Aula 15 – Introdução à Análise com Linguagem R e RStudio

Desvendando Dados com R e RStudio

Sua Jornada no Jornalismo de Dados

Bem-vindo(a) à Aula 15 do Curso de Jornalismo de Dados! Se você chegou até aqui, é porque já percebeu que o mundo está inundado de informações, e a capacidade de extrair sentido delas é uma habilidade de ouro. Hoje, vamos dar um passo fundamental nessa jornada, mergulhando no universo da linguagem R e do RStudio – ferramentas poderosas que transformarão a maneira como você interage com os dados.

Imagine que você tem em mãos uma montanha de documentos, relatórios e planilhas, e precisa encontrar a história escondida ali, aquela que ninguém mais viu. Sem as ferramentas certas, essa tarefa seria exaustiva e, muitas vezes, impossível. É exatamente aqui que R e RStudio entram em cena, não apenas como softwares, mas como seus novos aliados para desvendar padrões, testar hipóteses e, finalmente, contar histórias baseadas em evidências sólidas.

Ao final desta aula, você não apenas entenderá por que a programação é crucial para o jornalista de dados moderno, mas também estará apto(a) a instalar e navegar pela interface do RStudio, executar comandos básicos, importar dados e dar os primeiros passos com pacotes essenciais como readr, dplyr e ggplot2. Prepare-se para uma experiência prática que conectará teoria e aplicação, abrindo portas para um novo nível de análise e reportagem.

Por que Programar para Análise de Dados?

O Poder Além das Planilhas

No cenário atual, onde a informação flui em volumes sem precedentes, o jornalista de dados se depara com um desafio constante: como transformar grandes massas de dados brutos em narrativas claras e impactantes? Muitos de nós começamos nossa jornada com planilhas eletrônicas, e elas são, sem dúvida, ferramentas valiosas para tarefas mais simples. No entanto, quando a complexidade e o volume dos dados aumentam, as planilhas começam a mostrar suas limitações.

Planilhas Tradicionais

- Limitadas a milhares de linhas
- Processos manuais repetitivos
- Propenso a erros humanos
- Difícil reprodutibilidade

Programação com R

- Milhões de registros processados
- Automação completa
- Precisão e consistência
- Análises reproduzíveis

Pense em uma investigação jornalística que envolve milhares de registros de gastos públicos, ou uma análise de tendências em milhões de postagens de redes sociais. Tentar manipular esses dados manualmente em uma planilha seria como tentar esvaziar uma piscina com um copo: demorado, propenso a erros e, no fim das contas, ineficiente. A programação surge como a solução para esse problema, oferecendo a capacidade de automatizar tarefas repetitivas, processar grandes volumes de informação e garantir a reprodutibilidade do seu trabalho.

Superpoder do Jornalista: Aprender a programar para análise de dados é como ganhar um superpoder. Em vez de apenas "ler" os dados, você aprende a "conversar" com eles, a fazer perguntas complexas e a receber respostas precisas e rápidas.

A Era da Automação e IA na Coleta de Dados

O Jornalista como Arquiteto de Informação

O jornalismo de dados não se limita a analisar informações que já estão organizadas em tabelas bonitas. Muitas vezes, a matéria-prima para uma grande história está espalhada pela internet, em documentos PDF, em bancos de dados governamentais ou em APIs (Interfaces de Programação de Aplicativos) de plataformas digitais. Coletar esses dados de forma manual seria uma tarefa hercúlea, consumindo tempo e recursos preciosos.

Web Scraping

Técnicas que permitem "raspar" informações de páginas da web de forma programática, coletando dados em larga escala que seriam impossíveis de copiar e colar manualmente.

APIs

Uso de APIs possibilita a extração direta de dados de serviços online, como redes sociais ou plataformas de dados abertos, de maneira estruturada e eficiente.

Inteligência Artificial

A IA desempenha um papel crucial na identificação de padrões complexos em dados não estruturados, como textos ou imagens, ajudando a encontrar conexões e anomalias.

> Imagine que você está investigando a cobertura de um evento em diferentes veículos de mídia online. Em vez de visitar cada site e copiar manualmente as notícias, você pode usar um script para coletar os títulos, datas e links de milhares de artigos em questão de minutos.

Você se torna não apenas um repórter, mas um arquiteto de informação, construindo sua própria base de dados para contar histórias exclusivas.

Literacia de Dados

Não Apenas Ler, Mas Questionar e Criar Narrativas

Ter acesso a dados e saber programar para manipulá-los é um grande passo, mas não é o destino final. A verdadeira maestria reside na **literacia de dados**, que é a capacidade de ler, entender, criar e comunicar dados como informação. Em um mundo onde gráficos e estatísticas são usados para persuadir – ou, por vezes, enganar – a literacia de dados é sua armadura e sua bússola.

01

Questionar a Fonte

Quem coletou esses dados? Com que propósito? Há conflitos de interesse?

02

Analisar a Metodologia

Como os dados foram coletados? Qual foi o tamanho da amostra? Há vieses na coleta?

