Figure 2

Overview

This script is similar to demographics_plots.qmd, but with less data wrangling, so I can focus on plotting the donut plots together into a nice final figure. The data ingested by this script was produced by demographics_plots.qmd.

Import packages and utilities

```
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")))
```

Define functions

```
read_donut_data_from_file <- function(filename) {
  df <- read.csv(
    file.path(
      DATA_PATH,
      "data_for_plots",
      filename
    ),
    stringsAsFactors = FALSE,
    sep = "\t"
```

```
  )
  df$categories <- as.factor(df$categories)
  df <- reorder_factor_by_column(
    df,
    categories,
    n_responses,
    descending = TRUE
  )
}
```

This function is for the "simple" donut charts, i.e. the ones that don't have an exploding slice.

```
donut_chart <- function(
  df,
  cpalette = COLORS, # default: colors from utils.R
  legendpos = "right",
  plot_right_margin = 2.5,
  plot_left_margin = 2.5,
  label_dist = 2.3 # controls how close the labels are to the ring
) {
  ggplot(
    df,
    aes(
      ymax = ymax,
      ymin = ymin,
      xmax = 2,
      xmin = 1.5,
      fill = categories
    )
  ) +
    geom_rect() +
    # Add labels
    geom_text(
      x = label_dist,
      aes(y = labelPosition, label = label),
      size = 2
    ) +
    scale_fill_manual(values = cpalette) +
    theme_void() +
    coord_polar(theta = "y", clip = "off") +
    xlim(c(1, label_dist + 0.025)) +
```

```
    theme(
      legend.text = element_text(size = 5, margin = margin(l = 1), hjust = 0),
      legend.title = element_blank(),
      legend.position = legendpos,
      #legend.key.spacing.y = unit(0.25, "lines"),
      legend.key.size = unit(rel(0.3), "lines"),
      plot.title = element_text(
        hjust = 0.5,
        face = "bold",
        size = 7,
        margin = margin(t = 5, b = 7.5)
      ),
      plot.margin = margin(t = 10, r = plot_right_margin, b = 0, l = plot_left_margin),
      plot.background = element_rect(fill = "white", color = "white")
    )
}
```

Load data

```
campus_data <- read_donut_data_from_file("fig_campus_donut.tsv")
job_data <- read_donut_data_from_file("fig_job_donut.tsv")
field_data <- read_donut_data_from_file("fig_fields_donut.tsv")
staff_data <- read_donut_data_from_file("fig_staff_donut.tsv")
status_data <- read_donut_data_from_file("fig_status_donut.tsv")
```

Plot experience status

```
p_status <- donut_chart(status_data,
  label_dist = 2.25,
  cpalette = c(
    "#ddaa33",
    "#004488"
  ),
  legendpos = "left",
  plot_left_margin = 0
 ) +
  labs(title = "Respondents' Experience Levels")
```

Plot campus

```
# Add asterisks without changing factor level order
targets <- c(
  "UC Santa Barbara",
  "UC Los Angeles",
  "UC Davis",
  "UC Santa Cruz",
  "UC Berkeley",
  "UC San Diego"
)
lvls <- levels(campus_data$categories)
lvls[lvls %in% targets] <- paste0(lvls[lvls %in% targets], "*")
levels(campus_data$categories) <- lvls

# Plot
p_campus <- donut_chart(
  campus_data,
  # Paul Tol's "rainbow" color scheme
  cpalette = c(
    "#882d71",
    "#1964b0",
    "#518ac6",
    "#7bb0df",
    "#4db264",
    "#cae1ac",
    "#f7f057",
    "#f4a637",
    "#db060b",
    "#71190d"
  ),
  plot_right_margin = 5 #40
) +
  labs(title = "Campuses of Respondents")
```

Plot fields of study

```
p_field <- donut_chart(
  field_data,
```

```
  # Paul Tol's "bright" scheme
  cpalette = c(
    '#1a7937', #dark green
    '#acd49f', #light green
    '#e5e5e5', #gray
    '#c3a4d0', #light purple
    '#752a82' #dark purple
  ),
  legendpos = "left",
  plot_right_margin = 10 #40
) +
  labs(title = "Academic Respondents' Fields of Study")

  # # Paul Tol's "bright" scheme
  # cpalette = c(
  #   '#4477aa', #medium blue
  #   '#66ccee', #light blue
  #   '#228833', #green
  #   '#aa3377', #purple
  #   '#c6c6c6' #gray
  # ),
```

