MA678 Homework 6

Chang Lu

11/3/2024

Multinomial logit

Using the individual-level survey data from the 2000 National Election Study (data in folder NES), predict party identification (which is on a five-point scale) using ideology and demographics with an ordered multinomial logit model.

1. Summarize the parameter estimates numerically and also graphically.

```
nes_data <- read.table("nes.txt")
nes_data <- nes_data %>%
 mutate(partyid5 = case_when(
   partyid7 %in% c(1, 2) ~ 1,
   partyid7 == 3 ~ 2,
   partyid7 == 4 ~ 3,
   partyid7 == 5 ~ 4,
   partyid7 %in% c(6, 7) ~ 5
 ))
nes_data$partyid5 <- factor(nes_data$partyid5, ordered = TRUE)
model <- polr(partyid5 ~ ideo7 + age_new + gender + race + educ1, data = nes_data, Hess = TRUE)
summary(model)
```

```
## Call:
## polr(formula = partyid5 ~ ideo7 + age_new + gender + race + educ1,
##     data = nes_data, Hess = TRUE)
##
## Coefficients:
##            Value Std. Error t value
## ideo7    0.51417   0.010551  48.732
## age_new -0.04919   0.008402  -5.854
## gender  -0.09444   0.027257  -3.465
## race    -0.24195   0.015089 -16.035
## educ1    0.28114   0.015318  18.354
##
## Intercepts:
##     Value    Std. Error t value
## 1|2   1.9268   0.0820    23.5089
## 2|3   2.5168   0.0827    30.4505
## 3|4   2.9558   0.0835    35.3926
## 4|5   3.5748   0.0850    42.0668
##
```

```
## Residual Deviance: 51560.72
## AIC: 51578.72
## (16123 observations deleted due to missingness)
```

```
coeftest(model, vcov = vcov(model))
```

```
##
## t test of coefficients:
##
##           Estimate Std. Error  t value  Pr(>|t|)
## ideo7    0.5141693  0.0105509  48.7323 < 2.2e-16 ***
## age_new -0.0491864  0.0084024  -5.8538 4.884e-09 ***
## gender  -0.0944431  0.0272571  -3.4649 0.0005316 ***
## race    -0.2419482  0.0150888 -16.0349 < 2.2e-16 ***
## educ1    0.2811408  0.0153179  18.3538 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coef_table <- tidy(model, conf.int = TRUE)
coef_table
```

```
## # A tibble: 9 x 7
##   term    estimate std.error statistic conf.low conf.high coef.type
##   <chr>      <dbl>     <dbl>     <dbl>    <dbl>     <dbl> <chr>
## 1 ideo7     0.514    0.0106      48.7    0.494     0.535  coefficient
## 2 age_new  -0.0492   0.00840     -5.85  -0.0657   -0.0327 coefficient
## 3 gender   -0.0944   0.0273      -3.46  -0.148    -0.0410 coefficient
## 4 race     -0.242    0.0151     -16.0   -0.272    -0.212  coefficient
## 5 educ1     0.281    0.0153      18.4    0.251     0.311  coefficient
## 6 1|2       1.93     0.0820      23.5   NA        NA      scale
## 7 2|3       2.52     0.0827      30.5   NA        NA      scale
## 8 3|4       2.96     0.0835      35.4   NA        NA      scale
## 9 4|5       3.57     0.0850      42.1   NA        NA      scale
```

2. Explain the results from the fitted model.

Overall, this model helps us understand which factors (especially ideology and demographic characteristics) significantly influence individual party identification and provides a way to predict party leaning across different population groups.

3. Use a binned residual plot to assess the fit of the model.

```
nes_data <- read.table("nes.txt")
nes_data <- nes_data %>%
 mutate(partyid5 = case_when(
   partyid7 %in% c(1, 2) ~ 1,
   partyid7 == 3 ~ 2,
   partyid7 == 4 ~ 3,
   partyid7 == 5 ~ 4,
   partyid7 %in% c(6, 7) ~ 5
 ))
```

```r
nes_data$partyid5 <- factor(nes_data$partyid5, ordered = TRUE)
model_data <- nes_data %>%
 select(partyid5, ideo7, age_new, gender, race, educ1) %>%
 na.omit()
fit <- clm(as.factor(partyid5) ~ ideo7 + age_new + gender + race + educ1 ,
          data = model_data, link = "logit")
predicted_probs <- predict(fit, model_data, type = "prob")$fit
predicted_classes <- apply(predicted_probs, 1, which.max)
residuals <- as.numeric(model_data$partyid5) - as.numeric(predicted_classes)
num_bins <- 20
model_data$predicted_classes <- predicted_classes
model_data$residuals <- residuals
binned_data <- model_data %>%
 mutate(bin = cut(predicted_classes, breaks = num_bins)) %>%
 group_by(bin) %>%
 summarize(mean_residual = mean(residuals, na.rm = TRUE),
           bin_center = mean(predicted_classes, na.rm = TRUE))
```

