R e Data Science: como entender o que dizem os dados com o R?

Universidade Newton Paiva - Belo Horizonte

Vítor Wilher

Cientista de Dados | Mestre em Economia

Plano de Voo

Introdução

Exemplo: previsão de *Churn* em uma operadora de telecom

Slides disponíveis

Introdução

O avanço da informática e das telecomunicações possibilitou o armazenamento e a distribuição de conjuntos de dados cada vez mais complexos. Lidar com essas bases de dados exigiu a sistematização de diversas técnicas de coleta, tratamento, análise e apresentação de dados.

Essa sistematização de técnicas deu origem ao que hoje chamamos de **data science**, cujo objetivo principal é extrair informações úteis de conjuntos de dados aparentemente confusos.

Introdução

Algumas aplicações interessantes...

- Identificar mensagens indesejáveis em um e-mail (spam);
- Segmentação do comportamento de consumidores para propagandas direcionadas;
- Redução de fraudes em transações de cartão de crédito;
- Predição de eleições;
- Otimização do uso de energia em casas ou prédios;
- etc, etc, etc...

Introdução

De modo a responder esse tipo de pergunta, é necessário cumprir aquelas quatro etapas da ciência de dados.

- É preciso **coletar** os dados;
- Dados brutos precisam ser **tratados**;
- Uma vez disponíveis, os dados precisam ser **analisados** de forma a extrair informações relevantes e/ou responder determinados questionamentos;
- Com as respostas em mãos, é preciso **apresentar** os resultados.

Introdução

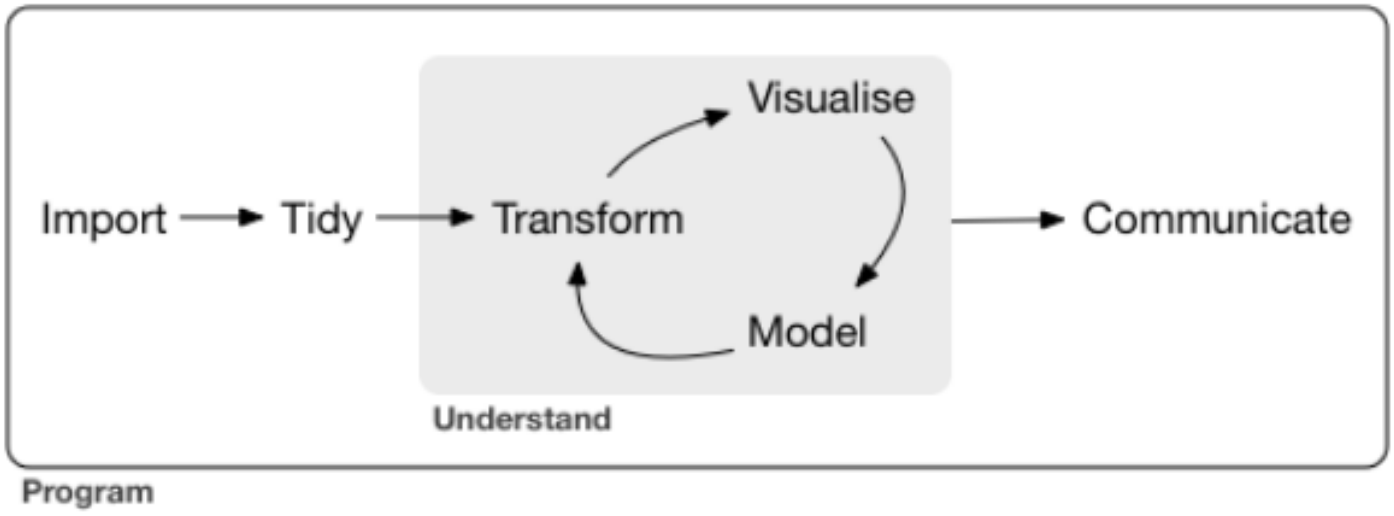


Figure 1: Fonte: R for Data Science.

Cada uma dessas etapas exige conhecimentos específicos, de modo a lidar com diferentes formatos de dados, bem como responder questões distintas.

Introdução

Era necessário construir uma plataforma que unisse todas essas etapas. O **R** é uma das melhores soluções atualmente disponíveis, dados os seguintes motivos:

- A existência de uma comunidade grande e bastante entusiasmada, que compartilha conhecimento todo o tempo;
- o **R** é gratuito, *open source*, de modo que você não precisa comprar licenças de software para instalá-lo;
- Tem inúmeras bibliotecas (pacotes) em estatística, *machine learning*, visualização, importação e tratamento de dados;
- Possui uma linguagem estabelecida para *data analysis*;
- Ferramentas poderosas para comunicação dos resultados da sua pesquisa, seja em forma de um website ou em pdf.

