Sumário

[Boas Vindas 3](#_Toc116654740)

[Uma ressalva sobre Ciência de Dados 4](#_Toc116654741)

[Por que usar a Linguagem de Programação R para Ciência de Dados? 4](#_Toc116654742)

[(FT) Tudo o que você precisa saber sobre como este curso está organizado 5](#_Toc116654743)

[1. ----- Contextualização R 7](#_Toc116654744)

[1.1 Linguagem R e RStudio 7](#_Toc116654745)

[1.1.1 Instalação do R e RStudio 7](#_Toc116654746)

[1.1.2 Um tour pelo RStudio e Conceitos básicos do R 9](#_Toc116654747)

[1.2 Bibliotecas, Helps e fontes de pesquisa para o R 10](#_Toc116654748)

[1.3 Dados tidy, Pipe (%>%), Tidyverse e Dialetos de manipulação de dados no R 12](#_Toc116654749)

[1.4 Rmarkdown, combinando códigos e comunicação 13](#_Toc116654750)

[2 ----- Importação 15](#_Toc116654751)

[2.1 Importando dados externos para o ambiente R 15](#_Toc116654752)

[2.1.1 Exportando dados, Diretórios E projetos no R (.Rproj) 15](#_Toc116654753)

[2.2 Verificações iniciais 16](#_Toc116654754)

[2.2.1 tipos de dados vs. classes 17](#_Toc116654755)

[3 ----- Exploração (EDA) 18](#_Toc116654756)

[3.1 EDA - bibliotecas de Investigações 18](#_Toc116654757)

[3.1.1 Medidas de Resumo 18](#_Toc116654758)

[3.1.2 Distribuições Empírica dos Dados 19](#_Toc116654759)

[3.1.3 Medidas de Associação 19](#_Toc116654760)

[3.1.4 considerações gerais sobre a EDA 19](#_Toc116654761)

[3.2 ----- Transformação (WRANGLE) 19](#_Toc116654762)

[3.2.1 Selecionando elementos de uma tabela 19](#_Toc116654763)

[3.2.2 Criando, excluindo e alterando elementos DE uma tabela 19](#_Toc116654764)

[3.2.3 Criando sumarizações e agrupamentos (summarize e group\_by) 19](#_Toc116654765)

[3.2.4 Concatenando dados (cbind, join e merge) 19](#_Toc116654766)

[3.2.5 Pivotando dados (wider e longer) 19](#_Toc116654767)

[3.2.6 bônus: manipulação avançada 20](#_Toc116654768)

[3.3 ----- Visualizando os dados (DATAVIZ) 20](#_Toc116654769)

[3.3.1 FundamentalsUm mapa mental: tipos de dados e objetivos de visualização 20](#_Toc116654770)

[3.3.2 Bibliotecas de visualizações: muitas perspectivas em poucas linhas de código 20](#_Toc116654771)

[3.3.3 gramática dos gráficos (ggplot): entendendo a estrutura 20](#_Toc116654772)

[3.3.4 exploração vs. ggplot: tipos de gráficos 21](#_Toc116654773)

[3.3.5 explanação vs. ggplot: refinando os outputs 21](#_Toc116654774)

[3.3.6 bonus: visualizações e recursos interativos: pacotes esquisser e plotly 21](#_Toc116654775)

[3.3.7 bonus: Tabelas 21](#_Toc116654776)

[4 ----- Preparando os dados (DATAPREP) 21](#_Toc116654777)

[4.1 Um mapa mental para o capítulo: o que é dataprep 22](#_Toc116654778)

[4.2 Lidando com dados duplicados 22](#_Toc116654779)

[4.3 Tratando dados faltantes e imputação de dados (missing) 22](#_Toc116654780)

[4.4 Identificando valores extremos (outliers) 23](#_Toc116654781)

[4.5 Aplicando padronizações 23](#_Toc116654782)

[4.6 Diferentes formas de encoding 23](#_Toc116654783)

[4.7 Bônus: feature engineering – trabalhando com datas, textos e fatores 23](#_Toc116654784)

[5 ----- MODELING 24](#_Toc116654785)

[5.1 Um mapa mental para o capítulo 24](#_Toc116654786)

[5.2 Machine Learning (ML) 24](#_Toc116654787)

[5.3 ML supervised ~ métodos aplicados para um problema de classificação e regressão 27](#_Toc116654788)

[5.4 ML unsupervised ~ métodos de clusterização e redução de dimensionalidade 28](#_Toc116654789)

[5.5 Inferência e Modelagem Estatística 31](#_Toc116654790)

[5.6 Bônus: Pontos Importantes 31](#_Toc116654791)

[6.1 Bônus: IA e dados não estruturados 32](#_Toc116654792)

[7 ----- COMUNICAÇÃO 32](#_Toc116654793)

[7.1 O que vai encontrar neste módulo 32](#_Toc116654794)

[7.2 Um pouco mais sobre relatórios com Rmarkdown 32](#_Toc116654795)

[7.3 Dashboards interativos com flexdashboard + plotly e Shiny 33](#_Toc116654796)

[7.4 Considerações gerais 33](#_Toc116654797)

[8 ----- CONCLUSÃO 33](#_Toc116654798)

[8.1 bônus | Interoperabilidade no R -- aproveitando a sua experiencia com: SQL, Python, SAS, etc 33](#_Toc116654799)

[8.2 QUIZ 33](#_Toc116654800)

# Boas Vindas

* ***Vídeo de boas-vindas***

Olá, boas-vindas ao curso "Linguagem de Programação R para Ciência de Dados", eu me chamo Nathália Demetrio, sou estatista e trabalho com dados há mais de 10 anos, além de ser mãe do Gustavo, apx pela lingugem R, e a sua instrutora! E vou te acompanhar nesta jornada.

Neste curso organizei os conteúdos considerando quatro níveis de objetivos em uma análise de dados, no caso: as análises com objetivos descritivos, diagnósticos, preditivos e prescritivos, que podem ser traduzidas como:

* "o que aconteceu?",
  + Examinar o histórico de dados para descrever uma questão
* "como aconteceu?",
  + Identificar padrões e relações nos dados
* "o que irá acontecer?" e
  + Prever os acontecimentos com base em padrões históricos
* "como fazemos acontecer?"
  + Identificar as ações que geram os resultados desejado por meio de experimentos

Note que existe uma hierarquia aqui, já que você não vai fazer um bom diagnóstico sem ter realizado uma boa análise descritiva, bem como não vai ter um bom modelo preditivo sem ter feito um bom diagnóstico, e assim por diante.

Agora, fazendo um de-para entre este objetivos e as etapas de um fluxo de trabalho na ciência de dados, teremos os seguintes módulos: obtenção e exploração de dados; manuseio e visualização das informações obtidas; pré-processamento e modelagem; e, por fim, testes e comunicação.

Aqui eu gostaria de fazer um disclaimer: nesta estrutura não vamos abordar o que poderia ser um quinto pilar, transversal, que eu vou chamar de “Produtizar” , onde teríamos um olhar de produto, algo visando facilitar um objetivo através do uso de dados, um “produto de dados”, contemplando coisas como algoritmos para automatização da tomada de decisão

Note que aspectos como deploy e manutenção de modelos não estão sendo considerados neste flow. Isto porque, apesar da incontextável importância, tais fases tendem a priorizar aspectos como integrações com sistemas, performance, e questões relacionadas, enquanto que para o ciclo pontuado acima, temos o entendimento e a modelagem do problema como meta principal.

