

Atividade 2

Deborah Pereira

12/06/2021

```
rm(list=ls())
setwd("C:/Users/debor/OneDrive/Propria - Estudo/Pós-Graduação Ciencia de Dados/Aprendizado supervisionado")
```

Pacotes

```
library(readr)
library(gclus)

## Loading required package: cluster
```

Função de apoio

```
normalizador = function(dado_entrada){
  for(i in 1:ncol(dado_entrada)){
    maximo = max(dado_entrada[,i])
    minimo = min(dado_entrada[,i])
    dif = ifelse(maximo-minimo == 0, 1, maximo-minimo)
    dado_entrada[,i] = (dado_entrada[,i]-minimo)/dif
  }
  return(dado_entrada)
}
```

Leitura dos dados

```
dado = read_csv("dados.csv")
head(dado)

## # A tibble: 6 x 6
##   InMichelin `Restaurant Name` Food Decor Service Price
##   <dbl> <chr>       <dbl> <dbl> <dbl> <dbl>
## 1 0 14 Wall Street     19    20     19    50
## 2 0 212                  17    17     16    43
```

```

## 3      0 26 Seats          23   17   21   35
## 4      1 44                19   23   16   52
## 5      0 A                 23   12   19   24
## 6      0 A.O.C.           18   17   17   36

```

```
summary(dado)
```

```

##   InMichelin  Restaurant Name       Food       Decor
##   Min.    :0.0000  Length:164     Min.    :15.00  Min.    :12.00
##   1st Qu.:0.0000  Class :character  1st Qu.:19.00  1st Qu.:16.00
##   Median :0.0000  Mode   :character  Median :21.00  Median :19.00
##   Mean    :0.4512                    Mean    :21.24  Mean    :19.16
##   3rd Qu.:1.0000                    3rd Qu.:23.00  3rd Qu.:22.00
##   Max.    :1.0000                    Max.    :28.00  Max.    :28.00
##   Service        Price
##   Min.    :13.00  Min.    : 13.0
##   1st Qu.:17.00  1st Qu.: 39.0
##   Median :19.00  Median : 45.0
##   Mean    :19.70  Mean    : 50.1
##   3rd Qu.:21.25  3rd Qu.: 53.0
##   Max.    :28.00  Max.    :201.0

```

1) Construa gráficos exploratórios que ajude a entender a relação entre as variáveis. Interprete.

```

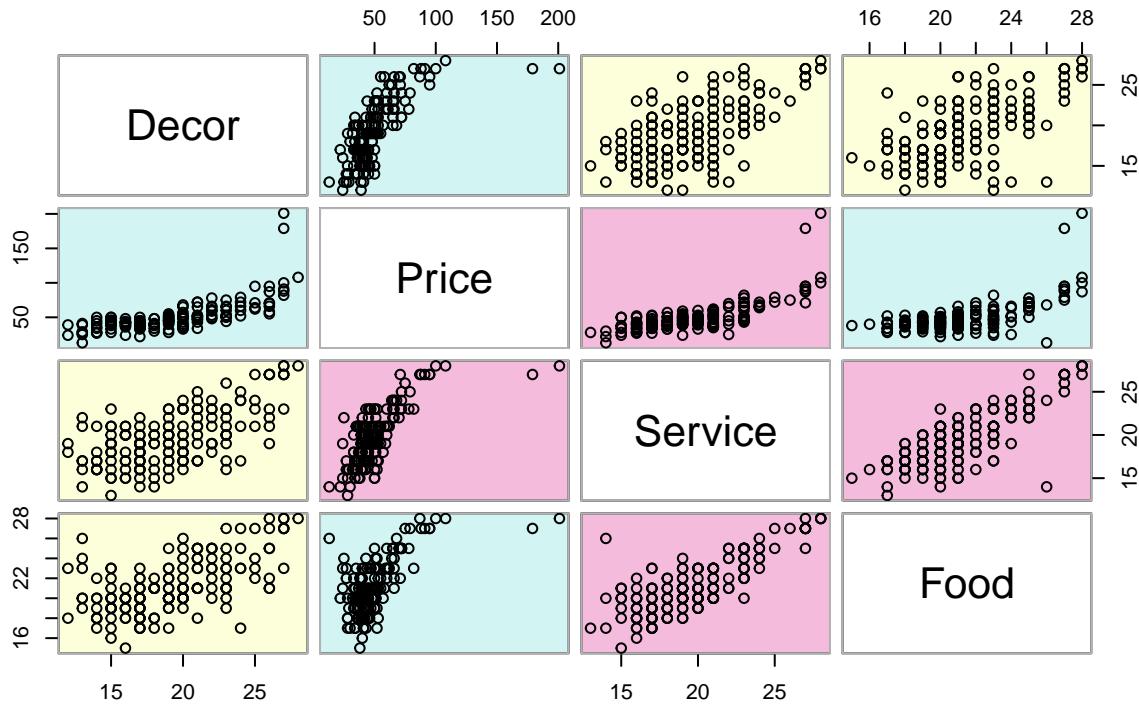
dta = dado[3:6]
dta.r = abs(cor(dta))
cor(dta)

##             Food   Decor   Service   Price
## Food    1.0000000 0.6079806 0.7989272 0.6387275
## Decor   0.6079806 1.0000000 0.6378978 0.7053523
## Service 0.7989272 0.6378978 1.0000000 0.7329377
## Price   0.6387275 0.7053523 0.7329377 1.0000000

dta.col = dmat.color(dta.r)
dta.o = order.single(dta.r)
cpairs(dta, dta.o, panel.colors=dta.col, gap=.5,
       main="Variaveis coloridas pela correlação." )