Introdução

Ao aprender **R**, você conseguirá integrar as etapas de coleta, tratamento, análise e apresentação de dados em um único ambiente. Você vai esquecer ter de abrir o excel, algum pacote estatístico, depois o power point ou o word, depois um compilador de pdf para gerar seu relatório. Todas essas etapas serão feitas em um único ambiente. E essa talvez seja a grande motivação para você entrar de cabeça nesse mundo.

Introdução

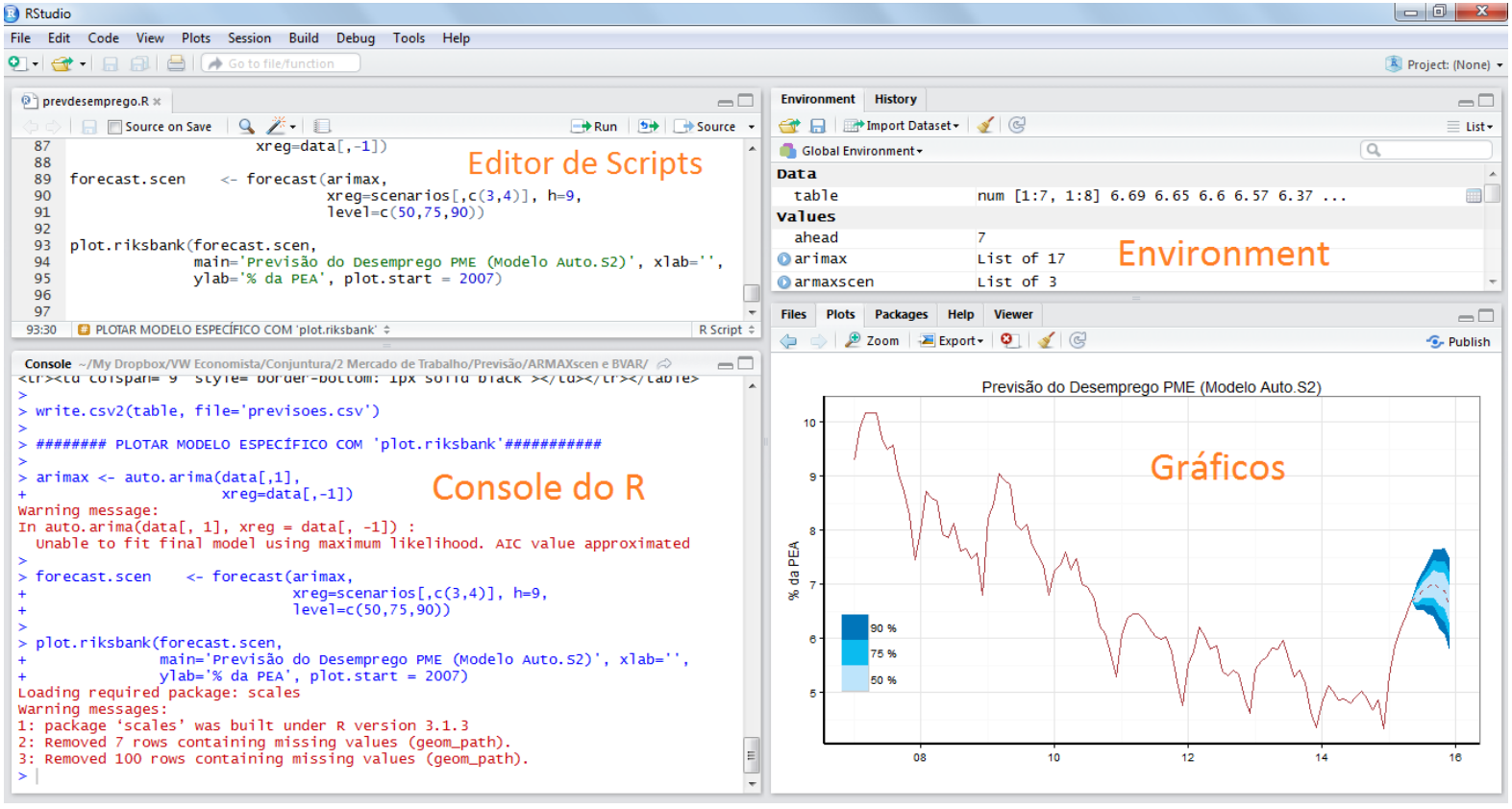


Figure 2: Ambiente do RStudio.

Introdução

1. Baixe o **R** em http://cran-r.c3sl.ufpr.br/;
2. Baixe o **RStudio** em https://www.rstudio.com/products/rstudio/download/;
3. Baixe o **MikTex** *se você for usuário de Windows* em http://miktex.org/download;
4. Baixe o **MacTex** *se você for usuário de Mac* em http://www.tug.org/mactex/.

