Universidade de Brasília

DEPARTAMENTO DE ESTATÍSTICA

30 maio 2024

Entrega 4

Prof. Dr. George von Borries

Aluno: Bruno Gondim Toledo

Matrícula: 15/0167636

Tópicos 2

1º/2024

11. Pesquisar funções disponíveis em pacotes R para classificação utilizando a função logística.

Apresentar um pequeno exemplo do uso das funções. Destacar vantagens e desvantagens em relação aos pacotes de Modelos Lineares Generalizados apresentados em aula.

Exemplos de pacotes para classificação no **R**: *caret*, *class*, *mlpack*.

Exemplos usando *stats::glm*

i) Reproduzindo o exemplo de Smolski

Os dados são originados do livro de Hosmer e Lemeschow (2000), tratando-se de uma amostra com 100 pessoas. A variável dependente é a ocorrência ou não (1 ou 0) de doença coronária cardíaca (CHD), associando-se com a idade (AGE) dos indivíduos.

```
head(chd)
```

```
## # A tibble: 6 x 3
##     AGE  AGRP CHD
##   <dbl> <dbl> <fct>
## 1    20     1 0
## 2    23     1 0
## 3    24     1 0
## 4    25     1 0
## 5    25     1 1
## 6    26     1 0
```

```
knitr::kable(summary(chd))
```

AGE	AGRP	CHD
Min. :20.00	Min. :1.00	0:57
1st Qu.:34.75	1st Qu.:2.75	1:43
Median :44.00	Median :4.00	NA
Mean :44.38	Mean :4.48	NA
3rd Qu.:55.00	3rd Qu.:7.00	NA
Max. :69.00	Max. :8.00	NA

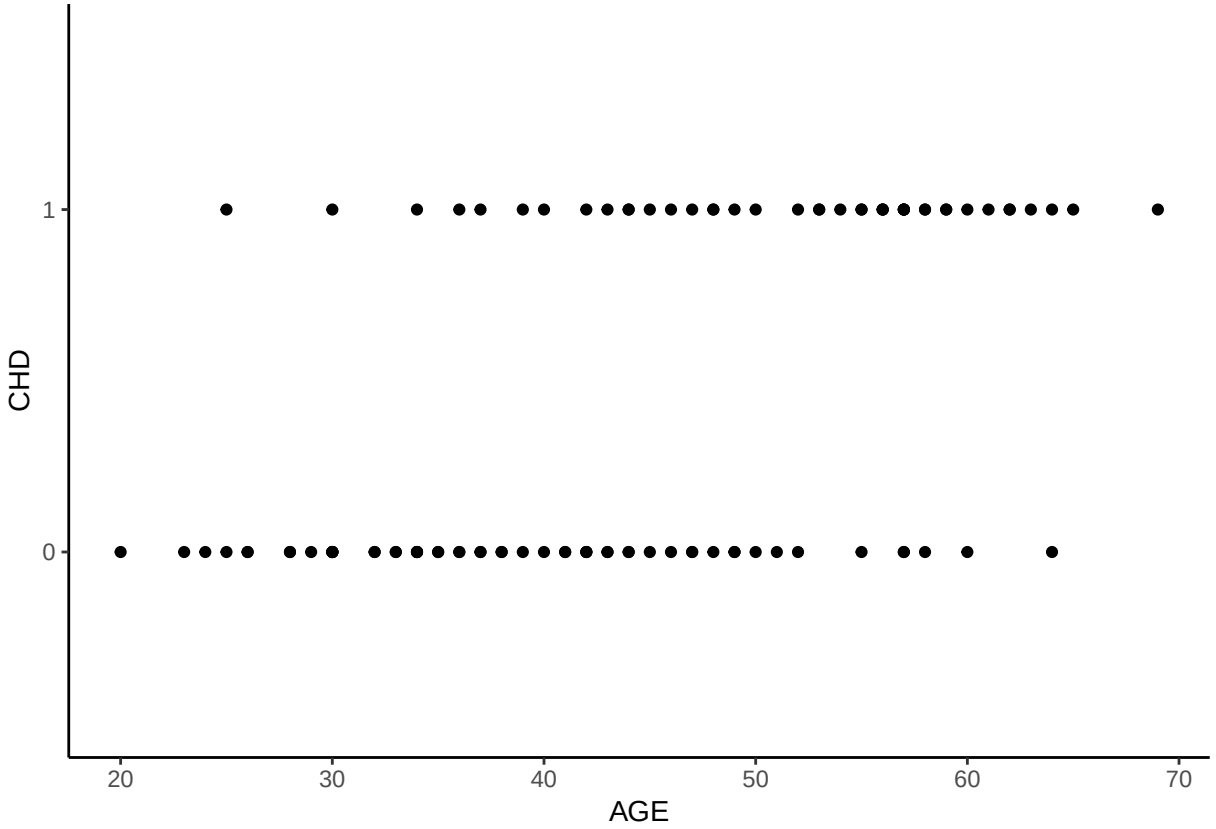
Pela tabela, vemos que existem 57 indivíduos sem a doença e 43 indivíduos com a doença na amostra.

