

Analysis of matrix spec bias-rumors

MTRNord

The following data is based purely on public knowledge. This means data is fetched from github and gitlab as best as I was able to.

Fetching the MSCs

```
library(gh)
library(tidyverse)
library(nplyr)

# Set theme
theme_set(theme_bw())

# gh_whoami()

# Fetch all issues that are a proposal
if (!exists("issues_all")) {
  issues_raw <- gh("GET /repos/matrix-org/matrix-spec-proposals/issues",
    sort = "created", state = "all", direction = "asc", labels = "proposal")
  issues <- as.data.frame(do.call(rbind, issues_raw))

  data <- list()
  data[[1]] <- issues

  # Paginate API
  while (TRUE) {
    issues_raw <- try({
      gh_next(issues_raw)
    }, silent = TRUE)
    if (inherits(issues_raw, "try-error")) {
      break
    } else {
      temp <- as.data.frame(do.call(rbind, issues_raw))
      data[[length(data) + 1]] <- temp
    }
  }
  issues_all <- do.call(rbind, data)
}

# Filter out corrupt issues without labels
filtered_issues <- issues_all |>
  filter(sapply(labels, length, simplify = TRUE) > 0) |>
```

```

rowwise()
filtered_issues$labels <- map(filtered_issues$labels, ~do.call(rbind,
.))

```

Get Employee association from Github and Gitlab

Please note that in the current PDF this is not yet hooked up to gitlab or checking the github workplace field. It may also exclude some users that are not detectable.

```

# TODO also check against gitlab TODO also check workplace
# thingy

# Compile a list of who is who
element_employee <- list()
famedly_employee <- list()
beeper_employee <- list()
users <- list()
# Get orgs of users on github
for (i in 1:nrow(filtered_issues)) {
  user <- filtered_issues[i, ]$user[[1]]$login
  if (user %in% users) {
    next
  }
  orgs_raw <- gh(sprintf("GET /users/%s/orgs", user))
  orgs <- as.data.frame(do.call(rbind, orgs_raw))

  if ("vector-im" %in% orgs$login) {
    element_employee[[length(element_employee) + 1]] <- user
  } else if ("beeper" %in% orgs$login) {
    beeper_employee[[length(beeper_employee) + 1]] <- user
  } else if ("Famedly" %in% orgs$login) {
    famedly_employee[[length(famedly_employee) + 1]] <- user
  }
  users[[length(users) + 1]] <- user
}

```

MSCs by Company (all kind)

Note that this does not adjust for private vs company MSCs.

```

# Filter MSCs by company
merged_element <- filtered_issues |>
  filter(user$login %in% element_employee) |>
  nrow()

merged_famedly <- filtered_issues |>
  filter(user$login %in% famedly_employee) |>
  nrow()

```

```

merged_beeper <- filtered_issues |>
  filter(user$login %in% beeper_employee) |>
  nrow()

merged_other <- filtered_issues |>
  filter(!(user$login %in% element_employee)) |>
  filter(!(user$login %in% famedly_employee)) |>
  filter(!(user$login %in% beeper_employee)) |>
  nrow()

# Display Data
data <- data.frame(group = c("Element", "Beeper", "Famedly",
  "Other"), value = c(merged_element, merged_beeper, merged_famedly,
  merged_other))

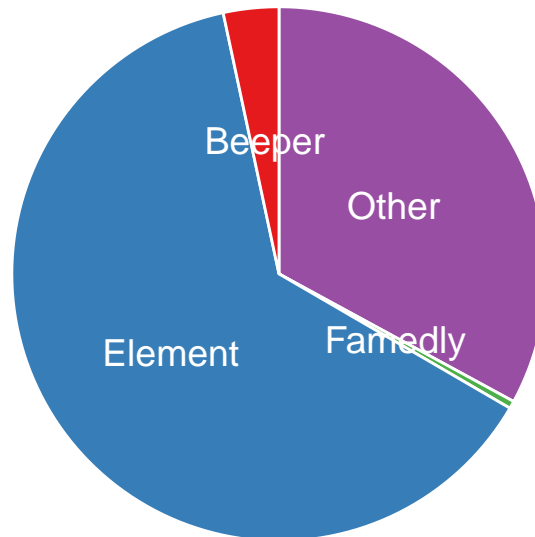
# Compute the position of labels
data <- data |>
  filter(value != 0) |>
  arrange(desc(group)) |>
  mutate(prop = value/sum(data$value) * 100) |>
  mutate(ypos = cumsum(prop) - 0.5 * prop)

# Basic piechart
ggplot(data, aes(x = "", y = prop, fill = group)) + geom_bar(stat = "identity",
  width = 1, color = "white") + coord_polar("y", start = 0) +
  theme_void() + labs(title = str_wrap("Percentage of MSCs by Contributors associated with companies"
  40), subtitle = str_wrap("Note that people may have gotten mixed or people with multiple hats may h
  60), caption = "source: Github API") + theme(legend.position = "none") +
  geom_text(aes(y = ypos, label = group), color = "white",
  size = 5) + scale_fill_brewer(palette = "Set1")

