

Modelos de Sobrevida, Regressão Quantílica e outras aplicações

Tiago Mendonça dos Santos



tiagoms.com



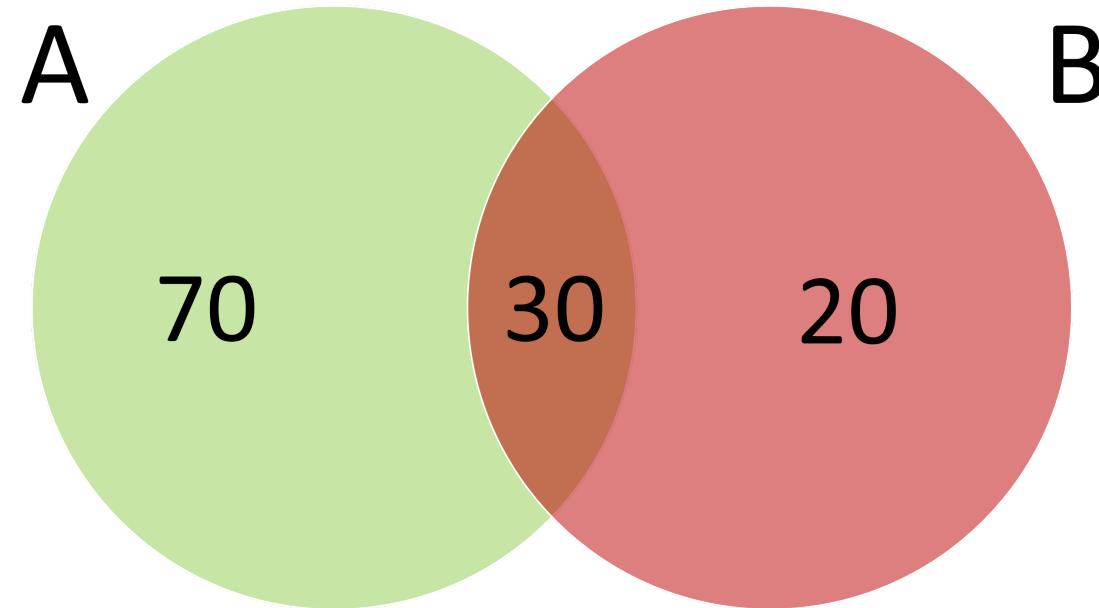
[tiagomendonca](https://github.com/tiagomendonca)



tiagoms1@insper.edu.br

Conceitos de Probabilidade

Probabilidade



$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$
 e
$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

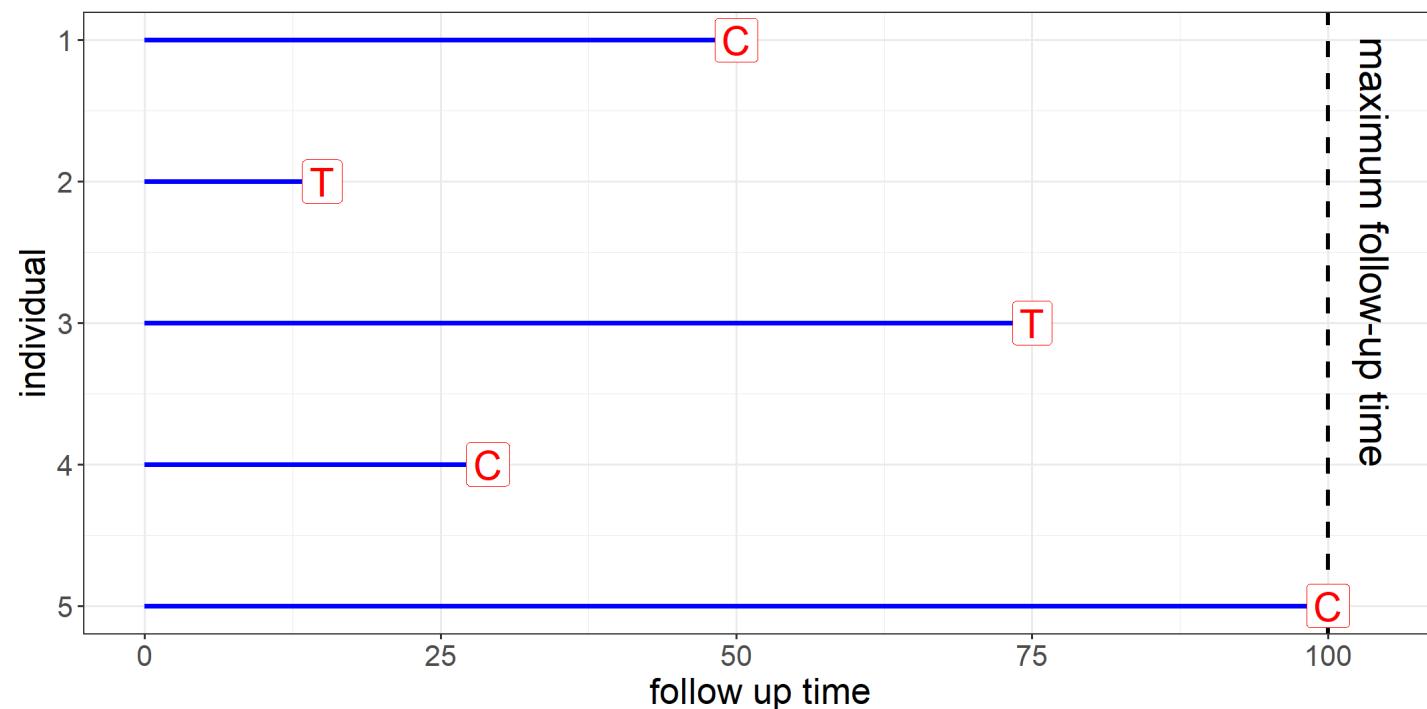
Modelos de Sobrevida

Contexto

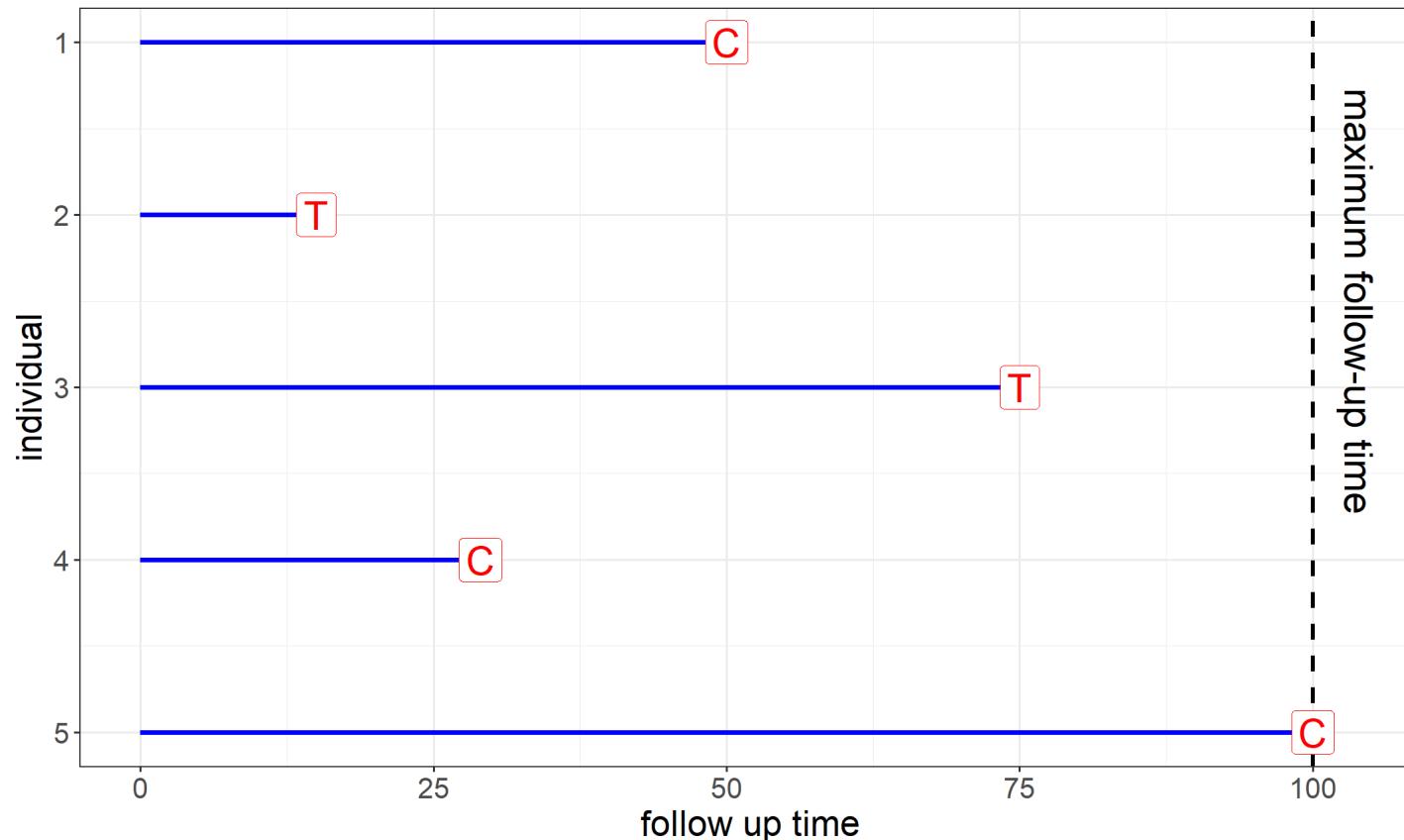
Nessa classe de modelos temos como objetivo estudar o tempo até a ocorrência de um evento de interesse.

