FUNDAÇÃO GETÚLIO VARGAS
ESCOLA DE ADMINISTRAÇÃO DE EMPRESAS DE SÃO PAULO

Programa Institucional de Bolsas de Iniciação Científica (PIBIC)

BIG DATA ANALYTICS USANDO O SOFTWARE R

ANÁLISE DE DADOS PARA TOMADA DE DECISÃO GERENCIAL

Um estudo sobre Shiny

THAIS FREIRE WU

JOÃO LUIZ CHELA

São Paulo – SP

2017

BIG DATA ANALYTICS USANDO O SOFTWARE R

ANÁLISE DE DADOS PARA TOMADA DE DECISÃO GERENCIAL

Um estudo sobre Shiny

Resumo

A complexidade e competitividade do mercado faz com que seja fundamental para a sobrevivência de uma organização compreender o cenário em que está inserido. Acompanhar suas mudanças e tendências é necessário para planejar e tomar boas decisões em busca de vantagem competitiva.

Nesse contexto a análise de dados é importante porque possibilita ao gestor realizar esse acompanhamento de maneira eficaz e eficiente, otimizando suas atividades e evitando desperdícios e subaproveitamentos.

Com base nessa necessidade foi desenvolvido um aplicativo protótipo utilizando o software R, através de um pacote funcional chamado Shiny. Para o protótipo foi utilizado um banco de dados que simula os de uma companhia telefônica. Assim, buscou-se compreender o motivo de alguns clientes cancelarem o serviço, para que, posteriormente, a empresa consiga realizar uma previsão de bons e maus assinantes, reduzindo seus custos. Para tanto são utilizadas ferramentas estatísticas com enfoque gráfico que permitem ao usuário customizar seus inputs.

A partir do refinamento da programação e da possibilidade de adaptação do modelo para outros bancos de dados, acredita-se que essa pesquisa possa ser implementada na realidade das pequenas empresas.

Palavras-chaves

Análise de dados, estratégia gerencial, R, Shiny, interface gráfica iterativa.

SUMÁRIO

1. INTRODUÇÃO

1.1 Apresentação do Tema

O mercado tem se tornado cada vez mais complexo e competitivo (PETENATE, 2016), a empresa está imersa em uma rede composta de diversos players: consumidores, fornecedores, rivais, etc. Portanto se torna essencial para a organização entender o cenário em que está inserido e acompanhar as suas tendências e mudanças de modo a planejar boas estratégias gerenciais e tomar decisões em busca de vantagem competitiva.

No ambiente externo geral, no qual a empresa não possui controle direto sobre aspectos como comportamento social, políticas locais, avanços tecnológicos e impactos econômicos, ter conhecimento sobre esses elementos podem ser fontes de ameaças e oportunidades. O mesmo acontece com relação ao ambiente externo setorial, no qual a organização deverá buscar o melhor posicionamento frente as forças que nela atuam ou influenciá-las a seu favor (PORTER, 1998).

Além disso, cada organização possui diferentes recursos, competências e atividades, o que irá refletir em estratégias distintas. Possuir conhecimento do ambiente interno da organização, ou seja, conhecer o próprio negócio em profundidade, possibilita ao gestor identificar forças e fraquezas a serem trabalhadas para melhor direcionar e otimizar as suas atividades, evitando desperdícios ou subaproveitamentos.

Nesse cenário a análise de dados é importante porque possibilita tanto a identificação de sinais e tendências de maneira eficaz e eficiente, quanto acompanhar os resultados de iniciativas anteriores. Visto que a tecnologia tem sido aprimorada constantemente nos últimos anos, e com isso, a capacidade de armazenamento, tratamento e processamento de dados, em meio a toneladas de informações coletadas, encontrar oportunidades que não tenham sido percebidas pelo mercado como necessidades específicas do público alvo resultam em diferenciação no mercado e vantagem competitiva.

Portanto a capacidade de prever oportunidades, ameaças, e analisar forças e fraquezas de maneira ágil é essencial para a manutenção de uma organização. Para tal, as empresas utilizam diferentes softwares que auxiliam na análise de dados e sustentam a tomada de decisão do gestor, como é o caso do SPSS, MaxStat, Analytica e do R.

1.2 Objetivos do Trabalho

O trabalho tem como objetivo principal o desenvolvimento de um aplicativo utilizando o software R, mais especificamente, um pacote funcional desse programa chamado Shiny, com a finalidade de proporcionar a um gestor informações sucintas de uma base de dados que o auxiliem a tomar decisões com fundamento teórico de forma prática, simples e rigorosa.

Para tanto os objetivos específicos propostos são: (1) Estudo do software R – funcionamento, linguagem de programação específica, ferramentas diversas – para manipulação de dados; (2) Estudo do Shiny para a implementação do código e aplicação na web; (3) Revisão e aprofundamento de conceitos teóricos matemáticos e estatísticos; (4) Implementação de um protótipo funcional; (5) Interpretação e análise dos dados gerados; (6) Redação do relatório da pesquisa.

1.3 Contribuições Esperadas

Espera-se que o protótipo desenvolvido possa ser aplicado na realidade de algumas empresas ou servir como base para o aperfeiçoamento de aplicativos direcionados a banco de dados específicos.

Além disso, também se espera despertar e ampliar o interesse de profissionais da área a respeito das vantagens e utilidade do R.

2. CONCEITOS FUNDAMENTAIS

2.1 Estudos Sobre o Tema

A origem de problemas e decisões a serem tomadas para solucioná-los dentro das empresas surgiu no momento em que elas foram criadas. O problema pode ser de várias naturezas, como por exemplo, posicionamento de marca, ineficiência de algum processo, falta de gestão em cadeias de suprimento, etc. No entanto, a maioria delas, para serem identificadas e solucionadas de modo eficiente precisa passar por uma série de estudos, assim, o gestor disporá de embasamento para tomar uma decisão. Segundo Killman e Mitroff esse processo pode ser realizado a partir de cinco passos:

1. Sentir o problema;
2. Definir o problema;
3. Entregar soluções;
4. Implementar soluções;
5. Avaliar os resultados.

A primeira parte é quando se cria a percepção de que algo não está certo, uma situação que poderia ou deveria ser diferente, o que iremos definir como “problema”. A partir daí é necessário entender o que de fato causa o problema. Um exemplo dado por Killman e Mitroff destaca que a alta rotatividade de funcionários de uma empresa pode ser fruto tanto de um ambiente organizacional inadequado quanto por falta de seleção apropriada dos empregados. Os passos 3, 4 e 5 derivam da boa definição dos passos 1 e 2, caso contrário poderia ser implementada uma solução que não resolve o problema.

Nesse contexto, a análise de dados auxilia tanto no vislumbre de que algo está errado ou que pode se modificar no futuro, necessitando de uma estratégia de readequação, como também para indicar variáveis que podem ser o problema, ou simplesmente estarem afetando o problema principal. Além de armazenar dados para futuramente avaliar os resultados da tomada de decisão.

Para tanto, existem diferentes serviços ofertados as empresas que oferecem dispositivos de análise e controle, sendo que o ideal varia de acordo com o perfil de cada organização. Portanto não existe um único método de análise de dados, e os estudos divergem conforme características específicas de cada setor. No entanto, é essencial compreender técnicas básicas que podem ser utilizadas em cenários generalizados.

2.2 Dados Quantitativos e Qualitativos

Dados quantitativos são aqueles que utilizam valores numéricos para indicar quantidade. Nesse caso é possível realizar operações aritméticas para obter resultados significativos, como por exemplo, a média.

Já os dados qualitativos (ou categóricos) são aqueles cujo número ou símbolo representam uma determinada categoria, podendo ser agrupados. Portanto não é possível realizar operações aritméticas para extração de informação, mas alguns cálculos como a contagem de ocorrências ou cruzamento com dados quantitativos auxiliam na análise (SWEENEY, 2015).

2.3 Variância

A variância é uma medida de variabilidade que utiliza todos os dados do conjunto (SWEENEY, 2015) com o objetivo de verificar o quão distante os valores estão com relação ao valor esperado, a média $\bar{x}$.

Se os dados forem extraídos de uma população, então a variância será denominada variância populacional (σ), e sua fórmula será:

$$\sigma^2 = \frac{\sum(xi - \mu)^2}{N} \quad (1)$$

Onde N é o número de observações e, μ, a média populacional. Como os valores são elevados ao quadrado, a variância sempre será positiva.

No entanto, por questões de custo, acesso a informação ou mesmo tempo para o tratamento dos dados, dificilmente se trabalha com dados de toda uma população. Em alguns casos é necessário trabalhar com dados amostrais, um subconjunto obtido aleatoriamente de uma população, para, a partir deles, realizar inferências a respeito da população. Nesse caso, a variância é denominada variância amostral (s), e sua fórmula será:

$$s^2 = \frac{\sum(xi - \bar{x})^2}{n - 1} \quad (2)$$

Onde n é o número de observações da amostra. Importante ressaltar que o denominador será (n –1) pois é possível demonstrar matematicamente que a variância amostral fornecerá uma estimativa não enviesada da variância populacional (SWEENEY, 2015).

2.4 Desvio Padrão

Assim como a variância, o desvio padrão também é uma medida de dispersão. Seu valor é obtido através do cálculo da raiz quadrada da variância tanto para dados populacionais quanto amostrais.

$$s = \sqrt[2]{s^2} \tag{3}$$

$$\sigma = \sqrt[2]{\sigma^2} \tag{4}$$

As unidades associadas à variância dos dados são elevadas ao quadrado. Portanto, o desvio padrão é mais facilmente comparado a outras estatísticas medidas nas mesmas unidades que os valores originais (SWEENEY, 2015).

2.5 Boxplots

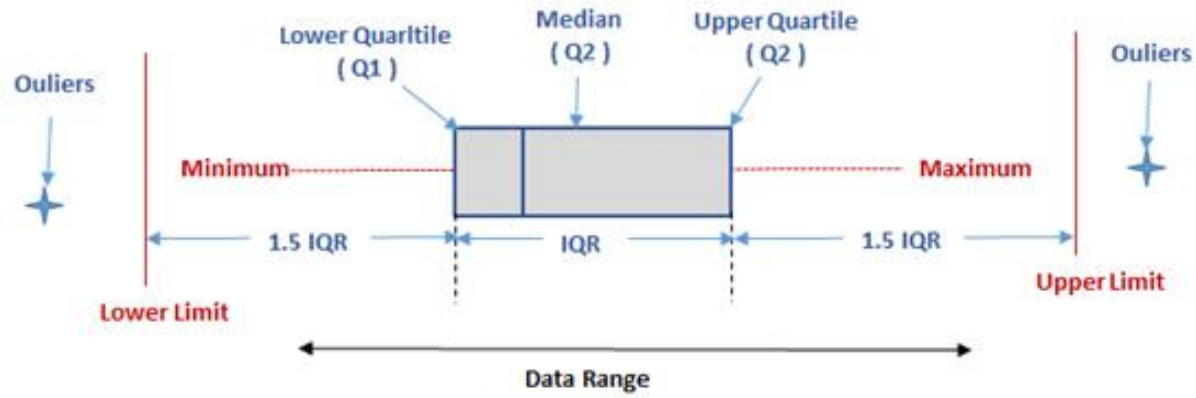
Um boxplot é um resumo gráfico de dados que se baseia na regra de cinco itens (SWEENEY, 2015):

1. Menor Valor;
2. Maior Valor;
3. Primeiro Quartil (Q1);
4. Segundo Quartil (Q2);
5. Terceiro Quartil (Q3).

Os quartis são os valores de dados localizados em determinada posição de um conjunto de dados organizados em ordem crescente. Por convenção, o número total do conjunto de dados é dividido em 4 partes iguais, a primeira parte, que contem os 25% primeiros dados indica a

posição do valor que determinará Q1. Portanto, a segunda parte, referente aos 50% primeiros dados será exatamente o valor da mediana, e assim por diante. Através destas medidas é possível calcular os limites inferiores e superiores, que serão os valores fundamentais para determinar se um dado é um outlier.

Figura 1 – Esquema de um boxplot



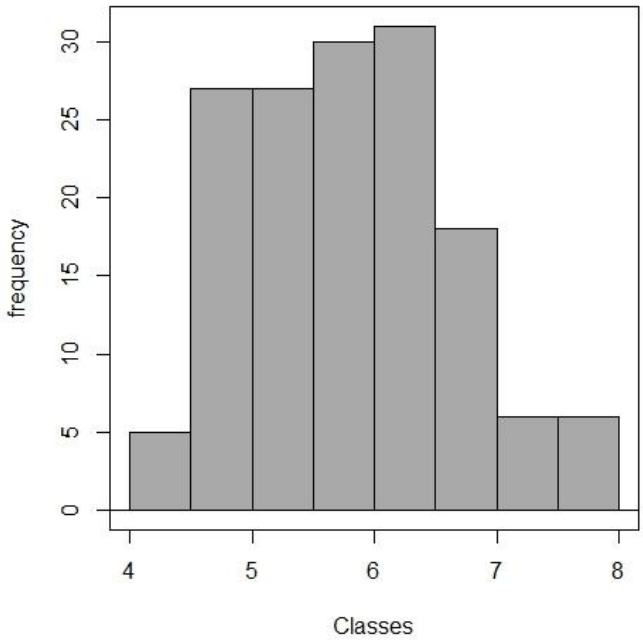
Fonte: What is Six Sigma, 2017

A grande vantagem do boxplot é a rápida visualização de outliers e a possibilidade de comparar vários grupos de dados ao mesmo tempo.

2.6 Histograma

O histograma é uma representação gráfica de um conjunto de dados quantitativo também conhecido como distribuição de frequências ou diagrama de frequências. Os dados de interesse são agrupados em classes uniformes e posicionados no eixo horizontal em forma retangular justaposto, enquanto no eixo vertical será atribuído o valor da frequência absoluta, ou frequência relativa ou ainda a frequência relativa percentual correspondente.

