

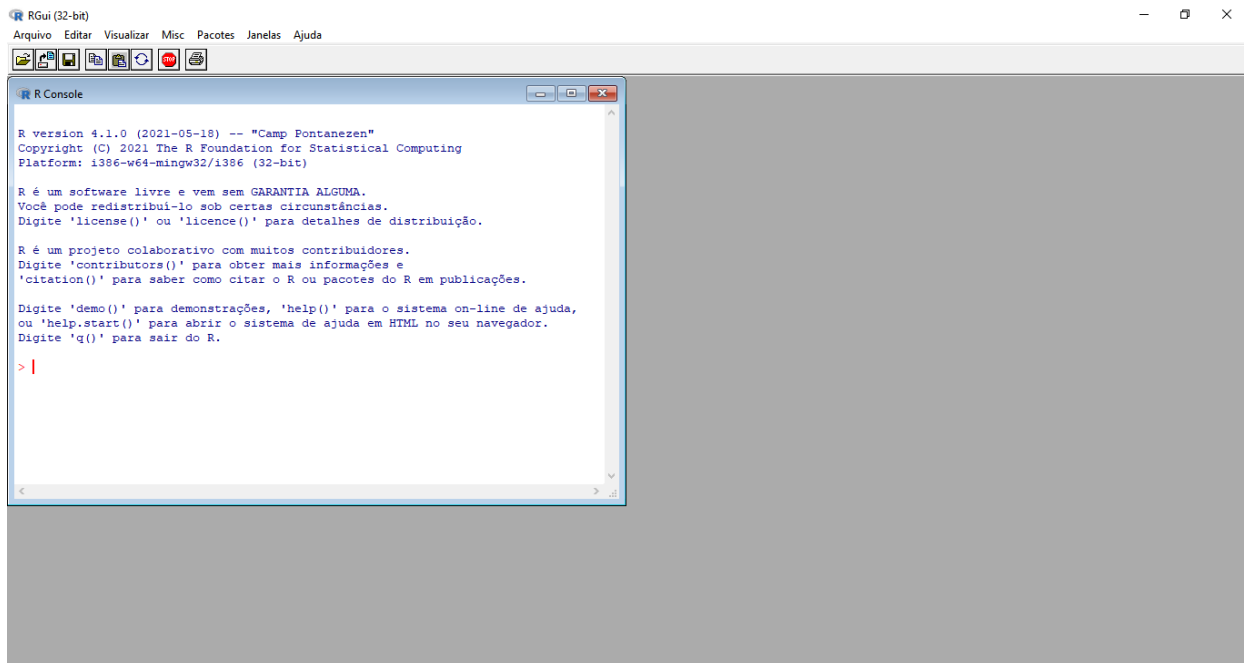
Aula 1 - Entendendo a plataforma R e sua linguagem

Eduardo Koerich Nery

Nesta aula iremos explorar aspectos gerais da plataforma e da linguagem R. O texto e as imagens da aula estão baseado na plataforma R base, logo você pode encontrar diferenças caso esteja usando a interface gráfica do RStudio. As diferenças entre interfaces gráficas serão abordadas durante a aula. Dado isto, vamos lá!