Se olharmos para a idade média de cada grupo

```
kable(chd |>
  mutate(Grupo = ifelse(CHD == 1, "Apresenta doenca", "Nao apresenta doenca")) |>
  group_by(Grupo) |>
  summarise(`Idade média do grupo` = round(mean(AGE))))
```

Grupo	Idade média do grupo
Apresenta doenca	51
Nao apresenta doenca	39

```r
ggplot(chd, aes(x=AGE, y=CHD)) +
  geom_point() + theme_classic()
```



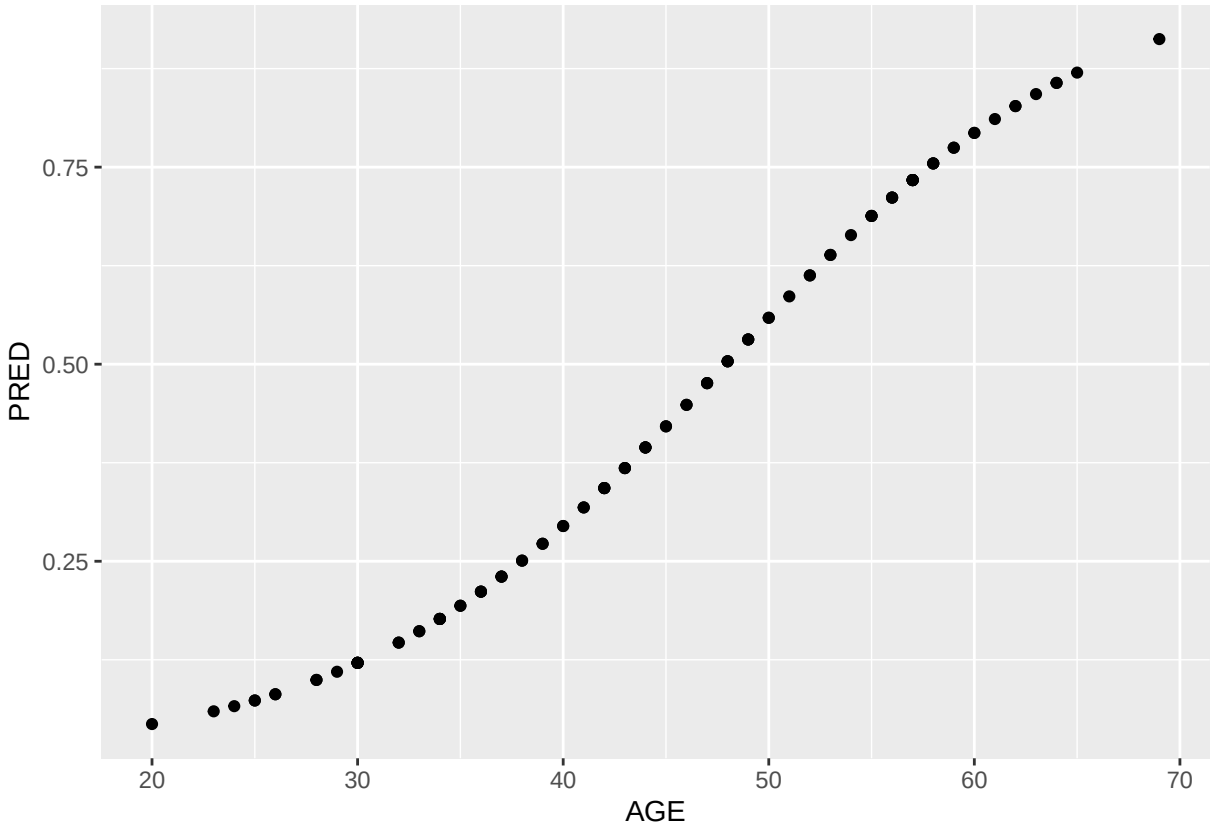
Podemos tentar ajustar um modelo de regressão logística para prever a ocorrência de doença coronária cardíaca a partir da idade.

```r
m1=glm(CHD~AGE, family = binomial(link="logit"), data = chd)
summary(m1)
```

```
##
## Call:
## glm(formula = CHD ~ AGE, family = binomial(link = "logit"), data = chd)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.30945    1.13365  -4.683 2.82e-06 ***
## AGE          0.11092    0.02406   4.610 4.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 136.66  on 99  degrees of freedom
## Residual deviance: 107.35  on 98  degrees of freedom
## AIC: 111.35
##
## Number of Fisher Scoring iterations: 4
```

