replication_study3_final

Asad Tariq, Isabella Mullen

2025-03-14

Study 3

Loading Data

```
s3 <- import("../original_materials/data/Study3.sav")
```

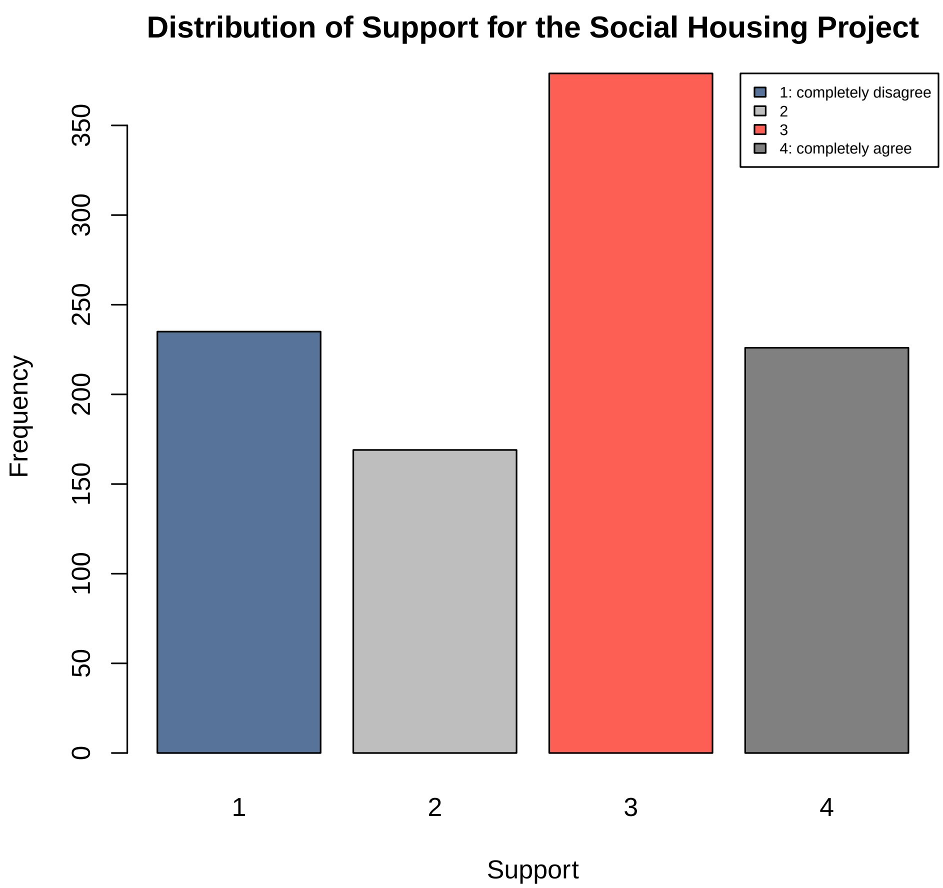
Data prep

```
s3$group = factor(s3$SKUPINA, levels = c(1, 2,
    3, 4), labels = c("Control", "Equality", "Proportionality",
    "Need"))
s3$SEX <- factor(s3$SEX, levels = c(1, 2), labels = c("Male",
    "Female"))
s3$AGECAT = factor(s3$AGECAT, levels = c(1, 2,
    3, 4, 5, 6), labels = c("18-24", "25-34",
    "35-44", "45-54", "55-64", "65+"))
s3$EDU = factor(s3$EDU, levels = c(1, 2, 3, 4),
    labels = c("Primary", "Secondary (no diploma)",
        "Secondary (complete)", "University"))
s3$SIZE = factor(s3$SIZE, levels = c(1, 2, 3,
    4, 5), labels = c("less than 1k", "1k-4 999",
    "5k-19 999", "20k - 99 999", "100k+"))
s3$REG = factor(s3$REG, levels = c(1, 2, 3, 4,
    5, 6, 7, 8), labels = c("Bratislavsky", "Trnavsky",
    "Trenciansky", "Nitriansky", "Zilinsky", "Banskobystricky",
    "Presovsky", "Kosicky"))


s3$income <- car::recode(s3$PINCOME, "'1'='below median';'2'='below median';'8'='NA';
                         '9'='NA'; else = 'above median'")
```

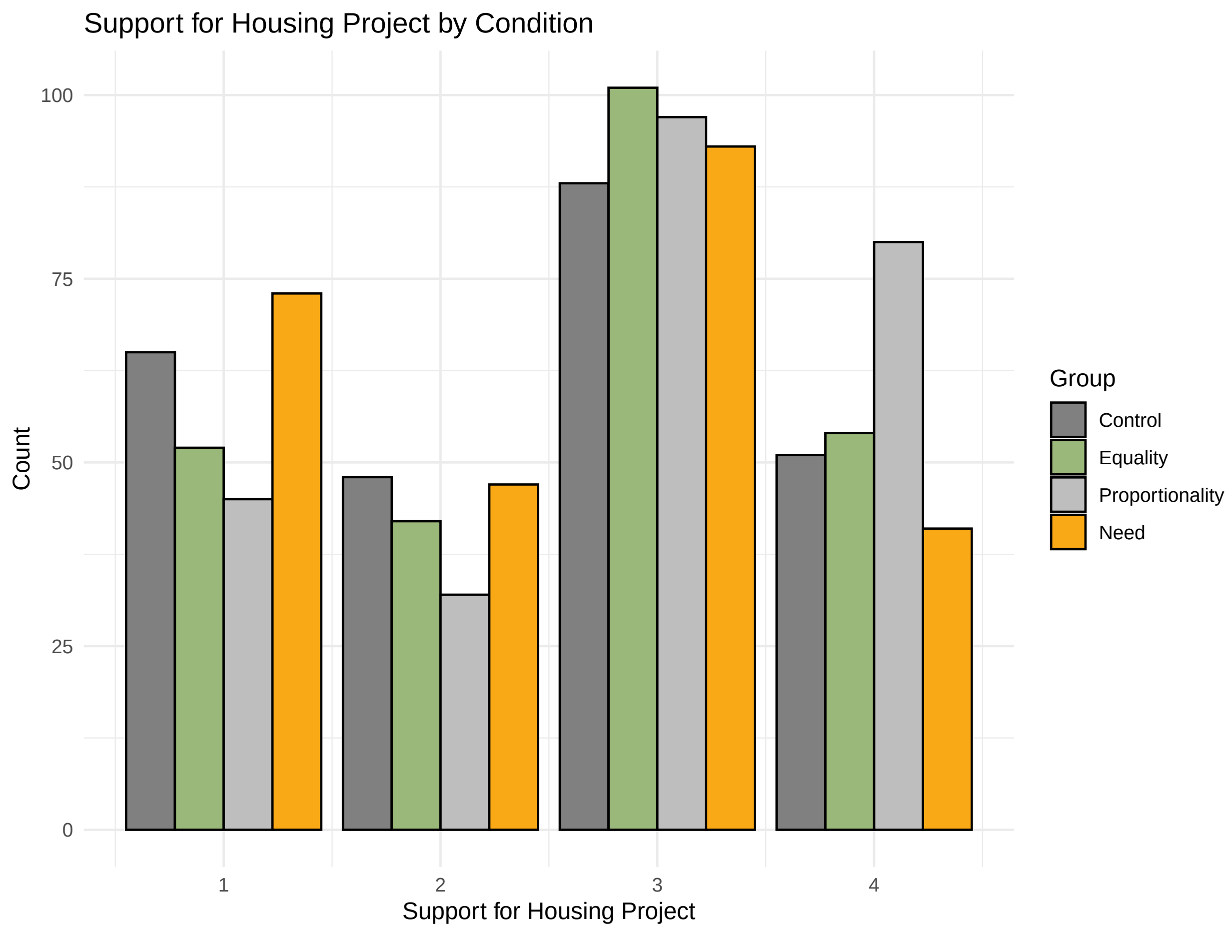
Visualization of Dependent Variable

```
counts <- table(s3$E3)
par(mar = c(5, 4, 3, 10))

barplot(counts, col = c("#577399", "grey", "#FE5F55",
    "#808080"), main = "Distribution of Support for the Social Housing Project",
    xlab = "Support", ylab = "Frequency", border = "black")

legend_labels <- c("1: completely disagree", "2",
```

```
    "3", "4: completely agree")

legend("topright", legend = legend_labels, fill = c("#577399",
    "grey", "#FE5F55", "#808080"), xpd = TRUE,
    c(-0.15, 0), cex = 0.6)
```




```
ggplot(s3, aes(x = E3, fill = group)) + geom_bar(position = "dodge",
    color = "black") + labs(title = "Support for Housing Project by Condition",
    x = "Support for Housing Project", y = "Count",
    fill = "Group") + scale_fill_manual(values = c("#808080",
    "#9AB87A", "grey", "#FAA916")) + theme_minimal()
```



