PARTE 4 – Análise Crédito - Modelagem

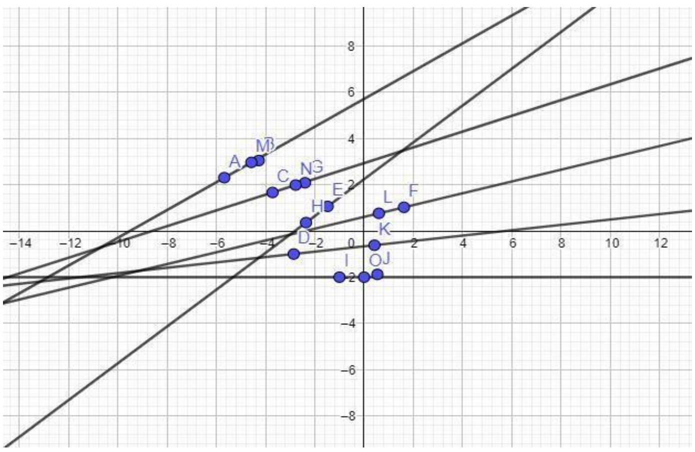
Estudo de caso 2: Implementação de regressão linear com 2 ou mais variáveis numéricas

O termo “Regressão” surgiu em 1885 com o antropólogo, matemático e estatístico Francis Galton. A sua aplicabilidade surgiu como metodologia usada na Antropometria, ou seja, estudo das medidas e da matemática dos corpos humanos.

Lembre-se que temos no nosso material o modelo do estudo antropométrico quando estudamos o caso do IMC.

O estudo realizado por Galton, observou as estaturas de pais e seus filhos. O resultado da observação demostrou que os filhos de pais com estatura baixa em relação à média, têm tendência a serem mais altos que seus pais, e filhos de pais com estatura alta em relação à média tendem a ser mais baixos que seus pais, ou seja, as alturas dos seres humanos em geral tendem a **regredir** à média. Para realizar a demonstração, podemos derivar essas medidas, cada qual, pela diferença da média das medidas observadas.

Quando falamos em derivar, podemos ter ao longo do eixo x, as diferenças entre os pontos A e B, B e C, C e D, e assim por diante, traçando tangentes a cada curva (reta) desses trechos segmentados, pode resultar em retas tangentes aproximadas. Veja a figura abaixo, dado o conjunto, a sua derivada são as retas tangentes as cada segmento. Isto é uma derivada. No entanto, não precisamos derivar cada segmento para expressá-los como uma reta (curva). Podemos aproximar cada segmento de um único modelo, então isto quer dizer que podemos regredir todas essas retas lineares a uma única, que represente todas.



Hoje, conhecemos a análise de regressão como uma técnica que permite estimar o comportamento médio de uma variável resposta em relação a uma ou mais variáveis explicativas. Por exemplo, estimar a altura média dos filhos a partir da altura de seus pais; estimar a produção média de uma lavoura a partir da quantidade de chuva, quantidade de adubo, etc. E como uma delas influência a outra.

É **importante notar que, apesar de ser uma possibilidade, a análise de regressão não tem como objetivo obter estimativas pontuais de eventos futuros, mas sim de estimar médias condicionais e efeitos, relativos ao comportamento dos dados, apenas uma previsão, um forecast.** A Análise de Regressão é chamada de Simples, quando existe apenas uma variável resposta e uma variável explicativa, e Múltipla quando existe uma variável resposta e mais de uma explicativa. Casos em que existe mais de uma variável resposta são analisados pela regressão Multivariada.

Implementando um modelo de Regressão Linear Simples:

O modelo de regressão linear simples consiste de 2 parâmetros, que correspondem aos coeficientes de uma equação da reta qualquer:

$$Y=\alpha+\beta X$$

Y é a esperança ou média da variável de resposta, você pode chamar de uma possível estimativa. O x é a variável que influência a variável de resposta, diretamente ou inversamente. Os coeficientes **α e β** são estimados através do Método dos Mínimos Quadrados.

Os mínimos quadrados são as variâncias de cada ponto, ou seja, é a diferença variável de cada ponto em relação a média, como cada ponto está equidistante diferentemente em relação a variância de outros pontos, cada um terá um erro distinto (erro mínimo quer dizer diferença). Cada variância pode ser positiva ou negativa, eleva-se ao quadrado para deixar em módulo. Regra dos mínimos quadrados. Quando se encontra estes mínimos pode-se traçar uma reta que os represente de uma forma estimada. O objetivo deste método é obter uma reta que minimiza as distâncias entre os valores estimados e os valores observados.

O **parâmetro α** representa o intercepto da reta, onde ela cruza o eixo **Y**, ou seja, o valor de **Y** para o qual **X = 0**. Este parâmetro só será interpretado na prática quando existir sentido real na variável explicativa assumindo valor zero.

Já o **parâmetro β** representa, neste caso, o efeito da variável explicativa sobre a variável resposta. O quanto ela pode influenciar o resultado Y.

A função que realiza o ajuste dessa reta ou modelo de regressão linear no R é a *lm()* na linguagem R. Neste exemplo, utilizaremos um conjunto de dados em que a variável resposta (Y) é o tempo de reação(curtida) da pessoa à uma certa notícia no Twiter, em segundos, e a variável explicativa

(X) é a idade do indivíduo. (Fonte: Bussab, 1988). No R, dados em tabelas são objetos do tipo *data frame*, nos quais cada coluna corresponde a uma variável e cada linha corresponde a uma observação.

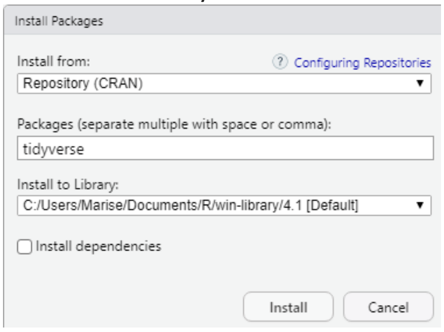
```
>dados<-data.frame(tempo=c(96,92,106,100,98,104,110,101,116,106,109,100,112,105,118,108,113,112,
127,117),idade = c(20,20,20,20,25,25,25,25,30,30,30,30,35,35,35,35,40,40,40,40))

> dados
   tempo idade
1     96  20
2     92  20
3    106  20
4    100  20
5     98  25
6    104  25
7    110  25
8    101  25
9    116  30
10   106  30
11   109  30
12   100  30
13   112  35
14   105  35
15   118  35
16   108  35
17   113  40
```

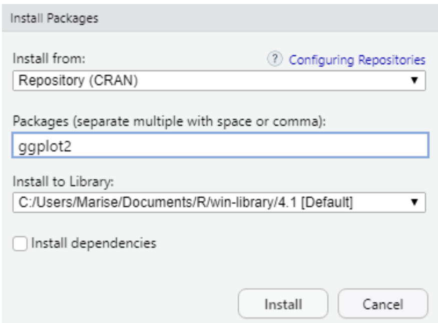
PARTE 4 – Análise Crédito - Modelagem

18 112   40
19 127   40
20 117   40

Precisamos da library ggplot
Vá em Tools e instale tidyverse



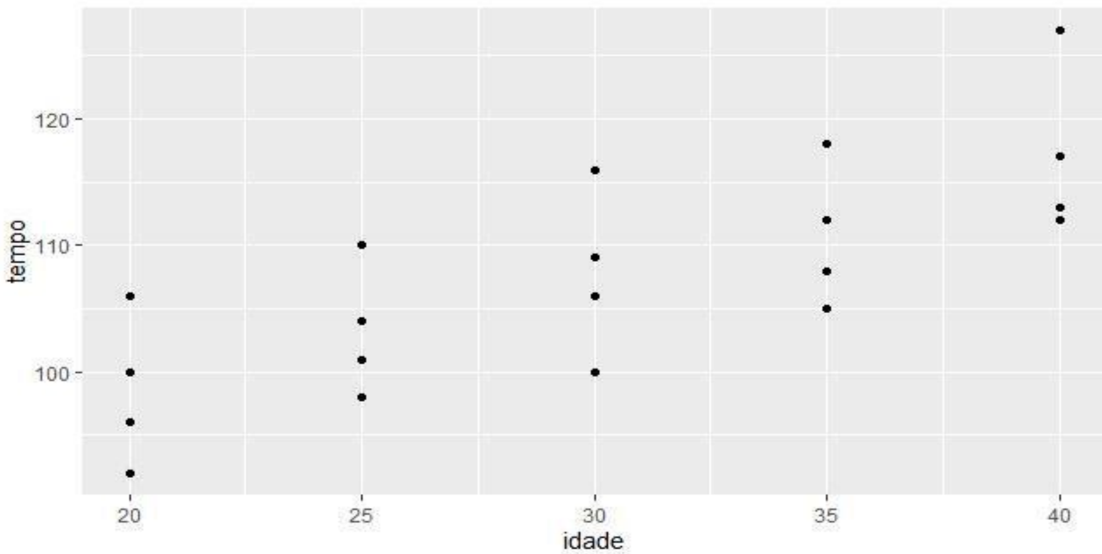
Depois ggplot2



>library(ggplot2)

Em caso de erro na biblioteca, feche o R e carregue novamente a biblioteca da ggplot2

> ggplot(dados, aes(x = idade, y = tempo))+
+ geom_point()



Observe o crescimento nos valores da variável tempo de acordo com o aumento das variáveis idade. Estima-se que haja um efeito da idade sobre o tempo de reação de curtidas diretamente. Coeficiente β > 0 é positivo.

