Thesis Code

Nicholas Johnson

2025-01-23

Loading Packages

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## 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
```

```r
library(tibble)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(ordinal)
```

```
## Warning: package 'ordinal' was built under R version 4.3.3
```

```
##
## Attaching package: 'ordinal'
##
## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
library(pscl)
```

```
## Warning: package 'pscl' was built under R version 4.3.3
```

```
## Classes and Methods for R originally developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University (2002-2015),
## by and under the direction of Simon Jackman.
## hurdle and zeroinfl functions by Achim Zeileis.
```

```r
library(nnet)
```

```
## Warning: package 'nnet' was built under R version 4.3.3
```

```r
library(ggplot2)
library(dplyr)
library(tidyr)
library(ggeffects)
```

```
## Warning: package 'ggeffects' was built under R version 4.3.3
```

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.3.3
```

```
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(effects)
```

```
## Warning: package 'effects' was built under R version 4.3.3
```

```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```r
library(officer)
```

```
## Warning: package 'officer' was built under R version 4.3.3
```

```r
library(flextable)
```

```
## Warning: package 'flextable' was built under R version 4.3.3
```

```
##
## Attaching package: 'flextable'
##
## The following object is masked from 'package:purrr':
##
##     compose
```

```r
library(nnet)
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.3.3
```

```r
library(knitr)
```

Reorganize NJ1-3 to be 2 rows: frame and value, using pivot longer/wider.

```r
load("C:/Users/nickj/OneDrive/Desktop/Thesis2025/NJ_Omnibus_1 (1).rdata")
data <- NJ_Omnibus_1
x <- pivot_longer(data, cols = c(NJ1, NJ2, NJ3),
               names_to = "Frame",
               values_to = "Importance",
               values_drop_na = TRUE)
```

Convert all necessary responses to their corresponding type (character/factor)

```r
x$Frame <- as.factor(x$Frame)
x$Frame <- relevel(x$Frame, ref = "NJ3")
```

Organizing Data for Ordinal Regression

```r
## Reclassifying variables
x$NJ6 <- as.factor(x$NJ6)
x$NJ11 <- as.factor(x$NJ11)
x$racethn <- as.factor(x$racethn)
x$sex <- as.factor(x$sex)
x$D5 <- as.numeric(x$D5)
```

```r
x$D8 <- as.factor(x$D8)
x$D1 <- as.factor(x$D1)
x$Importance <- factor(x$Importance, ordered = T)
x$NJ4 <- factor(x$NJ4, ordered = T)
x$NJ5 <- factor(x$NJ5, ordered = T)
```

Running an Ordinal Model to Predict the Importance Assigned to SSEC Reform

```r
## Running the ordinal logistic regression with pscl package
ordinal_model_imp <- clm(Importance ~ Frame + NJ6 + NJ7+ NJ8+ NJ9 +NJ10_1 +
           NJ11 + NJ12 + D1 + sex + agecat+ education + racethn + D5 + D8, data = x)

## Summarizing the model
summary(ordinal_model_imp)
```

```
## formula:
## Importance ~ Frame + NJ6 + NJ7 + NJ8 + NJ9 + NJ10_1 + NJ11 + NJ12 + D1 + sex + agecat + education + r
## data:    x
##
##  link  threshold nobs logLik   AIC     niter max.grad cond.H
##  logit flexible  1988 -2083.93 4233.86 5(0)  1.42e-08 9.8e+04
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## FrameNJ1   0.18122    0.11608   1.561  0.11848
## FrameNJ2   0.10576    0.11370   0.930  0.35231
## NJ62      -0.53681    0.12701  -4.227 2.37e-05 ***
## NJ63      -0.33670    0.12292  -2.739  0.00616 **
## NJ7       -0.06659    0.03886  -1.714  0.08659 .
## NJ8        0.03991    0.04440   0.899  0.36868
## NJ9        0.48946    0.04789  10.220  < 2e-16 ***
## NJ10_1     0.04507    0.03886   1.160  0.24612
## NJ112      0.04313    0.15851   0.272  0.78554
## NJ113     -0.75064    0.18793  -3.994 6.49e-05 ***
## NJ114     -0.12758    0.18154  -0.703  0.48220
## NJ115      0.37956    0.28250   1.344  0.17909
## NJ116      0.09296    0.19048   0.488  0.62553
## NJ12       0.24208    0.04071   5.947 2.73e-09 ***
## D12       -0.34861    0.77423  -0.450  0.65252
## D13       -0.29955    0.59504  -0.503  0.61468
## D14        0.17272    0.97644   0.177  0.85959
## D15       -0.84462    1.22388  -0.690  0.49012
## sex2       0.67668    0.76724   0.882  0.37779
## agecat     0.19481    0.01741  11.187  < 2e-16 ***
## education  0.05730    0.03736   1.534  0.12509
## racethn2  -0.10927    0.13368  -0.817  0.41369
## racethn3  -0.23989    0.15465  -1.551  0.12086
## racethn4  -0.42410    0.16577  -2.558  0.01052 *
## racethn5  -0.15573    0.15884  -0.980  0.32687
## D5         0.04289    0.01776   2.415  0.01575 *
## D82        0.14685    0.12174   1.206  0.22771
```

```
## D83        0.03963    0.12716   0.312  0.75532
## D84        0.05187    0.19831   0.262  0.79366
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##     Estimate Std. Error z value
## 1|2  -0.1729     0.3937  -0.439
## 2|3   1.4055     0.3797   3.702
## 3|4   2.5223     0.3814   6.613
## 4|5   3.6378     0.3862   9.420
## (15 observations deleted due to missingness)
```

```r
pR2(ordinal_model_imp)
```

```
## fitting null model for pseudo-r2
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -2083.9289566 -2344.6483631   521.4388129     0.1111977     0.2307145
##          r2CU
##     0.2548018
```

Rerun the regression as OLS for additional vetting

```r
lm_importance<- lm(as.numeric(x$Importance)~x$Frame + as.factor(x$NJ6) + x$NJ7+
x$NJ8+ x$NJ9 +x$NJ10_1 +as.factor(x$NJ11) + x$NJ12 + as.factor(x$D1)+
  as.factor(x$sex) + as.factor(x$agecat)+ x$education + as.factor(x$racethn) +
  x$D5 + as.factor(x$D8) + as.factor(x$vote))


summary(lm_importance)
```

```
##
## Call:
## lm(formula = as.numeric(x$Importance) ~ x$Frame + as.factor(x$NJ6) +
##     x$NJ7 + x$NJ8 + x$NJ9 + x$NJ10_1 + as.factor(x$NJ11) + x$NJ12 +
##     as.factor(x$D1) + as.factor(x$sex) + as.factor(x$agecat) +
##     x$education + as.factor(x$racethn) + x$D5 + as.factor(x$D8) +
##     as.factor(x$vote))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3009 -0.4404  0.2328  0.6189  2.1908
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             2.402539   0.177759  13.516  < 2e-16 ***
## x$FrameNJ1              0.042842   0.053893   0.795 0.426749
## x$FrameNJ2              0.027789   0.053963   0.515 0.606642
## as.factor(x$NJ6)2      -0.323216   0.064480  -5.013 5.86e-07 ***
## as.factor(x$NJ6)3      -0.152676   0.062038  -2.461 0.013942 *
```

```
## x$NJ7                 -0.035823   0.017780  -2.015 0.044070 *
## x$NJ8                  0.043675   0.020804   2.099 0.035915 *
## x$NJ9                  0.238540   0.021839  10.923  < 2e-16 ***
## x$NJ10_1              -0.009686   0.017807  -0.544 0.586555
## as.factor(x$NJ11)2     0.062962   0.071741   0.878 0.380248
## as.factor(x$NJ11)3    -0.326815   0.090787  -3.600 0.000326 ***
## as.factor(x$NJ11)4    -0.001500   0.084988  -0.018 0.985924
## as.factor(x$NJ11)5     0.261571   0.131112   1.995 0.046180 *
## as.factor(x$NJ11)6     0.131409   0.088701   1.481 0.138642
## x$NJ12                 0.105901   0.018632   5.684 1.52e-08 ***
## as.factor(x$D1)2      -0.342994   0.384207  -0.893 0.372111
## as.factor(x$D1)3      -0.291576   0.301865  -0.966 0.334206
## as.factor(x$D1)4       0.017324   0.433599   0.040 0.968133
## as.factor(x$D1)5      -0.511273   0.580206  -0.881 0.378323
## as.factor(x$sex)2      0.505444   0.381310   1.326 0.185145
## as.factor(x$agecat)2   0.181014   0.097828   1.850 0.064416 .
## as.factor(x$agecat)3   0.229587   0.093238   2.462 0.013888 *
## as.factor(x$agecat)4   0.357227   0.095601   3.737 0.000192 ***
## as.factor(x$agecat)5   0.561531   0.093262   6.021 2.07e-09 ***
## as.factor(x$agecat)6   0.404102   0.107079   3.774 0.000166 ***
## as.factor(x$agecat)7   0.623912   0.108289   5.762 9.67e-09 ***
## as.factor(x$agecat)8   0.643855   0.106462   6.048 1.76e-09 ***
## as.factor(x$agecat)9   0.843152   0.111274   7.577 5.42e-14 ***
## as.factor(x$agecat)10  0.808802   0.107865   7.498 9.77e-14 ***
## as.factor(x$agecat)11  0.819330   0.098518   8.317  < 2e-16 ***
## x$education            0.018592   0.017858   1.041 0.297962
## as.factor(x$racethn)2 -0.043590   0.063272  -0.689 0.490951
## as.factor(x$racethn)3 -0.113577   0.075155  -1.511 0.130892
## as.factor(x$racethn)4 -0.097099   0.082932  -1.171 0.241809
## as.factor(x$racethn)5 -0.060838   0.076646  -0.794 0.427430
## x$D5                   0.024683   0.008477   2.912 0.003637 **
## as.factor(x$D8)2       0.082583   0.072293   1.142 0.253451
## as.factor(x$D8)3       0.078235   0.065507   1.194 0.232510
## as.factor(x$D8)4       0.141832   0.103925   1.365 0.172486
## as.factor(x$vote)2     0.011943   0.069860   0.171 0.864275
## as.factor(x$vote)3     0.094415   0.148157   0.637 0.524027
## as.factor(x$vote)4    -0.198333   0.068007  -2.916 0.003582 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9695 on 1946 degrees of freedom
##   (15 observations deleted due to missingness)
## Multiple R-squared:  0.2316, Adjusted R-squared:  0.2154
## F-statistic:  14.3 on 41 and 1946 DF,  p-value: < 2.2e-16
```

```
## OLS confirms nonsignificance of frames, significance of other predictors.
```

Trump Ordinal Regression Interpreting Incomes as Cutoffs

```
## Defining the cutoff point for incomes above and below 34,00-44,000 USD
## As described by Trump, those above this threshold are eligible to recieve
## Tax cuts.
```

```r
## Creating the cutoff
x$D5 <- as.numeric(x$D5)
trump <- x |> filter(D5<4 | D5>5) |> mutate(cutoff = ifelse(D5 > 5, 1, 0)) # high values get a 1
trump$cutoff <- as.factor(trump$cutoff)
## Running the model
ordinal_model_t_cutoff <- clm(NJ4 ~ Frame*cutoff + NJ6 + NJ7+ NJ8+ NJ9 +NJ10_1 +
           NJ11 + NJ12 + D1 + sex + agecat+ education + racethn + D8, data = trump)

## Summarizing the model
summary(ordinal_model_t_cutoff)
```

```
## formula:
## NJ4 ~ Frame * cutoff + NJ6 + NJ7 + NJ8 + NJ9 + NJ10_1 + NJ11 + NJ12 + D1 + sex + agecat + education +
## data:    trump
##
##  link  threshold nobs logLik   AIC     niter max.grad cond.H
##  logit flexible  1581 -2293.67 4657.35 5(0)  1.75e-09 7.7e+04
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## FrameNJ1            0.04919    0.17435   0.282 0.777832
## FrameNJ2           -0.13452    0.17531  -0.767 0.442882
## cutoff1             0.24276    0.17200   1.411 0.158130
## NJ62               -0.16912    0.13377  -1.264 0.206144
## NJ63                0.09553    0.12267   0.779 0.436123
## NJ7                 0.09425    0.03813   2.472 0.013435 *
## NJ8                 0.07068    0.04447   1.590 0.111915
## NJ9                 0.14516    0.04705   3.085 0.002033 **
## NJ10_1              0.05934    0.03818   1.554 0.120149
## NJ112               0.24481    0.14858   1.648 0.099418 .
## NJ113              -0.34793    0.18464  -1.884 0.059513 .
## NJ114               0.02034    0.17354   0.117 0.906703
## NJ115               0.05301    0.26094   0.203 0.839020
## NJ116               0.16314    0.18249   0.894 0.371330
## NJ12                0.13250    0.03902   3.396 0.000684 ***
## D12                 0.26398    0.73830   0.358 0.720680
## D13                -0.05092    0.59380  -0.086 0.931658
## D14                 0.36269    0.96567   0.376 0.707228
## D15                -0.46392    0.94290  -0.492 0.622710
## sex2               -0.18972    0.73099  -0.260 0.795219
## agecat              0.09545    0.01664   5.736 9.72e-09 ***
## education           0.07088    0.03637   1.949 0.051309 .
## racethn2           -0.22164    0.13098  -1.692 0.090624 .
## racethn3           -0.20313    0.15861  -1.281 0.200316
## racethn4           -0.42128    0.16684  -2.525 0.011567 *
## racethn5           -0.20473    0.15530  -1.318 0.187410
## D82                -0.16729    0.11910  -1.405 0.160138
## D83                -0.28581    0.12610  -2.267 0.023420 *
## D84                -0.15576    0.19605  -0.795 0.426894
## FrameNJ1:cutoff1   -0.20183    0.22934  -0.880 0.378847
## FrameNJ2:cutoff1    0.28862    0.22973   1.256 0.208992
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Threshold coefficients:
##     Estimate Std. Error z value
## 1|2  -0.4303     0.3795  -1.134
## 2|3   0.6605     0.3756   1.758
## 3|4   2.1363     0.3789   5.639
## 4|5   3.2188     0.3836   8.392
## (12 observations deleted due to missingness)
```

```r
pR2(ordinal_model_t_cutoff)
```

```
## fitting null model for pseudo-r2
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -2.293675e+03 -2.384937e+03  1.825241e+02  3.826602e-02  1.090335e-01
##          r2CU
##  1.146452e-01
```

```r
# Delineating between groups at incomes who are guaranteed to benefit from Trump's
# policy, and those who are guaranteed to not benefit from Trumps policy is
# significant. However, individualist and collectivist frames have no impacts
# on these groups.
```

