Loops and Functions (HW 6)

Katie Clouse

2025-03-27

```
ec50.data <- read.csv("C:/Users/katie/Downloads/EC50_all.csv")
library(ggplot2)
install.packages("drc", repos = "https://cran.r-project.org")
```

Reading in the data

```
## Installing package into 'C:/Users/katie/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)

## package 'drc' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\katie\AppData\Local\Temp\RtmpisZgDX\downloaded_packages
```

```
library("drc")
```

```
## Warning: package 'drc' was built under R version 4.4.3

## Loading required package: MASS

##
## 'drc' has been loaded.

## Please cite R and 'drc' if used for a publication,

## for references type 'citation()' and 'citation('drc')'.

##
## Attaching package: 'drc'

## The following objects are masked from 'package:stats':
##
##     gaussian, getInitial
```

```r
library(tidyverse)
```

```
## Warning: package ’tidyverse’ was built under R version 4.4.2
```

```
## Warning: package ’dplyr’ was built under R version 4.4.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
# str() #allows you to see structure
# (5*(degree_f - 32)/9)#how to find Celsius
(5*(32 - 32)/9)
```

Examples of functions notes

```
## [1] 0
```

```r
(5*(36 - 32)/9)
```

```
## [1] 2.222222
```

```r
(5*(39 - 32)/9)
```

```
## [1] 3.888889
```

```r
(5*(40 - 32)/9)
```

```
## [1] 4.444444
```

```r
(5*(42 - 32)/9)
```

```
## [1] 5.555556
```

```
#f_temp is what we are going to input into the function
F_to_C <- function(f_temp){
  celsius <- (5*(f_temp - 32)/9)
  return(celsius)
}
# we wrote this function so that the resulting value will be celsius. This will happen for the data poi
# wehn you run it, it won't do anything bc you are setting this as a function

F_to_C(32)
```

Creating a syntax

```
## [1] 0
```

```
F_to_C(80)
```

```
## [1] 26.66667
```

```
# the function F_to_c is the same thing as the function. We are putting in the f_temp value and we are
```

```


# helps prevent copy and paste errors
# iterations function in base R
rep("A", 3) # repeats A 3 times
```

Iterations

```
## [1] "A" "A" "A"
```

```
rep(c("A", "B"),10)
```

```
##  [1] "A" "B" "A" "B" "A" "B" "A" "B" "A" "B" "A" "B" "A" "B" "A" "B" "A" "B" "A"
## [20] "B"
```

```
rep(c(1,2,5,3), 4, each = 5) #takes each number and repeats each number seperately 5 times and then it
```

```
##  [1] 1 1 1 1 1 2 2 2 2 2 5 5 5 5 5 3 3 3 3 3 1 1 1 1 1 2 2 2 2 2 5 5 5 5 5 3 3 3
## [39] 3 3 1 1 1 1 1 2 2 2 2 2 5 5 5 5 5 3 3 3 3 3 1 1 1 1 1 2 2 2 2 2 5 5 5 5 5 3
## [77] 3 3 3 3
```

```
1:7
```

```
## [1] 1 2 3 4 5 6 7
```

```
seq(from = 1, to = 7) #this is the same function at 1:7
```

```
## [1] 1 2 3 4 5 6 7
```

```
seq(from = 0, to = 10, by = 2) #this counts to 10 from 0 by two
```

```
## [1]  0  2  4  6  8 10
```

```
rep(seq(from = 0, to = 10, by = 2), times = 3, each = 2) #this goes through the pattern 00224466881010
```

```
##  [1]  0  0  2  2  4  4  6  6  8  8 10 10  0  0  2  2  4  4  6  6  8  8 10 10  0
## [26]  0  2  2  4  4  6  6  8  8 10 10
```

```
#seq_along()# allows you to generate sequence of numbers based on character vector

LETTERS #the alphabet
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
seq_along(LETTERS) #this allows you to go letter by letter, which corresponds to a certain number. This
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26
```

```
# whatever language youre coding in , this loop has a cetain algorithm. It sets up an iteration so you

for (i in 1:10){ #for each value of i (which is 1-10), it is going to multiply that value by 2
  (i*2)
  print(i*2)
}
```

