# EMISSÃO DE $CO_2$ DO SOLO EM ÁREAS AGRÍCOLAS: ABORDAGEM EM APRENDIZADO DE MÁQUINA ESTATÍSTICO


## Autor: Prof. Dr. Alan Rodrigo Panosso (https://www.fcav.unesp.br/#!/alan)

E-mail: alan.panosso@unesp.br (mailto:alan.panosso@unesp.br)

Departamento de Engenharia e Ciências Exatas

UNESP - Câmpus de Jaboticabal


# Objetivo

O objetivo do repositório `tese-fco2-ml-2023` é promover a transparência, a reprodutibilidade e a colaboração em pesquisa. Você é incentivado a explorar o código-fonte, utilizar os dados e contribuir com melhorias, se desejar. Sinta-se à vontade para entrar em contato caso tenha alguma dúvida ou precise de mais informações sobre minha pesquisa.

# Contribuições

Contribuições são bem-vindas! Se você deseja colaborar com melhorias nos códigos, correções de erros ou qualquer outro aprimoramento, sinta-se à vontade para abrir uma solicitação de `pull request`.

# Licença



# Base de dados

Apresentação do pacote `fco2r` construído para divulgação e análise dos resultados obtidos ao longo de mais de $21$ anos de ensaios em campo. Este pacote, permite a visualização dos dados, a execução de análises estatísticas avançadas e a geração de gráficos interativos para tornar os resultados mais acessíveis e compreensíveis para a comunidade científica.

## Instalação

Você pode instalar uma versão de desenvolvimento do pacote `fco2r` a partir do GitHub (https://github.com/) com os seguintes comandos:

```
# install.packages("devtools")
# devtools::install_github("arpanosso/fco2r")
```

## Problemas na instalação:

Possíveis problemas na instalação do pacote podem ser sanados com os seguintes comandos:

```
# Sys.getenv("GITHUB_PAT")
# Sys.unsetenv("GITHUB_PAT")
# Sys.getenv("GITHUB_PAT")
```

## Carregando os pacotes

```
library(fco2r)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.0

## Warning: package 'dplyr' was built under R version 4.4.0

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.2     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── 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 become errors
```

```
library(patchwork)
library(ggspatial)
```

```
## Warning: package 'ggspatial' was built under R version 4.3.1
```

```
library(readxl)
library(skimr)
library(tidymodels)
```

```
## ── Attaching packages ────────────────────────────────────── tidymodels 1.1.0 ──
## ✔ broom        1.0.4     ✔ rsample      1.1.1
## ✔ dials        1.2.0     ✔ tune         1.1.1
## ✔ infer        1.0.4     ✔ workflows    1.1.3
## ✔ modeldata    1.1.0     ✔ workflowsets 1.0.1
## ✔ parsnip      1.1.0     ✔ yardstick    1.2.0
## ✔ recipes      1.0.6
## ── Conflicts ───────────────────────────────────────── tidymodels_conflicts() ──
## ✖ scales::discard() masks purrr::discard()
## ✖ dplyr::filter()   masks stats::filter()
## ✖ recipes::fixed()  masks stringr::fixed()
## ✖ dplyr::lag()      masks stats::lag()
## ✖ yardstick::spec() masks readr::spec()
## ✖ recipes::step()   masks stats::step()
## • Use tidymodels_prefer() to resolve common conflicts.
```

```
library(ISLR)
library(modeldata)
library(vip)
```

```
##
## Attaching package: 'vip'
##
## The following object is masked from 'package:utils':
##
##     vi
```

```
library(ggpubr)
source("R/graficos.R")
theme_set(theme_bw())
```

## Conhecendo a base de dados de emissão de $CO_2$ do solo

Base proveniente de ensaios de campo.

```
glimpse(data_fco2)
```

```
## Rows: 15,397
## Columns: 39
## $ experimento       <chr> "Espacial", "Espacial", "Espacial", "Espacial", "Esp…
## $ data              <date> 2001-07-10, 2001-07-10, 2001-07-10, 2001-07-10, 200…
## $ manejo            <chr> "convencional", "convencional", "convencional", "con…
## $ tratamento        <chr> "AD_GN", "AD_GN", "AD_GN", "AD_GN", "AD_GN", "AD_GN"…
## $ revolvimento_solo <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL…
## $ data_preparo      <date> 2001-07-01, 2001-07-01, 2001-07-01, 2001-07-01, 200…
## $ conversao         <date> 1970-01-01, 1970-01-01, 1970-01-01, 1970-01-01, 197…
## $ cobertura         <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE…
## $ cultura           <chr> "milho_soja", "milho_soja", "milho_soja", "milho_soj…
## $ x                 <dbl> 0, 40, 80, 10, 25, 40, 55, 70, 20, 40, 60, 10, 70, 3…
## $ y                 <dbl> 0, 0, 0, 10, 10, 10, 10, 10, 20, 20, 20, 25, 25, 30,…
## $ longitude_muni    <dbl> 782062.7, 782062.7, 782062.7, 782062.7, 782062.7, 78…
## $ latitude_muni     <dbl> 7647674, 7647674, 7647674, 7647674, 7647674, 7647674…
## $ estado            <chr> "SP", "SP", "SP", "SP", "SP", "SP", "SP", "SP", "SP"…
## $ municipio         <chr> "Jaboticabal", "Jaboticabal", "Jaboticabal", "Jaboti…
## $ ID                <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1…
## $ prof              <chr> "0-0.1", "0-0.1", "0-0.1", "0-0.1", "0-0.1", "0-0.1"…
## $ FCO2              <dbl> 1.080, 0.825, 1.950, 0.534, 0.893, 0.840, 1.110, 1.8…
## $ Ts                <dbl> 18.73, 18.40, 19.20, 18.28, 18.35, 18.47, 19.10, 18.…
## $ Us                <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ pH                <dbl> 5.1, 5.1, 5.8, 5.3, 5.5, 5.7, 5.6, 6.4, 5.3, 5.8, 5.…
## $ MO                <dbl> 20, 24, 25, 23, 23, 21, 26, 23, 25, 24, 26, 20, 25, …
## $ P                 <dbl> 46, 26, 46, 78, 60, 46, 55, 92, 55, 60, 48, 71, 125,…
## $ K                 <dbl> 2.4, 2.2, 5.3, 3.6, 3.4, 2.9, 4.0, 2.3, 3.3, 3.6, 4.…
## $ Ca                <dbl> 25, 30, 41, 27, 33, 38, 35, 94, 29, 36, 37, 29, 50, …
## $ Mg                <dbl> 11, 11, 25, 11, 15, 20, 16, 65, 11, 17, 15, 11, 30, …
## $ H_Al              <dbl> 31, 31, 22, 28, 27, 22, 22, 12, 31, 28, 28, 31, 18, …
## $ SB                <dbl> 38.4, 43.2, 71.3, 41.6, 50.6, 60.9, 55.0, 161.3, 43.…
## $ CTC               <dbl> 69.4, 74.2, 93.3, 69.6, 77.9, 82.9, 77.0, 173.3, 74.…
## $ V                 <dbl> 55, 58, 76, 60, 65, 73, 71, 93, 58, 67, 67, 58, 82, …
## $ Ds                <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ Macro             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ Micro             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ VTP               <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ PLA               <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ AT                <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ SILTE             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ ARG               <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ HLIFS             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
```

Vamos conhecer, um pouco mais a nossa base de dados.

```
skimr::skim(data_fco2)
```

| | |
|---|---|
| Name | data_fco2 |
| Number of rows | 15397 |
| Number of columns | 39 |
| _______________________ | |
| Column type frequency: | |
| character | 7 |
| Date | 3 |
| logical | 2 |
| numeric | 27 |
| ________________________ | |
| Group variables | None |

Data summary

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| experimento | 0 | 1 | 8 | 8 | 0 | 2 | 0 |
| manejo | 0 | 1 | 6 | 15 | 0 | 10 | 0 |
| tratamento | 0 | 1 | 2 | 10 | 0 | 21 | 0 |
| cultura | 0 | 1 | 4 | 14 | 0 | 11 | 0 |

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| estado | 0 | 1 | 2 | 2 | 0 | 2 | 0 |
| municipio | 0 | 1 | 7 | 20 | 0 | 6 | 0 |
| prof | 0 | 1 | 5 | 7 | 0 | 2 | 0 |

**Variable type: Date**

| skim_variable | n_missing | complete_rate | min | max | median | n_unique |
|---|---|---|---|---|---|---|
| data | 0 | 1 | 2001-07-10 | 2019-12-01 | 2014-07-12 | 205 |
| data_preparo | 0 | 1 | 1986-03-01 | 2019-04-01 | 2002-01-01 | 14 |
| conversao | 0 | 1 | 1970-01-01 | 2009-07-03 | 1986-03-01 | 11 |

**Variable type: logical**

| skim_variable | n_missing | complete_rate | mean | count |
|---|---|---|---|---|
| revolvimento_solo | 0 | 1 | 0 | FAL: 15397 |
| cobertura | 0 | 1 | 1 | TRU: 15397 |

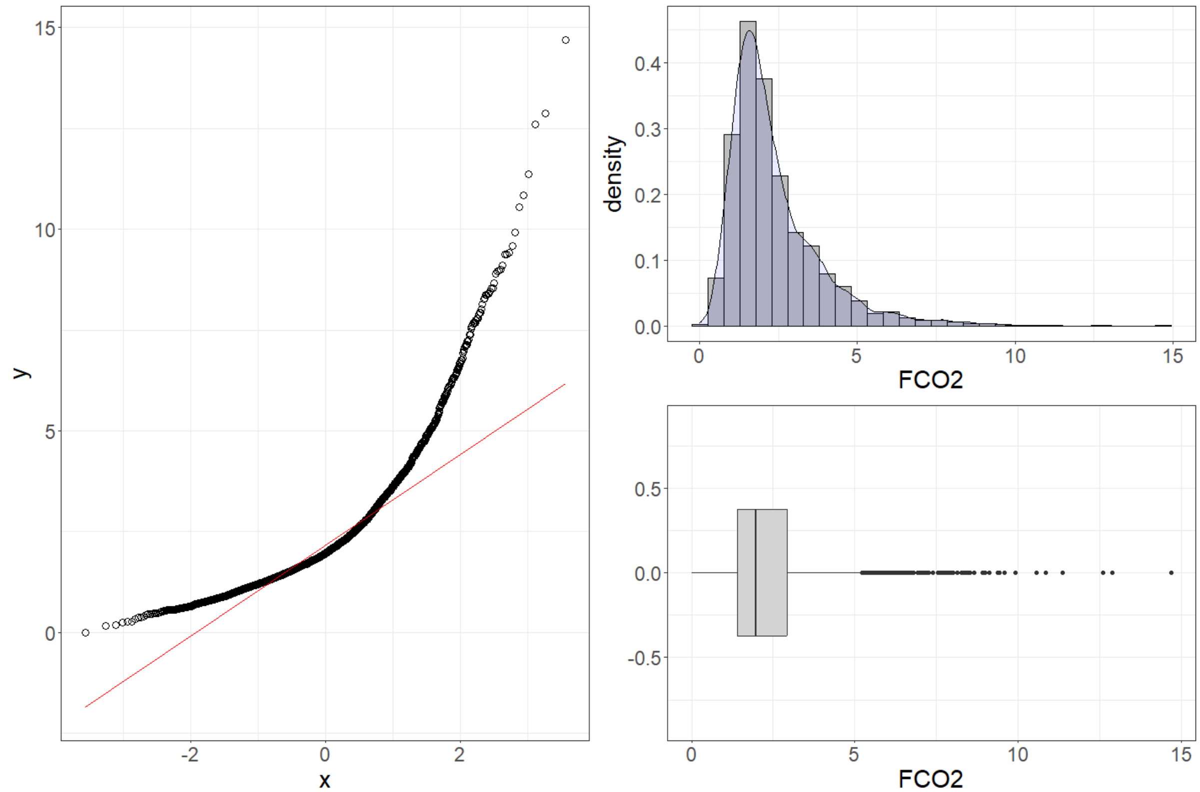
**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| x | 0 | 1.00 | 1392083.56 | 2923710.70 | 0.00 | 0.00 | 30.00 | 100.00 | 7749472.16 | |
| y | 0 | 1.00 | 495854.97 | 1722529.75 | 0.00 | 0.00 | 27.00 | 80.00 | 7630525.47 | |
| longitude_muni | 0 | 1.00 | 1067926.05 | 1796771.47 | 456798.63 | 458447.46 | 458447.46 | 792043.56 | 7638196.06 | |
| latitude_muni | 0 | 1.00 | 7231328.21 | 1754220.76 | 795907.06 | 7635356.70 | 7749398.84 | 7749821.85 | 7758831.37 | |
| ID | 0 | 1.00 | 40.52 | 31.52 | 1.00 | 13.00 | 35.00 | 60.00 | 141.00 | |
| FCO2 | 110 | 0.99 | 2.78 | 2.08 | -3.42 | 1.30 | 2.16 | 3.75 | 46.93 | |
| Ts | 317 | 0.98 | 21.84 | 6.76 | 1.00 | 19.33 | 22.50 | 26.15 | 195.63 | |
| Us | 1754 | 0.89 | 16.31 | 8.93 | 0.00 | 10.00 | 14.06 | 22.00 | 89.00 | |
| pH | 2802 | 0.82 | 4.64 | 1.13 | 3.50 | 4.00 | 4.50 | 5.15 | 52.00 | |
| MO | 1355 | 0.91 | 21.59 | 12.60 | 1.35 | 12.00 | 23.00 | 29.00 | 61.26 | |
| P | 1355 | 0.91 | 20.95 | 24.74 | 1.00 | 6.00 | 15.48 | 27.36 | 253.00 | |
| K | 1348 | 0.91 | 2.40 | 2.21 | 0.03 | 0.90 | 1.70 | 3.40 | 34.00 | |
| Ca | 1376 | 0.91 | 17.20 | 14.57 | 1.10 | 6.00 | 11.00 | 26.00 | 94.00 | |
| Mg | 1376 | 0.91 | 10.13 | 5.65 | 0.32 | 7.00 | 10.00 | 13.00 | 65.00 | |
| H_Al | 1362 | 0.91 | 46.89 | 29.38 | 0.00 | 26.00 | 42.29 | 72.00 | 121.00 | |
| SB | 1376 | 0.91 | 29.69 | 20.10 | 1.54 | 15.60 | 23.80 | 42.00 | 161.30 | |
| CTC | 1369 | 0.91 | 77.10 | 32.99 | 4.62 | 59.23 | 83.40 | 103.20 | 173.30 | |
| V | 1383 | 0.91 | 41.68 | 20.05 | 4.96 | 22.00 | 43.00 | 58.00 | 100.00 | |
| Ds | 3284 | 0.79 | 1.38 | 0.17 | 0.88 | 1.24 | 1.38 | 1.52 | 1.86 | |
| Macro | 3277 | 0.79 | 8.55 | 7.85 | -45.30 | 0.15 | 8.13 | 13.64 | 49.77 | |
| Micro | 3298 | 0.79 | 25.30 | 17.13 | 0.07 | 0.37 | 33.86 | 38.30 | 52.42 | |
| VTP | 3298 | 0.79 | 42.34 | 15.65 | -4.68 | 40.81 | 46.25 | 51.32 | 87.80 | |
| PLA | 3438 | 0.78 | 29.57 | 11.80 | -47.30 | 21.27 | 32.41 | 38.15 | 79.80 | |
| AT | 8083 | 0.48 | 1013.33 | 1358.81 | 11.72 | 236.00 | 593.62 | 816.00 | 4542.73 | |
| SILTE | 8048 | 0.48 | 229.26 | 336.37 | 1.26 | 50.87 | 73.65 | 188.00 | 1395.00 | |
| ARG | 8055 | 0.48 | 995.41 | 1560.32 | 27.19 | 173.27 | 403.69 | 609.50 | 5244.76 | |
| HLIFS | 10872 | 0.29 | 14590.11 | 17253.55 | 158.39 | 1110.15 | 2409.80 | 29707.78 | 84692.90 | |

## Alguns gráficos a respeito de nossa variável alvo, emissão de $CO_2$ do solo ($FCO_2$).

```
composition(FCO2,data_fco2)
```

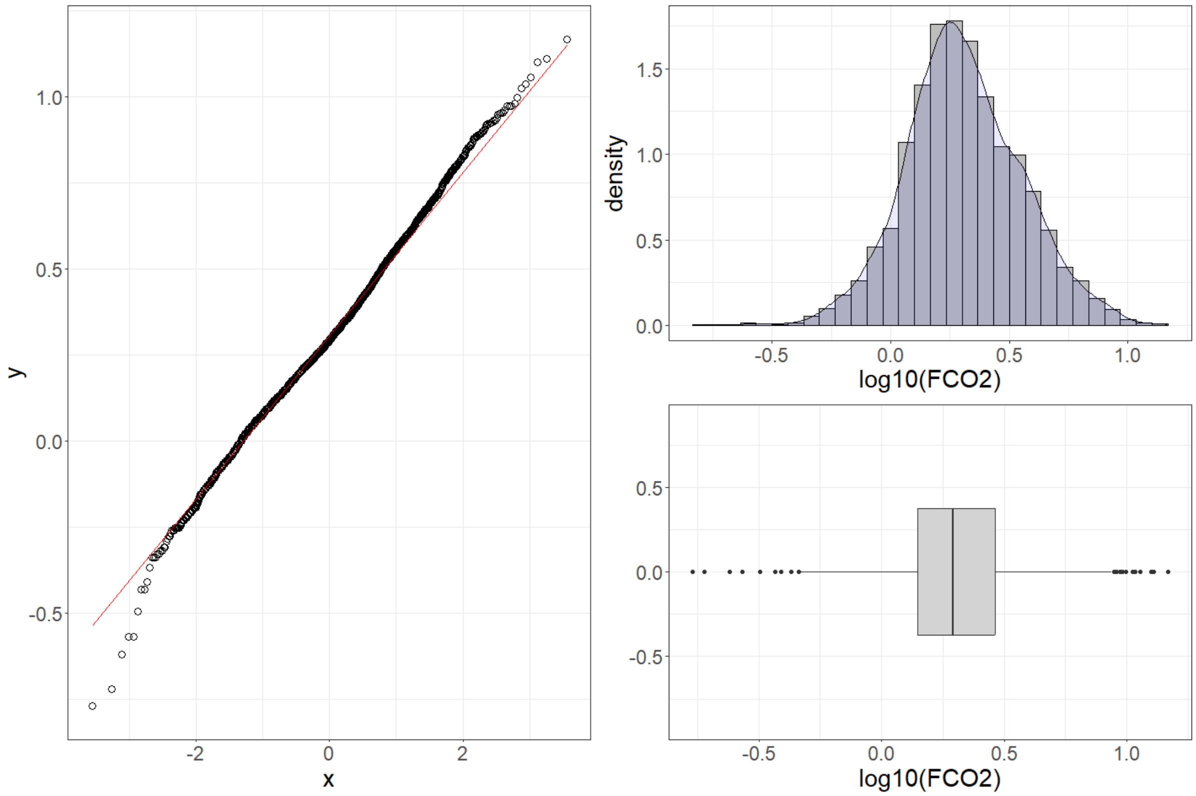
```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




## Aplicando a transformação logarítmica nos dados de $FCO_2$

```
composition(log10(FCO2) ,data_fco2)
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_qq()`).