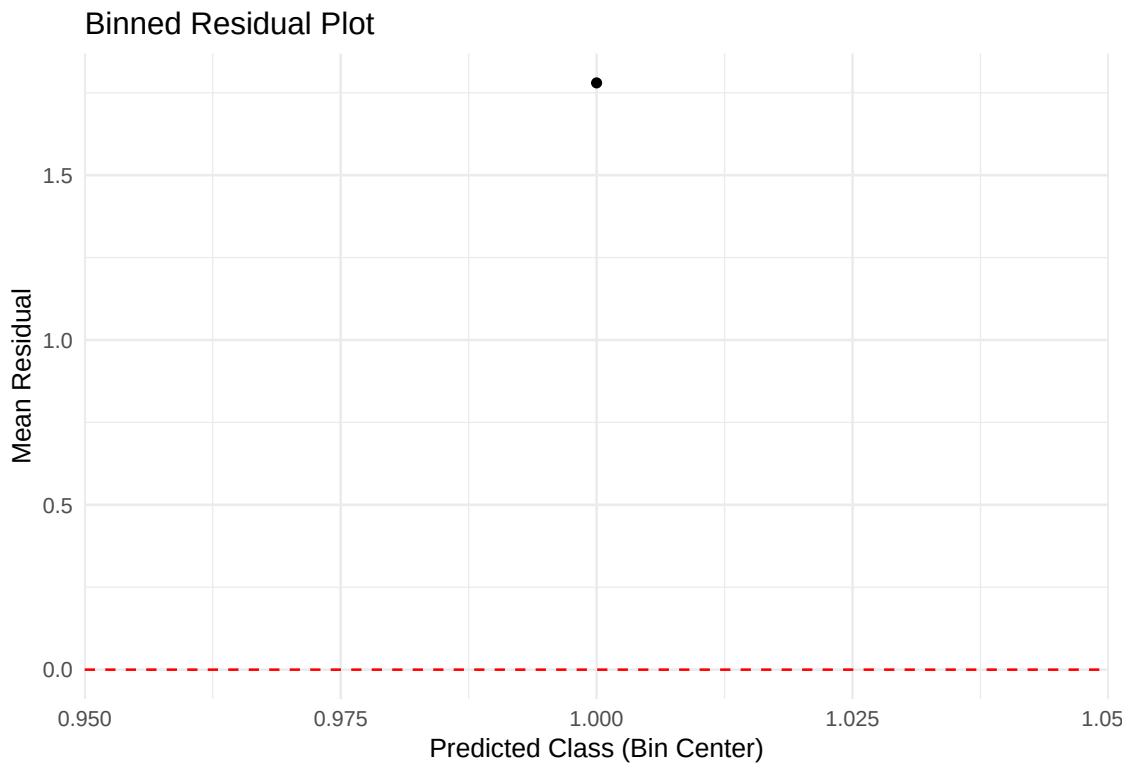
```r
# Check the distribution of predicted classes
summary(model_data$predicted_classes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
```

```r
num_bins <- 5  # try reducing the number of bins if there's a narrow range

binned_data <- model_data %>%
  mutate(bin = cut(predicted_classes, breaks = num_bins)) %>%
  group_by(bin) %>%
  summarize(mean_residual = mean(residuals, na.rm = TRUE),
            bin_center = mean(predicted_classes, na.rm = TRUE))

ggplot(binned_data, aes(x = bin_center, y = mean_residual)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(
    title = "Binned Residual Plot",
    x = "Predicted Class (Bin Center)",
    y = "Mean Residual"
  ) +
  theme_minimal()
```



Contingency table and ordered logit model

In a prospective study of a new living attenuated recombinant vaccine for influenza, patients were randomly allocated to two groups, one of which was given the new vaccine and the other a saline placebo. The responses were titre levels of hemaglutinin inhibiting antibody found in the blood six weeks after vaccination; they were categorized as "small", "medium" or "large".

treatment	small	moderate	large	Total
placebo	25	8	5	38
vaccine	6	18	11	35

The cell frequencies in the rows of table are constrained to add to the number of subjects in each treatment group (35 and 38 respectively). We want to know if the pattern of responses is the same for each treatment group.

