EDA - Dados de Crédito de Clientes de uma Instituição Financeira

Grupo 10 - Gabriel Nascimento, Ana Elisa e Juliana Câmara

2023-02-15

1. Contextualização

O crédito financeiro desempenha um grande papel no desenvolvimento economico dos países, porém controlar um objeto não tangível requer uma série de análises de hábitos financeiros por parte das empresas. Para reduzir significamente o risco de default, algumas perguntas precisam ser respondidas com o máximo de precisão pelas instituições interessadas:

1. Qual a faixa salarial do grupo predominante?
2. Qual a quantidade média de emprestimos desses clientes?
3. O salário líquido mensal pode influenciar de alguma forma nas taxas de juros que os clientes sofrem?
4. Qual o percentual de compromentimento da renda com dívidas?

2. Dataset

```
dados = read.csv('credito_tratado.csv')
```

O dataset escolhido traz informações sobre crédito de clientes de uma instituição financeira americana no período de dezembro/2022. Em seu formato original, o período de análise é maior e há repetições de um mesmo cliente em diversas entradas(atualizações), contando com 50.000 linhas e pode ser encontrado através deste LINK.
O dataset carregado acima já está tratado e sem valores faltantes.

As colunas(variáveis) contidas nesta base são:

```
colnames(dados)
```

```
##  [1] "Customer_ID"            "Month"                  "Age"
##  [4] "Monthly_Inhand_Salary"  "Num_Bank_Accounts"
"Num_Credit_Card"
```

```
##  [7] "Interest_Rate"          "Num_of_Loan"            "Type_of_Loan"
## [10] "Delay_from_due_date"    "Num_of_Delayed_Payment"
"Outstanding_Debt"
## [13] "Payment_of_Min_Amount"
```

Onde:

- 1.”Customer_ID” - ID do Cliente
- 2.”Month” - Mês do Ocorrência
- 3.”Age” - Idade
- 4.”Monthly_Inhand_Salary” - Salário Líquido Mensal (USD)
- 5.”Num_Bank_Accounts” - Número de Contas Bancárias
- 6.”Num_Credit_Card” - Número de cartões de crédito
- 7.”Interest_Rate” - Taxa de Juros (%)
- 8.”Num_of_Loan” - Número de Empréstimos
- 9.”Type_of_Loan” - Tipos de Empréstimos
- 10.”Delay_from_due_date” - Dias de Atraso da Fatura
- 11.”Num_of_Delayed_Payment” - Número de Empréstimos Atrasados
- 12.”Outstanding_Debt” - Valor da Dívida (USD)
- 13.”Payment_of_Min_Amount” - Pagamento do Mínimo da Fatura (SIM OU NÃO)

Os tipos de variáveis contidas nesta base são:

```
str(dados)
```

```
## 'data.frame':    10165 obs. of  13 variables:
##  $ Customer_ID           : int  11426 10820 10226 10058 11926 10536
10810 12375 10672 10667 ...
##  $ Month                 : chr  "2022-01-12" "2022-01-12" "2022-01-12"
"2022-01-12" ...
##  $ Age                   : int  28 45 33 39 21 36 27 35 55 29 ...
##  $ Monthly_Inhand_Salary : num  1419 2267 2258 10524 2729 ...
##  $ Num_Bank_Accounts     : int  5 3 4 2 3 8 10 5 0 8 ...
##  $ Num_Credit_Card       : int  7 10 4 4 6 6 6 7 7 7 ...
```

```
##  $ Interest_Rate         : int  10 15 7 4 9 25 16 15 9 10 ...
##  $ Num_of_Loan           : int  4 7 2 2 2 8 5 6 3 0 ...
##  $ Type_of_Loan          : chr  "Student Loan, Home Equity Loan,
Mortgage Loan, and Mortgage Loan" "Personal Loan, Auto Loan, Auto Loan,
Mortgage Loan, Debt Consolidation Loan, Home Equity Loan, and Debt
Consolidation Loan" "Credit-Builder Loan, and Student Loan" "Home Equity
Loan, and Mortgage Loan" ...
##  $ Delay_from_due_date   : int  21 9 5 28 6 31 55 2 0 28 ...
##  $ Num_of_Delayed_Payment: int  16 17 1 12 16 24 23 9 11 11 ...
##  $ Outstanding_Debt      : num  173 1962 702 1314 1307 ...
##  $ Payment_of_Min_Amount : chr  "No" "Yes" "No" "No" ...
```

