Week 2 report Sam portion

Sam Caruthers

10/18/25

Looking at Language, preferences as it relates to domain and research interests.

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(stringr)
```

```
# ---- 1) Load + normalize column names ----
#df <- read_csv("/Users/scaruthers/group-sandbox-table-4-1/module-0-class-survey-data-table-4/data/merg
df <- read_csv("/Users/scaruthers/Downloads/merged-clean.csv")
```

```
## Rows: 49 Columns: 34
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): prog.prof, math.prof, stat.prof, updv.num, dom.x, do_you_have_any_...
## 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.
```

```
# Create canonical columns if the source uses variants
# (these if-blocks are safe no-ops if the target already exists)
if (!"Language" %in% names(df) && "lang" %in% names(df))   df$Language <- df$lang
if (!"Majors"   %in% names(df) && "major" %in% names(df)) df$Majors   <- df$major
if (!"Group"    %in% names(df) && "group" %in% names(df)) df$Group    <- df$group
# keep dom/area as-is; if they were capitalized earlier, normalize them:
if (!"dom"  %in% names(df) && "Dom"  %in% names(df)) df$dom  <- df$Dom
if (!"area" %in% names(df) && "Area" %in% names(df)) df$area <- df$Area
```

```r
# dfm is the working copy used in later chunks
dfm <- df
```

```r
# 2. Expand into long form:
df_long <- df %>%
  separate_rows(dom.y,  sep = ";") %>%
  separate_rows(area, sep = ";") %>%
  mutate(across(c(dom.y, area), str_trim))

# ... now your tbl_dom_area, chi-square, etc. will work ...
```

```r
# 2. Filter pure Python vs. pure R
df2 <- df %>%
  filter(lang %in% c("Python", "R"))
```

```r
tbl <- table(df2$type, df2$Language)
chi <- chisq.test(tbl)
```

```
## Warning in chisq.test(tbl): Chi-squared approximation may be incorrect
```

```r
print(chi)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 11.718, df = 2, p-value = 0.002854
```

6. Optional: Mosaic plot with vcd

install.packages("vcd") # if not already installed

library(vcd) mosaic(tbl, shade = TRUE, legend = TRUE, main = "Mosaic Plot: Majors vs Language") # Optional: Cramer's V for effect size # install.packages("DescTools") library(DescTools) print(CramerV(tbl))

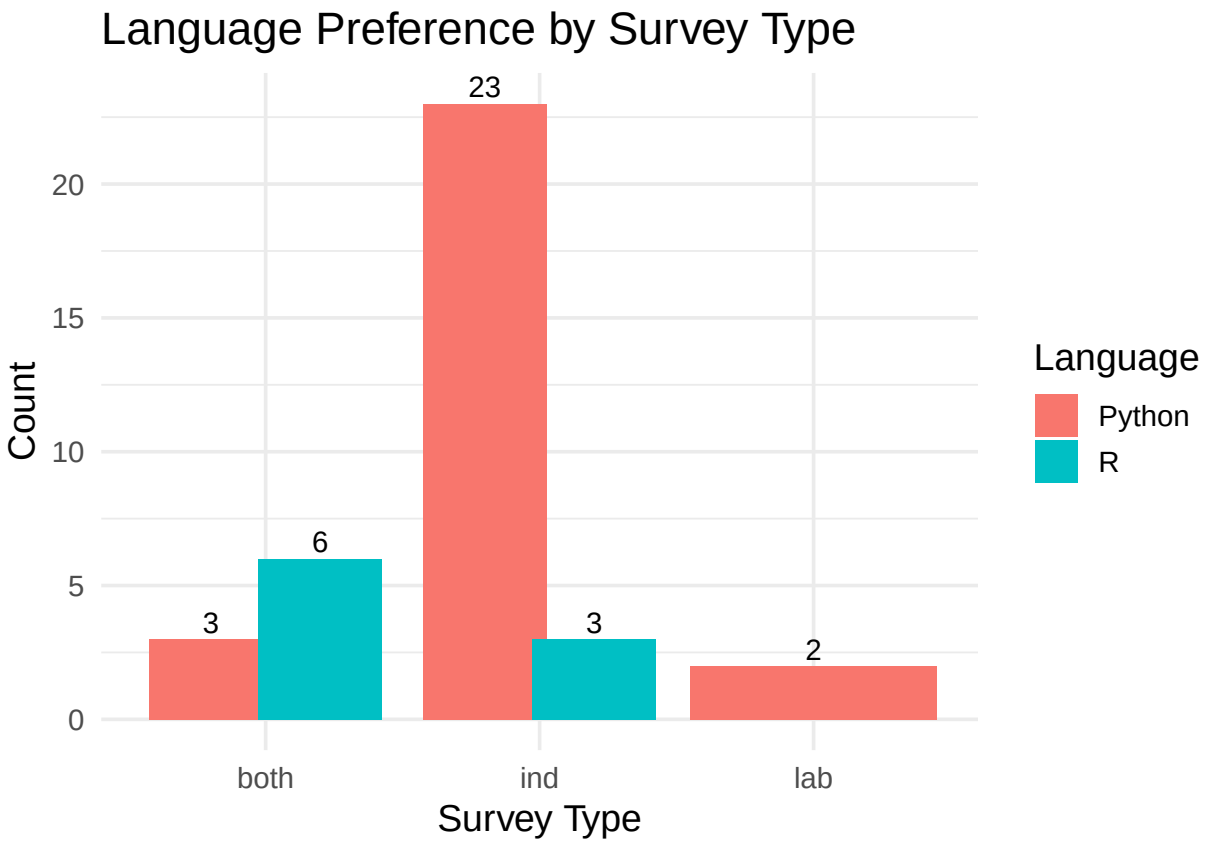
```r
# 3b. Contingency table: type vs language
tbl_type_lang <- table(df2$type, df2$Language)