Exemplo: previsão de *Churn* em uma operadora de telecom

A rotatividade de clientes ocorre quando clientes ou assinantes param de fazer negócios com uma empresa ou serviço, também conhecido como atrito com clientes. Também é referido como perda de clientes ou simplesmente *churn*. Um setor no qual as taxas de cancelamento são particularmente úteis é o setor de telecomunicações. Vamos prever a rotatividade de clientes usando um conjunto de dados de telecomunicações disponível no site da IBM, com base em modelos de regressão logística e Árvore de Decisão.

Pacotes utilizados

```
## Carregar pacotes necessários
library(plyr)
library(corrplot)
library(ggplot2)
library(gridExtra)
library(ggthemes)
library(MASS)
library(caret)
library(randomForest)
library(party)
library(stargazer)
```

Coleta de Dados

Os dados foram transferidos por download do IBM Sample Data Sets. Cada linha representa um cliente, cada coluna contém os atributos desse cliente:

```
churn <- read.csv('Telco-Customer-Churn.csv')
str(churn)
```

```
## 'data.frame':    7043 obs. of  21 variables:
##  $ customerID      : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",..: 5376 3963 2565 5536 6512 6552
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
##  $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
##  $ MultipleLines   : Factor w/ 3 levels "No","No phone service",..: 2 1 1 2 1 3 3 2 3 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2 2 2 1 2 1 ...
##  $ OnlineSecurity  : Factor w/ 3 levels "No","No internet service",..: 1 3 3 3 1 1 1 3 1 3 ...
##  $ OnlineBackup    : Factor w/ 3 levels "No","No internet service",..: 3 1 3 1 1 1 3 1 1 3 ...
##  $ DeviceProtection: Factor w/ 3 levels "No","No internet service",..: 1 3 1 3 1 3 1 1 3 1 ...
##  $ TechSupport     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ StreamingTV     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 3 1 3 1 ...
##  $ StreamingMovies : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 1 1 3 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1 1 1 1 2 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 4 4 1 3 3 2 4 3 1 ...
##  $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...
```

Coleta de Dados

As variáveis contidas no *dataset* são:

- customerID
- gender (female, male)
- SeniorCitizen (Whether the customer is a senior citizen or not (1, 0))
- Partner (Whether the customer has a partner or not (Yes, No))
- Dependents (Whether the customer has dependents or not (Yes, No))
- tenure (Number of months the customer has stayed with the company)
- PhoneService (Whether the customer has a phone service or not (Yes, No))
- MultipleLines (Whether the customer has multiple lines r not (Yes, No, No phone service)

Coleta de Dados

- InternetService (Customer's internet service provider (DSL, Fiber optic, No)
- OnlineSecurity (Whether the customer has online security or not (Yes, No, No internet service)
- OnlineBackup (Whether the customer has online backup or not (Yes, No, No internet service)
- DeviceProtection (Whether the customer has device protection or not (Yes, No, No internet service)
- TechSupport (Whether the customer has tech support or not (Yes, No, No internet service)

Coleta de Dados

- streamingTV (Whether the customer has streaming TV or not (Yes, No, No internet service)
- streamingMovies (Whether the customer has streaming movies or not (Yes, No, No internet service)
- Contract (The contract term of the customer (Month-to-month, One year, Two year)
- PaperlessBilling (Whether the customer has paperless billing or not (Yes, No))
- PaymentMethod (The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic)))
- MonthlyCharges (The amount charged to the customer monthly)
- TotalCharges (The total amount charged to the customer)
- Churn (Whether the customer churned or not (Yes or No))

Tratamento dos dados

Os dados brutos contém 7043 linhas (clientes) e 21 colunas (recursos). A coluna *Churn* é o nosso alvo. Usamos todas as outras colunas como recursos do nosso modelo. Usamos `sapply` para verificar o número, se houver valores ausentes em cada coluna. Descobrimos que há 11 valores ausentes nas colunas *TotalCharges*. Então, vamos remover essas linhas com valores ausentes.

Tratamento dos dados

```
sapply(churn, function(x) sum(is.na(x)))
```

```
##       customerID           gender    SeniorCitizen          Partner
##                0                0                0                0
##       Dependents           tenure     PhoneService    MultipleLines
##                0                0                0                0
##  InternetService   OnlineSecurity     OnlineBackup DeviceProtection
##                0                0                0                0
##      TechSupport      StreamingTV  StreamingMovies         Contract
##                0                0                0                0
## PaperlessBilling    PaymentMethod   MonthlyCharges     TotalCharges
##                0                0                0               11
##            Churn
##                0
```

Tratamento dos dados

```
churn <- churn[complete.cases(churn), ]
```

Retirados os *missing values*, agora nós trocamos *No internet service* para *No* em seis colunas: *OnlineSecurity*, *OnlineBackup*, *DeviceProtection*, *TechSupport*, *streamingTV* e *streamingMovies*.