03

Identificar Limitações

O que está sendo omitido? Quais são as margens de erro? Há outras interpretações possíveis?

04

Comunicar com Transparência

Como apresentar os dados de forma clara e honesta? Como evitar visualizações enganosas?

> **Analogia Poderosa:** Pense na literacia de dados como a diferença entre ler um livro e ser capaz de escrever um. Ler um livro (consumir dados) é importante, mas escrever um (analisar, interpretar e comunicar dados) exige um nível muito mais profundo de compreensão e criatividade.

Nosso curso é desenhado para construir essa base sólida, capacitando você a não apenas manipular os dados com R, mas a interpretá-los, questioná-los e, finalmente, transformá-los em narrativas jornalísticas poderosas e éticas que informam e engajam o público.

Ética e Transparência

A Bússola do Jornalista de Dados

Com o poder de coletar, analisar e interpretar grandes volumes de dados, vem uma responsabilidade imensa. A ética e a transparência não são apenas conceitos bonitos no jornalismo de dados; são pilares fundamentais que garantem a credibilidade do seu trabalho e protegem o público. Ignorá-los pode levar a consequências graves, desde a disseminação de informações enganosas até a violação da privacidade de indivíduos.

Considerações Éticas

- **Coleta:** Você tem permissão para acessar esses dados?
- **Análise:** Você está evitando vieses e interpretações tendenciosas?
- **Apresentação:** Suas visualizações são claras e não enganosas?
- **Privacidade:** Os dados sensíveis estão protegidos?

Práticas de Transparência

- Compartilhar fontes e metodologia
- Explicar como chegou às conclusões
- Reconhecer limitações dos dados
- Disponibilizar código (quando apropriado)

> Imagine que você está trabalhando com dados sensíveis, como registros de saúde ou informações financeiras de cidadãos. Uma análise descuidada ou a divulgação inadequada desses dados pode causar danos irreparáveis.

Princípio Fundamental: A ética e a transparência são a bússola que guia o jornalista de dados, garantindo que o poder da informação seja usado para o bem público, e não para manipulação ou prejuízo.

Conhecendo o R

O Coração da Análise de Dados

Agora que entendemos o "porquê", vamos ao "o quê". No centro da nossa jornada de análise de dados está a linguagem de programação R. Mas o que exatamente é o R? Pense nele como um motor potente e versátil, especialmente projetado para estatística e gráficos. Ele não é apenas uma ferramenta; é um ambiente completo para computação estatística e visualização, amplamente utilizado por cientistas de dados, estatísticos e, cada vez mais, por jornalistas.

Código Aberto

Gratuito para usar e modificar, com uma comunidade global de desenvolvedores contribuindo constantemente.

Extensível

Milhares de pacotes disponíveis para praticamente qualquer tipo de análise imaginável.

Especializado

Projetado especificamente para estatística, análise de dados e visualização.

Analogia Poderosa: Comparar o R com uma planilha eletrônica é como comparar uma calculadora de bolso com um supercomputador. Enquanto a planilha é excelente para tarefas rápidas e visuais, o R oferece um controle granular sobre cada etapa da análise.

Ele é o coração pulsante por trás de muitas das análises de dados mais sofisticadas e das visualizações mais impactantes que você vê hoje.

RStudio

Seu Painel de Controle para o R

Se o R é o motor potente da sua análise de dados, o RStudio é o painel de controle sofisticado que torna a pilotagem desse motor muito mais fácil e intuitiva. O RStudio não é uma linguagem de programação; é um Ambiente de Desenvolvimento Integrado (IDE) que facilita a escrita de código R, a execução de análises e a visualização de resultados. Pense nele como a cabine de um avião, onde todos os instrumentos e controles estão organizados para que o piloto possa operar a máquina de forma eficiente.

Sem RStudio

Interface de linha de comando básica, intimidadora para iniciantes, menos produtiva para análises complexas.

Com RStudio

Interface gráfica amigável, painéis organizados, fluxo de trabalho otimizado, visualização integrada.

Organização dos Painéis do RStudio

1 **Script Editor (Superior Esquerdo)**

Onde você escreve e salva seu código R. Como um editor de texto otimizado para programação.

2 **Console (Inferior Esquerdo)**

O "cérebro" do R. Comandos são executados aqui e resultados são exibidos.

3 **Environment/History (Superior Direito)**

Mostra objetos criados e histórico de comandos executados.

4 **Files/Plots/Packages/Help (Inferior Direito)**

Painel multifuncional para arquivos, gráficos, pacotes e documentação.

O RStudio transforma a experiência de programar em R de uma tarefa árdua para um processo mais fluido e agradável, permitindo que você se concentre na análise e na descoberta, em vez de lutar com a interface.

Instalação Descomplicada

Preparando Seu Ambiente de Trabalho

Chegou a hora de colocar a mão na massa e preparar seu ambiente de trabalho. A ideia de instalar softwares de programação pode parecer complexa à primeira vista, mas garanto que é um processo bastante direto. Pense nisso como montar sua estação de trabalho: você precisa do motor (o R) e do painel de controle (o RStudio) para começar a operar.