Running an Ordinal Model to Predict Support for Trump's SSEC Policy (no cutoff)

```r
## Running the ordinal logistic regression with pscl package
ordinal_model_t <- clm(NJ4 ~ Frame + NJ6 + NJ7+ NJ8+ NJ9 +NJ10_1 +
           NJ11 + NJ12 + D1 + sex + agecat+ education + racethn + D5 + D8, data = x)

## Summarizing the model
summary(ordinal_model_t)
```

```
## formula:
## NJ4 ~ Frame + NJ6 + NJ7 + NJ8 + NJ9 + NJ10_1 + NJ11 + NJ12 + D1 + sex + agecat + education + racethn
## data:    x
##
##  link  threshold nobs logLik   AIC     niter max.grad cond.H
##  logit flexible  1988 -2886.84 5839.68 5(0)  9.57e-10 1.2e+05
##
## Coefficients:
##            Estimate Std. Error z value Pr(>|z|)
## FrameNJ1    0.02173    0.09981   0.218   0.8276
## FrameNJ2    0.11156    0.09981   1.118   0.2637
## NJ62       -0.16635    0.11727  -1.419   0.1560
## NJ63        0.13172    0.11002   1.197   0.2312
## NJ7         0.07403    0.03372   2.196   0.0281 *
## NJ8         0.06894    0.04021   1.714   0.0865 .
## NJ9         0.16425    0.04149   3.959 7.53e-05 ***
## NJ10_1      0.05927    0.03366   1.761   0.0783 .
```

```
## NJ112      0.13370    0.13538   0.988   0.3234
## NJ113     -0.36651    0.16745  -2.189   0.0286 *
## NJ114      0.02112    0.15870   0.133   0.8941
## NJ115      0.07644    0.23929   0.319   0.7494
## NJ116      0.09303    0.16591   0.561   0.5750
## NJ12       0.14722    0.03515   4.189 2.81e-05 ***
## D12        0.23547    0.69979   0.336   0.7365
## D13        0.34751    0.53698   0.647   0.5175
## D14       -0.04198    0.90180  -0.047   0.9629
## D15       -0.45664    0.95217  -0.480   0.6315
## sex2      -0.16732    0.69426  -0.241   0.8096
## agecat     0.10990    0.01477   7.441 1.00e-13 ***
## education  0.05028    0.03274   1.536   0.1246
## racethn2  -0.20509    0.11615  -1.766   0.0775 .
## racethn3  -0.24728    0.13939  -1.774   0.0761 .
## racethn4  -0.36657    0.14976  -2.448   0.0144 *
## racethn5  -0.12732    0.13874  -0.918   0.3588
## D5         0.03877    0.01562   2.482   0.0131 *
## D82       -0.25926    0.10516  -2.465   0.0137 *
## D83       -0.26251    0.11166  -2.351   0.0187 *
## D84       -0.24638    0.17664  -1.395   0.1631
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##     Estimate Std. Error z value
## 1|2  -0.3119     0.3398  -0.918
## 2|3   0.8018     0.3361   2.385
## 3|4   2.2393     0.3391   6.603
## 4|5   3.2965     0.3437   9.592
## (15 observations deleted due to missingness)
```

```r
pR2(ordinal_model_t)
```

```
## fitting null model for pseudo-r2
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -2.886842e+03 -3.000428e+03  2.271725e+02  3.785668e-02  1.079846e-01
##          r2CU
##  1.135332e-01
```

```r
## Even ignoring the cutoff, people at higher incomes support the policy more than
## People at lower incomes.
```

Re-running this model with an interaction term

```r
## The goal is to investigate whether income and frame have any interaction
ordinal_model_t_int <- clm(NJ4 ~ Frame*D5 + NJ6 + NJ7+ NJ8+ NJ9 +NJ10_1 +
           NJ11 + NJ12 + D1 + sex + agecat+ education + racethn + D8, data = x)

## Summarizing the model
summary(ordinal_model_t_int)
```

```
## formula:
## NJ4 ~ Frame * D5 + NJ6 + NJ7 + NJ8 + NJ9 + NJ10_1 + NJ11 + NJ12 + D1 + sex + agecat + education + rac
## data:    x
##
##  link  threshold nobs logLik   AIC     niter max.grad cond.H
##  logit flexible  1988 -2885.57 5841.14 5(0)  9.83e-10 1.3e+05
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## FrameNJ1      0.20631    0.20240   1.019 0.308048
## FrameNJ2      0.01507    0.20388   0.074 0.941080
## D5            0.04505    0.02419   1.863 0.062500 .
## NJ62         -0.17275    0.11733  -1.472 0.140924
## NJ63          0.12634    0.11018   1.147 0.251514
## NJ7           0.07632    0.03374   2.262 0.023695 *
## NJ8           0.06456    0.04029   1.602 0.109071
## NJ9           0.16054    0.04159   3.860 0.000113 ***
## NJ10_1        0.05765    0.03370   1.711 0.087137 .
## NJ112         0.13487    0.13547   0.996 0.319480
## NJ113        -0.37447    0.16753  -2.235 0.025401 *
## NJ114         0.02205    0.15873   0.139 0.889509
## NJ115         0.07461    0.23933   0.312 0.755236
## NJ116         0.09310    0.16608   0.561 0.575086
## NJ12          0.14678    0.03524   4.165 3.11e-05 ***
## D12           0.25379    0.69773   0.364 0.716052
## D13           0.34721    0.53654   0.647 0.517552
## D14          -0.09276    0.89866  -0.103 0.917787
## D15          -0.48693    0.94536  -0.515 0.606505
## sex2         -0.17428    0.69205  -0.252 0.801172
## agecat        0.10947    0.01480   7.395 1.42e-13 ***
## education     0.05087    0.03275   1.554 0.120275
## racethn2     -0.20317    0.11616  -1.749 0.080275 .
## racethn3     -0.24079    0.13959  -1.725 0.084537 .
## racethn4     -0.36153    0.15000  -2.410 0.015942 *
## racethn5     -0.11971    0.13876  -0.863 0.388308
## D82          -0.26772    0.10534  -2.542 0.011036 *
## D83          -0.26511    0.11170  -2.373 0.017626 *
## D84          -0.23718    0.17659  -1.343 0.179230
## FrameNJ1:D5 -0.03489    0.03311  -1.054 0.292003
## FrameNJ2:D5  0.01820    0.03354   0.542 0.587490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##     Estimate Std. Error z value
## 1|2  -0.3015     0.3519  -0.857
## 2|3   0.8121     0.3484   2.331
## 3|4   2.2509     0.3515   6.405
## 4|5   3.3098     0.3558   9.302
## (15 observations deleted due to missingness)
```

```
## Does income influence how people percieve themselves in relation to the policy
## And subsequently influence the significance of the frames?
```

```
## The data suggests no - not with these frames.
```

Rerun Regression as OLS for additional vetting

```
# Trump Policy Support
lm_trump<- lm(as.numeric(x$NJ4)~x$Frame + as.factor(x$NJ6) + x$NJ7+ x$NJ8+ x$NJ9 +x$NJ10_1 +
           as.factor(x$NJ11) + x$NJ12 + as.factor(x$D1)+ as.factor(x$sex) + as.factor(x$agecat)
         + x$education + as.factor(x$racethn) + x$D5 + as.factor(x$D8)
         + as.factor(x$vote))

summary(lm_trump)
```

```
##
## Call:
## lm(formula = as.numeric(x$NJ4) ~ x$Frame + as.factor(x$NJ6) +
##     x$NJ7 + x$NJ8 + x$NJ9 + x$NJ10_1 + as.factor(x$NJ11) + x$NJ12 +
##     as.factor(x$D1) + as.factor(x$sex) + as.factor(x$agecat) +
##     x$education + as.factor(x$racethn) + x$D5 + as.factor(x$D8) +
##     as.factor(x$vote))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4930 -0.7342  0.1021  0.9729  2.3357
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.28568    0.21782  10.494  < 2e-16 ***
## x$FrameNJ1             0.01336    0.06604   0.202 0.839714
## x$FrameNJ2             0.06589    0.06612   0.997 0.319122
## as.factor(x$NJ6)2     -0.08567    0.07901  -1.084 0.278400
## as.factor(x$NJ6)3      0.15484    0.07602   2.037 0.041795 *
## x$NJ7                  0.04337    0.02179   1.991 0.046639 *
## x$NJ8                  0.04166    0.02549   1.634 0.102394
## x$NJ9                  0.09612    0.02676   3.592 0.000336 ***
## x$NJ10_1               0.02181    0.02182   1.000 0.317645
## as.factor(x$NJ11)2     0.10818    0.08791   1.231 0.218617
## as.factor(x$NJ11)3    -0.21328    0.11125  -1.917 0.055356 .
## as.factor(x$NJ11)4     0.03964    0.10414   0.381 0.703514
## as.factor(x$NJ11)5     0.07707    0.16066   0.480 0.631506
## as.factor(x$NJ11)6     0.09332    0.10869   0.859 0.390689
## x$NJ12                 0.07930    0.02283   3.473 0.000525 ***
## as.factor(x$D1)2       0.07827    0.47079   0.166 0.867974
## as.factor(x$D1)3       0.22976    0.36989   0.621 0.534562
## as.factor(x$D1)4      -0.10748    0.53131  -0.202 0.839713
## as.factor(x$D1)5      -0.40500    0.71096  -0.570 0.568975
## as.factor(x$sex)2     -0.01685    0.46724  -0.036 0.971235
## as.factor(x$agecat)2  -0.07079    0.11987  -0.591 0.554918
## as.factor(x$agecat)3   0.17125    0.11425   1.499 0.134059
## as.factor(x$agecat)4   0.10497    0.11714   0.896 0.370345
## as.factor(x$agecat)5   0.24807    0.11428   2.171 0.030069 *
## as.factor(x$agecat)6   0.34999    0.13121   2.667 0.007708 **
```

```
## as.factor(x$agecat)7   0.27696    0.13269   2.087 0.036997 *
## as.factor(x$agecat)8   0.40670    0.13045   3.118 0.001850 **
## as.factor(x$agecat)9   0.56440    0.13635   4.139 3.63e-05 ***
## as.factor(x$agecat)10  0.65682    0.13217   4.969 7.30e-07 ***
## as.factor(x$agecat)11  0.60572    0.12072   5.018 5.71e-07 ***
## x$education            0.03988    0.02188   1.822 0.068577 .
## as.factor(x$racethn)2 -0.11828    0.07753  -1.526 0.127284
## as.factor(x$racethn)3 -0.17151    0.09209  -1.862 0.062698 .
## as.factor(x$racethn)4 -0.20457    0.10162  -2.013 0.044243 *
## as.factor(x$racethn)5 -0.05297    0.09392  -0.564 0.572830
## x$D5                   0.02141    0.01039   2.061 0.039417 *
## as.factor(x$D8)2       0.10717    0.08858   1.210 0.226503
## as.factor(x$D8)3      -0.00288    0.08027  -0.036 0.971387
## as.factor(x$D8)4       0.04281    0.12735   0.336 0.736781
## as.factor(x$vote)2    -0.37990    0.08560  -4.438 9.59e-06 ***
## as.factor(x$vote)3    -0.32894    0.18155  -1.812 0.070157 .
## as.factor(x$vote)4    -0.28561    0.08333  -3.427 0.000622 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.188 on 1946 degrees of freedom
##   (15 observations deleted due to missingness)
## Multiple R-squared:  0.1049, Adjusted R-squared:  0.08604
## F-statistic: 5.562 on 41 and 1946 DF,  p-value: < 2.2e-16
```

```
## OLS results confirm non-significance of frames, significance of other predictors.
```

Running an Ordinal Model to Predict Support for Harris' SSEC Policy

```
## Running the model
ordinal_model_h <- clm(NJ5 ~ Frame + NJ6 + NJ7+ NJ8+ NJ9 +NJ10_1 +
          NJ11 + NJ12 + D1 + sex + agecat+ education + racethn + D5 + D8, data = x)

## Summarizing the model
summary(ordinal_model_h)
```

```
## formula:
## NJ5 ~ Frame + NJ6 + NJ7 + NJ8 + NJ9 + NJ10_1 + NJ11 + NJ12 + D1 + sex + agecat + education + racethn
## data:    x
##
##  link  threshold nobs logLik   AIC     niter max.grad cond.H
##  logit flexible  1988 -2950.34 5966.68 5(0)  3.69e-11 1.1e+05
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## FrameNJ1   0.05102    0.09918   0.514 0.606981
## FrameNJ2  -0.08450    0.09994  -0.846 0.397796
## NJ62      -0.21030    0.11737  -1.792 0.073160 .
## NJ63      -0.06958    0.11049  -0.630 0.528873
## NJ7       -0.04855    0.03399  -1.428 0.153161
## NJ8        0.14880    0.03971   3.747 0.000179 ***
```

```
## NJ9         0.30532    0.04164   7.333 2.25e-13 ***
## NJ10_1     -0.03118    0.03316  -0.940 0.347207
## NJ112       0.09212    0.13332   0.691 0.489592
## NJ113      -0.22448    0.16891  -1.329 0.183847
## NJ114       0.03407    0.15723   0.217 0.828439
## NJ115       0.03206    0.24303   0.132 0.895063
## NJ116       0.07867    0.16283   0.483 0.628997
## NJ12        0.06110    0.03499   1.746 0.080777 .
## D12        -0.79974    0.69983  -1.143 0.253138
## D13        -0.76149    0.50631  -1.504 0.132584
## D14        -0.76444    0.85194  -0.897 0.369565
## D15         0.26237    1.06345   0.247 0.805130
## sex2        0.55613    0.69420   0.801 0.423063
## agecat      0.04696    0.01463   3.210 0.001330 **
## education   0.11987    0.03237   3.703 0.000213 ***
## racethn2   -0.58164    0.11596  -5.016 5.28e-07 ***
## racethn3   -0.28606    0.13929  -2.054 0.039999 *
## racethn4   -0.50307    0.14760  -3.408 0.000654 ***
## racethn5   -0.29604    0.14142  -2.093 0.036317 *
## D5          0.03375    0.01542   2.190 0.028554 *
## D82         0.65396    0.10532   6.209 5.32e-10 ***
## D83         0.49824    0.10944   4.553 5.29e-06 ***
## D84         0.49642    0.18001   2.758 0.005822 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##     Estimate Std. Error z value
## 1|2  -0.1255     0.3308  -0.380
## 2|3   0.8660     0.3290   2.632
## 3|4   2.2074     0.3318   6.652
## 4|5   3.3198     0.3360   9.881
## (15 observations deleted due to missingness)
```

```
pR2(ordinal_model_h)
```

```
## fitting null model for pseudo-r2
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -2.950338e+03 -3.057467e+03  2.142568e+02  3.503828e-02  1.021704e-01
##          r2CU
##  1.071134e-01
```