```

Variáveis coloridas pela correlação.



```
rm(dta); rm(dta.col); rm(dta.r); rm(dta.o)
```

Todas as variáveis possuem correlação positiva de moderada à forte.

O preço possui valores muito maiores do que das outras variáveis, talvez colocando todos em um mesmo range, seja mais fácil analisar.

```
dados_normalizado=normalizador(dado[3:6])

par(mfrow=c(1,3))

boxplot(dados_normalizado
        ,main = "Com todos os restaurantes"
        ,names = colnames(dado[3:6])
        ,ylim = c(0, 1)
)

boxplot(dados_normalizado[dado[["InMichelin"]]==1,]
        ,main = "Com restaurantes InMichelin"
        ,names = colnames(dado[3:6])
        ,ylim = c(0, 1)
)

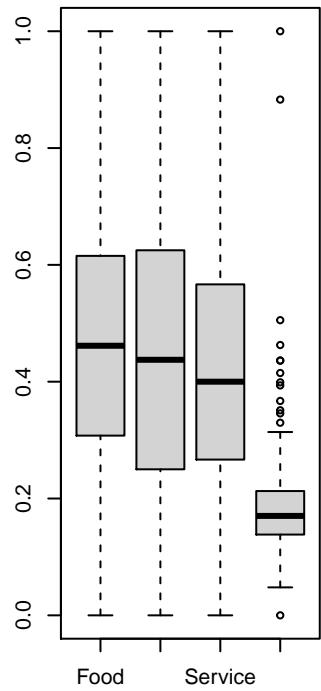
boxplot(dados_normalizado[dado[["InMichelin"]]!=1,]
        ,main = "Com restaurantes não InMichelin"
        ,names = colnames(dado[3:6])
```

```

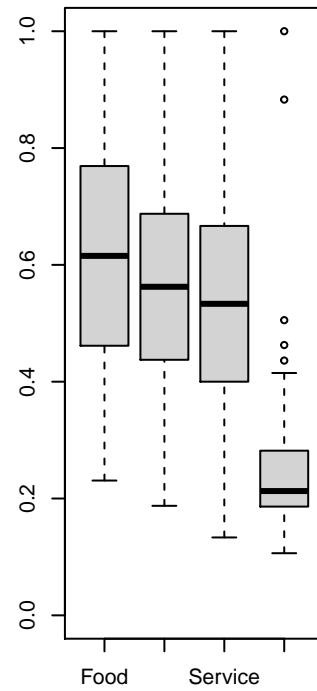
    ,ylim = c(0, 1)
)

