Deixa-a passar: maravilhosa, poderosa linguagem R

Fala meus guerreiros. É com muito entusiasmo que compartilho com vocês mais um artigo sobre a ciência de dados, dessa vez, falando sobre a maravilhosa, poderosa linguagem R. Embora ela seja antiga (nasceu em 1993, nos campos das universidades, criada por *Ross Ilhaka* e *Robert Gentleman*), hoje tem se tornando uma das principais ferramentas para trabalhar com ciência de dados contemplando comandos (já que é uma linguagem de programação voltada para análise estatística) para coletar, transformar, limpar, visualizar dados e aplicar modelos de aprendizado de máquina. Desculpem os vendedores das ferramentas *POWERBI, Qlik, Tableau* (usados na construção de *dashboards* contemplando gráficos), a *R* permite fazer muita coisa já e o melhor: é grátis e oferece alto nível de customização.

1. Gráficos.

Quando estamos no processo de análise de dados, isto é, coletando dados de diversas fontes, compreendendo cada um deles, é comum encontramos milhares de registros organizados em tabelas e tabelas. E se torna difícil compreender a relação entre os dados visualizando apenas a tabela. Carregar os dados num objeto do tipo *dataframe* (como se fosse uma planilha excel) e visualizá-lo através do comando *View(nome_dataset)* por um lado permite entender quais são as colunas da tabela, porém por outro a relação entre as variáveis não fica tão evidente. No *RStudio* (um ambiente de desenvolvimento para linguagem R), vamos visualizar o *dataframe* 'tips' usando os seguintes comandos:

```
# Carregando o dataframe tips, que está no pacote 'reshape2'
data(tips, package = 'reshape2')
# visualizando dataframe
view(tips)
```

O resultado (das 7 primeiras linhas) será:

Exercicio-Artigo.R × | tips ×

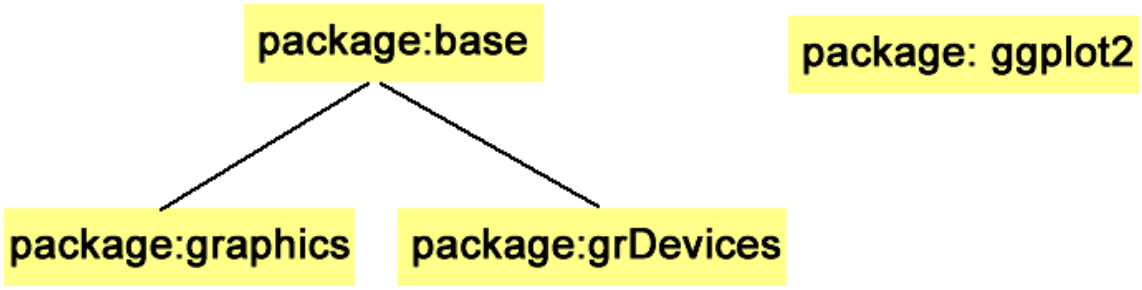
Filter

	total_bill	tip	sex	smoker	day	time	size
1	16.99	1.01	Female	No	Sun	Dinner	2
2	10.34	1.66	Male	No	Sun	Dinner	3
3	21.01	3.50	Male	No	Sun	Dinner	3
4	23.68	3.31	Male	No	Sun	Dinner	2
5	24.59	3.61	Female	No	Sun	Dinner	4
6	25.29	4.71	Male	No	Sun	Dinner	4
7	8.77	2.00	Male	No	Sun	Dinner	2

Esse *dataframe* descreve a gorjeta de várias pessoas num restaurante. O campo *'total_bill'* significa total da conta, *'tip'* é o valor da gorjeta, *'sex'* é o gênero do sexo

(Masculino, Feminino), *'smoker'* é se a pessoa é fumante ou não', *'day'* é o dia da semana, *'time'* é o tipo de refeição (janta, almoço, ...), *'size'* é a quantidade de pessoas na mesa. Como eu sei de tudo isso? Ora, porque são todas as palavras em inglês. Aposto que você me xingou me sua mente dizendo que não precisa saber inglês kkkk. Porém, é um diferencial e vai te ajudar (veja o artigo 'Muito Mais Que Saber O Verbo To Be' e entenda o porquê). Mas agora pergunto a você qual a relação entre a variável *'total_bill* (total da conta)' e '*tip* (gorjeta)'? Será que existe um padrão no sentido que quanto maior o valor da conta, maior a gorjeta? Ou não necessariamente? Para responder, podemos construir um gráfico e visualizar isso. A linguagem R possibilita construir uma série de gráficos de vários tipos (dispersão, barras, histogramas, pizza,...) e customizar cada um deles. E olha que tem os gráficos simples, mas também uns elegantes em termos de aparência, perfeito para impressionar seu chefe na hora de mostrar o resultado do trabalho kkkk.

