R Code – Best coding practices

Ken Mwai

1 – Naming conventions

- R has no standardised naming conventions
- Always choose a naming convention to work with; for example
  - all lowercase: e.g. adjustcolor
  - underscore separated: e.g. numeric_version
  - lowerCamelCase: e.g. addTaskCallback
  - UpperCamelCase: e.g. SignatureMethod
- **Avoid SPACES** while naming files

“There are only two hard things in Computer Science: cache invalidation and naming things.” — Phil Karlton

- Strive for names that are concise and meaningful
- R file names should be meaningful and end in .R.

Object names

- Variable and function names should be lowercase.

```
# Good
day_one
day_1

# Bad
first_day_of_the_month
DayOne
dayone
djm1
```

- avoid using names of existing functions and variables.

2 – Files organisation

- File organisation makes code and data analysis project readable
- Data should be separated from codes
- Documents should be separated from codes
- Use project facility of RStudio each time you start working on a new project

3 - organise the code within each file

- Start each file with a comment saying who wrote it and when, what it contains, and how it fits into the larger program
- Load all required packages (at the top of the script)
- Source required data files if any
- Modularize your code.

```r
#-----------------------------------------------
## f_StaR Introduction
## Ken Mwai - April 2022
#-----------------------------------------------
#-----------------------------------------------
# 0 - Load libraries
#-----------------------------------------------
library(dplyr) ## data cleaning
library(ggplot2) ## ploting
#-----------------------------------------------
# 1 - Source Data
#-----------------------------------------------
df1 <- read_csv("data/my_data.csv")
#-----------------------------------------------
# 2 - Start my code here: Calculating
#-----------------------------------------------
mean(df1$mpg)
```

3 – Syntax

- Place spaces around all infix operators (`=, +, -, <-, etc.`).
- Use <–, not =, for object assignment in R.
- Use comments to mark off sections of code.
- Comment your code with care. Comments should explain the why, not the what
- Each line of a comment should begin with the comment symbol and a single space
- Keep your lines less than 80 characters.

```r
# This is a comment
# Good
# Object assignment in R
x <- 10

#Bad
x=10
```

```r
# Good
average <- mean(feet / 12 + inches, na.rm = TRUE)

# Bad
average<-mean(feet/12+inches,na.rm=TRUE)
```

Use <-, not =, for assignment.

```r
# Good
x <- 5
# Bad
x = 5
```

Notes

- Do not include functions that change someone's computer i.e.
  - Installation of packages
  - `setwd()`
- Follow a style and be consistent.
  - https://google.github.io/styleguide/Rguide.html
  - https://style.tidyverse.org/files.html
  - https://bookdown.org/marius_mather/Rad/tips-for-effective-r-programming.html#writing-readable-code

Task

- Create a project folder with (data, code ,result , … folders)
- Use the mtcars dataset within R and check the structure
  - Take a first look at the data. Useful functions are `glimpse()`,`head()`,`str()` and `summary()`.
- Install the `janitor` package , use `get_dupes` to check duplicates in the mtcars `gear` variable.
- Remember to use good programming approach