Supplemental Digital Content 5:

1. **Example\_BBS\_final.R**

# Set up the computing environment: ----------------------------------------------

# Set the working directory. This should have the files example\_PBS.R,

# assign\_score\_single\_person.R, assign\_score\_multiple\_people.R,

# and score\_functions.rda

wd = "C://Users//your directory//" #You'll have to change this and all 4 files should be dowloaded in this same path/folder

setwd(wd)

source("assign\_score\_single\_person.R")

source("assign\_score\_multiple\_people.R")

load("score\_functions.rda")

################################################################################

# This assigns a physical activity score to a single

# person. Users can input values for all the physical behaviors and see the total

# score as well as the contribution to the total score by the 8 physical behaviors

################################################################################

# Put in the observed values of each physical behavior: ----------------------------

# Median in the NIH-AARP dataset

moderate.exercise = 8 # Moderate exercise (MET-hr/wk)

vigorous.exercise = 3 # Vigorous exercise (MET-hr/wk)

light.intensity.household = 3.5 # Light-intensity Household (MET-hr/wk)

moderate.vigorous.household = 3 # Moderate-vigorous Household (MET-hr/wk)

weight.training = 1 # Weight Training (MET-hr/wk)

sitting.other.than.tv = 5 # Sitting other than TV (hr/day)

television.sitting = 5 # Television sitting (hr/day)

sleep = 8 # Sleep (hr/day)

# After inputing reported physical behaviors above, run the calculate\_score function

# which will calculate the PBS and print the results to the console

total.score.median = calculate\_score(moderate.exercise = moderate.exercise, vigorous.exercise = vigorous.exercise,

light.intensity.household = light.intensity.household,

moderate.vigorous.household = moderate.vigorous.household,

weight.training = weight.training, sitting.other.than.tv = sitting.other.than.tv,

television.sitting = television.sitting, sleep = sleep)

################################################################################

# Begin specific cases of physical activity that are of interest:

# 2018 recommended levels, an inactive person and active with high sitting

###############################################################################

#Put in the physical behaviors at recommended levels ----------------------------

# Get physical behavior score for 2018 recommended physical behviors: -------------------

moderate.exercise = 7.5 # Moderate exercise (MET-hr/wk)

vigorous.exercise = 0 # Vigorous exercise (MET-hr/wk)

light.intensity.household = 2.5 # Light-intensity Household (MET-hr/wk)

moderate.vigorous.household = 2 # Moderate-vigorous Household (MET-hr/wk)

weight.training = 0 # Weight Training (MET-hr/wk)

sitting.other.than.tv = 3.0 # Sitting other than TV (hr/day)

television.sitting = 3 # Television sitting (hr/day)

sleep = 7 # Sleep (hr/day)

total.score.recommended = calculate\_score(moderate.exercise = moderate.exercise, vigorous.exercise = vigorous.exercise,

light.intensity.household = light.intensity.household,

moderate.vigorous.household = moderate.vigorous.household,

weight.training = weight.training, sitting.other.than.tv = sitting.other.than.tv,

television.sitting = television.sitting, sleep = sleep)

#Put in the physical behaviors for inactive person ----------------------------

moderate.exercise = 0 # Moderate exercise (MET-hr/wk)

vigorous.exercise = 0 # Vigorous exercise (MET-hr/wk)

light.intensity.household = 2 # Light-intensity Household (MET-hr/wk)

moderate.vigorous.household = 2 # Moderate-vigorous Household (MET-hr/wk)

weight.training = 0 # Weight Training (MET-hr/wk)

sitting.other.than.tv = 10 # Sitting other than TV (hr/day)

television.sitting = 7 # Television sitting (hr/day)

sleep = 6 # Sleep (hr/day)

total.score.inactive = calculate\_score(moderate.exercise = moderate.exercise, vigorous.exercise = vigorous.exercise,

light.intensity.household = light.intensity.household,

moderate.vigorous.household = moderate.vigorous.household,

weight.training = weight.training, sitting.other.than.tv = sitting.other.than.tv,

television.sitting = television.sitting, sleep = sleep)

