

Atividade 2

Rodrigo Malta Esteves

06/06/2021

```
library(readr)
library(tidyr)
library(dplyr)
library(ggplot2)
library(ggpubr)
library(MASS)
library(caret)
```

Questão A

```
dado <- read_csv("C:\\\\Users\\\\malta\\\\Desktop\\\\Pós Graduação\\\\Apredizado Supervisionado I\\\\Atividade 2\\\\da
dado$InMichelin <- as.factor(dado$InMichelin)
dado <- dado[,c(1,3:6)]
head(dado,n=20)
```

```
## # A tibble: 20 x 5
##   InMichelin Food Decor Service Price
##   <fct>      <dbl> <dbl>   <dbl> <dbl>
## 1 0          19    20     19    50
## 2 0          17    17     16    43
## 3 0          23    17     21    35
## 4 1          19    23     16    52
## 5 0          23    12     19    24
## 6 0          18    17     17    36
## 7 1          24    21     22    51
## 8 1          23    22     21    61
## 9 1          27    27     27   179
## 10 0         20    17     19    42
## 11 0         25    26     27    71
## 12 1         23    20     20    50
## 13 1         23    27     23    82
## 14 0         27    25     27    95
## 15 1         23    23     20    51
## 16 0         22    19     22    40
## 17 1         24    25     24    64
## 18 0         17    24     17    52
## 19 0         20    16     21    38
## 20 1         21    18     20    39
```

1) Análise exploratória

```

food_hist <- ggplot(dado,aes(x=Food))+  

  geom_bar()  

decor_hist <- ggplot(dado,aes(x=Decor))+  

  geom_bar()  

service_hist <- ggplot(dado,aes(x=Service))+  

  geom_bar()  

price_hist <- ggplot(dado,aes(x=Price))+  

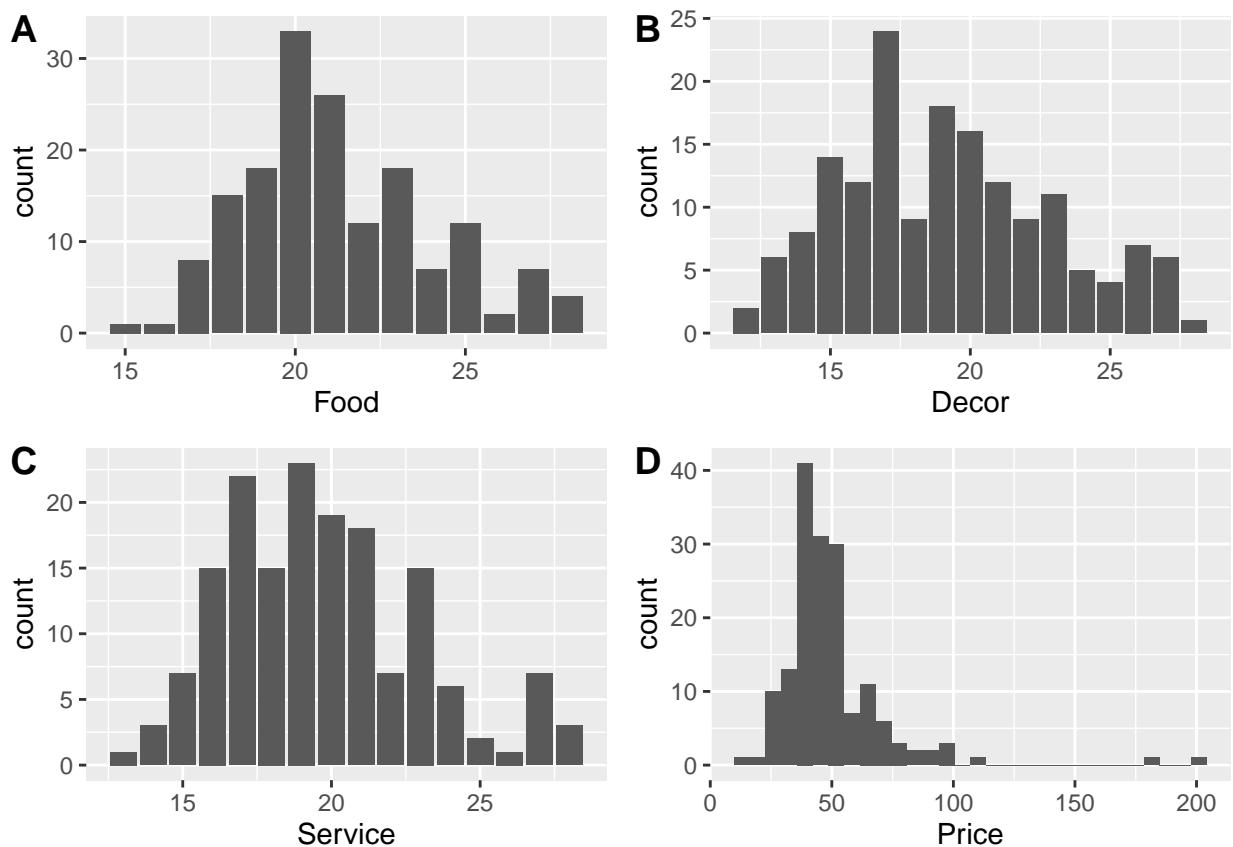
  geom_histogram()  

ggarrange(food_hist,decor_hist,service_hist,price_hist,  

  labels = c("A", "B", "C", "D"),  

  ncol = 2, nrow = 2)

```



```

food_in <- ggplot(dado,aes(x=InMichelin,y=Food))+  

  geom_boxplot()  

decor_in <- ggplot(dado,aes(x=InMichelin,y=Decor))+  

  geom_boxplot()  

service_in <- ggplot(dado,aes(x=InMichelin,y=Service))+  

  geom_boxplot()  

price_in <- ggplot(dado,aes(x=InMichelin,y=Price))+  

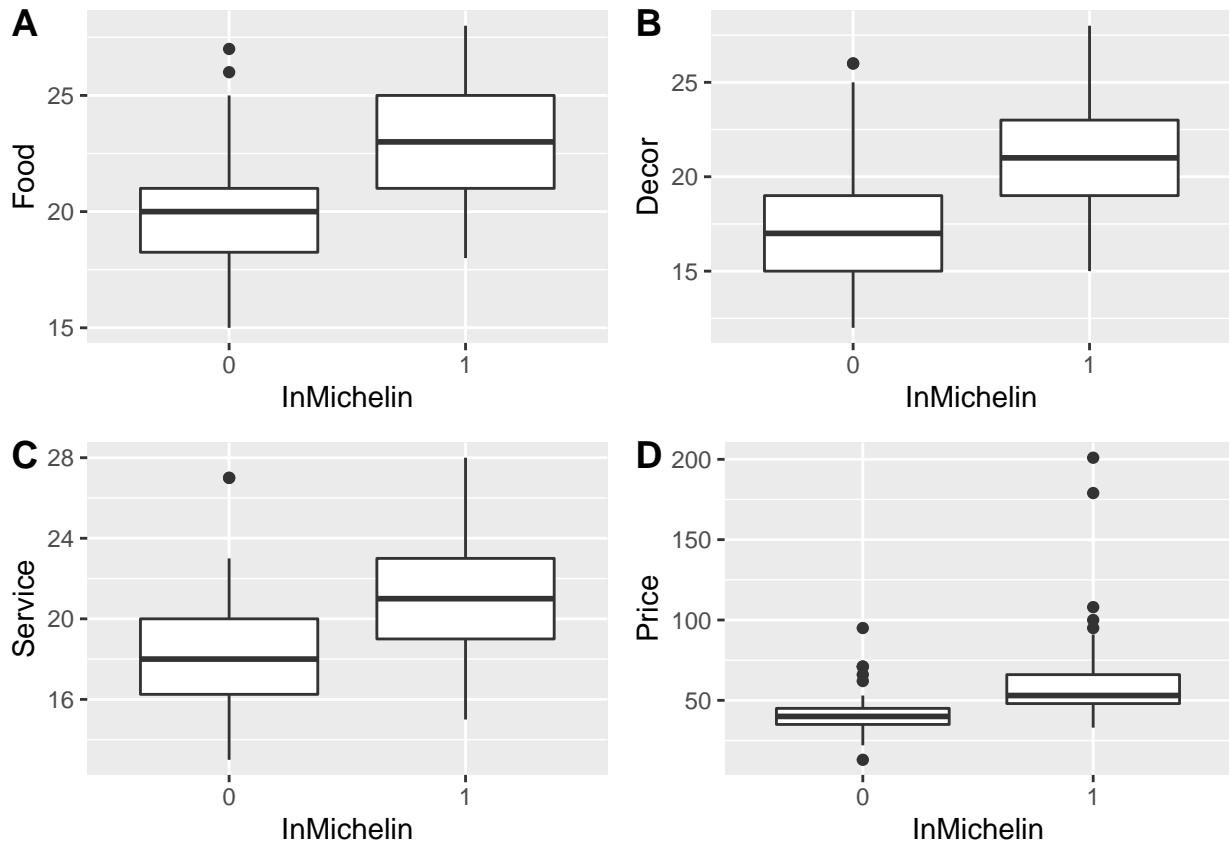
  geom_boxplot()  

ggarrange(food_in,decor_in,service_in,price_in,  

  labels = c("A", "B", "C", "D"),  

  ncol = 2, nrow = 2)