Instalar o R

Primeiro, instalaremos o R. Ele é a base, o software que realmente executa os comandos da linguagem R. Você pode baixá-lo gratuitamente no site oficial do CRAN (Comprehensive R Archive Network).

Escolher a Versão

Escolha a versão compatível com o seu sistema operacional (Windows, macOS ou Linux) e siga as instruções de instalação padrão.

Instalar o RStudio

Depois de instalar o R, é a vez do RStudio. Lembre-se, o RStudio precisa do R já instalado para funcionar, pois ele é apenas uma interface para o R.

Configuração Final

O RStudio Desktop (versão gratuita) pode ser baixado no site oficial. Uma vez instalados, o RStudio detectará automaticamente sua instalação do R.

Processo Simples: É como instalar qualquer outro programa no seu computador: "próximo", "próximo", "finalizar". Parabéns! Você acaba de montar seu laboratório de análise de dados.

Primeiros Passos no RStudio

Explorando a Interface

Ao abrir o RStudio pela primeira vez, você será recebido por uma interface que pode parecer um pouco cheia de informações, mas não se preocupe. Cada painel tem uma função específica e, juntos, eles formam um ambiente de trabalho muito eficiente. Pense na interface do RStudio como a cabine de um carro: cada botão e tela tem um propósito, e uma vez que você entende a função de cada um, dirigir se torna muito mais fácil.

Editor de Script

Localização: Canto Superior Esquerdo

É aqui que você escreverá e salvará seu código R. É como um editor de texto, mas otimizado para programação. Você pode executar linhas de código diretamente daqui.

Console

Localização: Canto Inferior Esquerdo

Este é o "cérebro" do R. Quando você executa um comando, ele aparece aqui, e os resultados são exibidos logo abaixo. É onde o R "conversa" com você.

Ambiente/Histórico

Localização: Canto Superior Direito

O painel "Environment" mostra todos os objetos (variáveis, dados) que você criou. O "History" guarda um registro dos comandos executados.

Arquivos/Plots/Pacotes/Ajuda

Localização: Canto Inferior Direito

Painel multifuncional que exibe arquivos, gráficos gerados, pacotes instalados, documentação de ajuda e visualizações interativas.

Dica de Exploração: Familiarizar-se com esses painéis é o primeiro passo para se sentir confortável no RStudio. Gaste um tempo explorando, clicando nas abas e observando como cada área se comporta. Em breve, você estará navegando por ele com a mesma naturalidade com que usa seu navegador de internet.

Comandos Básicos

Conversando com o R

Com o RStudio aberto e a interface explorada, é hora de dar os primeiros passos na "conversa" com o R. Pense nos comandos básicos como as primeiras palavras que você aprende em um novo idioma. Eles são a base para construir frases mais complexas e expressar ideias mais elaboradas. Não se preocupe em memorizar tudo agora; o importante é entender a lógica.

R como Calculadora

Vamos começar com algo simples: o R pode ser usado como uma calculadora poderosa. No painel do **Console** (canto inferior esquerdo), você pode digitar operações matemáticas:

```
2 + 2        # [1] 4
10 / 3       # [1] 3.333333
sqrt(16)     # [1] 4
```

Trabalhando com Variáveis

Além de cálculos diretos, podemos armazenar valores em **variáveis**. Variáveis são como caixas rotuladas onde você guarda informações:

```
minha_idade <- 30
nome <- "Maria"
saldo_bancario <- 1500.75
```

Operador de Atribuição

Usamos `<-` (ou `=`) para atribuir valores às variáveis

Visualizar Conteúdo

Para ver o conteúdo de uma variável, basta digitar o nome dela e pressionar Enter

Funções Básicas

O R possui muitas funções pré-definidas como `sqrt()`, `print()`, `mean()`

Conceito Fundamental: Uma função é como uma receita: você fornece os ingredientes (argumentos) e ela retorna um resultado. Esses são os blocos de construção mais fundamentais do R.

Pacotes

Expandindo as Capacidades do R

Uma das maiores forças do R reside em sua vasta coleção de **pacotes**. Pense nos pacotes como aplicativos que você instala em seu smartphone: eles adicionam novas funcionalidades e ferramentas que não vêm pré-instaladas com o sistema operacional básico. No caso do R, esses "aplicativos" são coleções de funções, dados e códigos que estendem as capacidades da linguagem para realizar tarefas específicas.

Comunidade Ativa

A comunidade R é incrivelmente ativa, e milhares de pacotes foram desenvolvidos para cobrir praticamente todas as necessidades de análise de dados imagináveis.

Biblioteca Gigantesca

É como ter acesso a uma biblioteca gigantesca de especialistas, cada um com sua área de conhecimento, prontos para ajudar em suas análises.

Ferramentas Prontas

Em vez de programar tudo do zero, você pode aproveitar ferramentas já testadas e aprovadas pela comunidade.