Exemplos: tempo de sobrevivência de indivíduos que recebem um determinado tratamento, o tempo de assinatura de um serviço ou o número de ligações de uma central de telemarketing até conseguir falar com o cliente.

O tempo até o evento de interesse será denotado por **T**. No entanto, não observaremos alguns indivíduos o tempo suficiente para que apresentem o evento de interesse. Para esses indivíduos, dizemos que foram **censurados** no instante **C**.



Contexto

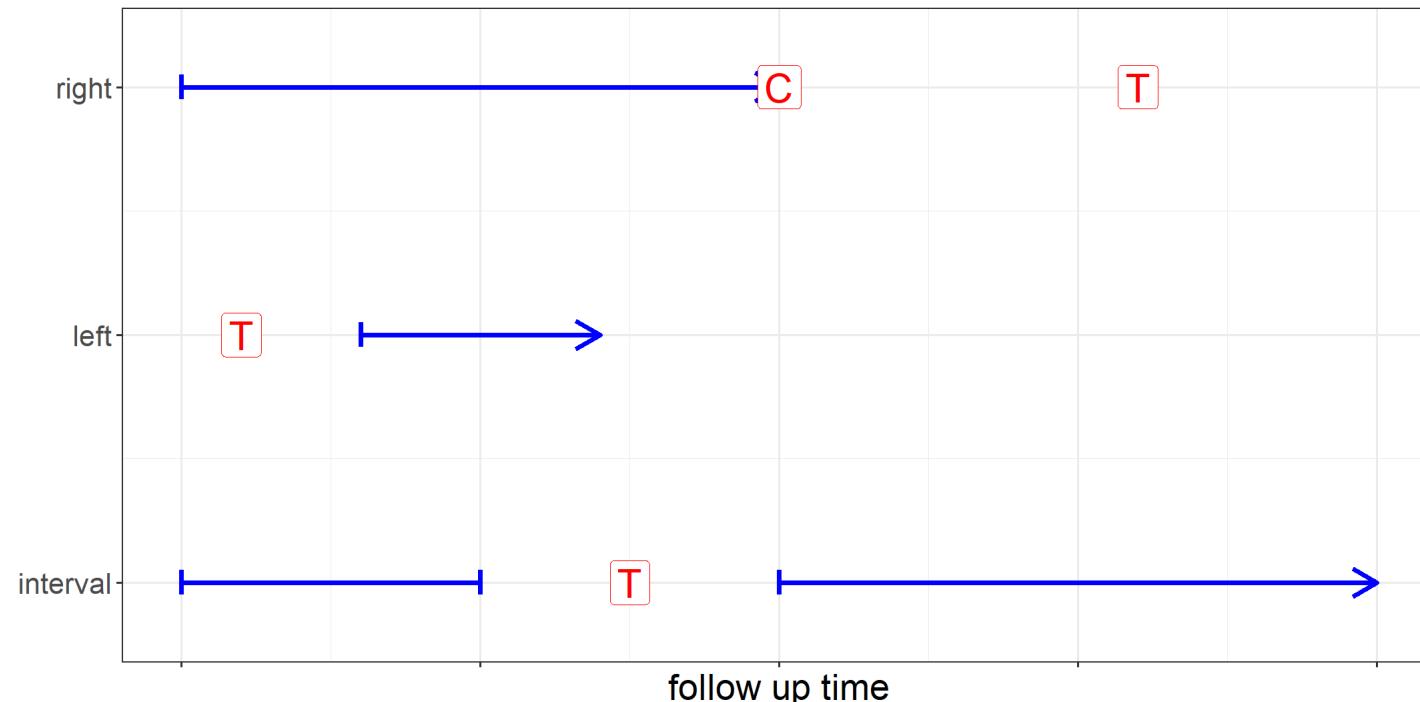


$$Y = \min(T, C) \quad \text{e} \quad \delta = \begin{cases} 1, & \text{se } T \leq C \\ 0, & \text{se } T > C \end{cases}$$

Tipos de censura

Podemos considerar os seguintes tipos de censura:

- **direita:** o evento ocorrerá após o tempo de censura observado (como apresentado anteriormente).
- **esquerda:** quando o evento ocorre antes do tempo registrado.
- **intervalar:** situação em que sabemos o intervalo de tempo em que o evento ocorreu. É comum ter esse tipo de censura em estudos que temos visitas de acompanhamento.



Função de sobrevivência

A curva/função de sobrevivência, que dá a probabilidade de um evento ocorrer após um instante de tempo t , é dada por:

$$S(t) = P(T > t)$$

Para obter essa probabilidade, considere $d_1 < d_2 < \dots < d_K$ os instantes de eventos distintos e q_k o número de indivíduos que apresentaram evento no instante k . Ainda, considere r_k o número de pacientes em risco em um instante anterior a d_k . Assim, temos

$$\begin{aligned} P(T > d_k) &= P(T > d_k \cap T > d_{k-1}) + P(T > d_k \cap T \leq d_{k-1}) \\ &= P(T > d_k | T > d_{k-1})P(T > d_{k-1}) + 0. \end{aligned}$$

Portanto, podemos escrever

$$S(d_k) = P(T > d_k) = P(T > d_k | T > d_{k-1})P(T > d_{k-1}) = P(T > d_k | T > d_{k-1})S(d_{k-1}).$$

E, aplicando a relação anterior recursivamente, temos

$$S(d_k) = P(T > d_k | T > d_{k-1}) \dots P(T > d_2 | T > d_1)P(T > d_1).$$

Estimador de Kaplan-Meier

$$\widehat{S}(d_k) = \prod_{j=1}^k \left(\frac{r_j - q_j}{r_j} \right)$$

Lembre que

$$S(d_k) = P(\text{T} > d_k | \text{T} > d_{k-1}) \dots P(\text{T} > d_2 | \text{T} > d_1) P(\text{T} > d_1)$$

e que podemos considerar o seguinte estimador

$$\widehat{P}(\text{T} > d_j | \text{T} > d_{j-1}) = \frac{r_j - q_j}{r_j}$$

Exemplo

Considere a situação em que temos os tempos observados e o indicador de evento. Construa a curva de sobrevivência utilizando o estimador de Kaplan-Meier.