```

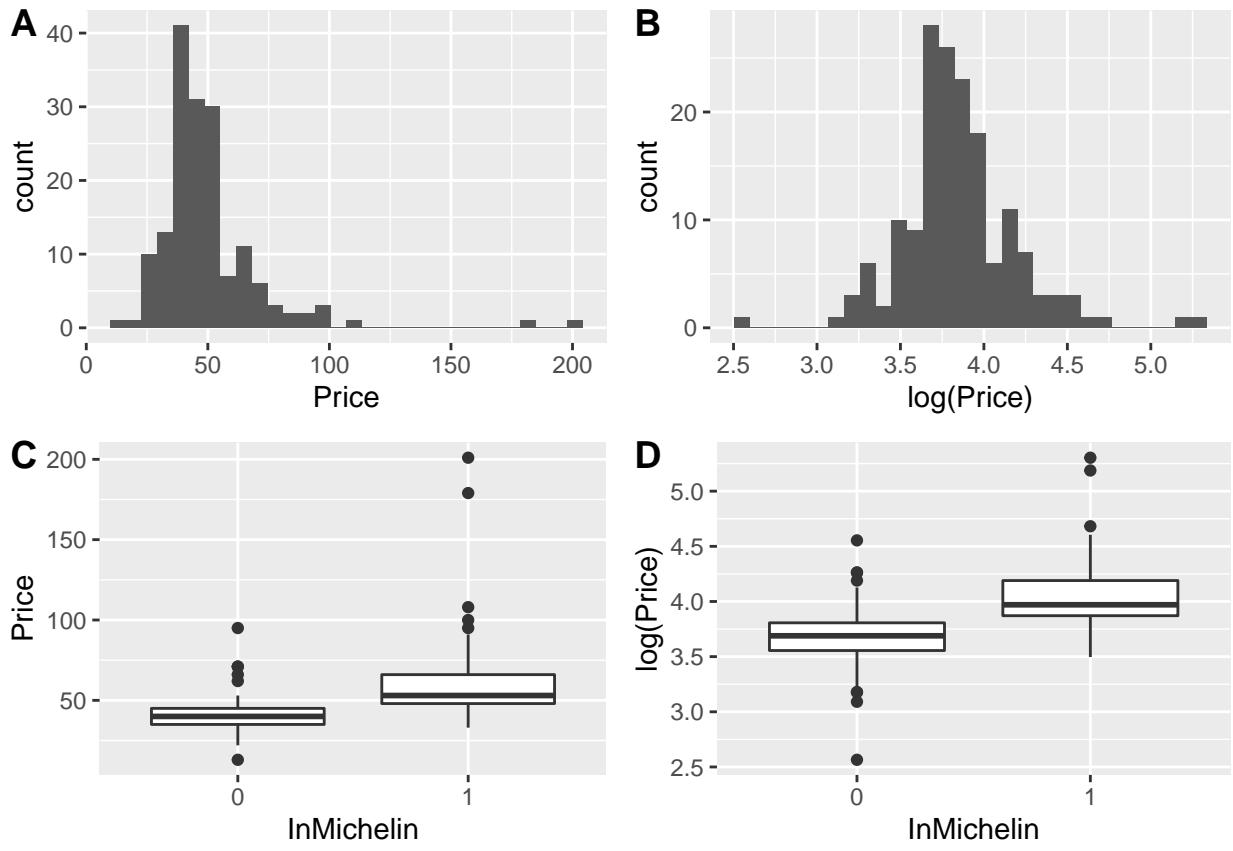


Observando os boxplots, podemos ver que os restaurantes que estão no guia Michelin têm em média notas maiores para todas as covariáveis. No entanto, com o suporte do histograma, a variável “Price” não parece apresentar a mesma simetria das outras três. Seria interessante incluir alguma transformação dessa variável ao modelo.

```
price_hist <- ggplot(dado,aes(x=Price))+
  geom_histogram()
price_hist_log <- ggplot(dado,aes(x=log(Price)))+
  geom_histogram()

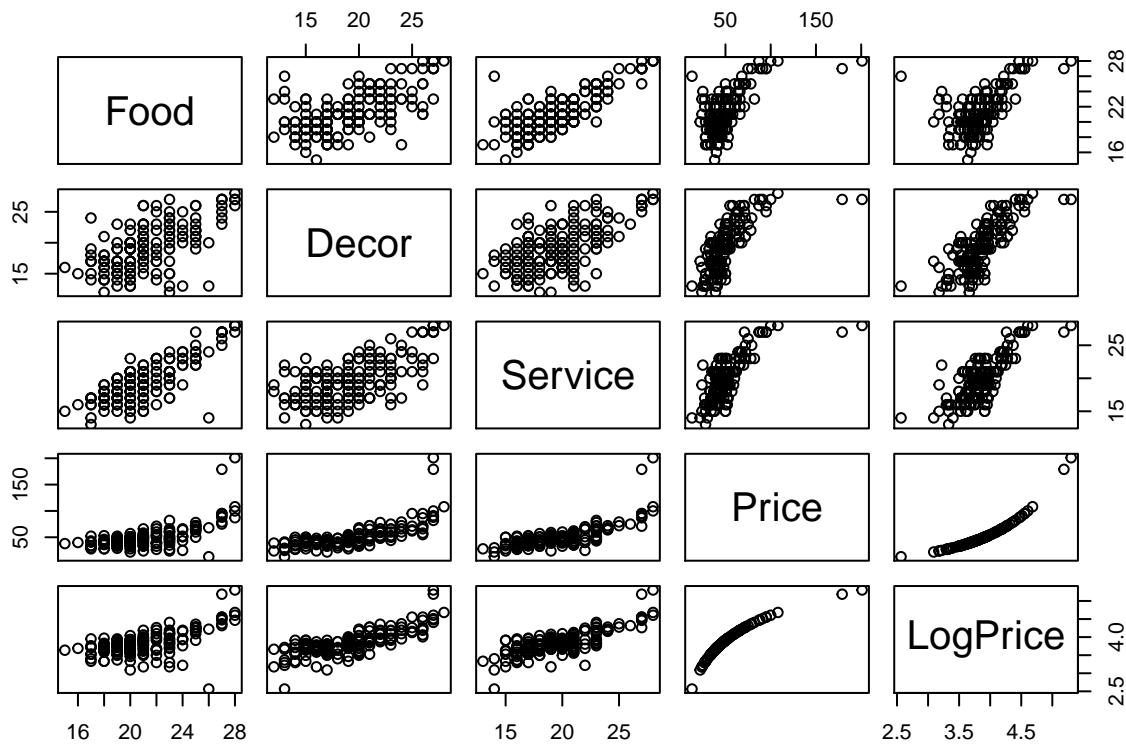
dado$LogPrice <- cbind(log(dado$Price))

g1 <- ggplot(dado,aes(x=InMichelin,y=log(Price)))+
  geom_boxplot()
ggarrange(price_hist,price_hist_log,price_in,g1,
          labels = c("A", "B", "C", "D"),
          ncol = 2, nrow = 2)
```



