Thresher Class Data

Hongtao Hu

Nov 1, 2025

```
### Your code here:
# List of files
files <- c(
  "humaF21.csv",
  "humaS22.csv",
  "humaF22.csv",
  "humaS23.csv",
  "humaF23.csv",
  "humaS24.csv",
  "humaF24.csv",
  "humaS25.csv",
  "humaF25.csv"
)

# Read and combine
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(ggplot2)

huma <- files |>
  lapply(read.csv) |>
  bind_rows()


names(huma)
```

```
##  [1] "TERM"              "SCHOOL"            "DEPARTMENT"
##  [4] "SUBJ"              "COURSE"            "SECTION"
##  [7] "CRN"               "TITLE"             "TYPE"
```

```
## [10] "CREDITS"            "INSTRUCTOR.NAME.S." "SECT.MAX"
## [13] "SECT.ENRL"          "UG.SECT.ENRL"       "UP.SECT.ENRL"
## [16] "GR.SECT.ENRL"       "VS.SECT.ENRL"       "VU.SECT.ENRL"
## [19] "VG.SECT.ENRL"       "XLST.GROUP"         "XLST.MAX"
## [22] "XLST.ENRL"          "TTL.ENRL"           "TTL.SEATS.AVAIL"
## [25] "XLST.WITH"          "XLST.CRN"           "XLST.LEAD.DEPT"
## [28] "HAS_SYLLABUS"
```

```
#  [1] "TERM"              "SCHOOL"            "DEPARTMENT"        "SUBJ"                "COURSE"
#  [8] "TITLE"             "TYPE"              "CREDITS"           "INSTRUCTOR.NAME.S." "SECT.MAX"
# [15] "UP.SECT.ENRL"      "GR.SECT.ENRL"      "VS.SECT.ENRL"      "VU.SECT.ENRL"        "VG.SECT.ENR
# [22] "XLST.ENRL"         "TTL.ENRL"          "TTL.SEATS.AVAIL"   "XLST.WITH"           "XLST.CRN"

unique(huma$TERM)
```

```
## [1] "Fall Semester 2021"   "Spring Semester 2022" "Fall Semester 2022"
## [4] "Spring Semester 2023" "Fall Semester 2023"   "Spring Semester 2024"
## [7] "Fall Semester 2024"   "Spring Semester 2025" "Fall Semester 2025"
```

```
# [1] "Fall Semester 2021"   "Spring Semester 2022" "Fall Semester 2022"   "Spring Semester 2023" "Fall
# [7] "Fall Semester 2024"   "Spring Semester 2025" "Fall Semester 2025"

huma <- huma %>%
  mutate(TERM = recode(TERM,
    "Fall Semester 2021" = "21_2",
    "Spring Semester 2022" = "22_1",
    "Fall Semester 2022" = "22_2",
    "Spring Semester 2023" = "23_1",
    "Fall Semester 2023" = "23_2",
    "Spring Semester 2024" = "24_1",
    "Fall Semester 2024" = "24_2",
    "Spring Semester 2025" = "25_1",
    "Fall Semester 2025" = "25_2"
  ))

# Chronological order of terms
term_levels <- c("21_2", "22_1", "22_2", "23_1", "23_2", "24_1", "24_2", "25_1", "25_2")

huma <- huma %>%
mutate(TERM_num = as.numeric(factor(TERM, levels = term_levels)))

# unique(huma$TERM_num)

huma$SECT.MAX <- as.numeric(as.character(huma$SECT.MAX))
```

```
## Warning: NAs introduced by coercion
```

```
#filter courses with NA, filter courses with huma$COURSE > 499

huma_clean <- huma %>%
  filter(
    !is.na(SECT.MAX),           # remove rows where SECT.MAX is NA
```

```
    !is.na(SECT.ENRL),          # remove rows where SECT.ENRL is NA (optional)
    COURSE <= 499               # only keep courses 499 or below
  )
#combine english + engl and creative writing, combine modern and classical lit & culture and modern cla
huma_clean <- huma_clean %>%
  mutate(DEPARTMENT = case_when(
    DEPARTMENT == "English" ~ "English and Creative Writing",
    DEPARTMENT == "Modrn & Classicl Lit & Culture" ~ "Modern Classic Lang, Lit, Cult",
    TRUE ~ DEPARTMENT  # keep all other departments unchanged
  ))


names(huma_clean)
```

```
##  [1] "TERM"               "SCHOOL"             "DEPARTMENT"
##  [4] "SUBJ"               "COURSE"             "SECTION"
##  [7] "CRN"                "TITLE"              "TYPE"
## [10] "CREDITS"            "INSTRUCTOR.NAME.S." "SECT.MAX"
## [13] "SECT.ENRL"          "UG.SECT.ENRL"       "UP.SECT.ENRL"
## [16] "GR.SECT.ENRL"       "VS.SECT.ENRL"       "VU.SECT.ENRL"
## [19] "VG.SECT.ENRL"       "XLST.GROUP"         "XLST.MAX"
## [22] "XLST.ENRL"          "TTL.ENRL"           "TTL.SEATS.AVAIL"
## [25] "XLST.WITH"          "XLST.CRN"           "XLST.LEAD.DEPT"
## [28] "HAS_SYLLABUS"       "TERM_num"
```

```
#group department into different colors
unique(huma_clean$DEPARTMENT)
```