```

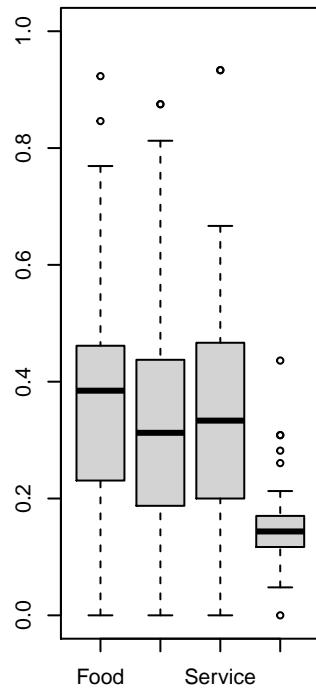
Com todos os restaurantes



Com restaurantes InMichelin



Com restaurantes não InMichelin



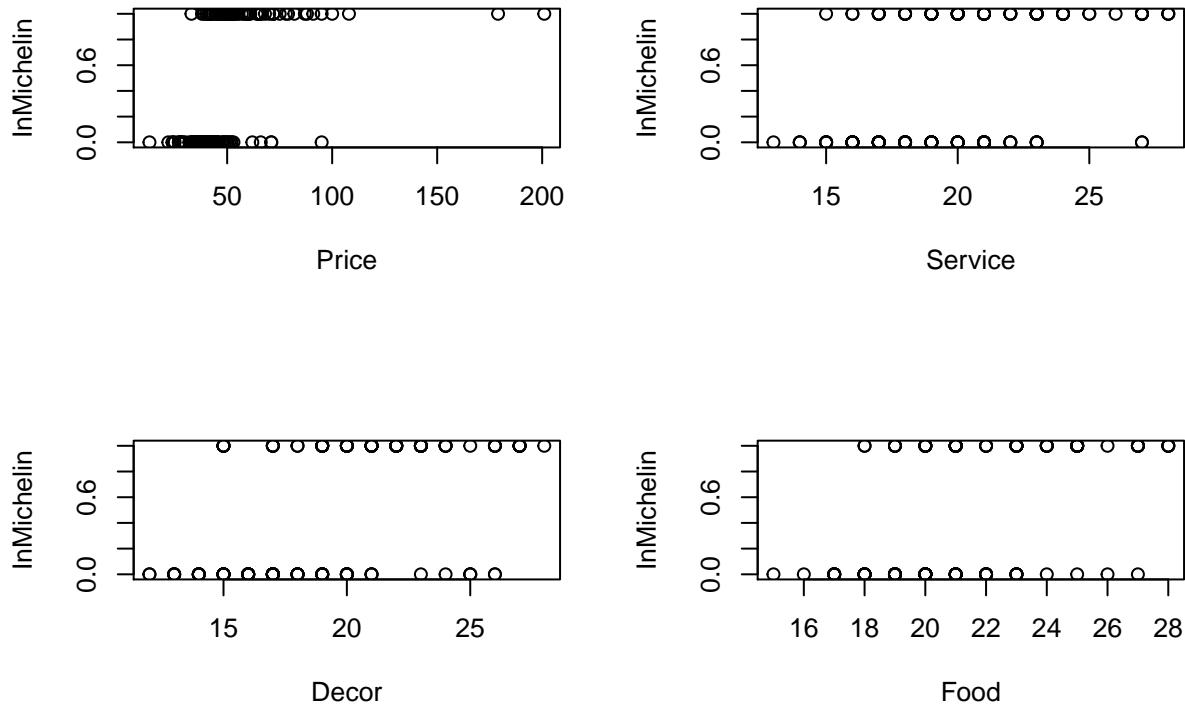
Em preço, existem muitos outliers, em geral as variáveis dos restaurantes “InMichelin” estão com os valores acima da média global.

```

par(mfrow=c(2,2))

plot(dado$Price,dado$InMichelin, ylab = "InMichelin", xlab="Price")
plot(dado$Service,dado$InMichelin, ylab = "InMichelin", xlab="Service")
plot(dado$Decor,dado$InMichelin, ylab = "InMichelin", xlab="Decor")
plot(dado$Food,dado$InMichelin, ylab = "InMichelin", xlab="Food")

```



Desta forma, a visualização do comportamento das variáveis não é tão explícita como o na visualização anterior. A não ser a variável preço, que a mudança de seu comportamento, para restaurantes “InMichel”, continua sendo visível neste gráfico.

2) Ajuste modelos de regressão variando as funções de ligação e covariáveis do modelo. Logit/Probit etc

Probit

```
mod1 = glm(InMichelin ~ Food + Decor + Service + Price
           ,family = binomial(link="probit")
           ,data = dado)
summary(mod1)

##
## Call:
## glm(formula = InMichelin ~ Food + Decor + Service + Price, family = binomial(link = "probit"),
##      data = dado)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -3.4021 -0.7111 -0.3922  0.7928  1.9245
##
```

```

## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.05759   1.20765 -5.016 5.28e-07 ***
## Food         0.22161   0.07171  3.090  0.0020 ** 
## Decor        0.07557   0.05096  1.483  0.1381  
## Service     -0.11318   0.07126 -1.588  0.1122  
## Price        0.04191   0.01734  2.417  0.0157 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 225.79  on 163  degrees of freedom
## Residual deviance: 151.68  on 159  degrees of freedom
## AIC: 161.68
##
## Number of Fisher Scoring iterations: 8

mod2 = glm(InMichelin ~ Food + Price
            ,family = binomial(link="probit")
            ,data = dado)
summary(mod2)

##
## Call:
## glm(formula = InMichelin ~ Food + Price, family = binomial(link = "probit"),
##      data = dado)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.5889 -0.7458 -0.4130  0.8638  1.8618
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.81818   1.13372 -5.132 2.87e-07 ***
## Food         0.16873   0.05875  2.872  0.00408 ** 
## Price        0.04433   0.01158  3.826  0.00013 *** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 225.79  on 163  degrees of freedom
## Residual deviance: 156.82  on 161  degrees of freedom
## AIC: 162.82
##
## Number of Fisher Scoring iterations: 8

```

Do modelo 1 para o modelo 2, foram retiradas as variáveis que foram indicadas como não significativas ao modelo. O AIC do modelo 1 é o melhor.

