Final Thoughts

Overview

Plotting results of coding analysis of Q12: “Are there any other challenges you’ve encountered in open source, or types of support that you would find helpful?”

Import packages and utilities

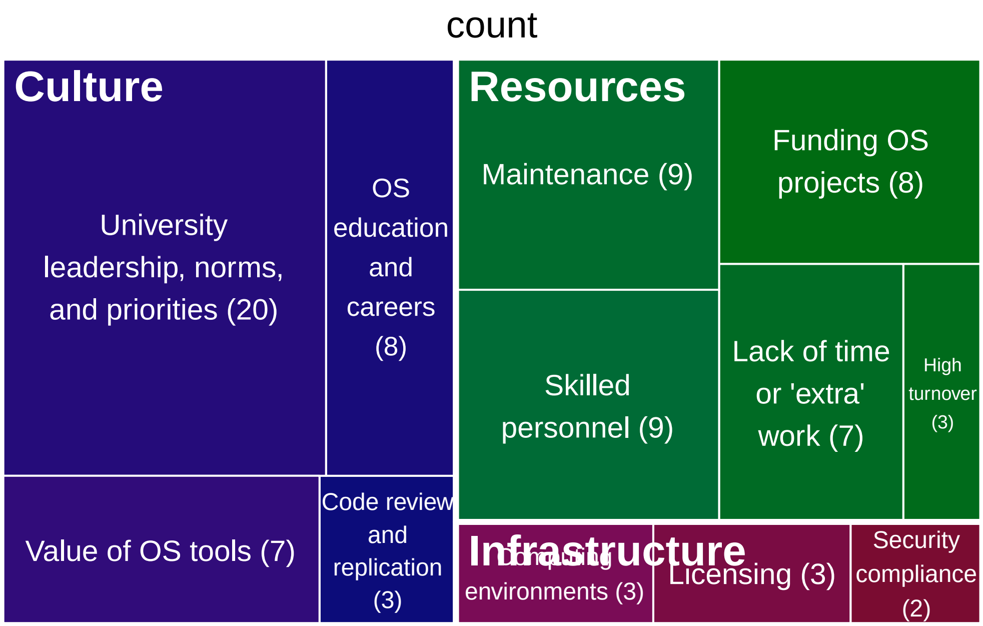
```r
project_root <- here::here() # requires that you be somewhere in the
# project directory (not above it)
# packages
suppressMessages(source(file.path(project_root, "scripts/packages.R")))
# functions and objects used across scripts
suppressMessages(source(file.path(project_root, "scripts/utils.R")))
```

I did this analysis in Taguette. Then I manually copied the number of comments associated with each tag from Taguette to this notebook. Each comment can have multiple tags. I’m creating the dataframe row-wise using tribble because that makes it a little easier to read and change, and I expect these fluid categories might change.

```r
results <- tribble(
  ~theme,                                    ~category,       ~count,
  "Maintenance",                                 "Resources",      9,
  "Funding OS projects",                         "Resources",      8,
  "High turnover",                               "Resources",      3,
  "Skilled personnel",                           "Resources",      9,
  "Lack of time or 'extra' work",                "Resources",      7,
  "Security compliance",                         "Infrastructure", 2,
  "Licensing",                                   "Infrastructure", 3,
  "Computing environments",         "Infrastructure", 3,
  "University leadership, norms, and priorities","Culture",       20,
  "Code review and replication",                 "Culture",        3,
  "OS education and careers",                    "Culture",        8,
  "Value of OS tools",                           "Culture",        7
)

results$theme <- paste0(results$theme, " (", results$count, ")")
```