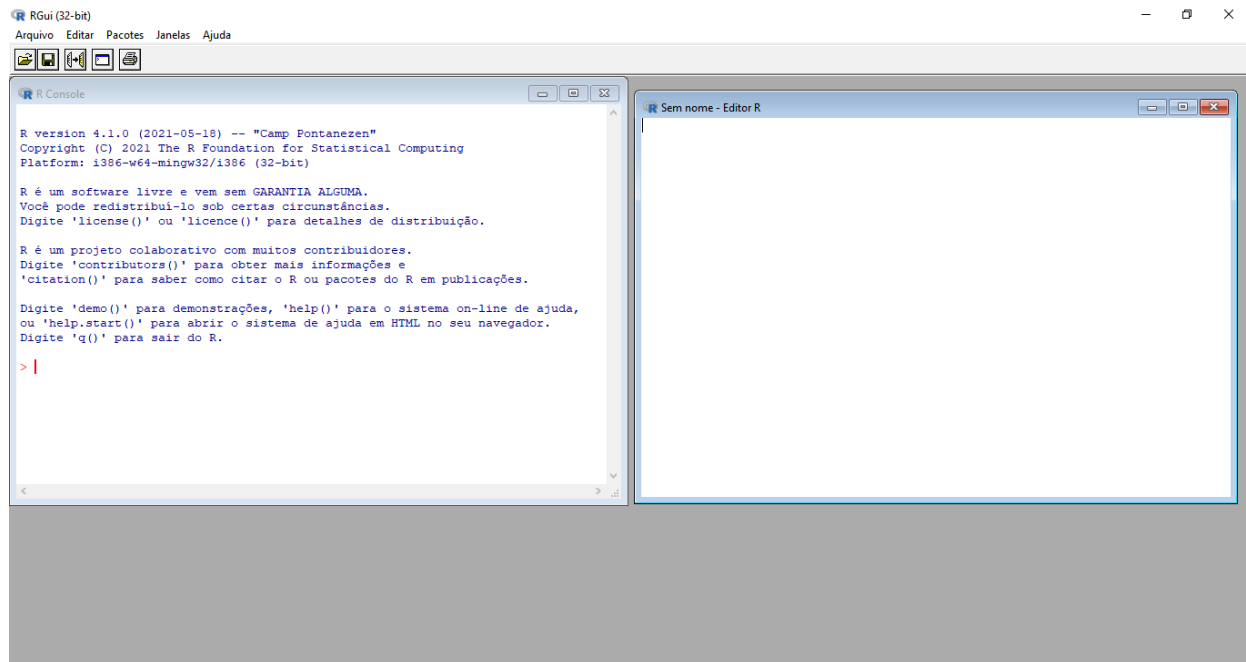
Console e Painel do Editor

Console Ao iniciar a plataforma R, você verá uma única janela (ou três janelas, no caso do RStudio). A janela que possui um sinal de `>` é o **console**, ou interpretador de linha de comando. Na janela do console, você escreve comandos para o computador executar, e o computador mostrará o resultado dos comandos. O sinal `>` mostra que o computador está pronto para receber comandos.

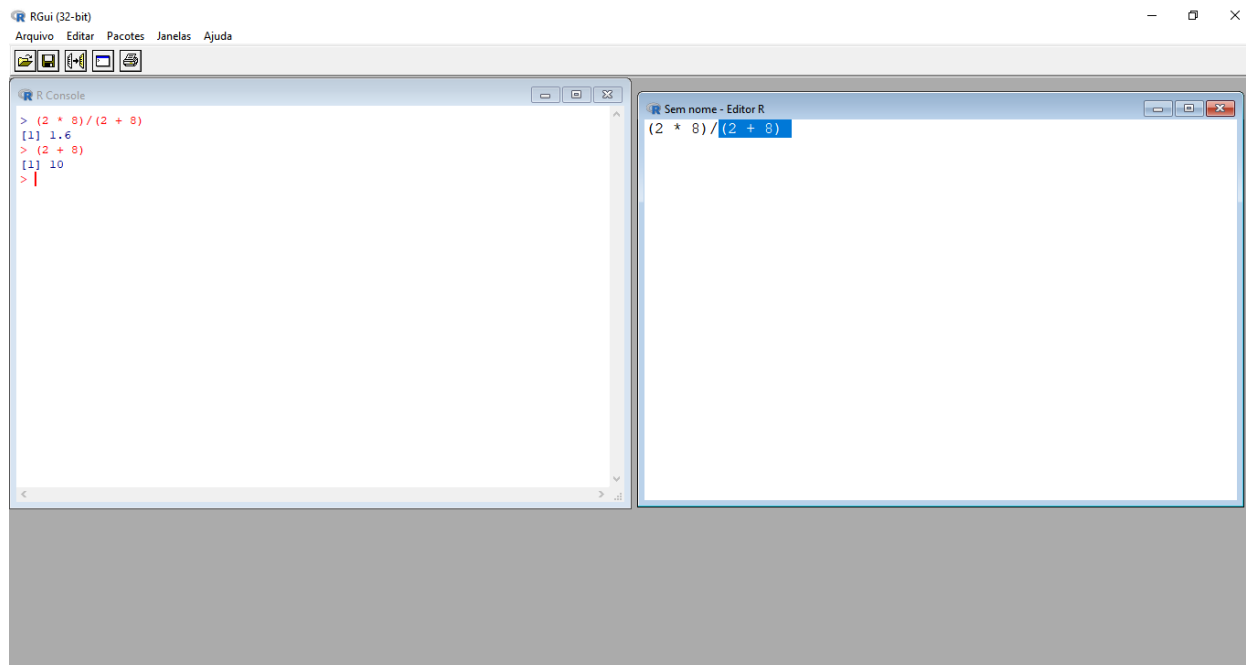


Escreva no console o seguinte comando $2 + 8$ e aperte a tecla “Enter”. Em seguida, execute esses comandos $2 * 8$, $2 - 8$ e $2 / 8$. O console tem memória dos comandos executados, e você pode voltar em comandos passados com as teclas “Acima” e “Abaixo” do seu teclado.

