FishR 101 – Functions Part 2

Homework Answers

Instructor: Emily Markowitz (Emily.Markowitz@noaa.gov)

February 10, 2021

Contents

First, load your libraries!

```
library(tidyverse)
library(here)
library(janitor)
```

Answers to Questions:

NOTE: For this homework, also refer to the PDF version. There are helpful graphics included in the PDF version!

1 Functions

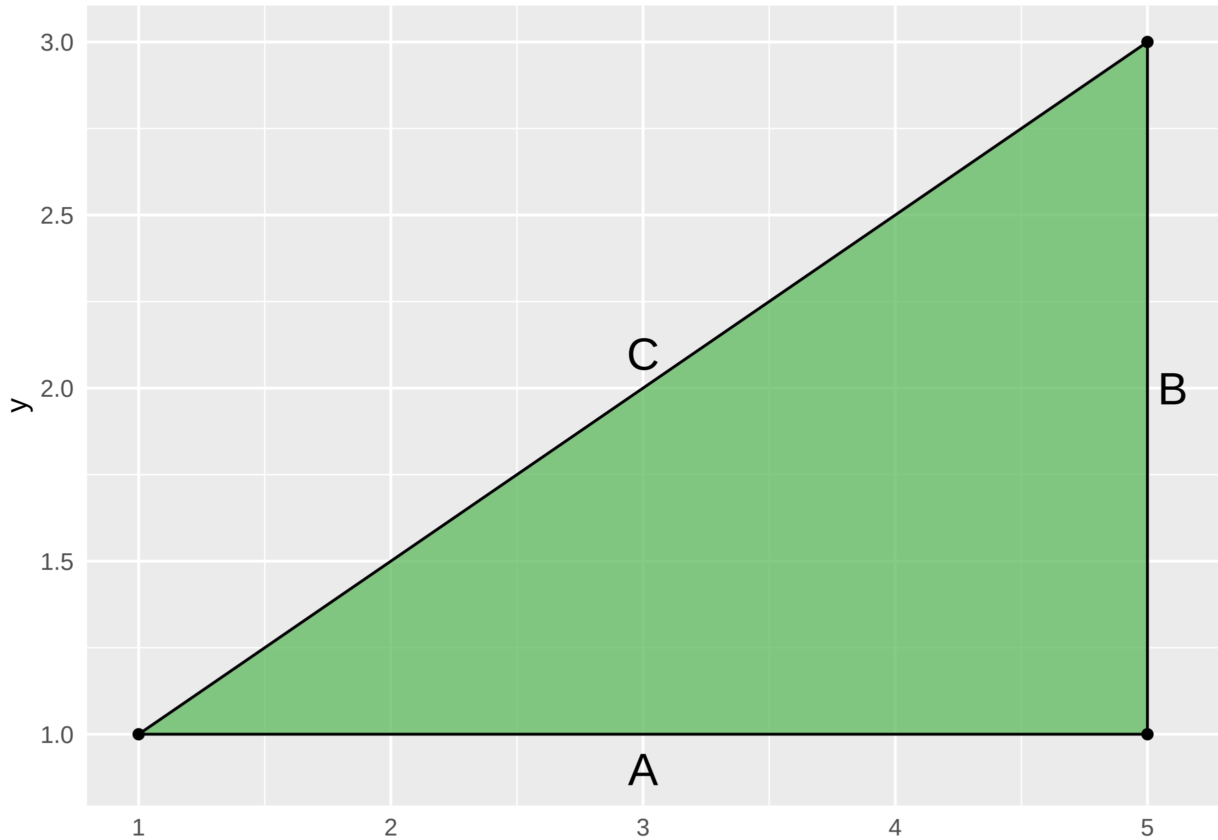
Though you can use funtions to automate processes like we did in the coursework, I am going to take a slightly different take here and focus of using functions for, well, mathimatical functions!

1.1 The Pythagorean Theorem

$$a^2 + b^2 = c^2$$

Where:

- a is the length of leg A
- b is leg length of leg B
- c is the length of leg C, otherwise known as the hypotenuse



Solve for c and include {roxygen2} skeletons!