Descriptives

Main outcomes

```
# group (Control, Equality, Proportionality,
# Need) and summary statistics for E3
# (Agreement with the statement |
# Construction of an apartment building)
s3 %>%
    dplyr::group_by(group) %>%
    dplyr::summarise(N = length(E3), Min = min(E3,
        na.rm = TRUE), Q1 = quantile(E3, probs = 0.25,
        na.rm = TRUE), Median = median(E3, na.rm = TRUE),
        Q3 = quantile(E3, probs = 0.75, na.rm = TRUE),
        Max = max(E3, na.rm = TRUE), Mean = mean(E3,
            na.rm = TRUE), SD = sd(E3, na.rm = TRUE),
        Skew = skewness(E3, na.rm = TRUE), Kurtosis = kurtosis(E3,
            na.rm = TRUE)) -> s3_personal

# Display summary statistics in a table
# (table A19 from paper)
knitr::kable(s3_personal, caption = "Study 3 - personal agreement") %>%
    kable_styling(full_width = F)
```

Table 1: Study 3 - personal agreement

group	N	Min	Q1	Median	Q3	Max	Mean	SD	Skew	Kurtosis
Control	252	1	1	3	3	4	2.50	1.08	-0.121	1.72
Equality	249	1	2	3	3	4	2.63	1.04	-0.328	1.93
Proportionality	254	1	2	3	4	4	2.83	1.06	-0.561	2.10
Need	254	1	1	3	3	4	2.40	1.07	-0.054	1.71

Original Model

```r
s3$E1 <- as.ordered(s3$E1)
s3$E2 <- as.ordered(s3$E2)
s3$E3 <- as.ordered(s3$E3)
s3$E4 <- as.ordered(s3$E4)

s3personal <- polr(E3 ~ group, data = s3, Hess = TRUE)
summary(s3personal)
```

```
## Call:
## polr(formula = E3 ~ group, data = s3, Hess = TRUE)
##
## Coefficients:
##                        Value Std. Error t value
## groupEquality          0.220      0.161   1.366
## groupProportionality   0.604      0.163   3.696
## groupNeed             -0.159      0.160  -0.989
##
## Intercepts:
##     Value  Std. Error t value
## 1|2 -1.050  0.123     -8.570
## 2|3 -0.250  0.118     -2.124
## 3|4  1.430  0.127     11.264
##
## Residual Deviance: 2682.53
## AIC: 2694.53
```

```r
ctable <- coef(summary(s3personal))
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) *
    2
(ctable <- cbind(ctable, `p value` = p))
```

```
##                        Value Std. Error t value
## groupEquality          0.220      0.161   1.366
## groupProportionality   0.604      0.163   3.696
## groupNeed             -0.159      0.160  -0.989
## 1|2                   -1.050      0.123  -8.570
## 2|3                   -0.250      0.118  -2.124
## 3|4                    1.430      0.127  11.264
##                                               p value
## groupEquality         0.171982948784216854720696687763 8
## groupProportionality 0.000218989805670838095629643360 6
## groupNeed             0.322501174203670082363260007696 2
## 1|2                   0.000000000000000010325397153651 2
```

```
## 2|3                    0.0336704264404697958812384683824
## 3|4                    0.0000000000000000000000000000199
```

```r
ci <- confint(s3personal)
```

```
## Waiting for profiling to be done...
```

```r
exp(cbind(OR = coef(s3personal), ci))
```

```
##                           OR 2.5 % 97.5 %
## groupEquality          1.246 0.909   1.71
## groupProportionality 1.829 1.329   2.52
## groupNeed              0.853 0.623   1.17
```