Notem que pilares mais relacionados a produtização da solução dados, construção de pipelines e deploy de maneira geral não serão cobertos neste curso. Mas ó, nos pautamos em dados por aqui, então comenta aí se for algo que você gostaria de aprender a fazer em R, que a gente prepara 😊

Mas não iremos contemplar este pilar aqui tanto por se tratar de um curso básico, quanto por que ao falar de produtização passamos a englobar novos desafios, particularmente relacionados ao universo de desenvolvimento de software, desde a criação de pipelines, com a automatização de tarefas, testes de código e deploy. Mas antes de pensar em automatizar as etapas anteriores, precisamos dominá-la, certo? E é isto que faremos neste curso, garantir que você se empondere do descritivo ao prescritivo.

Nos vemos nos próximos vídeos.

# Uma ressalva sobre Ciência de Dados

* ***Neste vídeo faremos uma rápida contextualização sobre ciencia de dados***

Existem diferentes formas de definir e entender a ciência de dados, se você conversar com 10 pessoas cientistas, possivelmente receberá 11 definições. Mas uma visão usual, trata da ciência de dados como uma intersecção entre três esferas:

* Matemática ou estatística
* Ciencia da computação/tecnologia
* Área de domínio, ou seja a área de interesse

Um ponto de atenção aqui, é que podemos ser levados a acreditar que seja necessário dominar estas três disciplinas, resultando em um profissional usualmente denominado como “unicórnio de dados”.

Contudo é mais realista considerar que existem diferentes profissionais atuando com dados. Sejam aqueles que possuem um maior domínio matemático, da computação, ou da área a qual estamos trabalhando.

A conclusão aqui é: unicórnios não existem, já ornitorrincos sim 😊

E porque ornitorrincos? Bem: são mamíferos que colocam ovo, bico de pato, cauda de castor, um verdadeiro cross-over no mundo animal! Uma realidade muito mais próxima do que é esperado de uma pessoa que trabalhe com dados: UM POUCO de tudo, pois para ter muito de tudo, voltamos ao unicórnio...

. [passar os slides para português] .

Dito isto, uma outra forma de pensar em ciência de dados, é dar enfoque no que é feito, e será este o viés que será considerado aqui. Contemplar os pilares mais essenciais para garantir a entrega de valor em uma análise de dados. Existem diferentes maneiras de enxergar estas etapas, podemos comentar do CRISP-DM, este frame introduzido pelo Hadley, ou ainda esta ilustração da Allison Horst – uma importante personalidade da comunidade R. E apesar de pontos como: importação, transformações e modelagem, serem transversais a quase todas as propostas, o principal ponto que eu gostaria que vocês absorvessem é a característica cíclica do processo. Pois a não-lineariedade é o ponto chave da ciência de dados. Afinal para cada etapa que avançamos, entendemos mais sobre os dados, sobre o problema, sobre as hipóteses envolvidas.

E com isto em mente, vamos lá?

# Por que usar a Linguagem de Programação R para Ciência de Dados?

* ***Neste vídeo falaremos sobre porque usar a linguagem de programação R.***

Vamos começar com:

* O que é o R?
  + R é uma linguagem de programação interpretada
  + Vale comentar que o R tem bastante tempo de estrada
  + R sendo uma brincadeira tanto para nome dos criadores, tanto para provocar o S, por ser uma implementação desta linguagem
* Mas pq o R? Quais vantagens?
  + Ambiente open-source ou seja x--- arrumar uma explicação/frase acessível ----x e cross-platform, basicamente possibilitando a integração com outras linguagens e ferramentas
  + Integrações
  + Linguagem focada: em contraste com alternativas tem isto como possibilidade e não foco, e aqui é a dinâmica especialista versus generalista, R é uma linguagem especialista.
    - Análises avançadas, tamanho de base e reprodutibilidade
  + comunidade
* E quais os meus pontos preferidos do R?
  + Bibliotecas
    - Hoje no CRAN, o repositório oficial tem mais de 18K
    - Você precisa ser muito criativo para pensar em um problema, ao passo que não precisa se preocupar em implementar rotinas para
    - E mesmo que pense, você poderá implementar o desenvolvimento, com a ajuda da comunidade e compartilha-la
  + Uso de recursos avançados por pessoas com menos know-how tecnico
    - Um ótimo exemplo é o shiny, em que mesmo sem saber nada sobre servidores, css , htmlou tantas outras tecnologias importantes, você ainda assim consegue desenvolver e compartilhar um dashboard.
    - MAs não se engane, o R possui também uma comunidade de usuários madura, com desenvolvimento de ponta, sendo inclusive a porta de entrada de muitas novas metodologias
  + Novas metodologias
    - Por ser uma ferramenta usual no desenvolvimento de pesquisa cientifica, além de ser a ferramenta usual de departamentos de estatística (área que tem como foco o desenvolvimento de metodologias para análise de dados
  + Comunidade
    - desenvolvimentos sejam relevantes para diversas áreas de atuação
    - eventos recorrentes
  + Por fim mas não menos importantes: os stickers!
    - São basicamente adesivos em formato de hexogramas, que representam normalmente bibliotecas ou comunidades relacionadas, como se fossem mascotes.
    - Aqui podemos ver por exemplo temos o sticker da comunidade R-ladies, um grupo que promove a diversidade de gênero dentro da comunidade R, ou o satRday, um evento que usuários de R organizam localmente para passar um sábado compartilhando conhecimento.

E dado tudo isto, espero que você esteja motivada ou motivado para iniciar esta jornada.

Vamos lá?

. [em algum dos pontos falar sobre o Python? ~ assim como SQL tem outros propósticos]

# (FT) Tudo o que você precisa saber sobre como este curso está organizado

* ***Antes de iniciarmos, gostaria de dar algumas dicas sobre como obter o máximo deste curso:***

. [o que apresento na tela? / Isto poderia ser um texto em algum lugar?] .

Vamos fazer alguns alinhamentos rápidos:

* Sobre como o curso está organizado:
  + Iniciaremos os módulos com mapas mentais, para te ajudar a entender sobre a importância do tema, e quais os principais fatores que devem ser levados em consideração para tomar as principais decisões. Seguido de aplicações e recursos que nos permitam obter resultados de forma rápida e estilizada. Seguido de alguns temas extras e referencias, visando garantir ferramentas e direcionamentos suficientes para que você possa seguir evoluindo nos temas de acordo com as suas próprias necessidades.
* Sobre a dinâmica dos videos
  + Vc pode so assisir os vídeos, mas recomendo fortemente que faça o download dos scripts, visto que são neles que os conceitos estão descritos com mais detalhes. Isto porque para garantir o tempo do curso, iremos apenas enunciar alguns trechos dos scripts, e o aprofundamento dos conteúdos fica por sua conta, combinado? 😊 Inclusive se vc quiser ver algum código com mais atenção, fique à vontade para pausar e voltar o vídeo.
* Sobre os pré requisitos
  + E o curso foi pensado para dar uma visão prática e generalista, saber programação e conceitos estatísticos será útil, mas não são mandatórios. O curso foi pensado, para ser acessível para quem está no básico, e ser interessante para quem já possua alguma trajetória com dados. Inclusive, mesmo nos capítulos que demandariam conceitos estatísticos mais avançados, não aprofundaremos na teoria, mas sim em conceitos chaves, heurísticas e nos resultados que podem ser obtidos, mas sempre compartilhando referencias para que você possa seguir evoluindo no tema.
* Sobre o que você precisa para conseguir executar
  + O curso foi realizado no Windows, considerando as versões R versão X e RStudio versão X, e apesar de ser possível trabalhar com versões anteriores, a recomendação é que você mantenha as ferramentas nas versões mais atualizadas, e consequentemente com mais recursos.
    - Instalação R, Rstudio, e os pacotes:

pacotes\_necessarios <- c(

"rmarkdown",

"knitr",

"tidyverse",

"janitor",

"prettydoc"

)

install.packages(pacotes\_necessarios)

* Sobre os exercícios
  + Ao final de cada modulo teremos quizzes
  + Mostrar a pasta com os scripts, e como estes estão estruturados. Cada script é denominado de acordo com o nome do vídeo, mesmo no caso em que o vídeo trate apenas de links, estes estarão listados nos scripts. No mais, utilizaremos scripts .R ao longo do primeiro capítulo, de contextualização, e posteriormente seguimos com arquivos .Rmd (Rmarkdown).

