

basic-pass-at-k

2022-08-06

Basic Pass at K Calculation

Goal of this R file is to read in the data from `all-pass-at-1-eval-run.csv` and plot it according to language and to sublanguage. Exact plot types are not at all final, but the idea is to do a first pass at reading in the data.

```
all_pass_at_1_eval_run <- read_csv("~/polyglot-codegen-evaluation/model_results/all-pass-at-1-eval-run.csv",
  col_names = FALSE, col_types = cols(X6 = col_integer(),
    X7 = col_integer(), X8 = col_integer()))
pass_at_1 <- tibble::as_tibble(all_pass_at_1_eval_run)
pass_at_1 <- dplyr::rename(pass_at_1, PL = X1, MODEL=X2, TEMP=X3, DOCS=X4, TERMS=X5,
  MIN_COMPLETE=X6, K=X7, MIN_PROBLEM=X8, RES=X9)

all_pass_at_10_eval_run <- read_csv("~/polyglot-codegen-evaluation/model_results/all-pass-at-10-eval-run.csv",
  col_names = FALSE, col_types = cols(X6 = col_integer(),
    X7 = col_integer(), X8 = col_integer()))
pass_at_10 <- tibble::as_tibble(all_pass_at_10_eval_run)
pass_at_10 <- dplyr::rename(pass_at_10, PL = X1, MODEL=X2, TEMP=X3, DOCS=X4, TERMS=X5,
  MIN_COMPLETE=X6, K=X7, MIN_PROBLEM=X8, RES10=X9)

frequency_map <- read_csv("frequency-map.csv", col_names = FALSE) %>%
  dplyr::rename(PL=X1, CATEGORY=X2)

## Rows: 19 Columns: 2
## -- Column specification -----
## Delimiter: ","
## chr (2): X1, X2
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
#pass_results<- tibble::add_column(pass_at_1, pass_at_10$RES10)

#pass_results<- dplyr::rename(pass_results, RES10 = `pass_at_10$RES10`)

#remove unneeded columns
#pass_results %>% dplyr::select(-c(MIN_COMPLETE, MIN_PROBLEM, K))

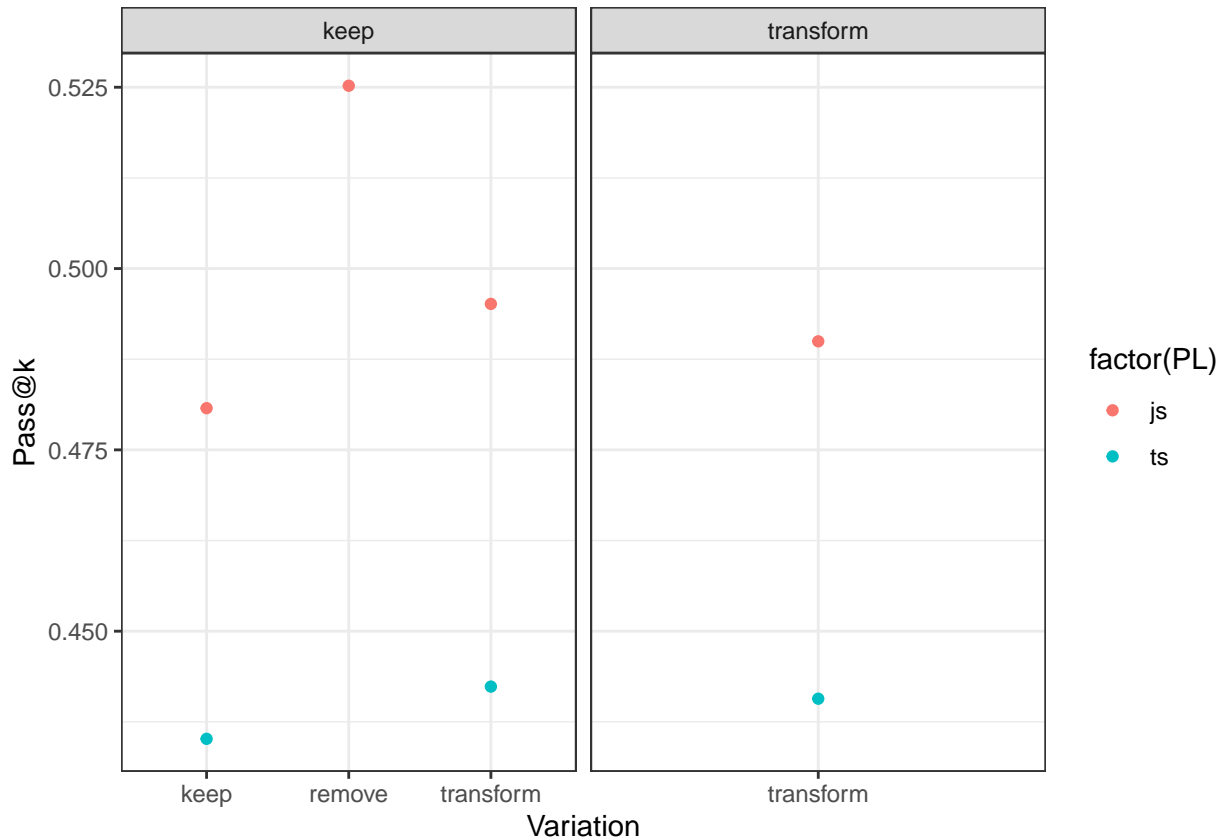
#make a davinci only data set
davinci_only <- dplyr::filter(pass_at_1, pass_at_1$MODEL == 'davinci')
```