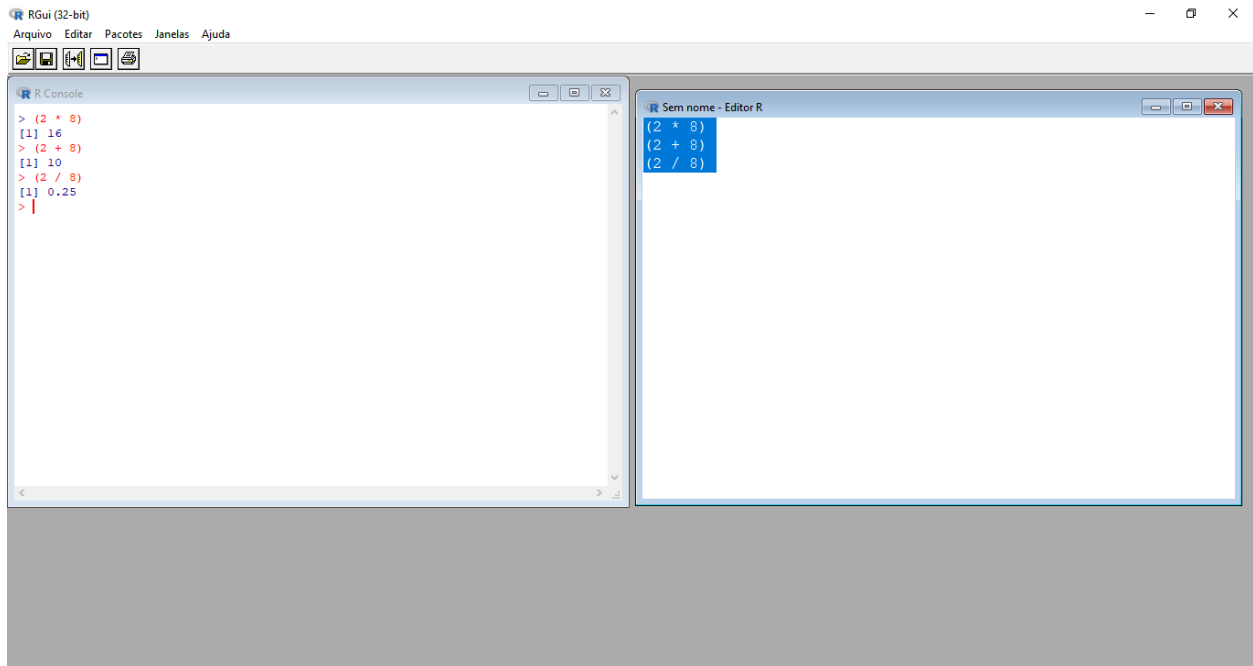
Painel do Editor Normalmente usamos o console para comandos pequenos e rápidos, mas normalmente precisamos executar múltiplos comandos no nosso trabalho. Para organizar e armazenar um conjunto de comandos, nós usaremos o **painel do editor**, ou *script*. Clique na aba “Arquivo”; depois “Novo script” (na interface RStudio: clique na aba “File”; “New File”; “R script”).



Agora, escreva o seguinte código no painel do editor $(2 * 8)/(2 + 8)$ e aperte “Ctrl + R” (no RStudio: “Ctrl + Enter”). Você acabou de executar essa linha de código. Você também pode selecionar uma parte da linha de código para executá-la separadamente com o mesmo comando.



Você pode executar múltiplas linhas de código ao selecioná-las e apertar “Ctrl + R” (no RStudio: “Ctrl + Enter”). **As linhas de códigos são sempre executadas na ordem em que são apresentadas.** Veremos que isso é a base para executarmos uma programação mais complexa.



Por fim, você pode adicionar comentários que explicam as suas linha de código, sem que eles afetem os comandos. Adicione o sinal `#` ao lado ou acima do código e descreva o que aquele código faz. Tente o seguinte `(2 * 8)/(2 + 8) # executa essa continha`. Comentários são muito úteis: ajudam a lembrar o propósito dos comandos dentro do nosso programa; organizam nossa linha de raciocínio; explicam nosso programa a um outro usuário. Portanto, adquira o hábito de comentar o seu código, sempre!