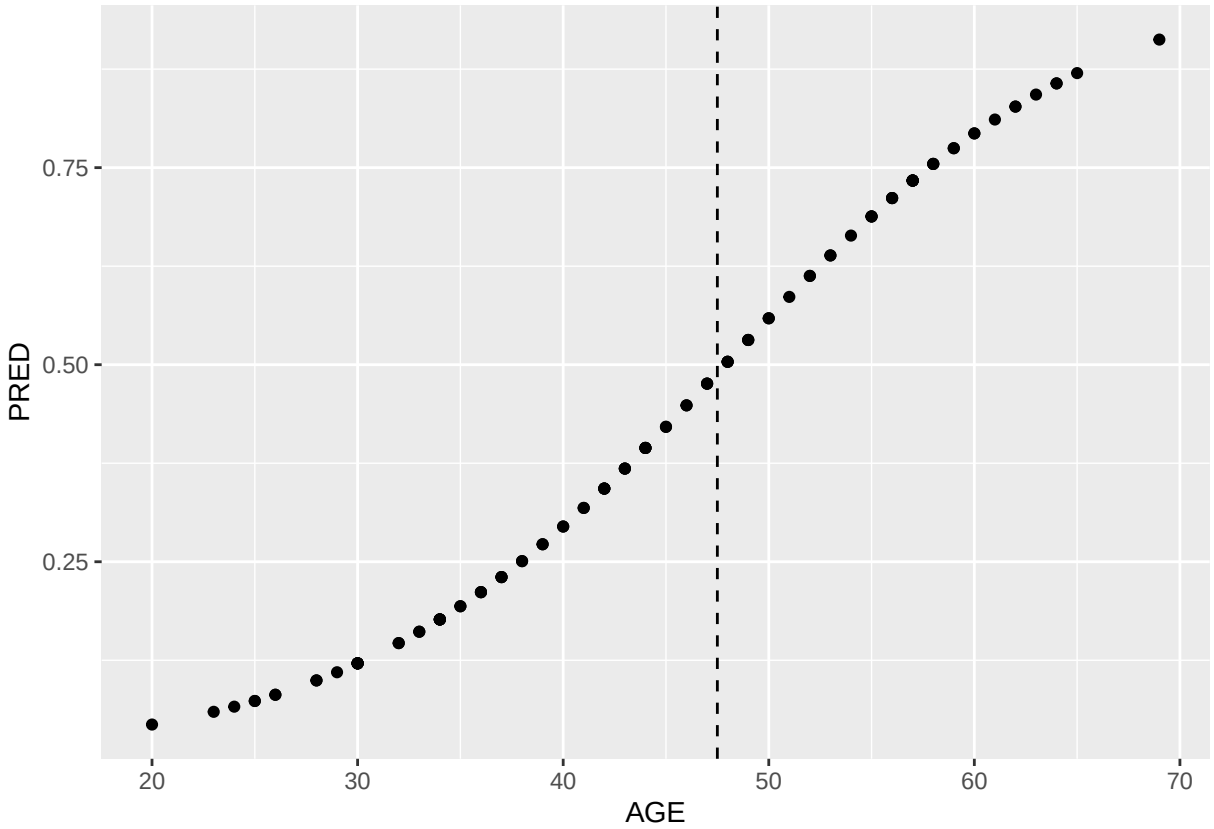
E então plotar a probabilidade de ocorrência da doença em função da idade.

```
IDADE<-chd[,1]
chd$PRED=predict(m1, newdata=IDADE, type="response")
ggplot(chd, aes(x=AGE, y=PRED)) +
  geom_point()
```



