UNIVERSIDADE DE SÃO PAULO

SME 0878 Mineração Estatística de Dados

Desafio 1 - Coverage

Pedro Vinícius Alves Silva - 10727865
alvespedro769@usp.br

```
#Bibliotecas requeridas
# require(ggplot2)
# require(GGally)
# require(dplyr)
# require(tibble)
# require(RColorBrewer)
# require(boot)
# require(ggsci)
# require(viridis)
# require(patchwork)
# require(hnp)
# require(Epi)
# require(car)
# require(R2jags)
# require(coda)
# require(formattable)
# require(randomForest)
# require(lattice)
```

Amostragem da base e separação treino/teste

Essa seção faz a amostragem dos conjunto de dados e salva os arquivos resultantes em baseprincipal.csv, basetreino.csv e basetest.csv. Os códigos estão disponíveis no arquivo .Rmd.

Análise Exploratória

Apresentação dos Dados

Uma das modalidades de seguro de veículos é conhecida como cobertura completa e pode ser acionada para cubrir os custos de danos ao automóvel por qualquer tipo de acidente, colisão, furto, vandalismos, enchentes ou impacto causado por um objeto inanimado.

Baseado em um lista de quatro características desejamos predizer a probabilidade de um indíviduo adquirir a cobertura completa e assim poder guiar o processo de escolha de potenciais clientes O conjunto de dados é formado por 2000 observações com as seguintes covariáveis:

- **AGE** - variável inteira, idade em anos dos indivíduos
- **MEN** - variável binária, sexo dos indivíduos (1- Masculino, 0 - Feminino)
- **URBAN** - variável binária, área na qual o indivíduo dirige (1 - Área Urbana, 0 - Área Rural)
- **PRIVATE** - variável binária, forma de utilização do veículo (1 - Privada, 0 - Comercial)
- **SENIORITY** - variável inteira, tempo de carta de motorista
- **y** - variável binária, se o indivíduo adquiriu cobertura completa? (1 - possui cobertura completa, 0- não possui cobertura completa)

```
data <- read.csv2('basetreino.csv', sep = ',')

data_test <- read.csv2('baseteste.csv', sep = ',')

data_test <- data_test %>%
  mutate_at(vars("URBAN", "MEN", "PRIVATE", "y"), as.factor)


head(data)
```

```
  y MEN URBAN PRIVATE AGE SENIORITY
1 1   1     0       1  71        27
2 0   1     0       1  51         8
3 1   0     1       1  35        14
4 0   0     0       1  47         8
5 0   1     0       1  77        14
6 1   1     0       1  21         5
```

Inconsistências

As colunas que podem apresentar algum tipo de inconsistência são Age e Seniority. Um indivíduo não poderia ter mais anos de carta de motorista do que de vida. Não temos nenhuma observação desse tipo no conjunto de treino.

```r
#inconsistência no conjunto de treino
inconsist1_treino <- data %>%
  filter(SENIORITY > AGE)
inconsist1_treino
```

```
[1] y         MEN       URBAN     PRIVATE   AGE       SENIORITY
<0 linhas> (ou row.names de comprimento 0)
```

Encontramos uma inconsistência no conjunto de teste e procedemos a sua deleção.

```r
inconsist1_teste <- data_test %>%
  filter(SENIORITY > AGE)

inconsist1_teste
```

```
  y MEN URBAN PRIVATE AGE SENIORITY
1 1   1     0       1  33        34
```

```r
todel_index <- which(data_test$SENIORITY > data_test$AGE)

data_test <- data_test[-c(todel_index),]
```

Filtramos também indivíduos menores de idade e encontramos duas observações no conjunto de treino. Não temos informação sobre o país na qual a amostra foi retirada, então não podemos concluir sobre a possibilidade de se conseguir a habilitação antes dos 18 anos. Assim, procedemos para a deleção desses casos.

```r
inconsist2 <- data %>%
  filter(AGE < 18)
inconsist2
```

```
  y MEN URBAN PRIVATE AGE SENIORITY
1 0   1     0       1  15         3
2 1   1     1       1  17         2
```

```r
todel_index <- which(data$AGE < 18)
data <- data[-c(todel_index),]
```

Igualmente para o conjunto de teste, encontramos uma observação inconsistente e a deletamos:

```r
inconsist2 <- data_test %>%
  filter(AGE < 18)

inconsist2
```

```
  y MEN URBAN PRIVATE AGE SENIORITY
1 0   1     0       1  16         3
```

```r
todel_index <-which(data_test$AGE < 18)
data_test <- data_test[-c(todel_index),]
```

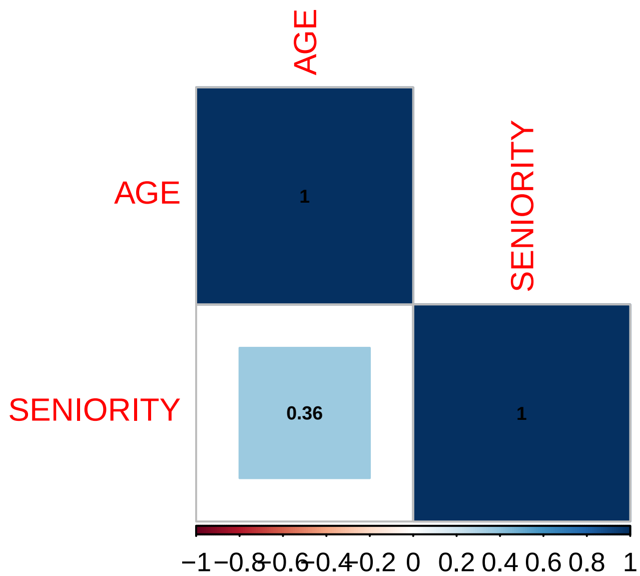
Correlações

Inicialmente, fazemos uma análise de correlação para descobrir se as covariáveis possuem alguma forte relação linear entre si (multicolineariedade) ou com a variável resposta.

Pearson

Aplicamos a correlação de Pearson para identificar correlação linear entre variáveis numéricas.

```r
numeric <- c("AGE", "SENIORITY")
df_pearson<- data[, numeric]



correl<- cor(df_pearson, method = 'pearson')

testRes = corrplot::cor.mtest(df_pearson, conf.level = 0.95, method = 'pearson')

corrplot::corrplot(correl, p.mat = testRes$p, addCoef.col ='black',method = 'square', orde
```

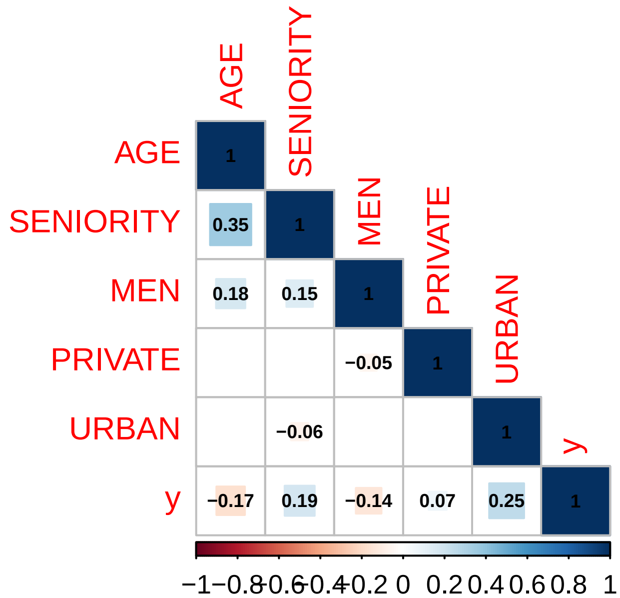


Vemos que Seniority e Age possuem uma correlação estatisticamente significante, ou seja, diferente de zero, porém de baixo módulo. Logo multicolinearidade não parece ser um problema.

Spearman

Calculamos também o coeficiente de Spearman para identificar correlações monotônicas entre as variáveis. Valores faltantes indicam que a correlação é estatististicamente igual a zero ao nível de 5% de confiança.

```
numeric <- c("AGE", "SENIORITY", "y", "MEN", "URBAN", "PRIVATE")
df_spearman<- data[, numeric]



correl<- cor(df_spearman, method = 'spearman')

testRes = corrplot::cor.mtest(df_spearman, conf.level = 0.95, method = 'spearman')

corrplot::corrplot(correl, p.mat = testRes$p, addCoef.col ='black',method = 'square', orde
```

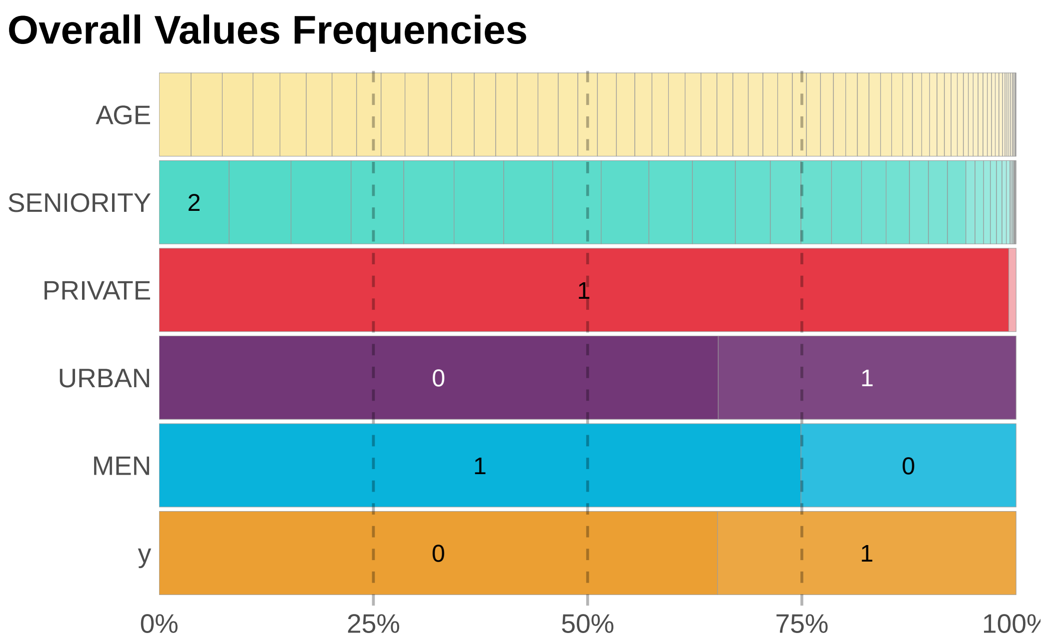


De forma geral, as correlações entre as covariáveis ou são fracas ou inexistentes. Em relação a variável resposta, temos que URBAN possui o maior valor em módulo, porém este também indica fraca correlação monotônica.

