# Homework 4 Solution Script

Ling Zhu and Songeun Emily Lee

10/24/2025

## 0. Loading Packages

```r
rm(list=ls()) # Start clean: remove all objects from the workspace
## Libraries
library(foreign)
library(ggplot2)
library(dplyr)
library(ggpubr)
library(VGAM)
library(MASS)
library(GGally)
library(censReg)
library(sampleSelection)
library(tidyr)
library(mvtnorm)
library(AER)        # dispersiontest for Poisson overdispersion
library(MASS)       # glm.nb (Negative Binomial)
library(dplyr)      # data wrangling
library(tidyr)      # pivot_longer
library(ggplot2)    # plotting
library(stargazer)  # model tables

## Working directory
setwd("/Users/songeunlee/Desktop/Ling/HW4")
#setwd("/Users/lingzhu/Dropbox/UH Teaching/POLS6382_2025/HW Assignments/2025 HW Review and Solution Scr

## Data
CESdata<-read.csv("CESdata.csv")
```

## 1. Analyzing 2022 Midterm Election Data

### Qeustion 1a and 1b: Data Preparation

```r
# ----------------------------------------------------------------
# Question 1a:
# Download the 2022 CES post-election data from https://cces.gov.harvard.edu/
# This code cleans and recodes key variables for analysis.
# ----------------------------------------------------------------

# Load packages
```

```r
library(dplyr)
library(tidyr)

# Remove missing cases only for the variables we need
CESdata <- CESdata %>%
  drop_na(race, birthyear, gender, party, votereg, vote, education)

# Recode variables
CESdata <- CESdata %>%
  mutate(
    white     = ifelse(race == 1, 1, 0),           # 1 = White, 0 = Non-White
    age       = 2022 - birthyear,                  # Age = 2022 minus birth year
    male      = ifelse(gender == 1, 1, 0),         # 1 = Male, 0 = Female/Other
    Democrat  = ifelse(party == 1, 1, 0),          # 1 = Democrat, 0 = Others
    votereg   = ifelse(votereg == 1, 1, 0),        # 1 = Registered, 0 = Not/Don't know
    vote      = ifelse(vote == 5, 1, 0),           # 1 = Voted, 0 = Did not vote
    college   = ifelse(education %in% c(3,4,5,6), 1, 0)  # 1 = Some college or more
  )

# Keep only the variables needed for this exercise
data1 <- CESdata %>%
  dplyr::select(white, age, male, Democrat, votereg, vote, college)

# Check data summary
summary(data1) # Check the dataset with View(data1) or colnames(data1)
```

```
     white             age             male           Democrat     
 Min.   :0.0000   Min.   :18.00   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:39.00   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :1.0000   Median :54.00   Median :1.0000   Median :0.0000  
 Mean   :0.7469   Mean   :52.31   Mean   :0.5241   Mean   :0.3204  
 3rd Qu.:1.0000   3rd Qu.:65.00   3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :95.00   Max.   :1.0000   Max.   :1.0000  
    votereg            vote          college      
 Min.   :0.0000   Min.   :0.000   Min.   :0.0000  
 1st Qu.:1.0000   1st Qu.:1.000   1st Qu.:1.0000  
 Median :1.0000   Median :1.000   Median :1.0000  
 Mean   :0.9753   Mean   :0.802   Mean   :0.7695  
 3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :1.000   Max.   :1.0000  
```

```r
# Save the cleaned dataset
write.csv(data1, "ces2022_clean.csv", row.names = FALSE)
```

## Question 1c: Estimating a Standard Logit

```r
# -----------------------------------------------------------------
# Question 1c: Predicting the Probability of Voting
# Estimate standard logit models using 2022 CES data
# -----------------------------------------------------------------

library(dplyr)
library(ggplot2)
```

```r
# --- Model 1: Predict voting using demographics and partisanship ---
logit_model1 <- glm(vote ~ white + age + male + Democrat + college,
                    data = data1, family = binomial)

summary(logit_model1)          # Model summary (log-odds)
```

```

Call:
glm(formula = vote ~ white + age + male + Democrat + college,
    family = binomial, data = data1)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.768356   0.091111  -30.38   <2e-16 ***
white        0.548546   0.048802   11.24   <2e-16 ***
age          0.050474   0.001502   33.60   <2e-16 ***
male         0.592213   0.045699   12.96   <2e-16 ***
Democrat     0.864663   0.053551   16.15   <2e-16 ***
college      1.138215   0.049065   23.20   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 15139  on 15209  degrees of freedom
Residual deviance: 12525  on 15204  degrees of freedom
AIC: 12537

Number of Fisher Scoring iterations: 5
```

```r
exp(coef(logit_model1))        # Exponentiated coefficients (odds ratios)
```

```
(Intercept)       white         age        male    Democrat     college
 0.06276513  1.73073411  1.05176904  1.80798522  2.37420573  3.12119299
```

```r
# --- Model 2: Add voter registration as an additional predictor ---
logit_model2 <- glm(vote ~ white + age + male + Democrat + college + votereg,
                    data = data1, family = binomial)

summary(logit_model2)
```

