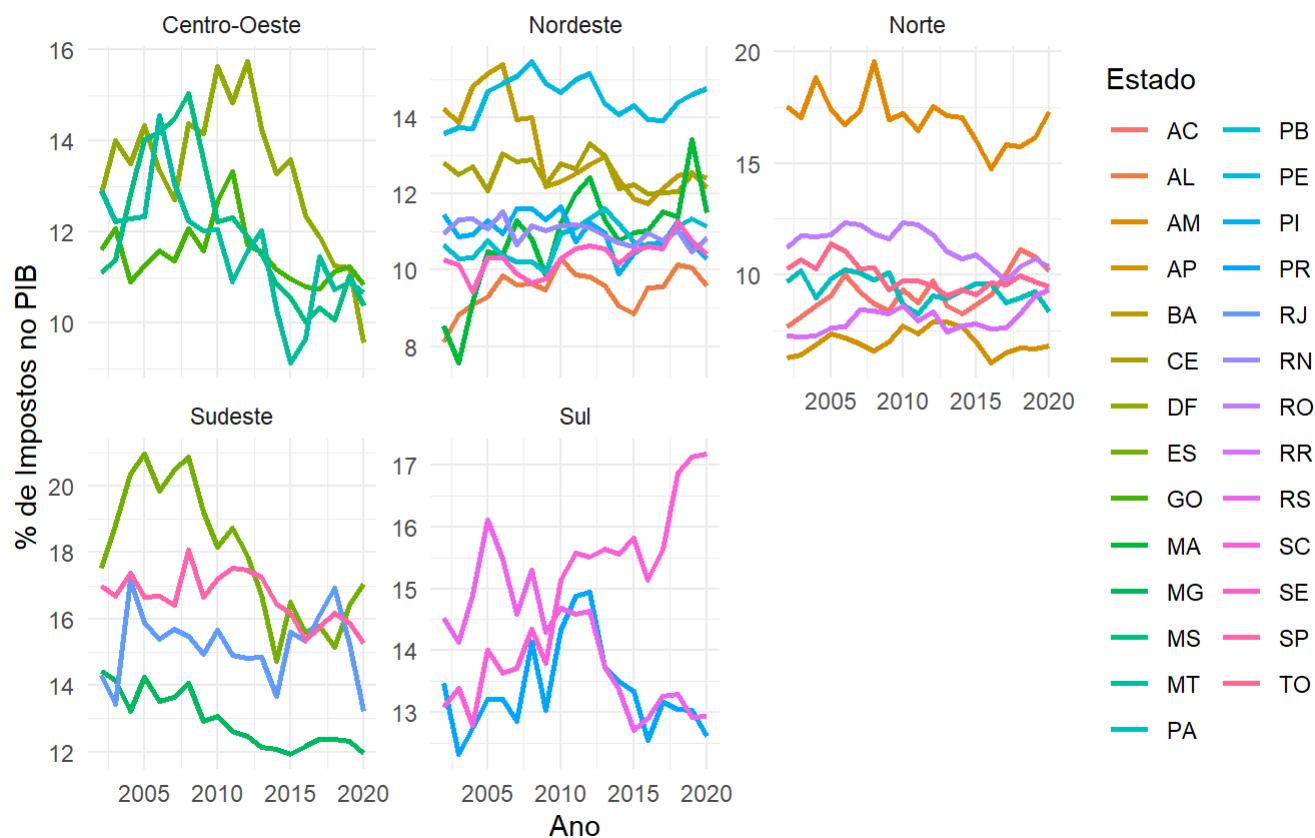
# Adicionar a coluna de região
PibImposto = PibImposto %>%
  left_join(regioes, by = "sigla_uf")

# Criar o gráfico da evolução da porcentagem de impostos por estado e região
ggplot(PibImposto, aes(x = ano, y = PctImposto, color = sigla_uf, group = sigla_uf)) +
  geom_line(size = 1) +
  facet_wrap(~ regioao, scales = "free_y") +
  theme_minimal() +
  labs(title = "Evolução da Porcentagem de Impostos no PIB por Estado",
        subtitle = "Analisando possíveis impactos na dinâmica do PIB",
        x = "Ano",
        y = "% de Impostos no PIB",
        color = "Estado")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

Evolução da Porcentagem de Impostos no PIB por Estado

Analisando possíveis impactos na dinâmica do PIB



Aprofundamento da análise

1)

```
### 1)

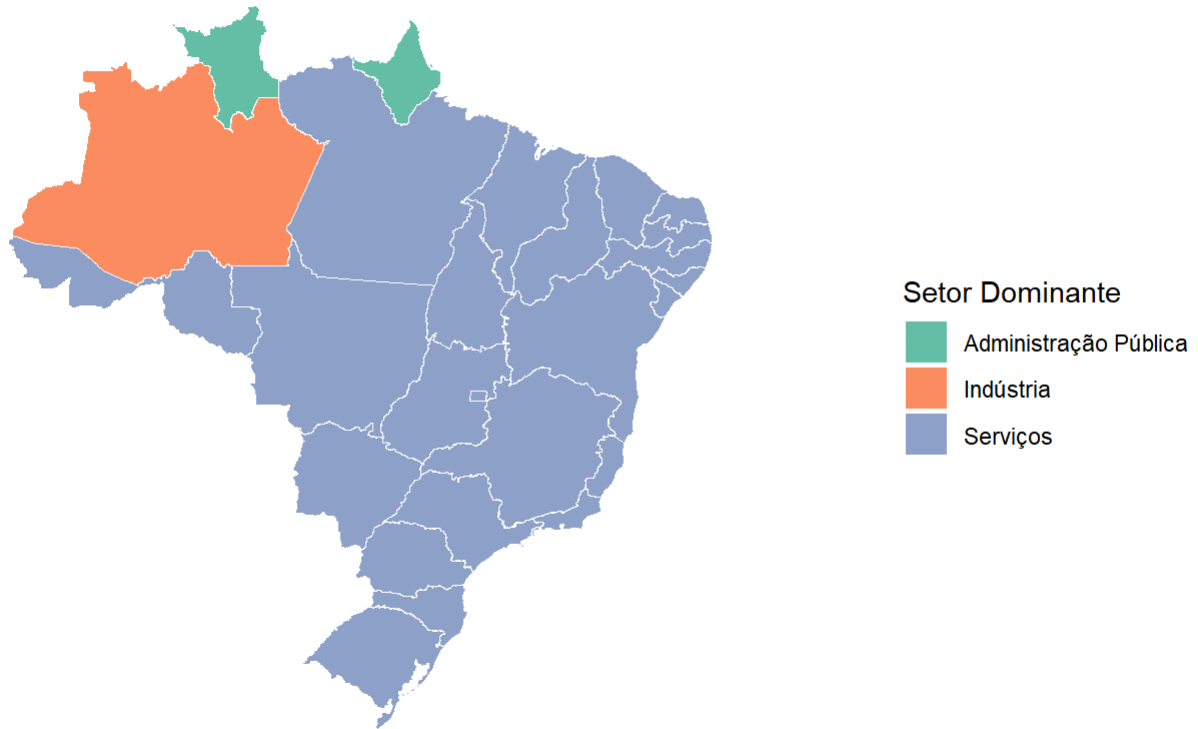
# Determinar o setor dominante em cada estado em 2002
SetorDominante = PibImpacto2002 %>%
  pivot_longer(cols = starts_with("Pct"),
               names_to = "Setor",
               values_to = "Percentual") %>%
  group_by(sigla_uf) %>%
  filter(Percentual == max(Percentual)) %>%
  ungroup() %>%
  mutate(Setor = recode(Setor,
                        PctAgroPecuarria = "Agropecuária",
                        PctImposto = "Impostos",
                        PctIndustria = "Indústria",
                        PctServicos = "Serviços",
                        PctAdespss = "Administração Pública"))

# Unir com os dados de geolocalização dos estados
MapaSetorDominante = MapaEstados %>%
  left_join(SetorDominante, by = c("abbrev_state" = "sigla_uf"))

# Criar o mapa
ggplot(MapaSetorDominante) +
  geom_sf(aes(fill = Setor), color = "white") +
  scale_fill_brewer(palette = "Set2", name = "Setor Dominante") +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),      # Remove as Linhas de grid
    axis.text = element_blank(),      # Remove os textos dos eixos
  ) +
  labs(title = "Setor Econômico Dominante por Estado (2002)",
       subtitle = "Análise baseada na maior contribuição ao PIB de cada estado",
       caption = "Fonte: Dados disponibilizados pelo governo")
```

Setor Econômico Dominante por Estado (2002)

Análise baseada na maior contribuição ao PIB de cada estado



Fonte: Dados disponibilizados pelo governo

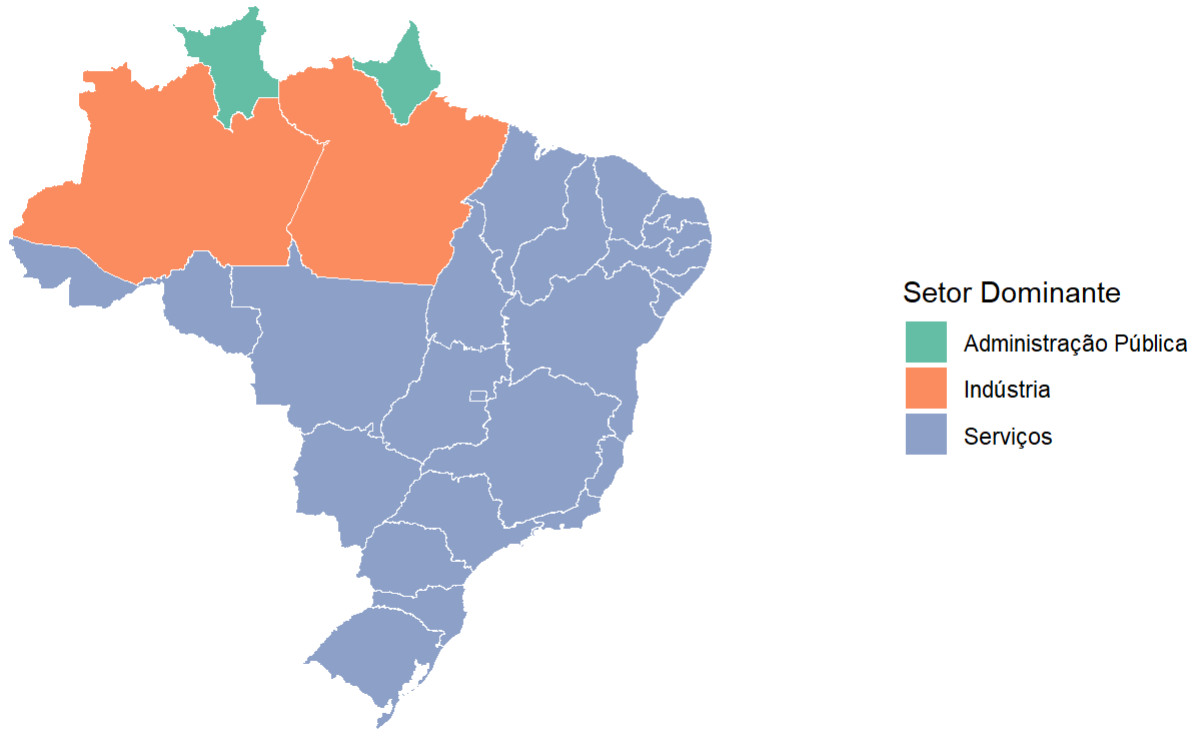
```
# Determinar o setor dominante em cada estado em 2020
SetorDominante = PibImpacto2020 %>%
  pivot_longer(cols = starts_with("Pct"),
               names_to = "Setor",
               values_to = "Percentual") %>%
  group_by(sigla_uf) %>%
  filter(Percentual == max(Percentual)) %>%
  ungroup() %>%
  mutate(Setor = recode(Setor,
                        PctAgroPecuararia = "Agropecuária",
                        PctImposto = "Impostos",
                        PctIndustria = "Indústria",
                        PctServicos = "Serviços",
                        PctAdespss = "Administração Pública"))

# Unir com os dados de geolocalização dos estados
MapaSetorDominante = MapaEstados %>%
  left_join(SetorDominante, by = c("abbrev_state" = "sigla_uf"))

# Criar o mapa
ggplot(MapaSetorDominante) +
  geom_sf(aes(fill = Setor), color = "white") +
  scale_fill_brewer(palette = "Set2", name = "Setor Dominante") +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),      # Remove as Linhas de grid
    axis.text = element_blank(),      # Remove os textos dos eixos
  ) +
  labs(title = "Setor Econômico Dominante por Estado (2020)",
       subtitle = "Análise baseada na maior contribuição ao PIB de cada estado",
       caption = "Fonte: Dados disponibilizados pelo governo")
```

