| Bem vindo ao swirl! Por favor, faça o login. Se você já esteve aqui antes, use o mesmo nome que você usou anteriormente. Se você é novo aqui, identifique-se com um nome original.

Como devo chamá-l@? Diego

| Obrigado, Diego. Permita-me explicar algumas coisinhas antes de começarmos nossa primeira lição. Antes de tudo, você precisa saber que quando você vir '...', isso significa que você deve pressionar Enter quando você

| terminar de ler e estiver pronto para continuar.

... <-- Esse é o sinal de que você deve pressionar Enter para continuar

| Da mesma forma, quando você vir 'ANSWER:', o prompt do R (>), ou quando lhe for pedido para selecionar itens de uma lista, isso significa que é a sua vez de dar uma resposta, e depois pressionar Enter para continuar.

Selecione 1, 2 ou 3 e pressione Enter

1: Continuar.

2: Avançar.

3: Vamos adiante!

Selection: 1

| Você pode sair do swirl e retornar ao prompt do R (>) a qualquer momento pressionando a tecla Esc. Se você já estiver no prompt, digite bye() para sair e salvar o seu progresso. Quando você sair corretamente, você verá uma

| breve mensagem lhe informando isso.

| Quando você estiver no prompt do R (>):

| -- Digitar skip() permite que você pule a questão atual.

| -- Digitar play() permite que você experimente com o R por conta própria; o swirl vai ignorar o que você fizer...

| -- ATÉ que você digite nxt() para ganhar a atenção do swirl novamente.

| -- Ao digitar bye() o swirl será fechado. O seu progresso será salvo.

| -- Ao digitar main(), você retornará ao menu principal do swirl.

| -- Ao digitar info(), essas informaçães serão apresentadas novamente.

| Vamos começar!

...

| Por favor, selecione um curso ou digite 0 para sair do swirl.

1: Aprenda R no R

2: Leve-me para o repositório de cursos do swirl!

Selection: 1

| Por favor, selecione uma lição ou digite 0 para retornar ao menu do curso.

1: Comandos basicos do R 2: Logicos 3: Sequencias numericas 4: Vetores 5: Valores ausentes 6: Data e Hora 7: Filtrando vetores 8: Matrizes e tabelas

9: Amostras e Simulacoes 10: Funcoes 11: Manipulacao com dplyr 12: Graficos Basicos 13: K Medias 14: Graficos Analiticos

Selection: 1

| Tentando carregar as dependências da lição...

| O pacote ‘openssl’ carregou corretamente!

| O pacote ‘jsonlite’ carregou corretamente!

| | 0%

| Oi Alunos! Nesta lição, vamos explorar alguns comandos básicos de programação na linguagem R.

...

|====== | 3%

| Se, a qualquer momento, você desejar mais informações sobre um determinado tópico relacionado a R, poderá digitar help.start() no prompt, que abrirá um menu de ajuda (no RStudio ou no navegador da Web padrão, dependendo da

| sua configuração). Como alternativa, uma simples pesquisa na web geralmente ajuda a encontrar a resposta que você está procurando.

...

|=========== | 5%

| Na sua forma mais básica, o R pode ser usado como uma calculadora interativa. Digite 5 + 7 e aperte Enter.

> 5+7

[1] 12

| Excelente!

|================= | 8%

| O R simplesmente imprime o resultado de 12 no console. No entanto, R também é uma linguagem de programação e, muitas vezes, o motivo pelo qual usamos uma linguagem de programação ao invés de uma calculadora é automatizar

| algum processo ou evitar repetições desnecessárias.

...

|====================== | 10%

| Nesse caso, podemos usar o resultado (12) acima em um segundo cálculo. Em vez de digitar novamente 5 + 7 toda vez que precisarmos, podemos criar uma nova variável que armazena o resultado.

...

|============================ | 13%

| A maneira que você atribui um valor a uma variável em R é usando um operador de atribuição, formado por um símbolo 'menor que' seguido por um sinal 'menos'. O que resulta no operador : <-

...

|================================= | 15%

| Pense no operador de atribuição como uma seta. Você está atribuindo o valor no lado direito da seta para o nome da variável no lado esquerdo da seta.

...

|======================================= | 18%

| As novas versões do R também suportam o operador 'igual' utilizado por diversas linguagens programação: =

...

|============================================ | 21%

| Para atribuir o resultado de 5 + 7 a uma nova variável chamada x, você digita x <- 5 + 7. Isso pode ser lido como 'x recebe 5 mais 7'. Experimente.