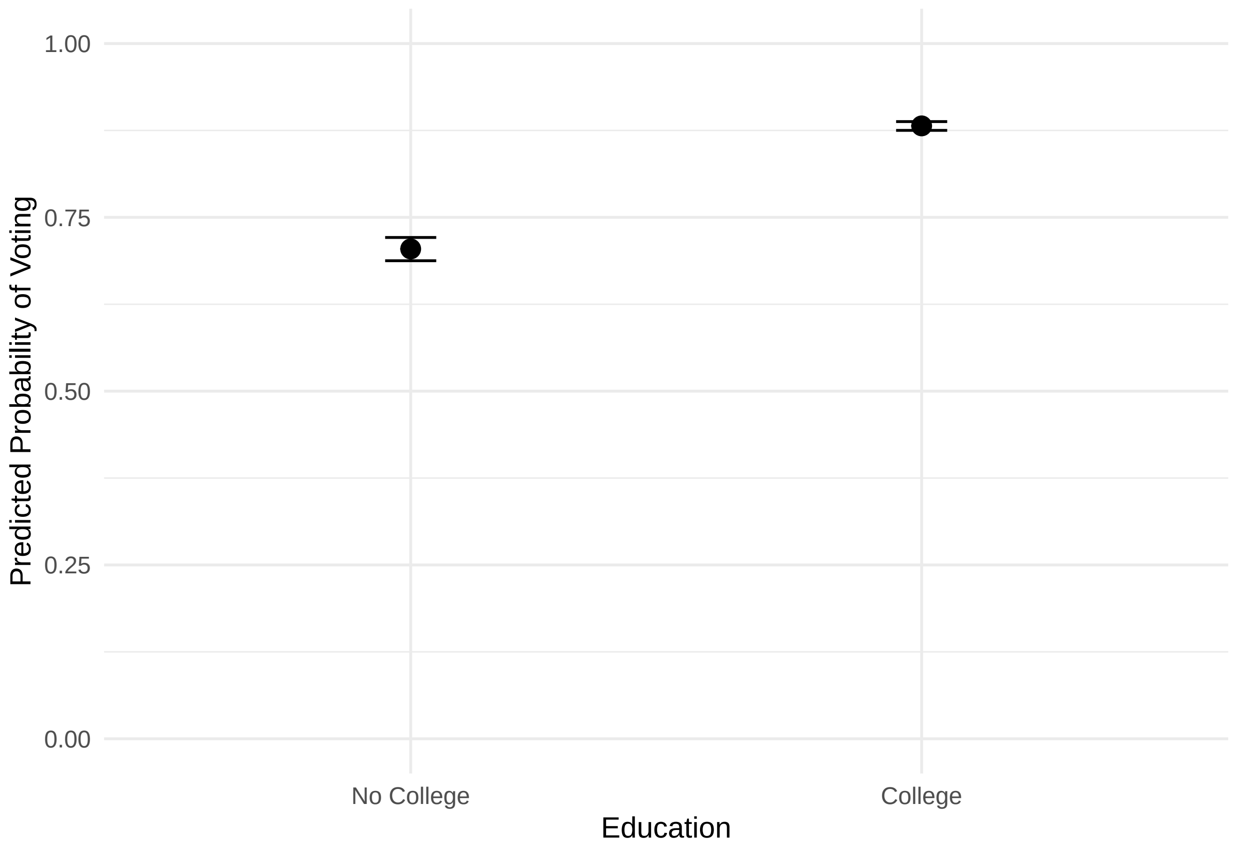
```

Call:
glm(formula = vote ~ white + age + male + Democrat + college +
    votereg, family = binomial, data = data1)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.983402   0.169777  -29.35   <2e-16 ***
white        0.539877   0.049750   10.85   <2e-16 ***
age          0.048461   0.001529   31.69   <2e-16 ***
male         0.587214   0.046604   12.60   <2e-16 ***
Democrat     0.829619   0.054450   15.24   <2e-16 ***
college      1.085820   0.050171   21.64   <2e-16 ***
votereg      2.443710   0.146225   16.71   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 15139  on 15209  degrees of freedom
Residual deviance: 12163  on 15203  degrees of freedom
AIC: 12177

Number of Fisher Scoring iterations: 5
```

