ESM 262: Computing Assignment 5

Taylor Cook and Kelsey Warren

2025-03-11

Design a function - you can pick any subject and you can even make up the equations as long as they conceptually make sense

- Make sure it has at least 2 inputs and 1 parameter (ideally more) and at least 2 outputs
- Code your function in R, save as a R file in subdirectory called R
- Make sure you include documentation (both at the top as we've shown in past examples, and inline)
- Include some error checking

```
#' To estimate the oil spill area (km^2) and thickness of oil spill spread (mm) after an oil spill in t
#'
#' @param volume the volume of oil that leaked in the spill (m^3)
#' @param wtemp the water temperature of the ocean (degrees C)
#' @param windspeed the wind speed (m/s)
#' @param viscosity the oil viscosity factor
#' @param ratecoeff the spread rate coefficient
#'
#'
#' @return list with the following items
#' \describe{
#' \item{area}{Area of oil spill spread (km^2)}
#' \item{thickness}{Thickness of the oil spill spread (mm)}
#' }
#'
#'
#' @examples
#' oil(volume, wtemp, windspeed, viscosity=0.0001, ratecoeff=0.16)
#' @references
#' https://link.springer.com/chapter/10.1007/978-1-4684-9019-0_5


oil = function(volume, wtemp, windspeed, viscosity, ratecoeff) {

  # error checking
  # check volume
  volume = ifelse( (volume<0), return("Caution: oil volume cannot be negative"), volume)

  # check wtemp
```

```
  wtemp = ifelse( (wtemp<0), return("Caution: ocean temperature is below freezing"), wtemp)

  # check windspeed
  windspeed = ifelse( (windspeed<0), return("Caution: windspeed cannot be negative"), windspeed)

  # check viscosity
  viscosity = ifelse( (viscosity<0), return("Caution: viscosity cannot be negative"), viscosity)

  # check ratecoeff
  ratecoeff = ifelse( (ratecoeff<0), return("Caution: ratecoeff cannot be negative"), ratecoeff)


# make some adjustments to oil spread based on ocean temperature and wind speed
# ocean temperature adjustment
wtemp_adj = 1 + (wtemp-15) * 0.01  # warmer water increases spread rate. for every 1C above 15C, spread
# windspeed adjustment
windspeed_adj = 1 + (windspeed * 0.05) # faster wind speed increases spread. each 1m/s increase in wind


# calculate oil spill area (km^2) using Fay's Model for Spill Area
area = ratecoeff * (volume^(2/3)) * wtemp_adj * windspeed_adj

# calculate oil spread thickness (mm)
thickness = (volume / (area * 1e6)) * 10^3 # multiply area*10^6 for km^2 --> m^2; multiply thickness (m

# output from function
return(list(area_km2=round(area,3), thickness_mm=round(thickness,3)))
}
```

In an Quatro file, generate some data for 2 of the function inputs

- use a *for* loop to run the function for the data
- repeat the "looping" using something from the "purrr" package
- Graph results (you can decide what the most interesting way to graph - you just need to make one graph)

```
#Generate data for 2 of the function inputs
volume = c(100, 200, 300, 400, 500)
wtemp = c(10, 15, 20, 25, 30)
windspeed = c(5, 10, 15, 20, 25)

# run the function for the data using a for loop
results = list()
for (i in 1:length(volume)) {
  results[[i]] = oil(volume[i], wtemp[i], windspeed[i], viscosity=0.0001, ratecoeff=0.16)
}

print(results)
```

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

```
## [[1]]$area_km2
## [1] 4.093
##
## [[1]]$thickness_mm
## [1] 0.024
##
##
## [[2]]
## [[2]]$area_km2
## [1] 8.208
##
## [[2]]$thickness_mm
## [1] 0.024
##
##
## [[3]]
## [[3]]$area_km2
## [1] 13.175
##
## [[3]]$thickness_mm
## [1] 0.023
##
##
## [[4]]
## [[4]]$area_km2
## [1] 19.11
##
## [[4]]$thickness_mm
## [1] 0.021
##
##
## [[5]]
## [[5]]$area_km2
## [1] 26.08
##
## [[5]]$thickness_mm
## [1] 0.019
```

```
# repeat the "looping" using something from the "purrr" package
results_purrr = purrr::pmap(list(volume, wtemp, windspeed), ~oil(.x, .y, 15, viscosity=0.0001, ratecoef

print(results_purrr)
```

```
## [[1]]
## [[1]]$area_km2
## [1] 5.731
##
## [[1]]$thickness_mm
## [1] 0.017
##
##
## [[2]]
## [[2]]$area_km2
## [1] 9.576
```

```
##
## [[2]]$thickness_mm
## [1] 0.021
##
##
## [[3]]
## [[3]]$area_km2
## [1] 13.175
##
## [[3]]$thickness_mm
## [1] 0.023
##
##
## [[4]]
## [[4]]$area_km2
## [1] 16.721
##
## [[4]]$thickness_mm
## [1] 0.024
##
##
## [[5]]
## [[5]]$area_km2
## [1] 20.285
##
## [[5]]$thickness_mm
## [1] 0.025
```

