

# 같은 이름의 변수

- 📖 함수 내와 함수 밖에서의 사용
- 📖 계산 보기
- 📖 요약 정리

Lorem ipsum dolor sit amet, ius an molestie facilisi erroribus, mutat nalerum delectus ei vis. Has ornatus conclusionemque id, an videri molestatis sit. In etqui praesent sit. An vel agan porro comprehensan, ad ludus constituto nea, et ius utroque scaevola assuaverit.

Vis cu nodus nulla feugait, oratio facilisi in usu, eilit vitae sea te. Ea fabulas accusamus dissentias sea, facete tacinales definitiones et per. Nihil dicant mediocrem pro eu, no mei nostro sensibus platonem. Qui id sunno perpetua neglegentur. Vel ipsum novum copiosae ut. Quo et liber detracto probatus. Nam augue scribentur an. Sea oporteat percipitur incidere at. Qui viris nemore an.





# 함수 내와 함수 밖에서의 사용



## R-언어에서 함수 사용

- + 함수 작성 시 사용한 변수와 같은 이름의 변수를 사용하는 일이 의도적이든 우연이든 발생할 수 있으며 이는 어떤 에러도 발생시키지 않음

→ 함수 내와 함수 바깥에서 같은 이름의 변수를 사용할 때 이 둘은 R에서 서로 구분되어 덮어 쓰는 등의 문제가 발생하지 않음



## 함수 내에서 사용한 변수의 이름

함수 내에서 호출한 해당 이름의 변수가  
함수 내에 정의된 경우



함수 내에서 정의된 변수를 사용

함수 내에서 호출한 해당 이름의 변수가  
함수 내에 정의되지 않은 경우



함수 바깥에서 정의된 변수를 사용

함수 내 및 함수 바깥 양쪽에서 모두  
정의되지 않은 변수



상황에 따라 에러 메시지를 주거나 NULL  
등의 값을 얻음



# 계산 보기



함수 바깥에 변수 x1이 정의되고 함수 내엔 x1이 없는 경우

```
x1 <- seq(1,10)
myfcn <- function() {
  mx1 <- mean(x1)
  print(mx1)
}
```

\* 함수 내에서 전혀 정의되지 않은 x1이 함수 밖에서 정의된 x1을 사용하여 값을 계산함을 알 수 있음



계산 결과

```
> myfcn()
[1] 5.5
```



# 계산 보기



함수 내외에서 같은 이름의 변수가 모두 정의된 경우

```
myfcn2 <- function() {  
  x1 <- seq(1,100)  
  mx1 <- mean(x1)  
  print(mx1)  
}
```

\* 함수 바깥의 x1은 1:10

\* 함수 내의 x1은 1:100

\* 함수 내의 변수가 사용됨

+ 계산 결과

```
> myfcn2()  
[1] 50.5
```

\* 함수 내의 x1을 사용한  
평균값이 반환됨

+ x1

```
> x1  
[1] 1 2 3 4 5 6 7 8 9 10
```

\* 원래의 x1은 변함없음



# 요약 정리



## R에 내장된 함수

- + 함수 내에서 정의된 변수와 함수 바깥에서 정의된 변수의 이름이 같을 수 있음
- + 함수 내에서 정의된 변수와 함수 바깥에서 정의된 변수

함수 내에 정의된 변수



local variable(지역변수)

함수 바깥에서 정의된 변수



global variable(전역변수)

- + 지역변수와 전역변수는 이름이 같더라도 다른 번지(address)를 사용하므로 같은 이름에 다른 값이 저장됨
- + 함수 내에서는 함수 내에 정의된 변수가 우선 사용됨
- + 함수 내에서 정의된 변수가 없을 시 전역변수를 찾으며, 전역변수에도 정의된 것이 없으면 **에러 메시지가 발생됨**

# 매개변수의 기본값 설정

📖 매개변수의 이름 값 설정하기

📖 계산 보기

Loren ipsum dolor sit amet, ius an molestie facilisi erroribus, mutat nalerum delectus ei vis. Has ornatus conclusionemque id, an vide molestatis sit. In etqui praesent sit. An vel agan porro comprehensan, ad ludus constituto nea, et ius utroque scaevola assumaverit.