> x <- 5+7

| Está correto!

|================================================== | 23%

| Você notará que o R não imprimiu o resultado 12 desta vez. Quando você usa o operador de atribuição, o R assume que você não quer ver o resultado imediatamente, mas sim que você pretende usar o resultado para outra coisa

| mais tarde.

...

|======================================================= | 26%

| Para visualizar o conteúdo da variável x, apenas digite x e pressione Enter. Tente agora.

> x

[1] 12

| Essa é a resposta que eu esperava.

|============================================================= | 28%

| Agora, armazene o resultado de x - 3 em uma nova variável chamada y.

> y <- x-3

| Sua dedicação é inspiradora!

|================================================================== | 31%

| Qual é o valor de y? Digite y para descobrir.

> y

[1] 9

| Maravilha!

|======================================================================== | 33%

| Agora, vamos criar uma pequena coleção de números chamada vetor. Qualquer objeto que contém dados é chamado de estrutura de dados e vetores numéricos são o tipo mais simples de estrutura de dados em R. Na verdade, até mesmo

| um único número é considerado um vetor de comprimento um.

...

|============================================================================== | 36%

| A maneira mais fácil de criar um vetor é com a função c(), que significa "concatenar" ou "combinar". Para criar um vetor contendo os números 1.1, 9 e 3.14, digite c(1.1, 9, 3.14). Experimente agora e armazene o resultado em

| uma variável chamada z.

>

> z <- c(1.1,9,3.14)

| Na mosca! Bom trabalho!

|=================================================================================== | 38%

| Sempre que você tiver dúvidas sobre uma função específica, você pode acessar a ajuda interna do R através do comando `?`. Por exemplo, se você quiser mais informações sobre a função c(), digite ?c sem os parênteses. Tenta

| aí.

> ?c

| Na mosca! Bom trabalho!

|========================================================================================= | 41%

| Digite z para ver o conteúdo da variável. Observe que não há vírgulas separando os valores na saída.

> z

[1] 1.10 9.00 3.14

| Você é muito bom, amig@!

|============================================================================================== | 44%

| Você pode combinar vetores para criar um novo vetor. Crie um novo vetor que contenha z, 555 e z novamente

| nessa ordem. Não atribua esse vetor a uma nova variável, para que possamos ver o resultado imediatamente.

> c(z,555,z)

[1] 1.10 9.00 3.14 555.00 1.10 9.00 3.14

| Ótimo trabalho!

|==================================================================================================== | 46%

| Vetores numéricos podem ser usados em expressões aritméticas. Digite o seguinte comando para ver o que

| acontece: z \* 2 + 100.

> z\*2+100

[1] 102.20 118.00 106.28

| Sua dedicação é inspiradora!

|========================================================================================================= | 49%

| Primeiro, o R multiplicou cada um dos três elementos em z por 2. Em seguida, acrescentou 100 a cada elemento

| para obter o resultado que você vê acima.

...

|=============================================================================================================== | 51%

| Outros operadores aritméticos comuns são `+`, `-`,` / `e` ^ `(onde x ^ 2 significa 'x ao quadrado'). Para

| pegar a raiz quadrada, use a função sqrt() e tire o valor absoluto, use a função abs().

...

|==================================================================================================================== | 54%

| Pegue a raiz quadrada de z - 1 e atribua a uma nova variável chamada my\_sqrt.

> my\_sqrt <- sqrt(z-1)

| Na mosca! Bom trabalho!

|========================================================================================================================== | 56%

| Antes de vermos o conteúdo da variável my\_sqrt, o que você acha que ela contém?

1: um vetor de comprimento 3

2: um único número (ou seja, um vetor de comprimento 1)

3: um vetor de comprimento 0 (ou seja, um vetor vazio)

Selection: 1

| Continue assim e você chegará lá!

|=============================================================================================================================== | 59%

| Imprima o conteúdo do my\_sqrt na tela.

> my\_sqrt

[1] 0.3162278 2.8284271 1.4628739

| Continue assim e você chegará lá!

|===================================================================================================================================== | 62%

| Como você deve ter imaginado, o R primeiro subtraiu 1 de cada elemento de z, depois tomou a raiz quadrada de

| cada elemento. Isso deixa você com um vetor do mesmo comprimento que o vetor z original.

...

|========================================================================================================================================== | 64%

| Agora, crie uma nova variável chamada my\_div que obtenha o valor de z dividido por my\_sqrt.

> my\_div <- z/my\_sqrt

| Essa é a resposta que eu esperava.

|================================================================================================================================================ | 67%

| Qual alternativa você acha que é verdadeira?

1: my\_div é indefinido

2: my\_div é um único número (ou seja, um vetor de comprimento 1)

3: O primeiro elemento de my\_div é igual ao primeiro elemento de z dividido pelo primeiro elemento de my\_sqrt e assim por diante ...

