Estatística Não Paramétrica - Regressão Não Paramétrica - Trabalho final

Humberto Alves,Fábio Augusto e Davi Ruas

2022-07-15

Introdução

Neste trabalho iremos realizar a análise dos dados de um **portal de apostas esportivas brasileiro** e, em seguida, utilizar técnicas de regressão não paramétricas para **prever a quantidade de novos clientes depositantes** nos sites de apostas parceiros - indicador muito conhecido na indústria como FTD (*First Time Depositors*) - a partir de duas variáveis selecionadas: a **quantidade de cliques nos links dos sites de apostas** e a **quantidade de usuários que visitaram o portal** no mesmo dia.

1- Descrição dos dados

Obtenção dos dados em uma base de dados MySQL

Para obter os dados, foi utilizado, incialmente, um script em Python para acessar a base de dados em MySQL e, então, filtrar e trabalhar as variáveis de interesse:

```
# Libraries
from sqlalchemy import create_engine
import pandas as pd
import access

#SQL Variables
kpis_sql_user = access.kpis_sql_user
kpis_sql_pass = access.kpis_sql_pass
sql_host = access.sql_host
sql_port = access.sql_port

engine = create_engine(f"mysql+pymysql://{kpis_sql_user}:\
  {kpis_sql_pass}@{sql_host}:{sql_port}/kpis")
kpis = pd.read_sql_query("SELECT * FROM bookies_kpis", con=engine)

# Creating Excel file with the relevant variables
kpis = kpis.fillna(value=0)
kpis['total_clicks'] = kpis['bet365_clicks'] + kpis['188bet_clicks'] +
kpis['betfair_clicks'] + kpis['sportsbet_clicks'] +
kpis['betmotion_clicks'] + kpis['pinnacle_clicks']

kpis['total_ftds'] = kpis['bet365_ftd'] + kpis['188bet_ftd'] +
```

```
kpis['betfair_ftd'] + kpis['sportsbet_ftd'] + kpis['betmotion_ftd'] + kpis['pinnacle_ftd']

kpis = kpis[['date', 'total_clicks', 'total_ftds']]

kpis.to_excel('C:/Users/DELL/Meu Drive (betoalvesrocha@gmail.com)/Estatística/3º Período/'
              'Estatística Não Paramétrica/Trabalho Final/ftds_database.xlsx')
```

Apresentação do conjunto de dados

Em seguida, obtivemos o número de usuários que visitaram o portal através de sua plataforma Google Analytics e, no arquivo Excel gerado acima, adicionamos essa nova coluna, resultando no seguinte conjunto de dados:

```
# Bibliotecas
library(gam)
```

```
## Carregando pacotes exigidos: splines
```

```
## Carregando pacotes exigidos: foreach
```

```
## Loaded gam 1.20.2
```

```
library(readxl)
library(glue)
library(ggplot2)

ftds_database <- ftds_database <- read_excel("C:/Users/Davi Ruas/Downloads/ftds_database.xlsx")
print.data.frame(head(ftds_database))
```

```
##         date total_clicks users total_ftds
## 1 2021-04-02          731  6085         35
## 2 2021-04-03          581  4748         15
## 3 2021-04-04          372  5598         10
## 4 2021-04-05          468  6535         22
## 5 2021-04-06          846  6939         67
## 6 2021-04-07          883  6964         43
```

Assim, na exibição das 5 primeiras linhas do conjunto de dados acima, vemos que temos 4 variáveis:

- **date**: A data das observações;
- **total_clicks**: O total de cliques em todos os links de sites de apostas afiliados ao portal;
- **users**: Usuários que acessaram o portal;
- **total_ftds**: Número de novos depositantes totais.

Estatística descritiva

```
summary(ftds_database)
```

```
##       date                       total_clicks        users         total_ftds
##  Min.   :2021-04-02 00:00:00   Min.   : 154.0   Min.   : 2478   Min.   : 2.00
##  1st Qu.:2021-07-26 06:00:00   1st Qu.: 422.2   1st Qu.: 5126   1st Qu.:12.00
##  Median :2021-11-16 12:00:00   Median : 525.0   Median : 6310   Median :17.00
##  Mean   :2021-11-15 20:38:24   Mean   : 639.5   Mean   : 6582   Mean   :18.62
##  3rd Qu.:2022-03-09 18:00:00   3rd Qu.: 707.8   3rd Qu.: 7614   3rd Qu.:23.00
##  Max.   :2022-06-30 00:00:00   Max.   :3315.0   Max.   :13161   Max.   :67.00
```

O resumo estatístico apresentado através da função *summary* nos dá uma noção sobre os dados disponíveis.

