R Notebook

```
names <- c("name", "n", "alloc", "free", "exec")
types <- c("c", "i", "i", "i")

ds1 <- read_csv("datasets/data.mac.csv", col_names = names, col_types = cols())
ds2 <- read_csv("datasets/data.gpu.csv", col_names = names, col_types = cols())
ds3 <- read_csv("datasets/data.gpuvec.csv", col_names = names, col_types = cols())

data <-
  full_join(ds1, ds2) %>%
  full_join(ds3)
```

```
## Joining, by = c("name", "n", "alloc", "free", "exec")
## Joining, by = c("name", "n", "alloc", "free", "exec")
```

```
# Adding helper columns for grouping
data <-
  data %>%
  mutate(algo = str_extract(name, "^.*(?=\\.)")) %>%
  mutate(setup = str_extract(name, "(?=.)\\w+$"))

df <-
  data %>%
  group_by(n, setup, algo) %>%
  summarise(
    exec = mean(exec)/1e3,
    free = mean(free)/1e3,
    alloc = mean(alloc)/1e3,
  ) %>%
  mutate(
    memtot = alloc + free
  )
```

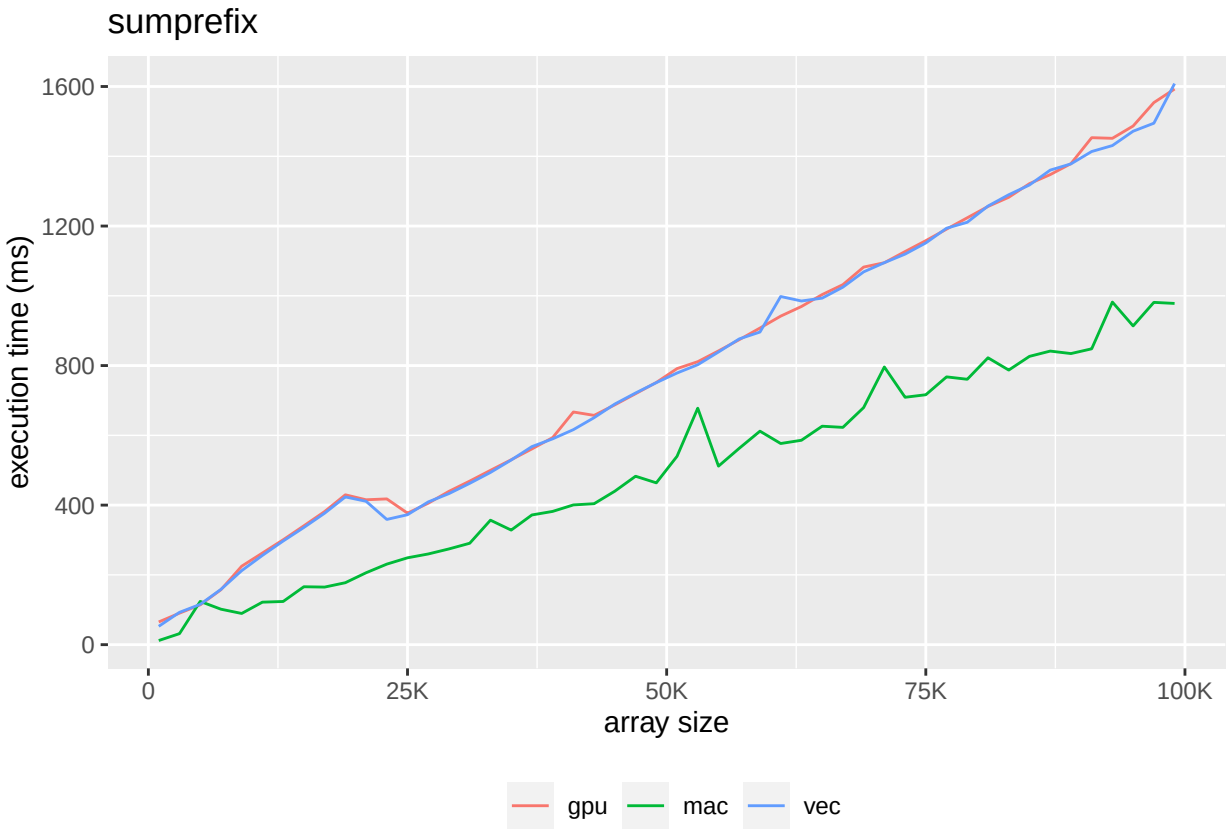
```
## `summarise()` regrouping output by 'n', 'setup' (override with `.groups` argument)
```

```
plot.mem <- function(df, a = "sumprefix") {
  df %>%
  group_by(free, alloc, memtot) %>%
  filter(algo == a) %>%
  ggplot() +
  ggtitle(a, subtitle = "memory measurements") +
  geom_line(aes(x = n, y = free, colour = "free()")) +
  geom_line(aes(x = n, y = alloc, colour = "alloc()")) +
  geom_line(aes(x = n, y = memtot, colour = "tot")) +
  scale_x_continuous("array size", labels = scales::label_number_si()) +
```

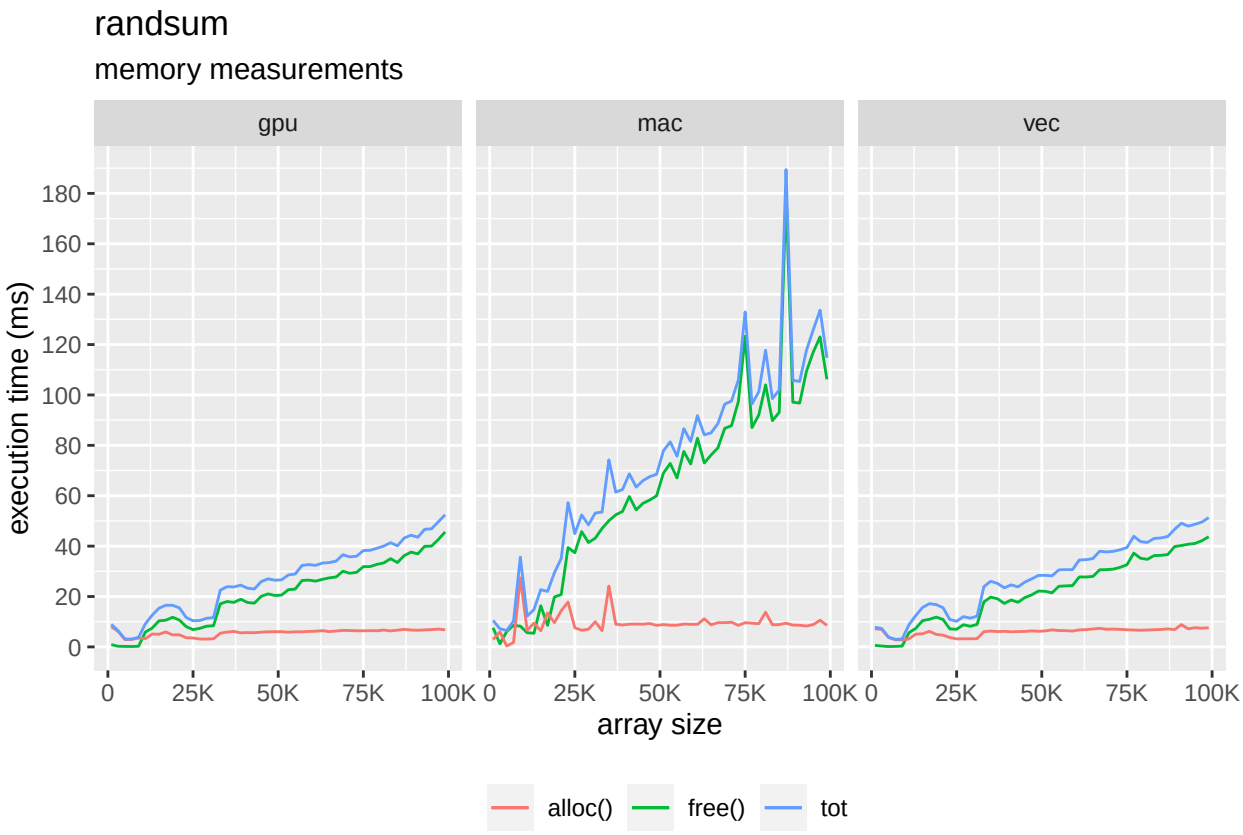
```
  scale_y_continuous("execution time (ms)", breaks = scales::breaks_width(20)) +