Como Usar Pacotes

1. Instalar

Baixar o pacote para seu computador (apenas uma vez por pacote)

```
install.packages("nome_do_pacote")
```

2. Carregar

Ativar o pacote na sua sessão R atual (a cada nova sessão)

```
library(nome_do_pacote)
```

Exemplo Prático

```
# Instalar o pacote readr (apenas uma vez)
install.packages("readr")

# Carregar o pacote (a cada nova sessão)
library(readr)
```

Poder da Extensibilidade: Essa capacidade de estender o R com pacotes é o que o torna tão poderoso e versátil para o jornalismo de dados.

Importando Dados

Trazendo Suas Histórias para o R

Dados são a matéria-prima do jornalismo de dados. Sem eles, não há história para contar, nem análise para fazer. Portanto, uma das primeiras e mais cruciais habilidades que você precisa desenvolver é a de importar dados para o R. Pense nisso como carregar os ingredientes para a sua cozinha antes de começar a preparar uma refeição.

Formatos Comuns

- CSV (Comma Separated Values)
- Excel (.xlsx, .xls)
- TSV (Tab Separated Values)
- JSON, XML

Funções Tradicionais

- read.csv()
- read.table()
- read.xlsx()

Pacote Moderno: readr

- Mais rápido e eficiente
- Melhor inferência de tipos
- Parte do Tidyverse

Exemplo de Importação com readr

```
# Importar arquivo CSV da pasta de trabalho
dados_jornalismo <- read_csv("meus_dados.csv")

# Importar de caminho completo
dados_jornalismo <- read_csv("C:/Users/SeuUsuario/Documentos/meus_dados.csv")
```

Resultado da Importação: Após a importação, seus dados serão armazenados em um objeto no R, geralmente um tibble (uma versão aprimorada de um data.frame), que você pode visualizar no painel "Environment" do RStudio.

É o primeiro passo para transformar dados brutos em insights.

O Pacote readr

Leitura Eficiente e Confiável

Conforme mencionamos, o pacote readr é a ferramenta preferida para importar dados no R, especialmente quando se trata de arquivos tabulares como CSVs e TSVs. Mas por que ele é tão elogiado em comparação com as funções de importação "base" do R? A resposta está na sua eficiência, robustez e na maneira como ele lida com os dados.

Velocidade Superior

Para conjuntos de dados muito grandes, o readr pode ser significativamente mais rápido que suas contrapartes base do R.

Inferência Inteligente

Tenta adivinhar automaticamente se uma coluna contém números, texto, datas, etc., e geralmente faz um trabalho excelente.

Consistência

As funções têm sintaxe similar (read_csv(), read_tsv(), read_delim()), facilitando o aprendizado.

Tibbles Modernos

Produz tibbles, uma versão moderna do data.frame com melhor impressão no console e comportamento mais previsível.

Exemplo Prático com Inferência de Tipos

```
# Certifique-se de ter o pacote readr carregado
library(readr)

# Criando um arquivo CSV de exemplo
dados_exemplo <- "nome,idade,cidade,data_registro
Alice,25,São Paulo,2023-01-15
Bruno,30,Rio de Janeiro,2023-02-20
Carla,28,Belo Horizonte,2023-03-10"

write_file(dados_exemplo, "dados_exemplo.csv")

# Importando o arquivo CSV
meus_dados <- read_csv("dados_exemplo.csv")

# Visualizando os dados e seus tipos
print(meus_dados)
```

Resultado da Inferência

nome	idade	cidade	data_registro
<chr>	<dbl>	<chr>	<date>
Alice	25	São Paulo	2023-01-15
Bruno	30	Rio de Janeiro	2023-02-20
Carla	28	Belo Horizonte	2023-03-10

Inteligência Automática: Note como read_csv() identificou nome e cidade como caracteres (<chr>), idade como número (<dbl>) e data_registro como data (<date>). Essa inteligência é o que torna o readr uma ferramenta indispensável.

Introdução ao Tidyverse

O Ecossistema da Análise Limpa

Depois de importar seus dados, o próximo passo é organizá-los e prepará-los para a análise. É aqui que entra o **Tidyverse**, um conceito que revolucionou a forma como muitos cientistas de dados trabalham com R. O Tidyverse não é apenas um pacote; é uma coleção de pacotes interligados, desenvolvidos com uma filosofia comum de design, que visa tornar a manipulação, exploração e visualização de dados mais intuitiva, consistente e eficiente.