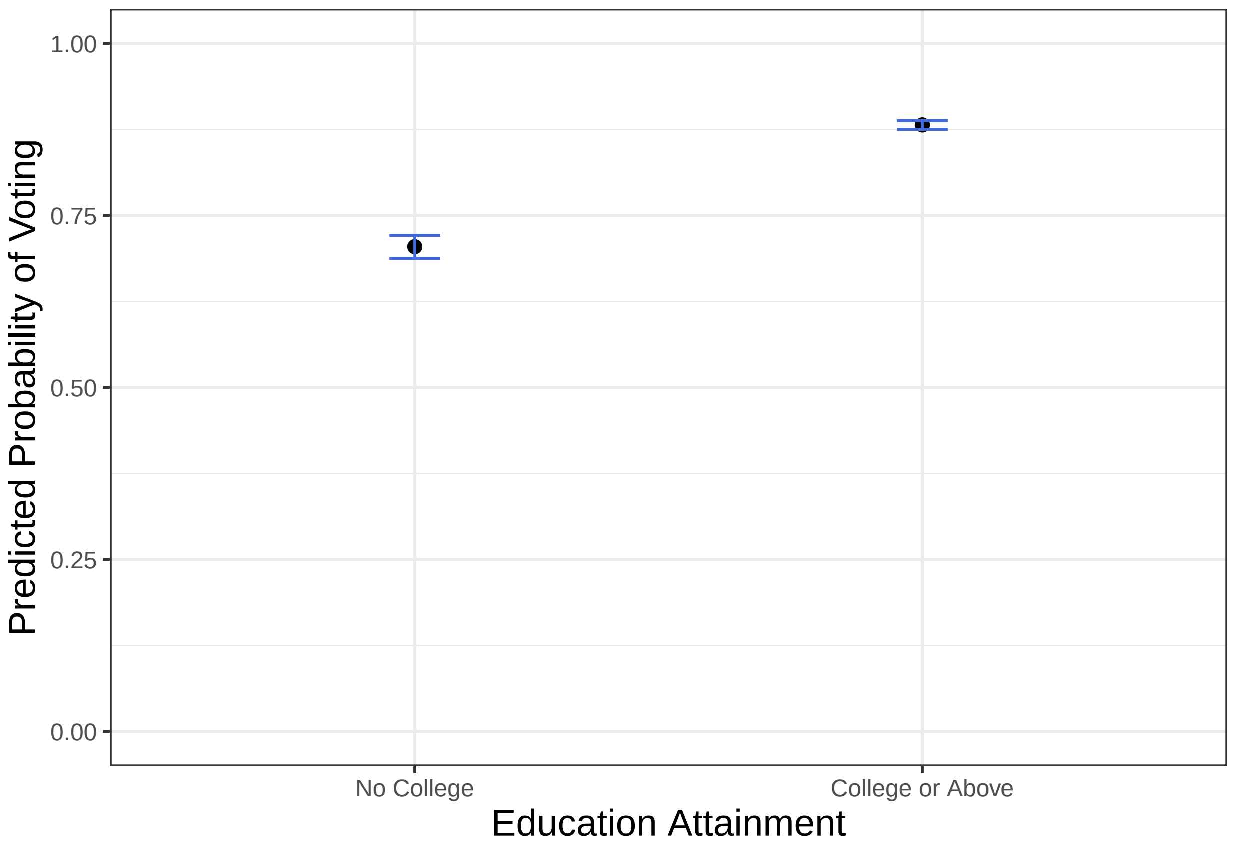
```r
exp(coef(logit_model2))
```

```
 (Intercept)        white          age         male     Democrat      college 
 0.006850715  1.715794995  1.049654065  1.798969006  2.292444068  2.961867321 
    votereg 
11.515685773 
```

```r
# ---------------------------------------------------------------
# Plotting Predicted Probabilities (Model 1)
# ---------------------------------------------------------------

# Calculate variable means for continuous and binary predictors
mean_white    <- mean(data1$white)
mean_age      <- mean(data1$age)
mean_male     <- mean(data1$male)
mean_Democrat <- mean(data1$Democrat)

# Create a new dataset varying college (0 = No College, 1 = College)
new_data <- data.frame(
  white     = mean_white,
  age       = mean_age,
  male      = mean_male,
  Democrat  = mean_Democrat,
  college   = c(0, 1)
)

# Predict log-odds and convert to probabilities with 95% CI
pred <- predict(logit_model1, newdata = new_data, type = "link", se.fit = TRUE)
prob <- plogis(pred$fit)
lower <- plogis(pred$fit - 1.96 * pred$se.fit)
upper <- plogis(pred$fit + 1.96 * pred$se.fit)

plot_data <- data.frame(
  college = factor(new_data$college, labels = c("No College", "College")),
  prob = prob, lower = lower, upper = upper
)

# --- Plot predicted probabilities with confidence intervals ---
ggplot(plot_data, aes(x = college, y = prob)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.1) +
  labs(x = "Education", y = "Predicted Probability of Voting") +
  ylim(0, 1) +
  theme_minimal()
```



```r
library(ggeffects)
pdturnout <- ggpredict(logit_model1, terms = "college [0,1]")

# make clean labels in the order you want
pdturnout$x <- factor(pdturnout$x,
                      levels = c(0, 1),
                      labels = c("No College", "College or Above"))

library(ggplot2)
ggplot(pdturnout, aes(x = x, y = predicted)) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high),
                width = 0.1, color = "royalblue") +
  labs(x = "Education Attainment",
       y = "Predicted Probability of Voting") +
  coord_cartesian(ylim = c(0, 1)) +  # optional: clamp to [0,1]
  theme_bw() +
  theme(
    legend.position = "none",
    axis.title = element_text(size = 14)
  )
```