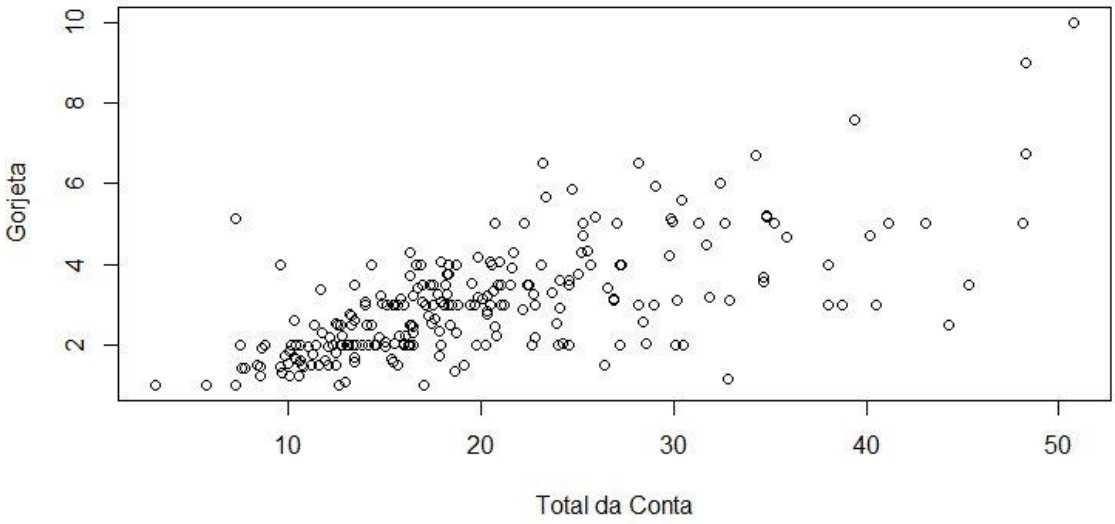
A linguagem R trabalha com pacotes. Existe um pacote chamado *'base'* que já vem instalado por padrão e carregado quando inicializa o *RStudio*. Ele é dividido em dois subpacotes: *graphics* (para trabalhar com gráficos simples) e *grDevices* (para compilar gráficos em arquivos *PDF*, *PNG* entre outros). Se quiséssemos gráficos mais elegantes devemos instalar e usar o pacote *'ggplot2'* (ele tem gráficos lindos e muito mais opções, porém não tão simples para manipular como o pacote *'base'*). Resumindo:



Calma, não me mate (*kkkkk*), eu sei que bastante informação para entender, porém confie em mim que no final dará certo. Continue na aventura guerreiro e você vai descobrir o poder da linguagem R. Bem, retornando ao nosso exemplo, vamos começar com o pacote *'base'* e construir um gráfico de dispersão (associa a relação entre duas variáveis: uma que é independente, outra que é dependente). Vamos executar os comandos:

```
# Comando para evitar referenciar toda hora o nome do dataframe
attach(tips)
# Construção do gráfico de dispersão
plot(x = total_bill, y = tip, xlab = 'Total da Conta', ylab = 'Gorjeta')
```