# 5d. Bar-plot: language preference by type with counts
library(ggplot2)

plt_type_lang <- ggplot(df2, aes(x = type, fill = Language)) +
  geom_bar(position = position_dodge(width = 0.8)) +
  geom_text(
    stat = "count",
    aes(label = after_stat(count)),
    position = position_dodge(width = 0.8),
    vjust = -0.3,
    size = 4
  ) +
  labs(
    title = "Language Preference by Survey Type",
    x = "Survey Type",
    y = "Count"
  ) +
```

```
  theme_minimal(base_size = 14)

print(plt_type_lang)
```



PLot 1

This plot shows the relationship between students language preference between Python and R as it relates to what type of capstone project they wanted to do, with the options being industry, lab, or both/no preference. The vast majority of responses are from students who want to work on an industry project in Python. This makes intuitive sense because I would say on average statistics majors are focused on landing a job post graduation, and a project in the data science industry, in the industry standard language of python, is the most transferrable project. The fact that those who picked both preferred is curious, because we normally associate R with lab work, or at least definitely lab and coursework at UCSB.

```
library(dplyr)
library(tidyr)
library(stringr)
library(ggplot2)

# --- assumes `df` already loaded ---

# Normalize names + expand semicolon lists
df_long <- df %>%
  rename_with(tolower) %>%
  separate_rows(dom.y,  sep = ";") %>%
  separate_rows(area, sep = ";") %>%
  mutate(
```

```
    dom  = str_squish(str_trim(dom.y)),
    area = str_squish(str_trim(area))
  )

# Exact-match buckets for area_group (your survey options)
tech_exact <- c(
  "Model deployment and software or web integrations",
  "Data acquisition and engineering",
  "Backend",
  "Analysis or classification of images",
  "Deep learning and neural networks",
  "Databases"
)
stats_exact <- c(
  "Statistical models and inference, generally",
  "Predictive modeling, generally",
  "Spatial statistics or time series analysis",
  "Natural language processing and analysis of text",
  "Algorithms",
  "Data visualization and interactive dashboards"
)

# Group areas and domains (no Statistics category)
df_grouped <- df_long %>%
  mutate(
    area_group = case_when(
      area %in% tech_exact  ~ "Tech/Software",
      area %in% stats_exact ~ "Stats/Research",
      TRUE                  ~ "Other"
    ),
    dom_group = case_when(
      str_detect(dom, regex("Social|political|Psychology", ignore_case = TRUE)) ~ "Social/Behavioral Sc
      str_detect(dom, regex("Environmental|Ecology",      ignore_case = TRUE)) ~ "Environmental/Ecology
      str_detect(dom, regex("Biology|Neuroscience",       ignore_case = TRUE)) ~ "Life Sciences",
      str_detect(dom, regex("Public health",              ignore_case = TRUE)) ~ "Health",
      str_detect(dom, regex("Technology|Software development|media/musical technology|Engineering",
                            ignore_case = TRUE)) ~ "Tech/Software/Engineering",
      str_detect(dom, regex("Chemistry|Physics|Math|Mathematics",
                            ignore_case = TRUE)) ~ "Physical/Math Sciences",
      TRUE ~ "Other"
    )
  )

