

Blake_Iter+Func

Kylie Blake

2025-03-26

Contents

Functions	2
Iterations	3
The for loop	4

1. Load in necessary packages

```
library(ggplot2)
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
```

```
library(tidyverse)
```

```
## -- 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  1.9.4      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
```

Functions

```
#Learning about functions
#(5*degree_f - 32)/9
#Calculating Celsius:
(5*(32 - 32)/9)
```

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

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

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

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

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

Rather than copy and pasting the same equation and changing the degree_f value, which can lead to typos, we can create a function instead.

Sample function

```
sample.function <- function(... variable goes here ...){ .... code goes here.... return(...
output ...) }
```

```
# Name a function
F_to_C <- function(f_temp){
  celsius <- (5*(f_temp -32)/9)
  return(celsius)
}
#in parentheses put the input for that function, it can be anything
#input f_temp into this caluculation, save that resulting value as Celsius, then return that value

#To use the function:Enter a value you want converted from Fahrenheit to Celsius
F_to_C(32)
```

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

```
F_to_C(90)
```

```
## [1] 32.22222
```

Iterations

- Useful for reproducibility to prevent copy and paste errors

```
#Repetition:
```

```
rep("A",3)
```

```
## [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) #repeat sequence of numbers 4 times with each number repeated 5 times
```

```
## [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
```

```
#Sequence: sequence of #s
```

```
1:7 #this is a sequence, but use seq() as get more complicated
```

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

```
seq(from=1, to =7)
```

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

```
seq(from= 0, to = 10, by=2)
```

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

```
#Seq_along: allows you to generate sequence of numbers based on character vector
```

```
LETTERS #gives alphabet of characters
```

```
## [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) #counts number of characters as sequence
```

```
## [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
```

The for loop

For loop: sets up iteration of initialization Ask question: Is that sequence over? Will repeat

Example 1

```
for (i in 1:10) {  
  print(i*2)  
}
```

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

Note: Inside parentheses for for loop we define our iteration. Usually use an i as general syntax. Setting i = 1 through 10. Print ix2 so essentially 1 through 10 x2. You use print so you can see the result in the console.

Example 2

```
#Calculate Fahrenheit for each value -30 through 100  
for (i in -30:100){  
  result <- F_to_C(i)  
  print(result)  
}
```

```
## [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] -8.88889
## [1] -8.333333
## [1] -7.77778
## [1] -7.22222
## [1] -6.66667
## [1] -6.11111
## [1] -5.55556
## [1] -5
## [1] -4.44444
## [1] -3.88889
## [1] -3.33333
## [1] -2.77778
## [1] -2.22222
## [1] -1.66667
## [1] -1.11111
## [1] -0.555556
## [1] 0
## [1] 0.555556
## [1] 1.11111
## [1] 1.66667
## [1] 2.22222
## [1] 2.77778
## [1] 3.33333
## [1] 3.88889
## [1] 4.44444
## [1] 5
## [1] 5.55556
## [1] 6.11111
## [1] 6.66667
## [1] 7.22222
```

```
## [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
```

```
#To store values from for loop

# Initialize an empty data frame to store the results
celsius.df <- NULL

# Loop through the range of temperatures from -30 to 100
for (i in -30:100) {
  # Convert the current temperature from Fahrenheit to Celsius and store it in a data frame
  result <- data.frame(F_to_C(i), i)

  # Add the result to the celsius.df data frame
  celsius.df <- rbind.data.frame(celsius.df, result)

  # Print the current result
  print(result)
}
```

```
##   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 -0.555556 31
##  F_to_C.i.  i
## 1      0 32
##  F_to_C.i.  i
## 1 0.555556 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 10.555556 51
## F_to_C.i. i
## 1 11.111111 52
## F_to_C.i. i
## 1 11.666667 53
## F_to_C.i. i
## 1 12.222222 54
## F_to_C.i. i
## 1 12.777778 55
## F_to_C.i. i
## 1 13.333333 56
## F_to_C.i. i
## 1 13.888889 57
## F_to_C.i. i
## 1 14.444444 58
## F_to_C.i. i
## 1      15 59
## F_to_C.i. i
## 1 15.555556 60
## F_to_C.i. i
## 1 16.111111 61
## F_to_C.i. i
## 1 16.666667 62
## F_to_C.i. i
## 1 17.222222 63
## F_to_C.i. i
## 1 17.777778 64
## F_to_C.i. i
## 1 18.333333 65
## F_to_C.i. i
## 1 18.888889 66
## F_to_C.i. i
## 1 19.444444 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
```

```
print(celsius.df)
```

```
##      F_to_C.i. i
## 1 -34.4444444 -30
## 2 -33.8888889 -29
## 3 -33.3333333 -28
## 4 -32.7777778 -27
## 5 -32.2222222 -26
## 6 -31.6666667 -25
## 7 -31.1111111 -24
## 8 -30.5555556 -23
## 9 -30.0000000 -22
## 10 -29.4444444 -21
## 11 -28.8888889 -20
## 12 -28.3333333 -19
## 13 -27.7777778 -18
## 14 -27.2222222 -17
## 15 -26.6666667 -16
## 16 -26.1111111 -15
## 17 -25.5555556 -14
## 18 -25.0000000 -13
## 19 -24.4444444 -12
## 20 -23.8888889 -11
## 21 -23.3333333 -10
## 22 -22.7777778 -9
## 23 -22.2222222 -8
## 24 -21.6666667 -7
## 25 -21.1111111 -6
## 26 -20.5555556 -5
## 27 -20.0000000 -4
## 28 -19.4444444 -3
## 29 -18.8888889 -2
## 30 -18.3333333 -1
## 31 -17.7777778 0
## 32 -17.2222222 1
## 33 -16.6666667 2
## 34 -16.1111111 3
## 35 -15.5555556 4
## 36 -15.0000000 5
## 37 -14.4444444 6
## 38 -13.8888889 7
## 39 -13.3333333 8
```

