Calling Functions - Scoping - Review

In-class practice

Marcus Birkenkrahe (pledged)

January 29, 2025

I have already uploaded test results to tinyurl.com/ds2-test1-data:

```
grades <- c(17.75, 19.08, 19, 20.83, 21.33, 21.67, 22, NA, 22.08, 22.67, 22.83, 23.33,
write(grades, file="../data/test1",ncolumns=1)
system("cat ../data/test1")
```

```
17.75
19.08
19
20.83
21.33
21.67
22
NA
22.08
22.67
22.83
23.33
23.33
23.33
24
NA
```

1. Download the results directly from my URL into R using `readLines`.

   ```
   x <- readLines("https://tinyurl.com/ds2-test1-data") |> as.numeric()
   x
   ```

   ```
   Warning message:
   ```

```
NAs introduced by coercion
 [1] 17.75 19.08 19.00 20.83 21.33 21.67 22.00    NA 22.08 22.67 22.83 23.33 23.33
```

2. Verify that the result is a numeric vector.

   ```
   is.vector(x) & is.numeric(x)
   ```

   ```
   [1] TRUE
   ```

3. Which namespace does `readLines` belong to?

   ```
   environment(readLines)
   ```

   ```
   <environment: namespace:base>
   ```

4. Show the source code of `readLines` and specify the namespace:

   ```
   base::readLines
   ```

   ```
   function (con = stdin(), n = -1L, ok = TRUE, warn = TRUE, encoding = "unknown",
       skipNul = FALSE)
   {
       if (is.character(con)) {
   con <- file(con, "r")
   on.exit(close(con))
       }
       .Internal(readLines(con, n, ok, warn, encoding, skipNul))
   }
   <bytecode: 0x63ad3a8b0538>
   <environment: namespace:base>
   ```

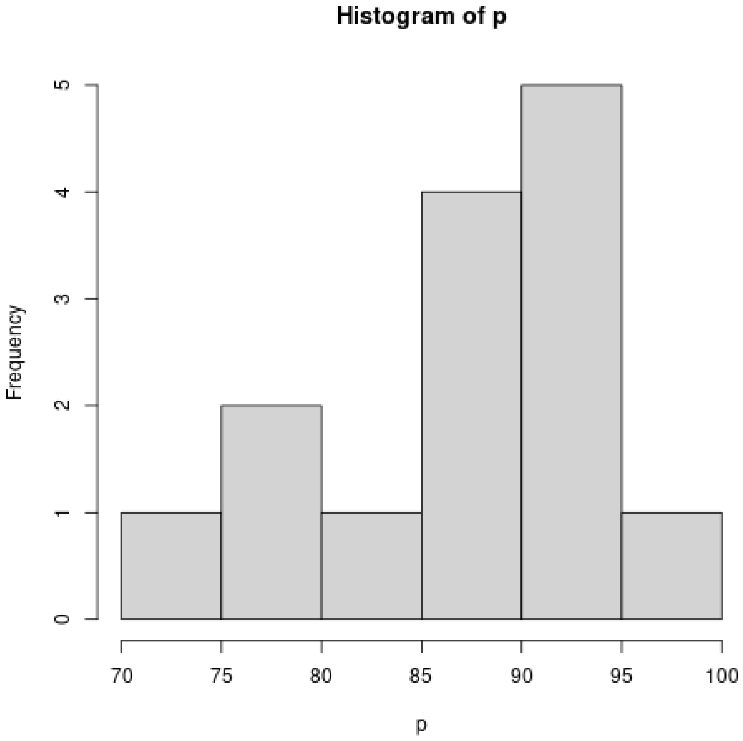
5. Remove the `NA` values and convert the points to percentages (the maximum point value was 25).

   ```
   x <- x[!is.na(x)] # only retain values that are NOT NA
   p <- (x/25)*100 # convert point values to percentage values
   p
   ```

   ```
   [1] 71.00 76.32 76.00 83.32 85.32 86.68 88.00 88.32 90.68 91.32 93.32 93.32 93.32
   ```

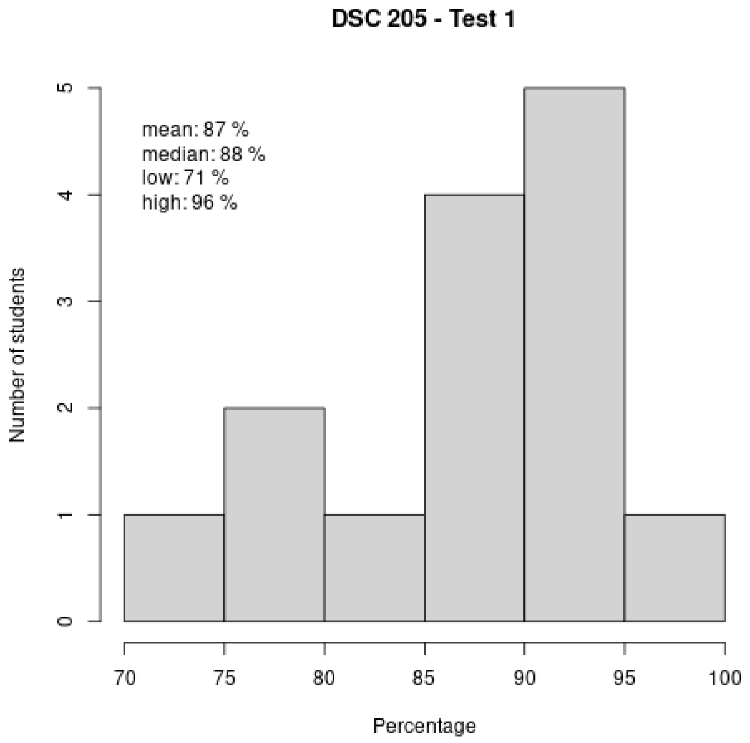
6. Plot the values as a histogram with `hist`.

```
hist(p)
```



7. Complete solution with `legend`:

```
hist(p,
     main="DSC 205 - Test 1",
     xlab="Percentage",
     ylab="Number of students")
legend("topleft",
       legend=paste(" mean:",format(mean(p),digits=2),"%\n",
    "median:",format(median(p),digits=2),"%\n",
    "low:",format(min(p),digits=2),"%\n",
    "high:",format(max(p),digits=2),"%\n"),
       bty="n")
```



8. How about putting the code into a script?

```
x <- readLines("https://tinyurl.com/ds2-test1-data") |> as.numeric()
x <- x[!is.na(x)]
p <- (x/25)*100
hist(p)
dev.off()
```

```
Warning message:
NAs introduced by coercion
null device
          1
```

9. What are the steps to run this script?

```
1. Emacs: M-x org-babel-tangle        -> file: ../src/grades.R
```

```
2. Shell: cd ../src; Rscript grades.R -> file: Rplots.pdf
3. Shell: evince Rplots.pdf           -> display histogram
```

10. How many tools did we just use?

    (a) Data uploading and sharing (`write`)
    (b) Check system data (`system`)
    (c) Fetch data directly from URL (`readLines`)
    (d) Check object characteristics (`is.vector`, `is.numeric`)
    (e) Check function environment (`environment`)
    (f) Access function definition in namespace (`::`)
    (g) Remove `NA` values from a vector (`!is.na`)
    (h) Plot a histogram with legend (`hist`, `legend`, `paste`, `format`)
    (i) Write and tangle an R script (`dev.off`)
    (j) Run R script on the shell (`Rscript`)