# Use the same rows for both variables (drop NAs)
df_tab <- df_grouped %>% filter(!is.na(dom_group), !is.na(area_group))

# Chi-square: dom_group vs area_group
tbl_domgrp_areagrp <- with(df_tab, table(dom_group, area_group))
print(tbl_domgrp_areagrp)
```

```
##                             area_group
## dom_group                    Stats/Research Tech/Software
##   Environmental/Ecology                  86            75
##   Health                                 95            83
```

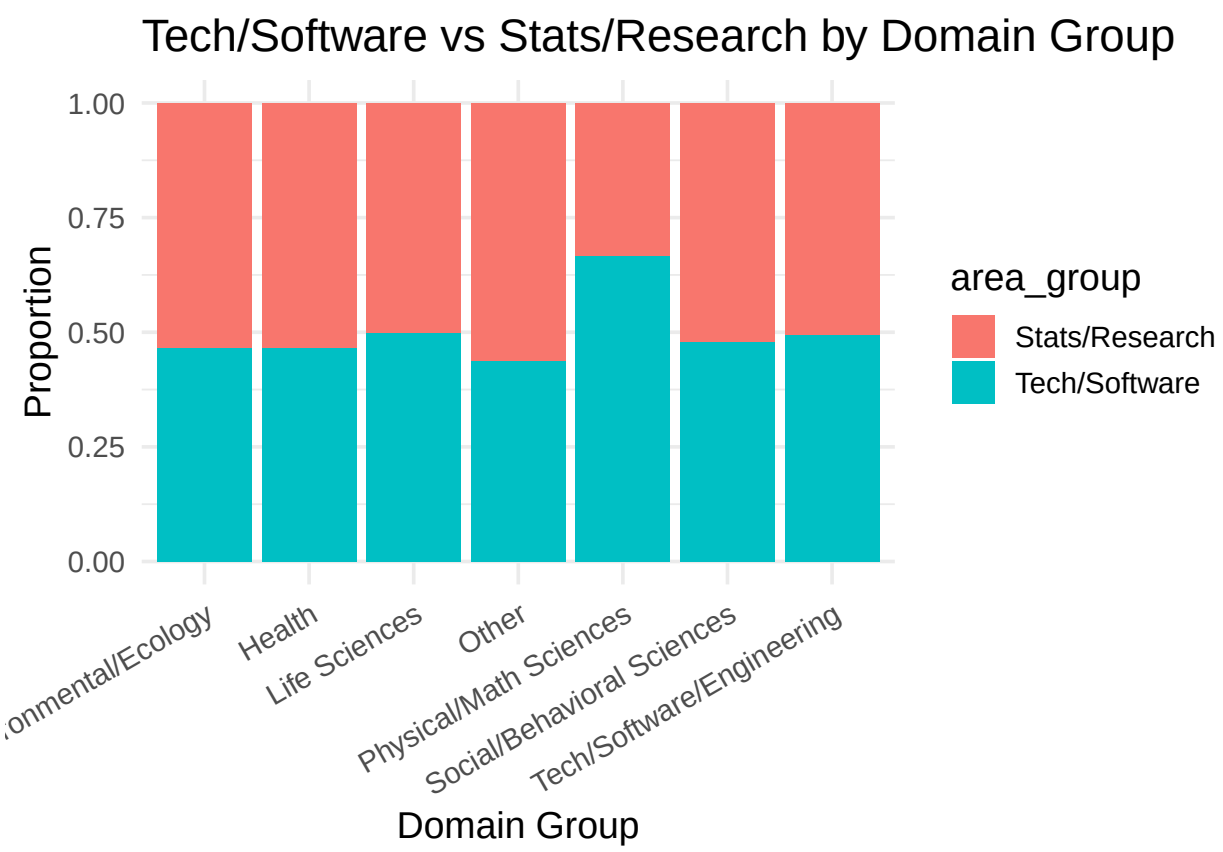
```
##   Life Sciences                    138          137
##   Other                              9            7
##   Physical/Math Sciences             1            2
##   Social/Behavioral Sciences        73           67
##   Tech/Software/Engineering        327          319
```