Para viabilizar a visualização gráfica das varáveis, iremos tomar uma amostra de 388 observações. O critério utilizado foi a proporção de clientes por faixas salariais.

```
#Verificando as medidas de posição da variável Salario Mensal para criar
as classes
summary(dados$Monthly_Inhand_Salary)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   303.6  1635.9  3097.0  4223.2  6001.1 15167.2
```

Sabendo os valores mínimo e máximo da variável "Salário Líquido Mensal", podemos criar classes intervalares para agrupar os clientes.

```
dados$Classes_Salarios = cut(dados$Monthly_Inhand_Salary, breaks =
c(0,2600,5200,7800,10400,13000,15600), include.lowest = TRUE,  labels =
c("(0,2600]", "(2600,5200]",
"(5200,7800]","(7800,10400]","(10400,13000]","(13000,15600]"))
```

Aqui iremos verificar a frequência absoluta das classes para tomarmos observações aleatórias de cada uma que estejam na mesma proporção da base original.

```
prop.table(table(dados$Classes_Salarios))
```

```
##
##      (0,2600]   (2600,5200]   (5200,7800]  (7800,10400] (10400,13000]
##    0.40649287    0.28135760    0.16202656    0.08785047    0.04545007
## (13000,15600]
##    0.01682243
```

```
#table(dados$Classes_Salarios))

amostra1 = dados[sample(which(dados$Classes_Salarios == "(0,2600]"),
160),]

amostra2 = dados[sample(which(dados$Classes_Salarios == "(2600,5200]"),
112),]

amostra3 = dados[sample(which(dados$Classes_Salarios == "(5200,7800]"),
64),]
```

```
amostra4 = dados[sample(which(dados$Classes_Salarios == "(7800,10400]"),
32),]

amostra5 = dados[sample(which(dados$Classes_Salarios == "(10400,13000]"),
16),]

amostra6 = dados[sample(which(dados$Classes_Salarios == "(13000,15600]"),
4),]

# Concatenando as amostras em um único dataframe
dados_amostrados = rbind(amostra1, amostra2, amostra3, amostra4,
amostra5, amostra6)
```

Agora podemos verificar a amostra que usaremos para as análises posteriores.

```
str(dados_amostrados)

## 'data.frame':    388 obs. of  14 variables:
##  $ Customer_ID           : int  19 222 868 4349 725 9906 11603 5089
6720 4556 ...
##  $ Month                 : chr  "2022-01-12" "2022-01-12" "2022-01-12"
"2022-01-12" ...
##  $ Age                   : int  32 33 31 41 25 39 36 25 35 29 ...
##  $ Monthly_Inhand_Salary : num  1512 2372 1240 842 2290 ...
##  $ Num_Bank_Accounts     : int  6 4 10 10 2 3 5 3 6 7 ...
##  $ Num_Credit_Card       : int  7 5 7 6 4 3 10 5 7 7 ...
##  $ Interest_Rate         : int  17 6 27 17 8 13 15 12 22 22 ...
##  $ Num_of_Loan           : int  5 3 8 8 0 4 3 4 6 2 ...
##  $ Type_of_Loan          : chr  "Mortgage Loan, Debt Consolidation
Loan, Payday Loan, Auto Loan, and Not Specified" "Not Specified, Credit-
Builder Loan, and Personal Loan" "Mortgage Loan, Personal Loan, Student
Loan, Home Equity Loan, Home Equity Loan, Credit-Builder Loan, Auto
Loan,"| __truncated__ "Home Equity Loan, Payday Loan, Auto Loan, Credit-
Builder Loan, Payday Loan, Debt Consolidation Loan, Home Equit"|
__truncated__ ...
##  $ Delay_from_due_date   : int  51 35 60 18 8 8 18 0 34 30 ...
##  $ Num_of_Delayed_Payment: int  22 17 24 21 12 12 12 5 17 19 ...
##  $ Outstanding_Debt      : num  2430 644 1790 3378 181 ...
##  $ Payment_of_Min_Amount : chr  "Yes" "No" "Yes" "Yes" ...
##  $ Classes_Salarios      : Factor w/ 6 levels
"(0,2600]","(2600,5200]",..: 1 1 1 1 1 1 1 1 1 1 ...
```