## 40	-12.7777778	9
## 41	-12.2222222	10
## 42	-11.6666667	11
## 43	-11.1111111	12
## 44	-10.5555556	13
## 45	-10.0000000	14
## 46	-9.4444444	15
## 47	-8.8888889	16
## 48	-8.3333333	17
## 49	-7.7777778	18
## 50	-7.2222222	19
## 51	-6.6666667	20
## 52	-6.1111111	21
## 53	-5.5555556	22
## 54	-5.0000000	23
## 55	-4.4444444	24
## 56	-3.8888889	25
## 57	-3.3333333	26
## 58	-2.7777778	27
## 59	-2.2222222	28
## 60	-1.6666667	29
## 61	-1.1111111	30
## 62	-0.5555556	31
## 63	0.0000000	32
## 64	0.5555556	33
## 65	1.1111111	34
## 66	1.6666667	35
## 67	2.2222222	36
## 68	2.7777778	37
## 69	3.3333333	38
## 70	3.8888889	39
## 71	4.4444444	40
## 72	5.0000000	41
## 73	5.5555556	42
## 74	6.1111111	43
## 75	6.6666667	44
## 76	7.2222222	45
## 77	7.7777778	46
## 78	8.3333333	47
## 79	8.8888889	48
## 80	9.4444444	49
## 81	10.0000000	50
## 82	10.5555556	51
## 83	11.1111111	52
## 84	11.6666667	53
## 85	12.2222222	54
## 86	12.7777778	55
## 87	13.3333333	56
## 88	13.8888889	57
## 89	14.4444444	58
## 90	15.0000000	59
## 91	15.5555556	60
## 92	16.1111111	61
## 93	16.6666667	62

##	94	17.2222222	63
##	95	17.7777778	64
##	96	18.3333333	65
##	97	18.8888889	66
##	98	19.4444444	67
##	99	20.0000000	68
##	100	20.5555556	69
##	101	21.1111111	70
##	102	21.6666667	71
##	103	22.2222222	72
##	104	22.7777778	73
##	105	23.3333333	74
##	106	23.8888889	75
##	107	24.4444444	76
##	108	25.0000000	77
##	109	25.5555556	78
##	110	26.1111111	79
##	111	26.6666667	80
##	112	27.2222222	81
##	113	27.7777778	82
##	114	28.3333333	83
##	115	28.8888889	84
##	116	29.4444444	85
##	117	30.0000000	86
##	118	30.5555556	87
##	119	31.1111111	88
##	120	31.6666667	89
##	121	32.2222222	90
##	122	32.7777778	91
##	123	33.3333333	92
##	124	33.8888889	93
##	125	34.4444444	94
##	126	35.0000000	95
##	127	35.5555556	96
##	128	36.1111111	97
##	129	36.6666667	98
##	130	37.2222222	99
##	131	37.7777778	100