# Put in the physical behaviors for active person but high sitting ----------------------------

moderate.exercise = 7.5 # Moderate exercise (MET-hr/wk)

vigorous.exercise = 0 # Vigorous exercise (MET-hr/wk)

light.intensity.household = 2.5 # Light-intensity Household (MET-hr/wk)

moderate.vigorous.household = 2 # Moderate-vigorous Household (MET-hr/wk)

weight.training = 0 # Weight Training (MET-hr/wk)

sitting.other.than.tv = 5 # Sitting other than TV (hr/day)

television.sitting = 9 # Television sitting (hr/day)

sleep = 6 # Sleep (hr/day)

total.score.high.sit = calculate\_score(moderate.exercise = moderate.exercise, vigorous.exercise = vigorous.exercise,

light.intensity.household = light.intensity.household,

moderate.vigorous.household = moderate.vigorous.household,

weight.training = weight.training, sitting.other.than.tv = sitting.other.than.tv,

television.sitting = television.sitting, sleep = sleep)

################################################################################

# This assigns a physical activity score to a a dataset with multiple subjects

# Users can input values for all the physical behaviors and see the total

# score as well as the contribution to the total score by the 8 physical behaviors

###############################################################################

# Make an dataset of 20 people

moderate.exercise = runif(20) # Moderate exercise (MET-hr/wk)

vigorous.exercise = runif(20) # Vigorous exercise (MET-hr/wk)

light.intensity.household = runif(20) # Light-intensity Household (MET-hr/wk)

moderate.vigorous.household = runif(20) # Moderate-vigorous Household (MET-hr/wk)

weight.training = runif(20) # Weight Training (MET-hr/wk)

sitting.other.than.tv = runif(20) # Sitting other than TV (hr/day)

television.sitting = runif(20) # Television sitting (hr/day)

sleep = runif(20) # Sleep (hr/day)

# Assign scores to each of the 20 people

# Replace the variables in the function call with your data

score.matrix = assign\_scores\_multiple\_people(moderate.exercise = moderate.exercise,

vigorous.exercise = vigorous.exercise,

light.intensity.household = light.intensity.household,

moderate.vigorous.household = moderate.vigorous.household,

weight.training = weight.training, sitting.other.than.tv = sitting.other.than.tv,

television.sitting = television.sitting, sleep = sleep)

print(score.matrix) # Show results

print(score.matrix$total.score) # Show the total score

hist(score.matrix$total.score)

1. **Assign\_score\_single\_person. R**

**\*\*Save this fine in the same folder but you do not have to open or modify.**

calculate\_score = function(moderate.exercise, vigorous.exercise, light.intensity.household,

moderate.vigorous.household, weight.training,

sitting.other.than.tv, television.sitting, sleep){

# Calculates the total Physical Behavior Score and the score for all the individual components

# for a single person.

#

# Args:

# moderate.exercise: reported moderate exercise in MET-hr/wk

# vigorous.exercise: reported vigorous exercise in MET-hr/wk

# light.intensity.household: reported light-intensity Household in MET-hr/wk

# moderate.vigorous.household: reported moderate-vigorous Household in MET-hr/wk

# weight.training: reported weight Training in MET-hr/wk

# sitting.other.than.tv: reported sitting other than TV in hr/day

# television.sitting: reported television sitting in hr/day

# sleep: reported leep (hr/day)

#

# Returns:

# the total physical behavior score

# Prints total Physical Behavior Score and score by component

moderate.score = score\_functions$`Moderate Activity`(moderate.exercise)

vigorous.score = score\_functions$`Vigorous Activity`(vigorous.exercise)

light.household.score = score\_functions$`Light Household Activity`(light.intensity.household)

moderate.vigorous.household.score = score\_functions$`MVPA Household Activity`(moderate.vigorous.household)

weight.training.score = score\_functions$`Weight Training`(weight.training)

sitting.not.TV.score = score\_functions$`Hours Sitting Other than TV`(sitting.other.than.tv)

tv.sitting.score = score\_functions$`Hours of TV Sitting`(television.sitting)

sleep.score = score\_functions$`Hours of Sleep`(sleep)

total.PBS = sum(moderate.score, vigorous.score, light.household.score,

moderate.vigorous.household.score,

weight.training.score, sitting.not.TV.score, tv.sitting.score,

sleep.score)

