Software quality analysis

Arnald Puy

Contents

1 Preliminary functions

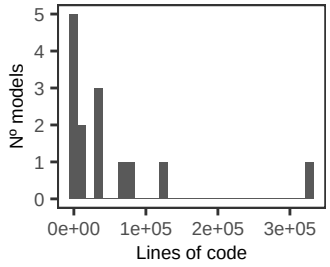
```
#   PRELIMINARY FUNCTIONS ###########################################################

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

# 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 ---------------------------------------------------------

selected.palette <- "Darjeeling1"
```

2 Dataset

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

dt <- data.table(read.xlsx("./datasets/results_sqa.xlsx"))
cols <- colnames(dt)
color_languages <- c("fortran" = "steelblue", "python" = "lightgreen")
```

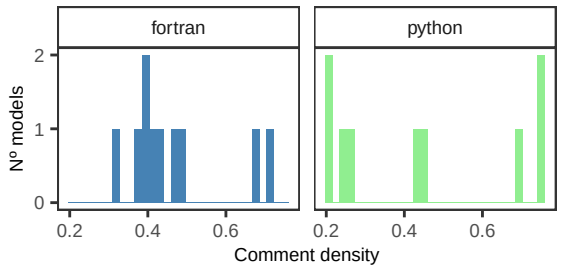
```r
# PLOT LINES OF CODE ##############################################################

plot_lines_code <- dt[, .(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`.
```




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

plot_comment_density <- dt[, .(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`.
```




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

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`.
```