

HW14 - Debug

Madhu Peduri

July 22, 2021

0.0.1 Q1(a)

```
calculate.exp <- function(my.number) {  
  exp.num <- (-my.number)^2  
  print(paste("Value of exp.num :", exp.num))  
  result <- exp(exp.num)  
  return(result)  
}
```

0.0.2 Q1(b)

```
calculate.exp(1)
```

```
## [1] "Value of exp.num : 1"
```

```
## [1] 2.72
```

0.0.3 Q1(c)

```
debugonce(calculate.exp)  
calculate.exp(1)
```

```
## debugging in: calculate.exp(1)  
## debug at <text>#1: {  
##   exp.num <- (-my.number)^2  
##   print(paste("Value of exp.num :", exp.num))  
##   result <- exp(exp.num)  
##   return(result)  
## }  
## debug at <text>#2: exp.num <- (-my.number)^2  
## debug at <text>#3: print(paste("Value of exp.num :", exp.num))  
## [1] "Value of exp.num : 1"  
## debug at <text>#4: result <- exp(exp.num)  
## debug at <text>#5: return(result)  
## exiting from: calculate.exp(1)  
## [1] 2.72
```

0.0.4 Q1(d)

- Debug: This method can be used to debug a function by setting a debugging flag. When a function flagged for debugging is entered, normal execution is suspended and the body of function is executed one statement at a time. A new browser context is initiated for each step. Once we flag a function

using this, this remains for the entire session, that is, every call to this function launches the debug browser context.

- Debugonce: This method is same as debug(), but it launches only for immediate next function call and further calls will be run in normal execution mode.

0.0.5 Q1(e)

```
calculate.exp <- function(my.number) {  
  exp.num <- -(my.number)^2  
  browser()  
  print(paste("Value of exp.num :", exp.num))  
  result <- exp(exp.num)  
  return(result)  
}  
calculate.exp(1)
```

```
## Called from: calculate.exp(1)  
## debug at <text>#4: print(paste("Value of exp.num :", exp.num))  
## [1] "Value of exp.num : -1"  
## debug at <text>#5: result <- exp(exp.num)  
## debug at <text>#6: return(result)  
## [1] 0.368
```

0.0.6 Q1(f)

- Editor breakpoints: Another common way to stop on a line of code is to set a breakpoint on that line. We can do this in RStudio by clicking to the left of the line number in the editor, or by pressing Shift+F9 with your cursor on the desired line.
- We call this an “editor breakpoint”. Editor breakpoints take effect immediately and don’t require you to change your code (unlike browser() breakpoints, below).

0.0.7 Q2

```
my.simulation <- function(mu) {  
  # initiate an empty value  
  mean.vec <- NA  
  for (i in 1:1e+05) {  
    # Step 1  
    simu.data <- rnorm(n = 10, mean = mu, sd = 1)  
    # Step 2  
    mean.simu <- mean(simu.data)  
    # Step 3  
    mean.vec <- c(mean.vec, mean.simu)  
  }  
  result <- mean(mean.vec)  
  return(result)  
}  
# print the time it takes to execute the function  
system.time(print(my.simulation(10)))
```

```
## [1] NA  
  
##      user  system elapsed  
##  10.80    0.11    10.91
```

0.0.8 Q2 (a)

```
my.simulation <- function(mu) {  
  # initiate an empty value  
  mean.vec <- NA  
  for (i in 1:10000) {  
    # Step 1  
    simu.data <- rnorm(n = 10, mean = mu, sd = 1)  
    # Step 2  
    mean.simu <- mean(simu.data)  
    # Step 3  
    mean.vec <- c(mean.vec, mean.simu)  
  }  
  result <- mean(mean.vec)  
  return(result)  
}  
  
# print the time it takes to execute the function  
system.time(print(my.simulation(10)))
```

```
## [1] NA  
  
##      user  system elapsed  
##    0.14    0.00    0.15
```

0.0.9 Q2 (b)

```
my.simulation <- function(mu) {  
  # initiate an empty value  
  mean.vec <- 0 ## Corrected statement  
  for (i in 1:100) {  
    # Step 1  
    simu.data <- rnorm(n = 10, mean = mu, sd = 1)  
    # Step 2  
    mean.simu <- mean(simu.data)  
    # Step 3  
    mean.vec <- c(mean.vec, mean.simu)  
  }  
  result <- mean(mean.vec)  
  return(result)  
}  
  
# print the time it takes to execute the function  
system.time(print(my.simulation(10)))
```

```
## [1] 9.89  
  
##      user  system elapsed  
##      0      0      0
```

