

# EMIÇÃO DE CO<sub>2</sub> 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) (<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

Este projeto é licenciado sob `MIT License`. Consulte o arquivo LICENSE (<https://github.com/arpanosso/tese-fco2-ml-2023/blob/master/LICENSE.md>) para obter mais detalhes.

## Base de dados

Apresentação do pacote `fco2r` construído para facilitar a divulgação e análise dos resultados obtidos ao longo de mais de 20 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)
library(patchwork)
library(ggspatial)
library(readxl)
library(skimr)
library(tidymodels)
library(ISLR)
library(modeldata)
library(vip)
library(ggpubr)
```

## Conhecendo a base de dados de emissão de CO<sub>2</sub> do solo

Base proveniente de ensaios de campo.

```
help(data_fco2)
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

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
cultura	0	1	4	14	0	11	0
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	

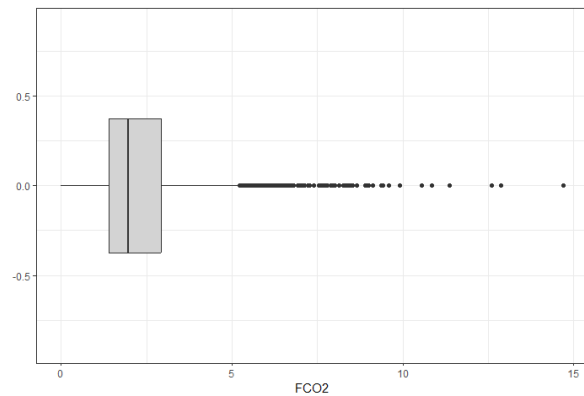
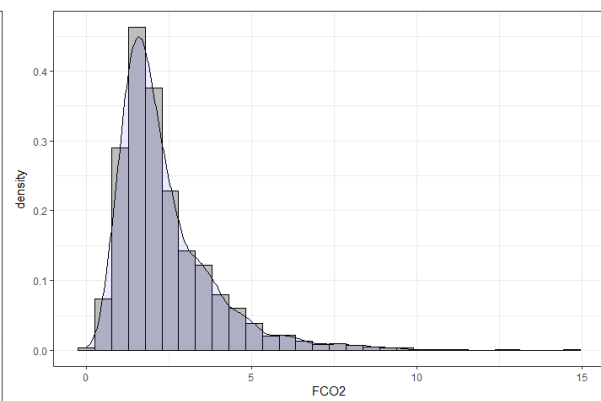
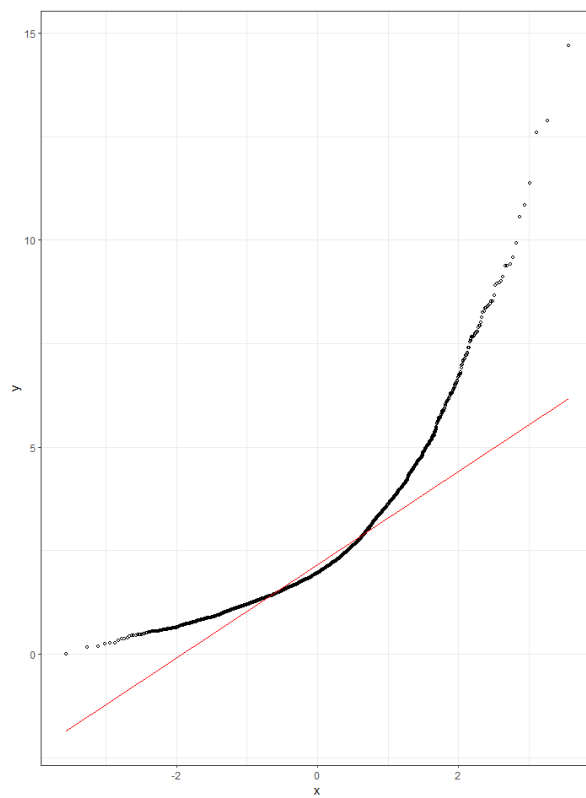
Alguns gráficos a respeito de nossa variável alvo, emissão de CO<sub>2</sub> do solo (FCO<sub>2</sub>).

```
theme_set(theme_bw())
fco2_historama <- data_fco2 %>%
  drop_na() %>%
  ggplot(aes(x=FCO2, y=..density..)) +
  geom_histogram(col="black", fill="gray") +
  geom_density(fill="blue", alpha=.08)

fco2_boxplot <- data_fco2 %>%
  drop_na() %>%
  ggplot(aes(x=FCO2)) +
  geom_boxplot(fill="lightgray") +
  coord_cartesian(ylim=c(-.9,.9))

fco2_qqplot <- data_fco2 %>%
  drop_na() %>%
  ggplot(aes(sample=FCO2)) +
  stat_qq(shape=1, size=1, color="black")+
  stat_qq_line(col="red")

fco2_qqplot | (fco2_historama)/(fco2_boxplot)
```



## Aplicando a transformação logarítmica nos dados de FCO<sub>2</sub>

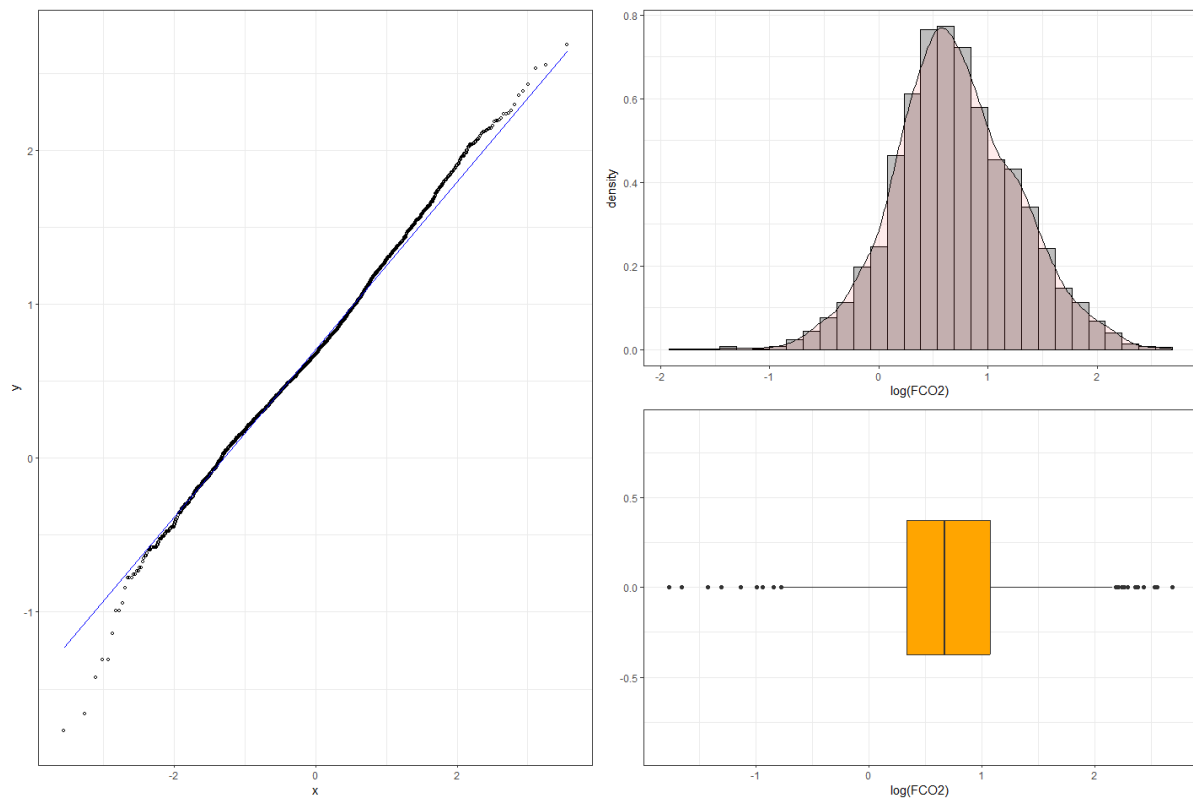
```
ggplot2::theme_set(theme_bw())

fco2_histograma <- data_fco2 %>%
  drop_na() %>%
  ggplot(aes(x=log(FCO2), y=.density..)) +
  geom_histogram(col="black",fill="gray") +
  geom_density(fill="red",alpha=.08)

fco2_boxplot <- data_fco2 %>%
  drop_na() %>%
  ggplot(aes(x=log(FCO2))) +
  geom_boxplot(fill="orange") +
  coord_cartesian(ylim=c(-.9,.9))

fco2_qqplot <- data_fco2 %>%
  drop_na() %>%
  ggplot(aes(sample=log(FCO2))) +
  stat_qq(shape=1,size=1,color="black")+
  stat_qq_line(col="blue")

fco2_qqplot | (fco2_histograma)/(fco2_boxplot)
```



```
# 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")
```

```

# muni <- read_municipality()
# write_rds(muni,"data/municipios.rds")
muni <- read_rds("data/municipios.rds")

sp <- muni %>%
  filter(abbrev_state == "SP")

ms <- muni %>%
  filter(abbrev_state == "MS")

sp_ms <- muni %>%
  filter(abbrev_state == "SP" | abbrev_state == "MS")

fsp<-if_else(sp$name_muni == "Jaboticabal" |
  sp$name_muni == "Guariba" |
  sp$name_muni == "Pardópolis" |
  sp$name_muni == "Rincão"|
  sp$name_muni == "Mococa"|
  sp$name_muni == "Ilha Solteira"
  ,"red","lightyellow")

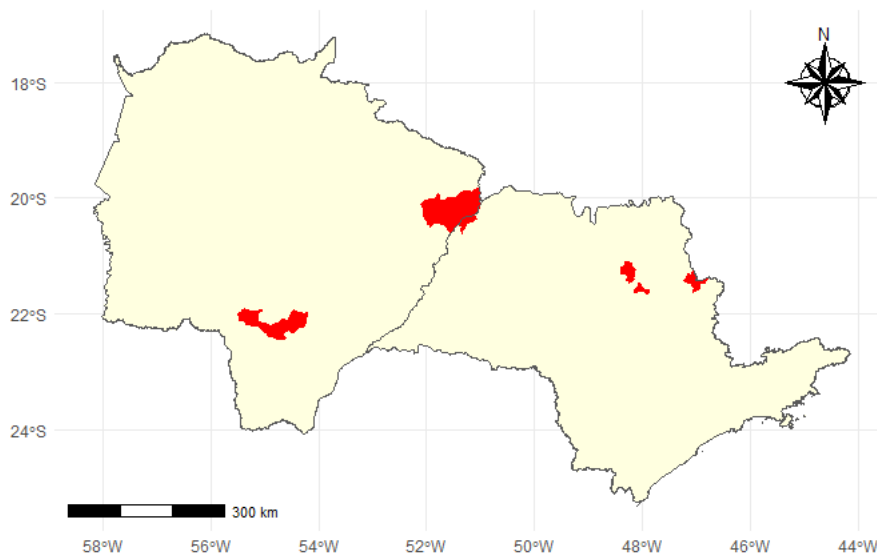
fms<-if_else(ms$name_muni == "Aparecida Do Taboado" |
  ms$name_muni == "Selvíria"|
  ms$name_muni == "Dourados"
  ,"red","lightyellow")

sp_ <- estados %>%
  filter(abbrev_state == "SP")

ms_ <- estados %>%
  filter(abbrev_state == "MS")

ggplot(sp_ms) +
  geom_sf(fill="lightyellow")+
  theme_minimal() +
  annotation_scale(location="bl")+
  annotation_north_arrow(location="tr",
    style = north_arrow_nautical(),
    width = unit(2,"cm"),
    height = unit(2,"cm")) +
  geom_sf(data= sp, fill=fsp,col=fsp) +
  geom_sf(data=sp_,fill="transparent") +
  geom_sf(data= ms, fill=fms,col=fms) +
  geom_sf(data=ms_,fill="transparent")