Figura 2 - Esquema gráfico de um histograma



Fonte: Elaborado pelo autor

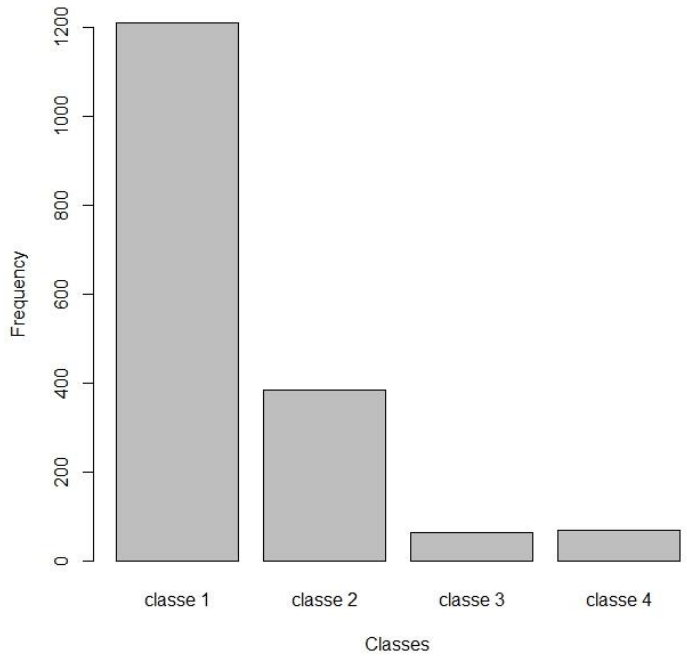
Um dos mais importantes usos de um histograma é fornecer informações a respeito do formato da distribuição (SWEENEY, 2015). Através dele é possível visualizar e comparar mais facilmente se os dados seguem determinados padrões históricos, como por exemplo, a distribuição normal, no qual o maior número de unidades estará localizada no centro da distribuição. Isso facilita a identificação de problemas e tendências do processo em análise. Quanto mais disperso os dados estiverem do ponto central, maior será a variabilidade (SCHLITTLER, 2014).

2.7 Gráfico de Barras

Também conhecido como gráfico de colunas, esse dispositivo é utilizado para representar dados categorizados sintetizados em uma distribuição de frequências absolutas, relativas ou relativas percentuais (SWEENEY, 2015).

Em um dos eixos do gráfico são especificados os rótulos dos dados categóricos em análise em forma retangular de mesma largura. No outro eixo são atribuídos os valores da frequência em uma escala bem definida, o qual serve de base para o comprimento do retângulo.

Figura 3 – Esquema de um gráfico de barras



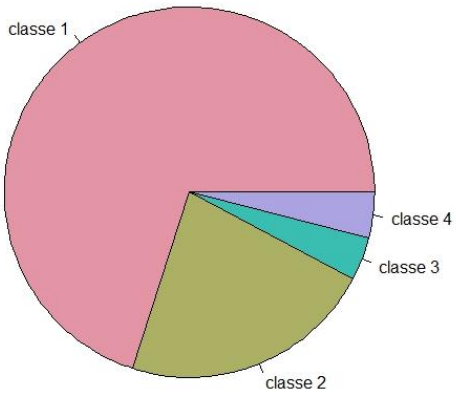
Fonte: Elaborado pelo autor

Embora essa ferramenta seja simples de ser construída e fácil de ser lida, a eficácia da análise é comprometida quando as frequências dos dados são próximas.

2.8 Gráfio de Pizza

Outro dispositivo utilizado para representar dados categorizados, apontando às distribuições das frequências relativas e relativas percentuais através da subdvisão de um círculo em vários setores, cujos ângulos são proporcionais a frequência de cada ocorrência.

Figura 4 – Esquema de um gráfico de pizza



Fonte: Elaborado pelo autor

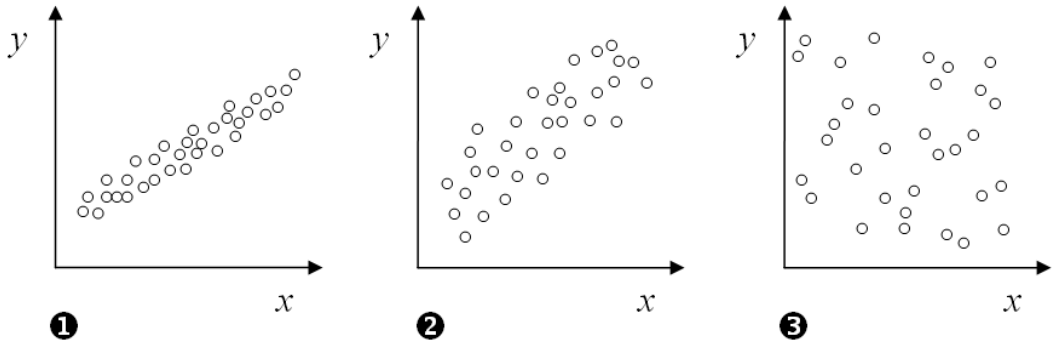
Assim como o gráfico de barras, a eficácia da análise é comprometida quando as frequências dos dados são próximas ou existem rótulos em demasia.

2.9 Diagrama de Dispersão e Correlação

O diagrama de dispersão é uma representação gráfica de relação entre duas variáveis quantitativas (SWEENEY, 2015), no qual uma é disposta no eixo horizontal, e outra no eixo vertical.

Junto ao gráfico é costume que seja traçada uma linha de tendência, que auxilia o analista a verificar facilmente se existe relação entre as variáveis, se são positivas ou negativas e ainda se são fortes ou fracas. No entanto, para obter uma resposta rigorosa para tal, é necessário o cálculo da correlação

Figura 5 – Esquema de diagramas de dispersão com distribuição de dados distinta



Fonte: Step Removed One, 2017.

A correlação varia entre os valores (-1; 1), sendo 1 uma correlação forte e positiva, 0 sem correlação, e -1, forte correlação negativa. A imagem a esquerda da figura (5) é um exemplo de correlação forte positiva, a do centro, fraca e positiva, e a direita sem correlação.

2.10 Regressão Linear

A partir do gráfico de dispersão é possível identificar se existe, e qual o tipo de relação entre duas variáveis (linear, quadrático, exponencial, etc.). A regressão linear é um modelo que vai quantificar a relação funcional entre uma variável dependente com uma ou mais variáveis dependentes através da obtenção de uma equação.

No entanto, como os dados são fenômenos observados, existirá uma lacuna entre o real e a equação matemática traçada. Assim, para conseguir o melhor ajuste e minimizar essa distância, o modelo utiliza o método dos mínimos quadrados. A demonstração matemática pode ser observada no livro “Análise de Regressão”, por Rodolfo Hoffmann, referência [5]. O resultado obtido é similar ao seguinte:

$$Yi = \beta 0 + \beta 1 x 1 + \varepsilon i \qquad (5)$$

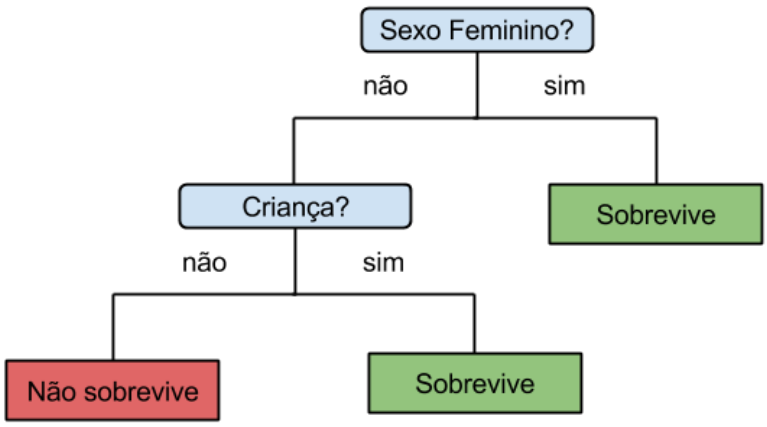
Neste modelo, Yi é a variável resposta no qual quer se descobrir o quanto sua variação é dependente da variação da variável x1. $\beta 0$ e $\beta 1$ são os coeficientes estimados da regressão que definem a reta e εi representa o erro experimental.

Esse modelo pode ser estendido e mais variáveis explicativas podem ser introduzidas, pois muitas vezes, a variação ocorre por mais de um fator. Para verificar sua adequabilidade é necessário realizar testes de hipótese, porém isso não será abordado no trabalho. Ademais, será apresentado o coeficiente de determinação R^2, no qual valores próximos a 1 indicam que o modelo proposto é adequado para descrever o fenômeno, enquanto valores próximos a 0 indicam que o modelo não é um bom parâmetro.

2.11 Árvores de Decisão

As árvores de decisão são algoritmos de classificação que têm amplo uso na prática não só por sua eficácia, mas, principalmente, pela facilidade de interpretação dos dados obtidos (SICSÚ, 2017). Esse método parte da divisão de um conjunto em subconjuntos a partir de características semelhantes, ou correlacionadas dado uma variável alvo a ser estudada. A estrutura final é semelhante a seguinte:

Figura 6 – Esquema de Árvore de Decisão



Fonte: Caelum, 2017.

Neste exemplo, para responder se uma pessoa sobrevive a uma determinada doença dado um banco de dados de um hospital, as variáveis selecionadas foram sexo e faixa etária. No caso, se for do sexo feminino, ou se for do sexo masculino e criança, ela sobrevive, caso contrário, não sobrevive.

Embora a construção do algoritmo não ser abordada, é importante ressaltar que existem limitações quanto ao número mínimo de dados e o tratamento dos nós terminais para que a análise seja de fato relevante (evitando por exemplo, overfittings). Isso pode ser feito através de métodos como o cross validation e de podas.

2.12 Regressão Logística

A regressão logística se assemelha ao modelo de regressão linear, no entanto, a variável resposta Yi é binária, assumindo os valores 0 e 1, que significam o "sucesso" ou "fracasso" de um evento. A regressão logística permite estimar as probabilidades que um evento tem de pertencer a cada um desses grupos a partir de suas características (SICSÚ, 2017).

Neste caso devemos assumir que a variável resposta segue uma distribuição de probabilidade binomial ($Yi \sim B(mi, \pi i)$), a qual é adequada ao modelo linear através da ligação logit que representa o logaritmo natural. A demonstração matemática pode ser observada no livro "Best Practices in Quantitative Methods", referência [8]. O resultado obtido é similar ao seguinte:

$$\ln\left(\frac{\pi i}{1-\pi i}\right) = \beta 0 + \beta 1 x1 + \cdots + \beta p x p \qquad (6)$$

Os parâmetros β são estimados através do método da máxima verossimilhança, que, de forma genérica, fornece valores que maximizam a probabilidade de se obter determinado conjunto de dados.

2.13 Validação dos Modelos

Existem vários indicadores na literatura que avaliam o desempenho de um modelo probabilístico, entre eles estão o MER, MWL e Gini. No entanto, o mais conhecido e

utilizado no mercado é o teste Kolmogorov-Smirnov (ou teste KS), dado à facilidade de cálculo e interpretação. Esse teste quantifica a máxima diferença entre as distribuições acumuladas das probabilidades de cada grupo (bons e maus) a partir de um conjunto de dados amostral retirados da base de dados principal em análise.

3. METODOLOGIA

3.1 RStudio

Para desenvolver o protótipo foi utilizado o RStudio, uma *graphical user interface* (GUI) do R, com ambiente de desenvolvimento integrado. Isso não afeta as funcionalidades do software original, mas torna a interface mais amigável e visualmente mais fácil de manipular as ferramentas disponíveis.

Além disso, o programa é gratuito e seu código é aberto, onde qualquer pessoa é livre para usar, copiar, estudar e fazer alterações no software, o que encoraja os usuários a voluntariamente melhorar o código e compartilhar com os demais. O programa pode ser encontrado para download no site oficial do RStudio.

3.2 Software R

3.2.1 Descrição Geral

R é um software gratuito voltado para estatística computacional e elaboração de gráficos, desenvolvido por Ihaka, Ross e Gentleman, Robert. A primeira versão beta do programa disponibilizada para o público foi em 2000, e desde então ela vem sendo aprimorada através de uma equipe de desenvolvimento especializada e dedicada, o “R Core Team”.

O software é substancialmente utilizado por profissionais estatísticos e data miners para a análise de dados. Isso porque o programa suporta tanto o desenvolvimento e compilação de scripts através de uma linguagem de programação específica (.R), considerada de alto nível, quanto comandos de linha dentro da própria interface, ou seja, é possível tratar dados externos de imediato uma vez que exportados.

O pacote de ferramentas e funções é amplo, contendo opções para maximizar a eficiência de um código, estilizar e personalizar gráficos e tratar dados. No que tange a estatística estão inclusos possibilidades como cálculo da média e variância, formação de modelos lineares e não lineares, modelos de regressão e análise de séries temporais. Portanto o software abrangendo tanto funções básicas, quanto avançadas.

Nesse cenário, o R foi escolhido como o software para o desenvolvimento da pesquisa por que:

1. É gratuito e de fácil acesso;

2. Funciona em diferentes plataformas, incluindo Windows, Unix e MacOS;
3. Possui linguagem de programação de alto nível, ou seja, é menos complexa de aprender (embora a curva de aprendizado seja inclinada);
4. Contém ferramentas estatísticas básicas e avançadas;
5. Têm ampla variedade de gráficos.

3.2.2 Funções Utilizadas

Para a montagem do protótipo foi essencial entender a lógica de programação e como é formada a estrutura de um código escrito em R. Para essa etapa foi necessário estudar os manuais disponíveis nos sites oficiais, assistir vídeo aulas, participar de tutoriais gratuitos e interagir com a comunidade de programadores do software.

É importante ressaltar que, ao escrever um código, existem diversas maneiras de se obter o mesmo resultado, no entanto, a grande diferença entre os métodos escolhidos se da pela eficiência com que o programa será implementado, consumindo tempo e memória da máquina, variando de acordo com o poder de processamento. Como o foco da pesquisa é o tratamento e análise de dados, o refinamento e eficiência do código não serão abordados.

O R possui uma vasta biblioteca de funções prontas para serem usadas, portanto, para calcular a média não é necessário realizar cada passo de operações aritméticas. Todas as operações já estão agrupadas em uma única função chamada **mean()**.

Para utilizar uma função, costuma ser necessário fornecer certos dados de entrada que serão manipulados de maneira automatizada para se obter a saída desejada. No caso de **mean()** é necessário fornecer o vetor que contém o conjunto de dados quantitativos que se quer obter a média. Além disso, existem parâmetros opcionais, que podem ser utilizados como guias para as funções, como por exemplo, definir **na.mr** como TRUE (verdadeiro), ou FALSE (falso), o que indica respectivamente se um conjunto de dados com algum valor sem definição (NA) deve ser levado em consideração (valor 0), ou excluído do cálculo.