Vis cu nodus nulla feugait, oratio facilisi ex usu, eilit vitae sea te. Ea fabulas accusamus dissentias sea, facete tacinates definitiones et per. Nihil dicant mediocrem pro eu, no mei nostro sensibus platonem. Qui id sunno perpetua neglegantur. Vel ipsum novum copiosae ut. Quo et liber detracto probatus. Nam augue scribentur an. Sea oporteat percipitur incidereat ab. Qui viris nemore an.



# 매개변수의 이름 값 설정하기



함수 작성시 매개변수의 값을 설정하지 않은 경우



자주 사용하는 값(매개변수의 기본값)을 사용하도록 설정할 수 있음



매개변수의 기본값을 설정하고자 할 때는 매개변수의 이름에 값을 설정함



매개변수의 이름 값 설정 예시

```
myfcn <- function(x1=0) {  
  mx1 <- mean(x1)  
  print(mx1)  
}
```

➤ 매개변수 x1

- 사용자가 따로 설정하지 않으면 0이 사용됨
- 사용자가 값을 설정하면 설정된 값이 사용됨







# 계산 보기



자료로 사용하기 위해 x1을 1부터 100 사이의 자연수로 두기

```
x1 <- seq(1:100)
```



x1으로 위의 함수를 호출한 결과

```
> myfcn(x1)  
[1] 50.5
```



매개변수 없이 위의 함수를 호출한 결과

```
> myfcn()  
[1] 0
```

x1은 0 한 값을 사용하여 평균이 0으로 계산



# 고급활용

- 📖 기본 예제
- 📖 apply 함수에서 사용자 함수 사용하기
- 📖 계산 보기
- 📖 참고사항

Lorem ipsum dolor sit amet, ius an molestie facilisi erroribus, mutat natorum delectus ei vis. Has ornatus conclusionemque id, an vide maiestatis sit. In etqui praesent sit. An vel agan porro comprehensan, ad ludus constituto nea, et ius utroque scaevola assuaverit.

Vix cu nodus nulla feugait, oratio facilisi in usu, eili vitae sea te. Ea fabulas accusamus dissentias sea, facete tacinates definitiones et per. Nihil dicant mediocrem pro eu, no mei nostro sensibus platonem. Qui id sunno perpetua neglegantur. Vel ipsum novum copiosae ut. Quo et liber detracto probatus. Nam augue scriben- tur an. Sea oporteat percipitur incidereit ab. Qui viris nemore an.



## 사용할 자료

```
> score <- read.table(file="d:/hwp/STI/score.txt", header = TRUE,  
  fileEncoding="UTF-8", stringsAsFactors=FALSE)
```

+ 50명 학생의 국어, 영어, 수학, 과학 및 사회 5개 과목별 점수를 저장한 텍스트 파일을 읽어 옴

➤ 첫 6줄의 자료

```
> head(score)
```

	이름	국어	영어	수학	과학	사회
1	한대성	67	87	60	85	80
2	강준호	63	73	82	73	85
3	김종욱	74	53	76	72	58
4	박상호	55	65	49	84	96
5	김소현	76	69	69	92	70
6	김태성	59	98	70	77	80



# 기본 예제



## 과목별 가중치 적용하기



### 자료

각 과목별의 가중치를 적용하여 학생의 점수를 계산해서 이를 score의 마지막 열에 붙여 보자. 기존의 행렬 또는 데이터 프레임에 열을 추가하는 것은 cbind 함수를 사용할 수 있다. 과목별 가중치는 국영수는 각각 25%, 과학 및 사회는 각각 12.5% 가정한다.



### 사용 함수

```
myscore <- function(x) {  
  total <- x[,2]*.25 + x[,3]*.25 + x[,4]*.25 + x[,5]*0.125 + x[,6]*0.125  
  list (total = total)  
}
```

➤ 개별 학생에 대해서 가중평균을 구하므로 각 학생의 5개 과목 점수가 매개변수로 전달됨



# 기본 예제



## 과목별 가중치 적용하기



### cbind 함수 사용 결과

```
> cbind(score, myscore(score)$total)
```

	이름	국어	영어	수학	과학	사회	myscore(score)\$total
1	한대성	67	87	60	85	80	74.125
2	강준호	63	73	82	73	85	74.250
3	김종욱	74	53	76	72	58	67.000
4	박상호	55	65	49	84	96	64.750
	⋮						

- 매개변수 x는 학생, 1열은 학생 이름, 2열부터 6열까지가 과목별 점수
- score의 첫 번째 열은 학생의 이름이므로 두 번째 열부터 가중치를 줌
- score의 각 열은 행의 첨자를 생략하여 얻음

[예] x[,2] 는 x의 두 번째 열임

# apply 함수에서 사용자 함수 사용하기



## 가중 합산 점수 구하기



### 자료

다음의 함수는 매개변수  $x$ 가 각 학생의 5개 과목별 점수인 벡터이며(myfcn 함수의 경우 행렬을 직접 사용하였음) 각 학생의 점수에 대해서 가중 합산 점수를 구해보자.