Observando os gráficos de dispersão e a matriz de correlação vemos que todas as variáveis apresentam alta correlação positiva entre elas. Seria interessante estudar essas relações antes da construção do modelo.

```
plot(dado[2:6])
```



```
round(cor(dado[2:6]),2)
```

```
##          Food  Decor  Service  Price  LogPrice
## Food     1.00   0.61    0.80   0.64    0.61
## Decor    0.61   1.00    0.64   0.71    0.77
## Service  0.80   0.64    1.00   0.73    0.78
## Price    0.64   0.71    0.73   1.00    0.93
## LogPrice 0.61   0.77    0.78   0.93    1.00
```

2) Modelagem

Como estamos tratando uma variável de interesse binária, a distribuição utilizada será de família binomial. Para as funções de ligação testaremos as opções apresentadas em aula e que estão disponíveis no R: ‘logit’, ‘probit’ e ‘cloglog’. Além disso, para a seleção das covariáveis do modelo será utilizada a função `stepAIC()`, onde é selecionado o modelo mais parcimonioso levando em consideração o menor AIC entre eles.

Função de ligação logística

```
m1 <- glm(InMichelin~Food+Decor+Service+Price+
Food:Decor+ Food:Service+ Decor:Service+ Food:Price+ Decor:Price+ Service:Price+
Food:Decor:Service+ Food:Decor:Price+ Food:Service:Price+ Decor:Service:Price+
Food:Decor:Service:Price,family=binomial("logit"),data=dado)
```

```

step.model1 <- stepAIC(m1,direction = "both",trace=FALSE)

m12 <- glm(formula = InMichelin ~ Food + Service +
Food:Service + Decor:Service + Food:Price +
Food:Decor:Service + Food:Service:Price +
Decor:Service:Price + Food:Decor:Service:Price-1, family = binomial("logit"),
data = dado)

summary(m12)

## 
## Call:
## glm(formula = InMichelin ~ Food + Service + Food:Service + Decor:Service +
##       Food:Price + Food:Decor:Service + Food:Service:Price + Decor:Service:Price +
##       Food:Decor:Service:Price - 1, family = binomial("logit"),
##       data = dado)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -2.60674 -0.53573 -0.06122  0.51840  2.22523
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## Food                   -2.445e+00 8.201e-01 -2.982  0.00286 **
## Service                 -2.240e+00 9.254e-01 -2.420  0.01552 *
## Food:Service            1.908e-01 6.004e-02  3.178  0.00148 **
## Service:Decor           1.925e-01 7.009e-02  2.746  0.00603 **
## Food:Price               5.717e-02 1.747e-02  3.272  0.00107 **
## Food:Service:Decor      -7.807e-03 3.118e-03 -2.503  0.01230 *
## Food:Service:Price      -2.198e-03 9.646e-04 -2.278  0.02270 *
## Service:Decor:Price    -2.337e-03 8.173e-04 -2.859  0.00425 **
## Food:Service:Decor:Price 9.090e-05 3.769e-05  2.412  0.01589 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 227.35 on 164 degrees of freedom
## Residual deviance: 119.43 on 155 degrees of freedom
## AIC: 137.43
##
## Number of Fisher Scoring iterations: 6

```

Função de ligação probit

```

m2 <- glm(InMichelin~Food+Decor+Service+Price+
Food:Decor+ Food:Service+ Decor:Service+ Food:Price+ Decor:Price+ Service:Price+
Food:Decor:Service+ Food:Decor:Price+ Food:Service:Price+ Decor:Service:Price+
Food:Decor:Service:Price,family=binomial("probit"),data=dado)

step.model2 <- stepAIC(m2,direction = "both",trace=FALSE)

```

```

m22 <- glm(formula = InMichelin ~ Food + Service +
Food:Service + Decor:Service + Food:Price +
Food:Decor:Service + Food:Service:Price +
Decor:Service:Price + Food:Decor:Service:Price-1, family = binomial("probit"),
data = dado)
summary(m22)

##
## Call:
## glm(formula = InMichelin ~ Food + Service + Food:Service + Decor:Service +
##       Food:Price + Food:Decor:Service + Food:Service:Price + Decor:Service:Price +
##       Food:Decor:Service:Price - 1, family = binomial("probit"),
##       data = dado)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -2.67157 -0.55355 -0.01893  0.53805  2.21145
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## Food                  -1.381e+00  4.374e-01 -3.157 0.001593 **
## Service                -1.263e+00  4.988e-01 -2.532 0.011333 *
## Food:Service            1.072e-01  3.239e-02  3.309 0.000936 ***
## Service:Decor           1.087e-01  3.888e-02  2.795 0.005187 **
## Food:Price              3.241e-02  9.317e-03  3.479 0.000504 ***
## Food:Service:Decor      -4.391e-03  1.715e-03 -2.560 0.010463 *
## Food:Service:Price      -1.238e-03  5.106e-04 -2.424 0.015353 *
## Service:Decor:Price    -1.325e-03  4.585e-04 -2.889 0.003862 **
## Food:Service:Decor:Price 5.123e-05  2.059e-05  2.488 0.012855 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 227.35 on 164 degrees of freedom
## Residual deviance: 119.56 on 155 degrees of freedom
## AIC: 137.56
##
## Number of Fisher Scoring iterations: 7

```

Função de ligação cloglog

```

m3 <- glm(InMichelin~Food+Decor+Service+Price+
Food:Decor+ Food:Service+ Decor:Service+ Food:Price+ Decor:Price+ Service:Price+
Food:Decor:Service+ Food:Decor:Price+ Food:Service:Price+ Decor:Service:Price+
Food:Decor:Service:Price,family=binomial("cloglog"),data=dado)

step.model3 <- stepAIC(m3,direction = "both",trace=FALSE)

m32 <- glm(formula = InMichelin ~ Food + Service +
Food:Service + Decor:Service + Food:Price +