Agora vamos ajustar o modelo para explicar este conjunto de dados.

Ajuste do Modelo:

```
> modelo <-lm(data = dados, formula = tempo ~idade)
> modelo$coefficients

(Intercept)      idade
       80.5        0.9
```

Podemos simplesmente consultar as estimativas dos parâmetros:

E temos a equação da reta ajustada:

$$Y = 80.5 + 0.9 * Idade$$

$$Y = \alpha + \beta x$$

Perceba que a equação ajusta os pontos para estimativas de cada um deles. Por este motivo, faz-se necessário testar qual significância desses novos valores estimados. Isto quer dizer, qual grau de confiança ou confiabilidade podemos afirmar que o modelo é linear, se aproximando de zero.

Vamos usar o comando summary para o modelo. Este comando estatístico indicará se os parâmetros estimados são significantes, isto quer dizer, se são valores distintos de zero, tem valor, tem significativo.

O summary no modelo linear também retorna a correlação entre essas duas variáveis, que é a medida R^2(adjusted R-squared). Este parâmetro indica o quanto da variação presente nos dados está sendo explicada pela covariável. Se há ou não uma correlação.

```
> summary(modelo)

Call:
lm(formula = tempo ~ idade, data = dados)

Residuals:
   Min     1Q Median     3Q    Max 
-7.500 -4.125 -0.750  2.625 10.500 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  80.5000     5.4510  14.768 1.67e-11
idade         0.9000     0.1769   5.089 7.66e-05

(Intercept) ***
idade       ***
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.593 on 18 degrees of freedom
Multiple R-squared:  0.5899,	Adjusted R-squared:  0.5672 
F-statistic:  25.9 on 1 and 18 DF,  p-value: 7.662e-05
```

O resultado do summary (modelo) revela que saídas estimadas relativas ao parâmetros, há um erro padrão associado a cada estimativa, uma estatística t e um p-valor associado, resultado do teste t utilizado para saber se as estimativas são realmente diferentes de zero. Quanto mais asteriscos presentes ao lado do efeito estimado, maior o nível de confiança com que podemos afirmar que o efeito não é nulo.

Quanto ao R^2, ao utilizar apenas uma variável é normal que o valor não seja extremamente alto. De qualquer maneira, na prática, 0.56 é um valor bastante razoável. De certa forma a idade afeta no tempo de curtidas.

Agora vamos qualificar o modelo linear.

Passando para a q**ualidade do ajuste**

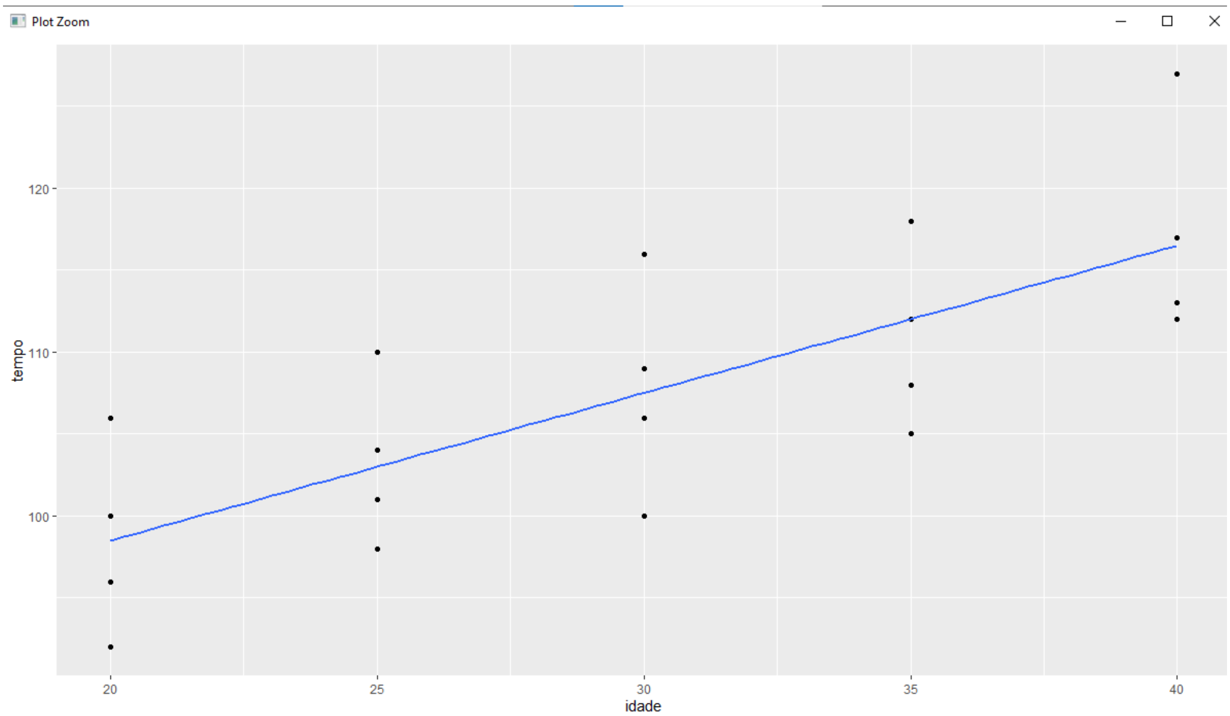
```
>ggplot(dados, aes(x = idade, y=tempo)) +
+         geom_point()+
+         geom_smooth(method =lm, se = FALSE)
```

Se houver erros , remover o pacote ggplot e instalar novamente da seguinte forma:

```
>remove.packages("ggplot2")
> install.packages("ggplot2")
> library(ggplot2)
```

Execute novamente

```
>ggplot(dados, aes(x = idade, y=tempo)) +
+         geom_point()+
+         geom_smooth(method =lm, se = FALSE)
```



A reta estimada claramente não coincidirá com todos os nossos dados. As medidas de distância entre os dados observados e a reta estimada são chamadas ***resíduos***. Os resíduos são utilizados para avaliar o ajuste do modelo, e a qualidade das estimativas feitas a partir dele.

Mostrando os dados estimados:

```
> predict(modelo)
   1    2    3    4     5     6     7     8     9    10
98.5 98.5 98.5 98.5 103.0 103.0 103.0 103.0 107.5 107.5
   11    12    13    14    15    16    17    18    19    20
107.5 107.5 112.0 112.0 112.0 112.0 116.5 116.5 116.5 116.5
```

```
> valor_aj<-predict(modelo)
> valor_aj
   1    2    3    4     5     6     7     8     9    10
98.5 98.5 98.5 98.5 103.0 103.0 103.0 103.0 107.5 107.5
   11    12    13    14    15    16    17    18    19    20
107.5 107.5 112.0 112.0 112.0 112.0 116.5 116.5 116.5 116.5
```

```
> valor_aj<-data.frame(predict(modelo))
> valor_aj
```

```
   predict.modelo.
1             98.5
2             98.5
3             98.5
4             98.5
5            103.0
6            103.0
7            103.0
8            103.0
9            107.5
10           107.5
11           107.5
12           107.5
13           112.0
14           112.0
15           112.0
16           112.0
17           116.5
18           116.5
19           116.5
20           116.5
```

```
> dados$id<-(c(1:20))
> dados
tempo idade id
1    96    20  1
2    92    20  2
3    106    20  3
4    100    20  4
5    98    25  5
6    104    25  6
7    110    25  7
8    101    25  8
9    116    30  9
10   106    30 10
11   109    30 11
12   100    30 12
13   112    35 13
14   105    35 14
15   118    35 15
16   108    35 16
17   113    40 17
18   112    40 18
19   127    40 19
20   117    40 20
```

Vamos ordenar o id para que fique do lado esquerdo da tabela.