Setor Econômico Dominante por Estado (2020)

Análise baseada na maior contribuição ao PIB de cada estado



Fonte: Dados disponibilizados pelo governo

2)

```
### 2)
```

```
# Unir os dados do IDEB com os dados do PIB em 1 único DF
```

```
PIB_Ideb_2005 <- PIB_UF %>%
  filter(ano == 2005) %>%
  select(sigla_uf, pib, ideb_ensino_publico_fundamental)
```

```
PIB_Ideb_2019 <- PIB_UF %>%
  filter(ano == 2019) %>%
  select(sigla_uf, pib, ideb_ensino_publico_fundamental)
```

```
# Inspeccionar os dados
head(PIB_Ideb_2005)
```

```
##   sigla_uf      pib ideb_ensino_publico_fundamental
## 1      RO 12511821181                3.1
## 2      AC  4300581271                3.4
## 3      AM 33980876529                2.6
## 4      RR  3193430415                3.2
## 5      PA 40522893486                3.2
## 6      AP  4306410130                3.4
```

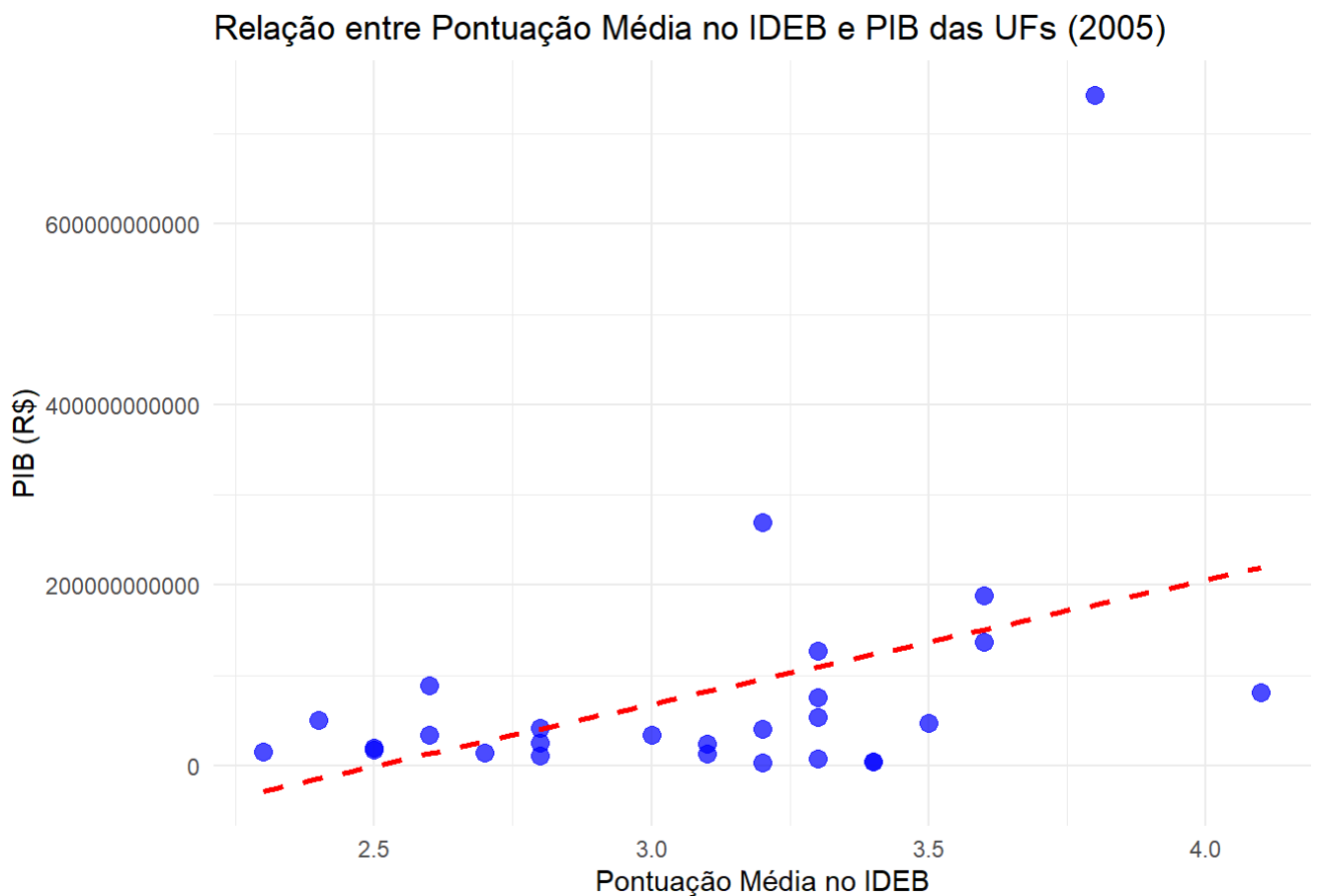
```
head(PIB_Ideb_2019)
```

```
##  sigla_uf      pib ideb_ensino_publico_fundamental
## 1      RO 47091335805                      4.8
## 2      AC 15630016942                      4.8
## 3      AM 108181090996                    4.5
## 4      RR 14292227131                     4.1
## 5      PA 178376983522                    3.9
## 6      AP 17496661074                     3.8
```

Gráficos de dispersão por ano

```
ggplot(PIB_Ideb_2005, aes(x = ideb_ensino_publico_fundamental, y = pib)) +
  geom_point(color = "blue", size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +
  labs(
    title = "Relação entre Pontuação Média no IDEB e PIB das UFs (2005)",
    x = "Pontuação Média no IDEB",
    y = "PIB (R$)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

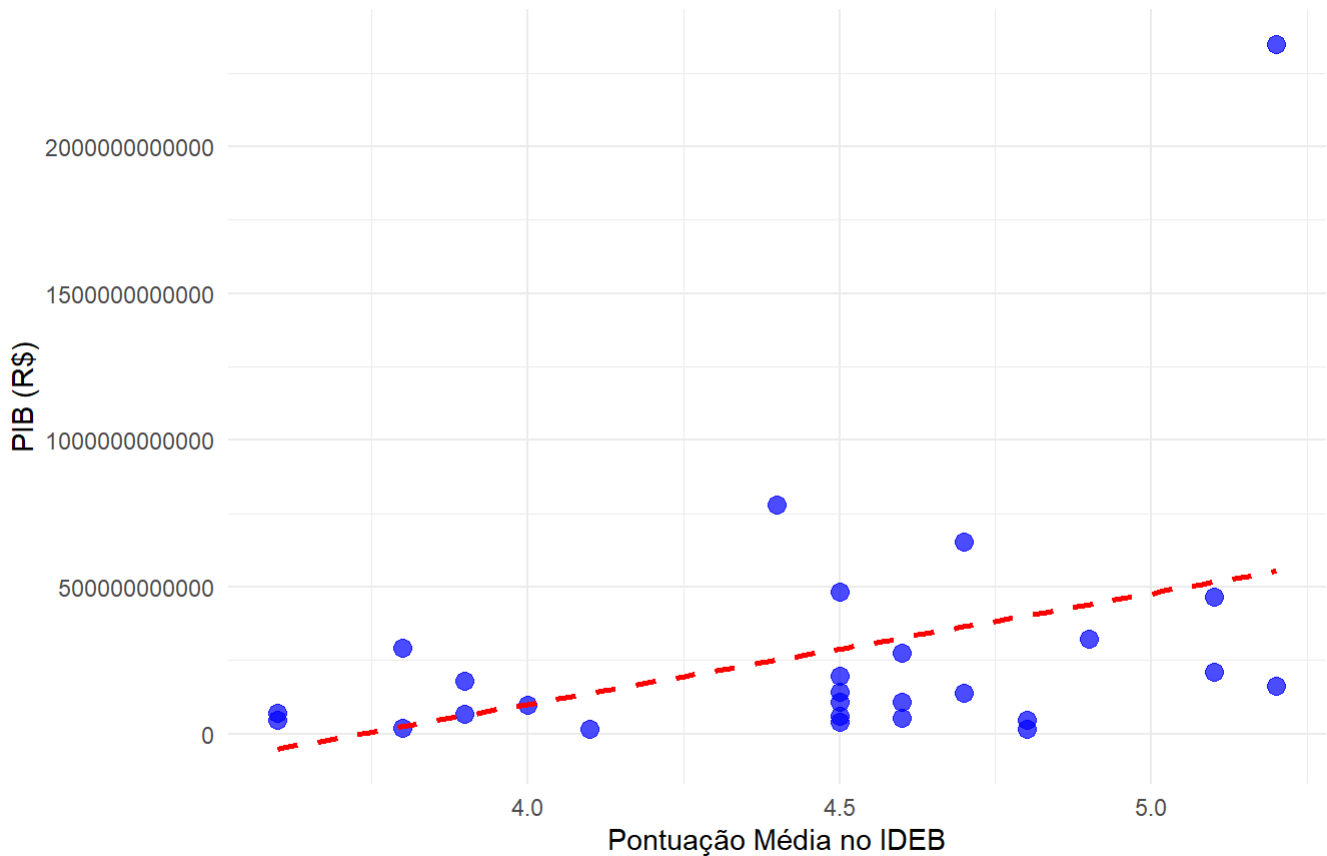


Fonte: Dados do projeto