Desta, podemos estabelecer como regra de classificação a probabilidade de 50% de ocorrência da doença..

```
ggplot(chd, aes(x=AGE, y=PRED)) +
  geom_point()+
  geom_vline(xintercept = 47.5, linetype = 2)
```



E este seria um exemplo (bem) simplificado de classificação utilizando regressão logística

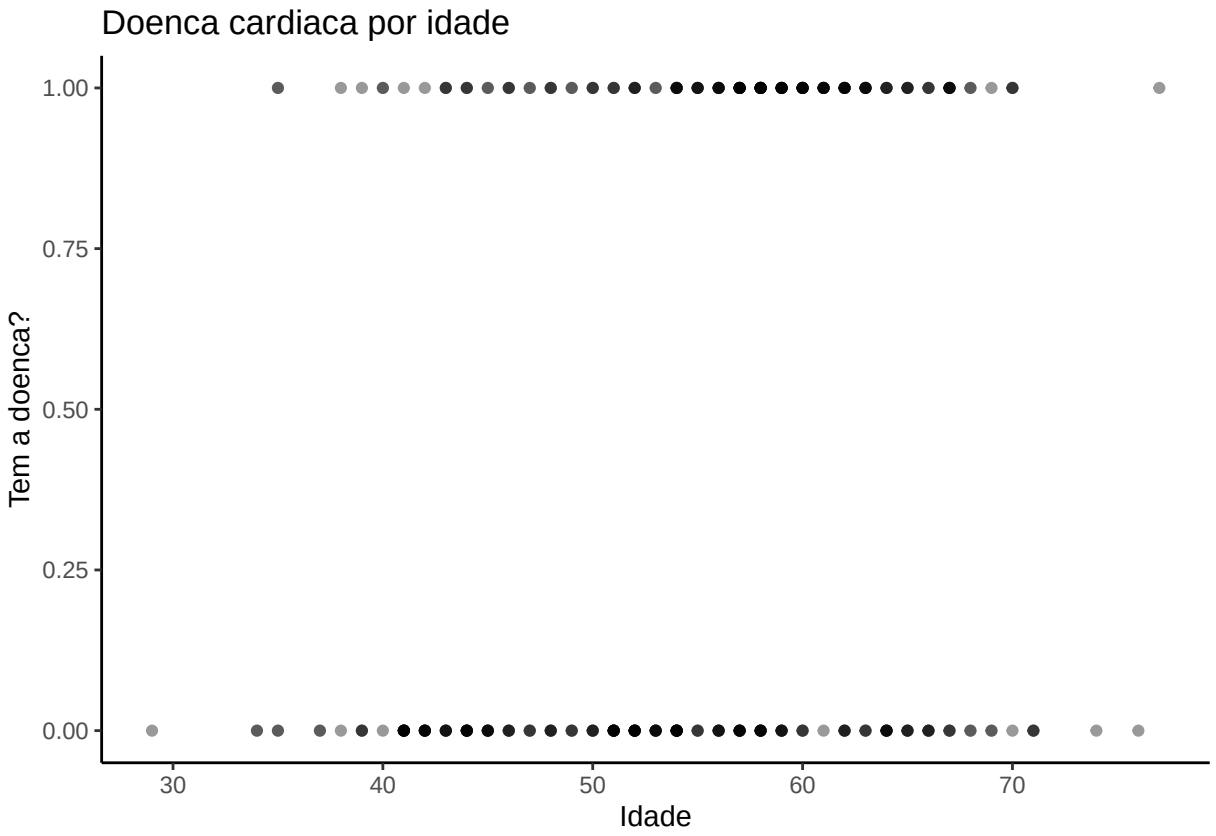
ii) Explorando a regressão logística com dados *heart_disease* do pacote *funModeling*