3. Análise Descritiva

Medidas de posição e dispersão

Através da função `summary` e `sd`, podemos ter as principais medidas de posição e desvio padrão das variáveis.

```
#Obtendo algumas medidas de posição
summary(dados_amostrados[,c("Age","Interest_Rate","Num_of_Loan")])
```

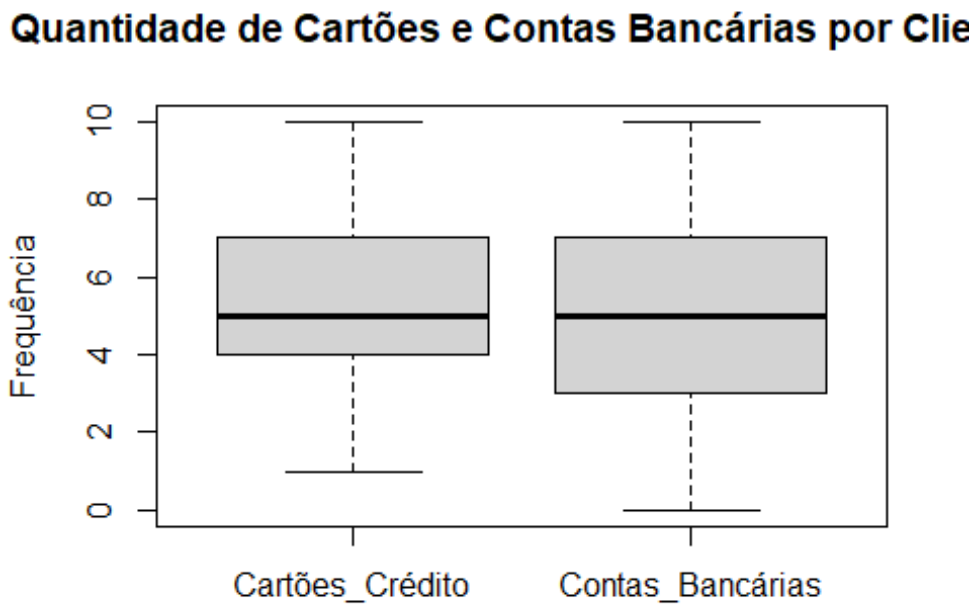
```
##       Age        Interest_Rate    Num_of_Loan
##  Min.   :18.00   Min.   : 1.00   Min.   :0.000
##  1st Qu.:26.00   1st Qu.: 7.00   1st Qu.:2.000
##  Median :35.00   Median :13.00   Median :3.000
##  Mean   :34.46   Mean   :13.65   Mean   :3.376
##  3rd Qu.:41.25   3rd Qu.:20.00   3rd Qu.:5.000
##  Max.   :56.00   Max.   :30.00   Max.   :9.000
```

```
#Obtendo o desvio padrão
desvio_padrao =
apply(dados_amostrados[,c("Age","Interest_Rate","Num_of_Loan")], 2, sd)
desvio_padrao
```

```
##           Age Interest_Rate   Num_of_Loan
##     10.176257      7.474786      2.484610
```