Outras funções utilizadas que automatizaram a necessidade de realizar múltiplas operações aritméticas foram:

1. **sd()**, retorna o resultado do cálculo do desvio padrão;
2. **median()**, retorna o resultado do cálculo da mediana;
3. **max()**, retorna o valor máximo do conjunto de dados;

4. **min()**, retorna o valor mínimo do conjunto de dados;
5. **discretize(),** separa um vetor em grupos de acordo com o método escolhido (frequência, intervalo...) e o número de categorias a serem particionadas;
6. **createDataParticion(),** retorna a seleção aleatória de uma porcentagem dos dados da base de dados original para comporem uma amostra;
7. **rpart(),** retorna a partição de um conjunto em subconjuntos de acordo com a sua correlação;
8. **predict(),** retorna a previsão de um resultado dado o modelo utilizado e as variáveis em análise;
9. **glm(),** retorna os coeficientes da regressão de um conjunto de dados;
10. **step(),** retorna os coeficientes da regressão mais significantes estatisticamente de acordo com a regra de decisão selecionada (both, backward ou forward);
11. **HMeasure(),** retorna medidas de validação de modelo, entre eles estão o índice KS e o AUC (area under the curve) aplicado em curvas ROC.

No entanto, existem funções cujas saídas não são necessariamente valores numéricos ou vetores. Para se obter um gráfico “x vs. y”, é possível utilizar a função **plot()**, que deverá ter como parâmetros obrigatórios o conjunto de dados referentes a x, e outro conjunto de dados referentes a y. Nesse caso a função aceita tanto dados quantitativos quanto qualitativos; ela automaticamente irá fornecer um diagrama de dispersão em caso de cruzamento entre dados quantitativos, boxplots em caso de dados quantitativos e qualitativo, e gráfico de barras em caso de cruzamento entre dados qualitativos.

Os parâmetros opcionais em caso de funções gráficas abrangem a possibilidade de definir um título para o gráfico, título para os eixos x e y, relação x/y, entre outros. Outras funções utilizadas no protótipo para criação de gráficos específicos foram:

1. **head()**, retorna os primeiros n valores de um conjunto de dados, onde n é o número de linhas a ser definido pelo programador;
2. **hist()**, retorna um histograma;
3. **boxplot()**, retorna um boxplot;
4. **barplot()**, retorna um gráfico de barras;
5. **pie()**, retorna um gráfico de pizza;
6. **fancyRpartPlot(),** retorna a estrutura da árvore de decisão.

Além dessas funções, existem aquelas que auxiliam essencialmente na elaboração lógica do código, ou em necessidades pontuais, como é o caso de **CrossTable()**, que retorna uma tabela cruzada das variáveis inseridas além das proporções totais e marginais referentes, **br()** que cria um parágrafo dentro de um texto, e **h3()**, **h4()** e **h5()**, que modificam o tamanho da fonte de um texto.

Às vezes nem todas as funções que o programador pretende utilizar estão previamente disponíveis no programa, como é o caso da função **HMeasure()**. Para tanto é necessário recorrer e instalar outros pacotes (ou bibliotecas).

3.2.3 Pacotes Utilizados

Existem diversos pacotes com as mais variadas funções disponíveis para download, isso porque eles podem ser desenvolvidos tanto pela equipe que trabalha diretamente com o R, como os próprios usuários. Para a realização do protótipo foram utilizados mais de 15 pacotes, sendo que os mais importantes foram:

1. **shiny**, retêm as funções que permitem a estruturação de um aplicativo de forma dinâmica e interativa, além da oportunidade de transmitir as informações diretamente para uma webpage;
2. **graphics**, possibilita a utilização de gráficos específicos, como o gráfico de barras;
3. **gmodels,** contêm ferramentas para modelagem e visualização de funções, como por exemplo, CrossTable, bastante utilizado nas análises;
4. **arules,** providencia infraestrutura para representar, manipular e analisar dados e padrões;
5. **caret,** direcionado para algoritmos de modelos preditivos, técnicas de ajustes, diagnóstico e visualização de modelos, além de ferramentas como cross-validation e bases de treino e teste;
6. **rpart,** retêm funções recursivas de partição para classificação, regressão e modelagem de árvores de decisão;

Além desses, também foram aplicados **colorspace**, **ggplot2**, **arules**, **car**, **stats**, **rpart.plot**, **ratlle**, **RColorBrewer**, **HMeasure** e **dummies**.

3.3 Shiny

Shiny é um pacote de funções voltado para criação de aplicativos interativos de web utilizando apenas o R. Portanto não é necessário ter conhecimento prévio de linguagens de programação específicas para isso, como HTML, CSS ou JavaScript. Isso porque, o compilador, que transforma o script em um programa funcional, realiza as mudanças necessárias para essas linguagens se necessário.

O grande diferencial do Shiny é que ele possibilita a execução do R em paralelo com o aplicativo, portanto, ao interagir com algum comando externo no app, ele automaticamente calcula, modifica e atualiza os dados para o usuário, incluindo alterações gráficas.

Para utilizar esta ferramenta ou qualquer outra pacote, em primeiro lugar é necessário instalar o pacote, para isso basta digitar o seguinte comando no R:

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

Depois de instalado, para utilizar as funções é preciso declarar isso nas primeiras linhas do código, caso contrário o software não irá reconhecer os comandos e acusará erro.

```
> library(shiny)
```

Após esse passo todas as funções básicas e as contidas nos pacotes declarados estão prontas para serem utilizadas. Para programar o aplicativo, o shiny necessita de uma divisão no código, isto é, uma parte é dedicada ao user-interface (*ui*), e a outra é dedicada às instruções que a máquina precisa para a construção do aplicativo (*server*).

Ui é responsável pelo controle e a aparência do layout. É nessa parte que são programados os botões interativos que capturam os inputs externos a serem enviados para tratamento no *server*. Depois de tratados, o modo como os outputs serão expostos também fazem parte de *ui*. Já o *server*, contém as instruções do que precisa ser feito com os inputs, como processar as informações e transferir isso para *ui*. Por exemplo, *ui* é responsável por enviar o vetor numérico escolhido pelo usuário para o *server*, que por sua vez contém as funções necessárias para calcular a média e retornar o resultado para *ui*, para assim disponibilizar para o usuário.

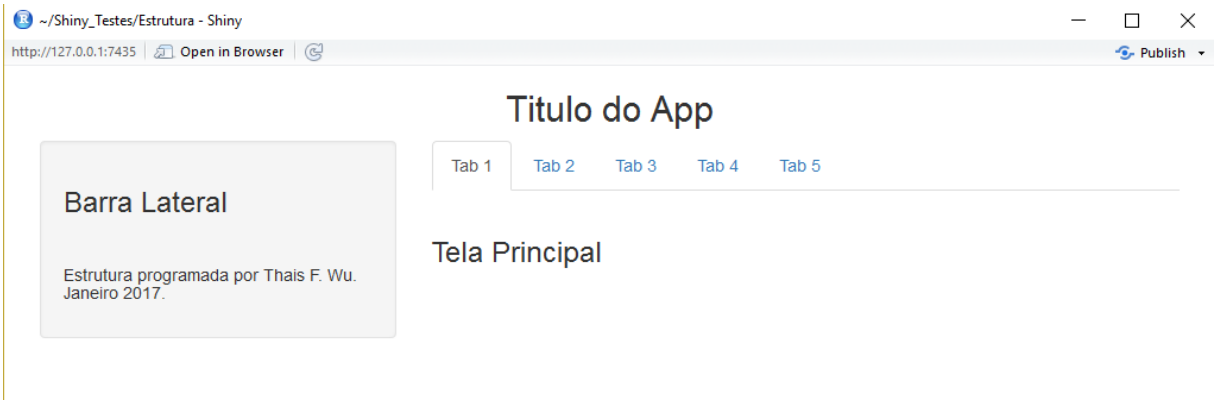
A partir da versão 0.10.2, o shiny passou a ter a opção de suportar no mesmo arquivo as informações contidas tanto em *ui* quanto no *server*. Antes disso, era necessário criar um script próprio para cada (ui.R e server.R) e salvá-los no mesmo diretório. Por motivos de organização o protótipo foi programado da maneira antiga, não causando qualquer diferença com relação a sua funcionalidade.

O layout do aplicativo pode ser composto de várias maneiras. Para a estruturação do protótipo foram utilizadas sete funções essenciais em *ui*:

1. **shinyUI()**, é a função que identifica e retorna todo o conteúdo presente em *ui*. Códigos escritos a partir da versão 0.10 não requerem mais dessa função, mas é necessário indicar na última expressão referente a ui.R que ele pertence a *ui*;
2. **pageWithSidebar()** define uma user interface que contém uma área para o título do aplicativo, uma barra lateral para controle de inputs e uma área central para visualização dos outputs;
3. **headerPanel()** retorna um título na parte superior do aplicativo;
4. **sidebarPanel()** define uma barra lateral a esquerda do aplicativo para controle de inputs, como por exemplo, painéis de seleção ou inserção de dados;
5. **conditionalPanel()** cria painéis condicionados, ou seja, que são visiveis ou não dependendo do valor da expressão que ativa a condição definida. Isso abre a possibilidade de abrir determinadas informações desejadas pelo usuário no painel principal;
6. **mainPanel()** define um painel principal ao lado direito da barra lateral que contém os outputs;
7. **tabsetPanel()** cria um conjunto de painés que podem ser definidos unitariamente por **tabPanel()**. É possível determinar o título de cada painel e ter controle de qual será ativado através do argumento **id**. Para isso uma alternativa é definir um valor para cada painel e passar o valor de argumento para a função **conditionalPanel()**.

Utilizando estas funções sem a adição de dados externos, o esqueleto do aplicativo gerado se assemelha a figura à baixo.

Figura 7 – Estrutura de um aplicativo em Shiny utilizando apenas *ui*



Fonte: Elaborado pelo autor

A partir dessa estrutura os conteúdos necessários para estudo e análise foram alocados para cada parte buscando manter coerência e organização.

4. RESULTADOS E DISCUSSÕES

4.1 Protótipo

4.1.1 Base de Dados

O protótipo foi montado tendo como base uma planilha de dados que simula as informações de uma companhia telefônica que oferece serviços de assinatura de linha telefônica (TEBA). O arquivo utilizado possui 2000 linhas e 10 variáveis, sendo que 6 delas são dados quantitativos, e 4 qualitativos.

1. **idade**, contêm a informação da idade em anos completos do cliente;
2. **linhas**, número de linhas que o cliente assina, podendo variar de 1 a 5;
3. **temp_cli**, tempo que o cliente é assinante do serviço, medido em meses;
4. **renda**, equivalente a renda familiar em G$, uma unidade fictícia equivalente ao real (R$);
5. **fatura**, despesa mensal que o cliente tem com a assinatura;
6. **temp_rsd**, tempo na residencia atual em anos;
7. **local**, região onde reside, divisão entre as áreas A, B, C e D;
8. **tvcabo**, se o cliente possui assinatura de televisão a cabo;
9. **debaut**, se o pagamento é realizado em débito automático;
10. **cancel**, se o assinante cancelou o contrato.

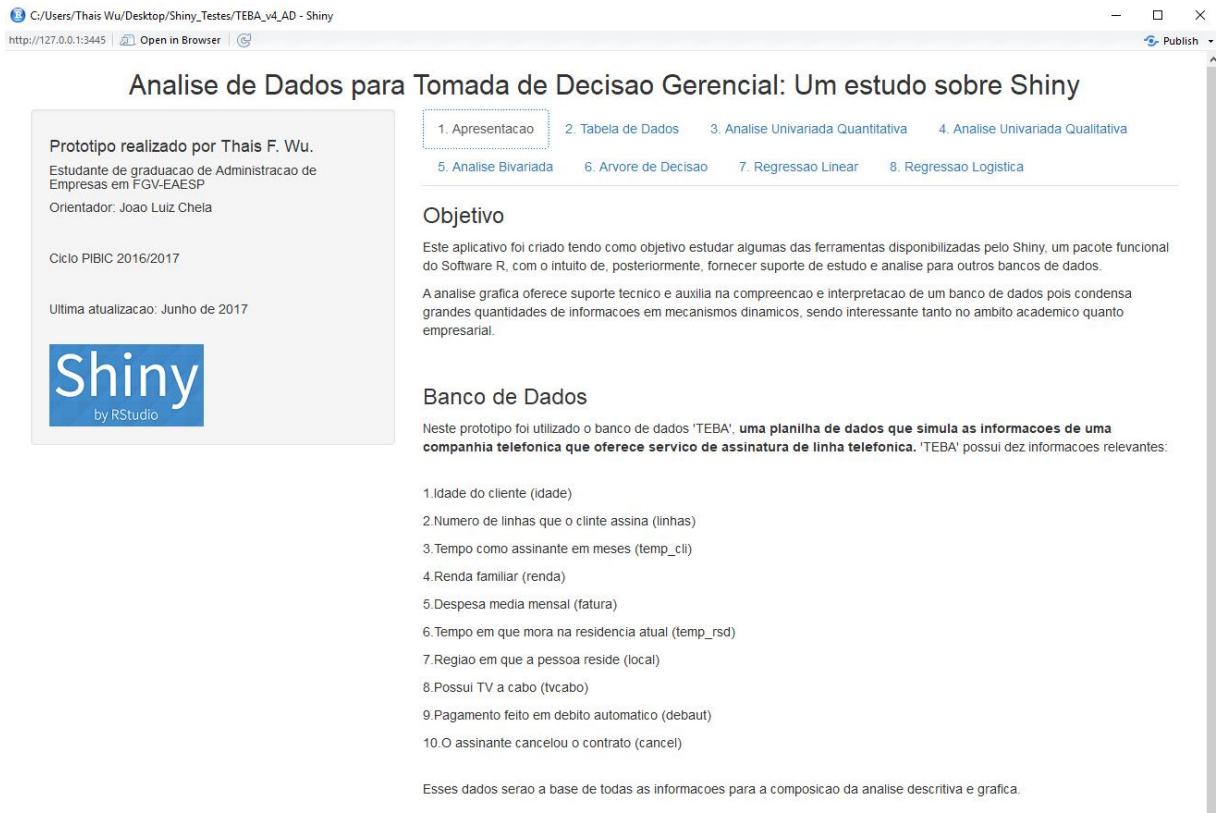
O objetivo é proporcionar informações relevantes utilizando ferramentas estatísticas que sejam suficientes para identificar e traçar o perfil dos assinantes que cancelam o serviço. Ter esse conhecimento abre espaço para a empresa entender o seu cliente para assim perceber oportunidades de investimento e suas fraquezas, de modo a reanalisar suas estratégias.

Nesse cenário o protótipo foi subdividido em 8 telas principais: (1) Apresentação, (2) Tabela de Dados, (3) Análise Univariada Quantitativa, (4) Análise Univariada Qualitativa, (5) Análise Bivariada, (6) Árvore de Decisão, (7) Regressão Linear e (8) Regressão Logística.