Selection: 3

| Está correto!

|====================================================================================================================================================== | 69%

| Vá em frente e imprima o conteúdo de my\_div.

> my\_div

[1] 3.478505 3.181981 2.146460

| Mantenha esse bom nível!

|=========================================================================================================================================================== | 72%

| Quando dados dois vetores do mesmo comprimento, o R simplesmente executa a operação aritmética especificada

| (`+`, `-`,` \* `, etc.) elemento por elemento. Se os vetores tiverem comprimentos diferentes, o R `recicla` o

| vetor mais curto até que tenha o mesmo comprimento que o vetor mais longo.

...

|================================================================================================================================================================= | 74%

| Quando fizemos z \* 2 + 100 em nosso exemplo anterior, z era um vetor de comprimento 3, mas tecnicamente 2 e

| 100 são vetores de comprimento 1.

...

|====================================================================================================================================================================== | 77%

| Nos bastidores, o R está 'reciclando' o 2 fazendo um vetor de 2s e o 100 e fazendo um vetor de 100s. Em

| outras palavras, quando você pede ao R para calcular z \* 2 + 100, a expressão realmente calculada é: z \*

| c(2, 2, 2) + c(100, 100, 100).

...

|============================================================================================================================================================================ | 79%

| Para ver outro exemplo de como a reciclagem funciona, tente somar c(1, 2, 3, 4) e c(0, 10). Não se preocupe

| em salvar o resultado em uma nova variável.

> c(1,2,3,4)+c(0,10)

[1] 1 12 3 14

| Bom trabalho!

|================================================================================================================================================================================= | 82%

| Se o comprimento do vetor mais longo não for múltiplo do vetor mais curto, o R ainda aplicará o método de

| `reciclagem`, mas emitirá um aviso para que você saiba que algo suspeito pode estar acontecendo.

...

|======================================================================================================================================================================================= | 85%

| Digite c(1, 2, 3, 4) + c(0, 10, 100) para ver um exemplo.

> c(1,2,3,4)+c(0,10,100)

[1] 1 12 103 4

Warning message:

In c(1, 2, 3, 4) + c(0, 10, 100) :

longer object length is not a multiple of shorter object length

| Excelente!

|============================================================================================================================================================================================ | 87%

| Antes de concluir esta lição, gostaria de mostrar alguns truques para economizar tempo.

...

|================================================================================================================================================================================================== | 90%

| Anteriormente na lição, você computou z \* 2 + 100. Vamos fingir que cometeu um erro e que você quis

| adicionar 1000 em vez de 100. Você pode redigitar a expressão ou ...

...

|======================================================================================================================================================================================================= | 92%

| Em muitos ambientes de programação, a seta para cima irá percorrer os comandos anteriores. Tente apertar a

| seta para cima no seu teclado até chegar a este comando 'z \* 2 + 100', então mude de 100 para 1000 e

| pressione Enter. Se a seta para cima não funcionar para você, basta digitar o comando corrigido.

> z\*2+1000

[1] 1002.20 1018.00 1006.28

| Maravilha!

|============================================================================================================================================================================================================= | 95%

| Finalmente, vamos fingir que você gostaria de ver o conteúdo de uma variável que você criou anteriormente,

| mas você não consegue lembrar se você a nomeou my\_div ou myDiv. Você poderia tentar os dois e ver o que

| funciona, ou ...

...my\_div

|================================================================================================================================================================================================================== | 97%

| Você pode digitar as duas primeiras letras do nome da variável e pressionar a tecla Tab (às vezes precia ter

| fé e apertar mais de uma vez). A maioria dos ambientes de programação fornecerá uma lista de variáveis que

| você criou e que começam com 'my'. Isso é chamado de preenchimento automático e pode ser bastante útil

| quando você tem muitas variáveis em sua área de trabalho. Tenta aí. (Se o preenchimento automático não

| funcionar para você, apenas digite my\_div e pressione Enter.)

> my\_div

[1] 3.478505 3.181981 2.146460

| Na mosca! Bom trabalho!

|========================================================================================================================================================================================================================| 100%

| Gostaria de informar ao professor sobre a conclusão desta lição

1: Não

2: Sim

Selection: 2

| Qual o código da sua turma? (exemplo FIAP-01IA)

24IA

| Qual seu código de aluno?

344154

| Qual seu nome?

Diego Cohen

| O que achou deste exercício?

Excelente!

[1] "Tentando submeter ao professor, tentativa 1 ... (max 5) ..."

[1] "saved"

#############################################################################################################

Seu resultado foi salvo!

#############################################################################################################