```



Conhecendo a base de dados de concentração de CO<sub>2</sub> atmosférico, oriundo do

sensor NASA-OCO2.

```
help(oco2_br)
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<sub>CO2</sub>

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
latitude	0	1	-1.179000e+01	7.850000e+00	-3.250000e+01
time_yyyymmddhhmmss	0	1	2.016952e+13	1.564571e+10	2.014091e+13
altitude_km	0	1	3.123200e+03	1.108800e+02	2.555700e+03
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
fluorescence_radiance_757nm_idp_ph_sec_1_m_2_sr_1_um_1	0	1	-1.358150e+18	1.946775e+20	-3.400736e+22
xco2_moles_mole_1	0	1	0.000000e+00	0.000000e+00	0.000000e+00
aerosol_total_aod	0	1	4.828100e+02	7.848572e+04	2.000000e-02
fluorescence_offset_relative_771nm_idp	0	1	-4.814400e+02	2.193698e+04	-9.999990e+05

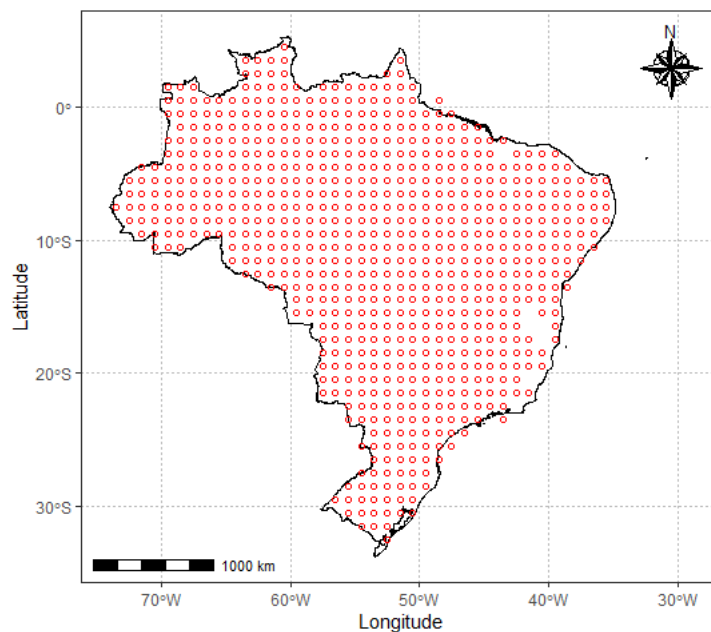
skim_variable	n_missing	complete_rate	mean	sd	p0	
fluorescence_at_reference_ph_sec_1_m_2_sr_1_um_1	0	1	1.296932e+18	2.245185e+18	-8.394901e+19	2.014561
fluorescence_radiance_771nm_idp_ph_sec_1_m_2_sr_1_um_1	0	1	1.904438e+18	2.236381e+18	-8.453983e+19	9.694701
fluorescence_offset_relative_757nm_idp	0	1	-3.744400e+02	1.934763e+04	-9.999990e+05	1.000000
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	4.695461
XCO2	0	1	3.858900e+02	3.120000e+00	3.383400e+02	3.844101

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

```
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)
```

Mapa das leituras do satélite OCO2-NASA

```
source("R/graficos.R")
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=0.1)+
  labs(x="Longitude",y="Latitude")
```



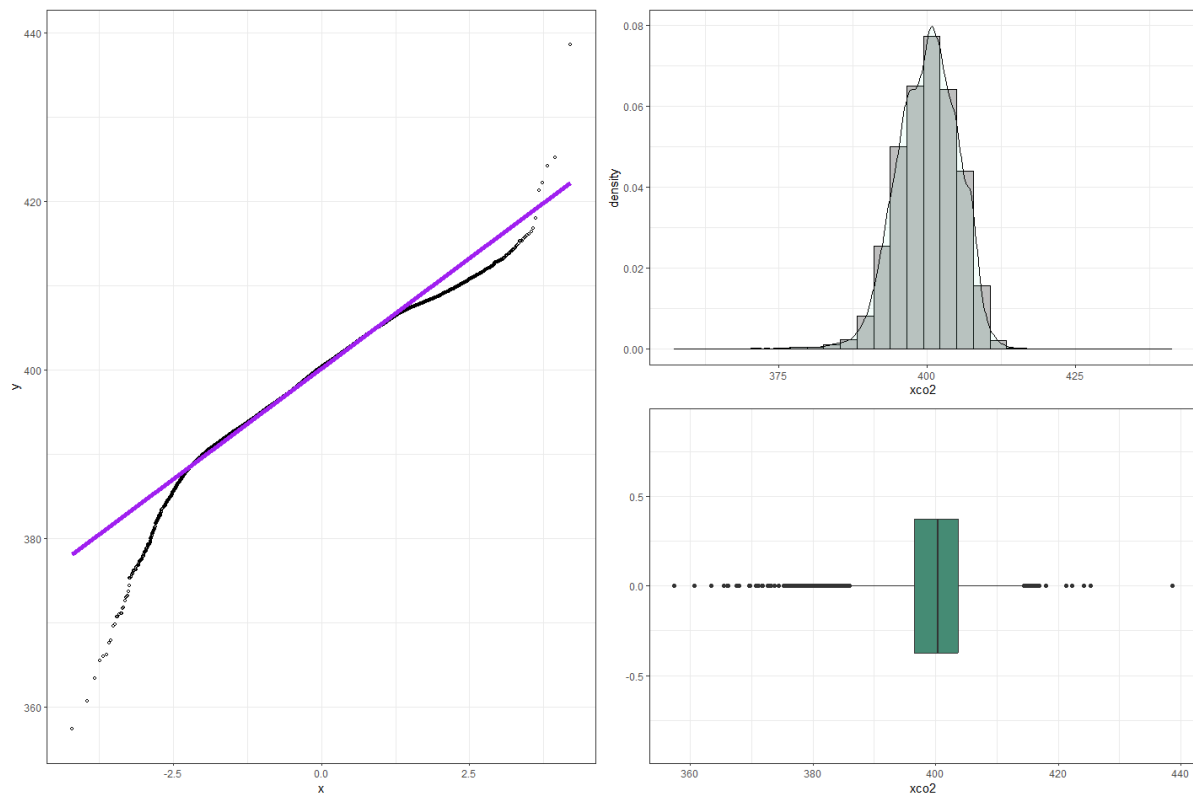


```
xco2_historama <- oco2_br %>%
  drop_na() %>%
  ggplot(aes(x=xco2, y=..density..)) +
  geom_histogram(col="black",fill="gray") +
  geom_density(fill="aquamarine2",alpha=.08)

xco2_boxplot <- oco2_br %>%
  drop_na() %>%
  ggplot(aes(x=xco2)) +
  geom_boxplot(fill="aquamarine4") +
  coord_cartesian(ylim=c(-.9,.9))

xco2_qqplot <- oco2_br %>%
  drop_na() %>%
  ggplot(aes(sample=xco2)) +
  stat_qq(shape=1,size=1,color="black")+
  stat_qq_line(col="purple",lwd=2)

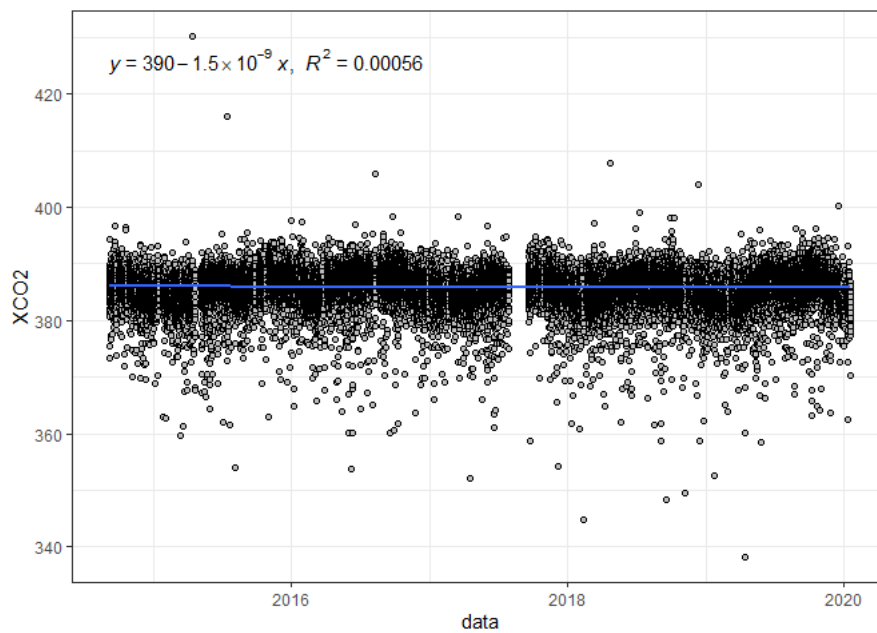
xco2_qqplot | (xco2_historama)/(xco2_boxplot)
```



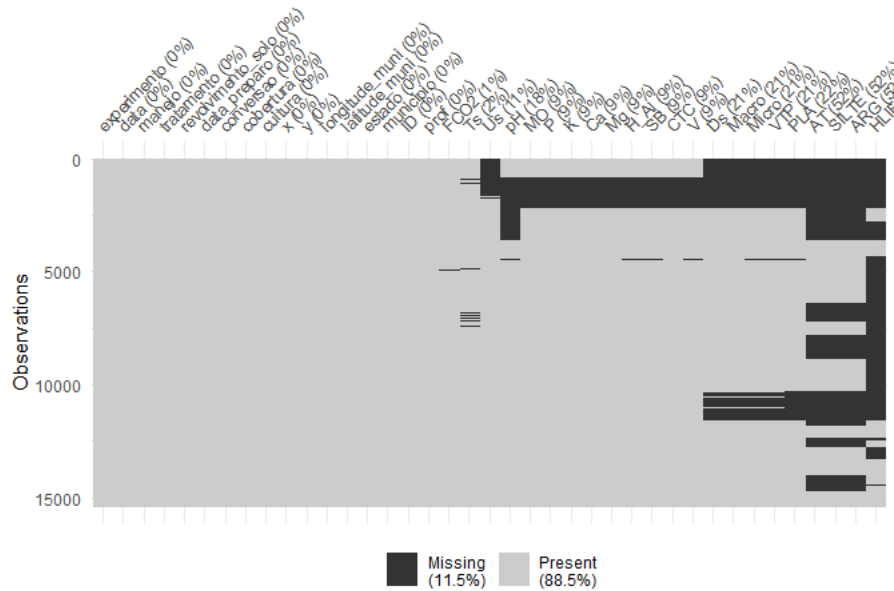
```
# Definindo o plano de multisession
future::plan("multisession")
```

