Functions - Pre-Class Work

Adam J Sullivan

September 29, 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("https://drive.google.com/uc?export=download&id=0B8CsRLdwqzbzYWxfN3ExQllBQkU")

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.

## Version 1 using map function  
standardize.me<-function(x){  
 avg<- map\_dbl(x,mean,na.rm=TRUE)  
 denominator<-map(x,sd,na.rm=TRUE)  
 answer<-(x-avg)/denominator  
 return(answer)  
}  
  
## version 2 using traditional subsetting  
standardize.me2<-function(x){  
 answer<-matrix(ncol=ncol(x),nrow=nrow(x))  
  
 for (i in 1:ncol(x)){  
 answer[,i]<-(x[,i]- colMeans(x[i],na.rm=TRUE))/sd(x[,i],na.rm=TRUE)  
 }  
return(answer)  
}

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

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.3.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

WCGS.new<-  
 wcgs%>%  
 select(age,height,weight,sbp,dbp,chol,ncigs,bmi)

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

library(purrr)

## Warning: package 'purrr' was built under R version 3.3.3

x=WCGS.new  
new\_standardize<-matrix(ncol=ncol(x),nrow=nrow(x))  
for ( i in 1:ncol(x)){  
 new\_standardize[i]<-standardize.me(x[i])  
   
}

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

y<-map\_dbl(new\_standardize,mean,na.rm=TRUE)  
 mean\_standardize<-mean(y,na.rm=TRUE)  
 new\_standardize<-as.data.frame(new\_standardize)  
 z<-map\_dbl(new\_standardize,sd,na.rm=TRUE)  
 sd\_standardized<-mean(z,na.rm=TRUE)  
 print(mean\_standardize)

## [1] 2.36691e-16

print(sd\_standardized)

## [1] 1

# I used the map functions because I wanted to test out my knowledge of th functions

### Question 2: Looping to Calculate

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

no\_chd<-numeric(0)  
MI\_SD<-numeric(0)  
silent\_MI<-numeric(0)  
angina<-numeric(0)  
  
no\_chd\_weight<-vector()  
MI\_SD\_weight<-vector()  
silent\_weight<-vector()  
angina\_weight<-vector()  
  
for (i in 1 :nrow(wcgs)){  
 if(wcgs$typchd69[i]=="no CHD"){   
 no\_chd\_weight[i]<- wcgs$weight[i]  
 no\_chd<-mean(no\_chd\_weight,na.rm=TRUE)  
  
 }else if (wcgs$typchd69[i]=="MI or SD") {  
 MI\_SD\_weight[i]<-wcgs$weight[i]  
 MI\_SD<-mean(MI\_SD\_weight,na.rm=TRUE)  
 } else if (wcgs$typchd69[i]=="silent MI"){  
 silent\_weight[i]<-wcgs$weight[i]  
 silent\_MI <- mean(silent\_weight,na.rm=TRUE)  
 } else if ( wcgs$typchd69[i]=="angina")  
 angina\_weight[i]<-wcgs$weight[i]  
 angina<-mean(angina\_weight,na.rm=TRUE)  
}  
print(c(no\_chd,MI\_SD,silent\_MI,angina))

## [1] 169.5537 172.6444 176.3521 176.6471

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

wcgs %>%  
 group\_by(typchd69) %>%  
 summarise(average= mean(weight,na.rm=TRUE))

## # A tibble: 4 × 2  
## typchd69 average  
## <fctr> <dbl>  
## 1 no CHD 169.5537  
## 2 MI or SD 172.6444  
## 3 silent MI 176.3521  
## 4 angina 176.6471