Ordinal Logistic Regression

Thomas Jensen

Ordered Logit Models

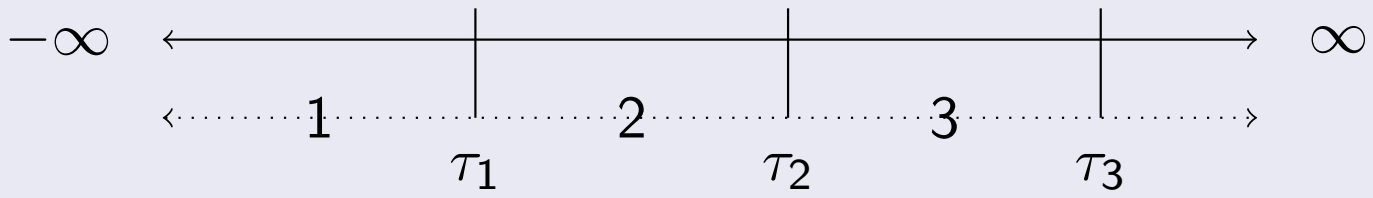
When should we use ordered logit models?

- Whenever the dependent variable is on an ordinal scale
  - Likert scale
  - Choice ranking
  - Ordered Categorization

Deriving the Ordered Logit Model

The latent variable

One way to derive the ordered logit model is to assume that we have an underlying continuous variable which is mapped onto our dependent variable.



Deriving the Ordered Logit Model

The Structural Model

As with the logit model the structural model is:

$$y_i^* = \alpha + x_i\beta + \epsilon_i \tag{1}$$

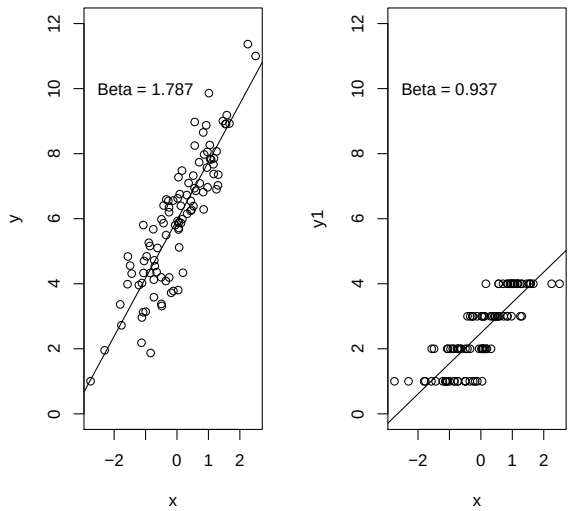
That is, the link between dependent and independent variables is linear in the parameters. This, of course, raises the question: why not just use OLS then?

Ordered Logit vs. OLS

Why not use OLS?

If we use OLS to estimate a model where the dependent variable is on an ordinal scale we make the assumption that the distance between each entry is the same.

```
> set.seed(123456)
> x <- rnorm(100)
> y <- 6 + 2 * x + rnorm(100)
> y1 <- y
> y1[y1 < 4.337] <- 1
> y1[y1 > 1 & y1 < 6.167] <- 2
> y1[y1 > 2 & y1 < 7.333] <- 3
> y1[y1 > 3] <- 4
> par(mfrow = c(1, 2))
> plot(x, y, ylim = c(0, 12))
> abline(lm(y ~ x))
> text(-1, 10, "Beta = 1.787")
> plot(x, y1, ylim = c(0, 12))
> abline(lm(y1 ~ x))
> text(-1, 10, "Beta = 0.937")
```



Distributional Assumptions

MLE

We can use MLE to estimate the regression of y^* on x. To do this we must assume a form for the error distribution. Here we will use the samme assumptions as we used with the logistic model (with the variance fixed at $\pi^2/3$)

