Software quality analysis of fourteen hydrological models

Arnald Puy

Contents

1 Preliminary functions

```
#   PRELIMINARY FUNCTIONS ###########################################################

sensobol::load_packages(c("data.table", "tidyverse", "openxlsx", "scales",
                          "cowplot", "readxl", "ggrepel", "tidytext"))

# Create custom theme ----------------------------------------------------------

theme_AP <- function() {
  theme_bw() +
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          legend.background = element_rect(fill = "transparent", color = NA),
          legend.key = element_rect(fill = "transparent", color = NA),
          strip.background = element_rect(fill = "white"),
          legend.text = element_text(size = 7.3),
          axis.title = element_text(size = 10),
          legend.key.width = unit(0.4, "cm"),
          legend.key.height = unit(0.4, "cm"),
          legend.key.spacing.y = unit(0, "lines"),
          legend.box.spacing = unit(0, "pt"),
          legend.title = element_text(size = 7.3),
          axis.text.x = element_text(size = 7),
          axis.text.y = element_text(size = 7),
          axis.title.x = element_text(size = 7.3),
          axis.title.y = element_text(size = 7.3),
          plot.title = element_text(size = 8),
          strip.text.x = element_text(size = 7.4),
          strip.text.y = element_text(size = 7.4))
}

# Select color palette ---------------------------------------------------------

color_languages <- c("fortran" = "steelblue", "python" = "lightgreen")
```

2 Results

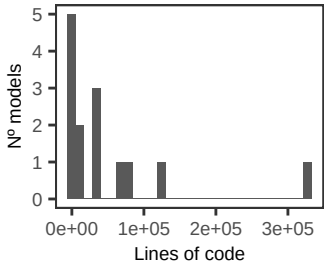
```
# READ IN DATASET ################################################################

# Get name of sheets -----------------------------------------------------------

sheets <- excel_sheets("./datasets/results_sqa.xlsx")

# Read all sheets --------------------------------------------------------------
```

```
dt <- lapply(sheets, function(x) data.table(read_excel("./datasets/results_sqa.xlsx",
                                                       sheet = x)))

# Name the slots ----------------------------------------------------------------

names(dt) <- sheets
```

2.1 Descriptive statistics

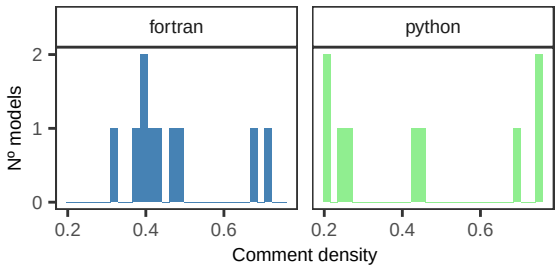
```
# PLOT LINES OF CODE ############################################################

plot_lines_code <- dt$descriptive_stats[, .(total_lines_code = sum(lines_code)), model] %>%
  ggplot(., aes(total_lines_code)) +
  geom_histogram() +
  labs(x = "Lines of code", y = "Nº models") +
  theme_AP()

plot_lines_code
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# PLOT COMMENT DENSITY ##########################################################

plot_comment_density <- dt$descriptive_stats[, .(total_lines_code = sum(lines_code),
       total_lines_comments = sum(lines_comments)), .(model, language)] %>%
  .[, comment_density:= total_lines_comments / total_lines_code] %>%
  ggplot(., aes(comment_density, fill = language)) +
  geom_histogram() +
  facet_wrap(~language) +
  scale_y_continuous(breaks = breaks_pretty(n = 3)) +
  scale_fill_manual(values = color_languages) +
  labs(x = "Comment density", y = "Nº models") +
  theme_AP() +
  theme(legend.position = "none")

plot_comment_density
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



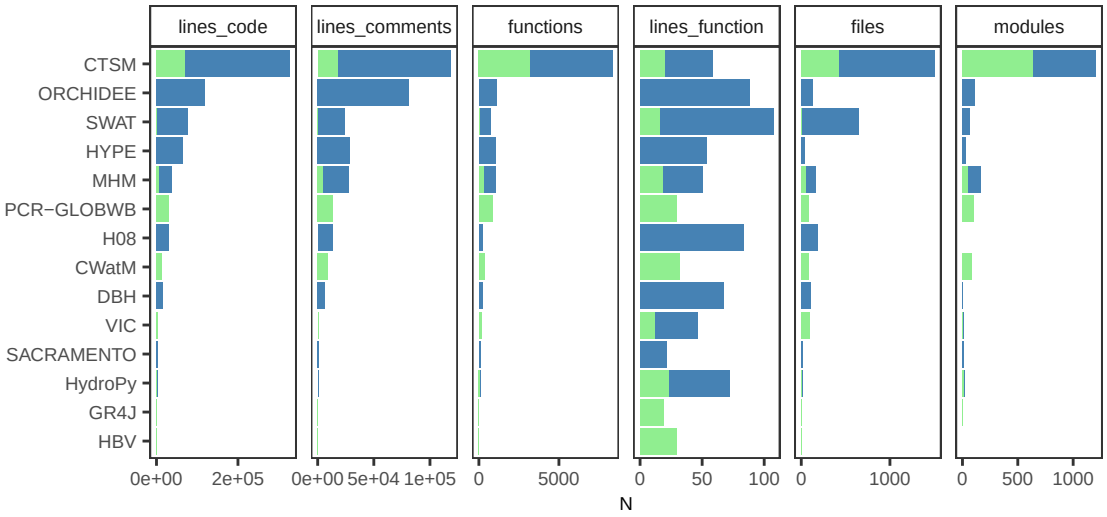