```r
print(chisq.test(tbl_domgrp_areagrp))
```

```
## Warning in chisq.test(tbl_domgrp_areagrp): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl_domgrp_areagrp
## X-squared = 1.4329, df = 6, p-value = 0.9638
```

```r
# Proportion plot: Tech vs Stats by domain group
ggplot(
  df_tab %>% filter(area_group %in% c("Tech/Software","Stats/Research")),
  aes(x = dom_group, fill = area_group)
) +
  geom_bar(position = "fill") +
  labs(
    title = "Tech/Software vs Stats/Research by Domain Group",
    x = "Domain Group", y = "Proportion"
  ) +
  theme_minimal(base_size = 14) +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



Plot 2

This plot shows the domain of interest for students relative to the research area they are interested. In the survey the question was phrased as "what area of data science are of interest to you", however I grouped them into stats and tech because a handful of topics were more CS/Software oriented. The classes are as follows, Stats/Research Statistical models and inference, generally", "Predictive modeling, generally", "Spatial statistics or time series analysis", "Natural language processing and analysis of text", "Algorithms", "Data visualization and interactive dashboards vs Tech:"Model deployment and software or web integrations", "Data acquisition and engineering","Backend" "Deep learning and neural networks", "Analysis or classification of images. I then created another graph, below, which considered deep learning and analysis/classification of images as a Statistics category, because there is an active debate between what is Computer Science vs Statistics vs Data science (which usually tends to be the intersection of a Stats and CS diagram).

```
tech_exact <- c(
  "Model deployment and software or web integrations",
  "Data acquisition and engineering",
  "Backend",
  "Databases"
)
stats_exact <- c(
  "Statistical models and inference, generally",
  "Predictive modeling, generally",
  "Spatial statistics or time series analysis",
  "Natural language processing and analysis of text",
  "Algorithms",
  "Analysis or classification of images",
  "Deep learning and neural networks",
  "Data visualization and interactive dashboards"
)

# Group areas and domains (no Statistics category)
df_grouped <- df_long %>%
  mutate(
    area_group = case_when(
      area %in% tech_exact  ~ "Tech/Software",
      area %in% stats_exact ~ "Stats/Research",
      TRUE                  ~ "Other"
    ),
    dom_group = case_when(
      str_detect(dom, regex("Social|political|Psychology", ignore_case = TRUE)) ~ "Social/Behavioral Sc
      str_detect(dom, regex("Environmental|Ecology",      ignore_case = TRUE)) ~ "Environmental/Ecology
      str_detect(dom, regex("Biology|Neuroscience",       ignore_case = TRUE)) ~ "Life Sciences",
      str_detect(dom, regex("Public health",              ignore_case = TRUE)) ~ "Health",
      str_detect(dom, regex("Technology|Software development|media/musical technology|Engineering",
                            ignore_case = TRUE)) ~ "Tech/Software/Engineering",
      str_detect(dom, regex("Chemistry|Physics|Math|Mathematics",
                            ignore_case = TRUE)) ~ "Physical/Math Sciences",
      TRUE ~ "Other"
    )
  )

# Use the same rows for both variables (drop NAs)
df_tab <- df_grouped %>% filter(!is.na(dom_group), !is.na(area_group))
```

```r
# Chi-square: dom_group vs area_group
tbl_domgrp_areagrp <- with(df_tab, table(dom_group, area_group))
print(tbl_domgrp_areagrp)
```