## Warning: Removed 1 rows containing non-finite values (`stat_qq_line()`).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 1 rows containing non-finite values (`stat_density()`).

## Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
```




### Carregando os dados do pacote `{geobr}`

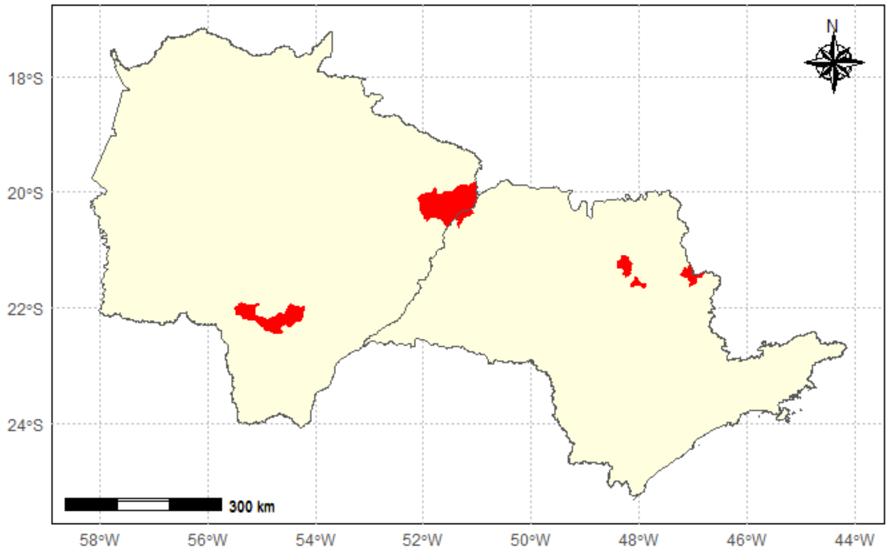
## Shape dos estados do Brasil

A fonte dos shapes abaixo utiizados é o pacote `{geobr}`, para maiores inofrmações acesse o link no GitHub, por comodidade, deixamos armazenados no repositório os arquivos que aqui serão utilizados.

```
# library(geobr)
# brasil_geobr <- geobr::read_country()
# estados <- read_state(code_state = "all")
# write_rds(estados,"data/estados.rds")
# write_rds(brasil_geobr,"data/brasil_geobr.rds")
estados <- read_rds("data/estados.rds")
```

## Shape dos municípios

```
# muni <- read_municipality()
# write_rds(muni,"data/municipios.rds")
muni <- read_rds("data/municipios.rds")
sp_ms <- muni %>%
  filter(abbrev_state == "SP" | abbrev_state == "MS")

fsp_ms<-if_else(sp_ms$name_muni == "Jaboticabal" |
             sp_ms$name_muni == "Guariba" |
             sp_ms$name_muni == "Padrópolis" |
             sp_ms$name_muni == "Rincão"|
             sp_ms$name_muni == "Mococa"|
             sp_ms$name_muni == "Ilha Solteira" |
             sp_ms$name_muni == "Aparecida Do Taboado" |
             sp_ms$name_muni == "Selvíria"|
             sp_ms$name_muni == "Dourados"
             ,"red","lightyellow")

sp_ms_ <- estados %>%
      filter(abbrev_state == "SP" | abbrev_state == "MS")

ggplot(sp_ms) +
  geom_sf(fill="lightyellow") +
  theme_minimal() +
  annotation_scale(location="bl") +
  geom_sf(data = sp_ms, fill=fsp_ms,col=fsp_ms) +
  # geom_sf(data = sp_ms, fill=fms,col=fms) +
  geom_sf(data = sp_ms_,fill="transparent") +
  tema_mapa()
```




## Conhecendo a base de dados de concentração de $CO_2$ atmosférico, oriundo do sensor orbital NASA-OCO2.

O satélite OCO-2 foi lançado em órbita em julho de 2014 pela NASA, e oferece um grande potencial nas estimativas dos fluxos de dióxido de carbono ($CO_2$). O satélite mede a concentração de $CO_2$ atmosférico indiretamente por meio da intensidade da radiação solar refletida em função da presença de dióxido de carbono em uma coluna de ar. Desta forma, faz-se a leitura em três faixas de comprimento de onda: a do O2, na faixa de $0,757$ a $0,775$ μm, e as do $CO_2$, que são subdividas em banda fraca $(1,594–1,627\ \mu m)$ e banda forte $(2,043–2,087\ \mu m)$.

Ele foi o primeiro satélite da NASA direcionado para o monitoramento dos fluxos de $CO_2$ atmosférico, sendo um dos mais recentes, e vem apresentando usos bem diversificados, mostrando-se capaz de monitorar as emissões de combustíveis fósseis, fotossíntese, e produção de biomassa.

```
glimpse(oco2_br)
```

```
## Rows: 37,387
## Columns: 18
## $ longitude                                          <dbl> -70.5, -…
## $ longitude_bnds                                     <chr> "-71.0:-…
## $ latitude                                           <dbl> -5.5, -4…
## $ latitude_bnds                                      <chr> "-6.0:-5…
## $ time_yyyymmddhhmmss                                <dbl> 2.014091…
## $ time_bnds_yyyymmddhhmmss                           <chr> "2014090…
## $ altitude_km                                        <dbl> 3307.8, …
## $ alt_bnds_km                                        <chr> "0.0:661…
## $ fluorescence_radiance_757nm_uncert_idp_ph_sec_1_m_2_sr_1_um_1 <dbl> 7.272876…
## $ fluorescence_radiance_757nm_idp_ph_sec_1_m_2_sr_1_um_1        <dbl> 2.537127…
## $ xco2_moles_mole_1                                  <dbl> 0.000394…
## $ aerosol_total_aod                                  <dbl> 0.148579…
## $ fluorescence_offset_relative_771nm_idp             <dbl> 0.016753…
## $ fluorescence_at_reference_ph_sec_1_m_2_sr_1_um_1   <dbl> 2.615319…
## $ fluorescence_radiance_771nm_idp_ph_sec_1_m_2_sr_1_um_1        <dbl> 3.088582…
## $ fluorescence_offset_relative_757nm_idp             <dbl> 0.013969…
## $ fluorescence_radiance_771nm_uncert_idp_ph_sec_1_m_2_sr_1_um_1 <dbl> 5.577878…
## $ XCO2                                               <dbl> 387.2781…
```

## Breve resumo do banco de dados de $X_{CO2}$

```
skimr::skim(oco2_br)
```

| | |
|---|---|
| Name | oco2_br |
| Number of rows | 37387 |
| Number of columns | 18 |
| _______________________ | |
| Column type frequency: | |
| character | 4 |

| | |
|---|---|
| numeric | 14 |
| ________________________ | |
| Group variables | None |

Data summary

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| longitude_bnds | 0 | 1 | 11 | 11 | 0 | 39 | 0 |
| latitude_bnds | 0 | 1 | 7 | 11 | 0 | 38 | 0 |
| time_bnds_yyyymmddhhmmss | 0 | 1 | 29 | 29 | 0 | 1765 | 0 |
| alt_bnds_km | 0 | 1 | 11 | 20 | 0 | 64 | 0 |

**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | |
|---|---|---|---|---|---|---|
| longitude | 0 | 1 | -5.120000e+01 | 8.280000e+00 | -7.350000e+01 | -5.65000 |
| latitude | 0 | 1 | -1.179000e+01 | 7.850000e+00 | -3.250000e+01 | -1.75000 |
| time_yyyymmddhhmmss | 0 | 1 | 2.016952e+13 | 1.564571e+10 | 2.014091e+13 | 2.01602 |
| altitude_km | 0 | 1 | 3.123200e+03 | 1.108800e+02 | 2.555700e+03 | 3.05635 |
| fluorescence_radiance_757nm_uncert_idp_ph_sec_1_m_2_sr_1_um_1 | 0 | 1 | 8.520719e+17 | 5.599367e+18 | -9.999990e+05 | 6.32325 |
| fluorescence_radiance_757nm_idp_ph_sec_1_m_2_sr_1_um_1 | 0 | 1 | -1.358150e+18 | 1.946775e+20 | -3.400736e+22 | 7.73515 |
| xco2_moles_mole_1 | 0 | 1 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.00000 |
| aerosol_total_aod | 0 | 1 | 4.828100e+02 | 7.848572e+04 | 2.000000e-02 | 1.10000 |
| fluorescence_offset_relative_771nm_idp | 0 | 1 | -4.814400e+02 | 2.193698e+04 | -9.999990e+05 | 1.00000 |
| fluorescence_at_reference_ph_sec_1_m_2_sr_1_um_1 | 0 | 1 | 1.296932e+18 | 2.245185e+18 | -8.394901e+19 | 2.01456 |
| fluorescence_radiance_771nm_idp_ph_sec_1_m_2_sr_1_um_1 | 0 | 1 | 1.904438e+18 | 2.236381e+18 | -8.453983e+19 | 9.69470 |
| fluorescence_offset_relative_757nm_idp | 0 | 1 | -3.744400e+02 | 1.934763e+04 | -9.999990e+05 | 1.00000 |
| fluorescence_radiance_771nm_uncert_idp_ph_sec_1_m_2_sr_1_um_1 | 0 | 1 | 5.235574e+17 | 7.580471e+16 | -9.999990e+05 | 4.69546 |
| XCO2 | 0 | 1 | 3.858900e+02 | 3.120000e+00 | 3.383400e+02 | 3.84410 |

## Manipulando a base `oco2_br` para criação das variáveis temporais e ajuste de unidade de xco2.

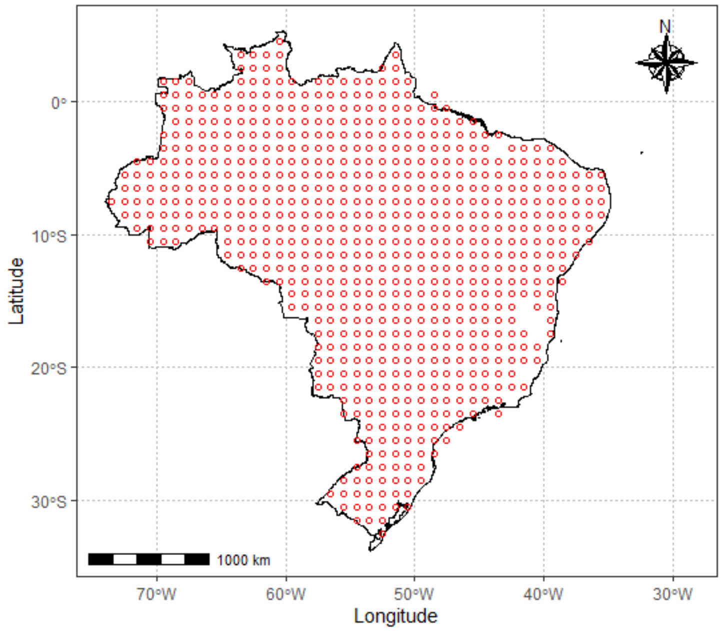
Inicialmente devemos transformar os dados de concentração de $CO_2$, variável xco2_moles_mole_1 para ppm em seguida devemos criar as variáveis de data a partir da variável time_yyyymmddhhmmss. Além disso, é necessário ajustar os valores de SIF, para compor a variável a partir dos dois sinais fornecidos pelo produto ("YU, L.; WEN, J.; CHANG, C. Y.; FRANKENBERG, C.; SUN, Y. High-Resolution Global Contiguous SIF of OCO-2. **Geophysical Research Letters**, v. 46, n. 3, p. 1449-1458, 2019.").

```
oco2_br <- oco2_br  %>%
        mutate(
           xco2 = xco2_moles_mole_1*1e06,
           data = ymd_hms(time_yyyymmddhhmmss),
           ano = year(data),
           mes = month(data),
           dia = day(data),
           dia_semana = wday(data),
           SIF = (fluorescence_radiance_757nm_idp_ph_sec_1_m_2_sr_1_um_1*2.6250912*10^(-19)  + 1.5*fluorescence_r
adiance_771nm_idp_ph_sec_1_m_2_sr_1_um_1* 2.57743*10^(-19))/2
           )
```

Mapa das leituras do satélite OCO2-NASA

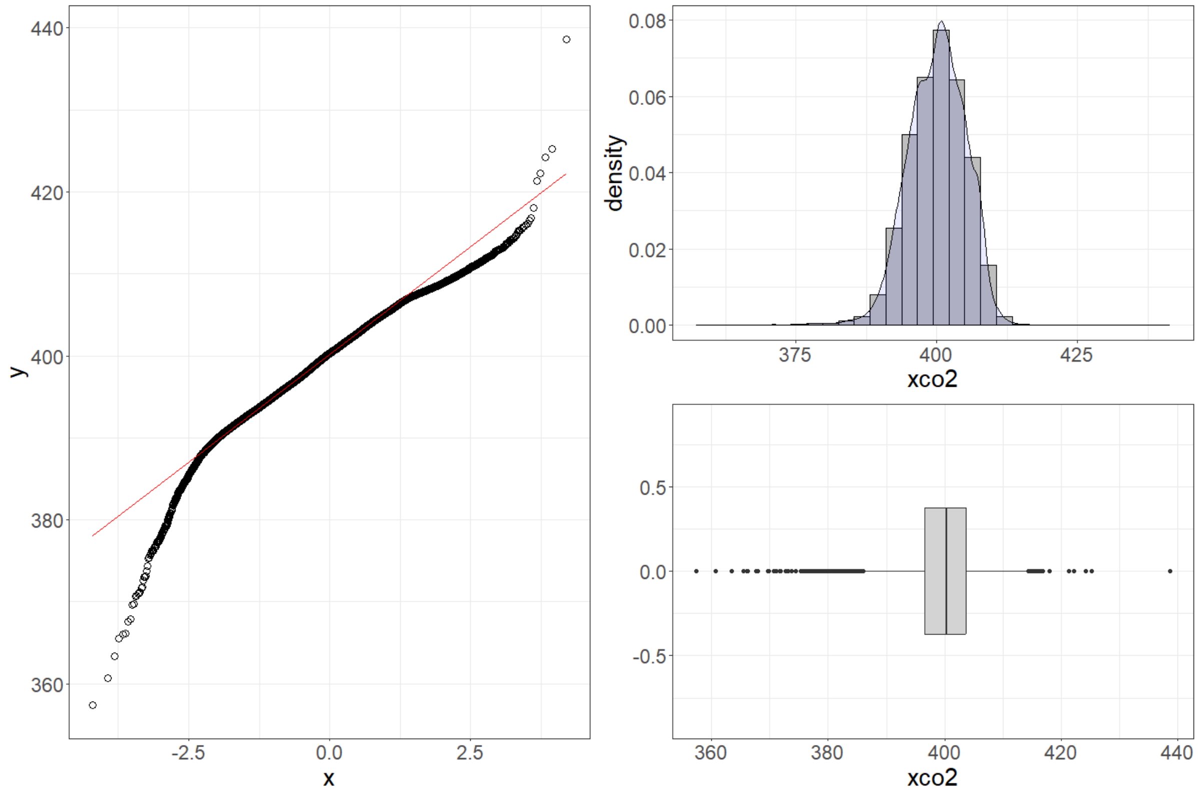
```
brasil_geobr <- read_rds("data/brasil_geobr.rds")
brasil_geobr %>%
  ggplot() +
  geom_sf(fill="white", color="black",
          size=.15, show.legend = FALSE) +
  tema_mapa() +
  geom_point(data=oco2_br %>%
                        sample_n(20000) ,
             aes(x=longitude,y=latitude),
             shape=1,
             col="red",
             alpha=01)+
  labs(x="Longitude",y="Latitude")
```