Instale o pacote select

```
> install.packages("dplyr")

> library(dplyr)
> dados<-dados %>%
+       select(id, tempo,idade)

> dados
> dados

id tempo idade
1   1    96    20
2   2    92    20
3   3    106    20
4   4    100    20
5   5    98    25
6   6    104    25
7   7    110    25
8   8    101    25
9   9    116    30
10 10    106    30
11 11    109    30
12 12    100    30
13 13    112    35
14 14    105    35
15 15    118    35
```

```
16 16   108    35
17 17   113    40
18 18   112    40
19 19   127    40
20 20   117    40
```

Vamos incluir o id na tabela valor_aj

```
> valor_aj$id<-(c(1:20))
> valor_aj
```

Vamos mudar a coluna de id para a primeira posição da tabela

```
> valor_aj<-valor_aj %>%
+         select(id,predict.modelo)
> valor_aj

predict.modelo. id
1          98.5  1
2          98.5  2
3          98.5  3
4          98.5  4
5         103.0  5
6         103.0  6
7         103.0  7
8         103.0  8
9         107.5  9
10        107.5 10
11        107.5 11
12        107.5 12
13        112.0 13
14        112.0 14
15        112.0 15
16        112.0 16
17        116.5 17
18        116.5 18
19        116.5 19
20        116.5 20
```

Vamos analisar como os valores ajustados se comportam. Estes valores são exatamente os valores recalculados a partir da equação
Y = 80.5 + 0.9 ∗ Idade

Vamos fazer um teste, recalcular os valores com a nova equação para validar o resultado obtido em cada ponto pelo comando predict:
Para isso faça:

```
> valor_aj$calculado<-80.5 +0.9*dados$idade
> valor_aj
```

```
   predict.modelo. id calculado
1             98.5  1      98.5
2             98.5  2      98.5
3             98.5  3      98.5
4             98.5  4      98.5
5            103.0  5     103.0
6            103.0  6     103.0
7            103.0  7     103.0
8            103.0  8     103.0
9            107.5  9     107.5
10           107.5 10     107.5
11           107.5 11     107.5
12           107.5 12     107.5
13           112.0 13     112.0
14           112.0 14     112.0
15           112.0 15     112.0
16           112.0 16     112.0
17           116.5 17     116.5
18           116.5 18     116.5
19           116.5 19     116.5
20           116.5 20     116.5
```

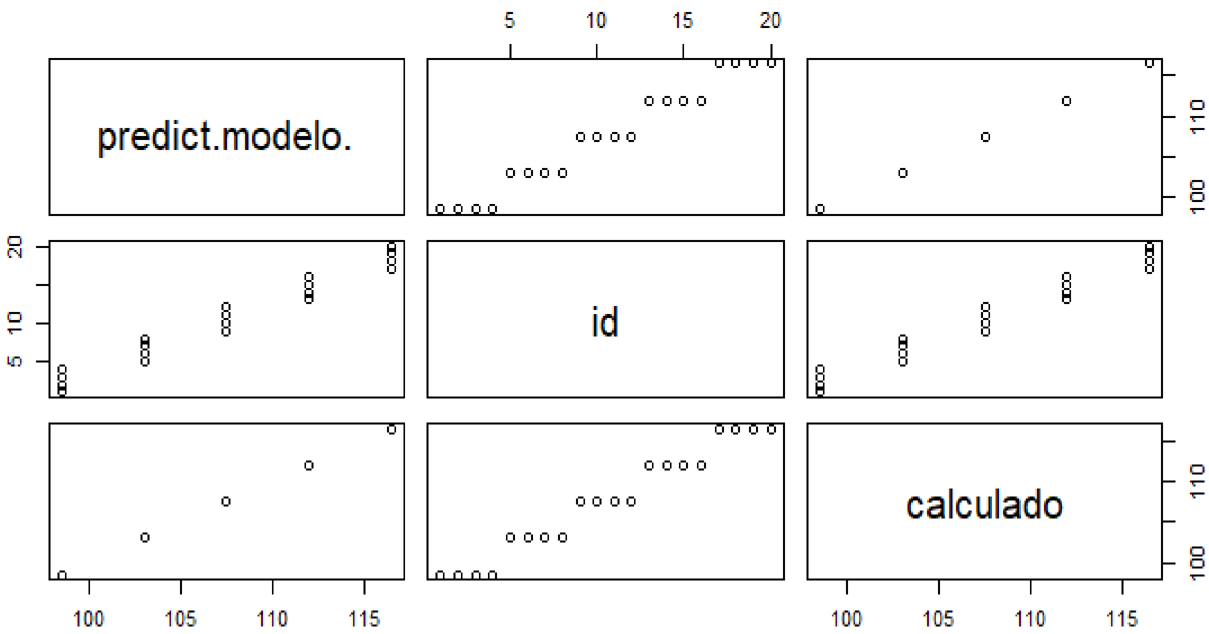
Veja que o resultado do valor na coluna predict.modelo. é exatamente o mesmo da coluna calculado pela fórmula do modelo linear.

Isto que dizer que os pontos estimados, seriam os melhores valores para um modelo linear.

Veja que o resultado do valor na coluna predict.modelo. é exatamente o mesmo da coluna calculado pela fórmula do modelo linear.

Isto que dizer que os pontos estimados, seriam os melhores valores para um modelo linear.

```
> plot(valor_aj)
```



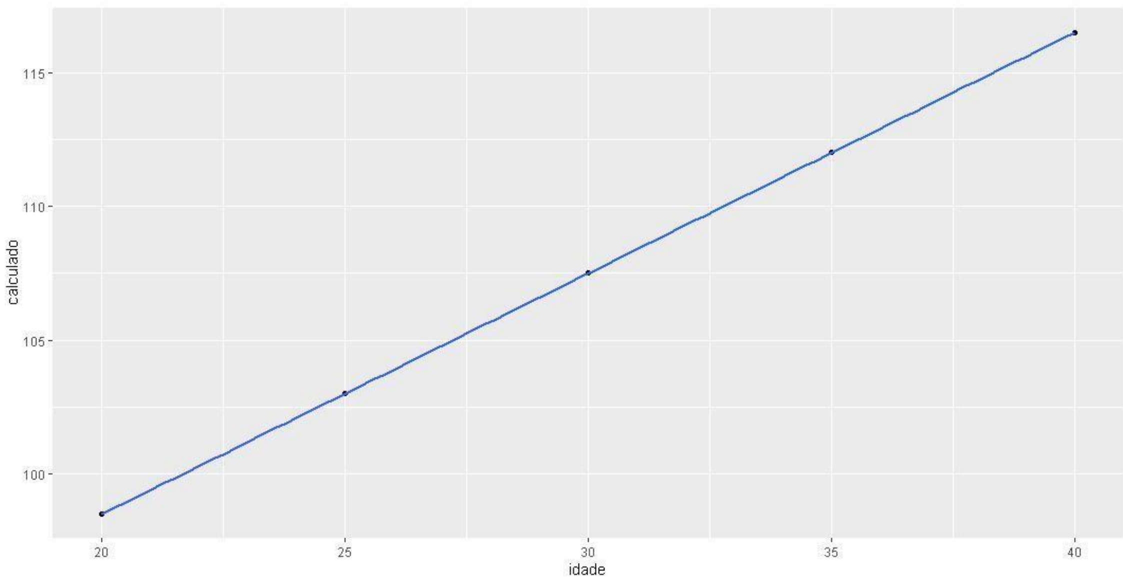
Vamos fazer um join entre as duas tabelas, criando um novo data set, com a idade e o novo valor estimado do modelo linear.

```
> dados1_model<-merge(dados,valor_aj)
> dados1_model
```

	id	tempo	idade	predict.modelo.	calculado
1	1	96	20	98.5	98.5
2	2	92	20	98.5	98.5
3	3	106	20	98.5	98.5
4	4	100	20	98.5	98.5
5	5	98	25	103.0	103.0
6	6	104	25	103.0	103.0
7	7	110	25	103.0	103.0
8	8	101	25	103.0	103.0
9	9	116	30	107.5	107.5
10	10	106	30	107.5	107.5
11	11	109	30	107.5	107.5
12	12	100	30	107.5	107.5
13	13	112	35	112.0	112.0
14	14	105	35	112.0	112.0
15	15	118	35	112.0	112.0
16	16	108	35	112.0	112.0
17	17	113	40	116.5	116.5
18	18	112	40	116.5	116.5
19	19	127	40	116.5	116.5
20	20	117	40	116.5	116.5

```
> ggplot(dados1_model,aes(x=idade,y=calculado)) +
+ geom_point()+
+ geom_smooth(method = lm,se =FALSE)
```

Caso ocorre erro no ggplo, instale novamente e restart o R



Observe o ajuste preciso do modelo.

Estes valores calculados seriam os valores previstos, estimados.

Condições para um Bom Ajuste de Modelo de Regressão Linear

Assim como qualquer método estatístico, a Regressão Linear, para ser corretamente utilizada, precisa que os dados estejam de acordo com algumas condições assumidas pelo modelo:

- **Normalidade dos Resíduos**

É necessário que os resíduos gerados pelo ajuste da reta sigam distribuição Normal.

