Olá alunos. Hoje vamos falar sobre valores ausentes em um conjunto de dados. Esses tipos de dados, também

| são conhecidos como valores omitidos ou mais comumente pelo termo em inglês, missing values. Esta lição usa

| algumas funções de amostras e simulações (números aleatórios), que veremos melhor em outra lição. Se você

| ainda não tiver visto isso em aula, não se preocupe que logo mais teremos esse conteúdo.

...

|==== | 4%

| Valores ausentes desempenham um papel importante na estatística e na análise de dados. Freqüentemente, os

| valores ausentes não devem ser ignorados, mas cuidadosamente estudados para ver se há um padrão subjacente

| ou alguma causa para sua ausência.

...

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

| Em R, NA é usado para representar qualquer valor que "não está disponível" (not available) ou seja está

| "ausente" (no sentido estatístico). Isso não quer dizer que o número não exista, ou seja, o valor pode

| existir mas você não tem como saber qual é este valor. Nesta lição, exploraremos os valores ausentes para

| você ganhar uma experiência nisso.

...

|============= | 12%

| Qualquer operação envolvendo NA geralmente produz NA como resultado. Para ilustrar, crie um vetor c(44, NA,

| 5, NA) e o atribua a uma variável x.

> x <- c(44,NA,5,NA)

| Bom trabalho!

|================= | 17%

| Agora multiplique x por 3.

> x\*3

[1] 132 NA 15 NA

| Você acertou!

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

| Observe que os elementos do vetor resultante que correspondem aos valores de NA em x também são NA.

...

|========================= | 25%

| Para tornar as coisas um pouco mais interessantes, crie um vetor contendo 1.000 desenhos de uma distribuição

| normal padrão com y <- rnorm(1000). Veremos mais sobre esta função `rnorm` em outra lição.

> y <- rnorm(1000)

| Sua dedicação é inspiradora!

|============================= | 29%

| Em seguida, vamos criar um vetor contendo 1000 NAs com z <- rep(NA, 1000).

> z <- rep(NA,1000)

| Bom trabalho!

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

| Finalmente, vamos selecionar 100 elementos aleatoriamente desses 2000 valores (combinando `y` e `z`) de tal

| forma que não sabemos quantos NAs teremos ou quais posições eles ocuparão em nosso vetor final. Para fazer

| isso, digite my\_data <- sample(c(y, z), 100).

> my\_data <- sample(c(x,y),100)

| Você quase acertou. Tente novamente. Ou digite info() para mais opções.

| A função sample() pega uma amostra aleatória dos dados fornecidos no primeiro argumento (neste caso, c(y,

| z)) do tamanho especificado pelo segundo argumento (100). O comando my\_data <- sample(c(y, z), 100) nos dará

| o que queremos.

> my\_data <- sample(c(y,z),100)

| Continue assim e você chegará lá!

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

| Vamos primeiro fazer a pergunta de onde nossos NAs estão localizados em nossos dados. A função is.na() nos

| diz se cada elemento de um vetor é NA. Rode is.na() no vetor my\_data e atribua o resultado à variável my\_na.

| (opa! comecei a complicar)

> my\_na <- is.na(my\_data)

| Mantenha esse bom nível!

|========================================== | 42%

| Agora, imprima my\_na no console, para ver o que você criou.

> my\_na

[1] TRUE TRUE FALSE FALSE TRUE FALSE FALSE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE

[18] FALSE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE TRUE FALSE FALSE FALSE

[35] TRUE FALSE TRUE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE FALSE

[52] TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE FALSE TRUE FALSE TRUE TRUE FALSE FALSE FALSE

[69] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE TRUE

[86] TRUE FALSE FALSE FALSE TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE

| Excelente trabalho!

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

| Onde você ver um valor TRUE, você sabe que o elemento correspondente de my\_data é NA. Da mesma forma, em

| todos os lugares que você vê um FALSE, você sabe que o elemento correspondente de my\_data é um de nossos

| sorteios aleatórios do vetor.

...

|================================================== | 50%

| Na lição sobre operadores lógicos, introduzimos o operador `==` como um método de testar a igualdade entre

| dois objetos. Então, você pode pensar que a expressão my\_data == NA produz os mesmos resultados que is.na().

| Tente isso.

