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.
        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-ru
        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)

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))
```

```
## # A tibble: 88 x 7
##    PL    MODEL    TEMP DOCS      TERMS       RES  RES10
##    <chr> <chr>   <dbl> <chr>     <chr>     <dbl>  <dbl>
##  1 py    davinci   0.2 transform transform 0.469 0.671
##  2 py    davinci   0.2 keep      keep      0.445 0.646
##  3 py    davinci   0.2 remove    keep      0.502 0.704
##  4 py    davinci   0.2 transform keep      0.467 0.662
##  5 py    incoder   0.2 transform transform 0.045 0.0985
##  6 py    incoder   0.2 keep      keep      0.131 0.215
##  7 py    incoder   0.2 remove    keep      0.180 0.280
##  8 py    incoder   0.2 transform keep      0.156 0.250
##  9 ts    davinci   0.2 transform transform 0.441 0.635
## 10 ts    davinci   0.2 keep      keep      0.435 0.631
## # ... with 78 more rows
## # i Use `print(n = ...)` to see more rows
```

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

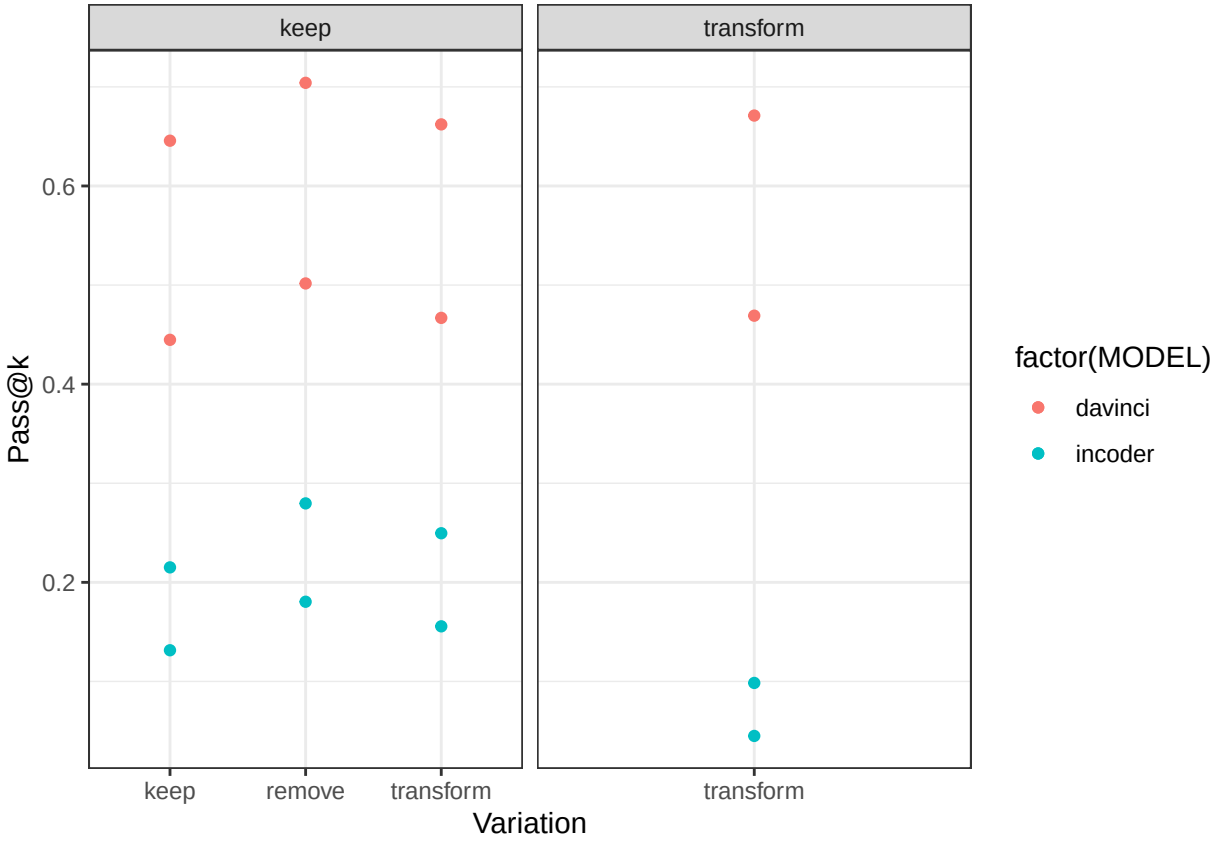
As a first pass, let's just plot all of the versions of the Python runs, which are arguably relatively simple to

handle