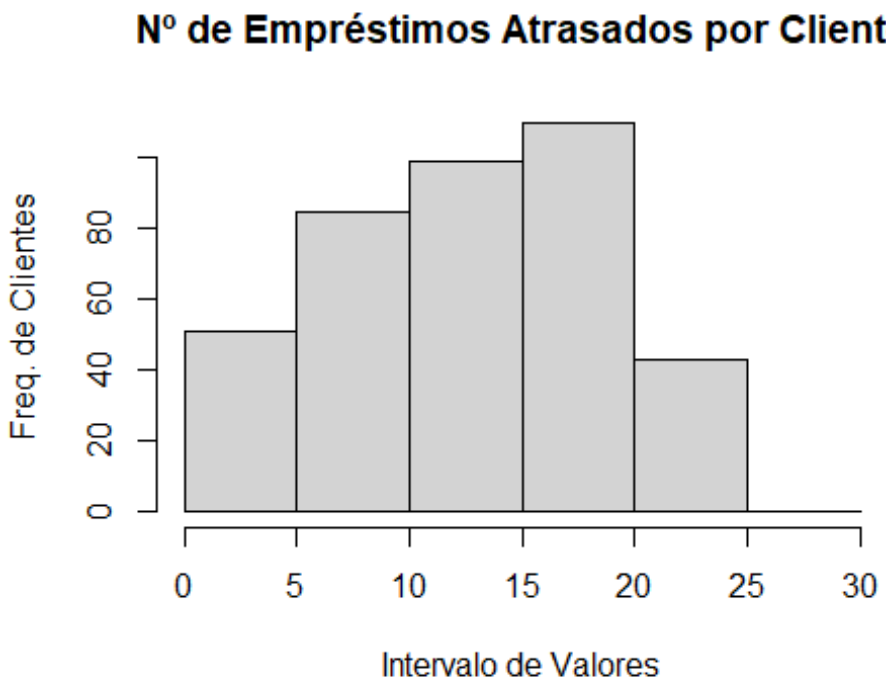
Outra forma de se obter informações sobre as variáveis é através de métodos gráficos.

```
boxplot(list(Cartões_Crédito = dados_amostrados$Num_Credit_Card,
Contas_Bancárias = dados_amostrados$Num_Bank_Accounts),
        main = "Quantidade de Cartões e Contas Bancárias por Cliente",
       ylab = "Frequência")
```



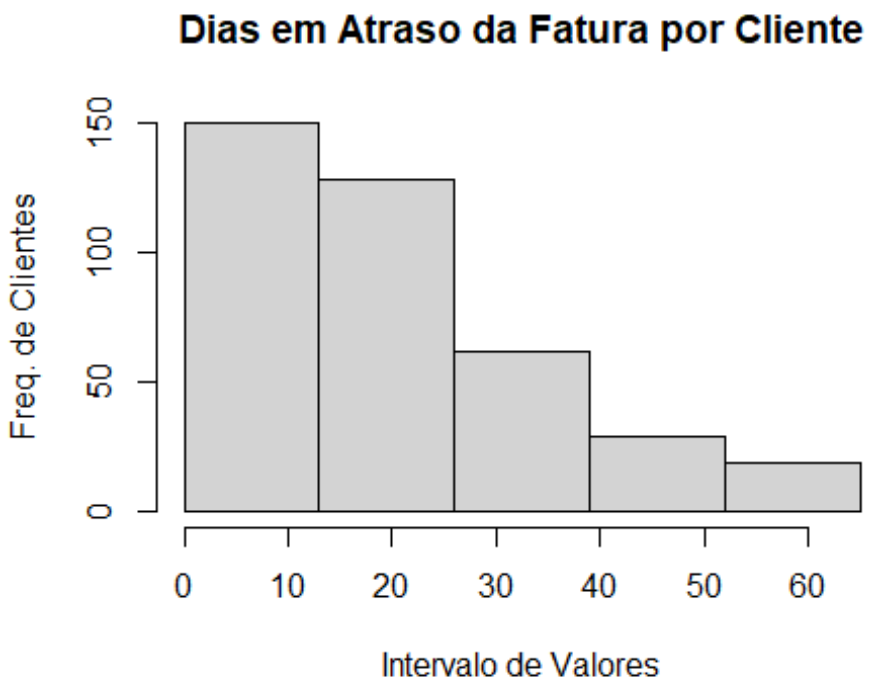
```r
#1. Histograma: Quantidade de Empréstimos Atrasados por Cliente
  #1.1. Agrupando os valores em classes
classes_num_of_delayed_payment = seq(0, 30, by = 5)

  #1.2. Plotando o histograma a partir do agrupamento
hist(dados_amostrados$Num_of_Delayed_Payment, breaks =
classes_num_of_delayed_payment, main = "Nº de Empréstimos Atrasados por
Cliente", xlab = "Intervalo de Valores", ylab = "Freq. de Clientes")
```