```
## Scale on map varies by more than 10%, scale bar may be inaccurate
```



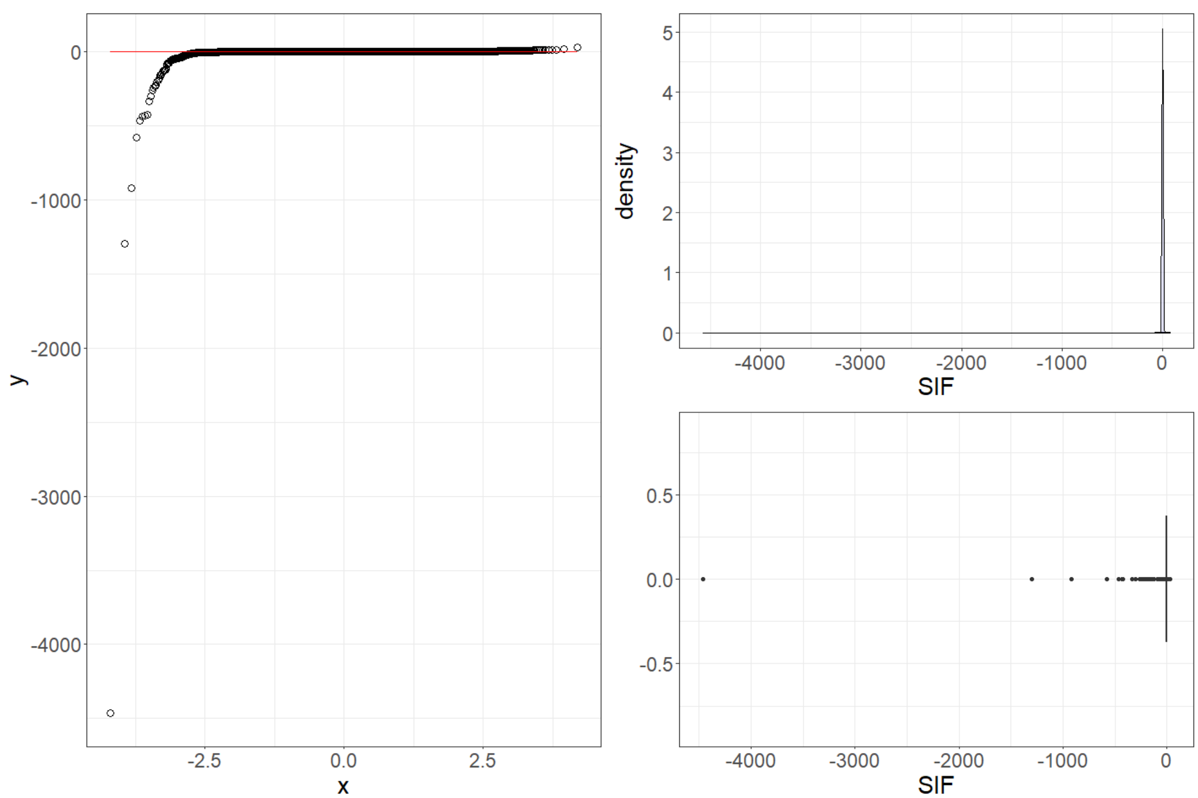

```
composition(xco2,oco2_br)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
composition(SIF,oco2_br)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




Necessário tratamento dos dados de SIF

```
oco2_br %>% filter (SIF > 0) %>%  pull(SIF) %>% summary
```

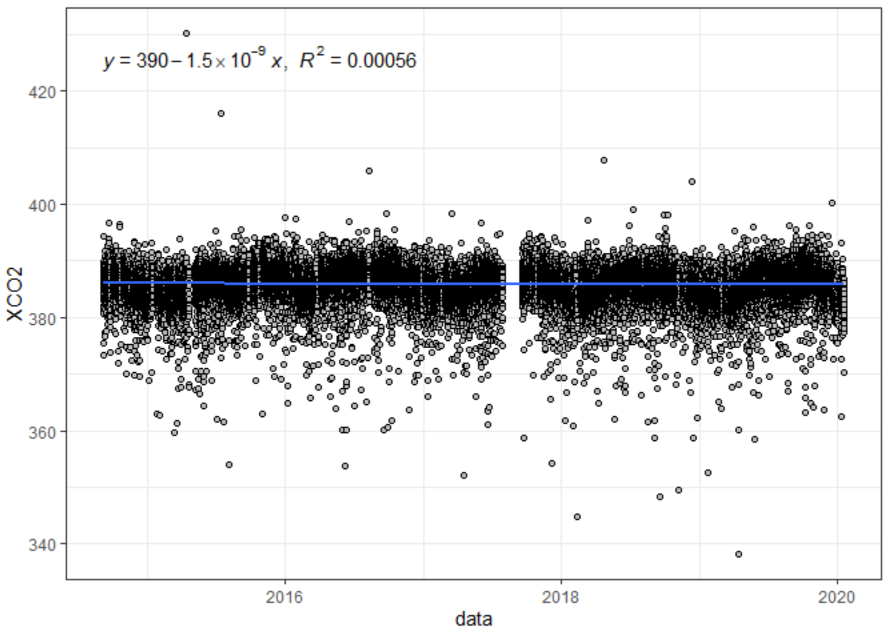
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##  0.00002  0.38094  0.64297  0.70352  0.89260 31.96757
```

```
sif_median <- 0.64297
oco2_br <- oco2_br %>%
  mutate(SIF = ifelse(SIF > 0, SIF, sif_median))
```

Existe uma tendência de aumento monotônica mundial da concentração de $CO_2$ na atmosfera, assim, ela deve ser retirada para podermos observar as tendências regionais. Observe que o sinal na variável `XCO2` não apresenta a tendência descrita.

```
oco2_br  %>%
  ggplot(aes(x=data,y=XCO2)) +
  geom_point(shape=21,color="black",fill="gray") +
  geom_smooth(method = "lm") +
  stat_regline_equation(ggplot2::aes(
  label =  paste(..eq.label.., ..rr.label.., sep = "*plain(\",\")~~")))
```

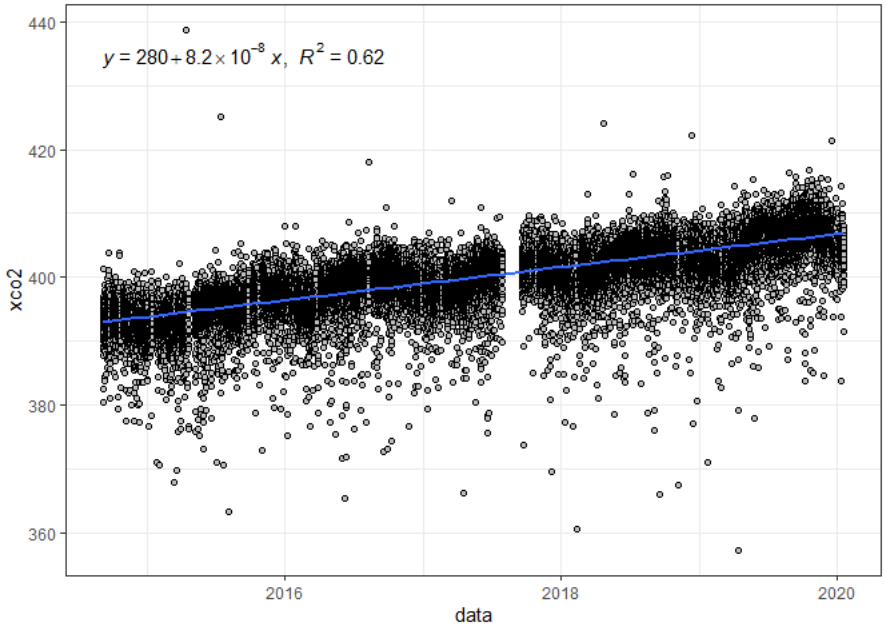
```
## `geom_smooth()` using formula = 'y ~ x'
```




Compare com os dados da variáveis `xco2` que apresenta a tendência de crescimento monotônica.

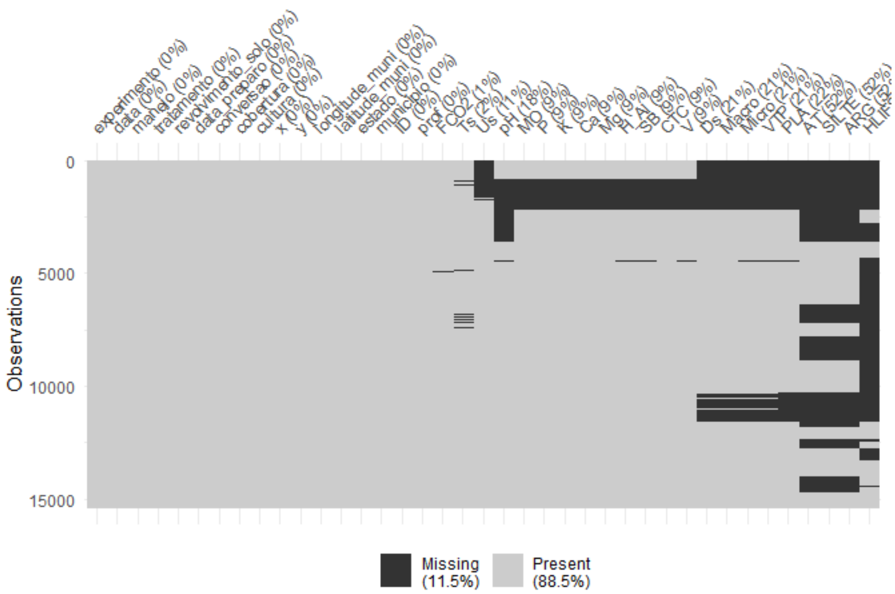
```
oco2_br  %>%
  ggplot(aes(x=data,y=xco2)) +
  geom_point(shape=21,color="black",fill="gray") +
  geom_smooth(method = "lm") +
  stat_regline_equation(ggplot2::aes(
  label =  paste(..eq.label.., ..rr.label.., sep = "*plain(\",\")~~")))
```

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




Agora, deve-se vizualizar os dados perdidos nas bases

```
visdat::vis_miss(data_fco2)
```




# Listando as datas dos arquivos

```
lista_data_fco2 <- unique(data_fco2$data)
lista_data_oco2 <- unique(oco2_br$data)
datas_fco2 <- paste0(lubridate::year(lista_data_fco2),"-",lubridate::month(lista_data_fco2)) %>% unique()

datas_oco2 <- paste0(lubridate::year(lista_data_oco2),"-",lubridate::month(lista_data_oco2)) %>% unique()
datas <- datas_fco2[datas_fco2 %in% datas_oco2]
```

Chaves para mesclagem

```
fco2 <- data_fco2 %>%
  mutate(ano_mes = paste0(lubridate::year(data),"-",lubridate::month(data))) %>%
  dplyr::filter(ano_mes %in% datas)

xco2 <- oco2_br %>%
  mutate(ano_mes=paste0(ano,"-",mes)) %>%
  dplyr::filter(ano_mes %in% datas)
```

Coordenadas das cidades

```
unique(xco2$ano_mes)[unique(xco2$ano_mes) %>% order()] ==
unique(fco2$ano_mes)[unique(fco2$ano_mes) %>% order()]
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

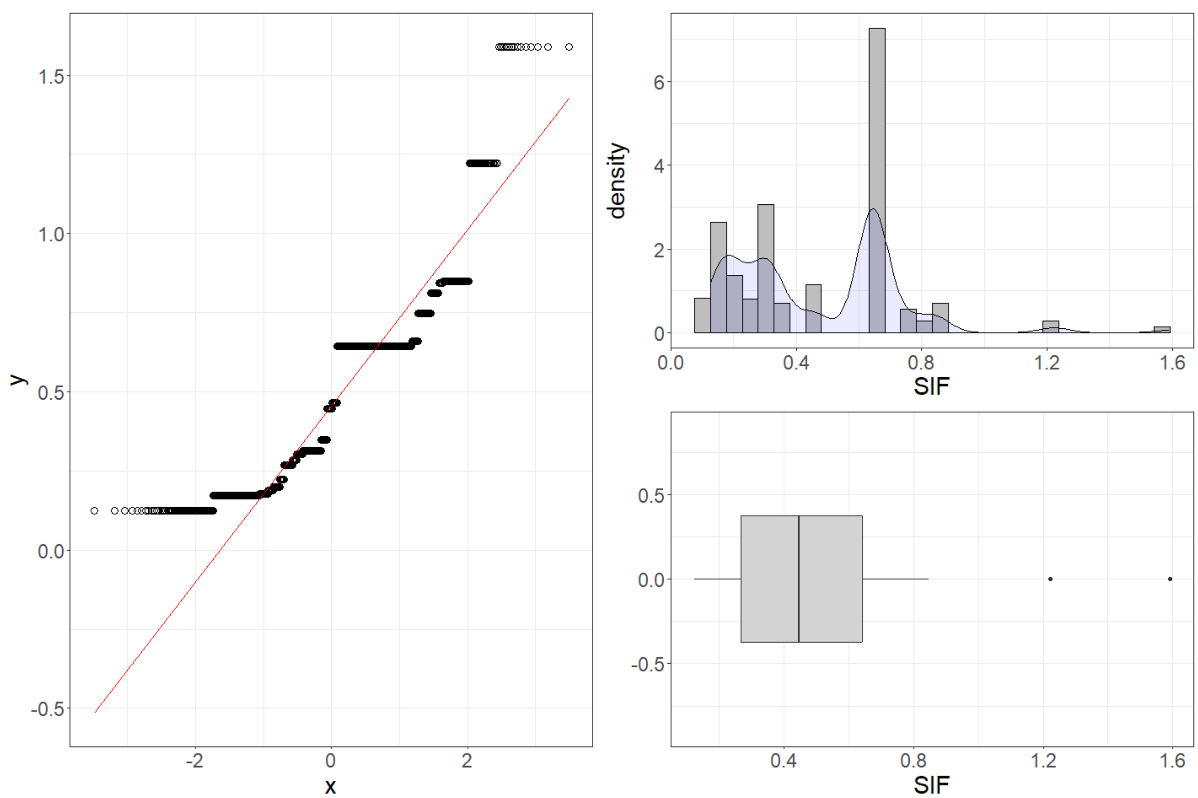
```
data_set <- left_join(fco2 %>%
            mutate(ano = lubridate::year(data),
                   mes = lubridate::month(data)
                   ),
          xco2 %>%
            select(data,mes,dia,longitude,latitude,XCO2,SIF,fluorescence_radiance_757nm_idp_ph_sec_1_m_2_sr_1_um_
1,fluorescence_radiance_771nm_idp_ph_sec_1_m_2_sr_1_um_1, ano_mes), by = "ano_mes") %>%
  mutate(dist = sqrt((longitude-(-51.423519))^2+(latitude-(-20.362911))^2),
         # SIF = (fluorescence_radiance_757nm_idp_ph_sec_1_m_2_sr_1_um_1*2.6250912*10^(-19)  + 1.5*fluorescence_r
adiance_771nm_idp_ph_sec_1_m_2_sr_1_um_1* 2.57743*10^(-19))/2
         )
```

```
## Warning in left_join(fco2 %>% mutate(ano = lubridate::year(data), mes = lubridate::month(data)), : Detected an
unexpected many-to-many relationship between `x` and `y`.
## ℹ Row 1 of `x` matches multiple rows in `y`.
## ℹ Row 1 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
data_set<-data_set %>%
  select(-fluorescence_radiance_757nm_idp_ph_sec_1_m_2_sr_1_um_1, -fluorescence_radiance_771nm_idp_ph_sec_1_m_2_s
r_1_um_1 )  %>%
  filter(dist <= .16, FCO2 <= 30 )
```

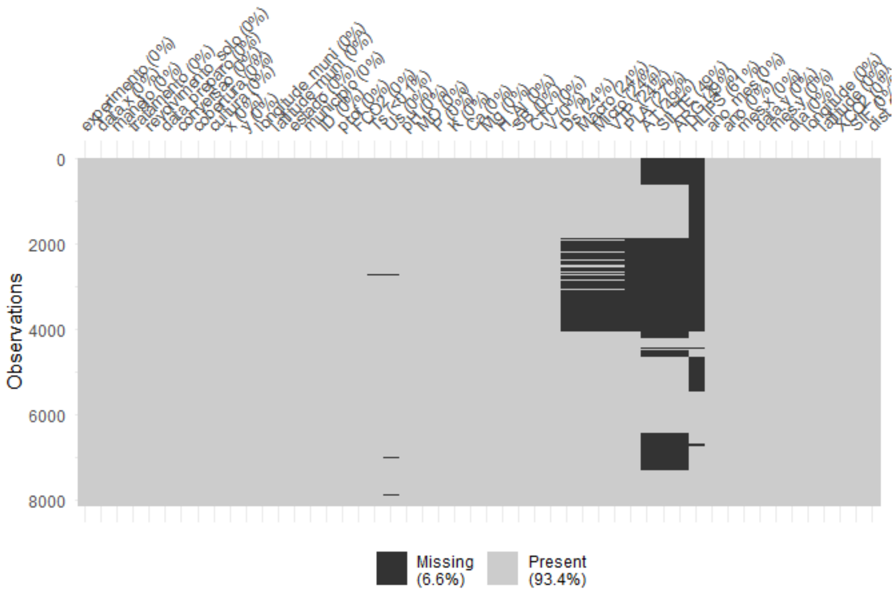
```
composition(SIF,data_set)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Definindo o plano de multisession
future::plan("multisession")
```

```
visdat::vis_miss(data_set)
```