```
python_only <- dplyr::filter(pass_results, (pass_results$PL == "py"))
print(python_only)
```

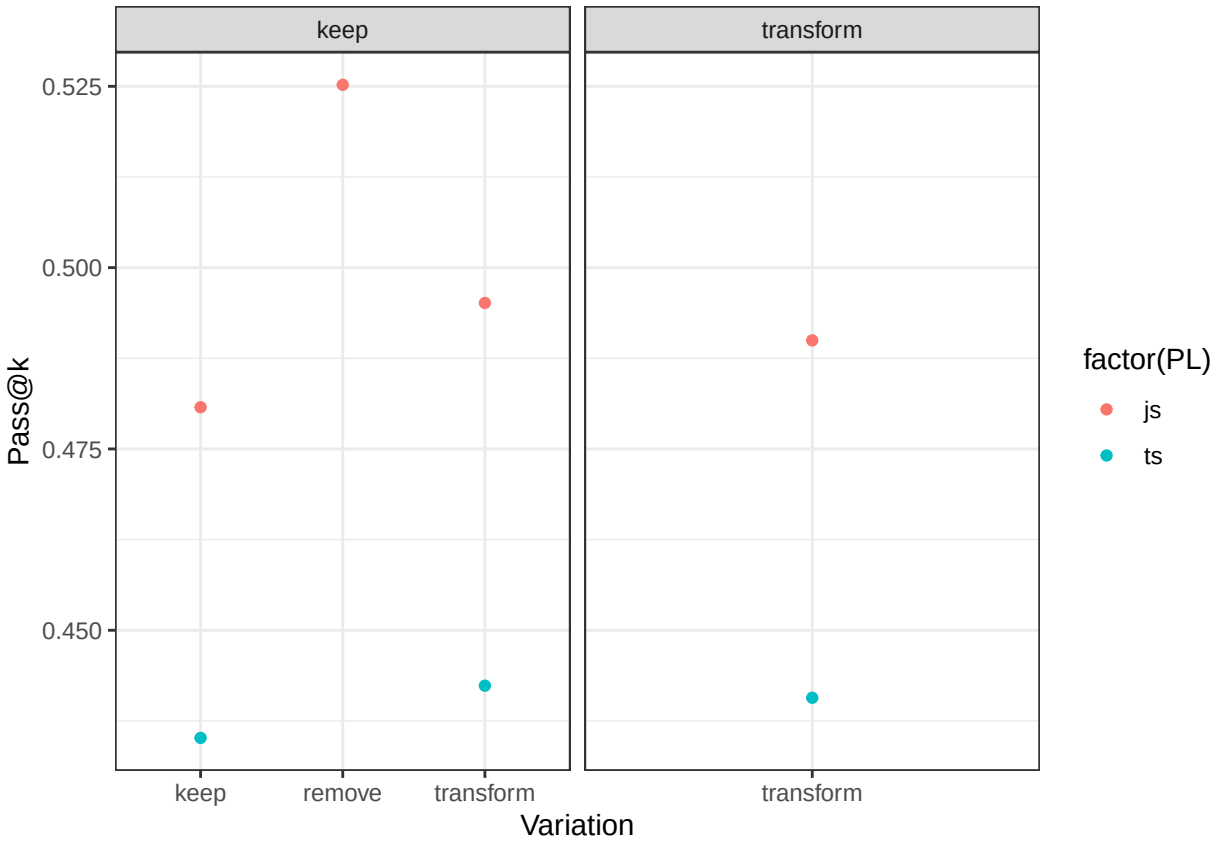
```
## # A tibble: 8 x 10
##   PL    MODEL    TEMP DOCS      TERMS     MIN_COMPL~1     K MIN_P~2   RES  RES10
##   <chr> <chr>   <dbl> <chr>     <chr>           <int> <int>   <int> <dbl>  <dbl>
## 1 py    davinci   0.2 transform transform          20     1     158 0.469 0.671
## 2 py    davinci   0.2 keep      keep              200     1     153 0.445 0.646
## 3 py    davinci   0.2 remove    keep              200     1     120 0.502 0.704
## 4 py    davinci   0.2 transform keep               20     1     161 0.467 0.662
## 5 py    incoder   0.2 transform transform          20     1      40 0.045 0.0985
## 6 py    incoder   0.2 keep      keep              200     1     156 0.131 0.215
## 7 py    incoder   0.2 remove    keep              200     1     120 0.180 0.280
## 8 py    incoder   0.2 transform keep               20     1     161 0.156 0.250
## # ... with abbreviated variable names 1: MIN_COMPLETE, 2: MIN_PROBLEM
```