```

```

Food:Decor:Service + Food:Service:Price +
Decor:Service:Price + Food:Decor:Service:Price-1, family = binomial("cloglog"),
  data = dado)
summary(m32)

##
## Call:
## glm(formula = InMichelin ~ Food + Service + Food:Service + Decor:Service +
##       Food:Price + Food:Decor:Service + Food:Service:Price + Decor:Service:Price +
##       Food:Decor:Service:Price - 1, family = binomial("cloglog"),
##       data = dado)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q        Max
## -2.9347   -0.6117   -0.1529    0.5800    2.0984
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## Food                      -1.574e+00 5.056e-01 -3.113  0.00185 **
## Service                   -1.689e+00 6.729e-01 -2.509  0.01209 *
## Food:Service                1.351e-01 4.119e-02  3.281  0.00104 **
## Service:Decor               1.350e-01 4.871e-02  2.772  0.00557 **
## Food:Price                  3.535e-02 1.034e-02  3.418  0.00063 ***
## Food:Service:Decor          -5.622e-03 2.220e-03 -2.533  0.01132 *
## Food:Service:Price           -1.375e-03 5.405e-04 -2.544  0.01096 *
## Service:Decor:Price         -1.471e-03 5.142e-04 -2.861  0.00422 **
## Food:Service:Decor:Price    5.811e-05 2.267e-05  2.563  0.01037 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 247.88 on 164 degrees of freedom
## Residual deviance: 122.81 on 155 degrees of freedom
## AIC: 140.81
##
## Number of Fisher Scoring iterations: 25

```

3) Selecionando e interpretando o melhor modelo

```

anova(m12,m22,m32,test="Chisq")

## Analysis of Deviance Table
##
## Model 1: InMichelin ~ Food + Service + Food:Service + Decor:Service +
##           Food:Price + Food:Decor:Service + Food:Service:Price + Decor:Service:Price +
##           Food:Decor:Service:Price - 1
## Model 2: InMichelin ~ Food + Service + Food:Service + Decor:Service +
##           Food:Price + Food:Decor:Service + Food:Service:Price + Decor:Service:Price +
##           Food:Decor:Service:Price - 1
## Model 3: InMichelin ~ Food + Service + Food:Service + Decor:Service +

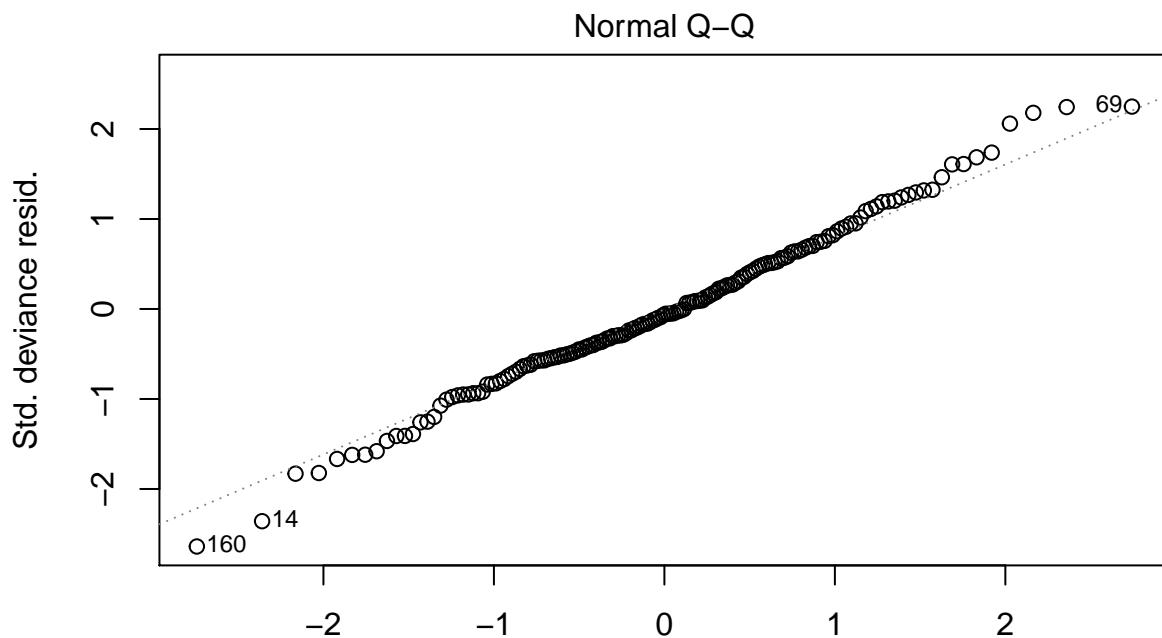
```

```

##      Food:Price + Food:Decor:Service + Food:Service:Price + Decor:Service:Price +
##      Food:Decor:Service:Price - 1
##      Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      155     119.43
## 2      155     119.56  0   -0.1321
## 3      155     122.81  0   -3.2470

plot(m12,which=2)

```



Theoretical Quantiles
 $\text{glm}(\text{InMichelin} \sim \text{Food} + \text{Service} + \text{Food:Service} + \text{Decor:Service} + \text{Food:Price} \dots)$

A comparação entre as deviances aponta para o modelo com função de ligação ‘logit’. O QQPlot é linear para a maior parte dos pontos, com exceção de algumas observações no começo e no final. O ajuste parece bom, mas talvez seja possível melhorá-lo removendo alguns outliers.

Pensando na razão de probabilidade (odds ratio), temos para cada aumento de uma unidade em “Food”, mantendo as outras covariáveis constantes, há um aumento de aproximadamente 0.09 na chance do restaurante estar no guia Michelin. Para cada aumento de unidade em “Service”, mantendo as outras covariáveis constantes, temos 0.10 a mais de chance. Para as interações “Food:Service” e “Service:Decor” temos um aumento de aproximadamente 1.21 e para o restante das interações “Food:Price”, “Food:Service:Decor”, “Food:Service:Price”, “Service:Decor:Price” e “Food:Service:Decor:Price” o aumento é de aproximadamente 1.00 do restaurante estar presente no guia.

```

OR <- exp(m12$coefficients) #odds ratio
OR

```

##	Food	Service	Food:Service
##	0.08668704	0.10650592	1.21022323

```

##          Service:Decor           Food:Price        Food:Service:Decor
##          1.21223832          1.05883859          0.99222329
##  Food:Service:Price  Service:Decor:Price Food:Service:Decor:Price
##          0.99780455          0.99766608          1.00009090