As observações vão desde o dia 02 de Abril de 2021 até o último dia de Junho de 2022. O portal possui em média 636 cliques, 6582 usuários e 18,62 novos depositantes por dia.

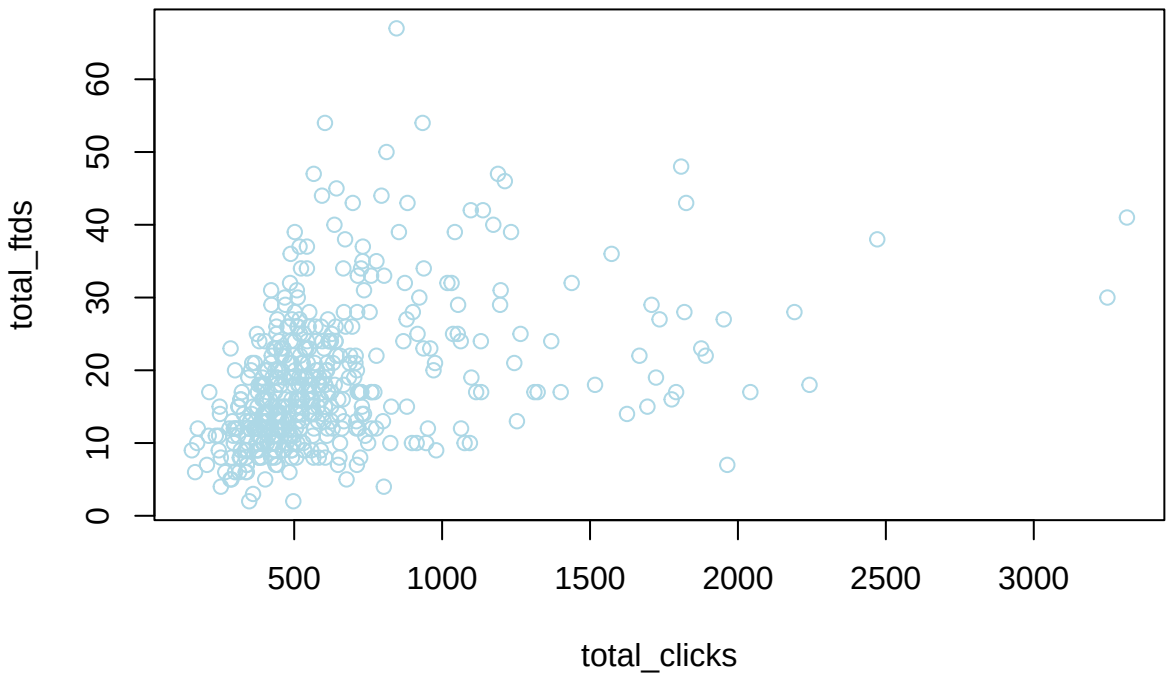
Conhecendo a relação entre variáveis

Como dito na apresentação, o nosso interesse será na variável **total_ftds**, que são os novos clientes depositantes gerados pelo portal para os sites parceiros. E duas variáveis dependentes serão utilizadas neste estudo: **total_clicks e users**.

Vamos dar uma olhada em como essas variáveis se correlacionam:

Cliques x FTDs

```
plot(data.frame(ftds_database['total_clicks'], ftds_database['total_ftds']), col='lightblue')
```



A quantidade de cliques é, naturalmente, a primeira candidata a variável explicativa para os novos depositantes, afinal se o usuário clica no site é porque está de alguma forma interessado em conhecê-lo.