Por fim, apesar deste curso ser em português existem muitos utilitários e referências importância da área que são apresentados em tela em inglês. Mas não se preocupe, sempre que se tratar de algo importante, farei a tradução para o português 😊

# ----- Contextualização R

***Olá, este capítulo tem por objetivo contextualizar a linguagem de programação R, e alguns dos seus conceitos mais importantes, como: instalação e dicas do R e da IDE RStudio, como escrever e executar scripts .R e .Rmd, e instalar e trabalhar com bibliotecas no R.***

Este é um módulo de apresentação dos conceitos mais básicos do R, tendo nos scripts uma série de textos, dicas e referências sobre a linguagem, que serão contextualizados nos vídeos, mas que possuem como objetivo que você possa utilizar para consulta ao longo do curso. Sendo assim trata-se de um módulo opcional, então sinta-se à vontade para pular caso você já tenha alguma experiência com o R, ou mesmo se preferir aprender fazendo. Contudo, independente da sua escolha, recomendo que baixe os scripts e se familiarize com alguns termos, para poder utilizá-los como consulta quando necessário. Recomendo também, que ao final do curso, faça este exercício novamente. Isto porque é comum que após ter uma maior vivência, consigamos aproveitar melhor os conceitos que você encontrará neste módulo.

Por outro lado, caso você sinta a necessidade de um aprofundamento maior do que o que veremos aqui, recomendo que consulte o curso da Jessica Temporal, em que os conceitos da Linguagem R são apresentados com maior detalhe. Combinado?

Então, vamos lá?

[concatenar os três primeiros módulos: instalação, Rstudio + IDE e tour/conceitos básicos]

# Linguagem R e RStudio

# Instalação do R e RStudio

***Neste video aprenderemos a instalar o R e o RStudio, pois sim, são coisas diferentes, e precisam ser instaladas nesta ordem. Por fim comentaremos como fazer o update de cada uma destas ferramentas.***

* Slide instalação

**Para instalar o R o primeiro passo é entrar** na página oficial do R Foundation: <http://www.r-project.org/>

(‘The R Project for Statistical Computing’), nela além de uma contextualização sobre a linguagem, encontramos também **informações sobre as versões mais recentes**, e alguns das principais referências da comunidade, **eventos de interesse a tutoriais.**

E na sessão Downloads, opção CRAN, escolha o servidor do local mais próximo ao qual você se encontra, ou vá direto para <https://cloud.r-project.org>, escolha o seu sistema operacional (utilizaremos o Windows como referência), o subdiretório “base”, e então fazer o download da versão mais recente do programa:

Click duas vezes no arquivo de instalação baixado, selecione o idioma de sua preferência, e então você estará no **Assistente de Instalação**.

Caso você não seja administrador, na tela em que especifica a pasta em que o programa será instalado, é necessário alterá-lo para um repositório dentro do seu usuário. No mais, basta seguir o padrão de instalação do Windows até a conclusão da instalação. Pronto!

Tendo o R na sua máquina, você pode acessá-lo via

**- via sistema de busca do seu sistema operacional**

**- ícone na área de trabalho**

**- ou pelo executável que se encontra na pasta em que o R foi instalado**

E fazer a estreia da linguagem com o clássico "Hello World" na janela “R Console”.

Via point-and-click temos opções que vão desde mudanças de diretórios até a instalação de bibliotecas. Porém**, por se tratar de um interpretador, tais opções são limitadas, e temos poucos recursos em termos de usabilidade**.

Como uma alternativa de interface mais amigável para trabalhar com o R, temos o RStudio, que é também a principal ambiente de desenvolvimento da linguagem R, e oferecendo recursos desde a edição de códigos até a visualização de dados. Para fazer o download do RStudio entre na página oficial do RStudio (<https://www.rstudio.com/>), na opção ‘Download’ RStudio, após passar pela descrição das opções disponíveis do RStudio, você encontrará a lista de instaladores organizados segundo os sistemas operacionais e processadores compatíveis. A partir do arquivo de instalação, selecione o idioma de sua preferência, e siga no **Assistente de Instalação**.

Uma vez instalado, para acessar o R Studio, podemos ir no diretório, ou ainda via buscador do sistema operacional.

É comum que o Rstudio seja um pouco intimidador inicialmente, mas não se preocupe, nos próximos vídeos você receberá um verdadeiro banho de loja dos recursos que o Rstudio oferece.

* Rstudio vs. posit
  + **Missão:** ajuda cientistas a dar sentido para o mundo por meio de dados
  + É legal comentar que esta organização se tornou uma **Empresa de Benefício Público**, B-corporation, basicamente um selo dado a empresas que reportam anualmente benefícios públicos.
  + **Produtos: desenvolvimento de ferramentas** gratuitas e open-source, para ciência de dados, pesquisa cientifica e comunicação técnica. E apesar de também existirem **ferramentas pagas**, manteremos o foco no arsenal gratuito disponível, mais do que suficiente para os nossos objetivos.
  + **people** E porque eu estou falando desta organização? Bem, pq iremos utilizar muitos dos recursos desenvolvidos por lá, isso porque muitos colaboradores e colaboradoras importantes da comunidades R atuam na RStudio, garantindo: materiais didáticos, desenvolvimento de pacotes, organização de eventos e muito mais.

Até lá.

# Um tour pelo RStudio e Conceitos básicos do R

***Neste vídeo conheceremos alguns dos recursos oferecidos pela IDE RStudio, e um script em que revisaremos alguns conceitos, termos e características básicas da linguagem R.***

* Slide paines do RStudio

Ao abrir o RStudio, trata-se de uma ferramenta para facilitar o desenvolvimento, com recursos para edição de códigos, execução de scripts, debugar. Trabalhando neste ambiente, temos, por exemplo, a identificação de códigos errados, a indicação de qual o problema no código, e até trabalhar com outras linguagens, quer ver? Ao abrir novo script, além dos formatos relacionados ao R, Temos java, c... vários sabores.

Apesar de ser um pouco intimidadora, visto a quantidade de opções, recomendo que investigue, sempre. E garanto que cada vez que o fizer, vai perceber um recurso novo. Recursos mais que suficientes para se apaixonar por esta IDE – Vc já ouviu a frase “muito do que torna um livro bom é te-lo lido no momento correto para nós”, pois bem, conforme você for ganhando maturidade e ampliando os seus horizontes, vai descobrindo mais recursos uteis para vc,

Nesta IDE encontramos diferentes paines que irão nos ajudar em diferentes momentos de uma análise de dados, vamos explorar:

* Console, é aqui que interagimos diretamente com a linguagem R , assim ao enviar um comando, recebemos o seu output.
* Contudo, trabalharemos por meio de scripts que iremos desenvolver as nossas análises de modo que de modo a garantir a posterior reprodutibilidade dos estudos feitos bem como uma melhor organização durante o trabalho. Note porém que para termos o nosso trabalho salvo é necessário salvar o script. Para este resumo iremos trabalhar com um script pronto que você poderá utilizar para posterior consulta e revisão.
  + Mostrar como salva
  + Arquivos .R, sua extenção
  + Qualquer nova alteração – print(“hello world”) bastante salvar novamente
  + Fechar e abrir novamente
  + Comparar um script na IDE vs. não ser: o .R é um texto, vc pode abrir em um editor de texto.
* Trabalharemos com um arquivo já preenchido
  + Mostrar as sessões/foldings
  + Tour
    - Passar pelo o que tem lá
    - Comentar do help – cheatsheet
    - Algumas dessas janelas só estarão disponíveis e atualizadas, em termos de funcionalidades, em versões mais recentes – Aqui temos informações como: a versão do R que estamos trabalhando, ou a quantidade de memória que está sendo utilizada – isto se você estiver utilizando uma versão.
    - Tutorial particularmente interessante é o Tutorial – legal né?
    - Todos entes painéis possuem muitos recursos uteis, trabalho há mais de 10 com rstudio, e sempre .
  + A mesma premissa vale para a barra de feramenta
    - Recursos para encontrar e substituir textos
    - Trabalhar com Identação
    - Debub de códigos
    - Uma aba particularmente interessante é o Tools> Global Options
      * tipos de layout
      * mudando a estrutura dos paines
  + Voltando ao script, conforme comentei, temos aqui uma contextualização geral da linguagem
    - Você pode acessar por aqui as opções
    - R como calculadora (uma das boas como diz a Jessica)
    - ....
    - Funções (trazer a
    - ....
    - Para entender computação no R, **John** McKinley **Chambers** , um importante nome da comunidade R, definiu: tudo o que existe no R, é um objeto, e tudo o que acontece é uma função.
    - ...
    - recomendo que leia este script, mesmo que nem tudo faça sentido inicialmente, vai te ajudar a ter uma noção mais clara das características da linguagem
    - Deixaremos aqui também um script com a contextualização sobre estruturas de controle e como criar suas próprias funções
    - Note que tudo que estamos vendo até aqui é built in do R, ou seja, você não precisa instalar nada a mais.

Até mais

# Bibliotecas, Helps e fontes de pesquisa para o R

***Neste vídeo falaremos sobre bibliotecas, um dos conceitos mais essenciais do R, visto permitir acessar todo o arsenal de recursos disponíveis para a linguagem. Bem como sobre pesquisas e recursos para obter ajuda no R, habilidade esta tão importante quanto a programação em si.***

Uma das maiores vantagem de trabalhar com o R são as suas muitas bibliotecas. Apenas no CRAN, o seu repositório oficial, existem mais de 17mil pacotes! Garantindo: maneiras mais otimizadas para trabalhar com metodologias já consolidadas, técnicas mais recentes ou mais especializadas, integrações e recursos tangentes à análise de dados de modo geral.

* Mas o que são bibliotecas?
  + Basicamente coleções de funções R, conjuntos de dados e código compilado em um formato bem definido.
  + um dos conceitos mais essenciais do R – por vezes também denominado como pacotes, apesar de existirem algumas diferenças técnicas entre estes termos, você irá encontra-las, em geral, de forma intercambiável
  + paralelo com livros
  + um conceito super importante no R! visto o poder de fogo
* aba packages
  + o que já temos instalado no computador, se vc nunca instalou nada, ainda assim teremos algumas bibliotecas, no caso, as bibliotecas nativas
* instalação
  + instalação via point-and-click ou comando
    - como o comando está aqui, vou mostrar o point-and-click
  + ver o que acontece na janela, que aparece após instalar
  + vamos testar com a biblioteca praise, uma biblioteca que possui uma função apenas
  + após instalar conseguimos sair usando? Não! Vamos testar: esta bibliote
    - existem dois momentos
  + carregar – comando library
* alternativas :
  + instalação: versão de desenvolvimento: devtools, github -- considerando não apenas CRAN, mas também fontes riquíssimas, como o github.
  + carregamento: require
* pontos importantes:
  + mostrar como checar sobre versões
  + formas de carregar apenas uma função – boa prática
    - funções homônimas
    - desta forma podemos consultar as funções e bases de dados de uma biblioteca
* extra
  + addins: são suplementos, extensões que fornecem um mecanismo simples para executar funções R avançadas de dentro do RStudio. ao executar um addin o código correspondente é executado sem que você precise escrever o código. Addins são cápsulas de código R que podem ser executados interativamente a partir do menu, servem para fazer diversos tipos de tarefas, como identificar seu código, ajudar nas tarefas de copiar e colar, insir códigos no seu script e muito mais. Os addins são criados em pacotes que são instalados, então cada pacote vai nos disponibilizar addins que facilitam algum tipo de tarefa.

E em relação à executar pesquisas e consultar documentações, algo tão importante quanto a programação em si, visto que sempre estaremos esbarrando em novos desafios.

* O que é
  + Primeiramente falaremos xxxx
* Aba help
  + Point-and click:
    - Pode ser intimidador
    - Pequeno tour: description, usage, arguments Examples
  + Busca ? e help
* Algumas outras opções de busca no R
  + Google – a mais importante.
    - Stack Overflow é uma ótima fonte de respostas para perguntas comuns de markdown. Também é um ótimo lugar para obter, depois de criar um exemplo reproduzível que ajude a ilustrar seu problema.
    - A comunidade RStudio é um lugar amigável para fazer perguntas sobre o rmarkdown e a família de pacotes R Markdown.
* Fontes de pesquisa
  + Ampliar o seu arsenal
  + Cheatsheet - Alguns links -- sticker
  + package finder
  + Aba Tutorial
  + Atalhos
    - Help / cheatsheet --- passar para o módulo da rstudio

E com tais ferramentas espero que você esteja cada vez mais seguro (a) para seguir evoluindo na linguagem.

# Dados tidy, Pipe (%>%), Tidyverse e Dialetos de manipulação de dados no R

***Neste vídeo falaremos sobre o tidyverse, bem como alguns dos conceitos chaves relacionados a este meta-pacote, como: dados tidy, operador pipe (%>%) e a relação entre as bibliotecas que compõem o tidyverse e o ciclo de análise da Ciência de Dados. De maneira complementar faremos uma breve comparação entre as diferentes sintaxes do R para Ciência de Dados, no caso: Rbase, data-table, e, é claro, o tidyverse.***

* Slide : tidyverse

....

* Slide : dados tidy e operador pipe

Neste vídeo falaremos sobre tidyverse, o pacote que irá nos acompanhar durante todo o curso, visto ser um sistema de pacotes coerente para manipulação de dados.

* Slide : tidyvserse e seus pacotes

Neste

* Slide de dialetos

Existem diferentes dialetos no R, ou seja, diferentes sintaxes, diferentes maneiras de escrever uma mesma ação, com especificidades e estruturas próprias. Neste script você encontrará referências para três das principais sintaxes da linguagen R: o Rbase, que é a linguagem nativa do R, ou seja, àquela que você terá acesso caso não instale nenhum pacote, o tidyverse que é um meta-pacote, ou seja, um conjunto de bibliotecas, XXXXX e o data-table, uma biblioteca apenas, nos permite um backend otimizado. Aqui você também encontrará a comparação de código entre estas três alternativas, considerando algumas das ações mais usuais em termos de manipulação de dados.

* Rbase
  + mais próximo da linguagem de programação pura
  + projeto mais longo prazo

* tidyverse
  + foi cunhado pelo Hadley, produz diversos materiais, a maior coleção de pacotes para DS do R, e encadeando comandos com pipe %>%
  + o pacote que mudou como o R é utilizado. Aprenda a utilidade dos pacotes que fazem parte do Tidyverse, como dplyr, ggplot2, tibble e muitos outros
  + quase como um google translate para ETL
  + este será o dialeto que utilizaremos primordialmente ao longo do curso
* data.table
  + Msm quando a base cresce muito

e você pode me perguntar, mas ué, se eu nem sei muito sobre uma, porque Nathália você está me contanto sobre várias. E aqui a questão é dar noção que existe mais de uma forma de fazer a mesma coisa. O que tende a ser confuso para quem ainda não está confortável com a linguagem, e ainda mais para quem sequer sabe sobre a variedade de sintaxes, de opções. O objetivo aqui é que você saiba que isso existe, e que ao longo do curso iremos dar o enfoque no tidyverse.