Visão Geral dos Dados

A seguir verificamos a distribuição dos dados na base de treino. Inicialmente, notamos que os dados são relativamente desbalanceados em relação a variável resposta, já que temos mais de 50% de observações para y = 0. As covariáveis binárias também apresentam desbalanceio, sendo caso mais evidente da variável PRIVATE.

```
library(lares)

freqs_df(data, plot = T)
```



Abaixo, temos um resumo quantitativo dos dados. A parte das variávies binárias, notamos que os indivíduos da base de treino tem em média 45 anos e 10 anos de carta de motorista.

```
summary(data)
```

```
       y               MEN              URBAN            PRIVATE      
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.0000  
 Median :0.0000   Median :1.0000   Median :0.0000   Median :1.0000  
 Mean   :0.3484   Mean   :0.7489   Mean   :0.3476   Mean   :0.9914  
 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
      AGE          SENIORITY    
 Min.   :18.00   Min.   : 2.00  
 1st Qu.:36.00   1st Qu.: 5.00  
 Median :45.00   Median : 9.00  
 Mean   :46.06   Mean   :10.61  
 3rd Qu.:55.00   3rd Qu.:15.00  
 Max.   :88.00   Max.   :40.00  
```

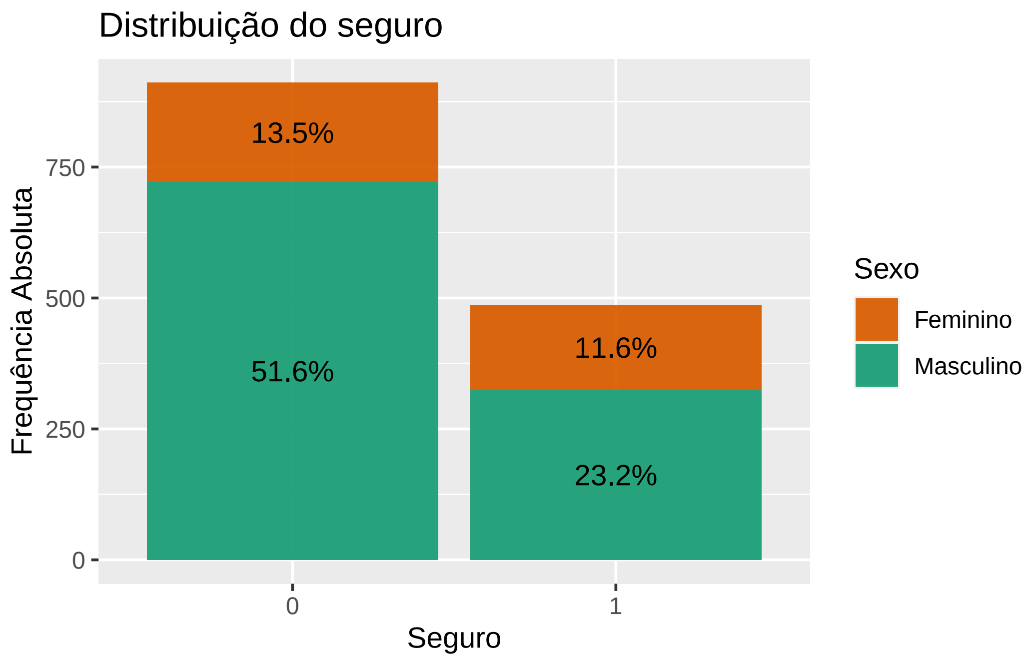
Distribuição do Seguro por Covariáveis

Nessa seção, avaliamos quantitativamente a relação entre covariáveis e variável resposta.

Seguro vs Sexo (MEN)

Como apresentado abaixo, a grande maioria dos dados são referentes a homens que não adquiriram o seguro completo. A porcentagem de mulheres é aproximadamente igual entre os grupos, indicando que não parece haver diferença entre a proporção de indivíduos do sexo feminino entre os dois grupos, contudo temos uma proporção maior de indivíduos do sexo masculino que não adquiriram seguro.

```
hist_men <- ggplot(data %>% count(y, MEN) %>%
         mutate(pct=n/sum(n)),
       aes(as.factor(y), n, fill=as.factor(MEN))) +
  geom_bar(stat="identity") +
  geom_text(aes(label=paste0(sprintf("%1.1f", pct*100),"%")), position=position_stack(vjus
scale_fill_manual(guide_legend(title="Sexo"), values = c("#D95F02", "#1B9E77"),labels = c(
  labs(x = 'Seguro', y = 'Frequência Absoluta', title = "Distribuição do seguro")


hist_men
```



Notamos também que a maioria dos dados são referentes a indivíduos que não possuem cobertura completa do seguro, tal desbalanceio pode dificultar a modelagem do problema.

```
print(paste('Indíviduos sem cobertura completa',nrow(filter(data, y == '0'))))
```

```
[1] "Indíviduos sem cobertura completa 911"
```

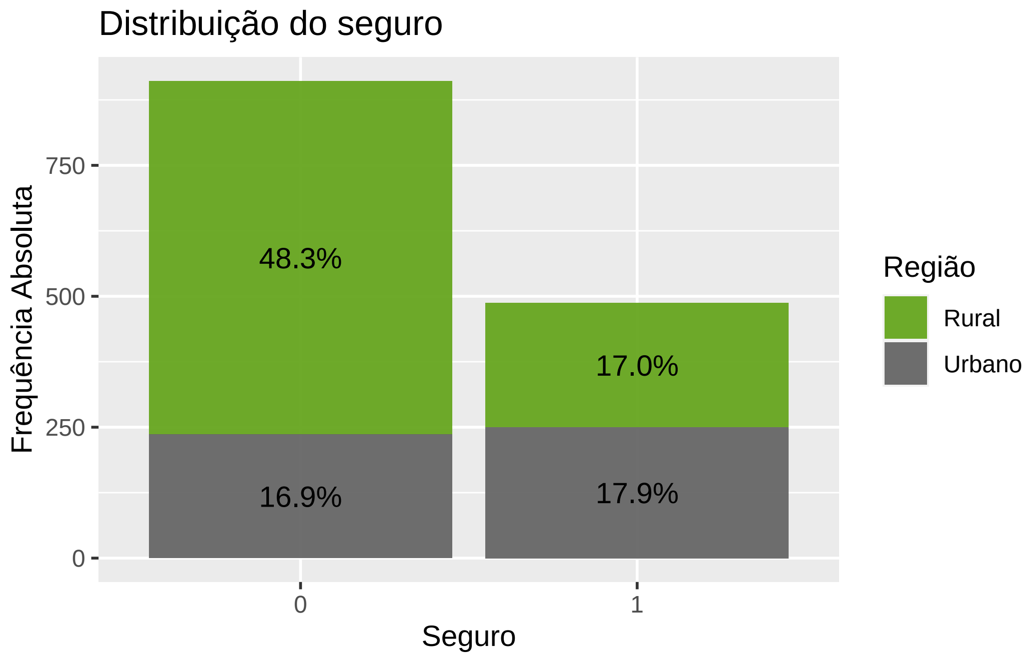
```
print(paste('Indíviduos com cobertura completa',nrow(filter(data, y == '1'))))
```

```
[1] "Indíviduos com cobertura completa 487"
```

Seguro vs região onde dirige (URBAN)

A maior parte dos indivíduos dirigem em áreas rurais e não adquiriram o seguro completo. Também não notamos diferenças significativas entre a proporção de indivíduos com seguro completo que dirigem na área urbana, porém a maior parte dos indivíduos que dirigem na área rural não adquiriram seguro completo.

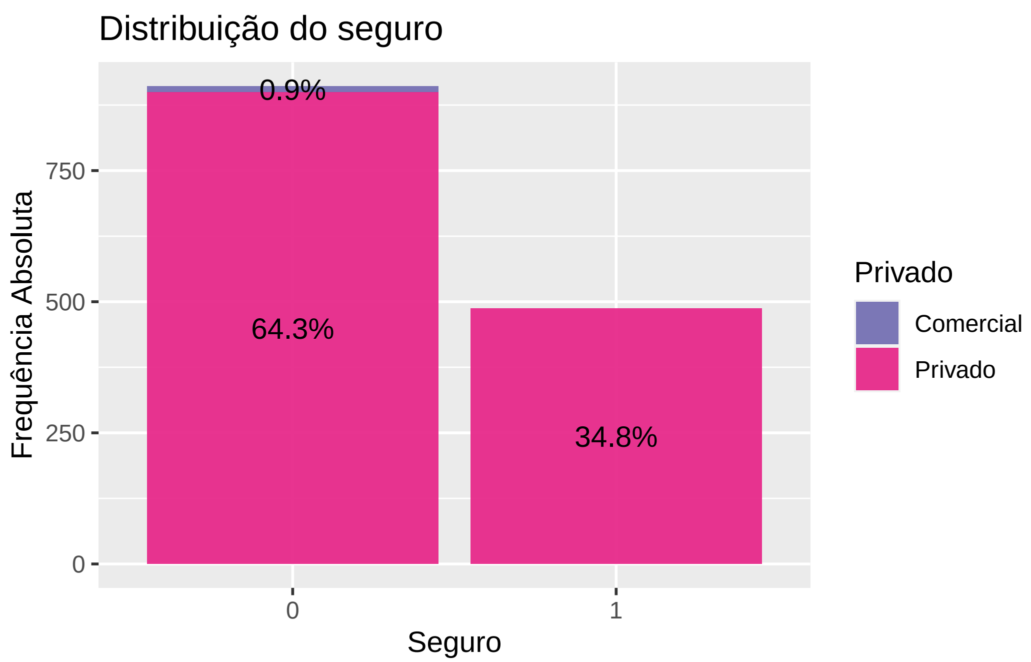
```
hist_urban<- ggplot(data %>% count(y, URBAN) %>%
         mutate(pct=n/sum(n)),
       aes(as.factor(y), n, fill=as.factor(URBAN))) +
  geom_bar(stat="identity") +
  geom_text(aes(label=paste0(sprintf("%1.1f", pct*100),"%")), position=position_stack(vjus
scale_fill_manual(guide_legend(title="Região"), values = c("#66A61E", "#666666"), labels =
  labs(x = 'Seguro', y = 'Frequência Absoluta', title = "Distribuição do seguro")

hist_urban
```



Seguro vs Uso de Veículo (PRIVATE)

Já para o tipo de uso do veículo, temos que 64.3% dos dados da amostra são de indivíduos que utilizam automóveis privativamente e não adquiriram o seguro completo. Nota-se também que apenas 0.9% dos dados referem-se a indivíduos que possuem veículos para uso comercial e todos eles não adquiriram seguro completo.

```
hist_private<- ggplot(data %>% count(y, PRIVATE) %>%
         mutate(pct=n/sum(n)),
       aes(as.factor(y), n, fill=as.factor(PRIVATE))) +
  geom_bar(stat="identity") +
  geom_text(aes(label=paste0(sprintf("%1.1f", pct*100),"%")), position=position_stack(vjus
scale_fill_manual(guide_legend(title="Privado"), values = c("#7570B3", "#E7298A"), labels
  labs(x = 'Seguro', y = 'Frequência Absoluta', title = "Distribuição do seguro")

hist_private
```



Se filtrarmos os indivíduos que usam veículos de forma comercial, também vemos que todos são do sexo masculino e não adquiriram o seguro completo.