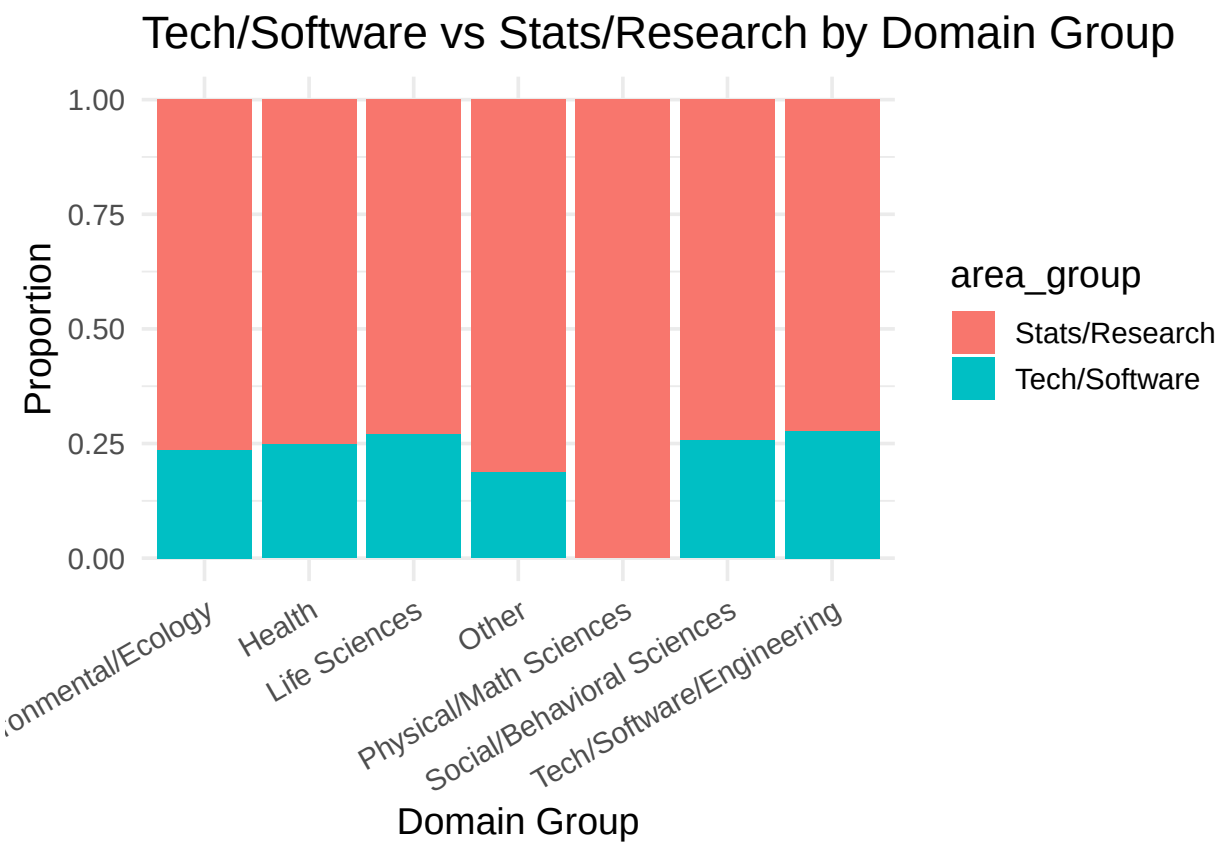
```
##                             area_group
## dom_group                    Stats/Research Tech/Software
##   Environmental/Ecology                 123            38
##   Health                                134            44
##   Life Sciences                         201            74
##   Other                                  13             3
##   Physical/Math Sciences                  3             0
##   Social/Behavioral Sciences            104            36
##   Tech/Software/Engineering             467           179
```

```r
print(chisq.test(tbl_domgrp_areagrp))
```

```
## Warning in chisq.test(tbl_domgrp_areagrp): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl_domgrp_areagrp
## X-squared = 3.1074, df = 6, p-value = 0.7952
```

```r
# Proportion plot: Tech vs Stats by domain group
ggplot(
  df_tab %>% filter(area_group %in% c("Tech/Software","Stats/Research")),
  aes(x = dom_group, fill = area_group)
) +
  geom_bar(position = "fill") +
  labs(
    title = "Tech/Software vs Stats/Research by Domain Group",
    x = "Domain Group", y = "Proportion"
  ) +
  theme_minimal(base_size = 14) +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



Answer with other plot

What we see is one that clearly neural networks and image classification are very popular as they are able to change the proportion of research interest in different domains by around forty percent. This intuitvely makes sense, as neural networks in particularly represent the hot new technology that everyone is trying to understand, and a year long research class is the perfect place to enhance one's understanding in that field while also building an impressive project.