- **Homocedasticidade**

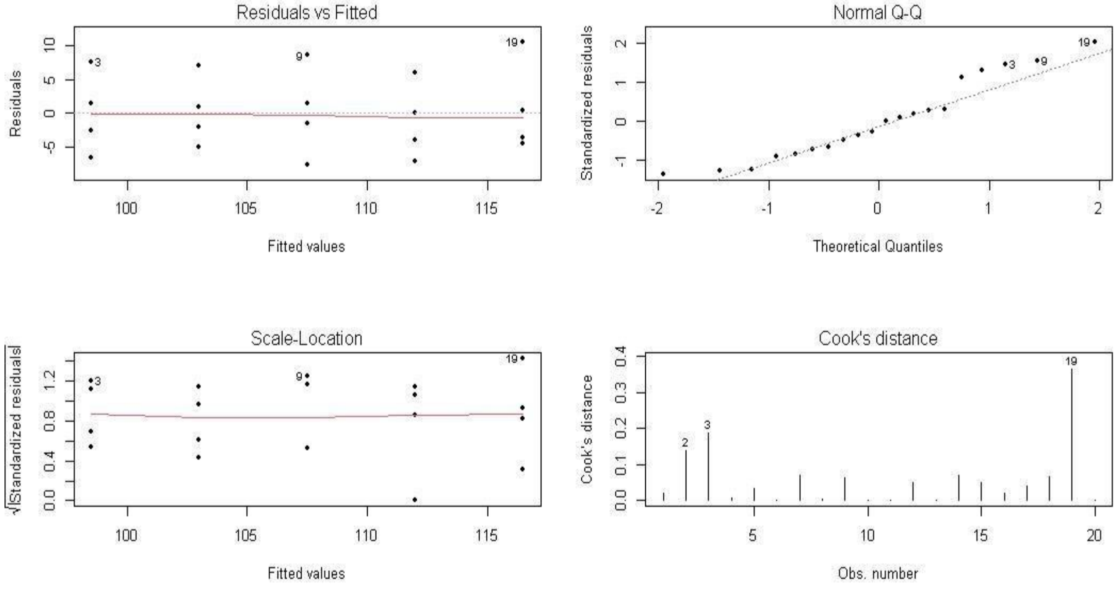
É necessário que a variância de Y seja constante para todos os valores de X. Ideal, mas com pouca diferença. Variáncia é a diferença do valor em relação ao valor médio de todos os dados.

- **Independência**

É necessário que não exista estrutura de dependência entre os dados, para que os resíduos sejam independentes e identicamente distribuídos.

Análise Residual ou de erros:
Dadas as suposições anteriores, validadas por meio dos gráficos residuais.

```
> par(mfrow = c(2,2))
> plot(modelo, which=c(1:4),pch=20)
```



Residuals vs Fitted – residual vs ajustado. No primeiro gráfico, temos os resíduos em função dos valores estimados. Podemos utilizar este gráfico para observar a independência e a **homocedasticidade**, se os resíduos se distribuem de maneira razoavelmente aleatória e com mesma amplitude em **torno do zero**.

Normal quantil-quantil. No segundo gráfico, podemos **avaliar a normalidade dos resíduos**. A linha diagonal pontilhada **representa a distribuição normal teórica**, e os pontos a distribuição dos resíduos observados. Espera-se que não exista grande fuga dos pontos em relação à reta teórica.

Scale-location. O terceiro gráfico pode ser avaliado da mesma maneira que o primeiro, observando a aleatoriedade e amplitude, desta vez dos resíduos padronizados. Este **gráfico** mostra se os resíduos são distribuídos igualmente ao longo dos intervalos de preditores. È assim que pode-se verificar a suposição de variância igual (homocedasticidade).

E o último gráfico permite visualizar as **Distâncias de Cook** das observações, uma medida de influência quando pode indicar a presença de *outliers* que possuem valor maior do que 1. Os números relacionados a cada linha vertical são as quantidades de observações em torno daquele valor.

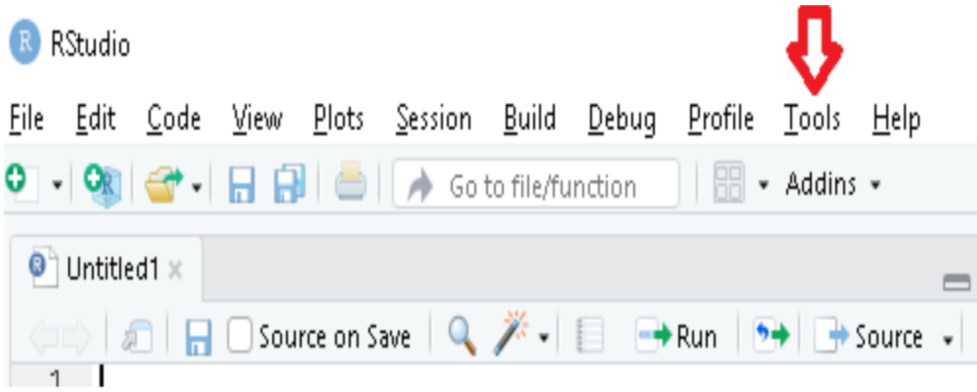
Quando a análise gráfica apresenta dúvidas, é possível também realizar testes estatísticos sobre os resíduos obtidos.

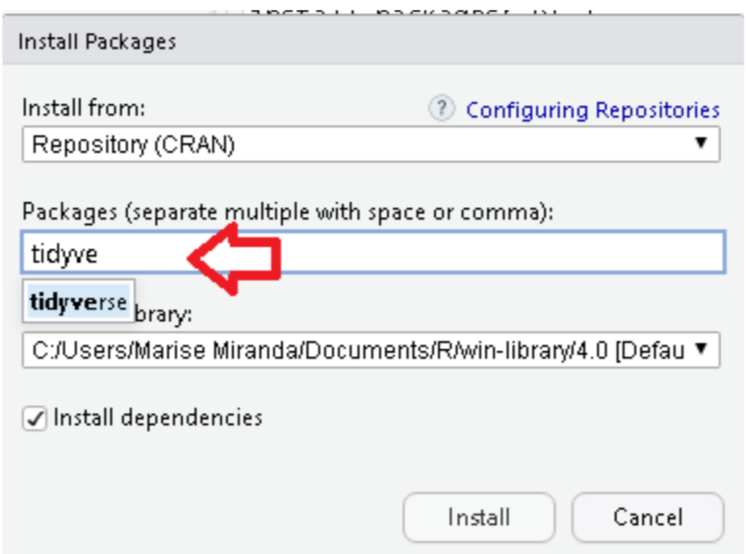
Estudo de caso 2: Base de dados LoanData em UCI Machine Learning (Dados empréstimo)

A análise do risco de crédito e a tomada de decisão na concessão de crédito é uma das operações mais importantes para as instituições financeiras. Levando em consideração os resultados anteriores, precisamos treinar um modelo para prever com precisão os resultados futuros (fonte: AWS Model R)

Carregue as bibliotecas que vamos usar.

Use Tools Install Packages Cran





PARTE 4 – Análise Crédito - Modelagem

```
library(tidyverse) # trabalha com vários tipos de estruturas de dados
library(ggthemes) # pacotes de temas e escalas aplicados a financeiro
library(corrplot)# pacote gráfico de matrix de correlação
library(GGally)# combina dados de matrix em interações geométricas
library(DT)# matrizes ou data frame que podem ser representados em HTML
library(caret)# simplificação de modelos preditivos
```

Carregue os dados disponíveis para análise. O conjunto de dados é constituído de registros do banco coletado sobre a situação de inadimplência e o perfil dos clientes.

Sugestão: carregue da área de trabalho

```
>loan=read.csv("C:/Users/Marise Miranda/Desktop/loan_data_set.csv",na =" ")
> View(loan)
> colnames(loan)
 [1]"Loan_ID"          "Gender"           "Married"
 [4]"Dependents"       "Education"        "Self_Employed"
[7] "ApplicantIncome"  "CoapplicantIncome" "LoanAmount"
[10] "Loan_Amount_Term" "Credit_History"  "Property_Area"
[13] "Loan_Status"
```

Seleção de recursos para a modelagem:

O conjunto de dados contém informações de idade, renda anual, grau de funcionário, casa própria que afetam a probabilidade de inadimplência do mutuário. As colunas que vamos usar são:

- **Loan_status** : tomou empréstimo? Yes, no
- **LoanAmount**: montante total do empréstimo tomado
- **Self_Employed** : emprego por conta própria
- **Property_Area** : Tipo de propriedade da casa/região
- **ApplicantIncome**: renda
- **CoapplicationIncome:** renda avalista
- **Loan_Amount_Term** : **período de** 36 ou 60 meses

Vamos agora atribuir uma seleção de colunas a uma nova variável, usaremos o select para as colunas da tabela, usando o operador pipe (%>%), para que seja usado o valor resultante da expressão do lado esquerdo como primeiro argumento da função do lado direito.

