**1. 벡터**

**- 여러개의 동일한 형태의 데이터를 모아서 함께 저장하는 것을 의미**

**- c( ) 함수 사용**

**- 동일한 데이터형이 저장되어야함**

**- 다른유형의 데이터가 있을경우 강제 형변환 또는 에러발생**

접기

**> c(1,2,3,4,5)**

**[1] 1 2 3 4 5**

**> # 숫자와 문자데이터가 섞여있을 경우 강제형변환 (모두다 문자로)**

**> c(1,2,3,4,"5")**

**[1] "1" "2" "3" "4" "5"**

**> # 벡터명을 지정하여 데이터 입력**

**> vec1 <- c(1,2,3,4,5)**

**> vec1**

**[1] 1 2 3 4 5**

접기

**① 특정위치값 제어하기**

**- 특정 항목의 요소를 보고싶을 경우 : 벡터명[번호]**

**- 기존 벡터에 새로운 데이터 추가 : 벡터명 <- 데이터**

**- append 함수를 사용하여 데이터 추가 가능 : append(vec1,10,after=3)**

접기

**> # 특정위치의 요소만 출력**

**> vec1[3]**

**[1] 3**

**>**

**> # 특정위치의 요소만 빼고 출력**

**> vec1[-3]**

**[1] 1 2 4 5**

**>**

**> # vec1의 총길이에서 2개를 뺀 갯수만큼 출력**

**> vec1[1:(length(vec1)-2)]**

**[1] 1 2 3**

**>**

**> # 1번부터 3번요소까지 뺀 나머지 요소만 출력**

**> vec1[-1:-3]**

**[1] 4 5**

**>**

**> # 2번부터 4번요소까지만 출력**

**> vec1[2:4]**

**[1] 2 3 4**

**>**

**> # 특정요소의 값 변경**

**> vec1[2] <- 6**

**> vec1**

**[1] 1 6 3 4 5**

**>**

**> # 새로운 내용을 추가**

**> vec1 <- c(vec1,7)**

**> vec1**

**[1] 1 6 3 4 5 7**

**>**

**> # 위치를 지정하여 값 추가**

**> vec1[9] <- 9  # vec1값 9번째자리에 9추가 -> 7,8번째자리는 비어있기때문에 NA**

**> vec1**

**[1]  1  6  3  4  5  7 NA NA  9**

**>**

**> # append함수이용**

**> append(vec1,10,after=3)**

**[1]  1  6  3 10  4  5  7 NA NA  9**

**>**

**> vec2 <- c(1,2,3,3,4,5)**

**> vec2**

**[1] 1 2 3 3 4 5**

**>**

**> # 데이터를 중간에 추가**

**> append(vec2,11,after=3)  # vec2값 3번째요소 뒤에 11추가**

**[1]  1  2  3 11  3  4  5**

**> append(vec2,10,after=3)  # vec3값 3번째요소 뒤에 10추가 -> 11사라지고 10추가됨**

**[1]  1  2  3 10  3  4  5**

**> append(vec2,12,after=0)  # after=0은 가장 앞자리**

**[1] 12  1  2  3  3  4  5**

접기

**② 벡터로 연산하기**

**- 벡터는 여러건의 동일한 데이터가 들어있는 세트이므로 자체 연산 가능**

**- 집합이라고도 함**

**- union : 데이터형이 다를경우 두 집합의 합집합**

**- setdiff : 두 집합간의 차이 값 ex) setdiff(var1, var2) -> var1에는 있지만 var2에는 없는 요소 출력**

**- intersect : 두 집합간에 공통적으로 있는 요소 찾기 (교집합)**

접기

**> c(1,2,3)+c(4,5,6)**

**[1] 5 7 9**

**>**

**> c(1,2,3)+1**

**[1] 2 3 4**

**>**

**> c(1,2,3)+c(4,5,6,7)**

**[1] 5 7 9 8**

**경고메시지:**

**In c(1, 2, 3) + c(4, 5, 6, 7) :**

**두 객체의 길이가 서로 배수관계에 있지 않습니다**

**>**

**> var1 <- c(1,2,3)**

**> var2 <- c(4,5,6)**

**> var1+var2**

**[1] 5 7 9**

**>**

**> var3 <- c(3,4,'5')**

**> var1+var3**

**다음에 오류가 있습니다var1 + var3 : 이항연산자에 수치가 아닌 인수입니다**

**>**

**> union(var1,var3)**

**[1] "1" "2" "3" "4" "5"**

**>**

**> var4 <- c(1,2,3,4,5)**

**> var1**

**[1] 1 2 3**

**> var4**

**[1] 1 2 3 4 5**

**> var1+var4**

**[1] 2 4 6 5 7**

**경고메시지:**

**In var1 + var4 : 두 객체의 길이가 서로 배수관계에 있지 않습니다**

**>**

**> var1**

**[1] 1 2 3**

**> var2**

**[1] 4 5 6**

**> var-var2**

**다음에 오류가 있습니다var - var2 : 이항연산자에 수치가 아닌 인수입니다**

**> var1-var2**

**[1] -3 -3 -3**

**>**

**> setdiff(var1,var2)**

**[1] 1 2 3**

**> setdiff(var2,var1)**

**[1] 4 5 6**

**>**

**> intersect(var1,var2)**

**numeric(0)**

접기

**③ 벡터의 각 컬럼에 이름지정 : names( ) 함수 사용**

접기

**> fruits <- c(10,20,30)**

**> fruits**

**[1] 10 20 30**

**>**

**> names(fruits) <- c('apple','banana','peach')**

**> fruits**

**apple banana  peach**

**10     20     30**

접기

**④ 벡터에 연속적인 데이터 할당 : seq( ), rep( ) 함수 사용**

접기