```
ggplot(PIB_Ideb_2019, aes(x = ideb_ensino_publico_fundamental, y = pib)) +
  geom_point(color = "blue", size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +
  labs(
    title = "Relação entre Pontuação Média no IDEB e PIB das UFs (2005)",
    x = "Pontuação Média no IDEB",
    y = "PIB (R$)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Relação entre Pontuação Média no IDEB e PIB das UFs (2005)

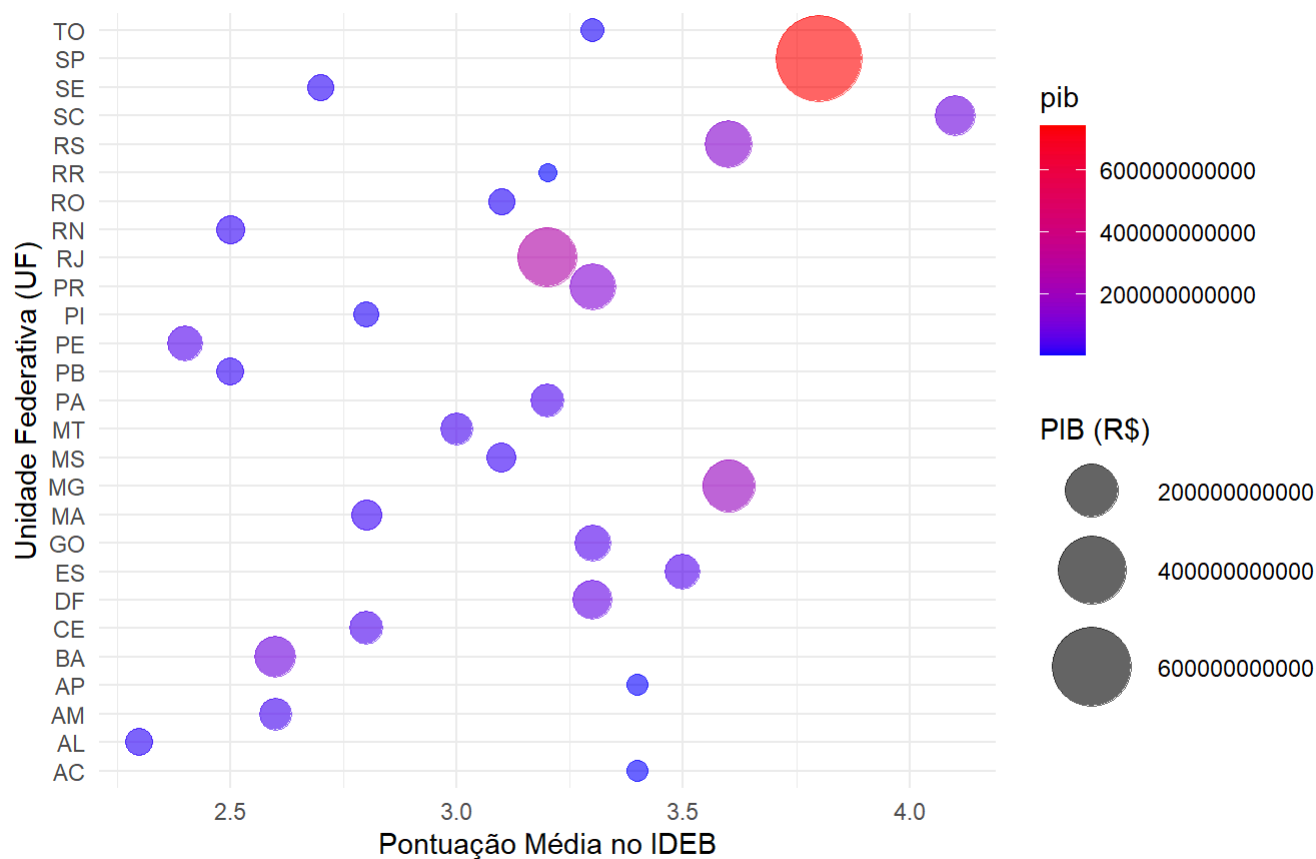


Fonte: Dados do projeto

Gráfico de Bolha por ano

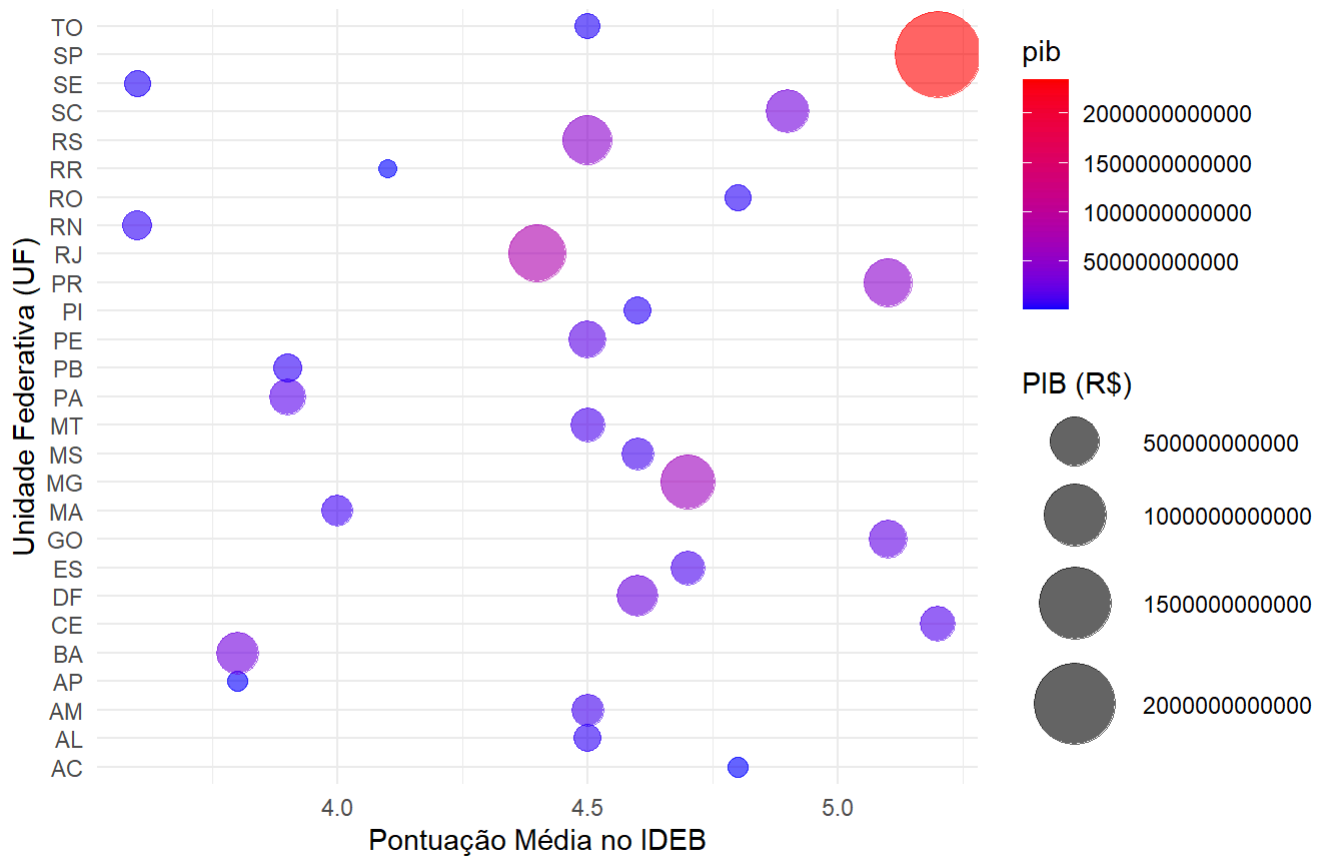
```
ggplot(PIB_Ideb_2005, aes(x = ideb_ensino_publico_fundamental, y = sigla_uf, size = pib, color = pib)) +
  geom_point(alpha = 0.6) +
  scale_size_continuous(name = "PIB (R$)", range = c(3, 15)) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(
    title = "PIB das UFs em relação ao IDEB (2005)",
    x = "Pontuação Média no IDEB",
    y = "Unidade Federativa (UF)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

PIB das UFs em relação ao IDEB (2005)



```
ggplot(PIB_Ideb_2019, aes(x = ideb_ensino_publico_fundamental, y = sigla_uf, size = pib, color = pib)) +
  geom_point(alpha = 0.6) +
  scale_size_continuous(name = "PIB (R$)", range = c(3, 15)) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(
    title = "PIB das UFs em relação ao IDEB (2019)",
    x = "Pontuação Média no IDEB",
    y = "Unidade Federativa (UF)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

PIB das UFs em relação ao IDEB (2019)



3)

```
### 3)

## 2010

# Unir os dados do PIB com os fatores do IDHM
PIB_IDHM <- PIB_UF %>%
  filter(ano == 2010) %>%
  select(sigla_uf, pib, idhm_2010, expectativa_vida_2010, indice_gini_2010, renda_pc_2010, indice_escolaridade_2010, indice_frequencia_escolar_2010)

# Inspeccionar os dados combinados
head(PIB_IDHM)
```