dplyr

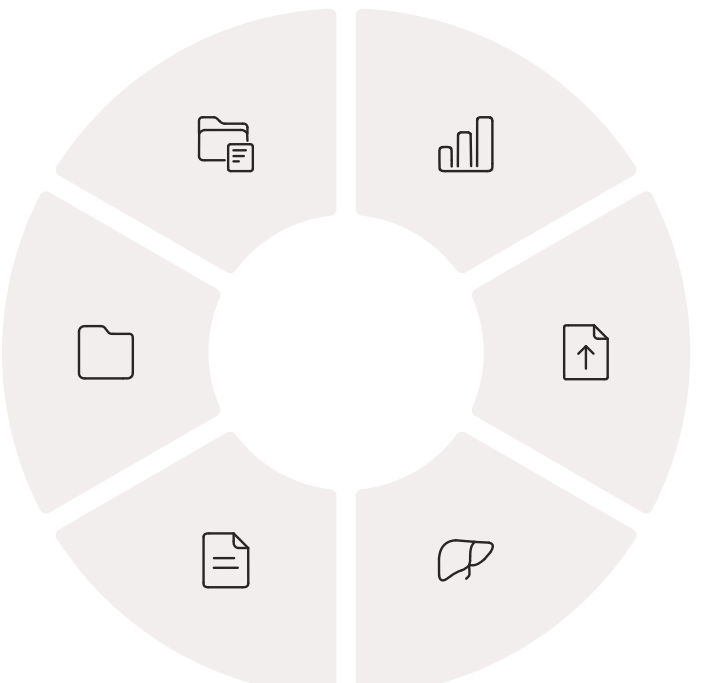
Manipulação de dados (filtrar, selecionar, transformar)

forcats

Trabalhar com fatores (variáveis categóricas)

stringr

Manipulação de texto



ggplot2

Visualização de dados (criar gráficos)

readr

Importação de dados

tidyr

Arrumar dados (transformar formatos)

Filosofia Unificada

> Pense no Tidyverse como uma caixa de ferramentas completa e bem organizada, onde cada ferramenta tem um propósito claro e todas elas funcionam em harmonia. Em vez de ter que aprender várias maneiras diferentes de fazer a mesma coisa, o Tidyverse oferece uma "gramática" unificada para a ciência de dados.

Instalação e Carregamento

```
# Instala todos os pacotes do Tidyverse
install.packages("tidyverse")

# Carrega os pacotes principais do Tidyverse
library(tidyverse)
```

Filosofia de Trabalho: Ao adotar o Tidyverse, você não está apenas aprendendo a usar um conjunto de ferramentas; você está adotando uma filosofia de trabalho que promove dados "arrumados" (tidy data), facilitando a análise e comunicação de resultados.

dplyr

Manipulando Seus Dados com Elegância

Com seus dados importados e o Tidyverse carregado, é hora de começar a moldá-los. O pacote **dplyr** é o cavalo de batalha do Tidyverse para a manipulação de dados. Ele oferece um conjunto de funções verbais e intuitivas que permitem realizar as operações mais comuns em conjuntos de dados de forma clara e eficiente. Pense no dplyr como um escultor que transforma um bloco de mármore bruto (seus dados) em uma obra de arte (os dados prontos para análise).

Os Cinco Verbos Principais

filter()

Seleciona linhas com base em condições. Quer ver apenas os dados de 2023? Use filter().

select()

Seleciona colunas (variáveis). Precisa apenas do nome e da idade? Use select().

mutate()

Cria novas colunas ou modifica as existentes. Quer calcular a idade em meses? Use mutate().

arrange()

Ordena as linhas por uma ou mais colunas. Quer ver do mais novo para o mais velho? Use arrange().

summarize() + group_by()

Calcula estatísticas resumidas para grupos de dados. Quer a média de idade por cidade? Use group_by() e summarize().

O Poder do Operador Pipe (%>%)

A beleza do dplyr está na sua sintaxe consistente e na capacidade de encadear operações usando o operador `%>%` (pipe), que significa "e então":

```
# Exemplo básico com tibble fictício
library(dplyr)

dados_ficticios <- tibble(
  nome = c("Ana", "Beto", "Carla", "Davi", "Eva"),
  idade = c(22, 35, 28, 40, 25),
  cidade = c("SP", "RJ", "SP", "MG", "RJ"),
  salario = c(3000, 5000, 3200, 6000, 3100)
)

# Filtrar pessoas de SP com mais de 25 anos
dados_filtrados <- dados_ficticios %>%
  filter(cidade == "SP", idade > 25)

print(dados_filtrados)
```

Leitura Natural: Com dplyr, você transforma seus dados de forma expressiva: "Pegue os dados_ficticios, **filtre** onde cidade é 'SP' E idade > 25". A leitura se torna quase uma frase em português!

dplyr em Ação

Um Exemplo Prático de Filtragem e Seleção

Para solidificar o entendimento do dplyr, vamos aplicar alguns de seus "verbos" em um cenário mais concreto. Imagine que você, como jornalista de dados, recebeu um conjunto de dados sobre reclamações de consumidores em diferentes estados e categorias. Sua tarefa é encontrar as reclamações mais recentes de um estado específico e focar apenas nas informações mais relevantes.