```
fit_glm=glm(has_heart_disease ~ age, data=heart_disease,
            family = binomial(link = 'logit'))
kable(tidy(fit_glm, conf.int = TRUE))
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-3.0059138	0.7591288	-3.959689	0.0000750	-4.5322435	-1.5485882
age	0.0519862	0.0136688	3.803282	0.0001428	0.0256978	0.0794146

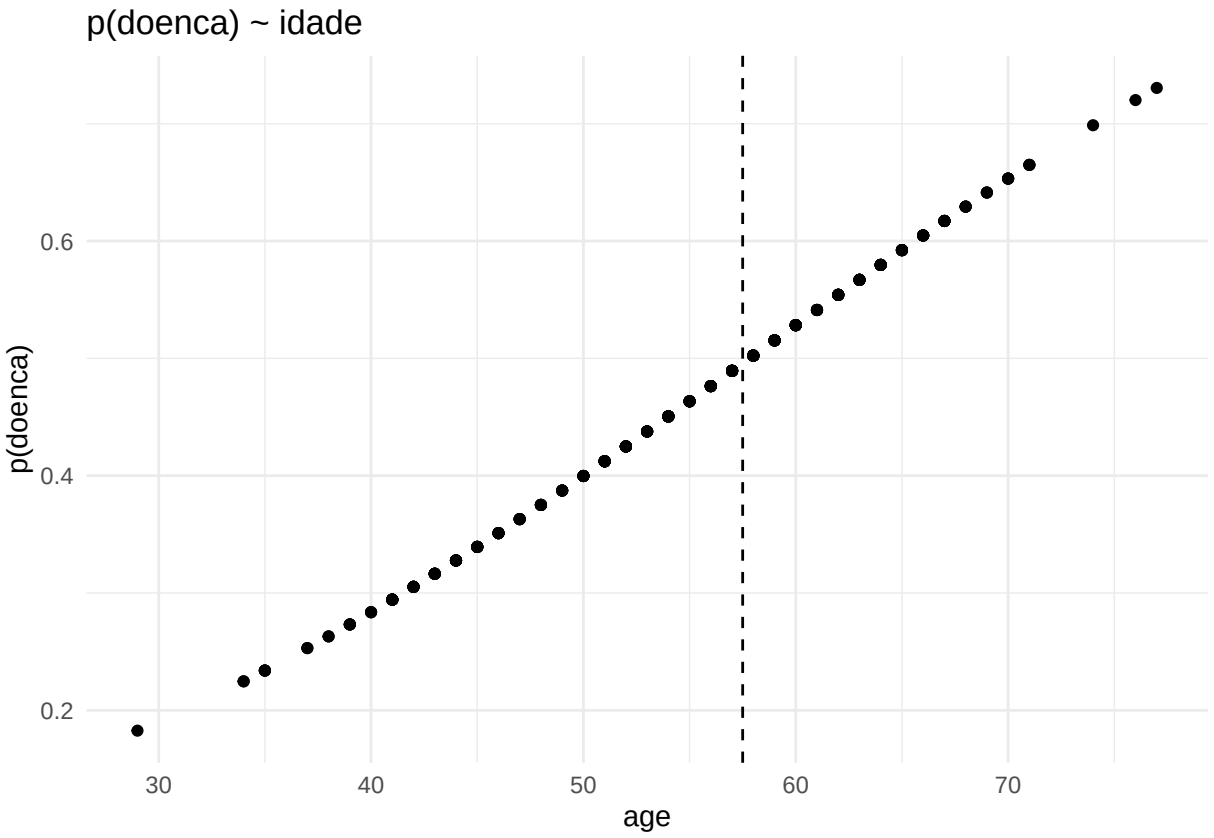
Podemos observar na tabela acima as métricas do modelo.