Tratamento dos dados

```
cols_recode1 <- c(10:15)
for(i in 1:ncol(churn[,cols_recode1])) {
        churn[,cols_recode1][,i] <- as.factor(mapvalues
                                              (churn[,cols_recode1][,i],
                                                from =c("No internet service"),to=c("No")))
}
```

Tratamento dos dados

Agora nós trocamos *No phone service* para *No* na coluna *MultipleLines*.

```
churn$MultipleLines <- as.factor(mapvalues(churn$MultipleLines,
                                           from=c("No phone service"),
                                           to=c("No")))
```

Tratamento dos dados

A posse (*tenure*) mínima de uma linha nessa empresa é de um mês e a máxima de é de 72 meses. Nós podemos agrupar essa posse em cinco categorias: “0–12 Month”, “12–24 Month”, “24–48 Months”, “48–60 Month” e “$>$ 60 Month”.

```
min(churn$tenure); max(churn$tenure)
```

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

```
## [1] 72
```

```
# Criar função
group_tenure <- function(tenure){
    if (tenure >= 0 & tenure <= 12){
        return('0-12 Month')
    }else if(tenure > 12 & tenure <= 24){
        return('12-24 Month')
    }else if (tenure > 24 & tenure <= 48){
        return('24-48 Month')
    }else if (tenure > 48 & tenure <=60){
        return('48-60 Month')
    }else if (tenure > 60){
        return('> 60 Month')
    }
}
```

Tratamento dos dados

```
# Aplicar função sobre coluna tenure
churn$tenure_group <- sapply(churn$tenure,group_tenure)
churn$tenure_group <- as.factor(churn$tenure_group)
```

Tratamento dos dados

Agora, mudamos os valores na coluna “SeniorCitizen” de 0 e 1 para “No” ou “Yes”.

```
churn$SeniorCitizen <- as.factor(mapvalues(churn$SeniorCitizen,
                                           from=c("0","1"),
                                           to=c("No", "Yes")))
```

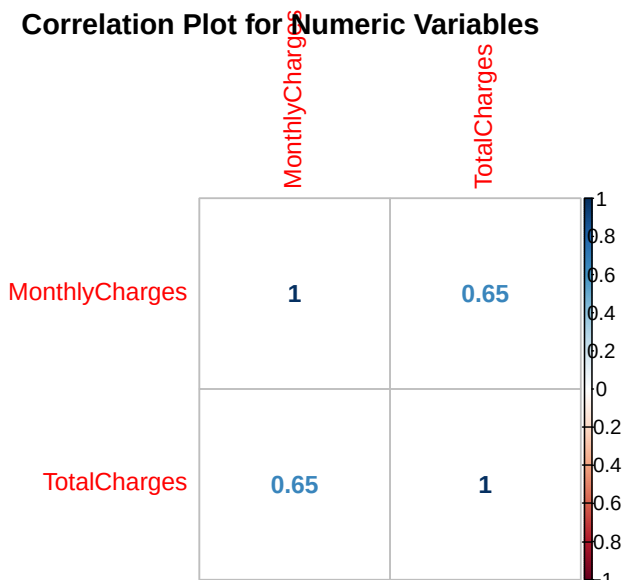
Tratamento dos dados

Por fim, removemos as colunas que não iremos utilizar.

```
churn$customerID <- NULL
churn$tenure <- NULL
```

Análise Exploratória dos Dados

```
numeric.var <- sapply(churn, is.numeric) ## Find numerical variables
corr.matrix <- cor(churn[,numeric.var])  ## Calculate the correlation matrix
corrplot(corr.matrix, main="\n\nCorrelation Plot for Numeric Variables", method="number")
```