```

Percentage of MSCs by Contributors associated with companies

Note that people may have gotten mixed or people with multiple hats may have MSCs landing in the wrong category



source: Github API

Merged MSCs by Company

Note that this does not adjust for private vs company MSCs.

```
# Filter for only merged MSCs
merged_mscs <- filtered_issues |>
  filter("proposal" %in% labels) |>
  filter(("disposition-merge" %in% labels) | ("merged" %in%
    labels))

# Filter MSCs by company
merged_element <- merged_mscs |>
  filter(user$login %in% element_employee) |>
  nrow()

merged_famedly <- merged_mscs |>
  filter(user$login %in% famedly_employee) |>
  nrow()

merged_beeper <- merged_mscs |>
  filter(user$login %in% beeper_employee) |>
  nrow()
```

```

merged_other <- merged_mscs |>
  filter(!(user$login %in% element_employee)) |>
  filter(!(user$login %in% famedly_employee)) |>
  filter(!(user$login %in% beeper_employee)) |>
  nrow()

# Display Data
data <- data.frame(group = c("Element", "Beeper", "Famedly",
  "Other"), value = c(merged_element, merged_beeper, merged_famedly,
  merged_other))

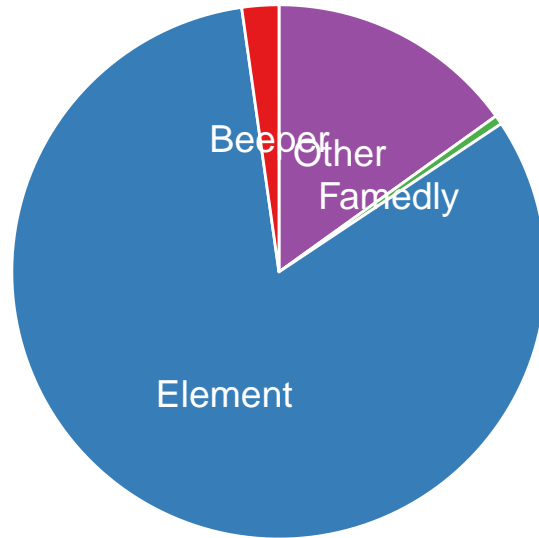
# Compute the position of labels
data <- data |>
  filter(value != 0) |>
  arrange(desc(group)) |>
  mutate(prop = value/sum(data$value) * 100) |>
  mutate(ypos = cumsum(prop) - 0.5 * prop)

# Basic piechart
ggplot(data, aes(x = "", y = prop, fill = group)) + geom_bar(stat = "identity",
  width = 1, color = "white") + coord_polar("y", start = 0) +
  theme_void() + labs(title = str_wrap("Percentage of merged MSCs by Contributors associated with comp
  40), subtitle = str_wrap("Note that people may have gotten mixed or people with multiple hats may h
  60), caption = "source: Github API") + theme(legend.position = "none") +
  geom_text(aes(y = ypos, label = group), color = "white",
  size = 5) + scale_fill_brewer(palette = "Set1")

```

Percentage of merged MSCs by
Contributors associated with companies

Note that people may have gotten mixed or people with multiple hats may have MSCs landing in the wrong category



source: Github API