**ATIVIDADE 1**

Realizar a atividade abaixo e postar a resolução no local indicado para envio de arquivo no APRENDER.

**EXERCÍCIOS**

1. Quais as diferenças entre NaN, NULL e Inf ? Digite exemplos de expressões que retornam cada um desses resultados.

Inf = > 1 / 0

matrix() = NA

log(-1) = NaN

1. Por que o código abaixo retorna erro? Arrume o código para retornar o valor TRUE.

Porque o sinal de igual simples é atribuição e não comparação

x <- 4

if(x = 4) { TRUE }

o certo é

if(x == 4){TRUE}

1. Usando if e else, escreva um código que retorne a string **“número”** caso o objeto x seja da classe numeric; a string **“palavra”** caso o objeto seja da classe character; e **NULL** caso contrário.

x=1

if(ifelse(is.numeric(x),"número","NULL") == "NULL"

ifelse(is.character(x),"palavra","NULL") =="NULL"

1. Sem rodar o código, calcule o que a expressão 5 + 3 \* 10 %/% 3 == 15 vai resultar no R. Em seguida, apenas utilizando parênteses, faça a expressão retornar o valor contrário (i.e., se originariamente for TRUE, faça retornar FALSE).

5 + (3 \* 10) %/% 3 == 15

1. Crie uma função para calcular a velocidade média de um objeto que percorreu 150 km em 2.5 horas. Formula: vm = d/t

vw <- function(d,t) {

return(d/t)

}

1. Use as funções **union, intersect e setdiff** para encontrar a união, o intersecto e as diferenças entre os conjuntos A e B. Aprenda no help como utilizar estas funções.

A = 1 ,2, 3, 4, 5

B = 4, 5, 6, 7

union(A,B)

[1] 1 2 3 4 5 6 7

> intersect(A,B)

[1] 4 5

> setdiff(A,B)

[1] 1 2 3

1. Qual a diferença entre uma matriz e um *data frame* no R?

A matriz só aceita dados do eso tipo e o datafrae de todos os tipos

1. O que ocorre se definirmos o argumento byrow = TRUE na construção de uma matriz?

Byrow=T os dados são preenchidos na sequencia em linha e Byrow=F são preenchidos em colunas

1. Crie um objeto chamado *info* que contém seu nome, idade, altura, peso, e-mail e telefone.

info=c("Luciano","49","1.86","105Kg","zicadopv@gmail.com","(18)98806.3287")

1. Usando o comando data.frame(), entre com a matriz de dados peso=(60,70,80) e altura=(160,150,170). Depois insira mais uma coluna chamada IMC=(baixo, médio, alto). Calcule a média da variável altura.

> peso=c(60,70,80)

> altura=c(160,150,170)

df=data.frame(peso,altura)

peso altura

1 60 160

2 70 150

3 80 170

> IMC=c("baixo","medio","alto")

> rownames(df)=IMC

> df

peso altura

baixo 60 160

medio 70 150

alto 80 170

sum(df[[2]])/3

[1] 160

1. Suponha que você marcou o tempo que leva para chegar a cada uma de suas parcelas no campo. Os tempos em minutos foram: 18, 14, 14, 15, 14, 34, 16, 17, 21, 26. Passe estes valores para o R, chame o objeto de *tempo*. Usando funções do R ache o tempo máximo, mínimo e o tempo médio que você gasta para chegar em suas parcelas.

> tempo = c(18, 14, 14, 15, 14, 34, 16, 17, 21, 26)

[1] 18 14 14 15 14 34 16 17 21 26

> max(tempo)

[1] 34

> min(tempo)

[1] 14

> diff(tempo)

[1] -4 0 1 -1 20 -18 1 4 5

**Obs.: Ops, o valor 34 foi um erro, ele na verdade é 15. Sem digitar tudo novamente, e usando colchetes [ ], mude o valor e calcule novamente o tempo médio.**

**> diff(tempo)**

**[1] -4 0 1 -1 1 1 1 4 5**

1. Suponha que você deseja jogar na megasena, mas não sabe quais números jogar, use a função *sample* do R para escolher seis números para você. Lembrando que a megasena tem valores de 1 a 60.

> sample(1:60,6)

[1] 36 25 19 59 22 11

1. Crie uma sequência de dados de 1 a 30 apenas com números ímpares. Use a função seq().

> seq(1, 30, by=2)

[1] 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29

1. Simule o resultado de 250 jogadas de um dado (sample). Descubra como usar a função **table** para saber quantas vezes cada número do dado foi sorteado. OBS: a página de help irá conter muitas informações que você ainda não sabe o que é e que não precisará utilizar. Veja os exemplos para ter uma ideia de como funciona.

a <- c(“A”,”B”,”C”)

table(sample(a,replace=TRUE))

> table(sample(1:6,250, replace=T))

1 2 3 4 5 6

34 36 46 41 43 50

1. Crie as seguintes sequências usando as funções rep ou seq:
2. Objeto chamado "letra.a" com caracteres:

a a a a a a

> rep('a',6)

[1] "a" "a" "a" "a" "a" "a"

1. Objeto chamado "numeros" com valores numéricos inteiros:

1 1 1 2 2 2 3 3 3

> rep(c(1:3), each=3)

[1] 1 1 1 2 2 2 3 3 3

1. Objeto chamado "decrescente" com valores numéricos inteiros:

1 1 1 2 2 3

> sort(t, decreasing=T)

[1] 3 2 2 1 1 1