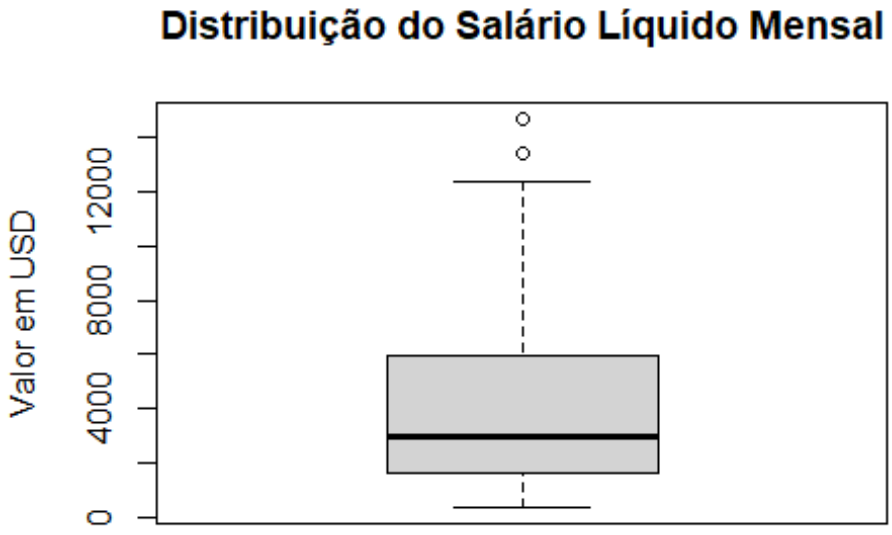



```r
#1. Histograma: Dias de Atraso da Fatura por Cliente
  #1.1. Agrupando os valores em classes
classes_delay_from_due_date = seq(0, 65, by = 13)

  #1.2. Plotando o histograma a partir do agrupamento
hist(dados_amostrados$Delay_from_due_date, breaks =
classes_delay_from_due_date, main = "Dias em Atraso da Fatura por
Cliente", xlab = "Intervalo de Valores", ylab = "Freq. de Clientes")
```