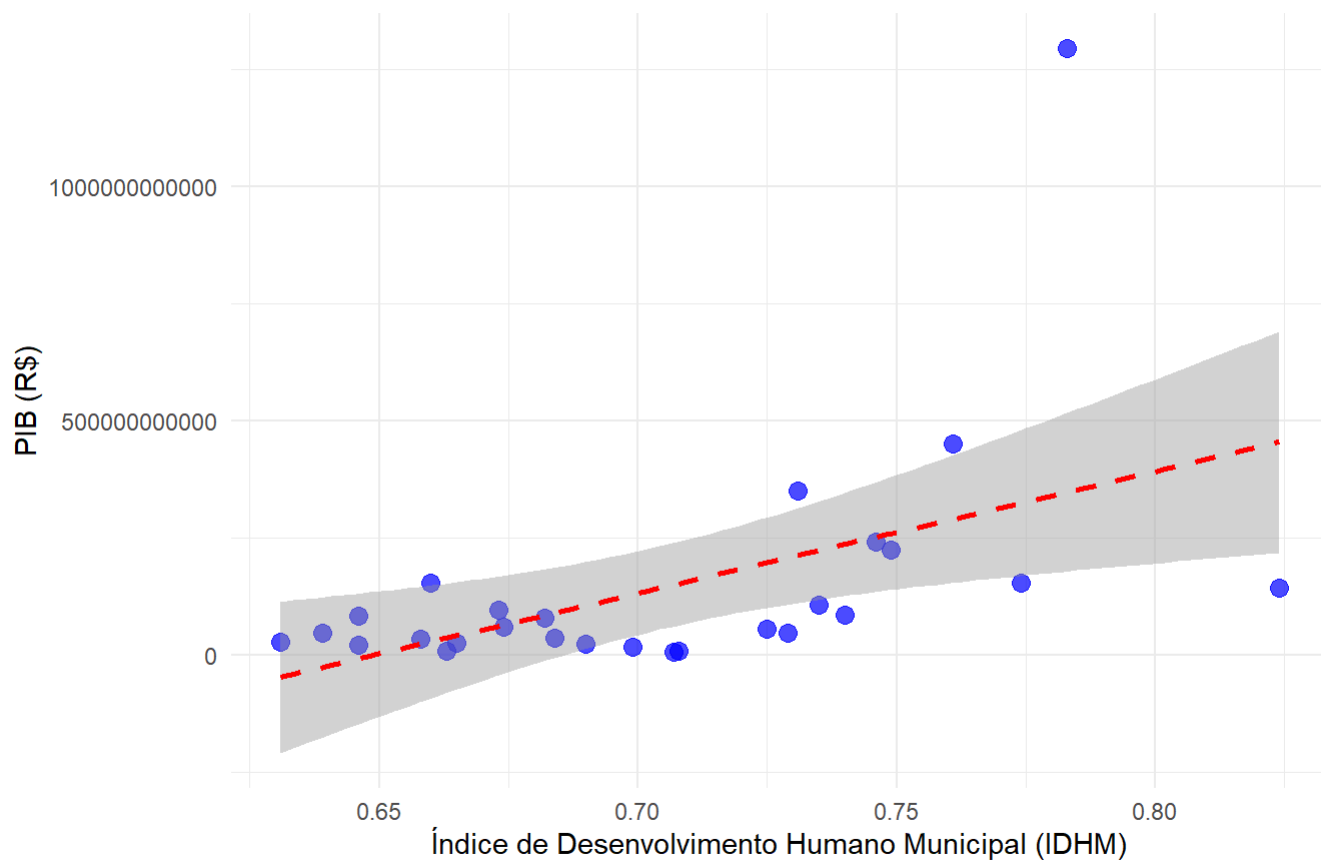
```
##      sigla_uf      pib idhm_2010 expectativa_vida_2010 indice_gini_2010
## 1      RO 23907886882      0.690              72.97              0.56
## 2      AC 8342355523      0.663              71.63              0.63
## 3      AM 60877122684      0.674              73.30              0.65
## 4      RR 6639150477      0.707              73.51              0.63
## 5      PA 82684517799      0.646              72.36              0.62
## 6      AP 8237795351      0.708              73.80              0.60
##      renda_pc_2010 indice_escolaridade_2010 indice_frequencia_escolar_2010
## 1      670.82              0.480              0.633
## 2      522.15              0.488              0.599
## 3      539.80              0.549              0.567
## 4      605.59              0.587              0.650
## 5      446.76              0.474              0.557
## 6      598.98              0.618              0.634
```

```
# IDHM e PIB
```

```
ggplot(PIB_IDHM, aes(x = idhm_2010, y = pib)) +
  geom_point(color = "blue", size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color = "red", linetype = "dashed") +
  labs(
    title = "Relação entre IDHM e PIB das UFs (2010)",
    x = "Índice de Desenvolvimento Humano Municipal (IDHM)",
    y = "PIB (R$)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Relação entre IDHM e PIB das UFs (2010)

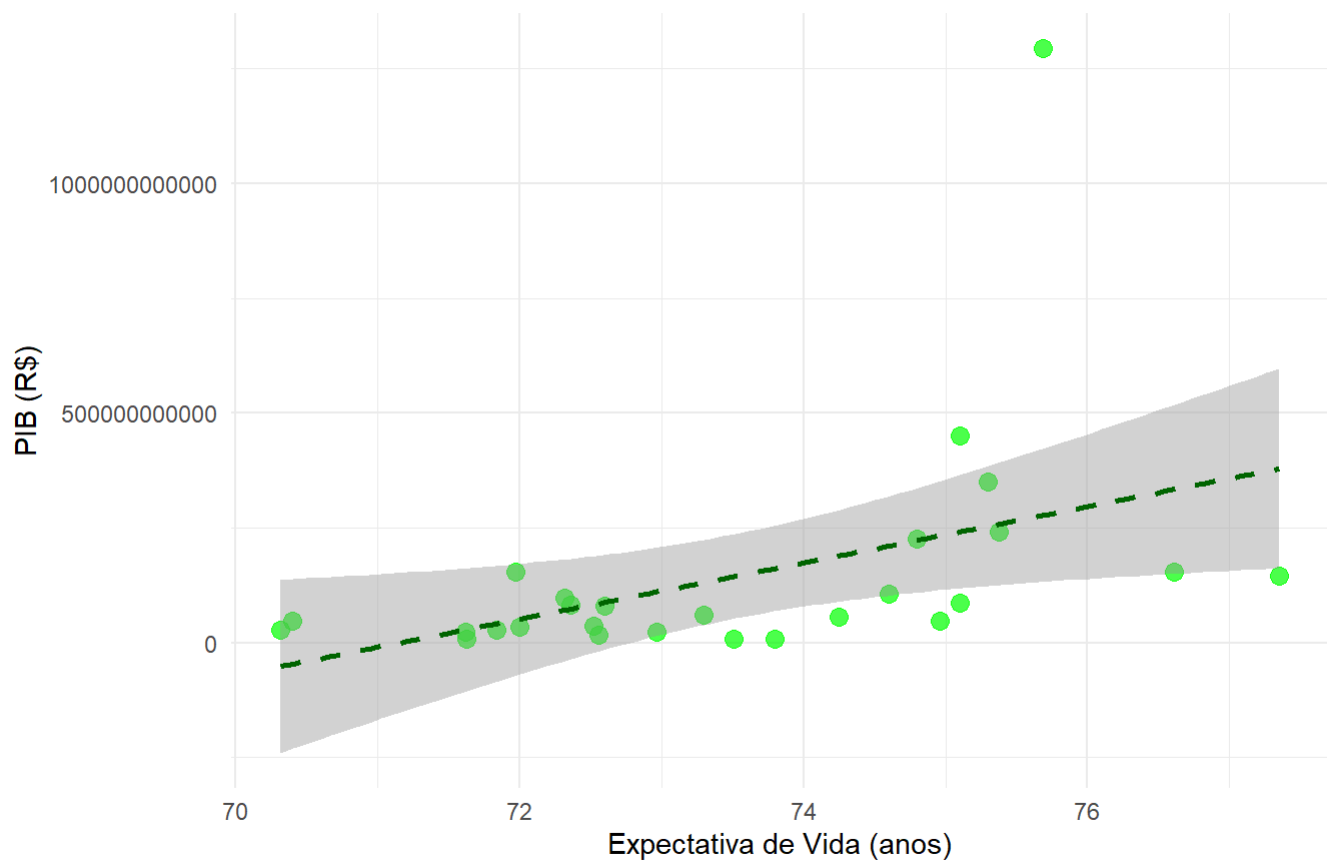


Fonte: Dados do projeto

```
# Expectativa de Vida e PIB
ggplot(PIB_IDHM, aes(x = expectativa_vida_2010, y = pib)) +
  geom_point(color = "green", size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color = "darkgreen", linetype = "dashed") +
  labs(
    title = "Relação entre Expectativa de Vida e PIB das UFs (2010)",
    x = "Expectativa de Vida (anos)",
    y = "PIB (R$)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Relação entre Expectativa de Vida e PIB das UFs (2010)