```
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 = "~~~")))

```



```
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)
```

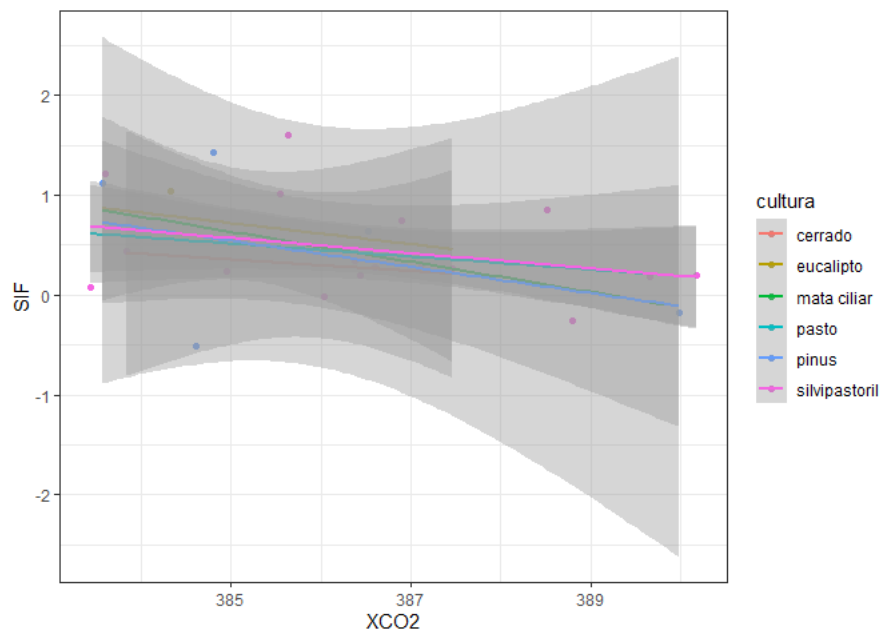
### Coordenadas das cidades

```
data_set <- left_join(fco2 %>%
  mutate(ano = lubridate::year(data),
    mes = lubridate::month(data)
  ),
  xco2 %>%
  select(data, mes, dia, longitude, latitude, XCO2, 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_radiance_771nm_idp_ph_sec_1_m_2_sr_1_um_1* 2.57743*10^(-19))/2)

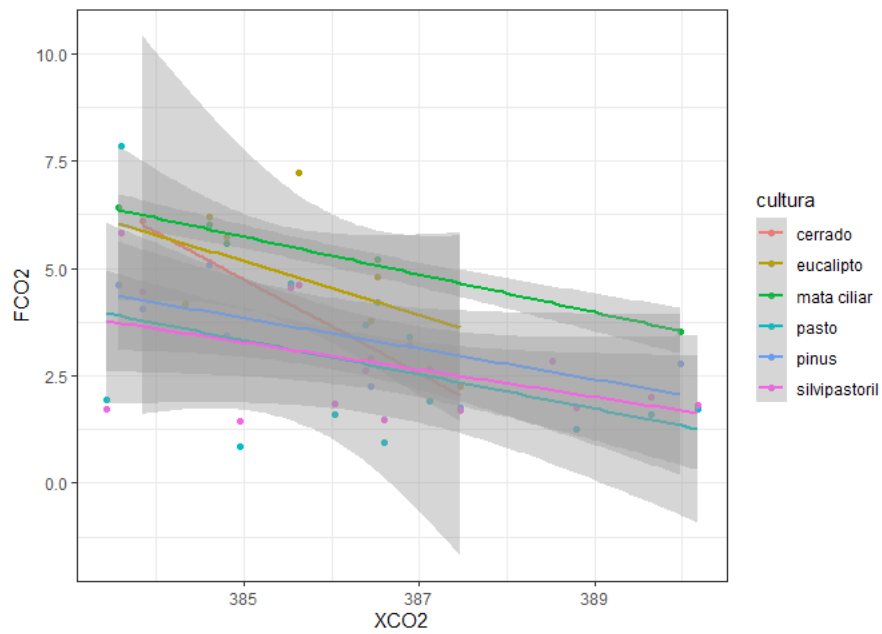
data_set<-data_set %>%
  select(-fluorescence_radiance_757nm_idp_ph_sec_1_m_2_sr_1_um_1, -fluorescence_radiance_771nm_idp_ph_sec_1_m_2_sr_1_um_1 ) %>%
  filter(dist <= .16, FCO2 <= 30 )
```

```
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))

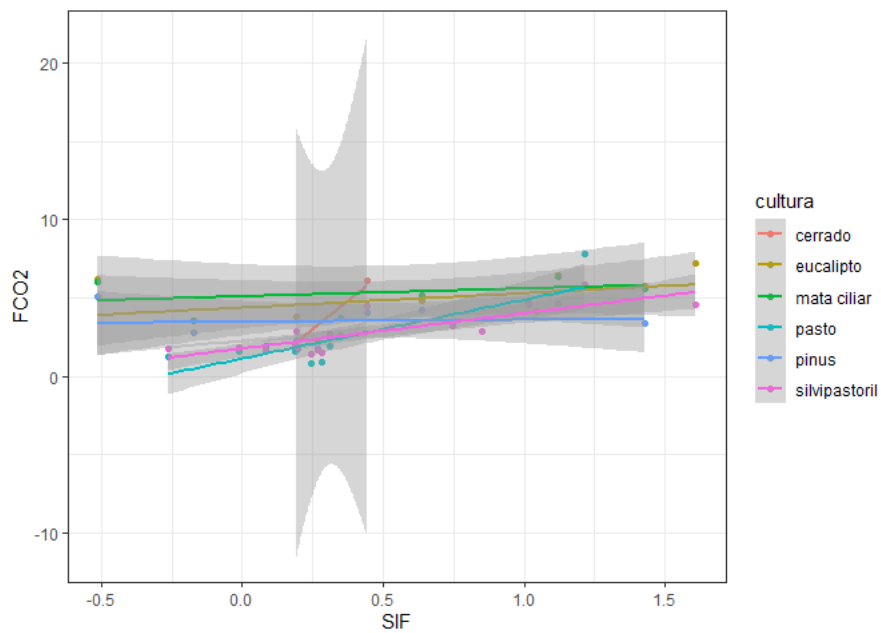
tab_medias %>%
  ggplot(aes(x=XCO2, y=SIF, color=cultura)) +
  geom_point()+
  geom_smooth(method = "lm")+
  theme_bw()
```



```
tab_medias %>%
  ggplot(aes(x=XCO2, y=FCO2,col=cultura)) +
  geom_point()+
  geom_smooth(method = "lm")+
  theme_bw()
```



```
tab_medias %>%
  ggplot(aes(y=FCO2, x=SIF, color=cultura)) +
  geom_point()+
  geom_smooth(method = "lm") +
  theme_bw()
```



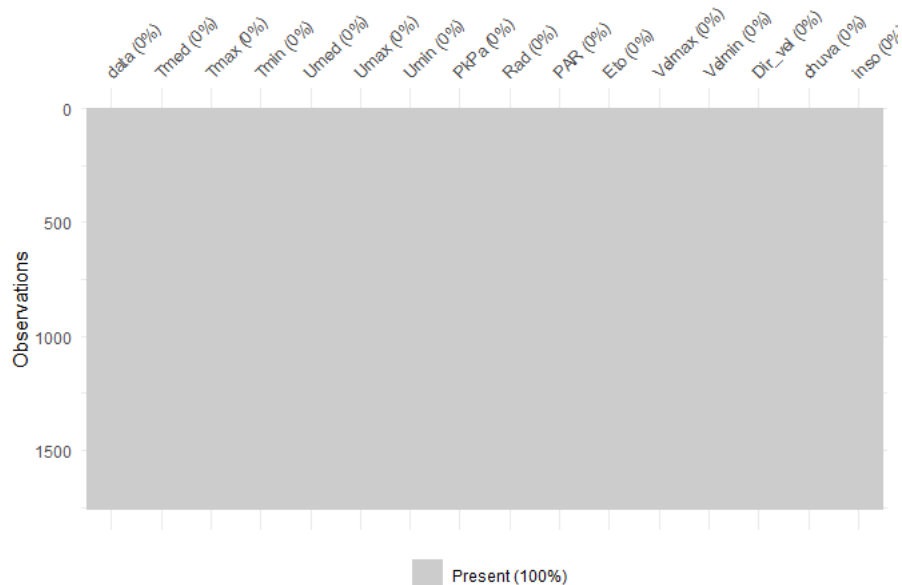
```
data_set_temporal <- data_set %>%
  filter(experimento == "Temporal")