1. Using a chi-square test and an appropriate log-linear model, test the hypothesis that the distribution of responses is the same for the placebo and vaccine groups.

```
# Creating the contingency table
response_data <- matrix(c(25, 8, 5, 6, 18, 11), nrow = 2, byrow = TRUE)
colnames(response_data) <- c("small", "moderate", "large")
rownames(response_data) <- c("placebo", "vaccine")

# Display the contingency table
response_data
```

```
##         small moderate large
```

```
## placebo    25        8     5
## vaccine     6       18    11
```

```
#           small moderate large
# placebo      25       8     5
# vaccine       6      18    11

# Chi-square test for independence
chi_square_test <- chisq.test(response_data)
chi_square_test
```

```
##
##  Pearson's Chi-squared test
##
## data:  response_data
## X-squared = 17.648, df = 2, p-value = 0.0001472
```

```
# Fitting a log-linear model to test the association
# Converting the data to a data frame format for glm
treatment <- factor(rep(c("placebo", "vaccine"), each = 3))
response <- factor(rep(c("small", "moderate", "large"), times = 2),
                   levels = c("small", "moderate", "large"))
frequency <- c(25, 8, 5, 6, 18, 11)
data <- data.frame(treatment, response, frequency)

# Fitting the log-linear model
log_linear_model <- glm(frequency ~ treatment * response, family = poisson, data = data)
summary(log_linear_model)
```

```
##
## Call:
## glm(formula = frequency ~ treatment * response, family = poisson,
##     data = data)
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        3.2189     0.2000  16.094  < 2e-16 ***
## treatmentvaccine                  -1.4271     0.4546  -3.139 0.001694 **
## responsemoderate                  -1.1394     0.4062  -2.805 0.005030 **
## responselarge                     -1.6094     0.4899  -3.285 0.001019 **
## treatmentvaccine:responsemoderate  2.2380     0.6223   3.597 0.000322 ***
## treatmentvaccine:responselarge     2.2156     0.7054   3.141 0.001684 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance:  2.3807e+01  on 5  degrees of freedom
## Residual deviance: -1.3323e-15  on 0  degrees of freedom
## AIC: 37.128
##
## Number of Fisher Scoring iterations: 3
```

```r
# Reduced model assuming independence (no interaction)
reduced_model <- glm(frequency ~ treatment + response, family = poisson, data = data)
summary(reduced_model)
```

```
##
## Call:
## glm(formula = frequency ~ treatment + response, family = poisson,
##     data = data)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        2.78111    0.21184  13.129   <2e-16 ***
## treatmentvaccine -0.08224    0.23428  -0.351   0.7256
## responsemoderate -0.17589    0.26593  -0.661   0.5083
## responselarge    -0.66140    0.30783  -2.149   0.0317 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 23.807  on 5  degrees of freedom
## Residual deviance: 18.643  on 2  degrees of freedom
## AIC: 51.771
##
## Number of Fisher Scoring iterations: 5
```

```r
# Likelihood ratio test to compare the models
anova(reduced_model, log_linear_model, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: frequency ~ treatment + response
## Model 2: frequency ~ treatment * response
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         2     18.642
## 2         0      0.000  2   18.642 8.95e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2. For the model corresponding to the hypothesis of homogeneity of response distributions, calculate the fitted values, the Pearson and deviance residuals, and the goodness of fit statistics X^2 and D. Which of the cells of the table contribute most to X^2 and D? Explain and interpret these results.