Diretório de trabalho

O R sempre vai designar uma pasta do seu computador para importar e exportar arquivos, o **diretório de trabalho**. Execute o comando `getwd()` para descobrir qual é o seu diretório de trabalho. Na interface do RStudio, o diretório é visualizado numa das janelas, sob a aba “Files”. Você pode designar o diretório de trabalho de duas formas:

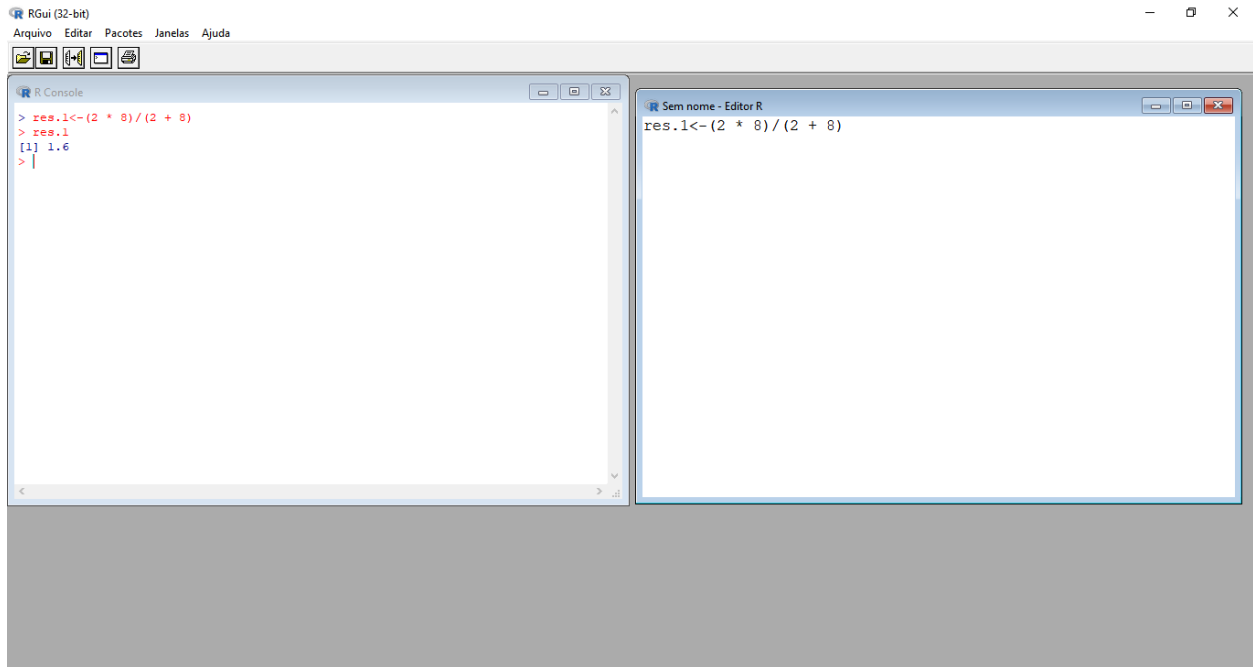
- 1) Por linha de comando. Execute a função `setwd()`, inserindo dentro dos parênteses o endereço da pasta que você quer como diretório, sendo que este endereço deve estar entre aspas `"`. Por exemplo, no meu computador, eu executei `setwd("C:/Users/eduar/Desktop/Minicurso")`.
- 2) Manualmente. No R base: clique na aba “Arquivo”; opção “Mudar Dir...”. No RStudio: clique em “Session”; opção “Set Working Directory”; opção “Choose Directory...”;

É uma boa prática de programação designar um diretório diferente para cada projeto/programa que você desenvolve. Crie uma pasta “Minicurso” em seu computador para este minicurso, e designe essa pasta como diretório de trabalho.