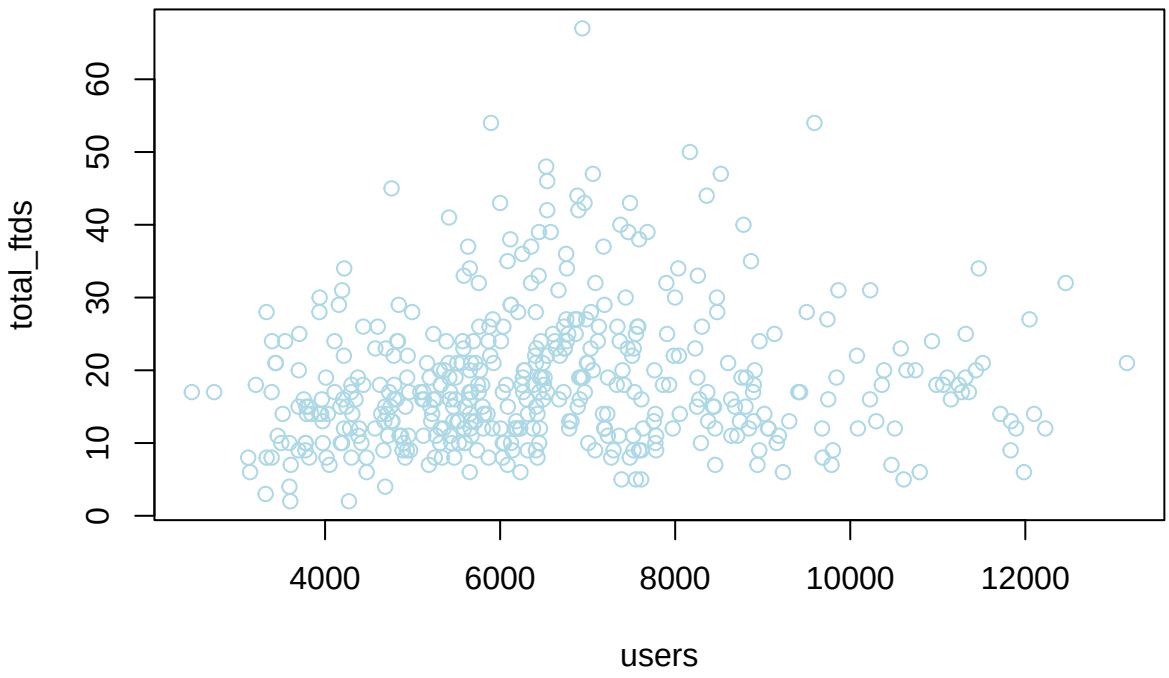
A correlação de Pearson abaixo nos mostra uma relação linear positiva fraca entre as duas variáveis:

```
glue('A correlação de Pearson entre as variáveis é de {round(cor(ftds_database["total_clicks"], ftds_da
```

```
## A correlação de Pearson entre as variáveis é de 0.382
```

Users x FTDs

```
plot(data.frame(ftds_database['users'], ftds_database['total_ftds']), col='lightblue')
```



A próxima varíavel é a quantidade de usuários visitantes. Essa variável pode ser interessante pois, obviamente, quanto mais usuários, tende-se a ter mais cliques e, com isso, mais depositantes. Mas, muito além disso, como o site possui uma seção de prognósticos para as principais partidas de futebol, entende-se que quanto mais usuários maior é o interesse pelas apostas esportivas naquele dia e, com isso, tentar refletir um pouco a "empolgação" da comunidade de apostadores naquele dia.

Diferente da primeira variável, não há uma relação linear clara entre essas duas variáveis:

```
glue('A correlação de Pearson entre as variáveis é de {round(cor(ftds_database["users"], ftds_database[
```

```
## A correlação de Pearson entre as variáveis é de 0.1175
```

2- Ajuste de Modelos

Modelo de regressão linear multipla

Padronização dos dados:

```
#Variaveis:
total_ftds <- ftds_database["total_ftds"]
total_clicks <- ftds_database["total_clicks"]
users <- ftds_database["users"]
```

```r
#Padronização dos dados:
total_ftds_padronizado<- (ftds_database$total_ftds - mean(ftds_database$total_ftds,
                        na.rm = TRUE)) / sd(ftds_database$total_ftds,
                                            na.rm = TRUE)

total_clicks_padronizado<-(ftds_database$total_clicks -
                             mean(ftds_database$total_clicks,
                            na.rm = TRUE))/sd(ftds_database$total_clicks,
                            na.rm = TRUE)

users_padronizado <- (ftds_database$users -
                       mean(ftds_database$users,
                            na.rm = TRUE))/sd(ftds_database$users, na.rm = TRUE)
```

Verificação de normalidade da variavel resposta:

```r
shapiro.test(total_ftds_padronizado)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  total_ftds_padronizado
## W = 0.91193, p-value = 1.635e-15
```

Como p-valor é menor do que 0.05 não podemos aceitar a hipotese de que a variavel resposta tem distribuição normal.