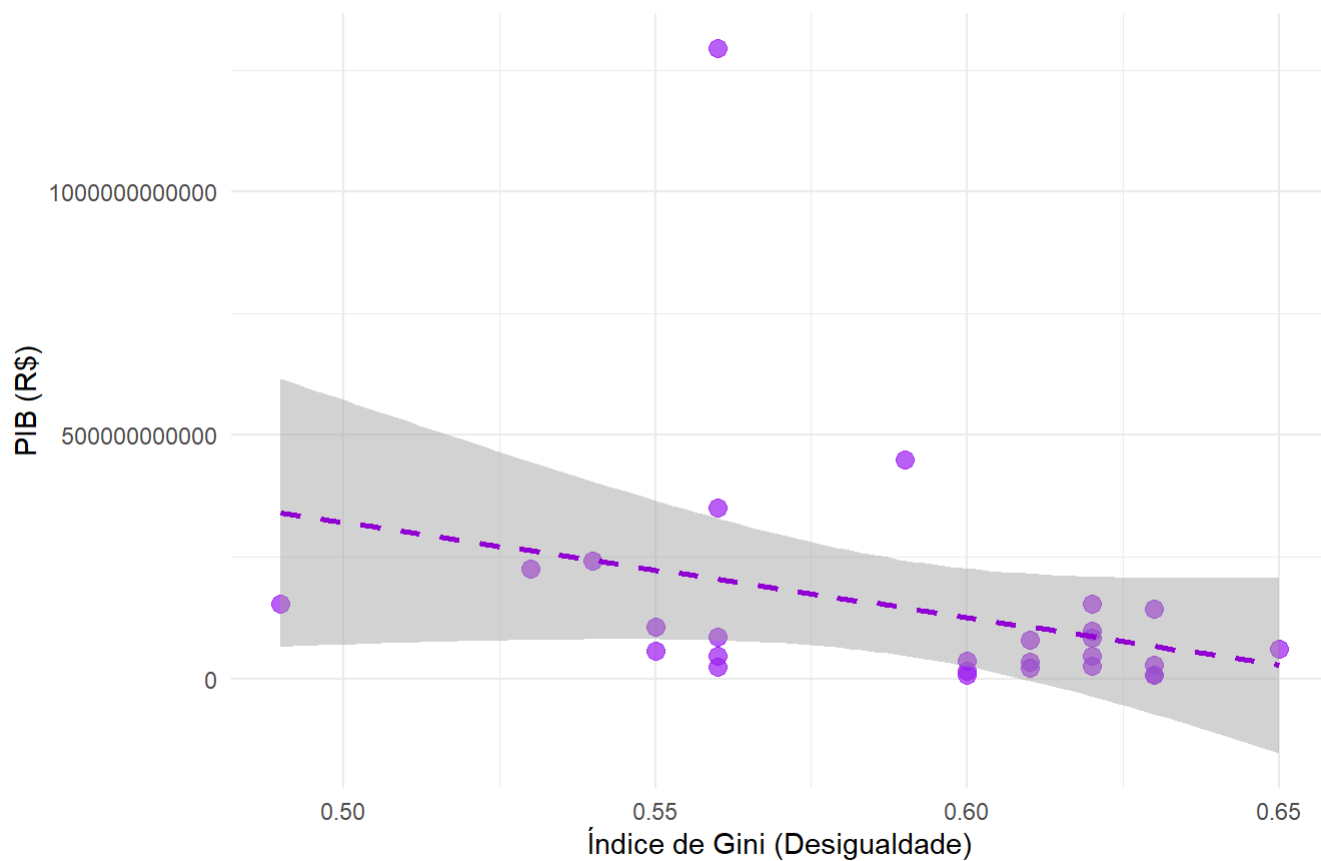


Fonte: Dados do projeto

```
# Índice de Gini e PIB
ggplot(PIB_IDHM, aes(x = indice_gini_2010, y = pib)) +
  geom_point(color = "purple", size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color = "darkviolet", linetype = "dashed") +
  labs(
    title = "Relação entre Índice de Gini e PIB das UFs (2020)",
    x = "Índice de Gini (Desigualdade)",
    y = "PIB (R$)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Relação entre Índice de Gini e PIB das UFs (2020)

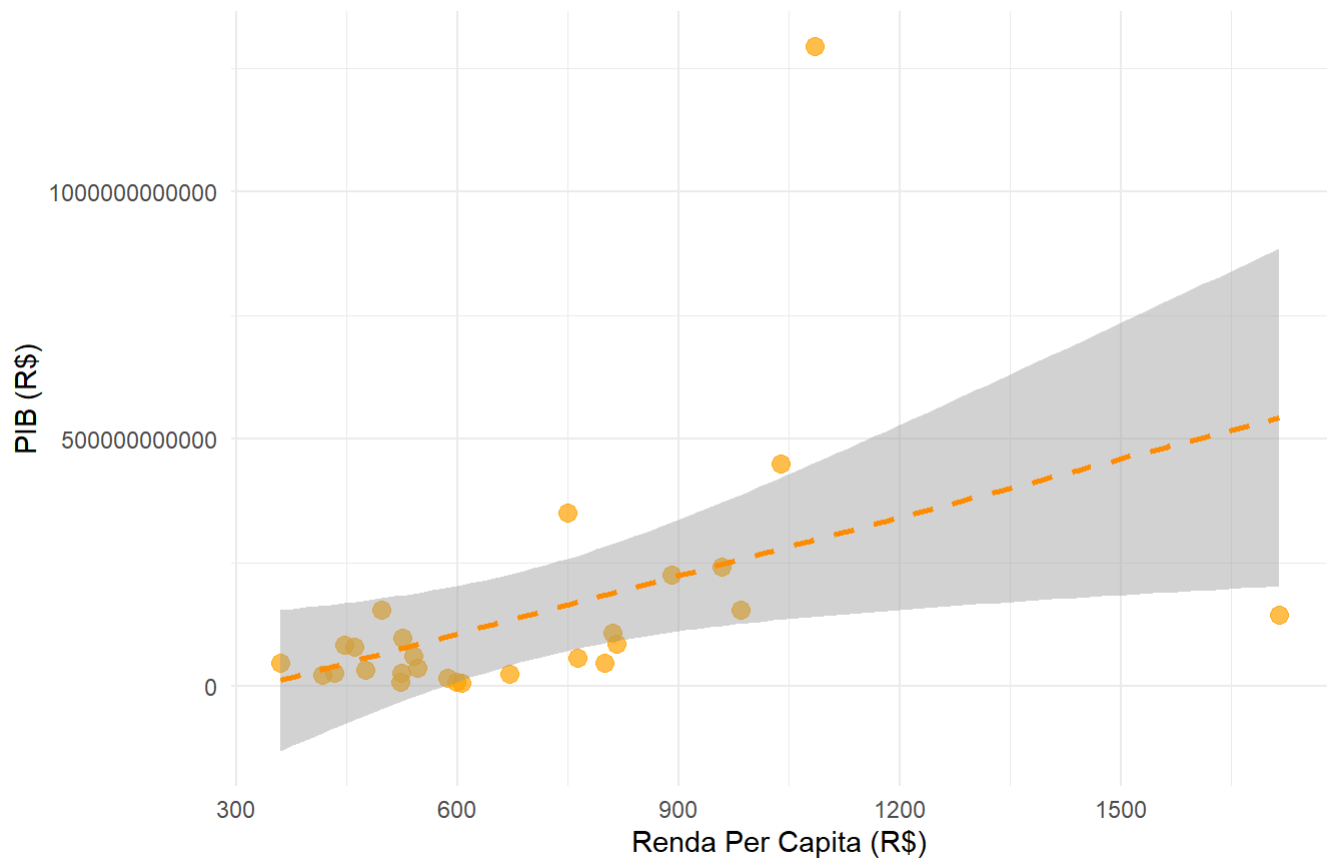


Fonte: Dados do projeto

```
# Renda Per Capita e PIB
ggplot(PIB_IDHM, aes(x = renda_pc_2010, y = pib)) +
  geom_point(color = "orange", size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color = "darkorange", linetype = "dashed") +
  labs(
    title = "Relação entre Renda Per Capita e PIB das UFs (2010)",
    x = "Renda Per Capita (R$)",
    y = "PIB (R$)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Relação entre Renda Per Capita e PIB das UFs (2010)

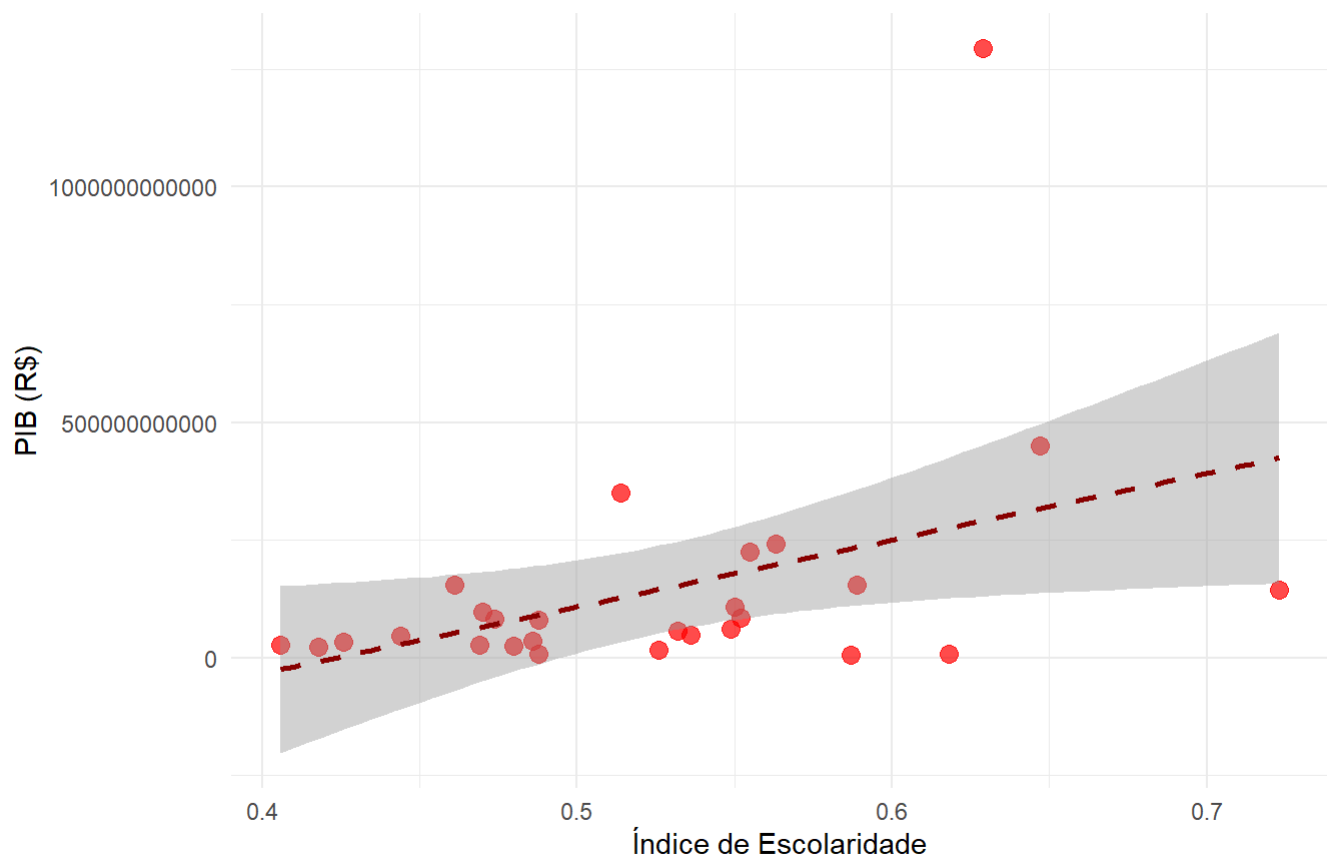


Fonte: Dados do projeto

```
# Índice de Escolaridade e PIB
ggplot(PIB_IDHM, aes(x = indice_escolaridade_2010, y = pib)) +
  geom_point(color = "red", size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color = "darkred", linetype = "dashed") +
  labs(
    title = "Relação entre Índice de Escolaridade e PIB das UFs (2010)",
    x = "Índice de Escolaridade",
    y = "PIB (R$)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


Relação entre Índice de Escolaridade e PIB das UFs (2010)

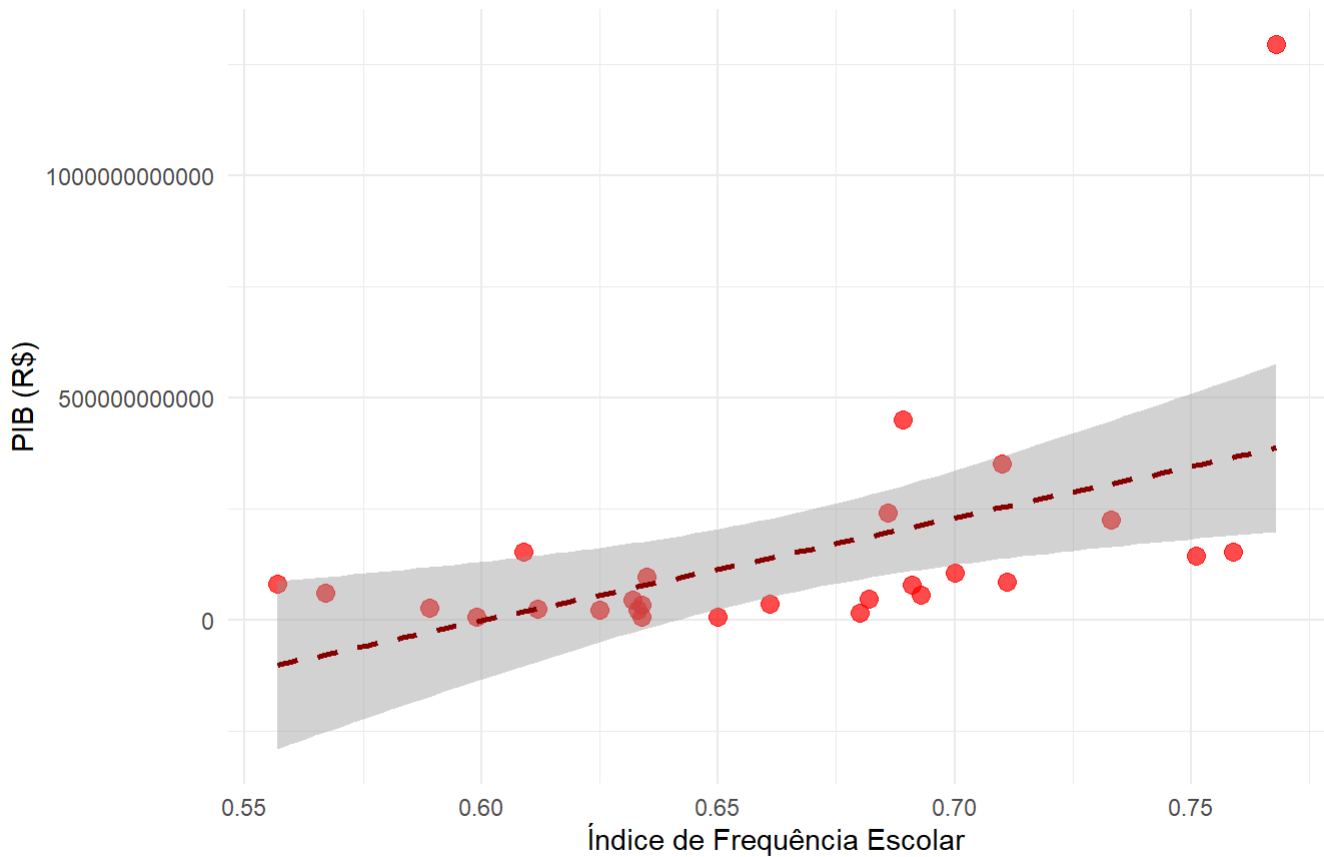


Fonte: Dados do projeto