4.1.2 Apresentação

Ao iniciar o aplicativo, o painel principal é programado para abrir a primeira aba “Apresentação”. Essa tela contém uma breve explicação dos objetivos da pesquisa, além das informações que compreendem as variáveis de TEBA.

Figura 8 – Interface do protótipo. Primeira Aba “Apresentação”



Fonte: Elaborado pelo autor

As funções utilizadas para redigir o texto foram **h5()**, **br()** e **strong()**. Note que o shiny não aceita acentuação gráfica em suas cadeias de caracteres (strings). No entanto, ele aceita imagens externas. Para isso, basta que o arquivo esteja no mesmo diretório que os scripts e utilizar a função **tags$img**, que aloca espaço para uma imagem em uma página HTML**.**

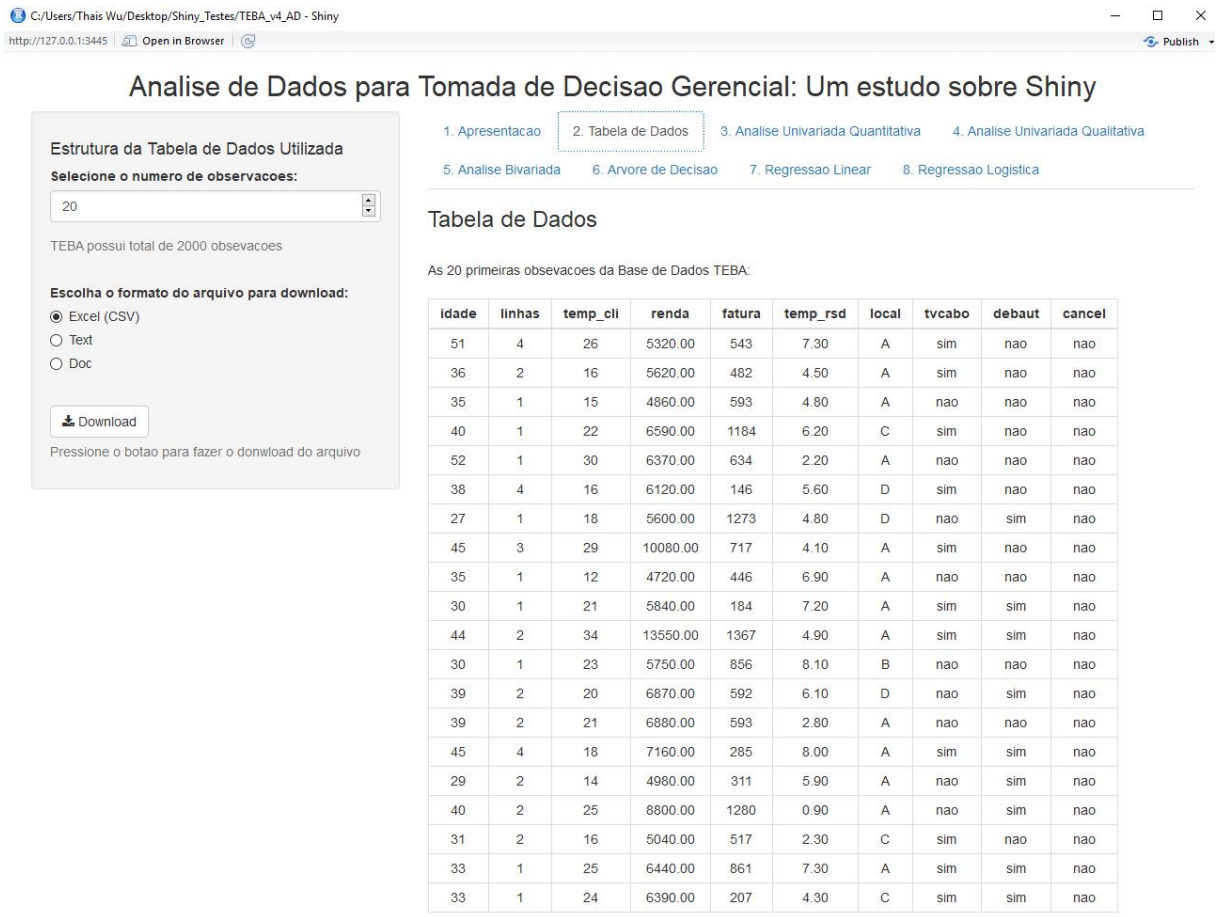
Repare que como foi utilizado apenas funções que representam saída de visualização para o usuário, não foi necessário programar em *server* para essa tela. As funções utilizadas em cada aba tanto em *ui* quanto em *server* estão disponíveis no anexo para maiores detalhes.

4.1.3 Tabela de Dados

A segunda aba, “Tabela de Dados”, contém a tabela original TEBA. O objetivo dela é complementar as informações das variáveis descritas em “Apresentação” através da visualização da estrutura dos dados. Como TEBA possui 2000 linhas, o protótipo foi

programado para mostrar apenas as 20 primeiras linhas, porém, o usuário tem a opção de escolher aumentar ou diminuir o número de observações através da barra lateral.

Figura 9 – Interface do protótipo. Segunda Aba “Tabela de Dados”



idade	linhas	temp_cli	renda	fatura	temp_rsd	local	tvcabo	debaut	cancel
51	4	26	5320.00	543	7.30	A	sim	nao	nao
36	2	16	5620.00	482	4.50	A	sim	nao	nao
35	1	15	4860.00	593	4.80	A	nao	nao	nao
40	1	22	6590.00	1184	6.20	C	sim	nao	nao
52	1	30	6370.00	634	2.20	A	nao	nao	nao
38	4	16	6120.00	146	5.60	D	sim	nao	nao
27	1	18	5600.00	1273	4.80	D	nao	sim	nao
45	3	29	10080.00	717	4.10	A	sim	nao	nao
35	1	12	4720.00	446	6.90	A	nao	nao	nao
30	1	21	5840.00	184	7.20	A	sim	sim	nao
44	2	34	13550.00	1367	4.90	A	sim	sim	nao
30	1	23	5750.00	856	8.10	B	nao	nao	nao
39	2	20	6870.00	592	6.10	D	nao	sim	nao
39	2	21	6880.00	593	2.80	A	nao	nao	nao
45	4	18	7160.00	285	8.00	A	sim	sim	nao
29	2	14	4980.00	311	5.90	A	nao	sim	nao
40	2	25	8800.00	1280	0.90	A	nao	sim	nao
31	2	16	5040.00	517	2.30	C	sim	nao	nao
33	1	25	6440.00	861	7.30	A	sim	sim	nao
33	1	24	6390.00	207	4.30	C	sim	sim	nao

Fonte: Elaborado pelo autor

Na barra lateral, em *ui*, é utilizada a seguinte função:

```
> numericInput("obs", "Selecione o numero de observacoes:", 20)
```

A função cria uma variável chamada “*obs*” que armazena o input do usuário. Note que é possível programar o número inicial de observações, no caso, 20. Essa variável é então enviada para o *server*, que utiliza seu valor em 2 funções, a primeira que atualiza o texto do painel central, e a segunda que atualiza a tabela.

```
#Atualiza o texto
> output$text <- renderText({
> paste("As", input$obs, "primeiras obsevacoes da Base de Dados TEBA:")})

#Atualiza a tabela
> output$view <- renderTable({
> head(teba, n = input$obs)})
```

Após atualizar os dados de input, o *server* manda de volta para *ui* os outputs *text* e *view* para serem disponibilizados para o usuário.

```
#Visualização de text
> textOutput("text")

#Visualização de view
> tableOutput("view")
```

Repare que as funções são diferentes, isso porque uma saída é em formato de texto, enquanto a outra, em forma de tabela.

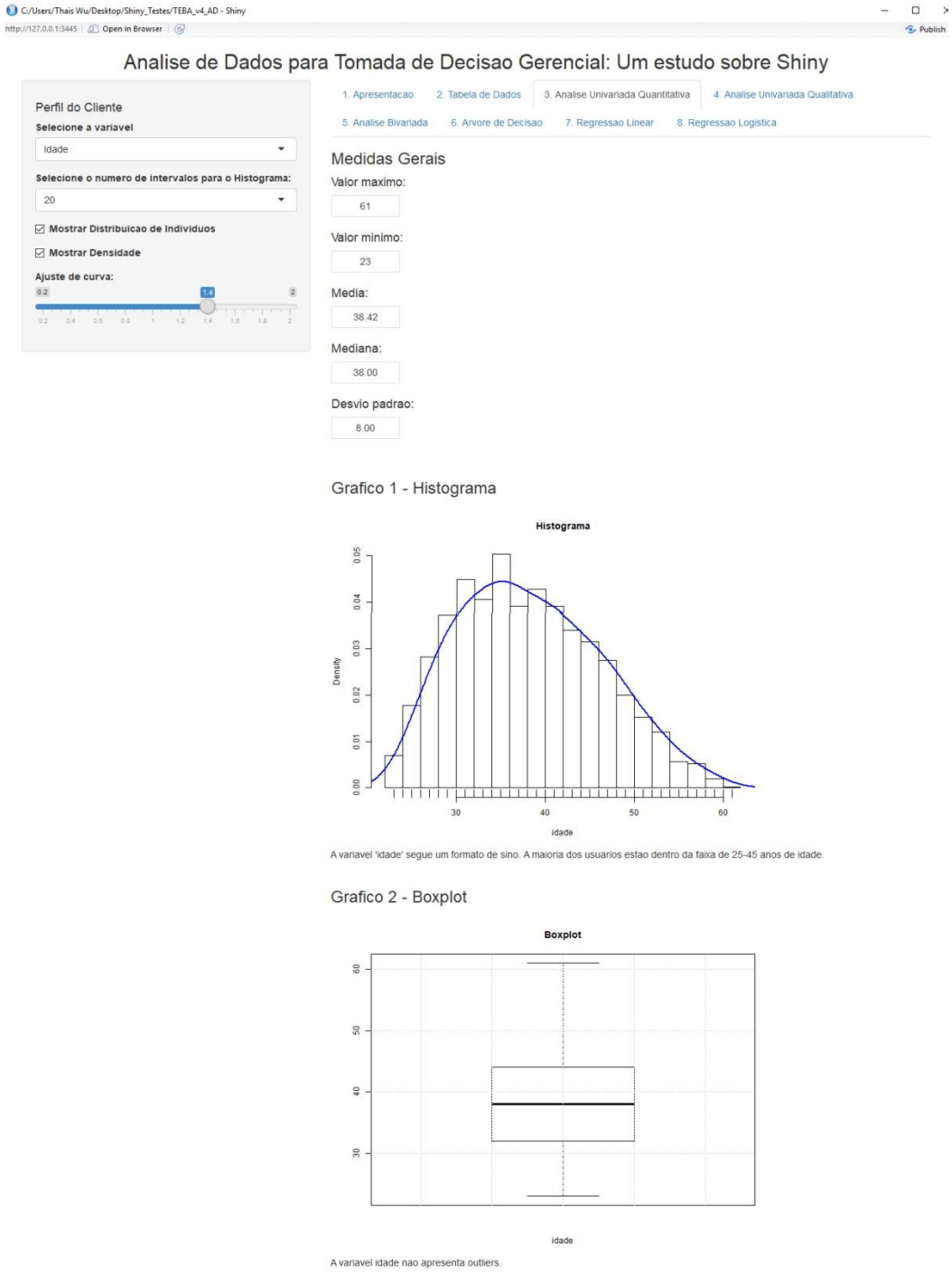
Além disso, ainda nessa aba, o usuário pode optar por fazer o download do arquivo em análise tanto com extensão “csv.” quando “txt.” ou “doc.”. O processo de comunicação entre *ui* e *server* serão similares a quase todas as próximas etapas: (1) armazenamento do input em *ui*, (2) recepção e tratamento em *server* e (3) output do resultado em *ui*.

Nesse caso, *server* deverá executar três passos: identificar a escolha enviada, redigir o nome do arquivo a ser salvo e formatar o arquivo conforme o padrão desejado.

4.1.4 Análise Univariada Quantitativa

A terceira aba utiliza os dados quantitativos apresentados na tabela de dados e realiza uma série de operações para auxiliar na análise do comportamento das variáveis em particular.

Figura 10 – Interface do protótipo. Terceira Aba “Análise Univariada Quantitativa”



Fonte: Elaborado pelo autor

Assim como “Tabela de Dados”, a aba lateral possibilita ao usuário escolher uma das variáveis quantitativas que será utilizada como input para todas as funções no *server*

direcionadas para essa aba. Além disso, ele também pode escolher mostrar a distribuição de indivíduos e ajustar a densidade da curva para o histograma.

```
#Cria um botão com as 6 variáveis quantitativas
> selectInput("var", "Selecione a variavel", choices = c("Idade" = 1, "Numero de linhas" = 2, "Tempo como assinantes" = 3, "Renda" = 4, "Fatura" = 5, "Tempo de Residencia" = 6)),
```

Note que a variável “var” que será transferida para *server* pode conter valores entre 1 a 6 dependendo da escolha do usuário. Isso é importante pois é o que identifica a coluna de dados que servirá como input. Em *server* teremos:

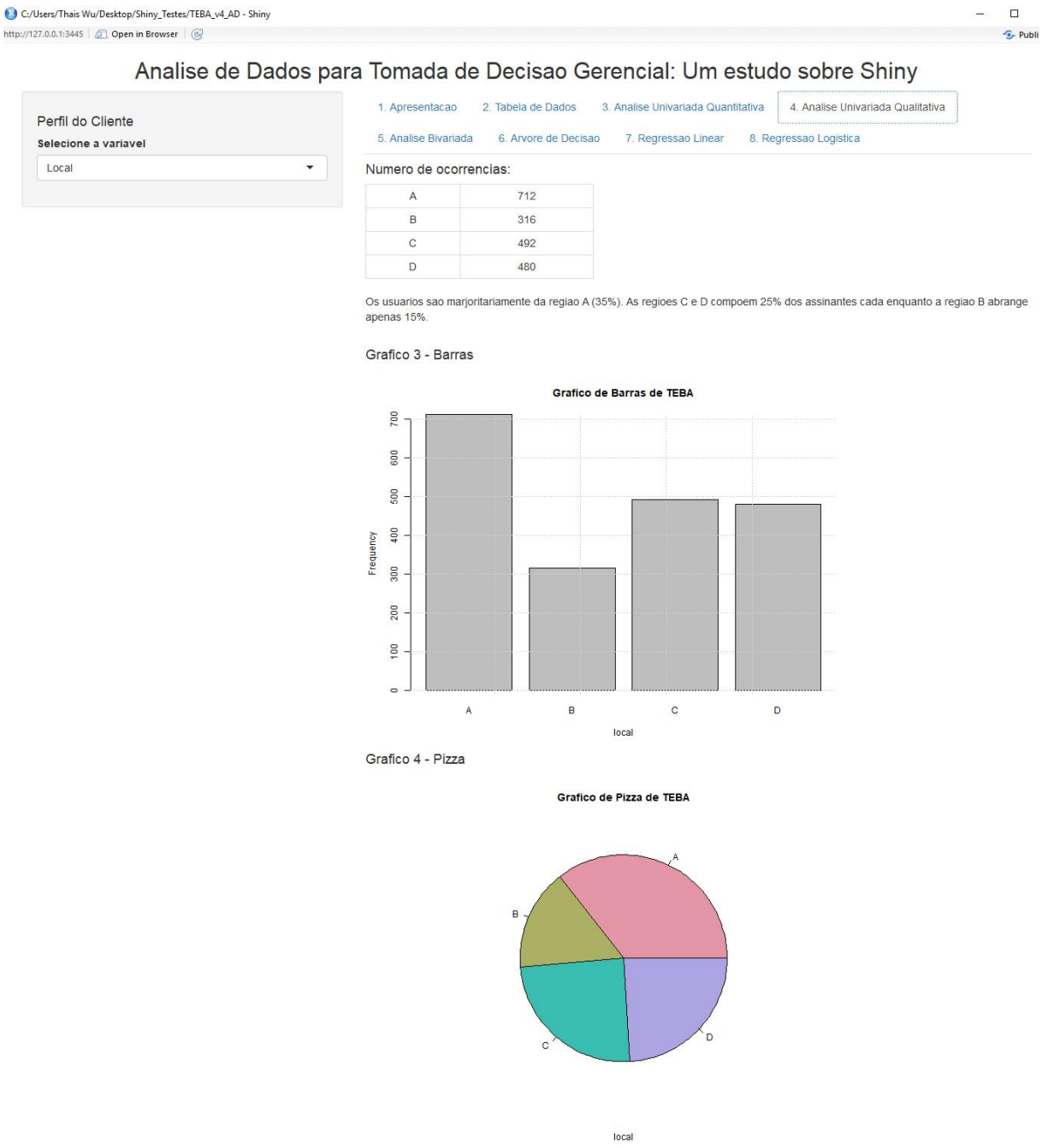
```
> output$media <- renderText({
> meanm <- as.numeric(input$var)
> mean(teba[,meanm], na.rm = TRUE)})
```

O valor numérico de “var” é transferido para “meanm”, que é utilizado na função para obter a média. Um processo similar acontece ao utilizar as funções que devolvem o cálculo do valor máximo, mínimo ou mesmo os gráficos de boxplot ou histograma. A diferença é o modo como eles são repassados para *ui*, isso porque as funções relativas a impressão de texto e impressão gráfica são distintas.

4.1.5 Análise Univariada Qualitativa

A quarta aba é dedicada aos dados qualitativos. O processo lógico de programação é parecido com o realizado na terceira aba. O usuário seleciona a variável desejada e o painel principal retorna o número bruto de ocorrências de cada dado, um gráfico de pizza e um gráfico de barras.

Figura 11 – Interface do protótipo. Quarta Aba “Análise Univariada Qualitativa”



Fonte: Elaborado pelo autor

Além disso, uma explicação sucinta é feita sobre os dados obtidos, por exemplo, se “TV a Cabo é selecionado”, um texto logo abaixo da tabela do número de ocorrências descreve:

> “68% dos usuarios possuem TV a Cabo, portanto poderia ser estudado o quao interessante e agregar esse servico junto a linha telefonica.”

Essas frases são programadas através de funções “if” e “else” que identificam a condição que deve existir para elas serem geradas.

4.1.6 Análise Bivariada

Após realizar a análise descritiva das variáveis individualmente, na quinta aba é possível visualizar graficamente como os dados se comportam com relação às demais variáveis. Para tanto foi utilizada a função **plot()** que retorna um diagrama de dispersão, um gráfico de barras ou boxplots dependendo do cruzamento entre dados quantitativos e qualitativos.

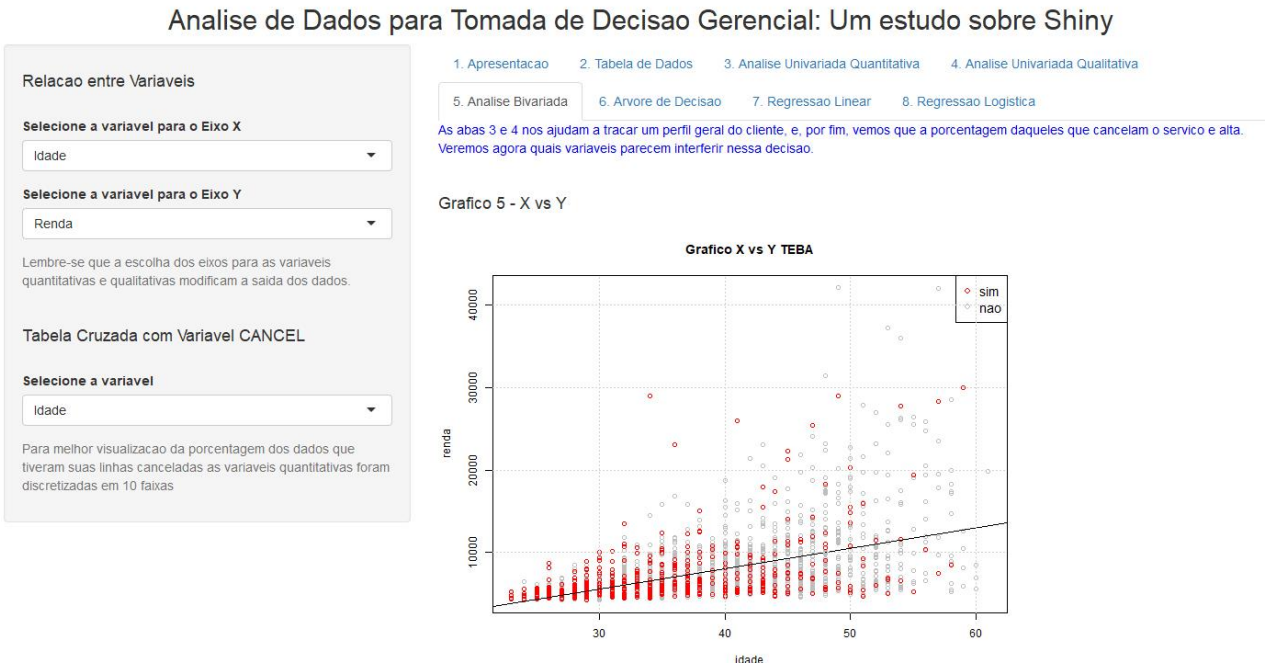
As funções utilizadas para o código funcionar nessa aba devem ser reativas. Note que nas demais aplicações o input selecionado sempre gerava apenas um output, porém, nesse caso, como **plot()** utiliza duas variáveis e o gráfico se altera com relação a mudança de qualquer uma delas é necessário que o input e o output sejam sensíveis a essas ações. Nesse caso utilizamos as função **reactive()** em *server* após armazenarmos os dados referentes ao eixo x na variável “varx” e os dados referentes ao eixo y, em “vary”.

```
#Eixo X
> x <- reactive({teba[,as.numeric(input$varx)]})

# #Eixo y
> y <- reactive({teba[,as.numeric(input$vary)]})
```

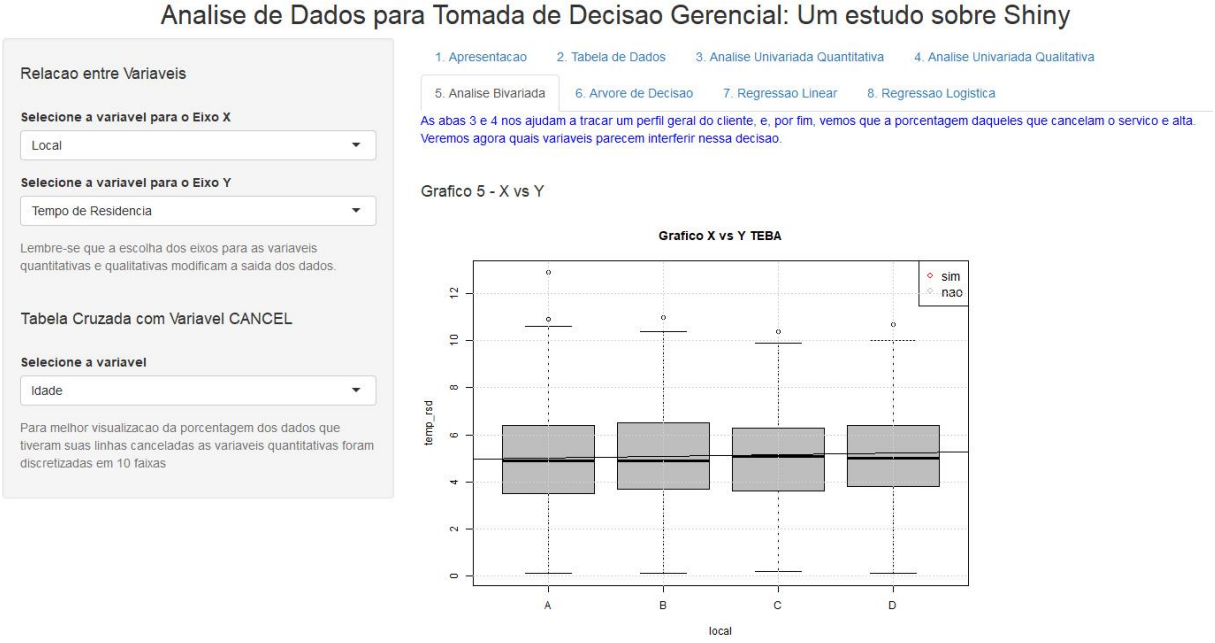
Após esse processo, é possível aplicar esses dados na função **plot()**. Para o gráfico ser mais compreensível, foi inserida a linha de tendência e uma grade. Além disso, existe uma distinção entre os dados que indicam indivíduos que cancelaram e não cancelaram a assinatura, destacados em vermelho e cinza, respectivamente.

Figura 12 – Interface do protótipo. Quinta Aba “Análise Bivariada”.



Fonte: Elaborado pelo autor

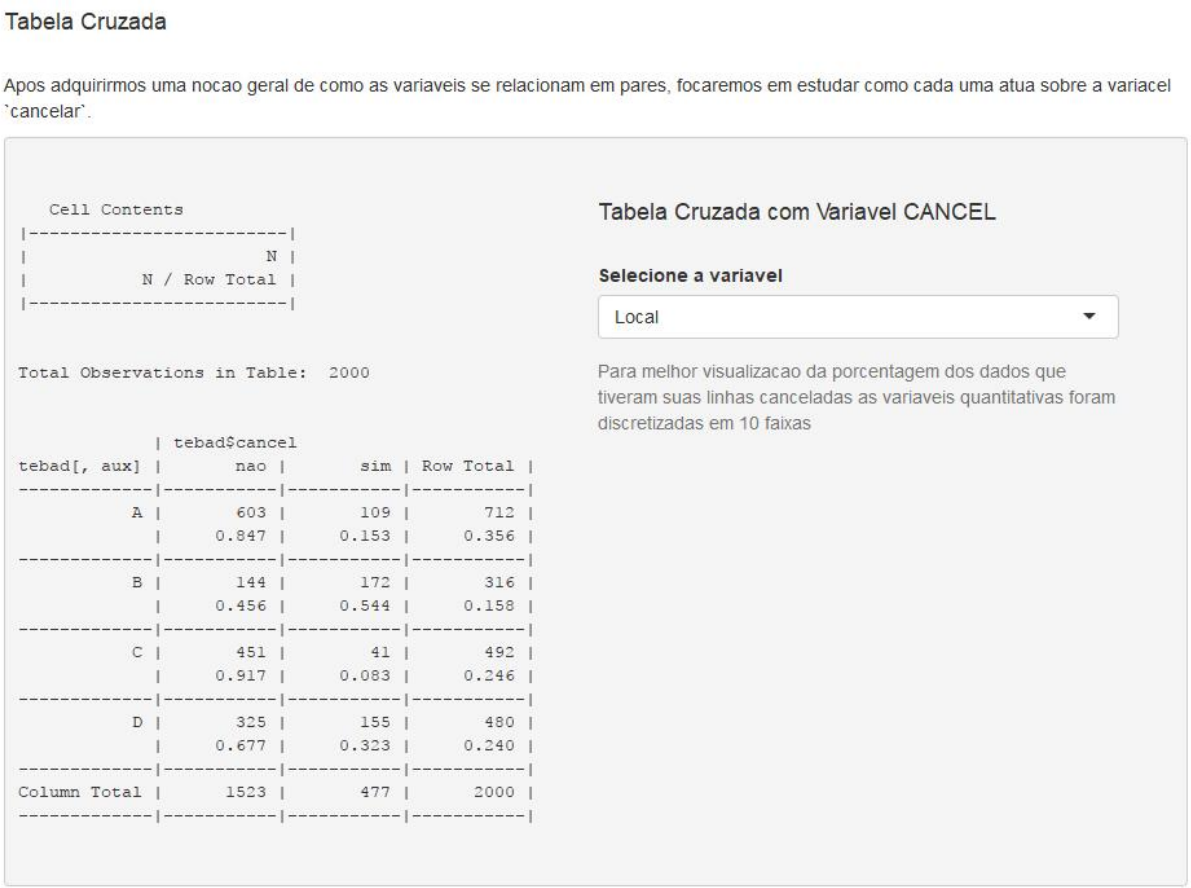
Figura 13 – Exemplo de uma variável quantitativa e qualitativa



Fonte: Elaborado pelo autor

Ainda nessa aba, o usuário tem a opção de selecionar qualquer variável para ser analisada em paralelo com o alvo “cancelar”. Isso foi realizado a partir da discretização dos dados em até dez faixas, a fim de verificar se existe um perfil específico.

Figura 14 – Exemplo de uma tabela cruzada entre “cancel” e “local”.