Alternate Model

```r
# We present an alternate model with a
# changed linear predictor
s3personal_alter <- polr(E3 ~ group + AGE + EDU +
    SEX + REG, data = s3, Hess = TRUE)
summary(s3personal_alter)
```

```
## Call:
## polr(formula = E3 ~ group + AGE + EDU + SEX + REG, data = s3,
##     Hess = TRUE)
##
## Coefficients:
##                                Value Std. Error t value
## groupEquality                0.21815    0.16296  1.3387
## groupProportionality         0.58589    0.16527  3.5451
## groupNeed                   -0.16201    0.16406 -0.9875
## AGE                          0.01363    0.00378  3.6083
## EDUSecondary (no diploma) -0.49254    0.21862 -2.2530
## EDUSecondary (complete)    0.22175    0.22146  1.0013
## EDUUniversity                0.47119    0.24168  1.9497
## SEXFemale                   -0.07541    0.11915 -0.6329
## REGTrnavsky                  0.12574    0.25362  0.4958
## REGTrenciansky               0.13763    0.24518  0.5613
## REGNitriansky               -0.40713    0.23840 -1.7077
## REGZilinsky                  0.34434    0.24199  1.4229
## REGBanskobystricky           0.13928    0.24494  0.5686
## REGPresovsky                -0.00284    0.23286 -0.0122
## REGKosicky                  -0.19348    0.23411 -0.8264
##
## Intercepts:
##     Value  Std. Error t value
## 1|2 -0.553  0.323     -1.710
## 2|3  0.287  0.323      0.889
## 3|4  2.066  0.330      6.255
##
## Residual Deviance: 2608.39
## AIC: 2644.39
```

```r
ctable <- coef(summary(s3personal_alter))
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) *
    2
```

```r
(ctable <- cbind(ctable, `p value` = p))
```

```
##                              Value Std. Error t value        p value
## groupEquality              0.21815    0.16296  1.3387 0.180664070460
## groupProportionality       0.58589    0.16527  3.5451 0.000392446561
## groupNeed                 -0.16201    0.16406 -0.9875 0.323398578360
## AGE                        0.01363    0.00378  3.6083 0.000308242600
## EDUSecondary (no diploma) -0.49254    0.21862 -2.2530 0.024260258374
## EDUSecondary (complete)    0.22175    0.22146  1.0013 0.316665545311
## EDUUniversity              0.47119    0.24168  1.9497 0.051216193507
## SEXFemale                 -0.07541    0.11915 -0.6329 0.526810240536
## REGTrnavsky                0.12574    0.25362  0.4958 0.620045839450
## REGTrenciansky             0.13763    0.24518  0.5613 0.574569872609
## REGNitriansky             -0.40713    0.23840 -1.7077 0.087684943085
## REGZilinsky                0.34434    0.24199  1.4229 0.154759332143
## REGBanskobystricky         0.13928    0.24494  0.5686 0.569614774917
## REGPresovsky              -0.00284    0.23286 -0.0122 0.990263885377
## REGKosicky                -0.19348    0.23411 -0.8264 0.408549111872
## 1|2                       -0.55262    0.32326 -1.7095 0.087350020191
## 2|3                        0.28715    0.32288  0.8893 0.373819487752
## 3|4                        2.06574    0.33027  6.2547 0.000000000398
```

```r
ci <- confint(s3personal_alter)
```

```
## Waiting for profiling to be done...
```

```r
exp(cbind(OR = coef(s3personal_alter), ci))
```

```
##                              OR 2.5 % 97.5 %
## groupEquality             1.244 0.904  1.712
## groupProportionality      1.797 1.300  2.486
## groupNeed                 0.850 0.616  1.173
## AGE                       1.014 1.006  1.021
## EDUSecondary (no diploma) 0.611 0.398  0.937
## EDUSecondary (complete)   1.248 0.808  1.927
## EDUUniversity             1.602 0.997  2.573
## SEXFemale                 0.927 0.734  1.171
## REGTrnavsky               1.134 0.690  1.865
## REGTrenciansky            1.148 0.710  1.856
## REGNitriansky             0.666 0.417  1.062
## REGZilinsky               1.411 0.878  2.269
## REGBanskobystricky        1.149 0.711  1.859
## REGPresovsky              0.997 0.632  1.574
## REGKosicky                0.824 0.520  1.304
```

General Good-ness of Fit

```r
# This test is an indicator of whether we
# have used an appropriate model to explain
# the data. The p-value > 0.05 confirms for
# us a good fit for our model.
print(lipsitz.test(s3personal))
```

```
##
##  Lipsitz goodness of fit test for ordinal response models
```

```
##
## data:  formula:  E3 ~ group
## LR statistic = 7, df = 9, p-value = 0.7
```

```r
print(lipsitz.test(s3personal_alter))
```

```
##
##  Lipsitz goodness of fit test for ordinal response models
##
## data:  formula:  E3 ~ group + AGE + EDU + SEX + REG
## LR statistic = 10, df = 9, p-value = 0.3
```

In-sample Predictive Performance

Confusion Matrix and Accuracy

```r
# Predicted classes
pred_original <- predict(s3personal, type = "class")
pred_alternate <- predict(s3personal_alter, type = "class")

# Confusion matrices
table(pred_original, s3$E3)
```

```
##
## pred_original   1   2   3   4
##             1   0   0   0   0
##             2   0   0   0   0
##             3 235 169 379 226
##             4   0   0   0   0
```

```r
table(pred_alternate, s3$E3)
```

```
##
## pred_alternate   1   2   3   4
##              1  63  38  57  18
##              2   0   0   0   0
##              3 167 130 300 179
##              4   5   1  22  29
```

```r
# Accuracy calculation
mean(pred_original == factor(s3$E3, ordered = FALSE))
```

```
## [1] 0.376
```

```r
mean(pred_alternate == factor(s3$E3, ordered = FALSE))
```

```
## [1] 0.389
```