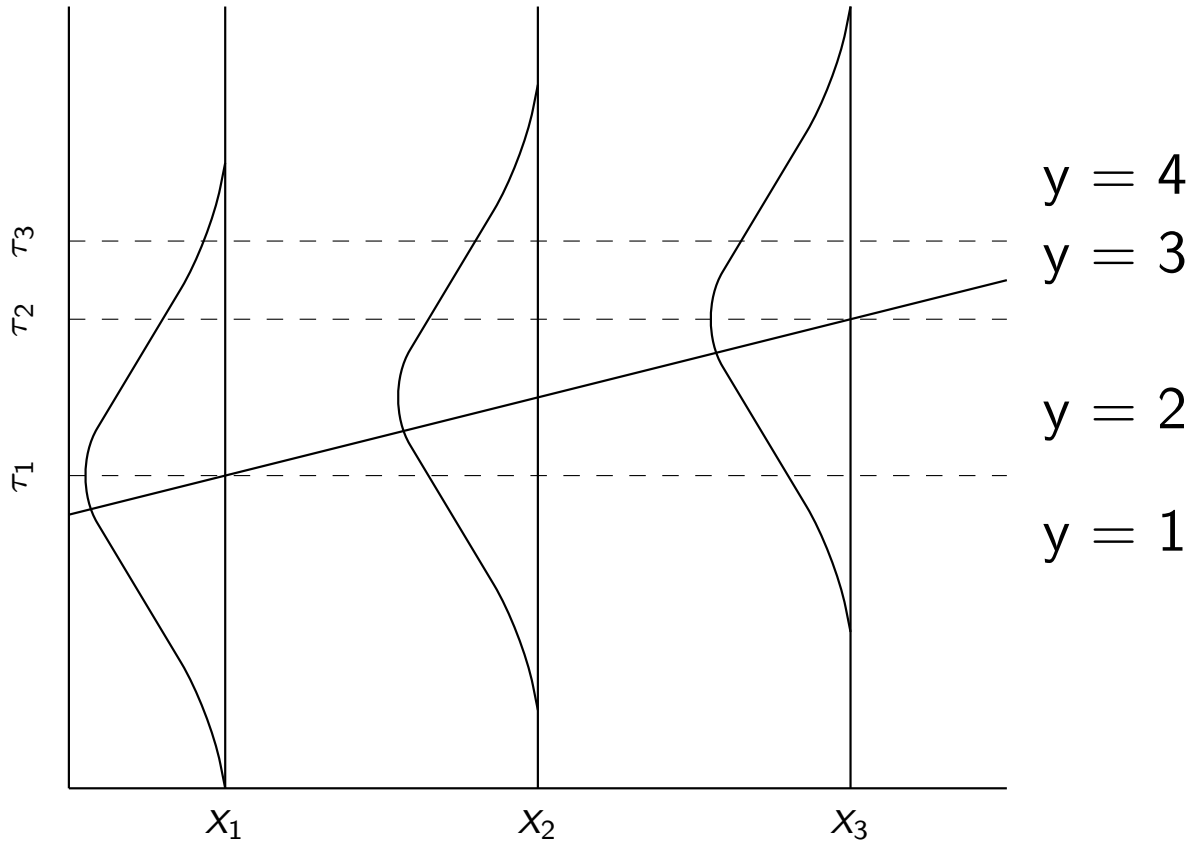
PDF

$$\lambda(\epsilon) = \frac{exp(\epsilon)}{[1 + exp(\epsilon)]^2} \tag{2}$$

CDF

$$\Lambda(\epsilon) = \frac{exp(\epsilon)}{1 + exp(\epsilon)} \tag{3}$$

Calculating Probabilities



Calculating Probabilities

The probability that y = 1

$$Pr(y_i = 1|x_i) = Pr(\tau_0 \leq y_i^* < |tau_1|x) \tag{4}$$

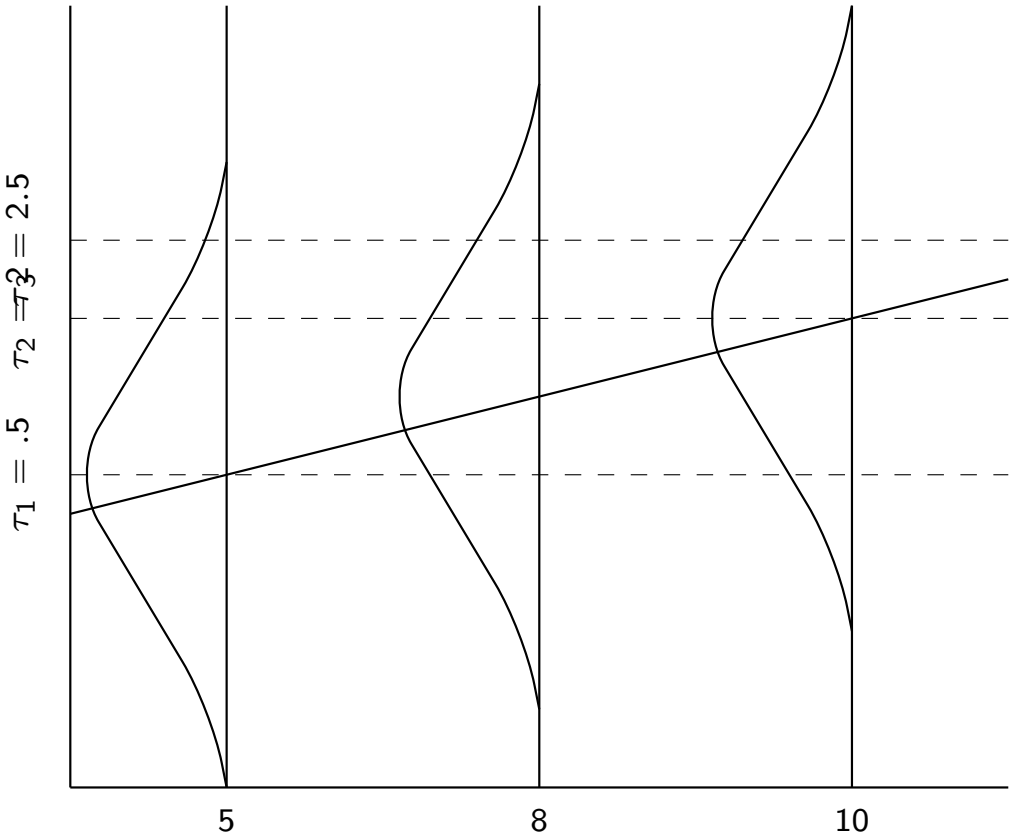
Substituting $y* = x\beta + \epsilon$

$$Pr(y_i = 1|x_i) = Pr(\tau_0 \leq x_i\beta + \epsilon < \tau_1|x_i) \tag{5}$$
$$Pr(y_i = 1|x_i) = Pr(\tau_0 - x_i\beta \leq \epsilon < \tau_1 - x_i\beta|x_i) \tag{6}$$
$$\tag{7}$$

The probability that a random variable is between two values is the difference in the CDF functions evaluated at the values, thus:

$$Pr(y_i = 1|x_i) = Pr(\epsilon < \tau_1 - x_i\beta|x_i) - Pr(\tau_0 - x_i\beta \leq) \tag{8}$$
$$= \Lambda(\tau_1 - x_i\beta) - \Lambda(\tau_0 - x_i\beta) \tag{9}$$