```
# PLOT PER MODEL ###################################################################

# Sort by model ----------------------------------------------------------------

model_ordered <- dt$descriptive_stats[, sum(lines), model] %>%
  .[order(V1)]

# Print ------------------------------------------------------------------------

model_ordered
```

```
##          model     V1
##         <char>  <num>
##  1:        HBV    180
##  2:       GR4J    423
##  3:    HydroPy   3739
##  4: SACRAMENTO   5294
##  5:        VIC   5952
##  6:        DBH  24334
##  7:      CWatM  27745
##  8:        H08  42917
##  9: PCR-GLOBWB  52686
## 10:        MHM  76286
## 11:       HYPE  89137
## 12:       SWAT  99976
## 13:   ORCHIDEE 211871
## 14:       CTSM 491592
```

```
# Extract column names ---------------------------------------------------------

col_names <- colnames(dt$descriptive_stats)

# Order facets -----------------------------------------------------------------

facet_order <- c("lines", "lines_code", "lines_comments", "functions",
                 "lines_function", "files", "modules")


# Plot -------------------------------------------------------------------------
```

```
plot_per_model <- melt(dt$descriptive_stats, measure.vars = col_names[-c(1, length(col_names))]
  .[, variable:= factor(variable, levels = facet_order)] %>%
  .[, model:= factor(model, levels = model_ordered[, model])] %>%
  .[!variable == "lines"] %>%
  ggplot(., aes(model, value, fill = language)) +
  geom_col() +
  coord_flip() +
  scale_y_continuous(breaks = breaks_pretty(n = 2)) +
  scale_fill_manual(values = color_languages) +
  facet_wrap(~ variable, ncol = 7, scales = "free_x") +
  labs(x = "", y = "N") +
  theme_AP() +
  theme(legend.position = "none")

plot_per_model
```

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_col()`).
```



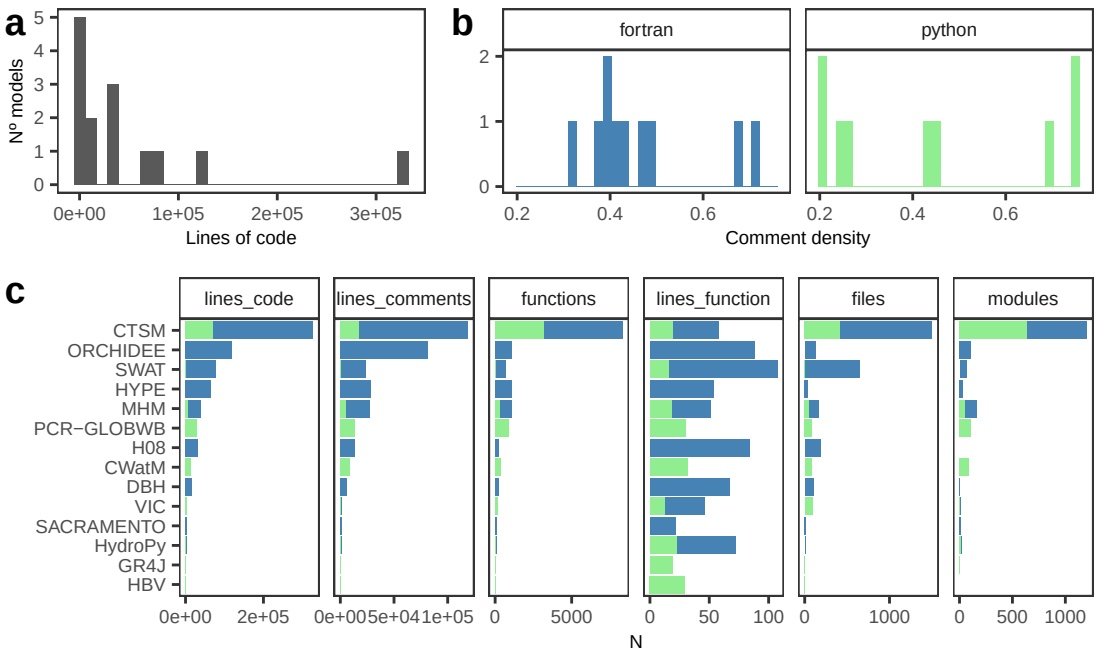

```
# MERGE PLOTS ######################################################################

top <- plot_grid(plot_lines_code, plot_comment_density + labs(x = "Comment density", y  = ""),
          labels = "auto", rel_widths = c(0.4, 0.6))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
p1 <- plot_grid(top, plot_per_model, ncol = 1, labels = c("", "c"), rel_heights = c(0.4, 0.6))
```