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      <dtm> 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)
```



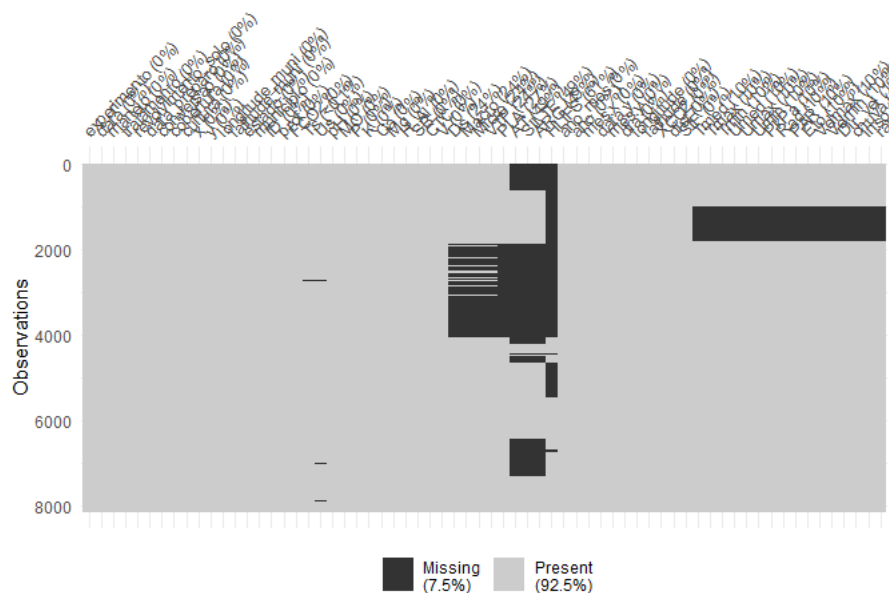
```
data_set_est_isa <- left_join(data_set %>%
  rename(data=data.x), dados_estacao, by = "data") %>%
  mutate(ra
nge_T = Tmax-Tmin)
```

## Terceira Aproximação

- Alvo: FCO2
- restrição dados após 2014
- Features: Atributos do Solo + Xco2 e SIF + Dados da Estação de ISA
- Modelo mais simples e geral
- Testar 3 métodos baseados em árvores de decisão

## visualização do banco de dados

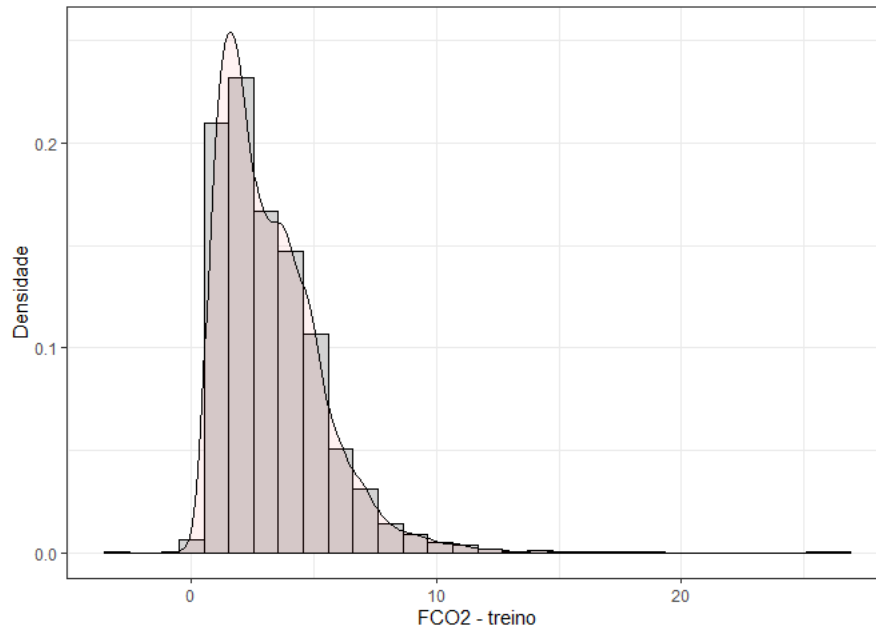
```
visdat::vis_miss(data_set_est_isa)
```



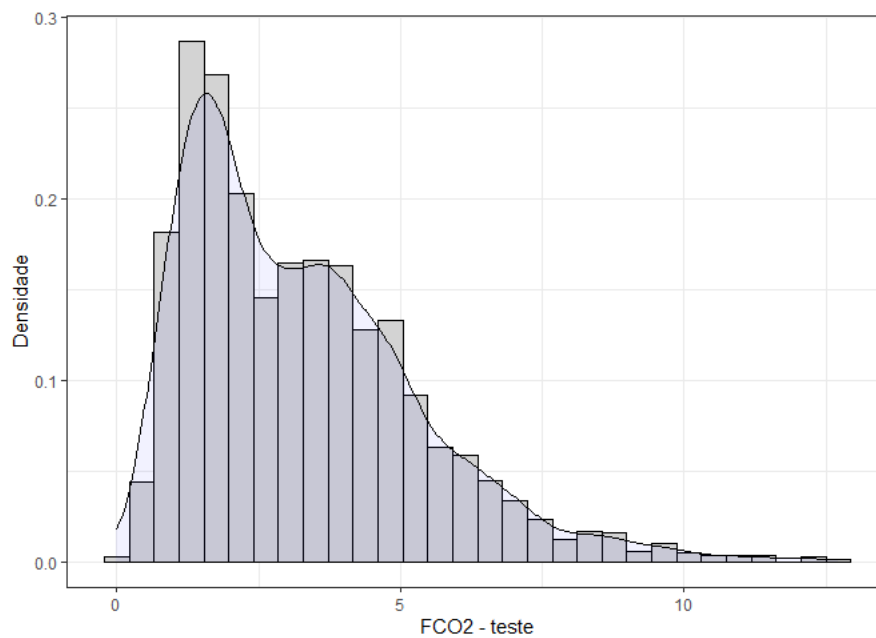
## Definindo a Base de treino e teste

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

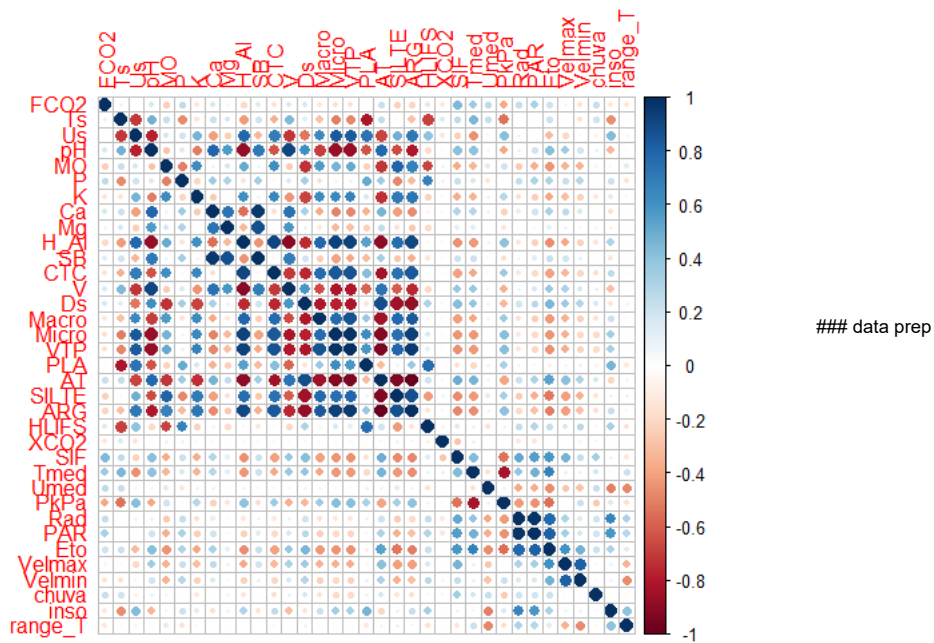
```
fco2_train <- training(fco2_initial_split)
# fco2_test <- testing(fco2_initial_split)
# visdat::vis_miss(fco2_test)
fco2_train %>%
  ggplot(aes(x=FCO2, y=..density..))+
  geom_histogram(bins = 30, color="black", fill="lightgray")+
  geom_density(alpha=.05,fill="red")+
  theme_bw() +
  labs(x="FCO2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



```
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)) %>% # imputação da mediana nos numéricos
  #step_poly(c(Us,Ts), degree = 2) %>%
  step_dummy(all_nominal_predictors())
bake(prepare(fco2_recipe), new_data = NULL)
#> # A tibble: 6,104 × 49
#>   Ts      Us      pH      MO      P      K      Ca      Mg      H_Al      SB
#>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 -0.803  0.989 -0.301  0.278 -0.786  0.313 -0.712 -1.25  1.18 -0.912
#> 2  0.574  1.65  0.141  0.871 -0.182 -0.354  1.39  2.64 -0.614  1.88
#> 3  0.264 -0.356  0.393 -0.910  1.33 -0.966 -0.329 -0.435 -1.38 -0.512
#> 4 -0.201  0.692 -0.0482  1.86  0.120  0.536  1.20  0.999 -0.293  1.29
#> 5  0.918 -0.275 -0.553  2.06 -0.384 -0.410 -0.903 -0.435  1.92 -0.863
#> 6  0.625 -0.824 -0.301  0.0797 -0.384 -0.743 -0.616 -1.46  0.862 -1.05
#> 7 -1.15  0.975 -0.174  0.871 -0.585 -0.0204 -0.233  0.179  1.18 -0.112
#> 8  0.419 -0.687 -0.301 -0.118  0.0192 -0.799 -0.712 -1.87  1.18 -1.26
#> 9 -0.0285 -0.0821 -0.237  0.574 -0.686  1.09 -0.520 -0.435  0.605 -0.393
#> 10 -1.03  0.854 -0.174  1.17 -0.686 -0.243 -0.425 -0.0256  0.862 -0.351
#> # 6,094 more rows
#> # 39 more variables: CTC <dbl>, V <dbl>, 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>, cultura_eucalipto <dbl>, cultura_mata.ciliar <dbl>, ...
```

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

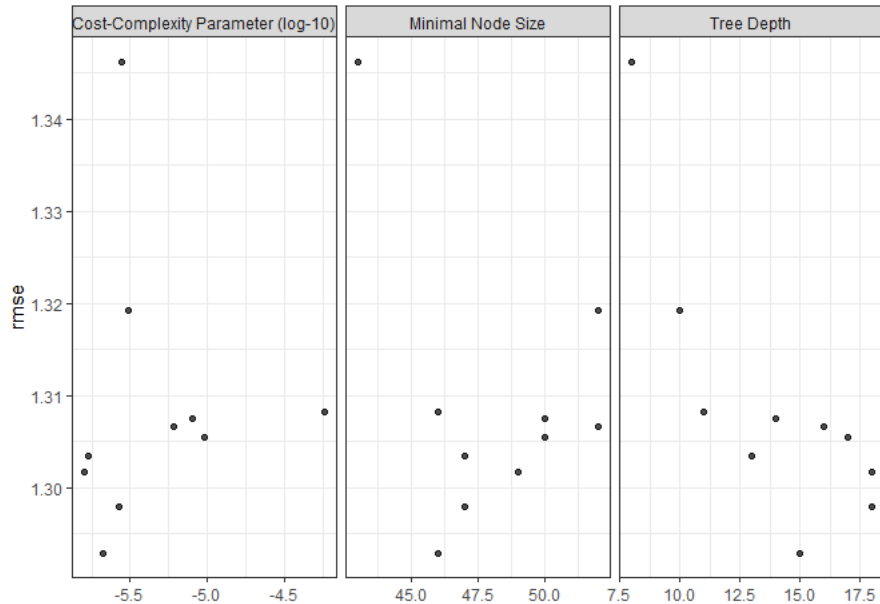


[illegible]