Logit

```
mod3 = glm(InMichelin ~ Food + Decor + Service + Price
           ,family = binomial(link="logit")
           ,data = dado)
summary(mod3)

##
## Call:
## glm(formula = InMichelin ~ Food + Decor + Service + Price, family = binomial(link = "logit"),
##      data = dado)
##
## Deviance Residuals:
##       Min      1Q   Median      3Q      Max
## -3.3923 -0.6723 -0.3810  0.7169  1.9694
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.19745   2.30896 -4.850 1.24e-06 ***
## Food         0.40485   0.13146   3.080  0.00207 **
## Decor        0.09997   0.08919   1.121  0.26235
## Service     -0.19242   0.12357  -1.557  0.11942
## Price        0.09172   0.03175   2.889  0.00387 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 225.79 on 163 degrees of freedom
## Residual deviance: 148.40 on 159 degrees of freedom
## AIC: 158.4
##
## Number of Fisher Scoring iterations: 6

mod4 = glm(InMichelin ~ Food + Price
           ,family = binomial(link="logit")
           ,data = dado)
summary(mod4)

##
## Call:
## glm(formula = InMichelin ~ Food + Price, family = binomial(link = "logit"),
##      data = dado)
##
## Deviance Residuals:
##       Min      1Q   Median      3Q      Max
## -3.5058 -0.7230 -0.3801  0.7569  1.9214
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.16740   2.24152 -4.982 6.29e-07 ***
## Food         0.31330   0.10892   2.876  0.00402 **
```

```

## Price          0.09317   0.02280   4.086 4.39e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 225.79  on 163  degrees of freedom
## Residual deviance: 152.55  on 161  degrees of freedom
## AIC: 158.55
##
## Number of Fisher Scoring iterations: 6

```

Aqui foi utilizado o mesmo método anterior para exclusão das variáveis no modelo 4. O AIC do modelo 4 é ligeiramente maior do que o do modelo 3.

3) Interprete os coeficientes estimados para o melhor modelo.

Escolhendo o modelo 4, que teve o AIC ligeiramente maior que o do modelo 3, mas mais parcimonioso. Temos que pela razão de chances $e^\beta = 1.36$ para cada ponto adicional da variável comida, a probabilidade de ser um restaurante “InMichelin” aumenta aproximadamente 36%. Já em relação ao preço, com $\beta = 0.09317$, pela razão de chances a cada ponto em que o restaurante fica mais caro, a probabilidade de ser classificado como um restaurante “InMichelin” aumenta aproximadamente 9%.

4) Existem pontos “outliers”? Comente

Sim, a variável preços possui muitos outliers, o que ficou claro com a visualização dos dados já normalizados.

```

rm(list=ls())
setwd("C:/Users/debor/OneDrive/Propria - Estudo/Pós-Graduação Ciencia de Dados/Aprendizado supervisionado")

```

Bibliotecas

```

library(readxl)
library(gclus)
library(caret)
library(erer)

```

Função de apoio

```

normalizador = function(dado_entrada){
  for(i in 1:ncol(dado_entrada)){
    maximo = max(dado_entrada[,i])
    minimo = min(dado_entrada[,i])
    dif = ifelse(maximo-minimo == 0, 1, maximo-minimo)
  }
}

```

```

    dado_entrada[,i] = (dado_entrada[,i]-minimo)/dif
}
return(dado_entrada)
}

```

Leitura dos dados

```

dado <- read.csv("dado2.txt", sep="")
head(dado)

##   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

summary(dado)

##          y            Length           Left           Right
##  Min.   :0.0000   Min.   :209.7   Min.   :126.6   Min.   :125.4
##  1st Qu.:1.0000   1st Qu.:213.3   1st Qu.:129.3   1st Qu.:129.3
##  Median :1.0000   Median :214.0   Median :130.0   Median :130.0
##  Mean   :0.8932   Mean   :214.0   Mean   :130.0   Mean   :130.0
##  3rd Qu.:1.0000   3rd Qu.:214.7   3rd Qu.:130.7   3rd Qu.:130.7
##  Max.   :1.0000   Max.   :217.8   Max.   :134.6   Max.   :134.2
##          Bottom        Top       Diagonal
##  Min.   : 5.970   Min.   : 9.023   Min.   :134.1
##  1st Qu.: 9.993   1st Qu.:10.672   1st Qu.:139.0
##  Median :11.004   Median :11.008   Median :140.0
##  Mean   :11.006   Mean   :11.005   Mean   :140.0
##  3rd Qu.:12.008   3rd Qu.:11.336   3rd Qu.:141.0
##  Max.   :16.193   Max.   :12.988   Max.   :145.4

