ForteLab1.R

andre

2022-05-21

# install.packages('tinytex')  
  
# Boolean logic is used to determine whether various statements are true or false. These are the only two   
# options in boolean logic. TRUE is equivalent to 1 and FALSE is equivalent to 0 in boolean logic.   
# There are three operations that can be applied in boolean logic. These operations are AND, OR, and NOT. I  
# will demonstrate these operations in the following examples:  
  
# First, let's discuss the AND operation. It is denoted by "&" in R. For an operation to result in TRUE   
# using the AND operation, both operands (the things on both sides of the AND operation) must be TRUE.   
# Here are the four possible outcomes:  
  
TRUE & TRUE

## [1] TRUE

TRUE & FALSE

## [1] FALSE

FALSE & TRUE

## [1] FALSE

FALSE & FALSE

## [1] FALSE

# Now let's discuss the OR operation. It is denoted by "|" in R. For an operation to result in TRUE using  
# the OR operation, either one of the operands or both must be TRUE. It only results in FALSE if both  
# operands are FALSE. Here are the possible outcomes:  
  
TRUE | TRUE

## [1] TRUE

TRUE | FALSE

## [1] TRUE

FALSE | TRUE

## [1] TRUE

FALSE | FALSE

## [1] FALSE

# The final operation in boolean logic is NOT. It is denoted by "!" in R. The NOT operation simply negates the  
# result of a statement. Here are some examples:  
  
!TRUE

## [1] FALSE

!FALSE

## [1] TRUE

!(TRUE & TRUE)

## [1] FALSE

# Take note of how the operations can be combined to form more complex statements. TRUE & TRUE should  
# result in TRUE, but is negated by the NOT operation and, therefore, results in FALSE. Also remember that  
# TRUE can be replaced by a 1 and FALSE can be replaced by a 0.  
  
# The following conditional statements will tell you whether the expected outcomes occur,  
# or if there is an error in the statement (I have replaced TRUE with 1 and FALSE with 0):  
  
if( 1&1 ) "1&1 is true" else "error somewhere"

## [1] "1&1 is true"

if( 1&0 ) "error somewhere" else "1&0 is false"

## [1] "1&0 is false"

if( 0&1 ) "error somewhere" else "0&1 is false"

## [1] "0&1 is false"

if( 0&0 ) "error somewhere" else "0&0 is false"

## [1] "0&0 is false"

if( 1|1 ) "1|1 is true" else "error somewhere"

## [1] "1|1 is true"

if( 1|0 ) "1|0 is true" else "error somewhere"

## [1] "1|0 is true"

if( 0|1 ) "0|1 is true" else "error somewhere"

## [1] "0|1 is true"

if( 0|0 ) "error somewhere" else "0|0 is false"

## [1] "0|0 is false"

if(!1) "error somewhere" else "!1 is false"

## [1] "!1 is false"

if(!0) "!0 is true" else "error somewhere"

## [1] "!0 is true"

if(!(TRUE & TRUE)) "error somewhere" else "!(TRUE & TRUE) is false"

## [1] "!(TRUE & TRUE) is false"

# The following code demonstrates various methods of vector math  
  
height <- c(59,60,61,58,67,72,70)  
weight <- c(150,140,180,220,160,140,130)  
a <- 150  
  
mean(height)

## [1] 63.85714

mean(weight)

## [1] 160

length(c(height, weight))

## [1] 14

sum(height)

## [1] 447

sum(height) / length(height)

## [1] 63.85714

maxH <- max(height)  
minW <- min(weight)  
  
newWeight <- weight + 5  
newWeight / height

## [1] 2.627119 2.416667 3.032787 3.879310 2.462687 2.013889 1.928571

if(maxH > 60) "yes" else "no"

## [1] "yes"

if(minW > a) "yes" else "no"

## [1] "no"