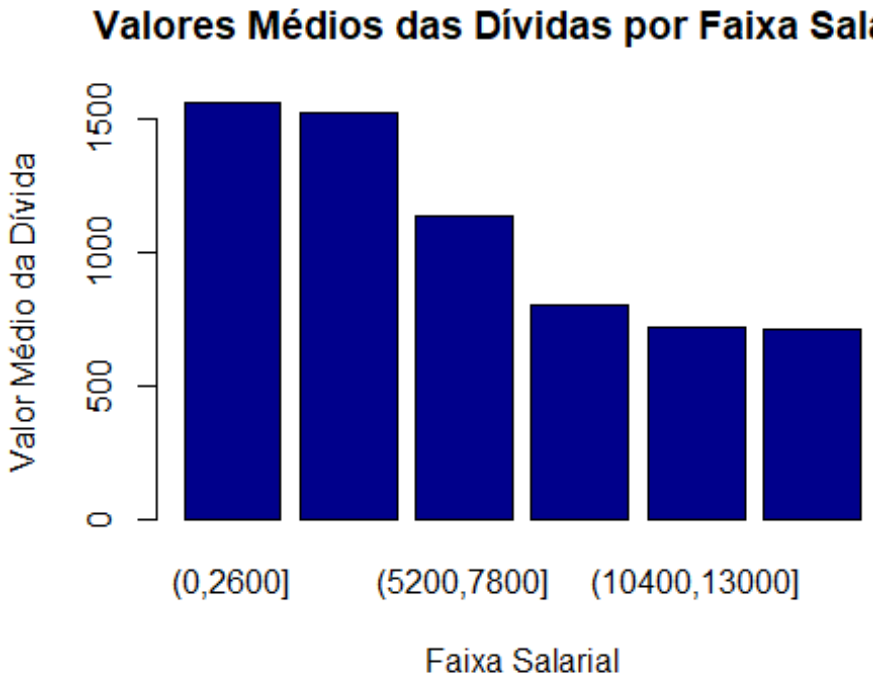

```r
#Boxplot: Salário Líquido Mensal em USD
boxplot(dados_amostrados$Monthly_Inhand_Salary, main = "Distribuição do
Salário Líquido Mensal", ylab = "Valor em USD")
```



```r
#Agrupando as faixas salariais por valor médio da dívida de cada uma
agrupamento_por_divida = aggregate(dados_amostrados$Outstanding_Debt ~
dados_amostrados$Classes_Salarios, data = dados_amostrados, FUN = mean)
agrupamento_por_divida
```

```
##   dados_amostrados$Classes_Salarios dados_amostrados$Outstanding_Debt
## 1                         (0,2600]                          1560.0230
## 2                      (2600,5200]                          1525.4540
## 3                      (5200,7800]                          1137.1745
## 4                     (7800,10400]                           799.1625
## 5                    (10400,13000]                           719.5812
## 6                    (13000,15600]                           707.4225
```

```r
barplot(agrupamento_por_divida$`dados_amostrados$Outstanding_Debt`,
names.arg = agrupamento_por_divida$`dados_amostrados$Classes_Salarios`,
main = "Valores Médios das Dívidas por Faixa Salarial", xlab = "Faixa
Salarial", ylab = "Valor Médio da Dívida", col = "darkblue")
```