**# seq( )**

**> var5 <- seq(1,5)**

**> var5**

**[1] 1 2 3 4 5**

**> var6 <- seq(2,-2)**

**> var6**

**[1]  2  1  0 -1 -2**

**> var7 <- seq(1,10,2)  # 1부터 10까지 2씩 증가시켜서 값을 출력**

**> var7**

**[1] 1 3 5 7 9**

**# rep( )**

**> var8 <- rep(1:3)**

**> var8**

**[1] 1 2 3**

**> var9 <- rep(1:3,2)  # 1부터3까지 두번출력**

**> var9**

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

**> var10 <- rep(1:3,each=2)  # 1부터 3까지 출력하되 각각2번씩 출력**

**> var10**

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

접기

**⑤ 벡터의 길이찾기**

**- length( )**

**- nrow( ) : 행렬일 경우 몇행인지 구하는 함수**

**- NROW( ) : 배열의 건수를 구할수 있음**

접기

**> var1**

**[1] 1 2 3**

**> length(var1)**

**[1] 3**

**> nrow(var1) # 행렬일 경우 몇행인지 구하는 함수**

**NULL**

**> NROW(var1) # 배열의 건수**

**[1] 3**

접기

**⑥ 벡터에 특정 문자의 포함여부 찾기**

**- 특정 데이터의 존재 유무를 검증하는 방법으로 많이 사용됨**

**- %in%**

접기

**> var7**

**[1] 1 3 5 7 9**

**> 3 %in% var7**

**[1] TRUE**

**> 4 %in% var7**

**[1] FALSE**

접기

**2. 행렬 (Matrix)**

**- matrix( )함수 사용**

**- 모든 컬럼과 행은 데이터형이 동일해야함**

**- 열 우선으로 입력됨**

**- nrow=n 값으로 행값입력**

**- byrow=T 가로로 입력을 우선으로**

**- rbind( ), cbind( ) : 행 또는 열 추가**

접기

**> mat1 <- matrix(c(1,2,3,4))**

**> mat1**

**[,1]**

**[1,]    1**

**[2,]    2**

**[3,]    3**

**[4,]    4**

**>**

**> mat2 <- matrix(c(1,2,3,4), nrow=2)**

**> mat2**

**[,1] [,2]**

**[1,]    1    3**

**[2,]    2    4**

**>**

**> mat3 <- matrix(c(1,2,3,4), nrow=2, byrow=T)**

**> mat3**

**[,1] [,2]**

**[1,]    1    2**

**[2,]    3    4**

접기

**① matrix의 데이터 조회**

접기

**> mat3**

**[,1] [,2]**

**[1,]    1    2**

**[2,]    3    4**

**>**

**> mat2[,1] #모든행의 1열값을 출력**

**[1] 1 2**

**>**

**> mat3[,1] #모든행의 1열값을 출력**

**[1] 1 3**

**>**

**> mat3[1,] #1행의 모든열값을 출력**

**[1] 1 2**

**>**

**> mat3[1,1] # 1행의 1열 값을 출력**

**[1] 1**

접기

**② 새로운 행과 열추가 : rbind( ), cbind( )**

접기

**> mat4 <- matrix(c(1,2,3,4,5,6,7,8,9), nrow=3, byrow=T)**

**> mat4**

**[,1] [,2] [,3]**

**[1,]    1    2    3**

**[2,]    4    5    6**

**[3,]    7    8    9**

**>**

**> # 컬럼의 이름 지정 : colnames**

**> colnames(mat4) <- c('First','Second','Third')**

**> mat4**

**First Second Third**

**[1,]     1      2     3**

**[2,]     4      5     6**

**[3,]     7      8     9**

**>**

**> # 행추가**

**> mat4 <- rbind(mat4, c(11,12,13))**

**> mat4**

**[,1] [,2] [,3]**

**[1,]    1    2    3**

**[2,]    4    5    6**

**[3,]    7    8    9**

**[4,]   11   12   13**

**>**

**> # 행길이가 다를경우 에러**

**> mat4 <- rbind(mat4, c(11,12,13,14))**

**경고메시지:**

**In rbind(mat4, c(11, 12, 13, 14)) :**

**number of columns of result is not a multiple of vector length (arg 2)**

**> mat4 <- cbind(mat4, c(9,9,9,9))**

**> mat4**

**[,1] [,2] [,3] [,4]**

**[1,]    1    2    3   9**

**[2,]    4    5    6   9**

**[3,]    7    8    9   9**

**[4,]   11   12   13  9**

접기

**3. list**

**- 키, 값 형태로 데이터를 저장하는 일종의 배열**

**- 데이터프레임의 기초**

**① List 생성하고 조회하기**

**- 특정키만 조회하고 싶을 경우는 변수이름$key 형식으로 조회**

접기

**> list1 <- list(name='jenny', address='Seoul', tel='010-xxxx-xxxx', pay=500)**

**> list1**

**$name**

**[1] "jenny"**

**$address**

**[1] "Seoul"**

**$tel**

**[1] "010-xxxx-xxxx"**

**$pay**

**[1] 500**

**> # 특정키만 조회하고 싶은 경우**

**> list1$name**

**[1] "jenny"**

**>**

**> list1[1:2]**

**$name**

**[1] "jenny"**

**$address**

**[1] "Seoul"**

접기