ROC Curve

```r
# Binary outcome 1: E3 = 1 vs Others
s3$E3_bin1 <- ifelse(s3$E3 == levels(s3$E3)[1],
    1, 0)

# Binary outcome 2: E3 <= 2 vs E3 >= 3
s3$E3_bin2 <- ifelse(s3$E3 %in% levels(s3$E3)[1:2],
    1, 0)
```

```r
# Binary outcome 3: E3 <= 3 vs E3 = 4
s3$E3_bin3 <- ifelse(s3$E3 %in% levels(s3$E3)[1:3],
    1, 0)

# Probabilities for each class
prob_logit <- predict(s3personal, data = s3, type = "prob")
prob_alter <- predict(s3personal_alter, data = s3,
    type = "prob")
```

```r
# ROC for predicting E3 = 1
roc_1_orig <- roc(s3$E3_bin1, prob_logit[, 1])
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

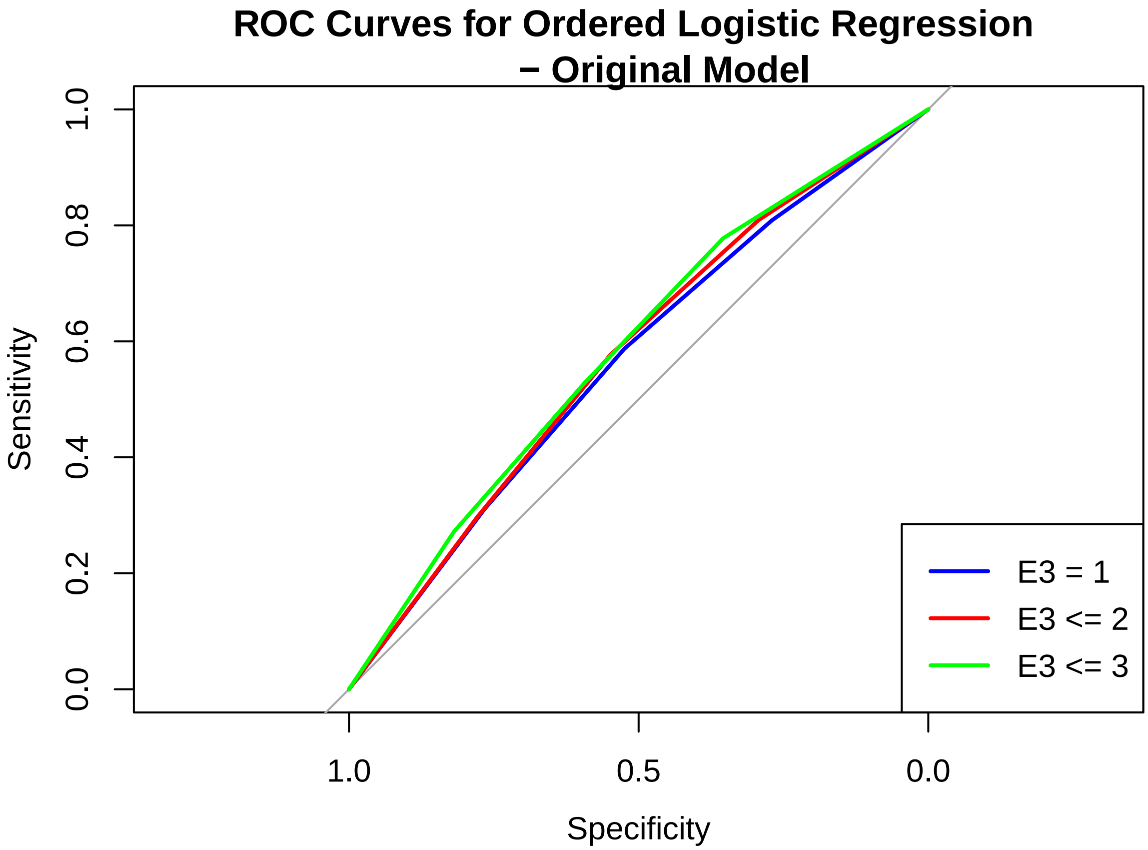
```r
# ROC for predicting E3 <= 2
roc_2_orig <- roc(s3$E3_bin2, prob_logit[, 1] +
    prob_logit[, 2])
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
# ROC for predicting E3 <= 3
roc_3_orig <- roc(s3$E3_bin3, prob_logit[, 1] +
    prob_logit[, 2] + prob_logit[, 3])
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
plot(roc_1_orig, col = "blue", main = "ROC Curves for Ordered Logistic Regression
     - Original Model")
plot(roc_2_orig, col = "red", add = TRUE)
plot(roc_3_orig, col = "green", add = TRUE)

legend("bottomright", legend = c("E3 = 1", "E3 <= 2",
    "E3 <= 3"), col = c("blue", "red", "green"),
    lwd = 2)
```




```r
# ROC for predicting E3 = 1
roc_1_alter <- roc(s3$E3_bin1, prob_alter[, 1])
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

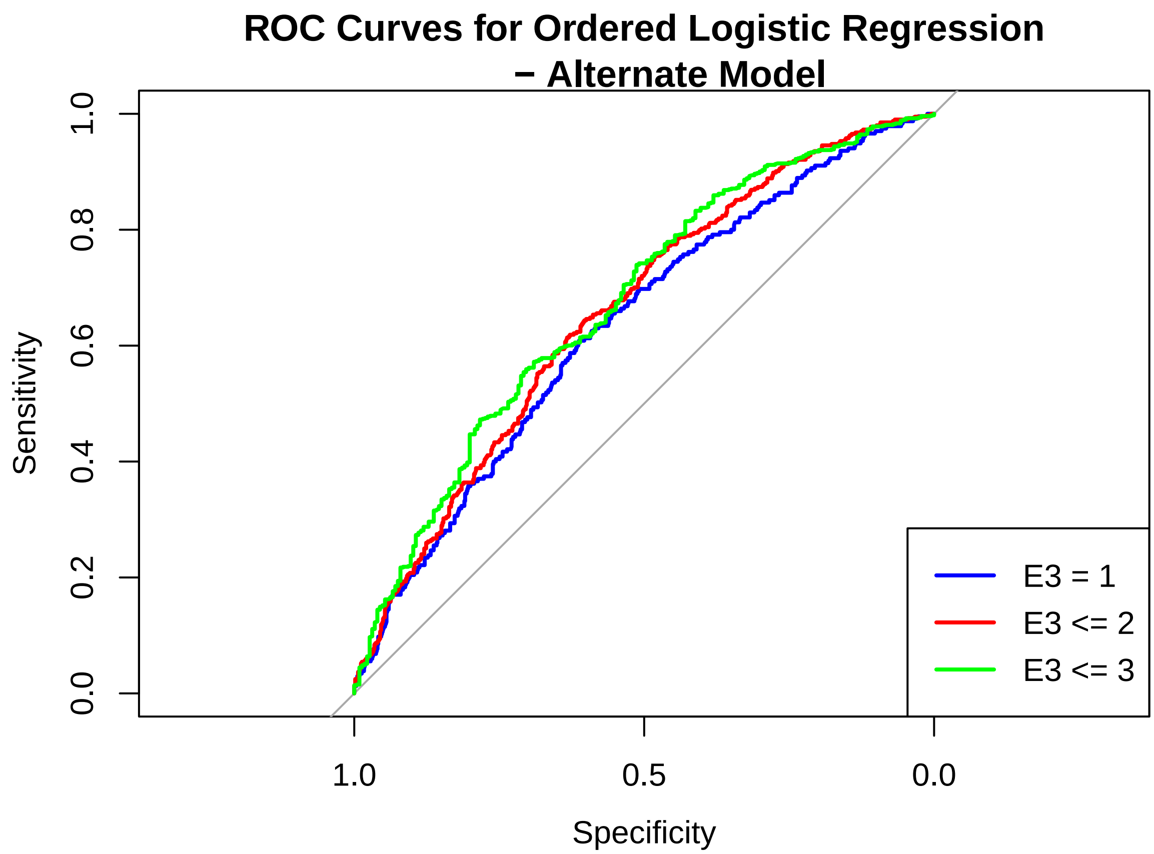
```r
# ROC for predicting E3 <= 2
roc_2_alter <- roc(s3$E3_bin2, prob_alter[, 1] +
    prob_alter[, 2])
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
# ROC for predicting E3 <= 3
roc_3_alter <- roc(s3$E3_bin3, prob_alter[, 1] +
    prob_alter[, 2] + prob_alter[, 3])
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
plot(roc_1_alter, col = "blue", main = "ROC Curves for Ordered Logistic Regression
     - Alternate Model")
plot(roc_2_alter, col = "red", add = TRUE)
plot(roc_3_alter, col = "green", add = TRUE)

legend("bottomright", legend = c("E3 = 1", "E3 <= 2",
    "E3 <= 3"), col = c("blue", "red", "green"),
    lwd = 2)
```