```
# Frequência escolar e PIB
ggplot(PIB_IDHM, aes(x = indice_frequencia_escolar_2010, y = pib)) +
  geom_point(color = "red", size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color = "darkred", linetype = "dashed") +
  labs(
    title = "Relação entre Índice de Frequência Escolar e PIB das UFs (2010)",
    x = "Índice de Frequência Escolar",
    y = "PIB (R$)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Relação entre Índice de Frequência Escolar e PIB das UFs (2010)



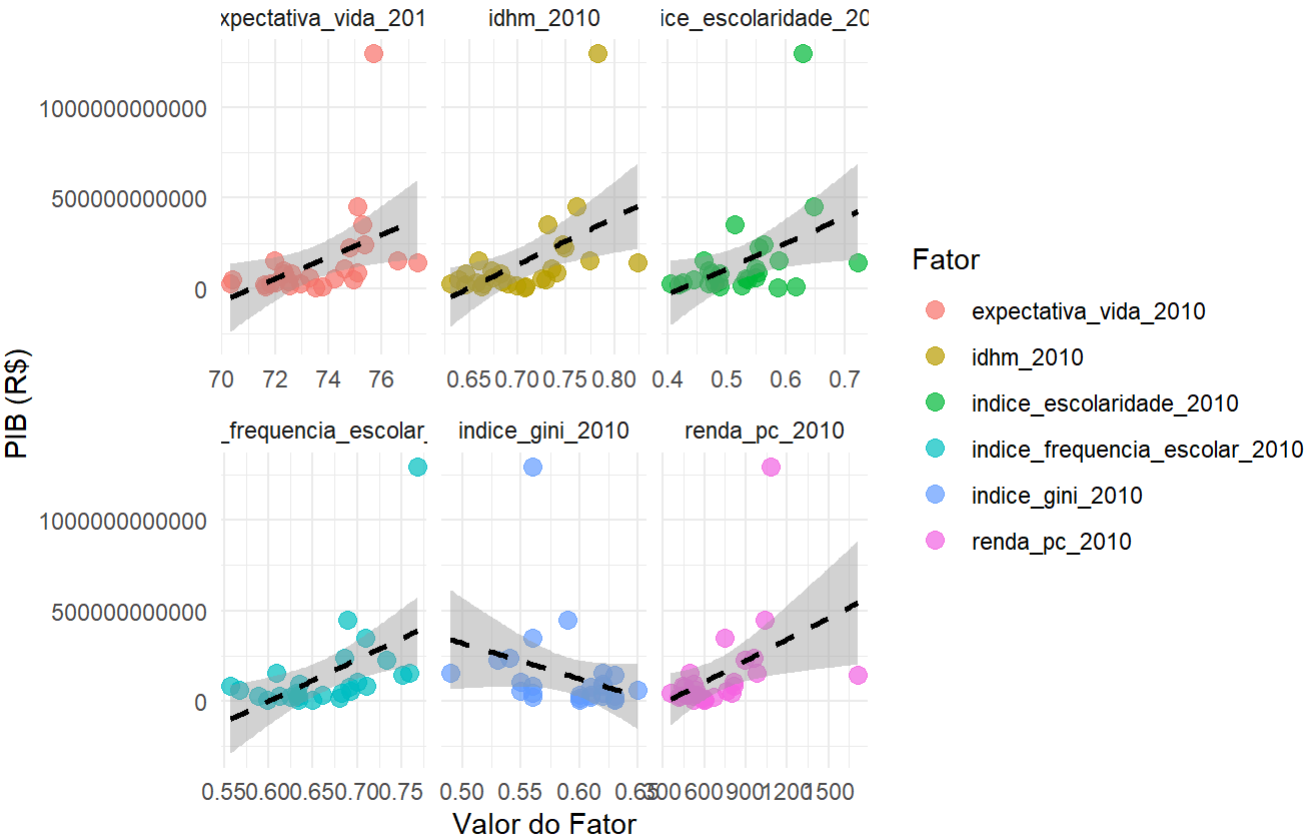
Fonte: Dados do projeto

```
# Transformar os dados para formato longo (long format)
PIB_IDHM_long <- PIB_IDHM %>%
  pivot_longer(
    cols = c(idhm_2010, expectativa_vida_2010, indice_gini_2010, renda_pc_2010, indice_escola
ridade_2010, indice_frequencia_escolar_2010),
    names_to = "Fator",
    values_to = "Valor"
  )

# Criar o gráfico combinado
ggplot(PIB_IDHM_long, aes(x = Valor, y = pib)) +
  geom_point(aes(color = Fator), size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color = "black", linetype = "dashed") +
  facet_wrap(~ Fator, scales = "free_x") +
  labs(
    title = "Relação entre PIB e Fatores do IDHM das UFs (2010)",
    x = "Valor do Fator",
    y = "PIB (R$)",
    caption = "Fonte: Dados do projeto"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Relação entre PIB e Fatores do IDHM das UFs (2010)



Questões Extras

UF escolhida: "SC"

1)

```
### 1)

## Calcular PIB total

# 2002
PibSul2002 = PibDelta %>%
  filter(ano == 2002)

PibSul2002 = PibSul2002 %>%
  mutate(PibTotal = sum(PibSul2002$pib))

# 2020
PibSul2020 = PibDelta %>%
  filter(ano == 2020)

PibSul2020 = PibSul2020 %>%
  mutate(PibTotal = sum(PibSul2020$pib))

## Calcular impacto das variáveis em %

# 2002
PibSul2002$PctPIB = round((PibSul2002$pib / PibSul2002$PibTotal) * 100, 2)

# 2020
PibSul2020$PctPIB = round((PibSul2020$pib / PibSul2020$PibTotal) * 100, 2)

## Filtrar apenas o que interessa

PibSul2002 = PibSul2002 %>%
  select(ano, sigla_uf, pib, va, impostos_liquidos, va_agropecuaria, va_industria, va_servico
s, va_adespss, PctPIB) %>%
  filter(sigla_uf %in% c("RS", "SC", "PR"))

PibSul2020 = PibSul2020 %>%
  select(ano, sigla_uf, pib, va, impostos_liquidos, va_agropecuaria, va_industria, va_servico
s, va_adespss, PctPIB) %>%
  filter(sigla_uf %in% c("RS", "SC", "PR"))

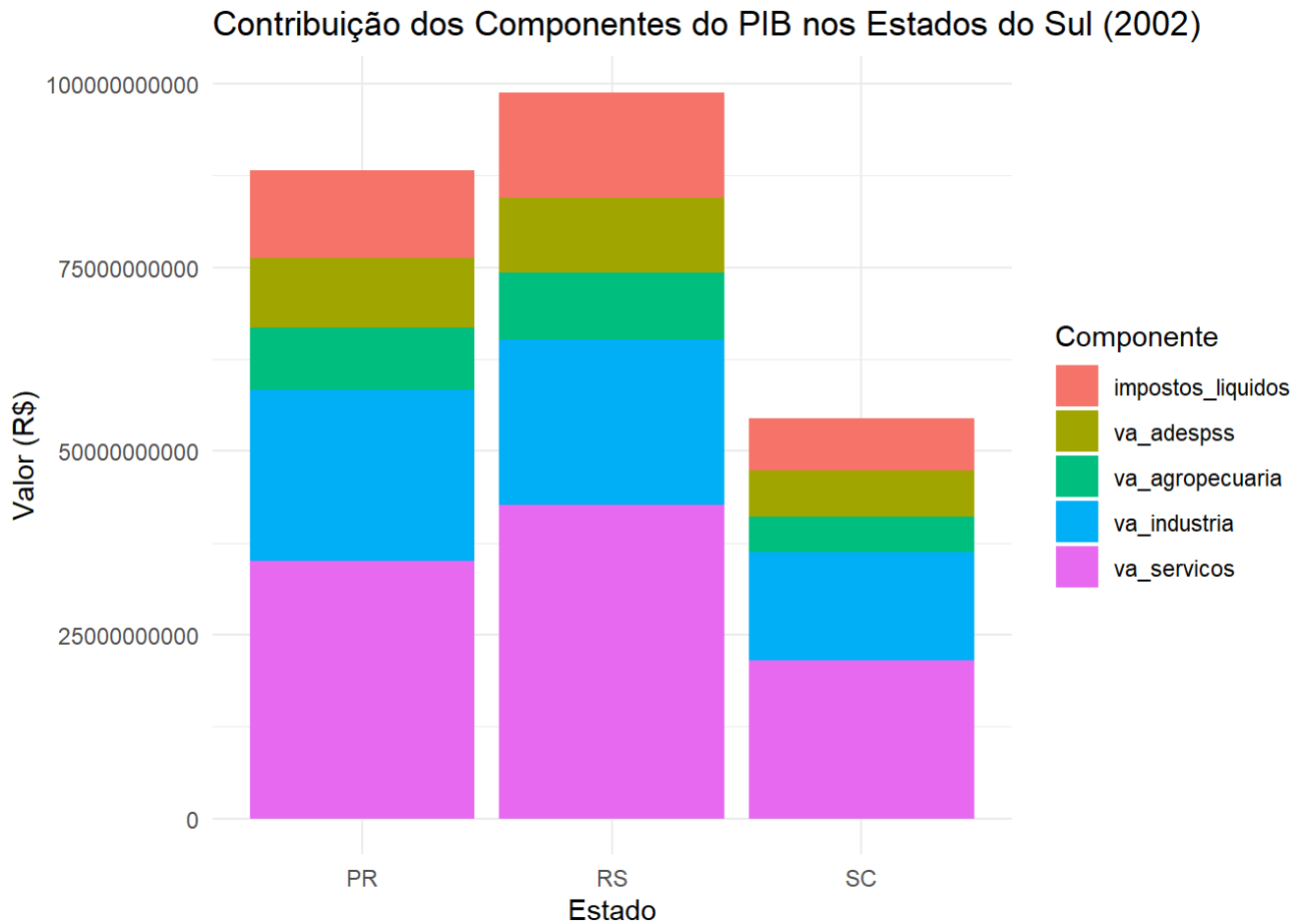
## Plot

#2002

# Transformar os dados para o formato Longo
df_long = PibSul2002 %>%
  select(sigla_uf, va_agropecuaria, va_industria, va_servicos, va_adespss, impostos_liquidos)
%>%
  pivot_longer(cols = -sigla_uf, names_to = "componente", values_to = "valor")