```

1) Divida os dados em uma amostra de treinamento e outra de teste.

```

set.seed(42)

dado["Aleatorio"] = rbinom(nrow(dado), 1, 0.3)

dado_teste = dado[dado["Aleatorio"]==1,-8]
dado_treino = dado[dado["Aleatorio"]==0,-8]
dado = dado[,-8]

summary(dado_teste)

```

```

##      y      Length      Left      Right
## Min. :0.0000  Min. :210.7  Min. :126.9  Min. :125.4
## 1st Qu.:1.0000 1st Qu.:213.3 1st Qu.:129.4 1st Qu.:129.3
## Median :1.0000 Median :214.0 Median :130.0 Median :130.0
## Mean   :0.8957 Mean  :214.0 Mean  :130.0 Mean  :130.0
## 3rd Qu.:1.0000 3rd Qu.:214.7 3rd Qu.:130.7 3rd Qu.:130.7
## Max.  :1.0000  Max. :217.5  Max. :133.6  Max. :133.6
##      Bottom      Top      Diagonal
## Min. : 6.205  Min. : 9.246  Min. :134.7
## 1st Qu.: 9.965 1st Qu.:10.686 1st Qu.:139.0
## Median :11.024 Median :11.020 Median :140.0
## Mean   :11.028 Mean  :11.011 Mean  :140.0
## 3rd Qu.:12.067 3rd Qu.:11.332 3rd Qu.:141.0
## Max.  :15.919  Max. :12.784  Max. :145.2

```

```
summary(dado_treino)
```

```

##      y      Length      Left      Right
## Min. :0.0000  Min. :209.7  Min. :126.6  Min. :125.5
## 1st Qu.:1.0000 1st Qu.:213.3 1st Qu.:129.3 1st Qu.:129.3
## Median :1.0000 Median :214.0 Median :130.0 Median :130.0
## Mean   :0.8921 Mean  :214.0 Mean  :130.0 Mean  :130.0
## 3rd Qu.:1.0000 3rd Qu.:214.7 3rd Qu.:130.7 3rd Qu.:130.7
## Max.  :1.0000  Max. :217.8  Max. :134.6  Max. :134.2
##      Bottom      Top      Diagonal
## Min. : 5.97  Min. : 9.023  Min. :134.1
## 1st Qu.:10.00 1st Qu.:10.668 1st Qu.:139.0
## Median :11.00 Median :11.004 Median :140.0
## Mean   :11.00 Mean  :11.002 Mean  :140.0
## 3rd Qu.:11.98 3rd Qu.:11.337 3rd Qu.:141.0
## Max.  :16.19  Max. :12.988  Max. :145.4

```

2) Construa gráficos exploratórios que ajude a entender a relação entre as variáveis. Interprete.

```

dta = dado_treino[2:7]
dta.r = abs(cor(dta))
cor(dta)