```r
# Calculate fitted values for the reduced model (homogeneity of response distributions)
fitted_values <- fitted(reduced_model)

# Calculate Pearson residuals
pearson_residuals <- residuals(reduced_model, type = "pearson")

# Calculate Deviance residuals
deviance_residuals <- residuals(reduced_model, type = "deviance")
```

```r
# Create a data frame with the observed, fitted values, Pearson residuals, and Deviance residuals
results <- data.frame(
  Treatment = treatment,
  Response = response,
  Observed = frequency,
  Fitted = fitted_values,
  Pearson_Residual = pearson_residuals,
  Deviance_Residual = deviance_residuals
)

# Display the results
print(results)
```

```
##   Treatment Response Observed    Fitted Pearson_Residual Deviance_Residual
## 1   placebo    small       25 16.136986         2.206329          2.040115
## 2   placebo moderate        8 13.534247        -1.504324         -1.629720
## 3   placebo    large        5  8.328767        -1.153435         -1.246900
## 4   vaccine    small        6 14.863014        -2.298942         -2.615460
## 5   vaccine moderate       18 12.465753         1.567470          1.468817
## 6   vaccine    large       11  7.671233         1.201852          1.127679
```

```r
# Calculate the goodness-of-fit statistics
X_squared <- sum(pearson_residuals^2)  # Pearson chi-square statistic
D <- sum(deviance_residuals^2)         # Deviance statistic

# Output the goodness-of-fit statistics
cat("Pearson Chi-Square Statistic (X^2):", X_squared, "\n")
```

```
## Pearson Chi-Square Statistic (X^2): 17.64783
```

```r
cat("Deviance Statistic (D):", D, "\n")
```

```
## Deviance Statistic (D): 18.64253
```

```r
# Determine which cells contribute most to X^2 and D
# Sorting the results based on the absolute values of residuals
results <- results %>%
  mutate(Abs_Pearson_Residual = abs(Pearson_Residual),
         Abs_Deviance_Residual = abs(Deviance_Residual)) %>%
  arrange(desc(Abs_Pearson_Residual), desc(Abs_Deviance_Residual))

print(results)
```

```
##   Treatment Response Observed    Fitted Pearson_Residual Deviance_Residual
## 4   vaccine    small        6 14.863014        -2.298942         -2.615460
## 1   placebo    small       25 16.136986         2.206329          2.040115
## 5   vaccine moderate       18 12.465753         1.567470          1.468817
## 2   placebo moderate        8 13.534247        -1.504324         -1.629720
## 6   vaccine    large       11  7.671233         1.201852          1.127679
## 3   placebo    large        5  8.328767        -1.153435         -1.246900
##   Abs_Pearson_Residual Abs_Deviance_Residual
```

```
## 4             2.298942             2.615460
## 1             2.206329             2.040115
## 5             1.567470             1.468817
## 2             1.504324             1.629720
## 6             1.201852             1.127679
## 3             1.153435             1.246900
```

3. Re-analyze these data using ordered logit model (use `polr`) to estimate the cut-points of a latent continuous response variable and to estimate a location shift between the two treatment groups. Sketch a rough diagram to illustrate the model which forms the conceptual base for this analysis.

```r
# Prepare the data in the appropriate format
treatment <- factor(c(rep("placebo", 3), rep("vaccine", 3)))
response <- ordered(c("small", "moderate", "large", "small", "moderate", "large"),
                    levels = c("small", "moderate", "large"))
frequency <- c(25, 8, 5, 6, 18, 11)
data <- data.frame(treatment, response, frequency)

# Expanding the data according to frequencies for polr to work correctly
expanded_data <- data[rep(1:nrow(data), data$frequency), 1:2]