As a first pass, let's just plot all of the versions of the Python runs, which are arguably relatively simple to handle

```
# python_only <- dplyr::filter(pass_results, (pass_results$PL == "py"))
# print(python_only)
# ggplot(python_only, aes(x=DOCS, y=RES, col=factor(MODEL))) + geom_point() + geom_point(data=python_only,
```

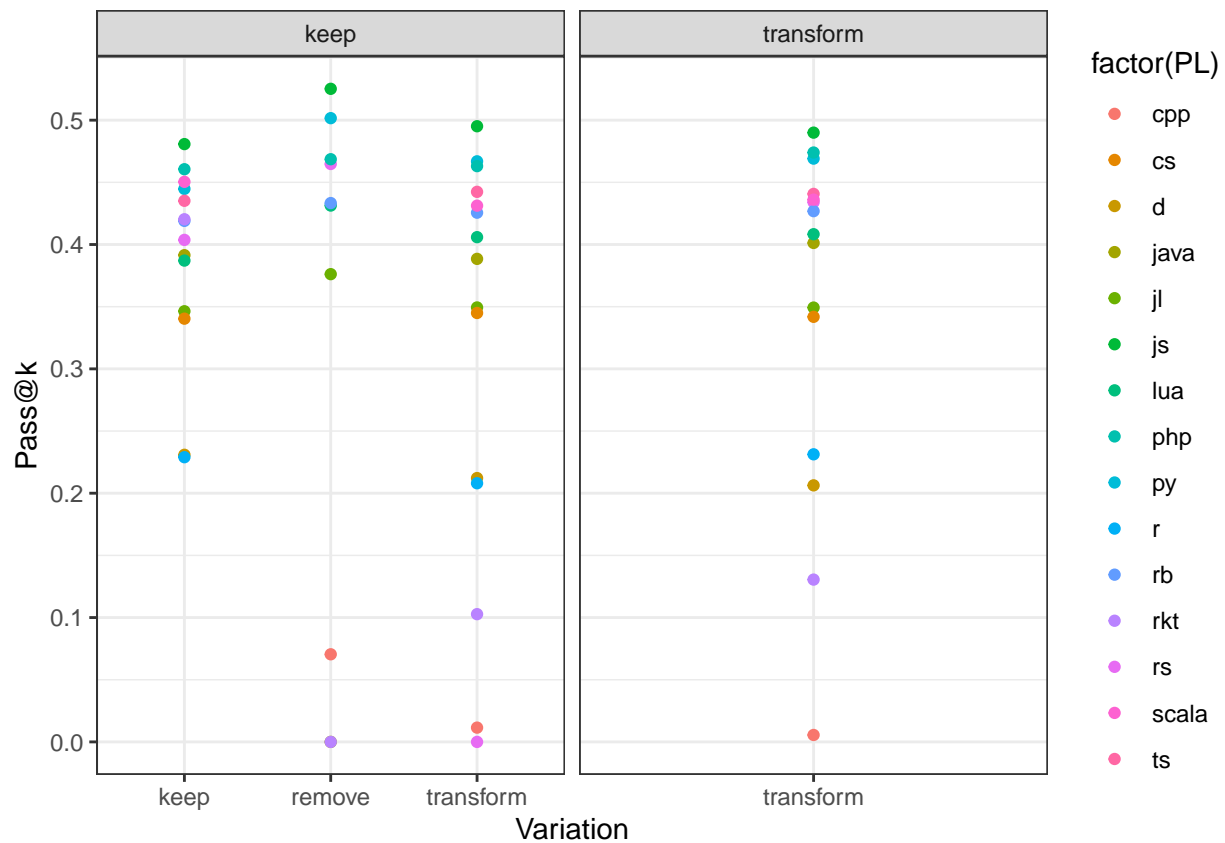
Plot only JavaScript versus TypeScript on Davinci:

```
js_v_ts <- dplyr::filter(davinci_only, (davinci_only$PL == "js" | davinci_only$PL == "ts"))
ggplot(js_v_ts, aes(x=DOCS, y=RES, col=factor(PL))) +
  ylab("Pass at K Rate") + xlab("Variation") +
  geom_point() +
  facet_grid(~ TERMS, scales = 'free') + theme_bw() + ylab("Pass@k") + xlab("Variation")
```



Plot all languages on Davinci 0.2:

```
ggplot(davinci_only, aes(x=DOCS, y=RES, col=factor(PL))) +
  ylab("Pass at K Rate") + xlab("Variation") +
  geom_point() +
  facet_grid(~ TERMS, scales = 'free') + theme_bw() + ylab("Pass@k") + xlab("Variation")
```



Do Some High Level Stats

```
category_data <- dplyr::left_join(davinci_only, frequency_map)
```

```
## Joining, by = "PL"
```