```

```

##      Length      Left      Right      Bottom      Top
## Length  1.0000000000 -0.005289007  0.005215626  0.0066580923  0.009115382
## Left    -0.005289007  1.0000000000  0.013154812 -0.0023912982  0.009071714
## Right   0.005215626  0.013154812  1.0000000000  0.0027087172  0.001738254
## Bottom  0.006658092 -0.002391298  0.002708717  1.0000000000  0.006046660
## Top     0.009115382  0.009071714  0.001738254  0.0060466602  1.000000000
## Diagonal -0.007537700 -0.001705918 -0.021672609  0.0009108161 -0.005161744
##      Diagonal
## Length  -0.0075377004
## Left    -0.0017059178
## Right   -0.0216726093

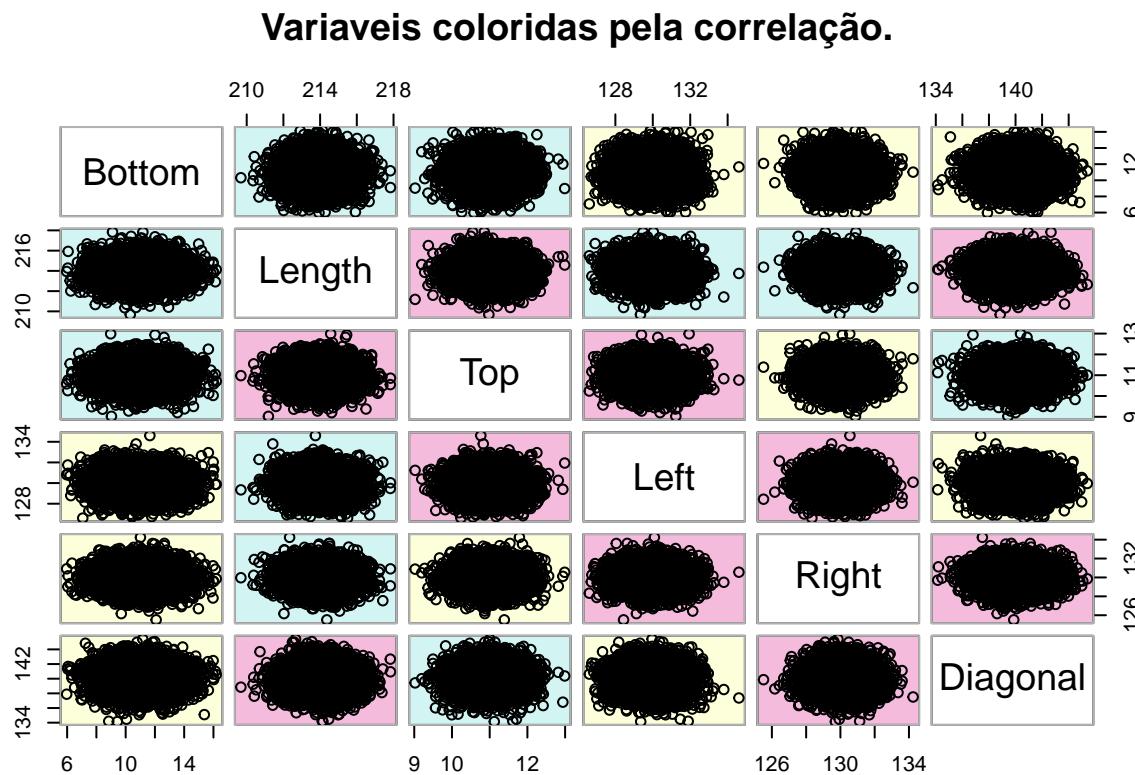
```

```

## Bottom      0.0009108161
## Top        -0.0051617438
## Diagonal   1.00000000000

dta.col = dmat.color(dta.r)
dta.o = order.single(dta.r)
cpairs(dta, dta.o, panel.colors=dta.col, gap=.5,
       main="Variaveis coloridas pela correlação." )

```



```
rm(dta); rm(dta.col); rm(dta.r); rm(dta.o)
```

A correlação entre as variáveis é desprezível.

A diagonal é muito maior do que as outras métricas, para que seja mais fácil analisar, os dados serão normalizados.

```

dado_normalizado=normalizador(dado_treino[2:7])

par(mfrow=c(1,3))

boxplot(dado_normalizado
         ,main = "Boxplot com todos as notas"
         ,names = colnames(dado_treino[2:7])
         ,ylim = c(0, 1)
)

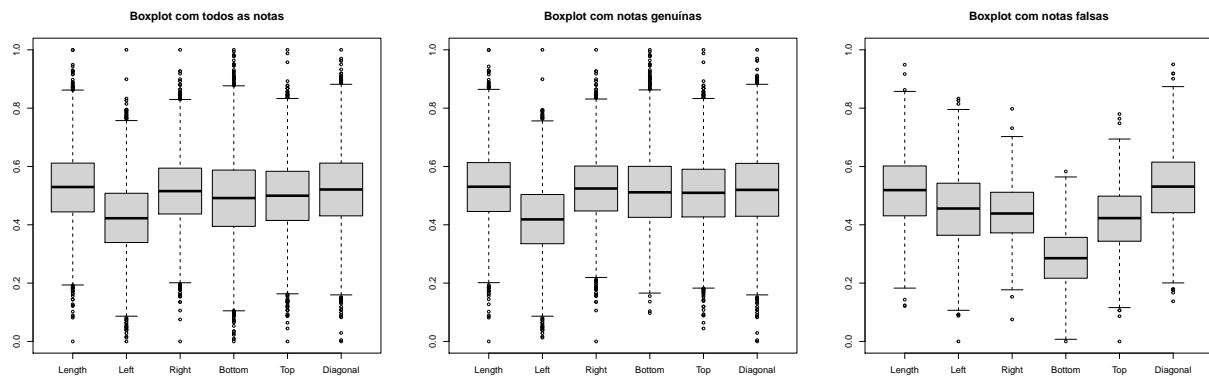
```

```

boxplot(dado_normalizado[dado_treino["y"]==1,]
        ,main = "Boxplot com notas genuínas"
        ,names = colnames(dado_treino[2:7])
        ,ylim = c(0, 1)
)

boxplot(dado_normalizado[dado_treino["y"]!=1,]
        ,main = "Boxplot com notas falsas"
        ,names = colnames(dado_treino[2:7])
        ,ylim = c(0, 1)
)

```



A medida Bottom parece ser a mais forte para indicar se a nota é falsa ou não. Por ser uma pequena parte da população sendo notas falsas, o boxplot com todas as notas genuínas é muito parecido com o de todas as notas.

