Functions - Pre-Class Work

Nour Audi

October 1, 2017

Standardizing a variable means subtracting the mean, and then dividing by the standard deviation. Let’s use a loop to standardize the numeric columns in the [Western Collaborative Group Study](https://clinicaltrials.gov/ct2/show/NCT00005174). This study began in 1960 with 3154 men ages 39-59, who were employed in one of 11 California based companies. They were followed until 1969 during this time, 257 of these men developed coronary heart disease (CHD). You can read this data in with the code below. You can access this dataset with the following code:

suppressMessages(library(foreign))  
wcgs <- read.dta("~/Documents/MPH Brown /PHP 1560 /assignment/functions-pre-class-Nour-Audi/wcgs2.dta")

The data has the following variables:

WCGS has the following variables:

|  |  |
| --- | --- |
| Name | Description |
| id | Subject identification number |
| age | Age in years |
| height | Height in inches |
| weight | Weight in lbs. |
| sbp | Systolic blood pressure in mm |
| dbp | Diastolic blood pressure in mm Hg |
| chol | Fasting serum cholesterol in mm |
| behpat | Behavior |
| 1 | A1 |
| 2 | A2 |
| 3 | B3 |
| 4 | B4 |
| ncigs | Cigarettes per day |
| dibpat | Behavior |
| 1 | type A |
| 2 | type B |
| chd69 | Coronary heart disease |
| 1 | Yes |
| 0 | no |
| typechd | Type of CHD |
| 1 | myocardial infarction or death |
| 2 | silent myocardial infarction |
| 3 | angina perctoris |
| time169 | Time of CHD event or end of follow-up |
| arcus | Arcus senilis |
| 0 | absent |
| 1 | present |
| bmi | Body Mass Index |

### Question 1: Standardize Function

A. Create a function called standardize.me() that takes a numeric vector as an argument, and returns the standardized version of the vector.

standardize.me <- function(x) {  
  
 (x - mean(x)) / sd(x)  
}

B. Assign all the numeric columns of the original WCGS dataset to a new dataset called WCGS.new.

WCGS.new <- wcgs [c("age","height","weight","sbp","dbp","chol","ncigs", "time169", "bmi" )]

C. Using a loop and your new function, standardize all the variables WCGS.new dataset.

for (i in seq\_along(WCGS.new)) {  
WCGS.new[,i] <- standardize.me(WCGS.new[ , i])  
   
}

D. What should the mean and standard deviation of all your new standardized variables be? Test your prediction by running a loop

for(i in seq\_along(WCGS.new)) {  
 print(paste("The mean of column number", i, "is", round(mean(WCGS.new[,i]), 2), "and the standard deviation is", round(sd(WCGS.new[,i]), 2)))  
}

## [1] "The mean of column number 1 is 0 and the standard deviation is 1"  
## [1] "The mean of column number 2 is 0 and the standard deviation is 1"  
## [1] "The mean of column number 3 is 0 and the standard deviation is 1"  
## [1] "The mean of column number 4 is 0 and the standard deviation is 1"  
## [1] "The mean of column number 5 is 0 and the standard deviation is 1"  
## [1] "The mean of column number 6 is NA and the standard deviation is NA"  
## [1] "The mean of column number 7 is 0 and the standard deviation is 1"  
## [1] "The mean of column number 8 is 0 and the standard deviation is 1"  
## [1] "The mean of column number 9 is 0 and the standard deviation is 1"

# didn't understand well:(

### Question 2: Looping to Calculate

A. Using a loop, calculate the mean weight of the subjects separated by the type of CHD they have.

typchd69\_weight.df <- data.frame("typchd69" = unique(wcgs$typchd69), "mean\_weight"=NA, stringsAsFactors = F)  
  
for(i in 1:nrow(typchd69\_weight.df)) {  
 typchd69.i <- typchd69\_weight.df$typchd69[i]  
 mean\_weight.i <- mean(subset(wcgs, typchd69 == typchd69.i)$weight)   
 typchd69\_weight.df$mean\_weight[i] <- mean\_weight.i  
}  
  
typchd69\_weight.df # answer was learned form others work. what i tried to do:

## typchd69 mean\_weight  
## 1 no CHD 169.5537  
## 2 MI or SD 172.6444  
## 3 silent MI 176.3521  
## 4 angina 176.6471

#wcgs.wCHD <- wcgs[c("typchd69", "weight")]  
 #for (i in 1:length(wcgs.wCHD[weight]))

B. Now do the same thing, but now don’t use a loop

wcgs %>%  
 group\_by(typchd69) %>%  
 summarise(mean(weight))  
  
## or using, aggregate(weight ~ typchd69, data = wcgs, FUN = mean)