```

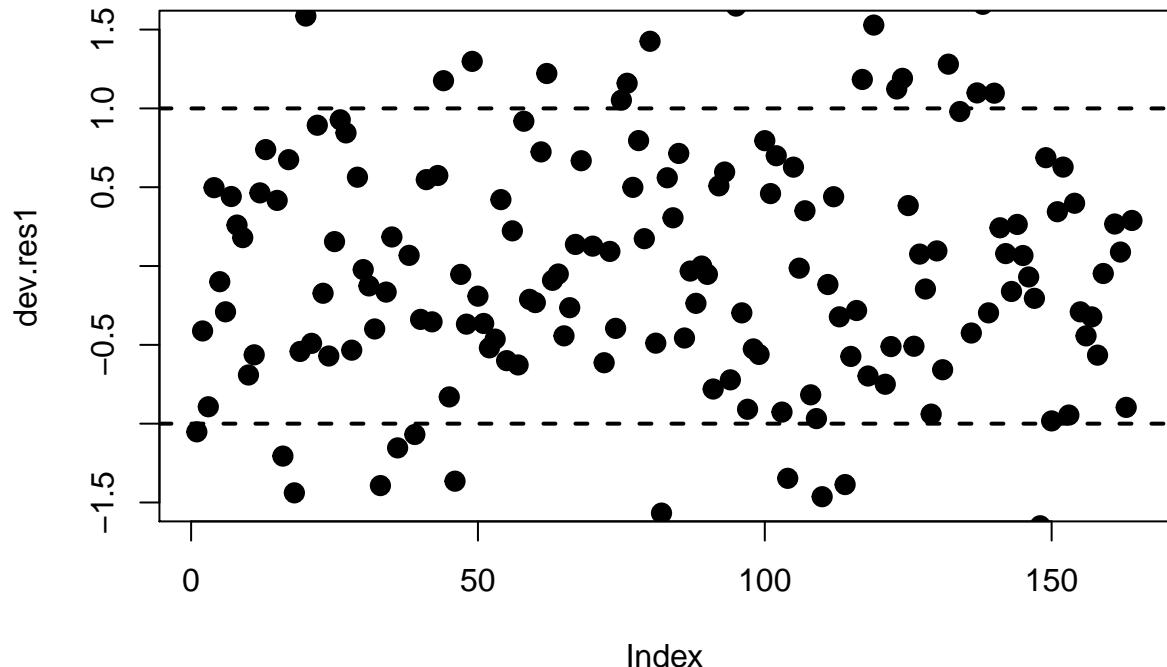
4) Outliers

O gráfico de pontos do valor ajustado versus os resíduos mostra que o modelo possui muitos pontos fora do intervalo (-1,1), indicando existem pontos contribuindo para a falta de ajuste. Além disso, quando observamos o plot de “Residuals vs Leverage” podemos perceber que existe uma grande quantidade de outlier, mas nenhum deles influentes (todos os pontos estão sob as linhas da distância de Cook).

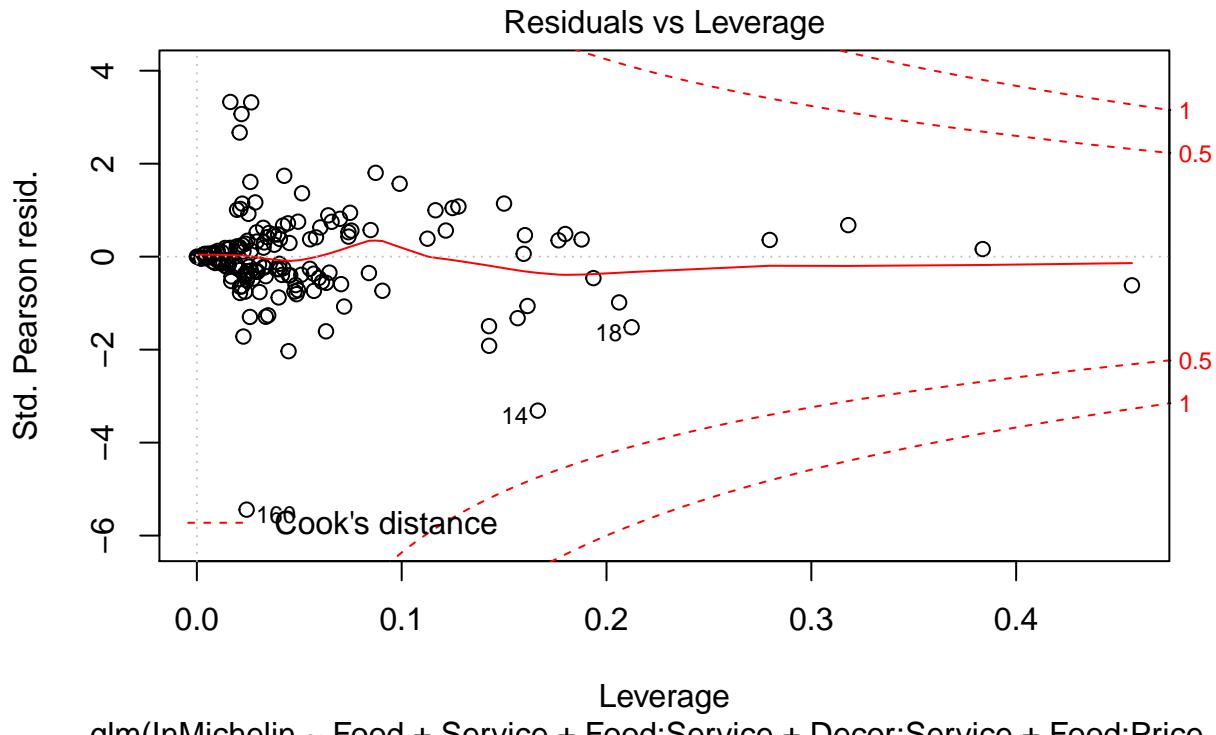
```

dev.res1 = residuals(m12)
plot(dev.res1,pch=20,cex=2,col=1,ylim=c(-1.5,1.5))
abline(h=c(-1,1),lty=2,lwd=2)

```



```
plot(m12,which=5)
```



Se pegarmos os 3 pontos em destaque no plot “Residuals vs Leverage” podemos observar algumas características. A observação 14 possui todas as notas e o preço máximo, mas não está no guia. Claramente um outlier que pode ser removido. A observação 16 também possui valores altos para “Food”, “Decor” e “Service”, mas um valor médio em “Price”. Já a observação 18 possui nota alta apenas em “Decor” e “Price” acima da média.

```
dado[c(14, 16, 18),]
```

```
## # A tibble: 3 x 6
##   InMichelin  Food  Decor  Service Price LogPrice[,1]
##   <fct>      <dbl> <dbl>   <dbl> <dbl>      <dbl>
## 1 0          27     25     27    95      4.55
## 2 0          22     19     22    40      3.69
## 3 0          17     24     17    52      3.95
```

```
dado %>%
  filter(InMichelin == "0") %>%
  summary()
```

```
##   InMichelin      Food        Decor       Service
##   0:90      Min.   :15.00   Min.   :12.00   Min.   :13.00
##   1: 0      1st Qu.:18.25  1st Qu.:15.00  1st Qu.:16.25
##   Median   :20.00   Median :17.00   Median :18.00
##   Mean     :19.94   Mean    :17.31   Mean    :18.36
##   3rd Qu.:21.00  3rd Qu.:19.00  3rd Qu.:20.00
##   Max.     :27.00   Max.    :26.00   Max.    :27.00
```

```

##      Price          LogPrice.V1
##  Min.   :13.00    Min.   :2.564949
##  1st Qu.:35.00   1st Qu.:3.555348
##  Median :40.00   Median :3.688879
##  Mean   :40.73   Mean   :3.672465
##  3rd Qu.:45.00   3rd Qu.:3.806662
##  Max.   :95.00   Max.   :4.553877

```

Questão B

```

dado2 <- read.table("C:\\\\Users\\\\malta\\\\Desktop\\\\Pós Graduação\\\\Apredizado Supervisionado I\\\\Atividade 2")
dado2$y <- as.factor(dado2$y)
head(dado2)

```

```

##   y   Length     Left     Right     Bottom      Top Diagonal
## 1 1 216.2512 130.7076 129.9948 12.517409 10.84310 139.7169
## 2 1 213.4440 131.6444 129.8536 11.297543 10.89005 139.0663
## 3 1 214.8599 131.1769 131.2336 15.969857 10.96258 140.0934
## 4 1 215.1562 130.6063 129.5702 12.670103 10.91676 141.6012
## 5 1 214.3922 129.5446 129.9485  9.856212 11.32475 138.3434
## 6 1 213.3132 129.8321 131.2414  9.724570 11.15741 140.1233

```

1) Gerando dados de treino e teste

```

#Criando dados de treino
set.seed(1234)
training.samples <- dado2$y %>% createDataPartition(p = 0.8, list = FALSE)
train.dt <- dado2[training.samples, ]

#Criando dados de teste
test.dt <- dado2[-training.samples, ]