### Question 1d: Estimating a Two-Step Heckman Selection Model

```r
# Load the package for Heckman selection models
library(sampleSelection)

# Estimate the Heckman two-step selection model
selectionmodel1 <- selection(
  selection = votereg ~ age + college,
  outcome   = vote ~ white + age + male + Democrat + college,
  data = data1,
  method = "2step"
)

# Display the model summary
summary(selectionmodel1)
```

```
--------------------------------------------
Tobit 2 model (sample selection model)
2-step Heckman / heckit estimation
15210 observations (376 censored and 14834 observed)
12 free parameters (df = 15199)
Probit selection equation:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.598875   0.075271   7.956  1.9e-15 ***
age         0.021775   0.001499  14.524  < 2e-16 ***
college     0.519222   0.047589  10.910  < 2e-16 ***
Outcome equation:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.6414035  0.0268734  23.868  < 2e-16 ***
white       0.0756957  0.0060049  12.606  < 2e-16 ***
age         0.0021870  0.0003772   5.797 6.87e-09 ***
male        0.0759778  0.0058483  12.991  < 2e-16 ***
Democrat    0.0960684  0.0062953  15.260  < 2e-16 ***
college     0.0432011  0.0140258   3.080  0.00207 **
Multiple R-Squared:0.1565,  Adjusted R-Squared:0.1561
   Error terms:
               Estimate Std. Error t value Pr(>|t|)
invMillsRatio  -1.8527        NaN     NaN      NaN
sigma           0.6922         NA      NA       NA
rho            -2.6766         NA      NA       NA
--------------------------------------------
```

```r
# install.packages("GJRM")
library(GJRM)

# Ensure both dependent variables are coded as 0/1 integers (Bernoulli responses expected by probit lin
data1 <- transform(
  data1,
  votereg = as.integer(votereg %in% 1),  # selection indicator: 1 = registered, 0 = not
  vote    = as.integer(vote %in% 1)      # outcome: 1 = voted, 0 = did not vote
)

# Specify the two equations:
# - eq_sel: selection (who is "in sample" for the outcome process-registered to vote)
# - eq_out: outcome (who votes among those eligible/selected)
eq_sel <- votereg ~ age + college
eq_out <- vote    ~ white + age + male + Democrat + college

# Fit a Bivariate Sample Selection (BSS) model:
# - FIML joint estimation of two Bernoulli-probit equations
# - Gaussian copula ties the latent errors, estimating their correlation (theta = rho)
# - margins = c("probit","probit") sets probit links for both equations
fit <- gjrm(
  list(eq_sel, eq_out),
  data    = data1,
  model   = "BSS",                  # bivariate probit selection model (FIML)
  margins = c("probit","probit")    # both equations are probit
)

# Summarize results:
# - Equation 1: probit coefficients for registration (selection)
# - Equation 2: probit coefficients for voting (outcome, conditional on selection)
# - theta: dependence parameter (latent error correlation = rho); tests endogenous selection
# - n / n.sel: total observations / number effectively in the selected regime
summary(fit)
```

```
COPULA: Gaussian
MARGIN 1: Bernoulli
MARGIN 2: Bernoulli

EQUATION 1
```

```
Link function for mu1: probit
Formula: votereg ~ age + college

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.601032   0.075181   7.994  1.3e-15 ***
age         0.021683   0.001497  14.483  < 2e-16 ***
college     0.521414   0.047525  10.971  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


EQUATION 2
Link function for mu2: probit
Formula: vote ~ white + age + male + Democrat + college

Parametric coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.309701   0.078548  -16.67   <2e-16 ***
white        0.304783   0.028693   10.62   <2e-16 ***
age          0.026423   0.001051   25.14   <2e-16 ***
male         0.324174   0.026216   12.37   <2e-16 ***
Democrat     0.439978   0.029769   14.78   <2e-16 ***
college      0.590905   0.035141   16.82   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


theta = -0.452(-0.825,0.278)
n = 15210  n.sel = 14834
total edf = 10
```