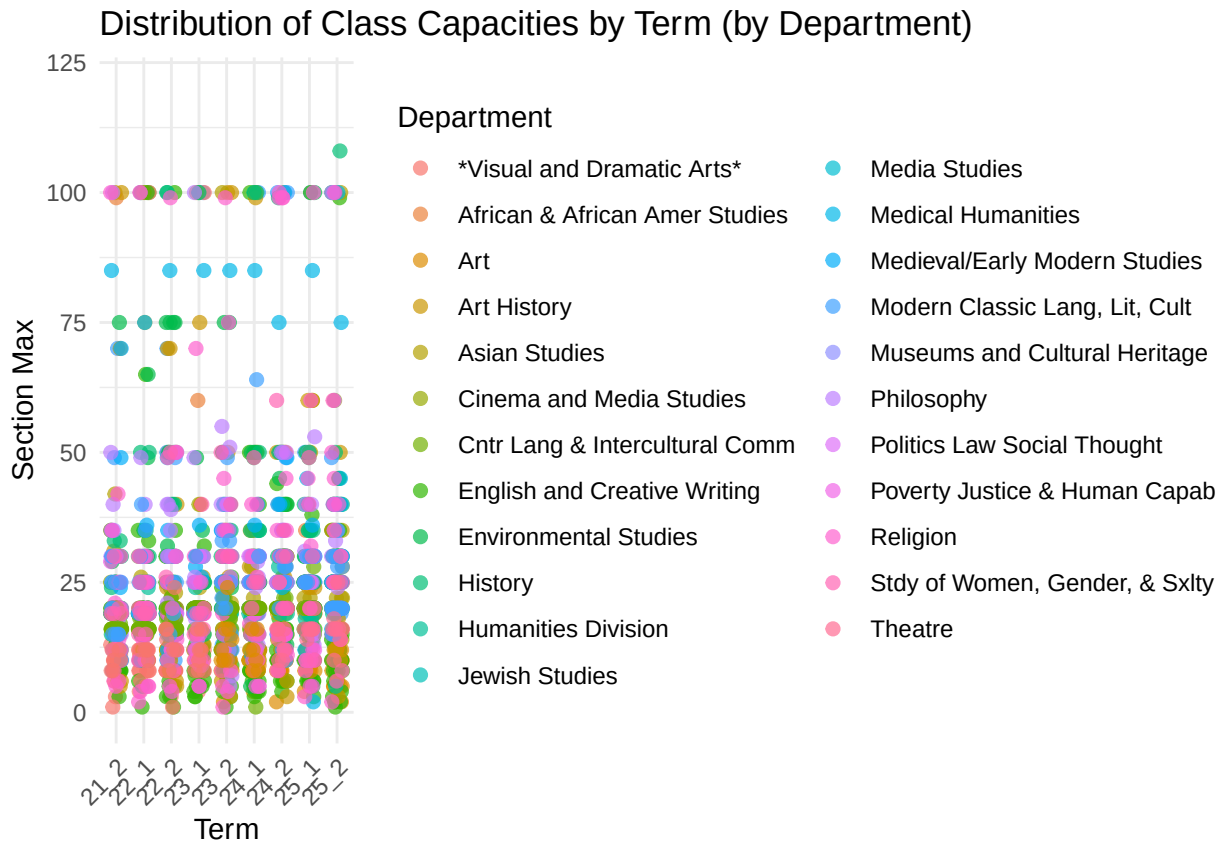
```
##  [1] "African & African Amer Studies" "Cntr Lang & Intercultural Comm"
##  [3] "*Visual and Dramatic Arts*"     "Asian Studies"
##  [5] "Modern Classic Lang, Lit, Cult" "Cinema and Media Studies"
##  [7] "English and Creative Writing"   "Environmental Studies"
##  [9] "Art History"                    "History"
## [11] "Humanities Division"            "Philosophy"
## [13] "Jewish Studies"                 "Medieval/Early Modern Studies"
## [15] "Medical Humanities"             "Poverty Justice & Human Capab"
## [17] "Politics Law Social Thought"    "Religion"
## [19] "Stdy of Women, Gender, & Sxlty" "Art"
## [21] "Museums and Cultural Heritage"  "Theatre"
## [23] "Media Studies"
```

```
#  [1] "African & African Amer Studies" "Cntr Lang & Intercultural Comm" "*Visual and Dramatic Arts*"
#  [4] "Asian Studies"                  "Modern Classic Lang, Lit, Cult" "Cinema and Media Studies"
#  [7] "English and Creative Writing"   "Environmental Studies"          "Art History"
# [10] "History"                        "Humanities Division"            "Philosophy"
# [13] "Jewish Studies"                 "Medieval/Early Modern Studies"  "Medical Humanities"
# [16] "Poverty Justice & Human Capab"  "Politics Law Social Thought"    "Religion"
# [19] "Stdy of Women, Gender, & Sxlty" "Art"                            "Museums and Cultural Heritage
# [22] "Theatre"                        "Media Studies"


#obtain counts of each class!
```

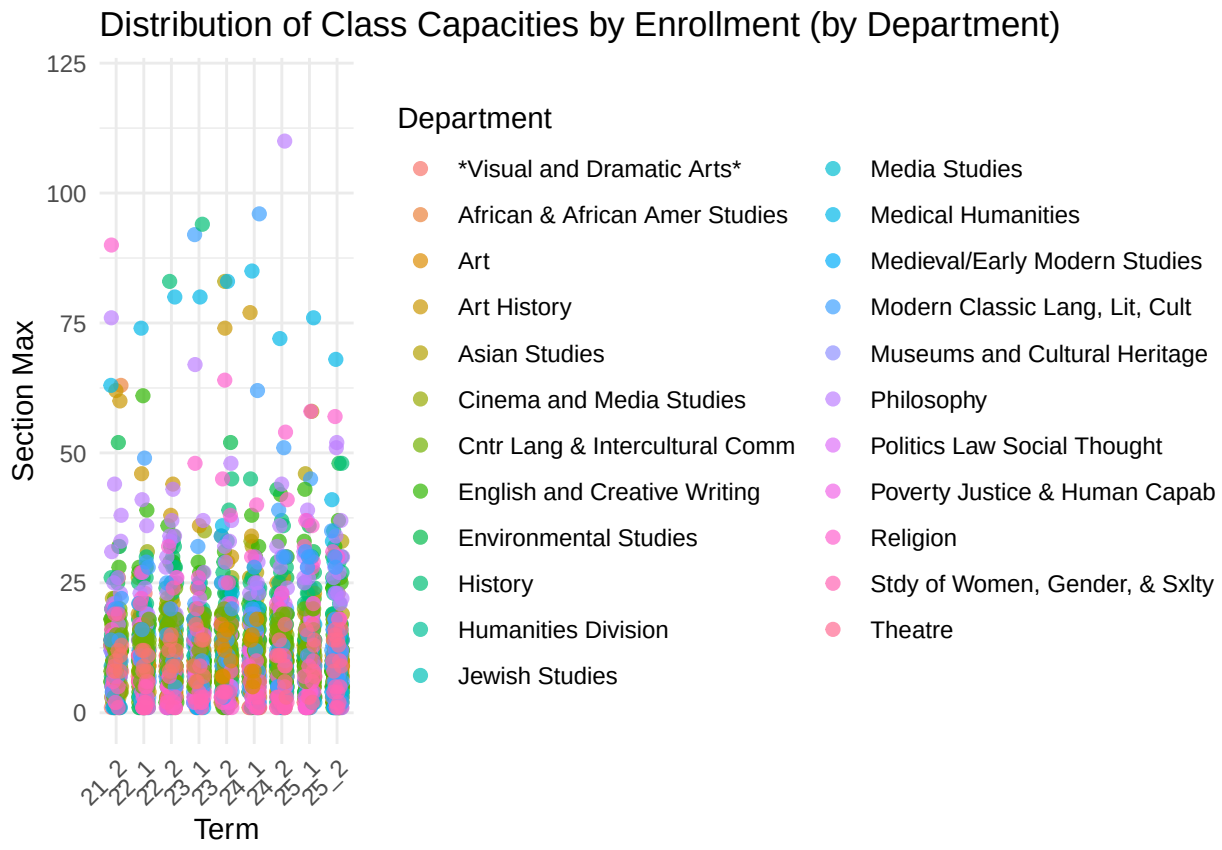
```r
dept_counts <- huma_clean %>%
  group_by(DEPARTMENT) %>%
  summarize(n_points = n(), .groups = "drop")
```

```r
library(ggplot2)

#CLASS CAP BY TERM#
ggplot(huma_clean, aes(x = TERM, y = SECT.MAX, color = DEPARTMENT)) +
  geom_jitter(size = 2, alpha = 0.7, width = 0.2, height = 0) +  # jitter avoids overlapping points
  labs(
    x = "Term",
    y = "Section Max",
    title = "Distribution of Class Capacities by Term (by Department)",
    color = "Department"  # legend title
  ) +
  ylim(0, 120) +  # adjust as needed
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

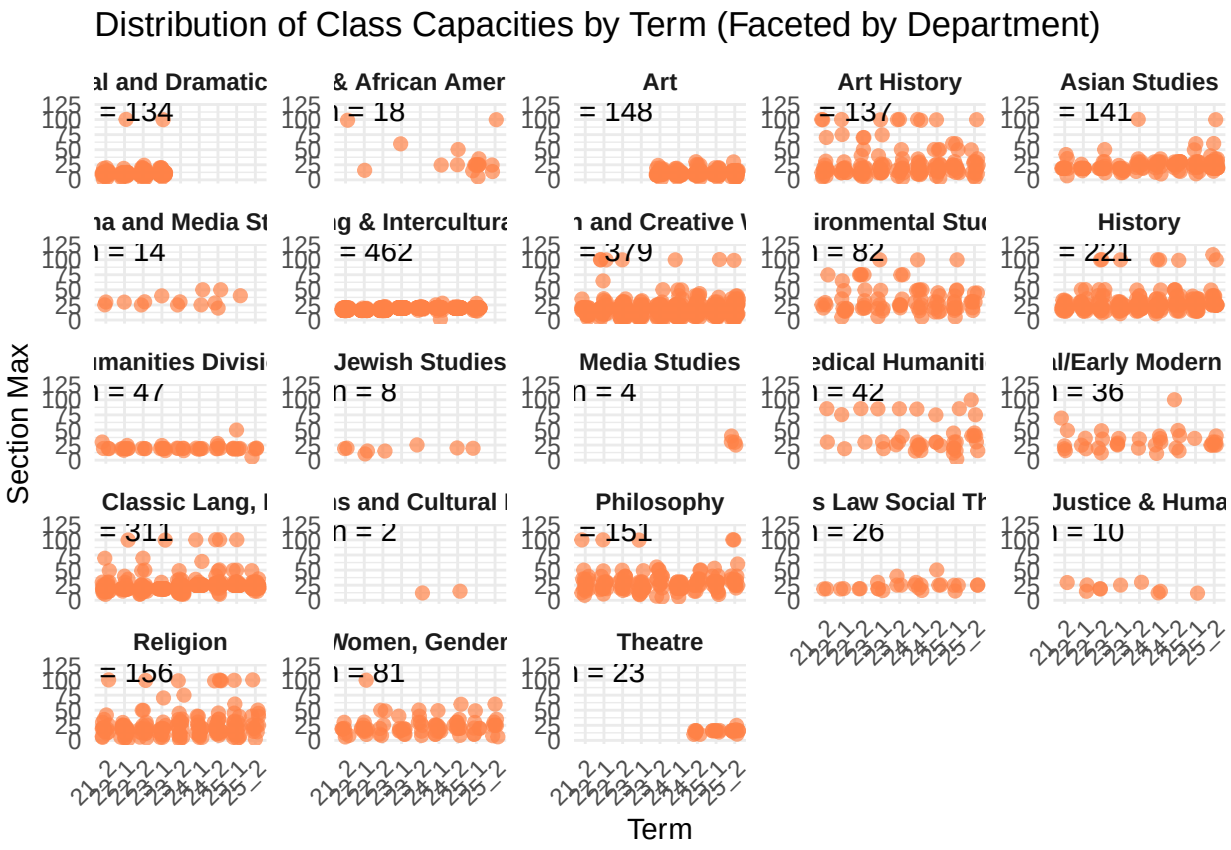


