Adarsh T10 (EDA)

Adarsh Nagar

2025-10-09

Introduction:

Today we will be exploring the relationship between the language comfortability/preference from the intake form and individual domain interest for 2025 Data Science Capstone projects.

Import the data

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   4.0.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# retrieve class survey data
url <- 'https://raw.githubusercontent.com/pstat197/pstat197a/main/materials/labs/lab2-tidyverse/data/'

background <- paste(url, 'background-clean.csv', sep = '') %>%
  read_csv()
```

```
## Rows: 51 Columns: 29
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): prog.prof, math.prof, stat.prof, updv.num, dom
## dbl (23): response_id, prog.comf, math.comf, stat.comf, PSTAT100, PSTAT115, ...
## lgl  (1): rsrch
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
interest <- paste(url, 'interest-clean.csv', sep = '') %>%
  read_csv()
```

```
## Rows: 52 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): type, lang, dom, area
## dbl (1): response_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
interest$lang[interest$response_id == 29] <- "Python"
interest$dom[interest$response_id == 29] <- "Technology;Software development;Product engineering"

metadata <- paste(url, 'survey-metadata.csv', sep = '') %>%
  read_csv()
```

```
## Rows: 34 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): variable.name, survey.section, variable.description, variable.type,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

EDA & Satisfying Expectations

```
library(ggplot2)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(dplyr)
library(forcats)
library(tidyr)

# Extract exact columns
consolidated_csv <- interest[3:4]
consolidated_df <- data.frame(consolidated_csv)

#Total Counts for each domain
expanded_df <- consolidated_csv |>
  separate_rows(dom,sep =";") |>
```

```
  mutate(dom = trimws(dom))

# summarize to lang × dom counts
plot_data <- expanded_df |>
  count(lang, dom, name = "n") |>
  mutate(lang = fct_reorder(lang, n, .fun = sum))

# Stacked Bar Plot (Frequency + Language)
stacked <- ggplot(plot_data, aes(y = n, x = lang, fill = dom)) +
  geom_col(position = position_dodge(width = 0.9)) +
  labs(title = "Domain Interest by Languge",
       x = "Language Preference", y = "Count", fill = "Domain") +
theme(legend.position = "right") + geom_hline(yintercept =5,color="black")

legend <- get_legend(
  ggplot(plot_data, aes(y = lang, x = n, fill = dom)) +
    geom_col(position = position_dodge()) +
    theme_minimal() +
    theme(legend.position = "right")
)
```

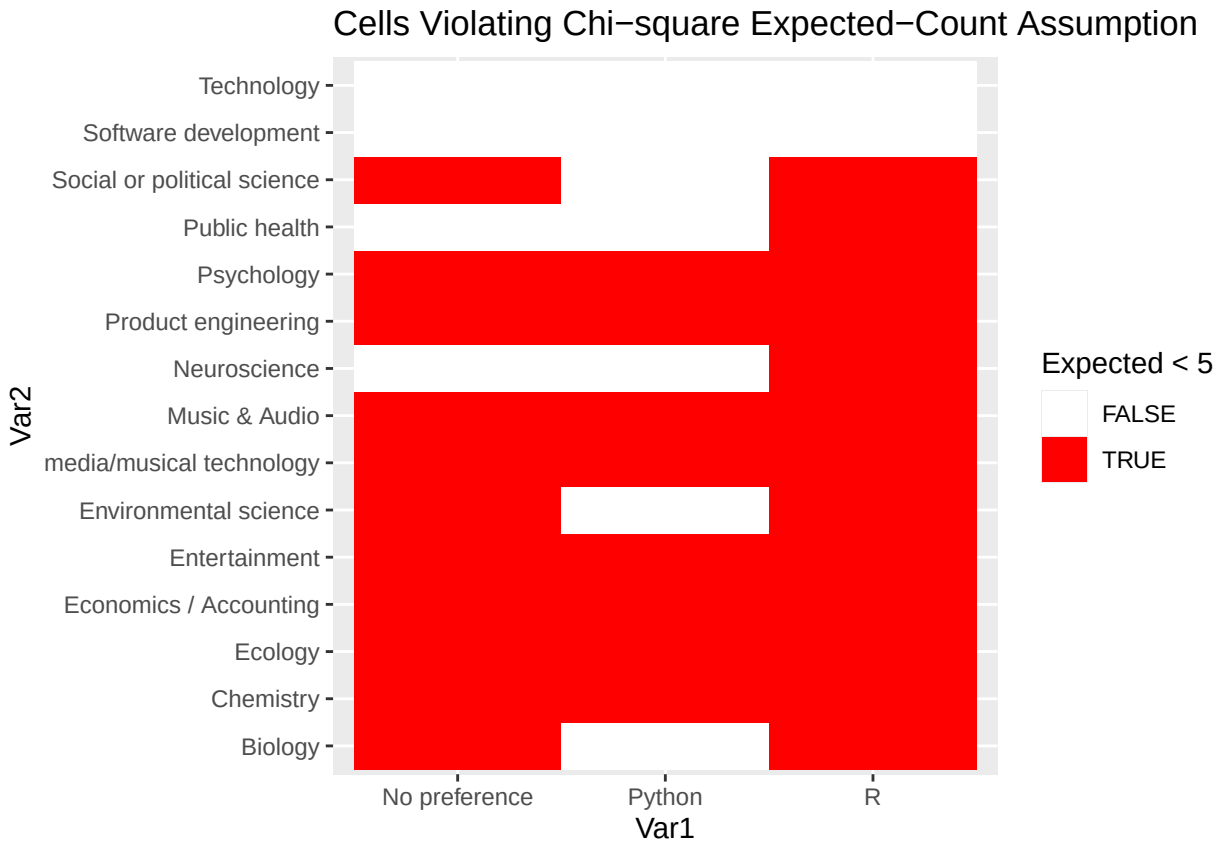
```
## Warning in get_plot_component(plot, "guide-box"): Multiple components found;
## returning the first one. To return all, use `return_all = TRUE`.
```

```
# List of Satisfying
tbl <- table(expanded_df$lang, expanded_df$dom)
chi <- suppressWarnings(chisq.test(tbl))
expected <- as.data.frame(as.table(chi$expected))
expected |> filter(Freq < 5)
```