# Fitting the ordered logit model
ordered_logit_model <- polr(response ~ treatment, data = expanded_data, method = "logistic")
summary(ordered_logit_model)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = response ~ treatment, data = expanded_data, method = "logistic")
##
## Coefficients:
##                   Value Std. Error t value
## treatmentvaccine 1.838     0.4882   3.764
##
## Intercepts:
##                 Value  Std. Error t value
## small|moderate 0.5654 0.3434     1.6465
## moderate|large 2.4414 0.4525     5.3949
##
## Residual Deviance: 139.6736
## AIC: 145.6736
```

interpretations

Thresholds (cut-points): The ordered logit model will provide estimates of the cut-points, which represent the thresholds on the latent continuous variable for moving from one category to the next. These cut-points divide the latent scale into sections corresponding to "small", "moderate", and "large" response levels.

Treatment Effect: The coefficient for the treatment variable (vaccine vs. placebo) represents a shift on the latent scale due to the treatment. A positive coefficient would suggest that the vaccine group tends to have higher response levels (e.g., higher antibody levels).

Conceptual Diagram

1. Latent Continuous Variable Axis: Draw a horizontal axis representing the latent continuous response variable (e.g., level of hemagglutinin inhibiting antibodies).
2. Cut-points (Thresholds): Mark two points on this axis labeled as τ_1 and τ_2 , representing the thresholds between response levels ("small" to "moderate" and "moderate" to "large").
3. Distribution Shifts: Draw two normal-shaped curves on this axis, one for the placebo group and one for the vaccine group. Shift the vaccine group curve to the right if the treatment effect is positive, indicating higher antibody levels for the vaccine group.
4. Response Categories: Divide the axis into three sections based on the cut-points: Left of τ_1: "Small" response Between τ_1 and τ_2: "Moderate" response Right of τ_2: "Large" response

High School and Beyond

The `hsb` data was collected as a subset of the High School and Beyond study conducted by the National Education Longitudinal Studies program of the National Center for Education Statistics. The variables are gender; race; socioeconomic status; school type; chosen high school program type; scores on reading, writing, math, science, and social studies. We want to determine which factors are related to the choice of the type of program—academic, vocational, or general—that the students pursue in high school. The response is multinomial with three levels.

```
data(hsb)
?hsb
```

```
## starting httpd help server ... done
```

1. Fit a trinomial response model with the other relevant variables as predictors (untransformed).

```
# Load necessary library
library(nnet)

# Fit the multinomial logistic regression model
hsb_model <- multinom(prog ~ gender + race + ses + schtyp + read + write + math + science + socst, data
```

```
## # weights:  42 (26 variable)
## initial  value 219.722458
## iter  10 value 171.814970
## iter  20 value 153.793692
## iter  30 value 152.935260
## final  value 152.935256
## converged
```

```
# Display a summary of the model to view coefficients and cut-points
summary(hsb_model)
```

```
## Call:
## multinom(formula = prog ~ gender + race + ses + schtyp + read +
##     write + math + science + socst, data = hsb)
##
```

```
## Coefficients:
##          (Intercept)  gendermale raceasian racehispanic racewhite     seslow
## general     3.631901 -0.09264717  1.352739   -0.6322019 0.2965156 1.09864111
## vocation    7.481381 -0.32104341 -0.700070   -0.1993556 0.3358881 0.04747323
##          sesmiddle schtyppublic        read       write       math    science
## general  0.7029621    0.5845405 -0.04418353 -0.03627381 -0.1092888 0.10193746
## vocation 1.1815808    2.0553336 -0.03481202 -0.03166001 -0.1139877 0.05229938
##                socst
## general  -0.01976995
## vocation -0.08040129
##
## Std. Errors:
##          (Intercept) gendermale raceasian racehispanic racewhite    seslow
## general     1.823452  0.4548778  1.058754    0.8935504 0.7354829 0.6066763
## vocation    2.104698  0.5021132  1.470176    0.8393676 0.7480573 0.7045772
##          sesmiddle schtyppublic       read      write       math    science
## general  0.5045938    0.5642925 0.03103707 0.03381324 0.03522441 0.03274038
## vocation 0.5700833    0.8348229 0.03422409 0.03585729 0.03885131 0.03424763
##              socst
## general  0.02712589
## vocation 0.02938212
##
## Residual Deviance: 305.8705
## AIC: 357.8705
```

2. For the student with id 99, compute the predicted probabilities of the three possible choices.

```
# Filter the data to get the row for student with ID 99
student_99 <- hsb[hsb$id == 99, ]

# Compute predicted probabilities for each program type for student with ID 99
predicted_probs_99 <- predict(hsb_model, newdata = student_99, type = "probs")

# Display the predicted probabilities
predicted_probs_99
```

```
##  academic   general  vocation
## 0.5076752 0.3753090 0.1170158
```

Happiness

Data were collected from 39 students in a University of Chicago MBA class and may be found in the dataset happy.

```
library(faraway)
data(happy)
```

Build a model for the level of happiness as a function of the other variables.

