```
> sum_twovals <- function(x, y){
+  return(x+y)
+ }
> sum_two_vals(x=3, y=10)
Error in sum_two_vals(x = 3, y = 10) : 
  could not find function "sum_two_vals"
> sum_twovals <- function(x, y){
+  return(x+y)
+ }
> 
> sum_two_vals(x=3, y=10)
Error in sum_two_vals(x = 3, y = 10) : 
  could not find function "sum_two_vals"
> sum_two_vals <- function(x, y){
+  return(x+y)
+ }
> 
> sum_two_vals(x=3, y=10)
[1] 13
> sum_twoo_vals <- function(x, y){
+  print(pase("value of x=", x))
+  print(pase("value of y=", y))
+  return(x+y)
+ }
> 
> sum_twoo_vals(x=3, y=10)
Error in pase("value of x=", x) : could not find function "pase"
> sum_twoo_vals <- function(x, y){
+  print(paste("value of x=", x))
+  print(paste("value of y=", y))
+  return(x+y)
+ }
> 
> sum_twoo_vals(x=3, y=10)
[1] "value of x= 3"
[1] "value of y= 10"
[1] 13
> great <- function(name){
+  cat("Hello, ", name, "!\n", sep = "")
+ }
> 
> great(name = "Ahasn")
Hello, Ahasn!
> check_even_odd <- function(number){
+  if (number %% 2==0){
+    return("Even")
+  } else {
```

```
+    return("Odd")
+  }
+ }
> print(check_even_odd(4))
[1] "Even"
> print(check_even_odd(7))
[1] "Odd"
> math.operation <- function(x, y, op) {
+  switch(op,
+         "add" = x + y,
+         "subtract" = x - y,
+         "multiply" = x * y,
+         "divide" = if (y != 0) { x / y } else { "Division by zero" },
+         "Invalid operation")
+ }
>
> math.operation(10, 5, "add")
[1] 15
> math.operation(10, 5, "divide")
[1] 2
> math.operation(10, 0, "divide")
[1] "Division by zero"
> calculate_sqrt <- function(val, digit=3){
+  return(round(sqrt(val), digit))
+ }
> sqrt(1001)
[1] 31.63858
> round(sqrt(1001), 2)
[1] 31.64
> calculate_sqrt <- function(val, digit=3){
+  return(round(sqrt(val), digit))
+ }
> calculate_sqrt(24, 2)
[1] 4.9
> calculate_sqrt(25, 2)
[1] 5
> # another way
> calculate_sqrt <- function(val, digit=3){
+  rooted <- sqrl(val)
+  out <- round(rooted, digit)
+  return(out)
+ }
> calculate_sqrt1(22, 2)
Error in calculate_sqrt1(22, 2) :
  could not find function "calculate_sqrt1"
> # another way
> calculate_sqrt1 <- function(val, digit=3){
```

```
+  rooted <- sqrl(val)
+  out <- round(rooted, digit)
+  return(out)
+ }
>
> calculate_sqrt1(22, 2)
Error in sqrl(val) : could not find function "sqrl"
> # Alternative way
>
> calculate_sqrt1 <- function(val, digit=3){
+  rooted <- sqrt(val)  # Corrected function name
+  out <- round(rooted, digit)
+  return(out)
+ }
>
> calculate_sqrt1(22, 2)
[1] 4.69
> calculate_sqrt1 <- function(val, digit=5){
+  rooted <- sqrt(val)  # Corrected function name
+  out <- round(rooted, digit)
+  return(out)
+ }
>
> calculate_sqrt1(22, 2)
[1] 4.69
> begin <- function() {
+  begin_time <- Sys.time()
+  message("Program started: ", begin_time)
+ }
> begin()
Program started: 2025-01-07 22:26:47.59766
> end <- function() {
+  end_time <- Sys.time()
+  runtime <- as.numeric(format(end_time, "%S")) -
as.numeric(format(begin_time, "%S"))
+  message("Program end: ", end_time)
+  message("Runtime: ", seconds_to_time_string(runtime))
+  message("Runtime stored in \"runtime\"")
+ }
>
> end()
Error in end() : object 'begin_time' not found
> timer_env <- new.env()
>
> begin <- function() {
+  timer_env$begin_time <- Sys.time()
+  message("Program started: ", timer_env$begin_time)
+ }
>
> end <- function() {
+  end_time <- Sys.time()
+  runtime <- as.numeric(difftime(end_time, timer_env$begin_time, units =
"secs"))
+  message("Program end: ", end_time)
+  message("Runtime: ", runtime, " seconds")
+ }
>
> # Run the functions
> begin()
Program started: 2025-01-07 22:30:03.051662
> Sys.sleep(3)
> end()
Program end: 2025-01-07 22:30:06.116359
Runtime: 3.06469702720642 seconds
> begin <- function() {
+  begin_time <<- Sys.time()  # Global assignment
+  message("Program started: ", begin_time)
+ }
> begin()
Program started: 2025-01-07 22:32:43.162321
>