```
heart_disease %>%
  mutate(
    has_heart_disease_numeric = case_when(
      has_heart_disease == "yes" ~ 1,
      has_heart_disease == "no" ~ 0 ))%>%
  ggplot(aes(y = has_heart_disease_numeric,
             x = age)) +
  geom_point(alpha = .4) +
  labs(y = "Tem a doenca?",
       x = "Idade",
       title = "Doenca cardiaca por idade") +
  theme_classic()
```



Podemos utilizar a mesma ideia do exemplo anterior para plotar a probabilidade de ocorrência da doença em função da idade.

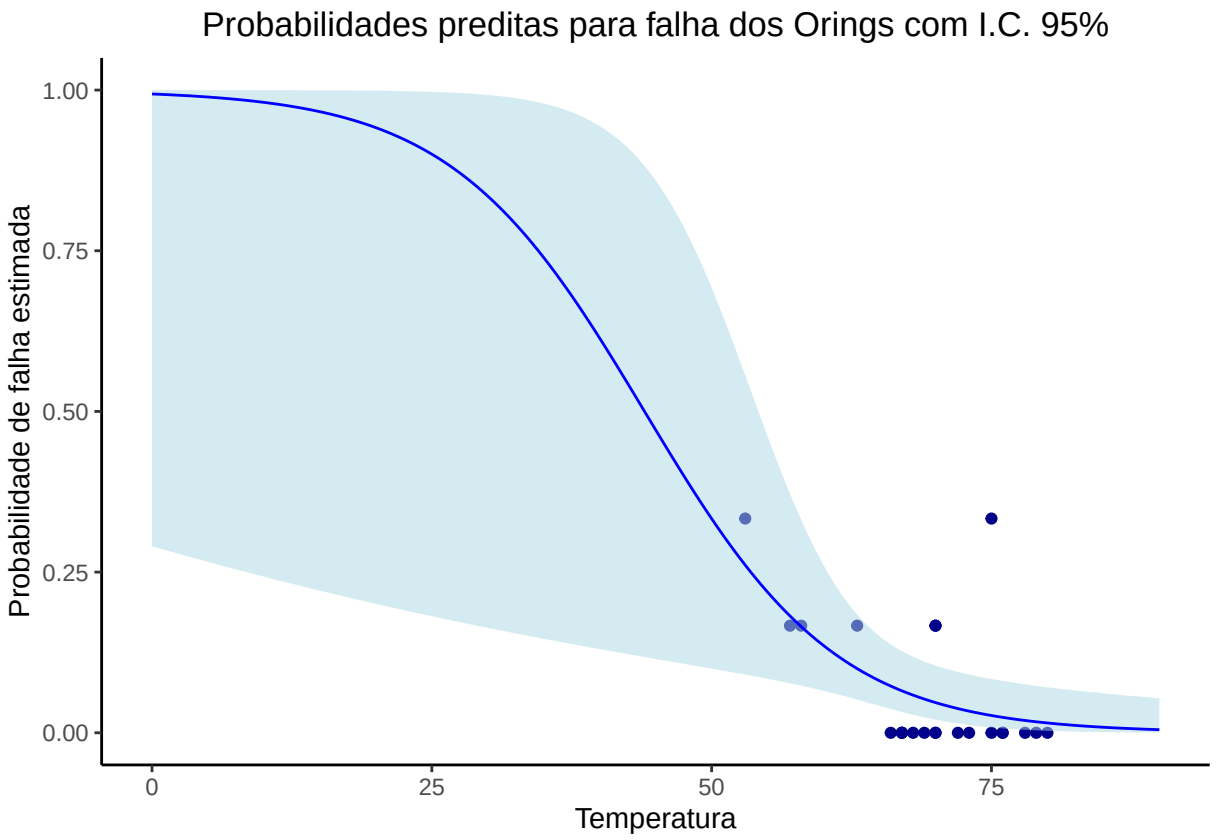
```
heart_disease$lr_log_odds = predict(fit_glm)
heart_disease$logistic_predictions = predict(fit_glm, type = "response")
heart_disease %>%
  ggplot(aes(x = age,
             y = logistic_predictions)) +
  geom_point() +
  labs(y = "p(doenca)",
       title = "p(doenca) ~ idade") +
  theme_minimal()+
  geom_vline(xintercept = 57.5, linetype = 2)
```



iii) O'rings - Não é exatamente classificação, mas é um exemplo interessante...