```
ggplot(python_only, aes(x=DOCS, y=RES, col=factor(MODEL))) + geom_point()+ geom_point(data=python_only,a
```



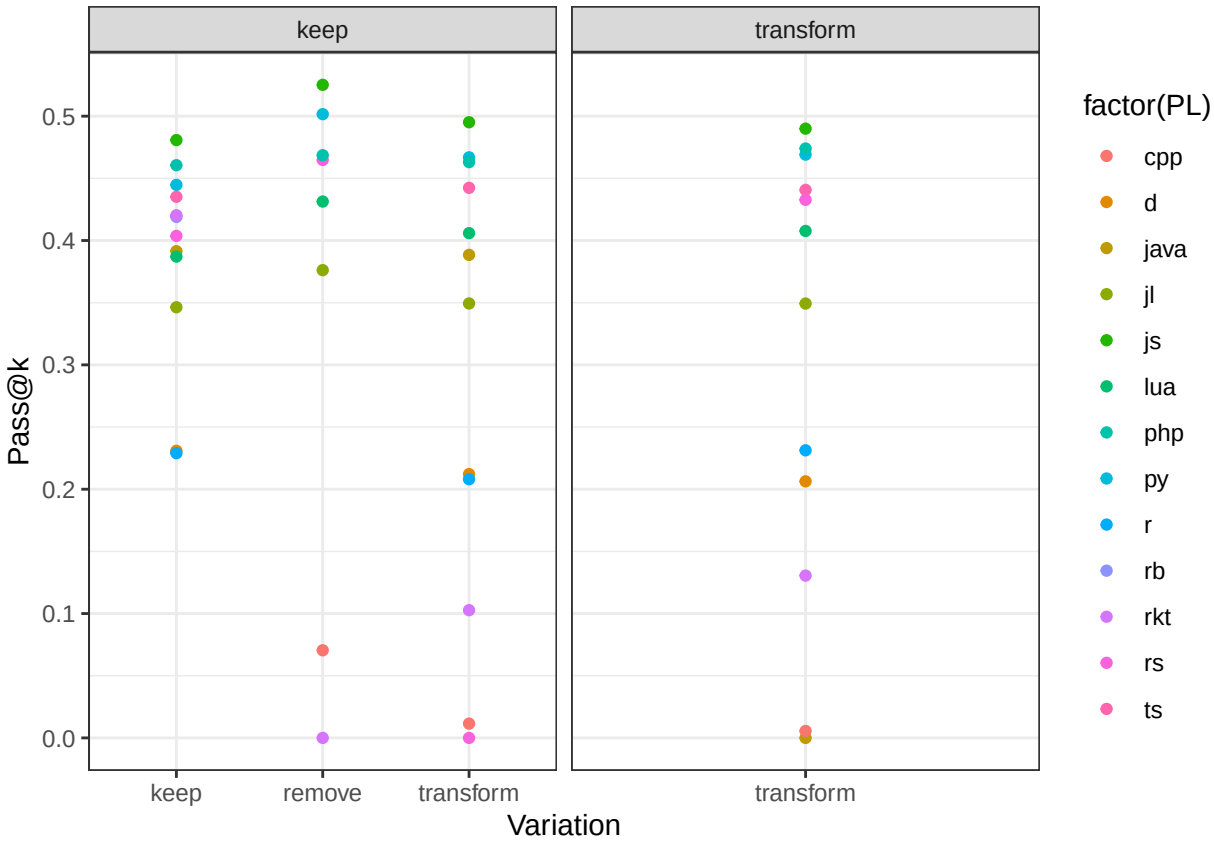
Plot only JavaScript versus TypeScript on Davinci:

```
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")
```


keep
transform
Pass@k
0.5
0.4
0.3
0.2
0.1
0.0
keep
remove
transform
transform
Variation
factor(PL)
cpp
d
java
jl
js
lua
php
py
r
rb
rkt
rs
ts