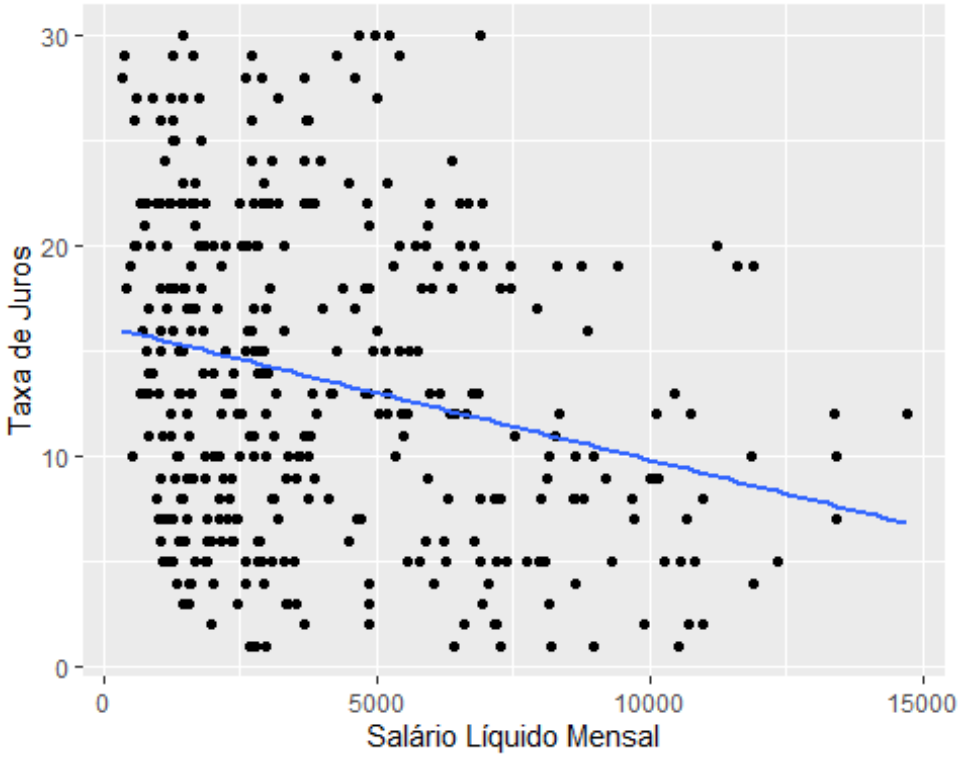
Associações

Aqui procuraremos associações entre as variáveis

```r
library("ggplot2")

grafico = ggplot(data = dados_amostrados, aes(x = Monthly_Inhand_Salary,
y = Interest_Rate))
grafico + geom_point() + geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Salário Líquido Mensal", y = "Taxa de Juros")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Através do gráfico acima é possível verificar uma certa associação entre “Salário Líquido Mensal” e “Taxa de Juros”. Porém, se trata de uma associação negativa e fraca, ou seja, os clientes com os menores salários sofrem com mais juros em relação á clientes com salários maiores.
Para entender um pouco melhor essa relação, chamamos a função corpara gerar um coeficiente de correlação de Person.

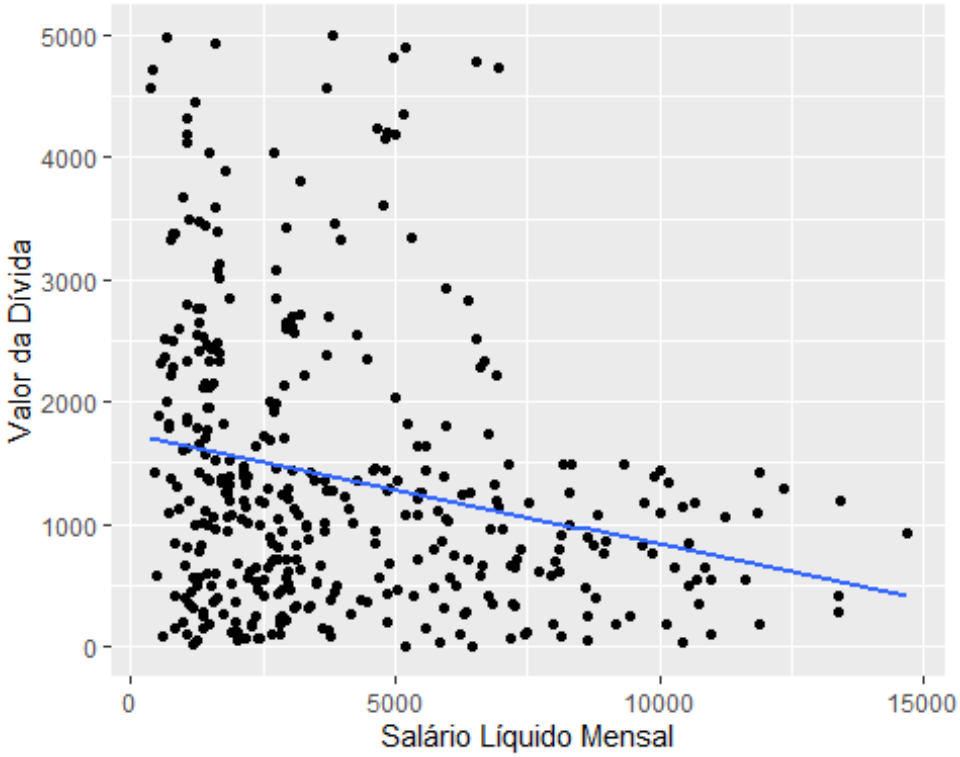
```
cor(dados_amostrados$Monthly_Inhand_Salary,
dados_amostrados$Interest_Rate)