# Criar o gráfico de barras empilhadas
ggplot(df_long, aes(x = sigla_uf, y = valor, fill = componente)) +
  geom_bar(stat = "identity") +
  labs(title = "Contribuição dos Componentes do PIB nos Estados do Sul (2002)",
    x = "Estado",
    y = "Valor (R$)",
```

```
fill = "Componente") +
theme_minimal()
```



```
# 2020
```

```
# Transformar os dados para o formato Longo
```

```
df_long2 = PibSul2020 %>%
```

```
  select(sigla_uf, va_agropecuaria, va_industria, va_servicos, va_adespss, impostos_liquidos)
%>%
```

```
  pivot_longer(cols = -sigla_uf, names_to = "componente", values_to = "valor")
```

```
# Criar o gráfico de barras empilhadas
```

```
ggplot(df_long2, aes(x = sigla_uf, y = valor, fill = componente)) +
```

```
  geom_bar(stat = "identity") +
```

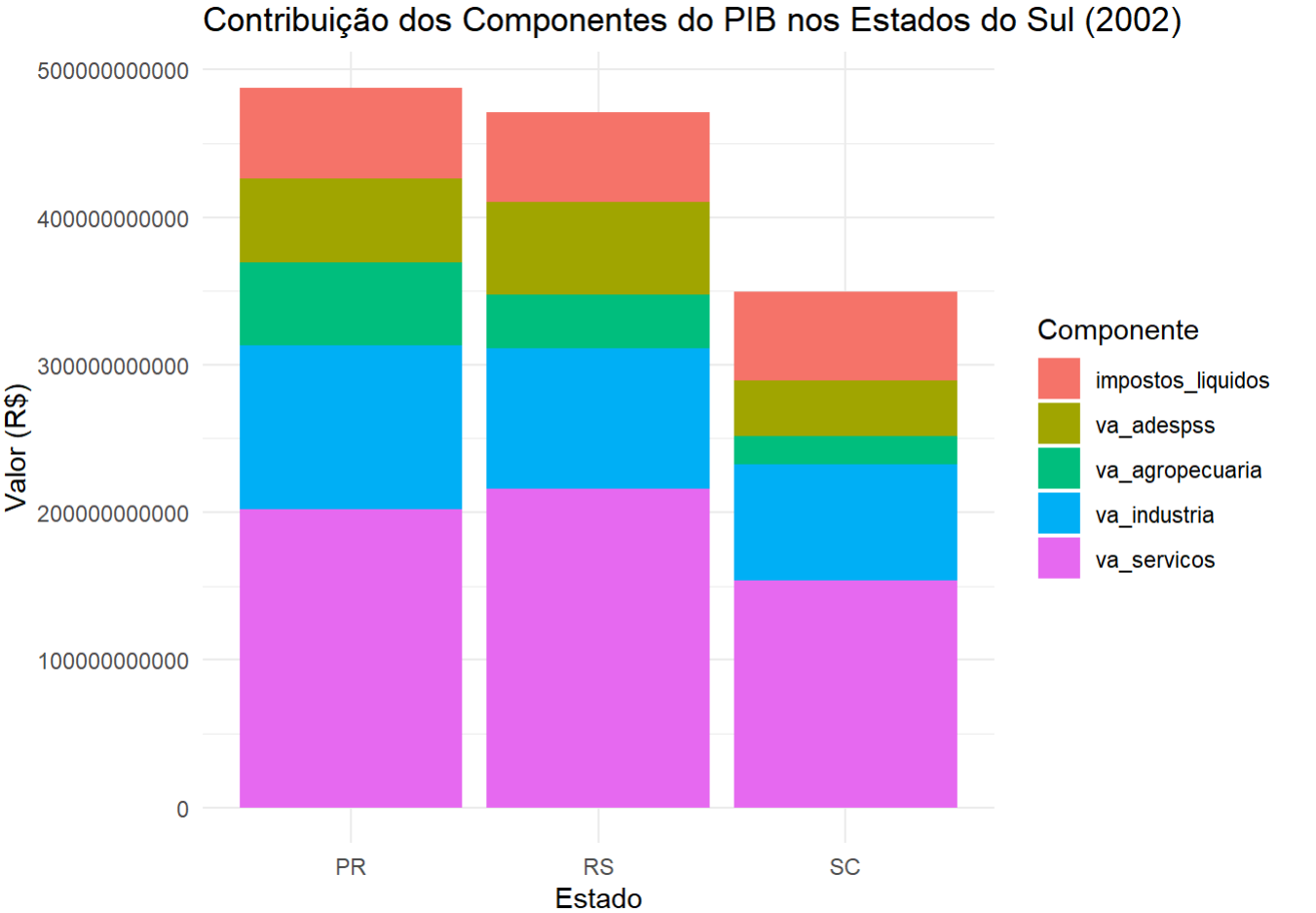
```
  labs(title = "Contribuição dos Componentes do PIB nos Estados do Sul (2002)",
```

```
        x = "Estado",
```

```
        y = "Valor (R$)",
```

```
        fill = "Componente") +
```

```
  theme_minimal()
```



2)

```
### 2)

# Filtrar dados para estado escolhido

PibSC = PIB_UF %>%
  select(ano, sigla_uf, pib) %>%
  filter(sigla_uf == "SC")

# Aumento ano a ano

PibSC = PibSC %>%
  mutate(aumento_pib = pib - lag(pib), # Diferença absoluta do PIB
         aumento_percentual = (aumento_pib / lag(pib)) * 100)

# Aumento médio

crescimento_medio = PibSC %>%
  group_by(sigla_uf) %>% # Sem o groupby, não preserva a UF
  summarise(crescimento_medio = mean(aumento_percentual, na.rm = TRUE))

# Aumento acumulado

crescimento_acumulado = PibSC %>%
  group_by(sigla_uf) %>%
  summarise(crescimento_acumulado = ((last(pib) / first(pib)) - 1) * 100)

# Juntar os resultados

CrescimentoAnalise = left_join(crescimento_medio, crescimento_acumulado, by = "sigla_uf")

# Visualizar resultados

CrescimentoAnoaAno = PibSC %>%
  select(ano, pib, aumento_pib, aumento_percentual)

CrescimentoAnoaAno
```

##	ano	pib	aumento_pib	aumento_percentual
## 1	2002	54481893216	NA	NA
## 2	2003	64098474683	9616581467	17.650968
## 3	2004	73618965923	9520491240	14.852914
## 4	2005	81549243683	7930277760	10.772058
## 5	2006	91063448624	9514204941	11.666822
## 6	2007	103728092246	12664643622	13.907494
## 7	2008	121477257847	17749165601	17.111243
## 8	2009	129098535042	7621277195	6.273830
## 9	2010	153726007373	24627472331	19.076492
## 10	2011	174068321733	20342314360	13.232839
## 11	2012	191794652139	17726330406	10.183548
## 12	2013	214512241571	22717589432	11.844746
## 13	2014	242553370851	28041129280	13.072042
## 14	2015	249079642281	6526271430	2.690654
## 15	2016	256754668524	7675026243	3.081354
## 16	2017	277270236580	20515568056	7.990339
## 17	2018	298227090046	20956853466	7.558277
## 18	2019	323263857398	25036767352	8.395202
## 19	2020	349275015538	26011158140	8.046417

CrescimentoAnalise

```
## # A tibble: 1 × 3
##   sigla_uf crescimento_medio crescimento_acumulado
##   <chr>          <dbl>          <dbl>
## 1 SC              11.0              541.
```

3)

```
### 3)

# filtrar Uf

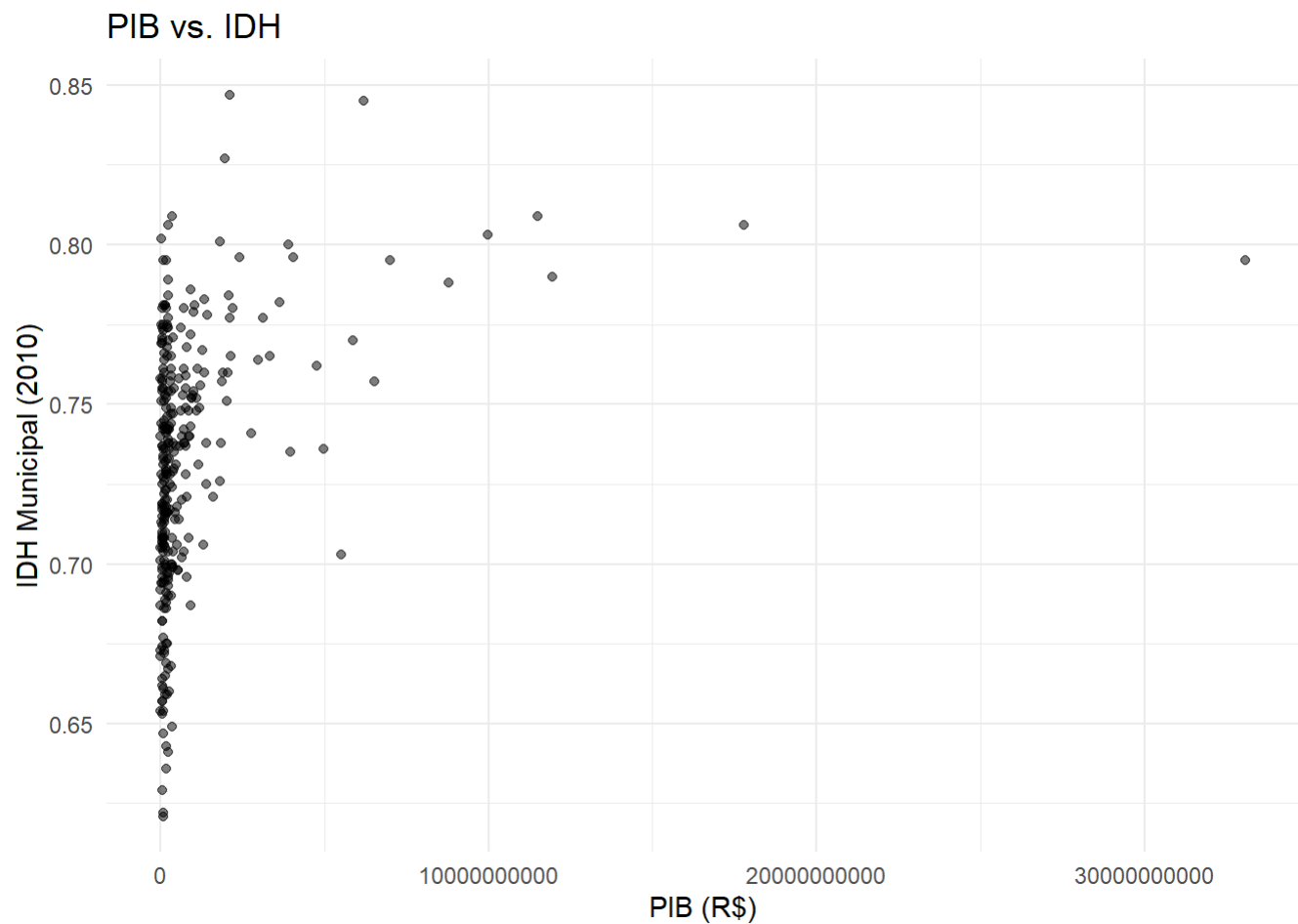
PibMunicipioSC = PIB_Municipio %>%
  filter(sigla_uf == "SC")