```
#CLASS ENROLLMENT BY TERM#
ggplot(huma_clean, aes(x = TERM, y = SECT.ENRL, color = DEPARTMENT)) +
  geom_jitter(size = 2, alpha = 0.7, width = 0.2, height = 0) +  # jitter avoids overlapping points
  labs(
    x = "Term",
    y = "Section Max",
    title = "Distribution of Class Capacities by Enrollment (by Department)",
    color = "Department"  # legend title
  ) +
  ylim(0, 120) +  # adjust as needed
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
ggplot(huma_clean, aes(x = TERM, y = SECT.MAX)) +
  geom_jitter(color = "sienna1", size = 2, alpha = 0.7, width = 0.2) +
  # Add the count label inside each facet
  geom_text(
    data = dept_counts,
    aes(x = 2, y = 115, label = paste0("n = ", n_points)), # adjust y for placement
    inherit.aes = FALSE
  ) +
  labs(
    x = "Term",
    y = "Section Max",
    title = "Distribution of Class Capacities by Term (Faceted by Department)"
```

```
  ) +
  coord_cartesian(ylim = c(0, 120)) +
  facet_wrap(~ DEPARTMENT, scales = "free_y") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    strip.text = element_text(face = "bold")
  )
```

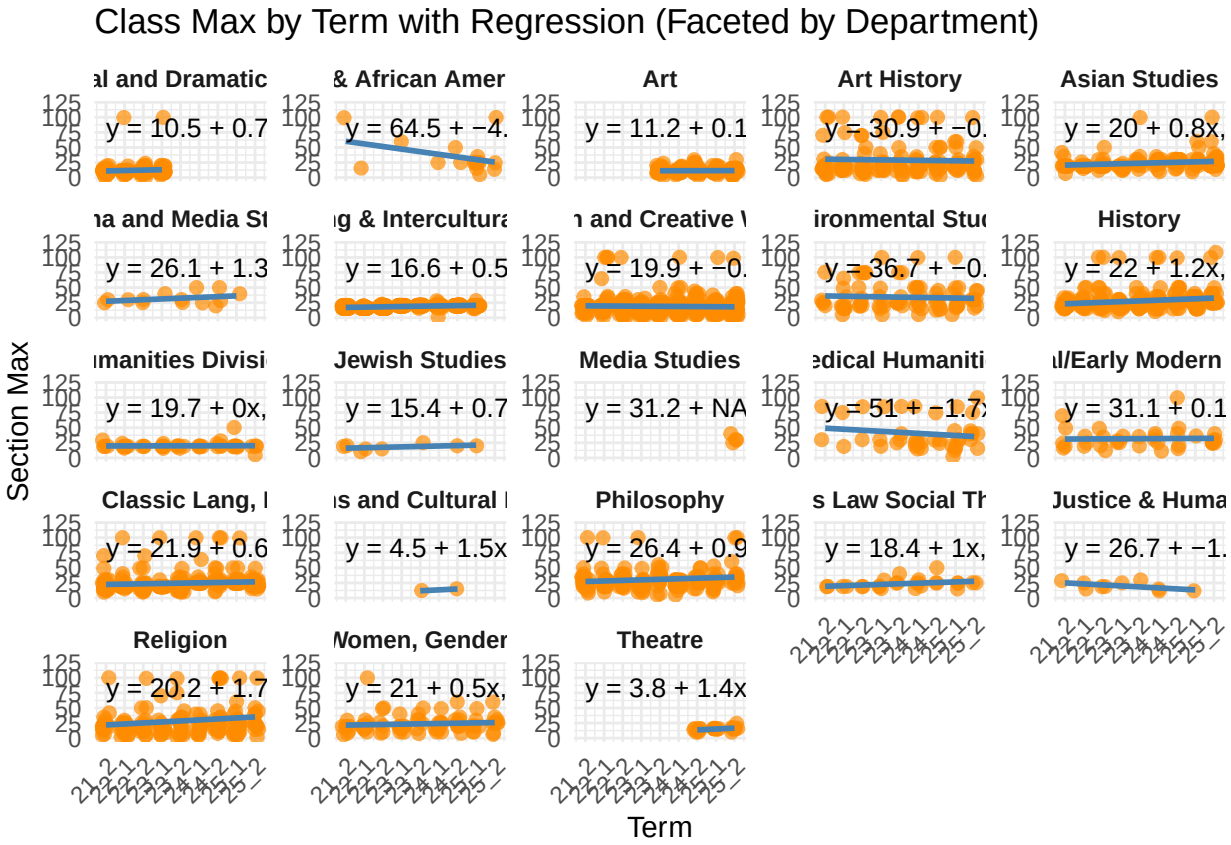



