Olá novamente! O R tem um modo especial de representar datas e horas, o que pode ser útil se você estiver

| trabalhando com dados que mostram como algo muda com o tempo (ou seja, dados de séries temporais) ou se seus

| dados contêm alguma outra informação temporal, como datas de nascimento.

...

|=== | 3%

| As datas são representadas pela classe "Date" e as horas são representadas pelas classes "POSIXct" e

| "POSIXlt". Internamente, as datas são armazenadas como o número de dias desde 1970-jan-01 e os horários são

| armazenados como o número de segundos desde 1970-jan-01 (para "POSIXct") ou uma lista de segundos, minutos,

| horas etc. (para "POSIXlt").

...

|====== | 6%

| Vamos começar criando uma variável chamada d1 para armazenar a data atual. Para isso digite o comando d1 <-

| Sys.Date() . (Essa é a letra "d" e o número "1".)

> d1 <- sys.Date()

Error in sys.Date() : could not find function "sys.Date"

> d1 <- Sys.Date()

| Você está em um bom ritmo!

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

| Use a função class() para confirmar que d1 é um objeto Date.

> class(d1)

[1] "Date"

| Continue assim e você chegará lá!

|=========== | 11%

| Podemos usar a função unclass() para ver como d1 é internamente. Experimente!

> unclass(d1)

[1] 19195

| Você é muito bom, amig@!

|============== | 14%

| Esse é o número exato de dias desde 01 de janeiro de 1970!

...

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

| No entanto, se você imprimir d1 no console, receberá a data de hoje - YEAR-MONTH-DAY. Digite aí para ver.

> d1

[1] "2022-07-22"

| Sua dedicação é inspiradora!

|==================== | 19%

| E se precisarmos fazer referência a uma data anterior a 1970-01-01? Crie uma variável d2 contendo o

| resultado de as.Date("1969-01-01").

> d2 <- as.Date(1969-01-01)

Error in as.Date.numeric(1969 - 1 - 1) : 'origin' must be supplied

> d2 <- as.Date("1969-01-01"")

+

+ d2 <- as.Date("1969-01-01")

Error: unexpected string constant in:

"

d2 <- as.Date(""

> as.Date('1969-01-01')

[1] "1969-01-01"

| Isso não é exatamente o que espero. Tente novamente. Ou digite info() para mais opções.

| Digite d2 <- as.Date("1969-01-01").

> d2 <- as.Date("1969-01-01")

| Mantenha esse bom nível!

|====================== | 22%

| Agora use unclass() para ver como d2 é internamente.

> unclass(d2)

[1] -365

| Excelente trabalho!

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

| Como você pode ter suposto, há um número negativo. Neste caso, é -365, desde 1969-01-01 é exatamente um ano

| civil (ou seja, 365 dias) ANTES de 1970-01-01.

...

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

| Agora, vamos dar uma olhada em como o R armazena horários. Você pode acessar a data e hora atuais usando a

| função Sys.time() sem argumentos. Faça isso e armazene o resultado em uma variável chamada t1.

> Sys.time()

[1] "2022-07-22 22:41:55 -03"

| Quase! Tente novamente. Ou digite info() para mais opções.

| t1 <- Sys.time() armazenará a data e hora atuais em uma variável chamada t1.

> t1 = Sys.time()

| Tente novamente. Acertar de primeira não tem graça mesmo! Ou digite info() para mais opções.

| t1 <- Sys.time() armazenará a data e hora atuais em uma variável chamada t1.

> t1 <- Sys.time()

| Excelente trabalho!

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

| Veja o conteúdo de t1.

> t1

[1] "2022-07-22 22:42:08 -03"

| Todo o trabalho duro está dando resultado!

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

| E verifique a classe de t1.

> class(t1)

[1] "POSIXct" "POSIXt"

| Continue assim e você chegará lá!

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

| Como mencionado anteriormente, POSIXct é apenas uma das duas maneiras pelas quais o R representa a

| informação de tempo/horários. Use unclass() para ver como t1 é internamente - o número (grande) de segundos

| desde 1970.

> unclass(t1)

[1] 1658540528

| Continue assim e você chegará lá!

|======================================= | 39%

| Por padrão, Sys.time() retorna um objeto da classe POSIXct, mas podemos converter o resultado para POSIXlt

| com a função as.POSIXlt(Sys.time()). Experimente e armazene o resultado em t2.

> as.POSIXlt(Sys.time())

[1] "2022-07-22 22:43:22 -03"

| Está perto... posso sentir! Tente novamente. Ou digite info() para mais opções.

| t2 <- as.POSIXlt(Sys.time()) irá forçar o resultado de Sys.time para 'POSIXlt' e armazenar o resultado em

| t2.

> t2 <- as.POSIXlt(Sys.time())

| Maravilha!

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

| Verifique a classe de t2.

> class(t2)

[1] "POSIXlt" "POSIXt"

| Ótimo trabalho!

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

| Agora veja seu conteúdo.

>

> t2

[1] "2022-07-22 22:43:37 -03"

| Você está indo muito bem!

|================================================ | 47%

| O formato impresso de t2 é idêntico ao de t1. Agora execute unclass() em t2 para ver como são diferentes

| internamente.

> unclass(t2)

$sec

[1] 37.77577

$min

[1] 43

$hour

[1] 22

$mday

[1] 22

$mon

[1] 6

$year

[1] 122

$wday

[1] 5

$yday

[1] 202

$isdst

[1] 0

$zone

[1] "-03"

$gmtoff

[1] -10800

attr(,"tzone")

[1] "" "-03" "-02"

| Excelente trabalho!

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

| t2, como todos os objetos POSIXlt, é apenas uma lista de valores que compõem a data e a hora. Digite

| str(unclass(t2)) para ter uma visão mais compacta.