Let:

- $a =$ `c(5,3,10,4)`
- $b =$ `c(3,5,10,20)`

```r
#' The Pythagorean Theorem
#'
#' @param a1 A numberic or a vector. The first leg of a right triangle.
#' @param b1 A numberic or a vector. The second leg of a right triangle.
#'
#' @return A numeric or a vector. The hypotenuse of the triangle.
#' @export
#'
#' @examples
#' pythagoreanTheorem(a1 = 5, b1 = 3)
pythagoreanTheorem <- function(a1, b1) {

  c2 <- a1^2 + b1^2

  c0 <- c2^(1/2) # could also use sqrt()

  return(c0)
}
```

```r
a <- c(5,3,10,4)
b <- c(3,5,10,20)
pythagoreanTheorem(a1 = a, b1 = b)
```

```
## [1]  5.830952  5.830952 14.142136 20.396078
```

1.2 Newton's Universal Law of Gravitation

$$F = G\frac{m_1 m_2}{d^2}$$

Where:

- F = force
- G = gravitational constant $(6.67430 * 10^{-11})$
- m_1 = mass of object 1
- m_2 = mass of object 2
- r = distance between centers of the masses

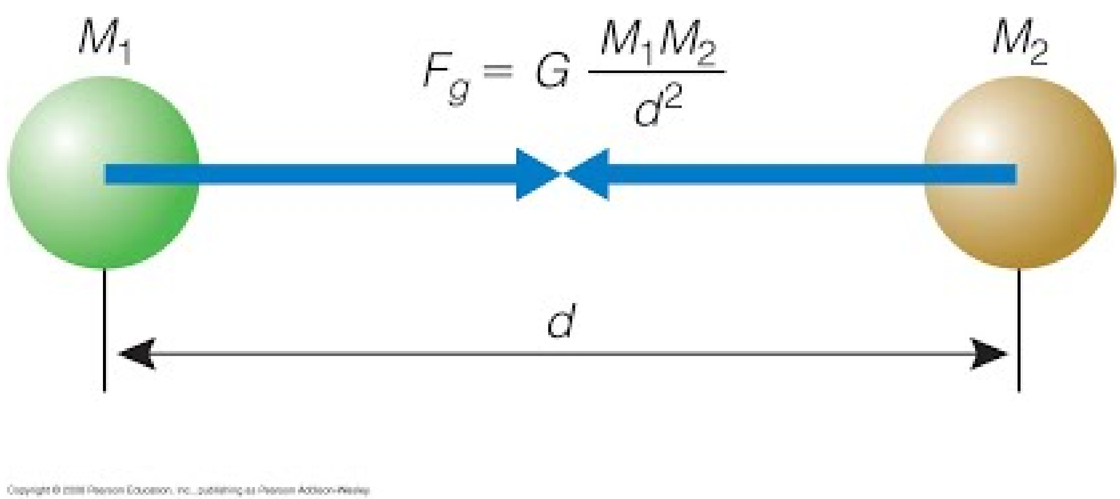


Figure 1: Newton's Universal Law of Gravitation

1.2.1 Solve for force (F) and include {roxygen2} skeletons:

Let:

- the mass of object 1 (m_1) = 5,

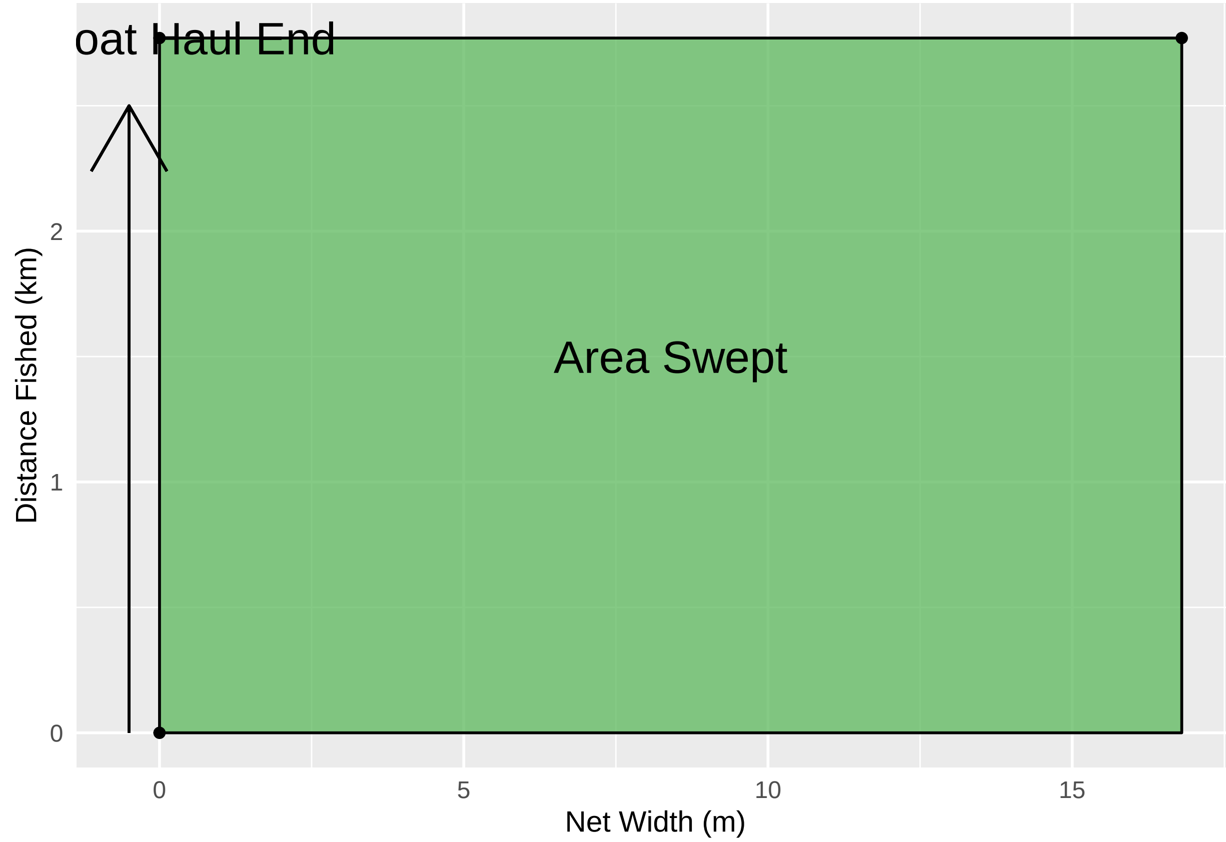
- the mass of object 2 (m_2) = 3, and
- distance between the two masses (d) = 2.