```
# Simple plain-text regression equation
lm_eqn_r2 <- function(df) {
  fit <- lm(SECT.MAX ~ TERM_num, data = df)
  r2 <- summary(fit)$r.squared
  paste0("y = ", round(coef(fit)[1], 1),
         " + ", round(coef(fit)[2], 1), "x",
         ", R² = ", round(r2, 2))
}

label_data <- huma_clean %>%
  group_by(DEPARTMENT) %>%
  summarise(
    n_points = n(),
    eq_r2 = lm_eqn_r2(cur_data()),
    .groups = "drop"
  )
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `eq_r2 = lm_eqn_r2(cur_data())`.
## i In group 1: `DEPARTMENT = "*Visual and Dramatic Arts*"`.
## Caused by warning:
## ! `cur_data()` was deprecated in dplyr 1.1.0.
## i Please use `pick()` instead.
```

```
ggplot(huma_clean, aes(x = TERM_num, y = SECT.MAX)) +
  geom_jitter(color = "darkorange", size = 2, alpha = 0.7, width = 0.2) +
  geom_smooth(method = "lm", se = FALSE, color = "steelblue") +
  geom_text(
    data = label_data,
    aes(x = 1, y = 115, label = paste0("n = ", n_points, "\n", eq_r2)),
    inherit.aes = FALSE,
    hjust = 0,
    size = 3.5
  ) +
  scale_x_continuous(
    breaks = 1:length(term_levels),
    labels = term_levels
  ) +
  facet_wrap(~ DEPARTMENT, scales = "free_y") +
  coord_cartesian(ylim = c(0, 120)) +
  labs(
    x = "Term",
    y = "Section Max",
    title = "Class Max by Term with Regression (Faceted by Department)"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    strip.text = element_text(face = "bold")
  )
```

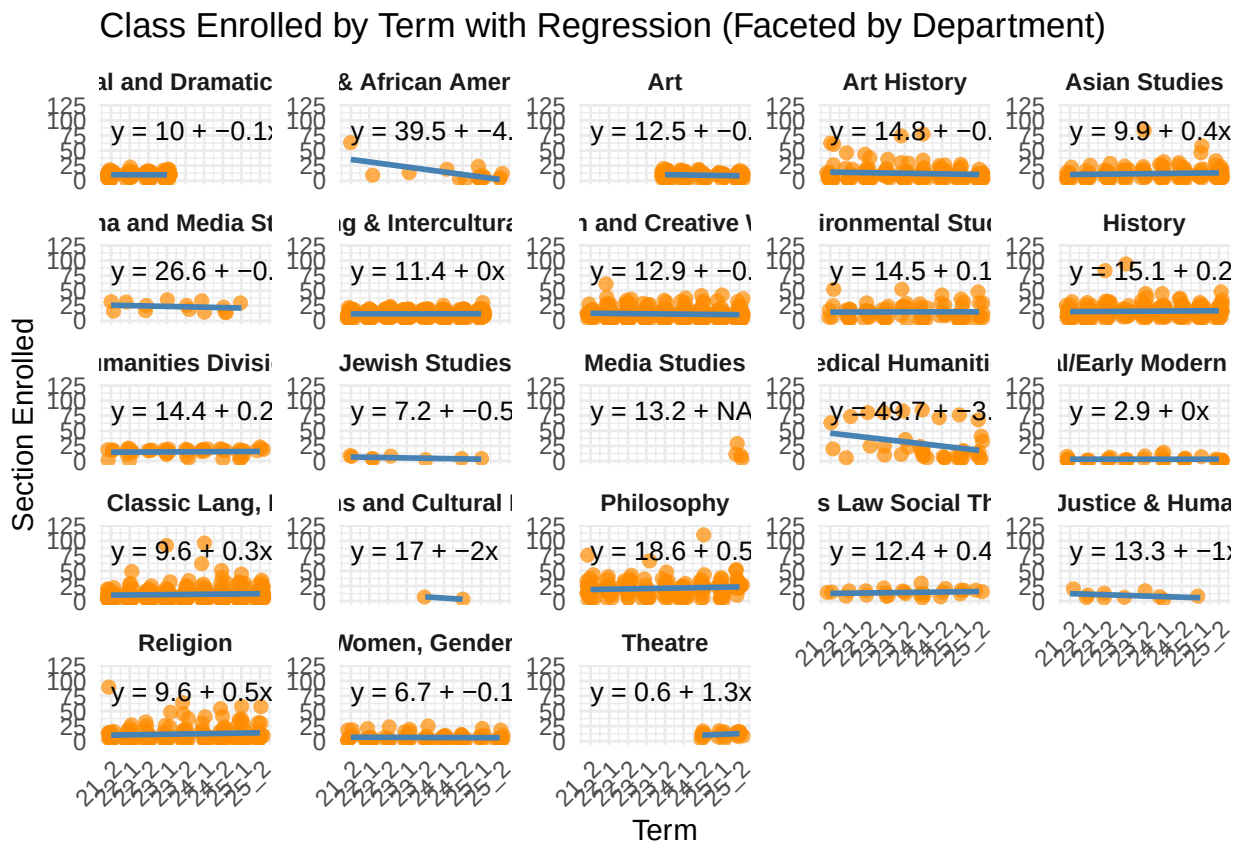
```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Simple plain-text regression equation
lm_eqn_enrl <- function(df) {
  fit <- lm(SECT.ENRL ~ TERM_num, data = df)
  # r2 <- summary(fit)$r.squared
  paste0("y = ", round(coef(fit)[1], 1),
         " + ", round(coef(fit)[2], 1), "x")
}

enrl_data <- huma_clean %>%
  group_by(DEPARTMENT) %>%
  summarise(
    n_points = n(),
    eq_enrl = lm_eqn_enrl(cur_data()),
    .groups = "drop"
  )



ggplot(huma_clean, aes(x = TERM_num, y = SECT.ENRL)) +
  geom_jitter(color = "darkorange", size = 2, alpha = 0.7, width = 0.2) +
  geom_smooth(method = "lm", se = FALSE, color = "steelblue") +
  geom_text(
    data = enrl_data,
    aes(x = 1, y = 115, label = paste0("n = ", n_points, "\n", eq_enrl)),
    inherit.aes = FALSE,
    hjust = 0,
```