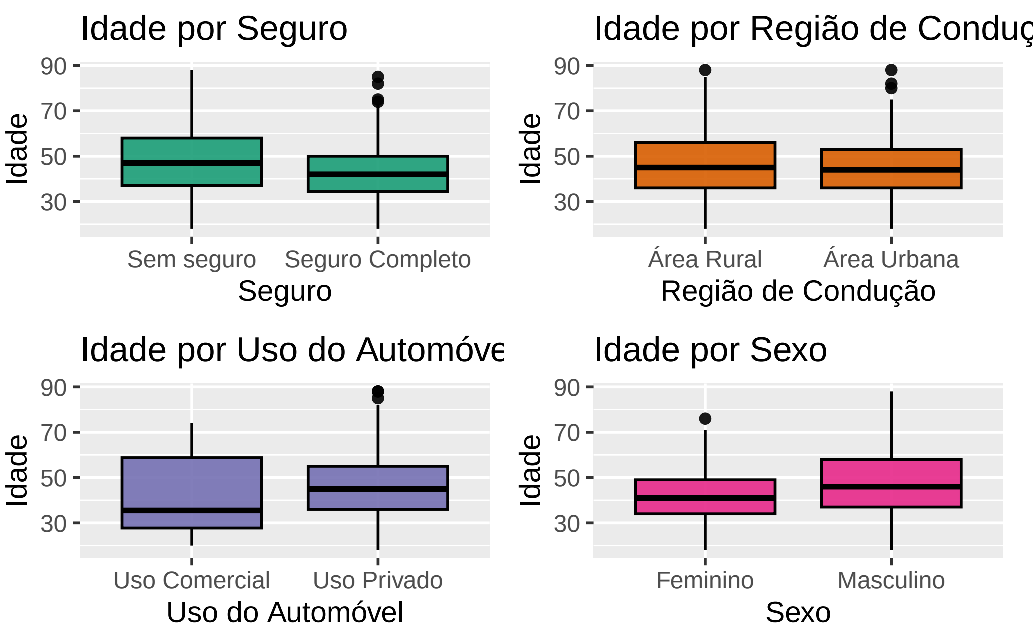
```
df <- data %>%
  filter(PRIVATE == 0)
df
```

```
   y MEN URBAN PRIVATE AGE SENIORITY
1  0   1     0       0  34         9
2  0   1     1       0  54        16
3  0   1     0       0  27         2
4  0   1     1       0  74         3
5  0   1     1       0  29         2
6  0   1     0       0  37        13
7  0   1     0       0  58        14
8  0   1     1       0  28         7
9  0   1     1       0  61        10
10 0   1     0       0  72        10
11 0   1     0       0  24         4
12 0   1     1       0  20         2
```

Idade e Tempo de Carteira

Avaliando a idade em relação as variáveis binárias, vemos que indivíduos que adquirem o seguro completo ou dirigem em área urbana tem em média uma menor idade. Por outro lado, indivíduos que fazem uso privado do automóvel ou são do sexo masculino são mais velhos que seus contrapontos.

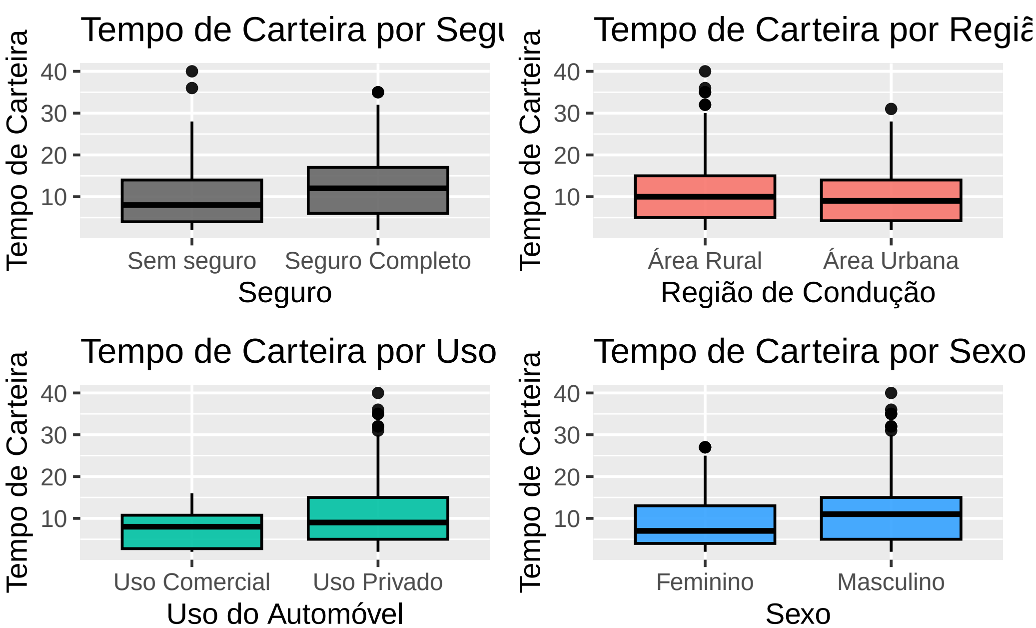
```r
p1 <- ggplot(data, aes(x=as.factor(y), y=AGE)) +
  geom_boxplot(notch=FALSE, fill = "#1B9E77", color = 'black') +
  scale_x_discrete(labels = c('Sem seguro', 'Seguro Completo'))+
  labs(title = 'Idade por Seguro',
       x = 'Seguro',
       y = 'Idade')

p2 <- ggplot(data, aes(x=as.factor(URBAN), y=AGE)) +
  geom_boxplot(notch=FALSE,fill = "#D95F02", color = 'black') +
    scale_x_discrete(labels = c('Área Rural', 'Área Urbana'))+
  labs(title = 'Idade por Região de Condução',
       x = 'Região de Condução',
       y = 'Idade')

p3 <- ggplot(data, aes(x=as.factor(PRIVATE), y=AGE)) +
  geom_boxplot(notch=FALSE,fill = "#7570B3", color = 'black') +
    scale_x_discrete(labels = c('Uso Comercial', 'Uso Privado'))+
  labs(title = 'Idade por Uso do Automóvel',
       x = 'Uso do Automóvel',
       y = 'Idade')

p4 <- ggplot(data, aes(x=as.factor(MEN), y=AGE, color = as.factor(y))) +
  geom_boxplot(notch=FALSE,fill = "#E7298A", color = 'black') +
    scale_x_discrete(labels = c('Feminino', 'Masculino'))+
  labs(title = 'Idade por Sexo',
       x = 'Sexo',
       y = 'Idade')

(p1|p2)/(p3|p4)
```



Para o tempo de carteira, notamos que indivíduos que não possuem seguro completo ou que fazem uso comercial do automóvel ou que sejam do sexo feminino, são em média mais novos do que seus opostos. Já indivíduos que dirigem pela área rural, parecem ter uma leve tendência a serem mais velhos do que indivíduos que dirigem na área urbana.

```
p1 <- ggplot(data, aes(x=as.factor(y), y=SENIORITY)) +
  geom_boxplot(notch=FALSE, fill = "#666666", color = 'black') +
  scale_x_discrete(labels = c('Sem seguro', 'Seguro Completo'))+
  labs(title = 'Tempo de Carteira por Seguro',
       x = 'Seguro',
       y = 'Tempo de Carteira')

p2 <- ggplot(data, aes(x=as.factor(URBAN), y=SENIORITY)) +
  geom_boxplot(notch=FALSE,fill = "#F8766D", color = 'black') +
    scale_x_discrete(labels = c('Área Rural', 'Área Urbana'))+
  labs(title = 'Tempo de Carteira por Região de Condução',
       x = 'Região de Condução',
       y = 'Tempo de Carteira')

p3 <- ggplot(data, aes(x=as.factor(PRIVATE), y=SENIORITY)) +
  geom_boxplot(notch=FALSE,fill = "#00C1A3", color = 'black') +
    scale_x_discrete(labels = c('Uso Comercial', 'Uso Privado'))+
```

```
  labs(title = 'Tempo de Carteira por Uso do Automóvel',
       x = 'Uso do Automóvel',
       y = 'Tempo de Carteira')

p4 <- ggplot(data, aes(x=as.factor(MEN), y=SENIORITY)) +
  geom_boxplot(notch=FALSE,fill = "#35A2FF", color = 'black') +
    scale_x_discrete(labels = c('Feminino', 'Masculino'))+
  labs(title = 'Tempo de Carteira por Sexo',
       x = 'Sexo',
       y = 'Tempo de Carteira')

(p1|p2)/(p3|p4)
```



Modelos Lineares Generalizados (Clássicos)

Nessa seção, temos o objetivo de ajustar o melhor modelo para predizer, dado as informações do indivíduo, se ele possui ou não cobertura completa do seguro.

```
data_reg <- data %>%
  mutate_at(vars("URBAN", "MEN", "PRIVATE", "y"), as.factor)
```

```
data_test <- data_test %>%
  mutate_at(vars("URBAN", "MEN", "PRIVATE", "y"), as.factor)
```

Regressão Logística (GLM com ligação logit)

Começamos ajustando um modelo de regressão logística clássico. Usamos todas as covariáveis possíveis e aplicamos testes de significância aos coeficientes.

```
m0 <- glm(formula = y ~  URBAN + SENIORITY+ AGE + MEN + PRIVATE,
 family = binomial(link = "logit"), data = data_reg)

summary(m0)
```

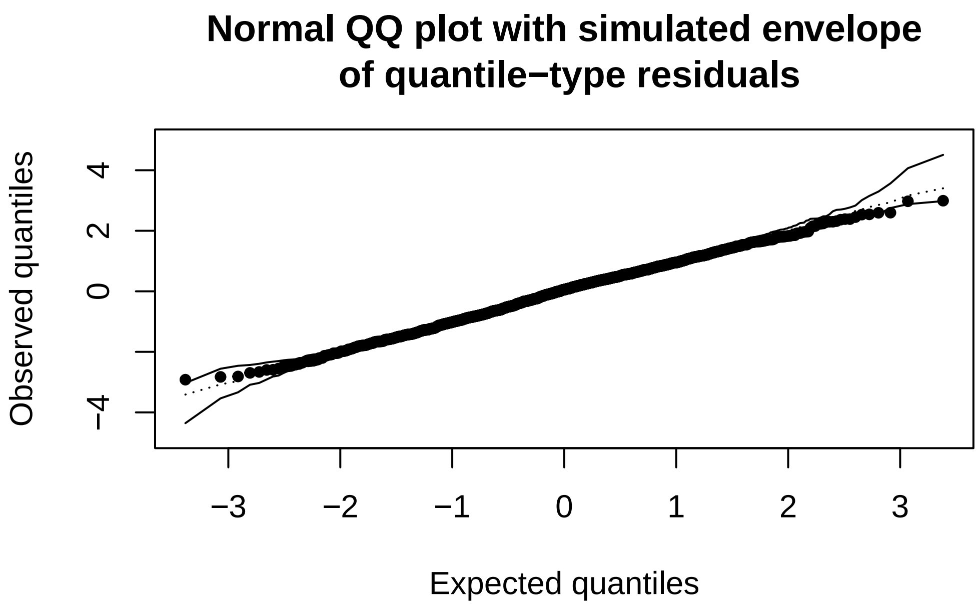
```
Call:
glm(formula = y ~ URBAN + SENIORITY + AGE + MEN + PRIVATE, family = binomial(link = "logit")
    data = data_reg)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4302  -0.7949  -0.4924   1.0330   2.6766

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -14.34106  376.42929  -0.038    0.970
URBAN1        1.38994    0.13302  10.449  < 2e-16 ***
SENIORITY     0.13340    0.01159  11.508  < 2e-16 ***
AGE          -0.05711    0.00599  -9.534  < 2e-16 ***
MEN1         -0.71958    0.14393  -5.000 5.75e-07 ***
PRIVATE1     14.81859  376.42923   0.039    0.969
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1807.4  on 1397  degrees of freedom
Residual deviance: 1494.0  on 1392  degrees of freedom
AIC: 1506

Number of Fisher Scoring iterations: 14
```

O intercepto e o coeficiente de Private são não significativos, mas procederemos para a retirada apenas do segundo.

```
m1_logit <- glm(formula = y ~  URBAN + SENIORITY+ AGE + MEN,
 family = binomial(link = "logit"), data = data_reg)
summary(m1_logit)
```