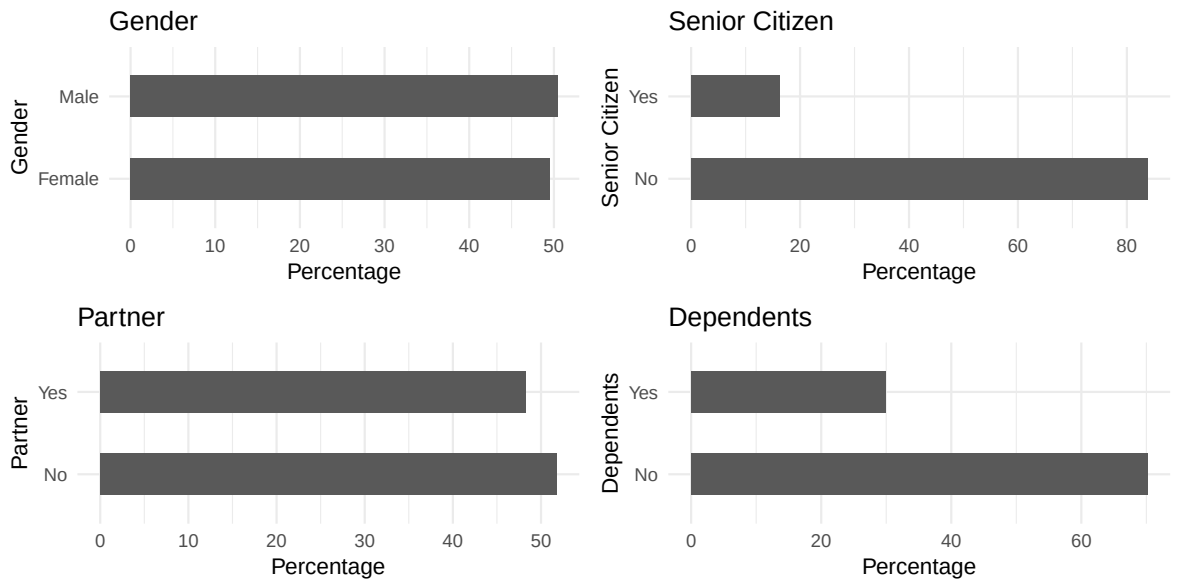


Análise Exploratória dos Dados

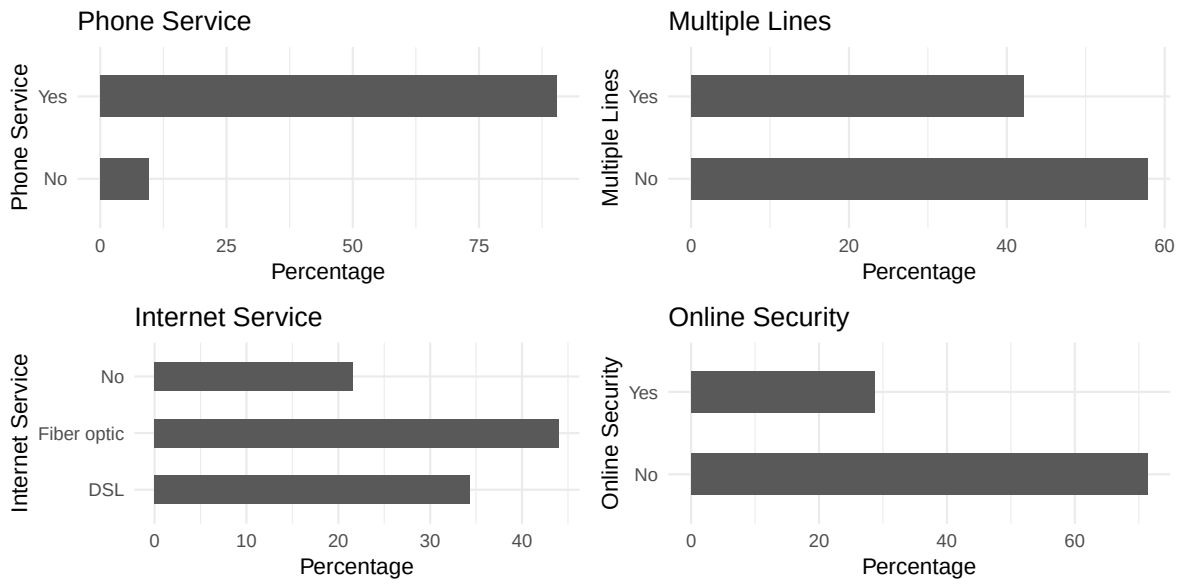
AS variáveis *Monthly Charges* e *Total Charges* são correlacionadas. Vamos utilizar apenas uma delas.

```
churn$TotalCharges <- NULL
```

