

```

library(dplyr)
library(ggplot2)

parse_duration <- function(duration) {
  components = strsplit(duration, ":")
  return(unlist(lapply(components, function(row) as.numeric(row[1])*60 + as.numeric(row[2]))))
}

next_smallest <- function(cmps, list) {
  return(unlist(lapply(cmps, function(cmp) {
    min_diff = 10
    min_el = 0
    for (el in list) {
      if (el < cmp && cmp-el < min_diff) {
        min_el = el
        min_diff = cmp-el
      }
    }
    return(min_el)})))
}

langs <- new.env()
langs$Julia <- "../julia_results"
langs$Python <- "../python_results"
langs$C <- "../c_results"
langs$JuliaNoTypes <- "../julia_results_noty"

all_data = lapply(names(langs), function (lang)
  read.csv(langs[[lang]],
    col.names=c("benchmark", "erase", "time"), stringsAsFactors=FALSE,
    sep=" ") %>% select(benchmark, time) %>%
    mutate(duration=parse_duration(time), language=lang) %>% select(-time) %>%
    filter(benchmark != "meteor_contest")) %>% bind_rows
all_data$language = as.factor(all_data$language)
all_data$benchmark = as.factor(all_data$benchmark)

all_data = all_data %>% group_by(benchmark) %>% mutate(frac = duration/max(duration)) %>%
  mutate(frac=frac-next_smallest(frac, frac))

ggplot(all_data, aes(x=benchmark, y=frac)) + geom_col(aes(fill=language))

```