Rerun regression as OLS for additional vetting

```
lm3<- lm(as.numeric(x$NJ5)~x$Frame + as.factor(x$NJ6) + x$NJ7+ x$NJ8+ x$NJ9 +x$NJ10_1 +
           as.factor(x$NJ11) + x$NJ12 + as.factor(x$D1)+ as.factor(x$sex) + as.factor(x$agecat)
         + x$education + as.factor(x$racethn) + x$D5 + as.factor(x$D8)
         + as.factor(x$vote))

summary(lm3)
```

```
##
## Call:
## lm(formula = as.numeric(x$NJ5) ~ x$Frame + as.factor(x$NJ6) +
##     x$NJ7 + x$NJ8 + x$NJ9 + x$NJ10_1 + as.factor(x$NJ11) + x$NJ12 +
##     as.factor(x$D1) + as.factor(x$sex) + as.factor(x$agecat) +
##     x$education + as.factor(x$racethn) + x$D5 + as.factor(x$D8) +
##     as.factor(x$vote))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6969 -0.7353  0.1049  0.9325  2.8028
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.0763875  0.2239178   9.273  < 2e-16 ***
## x$FrameNJ1             0.0497524  0.0678880   0.733  0.46373
## x$FrameNJ2            -0.0489778  0.0679757  -0.721  0.47129
## as.factor(x$NJ6)2     -0.0962175  0.0812243  -1.185  0.23632
## as.factor(x$NJ6)3      0.0172877  0.0781479   0.221  0.82495
## x$NJ7                 -0.0240419  0.0223976  -1.073  0.28322
## x$NJ8                  0.0949121  0.0262063   3.622  0.00030 ***
## x$NJ9                  0.1986935  0.0275099   7.223 7.28e-13 ***
## x$NJ10_1              -0.0322599  0.0224306  -1.438  0.15054
## as.factor(x$NJ11)2     0.0928698  0.0903699   1.028  0.30424
## as.factor(x$NJ11)3    -0.1489517  0.1143617  -1.302  0.19291
## as.factor(x$NJ11)4     0.0679268  0.1070575   0.634  0.52584
## as.factor(x$NJ11)5     0.0005109  0.1651586   0.003  0.99753
## as.factor(x$NJ11)6     0.0932902  0.1117346   0.835  0.40386
## x$NJ12                 0.0327501  0.0234705   1.395  0.16306
## as.factor(x$D1)2      -0.4029955  0.4839754  -0.833  0.40513
## as.factor(x$D1)3      -0.3944836  0.3802511  -1.037  0.29966
## as.factor(x$D1)4      -0.3974614  0.5461934  -0.728  0.46689
## as.factor(x$D1)5       0.1813088  0.7308703   0.248  0.80410
## as.factor(x$sex)2      0.2512493  0.4803256   0.523  0.60098
## as.factor(x$agecat)2  -0.1446073  0.1232309  -1.173  0.24075
## as.factor(x$agecat)3   0.0173984  0.1174490   0.148  0.88225
## as.factor(x$agecat)4  -0.0377165  0.1204258  -0.313  0.75417
## as.factor(x$agecat)5  -0.0067317  0.1174798  -0.057  0.95431
## as.factor(x$agecat)6  -0.0665317  0.1348847  -0.493  0.62189
## as.factor(x$agecat)7   0.1887092  0.1364089   1.383  0.16670
## as.factor(x$agecat)8  -0.0213832  0.1341068  -0.159  0.87333
## as.factor(x$agecat)9   0.0004878  0.1401685   0.003  0.99722
## as.factor(x$agecat)10  0.2118438  0.1358752   1.559  0.11913
## as.factor(x$agecat)11  0.3292499  0.1241009   2.653  0.00804 **
## x$education            0.0709963  0.0224953   3.156  0.00162 **
## as.factor(x$racethn)2 -0.3715626  0.0797023  -4.662 3.35e-06 ***
## as.factor(x$racethn)3 -0.1594832  0.0946712  -1.685  0.09223 .
## as.factor(x$racethn)4 -0.2907874  0.1044668  -2.784  0.00543 **
## as.factor(x$racethn)5 -0.1684506  0.0965484  -1.745  0.08119 .
## x$D5                   0.0220430  0.0106789   2.064  0.03913 *
## as.factor(x$D8)2       0.1639958  0.0910652   1.801  0.07188 .
## as.factor(x$D8)3       0.2209650  0.0825177   2.678  0.00747 **
## as.factor(x$D8)4       0.2017712  0.1309113   1.541  0.12341
## as.factor(x$vote)2     0.3595313  0.0880002   4.086 4.58e-05 ***
```

```
## as.factor(x$vote)3     0.2327352  0.1866289   1.247  0.21253
## as.factor(x$vote)4     0.1075249  0.0856672   1.255  0.20958
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.221 on 1946 degrees of freedom
##   (15 observations deleted due to missingness)
## Multiple R-squared:  0.1052, Adjusted R-squared:  0.08633
## F-statistic: 5.579 on 41 and 1946 DF,  p-value: < 2.2e-16
```

```
## OLS confirms non significance of frames, significance of other predictors
```

Re-running this model with an interaction term

```
## The goal is to investigate whether income and frame have any interaction
ordinal_model_h_int <- clm(NJ5 ~ Frame*D5 + NJ6 + NJ7+ NJ8+ NJ9 +NJ10_1 +
           NJ11 + NJ12 + D1 + sex + agecat+ education + racethn + D8, data = x)

## Summarizing the model
summary(ordinal_model_h_int)
```

```
## formula:
## NJ5 ~ Frame * D5 + NJ6 + NJ7 + NJ8 + NJ9 + NJ10_1 + NJ11 + NJ12 + D1 + sex + agecat + education + rac
## data:    x
##
##  link  threshold nobs logLik   AIC     niter max.grad cond.H
##  logit flexible  1988 -2950.33 5970.66 5(0)  3.70e-11 1.2e+05
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## FrameNJ1     0.0701161  0.2008293   0.349 0.726990
## FrameNJ2    -0.0809905  0.2045576  -0.396 0.692157
## D5           0.0351992  0.0241900   1.455 0.145637
## NJ62        -0.2107929  0.1174688  -1.794 0.072740 .
## NJ63        -0.0700318  0.1105639  -0.633 0.526469
## NJ7         -0.0484811  0.0340434  -1.424 0.154418
## NJ8          0.1485105  0.0397859   3.733 0.000189 ***
## NJ9          0.3049640  0.0417516   7.304 2.79e-13 ***
## NJ10_1      -0.0311502  0.0332180  -0.938 0.348372
## NJ112        0.0920054  0.1333952   0.690 0.490370
## NJ113       -0.2249608  0.1689666  -1.331 0.183060
## NJ114        0.0340750  0.1572394   0.217 0.828436
## NJ115        0.0317854  0.2431598   0.131 0.895998
## NJ116        0.0792066  0.1629429   0.486 0.626896
## NJ12         0.0612169  0.0351281   1.743 0.081390 .
## D12         -0.7993420  0.7000996  -1.142 0.253556
## D13         -0.7616093  0.5063346  -1.504 0.132540
## D14         -0.7696574  0.8536185  -0.902 0.367248
## D15          0.2592797  1.0631845   0.244 0.807331
## sex2         0.5564298  0.6943167   0.801 0.422896
## agecat       0.0468547  0.0146685   3.194 0.001402 **
```

```
## education    0.1199926  0.0323914   3.704 0.000212 ***
## racethn2    -0.5816878  0.1159724  -5.016 5.28e-07 ***
## racethn3    -0.2853664  0.1394356  -2.047 0.040699 *
## racethn4    -0.5026637  0.1477089  -3.403 0.000666 ***
## racethn5    -0.2957435  0.1414590  -2.091 0.036558 *
## D82          0.6533313  0.1054712   6.194 5.85e-10 ***
## D83          0.4981565  0.1094500   4.551 5.33e-06 ***
## D84          0.4964999  0.1801651   2.756 0.005855 **
## FrameNJ1:D5 -0.0036252  0.0331485  -0.109 0.912915
## FrameNJ2:D5 -0.0006939  0.0336140  -0.021 0.983529
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##     Estimate Std. Error z value
## 1|2  -0.1194     0.3432  -0.348
## 2|3   0.8722     0.3415   2.554
## 3|4   2.2135     0.3442   6.431
## 4|5   3.3260     0.3483   9.550
## (15 observations deleted due to missingness)
```

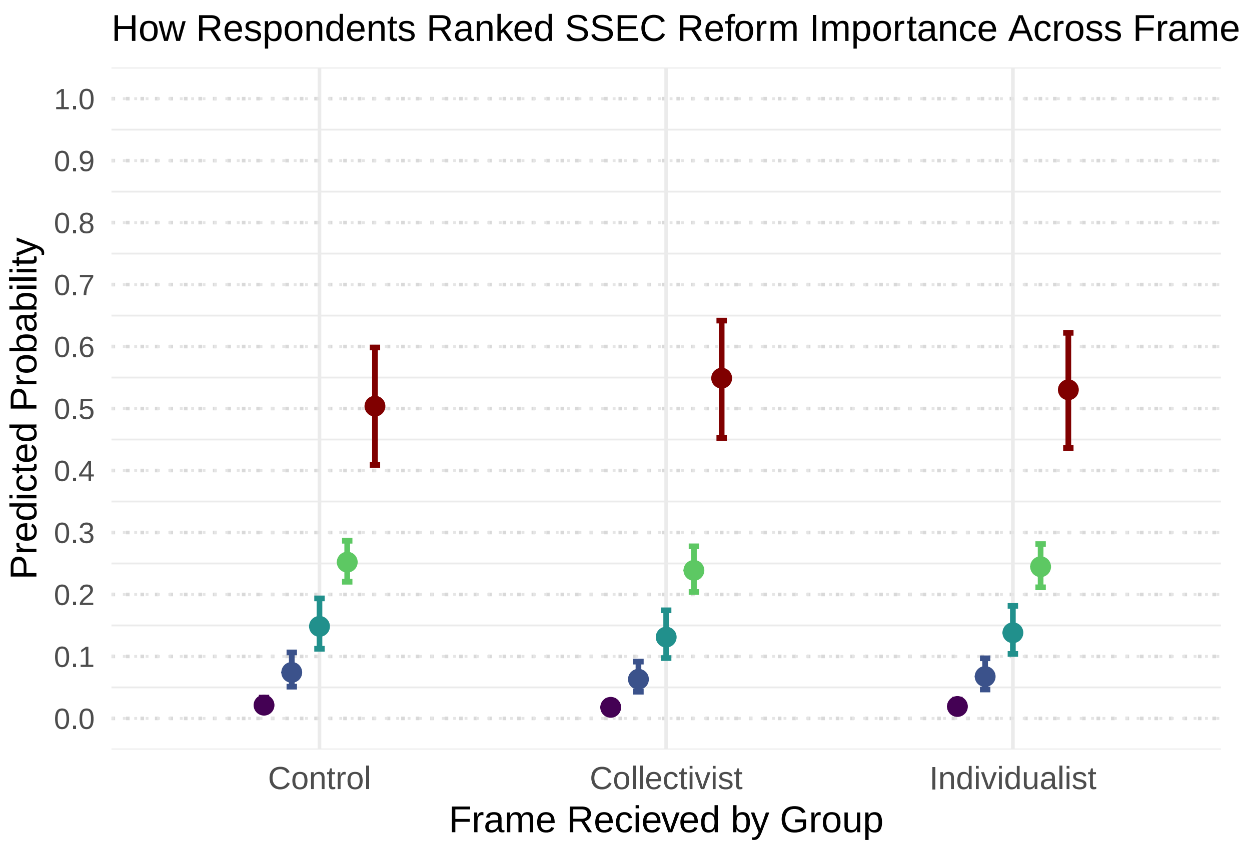
```
## Does income influence how people percieve themselves in relation to the policy
## And subsequently influence the significance of the frames?

## The data suggests no - not with these frames.
```

Plotting Regression Outputs

Confidence Intervals for Ordinal Regression on SSEC Reform Importance by Frame

```
# Compute predicted probabilities for Frame
frame_effect <- ggpredict(ordinal_model_imp, terms = "Frame")

# Convert to data.frame
frame_effect <- as.data.frame(frame_effect)

# Relabel response categories
frame_effect$response.label <- factor(frame_effect$response.level,
                                      levels = c("1", "2", "3", "4", "5"),
                                      labels = c("Not Important at All",
                                                 "Mildly Unimportant",
                                                 "Neither Important nor Unimportant",
                                                 "Mildly Important",
                                                 "Important"))

# Relabel Frame categories
frame_effect$x <- factor(frame_effect$x,
                         levels = c("NJ3", "NJ1", "NJ2"),
                         labels = c("Control", "Collectivist", "Individualist"))
```

```r
# Plot with proper groupings and restored colors
ggplot(frame_effect, aes(x = x, y = predicted,
                         ymin = conf.low, ymax = conf.high,
                         color = response.label,
                         shape = response.label,
                         group = response.label)) +

  geom_hline(yintercept = seq(0, 1, by = 0.1), color = "gray90", linetype = "dotted") +

  geom_errorbar(linewidth = 1, position = position_dodge(width = 0.4), width = 0.15, na.rm = TRUE) +

  geom_point(size = 3, position = position_dodge(width = 0.4), na.rm = TRUE) +

  scale_color_manual(name = "Level of Importance",
                     values = c("Not Important at All" = "#440154FF",
                                "Mildly Unimportant" = "#3B528BFF",
                                "Neither Important nor Unimportant" = "#21908CFF",
                                "Mildly Important" = "#5DC863FF",
                                "Important" = "#800000"))+

  scale_shape_manual(name = "Level of Importance",
                     values = c("Not Important at All" = 19,
                                "Mildly Unimportant" = 19, #from 17
                                "Neither Important nor Unimportant" = 19, #from 15
                                "Mildly Important" = 19, #from 18
                                "Important" = 19)) + #from 16

  scale_y_continuous("Predicted Probability", limits = c(0, 1), breaks = seq(0, 1, by = 0.1)) +

  labs(subtitle = str_wrap("How Respondents Ranked SSEC Reform Importance Across Frames"),
       x = "Frame Recieved by Group",
       y = "Predicted Probability of Ranking")+

  theme_minimal(base_size = 14) +
  theme(legend.position = c(2, 1),
        legend.justification = c(1, 1),
        legend.background = element_rect(fill = "white", color = NA),
        legend.key.size = unit(0.8, "cm"),
        legend.text = element_text(size = 11),
        legend.title = element_text(size = 12, face = "bold"),
        panel.grid.major.y = element_line(color = "gray85", linetype = "dotted"),
        axis.text.x = element_text(size = 12))
```




```
ggsave("framing_plot_word_ready.png", width = 7, height = 5, dpi = 450)
```