```
fails <- c(2, 0, 0, 1, 0, 0, 1, 0, 0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)
temp <- c(53, 66, 68, 70, 75, 78, 57, 67, 69, 70, 75,
          79, 58, 67, 70, 72, 76, 80, 63, 67, 70, 73, 76)

challenger <- tibble::tibble(fails, temp)

orings = 6
challenger <- challenger %>%
  dplyr::mutate(resp = fails/orings)

model_fit <- glm(resp ~ temp,
                 data = challenger,
                 weights = rep(6, nrow(challenger)),
                 family=binomial(link="logit"))

x_limits <- challenger %>%
  dplyr::summarise(min = 0, max = max(temp)+10)

x <- seq(x_limits[[1]], x_limits[[2]], by=0.5)

temp.data = data.frame(temp = x)

predicted.data <- as.data.frame(
  predict(model_fit,
          newdata = temp.data,
          type="link", se=TRUE))
```

```r
new.data <- cbind(temp.data, predicted.data)

std <- qnorm(0.95 / 2 + 0.5)
new.data$ymin <- model_fit$family$linkinv(new.data$fit - std * new.data$se)
new.data$ymax <- model_fit$family$linkinv(new.data$fit + std * new.data$se)
new.data$fit <- model_fit$family$linkinv(new.data$fit)

ggplot(challenger, aes(x=temp, y=resp))+
  geom_point(colour = "darkblue")+
  geom_ribbon(data=new.data,
              aes(y=fit, ymin=ymin, ymax=ymax),
              alpha = 0.5,
              fill = 'lightblue')+
  geom_line(data=new.data, aes(y=fit), colour = "blue") +
  labs(x="Temperatura", y="Probabilidade de falha estimada")+
  ggtitle("Probabilidades preditas para falha dos Orings com I.C. 95%")+
  theme_classic()+
  theme(panel.border = element_blank(), plot.title = element_text(hjust=0.5))
```

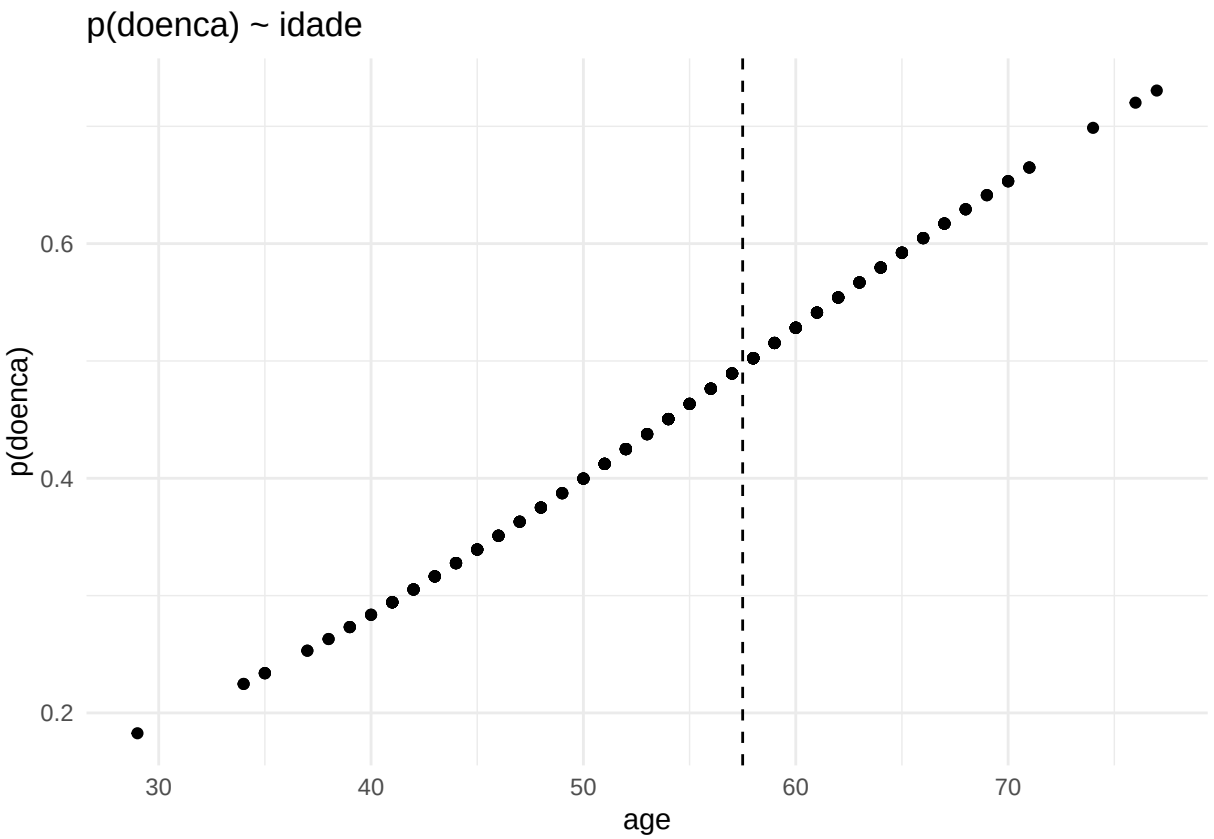


Exemplo ii), mas utilizando o *caret*