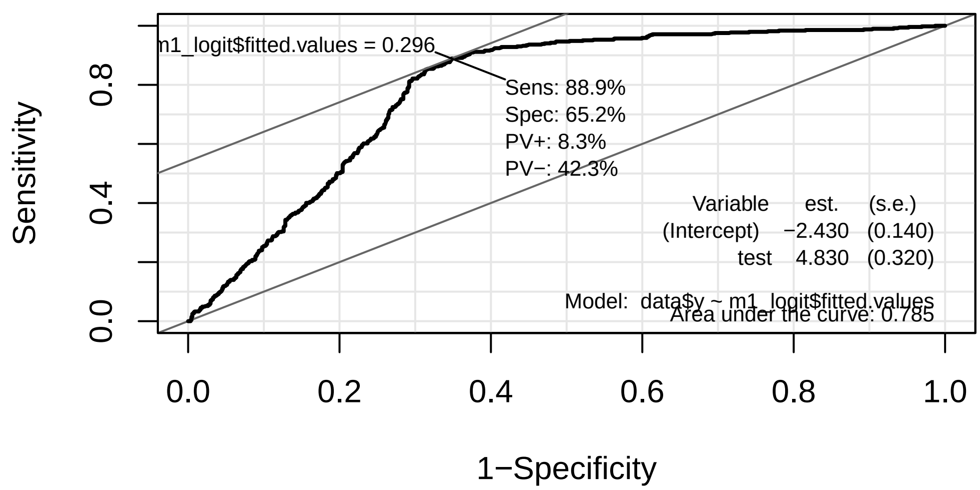
```
Call:
glm(formula = y ~ URBAN + SENIORITY + AGE + MEN, family = binomial(link = "logit"),
    data = data_reg)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4332  -0.7940  -0.5024   1.0405   2.6796

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.448334   0.244105   1.837   0.0663 .
URBAN1       1.375878   0.132474  10.386  < 2e-16 ***
SENIORITY    0.134149   0.011578  11.586  < 2e-16 ***
AGE         -0.056425   0.005951  -9.481  < 2e-16 ***
MEN1        -0.745306   0.143671  -5.188 2.13e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1807.4  on 1397  degrees of freedom
Residual deviance: 1503.9  on 1393  degrees of freedom
AIC: 1513.9

Number of Fisher Scoring iterations: 4
```

Verificamos agora o ajuste do modelo utilizando os resíduos de quantil randomizados. De forma geral, se o modelo faz um bom ajuste dos dados, esperamos que apenas 5% dos pontos se encontrem fora do envelope simulado. Pelo envelope simulado gerado abaixo não notamos fortes desvios.



Ao analisar a curva ROC, obtemos um valor de 0.785 pra área sob a curva.

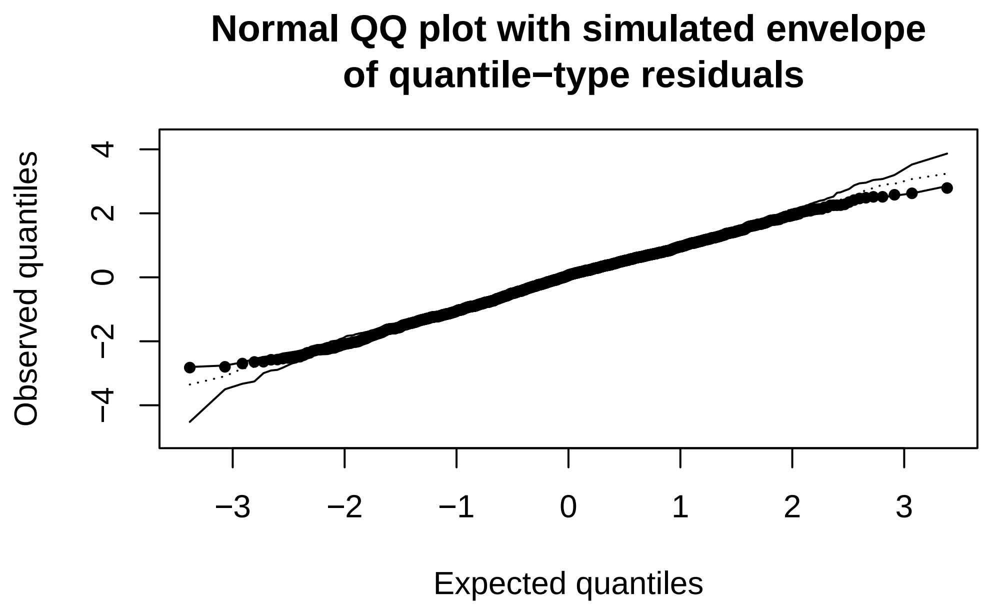
```
library(Epi)
Epi::ROC(m1_logit$fitted.values, data$y, plot= "ROC")
```

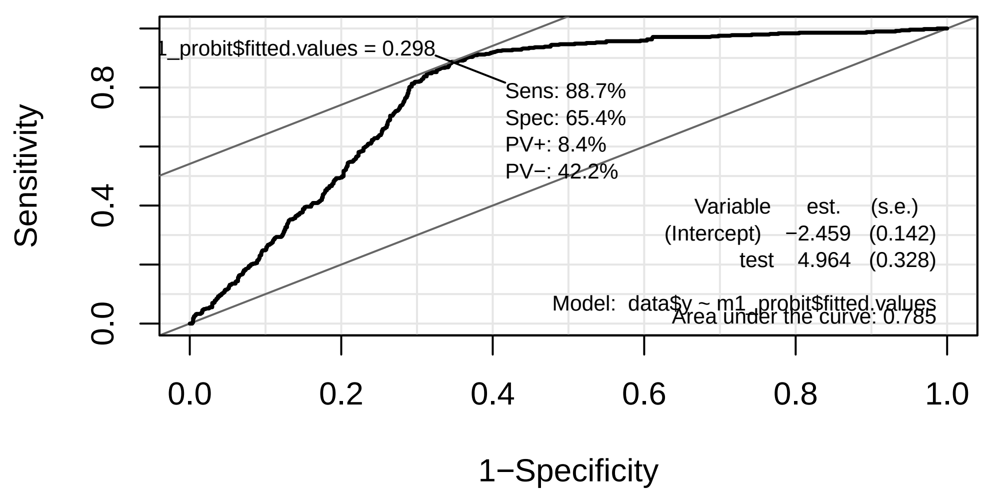


Regressão Logística (Probit, Cauchit, Cloglog, loglog)

Probit

Ao utlizar a função de ligação probito, temos um modelo também sem indícios de mau ajuste e com área sob a curva também de 0.785.



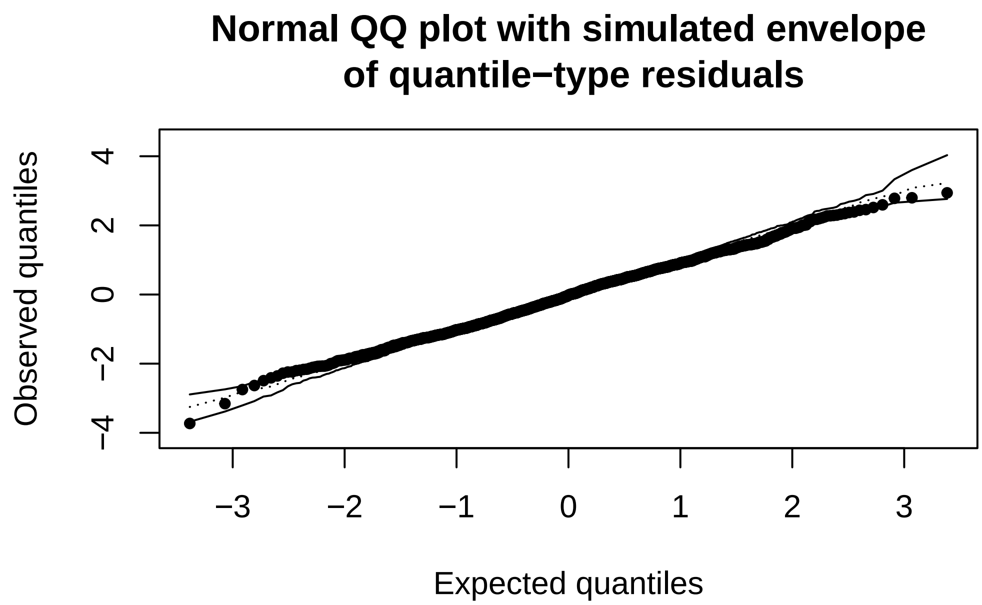


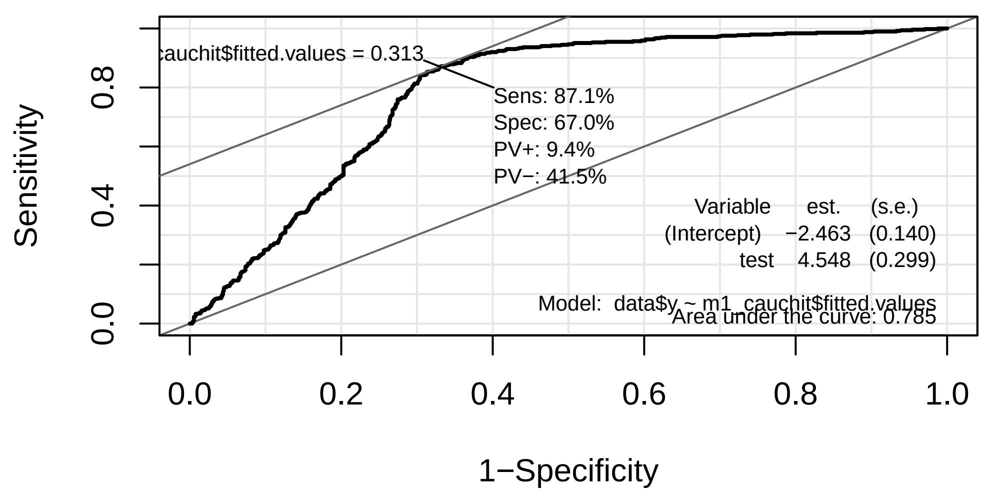

```
summary(m1_probit)
```

```
Call:
glm(formula = y ~ URBAN + SENIORITY + AGE + MEN, family = binomial(link = "probit"),
    data = data_reg)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4789  -0.7971  -0.4952   1.0597   2.8467

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.242315   0.145182   1.669   0.0951 .
URBAN1       0.822382   0.078025  10.540  < 2e-16 ***
SENIORITY    0.078098   0.006620  11.797  < 2e-16 ***
AGE         -0.033146   0.003405  -9.735  < 2e-16 ***
MEN1        -0.433073   0.085672  -5.055  4.3e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1807.4  on 1397  degrees of freedom
Residual deviance: 1502.4  on 1393  degrees of freedom
AIC: 1512.4

Number of Fisher Scoring iterations: 5
```

Cauchit

Usando a função de ligação Cauchit:



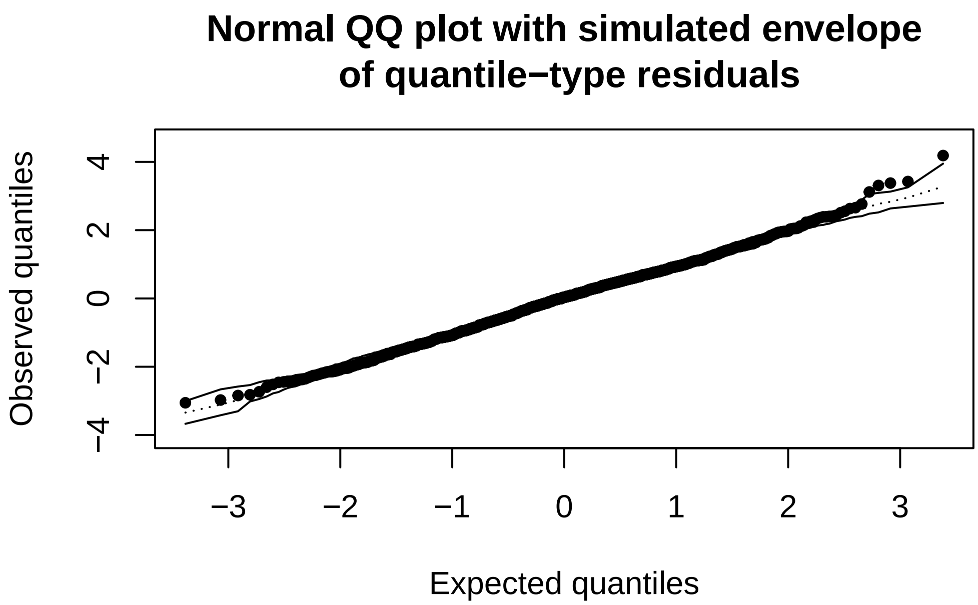


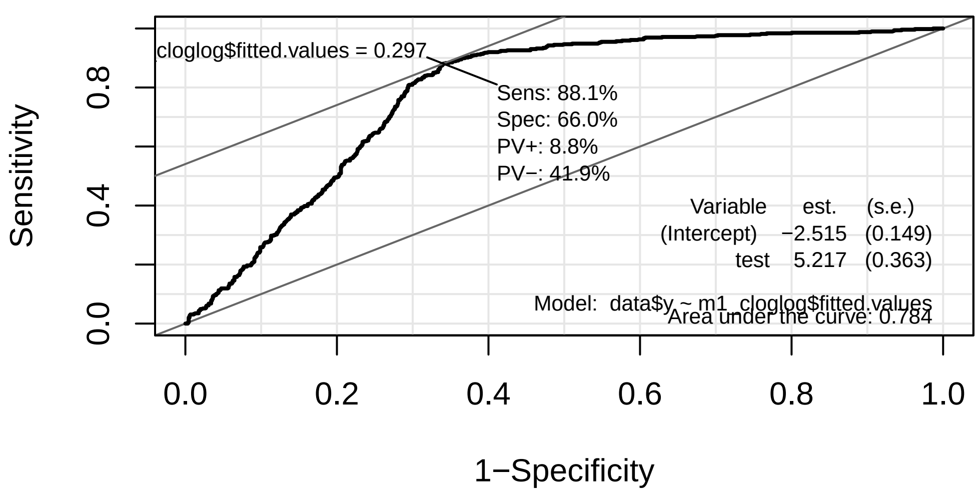

```
summary(m1_cauchit)
```

```
Call:
glm(formula = y ~ URBAN + SENIORITY + AGE + MEN, family = binomial(link = "cauchit"),
    data = data_reg)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1595  -0.8026  -0.5833   0.9427   2.1985

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.612386   0.244848   2.501   0.0124 *
URBAN1       1.356105   0.149978   9.042  < 2e-16 ***
SENIORITY    0.139137   0.014477   9.611  < 2e-16 ***
AGE         -0.056680   0.007021  -8.073 6.88e-16 ***
MEN1        -0.819398   0.148459  -5.519 3.40e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1807.4  on 1397  degrees of freedom
Residual deviance: 1524.5  on 1393  degrees of freedom
AIC: 1534.5

Number of Fisher Scoring iterations: 8
```

Cloglog

Para a função de ligação Cloglog:



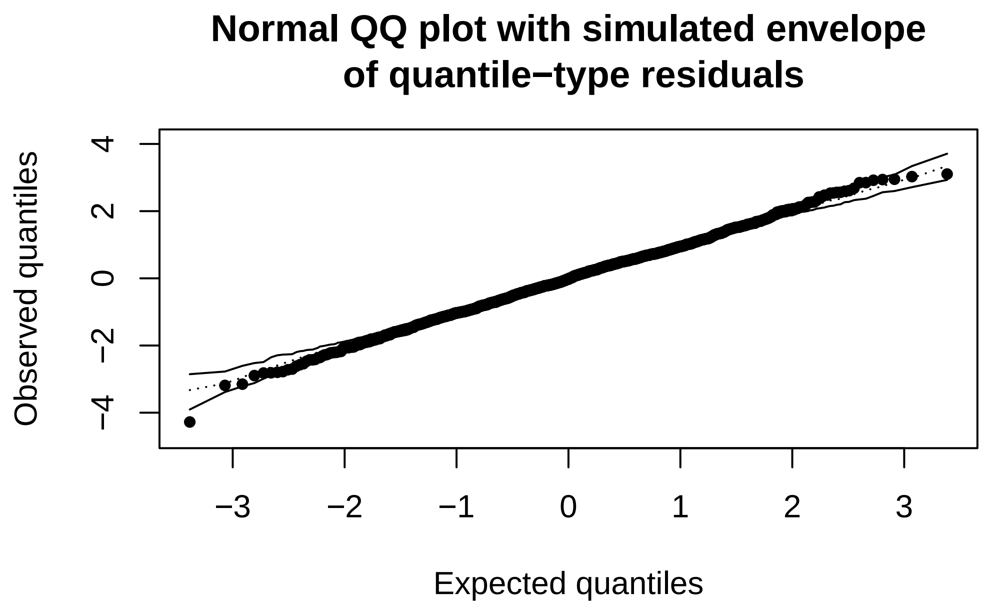


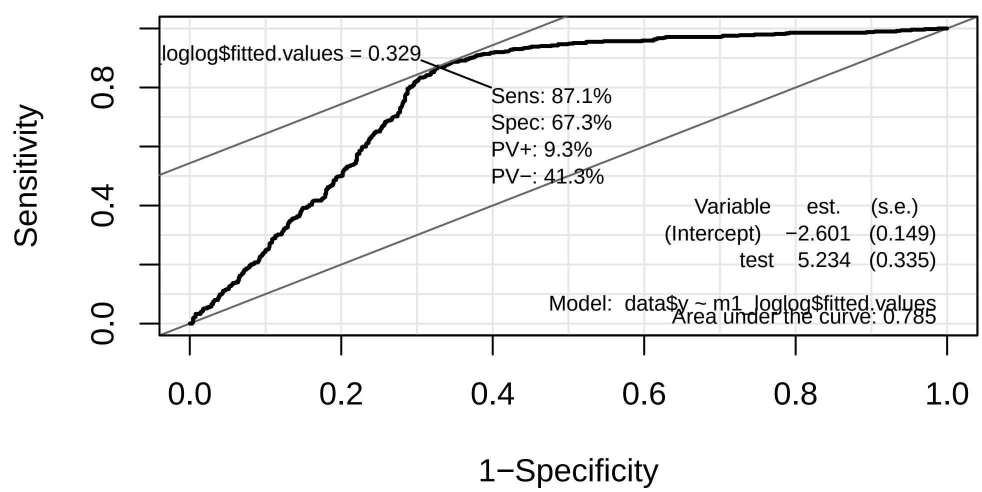

```
summary(m1_cloglog)
```

```
Call:
glm(formula = y ~ URBAN + SENIORITY + AGE + MEN, family = binomial(link = "cloglog"),
    data = data_reg)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7618  -0.8054  -0.5786   1.1169   2.3819

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.156985   0.185004  -0.849    0.396
URBAN1       0.921767   0.095309   9.671  < 2e-16 ***
SENIORITY    0.084626   0.007959  10.633  < 2e-16 ***
AGE         -0.038024   0.004453  -8.538  < 2e-16 ***
MEN1        -0.432057   0.103489  -4.175 2.98e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1807.4  on 1397  degrees of freedom
Residual deviance: 1538.1  on 1393  degrees of freedom
AIC: 1548.1

Number of Fisher Scoring iterations: 10
```

Loglog

O modelo com função de ligação loglog:






```
summary(m1_loglog)
```

```
Call:
glm(formula = y ~ URBAN + SENIORITY + AGE + MEN, family = binomial(link = loglog()),
    data = data_reg)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1272  -0.8219  -0.4540   1.0432   3.4517

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.632619   0.147459   4.290 1.79e-05 ***
URBAN1       0.889803   0.083497  10.657  < 2e-16 ***
SENIORITY    0.081249   0.006944  11.701  < 2e-16 ***
AGE         -0.033515   0.003320 -10.094  < 2e-16 ***
MEN1        -0.482435   0.090741  -5.317 1.06e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1807.4  on 1397  degrees of freedom
Residual deviance: 1484.1  on 1393  degrees of freedom
AIC: 1494.1

Number of Fisher Scoring iterations: 5
```

AIC

Utilizamos o critério de Akaike para selecionar um modelo:

```
# Dataframe para verificar o AIC
data.frame(Modelo=c("Modelo logito","Modelo probito","Modelo cauchito","Modelo cloglog","M
AIC = c(AIC(m1_logit),AIC(m1_probit),AIC(m1_cauchit),
AIC(m1_cloglog), AIC(m1_loglog)))
```

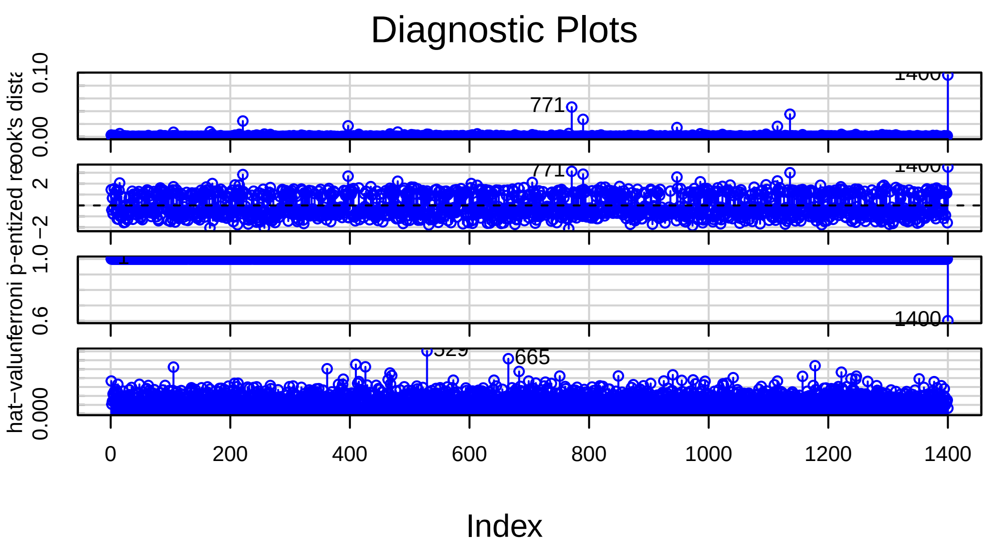
```
           Modelo      AIC
1   Modelo logito 1513.942
2  Modelo probito 1512.401
3 Modelo cauchito 1534.467
4  Modelo cloglog 1548.091
5   Modelo loglog 1494.055
```

Comparando os modelos pelo critério de Akaike, temos que o modelo loglog tende o mais balanceado em relação a qualidade de ajuste e quantidade de parâmetros. A seguir, analisamos alavancagem e influência dos pontos utilizados no ajuste.