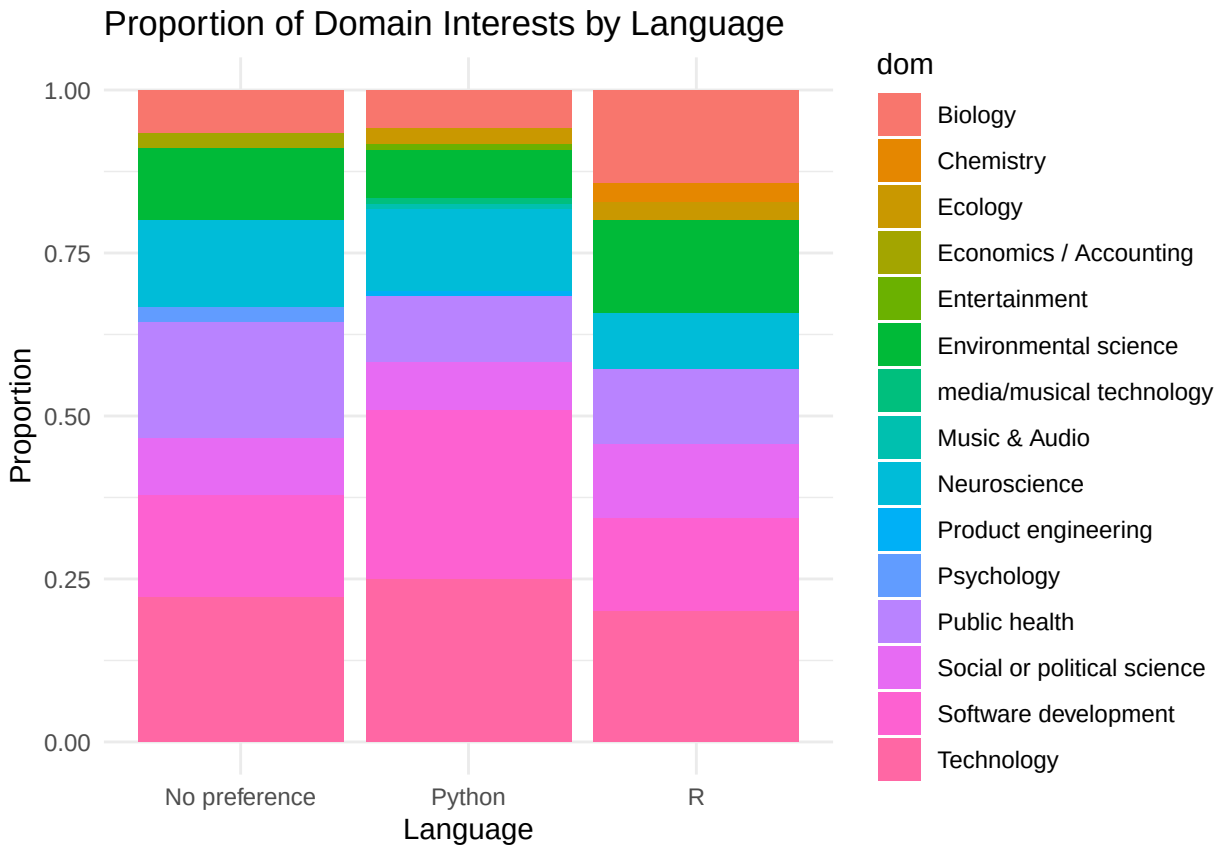
```
##             Var1                       Var2  Freq
## 1  No preference                    Biology 3.375
## 2              R                    Biology 2.625
## 3  No preference                  Chemistry 0.225
## 4         Python                  Chemistry 0.600
## 5              R                  Chemistry 0.175
## 6  No preference                    Ecology 0.900
## 7         Python                    Ecology 2.400
## 8              R                    Ecology 0.700
## 9  No preference      Economics / Accounting 0.225
## 10        Python      Economics / Accounting 0.600
## 11             R      Economics / Accounting 0.175
## 12 No preference               Entertainment 0.225
## 13        Python               Entertainment 0.600
## 14             R               Entertainment 0.175
## 15 No preference       Environmental science 4.275
## 16             R       Environmental science 3.325
## 17 No preference    media/musical technology 0.225
## 18        Python    media/musical technology 0.600
## 19             R    media/musical technology 0.175
## 20 No preference               Music & Audio 0.225
## 21        Python               Music & Audio 0.600
## 22             R               Music & Audio 0.175
```

```
## 23             R                Neuroscience 4.200
## 24 No preference         Product engineering 0.225
## 25        Python         Product engineering 0.600
## 26             R         Product engineering 0.175
## 27 No preference                  Psychology 0.225
## 28        Python                  Psychology 0.600
## 29             R                  Psychology 0.175
## 30             R               Public health 4.200
## 31 No preference Social or political science 3.825
## 32             R Social or political science 2.975
```

```
ggplot(expected, aes(x = Var1, y = Var2, fill = Freq < 5)) +
  geom_tile() +
  scale_fill_manual(values = c("FALSE" = "white", "TRUE" = "red")) +
  labs(title = "Cells Violating Chi-square Expected-Count Assumption",
       fill = "Expected < 5")
```




```
ggplot(expanded_df, aes(x = lang, fill = dom)) +
  geom_bar(position = "fill") +
  labs(title = "Proportion of Domain Interests by Language",
       x = "Language", y = "Proportion") +
  theme_minimal()
```