Vale ainda comentar que é possível fazer a tradução entre o tidyverse e o data.table, visto que os seus pontos fortes possuem uma certa complementariedade, por meio de um pacote chamado dtplyr que fornece um backend data.table para o pacote de manipulação de dados do tidyverse, o dplyr. Permitindo escrever um código dplyr que seja automaticamente traduzido para o código data.table equivalente,em geral, muito mais rápido: <https://dtplyr.tidyverse.org/>

E aqui gostaria de fazer um disclaimer: não se trata de melhor ou pior, existe primeiramente uma questão de preferência envolvida, mas principalmente de contexto. A depender do seu cenário, uma ou outra sintaxe pode fazer mais sentido, no fim do dia, quanto mais sintaxes, pacotes, linguagens de programação você dominar, mais recursos você terá na sua caixa de ferramentas.

Até,

# 1.4 Rmarkdown, combinando códigos e comunicação

***Neste vídeo faremos uma breve introdução ao R Markdown, um tipo de arquivo que nos permite uma maior interatividade entre códigos, outputs e textos, tornando a comunicação dos resultados muito mais prática e acessível.***

O que é ?

* uma alternativa ao script .R, que estávamos utilizando até então.
* Aqui passamos a ter como opção trabalhar com textos, códigos e outputs de forma mais prática e harmônica. Permitindo tanto uma comunicação  analyst-to-analyst, quanto a criação e disponibilização de um relatório, no caso uma comunicação analyst-to-decision-maker.
* A maneira mais fácil de criar um novo documento R Markdown é utilizando o modelo default dentro do RStudio. Vá para Arquivo > Novo Arquivo > R Markdown. No novo assistente de arquivo, você pode
  + Forneça um título de documento (opcional, mas recomendado),
  + Forneça um nome de autor (opcional, mas recomendado),
  + Selecione um formato de saída padrão - HTML é o formato recomendado para autoria e você pode alternar o formato de saída a qualquer momento (obrigatório)
  + Clique em OK (obrigatório).

Uma vez dentro do seu novo arquivo .Rmd, você poderá excluir todo o texto abaixo do YAML e preencher seu próprio .Rmd por:

• Adicionando pedaços de código (atalho de teclado: Ctrl + Alt + I; OS X: Cmd + Option + I),

• Escrever prosa com formatação Markdown e

• Executando cada pedaço de código interativamente clicando no ícone dentro do RStudio.

* + Estrutura
    - Cabeçalho
    - Chunks -- onde iremos escrever os códigos, e que, se rodarmos, teremos os outputs desejados
      * importante também comentar que os chunks não podem ter nomes repetidos
    - Textos markdowns - E textos, nos permitindo uma melhor descrição do trabalho, No caso dos textos, no caso, textos markdowns – uma sintaxe usada para padronizar e facilitar formatação de texto na web.
  + Para rodar precisaremos salvar
    - Extenção – não é .R
    - Use o botão “Knit” no IDE do RStudio para renderizar o arquivo e visualizar a saída com um único clique ou use o atalho de teclado Cmd/Ctrl + Shift + K.
    - Você também pode clicar em “Knit to HTML” novamente para renderizar o documento completo com todos os pedaços de código.
    - Vale mencionar que esse formato já permite uma melhor visualização e divulgação do que o script que estava trabalhando, visto que a pessoa teria que rodar no próprio RStudio ou algo assim.
    - Para obter mais ajuda para começar a usar o R Markdown, consulte o site do R Markdown ou use os links “Começar” na parte superior desta página

Adicionalmente a isto, este documento nos permite criar um documento, por default em extenção .html, que pode ser facilmente compartilhado. Para tal precisamos apenas salvar este arquivo, isto porque o R salvará este novo arquivo no mesmo diretório, e com o mesmo nome, mudando apenas a extensão. No caso, de .Rmd para .html, .pdf, ou o que vc desejar.

Agora que já sabemos como criar um arquivo modelo, vamos entender um pouco mais sobre a sua estrutura básica, e para isto vamos utilizar um arquivo modelo, em que adicionei algumas observações para te ajudar a utilizar melhor esta ferramenta:

* o cabeçalho é escrito na linguagem Yaml, não sei como pronuncia. Aqui é importante comentar que a indentação é muito importante, se você utilizar um espaço a mais ou a menos, o código não irá rodar
* # são utilizados para indicar Capitulos, ## para subcapítulos, e assim por diante
* Como trabalhar com códigos
  + Criar chunk - chunks, podemos criar aqui: ou via atalho
    - inclusive considerando outras linguagens de programação, mas não trataremos deste tema aqui.
  + Rodar
    - mencionar porque rodar com Knit e rodar com Run
* Como trabalhar com texto
  + Visual vs. Source
* Helps
  + acessar o Help, dentro do RStudio

Até o próximo vídeo

# 2 ----- Importação

Ref.: <https://www.linkedin.com/learning/data-wrangling-in-r-14135737/exercise-files?autoSkip=true&autoplay=true&resume=false>

# Importando dados externos para o ambiente R

* Slide das extensões

Tela de celular

Descrição gerada automaticamente com confiança média

* # Importando dados tabulares
  + Point and click
    - tipos de arquivos: excel, txt, csv.
    - Ver o resultado na aba objetos
  + Código
* Outras extensões de dados para importação
* Help R Data Import/Export
* Slide das Cheatsheet
* Importando Dados dos pacotes do R
  + Bases que existem no R
  + data() e como ver no environment
  + Praticamente todas as bibliotecas possuem bases de dados interessantes, além das bibliotecas com a finalidade de prover dados.

# Exportando dados, Diretórios E projetos no R (.Rproj)

[talvez separar do de importação]

* Exportando um objeto criado
  + Getwd() e Setwd()
  + A barra é invertida
  + Ver o resultado na aba files
  + Diferença R e Rmd
* Slide do Rproj

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

* Trabalhando com projetos
  + O QUE É, PORQUE É IMPORTANTE, E COMO FAZER ISTO COM PROJETOS NO R (.RPROJ)
  + Vantagem da linguagem cientifica, método cientifico, garantir reprodutibilidade

# Verificações iniciais

**Views**

* Visualizações do RStudio - Apertar o View pelo environment
* Head,

**Nomenclaturas**

* Linhas: observações
* Colunas: variáveis / features
* Valores: intersecção das variáveis e observações

**Tipos de Dados**

* Glimpse
* Cálculos dependem da natureza da variável
* Mas vamos começar pelo começo, entendendo a maneira como o R interpreta os dados. Nesta tabela temos: Character, data e double
* Summary
  + E porque é tão importante, pois isto limita os cálculos que podem ser feitos, com cada informação. A pergunta: Qual a média de duração, faz sentido! Já uma questão sobre a média do nome dos filmes, não faz sentido, não conseguimos fazer esta conta. Ordem de lançamento por sua vez faz sentido! Porém esta informação está codificada como character, o que faz com que o computador entenda os valores como se fossem letras, e não números.
  + Na verdade veremos o quanto o “tipo do dado” pode e deve direcionar a sua forma de pensar em praticamente todos os momentos de uma análise de dados.