## Question 1e: Comparing Specifications

```
summary(logit_model1)
```

```

Call:
glm(formula = vote ~ white + age + male + Democrat + college,
    family = binomial, data = data1)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.768356   0.091111  -30.38   <2e-16 ***
white        0.548546   0.048802   11.24   <2e-16 ***
age          0.050474   0.001502   33.60   <2e-16 ***
male         0.592213   0.045699   12.96   <2e-16 ***
Democrat     0.864663   0.053551   16.15   <2e-16 ***
college      1.138215   0.049065   23.20   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 15139  on 15209  degrees of freedom
Residual deviance: 12525  on 15204  degrees of freedom
AIC: 12537

Number of Fisher Scoring iterations: 5
```

```r
summary(selectionmodel1)
```

```
--------------------------------------------
Tobit 2 model (sample selection model)
2-step Heckman / heckit estimation
15210 observations (376 censored and 14834 observed)
12 free parameters (df = 15199)
Probit selection equation:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.598875   0.075271   7.956  1.9e-15 ***
age         0.021775   0.001499  14.524  < 2e-16 ***
college     0.519222   0.047589  10.910  < 2e-16 ***
Outcome equation:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.6414035  0.0268734  23.868  < 2e-16 ***
white       0.0756957  0.0060049  12.606  < 2e-16 ***
age         0.0021870  0.0003772   5.797 6.87e-09 ***
male        0.0759778  0.0058483  12.991  < 2e-16 ***
Democrat    0.0960684  0.0062953  15.260  < 2e-16 ***
college     0.0432011  0.0140258   3.080  0.00207 **
Multiple R-Squared:0.1565,  Adjusted R-Squared:0.1561
   Error terms:
              Estimate Std. Error t value Pr(>|t|)
invMillsRatio  -1.8527        NaN     NaN      NaN
sigma           0.6922         NA      NA       NA
rho            -2.6766         NA      NA       NA
--------------------------------------------
```

## 2. Analyzing Count Data

### Question 2a and 2b: Estimating a Poisson Model

```r
# Load the dataset (contains data on presidential vetoes)
load("vetos.RData")

# Load stargazer for formatted regression output tables
require(stargazer)

# (1) Poisson Regression Model (Full Specification)
# Dependent variable: nover (number of vetoes)
# Includes multiple predictors: congressional experience, governor experience, presidential party,
# House majority, Senate majority, and presidential election proximity
poissonmod <- glm(
  nover ~ congexpr + govexpr + prespty + hmajor + smajor + presepct,
  data = vetos,
  family = "poisson"    # Poisson regression for count data
)
```

```r
# (2) Poisson Regression Model (Reduced Specification)
# A more parsimonious model including only key predictors
poissonmod2 <- glm(
  nover ~ congexpr + govexpr + prespty,
  data = vetos,
  family = "poisson"
)

# Display both models side-by-side in a formatted table
# Title indicates comparison between a full and reduced Poisson model
stargazer(
  poissonmod, poissonmod2,
  type = "text",
  title = "Poisson Regression Models Predicting the Number of Vetoes"
)
```