The for loop

```
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
## [1] 12
## [1] 14
## [1] 16
## [1] 18
## [1] 20
```

```
#this will print to the console values 1 through 10 multiplied by 2
# you start out with "for" and parentheses. You define the iteration within these parentheses.
#Then you have the word "in" and 1 through 10 (or whatever numbers you need).
#This is setting i equal to one through 10.
#Then you open with curly brackets and press enter and put whatever you need in the iteration. The outp
```

```r
for (i in -30:100){
  result <- F_to_C(i)
  print(result)
} #you are multiplying the values -30 to 100 by the celsius formula and then you're result is what the
```

```
## [1] -34.44444
## [1] -33.88889
## [1] -33.33333
## [1] -32.77778
## [1] -32.22222
## [1] -31.66667
## [1] -31.11111
## [1] -30.55556
## [1] -30
## [1] -29.44444
## [1] -28.88889
## [1] -28.33333
## [1] -27.77778
## [1] -27.22222
## [1] -26.66667
## [1] -26.11111
## [1] -25.55556
## [1] -25
## [1] -24.44444
## [1] -23.88889
## [1] -23.33333
## [1] -22.77778
## [1] -22.22222
## [1] -21.66667
## [1] -21.11111
## [1] -20.55556
## [1] -20
## [1] -19.44444
## [1] -18.88889
## [1] -18.33333
## [1] -17.77778
## [1] -17.22222
## [1] -16.66667
## [1] -16.11111
## [1] -15.55556
## [1] -15
## [1] -14.44444
## [1] -13.88889
## [1] -13.33333
## [1] -12.77778
## [1] -12.22222
## [1] -11.66667
## [1] -11.11111
## [1] -10.55556
## [1] -10
## [1] -9.444444
## [1] -8.888889
## [1] -8.333333
## [1] -7.777778
## [1] -7.222222
## [1] -6.666667
## [1] -6.111111
## [1] -5.555556
## [1] -5
## [1] -4.444444
## [1] -3.888889
## [1] -3.333333
## [1] -2.777778
## [1] -2.222222
## [1] -1.666667
## [1] -1.111111
## [1] -0.5555556
## [1] 0
## [1] 0.5555556
## [1] 1.111111
## [1] 1.666667
## [1] 2.222222
## [1] 2.777778
## [1] 3.333333
## [1] 3.888889
## [1] 4.444444
## [1] 5
## [1] 5.555556
## [1] 6.111111
## [1] 6.666667
## [1] 7.222222
## [1] 7.777778
## [1] 8.333333
## [1] 8.888889
## [1] 9.444444
## [1] 10
## [1] 10.55556
## [1] 11.11111
## [1] 11.66667
## [1] 12.22222
## [1] 12.77778
## [1] 13.33333
## [1] 13.88889
## [1] 14.44444
## [1] 15
## [1] 15.55556
## [1] 16.11111
## [1] 16.66667
## [1] 17.22222
## [1] 17.77778
## [1] 18.33333
## [1] 18.88889
## [1] 19.44444
## [1] 20
## [1] 20.55556
## [1] 21.11111
## [1] 21.66667
## [1] 22.22222
## [1] 22.77778
## [1] 23.33333
## [1] 23.88889
## [1] 24.44444
## [1] 25
## [1] 25.55556
## [1] 26.11111
## [1] 26.66667
## [1] 27.22222
## [1] 27.77778
## [1] 28.33333
## [1] 28.88889
## [1] 29.44444
## [1] 30
## [1] 30.55556
## [1] 31.11111
## [1] 31.66667
## [1] 32.22222
## [1] 32.77778
## [1] 33.33333
## [1] 33.88889
## [1] 34.44444
## [1] 35
## [1] 35.55556
## [1] 36.11111
## [1] 36.66667
## [1] 37.22222
## [1] 37.77778
```

```
# looping the values basically means that you are running them through the code in the curly brackets
#the problem with these values is that they are not saved so you cannot plot them
```

```


#you need to type out what you want to cal it and set it equal to null

celcius.df <- NULL#this allows you to have a blank object so you can use it in the future

for (i in -30:100){
  result <- data.frame(F_to_C(i), i)
  print(result)
}
```