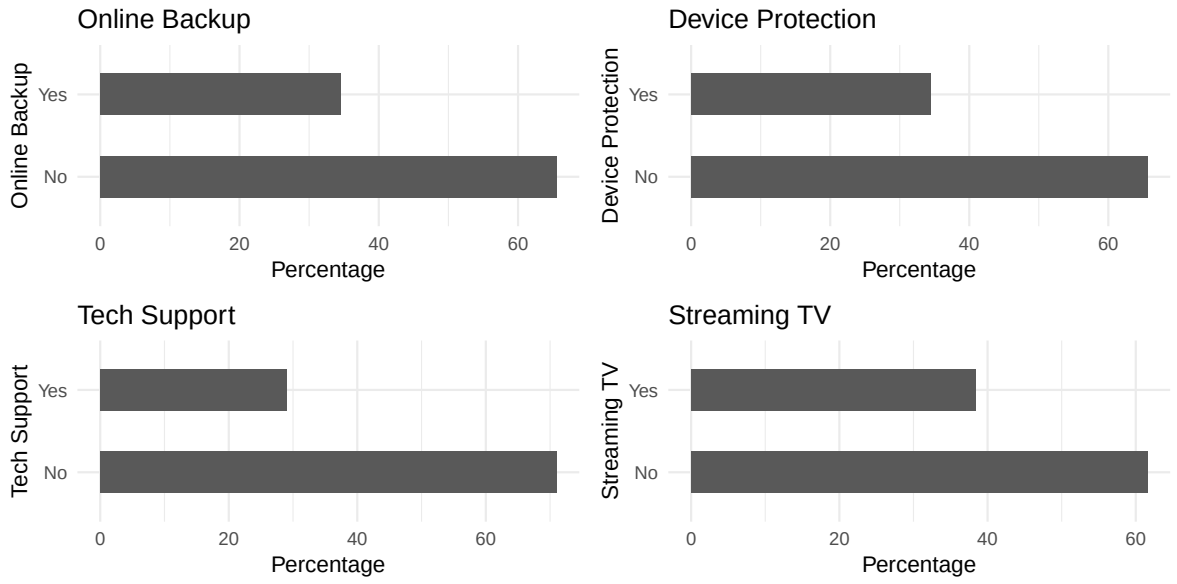
Análise Exploratória dos Dados



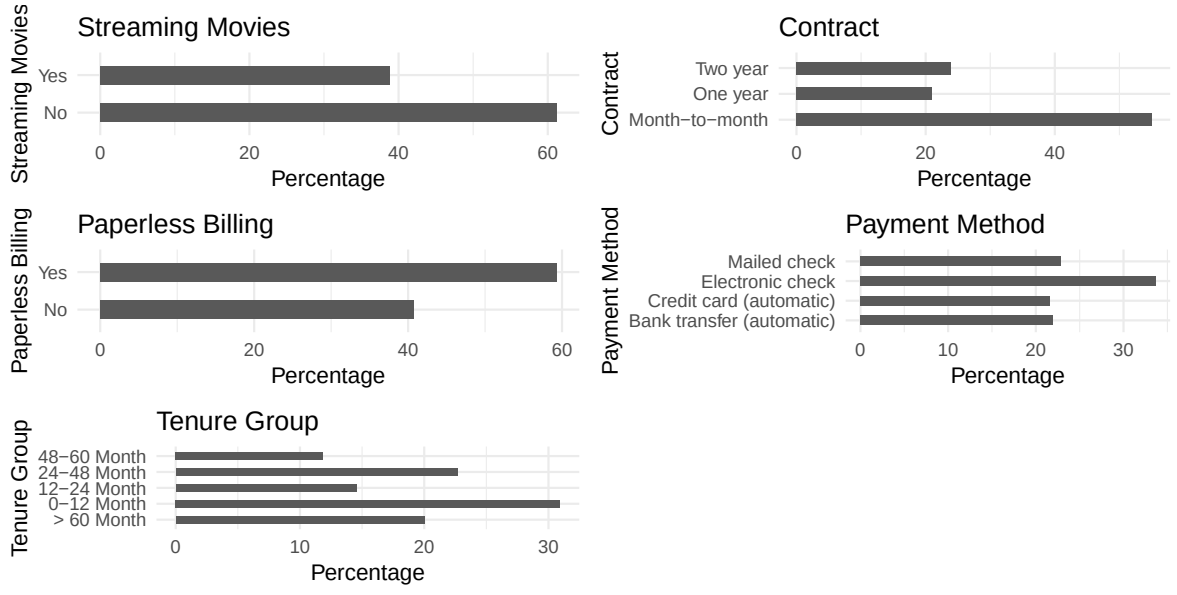
Análise Exploratória dos Dados



Análise Exploratória dos Dados



Análise Exploratória dos Dados



Todas as variáveis categóricas têm uma distribuição ampla razoável, portanto, todas elas serão mantidas para análise posterior.

Regressão Logística

Criar os conjuntos de treino (*training*) e de teste (*test*).

```
intrain <- createDataPartition(churn$Churn,p=0.7,list=FALSE)
set.seed(2017)
training <- churn[intrain,]
testing <- churn[-intrain,]
```

Confirmamos se a divisão está correta.

```
dim(training); dim(testing)
```

```
## [1] 4924   19
```

```
## [1] 2108   19
```

Regressão Logística

Estimamos, então, o modelo logístico.

```
LogModel <- glm(Churn ~ ., family=binomial(link="logit"),data=training)
summary(LogModel)
```

```
##
## Call:
## glm(formula = Churn ~ ., family = binomial(link = "logit"), data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9579  -0.6776  -0.2837   0.6596   3.1817
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -1.626791   0.976087  -1.667 0.095585
## genderMale                        -0.092963   0.077780  -1.195 0.232010
## SeniorCitizenYes                   0.323929   0.100361   3.228 0.001248
## PartnerYes                         0.048106   0.093049   0.517 0.605159
## DependentsYes                     -0.108615   0.107447  -1.011 0.312079
## PhoneServiceYes                   -0.182063   0.769394  -0.237 0.812942
## MultipleLinesYes                   0.339817   0.210521   1.614 0.106489
## InternetServiceFiber optic         1.248278   0.942856   1.324 0.185525
## InternetServiceNo                 -1.203082   0.955622  -1.259 0.208048
## OnlineSecurityYes                 -0.307312   0.211570  -1.453 0.146353
## OnlineBackupYes                   -0.145425   0.207346  -0.701 0.483076
## DeviceProtectionYes                0.206816   0.209178   0.989 0.322807
## TechSupportYes                    -0.247066   0.213583  -1.157 0.247366
## StreamingTVYes                     0.346754   0.387548   0.895 0.370928
## StreamingMoviesYes                 0.442755   0.386127   1.147 0.251523
## ContractOne year                  -0.794302   0.129177  -6.149 7.80e-10
## ContractTwo year                  -1.783629   0.219007  -8.144 3.82e-16
```