## 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)
```



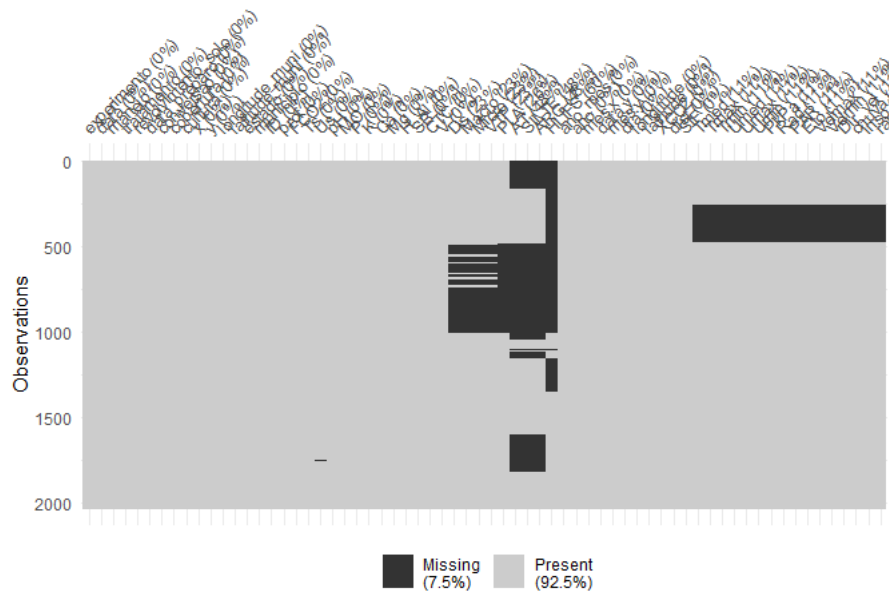
```
collect_metrics(fco2_dt_tune_grid)  
#> # A tibble: 10 × 9  
#>   cost_complexity tree_depth min_n .metric .estimator mean n std_err  
#>   <dbl> <int> <int> <chr> <chr> <dbl> <int> <dbl>  
#> 1 0.00000607 16 52 rmse standard 1.31 5 0.0378  
#> 2 0.00000170 13 47 rmse standard 1.30 5 0.0350  
#> 3 0.00000794 14 50 rmse standard 1.31 5 0.0376  
#> 4 0.00000963 17 50 rmse standard 1.31 5 0.0377  
#> 5 0.00000269 18 47 rmse standard 1.30 5 0.0357  
#> 6 0.00000211 15 46 rmse standard 1.29 5 0.0378  
#> 7 0.00000278 8 43 rmse standard 1.35 5 0.0307  
#> 8 0.0000577 11 46 rmse standard 1.31 5 0.0371  
#> 9 0.00000305 10 52 rmse standard 1.32 5 0.0368  
#> 10 0.00000159 18 49 rmse standard 1.30 5 0.0354  
#> # 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)
```

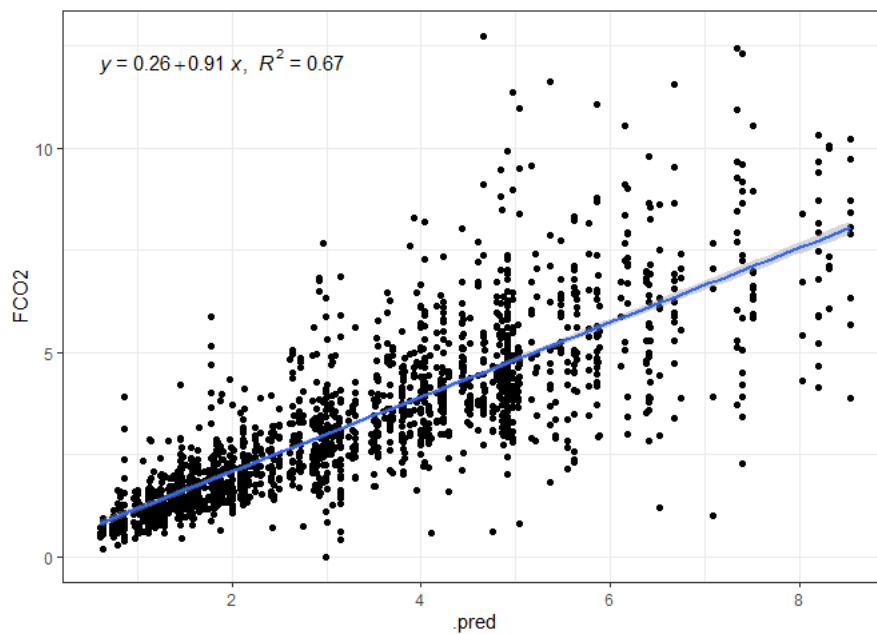
## Criando os preditos

```
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 = "~~~")))

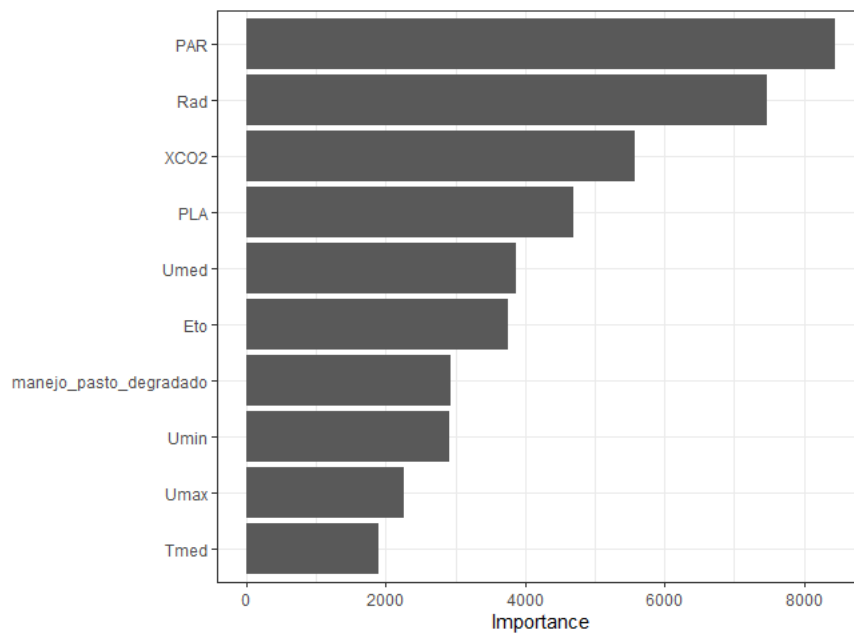
```



## Importância

```
fco2_dt_last_fit_model <- fco2_dt_last_fit$workflow[[1]]$fit$fit
vip(fco2_dt_last_fit_model)

```

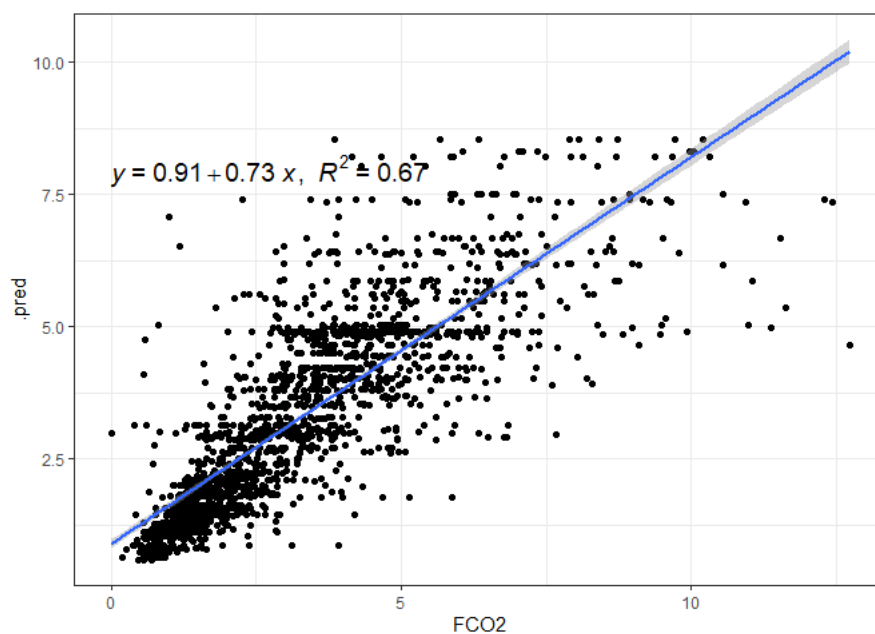


## Métricas

```
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

fco2_test_preds %>%
  ggplot(aes(x=FCO2,y=.pred))+
  geom_point()+
  geom_smooth(method = "lm")+
  stat_regline_equation(ggplot2::aes(
    label = paste(..eq.label.., ..rr.label.., sep = "~~~"),size=5)
```



```
# ggplot2::annotate('text',x=10.4,y=16.7,label=paste0('RMSE = ',round(my_rmse,2),', MAPE = ',
#                                     ,round(my_mape,2),'%'),size=5)+
# theme_bw()
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          0.8164169
#> R2         0.6665366
#> MSE        1.4020202
#> RMSE       1.1840693
#> MAE        0.7843213
#> MAPE       27.6688321
```

## Random Forest

### Definição do modelo

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

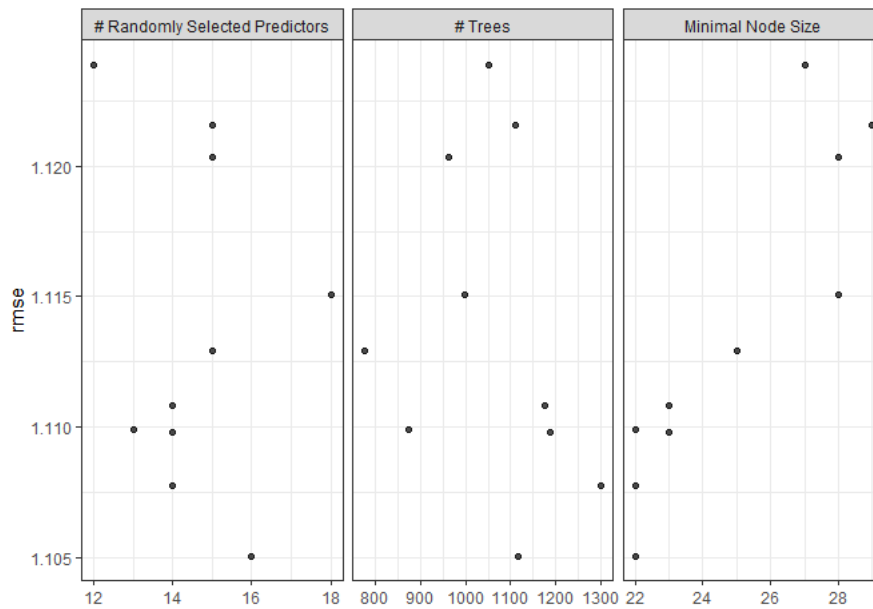
### Workflow

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

### Tune