>

> my\_data==NA

[1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA

[36] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA

[71] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA

| Mantenha esse bom nível!

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

| Ok, você acertou o comando! Mas poxa, parece que não deu certo :-(

...

|=========================================================== | 58%

| A razão de você ter um vetor todo cheio de NAs é que NA não é um valor, mas apenas um espaço reservado para

| uma quantidade que não está disponível. Se não está disponível, não se sabe o valor que realmente 'deveria

| estar lá' e não é possível saber se é uma igualdade.

...

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

| Se você não tem familiaridade com estatística, isso pode ser um pouco confuso. Se você é de tecnologia, NA é

| muito similar aos valores NULL e None que muitas linguagens tem - com algumas diferenças.

...

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

| Não se preocupe em ser PhD em NA, o principal objetivo é você entender que deve ser cauteloso ao usar

| expressões lógicas sempre que os NAs estiverem no meio, já que um único valor de NA pode inviabilizar a

| coisa toda.

...

|======================================================================== | 71%

| Então, de volta às tarefas ... Agora temos um vetor, chamado my\_na, que tem um TRUE para cada NA e FALSE

| para cada valor numérico. Podemos então calcular o número total de NAs em nossos dados.

...

|============================================================================ | 75%

| O truque é reconhecer que nos bastidores, o R representa TRUE como o número 1 e FALSE como o número 0.

| Portanto, se pegarmos a soma de um monte de TRUEs e FALSEs, obtemos o número total de TRUEs.

...

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

| Vamos tentar fazer isso. Chame a função sum() em my\_na para contar o número total de TRUEs em my\_na e,

| assim, o número total de NAs em my\_data. Não atribua o resultado a uma nova variável.

> sum(my\_na)

[1] 47

| Excelente!

|==================================================================================== | 83%

| Muito legal, né? Vamos dar uma olhada nos dados para nos convencer de que tudo "se soma". Imprime my\_data no

| console.

> my\_data

[1] NA NA -1.99341693 -0.13238153 NA 0.38703070 0.43650750 NA

[9] NA NA -1.25737242 NA NA 0.49037392 NA NA

[17] NA -0.07243295 NA NA NA 0.71929402 NA NA

[25] -0.22631231 NA 0.61864265 0.87445431 NA NA NA 0.14682763

[33] 0.66155577 0.34941250 NA 0.22432276 NA NA -1.40077675 0.50269511

[41] NA 0.46421589 NA NA NA 0.26961470 -0.02859672 NA

[49] NA -0.91936712 0.58726084 NA NA NA -0.37602809 1.79452441

[57] -0.72153710 0.08676694 NA NA -1.96281589 NA 0.43815943 NA

[65] NA -0.37196021 2.77226018 0.53172190 -0.98343377 -1.57147944 -0.56164737 0.47771508

[73] -0.63185578 -1.77268620 NA -0.28027511 -1.49272027 1.49504102 0.19306556 NA

[81] -2.04310347 NA -0.06609024 1.17230028 NA NA -0.22168293 -0.48479459

[89] -0.84647727 NA NA NA 1.09774405 0.92740141 0.34062784 NA

[97] 0.72744748 0.17387156 NA -0.23784830

| Ótimo trabalho!

|======================================================================================== | 88%

| Agora passamos pelos NAs, vamos ver um segundo tipo de valor ausente, o NaN, que significa "não um número".

| Para gerar NaN, tente dividir 0 por 0.

> 0/0

[1] NaN

| Você está em um bom ritmo!

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

| Vamos ver mais um tipo de dado especial, só por diversão. Em R, Inf significa infinito. O que acontece se

| você subtrair Inf de Inf?

> inf-inf

Error: object 'inf' not found

> Inf-Inf

[1] NaN

| Você acertou!

|================================================================================================= | 96%

| Pois é! Para o R, infinito menos infinito não é um número! Com o tempo você se acostumará com essas

| expressões e valores especiais.

...

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

| Gostaria de informar ao professor sobre a conclusão desta lição

1: Sim

2: Não

Selection: 1

| 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?

ótimo!

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

[1] "saved"

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

Seu resultado foi salvo!

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