```r
# Fit an ordered logistic regression model using polr from MASS package
happy_model <- polr(as.ordered(happy) ~ money + sex + love + work, data = happy, method = "logistic")

# View the summary of the model to interpret parameters
summary(happy_model)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = as.ordered(happy) ~ money + sex + love + work,
##     data = happy, method = "logistic")
##
## Coefficients:
##          Value Std. Error t value
## money  0.02246    0.01066  2.1064
## sex   -0.47344    0.79498 -0.5955
## love   3.60765    0.80114  4.5031
## work   0.88751    0.40826  2.1739
##
## Intercepts:
##      Value   Std. Error t value
## 2|3   5.4708  1.9891     2.7504
## 3|4   6.4684  1.9223     3.3650
## 4|5   9.1591  2.1698     4.2212
## 5|6  10.9725  2.3213     4.7268
## 6|7  11.5113  2.3720     4.8530
## 7|8  13.5433  2.6673     5.0776
## 8|9  17.2909  3.1454     5.4971
## 9|10 19.0112  3.3270     5.7142
##
## Residual Deviance: 94.86029
## AIC: 118.8603
```

Interpret the parameters of your chosen model.

1. money (0.02246):

The coefficient for money is 0.02246, which is positive and statistically significant (t-value = 2.1064). This means that for each unit increase in money (likely measured in thousands of dollars), the log odds of being in a higher happiness category increase by 0.02246, holding all other variables constant. In practical terms, higher income is associated with a higher level of happiness.

2. sex (-0.47344):

The coefficient for sex is -0.47344, which is negative and not statistically significant (t-value = -0.5955). This suggests that being sexually active (sex = 1) may be associated with a decrease in the log odds of being in a higher happiness category, though this relationship is not statistically significant. In this sample, sexual activity does not appear to have a strong association with happiness.

3. love (3.60765):

The coefficient for love is 3.60765, which is positive and statistically significant (t-value = 4.5031). This means that as the level of love (or perceived love) increases by one unit, the log odds of being in a higher happiness category increase substantially by 3.60765, holding other variables constant. This is a strong effect, indicating that feeling loved or having a strong romantic relationship is highly associated with higher levels of happiness.

4. work (0.88751):

The coefficient for work is 0.88751, which is positive and statistically significant (t-value = 2.1739). This suggests that for each unit increase in work satisfaction, the log odds of being in a higher happiness category increase by 0.88751, holding all other factors constant. Job satisfaction has a moderate and positive association with happiness.

5. Interpretation of Intercepts (Thresholds) The intercepts (labeled 2|3, 3|4, 4|5, etc.) represent the estimated thresholds on the latent continuous happiness scale that separate the different observed happiness levels. These cut-points allow the model to distinguish between the ordered levels of happiness.

For example, the threshold 2|3 (5.4708) represents the cut-point on the latent scale between happiness levels 2 and 3. Each successive threshold separates adjacent happiness categories, with higher thresholds indicating transitions to higher levels of happiness.

Predict the happiness distribution for subject whose parents earn $30,000 a year,

who is lonely, not sexually active and has no job.

```
# Define the new data for prediction
new_subject <- data.frame(money = 30, sex = 0, love = 1, work = 0)

# Predict the probability distribution for happiness levels
predicted_happiness <- predict(happy_model, newdata = new_subject, type = "probs")

# Display the predicted distribution
predicted_happiness
```

```
##            2            3            4            5            6            7
## 7.666374e-01 1.324359e-01 9.336979e-02 6.316565e-03 5.163141e-04 6.290945e-04
##            8            9           10
## 9.272845e-05 1.838218e-06 4.008120e-07
```

Newspaper survey on Vietnam War

A student newspaper conducted a survey of student opinions about the Vietnam War in May 1967. Responses were classified by sex, year in the program and one of four opinions. The survey was voluntary. The data may be found in the dataset `uncviet`. Treat the opinion as the response and the sex and year as predictors. Build a proportional odds model, giving an interpretation to the estimates.

```
data(uncviet)
```

```r
# Convert `policy` to an ordered factor, assuming it has an inherent ordering
uncviet$policy <- ordered(uncviet$policy, levels = c("A", "B", "C", "D"))