Criando o Dataset de Exemplo

```
library(dplyr)
library(readr)

# Criando um tibble de exemplo de reclamações
reclamacoes <- tibble(
  id_reclamacao = 101:108,
  data_registro = as.Date(c("2024-01-10", "2024-02-15", "2024-01-20",
                            "2024-03-01", "2024-02-28", "2024-01-05",
                            "2024-03-15", "2024-02-01")),
  estado = c("SP", "RJ", "MG", "SP", "RJ", "SP", "MG", "RJ"),
  categoria = c("Telefonia", "Bancos", "Energia", "Telefonia",
                "Internet", "Bancos", "Internet", "Telefonia"),
  status = c("Resolvida", "Pendente", "Resolvida", "Pendente",
             "Resolvida", "Pendente", "Pendente", "Resolvida"),
  descricao_curta = c("Problema com fatura", "Demora no atendimento",
                      "Aumento indevido", "Serviço intermitente",
                      "Conexão lenta", "Cobrança duplicada",
                      "Instalação atrasada", "Cancelamento não efetuado")
)
```

Aplicando Filtros e Seleções

Filtrar

Reclamações do estado "SP" com status "Pendente"

Selecionar

Apenas colunas relevantes: data, categoria e descrição

```
# Filtrando e selecionando dados
reclamacoes_sp_pendentes <- reclamacoes %>%
  filter(estado == "SP", status == "Pendente") %>%
  select(data_registro, categoria, descricao_curta)

print(reclamacoes_sp_pendentes)
```

Resultado

data_registro	categoria	descricao_curta
2024-03-01	Telefonia	Serviço intermitente
2024-01-05	Bancos	Cobrança duplicada

Fluidez Natural: Percebe como a leitura do código se torna quase uma frase em português? "Pegue as reclamacoes, **filtre** onde o estado é 'SP' E o status é 'Pendente', **e então selecione** as colunas data_registro, categoria e descricao_curta." Essa fluidez é a essência do dplyr e do Tidyverse.

ggplot2

Visualizando Suas Descobertas com Arte

Depois de manipular e organizar seus dados com dplyr, o próximo passo crucial é visualizá-los. Um bom gráfico pode comunicar uma história complexa de forma instantânea e impactante, algo que tabelas de números raramente conseguem. É aqui que o pacote **ggplot2** entra em cena, uma das ferramentas mais poderosas e elegantes para criar visualizações de dados no R.

Grammar of Graphics

O ggplot2 é baseado na "Grammar of Graphics" (Gramática dos Gráficos), uma filosofia que decompõe um gráfico em componentes lógicos:

Dados

As informações que você quer visualizar

Mapeamentos Estéticos

Como as variáveis se traduzem em elementos visuais (posição, cor, tamanho)

Geometrias

O tipo de gráfico (pontos, linhas, barras)

Estrutura Básica

A estrutura básica de um gráfico ggplot2 sempre começa com `ggplot()` e adiciona camadas usando o operador `+`:

```
ggplot(data = seus_dados, aes(x = sua_variavel_x, y = sua_variavel_y)) +
  geom_tipo_de_grafico()
```

01

ggplot()

Inicializa o gráfico e especifica os dados

02

aes() (aesthetics)

Mapeia variáveis para características visuais (eixos, cor, tamanho, forma)

03

geom_tipo_de_grafico()

Adiciona camada geométrica (geom_point(), geom_bar(), geom_line())

Construção em Camadas: Pense no ggplot2 como um conjunto de peças de Lego: você combina diferentes blocos (camadas) para construir o gráfico desejado, com total controle sobre cada detalhe. Essa abordagem permite construir gráficos complexos de forma incremental.

ggplot2 em Ação

Construindo Seu Primeiro Gráfico

Vamos aplicar a "Grammar of Graphics" para criar um gráfico simples, mas informativo. Usaremos os dados fictícios de reclamações que criamos anteriormente e o ggplot2 para visualizar a distribuição de reclamações por categoria. Isso nos dará uma ideia rápida de quais categorias geram mais problemas.

Preparando os Dados

```
library(ggplot2)
library(dplyr)

# Reutilizando o tibble de reclamações
reclamacoes <- tibble(
  id_reclamacao = 101:108,
  data_registro = as.Date(c("2024-01-10", "2024-02-15", "2024-01-20",
                            "2024-03-01", "2024-02-28", "2024-01-05",
                            "2024-03-15", "2024-02-01")),
  estado = c("SP", "RJ", "MG", "SP", "RJ", "SP", "MG", "RJ"),
  categoria = c("Telefonia", "Bancos", "Energia", "Telefonia",
                "Internet", "Bancos", "Internet", "Telefonia"),
  status = c("Resolvida", "Pendente", "Resolvida", "Pendente",
             "Resolvida", "Pendente", "Pendente", "Resolvida"),
  descricao_curta = c("Problema com fatura", "Demora no atendimento",
                      "Aumento indevido", "Serviço intermitente",
                      "Conexão lenta", "Cobrança duplicada",
                      "Instalação atrasada", "Cancelamento não efetuado")
)
```

Criando o Gráfico de Barras

```
# Criando um gráfico de barras para contar reclamações por categoria
grafico_categorias <- ggplot(data = reclamacoes, aes(x = categoria)) +
  geom_bar(fill = "steelblue") +  # Adiciona barras em azul
  labs(title = "Número de Reclamações por Categoria",
       x = "Categoria da Reclamação",
       y = "Contagem") +
  theme_minimal()  # Tema visual limpo

# Para exibir o gráfico
print(grafico_categorias)
```

Anatomia do Código

ggplot()

Inicializa o gráfico com nossos dados e mapeia 'categoria' para o eixo X

geom_bar()

Adiciona as barras, que automaticamente contam a frequência de cada categoria

labs()

Adiciona títulos e rótulos para tornar o gráfico mais informativo

theme_minimal()

Aplica um estilo visual limpo e moderno

Transformação Instantânea: Em poucas linhas, transformamos dados brutos em uma visualização compreensível! O ggplot2 automaticamente conta as ocorrências de cada categoria e cria as barras proporcionais.