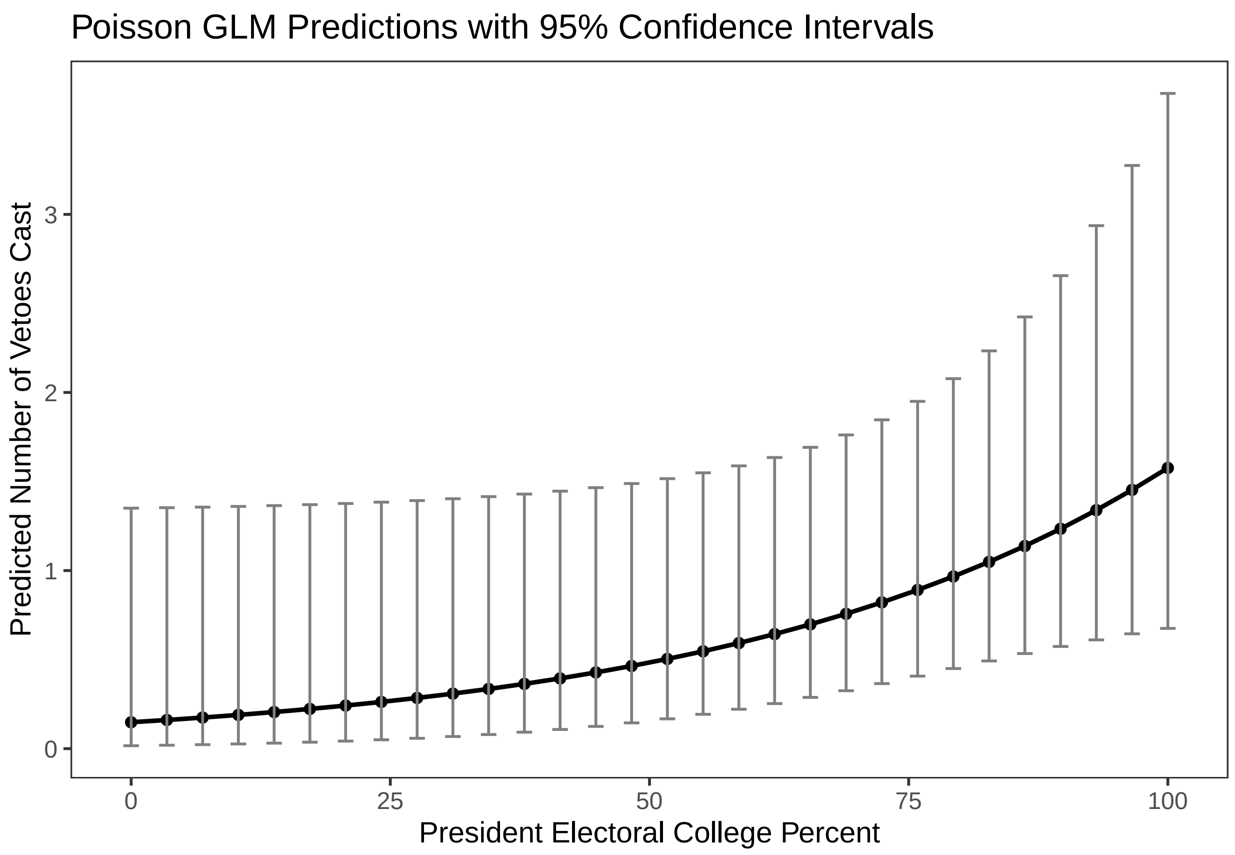
```
Poisson Regression Models Predicting the Number of Vetoes
==============================================
                      Dependent variable:
                  ----------------------------
                             nover
                      (1)            (2)
----------------------------------------------
congexpr            2.225***       1.845**
                    (0.778)        (0.739)

govexpr              1.584*         1.388*
                    (0.843)        (0.769)

prespty             -0.705*        -0.698**
                    (0.423)        (0.316)

hmajor               -1.025
                    (0.725)

smajor               0.074
                    (0.618)

presepct             0.024*
                    (0.013)

Constant             -2.069         -0.693
                    (1.318)        (0.707)

----------------------------------------------
Observations           26             26
Log Likelihood      -42.126        -46.167
Akaike Inf. Crit.    98.252        100.334
==============================================
Note:              *p<0.1; **p<0.05; ***p<0.01
```

## Question 2C: Methods of Substantively Interpreting Results

```r
# --- Incidence Rate Ratios (IRRs) with 95% CIs ------------------------------
# Do NOT overwrite the base 'coef' function name; store into 'coef_ci' instead.
# 'confint(poissonmod)' gives profile-likelihood CIs on the *log* scale.
# Exponentiating turns log-coefficients into IRRs and CIs onto the IRR scale.
coef_ci <- cbind(Estimate = coef(poissonmod),
                 confint(poissonmod))  # may profile; can be slow on big models
IRR <- exp(coef_ci)
IRR
```

```
             Estimate       2.5 %    97.5 %
(Intercept) 0.1262628 0.008146699  1.527394
congexpr    9.2530026 2.493316691 60.883153
govexpr     4.8735643 1.073307880 34.604845
prespty     0.4939730 0.207273235  1.103564
hmajor      0.3588653 0.084804432  1.502153
smajor      1.0763951 0.312913814  3.686567
presepct    1.0239137 0.999405144  1.050216
```

```r
# --- Predicted counts across 'presepct' with 95% CI (on response scale) -----
# IMPORTANT: For GLMs, standard errors are returned on the *link* scale.
# So: predict on 'link', build CIs on link, then transform via inverse link (exp)
# to the response (mean count) scale.

# Build a prediction grid: vary 'presepct' 0..100, hold others at their medians
newdata <- data.frame(
  presepct = seq(0, 100, length = 30),
  govexpr  = median(vetos$govexpr, na.rm = TRUE),
  prespty  = median(vetos$prespty, na.rm = TRUE),
  congexpr = median(vetos$congexpr, na.rm = TRUE),
  hmajor   = median(vetos$hmajor, na.rm = TRUE),
  smajor   = median(vetos$smajor, na.rm = TRUE)
)

# Predict on the LINK scale with SEs
pred_link <- predict(poissonmod, newdata, type = "link", se.fit = TRUE)

# Transform to RESPONSE scale (mean counts) and build 95% CIs correctly
predicdata <- within(newdata, {
  eta  = pred_link$fit                         # linear predictor
  se   = pred_link$se.fit                      # its standard error
  LL   = exp(eta - 1.96 * se)                  # lower CI on response scale
  UL   = exp(eta + 1.96 * se)                  # upper CI on response scale
  fit  = exp(eta)                              # mean predicted counts
})

# --- Plot: predicted mean counts vs. 'presepct' with 95% CI errorbars -------
# Points + line for the mean, error bars for 95% CI, clean theme
library(ggplot2)

ggplot(predicdata, aes(x = presepct, y = fit)) +
  geom_point() +
  geom_line(size = 0.8) +
  geom_errorbar(aes(ymin = LL, ymax = UL), width = 1.5, size = 0.5, color = "gray50") +
  theme_bw() +
  theme(panel.grid = element_blank()) +
  labs(x = "President Electoral College Percent",
       y = "Predicted Number of Vetoes Cast",
       title = "Poisson GLM Predictions with 95% Confidence Intervals")
```