## 함수의 계산 결과를 반환하는 return 함수 사용

```
myscore2 <- function(x) {  
  wt <- c(.25, .25, .25, .125, .125)  
  total <- sum(x * wt)  
  return(total)  
}
```



# apply 함수에서 사용자 함수 사용하기



## 가중 합산 점수 구하기

 apply 함수로 score의 각 학생별(행별) 점수 계산하기(첫 열은 문자열(이름)이므로 제외)

```
> apply(score[,-1], 1 , myscore2)
```

```
[1] 74.125 74.250 67.000 64.750 73.750 76.375 73.125 66.625 63.000 69.500  
[11] 67.500 75.875 73.500 70.250 67.125 67.750 74.375 69.625 66.750 70.750  
[21] 69.750 76.500 66.125 74.375 61.125 73.375 78.000 78.000 70.000 70.000  
[31] 69.625 60.000 79.750 75.750 72.875 70.000 72.375 70.625 79.625 70.000  
[41] 80.000 74.875 71.250 80.250 69.500 70.625 72.875 79.000 67.250 69.125
```



# apply 함수에서 사용자 함수 사용하기



## 가중 합산 점수 구하기

### 총점 구하기(필요한 경우)

```
> score$총점 <- apply(score[,-1], 1 , myscore2)
```

```
> head(score)
```

	이름	국어	영어	수학	과학	사회	총점
1	한대성	67	87	60	85	80	74.125
2	강준호	63	73	82	73	85	74.250
3	김종욱	74	53	76	72	58	67.000
4	박상호	55	65	49	84	96	64.750
5	김소현	76	69	69	92	70	73.750
6	김태성	59	98	70	77	80	76.300







# 계산 보기



## 표준화 점수를 얻기 위한 두 가지 방법



### 자료

score에서 각 과목의 표준화점수를 계산하여 zscore라는 데이터 프레임을 만드는 함수를 작성해 보자.



### 변수의 표준화

$$\text{표준화 변수값} = (\text{원자료값} - \text{평균}) / \text{표준편차}$$



# 계산 보기



## 표준화 점수를 얻기 위한 두 가지 방법



### 방법 1 : zfcn 이란 함수 작성하기

```
zfcn <- function(x) {  
  zscore <- x # 원래 자료의 이름 등을 그대로  
  ncol <- dim(x)[2] # x의 열의 수  
  for (i in 2:ncol) {  
    mx <- mean(x[,i])  
    stdev <- sd(x[,i])  
    zscore[,i] <- ( x[,i] - mx ) /stdev  
  } # end for  
  list(zscore=zscore)  
} # end function
```



# 계산 보기



## 표준화 점수를 얻기 위한 두 가지 방법



### 방법 1 : zfcfn이란 함수 작성하기

> zfcfn(score)

> zfcfn(score)

\$zscore

이름	국어	영어	수학	과학	사회	총점
1 한대성	-0.04965159	1.04209030	-0.62692387	0.51253665	0.44027414	0.53190800
2 강준호	-0.43158690	-0.21128291	1.77592022	-0.68404697	0.92730305	0.55795638
3 김종욱	0.61873520	-2.00181607	1.12059910	-0.78376227	-1.70265308	-0.95284988
4 박상호	-1.19545751	-0.92749617	-1.82834591	0.41282135	1.99876666	-1.42172079
5 김소현	0.80970285	-0.56938954	0.35605780	1.21054377	-0.53378369	0.45376285
6 김태성	-0.81352220	2.02688353	0.46527799	-0.28518576	0.44027414	1.00077891

score를 매개변수로 하여 호출한 결과

- 매개변수 x로 학생별 과목별 점수가 저장된 데이터 프레임을 전달받으며 첫 번째 열은 이름임
- ncol은 학생 과목 수를 얻기 위해 dim 함수의 결과 중 두 번째 원소를 얻었음
- 각각의 열에 대해서(이름이 1번 열이므로 2번 열부터 for를 사용) 평균과 분산을 계산하고 zscore의 값은 평균으로 뺀 후 표준편차로 나눈 값으로 저장함