# Fit the proportional odds model
policy_model <- polr(policy ~ sex + year, data = uncviet, method = "logistic")

# View the summary of the model to interpret coefficients and thresholds
summary(policy_model)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = policy ~ sex + year, data = uncviet, method = "logistic")
##
## Coefficients:
##                 Value Std. Error    t value
## sexMale     2.742e-16     0.5657  4.848e-16
## yearGrad    2.119e-16     0.8944  2.369e-16
## yearJunior  1.477e-17     0.8944  1.651e-17
## yearSenior  1.665e-16     0.8944  1.862e-16
## yearSoph   -1.110e-16     0.8944 -1.241e-16
##
## Intercepts:
##     Value   Std. Error t value
## A|B -1.0986  0.7303    -1.5043
## B|C  0.0000  0.7071     0.0000
## C|D  1.0986  0.7303     1.5043
##
## Residual Deviance: 110.9035
## AIC: 126.9035
```

Interpretation of Coefficients

1. sexMale (2.742e-16):

The coefficient for sexMale is close to zero, which is effectively zero and has no meaningful impact on the log odds. This suggests that there is no significant difference in the opinion about the Vietnam War between male and female students.

2. year Variables (Grad, Junior, Senior, Soph):

All coefficients for year (Grad, Junior, Senior, and Sophomore) are also extremely close to zero. This implies that the year in the program (e.g., freshman, sophomore, etc.) does not significantly influence students' opinions on the Vietnam War, as there is no substantial difference in the log odds across these groups. The very small values of the coefficients (close to zero) indicate that neither sex nor year has a strong association with the response variable (policy). In other words, these predictors do not seem to meaningfully explain the variation in opinions about the Vietnam War in this model.

Interpretation of Intercepts (Thresholds)

The intercepts (or thresholds) in the model represent the cut-points on the latent continuous scale of opinions that separate each adjacent category. Here's what each threshold represents:

A|B (-1.0986):

This is the threshold between opinions "A" and "B". A negative threshold means that the latent variable threshold for moving from "A" to "B" is relatively low, implying that a slight increase in the latent opinion variable would push an individual from the "A" to the "B" category. B|C (0.0000):

This threshold between "B" and "C" is exactly zero. This suggests that the latent variable levels separating "B" and "C" are symmetric around zero, meaning there is no shift in the underlying latent opinion variable needed to move between these categories. C|D (1.0986):

This threshold is positive, suggesting that a higher level on the latent variable is required to move from "C" to "D". It implies that transitioning to opinion "D" requires a relatively higher latent opinion score compared to the other transitions. ### Model Fit Residual Deviance (110.9035) and AIC (126.9035): These values give us an indication of model fit, though without a comparison to other models, they are difficult to interpret on their own. A lower AIC would indicate a better model if comparing with other models.

Pneumonoconiosis of coal miners

The pneumo data gives the number of coal miners classified by radiological examination into one of three categories of pneumonoconiosis and by the number of years spent working at the coal face divided into eight categories.

```
data(pneumo, package = "faraway")
```

1. Treating the pneumonoconiosis status as response variable as nominal, build a model for predicting the frequency of the three outcomes in terms of length of service and use it to predict the outcome for a miner with 25 years of service.

```
# Fit the multinomial logistic regression model
pneumo_model <- multinom(status ~ year, data = pneumo, weights = Freq)
```

```
## # weights:  9 (4 variable)
## initial  value 407.585159
## iter  10 value 208.724810
## final  value 208.724782
## converged
```

```
# View a summary of the model to examine the coefficients
summary(pneumo_model)
```

```
## Call:
## multinom(formula = status ~ year, data = pneumo, weights = Freq)
##
## Coefficients:
##        (Intercept)        year
## normal   4.2916723 -0.08356506
## severe  -0.7681706  0.02572027
##
```

```
## Std. Errors:
##        (Intercept)       year
## normal   0.5214110 0.01528044
## severe   0.7377192 0.01976662
##
## Residual Deviance: 417.4496
## AIC: 425.4496
```

```
# Create a new data frame for a miner with 25 years of service
new_miner <- data.frame(year = 25)

# Predict the probabilities of each status for the new miner
predicted_probs <- predict(pneumo_model, newdata = new_miner, type = "probs")

# Display the predicted probabilities
predicted_probs
```

```
##       mild     normal     severe
## 0.09148821 0.82778696 0.08072483
```

2. Repeat the analysis with the pneumonoconiosis status being treated as ordinal.