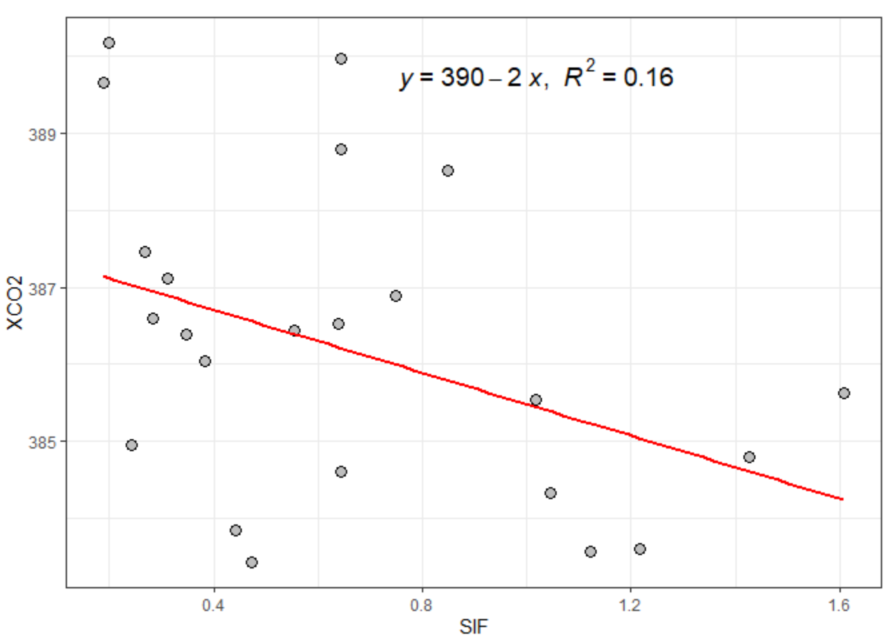



```
tab_medias <- data_set %>%
  # mutate(SIF = ifelse(SIF <=0, mean(data_set$SIF, na.rm=TRUE),SIF)) %>%
  group_by(ano_mes, cultura) %>%
  summarise(FCO2 = mean(FCO2, na.rm=TRUE),
            XCO2 = mean(XCO2, na.rm=TRUE),
            SIF = mean(SIF, na.rm=TRUE))
```

```
## `summarise()` has grouped output by 'ano_mes'. You can override using the
## `.groups` argument.
```

```
tab_medias %>% filter(SIF > 0) %>%
  ggplot(aes(y=XCO2, x=SIF)) +
  geom_point(size=3, shape=21, fill="gray")+
  geom_smooth(method = "lm", se=FALSE,
              ldw=2,color="red")+
  stat_regline_equation(aes(
    label =  paste(..eq.label.., ..rr.label.., sep = "*plain(\",\")~~")),size=5, label.x.npc = .4)
```

```
## Warning in geom_smooth(method = "lm", se = FALSE, ldw = 2, color = "red"):
## Ignoring unknown parameters: `ldw`

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



```
lm(XCO2 ~ SIF,
          data = tab_medias %>% filter(SIF > 0) ) %>%
  summary.lm()
```

```
##
## Call:
## lm(formula = XCO2 ~ SIF, data = tab_medias %>% filter(SIF > 0))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1219 -1.5150  0.0957  0.6931  3.7631
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 387.5248     0.4803 806.801  < 2e-16 ***
## SIF          -2.0433     0.6320  -3.233  0.00211 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.825 on 53 degrees of freedom
## Multiple R-squared:  0.1647, Adjusted R-squared:  0.149
## F-statistic: 10.45 on 1 and 53 DF,  p-value: 0.00211
```

```
lm(XCO2 ~ SIF + SIF2,
          data = tab_medias %>% filter(SIF > 0) %>% mutate(SIF2 = SIF^2))  %>%
  summary.lm()
```

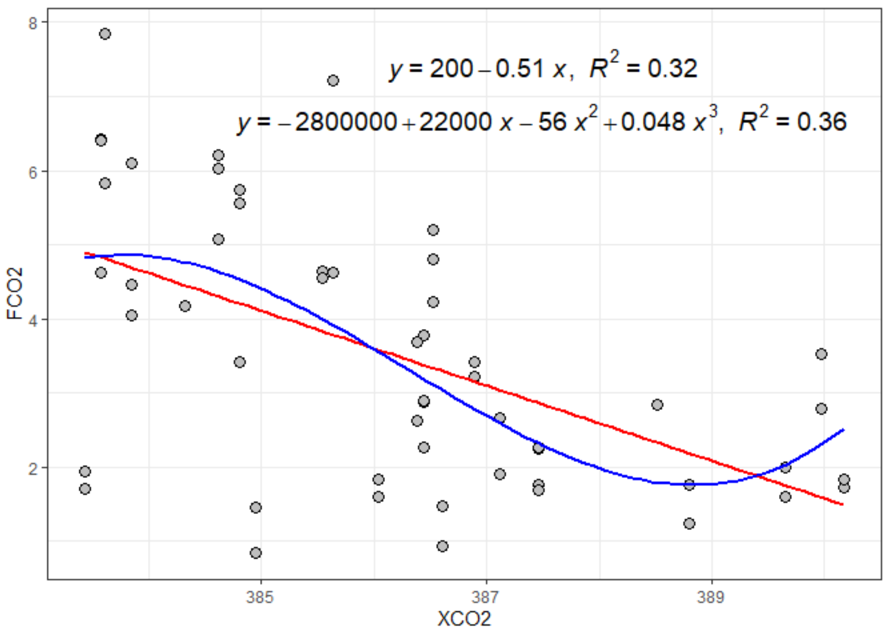
```
##
## Call:
## lm(formula = XCO2 ~ SIF + SIF2, data = tab_medias %>% filter(SIF >
##     0) %>% mutate(SIF2 = SIF^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0362 -1.3729  0.1043  0.6334  3.9880
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 388.2875     0.9331 416.126   <2e-16 ***
## SIF          -4.5561     2.7099  -1.681   0.0987 .
## SIF2          1.5192     1.5931   0.954   0.3447
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.826 on 52 degrees of freedom
## Multiple R-squared:  0.1791, Adjusted R-squared:  0.1475
## F-statistic: 5.672 on 2 and 52 DF,  p-value: 0.005913
```

```
lm(XCO2 ~ SIF + SIF2 + SIF3,
          data = tab_medias %>% filter(SIF > 0) %>% mutate(SIF2 = SIF^2,
                                                           SIF3 = SIF^3))  %>%
  summary.lm()
```

```
##
## Call:
## lm(formula = XCO2 ~ SIF + SIF2 + SIF3, data = tab_medias %>%
##     filter(SIF > 0) %>% mutate(SIF2 = SIF^2, SIF3 = SIF^3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9810 -1.3315  0.1091  0.7363  4.0294
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  388.654      1.734 224.086   <2e-16 ***
## SIF           -6.484      8.135  -0.797    0.429
## SIF2           4.208     10.808   0.389    0.699
## SIF3          -1.054      4.189  -0.252    0.802
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.843 on 51 degrees of freedom
## Multiple R-squared:  0.1801, Adjusted R-squared:  0.1319
## F-statistic: 3.734 on 3 and 51 DF,  p-value: 0.01676
```

```
formula <- y ~ poly(x, 3, raw = TRUE)
tab_medias %>% filter(SIF > 0) %>%
  ggplot(aes(x=XCO2, y=FCO2)) +
  geom_point(size=3, shape=21, fill="gray")+
  geom_smooth(method = "lm", se=FALSE,
              ldw=2,color="red") +
  stat_regline_equation(aes(
    label =  paste(..eq.label.., ..rr.label.., sep = "*plain(\",\")~~")),size=5,label.x.npc = .4) +
  stat_smooth(method="lm", se=TRUE, fill=NA,
                formula=y ~ poly(x, 3, raw=TRUE),colour="blue") +
  stat_regline_equation(aes(
    label =  paste(..eq.label.., ..rr.label.., sep = "*plain(\",\")~~")), formula = y ~ poly(x, 3, raw = TRUE)
    ,size=5,label.x.npc = .2,label.y.npc = .85)
```

```
## Warning in geom_smooth(method = "lm", se = FALSE, ldw = 2, color = "red"):
## Ignoring unknown parameters: `ldw`

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




```
lm(XCO2 ~ FCO2,
           data = tab_medias %>% filter(SIF > 0) ) %>%
  summary.lm()
```

```
##
## Call:
## lm(formula = XCO2 ~ FCO2, data = tab_medias %>% filter(SIF >
##     0))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8763 -0.9973  0.0098  0.7118  3.7944
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 388.3840     0.4956 783.599  < 2e-16 ***
## FCO2         -0.6252     0.1263  -4.952 7.87e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.651 on 53 degrees of freedom
## Multiple R-squared:  0.3163, Adjusted R-squared:  0.3034
## F-statistic: 24.52 on 1 and 53 DF,  p-value: 7.867e-06
```

```
lm(XCO2 ~ FCO2 + FCO22,
           data = tab_medias %>% filter(SIF > 0) %>% mutate(FCO22 = FCO2^2))  %>%
  summary.lm()
```

```
##
## Call:
## lm(formula = XCO2 ~ FCO2 + FCO22, data = tab_medias %>% filter(SIF >
##     0) %>% mutate(FCO22 = FCO2^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7982 -0.8280  0.0033  0.6508  3.6498
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 387.81592    1.01518 382.018   <2e-16 ***
## FCO2         -0.25984    0.58283  -0.446    0.658
## FCO22        -0.04630    0.07208  -0.642    0.523
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.66 on 52 degrees of freedom
## Multiple R-squared:  0.3217, Adjusted R-squared:  0.2956
## F-statistic: 12.33 on 2 and 52 DF,  p-value: 4.141e-05
```

```
lm(XCO2 ~ FCO2 + FCO22 + FCO23,
           data = tab_medias %>% filter(SIF > 0) %>% mutate(FCO22 = FCO2^2,
                                                             FCO23 = FCO2^3))  %>%
  summary.lm()
```

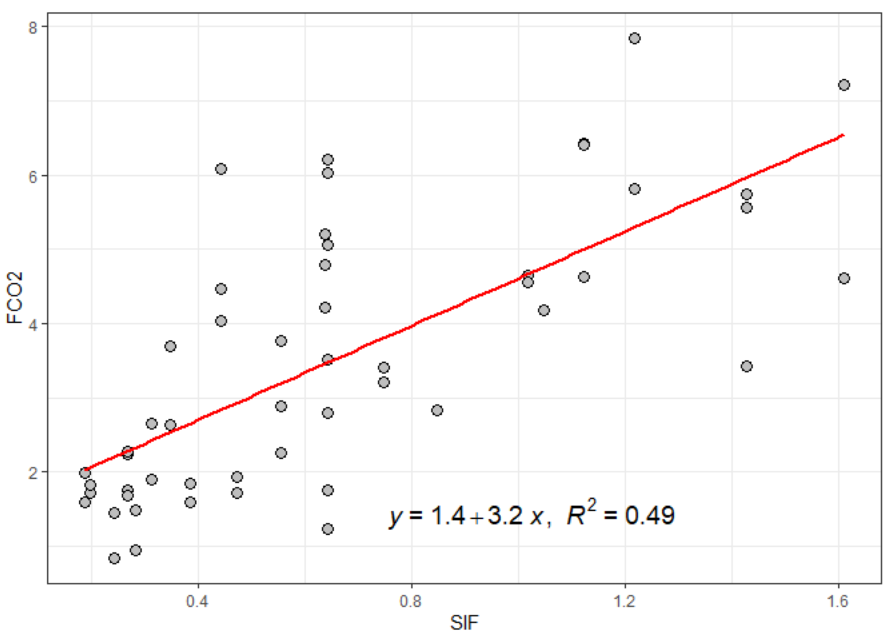
```
##
## Call:
## lm(formula = XCO2 ~ FCO2 + FCO22 + FCO23, data = tab_medias %>%
##     filter(SIF > 0) %>% mutate(FCO22 = FCO2^2, FCO23 = FCO2^3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9322 -0.9353  0.0675  0.6658  3.3518
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 384.0046     1.9473 197.201   <2e-16 ***
## FCO2          3.5011     1.7542   1.996   0.0513 .
## FCO22        -1.0680     0.4568  -2.338   0.0234 *
## FCO23         0.0810     0.0358   2.263   0.0279 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.598 on 51 degrees of freedom
## Multiple R-squared:  0.3836, Adjusted R-squared:  0.3473
## F-statistic: 10.58 on 3 and 51 DF,  p-value: 1.616e-05
```

```
lm(XCO2 ~ FCO2 + FCO22 + FCO23+ FCO24,
           data = tab_medias %>% filter(SIF > 0) %>% mutate(FCO22 = FCO2^2,
                                                             FCO23 = FCO2^3,
                                                             FCO24 = FCO2^4))  %>%
  summary.lm()
```

```
##
## Call:
## lm(formula = XCO2 ~ FCO2 + FCO22 + FCO23 + FCO24, data = tab_medias %>%
##     filter(SIF > 0) %>% mutate(FCO22 = FCO2^2, FCO23 = FCO2^3,
##     FCO24 = FCO2^4))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1290 -0.7578 -0.0005  0.8101  3.5196
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 380.75370    3.54284 107.471   <2e-16 ***
## FCO2          7.96676    4.42977   1.798   0.0781 .
## FCO22        -3.04007    1.85387  -1.640   0.1073
## FCO23         0.42194    0.31271   1.349   0.1833
## FCO24        -0.02000    0.01822  -1.097   0.2777
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.595 on 50 degrees of freedom
## Multiple R-squared:  0.3981, Adjusted R-squared:  0.3499
## F-statistic: 8.267 on 4 and 50 DF,  p-value: 3.373e-05
```

```
tab_medias %>% filter(SIF > 0) %>%
  ggplot(aes(y=FCO2, x=SIF)) +
  geom_point(size=3, shape=21, fill="gray")+
  geom_smooth(method = "lm", se=FALSE,
              ldw=2,color="red")+
  stat_regline_equation(aes(
    label =  paste(..eq.label.., ..rr.label.., sep = "*plain(\",\")~~")),size=5,label.x.npc = .4,label.y.npc = .
1)
```

```
## Warning in geom_smooth(method = "lm", se = FALSE, ldw = 2, color = "red"):
## Ignoring unknown parameters: `ldw`

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




```
lm(FCO2 ~ SIF,
           data = tab_medias %>% filter(SIF > 0) ) %>%
  summary.lm()
```

```
##
## Call:
## lm(formula = FCO2 ~ SIF, data = tab_medias %>% filter(SIF > 0))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5468 -0.7814 -0.2975  0.7166  3.2571
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.4350     0.3371   4.257 8.50e-05 ***
## SIF           3.1750     0.4436   7.158 2.51e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.281 on 53 degrees of freedom
## Multiple R-squared:  0.4915, Adjusted R-squared:  0.4819
## F-statistic: 51.23 on 1 and 53 DF,  p-value: 2.514e-09
```

```
lm(FCO2 ~ SIF + SIF2,
          data = tab_medias %>% filter(SIF > 0) %>% mutate(SIF2 = SIF^2))  %>%
  summary.lm()
```

```
##
## Call:
## lm(formula = FCO2 ~ SIF + SIF2, data = tab_medias %>% filter(SIF >
##     0) %>% mutate(SIF2 = SIF^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5244 -0.8894 -0.2424  0.7600  3.1917
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4461     0.6409   0.696   0.4895
## SIF           6.4331     1.8613   3.456   0.0011 **
## SIF2         -1.9698     1.0942  -1.800   0.0776 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.254 on 52 degrees of freedom
## Multiple R-squared:  0.5214, Adjusted R-squared:  0.5029
## F-statistic: 28.32 on 2 and 52 DF,  p-value: 4.791e-09
```

```
lm(FCO2 ~ SIF + SIF2 + SIF3,
          data = tab_medias %>% filter(SIF > 0) %>% mutate(SIF2 = SIF^2,
                                                            SIF3 = SIF^3))  %>%
  summary.lm()
```

```
##
## Call:
## lm(formula = FCO2 ~ SIF + SIF2 + SIF3, data = tab_medias %>%
##     filter(SIF > 0) %>% mutate(SIF2 = SIF^2, SIF3 = SIF^3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5431 -0.8731 -0.2585  0.7704  3.1687
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2809     1.1917   0.236    0.815
## SIF           7.3019     5.5894   1.306    0.197
## SIF2         -3.1818     7.4263  -0.428    0.670
## SIF3          0.4750     2.8779   0.165    0.870
##
## Residual standard error: 1.266 on 51 degrees of freedom
## Multiple R-squared:  0.5216, Adjusted R-squared:  0.4935
## F-statistic: 18.54 on 3 and 51 DF,  p-value: 2.901e-08
```

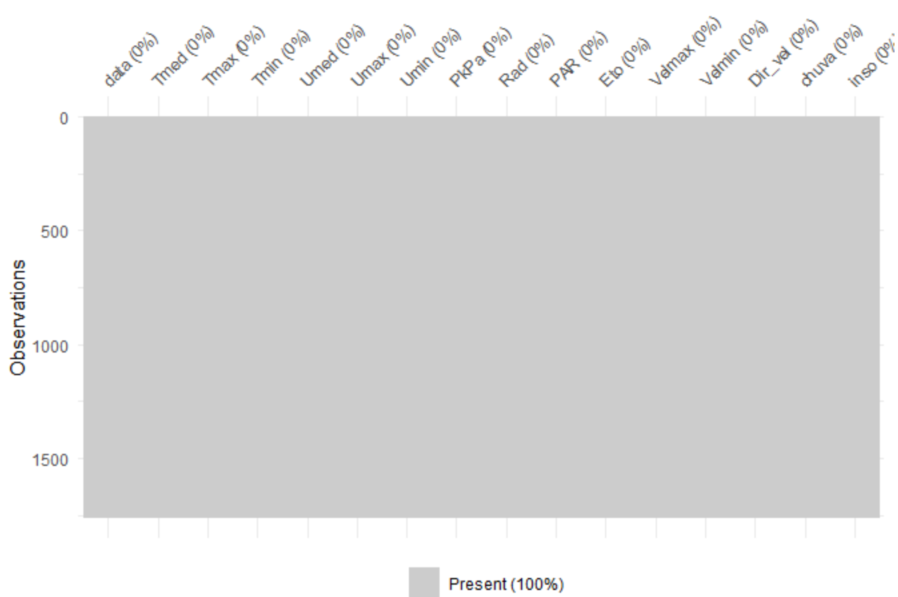
```
data_set_temporal <- data_set %>%
  filter(experimento == "Temporal")

data_set_espacial <- data_set %>%
  filter(experimento == "Espacial")
```

# Carregando dados Meteorológicos de Ilha Solteira