```

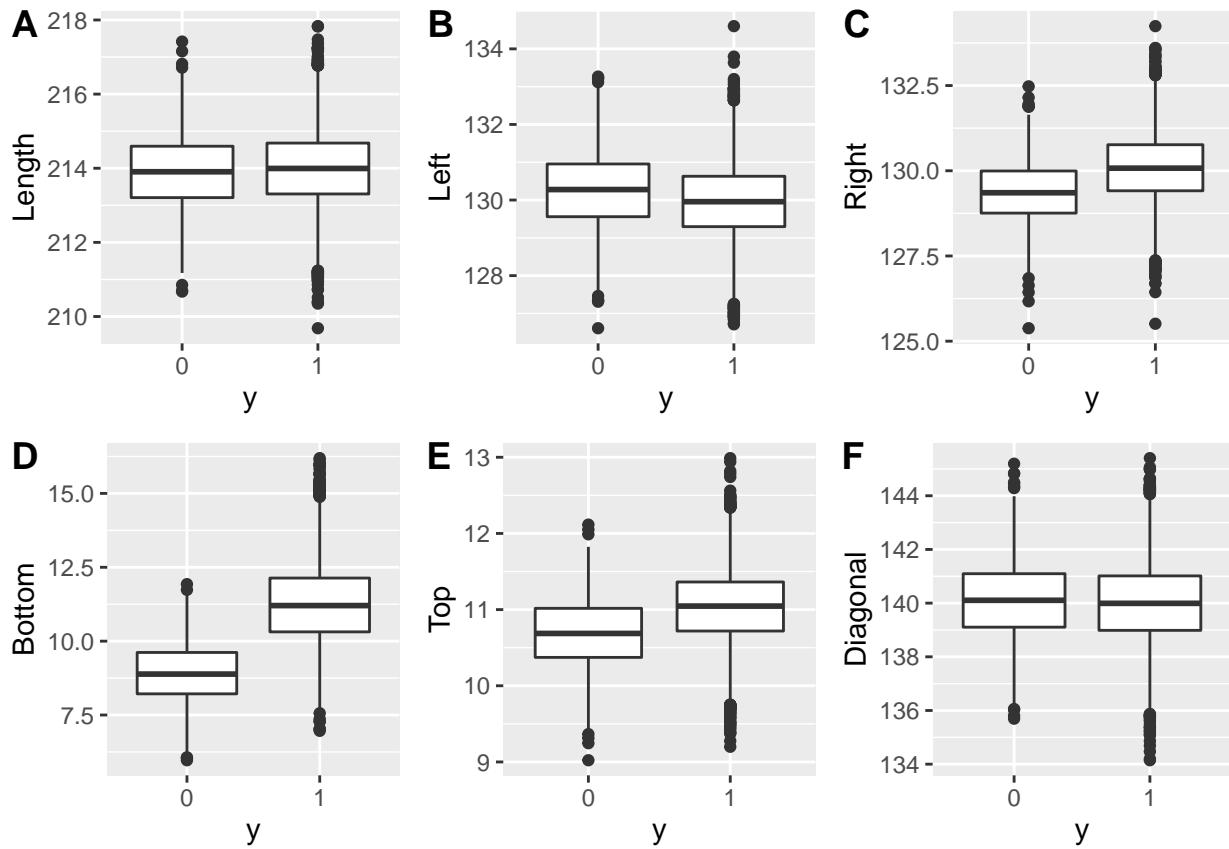
2) Gráficos exploratórios

```

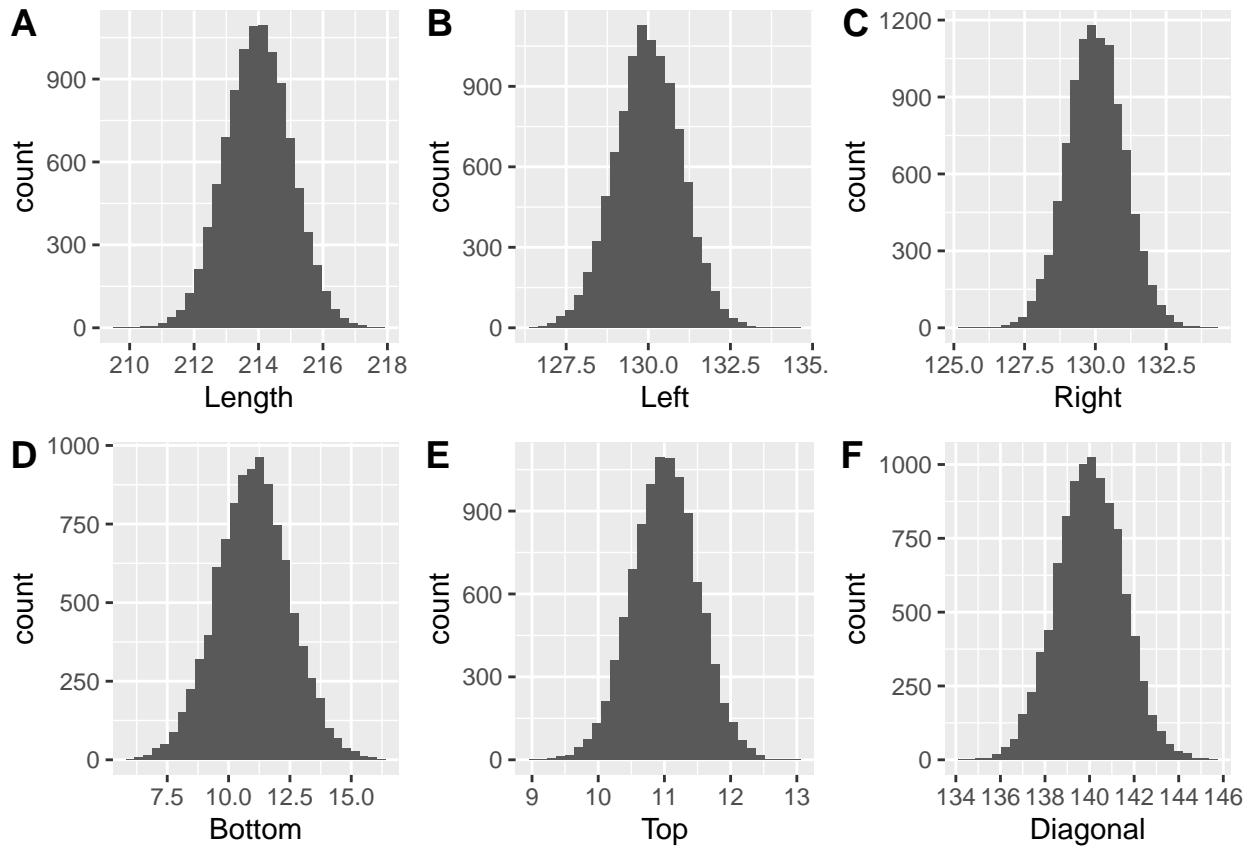
bp1 <- ggplot(dado2,aes(x=y,y=Length))+
  geom_boxplot()
bp2 <- ggplot(dado2,aes(x=y,y=Left))+
  geom_boxplot()
bp3 <- ggplot(dado2,aes(x=y,y=Right))+
  geom_boxplot()
bp4 <- ggplot(dado2,aes(x=y,y=Bottom))+
  geom_boxplot()
bp5 <- ggplot(dado2,aes(x=y,y=Top))+
  geom_boxplot()
bp6 <- ggplot(dado2,aes(x=y,y=Diagonal))+
  geom_boxplot()
ggarrange(bp1,bp2,bp3,bp4,bp5,bp6,