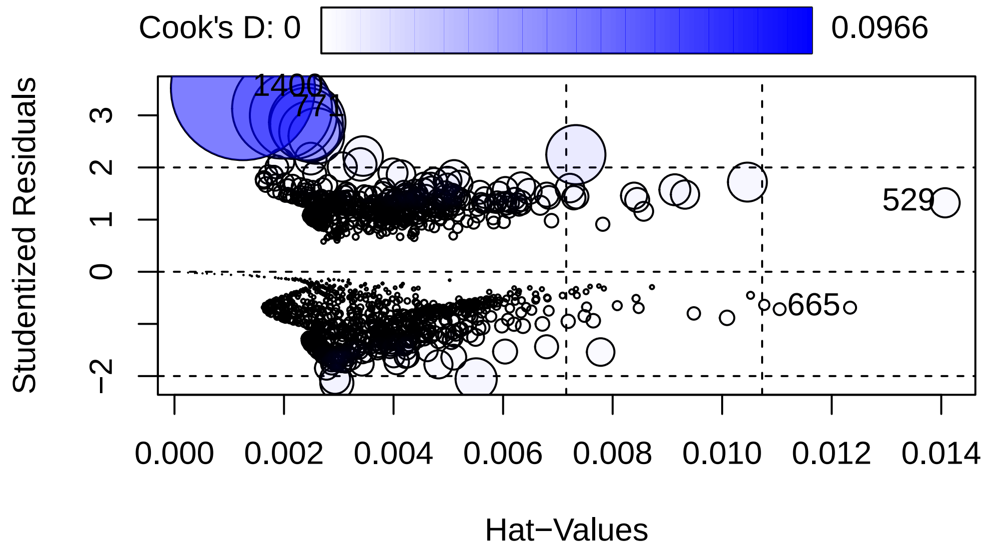
Retirada de pontos influentes Loglog

O gráfico em azul apresenta a distância de Cook para identificar pontos influentes. O segundo gráfico mostra os resíduos studentizados e a tabela apresenta informações de alguns pontos apontados como outlier usando o teste de Bonferroni para outliers.

```
#plot(m1_loglog)
influenceIndexPlot(m1_loglog,col='blue')
```




```
influencePlot(m1_loglog)
```




```
         StudRes         Hat        CookD
529   1.3226722 0.014070114 0.0039799795
665  -0.6879603 0.012335490 0.0006699527
771   3.1288347 0.001964524 0.0465263325
1400  3.5208389 0.001249995 0.0965990925
```

Pelos valores de resíduos studentizados, temos que os pontos 771 e 1400 encontram-se foram do intervalor (-2,2).

Para avaliar a alavancagem, procuramos valores de Hat superiores a 2*p/n = 0,007. Assim, concluimos que os pontos 529 e 665 apresentam indícios de impactarem nas predições do modelo. Como esses pontos também foram postos em evidência pela distância de Cook, vamos verificar o impacto de sua retirada.

Ajustamos um modelo considerando a retirada de cada ponto e um modelo considerando a remoção de ambos.

```
ajuste1 <- glm(formula = y ~URBAN + SENIORITY+ AGE + MEN,
 family = binomial(link = loglog()), subset = -c(529) , data = data_reg)

ajuste2 <- glm(formula = y ~  URBAN + SENIORITY+ AGE + MEN,
 family = binomial(link = loglog()), subset = -c(665) , data = data_reg)
```

```r
ajuste3 <- glm(formula = y ~  URBAN + SENIORITY+ AGE + MEN,
 family = binomial(link = loglog()), subset = -c(529,665) , data = data_reg)


compareCoefs(m1_loglog,ajuste1, ajuste2, ajuste3)
```

```
Calls:
1: glm(formula = y ~ URBAN + SENIORITY + AGE + MEN, family = binomial(link =
   loglog()), data = data_reg)
2: glm(formula = y ~ URBAN + SENIORITY + AGE + MEN, family = binomial(link =
   loglog()), data = data_reg, subset = -c(529))
3: glm(formula = y ~ URBAN + SENIORITY + AGE + MEN, family = binomial(link =
   loglog()), data = data_reg, subset = -c(665))
4: glm(formula = y ~ URBAN + SENIORITY + AGE + MEN, family = binomial(link =
   loglog()), data = data_reg, subset = -c(529, 665))

             Model 1  Model 2  Model 3  Model 4
(Intercept)    0.633    0.631    0.632    0.631
SE             0.147    0.147    0.147    0.147

URBAN1        0.8898   0.8881   0.8895   0.8878
SE            0.0835   0.0835   0.0835   0.0835

SENIORITY    0.08125  0.08117  0.08122  0.08114
SE           0.00694  0.00694  0.00695  0.00694

AGE         -0.03352 -0.03350 -0.03350 -0.03348
SE           0.00332  0.00332  0.00332  0.00332

MEN1         -0.4824  -0.4803  -0.4823  -0.4801
SE            0.0907   0.0908   0.0907   0.0908
```

Ao comparar os coeficientes entre os modelos não notamos diferenças significativas entre os coeficientes. Vejamos o impacto no AIC:

```r
data.frame(
Modelo= c("Completo", "Removendo 529", "Removendo 665",
"Removendo 529 e 665"),
AIC = c(AIC(m1_loglog),AIC(ajuste1), AIC(ajuste2), AIC(ajuste3)))
```

```
               Modelo      AIC
1            Completo 1494.055
2       Removendo 529 1493.381
3       Removendo 665 1494.008
4 Removendo 529 e 665 1493.334
```

Nota-se que o AIC diminui com a retirada de ambos os pontos. Sem uma base teórica que justifique a retirada desses pontos, não podemos simplesmente excluí-los da análise. Porém, vamos seguir utilizando o modelo reduzido para testar seu desempenho no conjunto de teste. Em seguida, ajustamos outros algoritmos para o conjunto de dados.

Predição no Conjunto de Teste

Abaixo, calculamos as métricas de acurácia, precisão, recall e f1score para os modelos loglog completo e reduzido. Os resultados são apresentados no momento oportuno no relatório.

```r
#cálculo de predição no conjunto de teste
accuracy <- function(model, model_roc, test_df) {

  #use roc returned by Epi roc function

  thresh_index <- which.max(rowSums(model_roc$res[, c("sens", "spec")]))

  recall <- model_roc$res[thresh_index,][1,1]
  precision <- model_roc$res[thresh_index,][1,3]

  f1_score <- 2*(recall*precision)/(recall+precision)


  thresh <- as.double(rownames(model_roc$res[thresh_index,][1]))
  predict_test <- ifelse(predict(model, newdata = test_df, type = 'response') >= thresh, 1

  acc <- mean(predict_test == test_df$y)

  return(c(acc,precision, recall,f1_score))

}
```

```r
#métricas para modelo reduzido
y_pred <- predict(ajuste3,newdata = data_test, type = 'response')
roc_logit <- Epi::ROC(y_pred, data_test$y, plot= T)
```

```
logitr_res_test <- accuracy(ajuste3, roc_logit, data_test )
logitr_acc_test <- logitr_res_test[1]
logitr_f1_test<-  logitr_res_test[4]
logitr_precision_test <- logitr_res_test[2]
logitr_recall_test <- logitr_res_test[3]
```

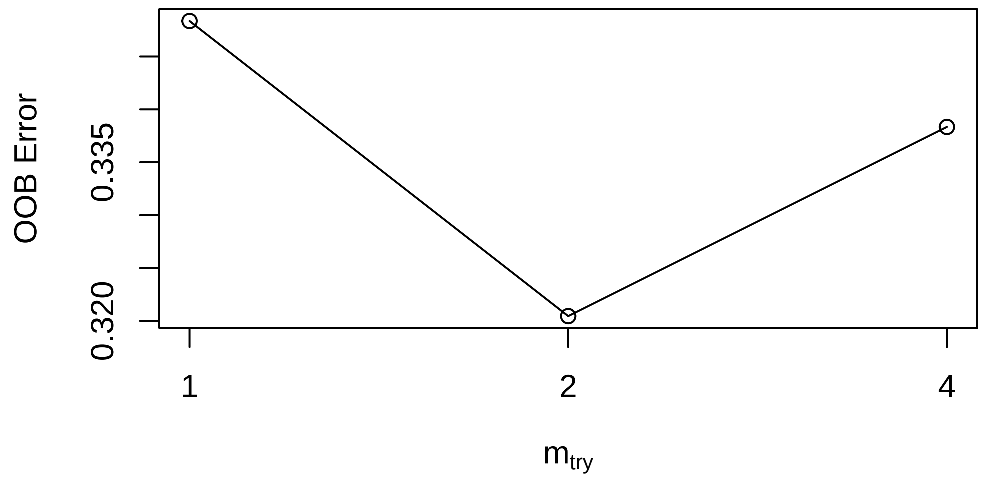
```
## métrica para modelo completo
y_pred <- predict(m1_loglog,newdata = data_test, type = 'response')
roc_loglog <- Epi::ROC(y_pred, data_test$y, plot= F)

loglog_res_test <- accuracy(m1_loglog, roc_loglog, data_test )
loglog_acc_test <- loglog_res_test[1]
loglog_f1_test<-  loglog_res_test[4]
loglog_precision_test <- loglog_res_test[2]
loglog_recall_test <- loglog_res_test[3]
```

Random Forest

O algoritmo de RandomForest trata-se de uma combinação de árvores de classificação. Os resultados gerados em diferentes árvores são posteriomente agregados em um único, de forma a reduzir a chance de superajuste (overfitting) e aumentar a acurácia.

Inicialmente, usamos a função tuneRF para encontrar o valor de mtry (número de variáveis aleatoriamente amostradas para ajustar as árvores).

```
mtry = 2  OOB error = 32.05%
Searching left ...
mtry = 4    OOB error = 33.83%
-0.05580357 0.05
Searching right ...
mtry = 1    OOB error = 34.84%
-0.08705357 0.05
```



Usando mtry = 2, usamos randomForest para ajustar o modelo.

```
rf <- randomForest(y~., data=data_reg, proximity=TRUE, mtry = 2)
print(rf)
```

```
Call:
 randomForest(formula = y ~ ., data = data_reg, proximity = TRUE,      mtry = 2)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 30.9%
Confusion matrix:
    0   1 class.error
0 690 221   0.2425906
1 211 276   0.4332649
```

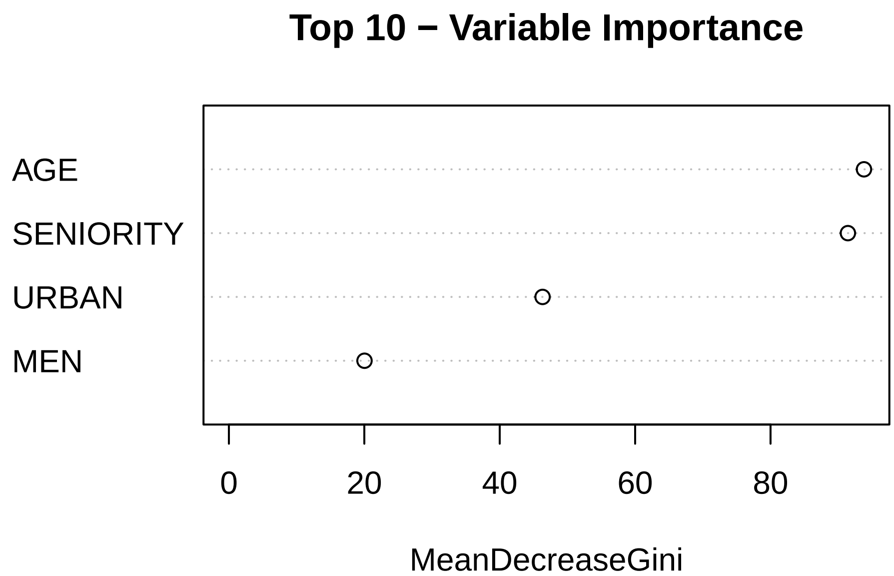
```
#plot(rf)
```

No conjunto de treino, temos um taxa de erro de 31.29%, vemos também que o modelo tende a errar mais para indivíduos que possuem cobertura completa do seguro. Abaixo calculamos as métricas de desempenho do modelo para serem comparadas.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 302  75
         1  97 124

               Accuracy : 0.7124
                 95% CI : (0.6743, 0.7484)
    No Information Rate : 0.6672
    P-Value [Acc > NIR] : 0.0101

                  Kappa : 0.3698

 Mcnemar's Test P-Value : 0.1093

            Sensitivity : 0.7569
            Specificity : 0.6231
         Pos Pred Value : 0.8011
         Neg Pred Value : 0.5611
             Prevalence : 0.6672
         Detection Rate : 0.5050
   Detection Prevalence : 0.6304
      Balanced Accuracy : 0.6900

       'Positive' Class : 0
```

Também olhamos para a importância das covariáveis na classificação usando o Decréscimo Médio no Índice de Gini e vemos que a idade (AGE) e tempo de carteira (SENIORITY) parecem serem os fatores mais determinantes para um indivíduo adquirir o seguro completo.