### Question 2D: Testing for Overdispersion and Estimating a Negative Binomial Model

```r
 # --- Fit competing count models ---------------------------------------------
# Poisson regression (PRM)
poissonmod <- glm(
  nover ~ congexpr + govexpr + prespty + hmajor + smajor + presepct,
  data   = vetos,
  family = poisson(link = "log")
)

# Overdispersion diagnostic for Poisson (H0: equidispersion)
dispersiontest(poissonmod)
```

```
    Overdispersion test

data:  poissonmod
z = 1.693, p-value = 0.04523
alternative hypothesis: true dispersion is greater than 1
```

```
sample estimates:
dispersion
  1.377474
```

```r
# Negative Binomial regression (accounts for overdispersion via theta)
negbinomod <- glm.nb(
  nover ~ congexpr + govexpr + prespty + hmajor + smajor + presepct,
  data = vetos
)

# Likelihood-based comparison (NB nests Poisson when theta)
anova(poissonmod, negbinomod, test = "Chisq")
```

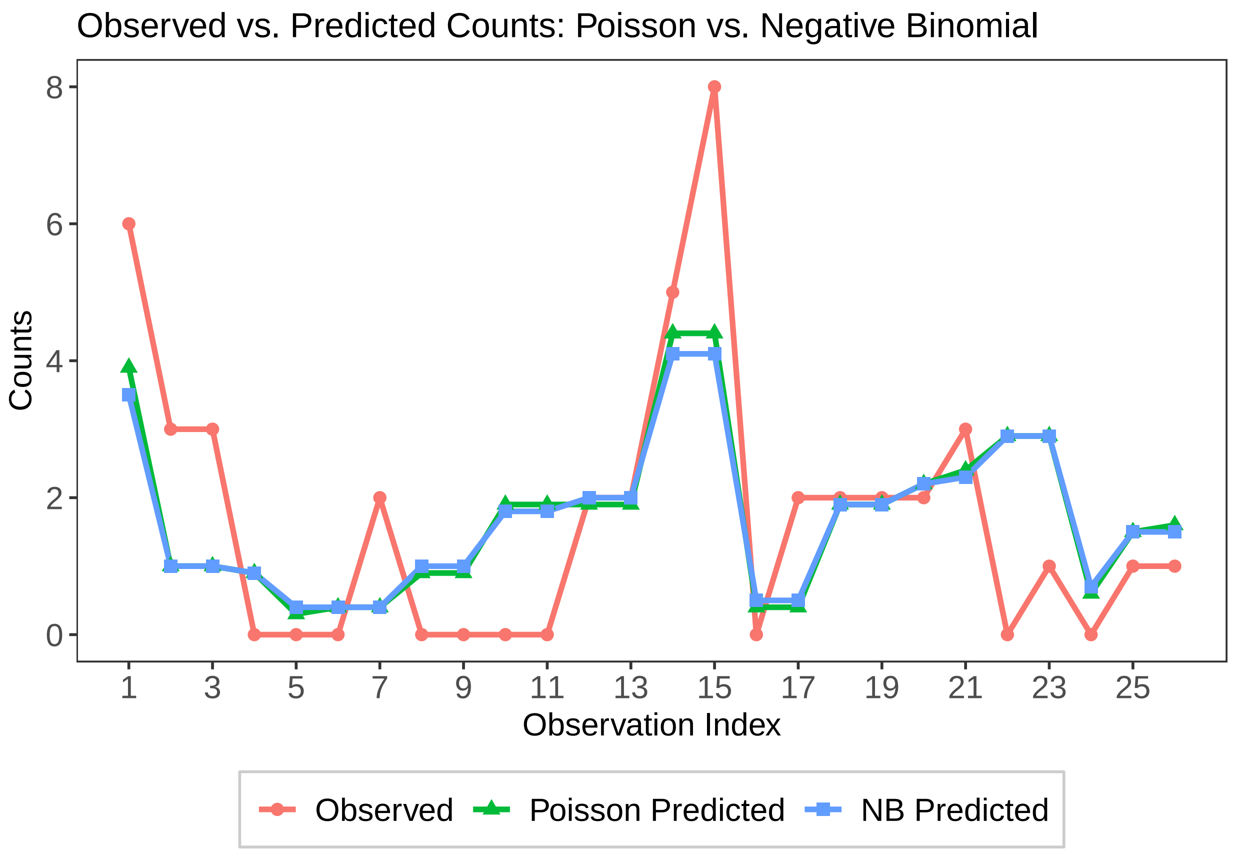
```
Analysis of Deviance Table

Model 1: nover ~ congexpr + govexpr + prespty + hmajor + smajor + presepct
Model 2: nover ~ congexpr + govexpr + prespty + hmajor + smajor + presepct
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        19     39.904
2        19     32.534  0   7.3704
```

```r
# --- Side-by-side model table -------------------------------------------------
stargazer(
  poissonmod, negbinomod,
  type  = "text",
  title = "Comparison of Poisson vs. Negative Binomial Regression for Veto Counts",
  dep.var.labels = "Number of Vetoes (nover)",
  covariate.labels = c("Congressional Experience",
                       "Governor Experience",
                       "President's Party (Indicator)",
                       "House Majority (President's Party)",
                       "Senate Majority (President's Party)",
                       "Pres. Electoral College %"),
  digits = 3
)
```

```
Comparison of Poisson vs. Negative Binomial Regression for Veto Counts
================================================================
                                       Dependent variable:
                                   ----------------------------
                                     Number of Vetoes (nover)
                                      Poisson       negative
                                                    binomial
                                        (1)           (2)
----------------------------------------------------------------
Congressional Experience              2.225***      2.103**
                                      (0.778)       (0.825)


Governor Experience                    1.584*        1.511*
                                      (0.843)       (0.906)


President's Party (Indicator)         -0.705*       -0.681