  facet_wrap(vars(setup)) +
  theme(legend.position = "bottom", legend.title = element_blank())
}

plot.exec <- function(df, a = "sumprefix") {
  df %>%
  filter(algo == a) %>%
  ggplot() +
  ggtitle(a) +
  geom_line(aes(x = n, y = exec, colour = setup)) +
  scale_x_continuous("array size", labels = scales::label_number_si()) +
  scale_y_continuous("execution time (ms)", breaks = scales::breaks_width(400)) +
  theme(legend.position = "bottom", legend.title = element_blank())
}
```

```
a <- "sumprefix"
plot.mem(df, a=a)
```

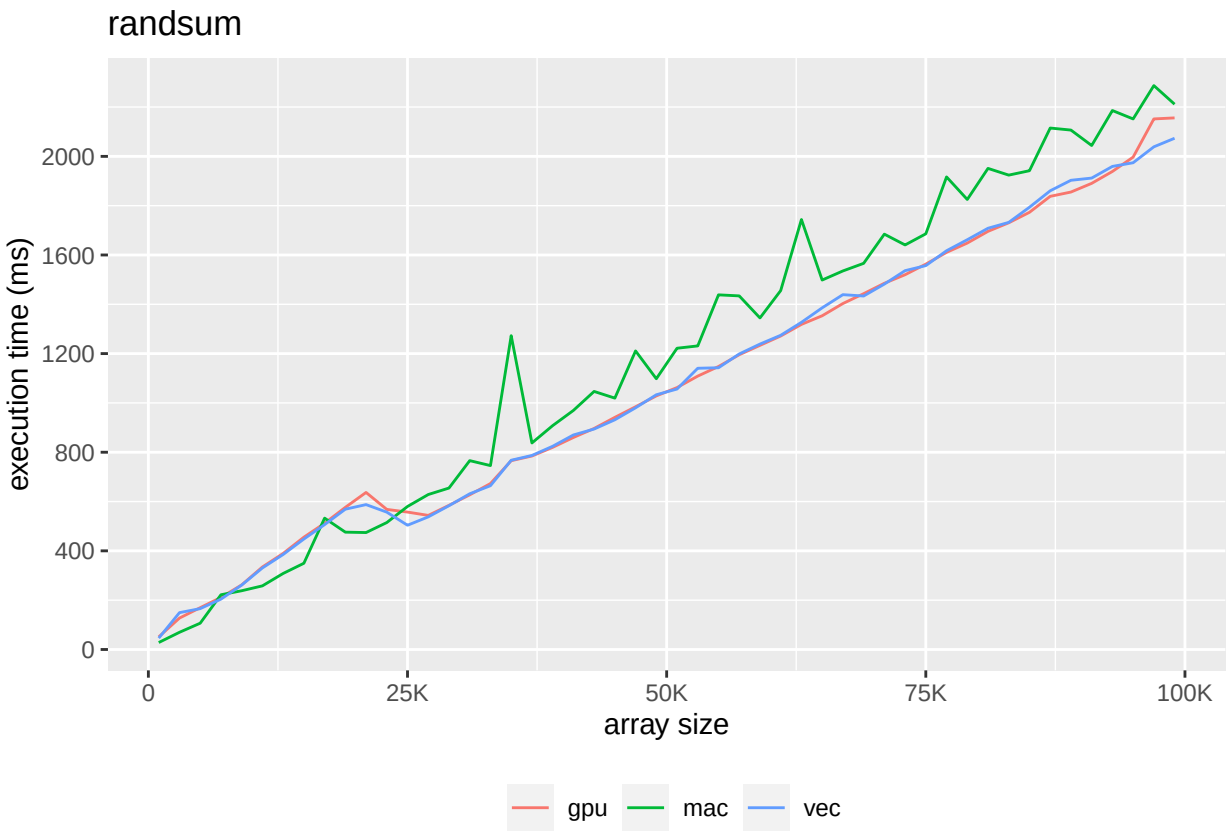

```
plot.exec(df, a=a)
```




```
a <- "randsum"
plot.mem(df, a=a)
```




```
plot.exec(df, a=a)
```


randsum
execution time (ms)
2000
1600
1200
800
400
0
0
25K
50K
75K
100K
array size
gpu
mac
vec