```
grid_rf <- grid_random(
  min_n(range = c(20, 30)),
  mtry(range = c(10,20)),
  trees(range = c(769,1500) ),
  size = 10 #<-----
)
```

```
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: 10 × 9
#>   mtry trees min_n .metric .estimator mean     n std_err .config
#>   <int> <int> <int> <chr>    <chr>    <dbl> <int>   <dbl> <chr>
#> 1    14  1176    23 rmse    standard  1.11     5  0.0407 Preprocessor1_Model...
#> 2    13   874    22 rmse    standard  1.11     5  0.0407 Preprocessor1_Model...
#> 3    16  1116    22 rmse    standard  1.11     5  0.0404 Preprocessor1_Model...
#> 4    14  1187    23 rmse    standard  1.11     5  0.0412 Preprocessor1_Model...
#> 5    15  1110    29 rmse    standard  1.12     5  0.0402 Preprocessor1_Model...
#> 6    15   776    25 rmse    standard  1.11     5  0.0402 Preprocessor1_Model...
#> 7    15   964    28 rmse    standard  1.12     5  0.0406 Preprocessor1_Model...
#> 8    14  1301    22 rmse    standard  1.11     5  0.0401 Preprocessor1_Model...
#> 9    12  1051    27 rmse    standard  1.12     5  0.0403 Preprocessor1_Model...
#> 10   18   997    28 rmse    standard  1.12     5  0.0404 Preprocessor1_Model...
```

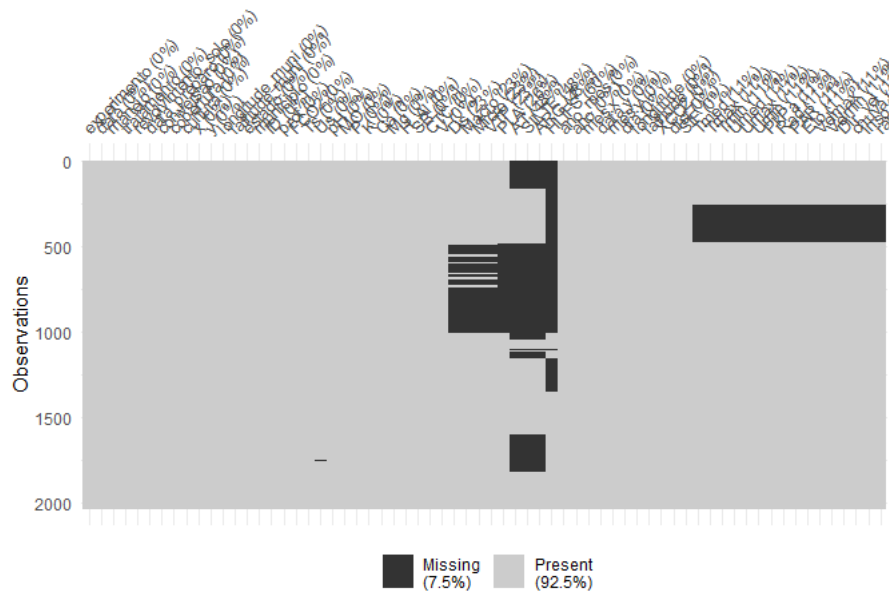
## Desempenho modelo final

```
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

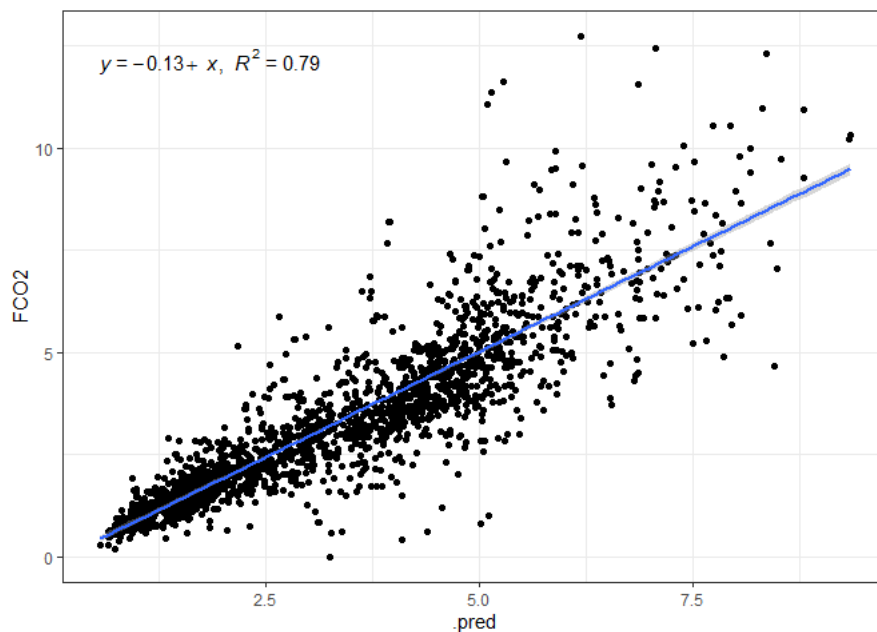
```
fco2_test_preds <- bind_rows(
  collect_predictions(fco2_rf_last_fit) %>% mutate(modelo = "rf")
)

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 = "~~~")))

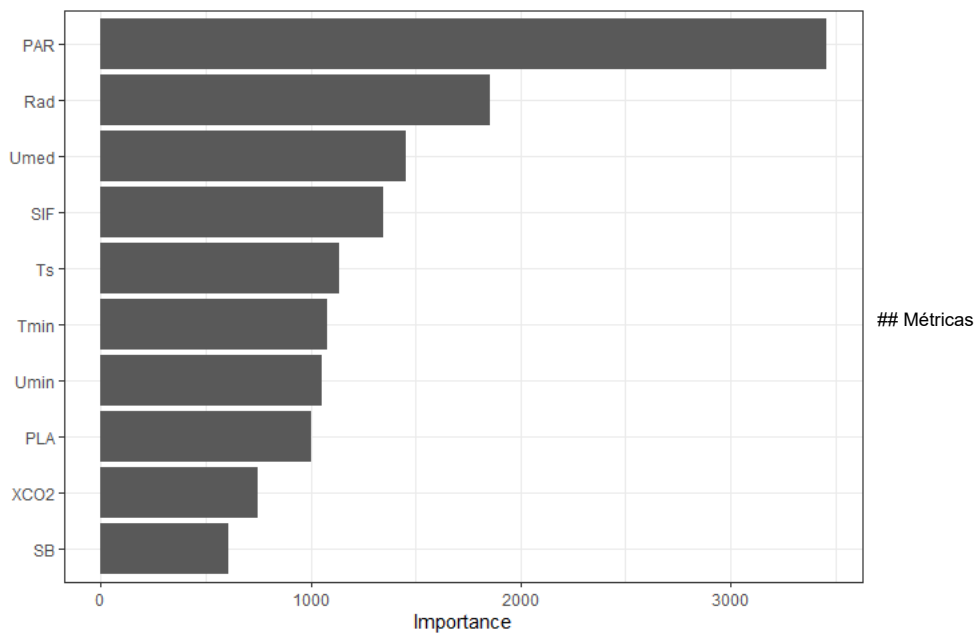
```



## Importância

```
fco2_rf_last_fit_model <- fco2_rf_last_fit$workflow[[1]]$fit$fit
vip(fco2_rf_last_fit_model)

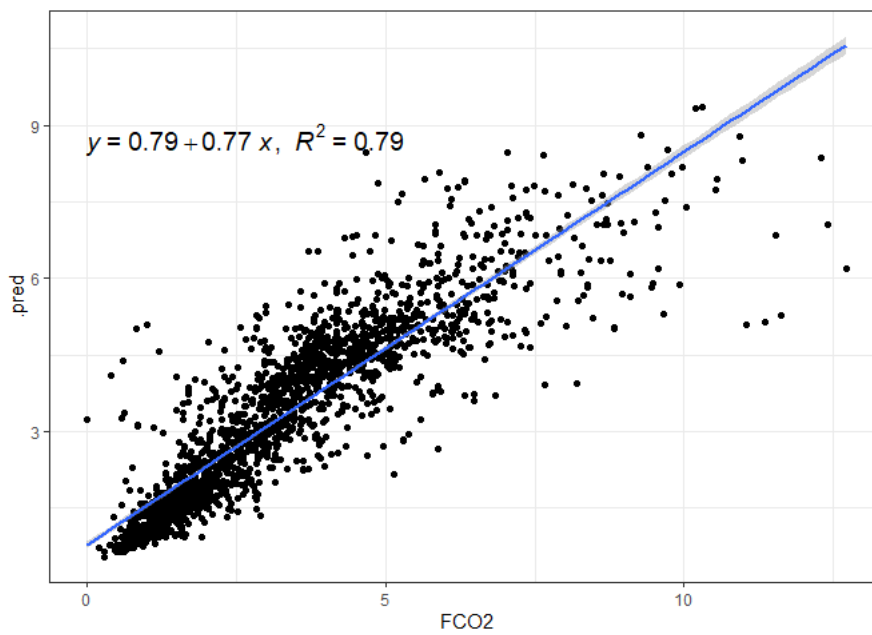
```



```
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

fco2_test_preds %>%
  ggplot(aes(x=FCO2,y=.pred))+
  geom_point()+
  geom_smooth(method = "lm")+
  stat_regline_equation(ggplot2::aes(
    label = paste(..eq.label.., ..rr.label.., sep = "plain(\\",\\")~~")),size=5)
```





```
# ggplot2::annotate('text',x=10.4,y=16.7,label=paste0('RMSE = ',round(my_rmse,2),', MAPE = ',
#                                     ,round(my_mape,2),'%'),size=5)+
# theme_bw()
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          0.8898384
#> R2         0.7918123
#> MSE        0.8623314
#> RMSE       0.9286180
#> MAE        0.5946322
#> MAPE       21.8970337
```