```
# Convert `status` to an ordered factor
pneumo$status <- ordered(pneumo$status, levels = c("normal", "mild", "severe"))
```

```
# Fit the ordered logistic regression model
pneumo_ordinal_model <- polr(status ~ year, data = pneumo, weights = Freq, method = "logistic")

# View the summary of the model to examine the coefficients and thresholds
summary(pneumo_ordinal_model)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = status ~ year, data = pneumo, weights = Freq,
##     method = "logistic")
##
## Coefficients:
##       Value Std. Error t value
## year 0.0959    0.01194   8.034
##
## Intercepts:
##             Value   Std. Error t value
## normal|mild  3.9558  0.4097     9.6558
## mild|severe  4.8690  0.4411    11.0383
##
## Residual Deviance: 416.9188
## AIC: 422.9188
```

```r
# Create a new data frame for a miner with 25 years of service
new_miner <- data.frame(year = 25)

# Predict the probabilities of each status for the new miner
predicted_probs_ordinal <- predict(pneumo_ordinal_model, newdata = new_miner, type = "probs")

# Display the predicted probabilities
predicted_probs_ordinal
```

```
##     normal       mild     severe
## 0.82610096 0.09601474 0.07788430
```

3. Now treat the response variable as hierarchical with top level indicating whether the miner has the disease and the second level indicating, given they have the disease, whether they have a moderate or severe case.

```r
# Step 1: Create top-level variable for disease presence
pneumo$disease <- ifelse(pneumo$status == "normal", 0, 1)

# Step 2: Create second-level variable for severity, filtering only those with disease
pneumo_severity <- subset(pneumo, disease == 1)
pneumo_severity$severity <- ifelse(pneumo_severity$status == "severe", 1, 0)

# Fit a logistic regression model for disease presence
disease_model <- glm(disease ~ year, data = pneumo, family = binomial, weights = Freq)

# View the summary of the model
summary(disease_model)
```

```
##
## Call:
## glm(formula = disease ~ year, family = binomial, data = pneumo,
##     weights = Freq)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.96635    0.41891  -9.468  < 2e-16 ***
## year         0.09627    0.01236   7.787 6.88e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 391.93  on 21  degrees of freedom
## Residual deviance: 305.93  on 20  degrees of freedom
## AIC: 309.93
##
## Number of Fisher Scoring iterations: 5
```

```r
# Fit a logistic regression model for severity given disease
severity_model <- glm(severity ~ year, data = pneumo_severity, family = binomial, weights = Freq)
```

```r
# View the summary of the model
summary(severity_model)
```

```
##
## Call:
## glm(formula = severity ~ year, family = binomial, data = pneumo_severity,
##     weights = Freq)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.11342    0.86248  -1.291    0.197
## year         0.03547    0.02350   1.509    0.131
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 113.24  on 13  degrees of freedom
## Residual deviance: 110.89  on 12  degrees of freedom
## AIC: 114.89
##
## Number of Fisher Scoring iterations: 3
```

```r
# Create a new data frame for a miner with 25 years of service
new_miner <- data.frame(year = 25)

# Step 1: Predict probability of having the disease
prob_disease <- predict(disease_model, newdata = new_miner, type = "response")

# Step 2: Predict probability of severe case given disease
prob_severe_given_disease <- predict(severity_model, newdata = new_miner, type = "response")

# Calculate overall probabilities
prob_no_disease <- 1 - prob_disease
prob_mild <- prob_disease * (1 - prob_severe_given_disease)
prob_severe <- prob_disease * prob_severe_given_disease

# Display the probabilities
cat("Probability of No Disease:", prob_no_disease, "\n")
```

```
## Probability of No Disease: 0.826299
```

```r
cat("Probability of Mild Disease:", prob_mild, "\n")
```

```
## Probability of Mild Disease: 0.09664999
```

```r
cat("Probability of Severe Disease:", prob_severe, "\n")
```

```
## Probability of Severe Disease: 0.07705105
```

4. Compare the three analyses.
5. Multinomial Model: Treats each status as unrelated, making it less interpretable in a progression context, but it is flexible for purely categorical outcomes.

6. Ordinal Model: Uses the ordering of the categories, which is appropriate for severity progression (normal < mild < severe) and aligns well with the nature of the data.

7. Hierarchical Model: Provides a nuanced approach that mirrors real-world decision processes (disease diagnosis followed by severity assessment). This model is intuitive for medical or progressive conditions and provides probabilities in two stages, adding interpretability.

All three approaches yield similar probability estimates for a miner with 25 years of service. However, the hierarchical and ordinal models align better with the context of pneumoconiosis as a disease with ordered progression. The hierarchical model adds further interpretability by breaking down the outcome into disease presence and severity.