# 계산 보기



## 표준화 점수를 얻기 위한 두 가지 방법



### 방법 2 : 벡터에 대해서 표준화하는 함수 zfcn2를 작성하기

```
zfcn2 <- function(x) {  
  zscore <- (x-mean(x))/sd(x)  
  return(zscore)  
}
```

➤ 벡터를 전달받아 각각의 값에 평균을 빼고 표준편차로 나누는 간단한 함수임



# 계산 보기



## 표준화 점수를 얻기 위한 두 가지 방법



### 방법 2 : 벡터에 대해서 표준화하는 함수 zfcn2를 작성하기

> apply(score[,-1], 2, zfcn2)

apply 함수를 사용하되 각 열별로 적용한 결과

	국어	영어	수학	과학	사회	총점
[1,]	-0.04965159	1.04209030	-0.62692387	0.51253665	0.44027414	0.53190800
[2,]	-0.43158690	-0.21128291	1.77592022	-0.68404697	0.92730305	0.55795638
[3,]	0.61873520	-2.00181607	1.12059910	-0.78376227	-1.70265308	-0.95284988
[4,]	-1.19545751	-0.92749617	-1.82834591	0.41282135	1.99876666	-1.42172079
[5,]	0.80970285	-0.56938954	0.35605780	1.21054377	-0.53378369	0.45376285
[6,]	-0.81352220	2.02688353	0.46527799	-0.28518576	0.44027414	1.00077891





# 계산 보기



## 표준화 점수를 얻기 위한 두 가지 방법



### 방법 2 : 벡터에 대해서 표준화하는 함수 zfcn2를 작성하기

이름을 포함한 새로운 데이터 프레임 작성 결과

```
> zscore <- data.frame( score[,1], apply(score[,-1], 2, zfcn2) )
```

```
> head(zscore)
```

score...1.	국어	영어	수학	과학	사회	총점
1 한대성	-0.04965159	1.04209030	-0.62692387	0.51253665	0.44027414	0.53190800
2 강준호	-0.43158690	-0.21128291	1.77592022	-0.68404697	0.92730305	0.55795638
3 김종욱	0.61873520	-2.00181607	1.12059910	-0.78376227	-1.70265308	-0.95284988
4 박상호	-1.19545751	-0.92749617	-1.82834591	0.41282135	1.99876666	-1.42172079
5 김소현	0.80970285	-0.56938954	0.35605780	1.21054377	-0.53378369	0.45376285
6 김태성	-0.81352220	2.02688353	0.46527799	-0.28518576	0.44027414	1.00077891





# 참고사항



이름의 score...1을 바꾸려면?



names 함수 사용

**names(zscore)**

[1] "score...1." "국어" "영어" "수학" "과학" "사회" "총점"

> **names(zscore)[1] <- "이름"**

> **head(zscore)**

이름	국어	영어	수학	과학	사회	총점
1 한대성	-0.04965159	1.0420903	-0.6269239	0.5125367	0.4402741	0.5319080
2 강준호	-0.43158690	-0.2112829	1.7759202	-0.6840470	0.9273031	0.5579564
3 김종욱	0.61873520	-2.0018161	1.1205991	-0.7837623	-1.7026531	-0.9528499
4 박상호	-1.19545751	-0.9274962	-1.8283459	0.4128214	1.9987667	-1.4217208
5 김소현	0.80970285	-0.5693895	0.3560578	1.2105438	-0.5337837	0.4537628
6 김태성	-0.81352220	2.0268835	0.4652780	-0.2851858	0.4402741	1.0007789





# 참고사항



이름의 score...1을 바꾸려면?



**주의** : names 함수 사용 시 data.frame 자리에 cbind를 사용하면 모든 자료가 문자열로 변경됨

cbind나 rbind 함수가 행렬(matrix 함수)에 대한 함수이며, 행렬은 데이터 프레임과 달리 모든 원소가 같은 형식의 값을 가져야 함

→ 한 값이라도 문자열이면 모든 자료가 문자열로 변환됨