Trump Policy Facet Graph

```
x$D5 <- as.numeric(x$D5)
trump <- x |> filter(D5<4 | D5>5) |> mutate(Cutoff_Income = ifelse(D5 > 5, 1, 0)) # high values get a 1

trump$Cutoff_Income <- as.factor(trump$Cutoff_Income)

ordinal_model_t_cutoff <- clm(NJ4 ~ Frame*Cutoff_Income + NJ6 + NJ7+ NJ8+ NJ9 +NJ10_1 +
           NJ11 + NJ12 + D1 + sex + agecat+ education + racethn + D8, data = trump)

ordinal_model_t_cutoff_no <- clm(NJ4 ~ Frame + Cutoff_Income + NJ6 + NJ7+ NJ8+ NJ9 +NJ10_1 +
           NJ11 + NJ12 + D1 + sex + agecat+ education + racethn + D8, data = trump)

frame_income_effect_trump <- ggpredict(ordinal_model_t_cutoff, terms = c("Frame", "Cutoff_Income"))
print(frame_income_effect_trump)
```

```
## # Predicted probabilities of NJ4
##
## NJ4: 1
## Cutoff_Income: 0
##
```

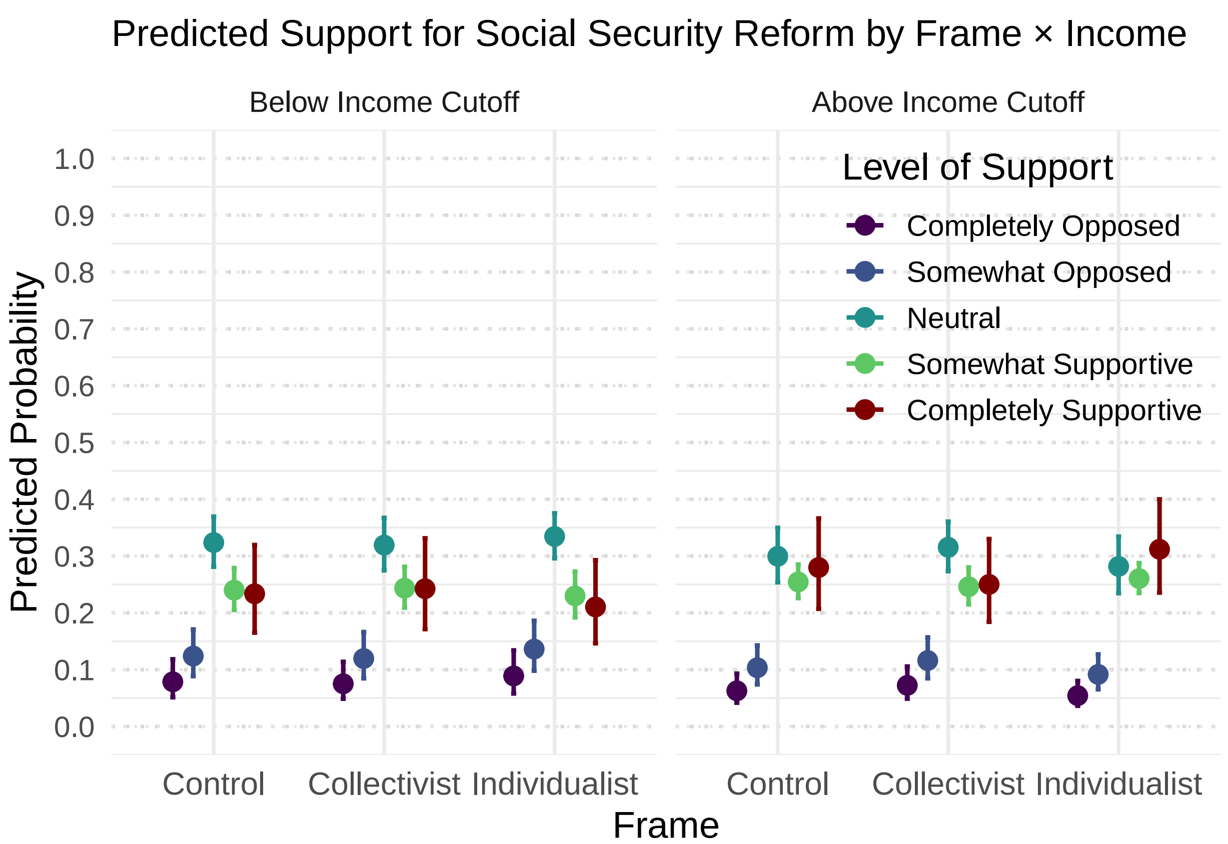
```
## Frame | Predicted |     95% CI
## ------------------------------
## NJ3   |      0.08 | 0.05, 0.12
## NJ1   |      0.08 | 0.05, 0.11
## NJ2   |      0.09 | 0.06, 0.13
##
## NJ4: 1
## Cutoff_Income: 1
##
## Frame | Predicted |     95% CI
## ------------------------------
## NJ3   |      0.06 | 0.04, 0.09
## NJ1   |      0.07 | 0.05, 0.11
## NJ2   |      0.05 | 0.04, 0.08
##
## NJ4: 2
## Cutoff_Income: 0
##
## Frame | Predicted |     95% CI
## ------------------------------
## NJ3   |      0.12 | 0.09, 0.17
## NJ1   |      0.12 | 0.08, 0.17
## NJ2   |      0.14 | 0.10, 0.19
##
## NJ4: 2
## Cutoff_Income: 1
##
## Frame | Predicted |     95% CI
## ------------------------------
## NJ3   |      0.10 | 0.07, 0.14
## NJ1   |      0.12 | 0.08, 0.16
## NJ2   |      0.09 | 0.07, 0.13
##
## NJ4: 3
## Cutoff_Income: 0
##
## Frame | Predicted |     95% CI
## ------------------------------
## NJ3   |      0.32 | 0.28, 0.37
## NJ1   |      0.32 | 0.27, 0.37
## NJ2   |      0.33 | 0.30, 0.38
##
## NJ4: 3
## Cutoff_Income: 1
##
## Frame | Predicted |     95% CI
## ------------------------------
## NJ3   |      0.30 | 0.25, 0.35
## NJ1   |      0.32 | 0.27, 0.36
## NJ2   |      0.28 | 0.23, 0.33
##
## NJ4: 4
## Cutoff_Income: 0
##
```

```
## Frame | Predicted |     95% CI
## ------------------------------
## NJ3   |      0.24 | 0.21, 0.28
## NJ1   |      0.24 | 0.21, 0.28
## NJ2   |      0.23 | 0.19, 0.27
##
## NJ4: 4
## Cutoff_Income: 1
##
## Frame | Predicted |     95% CI
## ------------------------------
## NJ3   |      0.25 | 0.23, 0.29
## NJ1   |      0.25 | 0.21, 0.28
## NJ2   |      0.26 | 0.23, 0.29
##
## NJ4: 5
## Cutoff_Income: 0
##
## Frame | Predicted |     95% CI
## ------------------------------
## NJ3   |      0.23 | 0.17, 0.32
## NJ1   |      0.24 | 0.17, 0.33
## NJ2   |      0.21 | 0.15, 0.29
##
## NJ4: 5
## Cutoff_Income: 1
##
## Frame | Predicted |     95% CI
## ------------------------------
## NJ3   |      0.28 | 0.21, 0.37
## NJ1   |      0.25 | 0.18, 0.33
## NJ2   |      0.31 | 0.24, 0.40
##
## Adjusted for:
## *       NJ6 =    1
## *       NJ7 = 2.00
## *       NJ8 = 2.00
## *       NJ9 = 3.00
## *    NJ10_1 = 3.00
## *      NJ11 =    1
## *      NJ12 = 3.00
## *        D1 =    1
## *       sex =    1
## *    agecat = 5.00
## * education = 3.00
## *   racethn =    1
## *        D8 =    1
```

```r
# Convert to data frame
frame_income_effect_trump <- as.data.frame(frame_income_effect_trump)


# Frame relabeling
frame_income_effect_trump$Frame <- factor(frame_income_effect_trump$x,
```

```r
                                          levels = c("NJ3", "NJ1", "NJ2"),
                                          labels = c("Control", "Collectivist", "Individualist"))

# Response labeling
frame_income_effect_trump$response.label <- factor(frame_income_effect_trump$response.level,
                                                   levels = c("1", "2", "3", "4", "5"),
                                                   labels = c("Completely Opposed",
                                                              "Somewhat Opposed",
                                                              "Neutral",
                                                              "Somewhat Supportive",
                                                              "Completely Supportive"))

# Income Group Relabeling
frame_income_effect_trump$Income_group <- factor(frame_income_effect_trump$group,
                                                   levels = c("0", "1"),
                                                   labels = c("Below Income Cutoff",
                                                              "Above Income Cutoff"))


# Plotting

ggplot(frame_income_effect_trump, aes(x = Frame, y = predicted,
                                      ymin = conf.low, ymax = conf.high,
                                      color = response.label,
                                      shape = response.label,
                                      group = interaction(response.label, Income_group))) +

  geom_hline(yintercept = seq(0, 1, by = 0.1), color = "gray90", linetype = "dotted") +

  geom_errorbar(linewidth = 0.8,
                position = position_dodge(width = 0.6), width = 0.15,
                na.rm = TRUE) +

  geom_point(size = 3,
            position = position_dodge(width = 0.6),
            na.rm = TRUE) +

  scale_color_manual(name = "Level of Support",
                    values = c("Completely Opposed" = "#440154FF",
                               "Somewhat Opposed" = "#3B528BFF",
                               "Neutral" = "#21908CFF",
                               "Somewhat Supportive" = "#5DC863FF",
                               "Completely Supportive" = "#800000")) +

  scale_shape_manual(name = "Level of Support",
                    values = c(19, 19, 19, 19, 19)) +

  scale_y_continuous("Predicted Probability", limits = c(0, 1), breaks = seq(0, 1, by = 0.1)) +

  labs(subtitle = "Predicted Support for Social Security Reform by Frame × Income",
      x = "Frame",
      y = "Predicted Probability") +

  facet_wrap(~ Income_group) +
```

```
  theme_minimal(base_size = 14) +
  theme(legend.position = c(1, 1),
        legend.justification = c(1, 1),
        panel.grid.major.y = element_line(color = "gray85", linetype = "dotted"),
        axis.text.x = element_text(size = 12))
```

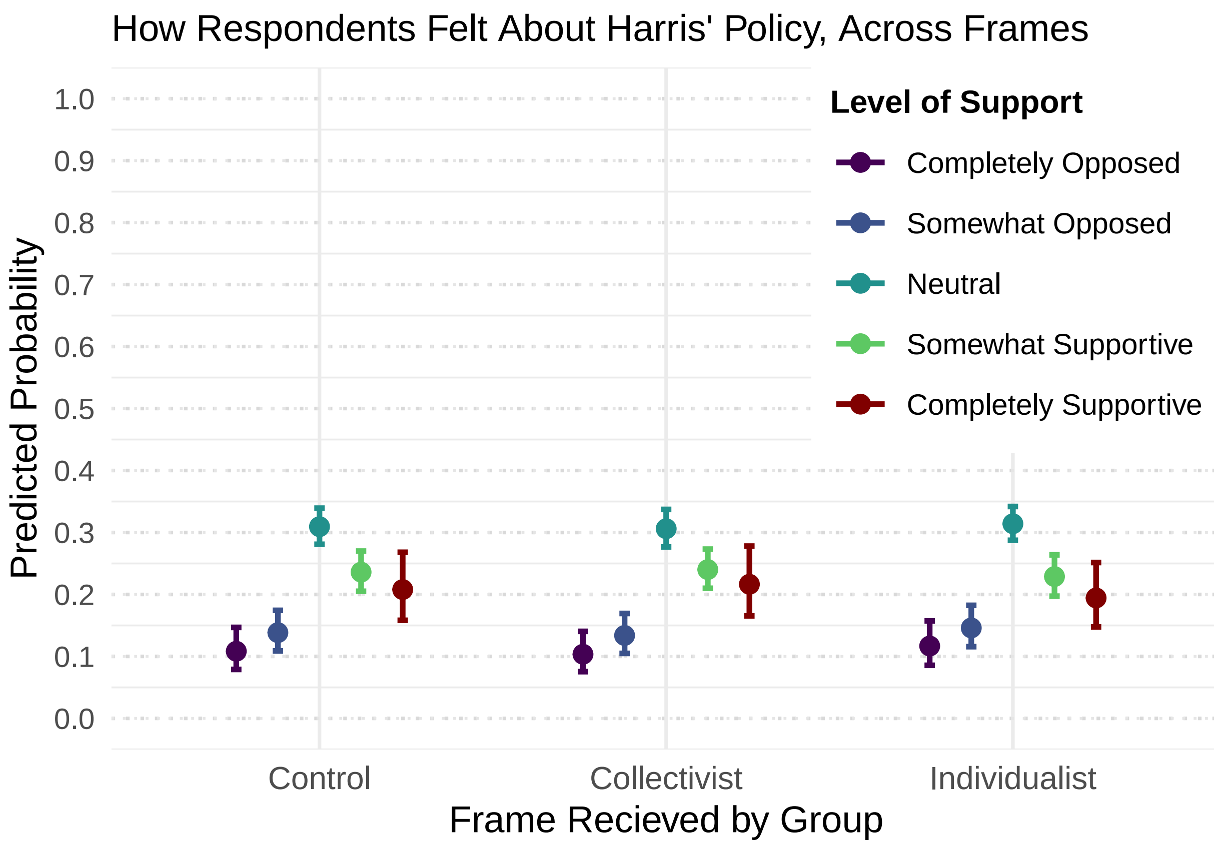



```
ggsave("framing_interaction_plot_word.png", width = 7, height = 5, dpi = 450)
```