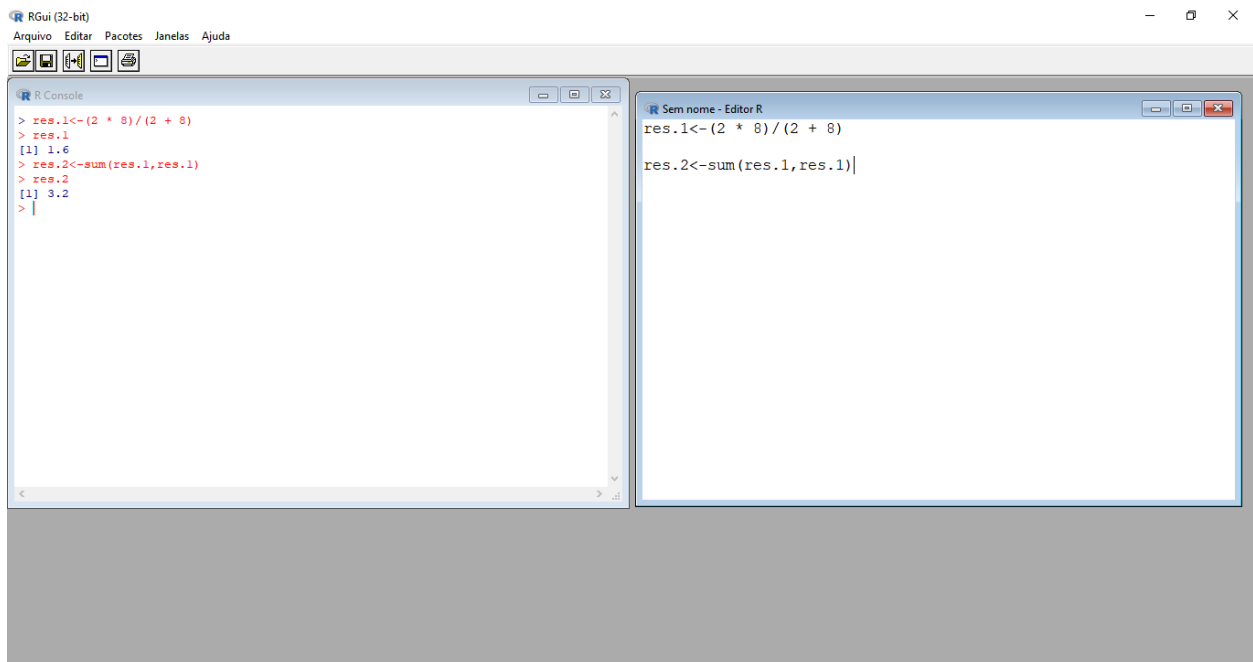
A linguagem R

O R é uma **linguagem orientada a objetos**. Isto quer dizer que os programas feitos precisam de unidades com propriedades definidas (objetos) e de relações entre essas unidades (funções).

Objetos Você deve utilizar os símbolos `=` ou `<-` para criar um objeto no R. Por exemplo, execute o comando `res.1 <- (2 * 8)/(2 + 8)`. Você criou um objeto chamado `res.1`, dentro do qual está armazenado o resultado da operação matemática $(2 * 8)/(2 + 8)$. Para acessar o conteúdo de um objeto, basta executar o nome desse objeto como se fosse um comando. Execute o objeto `res.1` no console ou no painel do editor.



Funções As funções estabelecem relações entre os objetos. A estrutura geral das funções é `função.genérica(x)`, onde `x` é chamado de “argumento da função”. Uma função pode ter um a muitos argumentos. Por exemplo, a função `sum()` soma os valores dos seus argumentos. Execute o comando `res.2<-sum(res.1,res.1)` e depois o objeto `res.2`.



IMPORTANTE: Você utilizará muitas funções desconhecidas ao longo deste minicurso e no seu trabalho.

Toda vez que tiver dúvidas sobre como uma função funciona, execute o comando de ajuda que é o operador `?` antes do nome da função. Por exemplo, execute o comando `?sum` para entender mais sobre a função `sum`. Isto abrirá uma página com explicações e exemplos sobre o uso da função.

Finalizando seu trabalho

Você terá investido um bom tempo do seu dia desenvolvendo o código do seu programa ou projeto, e com certeza não vai querer começar tudo do zero no outro dia. Assim você provavelmente vai querer salvar os comandos que usou e, em alguns casos, os objetos que você criou.

Para salvar os comandos usados, selecione a janela do **painel do editor**; clique em “Arquivo”; clique em “Salvar Como”. Os comandos escritos são armazenados em um arquivo com extensão `.R`. Este arquivo, o *script*, pode ser lido como bloco de notas. Para salvar os objetos criados, execute a função `save.image()` no seu console. Isto criará um arquivo com extensão `.RData` no seu diretório de trabalho. Caso queira continuar a trabalhar sobre os mesmos objetos, basta carregar o arquivo `.RData` criado em uma outra sessão do R.