```
    size = 3.5
  ) +
  scale_x_continuous(
    breaks = 1:length(term_levels),
    labels = term_levels
  ) +
  facet_wrap(~ DEPARTMENT, scales = "free_y") +
  coord_cartesian(ylim = c(0, 120)) +
  labs(
    x = "Term",
    y = "Section Enrolled",
    title = "Class Enrolled by Term with Regression (Faceted by Department)"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    strip.text = element_text(face = "bold")
  )
```

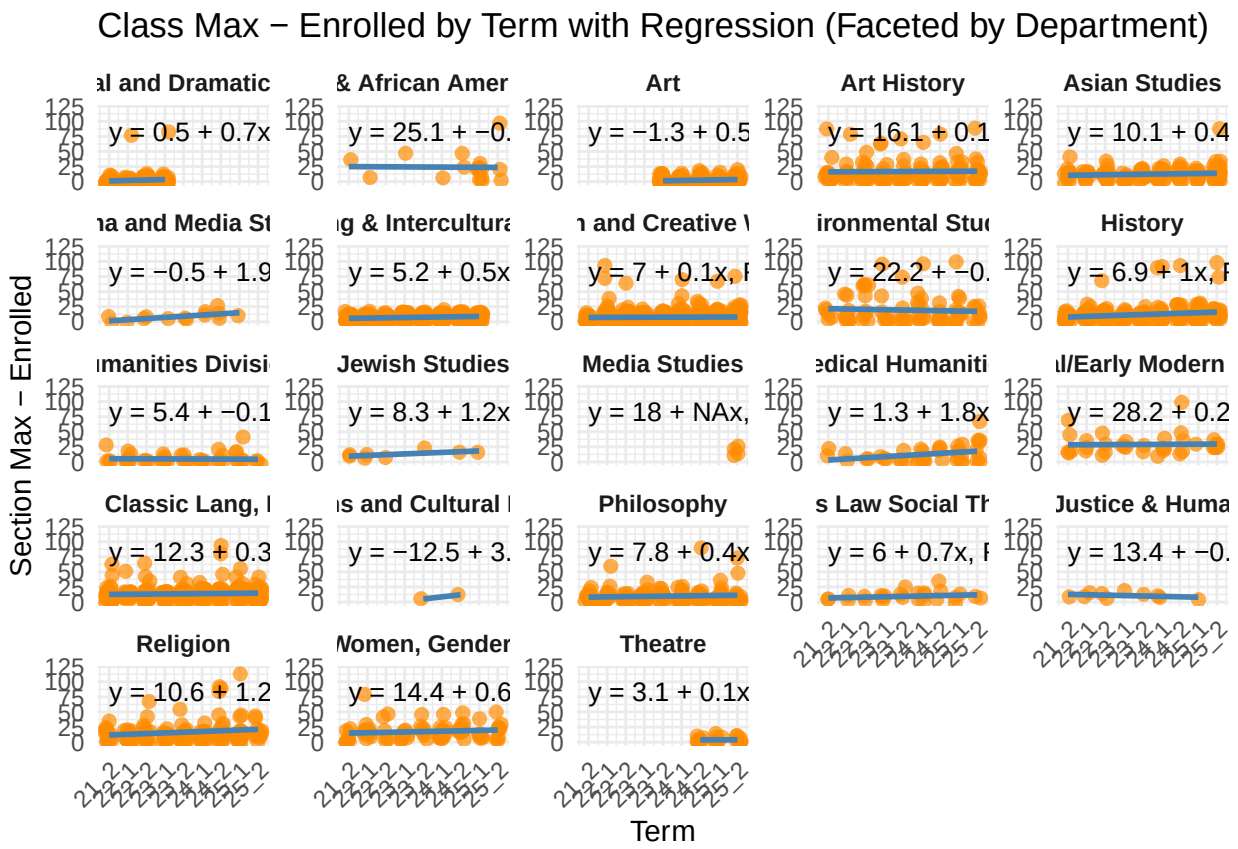
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Simple plain-text regression equation
lm_eqn_diff <- function(df) {
  fit <- lm((SECT.MAX-SECT.ENRL) ~ TERM_num, data = df)
  r2 <- summary(fit)$r.squared
```

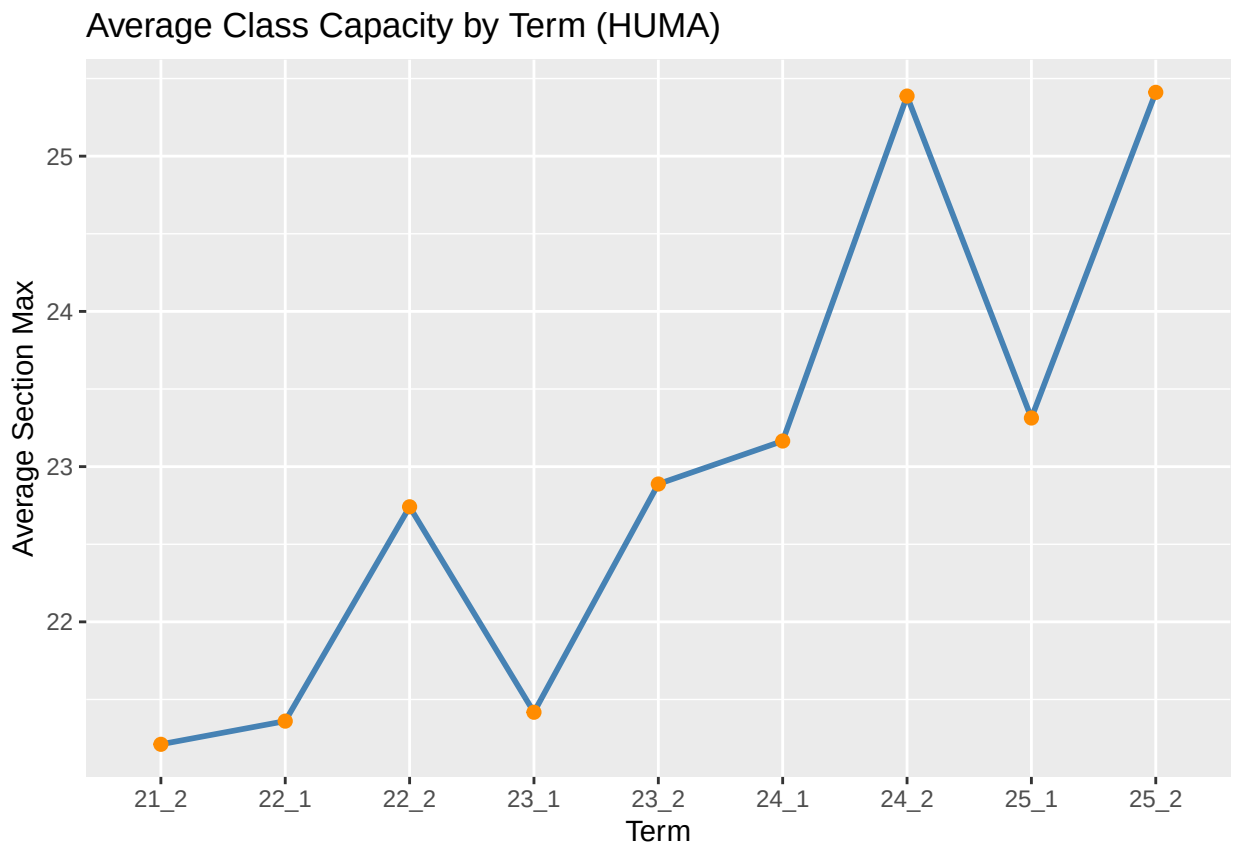
```
  paste0("y = ", round(coef(fit)[1], 1),
         " + ", round(coef(fit)[2], 1), "x",
         ", R² = ", round(r2, 2))
}

label_diff <- huma_clean %>%
  group_by(DEPARTMENT) %>%
  summarise(
    n_points = n(),
    eq_r2 = lm_eqn_diff(cur_data()),
    .groups = "drop"
  )

ggplot(huma_clean, aes(x = TERM_num, y = (SECT.MAX-SECT.ENRL))) +
  geom_jitter(color = "darkorange", size = 2, alpha = 0.7, width = 0.2) +
  geom_smooth(method = "lm", se = FALSE, color = "steelblue") +
  geom_text(
    data = label_diff,
    aes(x = 1, y = 115, label = paste0("n = ", n_points, "\n", eq_r2)),
    inherit.aes = FALSE,
    hjust = 0,
    size = 3.5
  ) +
  scale_x_continuous(
    breaks = 1:length(term_levels),
    labels = term_levels
  ) +
  facet_wrap(~ DEPARTMENT, scales = "free_y") +
  coord_cartesian(ylim = c(0, 120)) +
  labs(
    x = "Term",
    y = "Section Max - Enrolled",
    title = "Class Max - Enrolled by Term with Regression (Faceted by Department)"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    strip.text = element_text(face = "bold")
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

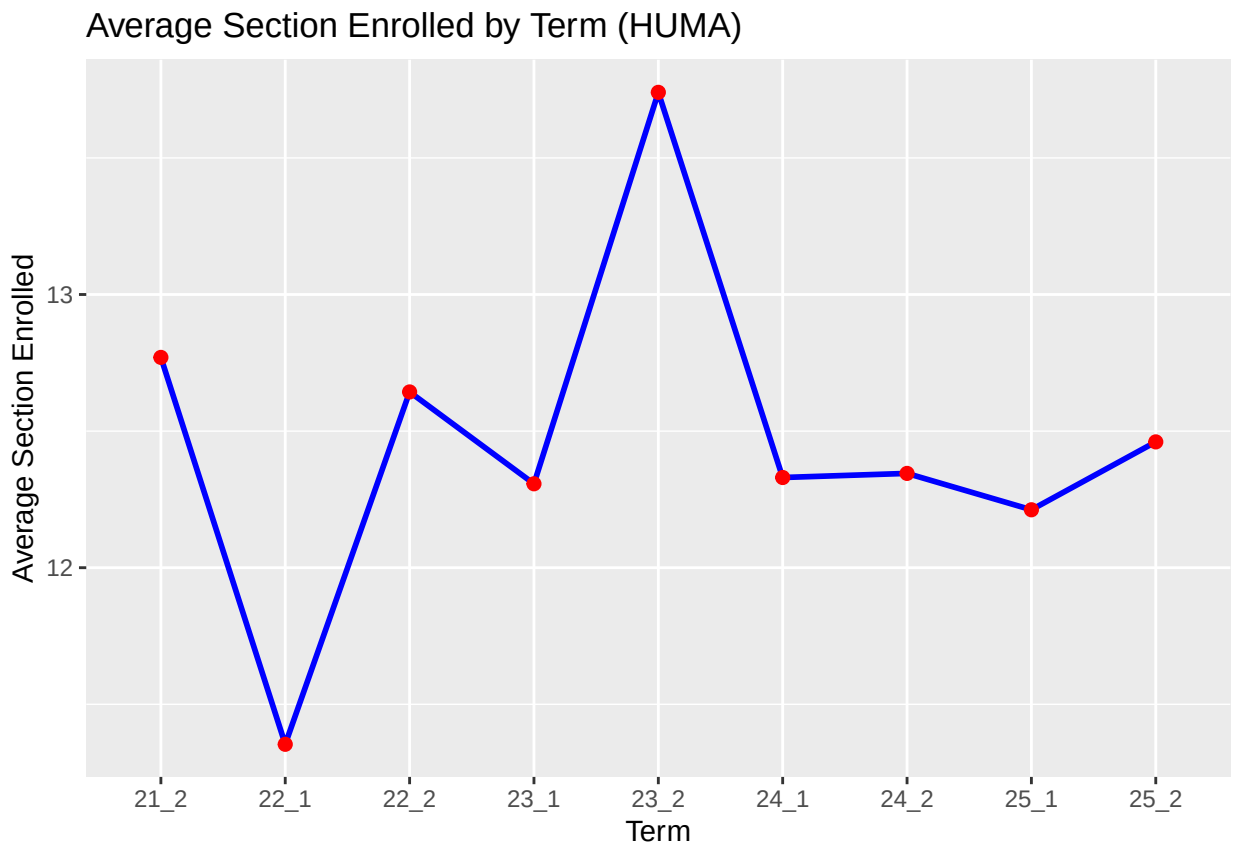