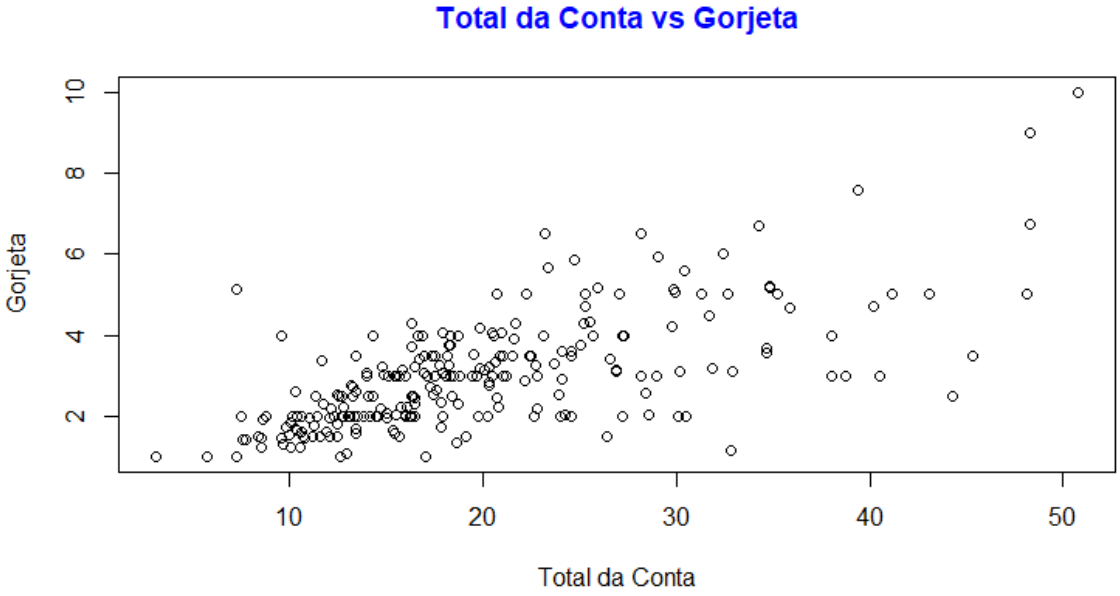
Sem desespero, meu filho (kkkk). Vou explicar tudo! O comando *attach(nome_dataframe)* é para evitar escrever toda hora o nome do dataframe nos comandos. A função *plot* é para construir um gráfico simples, de dispersão. E tem vários parâmetros (x é dado definido pela coordenada X, y é o dado definido pelo coordenada Y, *'xlab'* é o nome que aparecerá em X, *'ylab'* é o nome que aparecerá em Y). O resultado é:



Customização do gráfico é o que não falta aqui. Se quisermos adicionar um título ao gráfico basta usarmos a função *title().* Podemos ainda definir um cor para esse título como, por exemplo, azul. O atributo *'col.main'* é a cor do título principal, enquanto o atributo '*main'* é o título propriamente dito. Veja:

```
# Definindo um título para o gráfico e alterando a cor dele para AZUL
title(col.main = 'blue',main = 'Total da Conta vs Gorjeta')
```

O resultado será:



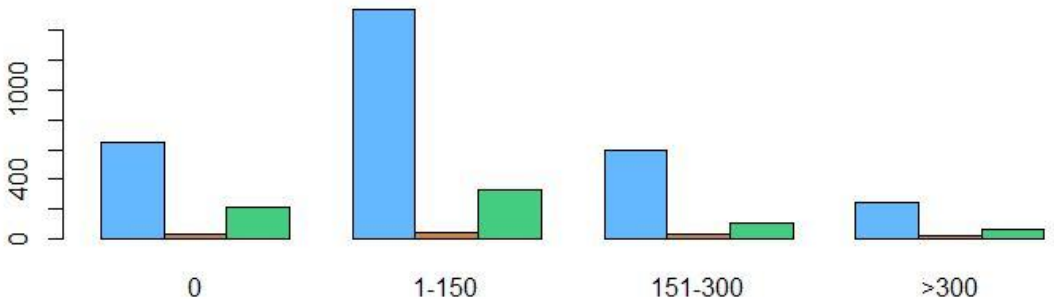
Podemos ainda fazer muita coisa nesse gráfico, porém não vou entrar nesse mérito (senão seria mais de cinquenta páginas escrevendo). Quero que você apenas foque nos pontos do gráfico. Note que há certo padrão quando o total da conta varia entre 8 a aproximadamente 25, as gorjetas ficam próximas de 2 a 4. Isso não conseguiria descobrir

facilmente apenas com o resultado do dataframe em formato de tabela. Para isso serve os gráficos, eles facilitam a visualização da informação.