y	delta
5	1
6	0
7	0
10	1
15	1
20	0

Exemplo

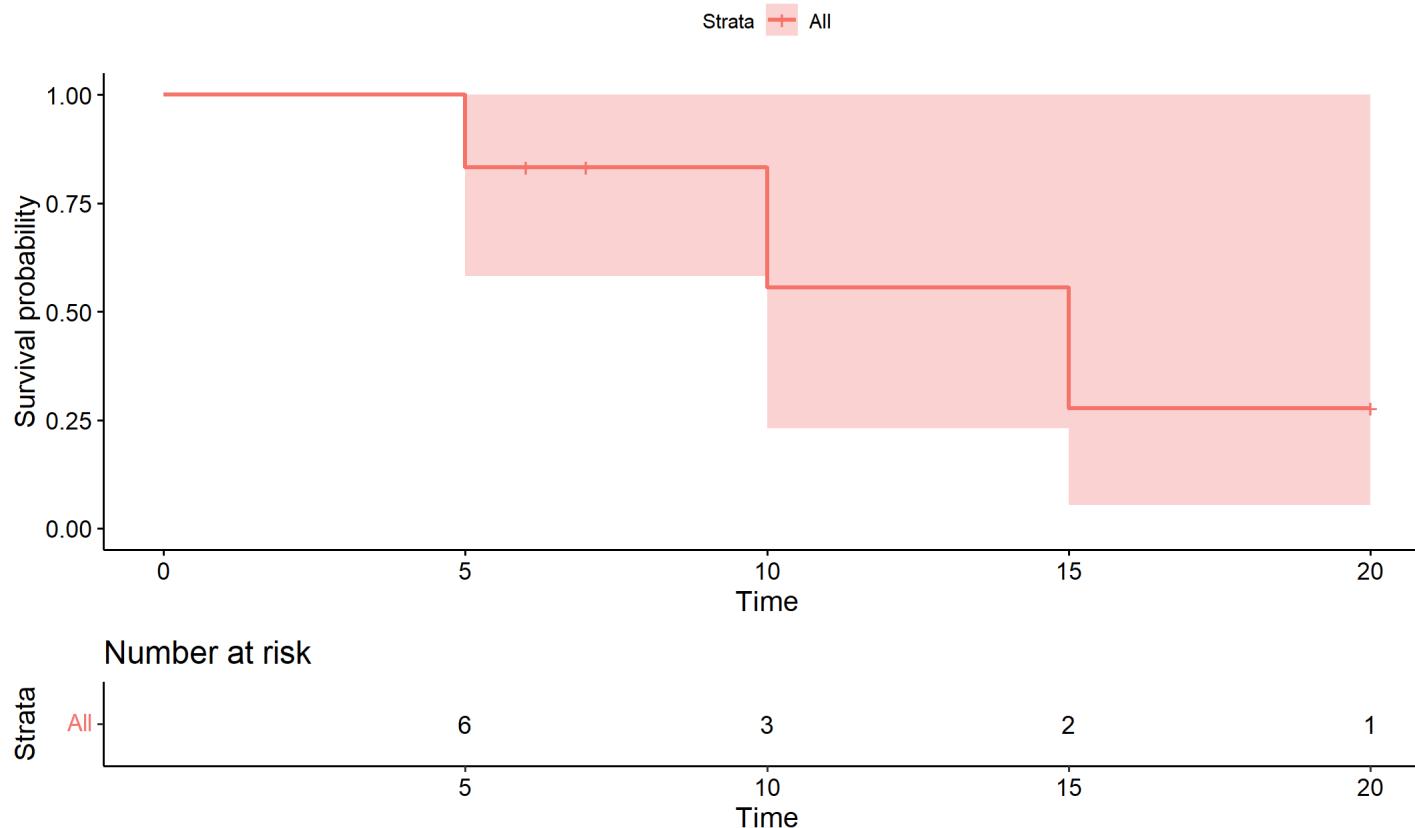
```
fit <- survfit(Surv(y, delta) ~ 1, data = df)

summary(fit)

## Call: survfit(formula = Surv(y, delta) ~ 1, data = df)
##
##   time n.risk n.event survival std.err lower 95% CI upper 95% CI
##      5      6       1     0.833   0.152    0.5827      1
##     10      3       1     0.556   0.248    0.2312      1
##     15      2       1     0.278   0.232    0.0539      1
```

Exemplo

```
survminer::ggsurvplot(fit, risk.table = TRUE)
```



Telco Churn

Vamos considerar o conjunto de dados Telco Customer Churn (analisado em outras aulas).

```
library(survival)
library(survminer)

dados <- read_csv("dados/WA_Fn-UseC_-Telco-Customer-Churn.csv") %>%
  mutate(Churn = case_when(Churn == "Yes" ~ 1, Churn == "No" ~ 0)) %>%
  mutate_if(is.character, factor)
```

Trabalharemos com as seguintes medidas:

- **gender:** Whether the customer is a male or a female
- **SeniorCitizen:** Whether the customer is a senior citizen or not (1, 0)
- **Dependents:** Whether the customer has dependents or not (Yes, No)
- **Contract:** The contract term of the customer (Month-to-month, One year, Two year)

O tempo de acompanhamento e variável indicadora de churn são apresentados a seguir:

- **tenure** Number of months the customer has stayed with the company
- **Churn:** Whether the customer churned or not (Yes or No)

```
(fit <- survfit(Surv(tenure, Churn) ~ 1, dados))

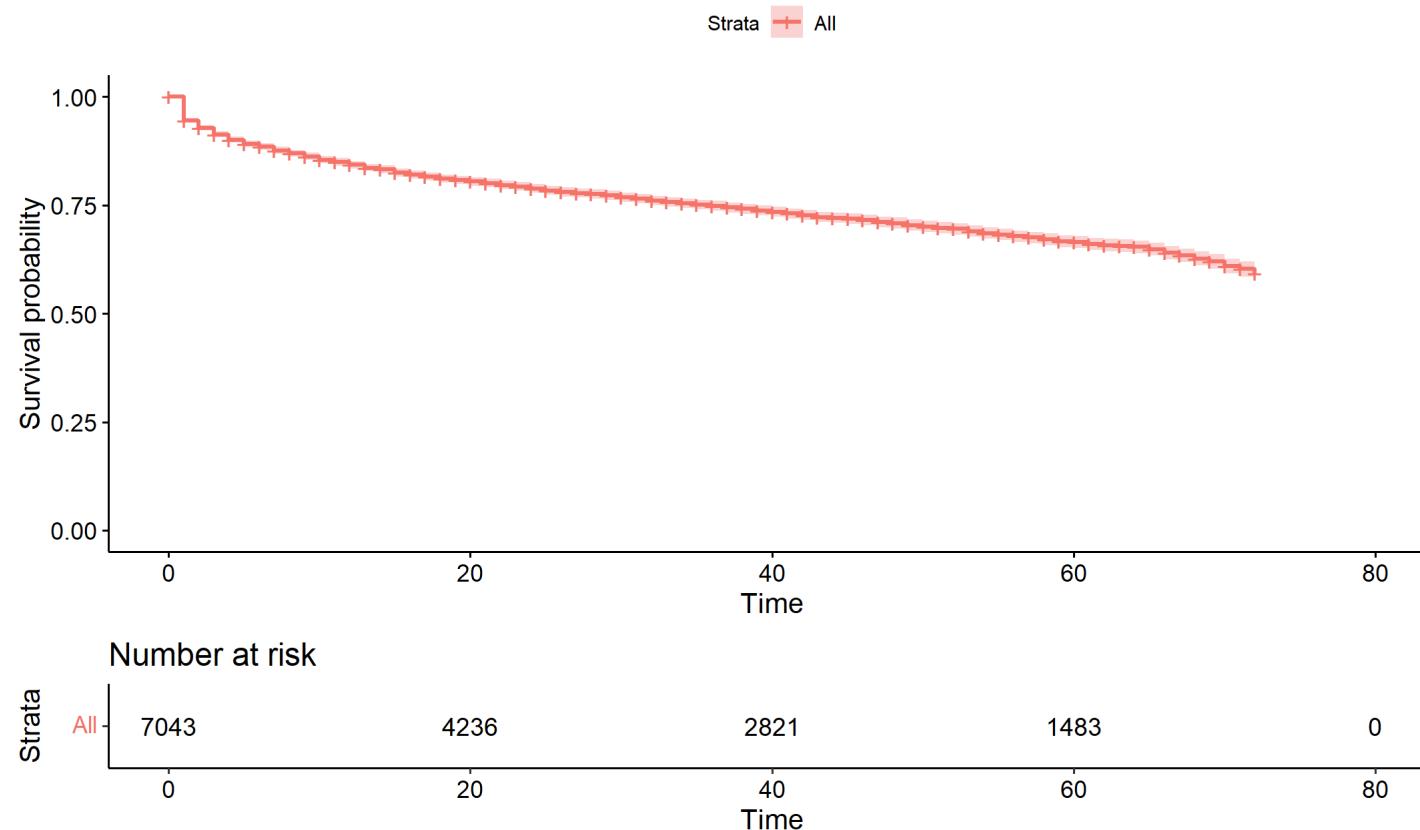
## Call: survfit(formula = Surv(tenure, Churn) ~ 1, data = dados)
##
##      n events median 0.95LCL 0.95UCL
## [1,] 7043    1869      NA      NA      NA

summary(fit)

## Call: survfit(formula = Surv(tenure, Churn) ~ 1, data = dados)
##
##   time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1    7032     380    0.946 0.00270     0.941     0.951
##     2    6419     123    0.928 0.00310     0.922     0.934
##     3    6181      94    0.914 0.00338     0.907     0.920
##     4    5981      83    0.901 0.00361     0.894     0.908
##     5    5805      64    0.891 0.00377     0.884     0.899
##     6    5672      40    0.885 0.00388     0.877     0.892
##     7    5562      51    0.877 0.00400     0.869     0.885
##     8    5431      42    0.870 0.00411     0.862     0.878
##     9    5308      46    0.862 0.00422     0.854     0.871
##    10    5189      45    0.855 0.00433     0.846     0.863
##    11    5073      31    0.850 0.00440     0.841     0.858
##    12    4974      38    0.843 0.00449     0.834     0.852
```

