ESM 262 assignment 3- SJKF

Jessica Copeland

2025-03-17

I am recycling my SJKF function with a few improvements, since I am working solo.

# Here is the function I created, which is also saved as a separate R file.   
  
#' San Joaquin Kit Fox weights and home ranges (an almost correct function)  
#'  
#' @param W is weight of the fox in pounds.   
#' @param G is gender of the fox. The only options are: 1 for female, 2 for male  
#' @param HR\_minimum is the minimum home range size of the fox in meters, in this case always 1000. #  
#' @returns HR\_indiv The home range for the individual in meters.   
  
  
SJKF\_range=function(W,G,HR\_minimum=1000,HR\_indiv) {  
  
 if(W<0) W=NA # Weight should not be less than zero  
   
 # Using ifelse() to set NA for values that are not 1 or 2, because we can only have 1 and 2 to specify the genders of the animals.   
 G <-ifelse(G != 1 & G != 2, NA, G)  
  
 # return("Home range cannot be less than 0")  
 if (HR\_minimum < 0) return("Home range cannot be less than 0")  
  
 # Calculate the home range of the individual SJKF   
 result <- (W \* G) + HR\_minimum  
  
return(result)  
}

I have this function saved as # SJKF\_func.R #

# I will bring in the function now  
  
source("C:/Users/jcope/OneDrive/Desktop/UCSB/R/SJKF\_func.R")

Let’s generate some data using a for loop!

# Lets say I want to generate 10 different weights..   
num\_weights <- 10  
  
# I want to create a vector (empty) to store the weights.   
weights <- numeric(num\_weights)  
  
# Then I can use a "for loop" to generate random weights  
set.seed(123) # Ensures reproducibility  
for (i in 1:num\_weights) {  
 weights[i] <- runif(1, min = 3, max = 15) # Generate a random weight  
}  
  
# Print the generated weights  
print(weights)

## [1] 6.450930 12.459662 7.907723 13.596209 14.285607 3.546678 9.337266  
## [8] 13.709029 9.617220 8.479377

Cool that gave me 10 different weights. Now I want to try some more looping with purrr, but I want it to give me whole integers instead.

library(purrr)  
  
set.seed(123)  
  
# Generate 10 random integer weights between 3 and 15  
weights <- map\_int(1:10, ~ sample(3:15, 1))  
  
print(weights)

## [1] 5 5 12 4 8 13 7 6 8 11

So now that I have weights, I will try and plug each one into the function…

#First I want to use these numbers and assume they are all females..  
SJKF\_range(G=1, W=5)

## [1] 1005

SJKF\_range(G=1, W=5)

## [1] 1005

SJKF\_range(G=1, W=12)

## [1] 1012

SJKF\_range(G=1, W=4)

## [1] 1004

SJKF\_range(G=1, W=8)

## [1] 1008

SJKF\_range(G=1, W=13)

## [1] 1013

SJKF\_range(G=1, W=7)

## [1] 1007

SJKF\_range(G=1, W=6)

## [1] 1006

SJKF\_range(G=1, W=8)

## [1] 1008

SJKF\_range(G=1, W=11)

## [1] 1011

Now if I do the same thing assuming they are all males….

SJKF\_range(G=2, W=5)

## [1] 1010

SJKF\_range(G=2, W=5)

## [1] 1010

SJKF\_range(G=2, W=12)

## [1] 1024

SJKF\_range(G=2, W=4)

## [1] 1008

SJKF\_range(G=2, W=8)

## [1] 1016

SJKF\_range(G=2, W=13)

## [1] 1026

SJKF\_range(G=2, W=7)

## [1] 1014

SJKF\_range(G=2, W=6)

## [1] 1012

SJKF\_range(G=2, W=8)

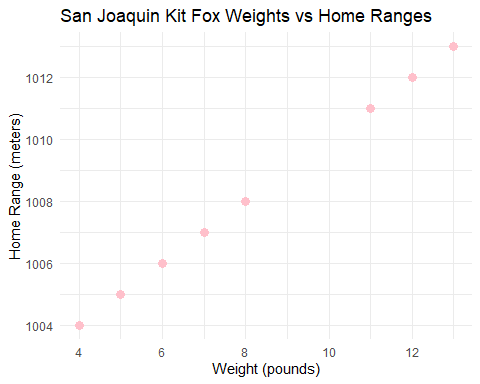
## [1] 1016

SJKF\_range(G=2, W=11)