Now let's plot a treemap!

```
my_treemap <- treemap(
  results,
  index = c("category", "theme"),  # first level "category", then theme
  vSize = "count",                     # area of each rectangle proportional to count
  type  = "index",                     # color by index (i.e. category)
  palette = c(colors[1], colors[8], colors[4]),                   # from utils.R

  # Label styling
  lowerbound.cex.labels = 0,                # multiplier between 0 and 1 that sets the lowerb
  fontsize.labels     = c(16,  11),                # e.g. 14pt for categories, 8pt for themes
  fontcolor.labels    = c("white", "white"),      # white for themes and categories
  align.labels = list(
    c("left", "top"),
    c("center", "center")
    ),

  # aesthetics
  border.col = "white",                # white borders for readability
  fontsize.title = 14,
  bg.labels = 0                        # transparent label backgrounds
)
```



```
my_treemap
```

```
$tm
         category                                             theme vSize
1         Culture                   Code review and replication (3)     3
2         Culture                                              <NA>    38
3         Culture                      OS education and careers (8)     8
4         Culture University leadership, norms, and priorities (20)    20
5         Culture                             Value of OS tools (7)     7
6  Infrastructure                        Computing environments (3)     3
7  Infrastructure                                     Licensing (3)     3
8  Infrastructure                                              <NA>     8
9  Infrastructure                           Security compliance (2)     2
10      Resources                           Funding OS projects (8)     8
11      Resources                                 High turnover (3)     3
12      Resources                  Lack of time or 'extra' work (7)     7
13      Resources                                   Maintenance (9)     9
14      Resources                                              <NA>    36
15      Resources                             Skilled personnel (9)     9
   vColor stdErr vColorValue level        x0        y0         w         h
1       1      3          NA     2 0.3243902 0.0000000 0.1390244 0.2631579
2       4     38          NA     1 0.0000000 0.0000000 0.4634146 1.0000000
```

```
3       1      8          NA     2 0.3310105 0.2631579 0.1324042 0.7368421
4       1     20          NA     2 0.0000000 0.2631579 0.3310105 0.7368421
5       1      7          NA     2 0.0000000 0.0000000 0.3243902 0.2631579
6       1      3          NA     2 0.4634146 0.0000000 0.2012195 0.1818182
7       1      3          NA     2 0.6646341 0.0000000 0.2012195 0.1818182
8       3      8          NA     1 0.4634146 0.0000000 0.5365854 0.1818182
9       1      2          NA     2 0.8658537 0.0000000 0.1341463 0.1818182
10      1      8          NA     2 0.7317073 0.6363636 0.2682927 0.3636364
11      1      3          NA     2 0.9195122 0.1818182 0.0804878 0.4545455
12      1      7          NA     2 0.7317073 0.1818182 0.1878049 0.4545455
13      1      9          NA     2 0.4634146 0.5909091 0.2682927 0.4090909
14      5     36          NA     1 0.4634146 0.1818182 0.5365854 0.8181818
15      1      9          NA     2 0.4634146 0.1818182 0.2682927 0.4090909
     color
1  #0C0C7A
2  #332288
3  #180C7A
4  #250C7A
5  #310C7A
6  #7A0C56
7  #7A0C43
8  #882255
9  #7A0C31
10 #006B12
11 #006B1B
12 #006B24
13 #006B2D
14 #117733
15 #006B36

$type
[1] "index"

$vSize
[1] "count"

$vColor
[1] NA

$stdErr
[1] "count"

$algorithm
```

```
[1] "pivotSize"

$vpCoorX
[1] 0.03579098 0.96420902

$vpCoorY
[1] 0.02812148 0.87187852

$aspRatio
[1] 1.729103

$range
[1] NA

$mapping
[1] NA NA NA

$draw
[1] TRUE
```

Oof, that's unfortunate. The labels for the infrastructure category are really small. Try as I might, I don't see a good way of allowing them to extend outside the chart. So instead I will just produce a chart where the Infrastructure labels are absent, and I'll add them in later using PowerPoint.

It looks like the lowerbound.cex.labels argument lets me control the plot's tolerance for ill-fitting labels. That's helpful. Also, I can hide the "category" labels by setting font size to 0. The way this plot looks is HIGHLY dependent on the size of the viewport, which in this case is controlled by the parameters of png().

```r
png(
  filename = file.path(Sys.getenv("FIGURE_PATH"), "treemap.png"),
  width = 1800,
  height = 1200,
  res = 300
)

treemap(
  results,
  index = c("category", "theme"),
  vSize = "count",
  type = "index",
  palette = c(colors[1], colors[8], colors[4]), # from utils.R
  title = "Comments from Free Response Box",

  # Label styling
  lowerbound.cex.labels = 0.5, # multiplier between 0 and 1 that sets the lowerbound for the
  fontsize.labels = c(0, 10), # first number is for categories, second is for themes
  fontcolor.labels = c("white", "white"), # white for themes and categories
  align.labels = list(
    c("center", "top"),
    c("center", "center")
  ),
  inflate.labels = FALSE, # does nothing?
  force.print.labels = FALSE, # does nothing?

  # aesthetics
  border.col = "white", # white borders for readability
  fontsize.title = 14,
  bg.labels = 0 # transparent label backgrounds
)

dev.off()
```

```
pdf
  2
```