Telco Churn

```
ggsurvplot(fit, risk.table = TRUE)
```

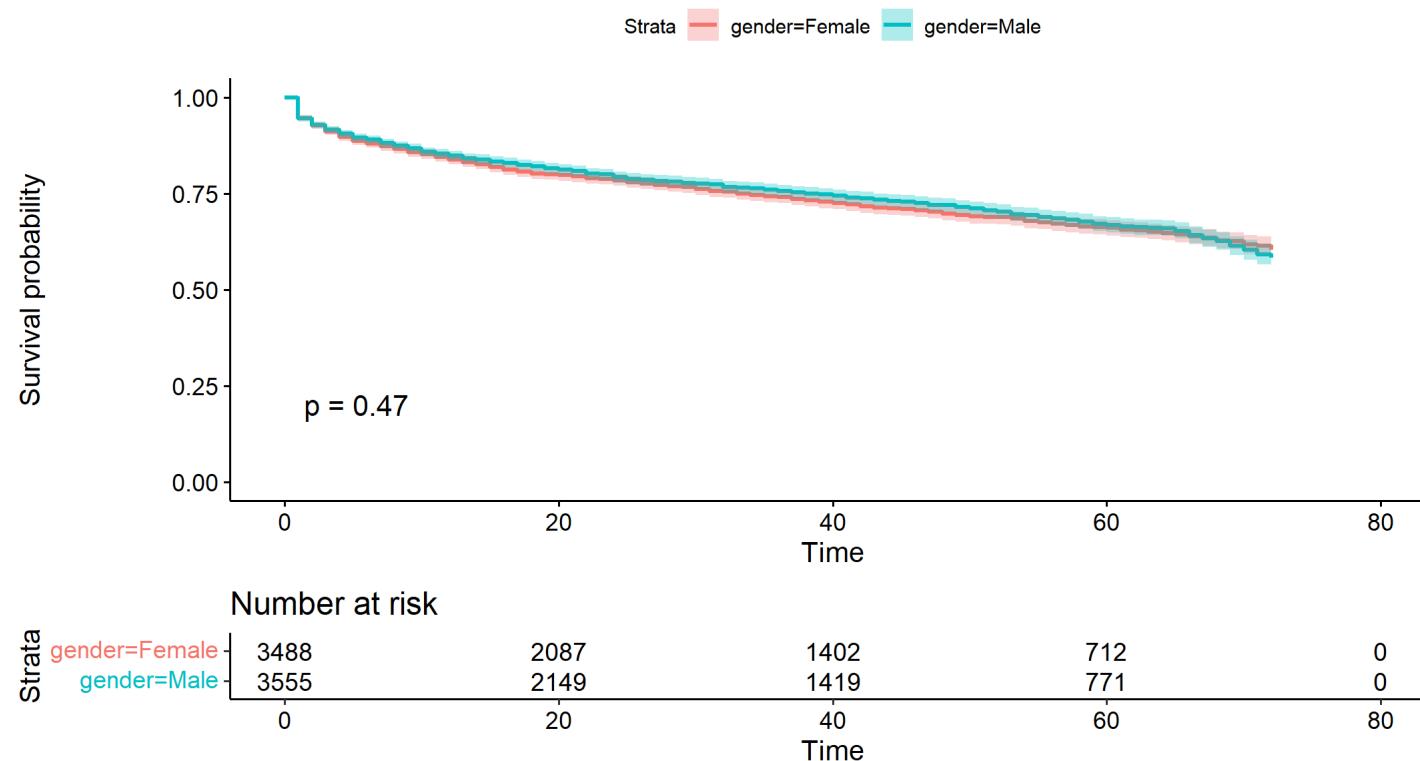


Telco Churn

Faça a curva de sobrevida para as medidas *gender*, *SeniorCitizen*, *Dependents* e *Contract*.

```
fit <- survfit(Surv(tenure, Churn) ~ gender, dados)

ggsurvplot(fit, risk.table = TRUE, conf.int = TRUE, pval = TRUE,
           censor = FALSE) # fun = "event"
```



Hazard Function / Taxa de Falha ou de Risco

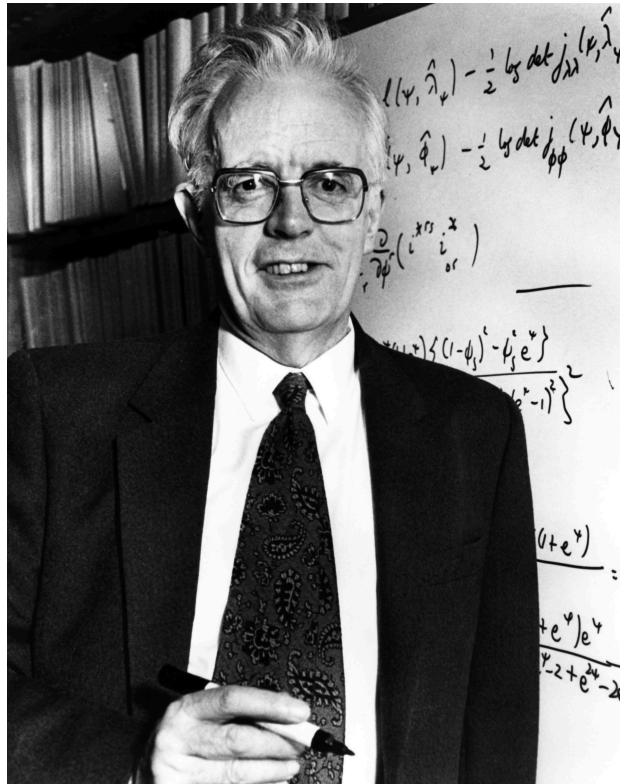
A seguir é apresentada a função de risco. Tomando um Δt bem pequeno, essa função representa a taxa de falha instantânea no determinado instante condicional à sobrevivência até esse tempo t .

$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{P(t < T \leq t + \Delta t | T > t)}{\Delta t}$$

Essa função se relaciona com a função densidade e de sobrevivência da seguinte forma:

$$\begin{aligned} h(t) &= \lim_{\Delta t \rightarrow 0} \frac{P(t < T \leq t + \Delta t | T > t)}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{P(t < T \leq t + \Delta t) / \Delta t}{P(T > t)} \\ &= \frac{f(t)}{S(t)} \end{aligned}$$

Modelos de Riscos Proporcionais de Cox



Sir David Cox

Vamos considerar um caso simples em que temos uma preditora indicadora de grupo. Ainda, considere que a função de risco é dada por:

$$h(t|x_1) = h_0(t) \exp\{\beta_1 x_1\}.$$

Assim, se tomarmos a razão de risco entre os grupos, teremos:

$$\frac{h_1(t)}{h_0(t)} = \frac{h_0(t) \exp\{\beta_1\}}{h_0(t) \exp\{0\}} = \exp\{\beta_1\} = K.$$

Isso indica que os riscos são proporcionais (constante K) independentemente do tempo.