Plot staff work areas

```
p_staff <- donut_chart(staff_data,
  # Paul Tol's YlOrBr palette
  cpalette = c(
    '#662606', #brown
    '#cc4c02', #dark orange
    '#fb9b2a', #light orange
    '#fde391', #pale yellow
    '#ffffe5' #almost-white-yellow
  ),
  plot_left_margin = 10 #40
  ) +
  labs(title = "Staff Respondents' Work Areas")

  # Paul Tol's "medium contrast" scheme
  # cpalette = c(
  #   '#994455', #magenta
```

```
  #   '#ee99aa', #pink
  #   '#997700', #dark gold
  #   '#eecc66', #yellow
  #   '#6699cc' #muted blue
  # )
```

Plot job categories

```
# Specify radius for exploded segment
job_data$explode <- c(0.30, rep(0, times = nrow(job_data)-1))

# Compute segment starts and ends
job_data <- job_data %>%
  mutate(
    proportion = n_responses / sum(n_responses),
    # Compute the cumulative percentages (top of each segment)
    ymax = cumsum(proportion),
    # lag finds the previous value in a vector
    ymin = lag(ymax, default = 0),
  ) %>%
    # we don't need this column anymore
    select(-proportion)
```

After many attempts to pre-compute label positions with trigonometry, I eventually gave up and just painstakingly chose label positions through visual inspection and trial and error.

```
p_job <- ggplot(job_data) +
  ggforce::geom_arc_bar(aes(
    x0 = 0,
    y0 = 0,
    r0 = 0.5, # r0=0 makes it a pie, 0.5 makes it a donut
    r = 1,
    start = ymin * 2 * pi, # Start angle in radians
    end = ymax * 2 * pi, # End angle in radians
    fill = categories,
    explode = explode # Map the explode column here
  ), color = NA   # remove borders
  ) +
  labs(title = "Respondents' Job Categories") +
```

```
  coord_fixed() + # Ensures the plot is a circle
  theme_void() +
  coord_fixed(clip = "off") + # Allow labels to overflow beyond plot area
  scale_fill_manual(values = COLORS) +
  geom_text(
    x = c(1.6,  0.2, -1.25, -1.3, -0.6, -0.2),
    y = c(0.5, -1.3, -0.6,  0.5,  1.15,  1.25),
    size = 2.2,
    aes(label=n_responses)
  ) +
  theme(
      legend.text = element_text(size = 5, margin = margin(l = 1)),
      legend.title = element_blank(),
      legend.position = "bottom",
      legend.key.spacing.y = unit(0, "cm"),
      legend.key.size = unit(rel(0.3), "lines"),
      legend.margin = margin(t = 10), # legend top margin
      plot.title = element_text(
        hjust = 0.5,
        face = "bold",
        size = 7,
        margin = margin(t = 5, b = 10)
      ),
      plot.margin = margin(t = 10, r = 40, b = 5, l = 40),
      plot.background = element_rect(fill = "white", color = "white")
  ) +
    guides(fill = guide_legend(byrow = TRUE))
```

```
Warning in ggforce::geom_arc_bar(aes(x0 = 0, y0 = 0, r0 = 0.5, r = 1, start =
ymin * : Ignoring unknown aesthetics: explode
```

```
Coordinate system already present. Adding new coordinate system, which will
replace the existing one.
```

Since patchwork does not have functionality to draw lines across panels, I will draw them by hand in an image editor.

```
#patchwork package
top_row <- plot_spacer() + p_status + plot_spacer() + p_campus + plot_spacer() + plot_layout
bottom_row <- p_field + p_job + p_staff
p_combined <- patchwork::wrap_plots(
```

```
  top_row / bottom_row
)

# FIGURE_PATH is in .Renviron file
ggsave(
  filename = file.path(FIGURE_PATH, "fig2_unfinished.tif"),
  plot = p_combined + plot_annotation(tag_levels = "A") & theme(plot.tag = element_text(size
  device = "tiff",
  width = 7.5, height = 3.75, units = "in",
  dpi = 300,
  compression = "none",
  bg = "white"
)
```