```
# ...existing code...
library(dplyr)
library(stringr)

# Normalize language categories
lang_counts <- dfm %>%
  mutate(
    lang_norm = case_when(
      str_detect(lang, regex("No preference", ignore_case = TRUE)) ~ "No preference",
      str_detect(lang, regex("\\bpython\\b", ignore_case = TRUE)) ~ "Python",
      str_detect(lang, regex("\\br\\b", ignore_case = TRUE)) ~ "R",
      TRUE ~ "Other"
    )
  ) %>%
  count(lang_norm, name = "n") %>%
  arrange(desc(n))

print(lang_counts)
```

```
## # A tibble: 3 x 2
##   lang_norm         n
##   <chr>         <int>
## 1 Python           29
## 2 No preference    11
## 3 R                 9
```

```
# Identify 3-digit course columns
cs_cols    <- grep("^CS[0-9]{3}$", names(dfm), value = TRUE, ignore.case = TRUE)
pstat_cols <- grep("^PSTAT[0-9]{3}$", names(dfm), value = TRUE, ignore.case = TRUE)

course_totals <- dfm %>%
  mutate(
    cs_count    = if (length(cs_cols)) rowSums(dplyr::across(all_of(cs_cols)), na.rm = TRUE) else 0,
    pstat_count = if (length(pstat_cols)) rowSums(dplyr::across(all_of(pstat_cols)), na.rm = TRUE) else
  ) %>%
  summarise(
    stats_gt4 = sum(pstat_count > 4, na.rm = TRUE),
    stats_eq4 = sum(pstat_count == 4, na.rm = TRUE),
    stats_le4 = sum(pstat_count < 4, na.rm = TRUE),
    cs_lt2    = sum(cs_count < 2, na.rm = TRUE),
    cs_eq2    = sum(cs_count == 2, na.rm = TRUE),
    cs_gt2    = sum(cs_count > 2, na.rm = TRUE)
  )
print(course_totals)
```

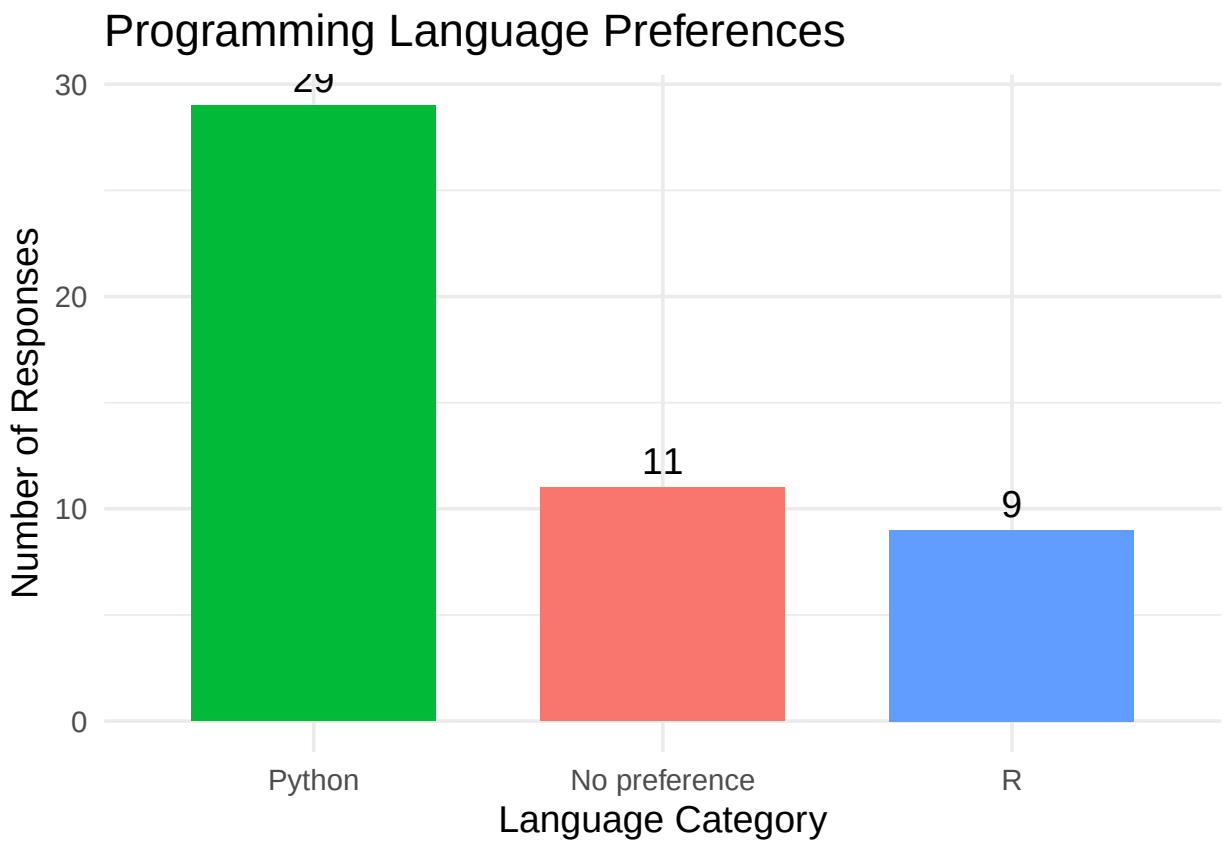
```
## # A tibble: 1 x 6
##   stats_gt4 stats_eq4 stats_le4 cs_lt2 cs_eq2 cs_gt2
##       <int>     <int>     <int>  <int>  <int>  <int>
## 1        21        17        11     46      3      0
```