Modelos de Riscos Proporcionais de Cox

De forma genérica, podemos escrever

$$h(t|x_i) = h_0(t) \exp\left\{ \sum_{j=1}^p \beta_j x_{ij} \right\}$$

e, consequentemente, teremos

$$\frac{h(t|x_i)}{h(t|x_k)} = \exp\left\{ \sum_{j=1}^p \beta_j (x_{ij} - x_{kj}) \right\}$$

Telco Churn

Vamos ajustar o modelo de Cox considerando as variáveis *gender*, *SeniorCitizen*, *Dependents* e *Contract*.

```
(fit <- coxph(Surv(tenure, Churn) ~ gender + SeniorCitizen + Dependents + Contract, dados))
```

```
## Call:  
## coxph(formula = Surv(tenure, Churn) ~ gender + SeniorCitizen +  
##         Dependents + Contract, data = dados)  
##  
##             coef  exp(coef)  se(coef)      z      p  
## genderMale     -0.05092   0.95035   0.04633  -1.099  0.272  
## SeniorCitizen  -0.05208   0.94925   0.05462  -0.954  0.340  
## DependentsYes -0.43374   0.64808   0.06229  -6.963 3.32e-12  
## ContractOne year -2.15024   0.11646   0.08396 -25.611 < 2e-16  
## ContractTwo year -4.15404   0.01570   0.15724 -26.419 < 2e-16  
##  
## Likelihood ratio test=2673 on 5 df, p=< 2.2e-16  
## n= 7043, number of events= 1869
```

Indice de concordância de Harrel / C-index

Considere o escore de risco dado por

$$\hat{\eta}_i = \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip} \quad \text{para } i = 1, \dots, n$$

Se $\hat{\eta}_i > \hat{\eta}_j$ (escores de risco), o modelo considera que j terá um tempo maior de sobrevivência do que i . Assim, o coeficiente de concordância é dado por:

$$C = \frac{\sum_{i,j:y_i>y_j} I(\eta_j > \eta_i) \delta_j}{\sum_{i,j:y_i>y_j} \delta_j}$$

Um indice de concordância de 0.792 pode indicar que o, dado um par de indivíduos aleatórios, o modelo tem uma acurácia estimada de 79.2% de indicar se uma observação apresentará o evento antes da outra.

Essa medida tem uma associação com a área sob a curva ROC. Para mais detalhes sobre estimativas dessa medida e outras avaliações preditivas no contexto de sobrevivência, consulte minha [dissertação de mestrado](#).

Indice de concordância de Harrel / C-index

```
summary(fit)

## Call:
## coxph(formula = Surv(tenure, Churn) ~ gender + SeniorCitizen +
##         Dependents + Contract, data = dados)
##
## n= 7043, number of events= 1869
##
##          coef exp(coef) se(coef)      z Pr(>|z|)
## genderMale -0.05092  0.95035  0.04633 -1.099   0.272
## SeniorCitizen -0.05208  0.94925  0.05462 -0.954   0.340
## DependentsYes -0.43374  0.64808  0.06229 -6.963 3.32e-12 ***
## ContractOne year -2.15024  0.11646  0.08396 -25.611 < 2e-16 ***
## ContractTwo year -4.15404  0.01570  0.15724 -26.419 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          exp(coef) exp(-coef) lower .95 upper .95
## genderMale      0.9504     1.052   0.86786   1.04068
## SeniorCitizen    0.9492     1.053   0.85289   1.05650
## DependentsYes    0.6481     1.543   0.57360   0.73223
## ContractOne year   0.1165     8.587   0.09879   0.13729
## ContractTwo year   0.0157    63.691   0.01154   0.02137
##
## Concordance= 0.792  (se = 0.004 )
```

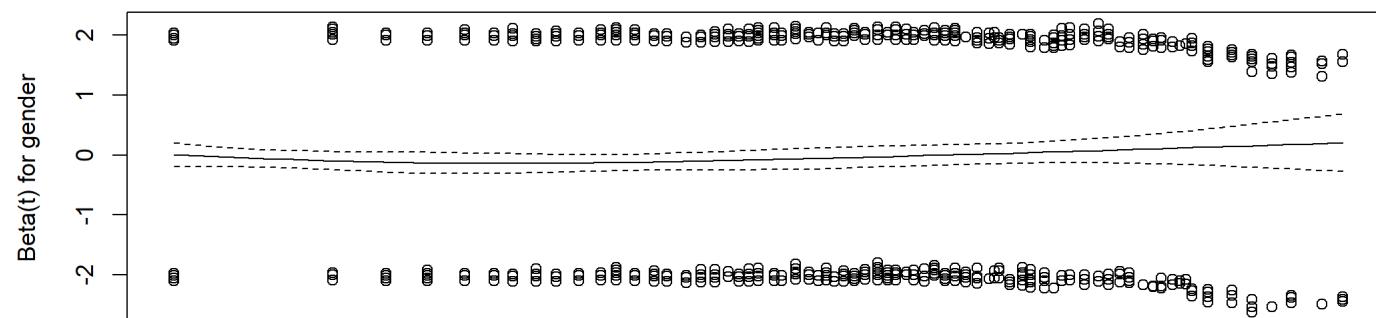
Telco Churn

Comentamos que o modelo de Cox é um modelo de riscos proporcionais, mas não verificamos se essa suposição é razoável. Para isso, podemos utilizar:

```
(teste_prop <- cox.zph(fit))
```

```
##                                chisq df   p
## gender                  0.2600  1  0.610
## SeniorCitizen            0.0162  1  0.899
## Dependents               4.0196  1  0.045
## Contract                109.9295 2 <2e-16
## GLOBAL                  115.0487 5 <2e-16
```

```
plot(teste.prop[1])
```



Telco Churn

Uma possível solução para a proporcionalidade dos riscos é estratificar o modelo de acordo com uma determinada variável. Nesse caso, *contract*.

```
fit <- coxph(Surv(tenure, Churn) ~ gender + SeniorCitizen + Dependents + strata(Contract),  
               data = dados)
```

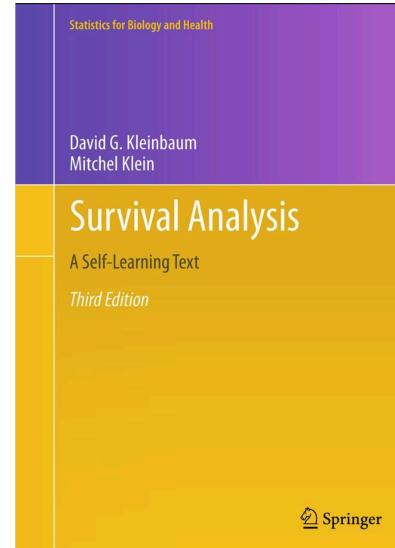
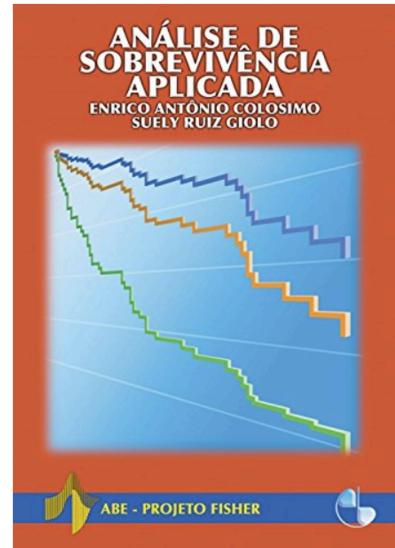
```
(teste_prop <- cox.zph(fit))
```

```
##          chisq df      p  
## gender     0.983  1 0.322  
## SeniorCitizen 3.033  1 0.082  
## Dependents   0.832  1 0.362  
## GLOBAL       5.489  3 0.139
```

Análise de Sobrevivência

- Modelos paramétricos
- Modelo de Cox com covariáveis dependentes do tempo
- Modelos de riscos competitivos
- Aplicações de modelos como Floresta Aleatória e Boosting