## [1] 1022

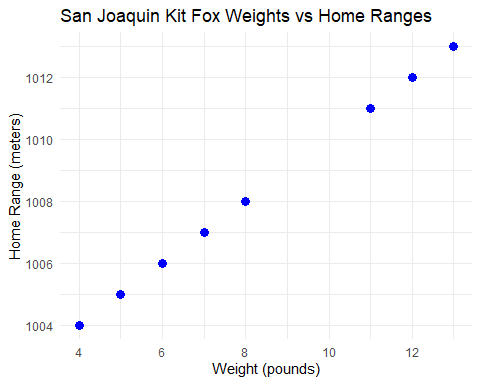
Now that I have inputs and outputs, I can attempt to plot these…

# Females first  
  
# Load ggplot2  
library(ggplot2)  
  
# Data from females  
data\_f <- data.frame(  
 Weight = c(5, 5, 12, 4, 8, 13, 7, 6, 8, 11), # Weights  
 HomeRange = c(1005, 1005, 1012, 1004, 1008, 1013, 1007, 1006, 1008, 1011), # Corresponding home ranges  
 Gender= "Female"  
)  
  
# Create a scatter plot  
ggplot(data\_f, aes(x = Weight, y = HomeRange)) +  
 geom\_point(color = "pink", size = 3) + # Add points  
 labs(  
 title = "San Joaquin Kit Fox Weights vs Home Ranges",  
 x = "Weight (pounds)",  
 y = "Home Range (meters)"  
 ) +  
 theme\_minimal() # Use a clean theme

 Well, this ends up being super boring because of how I set up my function, oops!

I will try the males anyway…

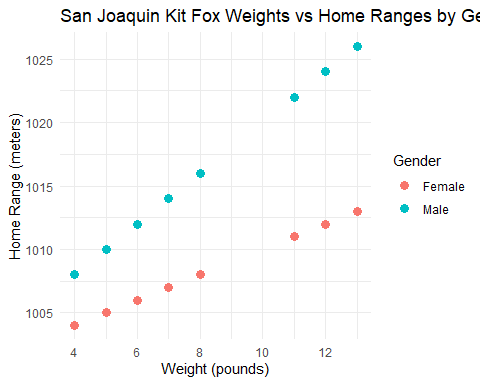
# Now males...   
  
data\_m <- data.frame(  
 Weight = c(5, 5, 12, 4, 8, 13, 7, 6, 8, 11), # Weights  
 HomeRange = c(1010, 1010, 1024, 1008, 1016, 1026, 1014, 1012, 1016, 1022), # Corresponding home ranges  
 Gender="Male"  
)  
  
# Create a scatter plot  
ggplot(data\_f, aes(x = Weight, y = HomeRange)) +  
 geom\_point(color = "blue", size = 3) + # Add points  
 labs(  
 title = "San Joaquin Kit Fox Weights vs Home Ranges",  
 x = "Weight (pounds)",  
 y = "Home Range (meters)"  
 ) +  
 theme\_minimal() # Use a clean theme

 I realized I want to combine my two datasets…

# Combine the two datasets..now I can plot beter (hopefully)  
data\_combined <- rbind(data\_m, data\_f)

Now plot!!

ggplot(data\_combined, aes(x = Weight, y = HomeRange, color = Gender)) +  
 geom\_point(size = 3) + # Add points, color-coded by gender  
 labs(  
 title = "San Joaquin Kit Fox Weights vs Home Ranges by Gender",  
 x = "Weight (pounds)",  
 y = "Home Range (meters)"  
 ) +  
 theme\_minimal()



Now I want to try running some tests of my function with the test file…

library(testthat)

##   
## Attaching package: 'testthat'

## The following object is masked from 'package:purrr':  
##   
## is\_null

testthat::test\_file("C:/Users/jcope/OneDrive/Desktop/UCSB/R/sjkf\_test.R")

## [ FAIL 0 | WARN 0 | SKIP 0 | PASS 0 ][ FAIL 0 | WARN 0 | SKIP 0 | PASS 1 ][ FAIL 0 | WARN 0 | SKIP 0 | PASS 2 ][ FAIL 0 | WARN 0 | SKIP 0 | PASS 3 ][ FAIL 0 | WARN 0 | SKIP 0 | PASS 4 ][ FAIL 0 | WARN 0 | SKIP 0 | PASS 5 ][ FAIL 0 | WARN 0 | SKIP 0 | PASS 6 ][ FAIL 0 | WARN 0 | SKIP 0 | PASS 7 ][ FAIL 0 | WARN 0 | SKIP 0 | PASS 8 ]

#remember where you saved that test file!

I got 0 fail, 0 warn, 0 skip, and 8 pass… which means that a total of 8 tests were conducted, and they all passed.

So my function should be working just fine!