Para ter acesso ao pipe use o package: install.packages("magrittr")
Caso o comando select retorne erro reinstale:

```
> install.packages("dplyr")

> library(dplyr)

>loanteste1 = loan %>%

+  select(Loan_Status, LoanAmount, Credit_History, Gender, ApplicantIncome, Loan_Amount_Term)

>loanteste1
```

```
Loan_Status LoanAmount Credit_History Gender ApplicantIncome Loan_Amount_Term
1    Y    NA    1  Male    5849    360
2    N    128   1  Male    4583    360
3    Y    66    1  Male    3000    360
4    Y    120   1  Male    2583    360
5    Y    141   1  Male    6000    360
6    Y    267   1  Male    5417    360
7    Y    95    1  Male    2333    360
8    N    158   0  Male    3036    360
9    Y    168   1  Male    4006    360
10   N    349   1  Male    12841   360
11   Y    70    1  Male    3200    360
12   Y    109   1  Male    2500    360
13   Y    200   1  Male    3073    360
14   N    114   1  Male    1853    360
15   Y    17    1  Male    1299    120
16   Y    125   1  Male    4950    360
17   Y    100   NA Male    3596    240
18   N    76    0 Female   3510    360
19   N    133   1  Male    4887    360
20   Y    115   1  Male    2600    NA
21   N    104   0  Male    7660    360
22   Y    315   1  Male    5955    360
23   N    116   0  Male    2600    360
24   N    112   0  <NA>    3365    360
……
```

Vamos verificar quantos NA temos em nossa base de dados:

```
> sapply(loanteste1 , function(x) sum(is.na(x)))

Loan_Status       LoanAmount  Credit_History        Gender ApplicantIncome Loan_Amount_Te
rm

          0          22          50          0          0          14
```

Vamos retirar esses valores ausentes:

PARTE 4 – Análise Crédito - Modelagem

```
> loanteste2 = loanteste1 %>%
+              filter(!is.na(Loan_Amount_Term),
+              !is.na(LoanAmount),
+              !is.na(Credit_History))

        > loanteste2
   Loan_Status LoanAmount Credit_History Gender ApplicantIncome Loan_Amount_Term
1           N        128              1   Male            4583              360
2           Y         66              1   Male            3000              360
3           Y        120              1   Male            2583              360
4           Y        141              1   Male            6000              360
5           Y        267              1   Male            5417              360
6           Y         95              1   Male            2333              360
7           N        158              0   Male            3036              360
8           Y        168              1   Male            4006              360
9           N        349              1   Male           12841              360
10          Y         70              1   Male            3200              360
11          Y        109              1   Male            2500              360
12          Y        200              1   Male            3073              360
13          N        114              1   Male            1853              360
14          Y         17              1   Male            1299              120
```

Outra maneira de retirar os Nas:

```
> loan1<-na.omit(loan)
> loan1teste

> loan1
   Loan_ID Gender Married Dependents     Education Self_Employed
2  LP001003   Male     Yes          1      Graduate            No
3  LP001005   Male     Yes          0      Graduate           Yes
4  LP001006   Male     Yes          0  Not Graduate            No
5  LP001008   Male      No          0      Graduate            No
6  LP001011   Male     Yes          2      Graduate           Yes
7  LP001013   Male     Yes          0  Not Graduate            No
8  LP001014   Male     Yes         3+      Graduate            No
9  LP001018   Male     Yes          2      Graduate            No
10 LP001020   Male     Yes          1      Graduate            No
11 LP001024   Male     Yes          2      Graduate            No
12 LP001027   Male     Yes          2      Graduate
13 LP001028   Male     Yes          2      Graduate            No
14 LP001029   Male      No          0      Graduate            No
15 LP001030   Male     Yes          2      Graduate            No
```

PARTE 4 – Análise Crédito - Modelagem

```
> sapply(loan1 , function(x) sum(is.na(x)))
      Loan_ID          Gender       Married      Dependents
            0               0             0               0
    Education   Self_Employed ApplicantIncome CoapplicantIncome
            0               0             0               0
   LoanAmount Loan_Amount_Term  Credit_History  Property_Area
            0               0             0               0
  Loan_Status
            0
```

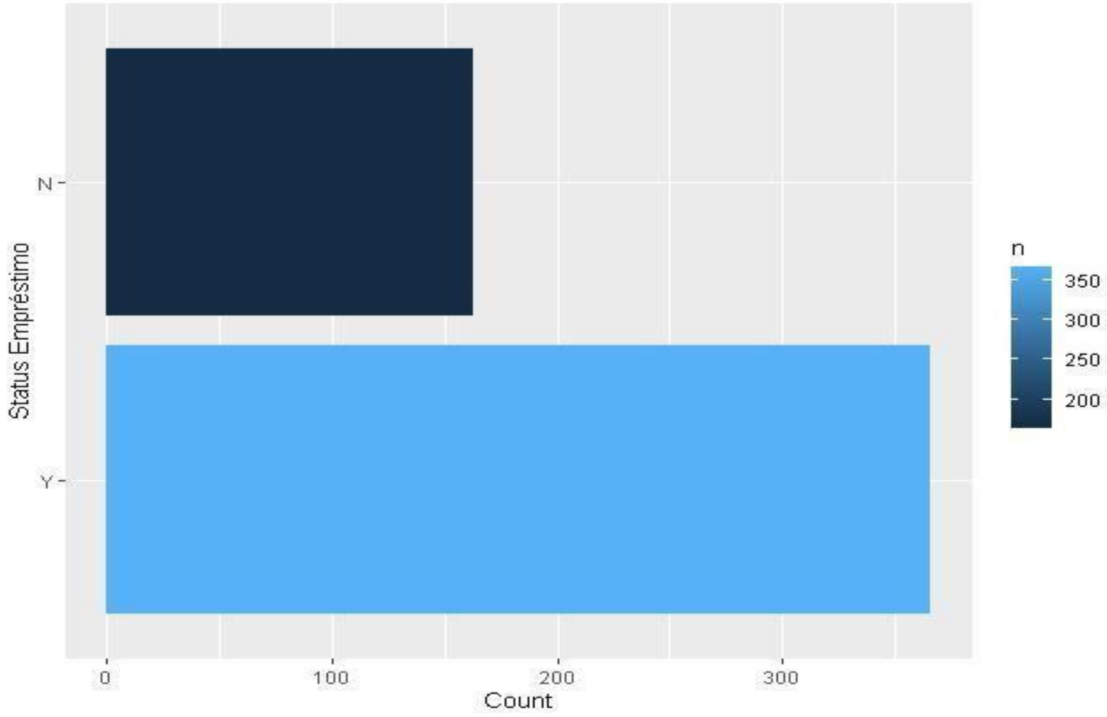
Análise Exploratória dos dados:

empréstimo_status :

```
> loanteste2 %>%
+       count(Loan_Status) %>%
+       ggplot(aes(x = reorder(Loan_Status , desc(n)), y = n , fill = n )) +
+       geom_col() +
+       coord_flip() +
+       labs(x = 'Status Empréstimo', y = 'Count')
```

Caso dê problema com erro em count

Chame a biblioteca
library(dplyr)



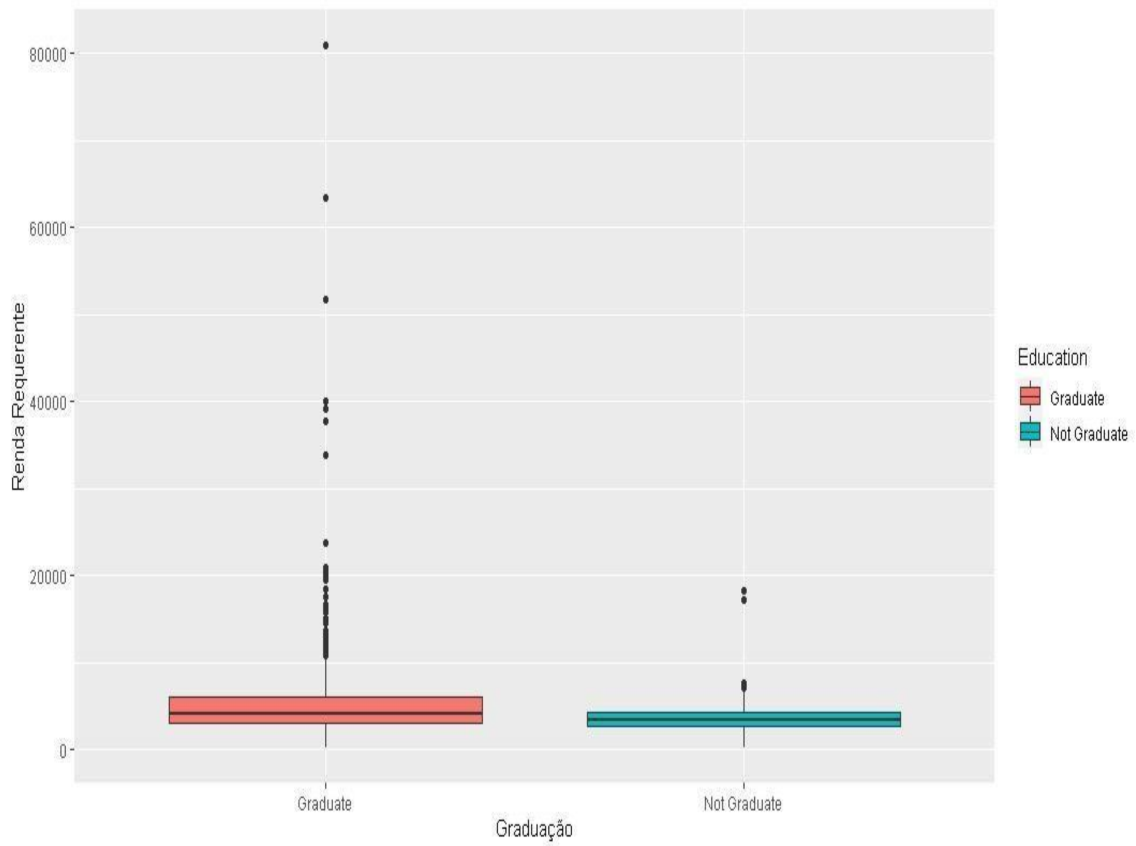
Análises de concessão de crédito:

Renda do Requerente do empréstimo e graduação

Vamos observar como essas variáveis podem ser úteis para a modelagem de risco de crédito. Sabe-se que quanto melhor a nota, menor a taxa de juros. Podemos visualizar isso perfeitamente com boxplots.

```
ggplot(loan, aes(x = Education, y = ApplicantIncome, fill = Education)) +
+    geom_boxplot() +
+    theme_gray()+
+ labs(y='Renda Requerente', x= 'Graduação')
```

Msc Eduardo Verri
Dra Marise Miranda



Assumimos que o grau de escolaridade é um grande indicador do volume de empréstimos Mas quantas delas não tiveram desempenho agrupado por série?