## [1] -0.2578687
```

Para conhecer um pouco mais sobre a associação de outras variáveis com o “Salário Mensal”, geramos mais um gráfico com a variável “Valor da dívida”

```
grafico2 = ggplot(data = dados_amostrados, aes(x = Monthly_Inhand_Salary,
y = Outstanding_Debt))
grafico2 + geom_point() + geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Salário Líquido Mensal", y = "Valor da Dívida")

## `geom_smooth()` using formula = 'y ~ x'
```



E novamente, analisaremos em conjunto com o coeficiente de correlação de Pearson.

```
cor(dados_amostrados$Delay_from_due_date,
dados_amostrados$Outstanding_Debt)

## [1] 0.6149205
```

Uma das variáveis desta base que sabemos que pode ter ligação com a taxa de juros é se o cliente pagou o mínimo da fatura ou não. Por isso, geramos uma tabela de contingência entre as variáveis “Classes Salariais” e “Pagamento Mínimo da Fatura” para conhecer melhor o comportamento dos dados desses clientes.

```
#Cruzando faixas salariais com pagamento mínimo da fatura
table(dados_amostrados$Classes_Salarios,
dados_amostrados$Payment_of_Min_Amount)

##
##                   No Yes
##   (0,2600]        51 109
##   (2600,5200]     39  73
##   (5200,7800]     30  34
##   (7800,10400]    23   9
##   (10400,13000]   12   4
##   (13000,15600]    4   0
```

4. Análise Inferencial

Teste de Shapiro-Wilk

Para trabalharmos com técnicas inferenciais, é interessante conhecer a distribuição dos dados em estudo. A distribuição normal permite a aplicação de diversos métodos e, para testarmos se as variáveis em questão seguem distribuição normal, aplicaremos o teste de Shapiro-Wilk.

O teste de Shapiro-Wilk é um teste estatístico que verifica se uma determinada amostra de dados segue uma distribuição normal ou não.
No R, o teste de Shapiro-Wilk pode ser aplicado usando a função `shapiro.test`. Esta função recebe como entrada um vetor de dados e retorna o resultado do teste, incluindo o valor da estatística de teste e o valor-p.
Se o valor-p for maior do que o nível de significância (geralmente 0,05), então não podemos rejeitar a hipótese nula de que a amostra segue uma distribuição normal. Por outro lado, se o valor-p for menor do que o nível de significância, então podemos rejeitar a hipótese nula e concluir que a amostra não segue uma distribuição normal.

```
shapiro.test(dados_amostrados$Outstanding_Debt)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dados_amostrados$Outstanding_Debt
## W = 0.87609, p-value < 2.2e-16
```

Intervalo de confiança para estimação da média com distribuição e desvio padrão populacionais desconhecidos.

```
# Calculando a média e o desvio padrão da amostra
media_amostral = mean(dados_amostrados$Monthly_Inhand_Salary)
desvio_padrao_amostral = sd(dados_amostrados$Monthly_Inhand_Salary)

# Definindo o nível de confiança
confianca = 0.95

# Calculando o tamanho da amostra
n = length(dados_amostrados$Monthly_Inhand_Salary)

# Calculando o erro padrão da média
erro_padrao = desvio_padrao_amostral / sqrt(n)

# Calculando o valor crítico t para o nível de confiança desejado e (n-1)
graus de liberdade
valor_critico = qt((1 - confianca) / 2, df = n - 1)

# Calculando os limites inferior e superior do intervalo de confiança
limite_inferior = media_amostral - valor_critico * erro_padrao
limite_superior = media_amostral + valor_critico * erro_padrao
```

```r
# Resultado do intervalo de confiança
cat("O intervalo de confiança de", confianca * 100, "% para a média
populacional é [", round(limite_inferior, 2), ", ",
round(limite_superior, 2), "].\n")
```

```
## O intervalo de confiança de 95 % para a média populacional é [ 4332.57
,  3728.56 ].
```