0.0.10 Q3 (a)

```
get.runs <- function(x, k) {  
  n <- length(x)  
  runs <- NULL  
  for (i in 1:(n - k)) {  
    if (all(x[i:(i + k - 1)] == TRUE)) {
```

```

        runs <- c(runs, i)
      }
    }
    return(runs)
  }

```

0.0.11 Q3 (b)

```

# this function below should return a vector of (4,5,8)
# because there are a run of two TRUEs in those indices.
get.runs(c(TRUE, FALSE, FALSE, TRUE, TRUE, TRUE, FALSE, TRUE,
           TRUE), 2)

```

```
## [1] 4 5
```

0.1 Document Information.

All of the statistical analyses in this document will be performed using R version 4.1.0 (2021-05-18). R packages used will be maintained using the packrat dependency management system.

```
sessionInfo()
```

```

## R version 4.1.0 (2021-05-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19041)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] grid      stats      graphics  grDevices  utils      datasets  methods
## [8] base
##
## other attached packages:
## [1] Rcpp_1.0.7      jtools_2.1.3      dobson_0.4        Matrix_1.3-4
## [5] psych_2.1.6     leaps_3.1         faraway_1.0.7     xtable_1.8-4
## [9] lmtest_0.9-38   zoo_1.8-9         PairedData_1.1.1  mvtnorm_1.1-2
## [13] gld_2.6.2       ggpubr_0.4.0      car_3.0-11        carData_3.0-4
## [17] mnormt_2.0.2    vcd_1.4-8         epiDisplay_3.5.0.1 nnet_7.3-16
## [21] foreign_0.8-81  Hmisc_4.5-0       Formula_1.2-4     survival_3.2-11
## [25] lattice_0.20-44 MASS_7.3-54       ggplot2_3.3.5     rmarkdown_2.8
## [29] knitr_1.33
##
## loaded via a namespace (and not attached):
## [1] nlme_3.1-152     RColorBrewer_1.1-2 tools_4.1.0
## [4] backports_1.2.1  utf8_1.2.1        R6_2.5.0
## [7] rpart_4.1-15     colorspace_2.0-1  withr_2.4.2
## [10] tidyselect_1.1.1 gridExtra_2.3     curl_4.3.1

```

## [13] compiler_4.1.0	formatR_1.11	htmlTable_2.2.1
## [16] scales_1.1.1	checkmate_2.0.0	proxy_0.4-26
## [19] stringr_1.4.0	digest_0.6.27	minqa_1.2.4
## [22] rio_0.5.27	base64enc_0.1-3	jpeg_0.1-8.1
## [25] pkgconfig_2.0.3	htmltools_0.5.1.1	lme4_1.1-27.1
## [28] htmlwidgets_1.5.3	rlang_0.4.11	readxl_1.3.1
## [31] rstudioapi_0.13	generics_0.1.0	dplyr_1.0.7
## [34] zip_2.2.0	magrittr_2.0.1	munsell_0.5.0
## [37] fansi_0.5.0	abind_1.4-5	lifecycle_1.0.0
## [40] stringi_1.6.1	yaml_2.2.1	parallel_4.1.0
## [43] forcats_0.5.1	crayon_1.4.1	lmom_2.8
## [46] haven_2.4.1	splines_4.1.0	pander_0.6.4
## [49] hms_1.1.0	tmvnsim_1.0-2	pillar_1.6.1
## [52] boot_1.3-28	ggsignif_0.6.2	glue_1.4.2
## [55] evaluate_0.14	latticeExtra_0.6-29	data.table_1.14.0
## [58] nloptr_1.2.2.2	png_0.1-7	vctrs_0.3.8
## [61] cellranger_1.1.0	gtable_0.3.0	purrr_0.3.4
## [64] tidyr_1.1.3	xfun_0.23	openxlsx_4.2.4
## [67] broom_0.7.8	e1071_1.7-7	rstatix_0.7.0
## [70] class_7.3-19	tibble_3.1.2	cluster_2.1.2
## [73] ellipsis_0.3.2		