Calculating Probapilities



(assume that $\alpha = 1$ and $\beta = 2$)

Calculating Probapilities

$$Pr(y = 1|x_i) = \Lambda(\tau_1 - \alpha - x_i\beta) \tag{10}$$
$$Pr(y = 2|x_i) = \Lambda(\tau_2 - \alpha - x_i\beta) - \Lambda(\tau_1 - \alpha - x_i\beta) \tag{11}$$
$$Pr(y = 3|x_i) = \Lambda(\tau_3 - \alpha - x_i\beta) - \Lambda(\tau_2 - \alpha - x_i\beta) \tag{12}$$
$$Pr(y = 4|x_i) = 1 - \Lambda(\tau_3 - \alpha - x_i\beta) \tag{13}$$

An Example

$$Pr(y = 1|x = 5) = \frac{exp(.5 - 1 - .2 * 5)}{1 + exp(.5 - 1 - .2 * 5)} = 0.18 \tag{14}$$

$$Pr(y = 1|x = 8) = \frac{exp(.5 - 1 - .2 * 8)}{1 + exp(.5 - 1 - .2 * 8)} = 0.10 \tag{15}$$

$$Pr(y = 1|x = 10) = \frac{exp(.5 - 1 - .2 * 10)}{1 + exp(.5 - 1 - .2 * 10)} = 0.08 \tag{16}$$

$$Pr(y = 2|x = 5) = \frac{exp(2 - 1 - .2 * 5)}{1 + exp(2 - 1 - .2 * 5)} - \frac{exp(.5 - 1 - .2 * 5)}{1 + exp(.5 - 1 - .2 * 5)} = 0.32 \tag{17}$$