```
# ...existing code...
```

```
# ...existing code...
library(dplyr)
library(stringr)
library(ggplot2)



# Plot language preference counts
ggplot(lang_counts, aes(x = reorder(lang_norm, -n), y = n, fill = lang_norm)) +
  geom_col(width = 0.7, show.legend = FALSE) +
  geom_text(aes(label = n), vjust = -0.5, size = 5) +
  labs(
    title = "Programming Language Preferences",
    x = "Language Category",
    y = "Number of Responses"
  ) +
  theme_minimal(base_size = 14)
```




```
# ...existing code...
library(dplyr)
library(stringr)
library(ggplot2)
library(forcats)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

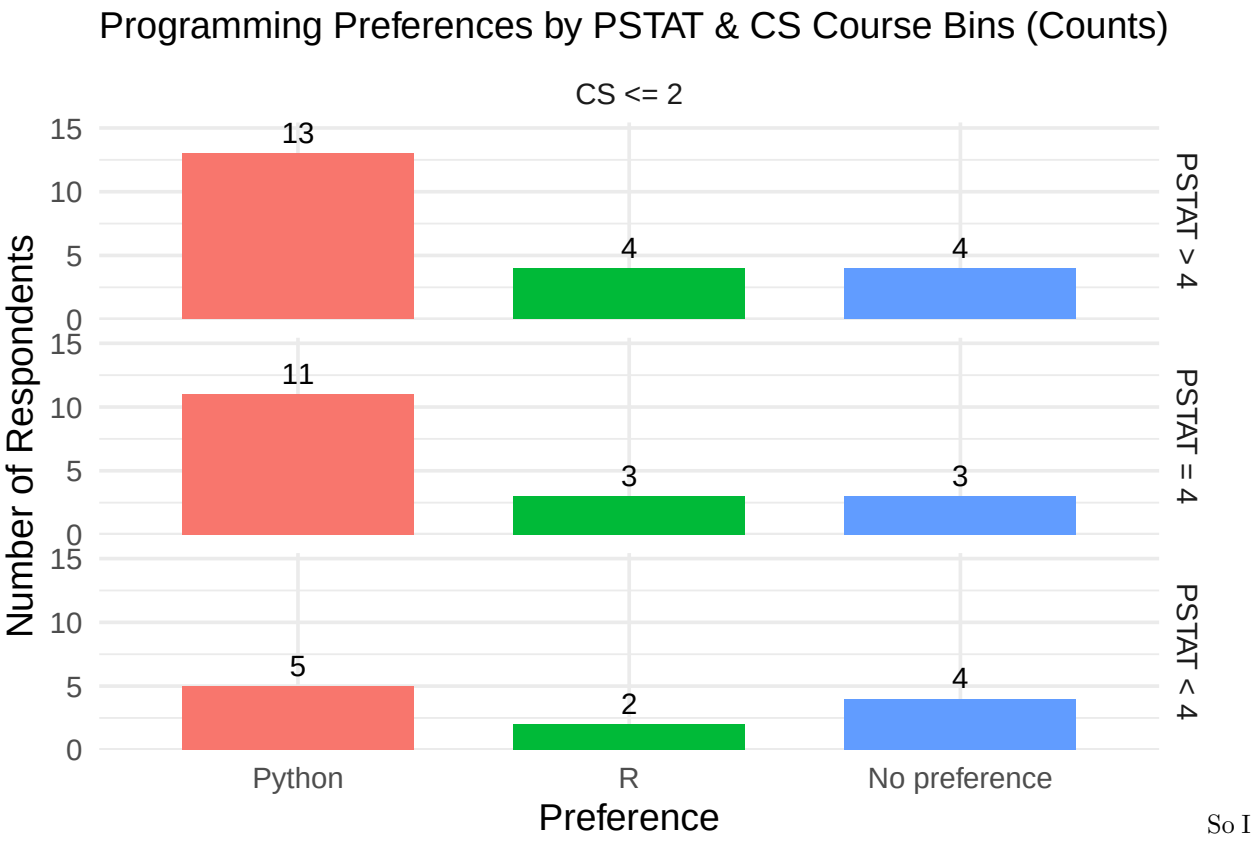
```
# Per-respondent CS/PSTAT counts + bins
dfm_binned <- dfm %>%
  mutate(
    lang_norm = case_when(
      str_detect(lang, regex("No preference", ignore_case = TRUE)) ~ "No preference",
      str_detect(lang, regex("\\bpython\\b", ignore_case = TRUE)) ~ "Python",
      str_detect(lang, regex("\\br\\b", ignore_case = TRUE)) ~ "R",
      TRUE ~ "Other"
    ),
    cs_count    = if (length(cs_cols))    rowSums(dplyr::across(all_of(cs_cols)), na.rm = TRUE) else 0,
    pstat_count = if (length(pstat_cols)) rowSums(dplyr::across(all_of(pstat_cols)), na.rm = TRUE) else
    pstat_bin = case_when(
      pstat_count > 4  ~ "PSTAT > 4",
      pstat_count == 4 ~ "PSTAT = 4",
```

```
      TRUE             ~ "PSTAT < 4"
    ),
    cs_bin = case_when(
      cs_count <= 2  ~ "CS <= 2",
      TRUE           ~ "CS > 2"
    ),
    # Set factor order for clean facets/axes
    lang_norm = fct_relevel(lang_norm, "Python", "R", "No preference", "Other"),
    pstat_bin = fct_relevel(factor(pstat_bin), "PSTAT > 4", "PSTAT = 4", "PSTAT < 4"),
    cs_bin    = fct_relevel(factor(cs_bin),    "CS <= 2", "CS = 2", "CS > 2")
  )
```

```