```

```
labels = c("A", "B", "C", "D", "E", "F"),
ncol = 3, nrow = 2)
```



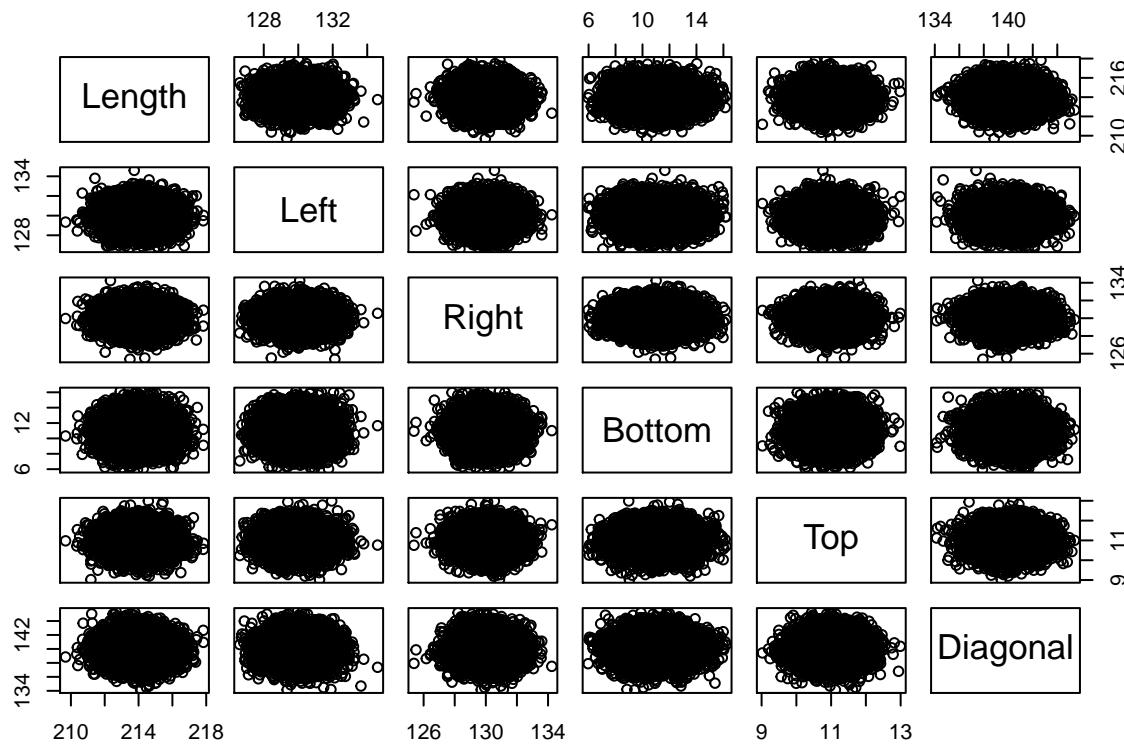
```
hist1 <- ggplot(dado2,aes(x=Length))+
  geom_histogram()
hist2 <- ggplot(dado2,aes(x=Left))+
  geom_histogram()
hist3 <- ggplot(dado2,aes(x=Right))+
  geom_histogram()
hist4 <- ggplot(dado2,aes(x=Bottom))+
  geom_histogram()
hist5 <- ggplot(dado2,aes(x=Top))+
  geom_histogram()
hist6 <- ggplot(dado2,aes(x=Diagonal))+
  geom_histogram()
ggarrange(hist1,hist2,hist3,hist4,hist5,hist6,
          labels = c("A", "B", "C", "D", "E", "F"),
          ncol = 3, nrow = 2)
```



Observando os boxplots podemos ver características distintas entre as notas genuínas e falsas. Apesar de terem “Length” e a “Diagonal” semelhantes, as notas falsas apresentam o tamanho do canto esquerdo (“Left”) maior e o tamanho do canto direito (“Right”), a distância da imagem até o parte de baixo (“Bottom”) e até a parte de cima (“Top”) menores.

Todas as variáveis parecem ter distribuição próxima da normal como evidenciado nos histogramas e não possuem correlação significativa.

```
plot(dado2[2:7])
```



```
cor(dado2[2:7])
```

```
##          Length      Left      Right      Bottom       Top
## Length  1.000000000 0.003152572 0.004567624 0.002245209 0.008988882
## Left    0.003152572 1.000000000 0.014815111 0.002904285 0.009128655
## Right   0.004567624 0.014815111 1.000000000 -0.002217551 0.001698941
## Bottom  0.002245209 0.002904285 -0.002217551 1.000000000 0.011655213
## Top     0.008988882 0.009128655 0.001698941 0.011655213 1.000000000
## Diagonal -0.008230275 0.003024692 -0.017114768 -0.002886119 -0.002697250
##                  Diagonal
## Length  -0.008230275
## Left    0.003024692
## Right   -0.017114768
## Bottom  -0.002886119
## Top     -0.002697250
## Diagonal 1.000000000
```

3) Modelagem

Assim como no exercício anterior, temos uma variável resposta binária. Selecionaremos um modelo `glm()` com família binomial e com uma função de ligação escolhida entre as disponíveis no R: ‘logit’, ‘probit’ e ‘cloglog’. Além disso, para a seleção das covariáveis do modelo será utilizada a função `stepAIC()`, onde é selecionado o modelo mais parcimonioso levando em consideração o menor AIC entre eles.

Função de ligação logística

```
m12 <- glm(y~.,family=binomial("logit"),data=train.dt)
step.model12 <- stepAIC(m12, direction = "both",
                         trace = FALSE)
summary(step.model12)

##
## Call:
## glm(formula = y ~ Length + Left + Right + Bottom + Top, family = binomial("logit"),
##      data = train.dt)
##
## Deviance Residuals:
##       Min      1Q   Median      3Q      Max
## -3.3898  0.0001  0.0016  0.0269  3.2626
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -451.23586  28.83684 -15.648 < 2e-16 ***
## Length       0.36699  0.07805  4.702 2.57e-06 ***
## Left        -1.38227  0.09853 -14.028 < 2e-16 ***
## Right        3.36032  0.16414  20.472 < 2e-16 ***
## Bottom       4.85533  0.21748  22.325 < 2e-16 ***
## Top          6.65060  0.33170  20.050 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 5439.1 on 8000 degrees of freedom
## Residual deviance: 1082.0 on 7995 degrees of freedom
## AIC: 1094
##
## Number of Fisher Scoring iterations: 10
```

Função de ligação probit

```
m22 <- glm(y~.,family=binomial("probit"),data=train.dt)
step.model22 <- stepAIC(m22, direction = "both",
                         trace = FALSE)
summary(step.model22)

##
## Call:
## glm(formula = y ~ Length + Left + Right + Bottom + Top, family = binomial("probit"),
##      data = train.dt)
##
## Deviance Residuals:
##       Min      1Q   Median      3Q      Max
## -3.7350  0.0000  0.0000  0.0041  3.5177
```

```

## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -245.50910  15.12340 -16.234 < 2e-16 ***
## Length       0.20295   0.04260   4.764  1.9e-06 ***
## Left         -0.75165   0.05187 -14.490 < 2e-16 ***
## Right        1.82463   0.08294  22.001 < 2e-16 ***
## Bottom       2.62776   0.10747  24.451 < 2e-16 ***
## Top          3.60568   0.16784  21.482 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 5439.1 on 8000 degrees of freedom
## Residual deviance: 1083.4 on 7995 degrees of freedom
## AIC: 1095.4
##
## Number of Fisher Scoring iterations: 10