```r
fit_glm_caret <- train(has_heart_disease ~ age,
                       data = heart_disease,
                       method = "glm",
                       family = binomial(link = 'logit'))
kable(tidy(fit_glm_caret$finalModel, conf.int = TRUE))
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-3.0059138	0.7591288	-3.959689	0.0000750	-4.5322435	-1.5485882
age	0.0519862	0.0136688	3.803282	0.0001428	0.0256978	0.0794146

```
heart_disease$lr_log_odds_caret = predict(fit_glm_caret, newdata = heart_disease)
heart_disease$logistic_predictions_caret = predict(fit_glm_caret,
                                                   newdata = heart_disease,
                                                   type = "prob")[,2]
heart_disease %>%
  ggplot(aes(x = age,
             y = logistic_predictions_caret)) +
  geom_point() +
  labs(y = "p(doenca)",
       title = "p(doenca) ~ idade") +
  theme_minimal()+
  geom_vline(xintercept = 57.5, linetype = 2)
```



Para estes casos simples, utilizar o *caret* não parece fazer diferença. Entretando, o pacote, através da função *train()* permite a utilização de diversas técnicas de validação cruzada, além de permitir a utilização de diversas métricas de avaliação de modelos.

Utilizando o pacote *mlpack* para modelar o exemplo ii)

```
heart_disease_mlpack = heart_disease %>%
  mutate(has_heart_disease = ifelse(has_heart_disease == "yes", 2, 1))

index <- sample(1:nrow(heart_disease_mlpack), 0.7*nrow(heart_disease_mlpack))
training_data <- matrix(heart_disease_mlpack[index,]$age)
```

```
training_label <- matrix(heart_disease_mlpack[index,]$has_heart_disease)

test_data <- matrix(heart_disease_mlpack[-index,]$age)
test_labels <- matrix(heart_disease_mlpack[-index,]$has_heart_disease)

output <- logistic_regression(training=training_data, labels=training_label,
                              lambda=0.1)
lr_model <- output$output_model

output <- logistic_regression(input_model=lr_model, test=test_data)
predictions <- output$predictions

kable(head(data.frame(predictions,test_labels)))
```

predictions	test_labels
2	1
2	2
1	1
2	1
2	1
1	1

```
kable(tail(data.frame(predictions,test_labels)))
```

	predictions	test_labels
86	2	2
87	1	1
88	2	2
89	2	2
90	2	2
91	1	1

```
kable(table(test_labels == predictions),
      col.names = c("Acertou?","Quantidade"))
```

Acertou?	Quantidade
FALSE	35
TRUE	56

O pacote *mlpack* é muito mais "enjoado" que os demais, requerendo uma série de etapas e procedimentos específicas. Entretando, o pacote tem a grande vantagem de ser compilado em FORTRAN, o que aumenta vertiginosamente a sua velocidade. Para exemplos simples como este, não há diferença. Entretando, para grandes conjuntos de dados e modelos mais complexos, este pode ser um grande diferencial na hora de performar uma modelagem.

Considerações

Em suma, o modelo logístico pode ser uma abordagem útil e interessante para classificação. Conforme discutido em sala, há suas desvantagens, como o fato de que este não funciona propriamente quando há separação perfeita entre os dados. Devemos ainda nos atentar por vezes no *threshold* de classificação,

sendo por vezes arriscado utilizar 0,5 como padrão. O pacote *caret* é uma excelente ferramenta para modelagem, permitindo a utilização de diversas técnicas de validação cruzada e métricas de avaliação de modelos. O pacote *mlpack* é muito mais rápido, mas requer um pouco mais de trabalho para ser utilizado. Enquanto que o *glm()* do *stats* é mais do que suficiente para a maioria das situações mais simples.