> str(unclass(t2))

List of 11

$ sec : num 37.8

$ min : int 43

$ hour : int 22

$ mday : int 22

$ mon : int 6

$ year : int 122

$ wday : int 5

$ yday : int 202

$ isdst : int 0

$ zone : chr "-03"

$ gmtoff: int -10800

- attr(\*, "tzone")= chr [1:3] "" "-03" "-02"

| Na mosca! Bom trabalho!

|===================================================== | 53%

| Se, por exemplo, nós quisermos apenas os minutos do tempo armazenado em t2, podemos acessá-los diretamente

| com t2$min. Tente.

> t2$min

[1] 43

| Excelente trabalho!

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

| Agora que exploramos todos os três principais tipos de objetos de data e hora, vamos ver algumas funções que

| extraem informações úteis de qualquer um desses objetos - weekdays(), months() e quarters().

...

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

| A função weekdays() retornará o dia da semana de qualquer objeto de data ou hora. Experimente em d1, que é o

| objeto Date que contém a data de hoje.

> weekdays(d1)

[1] "Friday"

| Toda a prática está rendendo frutos!

|============================================================== | 61%

| A função months() também funciona em qualquer objeto de data ou hora. Experimente em t1, que é o objeto

| POSIXct que contém a hora atual (bem, foi a hora atual quando você a criou).

> months(d1)

[1] "July"

| Não exatamente. Tente outra vez. Ou digite info() para mais opções.

| months(t1) lhe dará o mês atual.

> months(t1)

[1] "July"

| Mantenha esse bom nível!

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

| A função quarters() retorna o trimestre do ano (Q1-Q4) de qualquer objeto de data ou hora. Experimente em

| t2, que é o objeto POSIXlt que contém a hora em que você a criou.

> quarters(t1)

[1] "Q3"

| Quase! Tente novamente. Ou digite info() para mais opções.

| quarters(t2) lhe dará o trimestre atual.

> quarters(t2)

[1] "Q3"

| Todo o trabalho duro está dando resultado!

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

| Geralmente, as datas e horas em um conjunto de dados externos estarão em um formato que o R não reconhece. A

| função strptime() pode ser útil nesses casos.

...

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

| O strptime() converte os vetores de caracteres em POSIXlt. É semelhante ao as.POSIXlt(), exceto que a

| entrada não precisa estar em um formato específico (AAAA-MM-DD).

...

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

| Para ver como funciona, armazene a seguinte string em uma variável chamada t3: 'Outubro 17, 1986 08:24' (com

| as aspas).

> t3 <- 'Outubro 17, 1986 08:24'

| Você é muito bom, amig@!

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

| Agora, use strptime(t3, "%B %d, %Y %H:%M") para ajudar o R a converter nosso objeto de data / hora em um

| formato que ele entenda. Atribua o resultado a uma nova variável chamada t4. (Você pode ver a documentação

| do strptime() se quiser saber mais sobre como funciona.)

> strptime(t3, '%B %d, %y %H:%M')

[1] NA

| Não é bem isso. Tente novamente. Ou digite info() para mais opções.

| t4 <- strptime(t3, "%B %d, %Y %H:%M") converterá nosso objeto de data / hora em um formato que o R entenda.

> strptime(t3, '%B %d, %Y %H:%M')

[1] NA

| Mas uma vez. Você consegue! Ou digite info() para mais opções.

| t4 <- strptime(t3, "%B %d, %Y %H:%M") converterá nosso objeto de data / hora em um formato que o R entenda.

> t4 <- strptime(t3, '%B %d, %Y %H:%M')

| Na mosca! Bom trabalho!

|=============================================================================== | 78%

| Imprima o conteúdo de t4.

> t4

[1] NA

| Você acertou!

|================================================================================= | 81%

| Esse é o formato que esperamos. Agora, vamos verificar sua classe.

> class(t4)

[1] "POSIXlt" "POSIXt"

| Está correto!

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

| Finalmente, há várias operações que você pode executar em datas e horas, incluindo operações aritméticas (+

| e -) e comparações (<, ==, etc.)

...

|======================================================================================= | 86%

| A variável t1 contém a hora em que você a criou (lembre-se que você usou Sys.time()). Confirme que algum

| tempo se passou desde que você criou t1 usando o operador 'maior que' para compará-lo com a hora atual:

| Sys.time() > t1

> Sys.time()>t1

[1] TRUE

| Todo o trabalho duro está dando resultado!

|========================================================================================== | 89%

| Então, sabemos que algum tempo se passou, mas quanto? Tente subtrair t1 da hora atual usando Sys.time() -

| t1. Não esqueça os parênteses no final do Sys.time(), já que é uma função.

> Sys.time()-t1

Time difference of 6.298914 mins

| Continue assim e você chegará lá!

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

| O mesmo raciocínio se aplica à adição e aos outros operadores de comparação. Se você quiser mais controle

| sobre qual unidade de medida (dia, hora, etc.) ao calcular a diferença acima em tempos, você pode usar

| difftime(), que permite especificar um parâmetro de 'unidades'.

...

|=============================================================================================== | 94%

| Use difftime(Sys.time(), t1, units = 'days') para encontrar a quantidade de tempo em DIAS que passou desde

| que você criou t1.

> difftime(Sys.time(), t1, units='days')

Time difference of 0.004853941 days

| Maravilha!

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

| Nesta lição, você aprendeu o básico sobre como trabalhar com data e hora no R. Embora seja importante

| entender o básico, se você estiver trabalhando muito com datas e horários, talvez seja interessante procurar

| aprender sobre o pacote lubridate do Hadley Wickham.

...

|=====================================================================================================| 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?

Legal!

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

[1] "saved"

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

Seu resultado foi salvo!

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