$$Pr(y = 2|x = 8) = \frac{exp(2 - 1 - .2 * 8)}{1 + exp(2 - 1 - .2 * 8)} - \frac{exp(.5 - 1 - .2 * 8)}{1 + exp(.5 - 1 - .2 * 8)} = 0.25 \tag{18}$$

$$Pr(y = 2|x = 10) = \frac{exp(2 - 1 - .2 * 10)}{1 + exp(2 - 1 - .2 * 10)} - \frac{exp(.5 - 1 - .2 * 10)}{1 + exp(.5 - 1 - .2 * 10)} = 0.19 \tag{19}$$

$$\tag{20}$$

The problem of identification

Adding an arbitrary constant to α and τ does not make a difference when estimating probabilities, so the question is: How can we choose between the possible values for α and τ?

An illustration

Assume that:

$$\alpha^* = \alpha - \delta \text{ and } \tau^* = \tau - \delta \tag{21}$$

Then:

$$\begin{aligned}
Pr(y = m|x) &= \Lambda(\tau_m - \alpha - x_i\beta) - \Lambda(\tau_m - 1 - \alpha - x_i\beta) \\
Pr(y = m|x) &= \Lambda([\tau_m - \delta] - [\alpha - \delta] - x_i\beta) \\
&\quad - \Lambda([\tau_{m-1} - \delta] - [\alpha - \delta] - x_i\beta) \\
Pr(y = m|x) &= \Lambda(\tau_m^* - \alpha * - x_i\beta) - \Lambda(\tau_m^* - 1 - \alpha^* - x_i\beta)
\end{aligned}$$

Identfying Assumptions

1. Assume that $\tau_1 = 0$. This is equivalent to setting $\delta = \tau_1$
2. Assume that $\alpha = 0$. This is equivalent to setting $\delta = \alpha$

These assumptions are to a degree arbitrary, and does affect the estimated probabilities. This is also referred to as different *parameterizations* of the model. Thus depending on the parameterization a software program might estimate either α or τ_1, but not both.

Estimation

For any given level the probability associated with tha level is (with either β_0 or τ set equal to 0):

$$Pr(y = m|x_i, \beta, \tau) = \Lambda(\tau_m - \alpha - x_i\beta) - \Lambda(\tau_m - 1 - \alpha - x_i\beta) \quad (22)$$

The probability of y for the ith observation is

$$P_i = \begin{cases} Pr(y_i = 1|x_i, \beta, \tau) & \text{if } y = 1 \\ \dots & \dots \\ Pr(y_i = m|x_i, \beta, \tau) & \text{if } y = m \\ \dots & \dots \\ Pr(y_i = j|x_i, \beta, \tau) & \text{if } y = j \end{cases} \quad (23)$$

Estimation

The likelihood function

$$L(\beta, \tau | y, X) = \prod_{i=1}^{n} p_i \tag{24}$$

$$= \prod_{j=1}^{j} \prod_{y_i=j} Pr(y_i = j | x_i, \beta, \tau) \tag{25}$$

$$= \prod_{j=1}^{j} \prod_{y_i=j} [\Lambda(\tau_j - x_i\beta) - \Lambda(\tau_{j-1} - x_i\beta)] \tag{26}$$

Taking the log of the likelihood function we get:

$$lnL(\beta, \tau | y, X) = \sum_{j=1}^{j} \sum_{y_i=j} ln[\Lambda(\tau_j - x_i\beta) - \Lambda(\tau_{j-1} - x_i\beta)] \tag{27}$$

The Parallel Regression Assumption

The idea of parallel regressions is based on the fact that in the ordered logit model, the different levels correspond to the same regression with different intercepts. This can be illustrated by the fact that in terms of cumulative probability:

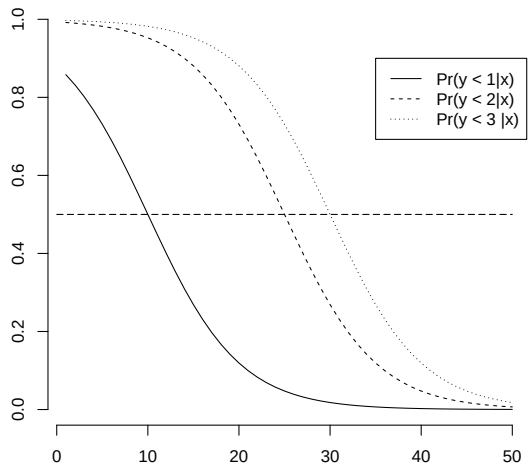
$$Pr(y \leq m|x) = \Lambda(\tau_m - x\beta) \tag{28}$$

The intercepts are defined by the τ values, and thus change with the different levels.

$$y \leq 1 \; : \; Pr(y \leq 1|x) = \Lambda(\tau_1 - \beta_0 - \sum_{k=1}^{k} \beta_k x_k)$$
$$y \leq 2 \; : \; Pr(y \leq 2|x) = \Lambda(\tau_2 - \beta_0 - \sum_{k=1}^{k} \beta_k x_k)$$

The Parallel Regression Assumption

```
> tau <- c(3, 6, 7)
> beta <- c(1, 0.2)
> x <- 1:50
> plot.new()
> plot.window(xlim = c(1, 50), ylim = c(0,
+     1))
> axis(1)
> axis(2)
> for (i in 1:3) {
+     prob <- exp(tau[i] - beta[1] - x *
+         beta[2])/(1 + exp(tau[i] - beta[1] -
+         x * beta[2]))
+     lines(x, prob, lty = i)
+ }
> segments(0, 0.5, 50, 0.5, lty = 5)
> legend(35, 0.9, c("Pr(y < 1|x)", "Pr(y < 2|x)",
+     "Pr(y < 3 |x)"), lty = 1:3)
```



The Parallel Regression Assumption

The parallel regression assumption is not violated if, for any point on the y-axis, the rate of change is constant across slopes. If this is not the case, i.e. the assumption is violated, then the ordinal logistic regression model is not appropriate

We can test the assumption of parallel regressions by running a series of logistic regressions for each $Pr(y \leq m|x) = \Lambda(\tau_m - x\beta_m)$. Thus the first binary regression has an outcome defined as 1 if $y \leq 1$, else 0. The second regression has an outcome defined as 1 if $y \leq 2$, else 0, and so on. If the assumption of parallel regressions is true the different betas should be the same across the regressions.

An example

The Study

For an illustration of ordinal regressio we will be using a study by Ryan Kennedy (2010). In this study Kennedy examines the effect on self-categorization (a proxy for identity) on political attitudes in Moldova.

An example

The Data

Variable	Operationalization	max	min
Support for democracy	ordinal variable expressing the level of support for democracy	5	0
EU Priority	Whether a respondent believes that the foreign policy should be oriented towards the EU	1	0
Gender	Male/female	1	0
Romanian	Whether a respondent primarily speaks romanian at home	1	0
Urban/Rural	Whether a respondent is from an urban or rural area	1	0
Trust in western media	Whether a respondent find western media trust worthy	1	0
Education	Ordinal variable classifying the respondents level of education	7	1
Age	Ordinal variable classifying a respondents age	4	1
Posessions	An index (continuous) of wealth based on a respondents possessings	1	0
ln(income)	The log of a respondents	11.513	5.298

Ordinal Logistic Regression in R

The Zelig Package

The zelig package was developed, and is still actively maintained, by Gary King at Harvard University. The package contains functions for estimating and interpreting the most used regression models in political science.

Install Zelig

To install the zelig package open R and type:

```
install.packages("Zelig", dep = T)
```

This will open a window where you can choose the mirror from which to download the package.

Ordinal Logistic Regression in R

the link function

All Zelig models follow the same syntax structure:

```
zelig(dep.var   ind.var, model, data)
```

An Example

Read libraries and the data

```
> library(foreign)
> library(Zelig)
> dat <- read.dta("/Users/thomasjensen/Downloads/Replication_Data/eup-09-0443-file003.dta")
```