Interface gráfica do usuário, Tabela

Descrição gerada automaticamente

# tipos de dados vs. classes

#No R existem 6 tipos de dados, em geral trabalhamos com 4: character > numeric > integer > logical

#Estes dados podem ser organizados em 5 estruturas: vetor atômico, matriz, array, data.frame e listas

**Ajustando o tipo de Dados**

* A maneira como o computador interpreta as informações é extremamente importante. Pois, como vimos, direciona a maneira como os cálculos são realizados. E não necessariamente o computador irá interpretar o dado corretamente, sendo necessário ajustar
* Exemplo – mutate
  + Não se preocupe que na sequencia falaremos mais sobre esta função!
* Tipos de dados geral
  + Qualitativas / Categoricas
    - Nominal
    - ordinal
  + Quantitativas / Numéricas
    - Discretas
    - Contínuas
  + Contudo, existem alguns dados que apesar de poderem ser vistos como subcasos da organização acima, são relevantes a ponto de terem características próprias, é o caso de variáveis: lógicas e de data, por exemplo.
  + Quantitativos - Quantifica ou mede 
    Contínuos: 
    Discretos: 
    Assumem valores 
    Assumem valores 
    em um intervalo 
    em um conjunto 
    especificado de 
    contínuo de 
    números. 
    números. 
    Qualitativos - Característica ou qualidade 
    Ordinal: 
    Nominal: 
    Característica que 
    Característica que 
    possui uma ordem 
    não possui ordem. 
    de grandeza. 

* Slide do tipo dos dados

Interface gráfica do usuário, Aplicativo, Tabela

Descrição gerada automaticamente

* Dica: estrutura de dados
  + Ou seja, se tentarmos colocar tipos diferentes de objetos em um mesmo vetor
  + o R irá modificar o tipo dos objetos de modo que todos passem a pertencer à mesma classe.

Adicionar algo sobre os tipos

# ----- Exploração (EDA)

Falar de EDA, o que significa o termo

# EDA - bibliotecas de Investigações

* janitor – meu pct preferido ,não vamos mexer muito com ele aqui, mas confia, ele é sucesso
* inspectfd
* skimr
* + dicas

"This is my favorite part about analytics: Taking boring flat data and bringing it to life through visualization." ~ John Tukey

# Medidas de Resumo

Primeiro vamos refletir sobre uma estrutura que nos permita ter um guia para seguirmos com as analises para podermos evoluir na investigação, usaremos dois conceitos:

[identificar um feature que a média não faça sentido

Objetivos de sumarização, que vão nos ajudar a descrever o atributo

Diagrama, Linha do tempo

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente com confiança média

[avaliar os resultados do skimr e pontuar um insight]

# Distribuições Empírica dos Dados

....

# Medidas de Associação

....

# considerações gerais sobre a EDA

...

"an aproximate answer to the right problem is worth a good deal more than na exact answer to na approximate problem" John Tukey

Na etapa de exploração começamos a estudar os dados visando compreendê-los segundo uma perspectiva analítica, entender características, padrões e relações que cada variável estudada apresenta.

# ----- Transformação (WRANGLE)

Limpeza e manipulação (data cleaning e wrangling)

# Selecionando elementos de uma tabela

…

# Criando, excluindo e alterando elementos DE uma tabela

....

# Criando sumarizações e agrupamentos (summarize e group\_by)

collapse large datasets into manageable summaries.

# Concatenando dados (cbind, join e merge)

....

# Pivotando dados (wider e longer)

...

# bônus: manipulação avançada

….

# ----- Visualizando os dados (DATAVIZ)

Ggplot: <https://www.linkedin.com/learning/data-visualization-in-r-with-ggplot2/using-the-exercise-files?autoSkip=true&autoplay=true&resume=false>

\* You'll then learn to turn this processed data into informative line plots, bar plots, histograms, and more with the ggplot2 package. You’ll get a taste of the value of exploratory data analysis and the power of Tidyverse tools

# FundamentalsUm mapa mental: tipos de dados e objetivos de visualização

....

# Bibliotecas de visualizações: muitas perspectivas em poucas linhas de código

Indicar o curso da Jessica

dataviz: representamos abstrações de forma mais acessível de encontrar tendências, padrões atípicos e segmentações

# gramática dos gráficos (ggplot): entendendo a estrutura

# exploração vs. ggplot: tipos de gráficos

tipo de variáveis

- categorical and numerical data

objetivos :

e a novidade: número de variáveis

data-to-viz

Vamos passar por alguns aqui.

* boxplot
* heatmap

# explanação vs. ggplot: refinando os outputs

scale

ggplot2tor.com

\* ainda mais representativos.

\* By the end of the course, you'll be able to answer these questions and more, while generating graphics that are both insightful and beautiful.

# bonus: visualizações e recursos interativos: pacotes esquisser e plotly

\* esquisser, ou repositórios como o ggplot2tor.com,

\* plotly::ggplotly()

\* highcharter

\* an interactive graphic would allow you to zoom in on a subset of your data without the need to create a new plot

Colorblind

# bonus: Tabelas

DT

# considerações finais

\* tanto as sumarizações estatísticas quanto as visualizações são visões parciais dos dados, otimizadas segundo algum propósito, implicando, necessariamente, na perda de outras informações: “diferentes níveis de pensamento pedem diferentes níveis de agregação” ou ainda “um gráfico mostra o que aquele gráfico mostra, e nada mais”.

# ----- Preparando os dados (DATAPREP)

Ref.: <https://www.linkedin.com/learning/cleaning-bad-data-in-r/data-is-messy?autoplay=true>

Tentaremos aqui endenreçar mais do que se trata cada opção, por que é importante, um exemplo simples, quais as principais consequencias quando não são bem endereçadas, e principais referencias

To ensure meaningful machine learning results, you must understand your data. Machine learning algorithms are often sensitive to specific characteristics of the data: outliers (data values that are very different from the typical values in your database), irrelevant columns, columns that vary together (such as age and date of birth), data coding, and data that you choose to include or exclude. Oracle Machine Learning can automatically perform much of the data preparation required by the algorithm. But some of the data preparation is typically specific to the domain or the machine learning problem. At any rate, you need to understand the data that was used to build the model to properly interpret the results when the model is applied.

Contadores únicos

# Um mapa mental para o capítulo: o que é dataprep

Por fim, temos a categoria de ajustes, nesta o termo preparação de dados ganha um significado mais literal, visto que aplicamos uma serie de ajustes para garantir a etapa de modelagem. Isto porque a depender do algoritmo utilizado, determinadas características não são permitidas. Tornando necessário ajustes como: o endereçamos dados faltantes, ou a aplicação de codificações para dados categóricos ou ainda a normalização/padronização de resultados numéricos.

Com todos esses cuidados, possibilitamos que as bases estejam no formato necessário para a aplicação dos modelos, afinal, qualquer modelo será tão bom quanto a qualidade dos dados permitir

Tanto que, usualmente, esta etapa concentra uma parte significativa de tempo de um fluxo de trabalho na ciência de dados, e, não à toa esta etapa é conhecida como wrangle, termo em inglês que pode ser traduzido como disputa ou batalha. Mas esta etapa pode também ser encontrada na literatura segundo termos como dataprep, data cleaning, ou data cleansing. Vale comentar que não há receita de bolo aqui, cada um dos ajustes citados deverá ser endereçado considerando a combinação de um olhar crítico em relação aos dados, alinhamentos com pessoas que possuam experiência com as informações em questão, e análises feitas de forma investigativa e interativa, tendo a exploração de dados como um dos principais recursos

\* The time spent cleaning is vital since analyzing dirty data can lead you to draw inaccurate conclusions.

\*It's commonly said that data scientists spend 80% of their time cleaning and manipulating data and only 20% of their time analyzing it.

# Lidando com dados duplicados

\* este ponto em geral diz respeito a sanitização básica dos dados entendimento da unidade observacional

# Tratando dados faltantes e imputação de dados (missing)

\* sua contagem e remoção

\* registros corrompidos e amostras tendenciosas

\* pacote naniar

\* referencia sobre tipos de dados faltantes

# Identificando valores extremos (outliers)

\* unusual data points

\* resultados discrepantes

anomalias ou outliers

Quartis

Boxplot

Dendrograma – citaremos mais na módulo sobre unsupervised

# Aplicando padronizações

\* methods to scale features, including mean centering and z-score standardization.