## Boosting gradient tree (xgb)

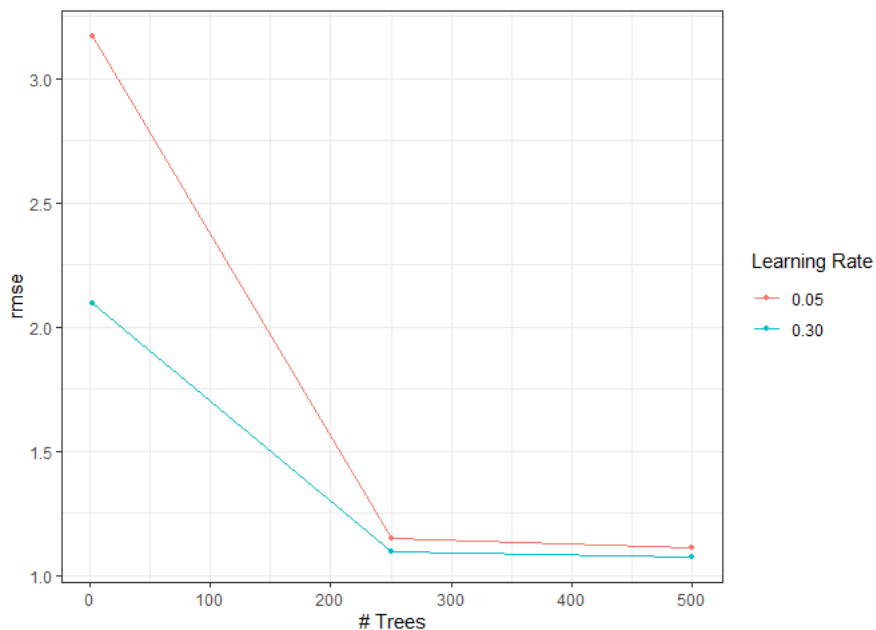
```
cores = 4
fco2_xgb_model <- boost_tree(
  mtry = 0.8,
  trees = tune(), # <-----
  min_n = 5,
  tree_depth = 4,
  loss_reduction = 0, # lambda
  learn_rate = tune(), # epsilon
  sample_size = 0.8
) %>%
  set_mode("regression") %>%
  set_engine("xgboost", nthread = cores, counts = FALSE)
```

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

```
grid_xgb <- expand_grid(
  learn_rate = c(0.05, 0.3),
  trees = c(2, 250, 500)
)
```

### Passo 1

```
fco2_xgb_tune_grid <- tune_grid(
  fco2_xgb_wf,
  resamples = fco2_resamples,
  grid = grid_xgb,
  metrics = metric_set(rmse)
)
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
#>   <dbl>      <dbl> <chr>    <chr>    <dbl> <int>   <dbl> <chr>
#> 1   500        0.3  rmse    standard  1.08     5  0.0440 Preprocessor1_Model6
#> 2   250        0.3  rmse    standard  1.10     5  0.0400 Preprocessor1_Model5
#> 3   500        0.05 rmse    standard  1.11     5  0.0335 Preprocessor1_Model3
#> 4   250        0.05 rmse    standard  1.15     5  0.0352 Preprocessor1_Model2
#> 5     2        0.3  rmse    standard  2.09     5  0.0400 Preprocessor1_Model4
#> 6     2        0.05 rmse    standard  3.17     5  0.0432 Preprocessor1_Model1
```

```
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
#>   <dbl>      <dbl> <chr>
#> 1   500        0.3 Preprocessor1_Model6
```

## Passo 2

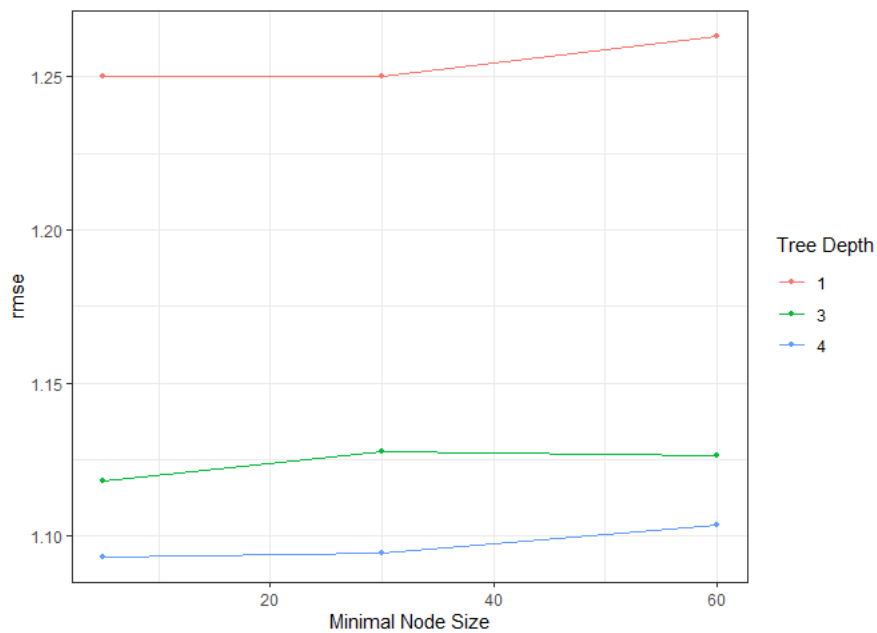
```
fco2_xgb_model <- boost_tree(
  mtry = 0.8,
  trees = fco2_xgb_select_best_passo1$trees,
  min_n = tune(),
  tree_depth = tune(),
  loss_reduction = 0,
  learn_rate = fco2_xgb_select_best_passo1$learn_rate,
  sample_size = 0.8
) %>%
  set_mode("regression") %>%
  set_engine("xgboost", nthread = cores,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

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

#### Grid
fco2_xgb_grid <- expand_grid(
  tree_depth = c(1, 3, 4),
  min_n = c(5, 30, 60)
)

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
#>   min_n tree_depth .metric .estimator mean      n std_err .config
#>   <dbl>   <dbl> <chr>   <chr>   <dbl> <int>   <dbl> <chr>
#> 1     5       4 rmse     standard 1.09     5 0.0314 Preprocessor1_Model3
#> 2    30       4 rmse     standard 1.09     5 0.0366 Preprocessor1_Model6
#> 3    60       4 rmse     standard 1.10     5 0.0287 Preprocessor1_Model9
#> 4     5       3 rmse     standard 1.12     5 0.0310 Preprocessor1_Model2
#> 5    60       3 rmse     standard 1.13     5 0.0329 Preprocessor1_Model8

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
#>   <dbl>   <dbl> <chr>
#> 1     5       4 Preprocessor1_Model3
```

## Passo 3

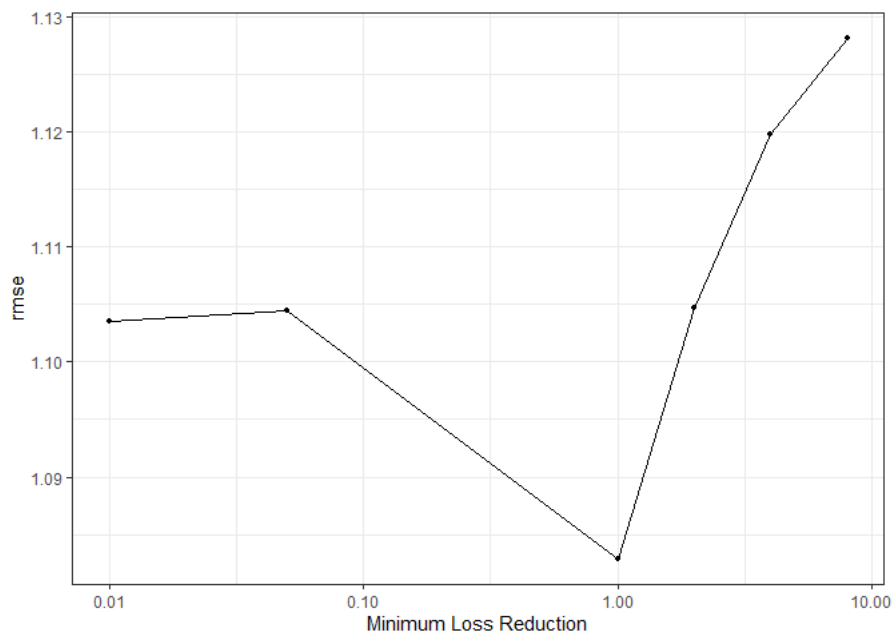
```
fco2_xgb_model <- boost_tree(
  mtry = 0.8,
  trees = fco2_xgb_select_best_passo1$trees,
  min_n = fco2_xgb_select_best_passo2$min_n,
  tree_depth = fco2_xgb_select_best_passo2$tree_depth,
  loss_reduction = tune(),
  learn_rate = fco2_xgb_select_best_passo1$learn_rate,
  sample_size = 0.8
) %>%
  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 <- expand_grid(
  loss_reduction = c(0.01, 0.05, 1, 2, 4, 8)
)

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  rmse    standard    1.08     5  0.0243 Preprocessor1_Model3
#> 2        0.01 rmse    standard    1.10     5  0.0319 Preprocessor1_Model1
#> 3        0.05 rmse    standard    1.10     5  0.0334 Preprocessor1_Model2
#> 4         2  rmse    standard    1.10     5  0.0317 Preprocessor1_Model4
#> 5         4  rmse    standard    1.12     5  0.0311 Preprocessor1_Model5
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 Preprocessor1_Model3
```

## Passo 4

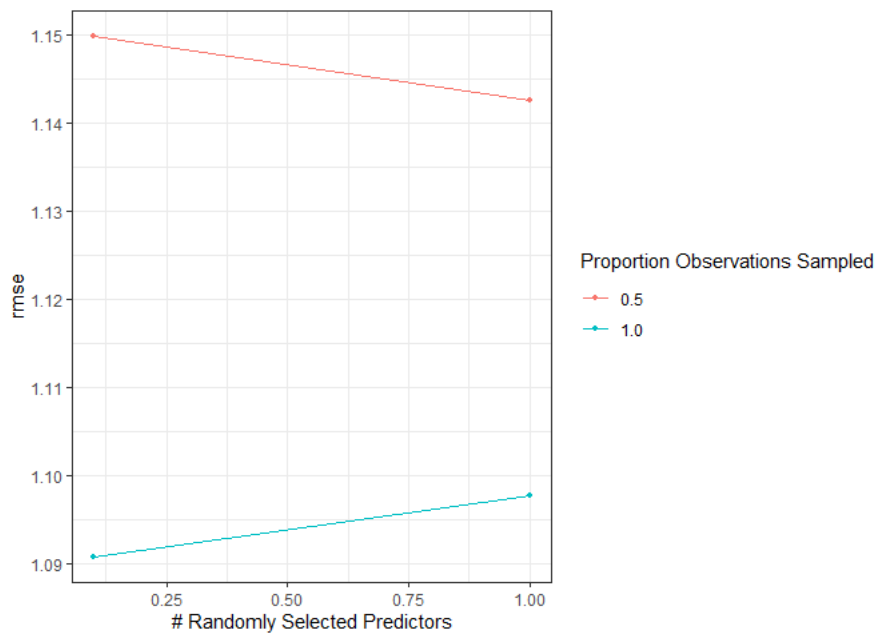
```
fco2_xgb_model <- boost_tree(
  mtry = tune(),
  trees = fco2_xgb_select_best_passo1$trees,
  min_n = fco2_xgb_select_best_passo2$min_n,
  tree_depth = fco2_xgb_select_best_passo2$tree_depth,
  loss_reduction = fco2_xgb_select_best_passo3$loss_reduction,
  learn_rate = fco2_xgb_select_best_passo1$learn_rate,
  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 = 2), ## <---
  mtry = seq(0.1, 1.0, length.out = 2) ## <---
)

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: 4 × 8
#>   mtry sample_size .metric .estimator mean      n std_err .config
#>   <dbl>      <dbl> <chr>    <chr>    <dbl> <int>   <dbl> <chr>
#> 1  0.1          1  rmse    standard  1.09      5  0.0329 Preprocessor1_Model2
#> 2  1            1  rmse    standard  1.10      5  0.0312 Preprocessor1_Model4
#> 3  1            0.5 rmse    standard  1.14      5  0.0329 Preprocessor1_Model3
#> 4  0.1          0.5 rmse    standard  1.15      5  0.0332 Preprocessor1_Model1
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  0.1          1 Preprocessor1_Model2
```