Check that the data has been read correctly

```
> str(dat)

'data.frame': 1116 obs. of  121 variables:
 $ nr_chest              : int  1 2 3 4 5 6 7 8 9 10 ...
 $ cod_oper              : int  1 1 1 1 1 1 1 1 1 1 ...
 $ data                  : int  24 24 25 25 26 26 24 27 28 29 ...
 $ ora_ince              : num  7 10 19.1 21 8.05 ...
 $ right_direction       : num  0 0 1 -1 -1 0 -1 1 -1 -1 ...
 $ cancel_elections      : num  1 1 0 1 0 0 1 1 1 -1 ...
 $ suspend_meetings      : num  0 1 1 1 1 0 -1 1 1 -1 ...
 $ media_censorship      : num  0 1 -1 1 1 0 -1 1 1 1 ...
 $ dissolve_parliament   : num  0 1 -1 1 0 0 1 1 1 -1 ...
 $ restrict_travel       : num  1 1 -1 0 1 1 1 1 1 -1 ...
 $ content_democracy     : num  0 1 1 0 -1 0 0 1 0 1 ...
 $ content_economic      : num  0 1 1 -1 1 0 0 1 0 1 ...
 $ concern_politics      : num  -2 0 1 -2 -1 -2 -1 0 0 0 ...
 $ influence_local       : num  0 -1 0 -2 -1 -2 -1 -2 -1 -1 ...
 $ influence_national    : num  0 -1 0 -2 -1 0 2 -2 -1 -2 ...
 $ prior_orientation     : num  1 1 1 1 1 0 1 1 1 -1 ...
 $ eu_vote               : num  1 1 1 1 1 0 1 1 1 0 ...
 $ rl_romania            : num  1 -1 1 1 1 1 -1 -1 -1 -1 ...
 $ rl_ukraine            : num  1 -1 1 1 1 1 0 -1 -1 1 ...
 $ rl_russia             : num  1 1 1 1 -1 1 -1 1 1 1 ...
```

An Example

Estimating an ordered logit model

```
> model1 <- zelig(as.factor(dem_support2) ~
+     eu_prior + rom_lang + male + urban +
+         education + age + possession_index +
+         log(income) + trust_western_media +
+         opinion_eu, data = dat, model = "ologit")
```

```
> summary(model1)

Call:
zelig(formula = as.factor(dem_support2) ~ eu_prior +
    male + urban + education + age + possession_index +
    trust_western_media + opinion_eu, model = "ologit

Coefficients:
                    Value Std. Error t value
eu_prior            0.4756    0.12421   3.829
rom_lang           -0.3827    0.12346  -3.100
male                0.2586    0.11330   2.282
urban              -0.1930    0.12868  -1.500
education           0.1639    0.03280   4.999
age                -0.1036    0.05765  -1.797
possession_index    0.8404    0.28902   2.908
log(income)         0.0671    0.02680   2.503
trust_western_media 0.1887    0.13402   1.408
opinion_eu          0.1422    0.08065   1.764

Intercepts:
    Value   Std. Error t value
0|1  0.5263  0.3242     1.6235
1|2  0.9597  0.3248     2.9551
2|3  1.3840  0.3262     4.2428
3|4  1.8996  0.3288     5.7778
4|5  2.4476  0.3319     7.3734

Residual Deviance: 3458.080
AIC: 3488.08
```

Testing the Parallel Regression Assumption

In the Kennedy data the dependent variable ranges between 0 and 5, thus we msut create five different dependent variables and compare the results from each variable.

Creating the variables

```
> dep0 <- ifelse(dat$dem_support2 <= 0,
+     1, 0)
> dat$dep0 <- dep0
> dep1 <- ifelse(dat$dem_support2 <= 1,
+     1, 0)
> dat$dep1 <- dep1
> dep2 <- ifelse(dat$dem_support2 <= 2,
+     1, 0)
> dat$dep2 <- dep2
> dep3 <- ifelse(dat$dem_support2 <= 3,
+     1, 0)
> dat$dep3 <- dep3
> dep4 <- ifelse(dat$dem_support2 <= 4,
+     1, 0)
> dat$dep4 <- dep4
```