PArticularmente importante no context em que a técnica é sensível a escala dos dados.

* Pré-processamento
  + Standardization
  + Positional standardization
  + Unitization
  + Positional UnitizationUnitization with zero minimum
  + Normalization in range [-1,1]
  + Positional normalization in range [-1,1]

# Diferentes formas de encoding

\* one-hot encoding and using binning for categorical features.

Dias da semana

# Bônus: feature engineering – trabalhando com datas, textos e fatores

Aqui é importante notar que a etapa Wrangle tem como cerne o desafio da ponte entre o problema que gostaríamos de resolver e os dados disponíveis, sendo preciso, muitas vezes, fazer uma engenharia reversa para entender algumas questões

more efficient

* stringr – pacote rex, que permite saber a expressão regular a partir de textos
* lubridate
* factor

Um exemplo de modelagem que fica melhor com a aplicação disso.

Fator

Questionr addin

[adicionar um capítulo como disclaimer antes da modelagem]

Alguns dos pontos que falaremos aqui serão particularmente relevantes no capítulo de modelagem, visto que algumas técnicas necessitam de palavras sejam categorizados de forma numérica, ou ainda não permitem dados faltantes por exemplo.

# ----- MODELING

# modelagem

...

# Machine Learning (ML), r e o Tidymodels

\* \* tidymodels, caret, mlr

\* data files, preprocessing data, creating models, improving models, and evaluating them to ultimately choose the best model.

\* Aprenda a dividir conjuntos de dados para validação cruzada, pré-processar dados com o pacote de receitas do arrumamodels e ajustar algoritmos de aprendizado de máquina. Você aprenderá os principais conceitos, como definir objetos de modelo e criar fluxos de trabalho de modelagem

\* pode ser segmentando segundo duas visões, sendo elas: o aprendizado supervisionado (supervised learning) e o aprendizado não supervisionado (unsupervised learning).

Tudo machine learning ok são técnicas algoritmos ok tem por objetivo fez identificar padrões em dados ok predizer resultados fez bem fez bem ok bom fez agrupar observ ações, ok isso fez isso fez bem ok bom o que enfim fez bem machine learning trata-se de um guarda-chuva de técnicas que nos ajudam está bom haha evoluirmos o nosso entendimento ok por meio de dados

Machine learning is a technique that discovers previously unknown relationships in data.

Machine learning and AI are often discussed together. An important distinction is that although all machine learning is AI, not all AI is machine learning. Machine learning automatically searches potentially large stores of data to discover patterns and trends that go beyond simple statistical analysis. Machine learning uses sophisticated algorithms that identify patterns in data creating models. Those models can be used to make predictions and forecasts, and categorize data.

The key features of machine learning are:

* Automatic discovery of patterns
* Prediction of likely outcomes
* Creation of actionable information
* Ability to analyze potentially large volumes of data

Machine learning can answer questions that cannot be addressed through traditional deductive query and reporting techniques

It is important to remember that the predictive relationships discovered through machine learning are not *causal* relationships. For example, machine learning might determine that males with incomes between $50,000 and $65,000 who subscribe to certain magazines are likely to buy a given product. You can use this information to help you develop a marketing strategy. However, you must not assume that the population identified through machine learning buys the product *because* they belong to this population.

Machine learning yields probabilities, not exact answers. It is important to keep in mind that rare events *can* happen; they do not happen very often.

Em linhas gerais o objetivo na aprendizagem de máquina é obter regras a partir dos dados disponíveis, de modo que tais regras generalizem os padrões existentes nos dados a ponto de serem aplicadas a novos dados, desconhecidos, e ainda assim trazerem bons resultados, como boas previsões por exemplo.

Organizando a explanação em torno das ideiais gerais, precisamos, primeiramente, refletir sobre os dados que temos disponíveis para análise, que serão utilizados como nossa referência de aprendizado – aqui denominada base/dados de treino. A partir destes dados iremos construir regras, modelos, que irão fornecer conhecimento sobre os padrões existentes nos dados. Para termos uma referência de como o modelo treinado irá performar quando estiver sendo aplicado fora da base de treino, em dados desconhecidos, é usual separar uma parcela desta base para o teste do modelo desenvolvido – sendo estes dados usualmente denominados base/dados de teste.

Este processo de aprendizagem, modelagem, ou treinamento, a partir da perspectiva do Aprendizagem de Máquina Clássica, pode ser segmentando segundo duas visões, sendo elas: o aprendizado supervisionado (supervised learning) e o aprendizado não supervisionado (unsupervised learning). Em ambos os casos temos o que chamamos de atributos descritivos, variáveis, características, features, isto é, informações que serão a nossa referência de aprendizado. E a partir destas características construímos regras, modelos, ajustes, que nos permitirão identificar e estimar padrões em dados não conhecidos.

No caso do aprendizado supervisionado os modelos visam uma característica em específico, de modo que as regras são definidas de modo a identificar esta característica, este rótulo, em novos dados. E por esta razão, a característica de interesse acaba por implicar em uma supervisão durante a construção das regras. Enquanto, no caso não supervisionado, não há interesse em nenhuma característica em específica, e as regras são construídos de uma forma mais ampla, sem nenhum rótulo em específico endereçado a priori.

* **Introduction**: A general introduction into statistical moedlling techniques and the {tidymodels} R package.
* **Quantitaive model fitting**: Using {parsnip} to perform simple and multiple linear regression, allowing us to fit a model.
* **Qualitative model fitting**: Using {parsnip} to perform logistic regression and fit the model.
* **Classification**: Brief introduction to the K-nearest neighbours (KNN) technqiue for cases where a categoric response has more than two possible outcomes.
* **Model assessment**: Testing the reliability of model predictions and finding optimum models using resampling methods like the validation set approach.
* **Model pre-processing with {recipes}**: Including variable trasnformations, one-hot encoding and normalisation

Nos métodos supervisionado temos algoritmos que usam os dados disponíveis zoom em ti denominados dados de treinamento para aprender os padrões e as relas ações que nos ajudam a prevê um determinado input dado. Em outras palavras temas como o input e a informação que desejamos prever como o auth pod os algoritmos de machine supervisionados procuram os padrões de relações que nos ajudam AA partir deste dos inputs obtermos out desejado. das outputs podem ser dados por informações categóricas ou contínuas no caso das categóricas ou classes denominamos algoritmos supervisionados de classificação discute no chamamos de algoritmos supervisionados de regressão. se você já teve contato com a área de modelagem estatística econometria estranhar o termo regressão sendo utilizado apenas para informações contínuas mas é isso mesmo trata-se de uma denominação pouco diferente.

* Classification are discrete and do not imply order.
* Continuous, floating-point values indicate a numerical, rather than a categorical.

No caso dos algoritmos de classificação o tipo mais simples são quando as classes são binárias ou seja quando a tarde te queremos prever tem apenas 2 opções isso como por exemplo: exemplo 1,2 exemplo 3, isso em detrimento de exemplos como alto médio e baixo ou a Netflix que atualmente trabalha com não gostei gostei amei. ok neste curso trabalharemos com classificações binárias e começaremos nossos exemplos com esta.

# ML supervised ~ métodos aplicados para um problema de classificação e regressão

Nossa biblioteca preferida será o tidymodels

Install.package(“tidymodels”)

\* \*logística, random forest

\*regressão linear, random forest

In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown.

Classification models are tested by comparing the predicted values to known target values in a set of test data. The historical data for a classification project is typically divided into two data sets: one for building the model; the other for testing the model.

* **Decision Tree**

Decision trees automatically generate rules, which are conditional statements that reveal the logic used to build the tree.