Fonte: Elaborado pelo autor

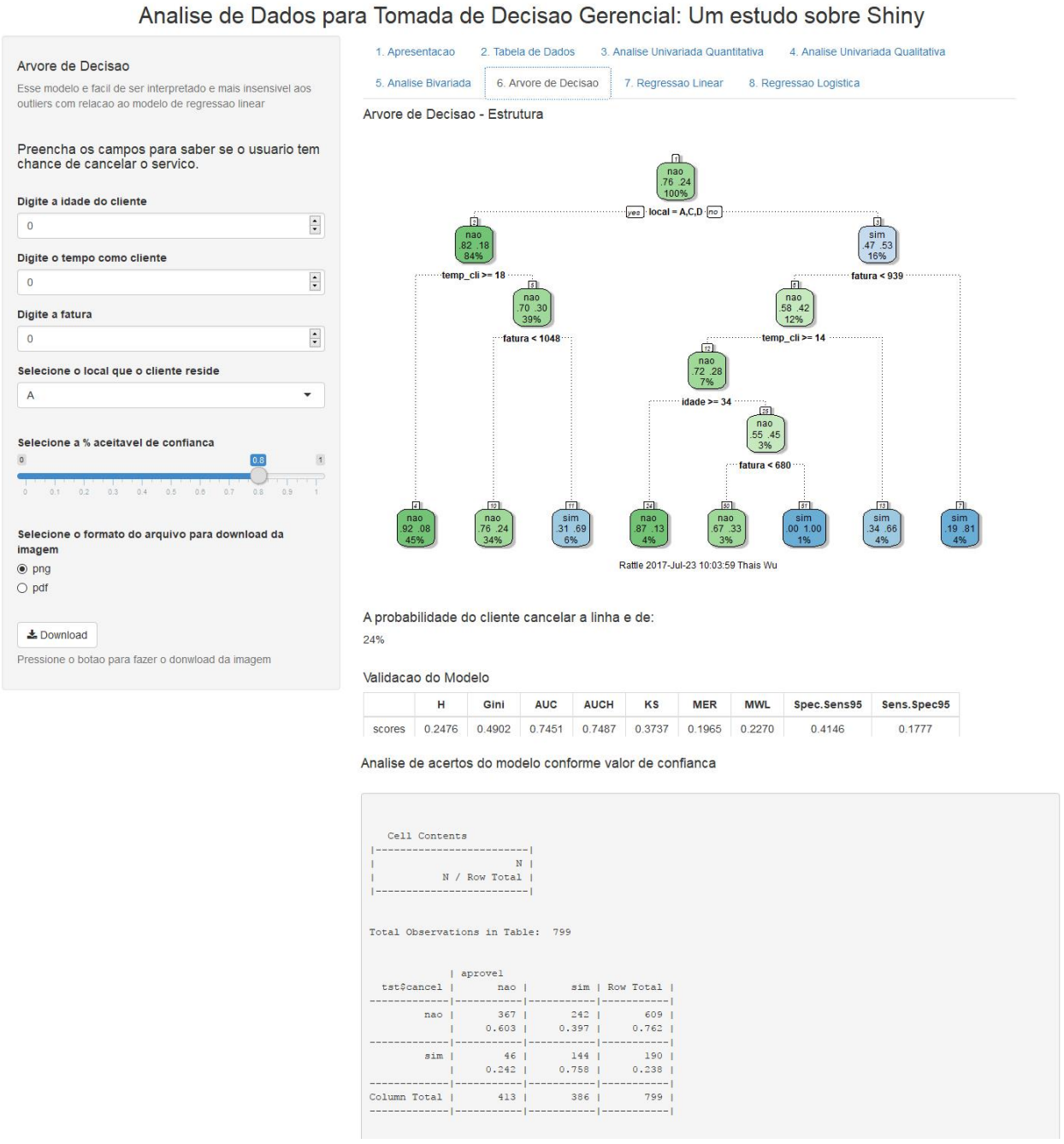
Nessa tabela, temos que 35,6% dos indivíduos moram no local A, sendo que 84,7% deles não cancelam a assinatura e 15,3% cancelam a assinatura. Já em B, 15,8% residem nesse local, sendo que 45,6% não cancelam e 54,4% cancelam.

4.1.7 Árvore de Decisão

A partir da sexta aba, o usuário tem a possibilidade de inserir dados externos e receber uma previsão da probabilidade do cliente cancelar a linha. Nessa foi utilizado o modelo de árvore

de decisão. O modelo não foi trabalho exclusivamente nos arquivos *server* e *ui*, para tanto foi criado um terceiro arquivo no R que desempenha o papel de uma estrutura auxiliar no qual foram realizadas as partições dos dados para estudo e validação, bem como a análise de acertos e custo.

Figura 15 – Interface do protótipo. Sexta Aba "Árvore de Decisão".



	H	Gini	AUC	AUCH	KS	MER	MWL	Spec.Sens95	Sens.Spec95
scores	0.2476	0.4902	0.7451	0.7487	0.3737	0.1965	0.2270	0.4146	0.1777

Fonte: Elaborado pelo autor

Tanto para o modelo de árvore, quanto o de regressão logística e linear, a base de dados foi subdividida em dois grupos, sendo que 60% dos dados (1200) foram alocados para formar as equações, e 40% para serem utilizadas como teste de acertos e possibilitar a validação final. Esse processo foi realizado a partir das linhas de código a seguir

```
> flag = createDataPartition(teba$cancel, p=.6,list=FALSE)
> lrn = teba[flag,]
> tst = teba[-flag,]
```

Especificamente para a árvore, o R possui a função **rpart()**, que, de forma recursiva, encontra os subgrupos finais automaticamente dado uma variável alvo, nesse caso, "cancel". Para minimizar o erro final identificado pelo valor retornado de cp, também foi utilizado o método de poda através da função **prune()**. A partir desse processo, as variáveis relevantes encontradas foram: idade do cliente, tempo como cliente, fatura e local de residência. Na aba lateral o usuário pode inserir as informações de um possível novo cliente e a probabilidade dessa pessoa cancelar a assinatura será disponibilizada na tela principal. O usuário também pode fazer o download da imagem da árvore tanto em png quanto em pdf.

Ademais, o usuário tem acesso a vários indicadores de validação de modelo obtidos através da função **HMeasure()** e uma tabela de acertos que varia de acordo com o nível de aceitação do usuário. Ambos utilizam os 40% dos dados separados para teste.

Figura 16 – Validação do modelo "Árvore de Decisão".

Validacao do Modelo

	H	Gini	AUC	AUCH	KS	MER	MWL	Spec.Sens95	Sens.Spec95
scores	0.2476	0.4902	0.7451	0.7487	0.3737	0.1965	0.2270	0.4146	0.1777

Fonte: Elaborado pelo autor

Repare que o valor KS para o modelo é relativamente baixo.

Figura 17 – Tabela de acertos “Árvore de Decisão”.

Fonte: Elaborado pelo autor

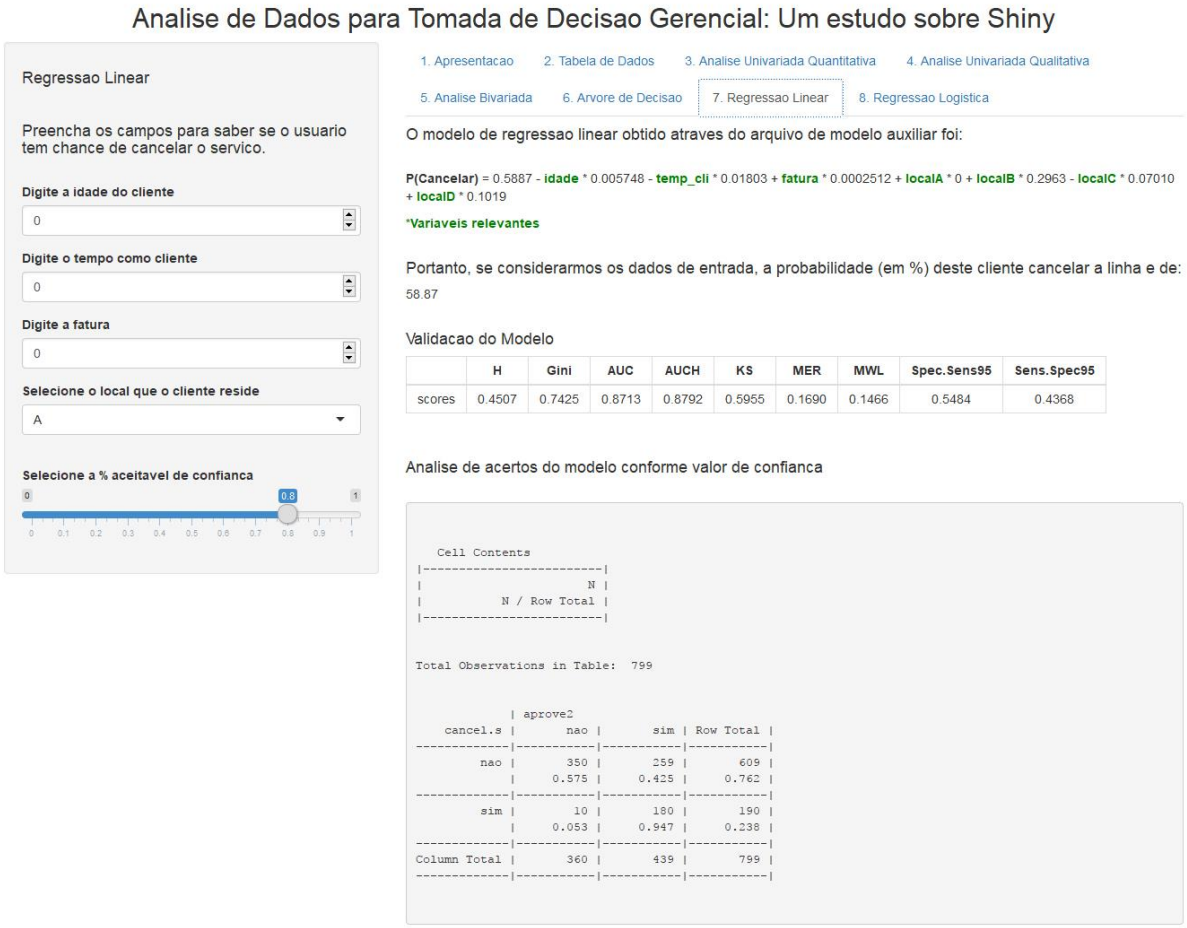
Conforme o nível de confiança aumenta, o modelo passa a considerar clientes que não cancelariam, mas que por ventura estão com suspeitas altas dadas suas características, como potenciais maus clientes. Por conta disso “não-não” diminui e “não-sim” aumenta gradativamente. Pela mesma logica, a chance do modelo acertar aqueles que de fato cancelam “sim-sim”, aumenta, e a chance de considerar um cliente com perfil que cancela erroneamente “sim-não” diminui.

Na imagem, dado uma confiança de 80%, o modelo acerta (“não-não” + “sim-sim”), 64% dos casos.

4.1.8 Regressão Linear

Nesta aba, assim como em “Árvore de Decisão”, o usuário insere os inputs a serem utilizados na equação de regressão para receber o resultado do quanto a variável alvo depende das características encontradas.

Figura 18 – Interface do protótipo. Sétima Aba “Regressão Linear”.



Fonte: Elaborado pelo autor

Novamente, esses passos foram realizados no modelo auxiliar. Foram utilizadas as funções **dummie()** para transformar as variáveis qualitativas em quantitativas, **lm()**, para obter os coeficientes da equação e, **step()**, que elimina as variáveis cujo valor de previsão não sejam significantes. A equação final encontrada foi:

$$P(cancel) = 0.5887 - idade * 0.0057 - temp.cli * 0.018 + fatura * 0.0002 + local.A * 0 + local.B * 0.2963 - local.C * 0.0701 + local.D * 0.1019$$

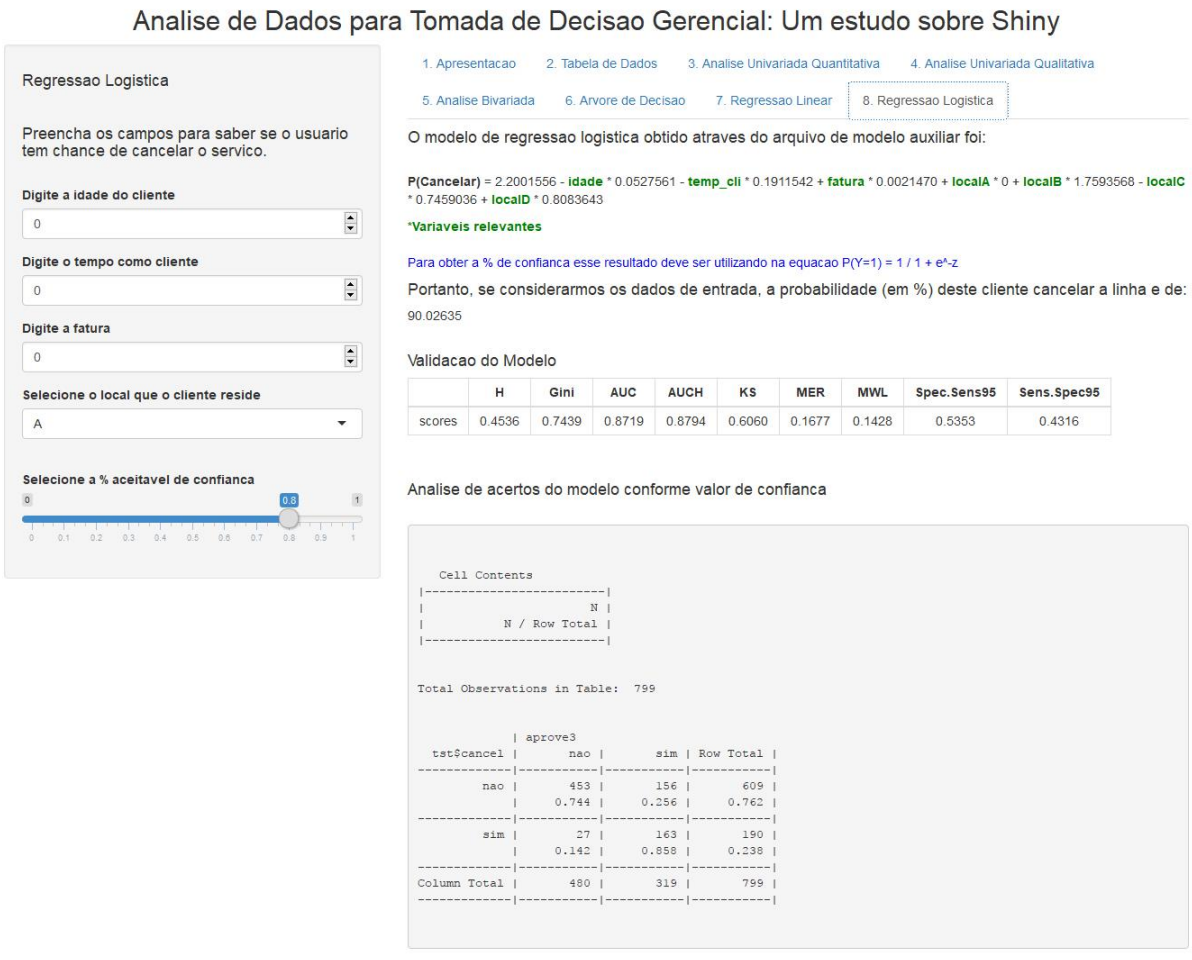
Note que as variáveis finais encontradas foram às mesmas do modelo anterior, sendo que a idade, tempo como cliente e morar no lugar C reduzem a influência sobre “cancel”, enquanto as demais aumentam. No entanto, ao verificar os índices de validação do modelo e a tabela de acertos, os números indicam maior valor no KS (0,59) e maior porcentagem de acertos (65%),

assim, o usuário consegue comparar a acurácia entre os modelos e optar pelo que melhor se molda a situação.