```
huma_summary <- huma_clean %>%
  group_by(TERM) %>%
  summarize(mean_max = mean(SECT.MAX, na.rm = TRUE))

huma_enrolled <- huma_clean %>%
  group_by(TERM) %>%
  summarize(mean_enrl = mean(SECT.ENRL, na.rm = TRUE))

ggplot(huma_summary, aes(x = TERM, y = mean_max, group = 1)) +
  geom_line(color = "steelblue", size = 1) +
  geom_point(color = "darkorange", size = 2) +
  labs(
    x = "Term",
    y = "Average Section Max",
    title = "Average Class Capacity by Term (HUMA)"
  )
```

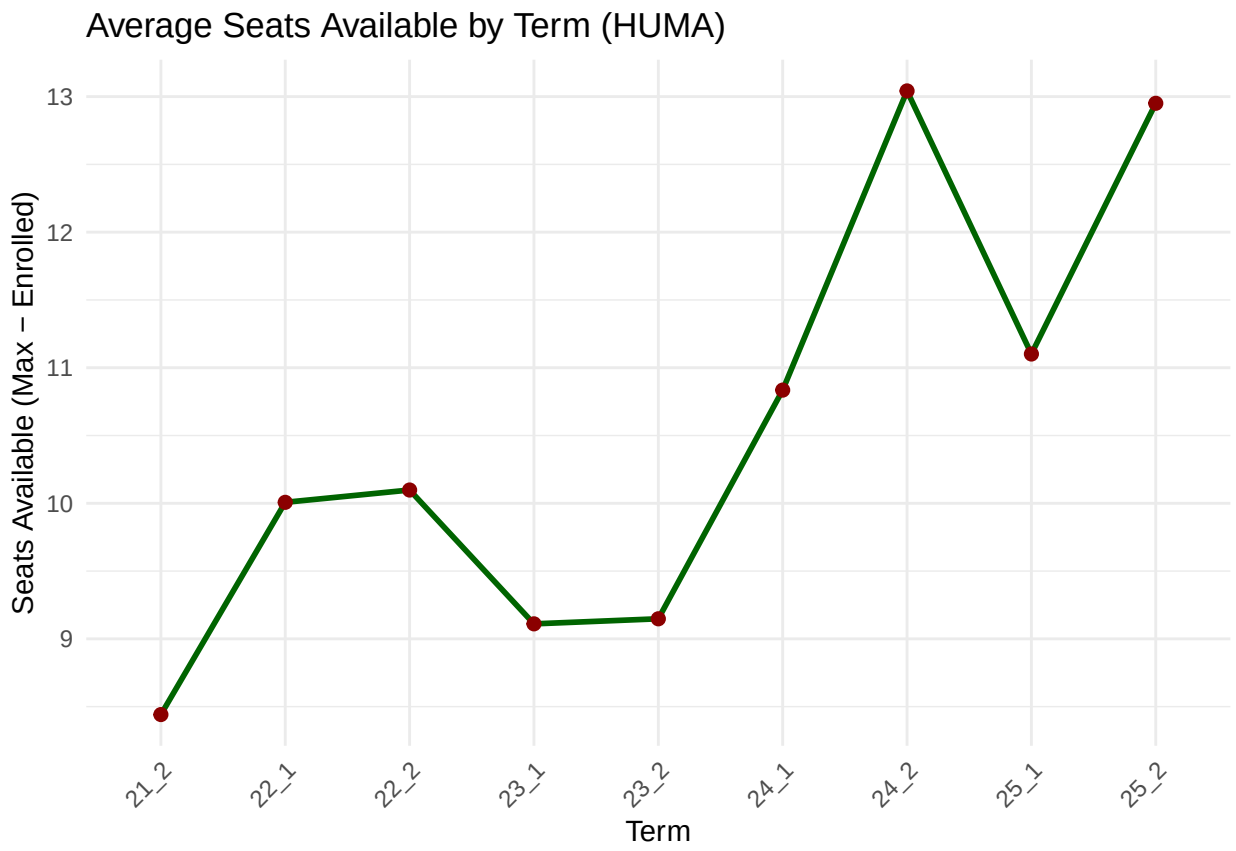
```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