* + - **Random Forest:** andom Forest is a powerful and popular machine learning algorithm that brings significant performance and scalability benefits. Random Forest is a classification algorithm that builds an **ensemble** (also called **forest**) of trees. The algorithm builds a number of Decision Tree models and predicts using the ensemble. An individual decision tree is built by choosing a random sample from the training data set as the input. At each node of the tree, only a random sample of predictors is chosen for computing the split point. This introduces variation in the data used by the different trees in the forest.
* **Generalized Linear Model**

Generalized Linear Model (GLM) is a popular statistical technique for linear modeling.

* + **logistica**
* **Naive Bayes**

Naive Bayes uses Bayes' Theorem, a formula that calculates a probability by counting the frequency of values and combinations of values in the historical data.

* **Support Vector Machine**

Support Vector Machine (SVM) is a powerful, state-of-the-art algorithm based on linear and nonlinear regression. OML4SQL implements SVM for binary and multiclass classification.

* **XGBoost**

XGBoost is machine learning algorithm for regression and classification that makes available the XGBoost open source package. Oracle Machine Learning for SQL XGBoost prepares training data, invokes XGBoost, builds and persists a model, and applies the model for prediction

# ML unsupervised ~ métodos de clusterização e redução de dimensionalidade

Olá, neste vídeo falaremos um pouco mais sobre modelos de aprendizagem não-supervisionada, aqui, nosso objetivo é entender melhor os padrões dos dados, sem direcionamentos a priori, isto é,

sem rótulos, ou características, pré especificadas. Com isto, podemos aprender os padrões existentes nos dados de maneira mais livre, aprendendo literalmente aprender com os dados. Existem diferentes objetivos possíveis dentro dos modelos não-supervisionados, são eles:

* regras de associação
* mais exemplos
* redução de dimensão
* agrupamentos ou clusterização

Neste vídeo falaremos sobre os dois últimos. Mais especificamente no caso das algoritmos de clusterização, consideraremos os algoritmos hierárquicos e particionados, este último muito conhecido devido ao k-means. O nosso pacote de estimação para aqui é o : <https://github.com/cran/factoextra>

#### Redução de Dimensão

O que é? Trabalha com uma releitura das variáveis, das colunas da base de dados, de modo que

Pq é Importante?

Aplicações?

Ideias básicas

Ou seja precisamos escolher:

...

Trabalhamos com as visualizações:

Scatterplot

Refs.

https://arxiv.org/pdf/1804.02502.pdf

#### Agrupamento / clusterização

O que é? A análise de agrupamento é o ato de separar objetos em grupos, ou, em inglês, separar em clusters, encontrando agrupamentos de objetos semelhantes entre si. De modo que os membros de um mesmo cluster sejam parecidos entre si, ou seja, garantindo uma similaridade intracluster, e os grupos sejam distintos, isto é, uma dissimilaridade entrecluster. Note que no agrupamento, ao contrário da classificação dos modelos supervisionados, os grupos não são definidos a priori.

Por que é importante? Clustering é útil para explorar dados, inclusive performa no top3 ranking de algoritmos de ML para kaggle. Você pode usar algoritmos de agrupamento para encontrar clusters naturais, literalmente aprender com os dados, podendo utilizar na etapa investigativa, para entender a relação entre as muitas características dos dados, para a detecção de anomalias – uma vez que dados atípicos acabam aparecendo como grupos à parte, ou até mesmo como um pré-processamento – como gerando novos rótulos nos dados.

Aplicações: identificação de personas, segmentação de produtos, caracterização de áreas.

Ideias básicas: xxxx

Escolhas -- All the settings have default values:

* Medidas de distancia -- Nestes algoritmos baseados em distância contam com uma função de distância para medir a similaridade entre os casos.
  + Euclidiana -- default
  + Manhattan (city-block)
  + Mahalanobis
  + Chebychev
  + Spearman
  + Kendall
  + Jaquaar
  + Gowen
* Metodos de ligação
  + Single Linkage (nearest neighbor)
  + Complete Linkage (farthest neighbor)
  + Average Linkage
  + Centroid Linkage
  + Ward’s Method (minimum variance)

Dataprep:

* Normalization is typically required by the *k*-Means algorithm.

Ex

Refs.:

**Clustering hierárquico**

Constrói uma hierarquia de partições a partir da similaridade entre as observações.

Escolhas:

* Aglomerativo

Dataviz

* Dendogramas - Interpretação e Identificação de outliers
* Compare dendrograma
* Mapas de calor

**Clustering particionado**

*Abordagens de agrupamento de particionamento, subdividem os dados em um conjunto de k grupos. Um dos métodos de particionamento populares é o agrupamento k-means.*

O algoritmo k-Means é um algoritmo de agrupamento que particiona os dados em um número especificado de agrupamentos. Os casos são atribuídos ao cluster mais próximo de acordo com o método de ligação escolhido. Basicamente, teremos pontos aleatoriamente escolhidos, o centroide, e a cada interação agregamos novas observações, considerando a distancia e o método de ligação escolhido, e a cada agregação conculamos novos centroide. Em linhas gerais o **centroid** represents the most typical case in a cluster. For example, in a data set of customer ages and incomes, the centroid of each cluster would be a customer of average age and average income in that cluster. The centroid does not necessarily describe any given case assigned to the cluster.

Escolhas -- All the settings have default values:

* Number of clusters- Escolha do K, do kmeans -- The term hyperparameter is also interchangeably used for model setting. Atenção para a seed! inicialização e escolha do k

Dataviz

* scatterplot
* Mapas de calor

Refs: colocar um artigo legal

# Inferência e Modelagem Estatística

...

# Bônus: Pontos Importantes

...

# ----- COMUNICAÇÃO

# O que vai encontrar neste módulo

Demanda muito tempo tornar o material, e códigos assustam

exploração vs. explanação

# Um pouco mais sobre relatórios com Rmarkdown

...

# Dashboards interativos com flexdashboard + plotly e Shiny

…

# Considerações gerais

….

# ----- CONCLUSÃO

# bônus | Interoperabilidade no R -- aproveitando a sua experiencia com: SQL, Python, SAS, etc

Existem muitas ferramentas que nos permitem trabalhar com dados, cada uma com seus próprios prós de uso, e comunidade especializada. Como uma forma de garantir que você possa aproveitar o conhecimento de outras comunidades caso o tenha, vamos repassar por alguns pacotes que nos permitem fazer a leitura de formatos externos, ou mesmo incorporando outras linguagens com o R.

# Como trabalhar com códigos em outras linguagens

Pois é, você pode! :) PAra isto você deve seguir o mesmo caminho do point-and-click feito para o chunk R, porém, escolhendo alguma das outras linguagens disponíveis. Assim teremos chunks similares aos criados para o R, porém ao invés de ter o r após as três crases invertidas que iniciam o chunk, teremos `python`, `sql`, e etc.

\* \* Interoperabilidade R & Python (reticulate)

Importante ressaltar que versões antigas do RStudio não dispoem desta opção.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

# QUIZ

Quis – tem perguntas legais para se inspirar:

Ciência de Dados com R - Introdução

Paulo Felipe de Oliveira Saulo Guerra Robert McDonnell

<file:///C:/Users/natyd/Downloads/Ci%C3%AAncia%20de%20dados%20com%20R.pdf>

#Vá em `Tools > Global Options` e personalize o RStudio

# considerando itens como `Appearance` ou `Pane Layout`

# pesquisas uma classe que seja do tipo hora

#unsupt -----

**A imagem abaixo descreve uma distribuição de dados (em verde) que foi representada usando um dendrograma (diagrama de hierarquia em árvore, em azul). Com base nessa imagem, podemos dizer que o dendrograma, em alguns casos, pode ser utilizado para detecção de outliers?**

Reduz distorções de escala no espaço dimensional dos dados;

Remove outliers dos dados;

Torna mais eficiente os cálculos dos algoritmos;

Evita que features com escalas maiores tenham maior relevância nos algoritmos;