Regressão Logística

A tabela 1 ilustra o modelo. As três variáveis mais relevantes para explicar *Churn* são: Contract, Paperless Billing e tenure group. A seguir, analisando a tabela de desvio, podemos ver a queda no desvio ao adicionar cada variável uma de cada vez. Adicionar InternetService, Contract e tenure_group reduz significativamente o desvio residual. As outras variáveis, como PaymentMethod e Dependents, parecem melhorar menos o modelo, embora todos tenham p-valores baixos.

```
anova(LogModel, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Churn
##
## Terms added sequentially (first to last)
##
##
##                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                              4923     5702.8
## gender            1     1.47      4922     5701.3  0.224947
## SeniorCitizen     1   124.55      4921     5576.7 < 2.2e-16 ***
## Partner           1   107.18      4920     5469.6 < 2.2e-16 ***
```

Verificando a acurácia do modelo

```
testing$Churn <- as.character(testing$Churn)
testing$Churn[testing$Churn=="No"] <- "0"
testing$Churn[testing$Churn=="Yes"] <- "1"
fitted.results <- predict(LogModel,newdata=testing,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
misClasificError <- mean(fitted.results != testing$Churn)
print(paste('Logistic Regression Accuracy',1-misClasificError))
```

```
## [1] "Logistic Regression Accuracy 0.792220113851992"
```

Verificando a acurácia do modelo

```
logit = cbind(as.numeric(testing$Churn),
              as.numeric(fitted.results))

teste = ifelse(logit[,1]==logit[,2], "Sim", "No")

sum(teste=="Sim")/nrow(logit)
```

```
## [1] 0.7922201
```

```
sum(teste=="No")/nrow(logit)
```

```
## [1] 0.2077799
```

Odds Ratio

Uma das medidas de desempenho interessantes na regressão logística é a *Odds Ratio*. Basicamente, *odds ratios* mede a chance de um evento acontecer.

```
exp(cbind(OR=coef(LogModel), confint(LogModel)))
```

```
##                                              OR      2.5 %     97.5 %
## (Intercept)                           0.1965593 0.02896214  1.3304990
## genderMale                            0.9112272 0.78229864  1.0612381
## SeniorCitizenYes                      1.3825485 1.13556113  1.6831041
## PartnerYes                            1.0492818 0.87448106  1.2594985
## DependentsYes                         0.8970755 0.72621179  1.1067497
## PhoneServiceYes                       0.8335486 0.18450498  3.7687956
## MultipleLinesYes                      1.4046909 0.93001568  2.1231890
## InternetServiceFiber optic            3.4843395 0.54990951 22.1755897
## InternetServiceNo                     0.3002675 0.04606687  1.9530457
## OnlineSecurityYes                     0.7354209 0.48546201  1.1128678
## OnlineBackupYes                       0.8646547 0.57579057  1.2982284
## DeviceProtectionYes                   1.2297561 0.81629422  1.8537870
## TechSupportYes                        0.7810893 0.51361847  1.1867376
## StreamingTVYes                        1.4144681 0.66204231  3.0258226
## StreamingMoviesYes                    1.5569914 0.73089897  3.3219411
## ContractOne year                      0.4518965 0.34974289  0.5805122
## ContractTwo year                      0.1680273 0.10776802  0.2548021
## PaperlessBillingYes                   1.2886089 1.08106204  1.5369328
## PaymentMethodCredit card (automatic)  1.0056650 0.77003520  1.3133231
## PaymentMethodElectronic check         1.4608579 1.16937597  1.8286013
## PaymentMethodMailed check             1.0324370 0.78863262  1.3531283
## MonthlyCharges                        0.9852161 0.91528711  1.0603722
## tenure_group0-12 Month                6.1122733 4.12005065  9.1388255
```