Plot Ordinal Model Harris Policy

```
# Compute predicted probabilities for Frame
frame_effect_h <- ggpredict(ordinal_model_h, terms = "Frame")


# Convert to data.frame
frame_effect_h <- as.data.frame(frame_effect_h)

# Relabeling response categories
frame_effect_h$response.label <- factor(frame_effect_h$response.level,
                                      levels = c("1", "2", "3", "4", "5"),
                                      labels = c("Completely Opposed",
                                                 "Somewhat Opposed",
```

```
                                                               "Neutral",
                                                               "Somewhat Supportive",
                                                               "Completely Supportive"))

# Relabel Frame categories
frame_effect_h$x <- factor(frame_effect_h$x,
                         levels = c("NJ3", "NJ1", "NJ2"),
                         labels = c("Control", "Collectivist", "Individualist"))

# Plotting
ggplot(frame_effect_h, aes(x = x, y = predicted,
                         ymin = conf.low, ymax = conf.high,
                         color = response.label,
                         shape = response.label,
                         group = response.label)) +

  geom_hline(yintercept = seq(0, 1, by = 0.1), color = "gray90", linetype = "dotted") +

  geom_errorbar(linewidth = 1, position = position_dodge(width = 0.6),width = 0.15, na.rm = TRUE) +

  geom_point(size = 3, position = position_dodge(width = 0.6), na.rm = TRUE) +

  scale_color_manual(name = "Level of Support",
                     values = c("Completely Opposed" = "#440154FF",
                                "Somewhat Opposed" = "#3B528BFF",
                                "Neutral" = "#21908CFF",
                                "Somewhat Supportive" = "#5DC863FF",
                                "Completely Supportive" = "#800000")) +

  scale_shape_manual(name = "Level of Support",
                     values = c("Completely Opposed" = 19,
                                "Somewhat Opposed" = 19,
                                "Neutral" = 19,
                                "Somewhat Supportive" = 19,
                                "Completely Supportive" = 19)) +

  scale_y_continuous("Predicted Probability", limits = c(0, 1.0), breaks = seq(0, 1, by = 0.1)) +

  labs(subtitle = str_wrap("How Respondents Felt About Harris' Policy, Across Frames"),
       x = "Frame Recieved by Group",
       y = "Predicted Probability of Support Level")+

  theme_minimal(base_size = 14) +
  theme(legend.position = c(1, 1),
        legend.justification = c(1, 1),
        legend.background = element_rect(fill = "white", color = NA),
        legend.key.size = unit(0.8, "cm"),
        legend.text = element_text(size = 11),
        legend.title = element_text(size = 12, face = "bold"),
        panel.grid.major.y = element_line(color = "gray85", linetype = "dotted"),
        axis.text.x = element_text(size = 12))
```




```
ggsave("framing_plot_harris_word.png", width = 7, height = 5, dpi = 450)
```

Experiment Part 2: Voting Patterns

Do Voters' Preferences Match their Votes

```
x <- x |>
  mutate(SupportCategory_4 = case_when(
    NJ4 %in% c(4, 5) ~ "Support",
    NJ4 == 3 ~ "Neutral",
    NJ4 %in% c(1, 2) ~ "Not Support"
  ))

x <- x |>
  mutate(SupportCategory_5 = case_when(
    NJ5 %in% c(4, 5) ~ "Support",
    NJ5 == 3 ~ "Neutral",
    NJ5 %in% c(1, 2) ~ "Not Support"
  ))

# Calculate the percentage of "Not Support" voters for each candidate
non_support_summary_trump <- x |>
  group_by(vote) |>
```

```r
  summarize(
    Total_Voters = n(),
    Non_Support_Voters = sum(SupportCategory_4 == "Not Support"),
    Percentage_Non_Support = (Non_Support_Voters / Total_Voters) * 100
  )

# Results
print(non_support_summary_trump)
```

```
## # A tibble: 4 x 4
##    vote Total_Voters Non_Support_Voters Percentage_Non_Support
##   <int>        <int>              <int>                  <dbl>
## 1     1          652                114                   17.5
## 2     2          768                171                   22.3
## 3     3           51                 11                   21.6
## 4     4          532                107                   20.1
```

```r
# 17 percent of Trump supporters do not support Trump's policies, while 22% of
# Harris voters did not support Trump's policies.

# Calculate the percentage of "Not Support" voters for each candidate for Harris Policy
non_support_summary_harris <- x |>
  group_by(vote) |>
  summarize(
    Total_Voters = n(),
    Non_Support_Voters = sum(SupportCategory_5 == "Not Support"),
    Percentage_Non_Support = (Non_Support_Voters / Total_Voters) * 100
  )

# Results
print(non_support_summary_harris)
```

```
## # A tibble: 4 x 4
##    vote Total_Voters Non_Support_Voters Percentage_Non_Support
##   <int>        <int>              <int>                  <dbl>
## 1     1          652                176                   27.0
## 2     2          768                145                   18.9
## 3     3           51                  9                   17.6
## 4     4          532                122                   22.9
```

```r
# 27% of Trump Supporters do not support Harris' policy. Conversely, 18 percent
# of Harris supporters do not support her own policy.

# The first table analyzes the percentage of each category of voters who do (not)
# support Trump's policy.

# The second table analyzes the percentage of each category of voters who do (not)
# Support Harris
```

```r
# Calculate the percentage of "Support" voters for each candidate Harris Policy
support_summary_5 <- x %>%
  group_by(vote) %>%
```

```
  summarize(
    Total_Voters = n(),
    Support_Voters = sum(SupportCategory_5 == "Support"),
    Percentage_Support = (Support_Voters / Total_Voters) * 100
  )

print(support_summary_5)
```

```
## # A tibble: 4 x 4
##    vote Total_Voters Support_Voters Percentage_Support
##   <int>        <int>          <int>              <dbl>
## 1     1          652            306               46.9
## 2     2          768            461               60.0
## 3     3           51             29               56.9
## 4     4          532            202               38.0
```

```
# Supporters for Trump Policy
support_summary_4 <- x %>%
  group_by(vote) %>%
  summarize(
    Total_Voters = n(),
    Support_Voters = sum(SupportCategory_4 == "Support"),
    Percentage_Support = (Support_Voters / Total_Voters) * 100
  )

print(support_summary_4)
```

```
## # A tibble: 4 x 4
##    vote Total_Voters Support_Voters Percentage_Support
##   <int>        <int>          <int>              <dbl>
## 1     1          652            404               62.0
## 2     2          768            384               50
## 3     3           51             29               56.9
## 4     4          532            207               38.9
```

```
sum_support_trump <- support_summary_4 |> summarize(Total = sum(Total_Voters), Support = sum(Support_Vo

sum_support_trump <- sum_support_trump |> mutate(Oppose = Total - Support)
print(sum_support_trump)
```

```
## # A tibble: 1 x 4
##   Total Support Percentage_Total_Support Oppose
##   <int>   <int>                    <dbl>  <int>
## 1  2003    1024                    0.511    979
```

```
sum_support_harris <- support_summary_5 |> summarize(Total = sum(Total_Voters), Support = sum(Support_V

sum_support_harris <- sum_support_harris |> mutate(Oppose = Total - Support)
print(sum_support_harris)
```

```
## # A tibble: 1 x 4
```

```
##   Total Support Percentage_Total_Support Oppose
##   <int>   <int>                    <dbl>  <int>
## 1  2003     998                    0.498   1005
```

Analyzing whether policy preferences are predictive of voting behavior

```
## Subsetting out non-voters
x2<- x |> filter(vote!=4)

# Fit multinomial logistic regression
multimodel <- multinom(vote ~ NJ4 + NJ5+ NJ6 + NJ7+
+             NJ8+ NJ9 +NJ10_1 +
+             NJ11 + NJ12 + D1+ sex +
+            agecat+ education + as.factor(racethn) + D5
+            + D8, data = x2)
```

```
## # weights:  111 (72 variable)
## initial  value 1605.072554
## iter  10 value 865.965198
## iter  20 value 743.006703
## iter  30 value 671.185749
## iter  40 value 654.576152
## iter  50 value 650.042683
## iter  60 value 649.597590
## iter  70 value 649.566278
## iter  80 value 649.546039
## iter  90 value 649.538816
## final  value 649.538733
## converged
```

```
# Extract coefficients and standard errors
summary_model <- summary(multimodel)
coefs <- summary_model$coefficients
std_errors <- summary_model$standard.errors

# Z-values
z_values <- coefs / std_errors

# P-values
p_values <- 2 * (1 - pnorm(abs(z_values)))
```

Creating an interpretable multinomial regression table

```
## Removing ordering for interpretability
x2$NJ4 <- factor(x2$NJ4, ordered = FALSE)
x2$NJ5 <- factor(x2$NJ5, ordered = FALSE)

# Fit multinomial logistic regression
multimodel <- multinom(vote ~ NJ4 + NJ5 + NJ6 + NJ7 + NJ8 + NJ9 + NJ10_1 +
```

```
                         NJ11 + NJ12 + D1 + sex + agecat + education +
                         as.factor(racethn) + D5 + D8, data = x2)
```

```
## # weights:  111 (72 variable)
## initial  value 1605.072554
## iter  10 value 865.676307
## iter  20 value 745.834325
## iter  30 value 671.420251
## iter  40 value 654.628768
## iter  50 value 650.127223
## iter  60 value 649.593496
## iter  70 value 649.562293
## iter  80 value 649.542374
## iter  90 value 649.538778
## final  value 649.538734
## converged
```

```
# Convert model output into a structured data frame
results_table <- tidy(multimodel, conf.int = TRUE) %>%
  mutate(Odds_Ratio = exp(estimate)) %>%
  dplyr::select(y.level, term, estimate, std.error, statistic, p.value, conf.low, conf.high)

# Rename columns
results_table <- results_table %>%
  rename(`Outcome` = y.level,
         `Variable` = term,
         `Coef.` = estimate,
         `Std. Err.` = std.error,
         `Z` = statistic,
         `P>|z|` = p.value,
         `[95% Conf. Interval] Lower` = conf.low,
         `[95% Conf. Interval] Upper` = conf.high)

# Format p-values to adjust for small values
results_table <- results_table %>%
  mutate(`P>|z|` = ifelse(`P>|z|` < 0.0001, format(`P>|z|`, scientific = TRUE, digits = 2),
                        round(`P>|z|`, 4)))

# Format numeric values to 4 decimal places
results_table <- results_table %>%
  mutate(across(where(is.numeric), ~round(., 4)))

# Print table
kable(results_table, format = "pipe", align = "r", caption = "Multinomial Logistic Regression Results")
```

Table 1: Multinomial Logistic Regression Results

Outcome	Variable	Coef.	Std. Err.	Z	P>|z|	[95% Conf. Interval] Lower	[95% Conf. Interval] Upper
2	(Intercept)	-3.5123	0.7645	-4.594100e+00	4.3e-06	-5.0107	-2.0139

Outcome	Variable	Coef.	Std. Err.	Z	P>|z|	[95% Conf. Interval] Lower	[95% Conf. Interval] Upper
2	NJ42	0.1779	0.3923	4.535000e-01	0.6502	-0.5909	0.9468
2	NJ43	0.2628	0.3570	7.362000e-01	0.4616	-0.4369	0.9625
2	NJ44	-0.4176	0.3545	-1.178100e+00	0.2388	-1.1124	0.2771
2	NJ45	-0.9183	0.3421	-2.684400e+00	0.0073	-1.5887	-0.2478
2	NJ52	-0.8580	0.3656	-2.346500e+00	0.019	-1.5747	-0.1413
2	NJ53	-0.5409	0.3226	-1.676900e+00	0.0936	-1.1731	0.0913
2	NJ54	-0.3231	0.3149	-1.026000e+00	0.3049	-0.9402	0.2941
2	NJ55	0.6426	0.3087	2.081800e+00	0.0374	0.0376	1.2476
2	NJ62	-0.3027	0.2494	-1.213900e+00	0.2248	-0.7915	0.1861
2	NJ63	0.0949	0.2448	3.878000e-01	0.6982	-0.3849	0.5748
2	NJ7	-0.0599	0.0678	-8.837000e-01	0.3769	-0.1928	0.0730
2	NJ8	0.0298	0.0791	3.769000e-01	0.7062	-0.1252	0.1848
2	NJ9	0.0732	0.0876	8.352000e-01	0.4036	-0.0986	0.2449
2	NJ10__1	-0.0043	0.0677	-6.340000e-02	0.9494	-0.1371	0.1285
2	NJ112	-0.0741	0.2627	-2.819000e-01	0.778	-0.5889	0.4408
2	NJ113	-0.3643	0.3366	-1.082100e+00	0.2792	-1.0240	0.2955
2	NJ114	-0.3969	0.3188	-1.245300e+00	0.213	-1.0217	0.2278
2	NJ115	0.2964	0.4907	6.040000e-01	0.5459	-0.6654	1.2581
2	NJ116	0.2672	0.3663	7.295000e-01	0.4657	-0.4508	0.9852
2	NJ12	0.0512	0.0701	7.297000e-01	0.4656	-0.0862	0.1885
2	D12	-1.1413	1.8956	-6.021000e-01	0.5471	-4.8567	2.5740
2	D13	-1.0868	1.3606	-7.987000e-01	0.4244	-3.7535	1.5800
2	D14	-24.7007	0.0000	-3.743785e+10	0.0e+00	-24.7007	-24.7007
2	D15	7.5354	0.0001	7.627897e+04	0.0e+00	7.5352	7.5356