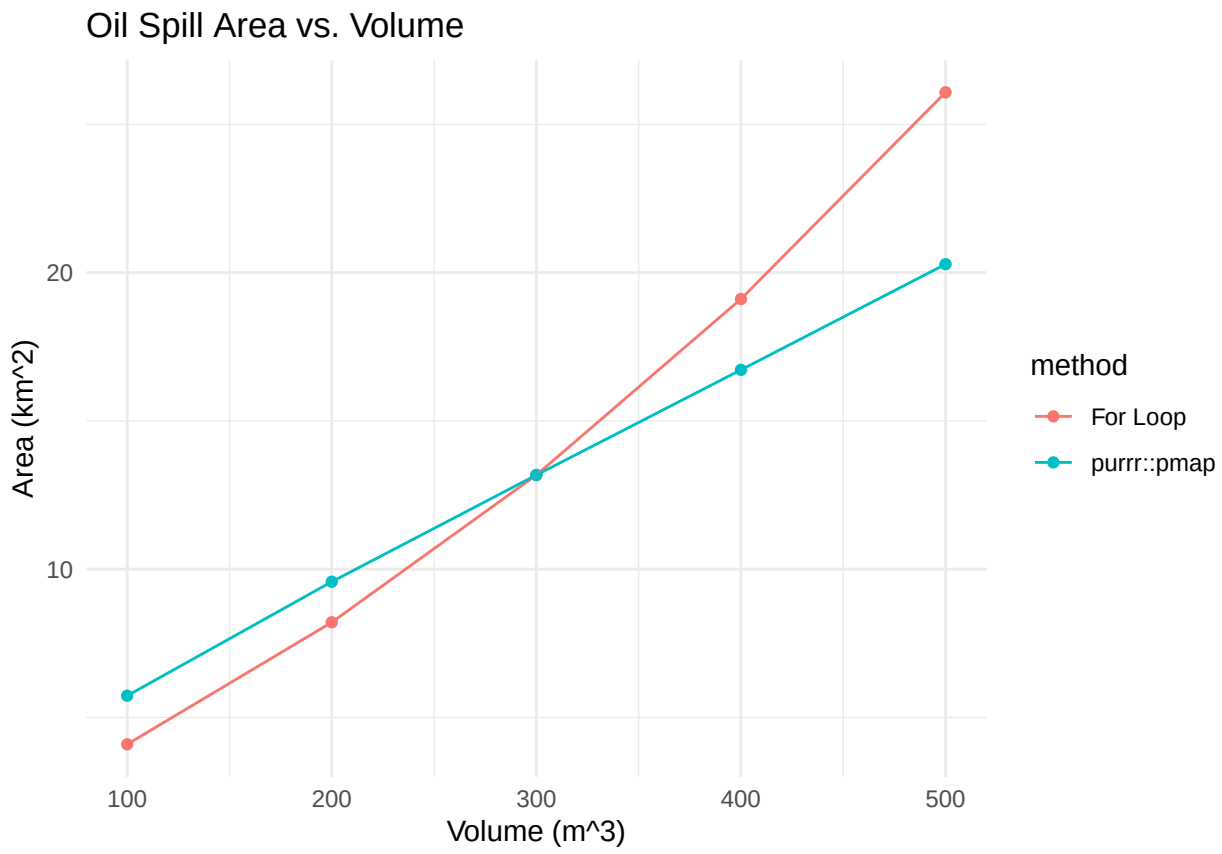
```r
# Graph results of both methods

results_df = bind_rows(results) %>%
  mutate(volume = volume, wtemp = wtemp, windspeed = windspeed, method = "For Loop")

results_purr_df = bind_rows(results_purrr) %>%
  mutate(volume = volume, wtemp = wtemp, windspeed = windspeed, method = "purrr::pmap")

# Combine both dataframes into one
results_combined = bind_rows(results_df, results_purr_df)


results_combined %>%
  ggplot(aes(x = volume, y = area_km2, color = method, group = method)) +
  geom_point() +
  geom_line() +
  labs(title = "Oil Spill Area vs. Volume", x = "Volume (m^3)", y = "Area (km^2)") +
  theme_minimal()
```



Write at least 2 tests for your function; store in a separate test file

Put this in a git repo that includes an R subdirectory, a tests subsdirectory and your Quatro file and submit the link on Canvas (you can put all of this in a new git repo or a subdirectory of an existing git repo)