```

Função de ligação cloglog

```

m32 <- glm(y~.,family=binomial("cloglog"),data=train.dt)
step.model32 <- stepAIC(m32, direction = "both",
                         trace = FALSE)
summary(step.model32)

## 
## Call:
## glm(formula = y ~ Length + Left + Right + Bottom + Top, family = binomial("cloglog"),
##      data = train.dt)
##
## Deviance Residuals:
##    Min     1Q Median     3Q    Max
## -4.988  0.000  0.000  0.000  2.605
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -248.72382  15.52343 -16.022 < 2e-16 ***
## Length       0.23097   0.04252   5.433 5.56e-08 ***
## Left         -0.75719   0.05226 -14.488 < 2e-16 ***
## Right        1.80815   0.08559  21.125 < 2e-16 ***
## Bottom       2.60386   0.11264  23.116 < 2e-16 ***
## Top          3.58521   0.17413  20.589 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 5439.1 on 8000 degrees of freedom
## Residual deviance: 1123.1 on 7995 degrees of freedom
## AIC: 1135.1

```

```
##  
## Number of Fisher Scoring iterations: 12
```

4) Comparação do poder preditivo

Utilizamos a amostra de teste para fazer previsões fora da amostra. Observando a porcentagem de acerto vemos que todos os três modelos entregam bons resultados, com o modelo 1 sendo marginalmente melhor. Quando observamos as porcentagens de acerto para cada caso percebemos algumas diferenças. Para as notas “Falsa” o modelo 3 com função de ligação “cloglog” tem uma taxa de acerto maior que a dos outros dois modelos, onde o segundo melhor é o modelo 1, porém com uma diferença bem pequena para o modelo 2. Já para as notas “Genuínas” o efeito se inverte. Os modelos 1 e 2 são superiores com taxas iguais e o modelo 3 acerta com probabilidade menor.

```
#Previsão na amostra de teste  
y.fit1 = predict.glm(step.model12,test.dt,type="response")  
y.fit2 = predict.glm(step.model22,test.dt,type="response")  
y.fit3 = predict.glm(step.model32,test.dt,type="response")  
  
classifica =  
  function(probabilidade) {  
    ifelse(probabilidade < 0.5, "Falsa", "Genuína")  
  }  
  
classifica_fit1 = classifica(y.fit1)  
classifica_fit2 = classifica(y.fit2)  
classifica_fit3 = classifica(y.fit3)  
  
tab1 <- table(test.dt$y,classifica_fit1)  
tab2 <- table(test.dt$y,classifica_fit2)  
tab3 <- table(test.dt$y,classifica_fit3)  
  
# % de acertos  
perc_acerto_logit <- (sum(diag(tab1))/sum(tab1)) %>% magrittr::multiply_by(100)  
# % de acertos  
perc_acerto_probit <- (sum(diag(tab2))/sum(tab2)) %>% magrittr::multiply_by(100)  
# % de acertos  
perc_acerto_cloglog <- (sum(diag(tab3))/sum(tab3)) %>% magrittr::multiply_by(100)  
  
cbind(perc_acerto_logit,perc_acerto_probit,perc_acerto_cloglog)  
  
##      perc_acerto_logit perc_acerto_probit perc_acerto_cloglog  
## [1,]        97.34867          97.29865         97.29865  
  
# % de casos "Genuína"  
tbl1 <- table(test.dt$y)  
perc_genuina <- (tbl1["1"] / sum(tbl1)) %>% magrittr::multiply_by(100)  
perc_falsa <- (tbl1["0"] / sum(tbl1)) %>% magrittr::multiply_by(100)  
cbind(perc_genuina,perc_falsa)  
  
##  perc_genuina perc_falsa  
## 1     89.34467   10.65533
```

```

# Especificidade: % de acertos de "Falsa"
specificity1 = tab1[1,1]/(tab1[1,1] + tab1[1,2])
specificity2 = tab2[1,1]/(tab2[1,1] + tab2[1,2])
specificity3 = tab3[1,1]/(tab3[1,1] + tab3[1,2])

perc_acerto_falsa_logit <- specificity1 %>% magrittr::multiply_by(100) %>% round(2)
perc_acerto_falsa_probit <- specificity2 %>% magrittr::multiply_by(100) %>% round(2)
perc_acerto_falsa_cloglog <- specificity3 %>% magrittr::multiply_by(100) %>% round(2)

cbind(perc_acerto_falsa_logit,perc_acerto_falsa_probit,perc_acerto_falsa_cloglog)

```

```

##      perc_acerto_falsa_logit perc_acerto_falsa_probit
## [1,]              84.98                  84.51
##      perc_acerto_falsa_cloglog
## [1,]              86.38

```

```

# Sensitividade: % de acertos de "Genuína"
sensitivity1 <- tab1[2,2]/(tab1[2,1] + tab1[2,2])
sensitivity2 <- tab2[2,2]/(tab2[2,1] + tab2[2,2])
sensitivity3 <- tab3[2,2]/(tab3[2,1] + tab3[2,2])

perc_acerto_genuina_logit <- sensitivity1 %>% magrittr::multiply_by(100)
perc_acerto_genuina_probit <- sensitivity2 %>% magrittr::multiply_by(100)
perc_acerto_genuina_cloglog <- sensitivity3 %>% magrittr::multiply_by(100)

cbind(perc_acerto_genuina_logit,perc_acerto_genuina_probit,perc_acerto_genuina_cloglog)

```

```

##      perc_acerto_genuina_logit perc_acerto_genuina_probit
## [1,]              98.82419                  98.82419
##      perc_acerto_genuina_cloglog
## [1,]              98.60022

```

5) Interpretando

Como a maior taxa de acerto para qualquer caso é do modelo 1 (97.35%), ele será eleito o melhor modelo de previsão. É importante observar que, caso a previsão tivesse como objetivo a identificação específica das notas falsas, em detrimento da identificação das notas verdadeiras, o modelo 3 seria mais indicado.

Do ponto de vista da regressão, buscando o modelo mais parcimonioso, e utilizando a deviance como critério de seleção, o modelo 1 com função de ligação “logit” também parece ser o melhor.

Pensando na razão de probabilidade (odds ratio) temos para cada aumento de uma unidade em “Length”, mantendo as outras covariáveis constantes, temos um aumento de 1.44 vezes na chance da nota ser genuína. Para cada aumento de unidade em “Left”, mantendo as outras covariáveis constantes, temos 0.25 de chance. Para cada aumento em “Right” temos 28.8, para “Bottom” 128.42 e para “Top” 773.25 vezes mais chance de ser genuína.

```
step.model12$coefficients
```

```

## (Intercept)      Length       Left       Right      Bottom       Top
## -451.235864    0.366995   -1.382268    3.360322    4.855334    6.650600

```

```

OR <- exp(step.model12$coefficients[2:6]) #odds ratio
OR

##      Length       Left       Right       Bottom       Top
## 1.4433907  0.2510087 28.7984659 128.4235762 773.2485095

anova(step.model12,step.model22,step.model32,test="Chisq")

## Analysis of Deviance Table
##
## Model 1: y ~ Length + Left + Right + Bottom + Top
## Model 2: y ~ Length + Left + Right + Bottom + Top
## Model 3: y ~ Length + Left + Right + Bottom + Top
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1     7995    1082.0
## 2     7995    1083.4  0   -1.313
## 3     7995    1123.1  0  -39.709

```