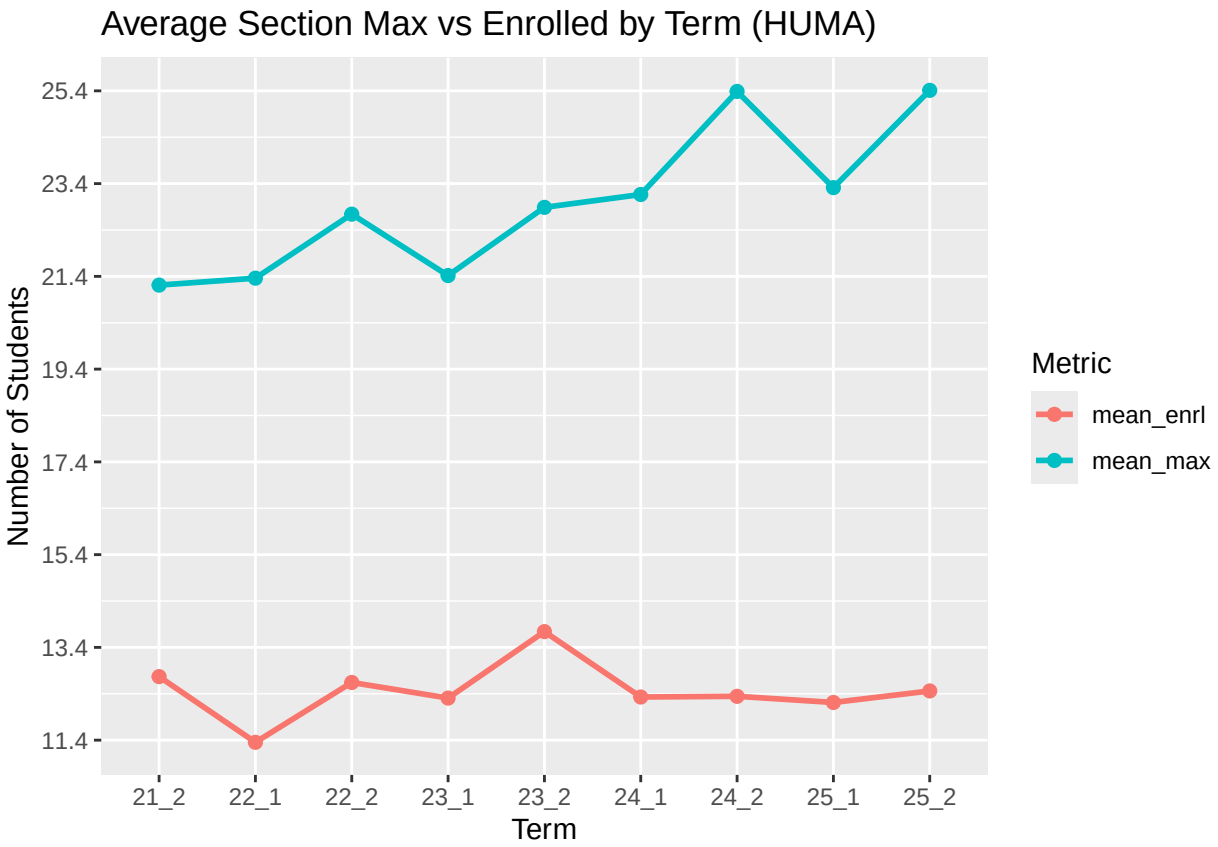

```
ggplot(huma_enrolled, aes(x = TERM, y = mean_enrl, group = 1)) +
  geom_line(color = "blue", size = 1) +
  geom_point(color = "red", size = 2) +
  labs(
    x = "Term",
    y = "Average Section Enrolled",
    title = "Average Section Enrolled by Term (HUMA)"
  )
```




```
huma_summary <- huma_clean %>%
  group_by(TERM) %>%
  summarize(
    mean_max = mean(SECT.MAX, na.rm = TRUE),
    mean_enrl = mean(SECT.ENRL, na.rm = TRUE)
  ) %>%
  mutate(
    seats_available = mean_max - mean_enrl
  )

ggplot(huma_summary, aes(x = TERM, y = seats_available, group = 1)) +
  geom_line(color = "darkgreen", size = 1) +
  geom_point(color = "darkred", size = 2) +
  labs(
    x = "Term",
    y = "Seats Available (Max - Enrolled)",
    title = "Average Seats Available by Term (HUMA)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



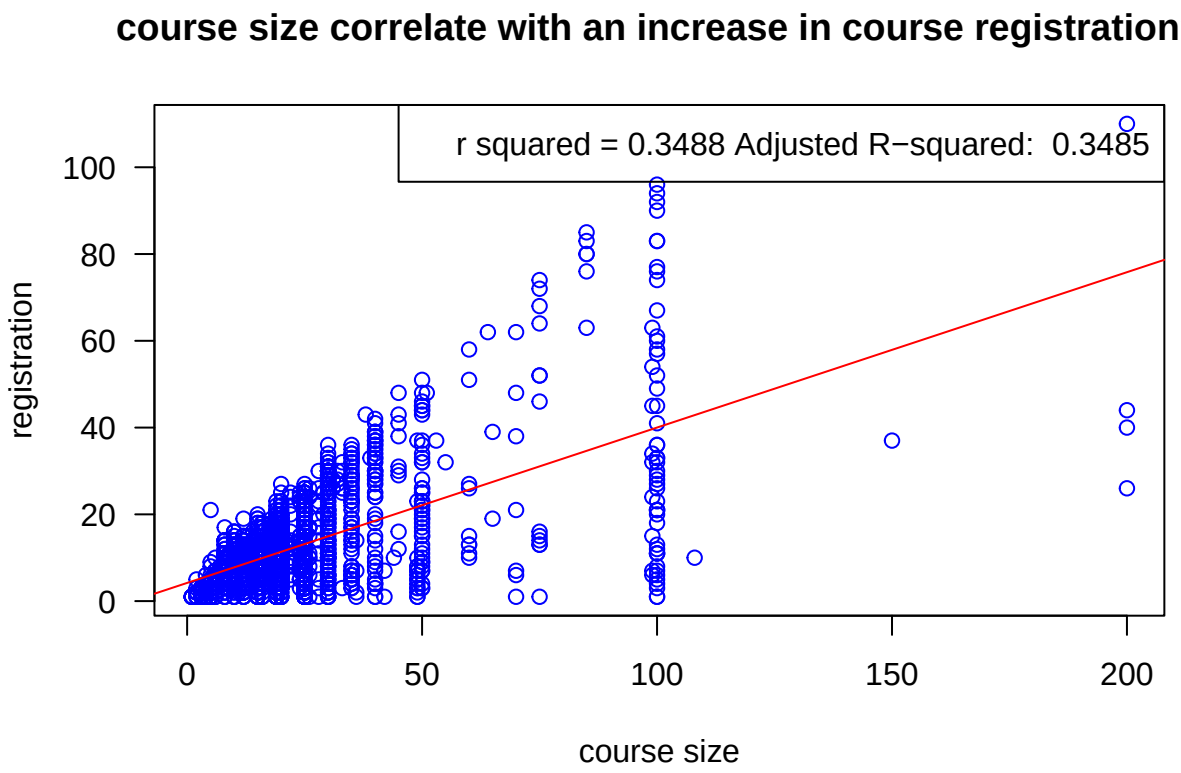