```
dados_estacao <- read_excel("data-raw/xlsx/estacao_meteorologia_ilha_solteira.xlsx", na = "NA")
glimpse(dados_estacao)
```

```
## Rows: 1,826
## Columns: 16
## $ data    <dttm> 2015-01-01, 2015-01-02, 2015-01-03, 2015-01-04, 2015-01-05, 2…
## $ Tmed    <dbl> 30.5, 30.0, 26.8, 27.1, 27.0, 27.6, 30.2, 28.2, 28.5, 29.9, 30…
## $ Tmax    <dbl> 36.5, 36.7, 35.7, 34.3, 33.2, 36.4, 37.2, 32.4, 37.1, 38.1, 38…
## $ Tmin    <dbl> 24.6, 24.5, 22.9, 22.7, 22.3, 22.8, 22.7, 24.0, 23.0, 23.3, 24…
## $ Umed    <dbl> 66.6, 70.4, 82.7, 76.8, 81.6, 75.5, 65.8, 70.0, 72.9, 67.6, 66…
## $ Umax    <dbl> 89.6, 93.6, 99.7, 95.0, 98.3, 96.1, 99.2, 83.4, 90.7, 97.4, 90…
## $ Umin    <dbl> 42.0, 44.2, 52.9, 43.8, 57.1, 47.5, 34.1, 57.4, 42.7, 38.3, 37…
## $ PkPa    <dbl> 97.2, 97.3, 97.4, 97.5, 97.4, 97.5, 97.4, 97.4, 97.4, 97.4, 97…
## $ Rad     <dbl> 23.6, 24.6, 20.2, 21.4, 17.8, 19.2, 27.0, 15.2, 21.6, 24.3, 24…
## $ PAR     <dbl> 496.6, 513.3, 430.5, 454.0, 378.2, 405.4, 565.7, 317.2, 467.5,…
## $ Eto     <dbl> 5.7, 5.8, 4.9, 5.1, 4.1, 4.8, 6.2, 4.1, 5.5, 5.7, 5.9, 6.1, 6.…
## $ Velmax  <dbl> 6.1, 4.8, 12.1, 6.2, 5.1, 4.5, 4.6, 5.7, 5.8, 5.2, 5.2, 4.7, 6…
## $ Velmin  <dbl> 1.0, 1.0, 1.2, 1.0, 0.8, 0.9, 0.9, 1.5, 1.2, 0.8, 0.8, 1.2, 1.…
## $ Dir_vel <dbl> 17.4, 261.9, 222.0, 25.0, 56.9, 74.9, 53.4, 89.0, 144.8, 303.9…
## $ chuva   <dbl> 0.0, 0.0, 3.3, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.…
## $ inso    <dbl> 7.9, 8.7, 5.2, 6.2, 3.4, 4.5, 10.5, 1.3, 6.3, 8.4, 8.6, 7.9, 1…
```

```
dados_estacao <- dados_estacao %>%
                   drop_na()
visdat::vis_miss(dados_estacao)
```




```
data_set_est_isa <- left_join(data_set %>%
                         rename(data=data.x), dados_estacao, by = "data") %>%                          mutate(ra
nge_T = Tmax-Tmin)
```

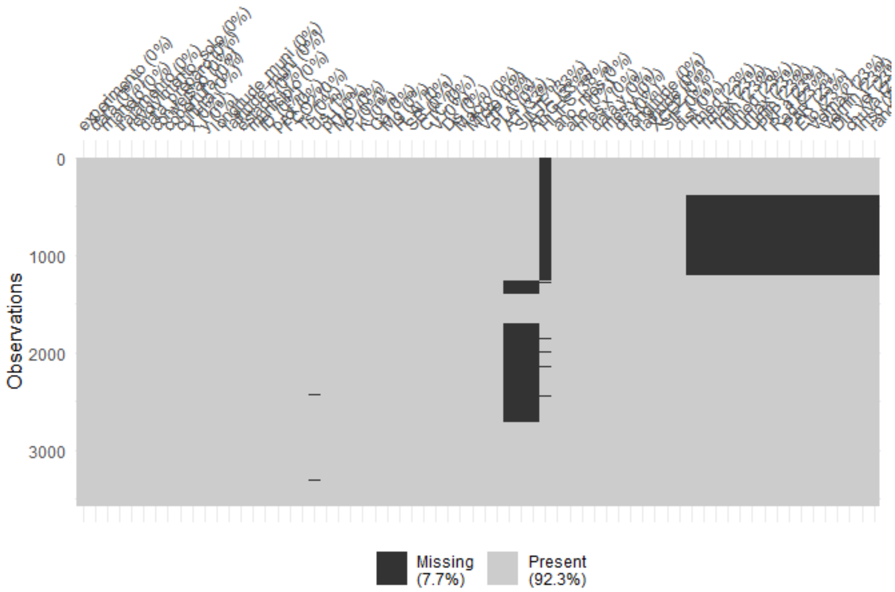
```
data_set_temporal <- data_set_est_isa %>%
  filter(experimento == "Temporal")

data_set_espacial <- data_set_est_isa %>%
  filter(experimento == "Espacial")
```

# Quarta Aproximação

- Alvo: FCO2 - temporal
- restrição dados após 2014
- Features: Atributos do Solo + Xco2 e SIF + Dados da Estação de ISA
- Modelo mais simples e geral
- Teste de três métodos baseados em árvores de decisão

## Visualização do banco de dados

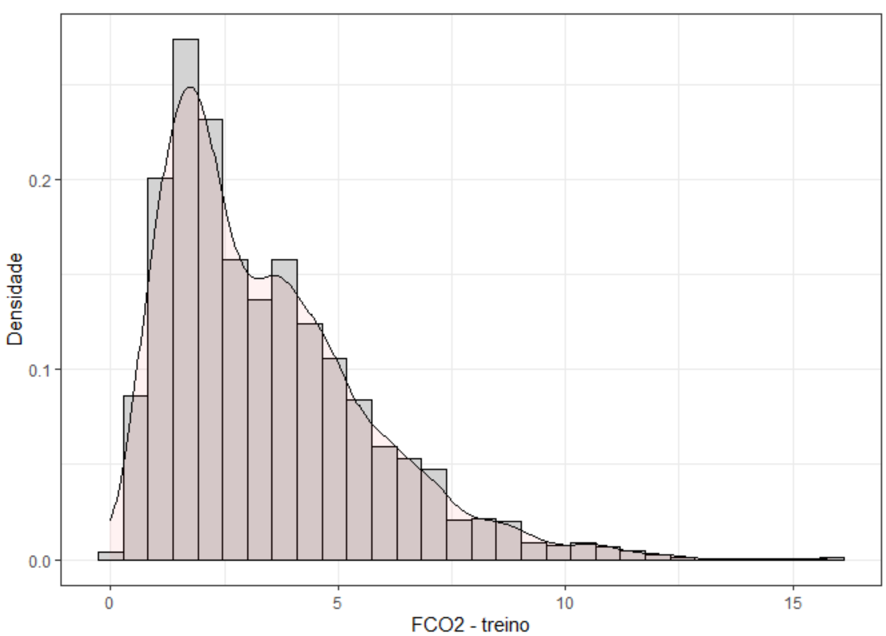
```
visdat::vis_miss(data_set_temporal)
```



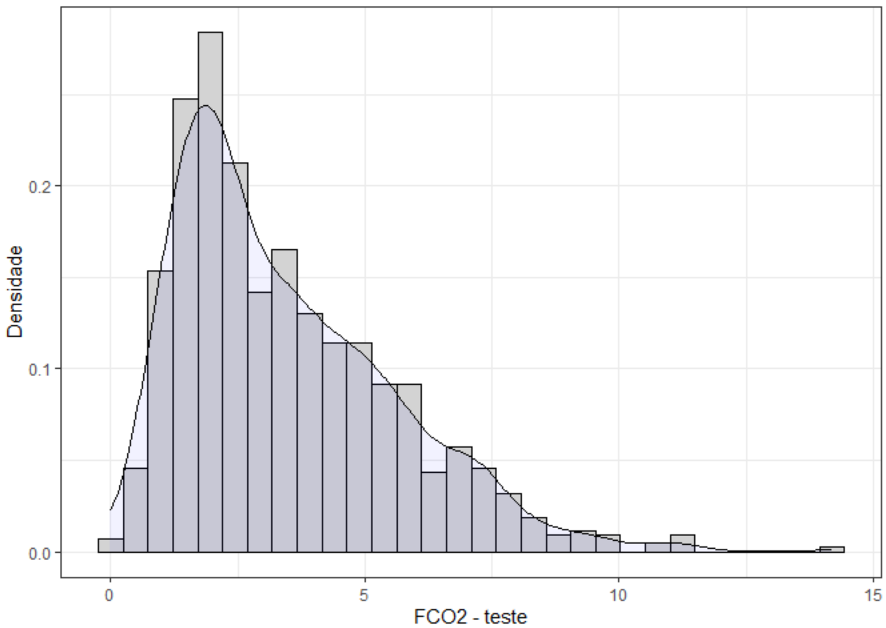

## Definindo a Base de treino e teste

```
# data_set_ml <- data_set_espacial  # <-------
data_set_ml <- data_set_temporal # <-------
fco2_initial_split <- initial_split(data_set_ml, prop = 0.75)
```

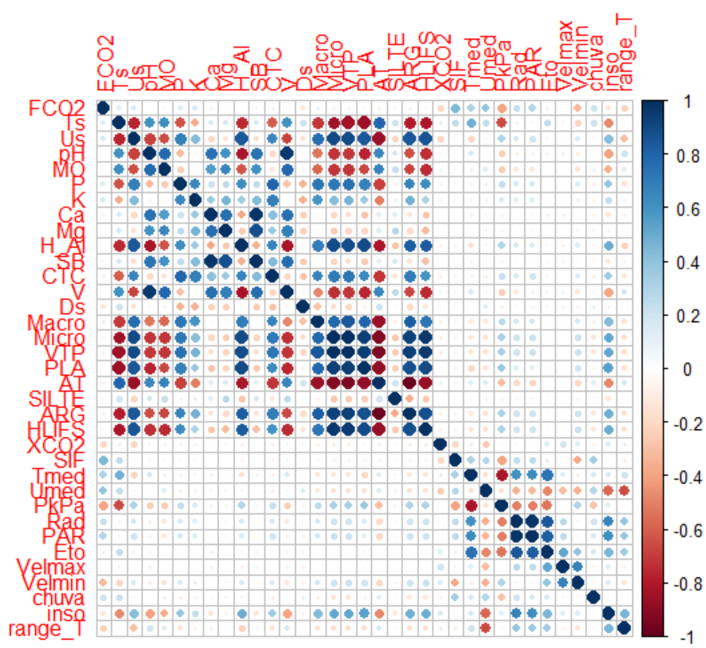
```
fco2_train <- training(fco2_initial_split)
# fco2_test <- testing(fco2_initial_split)
# visdat::vis_miss(fco2_test)
fco2_train  %>%
  ggplot(aes(x=FCO2, y=..density..))+
  geom_histogram(bins = 30, color="black",  fill="lightgray")+
  geom_density(alpha=.05,fill="red")+
  theme_bw() +
  labs(x="FCO2 - treino", y = "Densidade")
```



```
fco2_testing <- testing(fco2_initial_split)
fco2_testing  %>%
  ggplot(aes(x=FCO2, y=..density..))+
  geom_histogram(bins = 30, color="black",  fill="lightgray")+
  geom_density(alpha=.05,fill="blue")+
  theme_bw() +
  labs(x="FCO2 - teste", y = "Densidade")
```




```
fco2_train   %>%    select(FCO2:HLIFS,XCO2,SIF,Tmed:inso) %>%
  mutate(range_T = Tmax-Tmin) %>% select(-c(Tmax,Tmin,Umax,Umin,Dir_vel)) %>%   select(where(is.numeric)) %>%
  drop_na() %>%
  cor()  %>%
  corrplot::corrplot()
```

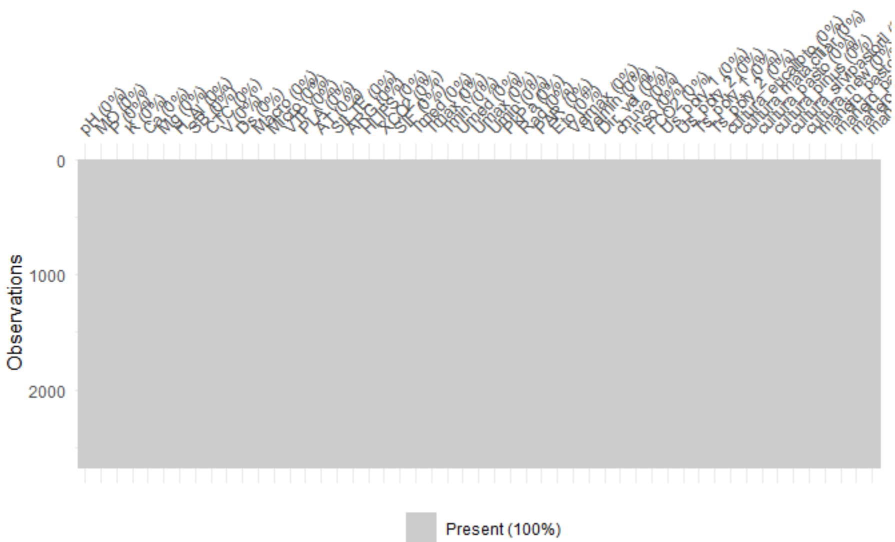


## data prep

```
fco2_recipe <- recipe(FCO2 ~ .,
                      data = fco2_train %>%
                        select(cultura, manejo, cobertura, FCO2:HLIFS,XCO2,SIF,Tmed:inso)
) %>%
  step_normalize(all_numeric_predictors())  %>%
  step_novel(all_nominal_predictors()) %>%
  step_zv(all_predictors()) %>%
  #step_naomit(c(Ts, Us)) %>%
  step_impute_median(where(is.numeric)) %>% # inputação da mediana nos numéricos
  step_poly(c(Us,Ts), degree = 2)  %>%
  step_dummy(all_nominal_predictors())
bake(prep(fco2_recipe), new_data = NULL)
```

```
## # A tibble: 2,676 × 51
##        pH      MO       P       K      Ca      Mg   H_Al     SB    CTC      V
##     <dbl>   <dbl>   <dbl>   <dbl>  <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1  1.26  -0.545  -1.18   -0.819 -0.108 -0.884  -1.68  -0.514 -1.92   1.18
##  2 -0.512 -0.864   1.50    1.55  -0.298  0.415   0.471  0.156  0.573 -0.285
##  3  0.292 -0.0142 -1.27   -0.723 -0.393 -0.327  -0.932 -0.501 -1.45   0.336
##  4  3.35   0.836  -0.0107 -0.385  3.13   2.27   -1.68   3.00  -0.106  2.54
##  5 -0.833  1.69   -0.848   0.195 -0.679  0.0438  0.471 -0.440  0.259 -0.668
##  6  2.06   1.37   -0.346  -0.288  2.36   1.71   -1.68   2.27  -0.313  2.16
##  7 -0.512 -0.864   1.50    1.55  -0.298  0.415   0.471  0.156  0.573 -0.285
##  8  2.06   0.304  -1.10   -0.578  0.558  1.53   -1.32   0.873 -1.19   2.01
##  9  0.935  0.517  -1.02   -0.771 -0.298  0.415  -1.08  -0.169 -1.46   0.909
## 10  0.453  1.05   -0.764  -0.192  1.03   2.46   -0.424  1.60   0.266  0.957
## # ℹ 2,666 more rows
## # ℹ 41 more variables: Ds <dbl>, Macro <dbl>, Micro <dbl>, VTP <dbl>,
## #   PLA <dbl>, AT <dbl>, SILTE <dbl>, ARG <dbl>, HLIFS <dbl>, XCO2 <dbl>,
## #   SIF <dbl>, Tmed <dbl>, Tmax <dbl>, Tmin <dbl>, Umed <dbl>, Umax <dbl>,
## #   Umin <dbl>, PkPa <dbl>, Rad <dbl>, PAR <dbl>, Eto <dbl>, Velmax <dbl>,
## #   Velmin <dbl>, Dir_vel <dbl>, chuva <dbl>, inso <dbl>, FCO2 <dbl>,
## #   Us_poly_1 <dbl>, Us_poly_2 <dbl>, Ts_poly_1 <dbl>, Ts_poly_2 <dbl>, …
```

```
visdat::vis_miss(bake(prep(fco2_recipe), new_data = NULL))
```




## Reamostragem definida e será padrão para todos os modelos

```
fco2_resamples <- vfold_cv(fco2_train, v = 10)
```

# Árvore de Decisão

## Definição do modelo

```
fco2_dt_model <- decision_tree(
  cost_complexity = tune(),
  tree_depth = tune(),
  min_n = tune()
)  %>%
  set_mode("regression")  %>%
  set_engine("rpart")
```

## Workflow