Referências bibliográficas



Sobrevivência em Floresta Aleatória

Podemos utilizar alguns modelos vistos no curso no contexto de sobrevivência. Por exemplo, para Floresta Aleatória

```
library(ranger)

fit <- ranger(Surv(tenure, Churn) ~ gender + SeniorCitizen + Dependents + Contract, dados)

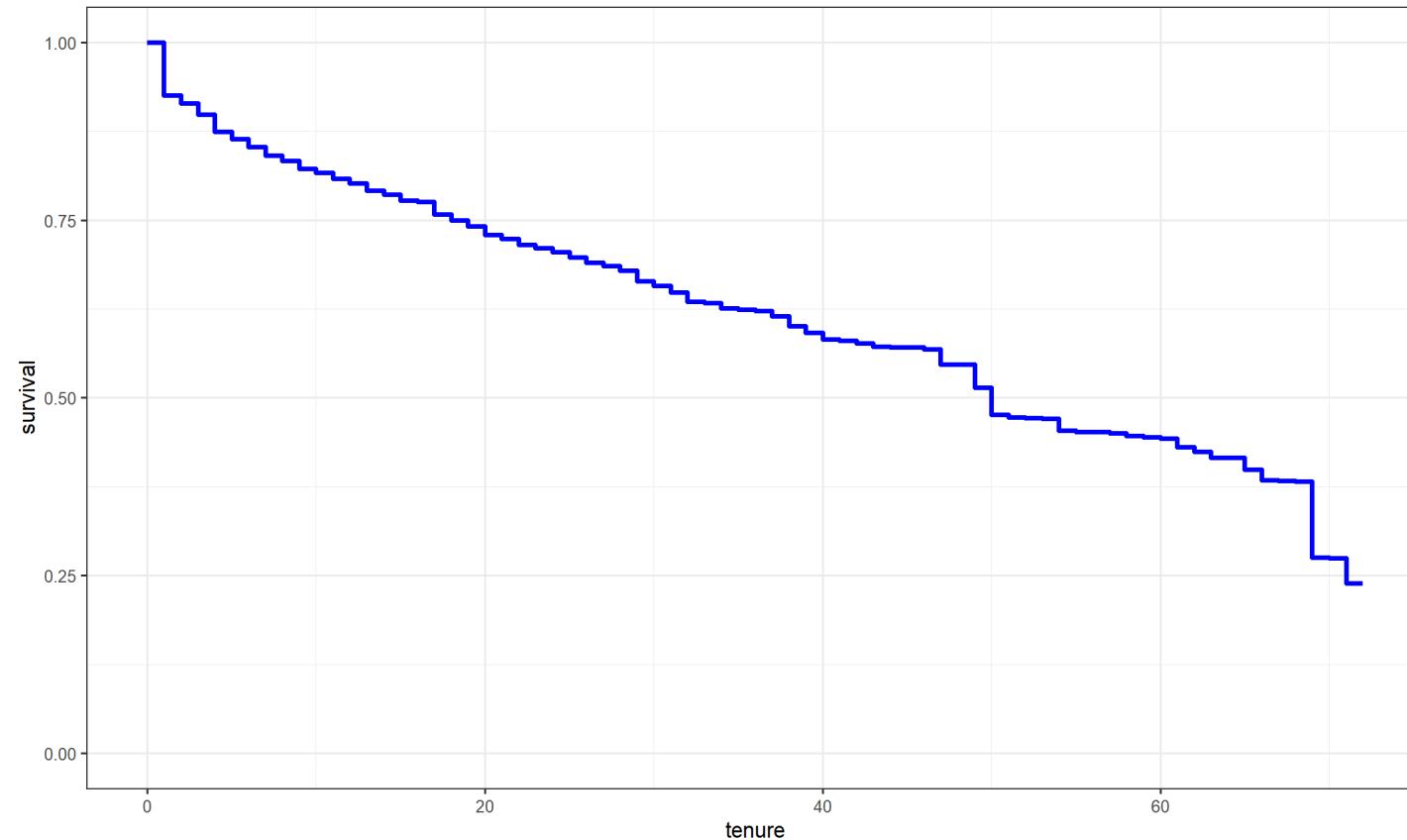
predito <- predict(fit, dados[7,])

tibble(time = predito$unique.death.times,
       surv = predito$survival) %>%
  ggplot(aes(time, surv)) +
  geom_step() +
  labs(x = "tenure", y = "survival") +
  scale_y_continuous(limits = c(0, 1)) +
  theme_bw()
```

Árvores de sobrevivência utilizam como critério de divisão a maximização das diferenças da curva de sobrevivência.

Sobrevivência em Floresta Aleatória

Podemos utilizar alguns modelos vistos no curso no contexto de sobrevivência. Por exemplo, para Floresta Aleatória



Sobrevivência em GBM

Além de Floresta Aleatória, também podemos utilizar boosting nesse contexto com o pacote `gbm`.

```
library(gbm)

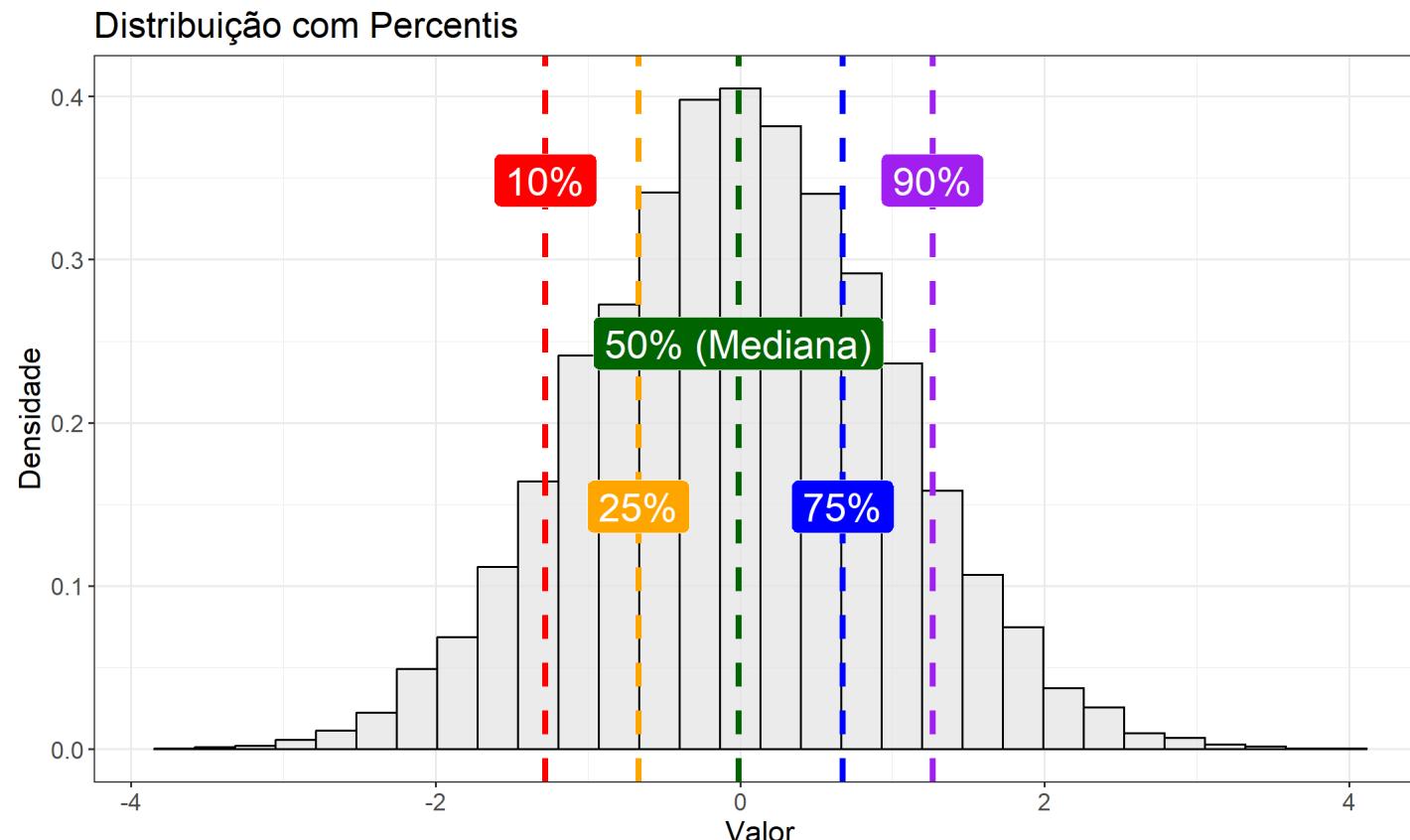
(fit <- gbm(Surv(tenure, Churn) ~ gender + SeniorCitizen + Dependents +
             Contract, distribution = "coxph", dados) )

## gbm(formula = Surv(tenure, Churn) ~ gender + SeniorCitizen +
##       Dependents + Contract, distribution = "coxph", data = dados)
## A gradient boosted model with coxph loss function.
## 100 iterations were performed.
## There were 4 predictors of which 4 had non-zero influence.
```

Régressão Quantílica

Regressão Quantílica

A Regressão Quantílica é uma técnica que permite modelar a relação entre variáveis explicativas e as diferentes quantis (percentis) de uma variável resposta. Ao contrário da regressão linear tradicional, que estima a média condicional da variável dependente, a regressão quantílica foca em diferentes pontos da distribuição condicional (por exemplo, o 25º, 50º, ou 90º percentil).