A decisão da análise requer a formação de subsets, que levem em consideração a tomada de empréstimos e ou o nível de graduação.

Perceba que no gráfico de renda do requerente pelo nível de estudo tem outliers.

Vamos limpar esses outliers.

```
> loanteste3 = loan %>%

+        select(Loan_Status, LoanAmount,Credit_History,
Gender, ApplicantIncome,Loan_Amount_Term, Education)


> loanteste3
   Loan_Status LoanAmount Credit_History Gender ApplicantIncome
Loan_Amount_Term  Education
1          Y        NA              1  Male          5849          360     Graduate
2          N        128             1  Male          4583          360     Graduate
3          Y         66             1 Male           3000          360     Graduate
4          Y        120             1  Male          2583          360 Not Graduate
5          Y        141             1  Male          6000          360     Graduate
6          Y        267             1  Male          5417          360     Graduate
```

Vamos retirar os NAs

```
> loanteste3
   Loan_Status LoanAmount Credit_History Gender ApplicantIncome
Loan_Amount_Term  Education
2          N       128            1  Male        4583          360     Graduate
3          Y        66            1 Male        3000          360     Graduate
4          Y       120            1  Male        2583          360 Not Graduate
5          Y       141            1  Male        6000          360     Graduate
6          Y       267            1  Male        5417          360     Graduate
7          Y        95            1 Male        2333          360 Not Graduate
8          N       158            0  Male        3036          360     Graduate
9          Y       168            1  Male        4006          360     Graduate
10         N       349            1  Male       12841          360     Graduate
11         Y        70            1  Male        3200          360     Graduate
12         Y       109            1  Male        2500          360     Graduate
```

```
> loansubset1<-na.omit(loanteste3)
```

```
> loansubset1
   Loan_Status     LoanAmount     Credit_History     Gender     ApplicantIncome
Loan_Amount_Term  Education
2          N       128           1  Male        4583          360     Graduate
3          Y        66           1  Male        3000          360     Graduate
4          Y       120           1  Male        2583          360 Not Graduate
5          Y       141           1  Male        6000          360     Graduate
6          Y       267           1  Male        5417          360     Graduate
7          Y        95           1  Male        2333          360 Not Graduate
8          N       158           0  Male        3036          360     Graduate
9          Y       168           1  Male        4006          360     Graduate
10         N       349           1  Male       12841          360     Graduate
11         Y        70           1  Male        3200          360     Graduate
12         Y       109           1  Male        2500          360     Graduate
13         Y       200           1  Male        3073          360     Graduate
14         N       114           1  Male        1853          360     Graduate
15         Y        17           1  Male        1299          120     Graduate
16         Y       125           1  Male        4950          360     Graduate
18         N        76           0 Female       3510          360     Graduate
19         N       133           1  Male        4887          360 Not Graduate
21         N       104           0  Male        7660          360 Not Graduate
22         Y       315           1  Male        5955          360     Graduate
```

```
>summary(loansubset1)
```

```
Loan_Status         LoanAmount     Credit_History     Gender           ApplicantIncome Loan_Amount_Term  Education
Length:529          Min.   :  9.0  Min.   :0.0000   Length:529         Min.   :  150   Min.   : 36.0    Length:529
Class :character    1st Qu.:100.0  1st Qu.:1.0000   Class :character   1st Qu.: 2900   1st Qu.:360.0    Class :character
Mode  :character    Median :128.0  Median :1.0000   Mode  :character   Median : 3816   Median :360.0    Mode  :character
                    Mean   :145.9  Mean   :0.8507                      Mean   : 5508   Mean   :342.4
                    3rd Qu.:167.0  3rd Qu.:1.0000                      3rd Qu.: 5815   3rd Qu.:360.0
                    Max.   :700.0  Max.   :1.0000                      Max.   :81000   Max.   :480.0
```

Vamos localizar os valores outliers para remoção. Observando o valor máximo de 81000 em ApplicantIncome.

```
> loansubset2<-loansubset1[loansubset1$ApplicantIncome>80000,]
> loansubset2
```

```
 410       N        360          0  Male
   ApplicantIncome  Loan_Amount_Term  Education
 410       81000              360  Graduate
```

O subset loansubset2 mostra as linhas com valores acima de 80000, para localizar a linha a ser removida.

Vamos fazer um subset novo a partir da remoção da linha encontrada com o outlier. No entando precisamos incluir uma coluna de ID.

```
> summary(loansubset1)
>
 Loan_Status          LoanAmount
 Length:529         Min.   :  9.0
 Class :character   1st Qu.:100.0
 Mode  :character   Median :128.0
                    Mean   :145.9
                    3rd Qu.:167.0
                    Max.   :700.0
 Credit_History       Gender
 Min.   :0.0000   Length:529
 1st Qu.:1.0000   Class :character
 Median :1.0000   Mode  :character
 Mean   :0.8507
 3rd Qu.:1.0000
 Max.   :1.0000
 ApplicantIncome Loan_Amount_Term
 Min.   :  150   Min.   : 36.0
 1st Qu.: 2900   1st Qu.:360.0
 Median : 3816   Median :360.0
 Mean   : 5508   Mean   :342.4
 3rd Qu.: 5815   3rd Qu.:360.0
 Max.   :81000   Max.   :480.0
  Education
 Length:529
 Class :character
 Mode  :character
```

Msc Eduardo Verri
Dra Marise Miranda

> loansubset1$id<-c(1:529)

> loansubset1

Veja que uma coluna de id foi criada

> loansubset2<-loansubset1[loansubset1$ApplicantIncome>80000,]

> loansubset2

```
    Loan_Status LoanAmount Credit_History Gender
410           N        360              0  Male
    ApplicantIncome Loan_Amount_Term Education id
410           81000              360 Graduate 352
```

E o id que porcuramos relativo ao outlier 81000 é o 352.

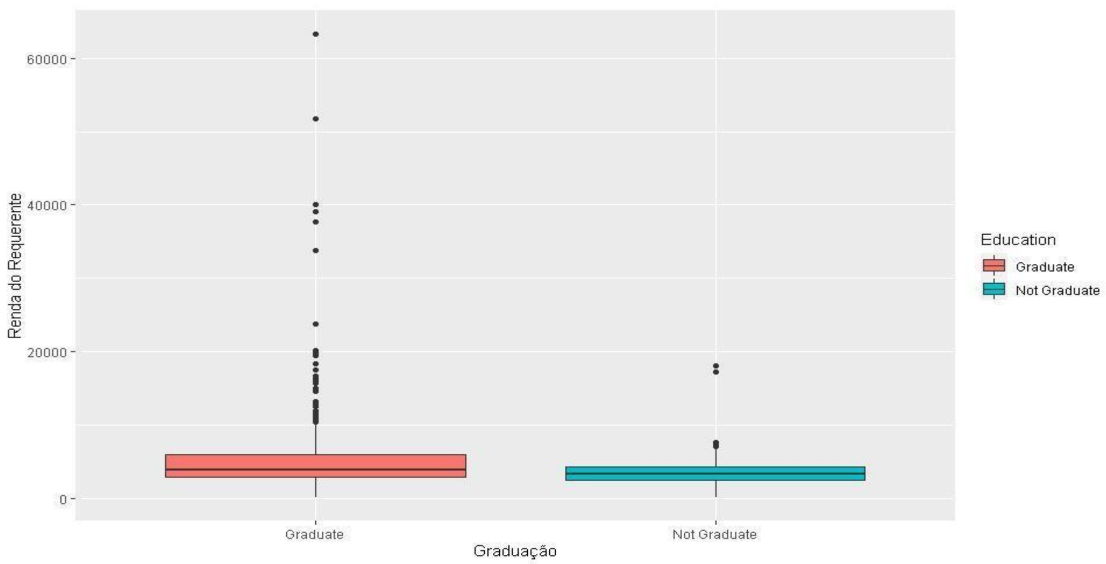
Vamos remover agora:

> loanremove<-loansubset1[-352,]

> summary(loanremove)

```
 Loan_Status          LoanAmount     Credit_History      Gender          ApplicantIncome Loan_Amount_Term  Education              id
 Length:528         Min.   :  9.0   Min.   :0.0000   Length:528         Min.   :  150   Min.   : 36.0    Length:528         Min.   :  1.0
 Class :character   1st Qu.:100.0   1st Qu.:1.0000   Class :character   1st Qu.: 2899   1st Qu.:360.0    Class :character   1st Qu.:132.8
 Mode  :character   Median :128.0   Median :1.0000   Mode  :character   Median : 3815   Median :360.0    Mode  :character   Median :264.5
                    Mean   :145.4   Mean   :0.8523                      Mean   : 5365   Mean   :342.3                       Mean   :264.8
                    3rd Qu.:166.2   3rd Qu.:1.0000                      3rd Qu.: 5804   3rd Qu.:360.0                       3rd Qu.:397.2
                    Max.   :700.0   Max.   :1.0000                      Max.   :63337   Max.   :480.0                       Max.   :529.0
```

O máximo valor de ApplicantIncome é 63337. Vamos ver como ficam o boxplot.