3) Ajuste modelos de regressão variando as funções de ligação e covariáveis do modelo.

Probit

```

mod1 = glm(y ~ Length + Left + Right + Bottom + Top + Diagonal
            ,family = binomial(link="probit")
            ,data = dado)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(mod1)

##
## Call:
## glm(formula = y ~ Length + Left + Right + Bottom + Top + Diagonal,
##      family = binomial(link = "probit"), data = dado)
##
## Deviance Residuals:

```

```

##      Min       1Q   Median       3Q      Max
## -3.7874   0.0000   0.0000   0.0037   3.5634
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -240.83551   14.25369 -16.896 < 2e-16 ***
## Length       0.19460    0.03841   5.067 4.05e-07 ***
## Left        -0.76528    0.04716 -16.228 < 2e-16 ***
## Right        1.84874    0.07633  24.220 < 2e-16 ***
## Bottom       2.67487    0.09838  27.189 < 2e-16 ***
## Top          3.60903    0.14973  24.103 < 2e-16 ***
## Diagonal     -0.03372    0.02697  -1.251   0.211
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 6795.4 on 9999 degrees of freedom
## Residual deviance: 1334.1 on 9993 degrees of freedom
## AIC: 1348.1
##
## Number of Fisher Scoring iterations: 10

```

```

mod2 = glm(y ~ Length + Left + Right + Bottom + Top
            ,family = binomial(link="probit")
            ,data = dado)

```

```

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

```

```

summary(mod2)

```

```

##
## Call:
## glm(formula = y ~ Length + Left + Right + Bottom + Top, family = binomial(link = "probit"),
##      data = dado)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.7926   0.0000   0.0000   0.0037   3.5531
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -244.83028   13.91304 -17.597 < 2e-16 ***
## Length       0.19349    0.03837   5.043 4.59e-07 ***
## Left        -0.76712    0.04719 -16.255 < 2e-16 ***
## Right        1.84676    0.07627  24.215 < 2e-16 ***
## Bottom       2.67325    0.09828  27.201 < 2e-16 ***
## Top          3.61092    0.14974  24.114 < 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: 6795.4  on 9999  degrees of freedom
## Residual deviance: 1335.7  on 9994  degrees of freedom
## AIC: 1347.7
##
## Number of Fisher Scoring iterations: 10

```

O modelo 2 possui melhor AIC, em relação ao modelo 1, foi retirada a variável Diagonal, que deu como não significativa.

Logit

```

mod3 = glm(y ~ Length + Left + Right + Bottom + Top + Diagonal
           ,family = binomial(link="logit")
           ,data = dado)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(mod3)

##
## Call:
## glm(formula = y ~ Length + Left + Right + Bottom + Top + Diagonal,
##      family = binomial(link = "logit"), data = dado)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -3.4269  0.0000  0.0014  0.0251  3.3001
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -446.37607   27.26151 -16.374 < 2e-16 ***
## Length       0.36507   0.07065   5.167 2.37e-07 ***
## Left        -1.42156   0.09021 -15.757 < 2e-16 ***
## Right        3.42289   0.15185  22.541 < 2e-16 ***
## Bottom       4.97010   0.20095  24.734 < 2e-16 ***
## Top          6.68477   0.29842  22.401 < 2e-16 ***
## Diagonal     -0.06352   0.04966  -1.279   0.201
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 6795.4  on 9999  degrees of freedom
## Residual deviance: 1328.7  on 9993  degrees of freedom
## AIC: 1342.7
##
## Number of Fisher Scoring iterations: 10

```

```

mod4 = glm(y ~ Length + Left + Right + Bottom + Top
           ,family = binomial(link="logit")
           ,data = dado)

```

```

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(mod4)

##
## Call:
## glm(formula = y ~ Length + Left + Right + Bottom + Top, family = binomial(link = "logit"),
##      data = dado)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -3.4297  0.0000  0.0014  0.0251  3.2936
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -453.98771   26.68370 -17.014 < 2e-16 ***
## Length       0.36312   0.07053   5.148 2.63e-07 ***
## Left        -1.42564   0.09031 -15.786 < 2e-16 ***
## Right        3.42022   0.15181  22.530 < 2e-16 ***
## Bottom       4.96700   0.20075  24.742 < 2e-16 ***
## Top          6.68860   0.29848  22.409 < 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: 6795.4 on 9999 degrees of freedom
## Residual deviance: 1330.3 on 9994 degrees of freedom
## AIC: 1342.3
##
## Number of Fisher Scoring iterations: 10

```

Novamente foi retirada a variável diagonal para por não ser significativa ao modelo. O modelo 4 possui o melhor AIC.