4.1.9 Regressão Logística

A oitava e ultima aba do protótipo possui uma interface similar a anterior, com os mesmos recursos e opções. No entanto, o modelo utilizado e o de regressão logística, assim, ao inserir os dados, o usuário pode verificar a probabilidade de um novo assinante cancelar ou não a linha.

Figura 19 – Interface do protótipo. Oitava Aba “Regressão Logística”.



Fonte: Elaborado pelo autor

Novamente são disponibilizadas as variáveis encontradas pelas funções **glm()** e **step()**, iguais às encontradas nos dois outro modelos, no entanto, os coeficientes são diferentes. A função final encontrada é:

$$P(cancel) = 2.2 - idade * 0.0527 - tem.cli * 0.1911 + fatura * 0.002 + local.A * 0 + local.B * 1.7593 - local.C * 0.7459 + local.D * 0.8083$$

Ademais, o índice de validação KS indica um valor próximo ao de regressão linear (0,60), porém com relação à tabela de acertos, o modelo acerta 77% dos casos demonstrando ser o mais apropriado para essa base de dados em particular.

4.2 Estudo de Caso

A fim de exemplificar como seria o processo de avaliação de novos assinantes iremos supor que três pessoas fictícias, Tiago, Pedro e Alice tenham pedido para assinar um novo plano, ou mudar um contrato já existente com a companhia telefônica Tesla. Para isso ser feito é necessário que ocorra a atualização dos dados dessas pessoas no sistema da empresa seguindo a estrutura da tabela abaixo.

Tabela 1 – Dados dos assinantes

Nome	idade	linhas	temp_cli	renda	fatura	temp_rsd	local	tvcabo	debaut
Tiago	24	1	0	1700	275	1.70	B	nao	sim
Pedro	37	3	18	7150	1700	7.00	A	sim	nao
Alice	39	2	20	9500	700	11.00	B	sim	sim

Fonte: Elaborado pelo autor

Antes de determinar se eles são bons clientes ou não para a empresa, é preciso entender primeiramente as características que levam as pessoas a cancelarem a linha e qual é o perfil geral dos assinantes, assim, o gestor tem opções estratégicas de como atacar o problema de fato para reter e atrair mais consumidores e não apenas aceitar ou negar a assinatura.

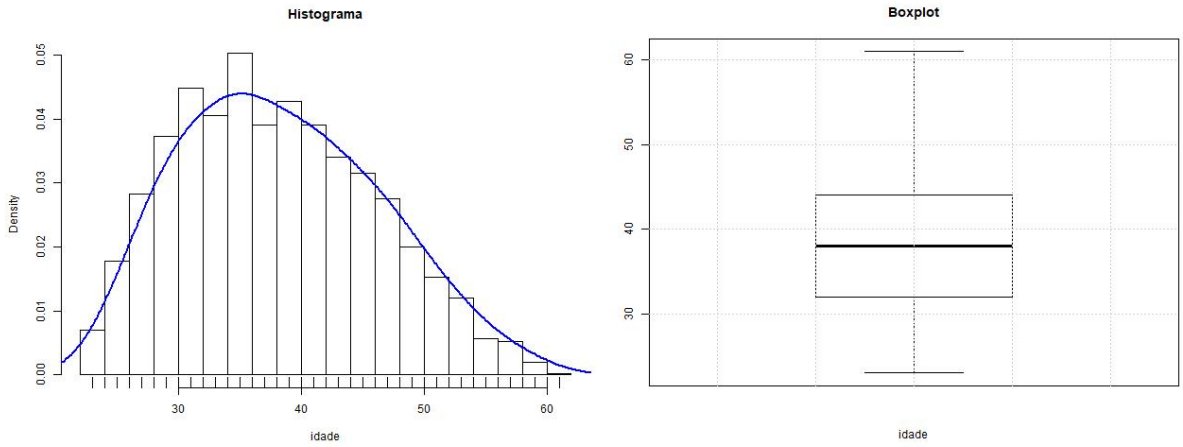
4.2.1 Perfil Geral dos Assinantes

A partir das informações disponibilizadas no protótipo, é possível identificar um perfil que se sobressai aos demais assinantes do serviço. Na aba “Análise Univariada Quantitativa”, o

histograma oferece uma visão da distribuição da variável em análise, isto é, se ela segue uma função normal ou dispersa. Além disso, o boxplot possibilita a verificação de como os dados se comportam perante a média e a variância, e quantos deles são outliers, o que prejudicaria a confiabilidade de uma análise de menor profundidade.

Quando a variável selecionada é a "idade", o histograma indica uma frequência próxima da normal (curva de sino). Pela regra empírica do teorema de Chebyshev isso significa que aproximadamente 63% dos dados estão contidos dentro de um desvio padrão da média. De fato, a média (38,42) é muito próxima da mediana (38,00) e o desvio padrão é de apenas 8,00, além disso o boxplot não indica outliers. Portanto é possível afirmar que a faixa etária dos clientes assinantes se concentra em torno dos 38 anos de idade.

Figura 20 – Histograma e Boxplot da variável "idade"

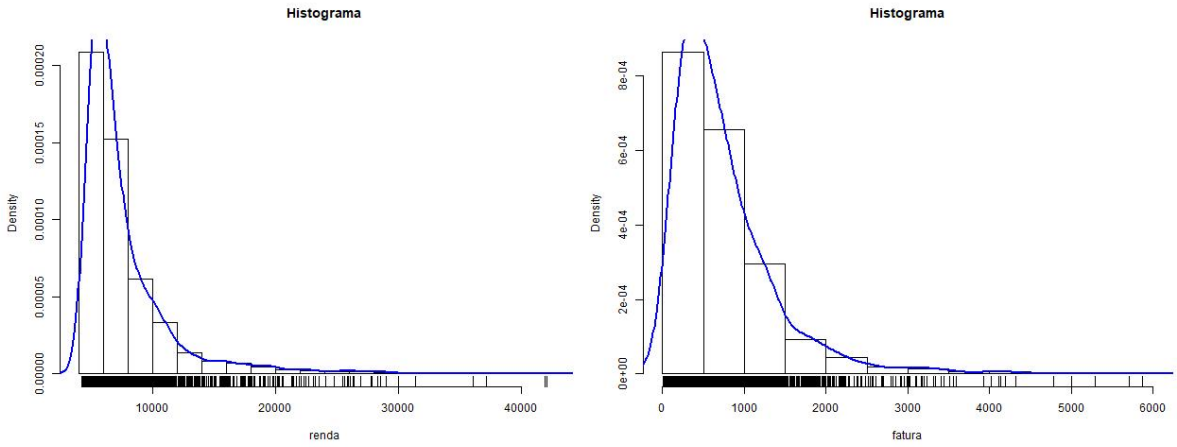


Fonte: Elaborado pelo autor

Ao escolher "Renda" o histograma deixa claro que existe um pico na frequência próxima ao valor 6.000,00. No entanto existem vários valores dispersos que geram uma cauda a direita do gráfico, portanto existe viés à direita, de fato, nesse caso a média (7664,51) é superior a mediana (6400,00) e o desvio padrão (3932,51) equivale a 51,3% do valor da mediana. Ao contrário da variável "idade", o boxplot nessa situação apresenta vários valores em situação de outliers. Esse acontecimento se repete com relação a variável "Fatura" cuja frequência se concentra ao redor de 500,00 apesar da média ser 752,60, e a mediana 582,00. Isso expressa à importância da análise gráfica e a necessidade de considerar várias ferramentas estatísticas em paralelo para o estudo.

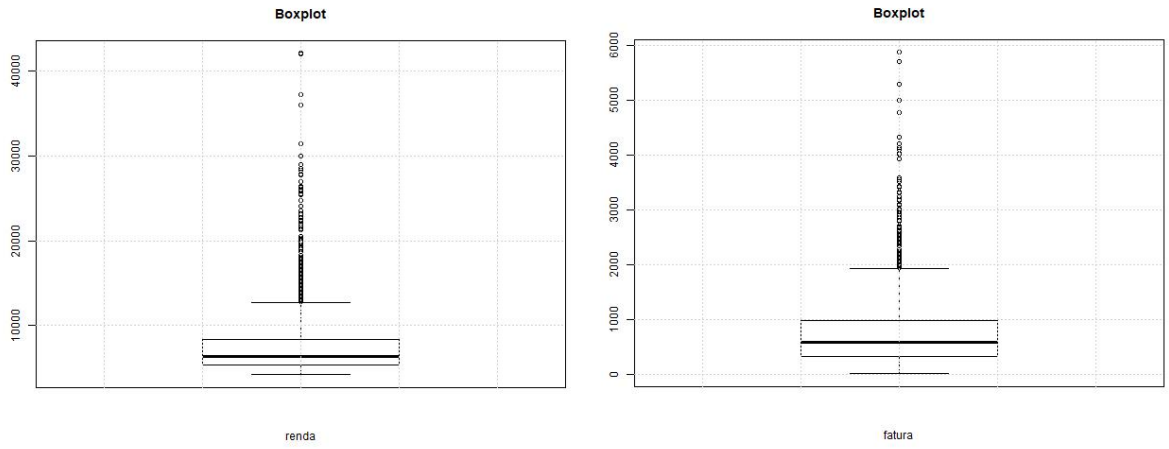
Porém, note que o bem oferecido é um bem básico para qualquer classe social, portanto, indefere da renda familiar, nesse cenário, os outliers fazem sentido uma vez que existe desigualdade social. A mesma lógica não pode ser usada para “fatura” no qual existem vários usuários que possuem gastos que excedem os limites estipulados estatisticamente. Essa ocorrência deveria ser estudada a parte, isso porque, pode ser que a empresa ofereça serviços extras que atraem alguns dos clientes. Nesse caso, essa ocorrência se torna uma oportunidade a ser explorada pelo gestor.

Figura 21 – Histograma de “renda” e “fatura” respectivamente



Fonte: Elaborado pelo autor

Figura 22 – Boxplot de “Renda” e “Fatura” respectivamente

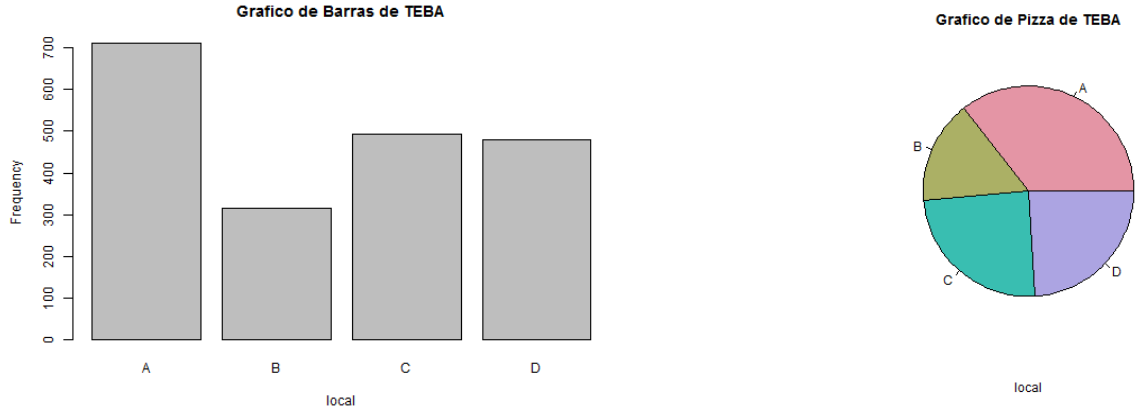


Fonte: Elaborado pelo autor

O mesmo processo lógico pode ser realizado com as demais variáveis quantitativas. Portanto é possível verificar que o cliente costuma assinar de uma a duas linhas, no qual grande parte deles costuma manter a assinatura por, em média, 20 meses (1 ano e 8 meses) e ter tempo médio de residência de 5 anos, com desvio padrão de 2,01 anos.

Quanto aos dados qualitativos, tanto o gráfico de barras quanto o gráfico de pizza presentes na aba “Análise Univariada Qualitativa” são ferramentas visuais que possibilitam rápida compreensão das diferenças de ocorrências. Ao selecionar a variável “Local” fica perceptível que a região A é onde existem mais assinantes, e a região B, o lugar onde tem menos assinantes.

Figura 23 – Gráfica de Barras e gráfico de Pizza da variável “local”



Fonte: Elaborado pelo autor

Dados como esses, embora aparentemente simples, podem mover uma organização a entender o porquê de essa diferença existir para assim criarem novos planos e conseguirem mais clientes em potencial. Por exemplo, 68% dos usuários possuem TV a Cabo, portanto poderia ser estudado o quão interessante e agregar esse serviço junto à linha telefônica. Além disso, aproximadamente 40% dos clientes pagam o serviço em debito automático. Logo, a maior parte recorre a outros métodos, como por exemplo, pagamento de boletos. Nesse contexto deve-se levar em consideração o erro humano, no qual as pessoas são sucessíveis ao esquecimento sendo interessante verificar os meios de comunicação com o cliente.

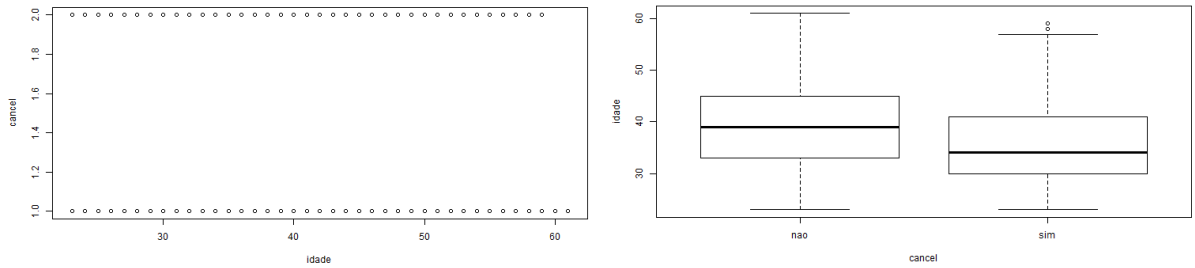
Por fim, 24% dos assinantes optam por cancelar o contrato com a empresa, o que é um número relativamente alarmante.