```
fco2_dt_wf <- workflow()   %>%
  add_model(fco2_dt_model) %>%
  add_recipe(fco2_recipe)
```

## Criando a matriz (grid) com os valores de hiperparâmetros a serem testados

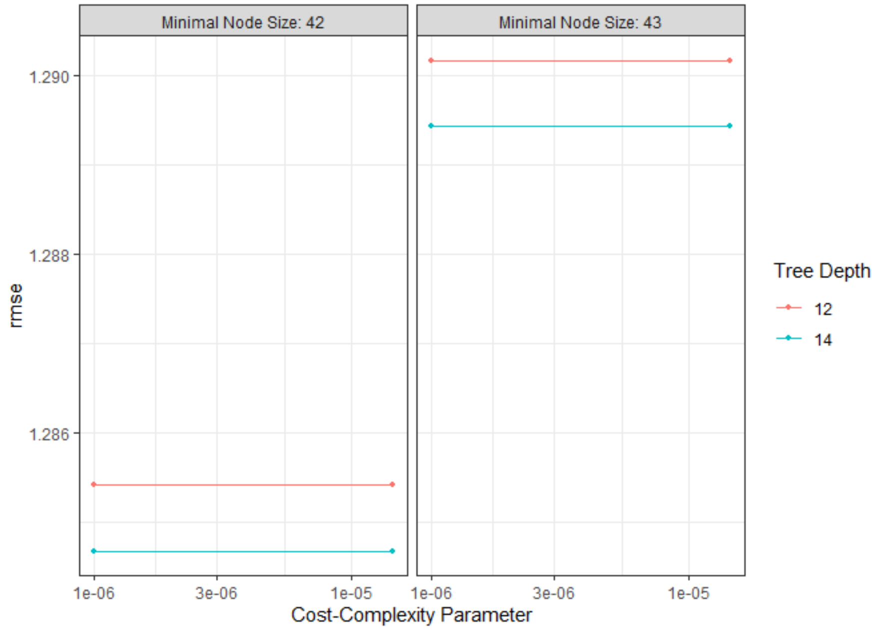
```
# grid_dt <- grid_regular(
#   cost_complexity(c(-6, -4)),
#   tree_depth(range = c(8, 18)),
#   min_n(range = c(42, 52)),
#   levels = 20 # <-----------------------------
# )

## melhor hiperparâmetros
grid_dt <- expand.grid(
  cost_complexity = c(1.438450e-05, 1.000000e-06),
  tree_depth = c(12,14),
  min_n  = c(42, 43)
)
glimpse(grid_dt)
```

```
## Rows: 8
## Columns: 3
## $ cost_complexity <dbl> 1.43845e-05, 1.00000e-06, 1.43845e-05, 1.00000e-06, 1.…
## $ tree_depth      <dbl> 12, 12, 14, 14, 12, 12, 14, 14
## $ min_n           <dbl> 42, 42, 42, 42, 43, 43, 43, 43
```

## Tuning de hiperparâmetros

```
fco2_dt_tune_grid <- tune_grid(
  fco2_dt_wf,
  resamples = fco2_resamples,
  grid = grid_dt,
  metrics = metric_set(rmse)
)
```

```
autoplot(fco2_dt_tune_grid)
```




```
collect_metrics(fco2_dt_tune_grid)
```

```
## # A tibble: 8 × 9
##   cost_complexity tree_depth min_n .metric .estimator  mean     n std_err
##             <dbl>      <dbl> <dbl> <chr>   <chr>      <dbl> <int>   <dbl>
## 1       0.0000144         12    42 rmse    standard    1.26    10  0.0394
## 2       0.000001          12    42 rmse    standard    1.26    10  0.0394
## 3       0.0000144         14    42 rmse    standard    1.26    10  0.0384
## 4       0.000001          14    42 rmse    standard    1.26    10  0.0384
## 5       0.0000144         12    43 rmse    standard    1.26    10  0.0398
## 6       0.000001          12    43 rmse    standard    1.26    10  0.0398
## 7       0.0000144         14    43 rmse    standard    1.26    10  0.0388
## 8       0.000001          14    43 rmse    standard    1.26    10  0.0388
## # ℹ 1 more variable: .config <chr>
```

```
fco2_dt_tune_grid %>%   show_best(metric = "rmse", n = 6)
```

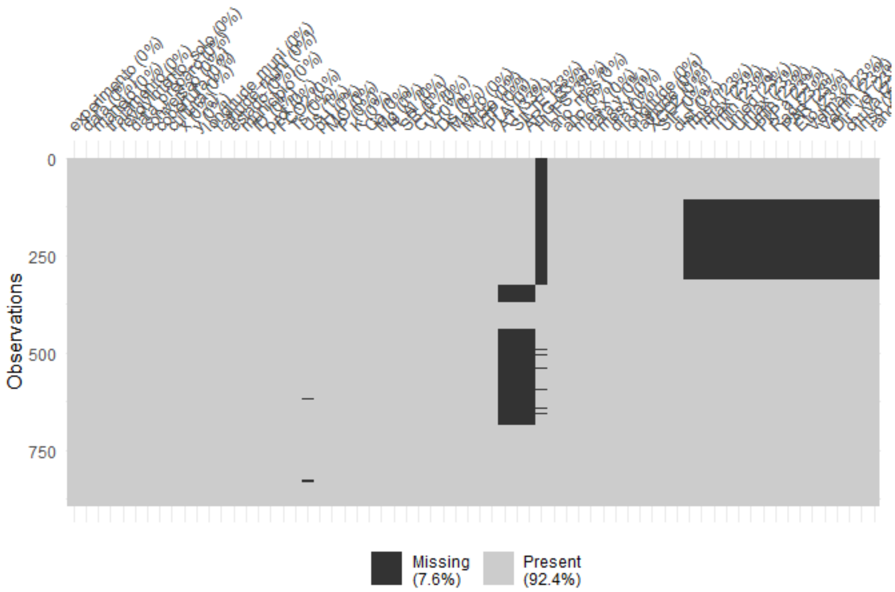
```
## # A tibble: 6 × 9
##   cost_complexity tree_depth min_n .metric .estimator  mean     n std_err
##             <dbl>      <dbl> <dbl> <chr>   <chr>      <dbl> <int>   <dbl>
## 1       0.0000144         14    42 rmse    standard    1.26    10  0.0384
## 2       0.000001          14    42 rmse    standard    1.26    10  0.0384
## 3       0.0000144         12    42 rmse    standard    1.26    10  0.0394
## 4       0.000001          12    42 rmse    standard    1.26    10  0.0394
## 5       0.0000144         14    43 rmse    standard    1.26    10  0.0388
## 6       0.000001          14    43 rmse    standard    1.26    10  0.0388
## # ℹ 1 more variable: .config <chr>
```

## Desempenho dos modelos finais

```
fco2_dt_best_params <- select_best(fco2_dt_tune_grid, "rmse")
fco2_dt_wf <- fco2_dt_wf %>% finalize_workflow(fco2_dt_best_params)
fco2_dt_last_fit <- last_fit(fco2_dt_wf, fco2_initial_split)
```

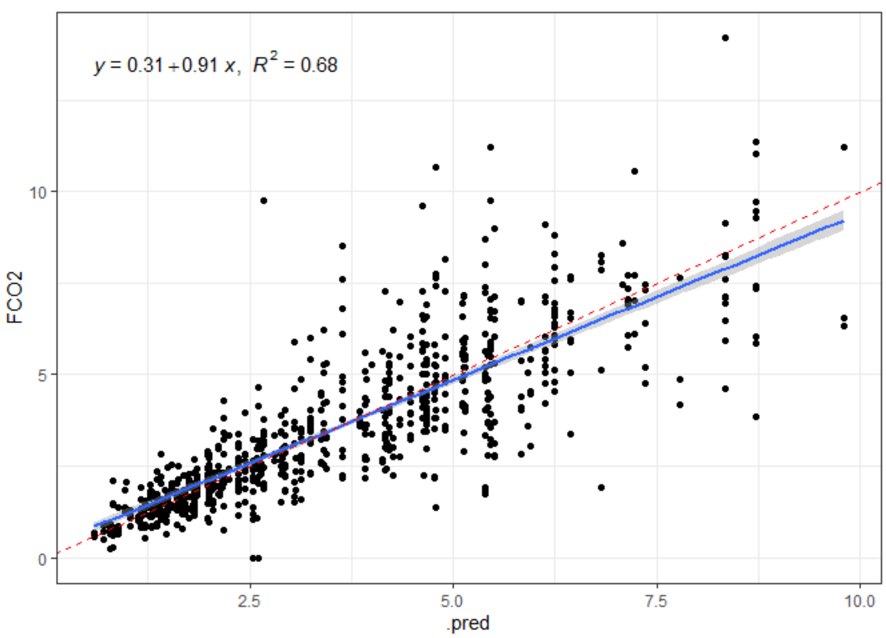
## Criando os preditos

```
fco2_test_preds <- bind_rows(
  collect_predictions(fco2_dt_last_fit)  %>%   mutate(modelo = "dt")
)

fco2_test <- testing(fco2_initial_split)
visdat::vis_miss(fco2_test)
```



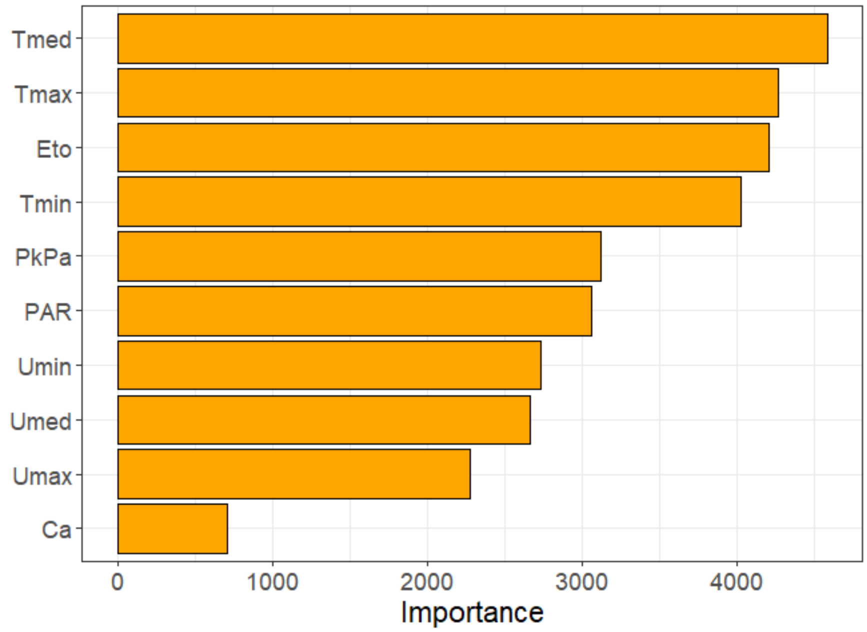

```
fco2_test_preds %>%
  ggplot(aes(x=.pred, y=FCO2)) +
  geom_point()+
  theme_bw() +
  geom_smooth(method = "lm") +
  stat_regline_equation(ggplot2::aes(
  label =  paste(..eq.label.., ..rr.label.., sep = "*plain(\",\")~~"))) +
  geom_abline (slope=1, linetype = "dashed", color="Red")
```

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




# Importância

```
fco2_dt_last_fit_model <-fco2_dt_last_fit$.workflow[[1]]$fit$fit
vip(fco2_dt_last_fit_model,
    aesthetics = list(color = "black", fill = "orange")) +
    theme(axis.text.y=element_text(size=rel(1.5)),
          axis.text.x=element_text(size=rel(1.5)),
          axis.title.x=element_text(size=rel(1.5))
          )
```



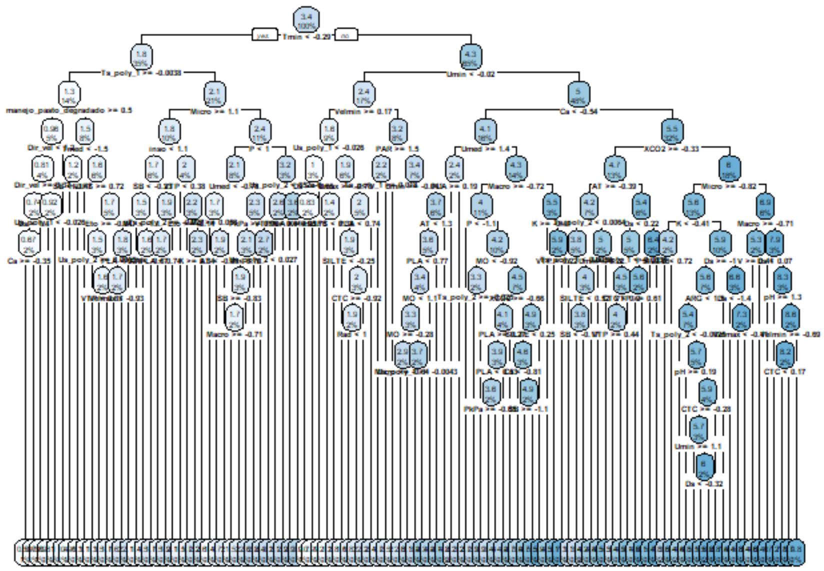

# Métricas

```
da <- fco2_test_preds %>%
  filter(FCO2 > 0, .pred>0 )

my_r <- cor(da$FCO2,da$.pred)
my_r2 <- my_r*my_r
my_mse <- Metrics::mse(da$FCO2,da$.pred)
my_rmse <- Metrics::rmse(da$FCO2,
                         da$.pred)
my_mae <- Metrics::mae(da$FCO2,da$.pred)
my_mape <- Metrics::mape(da$FCO2,da$.pred)*100

vector_of_metrics <- c(r=my_r, R2=my_r2, MSE=my_mse, RMSE=my_rmse, MAE=my_mae, MAPE=my_mape)
print(data.frame(vector_of_metrics))
```

```
##      vector_of_metrics
## r            0.8390179
## R2           0.7039510
## MSE          1.4897133
## RMSE         1.2205381
## MAE          0.8059660
## MAPE        25.8478576
```

```
tree_fit_rpart <- extract_fit_engine(fco2_dt_last_fit)
rpart.plot::rpart.plot(tree_fit_rpart,cex=.4)
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binary for the
variables).
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.

## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



# Random Forest

## Definição do modelo

```
fco2_rf_model <- rand_forest(
  min_n = tune(),
  mtry = tune(),
  trees = tune()
)   %>%
  set_mode("regression")  %>%
  set_engine("randomForest")
```

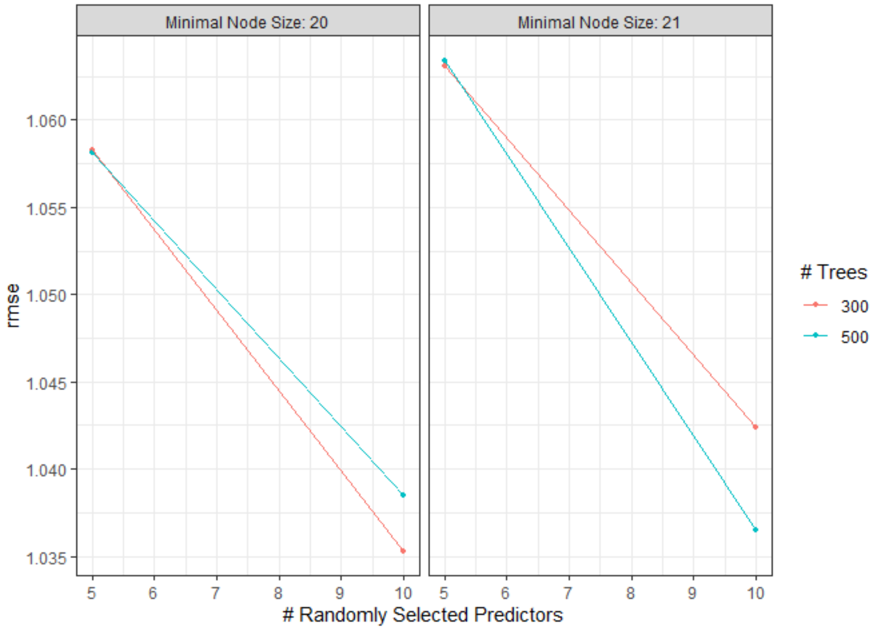
## Workflow

```
fco2_rf_wf <- workflow()   %>%
  add_model(fco2_rf_model) %>%
  add_recipe(fco2_recipe)
```

## Tune

```
# grid_rf <- grid_regular(
#   min_n(range = c(20, 30)),
#   mtry(range = c(5,10)),
#   trees(range = c(100,500) ),
#   levels = 5 #<-----------------------
# )


grid_rf <- expand.grid(
  min_n = c(20,21),
  mtry = c(5,10),
  trees = c(300,500)  #<-----------------------
)
```

```
fco2_rf_tune_grid <- tune_grid(
 fco2_rf_wf,
  resamples = fco2_resamples,
  grid = grid_rf,
  metrics = metric_set(rmse)
)
autoplot(fco2_rf_tune_grid)
```




```
collect_metrics(fco2_rf_tune_grid)
```

```
## # A tibble: 8 × 9
##    mtry trees min_n .metric .estimator  mean     n std_err .config
##   <dbl> <dbl> <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1     5   300    20 rmse    standard    1.06    10  0.0439 Preprocessor1_Model1
## 2     5   300    21 rmse    standard    1.06    10  0.0431 Preprocessor1_Model2
## 3    10   300    20 rmse    standard    1.04    10  0.0446 Preprocessor1_Model3
## 4    10   300    21 rmse    standard    1.04    10  0.0430 Preprocessor1_Model4
## 5     5   500    20 rmse    standard    1.06    10  0.0442 Preprocessor1_Model5
## 6     5   500    21 rmse    standard    1.06    10  0.0439 Preprocessor1_Model6
## 7    10   500    20 rmse    standard    1.04    10  0.0446 Preprocessor1_Model7
## 8    10   500    21 rmse    standard    1.04    10  0.0439 Preprocessor1_Model8
```

```
fco2_rf_tune_grid %>%   show_best(metric = "rmse", n = 6)
```