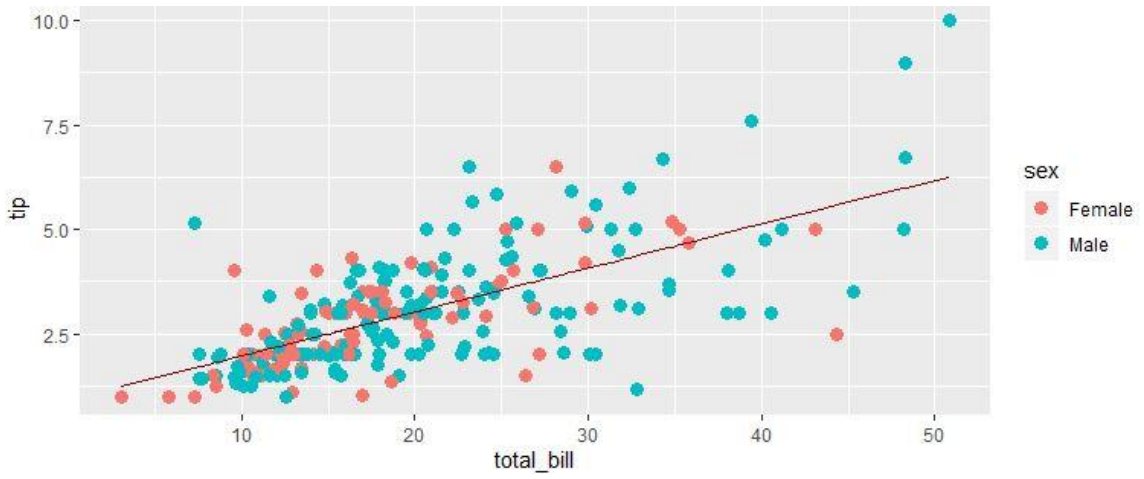
Ainda no pacote base podemos criar outros tipos como, por exemplo, gráfico de barras através da função *barplot().* Note os comandos abaixo. Observe apenas o comando destacado em amarelo, não vou entrar nos aspectos técnicos dos demais. '*Dados'* é o conjunto de dados, ou seja, o dataframe do número de casamentos criado, '*col'* se refere às cores, '*beside'* é quanto à orientação das barras (lado a lado).

```
# Preparando os dados - número de casamentos em uma igreja de SP
dados <- matrix(c(652,1537,598,242,36,46,38,21,218,327,106,67), nrow = 3, byrow = T)
dados

# Nomeando linhas e colunas na matriz
colnames(dados) <- c("0","1-150","151-300",">300")
rownames(dados) <- c("Jovem","Adulto","Idoso")
dados

# grafico de barras
barplot(dados, col = c("steelblue1", "tan3", "seagreen3"),beside = T)
```

Ao executar o comando, o resultado é:



Já um exemplo de gráfico criado pelo pacote *ggplot2* está descrito abaixo. Aqui temos uma 'linha vermelha' que indica um modelo linear (previsão da relação entre os valores '*tip'* e '*total_bill'*, indicando um padrão de comportamento). Podemos ver aqueles pontos que estão próximos dela, como aqueles bem distantes.



2. 0800.

O *RStudio*, o interpretador da linguagem R assim como toda documentação é tudo disponibilizado gratuitamente pelo site oficial deles (https://www.r-project.org/). Se tiver dúvidas, existe uma grande comunidade na internet para ajudá-lo. Diferente de muitas ferramentas de construção de dashboards e gráficos que são pagas, aqui tem muita coisa gratuita (um ou outro pacote mais sofisticado, elaborado é cobrado à parte).

3. Nem tudo são flores

A essa altura do campeonato você deve estar convencido que a linguagem R resolve 100% dos seus problemas, você nunca mais vai precisar do *PowerBI, Qlick, Tableau*, porém tenha cautela. Saiba de antemão, que nenhuma tecnologia vai resolver todos os seus problemas. Na prática, especialmente na ciência de dados, podemos usar várias, cada uma com seu propósito, não existe uma melhor que outra. Como cientista de dados, podemos fazer muita coisa com a linguagem R conforme vimos: na análise de dados construímos os mais variados gráficos possíveis para observar o comportamento dos dados. Não há necessidade, de ficar exportando todos os dados para as ferramentas *(PowerBI, Qlick, Tableau)* para gerar os gráficos. A linguagem R é uma mão na roda.

Como a linguagem R, é open-source, não há suporte oficial da empresa. Se acontecer um erro, *bug*, pode não ter solução e levar muito tempo para a equipe técnica resolve-lo (questão que na maioria das vezes é impossível para a empresa esperar). Quando temos uma empresa proprietária como, por exemplo, a *Microsoft* para o produto *PowerBI*, ela se responsabiliza contratualmente pelo suporte, oferecendo toda a assistência necessária para instalação, configuração e uso incluindo correção de erros e falhas. Esse motivo é que leva as empresas preferirem comprar produtos proprietários ao invés de usar *open-source*.

4. E no fim...

E no fim tudo gira em torno da resposta ‘depende’. Devo usar ou não *PowerBI*? Depende. Devo usar ou não R para análise? Depende ... A linguagem R é poderosa e oferece muitos recursos (como, por exemplo, gráficos de vários tipos e permite personalizá-los) e de forma gratuita (você não paga nada), diferente de ferramentas do tipo *PowerBI, Qlick, Tableau.* Para um trabalho profundo de análise, a linguagem R atende muito bem. Porém se você precisa de um suporte dedicado 100% para seus produtos (e não deixar a ‘deriva’ quando acontecer erros ou falhas), talvez a melhor opção seja usar as ferramentas *PowerBI, Qlick, Tableau.* Abraços e até a próxima pessoal.