> end <- function() {
+  end_time <- Sys.time()
+  runtime <- as.numeric(format(end_time, "%S")) -
as.numeric(format(begin_time, "%S"))
+  message("Program end: ", end_time)
+  message("Runtime: ", runtime, " seconds")
+ }
>
> end()
Program end: 2025-01-07 22:32:43.163607
Runtime: 0 seconds
> trial_func_ret <- function(x) {
+  res <- mean(x)
+  return(res)
+ }
>
> trial_func_inv <- function(x) {
+  res <- mean(x)
+  invisible(res)
+ }
> vals <- c(4, 5, 2, 6, 9)
> trial_func_ret_x <- vals
> trial_func_inv_x <- vals
> visible_sum <- function(x, y) {
+  result <- x + y
+  return(result)  # Explicitly returns and shows the result
+ }
>
> # Call the function
> visible_sum(5, 3)
[1] 8
> invisible_sum <- function(x, y) {
+  result <- x + y
+  invisible(result)  # Returns the result but hides it from the console
+ }
>
> # Call the function
> invisible_sum(5, 3)
> pipeline_example <- function(x) {
+  step1 <- x + 1
+  step2 <- step1 * 2
+  invisible(step2)  # Hides intermediate steps while keeping the result
available
+ }
>
> result <- pipeline_example(5)
> print(result)  # Only prints if explicitly called
[1] 12
> # temporary function
> mat1 <- matrix(c(1, 2, 3, 4), ncol = 2, nrow = 2, byrow = TRUE)
>
> apply(mat1, 1, function(x) sum(x)^2)
[1]  9 49
> apply(mat1, 2, function(x) sum(x)^2)
[1] 16 36
> # wrapper mean function
> wrapper_mean <- function(data, ...) {
+  if (is.numeric(data)) {
+    mean(data, ...)
+  } else {
+    warning("Data is not numeric")
+  }
+ }
>
> wrapper_mean(c(2, 4, 5, 4, 3))
[1] 3.6
> wrapper_mean(c(2, 4, 5, NA, '3'))
Warning message:
In wrapper_mean(c(2, 4, 5, NA, "3")) : Data is not numeric
> wrapper_mean(c(2, 4, 5, NA, '3'), na.rm = TRUE)
```

```
Warning message:
In wrapper_mean(c(2, 4, 5, NA, "3"), na.rm = TRUE) : Data is not numeric
> wrapper_mean(c(2, 4, 5, NA, 3), na.rm = TRUE)
[1] 3.5
>
> # function as argument
> apply_function <- function(vec, funct) {
+  return(funct(vec))
+ }
> wrapper_mean <- function(data, ...) {
+  if (is.numeric(data)) {
+    mean(data, ...)
+  } else {
+    warning("Data is not numeric")
+  }
+ }
>
> wrapper_mean(c(2, 4, 5, 4, 3))
[1] 3.6
> wrapper_mean(c(2, 4, 5, NA, '3'))
Warning message:
In wrapper_mean(c(2, 4, 5, NA, "3")) : Data is not numeric
> wrapper_mean(c(2, 4, 5, NA, '3'), na.rm = TRUE)
Warning message:
In wrapper_mean(c(2, 4, 5, NA, "3"), na.rm = TRUE) : Data is not numeric
> wrapper_mean(c(2, 4, 5, NA, 3), na.rm = TRUE)
[1] 3.5
> wrapper_mean <- function(data, ...) {
+  if (is.numeric(data)) {
+    mean(data, ...)
+  } else {
+    warning("Data is not numeric")
+  }
+ }
> wrapper_mean(c(2, 4, 5, 4, 3))
[1] 3.6
> wrapper_mean(c(2, 4, 5, NA, '3'))
Warning message:
In wrapper_mean(c(2, 4, 5, NA, "3")) : Data is not numeric
> wrapper_mean(c(2, 4, 5, NA, '3'), na.rm = TRUE)
Warning message:
In wrapper_mean(c(2, 4, 5, NA, "3"), na.rm = TRUE) : Data is not numeric
> wrapper_mean(c(2, 4, 5, NA, 3), na.rm = TRUE)
[1] 3.5
> apply_function <- function(vec, funct){
+  return(fun(vec))
+ }
```

```
>
>
> apply_function(c(1,2,3,4,), fun=mean)
Error in fun(vec) : could not find function "fun"
> apply_function <- function(vec, fun){
+  return(fun(vec))
+ }
>
>
> apply_function(c(1,2,3,4,), fun=mean)
Error in c(1, 2, 3, 4, ) : argument 5 is empty
> apply_function <- function(vec, fun){
+  return(fun(vec))
+ }
>
> apply_function(c(1,2,3,4), fun=mean)
[1] 2.5
> # Roxygen comments
>
> #' @title calculate sort
> #' @description Calculates square root of the numbers
> #' @param val numeric vector or length 1 or more
> #' @param digit numeric value
> #' @return numeric vector that is squared root of the passed vector
> #' # @examples
> #' # calculate_sqrt(c(6,10,453), 4)
>
> # calculate_sort(c(2,3,4), 2)  # This line is likely a call to a function not
shown in the image.
>
> calculate_sqrt <- function(val, digit = 3) {
+  return(round(sqrt(val), digit))
+ }
```