Outcome	Variable	Coef.	Std. Err.	Z	P>|z|	[95% Conf. Interval] Lower	[95% Conf. Interval] Upper
2	sex2	1.6266	1.8864	8.623000e-01	0.3885	-2.0706	5.3238
2	agecat	0.0106	0.0309	3.422000e-01	0.7322	-0.0499	0.0710
2	education	0.1970	0.0654	3.013900e+00	0.0026	0.0689	0.3251
2	as.factor(racethn)2	1.0587	0.2494	4.244800e+00	2.2e-05	0.5698	1.5475
2	as.factor(racethn)3	-0.1799	0.2694	-6.678000e-01	0.5043	-0.7079	0.3481
2	as.factor(racethn)4	0.1456	0.3337	4.364000e-01	0.6626	-0.5084	0.7996
2	as.factor(racethn)5	0.1972	0.2988	6.601000e-01	0.5092	-0.3884	0.7829
2	D5	-0.0023	0.0327	-7.060000e-02	0.9437	-0.0663	0.0617
2	D82	4.7233	0.2393	1.973810e+01	1.0e-86	4.2543	5.1923
2	D83	2.4404	0.2286	1.067460e+01	1.3e-26	1.9923	2.8885
2	D84	2.6861	0.4273	6.285900e+00	3.3e-10	1.8486	3.5237
3	(Intercept)	-5.1330	1.6896	-3.037900e+00	0.0024	-8.4446	-1.8213
3	NJ42	-0.8874	0.7536	-1.177500e+00	0.239	-2.3644	0.5897
3	NJ43	-0.7430	0.6490	-1.144800e+00	0.2523	-2.0150	0.5290
3	NJ44	-0.8757	0.6277	-1.395200e+00	0.163	-2.1059	0.3545
3	NJ45	-1.2878	0.6213	-2.072600e+00	0.0382	-2.5056	-0.0700
3	NJ52	0.9206	0.9537	9.653000e-01	0.3344	-0.9487	2.7900
3	NJ53	1.1284	0.8975	1.257300e+00	0.2086	-0.6306	2.8874
3	NJ54	1.3163	0.8760	1.502700e+00	0.1329	-0.4006	3.0332
3	NJ55	1.1151	0.8899	1.253000e+00	0.2102	-0.6291	2.8593
3	NJ62	0.6428	0.4669	1.376700e+00	0.1686	-0.2723	1.5579
3	NJ63	0.4797	0.4627	1.036700e+00	0.2999	-0.4272	1.3866
3	NJ7	-0.0243	0.1412	-1.718000e-01	0.8636	-0.3010	0.2525
3	NJ8	-0.3542	0.1969	-1.798900e+00	0.072	-0.7400	0.0317
3	NJ9	-0.1533	0.1824	-8.407000e-01	0.4005	-0.5107	0.2041
3	NJ10_1	-0.0633	0.1314	-4.821000e-01	0.6298	-0.3208	0.1941
3	NJ112	-0.4295	0.4616	-9.304000e-01	0.3521	-1.3343	0.4753

Outcome	Variable	Coef.	Std. Err.	Z	P>|z|	[95% Conf. Interval] Lower	[95% Conf. Interval] Upper
3	NJ113	-1.7479	0.8946	-1.953800e+00	0.0507	-3.5012	0.0055
3	NJ114	-0.6786	0.6070	-1.117900e+00	0.2636	-1.8683	0.5112
3	NJ115	0.4898	0.9273	5.283000e-01	0.5973	-1.3276	2.3073
3	NJ116	-0.3928	0.6718	-5.847000e-01	0.5588	-1.7095	0.9239
3	NJ12	-0.2132	0.1447	-1.473800e+00	0.1405	-0.4968	0.0703
3	D12	-0.5961	2.2206	-2.684000e-01	0.7884	-4.9483	3.7562
3	D13	1.8127	1.9696	9.204000e-01	0.3574	-2.0476	5.6730
3	D14	0.9748	1.9842	4.913000e-01	0.6232	-2.9143	4.8638
3	D15	-1.1439	0.0000	-1.773655e+05	0.0e+00	-1.1440	-1.1439
3	sex2	1.2953	2.1970	5.896000e-01	0.5555	-3.0108	5.6014
3	agecat	0.1680	0.0668	2.513800e+00	0.0119	0.0370	0.2989
3	education	0.1812	0.1333	1.359400e+00	0.174	-0.0800	0.4424
3	as.factor(racethn)2	0.5048	0.5469	9.230000e-01	0.356	-0.5671	1.5766
3	as.factor(racethn)3	0.1320	0.5819	2.268000e-01	0.8206	-1.0085	1.2724
3	as.factor(racethn)4	0.1335	0.6073	2.198000e-01	0.826	-1.0568	1.3237
3	as.factor(racethn)5	0.6319	0.5261	1.201100e+00	0.2297	-0.3992	1.6630
3	D5	0.0637	0.0636	1.001700e+00	0.3165	-0.0609	0.1883
3	D82	2.4699	0.6663	3.707000e+00	2e-04	1.1640	3.7758
3	D83	3.0315	0.5543	5.468700e+00	4.5e-08	1.9450	4.1180
3	D84	4.4060	0.7280	6.052300e+00	1.4e-09	2.9792	5.8329

Performing Chi-Squared Test to assess collinearity

```r
## Trump associations
chi_trump_table <- table(x2$NJ4, x2$D8)

# Perform the Chi-Square test
chi_trump_test <- chisq.test(chi_trump_table)
```

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

```r
# View results
chi_trump_test
```

```
##
##  Pearson's Chi-squared test
##
## data:  chi_trump_table
## X-squared = 34.31, df = 12, p-value = 0.0006027
```

```r
## Trump associations
chi_harris_table <- table(x2$NJ5, x2$D8)

# Perform the Chi-Square test
chi_harris_test <- chisq.test(chi_harris_table)
```

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

```r
# View results
chi_harris_test
```

```
##
##  Pearson's Chi-squared test
##
## data:  chi_harris_table
## X-squared = 37.378, df = 12, p-value = 0.0001939
```

Creating Stacked Barplot Dataframes for Visual Aid

```r
partisan_trump <- x |> group_by(D8, NJ4)|>
  summarise(Count = n(), .groups = "drop") |>
  filter(D8 ==1|D8==2)

## DF for Trump Policy Bar among Republicans
R_trump <- partisan_trump |> filter(D8 ==1) |>
  mutate(Percentage = Count / sum(Count) * 100)

sum(unique(R_trump$Percentage)) ## checking for accuracy
```

```
## [1] 100
```

```r
## DF for Trump Policy Bar among Democrats
D_trump <- partisan_trump |> filter(D8 ==2) |>
  mutate(Percentage = Count / sum(Count) * 100)

sum(unique(D_trump$Percentage)) ## checking for accuracy
```

```
## [1] 100
```

```r
partisan_harris <- x |> group_by(D8, NJ5) |> summarise(Count = n(), .groups = "drop")

R_harris <- partisan_harris |> filter(D8 ==1) |>
  mutate(Percentage = Count / sum(Count) * 100)

sum(unique(R_harris$Percentage)) ## checking for accuracy
```

```
## [1] 100
```

```r
D_harris <- partisan_harris |> filter(D8 ==2) |>
  mutate(Percentage = Count / sum(Count) * 100)

sum(unique(D_harris$Percentage)) ## checking for accuracy
```

```
## [1] 100
```

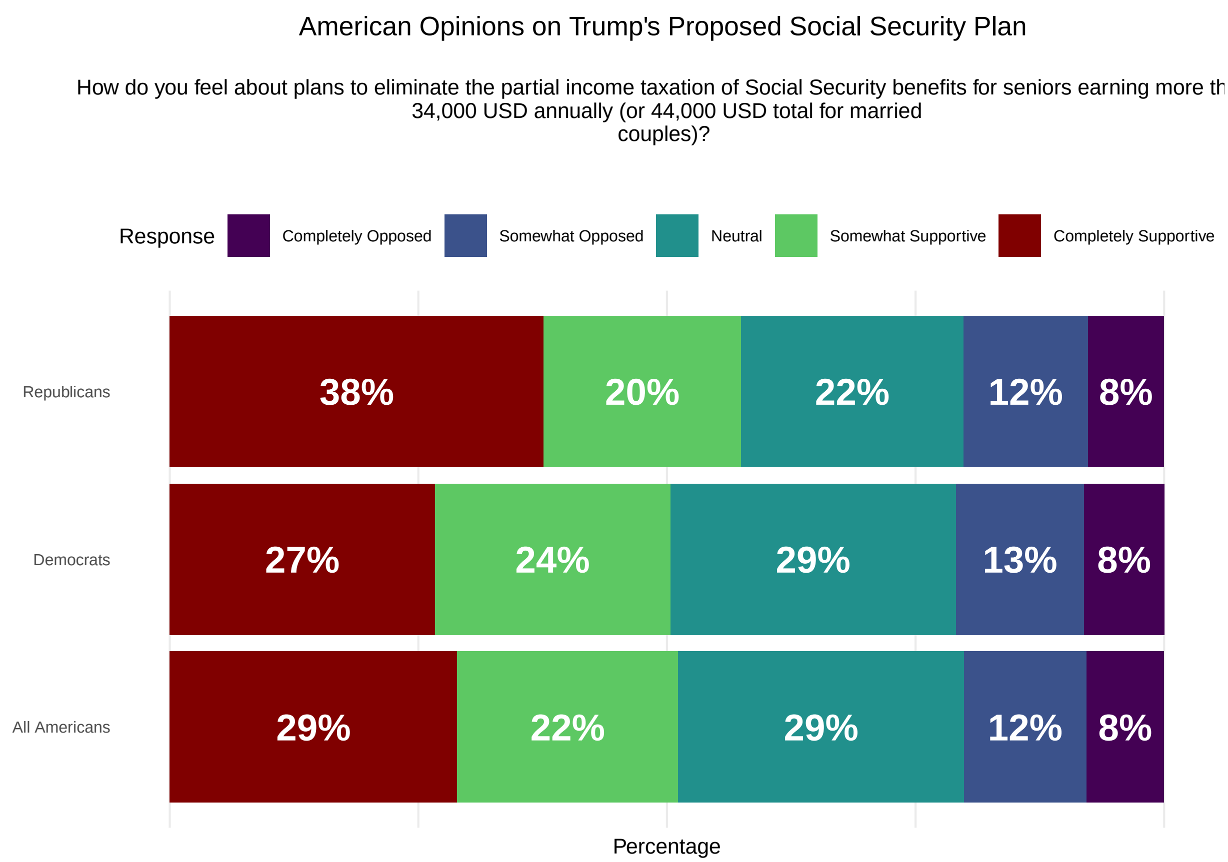
```r
all_american_trump <- x |> group_by(NJ4) |>
  summarise(Count = n(), .groups = "drop") %>%
  mutate(Percentage = Count / sum(Count) * 100)

all_american_harris <- x |> group_by(NJ5) |>
  summarise(Count = n(), .groups = "drop") |>
  mutate(Percentage = Count / sum(Count) * 100)

## If all dataframes are accurate, then the value 100 should be printed 4 times
```

Reorganizing these Dataframes for Subsequent Plots

```r
# Specifying the group
R_trump <- R_trump |> mutate(Group = "Republicans")
D_trump <- D_trump |> mutate(Group = "Democrats")
all_american_trump <- all_american_trump |> mutate(Group = "All Americans")

# Combining into one df
stacked_data <- bind_rows(R_trump, D_trump, all_american_trump)

# Renaming to match response levels
stacked_data$NJ4 <- factor(stacked_data$NJ4,
                                     levels = c("1", "2", "3", "4", "5"),
                                     labels = c("Completely Opposed",
                                                "Somewhat Opposed",
                                                "Neutral",
                                                "Somewhat Supportive",
                                                "Completely Supportive"))
```

Plotting Trump Policy

```r
ggplot(stacked_data, aes(x = Group, y = Percentage, fill = NJ4)) +
  geom_bar(stat = "identity", position = "stack") +
  geom_text(aes(label = paste0(round(Percentage, 0), "%")),
            position = position_stack(vjust = 0.5),
            size = 5,
            color = "white",
            fontface = "bold") +
  coord_flip() +
  scale_fill_manual(values = c("Completely Opposed" = "#440154FF",
                               "Somewhat Opposed" = "#3B528BFF",
                               "Neutral" = "#21908CFF",
                               "Somewhat Supportive" = "#5DC863FF",
                               "Completely Supportive" = "#800000")) +
  labs(
    title = "American Opinions on Trump's Proposed Social Security Plan \n",
    subtitle = "How do you feel about plans to eliminate the partial income taxation of Social Security
couples)?
\n",
    x = "",
    y = "Percentage",
    fill = "Response"
  ) +

  theme_minimal(base_size = 8) +
  theme(
    plot.title = element_text(hjust = 0.5),
    plot.subtitle=element_text(hjust = 0.5),
    panel.grid.major.y = element_blank(),
    panel.grid.minor = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    legend.position = "top"
  )
```