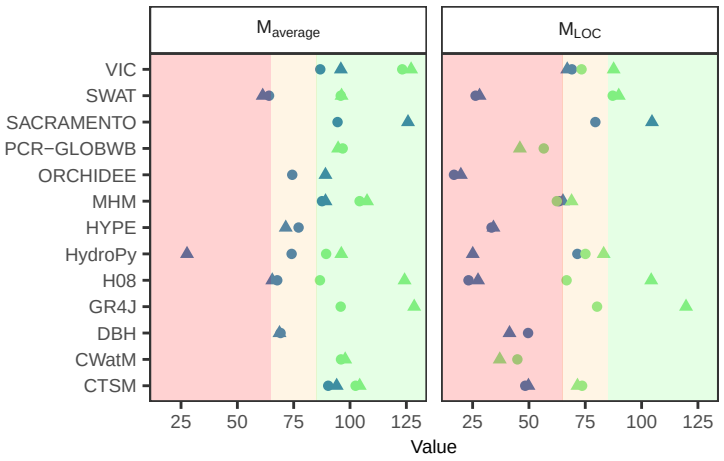
```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_col()`).
```

```
p1
```



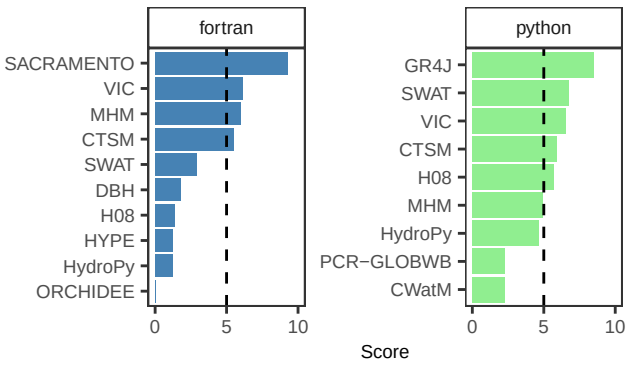
2.2 Maintainability index

```
# PLOT MAINTAINABILITY INDEX ###################################################

plot_maintainability_index <- dt$maintainability_index %>%
  melt(., measure.vars = c("M_loc", "M_average")) %>%
  .[, variable:= factor(variable, levels = c("M_average", "M_loc"))] %>%
  ggplot(., aes(model, value, color = language, shape = type)) +
  geom_point() +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = -Inf, ymax = 65,
           fill = "red", alpha = 0.18) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = 65, ymax = 85,
           fill = "orange", alpha = 0.1) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = 85, ymax = Inf,
           fill = "green", alpha = 0.1) +
  facet_wrap(~variable, labeller = as_labeller(c(M_loc = "M[LOC]",
                                                 M_average = "M[average]"),
                                               default = label_parsed)) +
  labs(x = "", y = "Value") +
  scale_color_manual(values = color_languages, guide = "none") +
  theme_AP() +
  theme(legend.position = "none") +
  coord_flip()

plot_maintainability_index
```