Initializing and r object

```
##   F_to_C.i.   i
## 1 -34.44444 -30
##   F_to_C.i.   i
## 1 -33.88889 -29
##   F_to_C.i.   i
## 1 -33.33333 -28
##   F_to_C.i.   i
## 1 -32.77778 -27
##   F_to_C.i.   i
## 1 -32.22222 -26
##   F_to_C.i.   i
## 1 -31.66667 -25
##   F_to_C.i.   i
## 1 -31.11111 -24
##   F_to_C.i.   i
## 1 -30.55556 -23
##   F_to_C.i.   i
## 1       -30 -22
##   F_to_C.i.   i
## 1 -29.44444 -21
##   F_to_C.i.   i
## 1 -28.88889 -20
##   F_to_C.i.   i
## 1 -28.33333 -19
##   F_to_C.i.   i
## 1 -27.77778 -18
##   F_to_C.i.   i
## 1 -27.22222 -17
##   F_to_C.i.   i
## 1 -26.66667 -16
##   F_to_C.i.   i
## 1 -26.11111 -15
##   F_to_C.i.   i
## 1 -25.55556 -14
##   F_to_C.i.   i
## 1       -25 -13
##   F_to_C.i.   i
## 1 -24.44444 -12
##   F_to_C.i.   i
## 1 -23.88889 -11
##   F_to_C.i.   i
## 1 -23.33333 -10
##   F_to_C.i.  i
## 1 -22.77778 -9
##   F_to_C.i.  i
## 1 -22.22222 -8
##   F_to_C.i.  i
## 1 -21.66667 -7
##   F_to_C.i.  i
## 1 -21.11111 -6
##   F_to_C.i.  i
## 1 -20.55556 -5
##   F_to_C.i.  i
## 1       -20 -4
##   F_to_C.i.  i
## 1 -19.44444 -3
##   F_to_C.i.  i
## 1 -18.88889 -2
##   F_to_C.i.  i
## 1 -18.33333 -1
##   F_to_C.i. i
## 1 -17.77778 0
##   F_to_C.i. i
## 1 -17.22222 1
##   F_to_C.i. i
## 1 -16.66667 2
##   F_to_C.i. i
## 1 -16.11111 3
##   F_to_C.i. i
## 1 -15.55556 4
##   F_to_C.i. i
## 1       -15 5
##   F_to_C.i. i
## 1 -14.44444 6
##   F_to_C.i. i
## 1 -13.88889 7
##   F_to_C.i. i
## 1 -13.33333 8
##   F_to_C.i. i
## 1 -12.77778 9
##   F_to_C.i.  i
## 1 -12.22222 10
##   F_to_C.i.  i
## 1 -11.66667 11
##   F_to_C.i.  i
## 1 -11.11111 12
##   F_to_C.i.  i
## 1 -10.55556 13
##   F_to_C.i.  i
## 1       -10 14
##   F_to_C.i.  i
## 1 -9.444444 15
##   F_to_C.i.  i
## 1 -8.888889 16
##   F_to_C.i.  i
## 1 -8.333333 17
##   F_to_C.i.  i
## 1 -7.777778 18
##   F_to_C.i.  i
## 1 -7.222222 19
##   F_to_C.i.  i
## 1 -6.666667 20
##   F_to_C.i.  i
## 1 -6.111111 21
##   F_to_C.i.  i
## 1 -5.555556 22
##   F_to_C.i.  i
## 1        -5 23
##   F_to_C.i.  i
## 1 -4.444444 24
##   F_to_C.i.  i
## 1 -3.888889 25
##   F_to_C.i.  i
## 1 -3.333333 26
##   F_to_C.i.  i
## 1 -2.777778 27
##   F_to_C.i.  i
## 1 -2.222222 28
##   F_to_C.i.  i
## 1 -1.666667 29
##   F_to_C.i.  i
## 1 -1.111111 30
##    F_to_C.i.  i
## 1 -0.5555556 31
##   F_to_C.i.  i
## 1         0 32
##   F_to_C.i.  i
## 1 0.5555556 33
##   F_to_C.i.  i
## 1  1.111111 34
##   F_to_C.i.  i
## 1  1.666667 35
##   F_to_C.i.  i
## 1  2.222222 36
##   F_to_C.i.  i
## 1  2.777778 37
##   F_to_C.i.  i
## 1  3.333333 38
##   F_to_C.i.  i
## 1  3.888889 39
##   F_to_C.i.  i
## 1  4.444444 40
##   