# Filtrar apenas o último ano disponível
PibMunicipioSC = PibMunicipioSC %>%
  filter(ano == 2020)

## Criar os gráficos

# PIB vs IDH
ggplot(PibMunicipioSC, aes(x = pib, y = idhm_2010)) +
  geom_point(alpha = 0.5) +
  labs(x = "PIB (R$)", y = "IDH Municipal (2010)", title = "PIB vs. IDH") +
  theme_minimal()
```

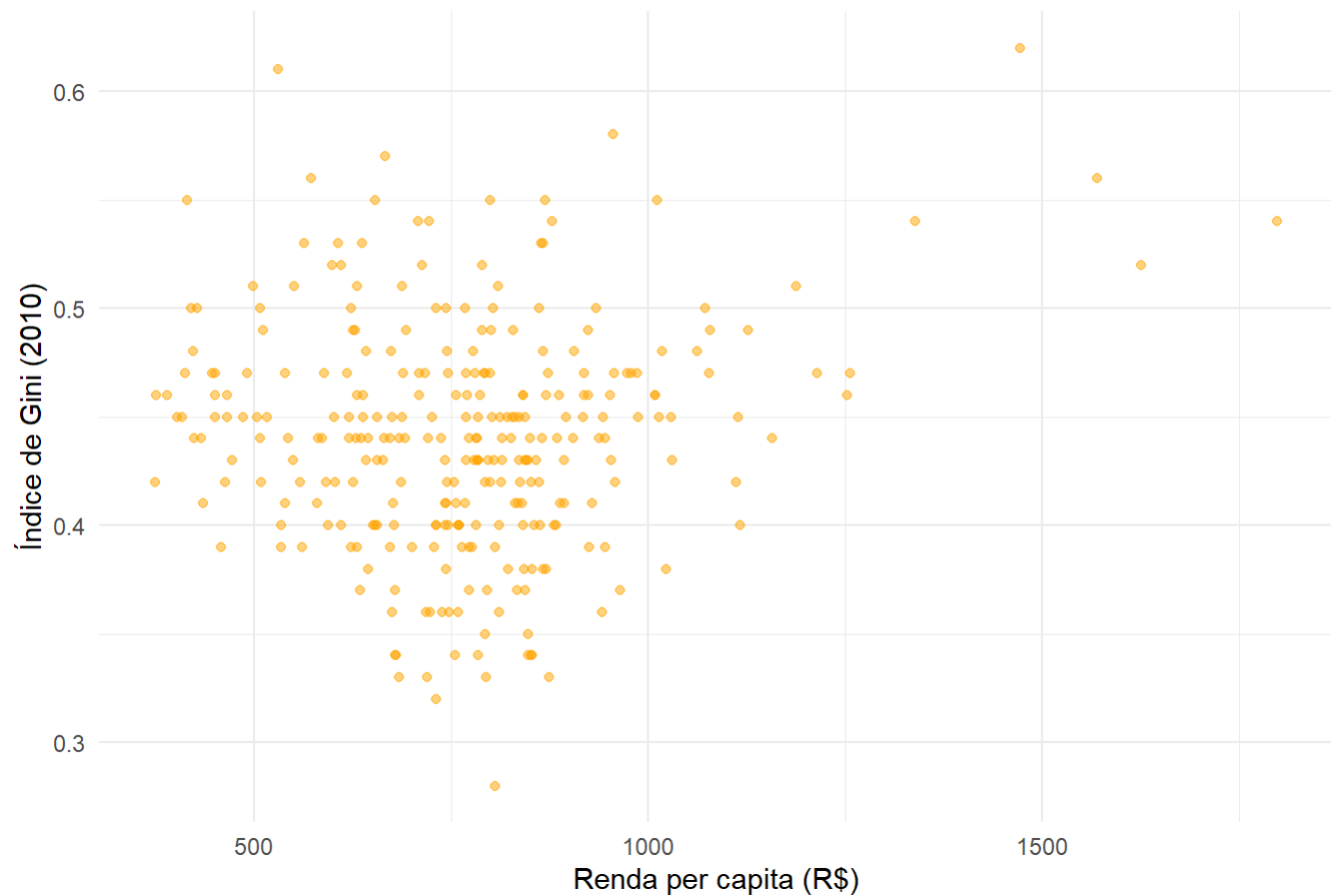
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
# Renda per capita vs Índice de Gini
ggplot(PibMunicipioSC, aes(x = renda_pc_2010, y = indice_gini_2010)) +
  geom_point(alpha = 0.5, color = "orange") +
  labs(x = "Renda per capita (R$)", y = "Índice de Gini (2010)", title = "Renda per capita v
s. Índice de Gini") +
  theme_minimal()
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

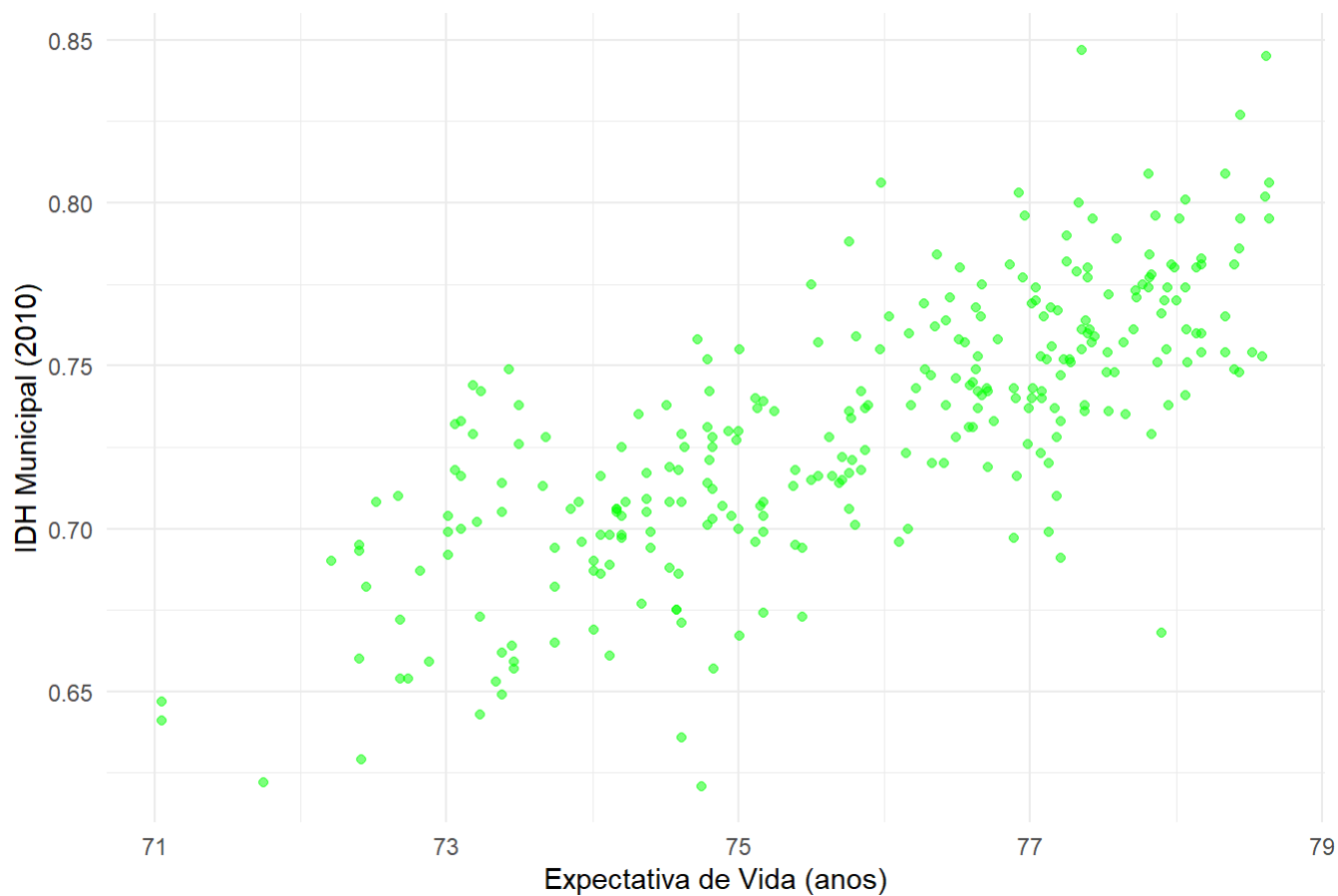
Renda per capita vs. Índice de Gini



```
# Expectativa de Vida vs IDH
ggplot(PibMunicipioSC, aes(x = expectativa_vida_2010, y = idhm_2010)) +
  geom_point(alpha = 0.5, color = "green") +
  labs(x = "Expectativa de Vida (anos)", y = "IDH Municipal (2010)", title = "Expectativa de
Vida vs. IDH") +
  theme_minimal()
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Expectativa de Vida vs. IDH



```
# Valor Adicionado da Indústria vs PIB
ggplot(PibMunicipioSC, aes(x = va_industria, y = pib)) +
  geom_point(alpha = 0.5, color = "red") +
  labs(x = "Valor Adicionado da Indústria (R$)", y = "PIB (R$)", title = "Valor Adicionado da
Indústria vs. PIB") +
  theme_minimal()
```

Valor Adicionado da Indústria vs. PIB