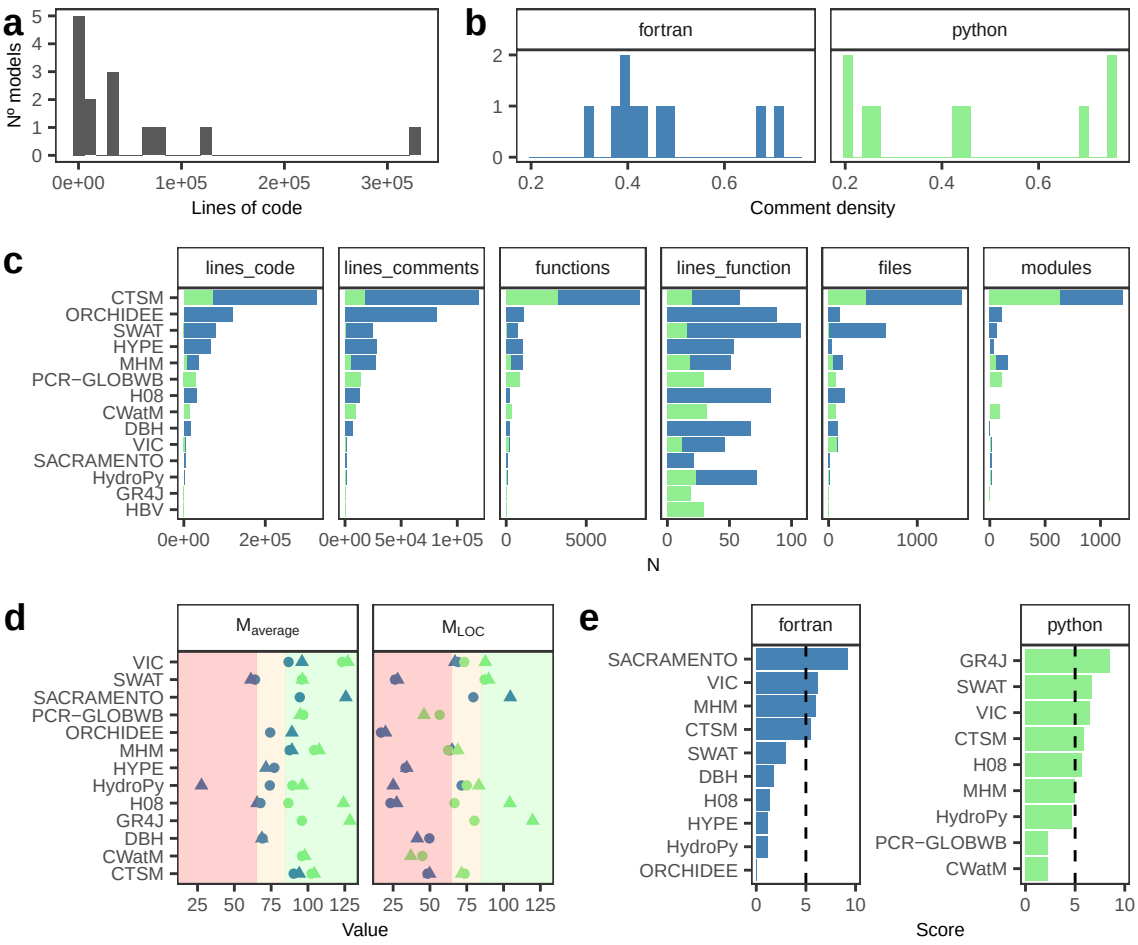


2.3 Score

```
# PLOT SCORE ##################################################################

plot_score <- dt$score %>%
  ggplot(aes(x = reorder_within(model, score, language), y = score, fill = language)) +
  geom_bar(stat = "identity") +
  facet_wrap(~ language, scales = "free_y") +
  labs(x = "", y = "Score") +
  scale_fill_manual(values = color_languages) +
  geom_hline(yintercept = 5, lty = 2) +
  scale_x_reordered() +
  scale_y_continuous(limits = c(0, 10), breaks = c(0, 5, 10)) +
  coord_flip() +
  theme_AP() +
  theme(legend.position = "none")

plot_score
```




```
# MERGE PLOTS #################################################################

bottom <- plot_grid(plot_maintainability_index, plot_score, ncol = 2, labels = c("d", "e"))

plot_grid(p1, bottom, ncol = 1, rel_heights = c(0.62, 0.38))
```


a
Nº models
Lines of code
b
fortran
python
Comment density
c
lines_code
lines_comments
functions
lines_function
files
modules
CTSM
ORCHIDEE
SWAT
HYPE
MHM
PCR−GLOBWB
H08
CWatM
DBH
VIC
SACRAMENTO
HydroPy
GR4J
HBV
N
d
M_average
M_LOC
Value
e
fortran
python
Score