## Warning: There were 2 warnings in `mutate()`.
## The first warning was:
## i In argument: `lang_norm = fct_relevel(lang_norm, "Python", "R", "No
##   preference", "Other")`.
## Caused by warning:
## ! 1 unknown level in `f`: Other
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
# Summaries by bins
pref_by_bins <- dfm_binned %>%
  count(pstat_bin, cs_bin, lang_norm, name = "n") %>%
  group_by(pstat_bin, cs_bin) %>%
  mutate(prop = n / sum(n)) %>%
  ungroup()


# ===== Plot 2: Raw counts within each (PSTAT x CS) bin =====
ggplot(pref_by_bins, aes(x = lang_norm, y = n, fill = lang_norm)) +
  geom_col(show.legend = FALSE, width = 0.7) +
  geom_text(aes(label = n), vjust = -0.5, size = 4) +
  facet_grid(pstat_bin ~ cs_bin) +
  scale_y_continuous(
    expand = expansion(mult = c(0, 0.187))  # adds headroom above bars
  ) +
  labs(
    title = "Programming Preferences by PSTAT & CS Course Bins (Counts)",
    x = "Preference",
    y = "Number of Respondents"
  ) +
  theme_minimal(base_size = 14) +
theme(
  plot.title = element_text(size = 14)   # shrink title text
)
```



So I decided also to look at language preference by classes taken. It turns out most people in the class (all but 3) have taken 2 Computer Science classes, so I decided to break it up by PSTAT class. I found this interesting because for the most part, UCSB PSTAT Classes are taught in R. That intuition suggests that the more PSTAT classes you take the more you would prefer R. However, there are several possibilities that can contribute to this result. I think as we saw previously people mainly want to do a Python project in the industry, so it can also be true that the more PSTAT classes you take the more you are thinking about getting a project in the industry. It is also true that in general students may see Python as a future career skill.