Conectando os Pontos

Tidyverse para um Fluxo de Trabalho Integrado

Até agora, exploramos o readr para importar dados, o dplyr para manipulá-los e o ggplot2 para visualizá-los. A verdadeira magia do Tidyverse, no entanto, reside em como esses pacotes trabalham juntos de forma coesa, permitindo um fluxo de trabalho de análise de dados suave e integrado. Pense nisso como uma linha de montagem bem azeitada, onde cada estação (pacote) realiza uma tarefa específica e passa o resultado para a próxima, sem interrupções.

O Segredo: Operador Pipe (%>%)

O segredo para essa integração é o operador **pipe (%>%)**, que permite pegar o resultado de uma função e "alimentá-lo" diretamente como entrada para a próxima função. Isso torna seu código incrivelmente legível e evita a criação de muitas variáveis intermediárias.

Exemplo Completo: Da Importação à Visualização

```
library(tidyverse)  # Carrega readr, dplyr, ggplot2 e outros

# Recriando o dataset de reclamações
reclamacoes <- tibble(
  id_reclamacao = 101:108,
  data_registro = as.Date(c("2024-01-10", "2024-02-15", "2024-01-20",
                            "2024-03-01", "2024-02-28", "2024-01-05",
                            "2024-03-15", "2024-02-01")),
  estado = c("SP", "RJ", "MG", "SP", "RJ", "SP", "MG", "RJ"),
  categoria = c("Telefonia", "Bancos", "Energia", "Telefonia",
                "Internet", "Bancos", "Internet", "Telefonia"),
  status = c("Resolvida", "Pendente", "Resolvida", "Pendente",
             "Resolvida", "Pendente", "Pendente", "Resolvida"),
  descricao_curta = c("Problema com fatura", "Demora no atendimento",
                      "Aumento indevido", "Serviço intermitente",
                      "Conexão lenta", "Cobrança duplicada",
                      "Instalação atrasada", "Cancelamento não efetuado")
)

# Fluxo integrado: filtrar → agrupar → contar → visualizar
grafico_pendentes_por_estado <- reclamacoes %>%
  filter(status == "Pendente") %>%          # Filtra apenas as pendentes
  group_by(estado) %>%                      # Agrupa por estado
  summarize(total_pendente = n()) %>%       # Conta por estado
  ggplot(aes(x = estado, y = total_pendente, fill = estado)) +  # Inicia gráfico
  geom_col() +                              # Adiciona colunas
  labs(title = "Reclamações Pendentes por Estado",
       x = "Estado",
       y = "Total de Reclamações Pendentes") +
  theme_minimal()

print(grafico_pendentes_por_estado)
```

Filtrar

Apenas reclamações pendentes

Agrupar

Por estado

Contar

Total por grupo

Visualizar

Gráfico de colunas

Fluxo Natural: Neste exemplo, começamos com os dados brutos, filtramos as reclamações pendentes, agrupamos por estado para contar quantas há em cada um, e então passamos o resultado diretamente para o ggplot2. Essa é a essência do fluxo de trabalho Tidyverse: uma sequência lógica de transformações que leva você da pergunta aos dados e, finalmente, à visualização da resposta.

Consolidação e Próximos Passos

Sua Jornada no Jornalismo de Dados

Chegamos ao fim da nossa introdução à análise de dados com R e RStudio. Percorremos um caminho que começou com a compreensão do "porquê" programar para o jornalismo de dados, passando pela importância da literacia e ética, até a instalação das ferramentas e os primeiros passos práticos. Você aprendeu que R é o motor estatístico e RStudio é seu painel de controle, e que pacotes como readr, dplyr e ggplot2 são seus aliados para importar, manipular e visualizar dados de forma eficiente e elegante.