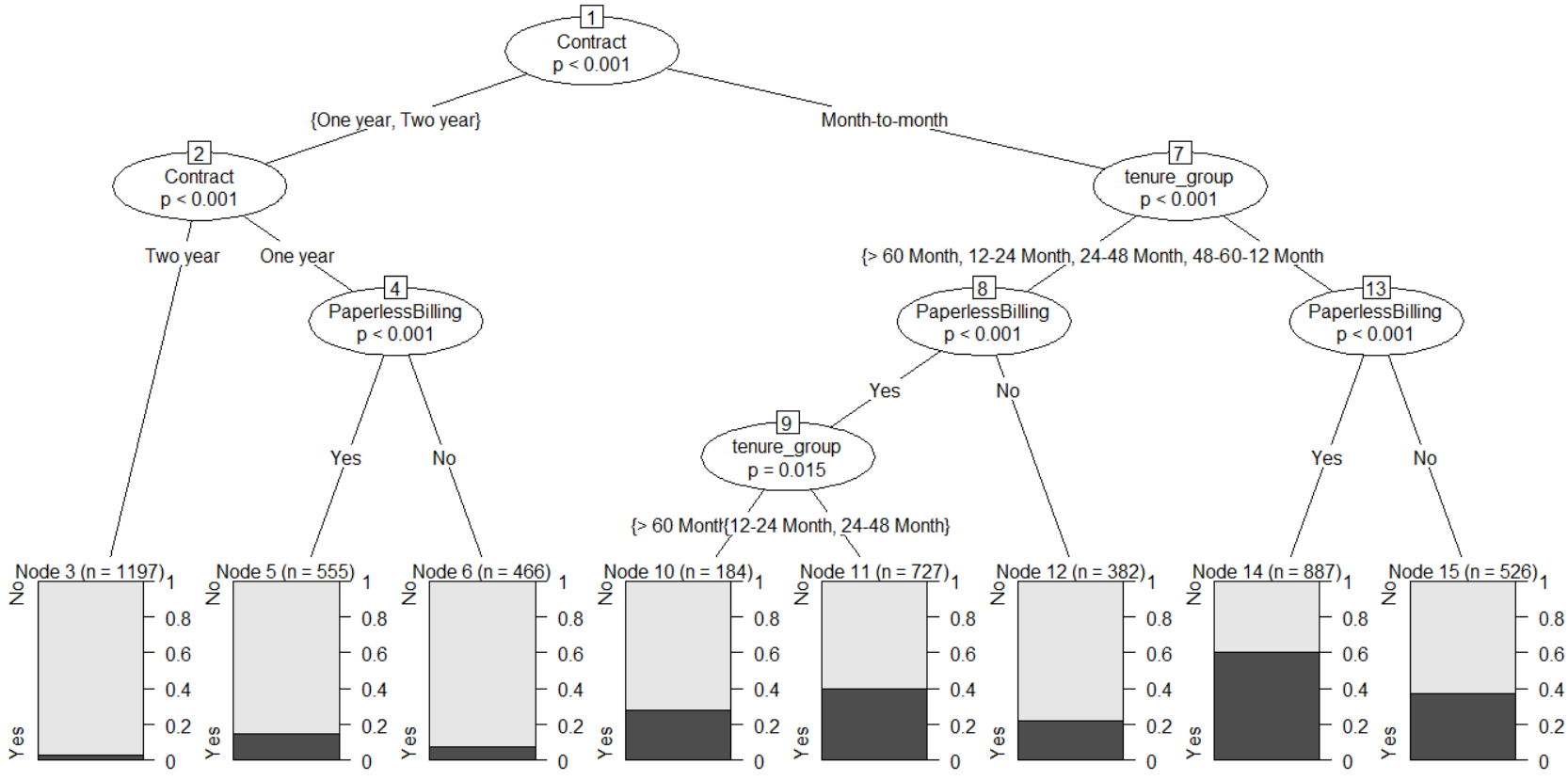
Decision Tree

Árvores de decisão são métodos de aprendizado de máquinas supervisionado não-paramétricos, muito utilizados em tarefas de classificação e regressão. Vamos utilizar uma para prever nosso *Churn*. Para ilustrarmos, vamos utilizar apenas três variáveis: "Contract", "tenure_group" e "PaperlessBilling".

```
tree <- ctree(Churn~Contract+tenure_group+PaperlessBilling, training)
```

```
plot(tree)
```

Decision Tree



Decision Tree

Das três variáveis que usamos, *Contract* é a variável mais importante para prever a rotatividade de clientes ou não. Se um cliente em um contrato de um ano ou dois anos, não importa se ele (ela) tem ou não a PapelessBilling, ele (ela) tem menos probabilidade de sair. Por outro lado, se um cliente estiver em um contrato mensal, e no grupo de posse de 0 a 12 meses, e usando o PaperlessBilling, esse cliente terá maior probabilidade de sair.[1]

[1] Na hora de rodar o código, coloque `plot(tree)` e dê um zoom para ver a árvore completa. A interpretação fica mais fácil.

Avaliando a acurácia da Árvore de Decisão

```
pred_tree <- predict(tree, testing)
tab <- table(Predicted = pred_tree, Actual = testing$Churn)
print(paste('Decision Tree Accuracy',sum(diag(tab))/sum(tab)))
```

```
## [1] "Decision Tree Accuracy 0.772296015180266"
```

Slides disponíveis

Obrigado!

Os slides e códigos dessa apresentação estarão disponíveis no Blog da Análise Macro amanhã: http://analisemacro.com.br/blog