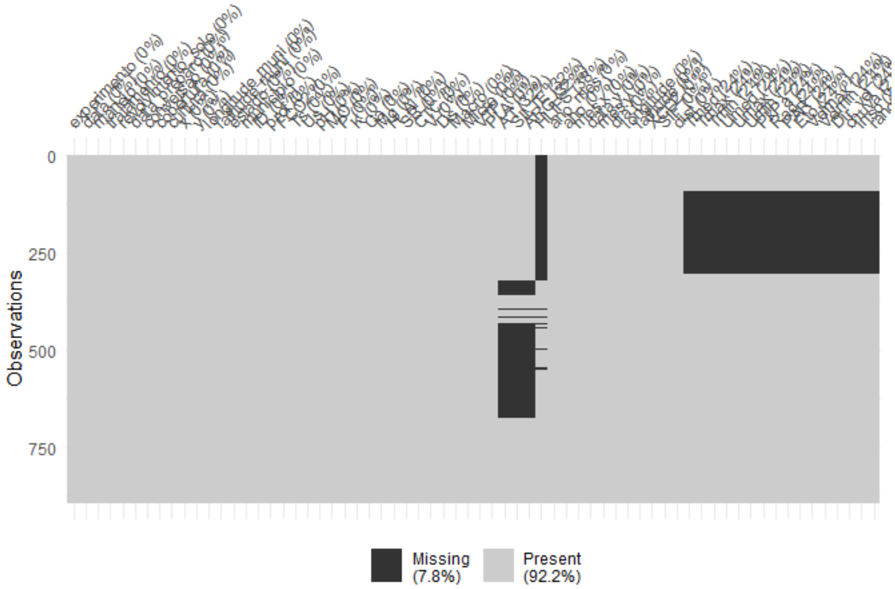
```
## # A tibble: 6 × 9
##    mtry trees min_n .metric .estimator  mean     n std_err .config
##   <dbl> <dbl> <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1    10   300    20 rmse    standard    1.04    10  0.0446 Preprocessor1_Model3
## 2    10   500    21 rmse    standard    1.04    10  0.0439 Preprocessor1_Model8
## 3    10   500    20 rmse    standard    1.04    10  0.0446 Preprocessor1_Model7
## 4    10   300    21 rmse    standard    1.04    10  0.0430 Preprocessor1_Model4
## 5     5   500    20 rmse    standard    1.06    10  0.0442 Preprocessor1_Model5
## 6     5   300    20 rmse    standard    1.06    10  0.0439 Preprocessor1_Model1
```

## Desempenho modelo final

```
fco2_rf_best_params <- select_best(fco2_rf_tune_grid, "rmse")
fco2_rf_wf <- fco2_rf_wf %>% finalize_workflow(fco2_rf_best_params)
fco2_rf_last_fit <- last_fit(fco2_rf_wf, fco2_initial_split)
```

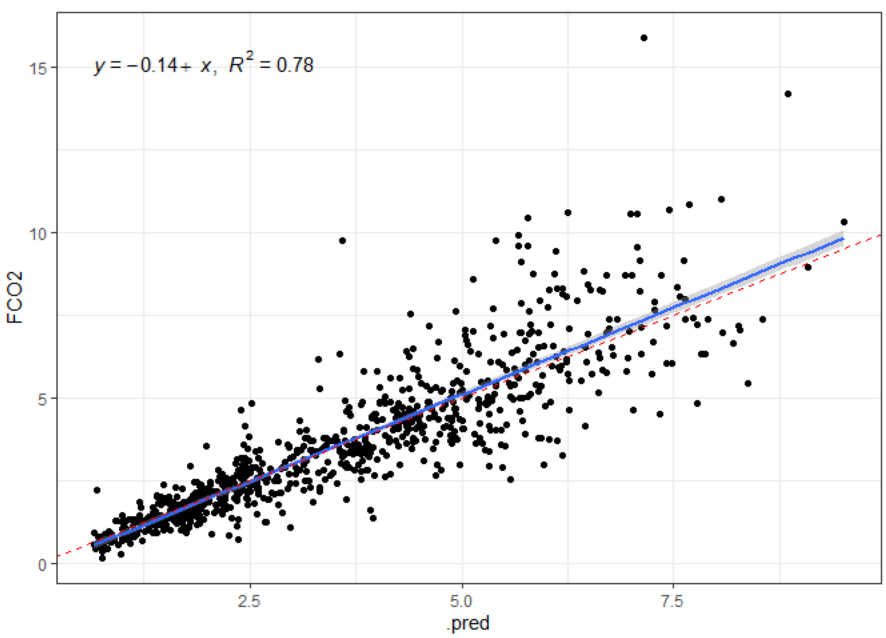
## Criando os preditos

```
fco2_test_preds <- bind_rows(
  collect_predictions(fco2_rf_last_fit)  %>%   mutate(modelo = "rf")
)

fco2_test <- testing(fco2_initial_split)
visdat::vis_miss(fco2_test)
```



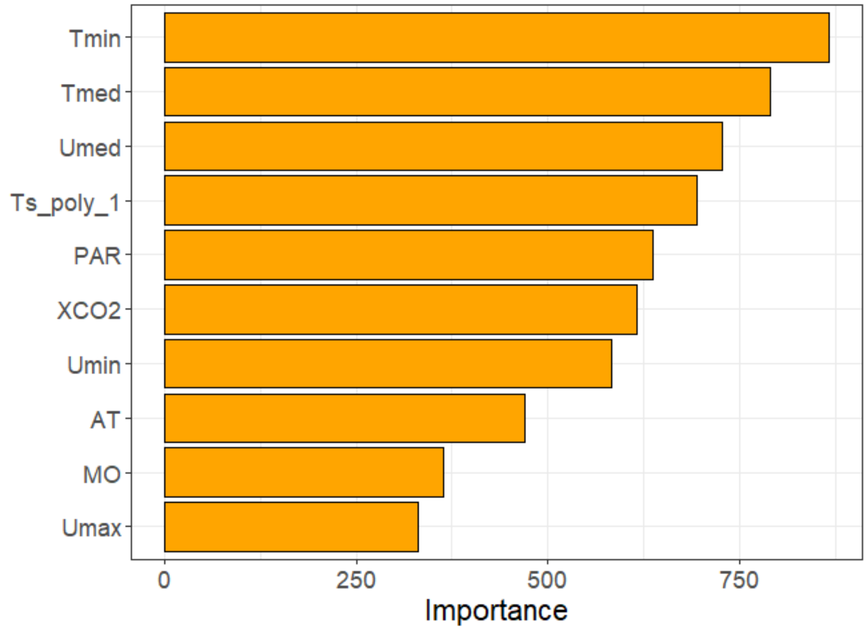

```
fco2_test_preds %>%
  ggplot(aes(x=.pred, y=FCO2)) +
  geom_point()+
  theme_bw() +
  geom_smooth(method = "lm") +
  stat_regline_equation(ggplot2::aes(
  label =  paste(..eq.label.., ..rr.label.., sep = "*plain(\",\")~~"))) +
  geom_abline (slope=1, linetype = "dashed", color="Red")
```

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




# Importância

```
fco2_rf_last_fit_model <-fco2_rf_last_fit$.workflow[[1]]$fit$fit
vip(fco2_rf_last_fit_model,
    aesthetics = list(color = "black", fill = "orange")) +
    theme(axis.text.y=element_text(size=rel(1.5)),
          axis.text.x=element_text(size=rel(1.5)),
          axis.title.x=element_text(size=rel(1.5))
          )
```




## Métricas

```
da <- fco2_test_preds %>%
  filter(FCO2 > 0, .pred>0 )

my_r <- cor(da$FCO2,da$.pred)
my_r2 <- my_r*my_r
my_mse <- Metrics::mse(da$FCO2,da$.pred)
my_rmse <- Metrics::rmse(da$FCO2,
                         da$.pred)
my_mae <- Metrics::mae(da$FCO2,da$.pred)
my_mape <- Metrics::mape(da$FCO2,da$.pred)*100


vector_of_metrics <- c(r=my_r, R2=my_r2, MSE=my_mse, RMSE=my_rmse, MAE=my_mae, MAPE=my_mape)
print(data.frame(vector_of_metrics))
```

```
##      vector_of_metrics
## r            0.8857952
## R2           0.7846332
## MSE          1.0853958
## RMSE         1.0418233
## MAE          0.6539786
## MAPE        20.4465015
```

# Boosting gradient tree (xgb)

```
cores = 4
fco2_xgb_model <- boost_tree(
  mtry = 0.8,
  trees = tune(), # <---------------
  min_n = 5,
  tree_depth = 4,
  loss_reduction = 0, # lambda
  learn_rate = tune(), # epsilon
  sample_size = 0.8
)  %>%
  set_mode("regression")  %>%
  set_engine("xgboost", nthread = cores, counts = FALSE)
```

```
fco2_xgb_wf <- workflow()  %>%
  add_model(fco2_xgb_model) %>%
  add_recipe(fco2_recipe)
```

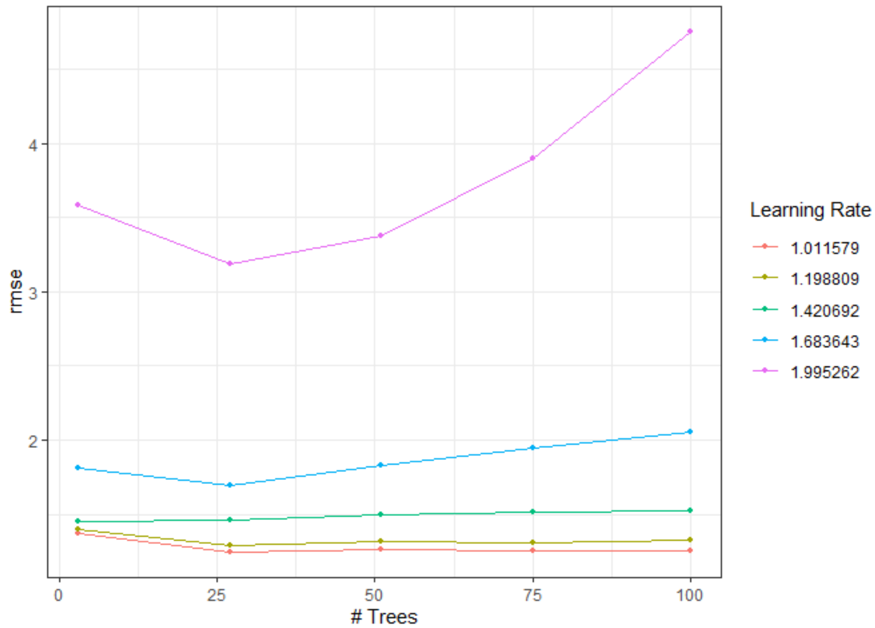
```
grid_xgb <- grid_regular(
  learn_rate(range =  c(0.005, 0.3)),
  trees(range = c(3, 100)),
  levels = 5
)
```

Passo 1

```
fco2_xgb_tune_grid <- tune_grid(
 fco2_xgb_wf,
  resamples = fco2_resamples,
  grid = grid_xgb,
  metrics = metric_set(rmse)
)
```

```
## Warning: package 'xgboost' was built under R version 4.3.1
```

```
autoplot(fco2_xgb_tune_grid)
```




```
fco2_xgb_tune_grid   %>%   show_best(metric = "rmse", n = 6)
```

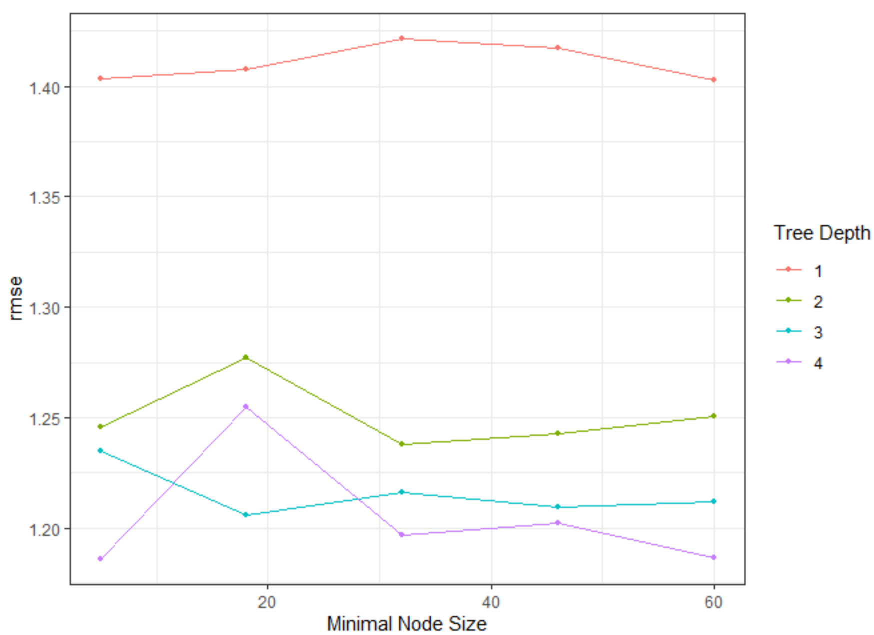
```
## # A tibble: 6 × 8
##   trees learn_rate .metric .estimator  mean     n std_err .config
##   <int>      <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1    27       1.01 rmse    standard    1.25    10  0.0496 Preprocessor1_Model02
## 2    75       1.01 rmse    standard    1.25    10  0.0558 Preprocessor1_Model04
## 3   100       1.01 rmse    standard    1.25    10  0.0545 Preprocessor1_Model05
## 4    51       1.01 rmse    standard    1.26    10  0.0528 Preprocessor1_Model03
## 5    27       1.20 rmse    standard    1.29    10  0.0353 Preprocessor1_Model07
## 6    75       1.20 rmse    standard    1.31    10  0.0380 Preprocessor1_Model09
```

```
fco2_xgb_select_best_passo1 <- fco2_xgb_tune_grid %>%
  select_best(metric = "rmse")
fco2_xgb_select_best_passo1
```

```
## # A tibble: 1 × 3
##   trees learn_rate .config
##   <int>      <dbl> <chr>
## 1    27       1.01 Preprocessor1_Model02
```

Passo 2

```r
fco2_xgb_model <- boost_tree(
  mtry = 0.8,
  trees = fco2_xgb_select_best_passo1$trees,
  min_n = tune(),
  tree_depth = tune(),
  loss_reduction = 0,
  learn_rate = fco2_xgb_select_best_passo1$learn_rate,
  sample_size = 0.8
) %>%
  set_mode("regression")  %>%
  set_engine("xgboost", nthread = cores, counts = FALSE)

#### Workflow
fco2_xgb_wf <- workflow() %>%
    add_model(fco2_xgb_model)   %>%
    add_recipe(fco2_recipe)

#### Grid
fco2_xgb_grid <- grid_regular(
  tree_depth(range = c(1, 4)),
  min_n(range = c(5, 60)),
  levels = 5
)

fco2_xgb_tune_grid <- fco2_xgb_wf   %>%
  tune_grid(
    resamples =fco2_resamples,
    grid = fco2_xgb_grid,
    control = control_grid(save_pred = TRUE, verbose = FALSE, allow_par = TRUE),
    metrics = metric_set(rmse)
  )

#### Melhores hiperparâmetros
autoplot(fco2_xgb_tune_grid)
```




```r
fco2_xgb_tune_grid  %>%   show_best(metric = "rmse", n = 5)
```

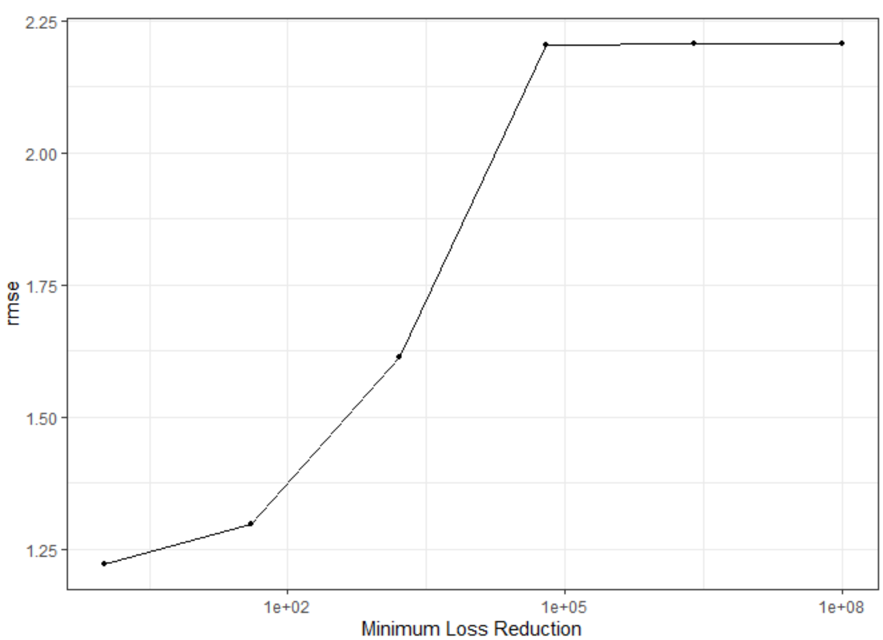
```
## # A tibble: 5 × 8
##   min_n tree_depth .metric .estimator  mean     n std_err .config
##   <int>      <int> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1     5          4 rmse    standard    1.19    10  0.0473 Preprocessor1_Model04
## 2    60          4 rmse    standard    1.19    10  0.0407 Preprocessor1_Model20
## 3    32          4 rmse    standard    1.20    10  0.0425 Preprocessor1_Model12
## 4    46          4 rmse    standard    1.20    10  0.0409 Preprocessor1_Model16
## 5    18          3 rmse    standard    1.21    10  0.0377 Preprocessor1_Model07
```

```r
fco2_xgb_select_best_passo2 <- fco2_xgb_tune_grid  %>%   select_best(metric = "rmse")
fco2_xgb_select_best_passo2
```

```
## # A tibble: 1 × 3
##   min_n tree_depth .config
##   <int>      <int> <chr>
## 1     5          4 Preprocessor1_Model04
```