4	5	3
Ferramentas Principais	**Verbos do dplyr**	**Componentes ggplot2**
R, RStudio, readr, dplyr, ggplot2 e Tidyverse	filter(), select(), mutate(), arrange(), summarize()	Dados, mapeamentos estéticos e geometrias

Recapitulação dos Conceitos-Chave

Fundamentos

- Por que programar para jornalismo
- Importância da literacia de dados
- Ética e transparência

Ferramentas

- R como motor estatístico
- RStudio como interface
- Tidyverse como ecossistema

Prática

- Importação com readr
- Manipulação com dplyr
- Visualização com ggplot2

Em Prática - Seus Próximos Passos

01

Baixe Dados Reais

Procure conjuntos de dados públicos (ex: dados abertos governamentais)

02

Importe para o RStudio

Use read_csv() para carregar os dados

03

Explore com dplyr

Use filter() e select() para explorar subconjuntos

04

Visualize com ggplot2

Crie um gráfico simples para visualizar uma variável

Lembre-se: A capacidade de transformar dados brutos em histórias significativas é uma habilidade cada vez mais valorizada. A prática leva à perfeição - continue experimentando com os comandos, buscando novos conjuntos de dados e tentando replicar análises que você vê em reportagens.

Autoavaliação

Teste Seus Conhecimentos

Questões de Múltipla Escolha

1

Vantagem do R e RStudio

Qual das seguintes opções melhor descreve a principal vantagem de usar R e RStudio para análise de dados em comparação com planilhas eletrônicas para grandes volumes de dados?

- a) A interface gráfica do RStudio é mais colorida
- b) R e RStudio são mais rápidos para cálculos simples
- **c) Permitem automação, reprodutibilidade e processamento de grandes volumes de dados**
- d) Planilhas eletrônicas não conseguem realizar nenhuma análise estatística

2

Pacote para Manipulação

Qual pacote do Tidyverse é primariamente utilizado para a manipulação e transformação de dados, como filtrar linhas ou selecionar colunas?

- a) readr
- b) ggplot2
- **c) dplyr**
- d) stringr

3

Carregamento de Pacotes

Para carregar um pacote na sua sessão R atual, após ele já ter sido instalado, qual função você deve usar?

- a) install.packages()
- b) load.package()
- **c) library()**
- d) require_package()

4

Grammar of Graphics

A "Grammar of Graphics", base do ggplot2, decompõe um gráfico em quais componentes lógicos principais?

- a) Título, legenda e cores
- **b) Dados, mapeamentos estéticos e geometrias**
- c) Eixos X, Y e Z
- d) Tabelas, listas e texto

Questão Dissertativa

Questão 5

Explique brevemente por que a literacia de dados e a ética são cruciais para o jornalista de dados, especialmente no contexto de automação e IA.

(Resposta esperada: 3-5 linhas)

Gabarito

Respostas e Explicações

Respostas das Questões Objetivas

Questão 1

Resposta: c)

R e RStudio permitem automação, reprodutibilidade e processamento de grandes volumes de dados, superando as limitações das planilhas.

Questão 2

Resposta: c)

O dplyr é o pacote especializado em manipulação de dados com seus cinco verbos principais.

Questão 3

Resposta: c)

A função library() é usada para carregar pacotes já instalados na sessão atual do R.

Questão 4

Resposta: b)

A Grammar of Graphics decompõe gráficos em dados, mapeamentos estéticos e geometrias.

Resposta da Questão Dissertativa

Questão 5 - Resposta Modelo

A literacia de dados permite ao jornalista interpretar, questionar e comunicar dados de forma crítica, evitando a desinformação. A ética, por sua vez, garante que a coleta, análise e divulgação de dados sejam feitas com responsabilidade, protegendo a privacidade e evitando vieses, o que é ainda mais vital com a capacidade de processar grandes volumes de dados via automação e IA.

Pontos-Chave da Resposta: A resposta deve abordar tanto a literacia (capacidade crítica de interpretar dados) quanto a ética (responsabilidade na coleta e divulgação), especialmente considerando o poder ampliado da automação e IA no processamento de grandes volumes de informação.

Recursos e Próximos Passos

Continue Sua Jornada

Próxima Aula

Aula 16

Análise de Dados com R (Parte 1)

Na próxima aula, aprofundaremos nas técnicas de análise de dados com R, explorando mais funções do dplyr e tidyr para preparar seus dados para análises mais complexas.

Recursos Adicionais

R for Data Science

Autor: Hadley Wickham

Livro online gratuito e essencial para o Tidyverse. Considerado a bíblia do R moderno para ciência de dados.

RStudio Cheatsheets

Tipo: Guias Rápidos Visuais

Guias rápidos visuais para dplyr, ggplot2 e outros pacotes. Perfeitos para consulta rápida durante o trabalho.

Comunidade R Brasil

Plataforma: Fóruns e Grupos

Fóruns e grupos para tirar dúvidas e compartilhar conhecimento com outros usuários brasileiros do R.

Links Úteis

Documentação Oficial

- CRAN R Project
- RStudio Documentation
- Tidyverse.org

Comunidades

- R-Ladies Global
- Stack Overflow (tag: r)
- Reddit r/rstats

NOTA IMPORTANTE: As informações regulatórias/legais/técnicas desta aula estão atualizadas até 2025. Consulte sempre fontes oficiais para verificar alterações.

> Parabéns por completar esta introdução ao R e RStudio! Você deu o primeiro passo em uma jornada transformadora no jornalismo de dados. Continue praticando, explorando e, principalmente, questionando os dados que encontrar pelo caminho.