                                      (0.423)       (0.489)


House Majority (President's Party)     -1.025        -0.870
```

```
                                        (0.725)        (0.849)

Senate Majority (President's Party)      0.074          0.058
                                        (0.618)        (0.733)

Pres. Electoral College %                0.024*         0.020
                                        (0.013)        (0.015)

Constant                                 -2.069        -1.806
                                        (1.318)        (1.522)

----------------------------------------------------------------
Observations                               26             26
Log Likelihood                          -42.126        -42.779
theta                                               5.231 (7.358)
Akaike Inf. Crit.                        98.252         99.559
================================================================
Note:                                  *p<0.1; **p<0.05; ***p<0.01
```

```r
# --- Predicted values: Poisson vs. NB (Simple Version) -----------------------
# Create a simple data frame with observed and predicted counts
compare <- data.frame(
  id             = seq_len(nrow(vetos)),
  nover          = vetos$nover,
  prmpredicted  = round(fitted(poissonmod), 1),
  nbrmpredicted = round(fitted(negbinomod), 1)
)

# Convert to long format for ggplot (base R reshape)
compare_long <- reshape(
  compare,
  varying = c("nover", "prmpredicted", "nbrmpredicted"),
  v.names = "Counts",
  timevar = "model",
  times = c("Observed", "Poisson Predicted", "NB Predicted"),
  direction = "long"
)
rownames(compare_long) <- NULL

# Make model a factor to control legend order/labels
compare_long$model <- factor(
  compare_long$model,
  levels = c("Observed", "Poisson Predicted", "NB Predicted")
)

# --- Plot observed vs. fitted (PRM vs. NB) ----------------------------------
ggplot(compare_long, aes(x = id, y = Counts, color = model, group = model, shape = model)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  scale_x_continuous(breaks = seq(1, max(compare_long$id), 2)) +
  labs(x = "Observation Index", y = "Counts",
       title = "Observed vs. Predicted Counts: Poisson vs. Negative Binomial") +
  theme_bw() +
  theme(
    panel.grid         = element_blank(),
    legend.position    = "bottom",
    legend.background = element_rect(color = "grey80"),
    legend.title       = element_blank(),
    legend.text        = element_text(size = 12),
    axis.text          = element_text(size = 12),
    axis.title         = element_text(size = 12)
  )
```


Observed vs. Predicted Counts: Poisson vs. Negative Binomial
Counts
Observation Index
Observed
Poisson Predicted
NB Predicted