Hint! The Gravitational Constant Shouldn't change, unless you are testing this out on other planets. To save time, add the gravitational constant on earth to where you define the arguments in your function.

```r
#' Calculate Newton's Universal Law of Gravitation
#'
#' @param G0 A numeric or a vector. The gravitational constant. The default is 6.67430*10^-11.
#' @param m_1 A numeric or a vector. Mass of the first object.
#' @param m_2 A numeric or a vector. Mass of the second object.
#' @param d2 A numeric or a vector. distance between centers of the masses.
#'
#' @return A numeric or a vector. The force.
#' @export
#'
#' @examples
#' NewtonGrav(m_1 = 5, m_2 = 3, d2 = 2)
NewtonGrav <- function(G0 = 6.67430*10^-11,
                       m_1,
                       m_2,
                       d2){

  F1 = G0 * (m_1*m_2)/d2

  return(F1)
}

NewtonGrav(m_1 = 5,
           m_2 = 3,
           d2 = 2)
```

```
## [1] 5.005725e-10
```

1.3 Area Swept (from our surveys!)



This is the area sampled for each observation when we survey. Here I'll write without variables to make it easier to read:

$$AreaSwept_{haul}(ha) = \frac{DistanceFished_{haul}(km) * NetWidth_{haul}(m)}{10(\frac{m*km}{ha})}$$

1.3.1 Write a function for this equation and solve for $AreaSwept(km^2)$.

Let:

- dist_fish = Distance Fished (2.77 km)
- net_width = Net Width (16.8 m)

Solve for Area and include {roxygen2} skeletons.

```
#' Area Swept
#'
#' @param dist_fish A numeric or a vector. Distnace survey fished on this trip.
#' @param net_width A numeric or a vector. Width of net.
#'
#' @return
#' @export
```

```r
#'
#' @examples area_swept(dist_fish = 2.77, net_width = 16.8)
area_swept<-function(dist_fish, net_width) {
  return((dist_fish * net_width)/10)
}

area_swept(dist_fish = 2.77, net_width = 16.8)
```

```
## [1] 4.6536
```

1.3.2 Now we will apply this function to some real data! Here I've combined two datasets, *haul* and *catch* for Eastern Bering Sea data (people familiar with the survey data on Oracle will know what this means).

```r
EBS<-read_csv(file = here("data", "haul_catch.csv"))
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```r
EBS<-janitor::clean_names(EBS)
EBS_summary <- EBS %>%
  dplyr::filter(year == 2017) %>%
  dplyr::select(hauljoin, distance_fished, net_width) %>%
  # Now notice that this equation is calculated by haul.
  # Right now the data includes species data so each haul number is
  # listed multiple times in the haul column. We'll need to find the
  # unqiue rows of the data so there is one row per haul.
  dplyr::distinct(hauljoin, distance_fished, net_width)

head(EBS_summary)
```

hauljoin	distance_fished	net_width
-16940	2.768	16.811
-16941	2.734	16.248
-16520	2.872	16.505
-16533	2.916	17.834
-16345	2.824	16.007
-16551	2.838	16.745

```r
str(EBS_summary)
```

```
## tibble [376 x 3] (S3: tbl_df/tbl/data.frame)
##  $ hauljoin       : num [1:376] -16940 -16941 -16520 -16533 -16345 ...
##  $ distance_fished: num [1:376] 2.77 2.73 2.87 2.92 2.82 ...
##  $ net_width      : num [1:376] 16.8 16.2 16.5 17.8 16 ...
```