Log-likelihood Ratio Test, AIC, and BIC

```
s3personal$AIC <- round(AIC(s3personal), 1)
s3personal$BIC <- round(BIC(s3personal), 1)

s3personal_alter$AIC <- round(AIC(s3personal_alter),
    1)
s3personal_alter$BIC <- round(BIC(s3personal_alter),
    1)

# Perform the likelihood ratio test
lr_test <- lrtest(s3personal, s3personal_alter)

# Extract log-likelihood and p-value
logLik_value_orig <- as.numeric(logLik(s3personal))
logLik_value_alter <- as.numeric(logLik(s3personal_alter))
p_value <- lr_test$`Pr(>Chisq)`[2]  # Extract p-value from second row

stargazer::stargazer(s3personal, s3personal_alter,
    type = "text", keep.stat = c("bic", "aic",
        "n"), add.lines = list(c("Log-Likelihood",
        round(logLik_value_orig, 2), paste0(round(logLik_value_alter,
            2), " (", signif(p_value, 3), ")"))))
```

```
##
## ============================================================
```

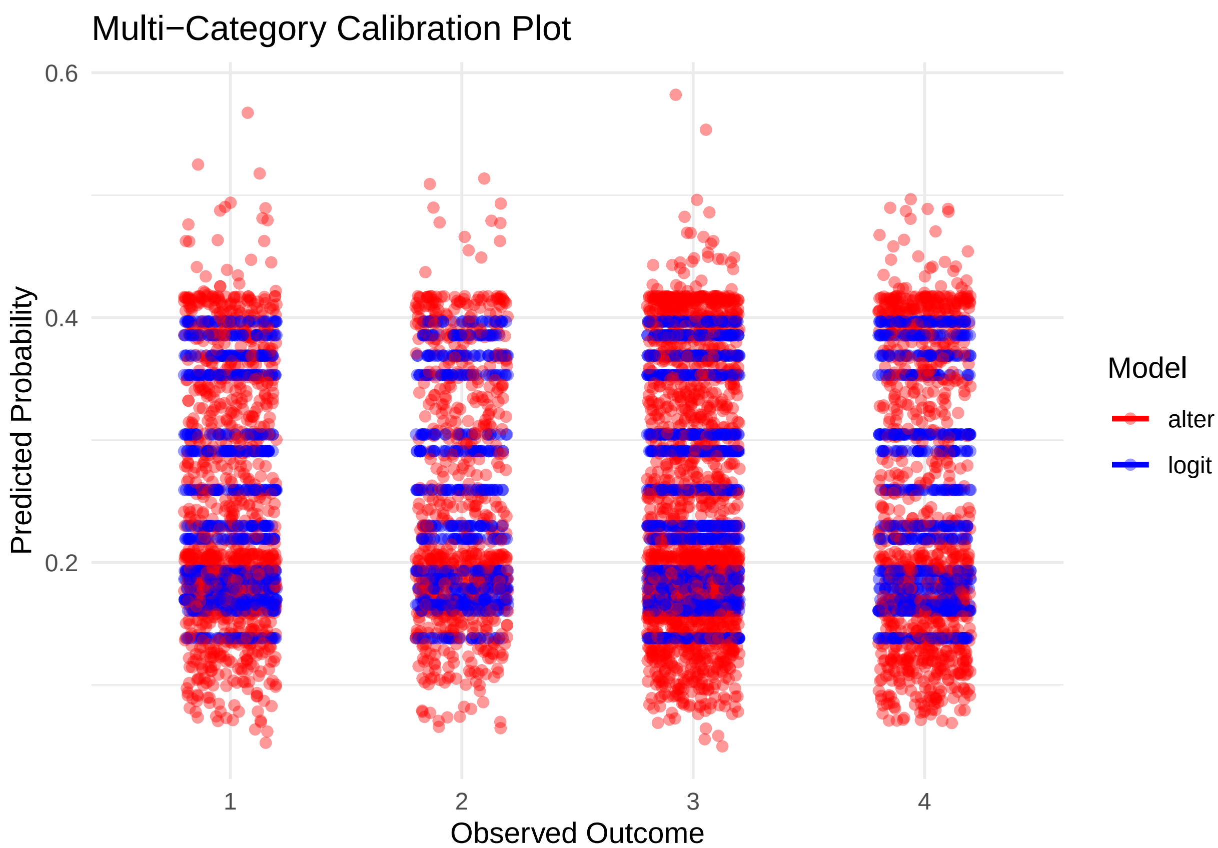
```
##                                   Dependent variable:
##                           -----------------------------------
##                                           E3
##                              (1)                (2)
## -------------------------------------------------------------
## groupEquality              0.220              0.218
##                           (0.161)            (0.163)
##
## groupProportionality      0.604***           0.586***
##                           (0.163)            (0.165)
##
## groupNeed                 -0.159             -0.162
##                           (0.160)            (0.164)
##
## AGE                                          0.014***
##                                              (0.004)
##
## EDUSecondary (no diploma)                    -0.493**
##                                              (0.219)
##
## EDUSecondary (complete)                       0.222
##                                              (0.221)
##
## EDUUniversity                                 0.471*
##                                              (0.242)
##
## SEXFemale                                    -0.075
##                                              (0.119)
##
## REGTrnavsky                                   0.126
##                                              (0.254)
##
## REGTrenciansky                                0.138
##                                              (0.245)
##
## REGNitriansky                                -0.407*
##                                              (0.238)
##
## REGZilinsky                                   0.344
##                                              (0.242)
##
## REGBanskobystricky                            0.139
##                                              (0.245)
##
## REGPresovsky                                 -0.003
##                                              (0.233)
##
## REGKosicky                                   -0.193
##                                              (0.234)
##
## -------------------------------------------------------------
## Log-Likelihood            -1341.27  -1304.2 (0.0000000000533)
## Observations               1,009              1,009
## Akaike Inf. Crit.         2,694.000          2,644.000
```

```
## Bayesian Inf. Crit.       2,724.000        2,733.000
## ==============================================================
## Note:                              *p<0.1; **p<0.05; ***p<0.01
```