Vamos continuar nossa análise exploratória.

Não melhorou muito a análise de concessão de crédito. Vamos ajustar um novo subset desta vez removendo os outliers acima de 40000, mas vamos verificar qunatos teremos que remover.

```
> loansubset2<-loansubset1[loansubset1$ApplicantIncome>40000,]
> loansubset2
```

```
  172      Y       700         1
  334      Y       490         1  Male
  410      N       360         0  Male

  172       51763         300 Graduate 144
  334       63337         180 Graduate 285
  410       81000         360 Graduate 352
```

Vamos remover os três id, lembre-se que não removemos da base original já tratada criamos outro subset de remoção

```
> loanremove1<-loansubset1[-352,]

> loanremove2<-loanremove1[-285,]

> loanremove3<-loanremove2[-144,]

> loanremove3
```

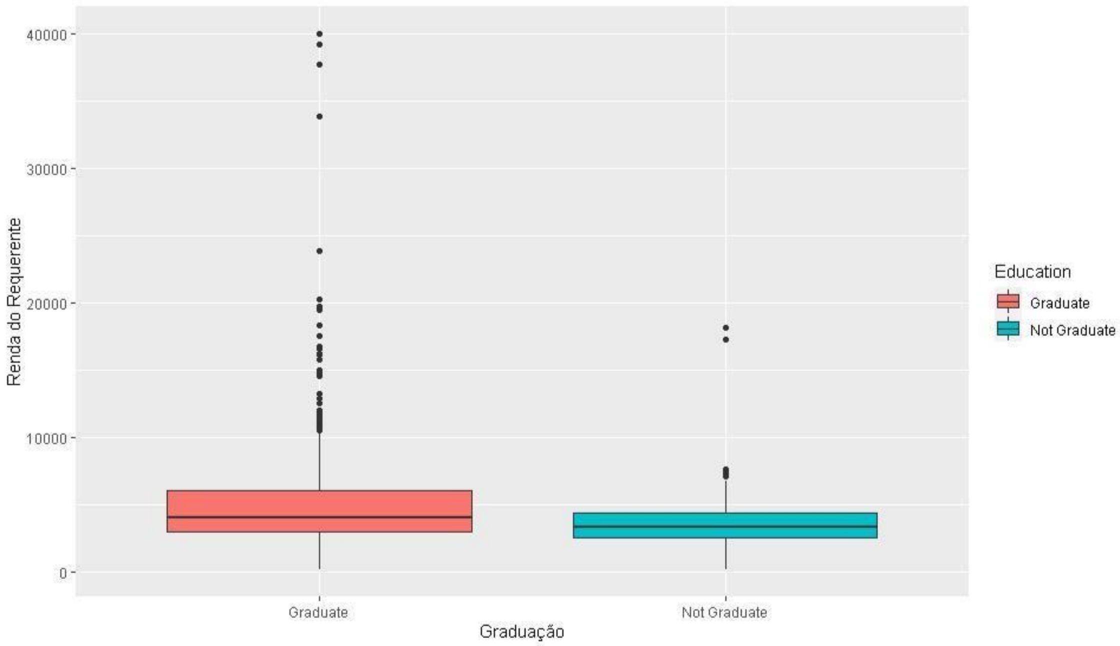
```
> summary(loanremove3)

 Loan_Status         LoanAmount    Credit_History
 Length:526        Min.  : 9.0  Min.  :0.0000
 Class :character 1st Qu.:100.0 1st Qu.:1.0000
 Mode :character    Median :128.0  Median :1.0000
                    Mean :143.7  Mean  :0.8517 3rd
                    Qu.:165.0  3rd Qu.:1.0000Max.
                    :600.0  Max. :1.0000
   Gender     ApplicantIncome     Loan_Amount_Term
 Length:526 Min.  : 150  Min.  : 36.0
 Class :character 1st Qu.: 2896 1st Qu.:360.0 Mode
 :character  Median : 3814  Median :360.0
                    Mean  : 5166  Mean  :342.7
                    3rd Qu.: 5766 3rd Qu.:360.0
                    Max.   :39999   Max.     :480.0
  Education              id
 Length:526        Min. : 1.0
 Class :character  1st Qu.:132.2
```

```
 Mode :character   Median :264.5
                   Mean   :265.0
                   3rd Qu.:397.8
                   Max.  :529.0
```

Veja que foram removidos apenas três dados ficando com 526 observações.

Vamos novamente gerar o boxplot, mas desta vez com o novo subset loanremove3.



Podemos melhorar mais ainda, removendo os ouliers que afetam este conjunto de análise. Mas vcs podem fazer isso depois. Agora vamos analisar o gráfico boxplot.

Ele tenta nos mostrar a distribuição dos dados, com o valor mínimo, 1º quartil, média, mediana, 3º quartil, máximo.

Até que tenhamos uma boa situação de análise para que possamos ser acertivos na concessão de crédito a mais pessoas de interesse.

E assim vamos criando novos data sets a partir do original, fazendo melhorias e diminuindo os outliers de modo a ter um cluster mais estável para tomada de decisão que possa atender um nicho muito característico.

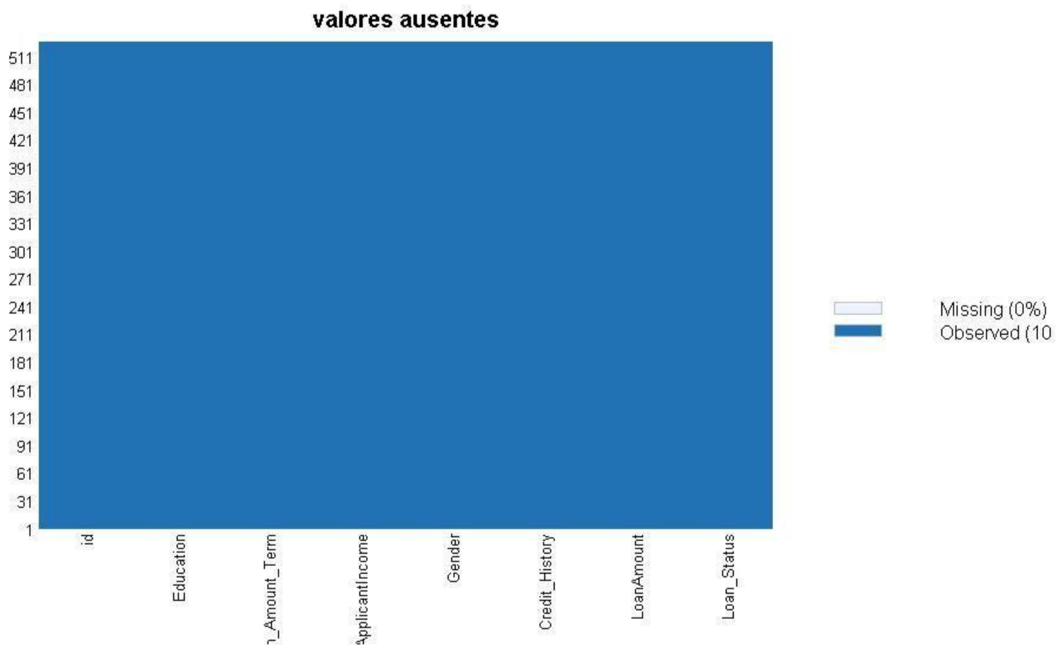
Vamos trabalhar com um modelo de Regressão Logística:

No R instale em tools Install Packages:
Amelia

Na console digite:

PARTE 4 – Análise Crédito - Modelagem

```
>library(Amelia)
> missmap(loanremove3,main = "valores ausentes")
```



Verificamos que não temos valores ausentes no dataset loanremove3.

Modelos lineares generalizados: Como um lembrete, os Modelos Lineares Generalizados são uma extensão dos modelos de regressão linear que permitem que a variável dependente seja não normal.

A regressão logística é amplamente utilizada dentre os modelos generalizados. A regressão logística é usada para prever uma classe, ou seja, uma probabilidade. A regressão logística pode prever um resultado binário com precisão.

Imagine que você deseja prever se um empréstimo será negado / aceito com base em muitos atributos. A regressão logística é da forma 0/1. y = 0 se um empréstimo for rejeitado, y = 1 se aceito.

Um modelo de regressão logística difere do modelo de regressão linear de duas maneiras.