Regressão Quantílica

```
library(ISLR)
library(quantreg)
library(ranger)
library(gbm)

reg_quant <- rq(Balance ~ Income + Student, tau = 0.25,
                 data = Credit)

rf <- ranger(Balance ~ Income + Student, quantreg = TRUE,
              data = Credit)

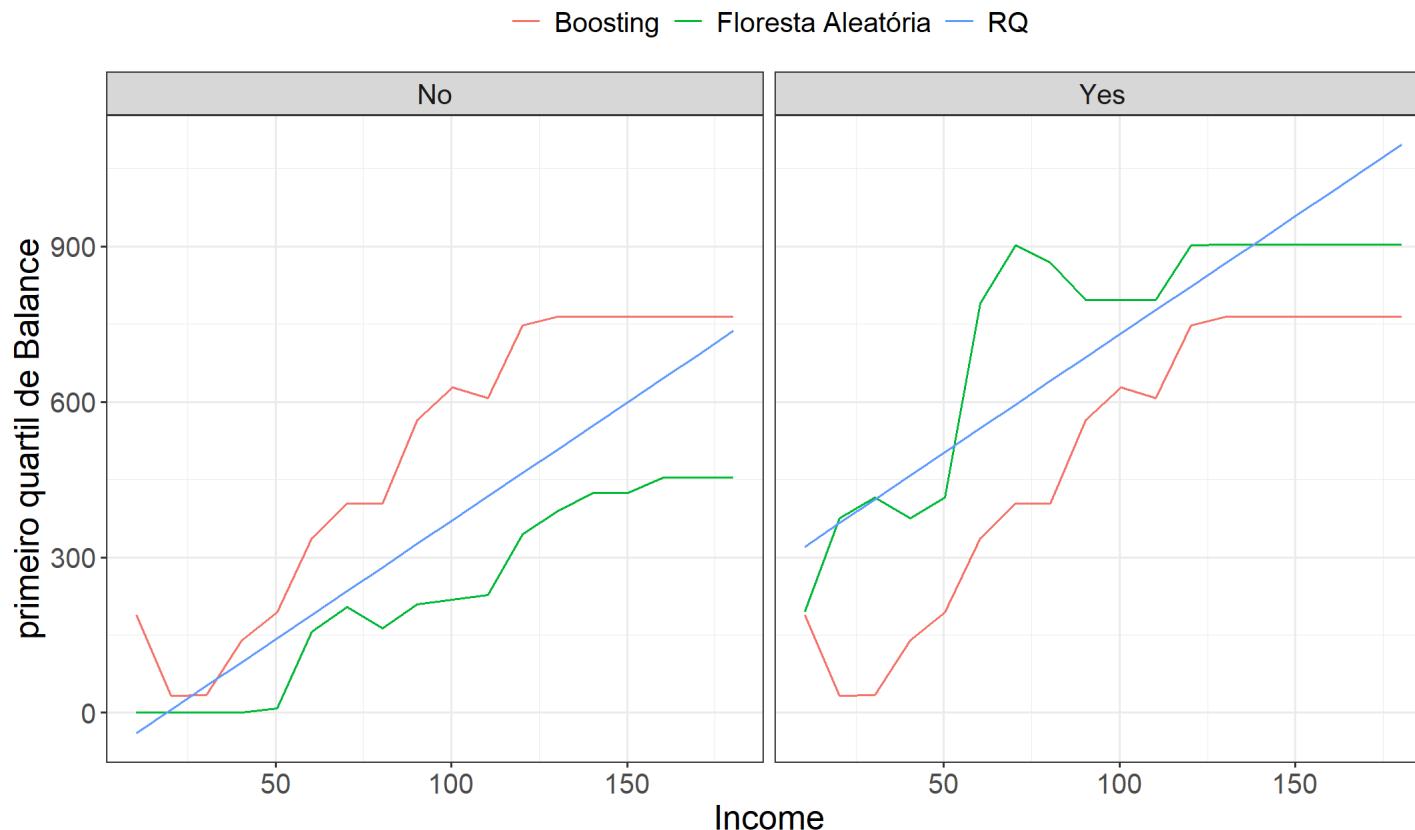
bst <- gbm(Balance ~ Income + Student,
            distribution = list(name = "quantile", alpha = 0.25),
            data = Credit)
```

Regressão Quantílica

```
df <- crossing(Income = seq(min(ISLR::Credit$Income),
                           max(ISLR::Credit$Income), 10),
                 Student = c("No", "Yes"))

df %>%
  mutate(modelo = "RQ") %>%
  bind_cols(previsao = predict(reg_quant, df)) %>%
  bind_rows(df %>%
    mutate(modelo = "Floresta Aleatória") %>%
    bind_cols(previsao = predict(rf, df, type = "quantiles",
                                 quantiles = 0.25)$predictions[,1])) %>%
  bind_rows(df %>%
    mutate(modelo = "Boosting") %>%
    bind_cols(previsao = predict(bst, df))) %>%
  ggplot(aes(Income, previsao, group = modelo, color = modelo)) +
  geom_line() +
  facet_grid(~ Student) +
  labs(y = "primeiro quartil de Balance", group = NULL, color = NULL) +
  theme_bw() +
  theme(legend.position = "top")
```

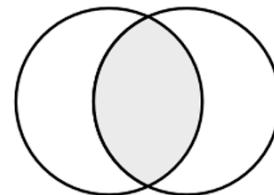
Regressão Quantílica



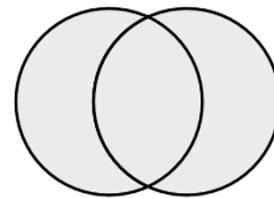
Join's e mapas

Join's

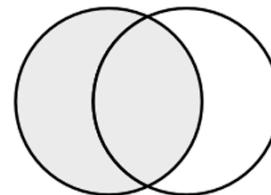
Para agrupar diferentes bancos de dados no R, podemos utilizar as funções da família `join`.



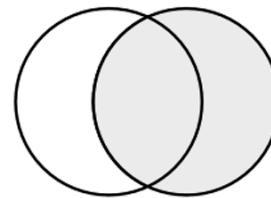
`inner_join(x, y)`



`full_join(x, y)`



`left_join(x, y)`



`right_join(x, y)`

SEADE

Como exemplo, vamos utilizar alguns dados do SEADE sobre IDH, renda e saúde. Inspecione os arquivos e verifique as informações apresentadas.