4) Faça a previsão fora da amostra de treino e compare os modelos. Comente.

Será comparado os melhores modelos de cada função de ligação.

Matriz de confusão modelo 2:

```

dado_teste$mod2 = predict(mod2
                           ,newdata = dado_teste
                           ,type="response")

summary(dado_teste$mod2)

##      Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.0000  0.9984 1.0000  0.8935 1.0000 1.0000

```

```

#Matriz de confusão

confusionMatrix(data = as.factor(round(dado_teste$mod2,0))
               ,reference = as.factor(dado_teste$y)
               ,positive = "1")

## Confusion Matrix and Statistics
##
##             Reference
## Prediction      0      1
##           0   271    36
##           1    44  2670
##
##                  Accuracy : 0.9735
##                  95% CI : (0.9671, 0.9789)
##      No Information Rate : 0.8957
##      P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.8566
##
## McNemar's Test P-Value : 0.4338
##
##                  Sensitivity : 0.9867
##                  Specificity : 0.8603
##      Pos Pred Value : 0.9838
##      Neg Pred Value : 0.8827
##                  Prevalence : 0.8957
##      Detection Rate : 0.8838
## Detection Prevalence : 0.8984
##      Balanced Accuracy : 0.9235
##
##      'Positive' Class : 1
##

```

Matriz de confusão modelo 4:

```

# Teste
dado_teste$mod4 = predict(mod4
                           ,newdata = dado_teste
                           ,type="response")

summary(dado_teste$mod4)

##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##  0.0000  0.9957  1.0000  0.8933  1.0000  1.0000

#Matriz de confusão

confusionMatrix(data = as.factor(round(dado_teste$mod4,0))
               ,reference = as.factor(dado_teste$y)
               ,positive = "1")

```

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   0      1
##           0 271    37
##           1  44 2669
##
##                 Accuracy : 0.9732
##                 95% CI : (0.9668, 0.9787)
##     No Information Rate : 0.8957
##     P-Value [Acc > NIR] : <2e-16
##
##                 Kappa : 0.855
##
## McNemar's Test P-Value : 0.505
##
##                 Sensitivity : 0.9863
##                 Specificity : 0.8603
## Pos Pred Value : 0.9838
## Neg Pred Value : 0.8799
## Prevalence : 0.8957
## Detection Rate : 0.8835
## Detection Prevalence : 0.8980
## Balanced Accuracy : 0.9233
##
## 'Positive' Class : 1
##

```

A diferença entre os dois modelos não é muito grande, foi predito apenas um a mais de forma errada no modelo 4. Neste caso, a sensibilidade é a melhor métrica para decidirmos qual melhor modelo, pois também é conhecido como “probabilidade de detecção”, então será adotado como melhor modelo, o modelo 2.

5) Interprete os coeficientes estimados para o melhor modelo em termos de ajuste e preditivo.

```

summary(mod2)

##
## Call:
## glm(formula = y ~ Length + Left + Right + Bottom + Top, family = binomial(link = "probit"),
##      data = dado)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -3.7926    0.0000    0.0000    0.0037   3.5531
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -244.83028  13.91304 -17.597 < 2e-16 ***
## Length       0.19349   0.03837   5.043 4.59e-07 ***

```

```

## Left          -0.76712   0.04719 -16.255 < 2e-16 ***
## Right         1.84676   0.07627  24.215 < 2e-16 ***
## Bottom        2.67325   0.09828  27.201 < 2e-16 ***
## Top           3.61092   0.14974  24.114 < 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: 6795.4 on 9999 degrees of freedom
## Residual deviance: 1335.7 on 9994 degrees of freedom
## AIC: 1347.7
##
## Number of Fisher Scoring iterations: 10

```

Todas as variáveis são significativas no modelo. Como esperado pela visualização do boxplot, os coeficientes de Bottom e Top são positivos, o que indica que o aumento no preditor, aumenta a probabilidade de ser uma nota genuína. Já o coeficiente de Left é negativo, o que significa que seu aumento leva uma diminuição na probabilidade de ser uma nota genuína.