```
library(dplyr)
library(tidyr)
huma_long <- huma_summary %>%
  select(TERM, mean_max) %>%
  left_join(huma_enrolled %>% select(TERM, mean_enrl), by = "TERM") %>%
  pivot_longer(
    cols = c(mean_max, mean_enrl),
    names_to = "Type",
    values_to = "Value"
  )


ggplot(huma_long, aes(x = TERM, y = Value, color = Type, group = Type)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(
    x = "Term",
    y = "Number of Students",
    color = "Metric",
    title = "Average Section Max vs Enrolled by Term (HUMA)"
  ) +
  scale_y_continuous(breaks = round(seq(min(huma_long$Value), max(huma_long$Value), by = 2),1))
```



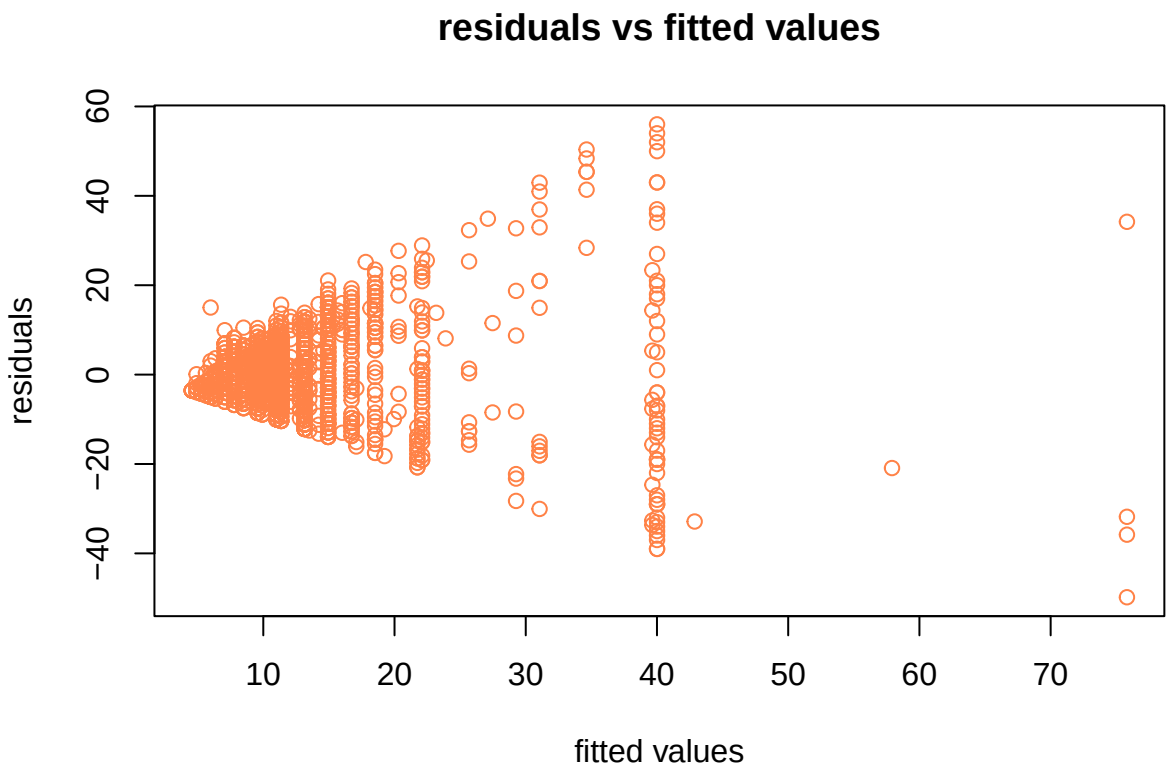

```
#department, max, enrolled

# summary(course_reg)
x <- huma_clean$SECT.MAX
y <- huma_clean$SECT.ENRL
course_linreg <- lm(y ~ x)
summary(course_linreg)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.814  -4.927  -0.359   4.073  55.994
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.197814   0.275508   15.24   <2e-16 ***
## x           0.358080   0.009539   37.54   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.516 on 2631 degrees of freedom
## Multiple R-squared:  0.3488, Adjusted R-squared:  0.3485
## F-statistic:  1409 on 1 and 2631 DF,  p-value: < 2.2e-16
```

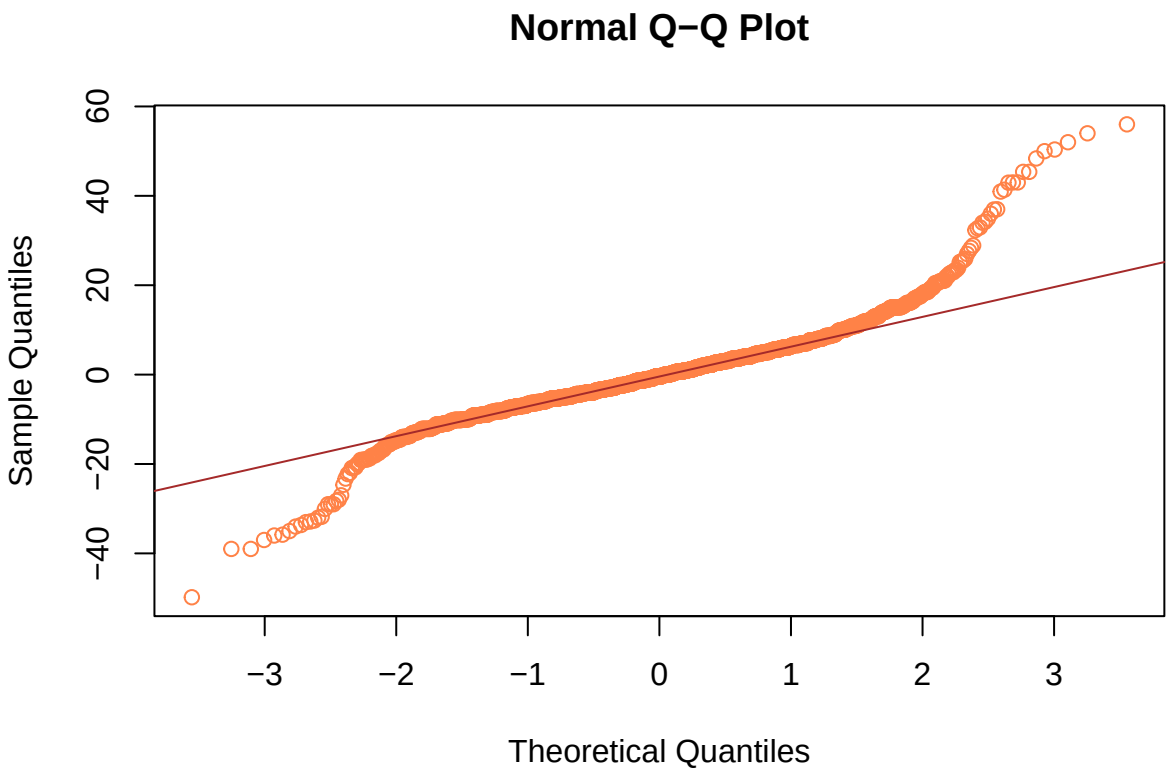
```r
plot(x,y, main="course size correlate with an increase in course registration?", xlab="course size", yla
legend("topright", legend=c("r squared = 0.3488 Adjusted R-squared:  0.3485"))
abline(a = course_linreg$coefficients[1], b = course_linreg$coefficients[2], col = "red")
```




```r
res <- residuals(course_linreg)
fit <- fitted.values(course_linreg)
pch_vec <- c(1, 0)

plot(fit, res, main = "residuals vs fitted values", xlab = "fitted values", ylab = "residuals", col = "
```




```
qqnorm(res, col = "sienna1")
qqline(res, col = "brown")
```


Normal Q−Q Plot
Sample Quantiles
60
40
20
0
−20
−40
−3
−2
−1
0
1
2
3
Theoretical Quantiles