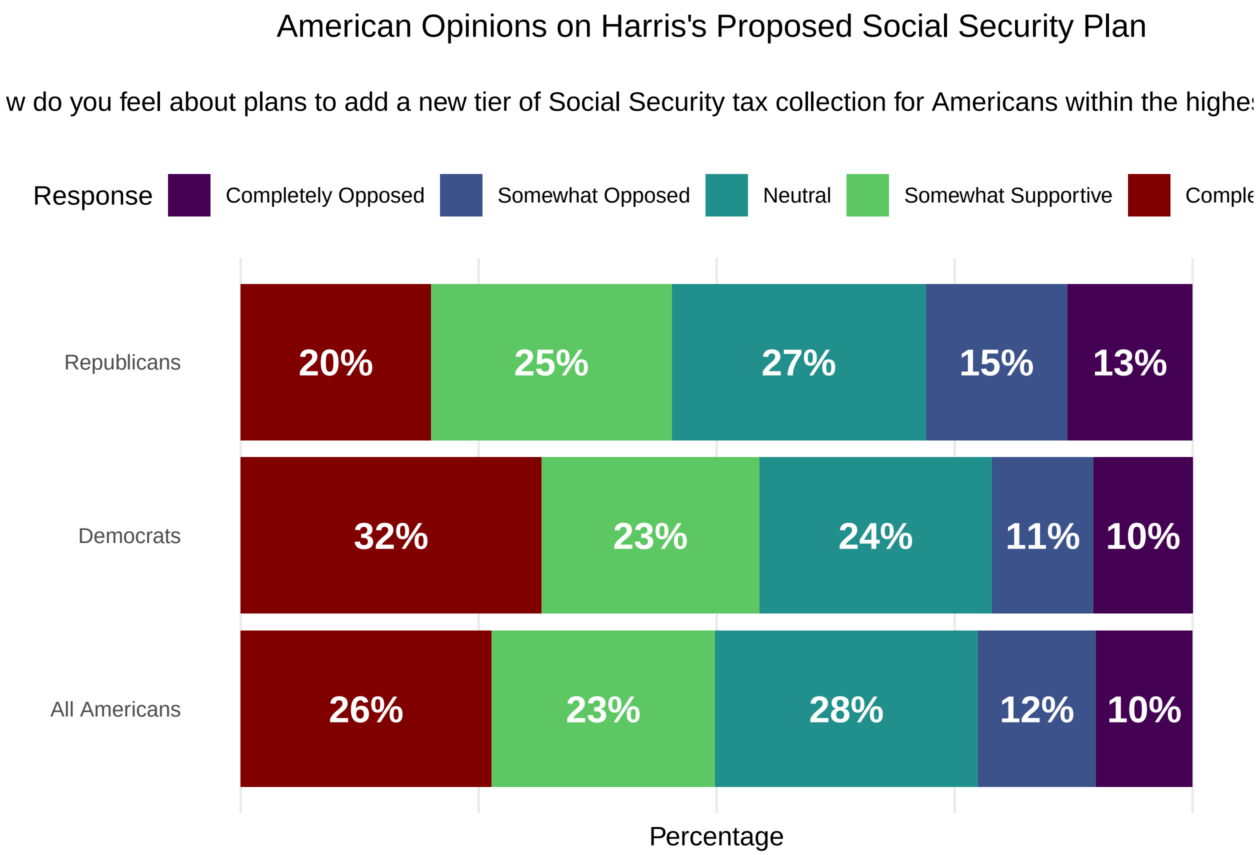

```r
ggsave("descriptive_trump.png", width = 7, height = 5, dpi = 600)
```

Organizing Data for Harris

```r
# Group specification
R_harris <- R_harris |> mutate(Group = "Republicans")
D_harris <- D_harris |> mutate(Group = "Democrats")
all_american_harris <- all_american_harris |> mutate(Group = "All Americans")

# Creating one df
stacked_data <- bind_rows(R_harris, D_harris, all_american_harris)

# Renaming NJ5 to match response levels
stacked_data$NJ5 <- factor(stacked_data$NJ5,
                           levels = c("1", "2", "3", "4", "5"),
                           labels = c("Completely Opposed",
                                      "Somewhat Opposed",
                                      "Neutral",
                                      "Somewhat Supportive",
                                      "Completely Supportive"))
```

##Plotting Harris

```r
ggplot(stacked_data, aes(x = Group, y = Percentage, fill = NJ5)) +
  geom_bar(stat = "identity", position = "stack") +
  geom_text(aes(label = paste0(round(Percentage, 0), "%")),
            position = position_stack(vjust = 0.5),
            size = 5,
            color = "white",
            fontface = "bold") +

  coord_flip() +
  scale_fill_manual(values = c("Completely Opposed" = "#440154FF",
                               "Somewhat Opposed" = "#3B528BFF",
                               "Neutral" = "#21908CFF",
                               "Somewhat Supportive" = "#5DC863FF",
                               "Completely Supportive" = "#800000")) +
  labs(
    title = "American Opinions on Harris's Proposed Social Security Plan \n",
    subtitle = "How do you feel about plans to add a new tier of Social Security tax collection for Ame
    x = "",
    y = "Percentage",
    fill = "Response"
  ) +

  theme_minimal(base_size = 10) +
  theme(
    plot.title = element_text(hjust = 0.5),
    plot.subtitle=element_text(hjust = 0.5),
    panel.grid.major.y = element_blank(),
    panel.grid.minor = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    legend.position = "top"
  )
```




```r
ggsave("descriptive_harris.png", width = 7, height = 5, dpi = 600)
```

Transferring Regression Tables to Word

Ordinal Regression 1: Importance

```r
coef(summary(ordinal_model_imp))
```

```
##               Estimate Std. Error    z value     Pr(>|z|)
## 1|2        -0.17291120 0.39371637 -0.4391771 6.605332e-01
## 2|3         1.40553268 0.37965792  3.7021028 2.138199e-04
## 3|4         2.52234935 0.38140515  6.6133070 3.758279e-11
## 4|5         3.63779092 0.38616080  9.4204046 4.493533e-21
## FrameNJ1    0.18122279 0.11607876  1.5612053 1.184753e-01
## FrameNJ2    0.10575583 0.11370204  0.9301137 3.523122e-01
## NJ62       -0.53681420 0.12700718 -4.2266446 2.372018e-05
## NJ63       -0.33669754 0.12291538 -2.7392629 6.157710e-03
## NJ7        -0.06659236 0.03885901 -1.7136917 8.658537e-02
## NJ8         0.03990999 0.04439624  0.8989497 3.686795e-01
## NJ9         0.48945997 0.04789318 10.2198262 1.616305e-24
## NJ10_1      0.04507169 0.03886095  1.1598196 2.461223e-01
## NJ112       0.04313096 0.15850849  0.2721050 7.855413e-01
## NJ113      -0.75064068 0.18792992 -3.9942586 6.489700e-05
```

```
## NJ114     -0.12757923 0.18153921 -0.7027640 4.822028e-01
## NJ115      0.37955512 0.28250020  1.3435570 1.790917e-01
## NJ116      0.09295818 0.19047583  0.4880314 6.255276e-01
## NJ12       0.24208115 0.04070577  5.9470973 2.729391e-09
## D12       -0.34861199 0.77423300 -0.4502675 6.525175e-01
## D13       -0.29954577 0.59503928 -0.5034050 6.146795e-01
## D14        0.17272496 0.97644290  0.1768920 8.595932e-01
## D15       -0.84462040 1.22387675 -0.6901188 4.901195e-01
## sex2       0.67668080 0.76723784  0.8819701 3.777930e-01
## agecat     0.19480800 0.01741305 11.1874737 4.696236e-29
## education  0.05730205 0.03736047  1.5337614 1.250884e-01
## racethn2  -0.10927040 0.13367703 -0.8174209 4.136879e-01
## racethn3  -0.23989454 0.15465461 -1.5511632 1.208626e-01
## racethn4  -0.42409949 0.16577064 -2.5583511 1.051699e-02
## racethn5  -0.15573206 0.15883868 -0.9804417 3.268681e-01
## D5         0.04288861 0.01776099  2.4147647 1.574538e-02
## D82        0.14685259 0.12174077  1.2062729 2.277123e-01
## D83        0.03962851 0.12716419  0.3116326 7.553197e-01
## D84        0.05187003 0.19831324  0.2615561 7.936637e-01
```

```r
coef_table <- as.data.frame(coef(summary(ordinal_model_imp)))
print(coef_table)
```

```
##               Estimate Std. Error    z value     Pr(>|z|)
## 1|2       -0.17291120 0.39371637 -0.4391771 6.605332e-01
## 2|3        1.40553268 0.37965792  3.7021028 2.138199e-04
## 3|4        2.52234935 0.38140515  6.6133070 3.758279e-11
## 4|5        3.63779092 0.38616080  9.4204046 4.493533e-21
## FrameNJ1   0.18122279 0.11607876  1.5612053 1.184753e-01
## FrameNJ2   0.10575583 0.11370204  0.9301137 3.523122e-01
## NJ62      -0.53681420 0.12700718 -4.2266446 2.372018e-05
## NJ63      -0.33669754 0.12291538 -2.7392629 6.157710e-03
## NJ7       -0.06659236 0.03885901 -1.7136917 8.658537e-02
## NJ8        0.03990999 0.04439624  0.8989497 3.686795e-01
## NJ9        0.48945997 0.04789318 10.2198262 1.616305e-24
## NJ10_1     0.04507169 0.03886095  1.1598196 2.461223e-01
## NJ112      0.04313096 0.15850849  0.2721050 7.855413e-01
## NJ113     -0.75064068 0.18792992 -3.9942586 6.489700e-05
## NJ114     -0.12757923 0.18153921 -0.7027640 4.822028e-01
## NJ115      0.37955512 0.28250020  1.3435570 1.790917e-01
## NJ116      0.09295818 0.19047583  0.4880314 6.255276e-01
## NJ12       0.24208115 0.04070577  5.9470973 2.729391e-09
## D12       -0.34861199 0.77423300 -0.4502675 6.525175e-01
## D13       -0.29954577 0.59503928 -0.5034050 6.146795e-01
## D14        0.17272496 0.97644290  0.1768920 8.595932e-01
## D15       -0.84462040 1.22387675 -0.6901188 4.901195e-01
## sex2       0.67668080 0.76723784  0.8819701 3.777930e-01
## agecat     0.19480800 0.01741305 11.1874737 4.696236e-29
## education  0.05730205 0.03736047  1.5337614 1.250884e-01
## racethn2  -0.10927040 0.13367703 -0.8174209 4.136879e-01
## racethn3  -0.23989454 0.15465461 -1.5511632 1.208626e-01
## racethn4  -0.42409949 0.16577064 -2.5583511 1.051699e-02
## racethn5  -0.15573206 0.15883868 -0.9804417 3.268681e-01
## D5         0.04288861 0.01776099  2.4147647 1.574538e-02
```

```
## D82        0.14685259 0.12174077  1.2062729 2.277123e-01
## D83        0.03962851 0.12716419  0.3116326 7.553197e-01
## D84        0.05187003 0.19831324  0.2615561 7.936637e-01
```

```r
colnames(coef_table) <- c("Estimate", "Std. Error", "z value", "p value")



coef_table <- coef_table %>%
  mutate(`p value` = ifelse(`p value` < 0.0001, format(`p value`, scientific = TRUE, digits = 2),
                           round(`p value`, 4)))

# Making a flextable
ft <- flextable(coef_table) %>%
  colformat_num(j = c("Estimate", "Std. Error", "z value"), digits = 4) %>%
  colformat_char(j = "p value") %>%
  autofit()

# Saving as a Word document
save_as_docx(ft, path = "ordinal_regression_results_1.docx")
```

##Trump Ordinal Model

```r
coef(summary(ordinal_model_t_cutoff_no))
```

```
##                   Estimate Std. Error     z value     Pr(>|z|)
## 1|2            -0.39680758 0.37484973 -1.05857775 2.897921e-01
## 2|3             0.69427366 0.37092287  1.87174672 6.124165e-02
## 3|4             2.16758598 0.37415602  5.79326775 6.902992e-09
## 4|5             3.24637336 0.37900307  8.56555961 1.075522e-17
## FrameNJ1       -0.06575747 0.11240237 -0.58501855 5.585352e-01
## FrameNJ2        0.03457993 0.11301228  0.30598384 7.596170e-01
## Cutoff_Income1  0.26861143 0.10851258  2.47539448 1.330891e-02
## NJ62           -0.16199565 0.13368282 -1.21179106 2.255924e-01
## NJ63            0.09854182 0.12244698  0.80477133 4.209516e-01
## NJ7             0.08913856 0.03806652  2.34165246 1.919858e-02
## NJ8             0.07458839 0.04440744  1.67963706 9.302795e-02
## NJ9             0.15058662 0.04686800  3.21299450 1.313588e-03
## NJ10_1          0.06324046 0.03813918  1.65814931 9.728733e-02
## NJ112           0.23629279 0.14833188  1.59300074 1.111600e-01
## NJ113          -0.33958382 0.18457971 -1.83976788 6.580232e-02
## NJ114           0.01978051 0.17344364  0.11404573 9.092015e-01
## NJ115           0.05317574 0.26080702  0.20388923 8.384401e-01
## NJ116           0.16704085 0.18225311  0.91653224 3.593878e-01
## NJ12            0.13385404 0.03890309  3.44070435 5.802021e-04
## D12             0.20442067 0.73989070  0.27628496 7.823292e-01
## D13            -0.04903292 0.59342882 -0.08262646 9.341486e-01
## D14             0.36873351 0.96892763  0.38055836 7.035310e-01
## D15            -0.44589452 0.95376183 -0.46751139 6.401340e-01
## sex2           -0.14913859 0.73288477 -0.20349527 8.387479e-01
## agecat          0.09479590 0.01658017  5.71742622 1.081496e-08
## education       0.06980129 0.03636391  1.91952088 5.491845e-02
## racethn2       -0.22609566 0.13100594 -1.72584276 8.437570e-02
```

```
## racethn3       -0.21148210 0.15845386 -1.33466044 1.819875e-01
## racethn4       -0.42650816 0.16637659 -2.56351058 1.036195e-02
## racethn5       -0.21491631 0.15527869 -1.38406831 1.663375e-01
## D82            -0.15357844 0.11867882 -1.29406781 1.956420e-01
## D83            -0.28340740 0.12597632 -2.24968795 2.446876e-02
## D84            -0.16883845 0.19579268 -0.86233282 3.885044e-01
```

```r
coef_table <- as.data.frame(coef(summary(ordinal_model_t_cutoff_no)))
print(coef_table)
```

```
##                   Estimate Std. Error     z value     Pr(>|z|)
## 1|2            -0.39680758 0.37484973 -1.05857775 2.897921e-01
## 2|3             0.69427366 0.37092287  1.87174672 6.124165e-02
## 3|4             2.16758598 0.37415602  5.79326775 6.902992e-09
## 4|5             3.24637336 0.37900307  8.56555961 1.075522e-17
## FrameNJ1       -0.06575747 0.11240237 -0.58501855 5.585352e-01
## FrameNJ2        0.03457993 0.11301228  0.30598384 7.596170e-01
## Cutoff_Income1  0.26861143 0.10851258  2.47539448 1.330891e-02
## NJ62           -0.16199565 0.13368282 -1.21179106 2.255924e-01
## NJ63            0.09854182 0.12244698  0.80477133 4.209516e-01
## NJ7             0.08913856 0.03806652  2.34165246 1.919858e-02
## NJ8             0.07458839 0.04440744  1.67963706 9.302795e-02
## NJ9             0.15058662 0.04686800  3.21299450 1.313588e-03
## NJ10_1          0.06324046 0.03813918  1.65814931 9.728733e-02
## NJ112           0.23629279 0.14833188  1.59300074 1.111600e-01
## NJ113          -0.33958382 0.18457971 -1.83976788 6.580232e-02
## NJ114           0.01978051 0.17344364  0.11404573 9.092015e-01
## NJ115           0.05317574 0.26080702  0.20388923 8.384401e-01
## NJ116           0.16704085 0.18225311  0.91653224 3.593878e-01
## NJ12            0.13385404 0.03890309  3.44070435 5.802021e-04
## D12             0.20442067 0.73989070  0.27628496 7.823292e-01
## D13            -0.04903292 0.59342882 -0.08262646 9.341486e-01
## D14             0.36873351 0.96892763  0.38055836 7.035310e-01
## D15            -0.44589452 0.95376183 -0.46751139 6.401340e-01
## sex2           -0.14913859 0.73288477 -0.20349527 8.387479e-01
## agecat          0.09479590 0.01658017  5.71742622 1.081496e-08
## education       0.06980129 0.03636391  1.91952088 5.491845e-02
## racethn2       -0.22609566 0.13100594 -1.72584276 8.437570e-02
## racethn3       -0.21148210 0.15845386 -1.33466044 1.819875e-01
## racethn4       -0.42650816 0.16637659 -2.56351058 1.036195e-02
## racethn5       -0.21491631 0.15527869 -1.38406831 1.663375e-01
## D82            -0.15357844 0.11867882 -1.29406781 1.956420e-01
## D83            -0.28340740 0.12597632 -2.24968795 2.446876e-02
## D84            -0.16883845 0.19579268 -0.86233282 3.885044e-01
```

```r
colnames(coef_table) <- c("Estimate", "Std. Error", "z value", "p value")




coef_table <- coef_table %>%
  mutate(`p value` = ifelse(`p value` < 0.0001, format(`p value`, scientific = TRUE, digits = 2),
                            round(`p value`, 4)))
```

```r
# Making a flextable
ft <- flextable(coef_table) %>%
  colformat_num(j = c("Estimate", "Std. Error", "z value"), digits = 4) %>%
  colformat_char(j = "p value") %>%
  autofit()

# Saving as a Word document
save_as_docx(ft, path = "ordinal_regression_results_t.docx")
```