Modelo de regressão multipla:

```r
REG <- lm(total_ftds_padronizado ~ total_clicks_padronizado + users_padronizado)
summary(REG)
```

```
##
## Call:
## lm(formula = total_ftds_padronizado ~ total_clicks_padronizado +
##     users_padronizado)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6412 -0.6176 -0.1471  0.4217  4.8536
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               6.717e-17  4.338e-02   0.000   1.0000
## total_clicks_padronizado 3.785e-01  4.345e-02   8.711   <2e-16 ***
## users_padronizado         1.049e-01  4.345e-02   2.414   0.0162 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.9202 on 447 degrees of freedom
## Multiple R-squared:  0.1569, Adjusted R-squared:  0.1532
## F-statistic:  41.6 on 2 and 447 DF,  p-value: < 2.2e-16
```

Podemos fazer o teste F-Fisher para verificar a hipótese:

H0: Nenhuma das variáveis explicativas (**total_clicks** e **users**) está relacionada à variável resposta (o que significa que Beta1 e Beta2 são iguais a zero)

H1: Pelo menos uma das variáveis explicativas está relacionada à variável resposta (o que significa que Beta1 ou Beta2ou ambos são diferentes de zero).

Como o valor-p para o teste F (<2.2e-16) é menor que 0,05, podemos dizer que pelo menos uma das duas variáveis é significativa.

Podemos também fazer o teste t-Student para verificar a hipótese: H0: Betai = 0 , i = 1 ou 2 H1: Betai != 0 , i = 1 ou 2 Como o valor-p para o teste t-Student, tanto para **total_clicks** (<2e-16) quanto para **users** (0.0162) são menores que 0,05, podemos dizer que as duas variáveis são significativas.

Modelo de regressão aditivo generalizado usando o Lowess:

```
o=order(total_clicks_padronizado)
n=length(total_clicks_padronizado)
SQRes=NULL

mdat= matrix(c(total_clicks_padronizado[o],total_ftds_padronizado[o]), nrow = n, ncol = 2)

for(i in 4:8){
  residuo = total_ftds_padronizado[o] - lowess(mdat, f = i/10, iter=0)$y
 SQRes[i-3]=sum(residuo^2)
}

lambdas=c(.4,.5,.6,.7,.8)
cbind(lambdas,SQRes)
```

```
##      lambdas    SQRes
## [1,]     0.4 347.4251
## [2,]     0.5 348.2333
## [3,]     0.6 348.7321
## [4,]     0.7 349.1846
## [5,]     0.8 349.6769
```

```
#Melhor valor de lambda = 0.4

MAGL=gam(total_ftds_padronizado ~ lo(total_clicks_padronizado,span=0.4)+users_padronizado)
summary(MAGL)
```

```
##
## Call: gam(formula = total_ftds_padronizado ~ lo(total_clicks_padronizado,
##     span = 0.4) + users_padronizado)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.0831 -0.5370 -0.1081  0.4610  4.4704
##
## (Dispersion Parameter for gaussian family taken to be 0.7661)
##
##     Null Deviance: 449 on 449 degrees of freedom
## Residual Deviance: 338.3962 on 441.724 degrees of freedom
## AIC: 1167.333
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                                           Df Sum Sq Mean Sq F value
## lo(total_clicks_padronizado, span = 0.4)   1.00  65.53  65.526  85.534
## users_padronizado                          1.00   7.94   7.944  10.370
## Residuals                                441.72 338.40   0.766
##                                             Pr(>F)
## lo(total_clicks_padronizado, span = 0.4) < 2.2e-16 ***
## users_padronizado                         0.001376 **
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                                          Npar Df Npar F    Pr(F)
## (Intercept)
## lo(total_clicks_padronizado, span = 0.4)     5.3 9.9318 2.26e-09 ***
## users_padronizado
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como a Anova para efeitos paramétricos indicou valor-p = 0.001376, **users** é significativa. Como o a Anova para efeitos não-paramétricos indicou valor-p = 2.26e-09, **total_clicks** é significativa.

Modelo de regressão aditivo generalizado usando o B-Spline:

```
require(splines)
fitdf3 <- lm(total_ftds_padronizado ~ bs(total_clicks_padronizado, df = 3))# 3 nós
AIC(fitdf3)
```

```
## [1] 1172.974
```

```
fitdf4 <- lm(total_ftds_padronizado ~ bs(total_clicks_padronizado, df = 4))# 4 nós
AIC(fitdf4)
```

```
## [1] 1174.705
```

```
fitdf5 <- lm(total_ftds_padronizado ~ bs(total_clicks_padronizado, df = 5))# 5 nós
AIC(fitdf5)
```