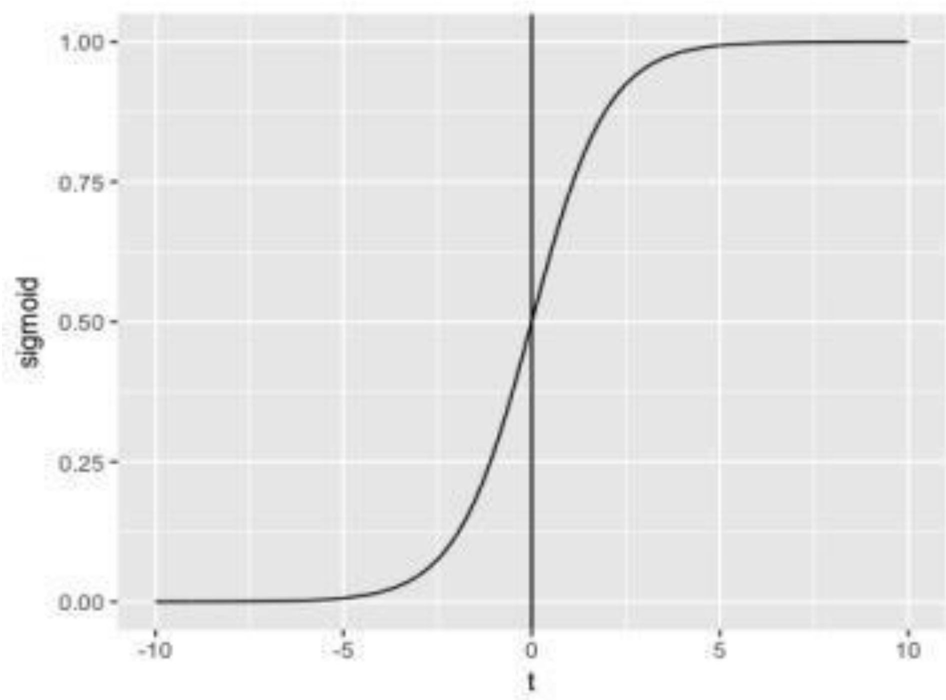
- Em primeiro lugar, a regressão logística aceita apenas entrada dicotômica (binária) como variável dependente (ou seja, um vetor de 0 e 1).
- Em segundo lugar, o resultado é medido pela seguinte função de ligação probabilística chamada **sigmóide** devido à sua forma de S:

$$Prlog = \frac{1}{1 + e^{-(b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_k X_k)}}$$

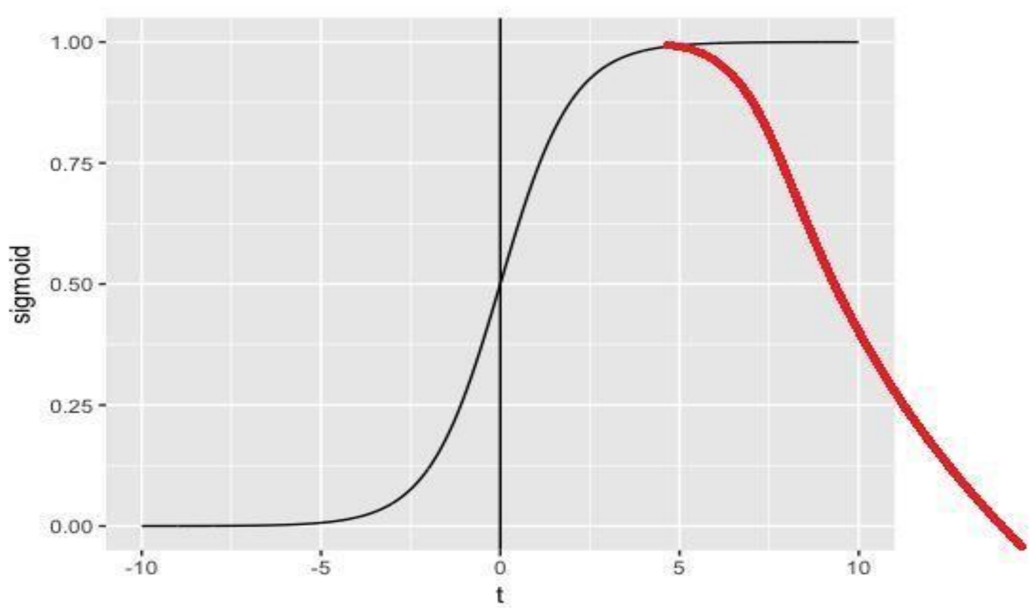
Msc Eduardo Verri
Dra Marise Miranda

Aqui é possível perceber que quanto mais elementos de análise são incluidos nas variáveis idependentes mais chances de sucesso são possíveis.

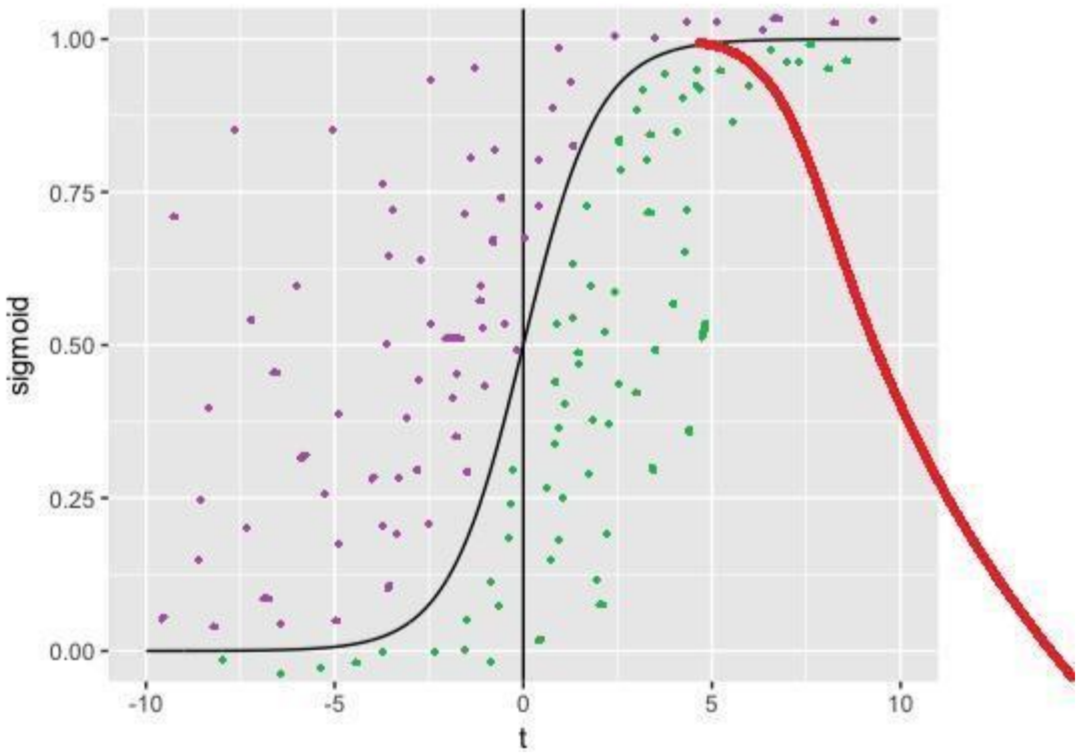
A saída da função está sempre entre 0 e 1. Verifique a imagem abaixo



Observe melhor e compare a um modelo de distribuição binominal parcial, aproximada de uma normal.



Msc Eduardo Verri
Dra Marise Miranda



Por este motivo devemos usar um modelo generalizado.

Devemos considerar os GLMs principalmente quando a variável resposta é expressa em:

- contagens simples
- contagem expressa em proporções
- número de sucesso e tentativa
- variáveis binárias (ex. morto x vivo)
- tempo para o evento ocorrer (modelos de sobrevivência)

Os modelos lineares generalizados (**GLMs**) são uma ampliação dos modelos lineares. Os **GLM's** são usados quando os resíduos (erro) do modelo apresentam distribuição diferente da normal (gaussiana). Por exemplos, variáveis de contagem são inteiras e apresentam os valores limitados no zero. Esse tipo de variável, em geral, tem uma distribuição de erros assimétrica para valores baixos e uma variância que aumenta com a média dos valores preditos, violando duas premissas dos modelos lineares.

Funções de ligações canônicas

Para alguns tipos de famílias de variáveis temos funções de ligações padrões. As mais usadas são:

Natureza da resposta	Estrutura dos resíduos (erro)	Função de ligação
contínua	normal	identidade
contagem	poisson	log
proporção	binomial	logit

Só pra lembrar a forma canônica é a forma matemática mais simples que explica o modelo ou método. Em linguagem léxica, canônica é a junção de uma sílaba com uma vogal (mato= ma – to)

Em linguagem R temos as funções implícitas dentro do Package GLM – Modelos lineares generalizados. Sendo o objeto da classe "family" que contém os modelos das distribuições dos erros e o link da função a ser usado no modelo. Conforme tabela abaixo. Apenas assimile.

family(object, …)

binomial(link = "logit") → eventos binários (fumante ou não fumante)

gaussian(link = "identity") → medidas físicas (peças x defeitos)

Gamma(link = "inverse") → tempo de vida de produto

inverse.gaussian(link = "1/mu^2") → lançamento de um novo produto

poisson(link = "log") → eventos imprevisíveis (sentenças criminais)

quasi(link = "identity", variance = "constant") □ cadeia produtiva x produção

quasibinomial(link = "logit") → usado como parâmetro extra quasipoisson(link = "log")→ usado como parâmetro extra