```r
#hist(treesize(rf),
#     main = "No. of Nodes for the Trees",
#     col = "green")
varImpPlot(rf,
           sort = T,
           n.var = 4,
           main = "Top 10 - Variable Importance")
```




```
importance(rf)
```

```
          MeanDecreaseGini
MEN              20.029378
URBAN            46.327670
PRIVATE           2.838403
AGE              93.800786
SENIORITY        91.425791
```

Soluções Bayesianas

Modelo Cauchit

A seguir, ajustaremos um modelo linear generalizado bayesiano com função de ligação de potência inversa de cauchit. Essa função de ligação

$$Y_i|\beta,\delta \sim Bernoulli(F(x_i\beta))$$

$$\beta \sim N(0,100)$$

$$\lambda \sim Uniform(-2, 2)$$

$$\lambda = \exp(\delta)$$

Em que:

$$F(x_i|\beta) = 1 - (\frac{1}{\pi} arctan(z) + 0.5)^{\lambda}$$

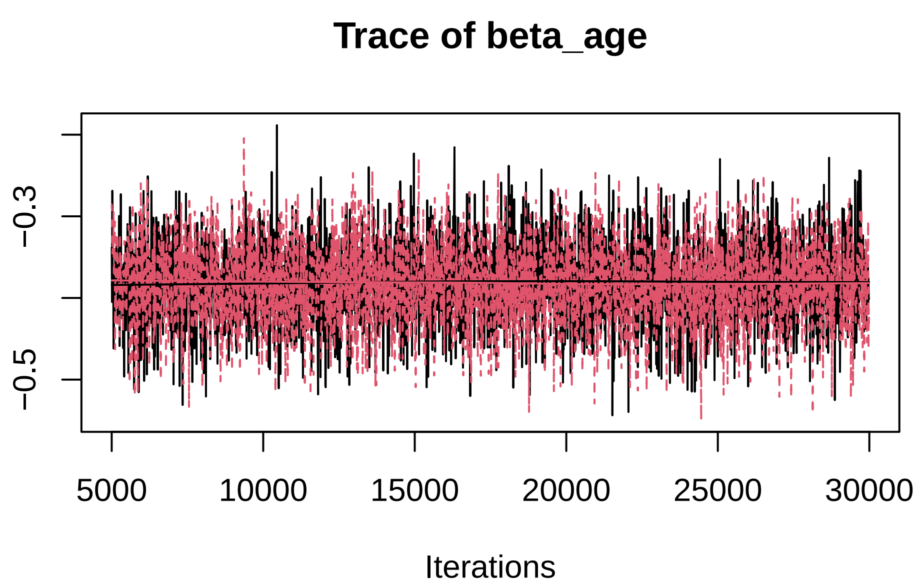
```
modelString="
model{
  for (i in 1:N) {
    y[i] ~ dbern(p[i])
    m[i] <- beta0+beta_urban*URBAN[i] + beta_sen*SEN[i] + beta_age*AGE[i] + beta_men*MEN[i
    pstar[i] <- 1/3.141592* arctan(-m[i]) + 1/2
    p[i] <- 1-pow(pstar[i], lambda)

  }
  beta0 ~ dnorm(0,100)
  delta ~ dunif(-2,2)
  lambda <- exp(delta)
  beta_urban ~ dnorm(0,100)
  beta_sen ~ dnorm(0,100)
  beta_age ~ dnorm(0,100)
  beta_men ~ dnorm(0,100)

  }

"
writeLines(modelString, con='models/m1_cauchit.bug')
```

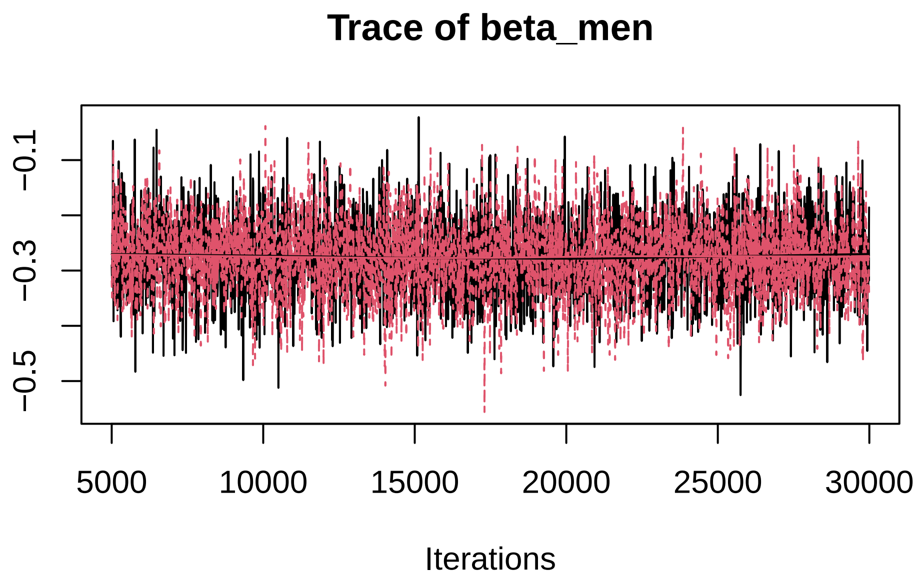
```
data_scaled = data
data_scaled$AGE <- as.numeric(scale(data$AGE))
data_scaled$SENIORITY <- as.numeric(scale(data$SENIORITY))

N = nrow(data_scaled)
jagsData <- list(N = N, y = data_scaled$y,
    URBAN = data_scaled$URBAN, SEN = data_scaled$SENIORITY    ,AGE = data_scaled$AGE, MEN
```

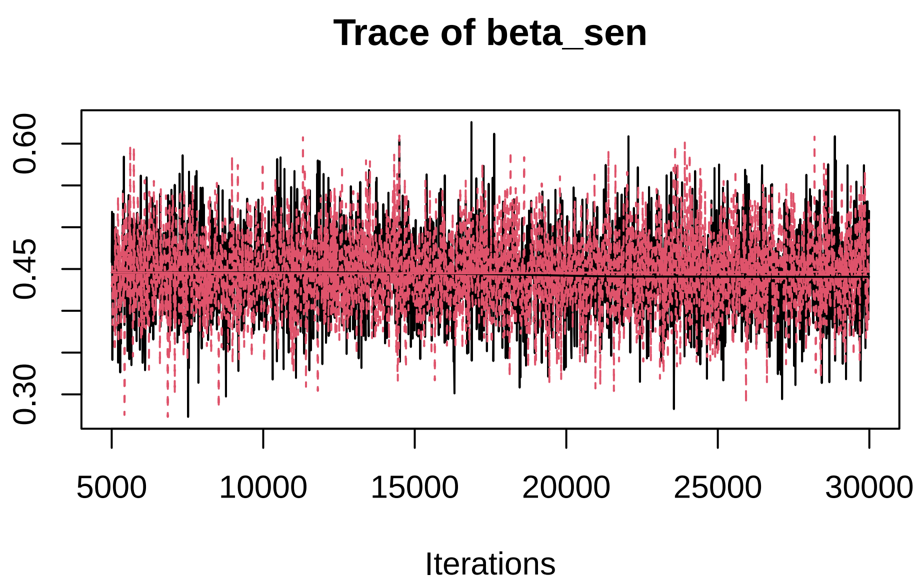
```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 1398
   Unobserved stochastic nodes: 6
   Total graph size: 14921