4.2.2 Perfil do Assinante que Cancela a Linha

Através da aba “Análise Bivariada” é possível realizar uma breve análise de como as variáveis estão correlacionadas, auxiliando a traçar o perfil dos clientes que cancelam a assinatura. A compreensão dos fatores que impelem esse acontecimento é de vital importância para a organização, pois possibilita tomar medidas estratégicas que garantem a vantagem competitiva da empresa.

Na barra lateral selecionamos a variável “cancel” no eixo X e as demais variáveis do eixo Y. Note que se “cancel” for selecionado no eixo Y os diagramas de dispersão apresentarão uma distribuição de dados diferente. Isso acontece porque “cancel” é uma variável qualitativa e o programa assume o símbolo “sim” como tendo valor 1, e “não”, como valor 2.

Figura 24 – Exemplo de inversão entre as variáveis e os eixos



A imagem a esquerda representa um diagrama de dispersão quando é realizado “idade vs cancel”. A direita são boxplots quando realizado “cancel vs idade”.

Fonte: Elaborado pelo autor

Os boxplot acima representam como a idade se comporta com relação àqueles que não cancelam e cancelam a assinatura, respectivamente. Nota-se que existe certo desnível entre as medianas e poucos outliers, portanto é possível presumir que pessoas mais jovens, entre 35 anos, são mais propensas a cancelar do que pessoas mais velhas. Isso também pode ser verificado através da tabela abaixo, que divide a variável idade em dez segmentos de mesmo tamanho, nele, a maior concentração de pessoas que cancelam estão presentes no intervalo de

29 a 39 anos, marcados em preto, enquanto aqueles que não cancelam, estão concentrados nos intervalos de 36 a 47 anos, em vermelho.

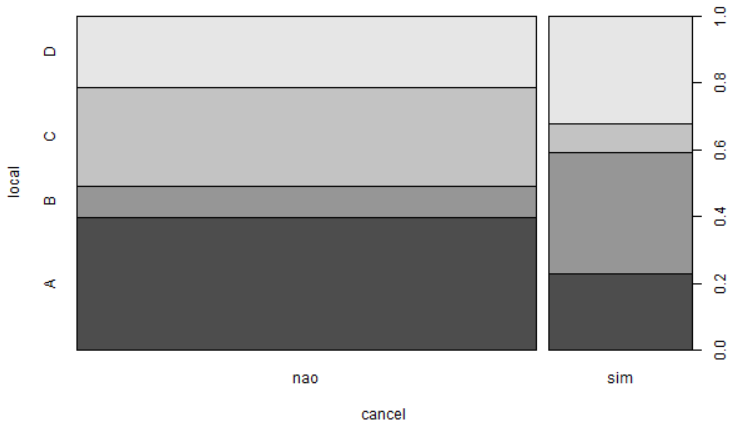
Tabela 2 – Concentração de dados entre variável “idade” e “cancel”

	tebad$cancel		
tebad[, aux]	nao	sim	Row Total
[23,29)	121	91	212
	0.571	0.429	0.106
[29,32)	152	78	230
	0.661	0.339	0.115
[32,34)	128	48	176
	0.727	0.273	0.088
[34,36)	134	49	183
	0.732	0.268	0.091
[36,39)	198	60	258
	0.767	0.233	0.129
[39,41)	144	27	171
	0.842	0.158	0.086
[41,44)	175	46	221
	0.792	0.208	0.110
[44,47)	167	30	197
	0.848	0.152	0.098
[47,50)	135	20	155
	0.871	0.129	0.077
[50,61]	169	28	197
	0.858	0.142	0.098
Column Total	1523	477	2000

Fonte: Elaborado pelo autor

Esse procedimento é repetido para as demais, e é verificado que tanto a variável “fatura” quanto “temp_cli” parecem ser relevantes apesar de apresentarem outliers (boxplots a direita da imagem abaixo). Ademais, verifica-se que o maior número de cancelamentos deriva das regiões B e D, mesmo elas sendo as que menos possuem clientes. Como dito anteriormente, um estudo que explique as causas desse fenômeno possibilitaria potencial crescimento de mercado.

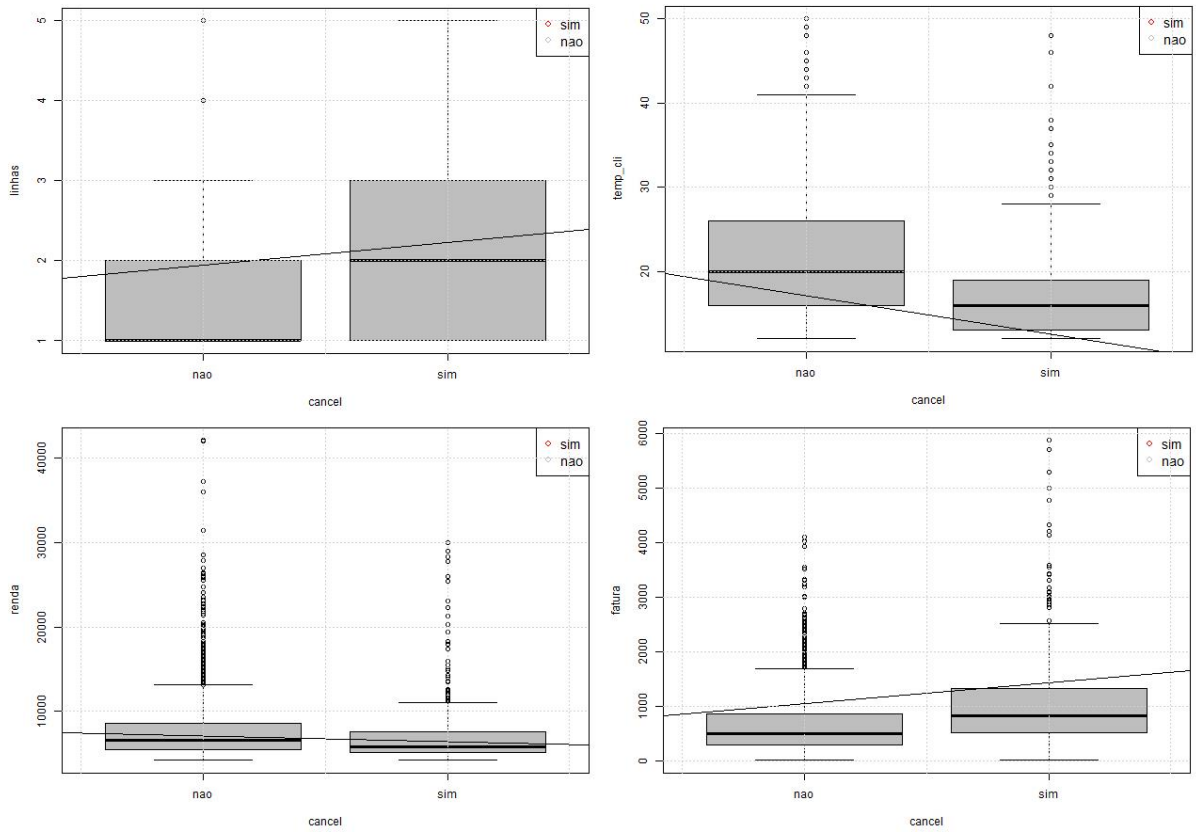
Figura 25 – Cancelamento de linhas de acordo com as regiões



Fonte: Elaborado pelo autor

No entanto, as demais variáveis apresentam boxplots com médias e limites inferiores e superiores próximos, o que impossibilita a afirmação de que elas são relevantes para a discriminação do cancelamento da assinatura. Para isso ser feito fazem-se necessárias técnicas estatísticas mais avançadas.

Figura 26 – Estudo da variável “cancel”



A imagem superior à esquerda refere a variável “numero de linhas”, a superior a direita, “tempo como cliente”, a inferior a esquerda, “renda” e inferior a direita, “fatura”.

Fonte: Elaborado pelo autor

4.2.3 Avaliação dos novos assinantes

Mesmo através de uma análise visual, as variáveis que se destacaram foram às mesmas obtidas nos modelos de árvore de decisão, regressão linear e regressão logística. Depois de estudado o perfil geral do assinante e as características que impelem alguns a cancelarem a linha e prejudicar a empresa, por fim, iremos verificar se é interessante que a empresa tenha despesas com um novo assinante que tem grande probabilidade de interromper o serviço.

Recorde que Tiago, Pedro e Alice recorreram à empresa Tesla, detentora do banco de dados TEBA (tabela 1). Iremos inserir seus dados nas abas 6, 7 e 8 a fim de investigar qual a propensão de cada um deles de cancelar a linha. Os dados a seguir sintetizam os resultados:

Tabela 3 – Resultado dos modelos

	Tiago	Pedro	Alice	KS	Acertos (80%)
Regressão Linear	81.61%	40.31%	47.60%	0.59	65%
Regressão Logística	82.11%	45.34%	39.69%	0.60	77%
Árvore de Decisão	66%	8%	13%	0.37	64%

Fonte: Elaborado pelo autor

O modelo que apresenta ter maior valor KS e maior porcentagem de acertos, e, portanto, o que parece mais adequado é o de regressão logística, seguido da regressão linear e árvore. A aceitação de um novo cliente depende do quanto à empresa está disposta a assumir o risco dado que existe um custo tanto de negar um cliente, quanto aceitar e perder um cliente. O Tiago possui características de um jovem que ainda não possui emprego ou residência fixa, e por isso, apresenta o maior risco entre os avaliados (82.11%). Já a Alice, é uma adulta que apresenta renda alta e moradia estável em um bom lugar, portanto, ela representa o menor risco (39.69%). Mesmo que existam discrepâncias entre os modelos, todos apresentam números coerentes quando combinados os resultados e validação, portanto, se torna responsabilidade do usuário saber qual aquele que melhor se encaixa para a situação.

5. CONCLUSÃO

O shiny possui uma curva de aprendizado elevado. Utilizar as suas estruturas básicas não requer conhecimentos especializados de programação, no entanto, se o programador pretende ter maior controle sobre o código, cujas funções fogem das pré-disponibilizadas pelo software e pela comunidade, ele deverá precisar de conhecimentos computacionais específicos.

Apesar disso existe uma variedade enorme de ferramentas matemáticas e estatísticas que compreendem tanto funções básicas quanto avançadas, enquanto a estrutura é facilmente moldável e adaptável para outros cenários. Além disso, a interação gráfica auxilia a interpretação e a análise por parte do usuário.

O protótipo atinge o objetivo de possibilitar a compreensão e previsão do comportamento das variáveis, e acredita-se ser possível implementar essa ferramenta na realidade das pequenas empresas principalmente devido a limitação de custos e menor complexidade e variedade dos conjuntos de dados. No entanto faz-se necessário o estudo e programação de ferramentas para realizar uma análise rigorosa quando os dados estão muito dispersos, pois isso impossibilita o analista de fazer boas inferências, afetando as decisões gerenciais.

6. REFERÊNCIAS

[1] AGRAWAL, A. *R Shiny app tutorial #1 – How to make shiny apps – An introduction to Shiny*. Acessado em Janeiro de 2017. Disponível em: https://www.youtube.com/watch?v=_0ORRJqctHE&list=PL6wLL_RojB5xNOhe2OTSd-DPkMLVY9DfB

[2] Caelum. *R, Titanic and Data Science*. Acessado em Julho de 2017. Disponível em: http://blog.caelum.com.br/r-titanic-e-data-science/

[3] CHANG. W. Package 'Shiny'. Acessado em Janeiro de 2017. Disponível em: https://cran.r-project.org/web/packages/shiny/shiny.pdf

[4] Github. *UI & Server*. Acessado em Janeiro de 2017. Disponível em: http://rstudio.github.io/shiny/tutorial/#ui-and-server

[5] HOFFMAN, R. *Análise de Regressão* - Uma introdução a econometria. Piracicaba. 5a ed. 2016, p 44.

[6] KILLMAN & MITROFF. *Problem defining and the consulting/intervention process*. Calif Manage Rev. 1979 Spring, p.26-33.

[7] SICSÚ, A. *Modelagem Preditiva Árvores de Decisão*. São Paulo, 2017. FGV-EAESP.

[8] OSBORNE, J. *Best Practices in Quantitative Methods*. Califórnia: Sage Publications, Inc. 2008, p. 358.

[9] PETENATE, M. *A importância da análise de dados para um negócio*. Acessado em Janeiro de 2017. Disponível em: http://www.escolaedti.com.br/a-importancia-da-analise-de-dados-para-um-negocio/

[10] PORTER, Michael, MONTOGOMERY, Cynthia. *Estratégia a Busca da Vantagem.* Rio de Janeiro. 13 ed. 1998.

[11] R. *The R Base Package*. Acessado em Janeiro de 2017. Disponível em: http://stat.ethz.ch/R-manual/R-devel/library/base/html/00Index.html

[12] SCHLITTLEER, C. A. *Ferramentas de Qualidade – Histograma*. Acessado em Janeiro de 2017. Disponível em: http://koeso.com.br/2014/01/ferramentas-da-qualidade-histograma/

[13] Shiny by RStudio. *Lesson 1: Welcome to Shiny. Acessado em Janeiro de 2017.* Disponível em: https://shiny.rstudio.com/tutorial/lesson1/

[14] SWEENEY, D. J. WILLIAMS, T. A. ANDERSON, D, R. *Estatística Aplicada* à administração e economia. São Paulo: Cengage Learning, 3ª Ed, 2015.

[15] Step Removed One. *Correlación vs. Cointegratión*. Acessado em Janeiro de 2017. Disponível em: https://www.onestepremoved.com/es/correlation-vs-cointegration/.

[16] VIALI, L. Estatística Básica: *Texto I Descritiva*. Acessado em Janeiro de 2017. Disponível em: http://www.pucrs.br/famat/viali/graduacao/engenharias/material/apostilas/Apostila_1.pdf

[17] What is Six Sigma. *Box Plot Diagram to Identify Outliers*. Acessado em Janeiro de 2017. Disponível em: http://www.whatissixsigma.net/box-plot-diagram-to-identify-outliers/

[18] ZevRoss, Know Your Data. *R powered web applications with Shiny (a tutorial and cheat shet with 40 examples app)*. Acessado em Janeiro de 2017. Disponível em: http://zevross.com/blog/2016/04/19/r-powered-web-applications-with-shiny-a-tutorial-and-cheat-sheet-with-40-example-apps/