```
## [1] 1175.346
```

```r
#Melhor: 3 nós

MAGS=gam(total_ftds_padronizado ~ s(total_clicks_padronizado,3)+users_padronizado)
summary(MAGS)
```

```
##
## Call: gam(formula = total_ftds_padronizado ~ s(total_clicks_padronizado,
##     3) + users_padronizado)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1742 -0.5584 -0.1381  0.4179  4.5378
##
## (Dispersion Parameter for gaussian family taken to be 0.773)
##
##     Null Deviance: 449 on 449 degrees of freedom
## Residual Deviance: 344.0054 on 444.9998 degrees of freedom
## AIC: 1168.179
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                                Df Sum Sq Mean Sq F value    Pr(>F)
## s(total_clicks_padronizado, 3)   1  65.53  65.526  84.763 < 2.2e-16 ***
## users_padronizado                1   6.67   6.669   8.627  0.003484 **
## Residuals                      445 344.01   0.773
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                                Npar Df Npar F     Pr(F)
## (Intercept)
## s(total_clicks_padronizado, 3)       2 22.334 5.706e-10 ***
## users_padronizado
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como a Anova para efeitos paramétricos indicou valor-p=0.003484, **users** é significativa. Como o a Anova para efeitos não-paramétricos indicou valor-p= 5.706e-10, **total_clicks** é significativa.

3- Comparação de modelos

Comparando os valores de AIC dos 3 modelos:

```r
AICREG=AIC(REG)
AICMAGL=AIC(MAGL)
AICMAGS=AIC(MAGS)
cbind(AICREG,AICMAGL,AICMAGS)
```

```
##        AICREG  AICMAGL  AICMAGS
## [1,] 1207.227 1167.333 1168.179
```

Melhor modelo: MAG com B-spline(MAGS), pois apresentou o menor AIC (1168.179).

4- ANÁLISE DE RESÍDUOS

```
shapiro.test(REG$res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  REG$res
## W = 0.93064, p-value = 1.348e-13
```

```
shapiro.test(MAGL$res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  MAGL$res
## W = 0.95342, p-value = 1.035e-10
```

```
shapiro.test(MAGL$res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  MAGL$res
## W = 0.95342, p-value = 1.035e-10
```

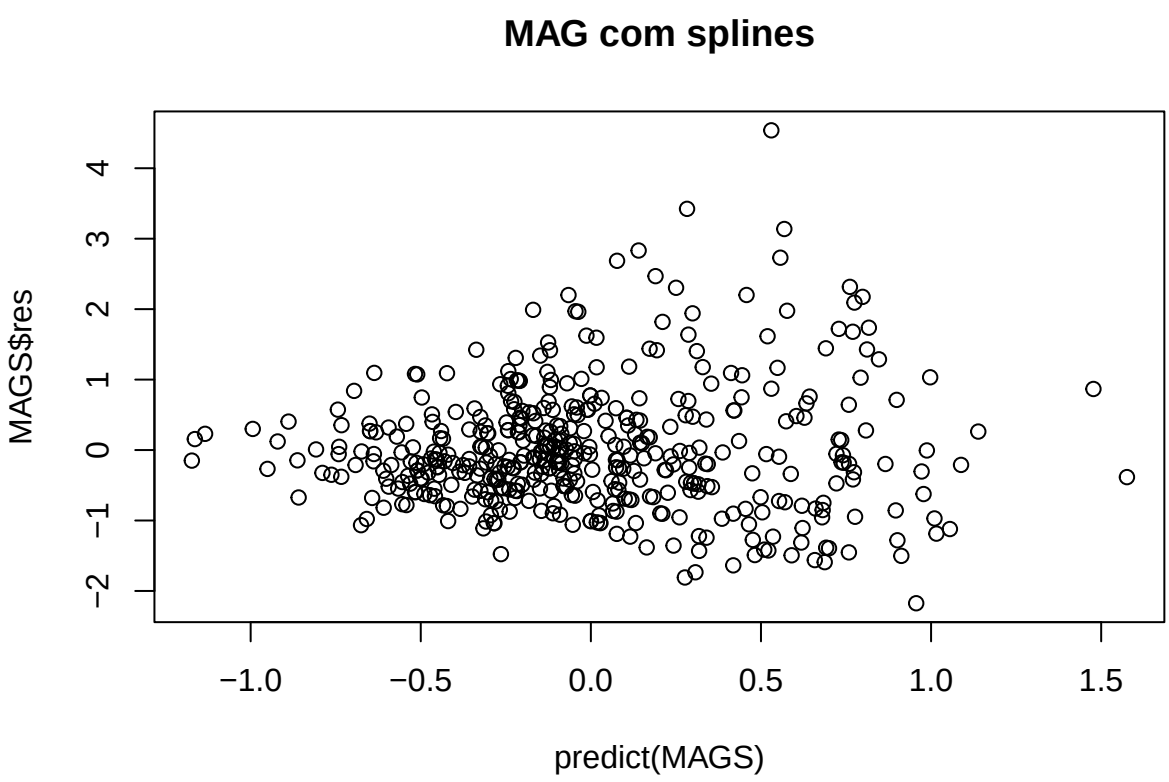
A distribuição dos resíduos, para todos os procedimentos, não é normal.