Initializing model
```

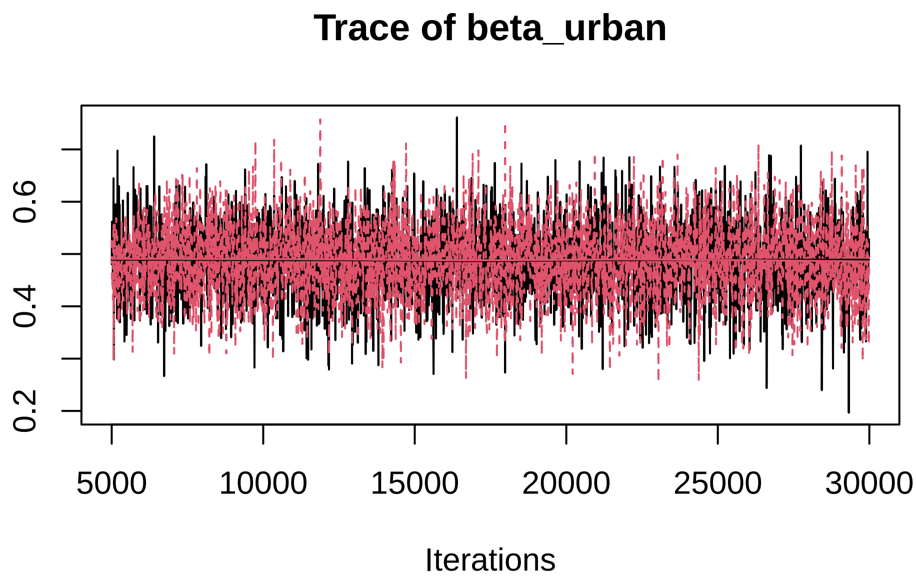
Analisamos os diagnósticos de convergência e não encontramos nenhum indicativo de não convergência. Apresentamos também os gráfico das posteriores dos parâmetros:

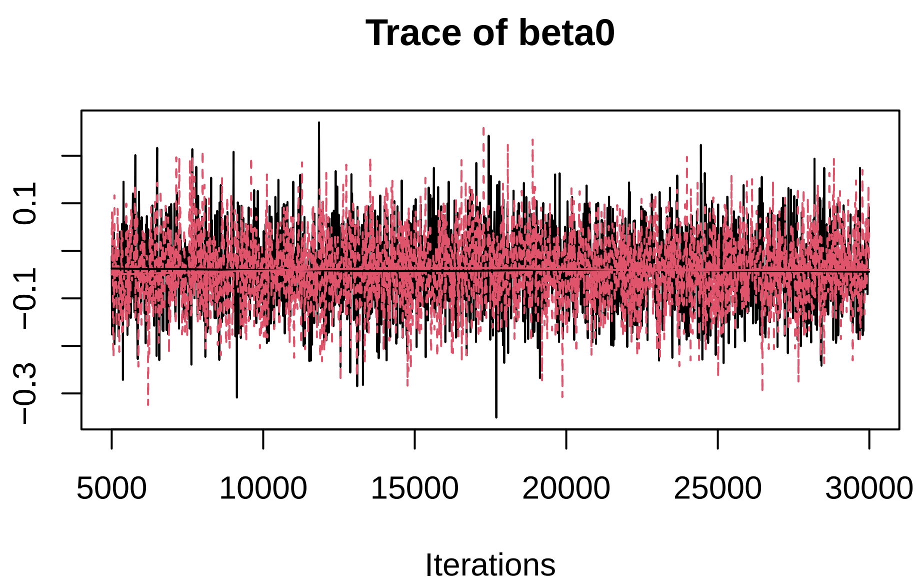
```r
mcmc.samples <- as.mcmc(m1_blogit)
traceplot(mcmc.samples)
```

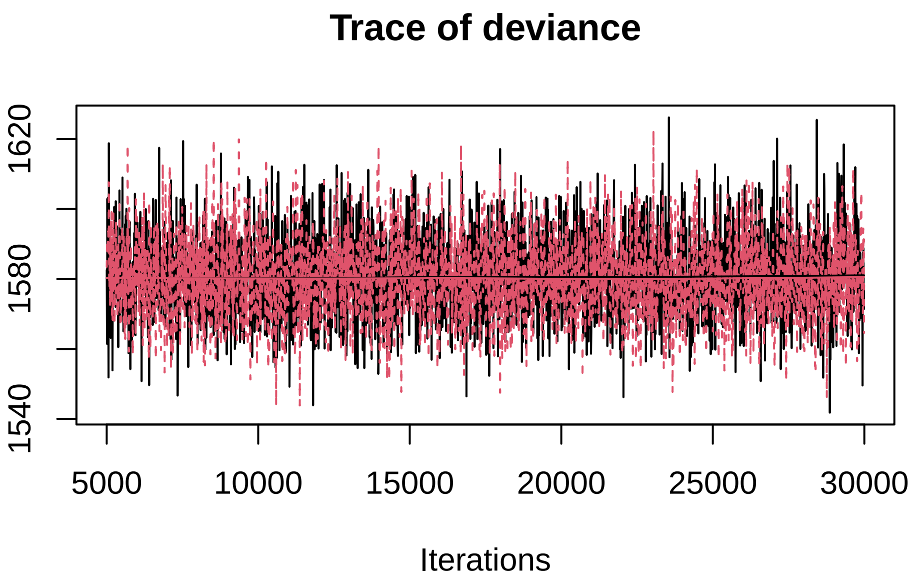


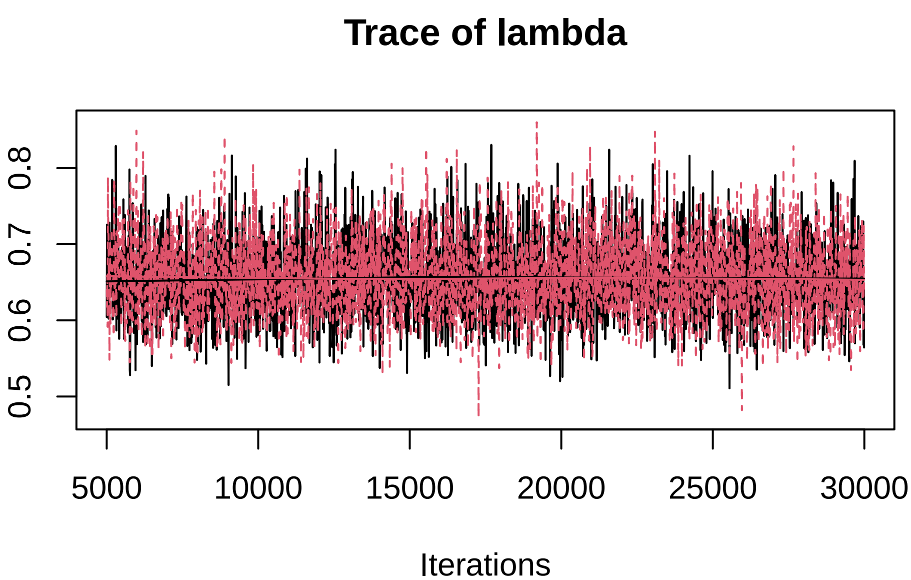

Trace of beta_men
−0.1
−0.3
−0.5
5000
10000
15000
20000
25000
30000
Iterations


Trace of beta_sen
0.60
0.45
0.30
5000
10000
15000
20000
25000
30000
Iterations

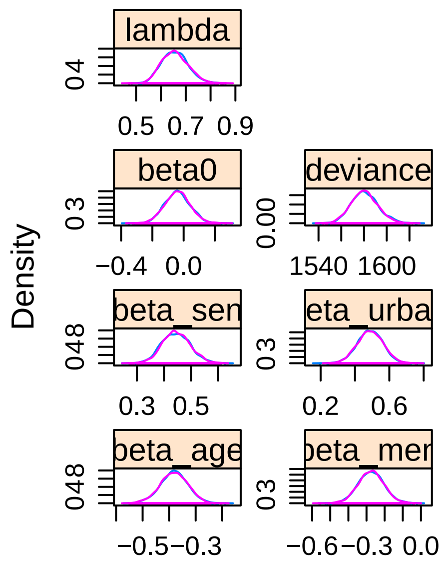

Trace of beta_urban
0.2
0.4
0.6
5000
10000
15000
20000
25000
30000
Iterations


Trace of beta0
−0.3
−0.1
0.1
5000
10000
15000
20000
25000
30000
Iterations






```
densityplot(mcmc.samples)
```



Predição nos conjuntos de treino e teste

```
predict_bcauchit <- function(jagsfit.mcmc, data) {
# data deve ser do tipo dataframe

#vetor da média dos coeficientes beta_age/beta_men/beta_sen/beta_urban
coef <- as.matrix(summary(jagsfit.mcmc)$statistics[, 'Mean'][1:5])
lambda <- as.matrix(summary(jagsfit.mcmc)$statistics[, 'Mean'])[7]


data_matrix <- data.matrix(data)[,c(-1,-4)][, c(3,1,4,2)]
data_matrix <- cbind(data_matrix, intercept = 1)
n <- dim(data_matrix)[1]


#coef * data
res <- t(coef) %*% t(data_matrix)
res_vec <- as.vector(res)


pstar <- 1/pi* atan(-res_vec) + 0.5
```

```r
p <- 1- pstar^lambda

predictions <- rbinom(n, 1, p)

return(predictions)
}



accuracy_bayes <- function(model_roc, test_df, predict) {

  #use roc returned by Epi roc function

  thresh_index <- which.max(rowSums(model_roc$res[, c("sens", "spec")]))

  recall <- model_roc$res[thresh_index,][1,1]
  precision <- model_roc$res[thresh_index,][1,3]

  f1_score <- 2*(recall*precision)/(recall+precision)


  thresh <- as.double(rownames(model_roc$res[thresh_index,][1]))


  acc <- mean(predict == test_df$y)

  return(c(acc,precision, recall,f1_score))

}

pred_test <- predict_bcauchit(mcmc.samples,data_test)


roc_bayes <- Epi::ROC(pred_test, data_test$y, plot= F)


thresh_index <- which.max(rowSums(roc_bayes$res[, c("sens", "spec")]))

recall <- roc_bayes$res[thresh_index,][1,1]
precision <- roc_bayes$res[thresh_index,][1,3]
```

```
metrics_bayes <- accuracy_bayes(roc_bayes, data_test, pred_test )
acc_bayes <- metrics_bayes[1]
prec_bayes <- metrics_bayes[2]
recall_bayes<-metrics_bayes[3]
f1_bayes <- metrics_bayes[4]
```

Predição no conjunto de teste

A seguir,testamos o poder de predição dos modelos loglog com o conjunto de dados completo e reduzido e os outros modelos implementados. Aplicamos novamente a curva ROC e calculos as seguintes métricas:

- **Acurácia**: para mensurar o desempenho médio do modelo na classificação de indivíduos propensos a adquirir seguro completo
- **Precisão**: mensura a proporção de classificações corretas na classe positiva entre todas as classificações positivas
- **Recall**: mensura a proporção de casos corretamente classificados como positivo entre todos os casos que são de fato positivos.
- **F1-Score:** média harmônica entre precisão e recall

Para o caso do seguro, podemos incorrer em erros de falsos positivos(classificar como 1 quando o correto seria 0) ou falsos negativos (classificar como 0 quando o correto seria 1). O primeiro caso ocasionaria a abordagem de clientes com baixa propensão a adquirir o seguro completo, já o segundo resultaria em não abordar clientes com alto potencial de comprar o seguro. Como o segundo tipo de erro é financeiramente mais danoso a uma companhia, usaremos o recall como critério de escolha preferencial para o melhor modelo, uma vez que um valor alto recall significa um baixo número de falsos negativos.

```
# Load the required libraries

# Create a data frame with the data
data <- data.frame(
  Modelo = c("Modelo loglog", "Modelo loglog Reduzido", "Random Forest", "Cauchit Potência
  'Acurácia' = round(c(loglog_acc_test, logitr_acc_test, acc_rf, acc_bayes),3),
  'Precisão' = round(c(loglog_precision_test, logitr_precision_test, precision_forest, pre
  'Recall' = round(c(loglog_recall_test, logitr_recall_test, recall_forest, recall_bayes),

 'F1Score' = round(c(loglog_f1_test, logitr_f1_test, f1_forest, f1_bayes),3)
)
```

```
data
```

```
                    Modelo Acurácia Precisão Recall F1Score
1             Modelo loglog    0.722    0.082  0.884   0.151
2   Modelo loglog Reduzido    0.722    0.082  0.884   0.151
3             Random Forest    0.712    0.801  0.757   0.778
4 Cauchit Potência Inversa    0.666    0.330  0.030   0.055
```

Notamos que os modelos loglog tem desempenhos similares em todas as métricas após o arredondamento. Tomando o recall como referência, esses modelos mostraram-se como os melhores para a predição de indivíduos com maior probabilidade de adquirir o seguro completo. Em segundo lugar, temos o modelo de RandomForest que é o modelo com maior F1-Score, indicando que este é o mais balanceado em relação ao número de falsos positivos e falsos negativos. Por fim, o modelo baysiano teve o pior desempenho em todas as métricas.

Levando estes resultados em conta, podemos sugerir o modelo loglog como o mais apropriado para reduzir falsos negativos e interpretar o impacto dos coeficientes (AGE, PRIVATE,URBAN e etc) na propensão dos indivíduos em adquirir seguro completo. Se a interpretação dos coeficientes não é necessária e é de interesse minimizar tanto falsos positivos quanto negativos, o modelo de RandomForest é mais recomendado.

Bibliografia

1. BÁZAN, J.L.Power and reversal power links for binary regressions: An application for motor insurance policyholders. **Wiley Online Library,** wileyonlinelibrary.com, 23/11/2016
2. **Bayesian Logistic Regression Tutorial**, https://www.flutterbys.com.au/stats/tut/tut10.5b.html, **Acessado por último em** :11/05/2023
3. SHUNG, Koo Ping, **Accuracy, Precision, Recall or F1?**, https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9. **Acessado por último em:** 11/05/2023
4. BROWLEE, Jason, **Tune Machine Learning Algorithms in R (random forest case study)**, https://machinelearningmastery.com/tune-machine-learning-algorithms-in-r/, **Acesso por último em:** 11/05/2023.