# Passo 3

```
fco2_xgb_model <- boost_tree(
  mtry = 0.8,
  trees = fco2_xgb_select_best_passo1$trees,
  min_n = fco2_xgb_select_best_passo2$min_n,
  tree_depth = fco2_xgb_select_best_passo2$tree_depth,
  loss_reduction =tune(),
  learn_rate = fco2_xgb_select_best_passo1$learn_rate,
  sample_size = 0.8
)  %>%
  set_mode("regression")  %>%
  set_engine("xgboost", nthread = cores, counts = FALSE)

#### Workflow
fco2_xgb_wf <- workflow()  %>%
    add_model(fco2_xgb_model)  %>%
    add_recipe(fco2_recipe)

#### Grid
fco2_xgb_grid <- grid_regular(
  loss_reduction(range = c(0.01, 8)),
  levels = 6
)

fco2_xgb_tune_grid <- fco2_xgb_wf   %>%
  tune_grid(
    resamples = fco2_resamples,
    grid = fco2_xgb_grid,
    control = control_grid(save_pred = TRUE,
                           verbose = FALSE,
                           allow_par = TRUE),
    metrics = metric_set(rmse)
  )

#### Melhores hiperparâmetros
autoplot(fco2_xgb_tune_grid)
```




```
fco2_xgb_tune_grid   %>%   show_best(metric = "rmse", n = 5)
```

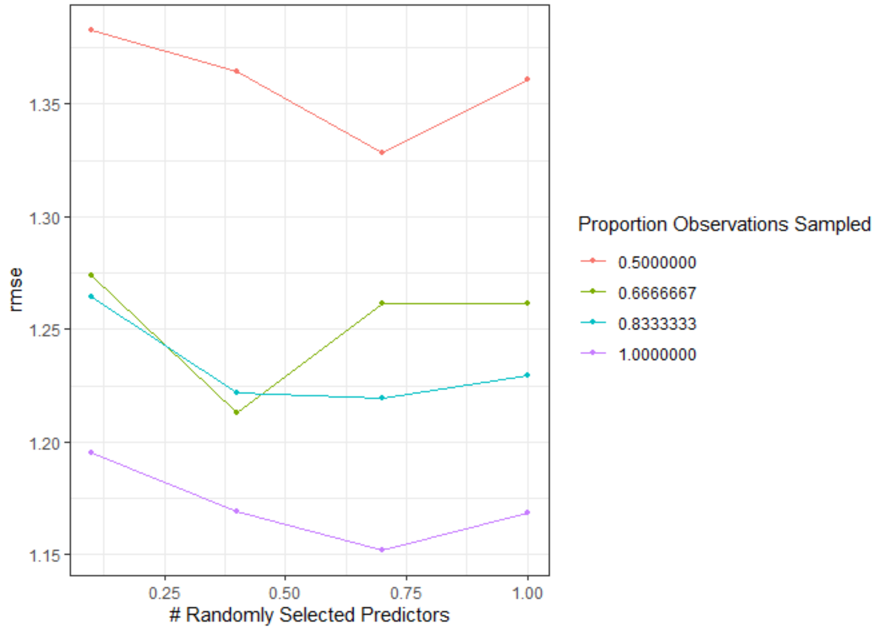
```
## # A tibble: 5 × 7
##   loss_reduction .metric .estimator  mean     n std_err .config
##            <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1           1.02 rmse    standard    1.22    10  0.0461 Preprocessor1_Model1
## 2          40.6  rmse    standard    1.30    10  0.0435 Preprocessor1_Model2
## 3        1607.   rmse    standard    1.61    10  0.0329 Preprocessor1_Model3
## 4       63680.   rmse    standard    2.20    10  0.0463 Preprocessor1_Model4
## 5   100000000    rmse    standard    2.21    10  0.0463 Preprocessor1_Model6
```

```
fco2_xgb_select_best_passo3 <- fco2_xgb_tune_grid %>% select_best(metric = "rmse")
fco2_xgb_select_best_passo3
```

```
## # A tibble: 1 × 2
##   loss_reduction .config
##            <dbl> <chr>
## 1           1.02 Preprocessor1_Model1
```

## Passo 4

```
fco2_xgb_model <- boost_tree(
  mtry = tune(),
  trees = fco2_xgb_select_best_passo1$trees,
  min_n = fco2_xgb_select_best_passo2$min_n,
  tree_depth = fco2_xgb_select_best_passo2$tree_depth,
  loss_reduction = fco2_xgb_select_best_passo3$loss_reduction,
  learn_rate = fco2_xgb_select_best_passo1$learn_rate,
  sample_size = tune()
)%>%
  set_mode("regression")  |>
  set_engine("xgboost", nthread = cores, counts = FALSE)

#### Workflow
fco2_xgb_wf <- workflow()  %>%
    add_model(fco2_xgb_model)  %>%
    add_recipe(fco2_recipe)

#### Grid
fco2_xgb_grid <- expand.grid(
    sample_size = seq(0.5, 1.0, length.out = 4), ## <---
    mtry = seq(0.1, 1.0, length.out = 4) ## <---
)

fco2_xgb_tune_grid <- fco2_xgb_wf   %>%
  tune_grid(
    resamples = fco2_resamples,
    grid = fco2_xgb_grid,
    control = control_grid(save_pred = TRUE,
                           verbose = FALSE,
                           allow_par = TRUE),
    metrics = metric_set(rmse)
  )

autoplot(fco2_xgb_tune_grid)
```




```
fco2_xgb_tune_grid  |>  show_best(metric = "rmse", n = 5)
```

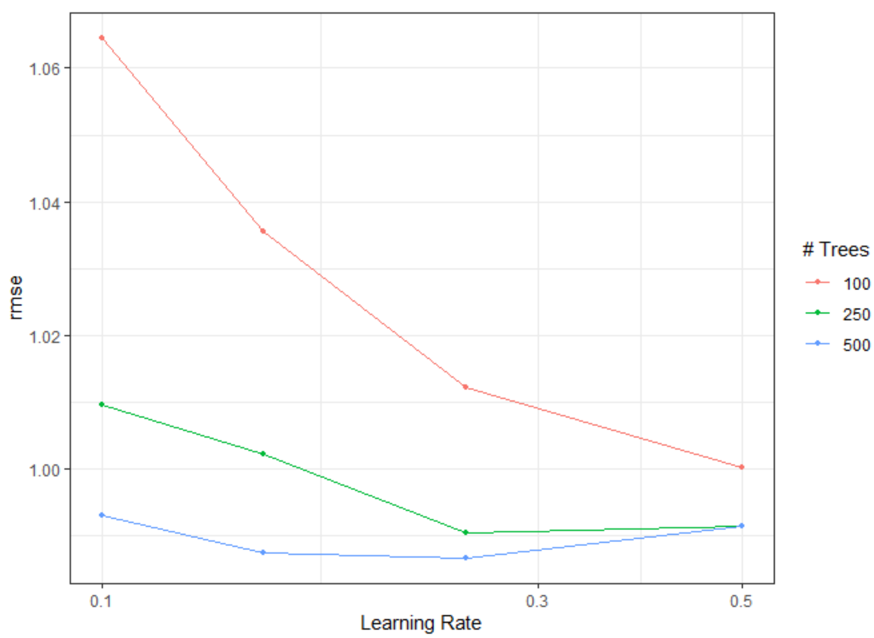
```
## # A tibble: 5 × 8
##    mtry sample_size .metric .estimator  mean     n std_err .config
##   <dbl>       <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1   0.7       1     rmse    standard    1.15    10  0.0472 Preprocessor1_Model12
## 2   1         1     rmse    standard    1.17    10  0.0322 Preprocessor1_Model16
## 3   0.4       1     rmse    standard    1.17    10  0.0523 Preprocessor1_Model08
## 4   0.1       1     rmse    standard    1.20    10  0.0483 Preprocessor1_Model04
## 5   0.4       0.667 rmse    standard    1.21    10  0.0461 Preprocessor1_Model06
```

```
fco2_xgb_select_best_passo4 <- fco2_xgb_tune_grid   %>%   select_best(metric = "rmse")
fco2_xgb_select_best_passo4
```

```
## # A tibble: 1 × 3
##    mtry sample_size .config
##   <dbl>       <dbl> <chr>
## 1   0.7           1 Preprocessor1_Model12
```

## Passo 5

```
fco2_xgb_model <- boost_tree(
  mtry = fco2_xgb_select_best_passo4$mtry,
  trees = tune(),
  min_n = fco2_xgb_select_best_passo2$min_n,
  tree_depth = fco2_xgb_select_best_passo2$tree_depth,
  loss_reduction = fco2_xgb_select_best_passo3$loss_reduction,
  learn_rate = tune(),
  sample_size = fco2_xgb_select_best_passo4$sample_size
) |>
  set_mode("regression")  %>%
  set_engine("xgboost", nthread = cores, counts = FALSE)

#### Workflow
fco2_xgb_wf <- workflow() %>%
    add_model(fco2_xgb_model)  %>%
    add_recipe(fco2_recipe)

#### Grid
fco2_xgb_grid <- expand.grid(
    learn_rate = c(0.10, 0.15, 0.25, 0.50),
    trees = c(100, 250, 500)
)

fco2_xgb_tune_grid <- fco2_xgb_wf   %>%
  tune_grid(
    resamples = fco2_resamples,
    grid = fco2_xgb_grid,
    control = control_grid(save_pred = TRUE,
                           verbose = FALSE,
                           allow_par = TRUE),
    metrics = metric_set(rmse)
  )

#### Melhores hiperparâmetros
autoplot(fco2_xgb_tune_grid)
```




```
fco2_xgb_tune_grid  %>%   show_best(metric = "rmse", n = 5)
```

```
## # A tibble: 5 × 8
##   trees learn_rate .metric .estimator  mean     n std_err .config
##   <dbl>      <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1   500       0.25 rmse    standard   0.987    10  0.0499 Preprocessor1_Model09
## 2   500       0.15 rmse    standard   0.987    10  0.0510 Preprocessor1_Model06
## 3   250       0.25 rmse    standard   0.991    10  0.0502 Preprocessor1_Model08
## 4   500       0.5  rmse    standard   0.991    10  0.0416 Preprocessor1_Model12
## 5   250       0.5  rmse    standard   0.991    10  0.0409 Preprocessor1_Model11
```

```
fco2_xgb_select_best_passo5 <- fco2_xgb_tune_grid   %>%   select_best(metric = "rmse")
fco2_xgb_select_best_passo5
```

```
## # A tibble: 1 × 3
##   trees learn_rate .config
##   <dbl>      <dbl> <chr>
## 1   500       0.25 Preprocessor1_Model09
```

# Desempenho dos modelos finais

```
fco2_xgb_model <- boost_tree(
  mtry = fco2_xgb_select_best_passo4$mtry,
  trees = fco2_xgb_select_best_passo5$trees,
  min_n = fco2_xgb_select_best_passo2$min_n,
  tree_depth = fco2_xgb_select_best_passo2$tree_depth,
  loss_reduction = fco2_xgb_select_best_passo3$loss_reduction,
  learn_rate = fco2_xgb_select_best_passo5$learn_rate,
  sample_size = fco2_xgb_select_best_passo4$sample_size
) %>%
  set_mode("regression")  %>%
  set_engine("xgboost", nthread = cores, counts = FALSE)
```

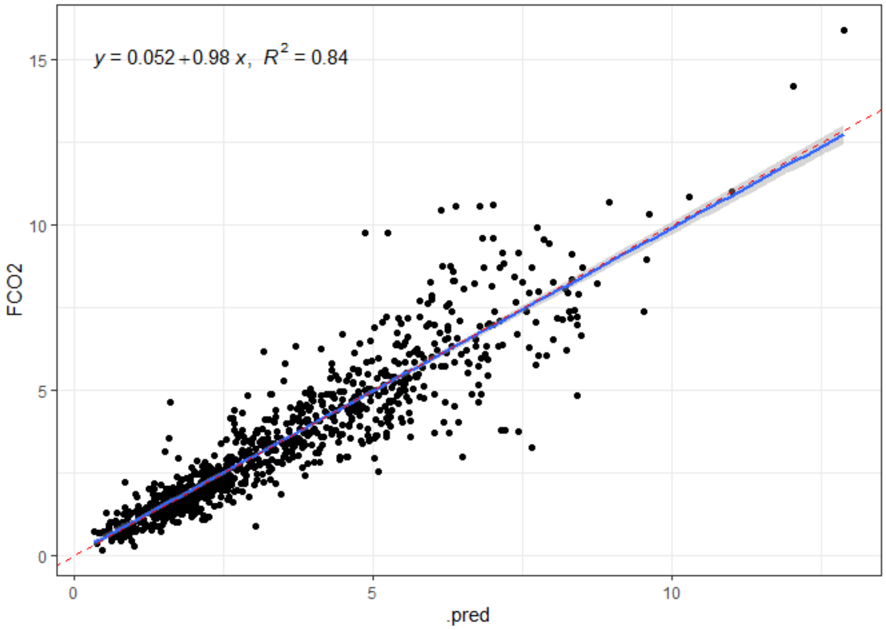
```
df <- data.frame(
  mtry = fco2_xgb_select_best_passo4$mtry,
  trees = fco2_xgb_select_best_passo5$trees,
  min_n = fco2_xgb_select_best_passo2$min_n,
  tree_depth = fco2_xgb_select_best_passo2$tree_depth,
  loss_reduction = fco2_xgb_select_best_passo3$loss_reduction,
  learn_rate = fco2_xgb_select_best_passo5$learn_rate,
  sample_size = fco2_xgb_select_best_passo4$sample_size
)
fco2_xgb_wf <- fco2_xgb_wf %>% finalize_workflow(df) # <------
fco2_xgb_last_fit <- last_fit(fco2_xgb_wf, fco2_initial_split) # <--------
```

# Criar Preditos

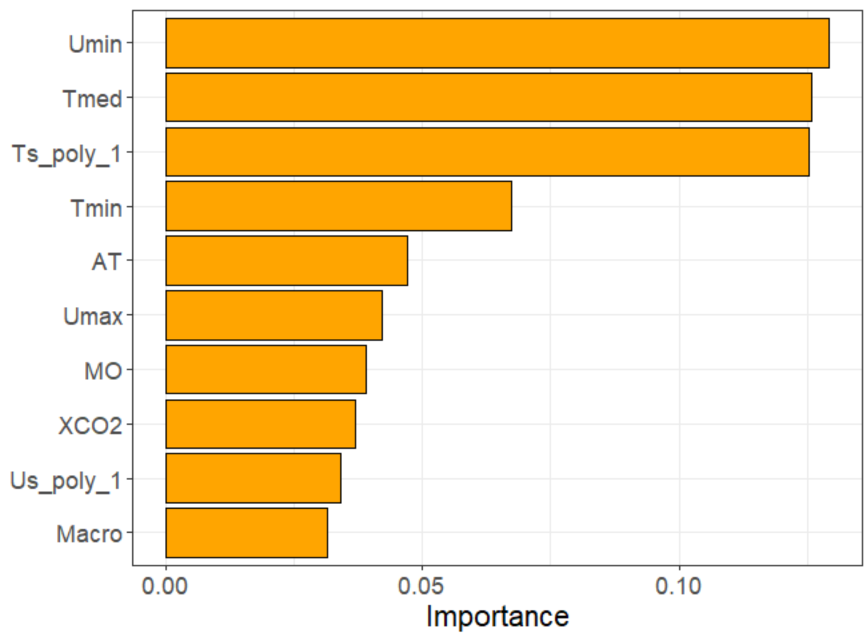
```
fco2_test_preds <- bind_rows(
  collect_predictions(fco2_xgb_last_fit)  %>%   mutate(modelo = "xgb")
)
```

```
fco2_test_preds %>%
  ggplot(aes(x=.pred, y=FCO2)) +
  geom_point()+
  theme_bw() +
  geom_smooth(method = "lm") +
  stat_regline_equation(ggplot2::aes(
  label =  paste(..eq.label.., ..rr.label.., sep = "*plain(\",\")~~")))+
  geom_abline (slope=1, linetype = "dashed", color="Red")
```

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




```
fco2_xgb_last_fit_model <-fco2_xgb_last_fit$.workflow[[1]]$fit$fit
vip(fco2_xgb_last_fit_model,
    aesthetics = list(color = "black", fill = "orange")) +
    theme(axis.text.y=element_text(size=rel(1.5)),
          axis.text.x=element_text(size=rel(1.5)),
          axis.title.x=element_text(size=rel(1.5))
          )
```




# Métricas

```
da <- fco2_test_preds %>%
  filter(FCO2 > 0, .pred>0 )

my_r <- cor(da$FCO2,da$.pred)
my_r2 <- my_r*my_r
my_mse <- Metrics::mse(da$FCO2,da$.pred)
my_rmse <- Metrics::rmse(da$FCO2,
                         da$.pred)
my_mae <- Metrics::mae(da$FCO2,da$.pred)
my_mape <- Metrics::mape(da$FCO2,da$.pred)*100

vector_of_metrics <- c(r=my_r, R2=my_r2, MSE=my_mse, RMSE=my_rmse, MAE=my_mae, MAPE=my_mape)
print(data.frame(vector_of_metrics))
```

```
##      vector_of_metrics
## r            0.9142274
## R2           0.8358117
## MSE          0.8211791
## RMSE         0.9061893
## MAE          0.6019709
## MAPE        19.3042846
```