F_to_C.i.  i
## 1         5 41
##   F_to_C.i.  i
## 1  5.555556 42
##   F_to_C.i.  i
## 1  6.111111 43
##   F_to_C.i.  i
## 1  6.666667 44
##   F_to_C.i.  i
## 1  7.222222 45
##   F_to_C.i.  i
## 1  7.777778 46
##   F_to_C.i.  i
## 1  8.333333 47
##   F_to_C.i.  i
## 1  8.888889 48
##   F_to_C.i.  i
## 1  9.444444 49
##   F_to_C.i.  i
## 1        10 50
##   F_to_C.i.  i
## 1  10.55556 51
##   F_to_C.i.  i
## 1  11.11111 52
##   F_to_C.i.  i
## 1  11.66667 53
##   F_to_C.i.  i
## 1  12.22222 54
##   F_to_C.i.  i
## 1  12.77778 55
##   F_to_C.i.  i
## 1  13.33333 56
##   F_to_C.i.  i
## 1  13.88889 57
##   F_to_C.i.  i
## 1  14.44444 58
##   F_to_C.i.  i
## 1        15 59
##   F_to_C.i.  i
## 1  15.55556 60
##   F_to_C.i.  i
## 1  16.11111 61
##   F_to_C.i.  i
## 1  16.66667 62
##   F_to_C.i.  i
## 1  17.22222 63
##   F_to_C.i.  i
## 1  17.77778 64
##   F_to_C.i.  i
## 1  18.33333 65
##   F_to_C.i.  i
## 1  18.88889 66
##   F_to_C.i.  i
## 1  19.44444 67
##   F_to_C.i.  i
## 1        20 68
##   F_to_C.i.  i
## 1  20.55556 69
##   F_to_C.i.  i
## 1  21.11111 70
##   F_to_C.i.  i
## 1  21.66667 71
##   F_to_C.i.  i
## 1  22.22222 72
##   F_to_C.i.  i
## 1  22.77778 73
##   F_to_C.i.  i
## 1  23.33333 74
##   F_to_C.i.  i
## 1  23.88889 75
##   F_to_C.i.  i
## 1  24.44444 76
##   F_to_C.i.  i
## 1        25 77
##   F_to_C.i.  i
## 1  25.55556 78
##   F_to_C.i.  i
## 1  26.11111 79
##   F_to_C.i.  i
## 1  26.66667 80
##   F_to_C.i.  i
## 1  27.22222 81
##   F_to_C.i.  i
## 1  27.77778 82
##   F_to_C.i.  i
## 1  28.33333 83
##   F_to_C.i.  i
## 1  28.88889 84
##   F_to_C.i.  i
## 1  29.44444 85
##   F_to_C.i.  i
## 1        30 86
##   F_to_C.i.  i
## 1  30.55556 87
##   F_to_C.i.  i
## 1  31.11111 88
##   F_to_C.i.  i
## 1  31.66667 89
##   F_to_C.i.  i
## 1  32.22222 90
##   F_to_C.i.  i
## 1  32.77778 91
##   F_to_C.i.  i
## 1  33.33333 92
##   F_to_C.i.  i
## 1  33.88889 93
##   F_to_C.i.  i
## 1  34.44444 94
##   F_to_C.i.  i
## 1        35 95
##   F_to_C.i.  i
## 1  35.55556 96
##   F_to_C.i.  i
## 1  36.11111 97
##   F_to_C.i.  i
## 1  36.66667 98
##   F_to_C.i.  i
## 1  37.22222 99
##   F_to_C.i.   i
## 1  37.77778 100
```

```r
#this allows you to create a data frame that will be 2 columns that contains result of each function an

celsius.df <- NULL
for (i in -30:100){
  result <- data.frame(F_to_C(i), i)
  celcius.df <- rbind.data.frame(celcius.df, result)
}
#the first iteration, -30, gets plugged into the iteration i. then a one row data frame will be created
#the result will be the numbers -30 to 100 and they will all have gone through the equation to get that
```

“‘