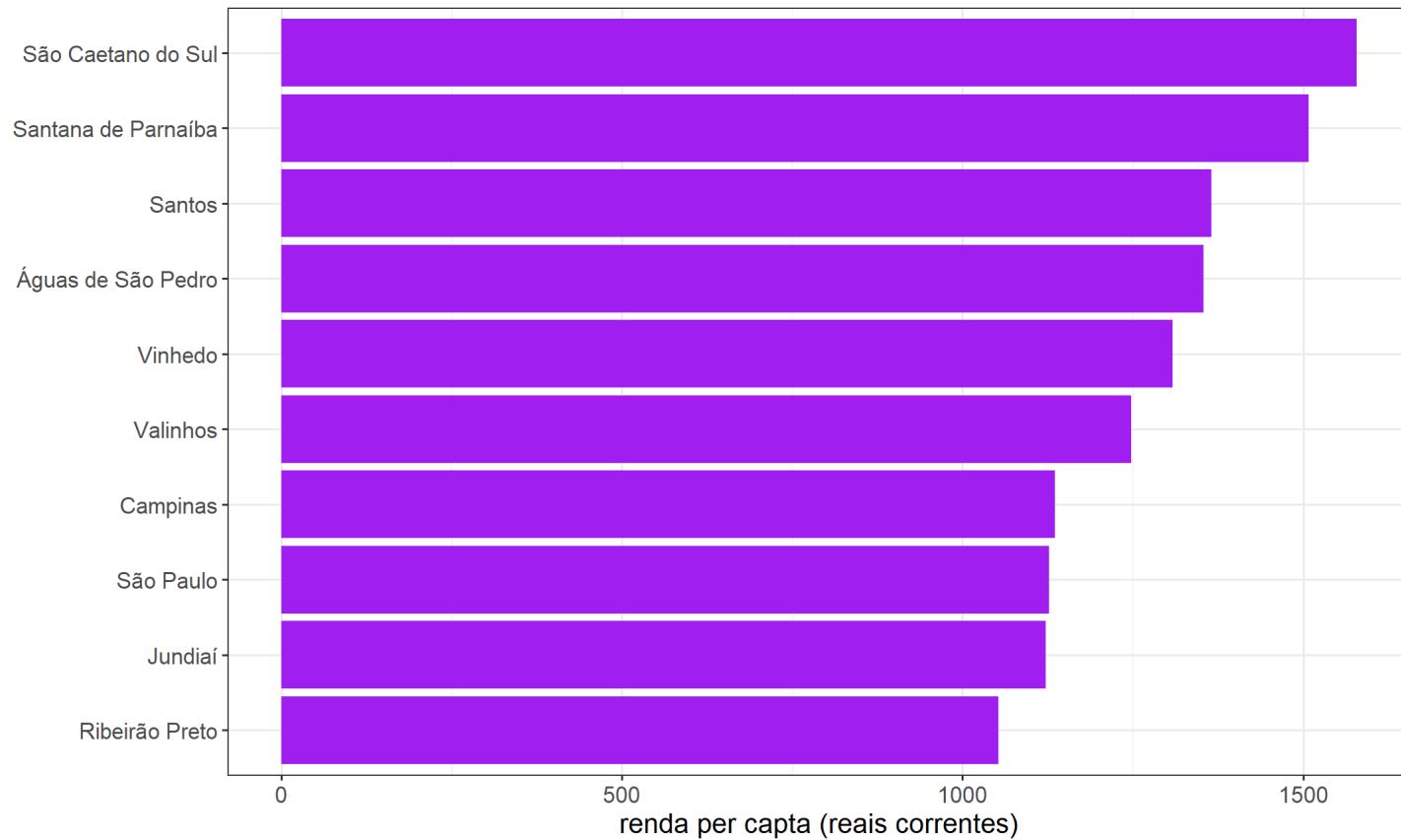
```
idh <- read.csv2("dados/idh.csv") %>%  
  janitor::clean_names()  
  
skimr::skim(idh)  
  
renda <- read.csv2("dados/renda.csv") %>%  
  janitor::clean_names()  
  
saude <- read.csv2("dados/saude_mun.csv") %>%  
  janitor::clean_names()
```

Agrupe esses bancos de dados considerando apenas dados de 2010 e identifique os 10 municípios com maior renda per capita.

```
dados <- idh %>%
  filter(periodos == 2010) %>%
  left_join(renda %>%
              filter(periodos == 2010), by = c("cod_ibge", "periodos", "localidades")) %>%
  left_join(saude %>%
              filter(periodos == 2010), by = c("cod_ibge", "periodos", "localidades"))

dados %>%
  slice_max(renda_per_capita_censo_demografico_em_reais_correntes, n = 10) %>%
  ggplot(aes(renda_per_capita_censo_demografico_em_reais_correntes,
             reorder(localidades, renda_per_capita_censo_demografico_em_reais_correntes))) +
  geom_col(fill = "purple") +
  labs(x = "renda per capita (reais correntes)", y = NULL) +
  theme_bw() +
  theme(text = element_text(size = 14))
```

Agrupe esses bancos de dados considerando apenas dados de 2010 e identifique os 10 municípios com maior renda per capita.



A seguir verificaremos como trabalhar com mapas no R. O pacote `geobr` conta com diversos mapas e também tem uma versão para Python! Nessa aplicação faremos um mapa da renda per capita dos municípios de São Paulo.

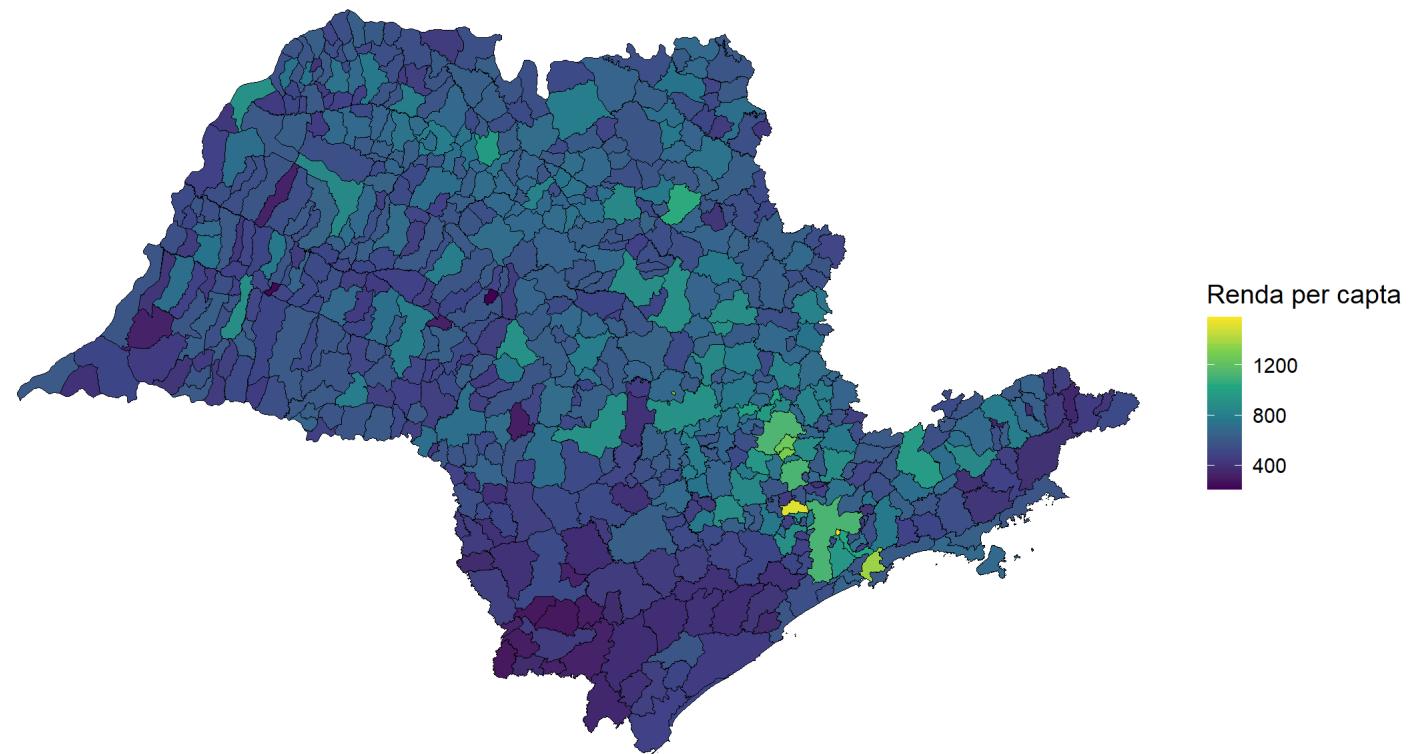
```
(mapa <- geobr::read_municipality(code_muni = "SP", showProgress = FALSE))

## Simple feature collection with 645 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: -53.10986 ymin: -25.31232 xmax: -44.16137 ymax: -19.77966
## Geodetic CRS: SIRGAS 2000
## First 10 features:
##   code_muni      name_muni code_state abbrev_state
## 1 3500105        Adamantina      35          SP
## 2 3500204         Adolfo       35          SP
## 3 3500303        Aguai       35          SP
## 4 3500402 Águas Da Prata     35          SP
## 5 3500501 Águas De Lindóia    35          SP
## 6 3500550 Águas De Santa Bárbara 35          SP
## 7 3500600 Águas De São Pedro   35          SP
## 8 3500709        Agudos       35          SP
## 9 3500758        Alambari     35          SP
## 10 3500808 Alfredo Marcondes  35          SP
## 
##   geom
## 1 MULTIPOLYGON (((-51.09093 -...
```

A seguir verificaremos como trabalhar com mapas no R. O pacote `geobr` conta com diversos mapas e também tem uma versão para Python! Nessa aplicação faremos um mapa da renda per capita dos municípios de São Paulo.

```
(fig <- mapa %>%
  left_join(dados, by = c("code_muni" = "cod_ibge")) %>%
  ggplot(aes(fill = renda_per_capita_censo_demografico_em_reais_correntes)) +
  geom_sf(color = "black", lwd = .3) +
  scale_fill_viridis_c() +
  labs(fill = "Renda per capita") +
  theme_void() )

# plotly::ggplotly(fig)
```



Obrigado!

 tiagoms.com

 [tiagomendonca](#)

 tiagoms1@insper.edu.br