5- GRÁFICOS DE AJUSTE

Como o melhor modelo definido no item 3 foi o MAG com Spilines (MAGS) vamos fazer os graficos apenas para tal modelo.

Valores observados de y vs valores ajustados:

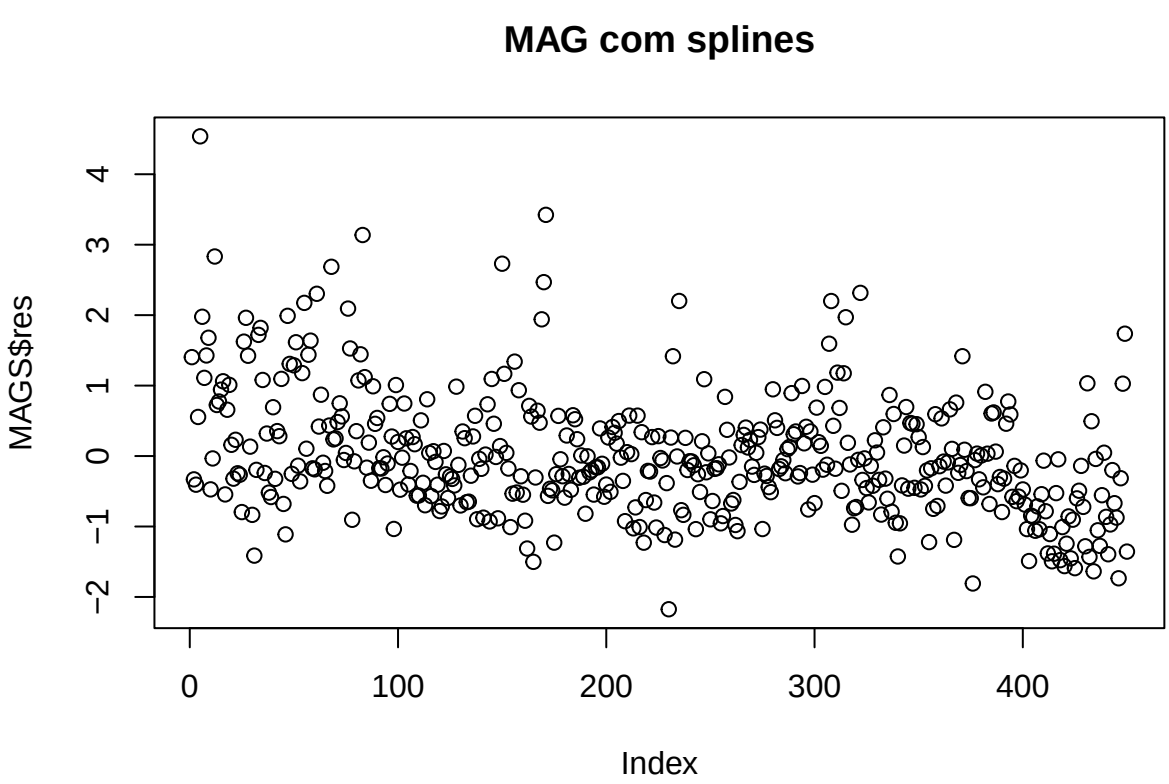
```
plot(predict(MAGS),MAGS$res, main= "MAG com splines")
```



Os residuos estão razoavelmente dispostos de forma aleatória em torno de zero.

Grafico de valores ajustados vs variavel de relação linear com y:

```
plot(MAGS$res, main= "MAG com splines")
```



Não parece haver nenhum problema de independência, pois todos os resíduos estão dispostos de forma aleatória em torno de zero.