Statistical Testing

```
# Monte Carlo Simulation Chi-Square Test
chi_sim <- chisq.test(tbl, simulate.p.value = TRUE, B = 10000)
print(chi_sim)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 10000
##  replicates)
##
## data:  tbl
## X-squared = 25.117, df = NA, p-value = 0.6562
```

#Result

Discarded:

```
# Get unique languages and domains
langs   <- unique(expanded_df$lang)
domains <- unique(expanded_df$dom)
```

```r
domain_test <- function(d) {
  # Build the 2×K table for this domain
  tab <- expanded_df %>%
    mutate(present = dom == d) %>%
    count(lang, present) %>%
    complete(lang = langs, present = c(FALSE, TRUE), fill = list(n = 0)) %>%
    pivot_wider(names_from = present, values_from = n) %>%
    as.data.frame()

  mat <- as.matrix(tab[, c("FALSE", "TRUE")])
  rownames(mat) <- tab$lang

  # Try Chi-square first
  chi_try <- suppressWarnings(chisq.test(mat))
  exp_ok  <- all(chi_try$expected >= 5)

  if (exp_ok) {
    tibble(domain = d, test = "Chi-square", p = chi_try$p.value,
           min_expected = min(chi_try$expected))
  } else {
    fish <- fisher.test(mat)
    tibble(domain = d, test = "Fisher", p = fish$p.value,
           min_expected = min(chi_try$expected))
  }
}

pairwise_results <- map_dfr(domains,domain_test) |>
  mutate(p_adj = p.adjust(p, method = "BH")) |>
  arrange(p_adj)

knitr::kable(pairwise_results)
```

domain	test	p	min_expected	p_adj
Environmental science	Fisher	0.3875315	3.325	0.8571429
Biology	Fisher	0.2647129	2.625	0.8571429
Public health	Fisher	0.3611974	4.200	0.8571429
Software development	Chi-square	0.1866923	7.525	0.8571429
Chemistry	Fisher	0.1750000	0.175	0.8571429
Psychology	Fisher	0.4000000	0.175	0.8571429
Economics / Accounting	Fisher	0.4000000	0.175	0.8571429
Ecology	Fisher	0.6307750	0.700	1.0000000
Neuroscience	Fisher	0.8670987	4.200	1.0000000
Technology	Chi-square	0.8067232	8.225	1.0000000
Social or political science	Fisher	0.6356176	2.975	1.0000000
media/musical technology	Fisher	1.0000000	0.175	1.0000000
Product engineering	Fisher	1.0000000	0.175	1.0000000
Entertainment	Fisher	1.0000000	0.175	1.0000000
Music & Audio	Fisher	1.0000000	0.175	1.0000000