Let:

- dist_fish = EBS_summary$distance_fished

- net_width = EBS_summary$net_width

Add a new column called area_swept to our EBS_summary data frame.

```
EBS_summary$area_swept<-area_swept(dist_fish = EBS_summary$distance_fished,
                                   net_width = EBS_summary$net_width)
summary(EBS_summary)
```

```
##     hauljoin      distance_fished   net_width       area_swept
##  Min.   :-17060   Min.   :1.337   Min.   :13.61   Min.   :2.361
##  1st Qu.:-16965   1st Qu.:2.769   1st Qu.:16.30   1st Qu.:4.591
##  Median :-16781   Median :2.828   Median :16.93   Median :4.787
##  Mean   :-16728   Mean   :2.816   Mean   :16.98   Mean   :4.779
##  3rd Qu.:-16532   3rd Qu.:2.896   3rd Qu.:17.62   3rd Qu.:5.004
##  Max.   :-16323   Max.   :3.439   Max.   :20.12   Max.   :5.799
```

```
str(EBS_summary)
```

```
## tibble [376 x 4] (S3: tbl_df/tbl/data.frame)
##  $ hauljoin       : num [1:376] -16940 -16941 -16520 -16533 -16345 ...
##  $ distance_fished: num [1:376] 2.77 2.73 2.87 2.92 2.82 ...
##  $ net_width      : num [1:376] 16.8 16.2 16.5 17.8 16 ...
##  $ area_swept     : num [1:376] 4.65 4.44 4.74 5.2 4.52 ...
```

1.4 Catch Per Unit Effort (CPUE)

CPUE is calculated by dividing the catch weight of each fishing trip by the area swept (from the last equation!). This gives CPUE in units of kilograms per hectar.

$$CPUE_{haul}(kg/ha) = \frac{FishWeight_{haul}(kg)}{(DistanceFished_{haul}(km) * NetWidth_{haul}(m))/10(\frac{m*km}{ha})} = \frac{FishWeight_{haul}(kg)}{AreaSwept_{haul}(ha)}$$

Write a function for the above CPUE equation and solve for $CPUE$ using the below data and using your function for area swept from the last question.

```
# EBS<-read_csv(file = here("data", "haul_catch.csv"))
# EBS<-janitor::clean_names(EBS)
EBS_summary <- EBS %>%
  dplyr::select(year, hauljoin, distance_fished, net_width, weight) %>%
  dplyr::group_by(year, hauljoin, distance_fished, net_width) %>%
  dplyr::summarise(sum_weight = sum(weight, na.rm = TRUE))

head(EBS_summary)
```

year	hauljoin	distance_fished	net_width	sum_weight
1982	857	2.667	16.055	392.992
1982	858	2.889	16.055	2195.443
1982	859	2.667	16.440	2547.919
1982	860	2.889	16.440	2264.883
1982	862	2.685	16.454	988.244
1982	863	2.519	16.440	1283.893

Let:

- catch = catch from the survey (catch = EBS_summary$number_fish),
- time = time in hours from the survey (EBS_summary$duration),
- trips = number of trips taken during survey (EBS_summary$haul).

Solve for CPUE (CPUE) and include {roxygen2} skeleton.

Below the full text for my description of dat is: '#' @param dat A data.frame. This contains at least 3 headers named sum_weight, distance_fished, and net_width. All 3 columns are A numeric or a vector. sum_weight is a numeric or a vector that represents the total weight of catch of a haul. dist_fished is the distnace survey fished on this trip and net_width and width of net.'

```r
#' Calculate CPUE of Survey
#'
#' @param dat A data.frame. This contains at least 3 headers named sum_weight, distance_fished, and net_
#' @return A numeric or a data.frame of hauljoin and CPUE (kg/hect).
#' @export
#'
#' @examples
#' CPUE0<-CPUE(dat)
CPUE <- function(dat){

  CPUE0 <- dat %>%
      dplyr::group_by(hauljoin) %>%
      dplyr::summarise(CPUE = (sum_weight/
                                 area_swept(dist_fish = distance_fished,
                                            net_width = net_width)))

  return(CPUE0)
}

CPUE0<-CPUE(dat = EBS_summary)
head(CPUE0)
```

hauljoin	CPUE
-19495	117.95633
-19484	104.23919
-19483	59.33973
-19482	60.27348
-19481	123.40286
-19480	283.15348