Observed vs. Predicted Probability Plot

```r
# Reshaping the Data to Long Format
calibration_data_long <- data.frame(Observed = as.factor(s3$E3),
    prob_logit_1 = prob_logit[, 1], prob_logit_2 = prob_logit[,
        2], prob_logit_3 = prob_logit[, 3], prob_logit = prob_logit[,
        4], prob_alter_1 = prob_alter[, 1], prob_alter_2 = prob_alter[,
        2], prob_alter_3 = prob_alter[, 3], prob_alter = prob_alter[,
        4]) %>%
    pivot_longer(cols = -Observed, names_to = c("Model",
        "Category"), names_pattern = "prob_(logit|alter)_?(\\d*)",
        values_to = "Predicted_Probability") %>%
    mutate(Category = as.factor(Category))  # Ensure category is treated as a factor

# Plot
ggplot(calibration_data_long, aes(x = Observed,
    y = Predicted_Probability, color = Model)) +
    geom_point(position = position_jitter(width = 0.2,
        height = 0), alpha = 0.4) + geom_smooth(method = "loess",
    se = FALSE) + labs(title = "Multi-Category Calibration Plot",
    x = "Observed Outcome", y = "Predicted Probability") +
    scale_color_manual(values = c(logit = "blue",
        alter = "red")) + theme_minimal()
```

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



Out-of-Sample Predictive Performance

K-fold Cross-Validation

```
# Cross-validation control
ctrl <- trainControl(method = "cv", number = 10)

# Fit models with cross-validation
logit_cv <- train(E3 ~ group, data = s3, method = "polr",
    trControl = ctrl)
alter_cv <- train(E3 ~ group + AGE + SEX + EDU +
    REG, data = s3, method = "polr", trControl = ctrl)

# Extract results
logit_results <- logit_cv$results
alter_results <- alter_cv$results

# Add final model details
logit_results$Final_Model <- logit_cv$finalModel$method
alter_results$Final_Model <- alter_cv$finalModel$method

# Format results for stargazer
logit_summary <- data.frame(Model = "Original",
    Method = logit_results$method, Accuracy = round(logit_results$Accuracy,
        3), Kappa = round(logit_results$Kappa,
        3), Final_Model = logit_cv$finalModel$method)
```

```r
alter_summary <- data.frame(Model = "Alternative",
    Method = alter_results$method, Accuracy = round(alter_results$Accuracy,
        3), Kappa = round(alter_results$Kappa,
        3), Final_Model = alter_cv$finalModel$method)

# Combine results
cv_results <- rbind(logit_summary, alter_summary)

# Display using stargazer
library(stargazer)
```

```
##
## Please cite as:
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```r
stargazer(cv_results, summary = FALSE, title = "Cross-Validation Results",
    type = "text")
```

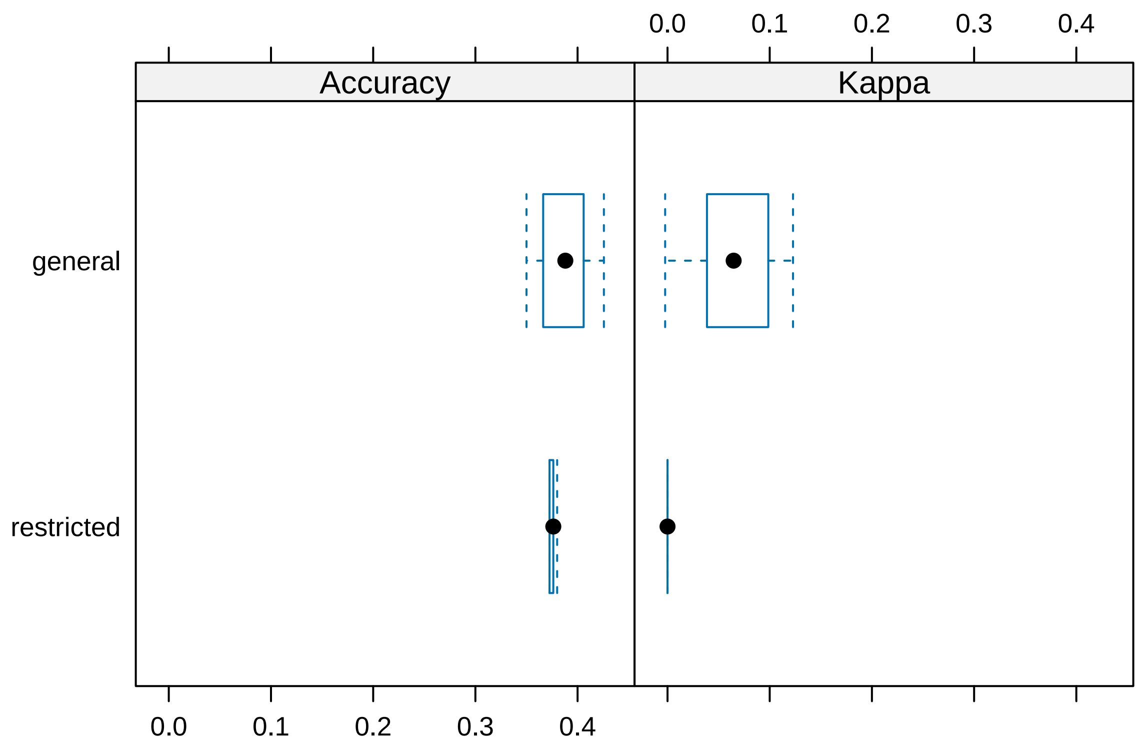
```
##
## Cross-Validation Results
## ==================================================
##       Model     Method  Accuracy Kappa Final_Model
## --------------------------------------------------
## 1   Original   cauchit   0.376     0     cauchit
## 2   Original   cloglog   0.376     0     cauchit
## 3   Original   logistic  0.376     0     cauchit
## 4   Original    loglog   0.376     0     cauchit
## 5   Original    probit   0.376     0     cauchit
## 6  Alternative cauchit   0.383   0.062   cloglog
## 7  Alternative cloglog   0.387   0.066   cloglog
## 8  Alternative logistic  0.384   0.064   cloglog
## 9  Alternative  loglog   0.372   0.041   cloglog
## 10 Alternative  probit   0.384   0.061   cloglog
## --------------------------------------------------
```

```r
# Compare resampling results
resamples <- resamples(list(general = alter_cv,
    restricted = logit_cv))
summary(resamples)
```

```
##
## Call:
## summary.resamples(object = resamples)
##
## Models: general, restricted
## Number of resamples: 10
##
## Accuracy
##             Min. 1st Qu. Median  Mean 3rd Qu.  Max. NA's
## general    0.350   0.369  0.388 0.387   0.404 0.426    0
## restricted 0.373   0.373  0.376 0.376   0.376 0.380    0
##
## Kappa
```