Harris Ordinal Model

```r
coef(summary(ordinal_model_h))
```

```
##                Estimate Std. Error    z value     Pr(>|z|)
## 1|2         -0.12554270 0.33078065 -0.3795346 7.042909e-01
## 2|3          0.86602516 0.32903121  2.6320456 8.487247e-03
## 3|4          2.20740483 0.33183470  6.6521218 2.888975e-11
## 4|5          3.31981973 0.33599053  9.8806944 5.048186e-23
## FrameNJ1     0.05101759 0.09918127  0.5143873 6.069812e-01
## FrameNJ2    -0.08450296 0.09993683 -0.8455638 3.977961e-01
## NJ62        -0.21029960 0.11736563 -1.7918330 7.315973e-02
## NJ63        -0.06957866 0.11049011 -0.6297275 5.288729e-01
## NJ7         -0.04855047 0.03398811 -1.4284547 1.531610e-01
## NJ8          0.14879926 0.03970731  3.7474025 1.786752e-04
## NJ9          0.30532396 0.04163582  7.3332043 2.247139e-13
## NJ10_1      -0.03117555 0.03316477 -0.9400200 3.472073e-01
## NJ112        0.09212087 0.13332348  0.6909576 4.895922e-01
## NJ113       -0.22448133 0.16890944 -1.3290040 1.838466e-01
## NJ114        0.03407263 0.15723138  0.2167037 8.284392e-01
## NJ115        0.03205561 0.24302967  0.1319000 8.950634e-01
## NJ116        0.07866827 0.16282731  0.4831392 6.289969e-01
## NJ12         0.06109877 0.03498966  1.7461952 8.077704e-02
## D12         -0.79973638 0.69982799 -1.1427613 2.531377e-01
## D13         -0.76148616 0.50631129 -1.5039881 1.325844e-01
## D14         -0.76443850 0.85194314 -0.8972882 3.695652e-01
## D15          0.26236759 1.06345032  0.2467135 8.051299e-01
## sex2         0.55613302 0.69419601  0.8011181 4.230633e-01
## agecat       0.04696049 0.01463162  3.2095204 1.329566e-03
## education    0.11986942 0.03237305  3.7027531 2.132724e-04
## racethn2    -0.58164377 0.11596378 -5.0157365 5.283070e-07
## racethn3    -0.28606039 0.13928595 -2.0537634 3.999860e-02
## racethn4    -0.50306590 0.14760209 -3.4082573 6.537923e-04
## racethn5    -0.29604125 0.14141903 -2.0933623 3.631683e-02
## D5           0.03375356 0.01541549  2.1895869 2.855421e-02
## D82          0.65396485 0.10531997  6.2093149 5.321609e-10
## D83          0.49824027 0.10943631  4.5527876 5.293971e-06
## D84          0.49641757 0.18001499  2.7576457 5.821925e-03
```

```r
coef_table <- as.data.frame(coef(summary(ordinal_model_h)))
print(coef_table)
```

```
##                Estimate Std. Error    z value     Pr(>|z|)
## 1|2        -0.12554270 0.33078065 -0.3795346 7.042909e-01
## 2|3         0.86602516 0.32903121  2.6320456 8.487247e-03
## 3|4         2.20740483 0.33183470  6.6521218 2.888975e-11
## 4|5         3.31981973 0.33599053  9.8806944 5.048186e-23
## FrameNJ1    0.05101759 0.09918127  0.5143873 6.069812e-01
## FrameNJ2   -0.08450296 0.09993683 -0.8455638 3.977961e-01
## NJ62       -0.21029960 0.11736563 -1.7918330 7.315973e-02
## NJ63       -0.06957866 0.11049011 -0.6297275 5.288729e-01
## NJ7        -0.04855047 0.03398811 -1.4284547 1.531610e-01
## NJ8         0.14879926 0.03970731  3.7474025 1.786752e-04
## NJ9         0.30532396 0.04163582  7.3332043 2.247139e-13
## NJ10_1     -0.03117555 0.03316477 -0.9400200 3.472073e-01
## NJ112       0.09212087 0.13332348  0.6909576 4.895922e-01
## NJ113      -0.22448133 0.16890944 -1.3290040 1.838466e-01
## NJ114       0.03407263 0.15723138  0.2167037 8.284392e-01
## NJ115       0.03205561 0.24302967  0.1319000 8.950634e-01
## NJ116       0.07866827 0.16282731  0.4831392 6.289969e-01
## NJ12        0.06109877 0.03498966  1.7461952 8.077704e-02
## D12        -0.79973638 0.69982799 -1.1427613 2.531377e-01
## D13        -0.76148616 0.50631129 -1.5039881 1.325844e-01
## D14        -0.76443850 0.85194314 -0.8972882 3.695652e-01
## D15         0.26236759 1.06345032  0.2467135 8.051299e-01
## sex2        0.55613302 0.69419601  0.8011181 4.230633e-01
## agecat      0.04696049 0.01463162  3.2095204 1.329566e-03
## education   0.11986942 0.03237305  3.7027531 2.132724e-04
## racethn2   -0.58164377 0.11596378 -5.0157365 5.283070e-07
## racethn3   -0.28606039 0.13928595 -2.0537634 3.999860e-02
## racethn4   -0.50306590 0.14760209 -3.4082573 6.537923e-04
## racethn5   -0.29604125 0.14141903 -2.0933623 3.631683e-02
## D5          0.03375356 0.01541549  2.1895869 2.855421e-02
## D82         0.65396485 0.10531997  6.2093149 5.321609e-10
## D83         0.49824027 0.10943631  4.5527876 5.293971e-06
## D84         0.49641757 0.18001499  2.7576457 5.821925e-03
```

```r
colnames(coef_table) <- c("Estimate", "Std. Error", "z value", "p value")




coef_table <- coef_table %>%
  mutate(`p value` = ifelse(`p value` < 0.0001, format(`p value`, scientific = TRUE, digits = 2),
                            round(`p value`, 4)))

# Making a flextable
ft <- flextable(coef_table) %>%
  colformat_num(j = c("Estimate", "Std. Error", "z value"), digits = 4) %>%
  colformat_char(j = "p value") %>%
  autofit()

# Saving as Word document
save_as_docx(ft, path = "ordinal_regression_results_h.docx")
```

Exporting multinomial regression to word

```r
# Making a flextable
flex_table <- flextable(results_table) %>%
  theme_vanilla() %>%
  autofit()

# Save as Word document
doc <- read_docx() %>%
  body_add_flextable(flex_table) %>%
  body_add_par(" ")

print(doc, target = "Multinomial_Regression_Results_2.docx")

# Showing the results
flex_table
```

```
## Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘ engine is
## used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning by using the
## ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a compatible engine
## by defining ‘latex_engine: xelatex‘ in the YAML header of the R Markdown
## document.
```

Outcome	Variable	Coef.	Std. Err.	Z	P>|z|	[
2	(Intercept)	-3.5123	0.7645	-4.5941	4.3e-06	
2	NJ42	0.1779	0.3923	0.4535	0.6502	
2	NJ43	0.2628	0.3570	0.7362	0.4616	
2	NJ44	-0.4176	0.3545	-1.1781	0.2388	
2	NJ45	-0.9183	0.3421	-2.6844	0.0073	
2	NJ52	-0.8580	0.3656	-2.3465	0.019	
2	NJ53	-0.5409	0.3226	-1.6769	0.0936	
2	NJ54	-0.3231	0.3149	-1.0260	0.3049	
2	NJ55	0.6426	0.3087	2.0818	0.0374	
2	NJ62	-0.3027	0.2494	-1.2139	0.2248	
2	NJ63	0.0949	0.2448	0.3878	0.6982	
2	NJ7	-0.0599	0.0678	-0.8837	0.3769	
2	NJ8	0.0298	0.0791	0.3769	0.7062	
2	NJ9	0.0732	0.0876	0.8352	0.4036	
2	NJ10_1	-0.0043	0.0677	-0.0634	0.9494	
2	NJ112	-0.0741	0.2627	-0.2819	0.778	
2	NJ113	-0.3643	0.3366	-1.0821	0.2792	
2	NJ114	-0.3969	0.3188	-1.2453	0.213	

Outcome	Variable	Coef.	Std. Err.	Z	P>|z|	[
2	NJ115	0.2964	0.4907	0.6040	0.5459	
2	NJ116	0.2672	0.3663	0.7295	0.4657	
2	NJ12	0.0512	0.0701	0.7297	0.4656	
2	D12	-1.1413	1.8956	-0.6021	0.5471	
2	D13	-1.0868	1.3606	-0.7987	0.4244	
2	D14	-24.7007	0.0000	-37,437,845,768.2020	0.0e+00	
2	D15	7.5354	0.0001	76,278.9701	0.0e+00	
2	sex2	1.6266	1.8864	0.8623	0.3885	
2	agecat	0.0106	0.0309	0.3422	0.7322	
2	education	0.1970	0.0654	3.0139	0.0026	
2	as.factor(racethn)2	1.0587	0.2494	4.2448	2.2e-05	
2	as.factor(racethn)3	-0.1799	0.2694	-0.6678	0.5043	
2	as.factor(racethn)4	0.1456	0.3337	0.4364	0.6626	
2	as.factor(racethn)5	0.1972	0.2988	0.6601	0.5092	
2	D5	-0.0023	0.0327	-0.0706	0.9437	
2	D82	4.7233	0.2393	19.7381	1.0e-86	
2	D83	2.4404	0.2286	10.6746	1.3e-26	
2	D84	2.6861	0.4273	6.2859	3.3e-10	
3	(Intercept)	-5.1330	1.6896	-3.0379	0.0024	
3	NJ42	-0.8874	0.7536	-1.1775	0.239	
3	NJ43	-0.7430	0.6490	-1.1448	0.2523	
3	NJ44	-0.8757	0.6277	-1.3952	0.163	
3	NJ45	-1.2878	0.6213	-2.0726	0.0382	
3	NJ52	0.9206	0.9537	0.9653	0.3344	
3	NJ53	1.1284	0.8975	1.2573	0.2086	
3	NJ54	1.3163	0.8760	1.5027	0.1329	
3	NJ55	1.1151	0.8899	1.2530	0.2102	
3	NJ62	0.6428	0.4669	1.3767	0.1686	
3	NJ63	0.4797	0.4627	1.0367	0.2999	
3	NJ7	-0.0243	0.1412	-0.1718	0.8636	
3	NJ8	-0.3542	0.1969	-1.7989	0.072	
3	NJ9	-0.1533	0.1824	-0.8407	0.4005	
3	NJ10_1	-0.0633	0.1314	-0.4821	0.6298	
3	NJ112	-0.4295	0.4616	-0.9304	0.3521	
3	NJ113	-1.7479	0.8946	-1.9538	0.0507	

Outcome	Variable	Coef.	Std. Err.	Z	P>|z|	[
3	NJ114	-0.6786	0.6070	-1.1179	0.2636	
3	NJ115	0.4898	0.9273	0.5283	0.5973	
3	NJ116	-0.3928	0.6718	-0.5847	0.5588	
3	NJ12	-0.2132	0.1447	-1.4738	0.1405	
3	D12	-0.5961	2.2206	-0.2684	0.7884	
3	D13	1.8127	1.9696	0.9204	0.3574	
3	D14	0.9748	1.9842	0.4913	0.6232	
3	D15	-1.1439	0.0000	-177,365.5406	0.0e+00	
3	sex2	1.2953	2.1970	0.5896	0.5555	
3	agecat	0.1680	0.0668	2.5138	0.0119	
3	education	0.1812	0.1333	1.3594	0.174	
3	as.factor(racethn)2	0.5048	0.5469	0.9230	0.356	
3	as.factor(racethn)3	0.1320	0.5819	0.2268	0.8206	
3	as.factor(racethn)4	0.1335	0.6073	0.2198	0.826	
3	as.factor(racethn)5	0.6319	0.5261	1.2011	0.2297	
3	D5	0.0637	0.0636	1.0017	0.3165	
3	D82	2.4699	0.6663	3.7070	2e-04	
3	D83	3.0315	0.5543	5.4687	4.5e-08	
3	D84	4.4060	0.7280	6.0523	1.4e-09	