R Track Week 2 Assignment

Kumudini Bhave

July 23, 2016

Table of Contents

# **Function ncombr()**

This is an R Markdown document for providing documentation for the function **ncombr()**.

**ncombr(num, rsub)** takes two arguments

* *num : positive integer*, objects in a collection
* *rsub : positive integer*, number of objects to be choosen

*ncombr()* calculates the the number of different ways one can choose *rsub*, number of objects from a collection of *num*, number of objects

*ncombr()* uses an internal function **intfact()** to calculate the factorial

**intfact(num)** takes one argument

* *numfac : positive integer*, number whose factorial is to be calculated

## R Code for function ncombr()

# @ Kumudini Bhave   
# R Bridge Course Week 2 Problem A choose B = A! /() B! \* (A-B)! )  
#  
# This is a generic function to calculate Combination a\_Choose\_b on any given positive number  
# ncombr() function will take two arguments of type "number"   
# Ideally num>=rsub>0  
# It validates the value of the input argument and calculates the Combination for set of positive   
# integer numbers : num, rsub  
# It uses an internal function intfact() for calculating the required factorials  
#  
# The function choose() available in R calculates for -ve and fractional   
# value of num abd rsub as well   
# However the ncombr() assumes integer values only. Hence for such values   
# the results of the two evaluations may differ#  
  
  
ncombr<- function(num, rsub){  
   
 if(nargs()<2) {  
 if (missing(num)) {  
 print("USAGE : ncombr(n, r) : For Positive Integer Values of n ")  
 return("NA")  
 }   
 if (missing(rsub)) {  
 print("USAGE : ncombr(n, r) : For Positive Integer Values of r ")  
 return("NA")  
 }  
 }  
   
 if ((class(num) != "numeric") | (class(rsub) != "numeric")) {  
 print("Enter positive intergers only")  
 print("USAGE : ncombr(n, r) : For Positive Integer Values of n And r ")  
   
 return("NA")  
   
 } else {  
 num <- as.integer(num)  
 rsub <- as.integer(rsub)  
 }  
   
 cat(num," choose ",rsub,"\n")  
   
 choosefac <- 1  
   
   
 # Internal function intfact() for calculating factorial  
   
 intfact <- function(numfac){  
 numfac <- (as.integer(numfac))  
   
 fac <- 1  
   
 if (numfac < 0) {  
 fac <- NaN  
 } else if(numfac == 0) {  
 fac  
 } else {  
 for(i in 1:numfac){  
 fac <- fac \* i  
 }  
 }  
 return (fac)  
 }  
   
 # intfact() ends  
   
   
 # Calculating combination for positive integer values of num and rsub  
 # Ideally num >= rsub > 0  
   
   
 if (num > 0 & rsub > 0 & (num >= rsub)) {  
   
 choosefac <- (intfact(num) / (intfact(num - rsub) \* intfact(rsub)))  
   
 } else if ((num >= 0 & ( (num < rsub) | (rsub < 0) )) | (num < 0 & rsub < 0)) {  
   
 choosefac <- 0  
   
 } else if(rsub == 0) {  
   
 # Case of Combination(0,0) = 1  
   
 choosefac <- 1  
 }  
   
 cat("Combination (",num,",",rsub,") is: ",choosefac)  
   
} # Enf od ncombr()

## Test Script for ncombr()

The code snippets below tests for different combinations of arguments passed to ncombr()

ncombr(5,3)

## 5 choose 3   
## Combination ( 5 , 3 ) is: 10

choose(5,3) # Verifying with choose() in R

## [1] 10

ncombr(5,2)

## 5 choose 2   
## Combination ( 5 , 2 ) is: 10

choose(5,2) # Verifying with choose() in R

## [1] 10

ncombr(5,5)

## 5 choose 5   
## Combination ( 5 , 5 ) is: 1

choose(5,5) # Verifying with choose() in R

## [1] 1

ncombr(1,5)

## 1 choose 5   
## Combination ( 1 , 5 ) is: 0

choose(1,5) # Verifying with choose() in R

## [1] 0

ncombr(5,0)

## 5 choose 0   
## Combination ( 5 , 0 ) is: 1

choose(5,0) # Verifying with choose() in R

## [1] 1

ncombr(0,5)

## 0 choose 5   
## Combination ( 0 , 5 ) is: 0

choose(0,5) # Verifying with choose() in R

## [1] 0

ncombr(5,-3)

## 5 choose -3   
## Combination ( 5 , -3 ) is: 0

choose(5,-3) # Verifying with choose() in R

## [1] 0

ncombr(-5,3)

## -5 choose 3   
## Combination ( -5 , 3 ) is: 1

choose(-5,3) # Verifying with choose() in R

## [1] -35

ncombr(0,0)

## 0 choose 0   
## Combination ( 0 , 0 ) is: 1

choose(0,0) # Verifying with choose() in R

## [1] 1

ncombr(-5,0)

## -5 choose 0   
## Combination ( -5 , 0 ) is: 1

choose(-5,0) # Verifying with choose() in R

## [1] 1

ncombr(0,-5)

## 0 choose -5   
## Combination ( 0 , -5 ) is: 0

choose(0,-5) # Verifying with choose() in R

## [1] 0

Some combination results differ for negative values and factorial value of *num* and *rsub*

# Result differs for negative values of num  
ncombr(-5,2)

## -5 choose 2   
## Combination ( -5 , 2 ) is: 1

choose(-5,2) # Verifying with choose() in R

## [1] 15

# Result differs for fractional arguments  
ncombr(5.2,3)

## 5 choose 3   
## Combination ( 5 , 3 ) is: 10

choose(5.2,3) # Verifying with choose() in R

## [1] 11.648