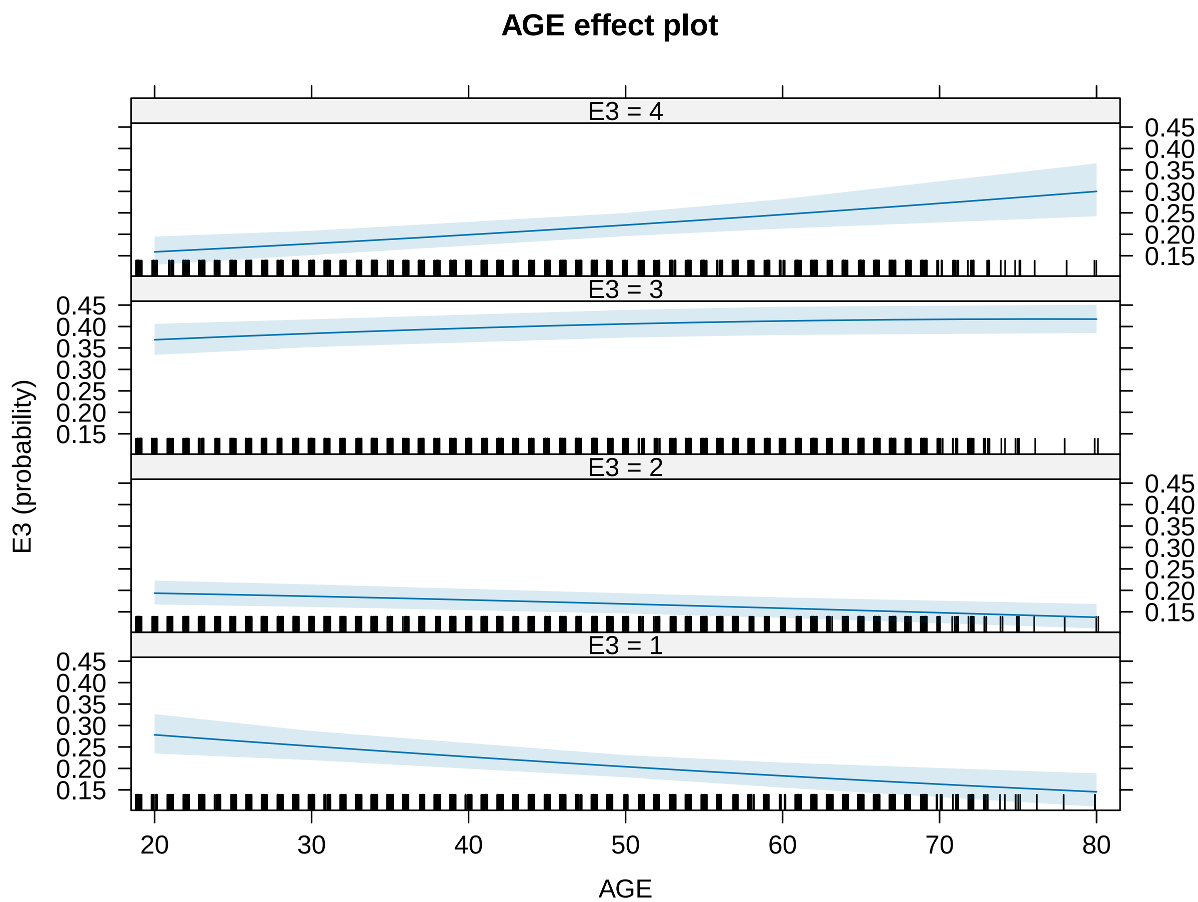
```
##                  Min. 1st Qu. Median   Mean 3rd Qu.  Max. NA's
## general    -0.00231  0.0409 0.0647 0.0657  0.0966 0.123    0
## restricted  0.00000  0.0000 0.0000 0.0000  0.0000 0.000    0
```

```r
bwplot(resamples)
```

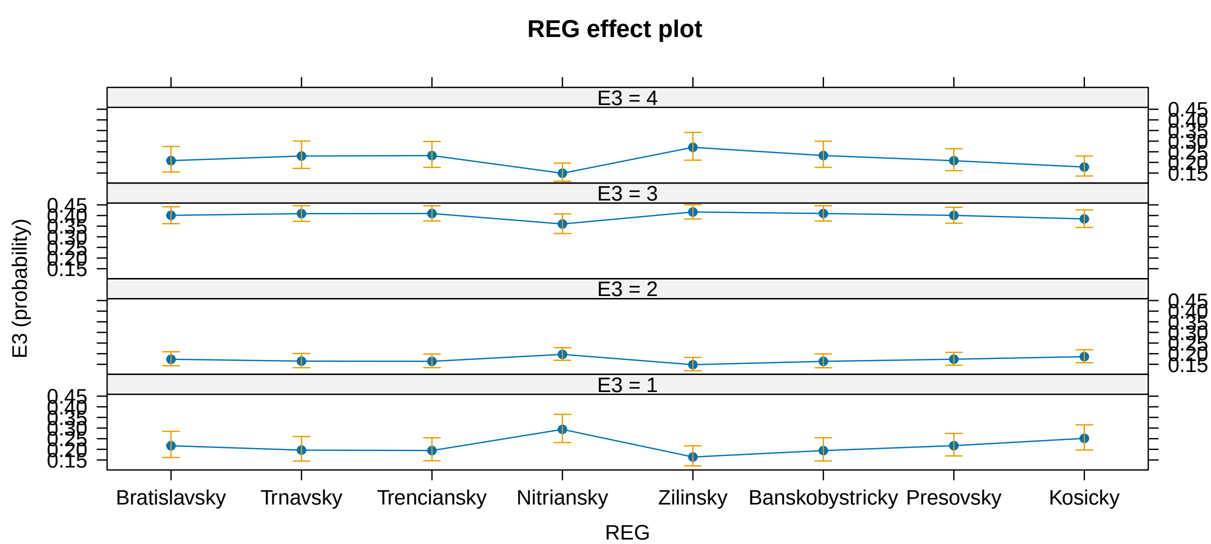


Independent Variable Effects

```r
plot(Effect(focal.predictors = "AGE", s3personal_alter))
```




```
plot(Effect(focal.predictors = "REG", s3personal_alter))
```