## Passo 5

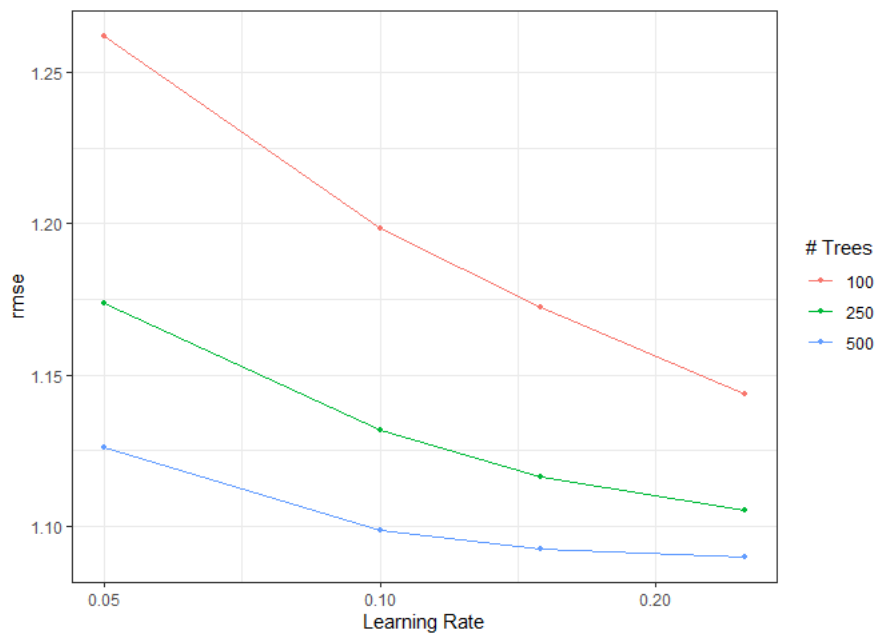
```
fco2_xgb_model <- boost_tree(
  mtry = fco2_xgb_select_best_passo4$mtry,
  trees = tune(),
  min_n = fco2_xgb_select_best_passo2$min_n,
  tree_depth = fco2_xgb_select_best_passo2$tree_depth,
  loss_reduction = fco2_xgb_select_best_passo3$loss_reduction,
  learn_rate = tune(),
  sample_size = fco2_xgb_select_best_passo4$sample_size
) |>
  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(0.05, 0.10, 0.15, 0.25),
  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  1.09     5  0.0311 Preprocessor1_Model12
#> 2   500      0.15 rmse    standard  1.09     5  0.0307 Preprocessor1_Model09
#> 3   500      0.1  rmse    standard  1.10     5  0.0308 Preprocessor1_Model06
#> 4   250      0.25 rmse    standard  1.11     5  0.0341 Preprocessor1_Model11
#> 5   250      0.15 rmse    standard  1.12     5  0.0300 Preprocessor1_Model08
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_Model12
```

## 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)
```

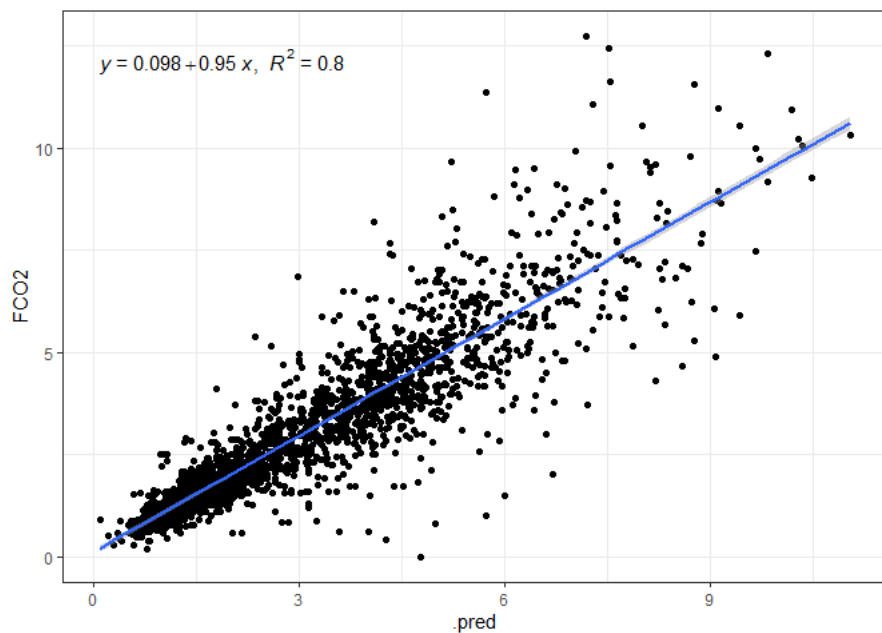
```
df <- data.frame(
  mtry = fco2_xgb_select_best_passo4$mtry,
  trees = fco2_xgb_select_best_passo5$trees,
  min_n = fco2_xgb_select_best_passo2$min_n,
  tree_depth = fco2_xgb_select_best_passo2$tree_depth,
  loss_reduction = fco2_xgb_select_best_passo3$loss_reduction,
  learn_rate = fco2_xgb_select_best_passo5$learn_rate,
  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")
)
```

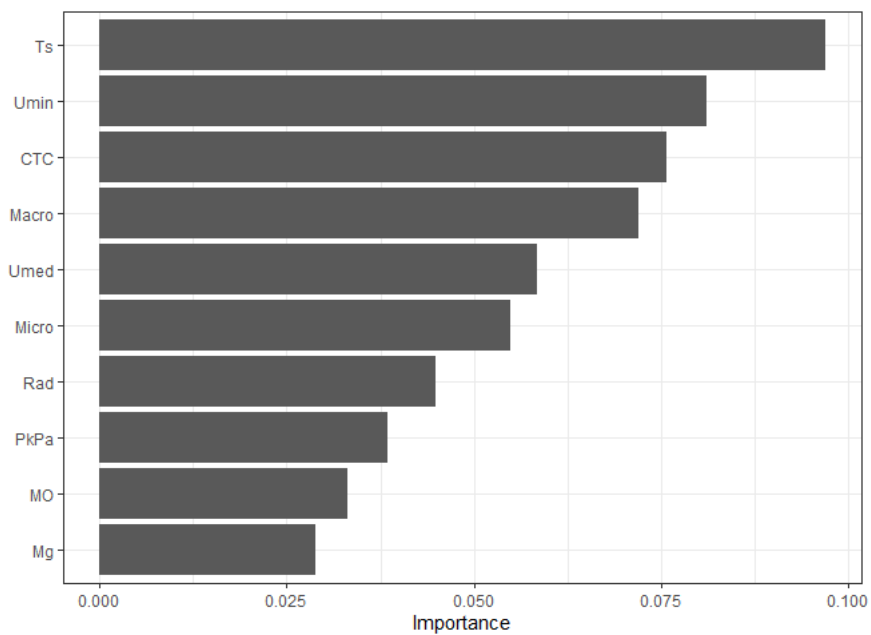
```
fco2_test_preds %>%
  ggplot(aes(x=.pred, y=FCO2)) +
  geom_point() +
  theme_bw() +
  geom_smooth(method = "lm") +
  stat_regline_equation(ggplot2::aes(
    label = paste(..eq.label.., ..rr.label.., sep = "~~~")))

```



```
fco2_xgb_last_fit_model <- fco2_xgb_last_fit$.workflow[[1]]$fit$fit
vip(fco2_xgb_last_fit_model)

```

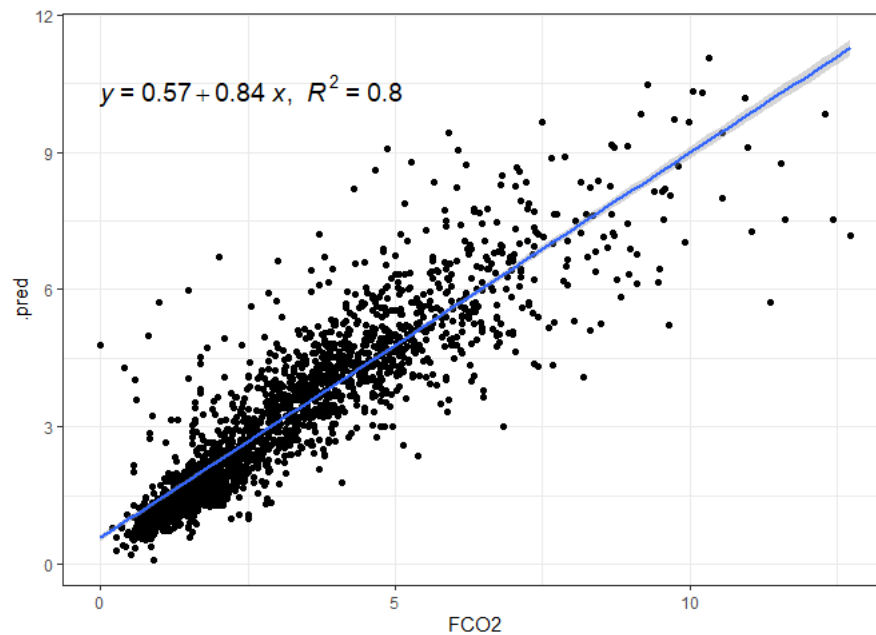


# Métricas

```
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

fco2_test_preds %>%
  ggplot(aes(x=FCO2,y=.pred))+
  geom_point()+
  geom_smooth(method = "lm")+
  stat_regline_equation(ggplot2::aes(
    label = paste(..eq.label.., ..rr.label.., sep = "plain(\"\",\"\")~~"),size=5)
```



```
# ggplot2::annotate('text',x=10.4,y=16.7,label=paste0('RMSE = ',round(my_rmse,2),', MAPE = ',
#                                                     ,round(my_mape,2),'%'),size=5)+
# theme_bw()
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              0.8966855
#> R2             0.8040449
#> MSE            0.8196266
#> RMSE           0.9053323
#> MAE            0.5919036
#> MAPE           22.0173046
```