# Print the total score, rounded to two decimal places

print(paste("Total PBS Score is:", round(total.PBS,2)))

print("-------------------------------")

# Print the individual score components

print("This score is made up of:")

print(paste("Moderate Activity:", round(moderate.score,2)))

print(paste("Vigorous Activity:", round(vigorous.score,2)))

print(paste("Light Household Activity:", round(light.household.score,2)))

print(paste("Moderate-Vigorous Household Activity:",

round(moderate.vigorous.household.score,2)))

print(paste("Weight Training:",

round(weight.training.score,2)))

print(paste("Sitting other than TV:", round(sitting.not.TV.score,2)))

print(paste("Television sitting:", round(tv.sitting.score,2)))

print(paste("Sleep:", round(sleep.score,2)))

# Get health outcomes associated with this score -------------

# This is the coefficient for the PBS in a Cox Model using the NIH-AARP dataset from the paper

cox.coefficient = -0.02742

# This is the median score assigned to people in the NIH-AARP Dataset

median.score = 76.41301

fifth.quantile = 53.5

relative.hazard.ratio.at.zero = exp(cox.coefficient \* (total.PBS - fifth.quantile))

relative.hazard.ratio.at.median = exp(-0.02742 \* (total.PBS - median.score))

print(paste("This score has a hazard ratio of", round(relative.hazard.ratio.at.zero,2),

"when compared to a score at the 5th percentile (PBS = 53.5)"))

print(paste("This score has a hazard ratio of", round(relative.hazard.ratio.at.median,2),

"when compared to the median score in the NIH-AARP dataset"))

return(total.PBS)

}

1. Assign\_scores\_multiple\_people.R

assign\_scores\_multiple\_people = function(moderate.exercise, vigorous.exercise,

light.intensity.household,

moderate.vigorous.household, weight.training,

sitting.other.than.tv, television.sitting, sleep){

# Calculates the total Physical Behavior Score and the score for all the individual components

# for a dataset of physical activity measurements

#

# Args:

# moderate.exercise: vector of reported moderate exercise in MET-hr/wk

# vigorous.exercise: vector of reported vigorous exercise in MET-hr/wk

# light.intensity.household: vector of reported light-intensity Household in MET-hr/wk

# moderate.vigorous.household: vector of reported moderate-vigorous Household in MET-hr/wk

# weight.training: vector of reported weight Training in MET-hr/wk

# sitting.other.than.tv: vector of reported sitting other than TV in hr/day

# television.sitting: vector of reported television sitting in hr/day

# sleep: vector of reported leep (hr/day)

#

# Returns:

# A (n x 9) data frame. Columns 1 through 8 are the contribution to the total score

# by each component. Column 9 is the total physical behavior score.

moderate.score = score\_functions$`Moderate Activity`(moderate.exercise)

vigorous.score = score\_functions$`Vigorous Activity`(vigorous.exercise)

light.household.score = score\_functions$`Light Household Activity`(light.intensity.household)

moderate.vigorous.household.score = score\_functions$`MVPA Household Activity`(moderate.vigorous.household)

weight.training.score = score\_functions$`Weight Training`(weight.training)

sitting.not.TV.score = score\_functions$`Hours Sitting Other than TV`(sitting.other.than.tv)

tv.sitting.score = score\_functions$`Hours of TV Sitting`(television.sitting)

sleep.score = score\_functions$`Hours of Sleep`(sleep)

scores = cbind(moderate.score, vigorous.score, light.household.score, moderate.vigorous.household.score,

weight.training.score, sitting.not.TV.score, tv.sitting.score,sleep.score)

scores = cbind(scores, "total.score" = rowSums(scores))

scores = as.data.frame(scores)

return(scores)

}