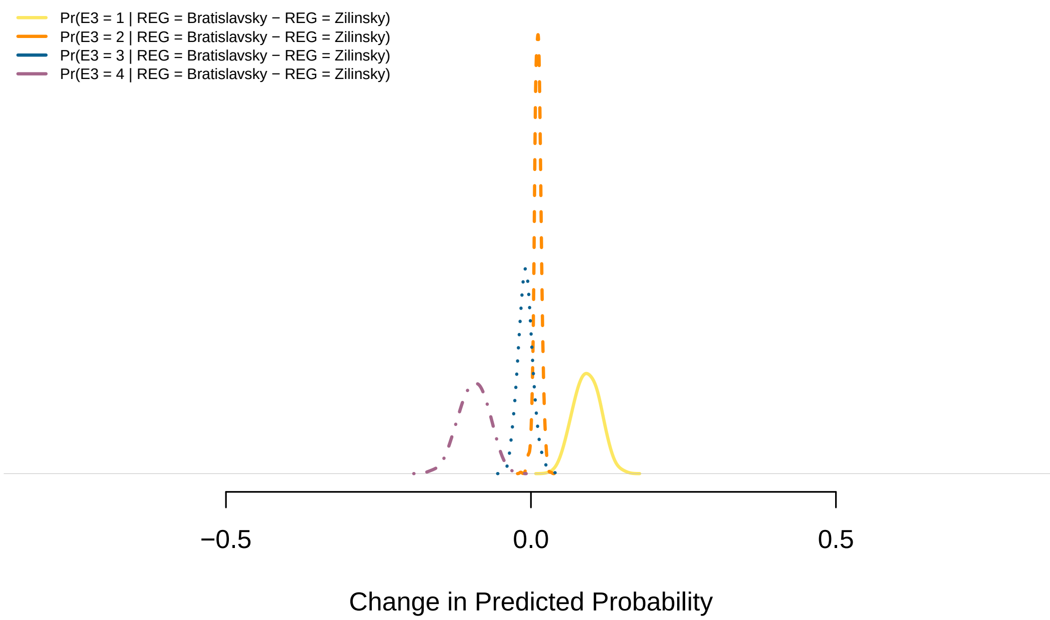
Quantity of Interest

```
# This function gets the most often
# repeating value in a column
mode_char <- function(x) unique(x)[which.max(tabulate(match(x,
    unique(x))))]
```

```
s3personal_alter <- polr(E3 ~ as.numeric(group) + AGE + as.numeric(EDU) +
                              as.numeric(SEX) + as.numeric(REG), data = s3,
                            Hess=TRUE, method = "probit")

#We create a scenario where we look at the following for the Bratislavsky and
# Zilinsky regions
# Median AGE
# Most often repeating (mode) EDU
# Most often repeating (mode) SEX
# REG = 1/5 (Bratislavsky/Zilinsky)

X.low<-c(group = mode_char(as.numeric(filter(s3, REG == "Bratislavsky")$group)),
         AGE = median(filter(s3, REG == "Bratislavsky")$AGE),
         EDU = mode_char(as.numeric(filter(s3, REG == "Bratislavsky")$EDU)),
         SEX = mode_char(as.numeric(filter(s3, REG == "Bratislavsky")$SEX)),
         REG = 1) #REG=Bratislavsky

X.high<-c(group = mode_char(as.numeric(filter(s3, REG == "Zilinsky")$group)),
        AGE = median(filter(s3, REG == "Zilinsky")$AGE),
        EDU = mode_char(as.numeric(filter(s3, REG == "Zilinsky")$EDU)),
        SEX = mode_char(as.numeric(filter(s3, REG == "Zilinsky")$SEX)),
        REG = 5) #REG=Zilinsky

draws<-mvrnorm(1000, #1000 draws;
               c(coef(s3personal_alter),
                 s3personal_alter$zeta), #note inclusion of cutpoints
               solve(s3personal_alter$Hessian))

B<-draws[,1:length(coef(s3personal_alter))]
Taus<-draws[,(length(coef(s3personal_alter))+1):ncol(draws)]

# Predicted probabilities for coop = 1 and coop = 4
pi.class1.sc1 <- plogis(Taus[, 1] - B %*% X.low) # Pr(Y = 1)
pi.class1.sc2 <- plogis(Taus[, 1] - B %*% X.high)

pi.class2.sc1 <- plogis(Taus[, 2] - B %*% X.low) -
  plogis(Taus[, 1] - B %*% X.low) # Pr(Y = 2)
pi.class2.sc2 <- plogis(Taus[, 2] - B %*% X.high) -
  plogis(Taus[, 1] - B %*% X.high)

pi.class3.sc1 <- plogis(Taus[, 3] - B %*% X.low) -
  plogis(Taus[, 2] - B %*% X.low) # Pr(Y = 3)
pi.class3.sc2 <- plogis(Taus[, 3] - B %*% X.high) -
  plogis(Taus[, 2] - B %*% X.high)

pi.class4.sc1 <- 1 - plogis(Taus[, 3] - B %*% X.low) # Pr(Y = 4)
pi.class4.sc2 <- 1 - plogis(Taus[, 3] - B %*% X.high)
```

```r
# Computing difference in probabilities
fd.class1 <- pi.class1.sc2 - pi.class1.sc1
fd.class2 <- pi.class2.sc2 - pi.class2.sc1
fd.class3 <- pi.class3.sc2 - pi.class3.sc1
fd.class4 <- pi.class4.sc2 - pi.class4.sc1


plot(density(fd.class1, adjust = 1.5),
     xlim = c(-0.8, 0.8),
     ylim = range(density(fd.class1)$y, density(fd.class2)$y,
                  density(fd.class3)$y, density(fd.class4)$y),
     xlab = "Change in Predicted Probability",
     col = "#FCE762", bty = "n",
     yaxt = "n", lwd = 2,
     main = "Implied effect on E3 (Personal Agreement of Construction of Apartment)",
     ylab = "",
     )

lines(density(fd.class2, adjust = 1.5), col = "darkorange", lwd = 2, lty = 2)
lines(density(fd.class3, adjust = 1.5), col = "#0C6291", lwd = 2, lty = 3)
lines(density(fd.class4, adjust = 1.5), col = "#A5668B", lwd = 2,
      lty = 4)

legend(
  "topleft",  # Adjust x and y to move the legend
  legend = c(
    "Pr(E3 = 1 | REG = Bratislavsky - REG = Zilinsky)",
    "Pr(E3 = 2 | REG = Bratislavsky - REG = Zilinsky)",
    "Pr(E3 = 3 | REG = Bratislavsky - REG = Zilinsky)",
    "Pr(E3 = 4 | REG = Bratislavsky - REG = Zilinsky)"
  ),
  col = c("#FCE762", "darkorange", "#0C6291", "#A5668B"),
  lwd = 2,
  bty = "n", cex = 0.6
)
```

Implied effect on E3 (Personal Agreement of Construction of Apartment)



Appendix

We used ChatGPT 4o LLM/AI tool in this report. We used this tool to understanding the author's original code. We also used ChatGPT to save time debugging our code, helping with latex formatting, and translating some of the original materials from Slovak to English. The tool was helpful and efficient for these tasks.

Click here to view my conversation with ChatGPT.

ChatGPT 4o LLM/AI was also used to help interpret the calibration plot and ROC curves we made for the in-sample predictive performance of the two models. It helped us understand how the models perfomed for in-sample predictions compared to each other.

Click here to view my conversation with ChatGPT.