Testing the Parallel regression Assumption

Running the regressions

```
> test0 <- zelig(as.factor(dep0) ~ eu_prior + rom_lang + male +
+     urban + education + age + possession_index + log(income) +
+     trust_western_media + opinion_eu, data = dat, model = "logit")
> test1 <- zelig(as.factor(dep1) ~ eu_prior + rom_lang + male +
+     urban + education + age + possession_index + log(income) +
+     trust_western_media + opinion_eu, data = dat, model = "logit")
> test2 <- zelig(as.factor(dep2) ~ eu_prior + rom_lang + male +
+     urban + education + age + possession_index + log(income) +
+     trust_western_media + opinion_eu, data = dat, model = "logit")
> test3 <- zelig(as.factor(dep3) ~ eu_prior + rom_lang + male +
+     urban + education + age + possession_index + log(income) +
+     trust_western_media + opinion_eu, data = dat, model = "logit")
> test4 <- zelig(as.factor(dep4) ~ eu_prior + rom_lang + male +
+     urban + education + age + possession_index + log(income) +
+     trust_western_media + opinion_eu, data = dat, model = "logit")
```

Testing the Parallel Regression Assumption

```
> test <- cbind(coef(test0), coef(test1), coef(test2), coef(test3),
+     coef(test4))
> test
```

	[,1]	[,2]	[,3]	[,4]	[,5]
(Intercept)	0.28872149	0.91028126	1.46609146	2.03049306	2.30357137
eu_prior	-0.75504120	-0.69782415	-0.68898569	-0.35512504	-0.19843384
rom_lang	0.36170703	0.58705568	0.55610522	0.50557223	0.29888127
male	-0.22363283	-0.40415817	-0.36029409	-0.24938801	-0.17481164
urban	0.11619456	0.20096857	0.12047119	0.17348911	0.18867381
education	-0.21625831	-0.21823312	-0.16821526	-0.17315014	-0.09782536
age	0.21244179	0.15704465	0.10352972	0.04690874	-0.02101613
possession_index	-0.94932013	-0.90327651	-0.81882996	-0.87183176	-0.66481874
log(income)	-0.03290315	-0.05976662	-0.08340315	-0.08098908	-0.07213102
trust_western_media	0.12731668	-0.01703316	-0.09776695	-0.19026401	-0.41571608
opinion_eu	-0.20191600	-0.09863653	-0.07807418	-0.16019949	-0.12249111

Testing the Parallel Regression Assumption

Is the parallel regression assumption violated

- In a strict sense yes!
- All variables, except one, are fairly similar.
- The `trust_western_media` variable changes sign across the regressions, and thus this variable clearly violates the assumption.

Interpreting the Model

Predicted Probabilities

As a first step it is useful to examine the range of predicted probabilities in the sample, if the range is very small further analysis is not necessary. To do this we must get the predicted probabilities for each value of the dependent variable:

$$\bar{Pr}(y = m|x) = \frac{1}{N}\sum_{i=1}^{N}\hat{Pr}(y = m|x) \tag{29}$$

$$minPr(y = m|x) = \frac{1}{N}\sum_{i=1}^{N}\hat{Pr}(y = m|x) \tag{30}$$

$$maxPr(y = m|x) = \frac{1}{N}\sum_{i=1}^{N}\hat{Pr}(y = m|x) \tag{31}$$

Interpreting the Model

How to do this in R

```
> pre.prob <- matrix(0, nr = 4, nc = 6)
> for (i in 1:6) {
+     pre.prob[1, i] <- mean(model1$fitted.values[,
+         i])
+     pre.prob[2, i] <- min(model1$fitted.values[,
+         i])
+     pre.prob[3, i] <- max(model1$fitted.values[,
+         i])
+     pre.prob[4, i] <- diff(range(model1$fitted.values[,
+         i]))
+ }
> pre.prob

          [,1]       [,2]       [,3]       [,4]
[1,] 0.3138101 0.08758924 0.09169569 0.11149794
[2,] 0.0493925 0.02480367 0.03493618 0.04861788
[3,] 0.7367781 0.10791632 0.10568271 0.12820373
[4,] 0.6873856 0.08311265 0.07074653 0.07958585
           [,5]       [,6]
[1,] 0.10845981 0.28694722
[2,] 0.03326358 0.04971031
[3,] 0.13613565 0.73808469
[4,] 0.10287207 0.68837438
```

Interpreting the Model

We can see that for the values 1,2,3 the range between the highest and lowest predicted probability is below .1, thus for these cases there is not much of an effect in the model. This is caused by the distribution of cases on the dependent variable:

```
> table(dat$dem_support2)

  0   1   2   3   4   5
346  94  99 123 125 329
```

Interpreting the Model

Predicted Probabilities

We can also plot the effect of each variable in much the same way as we did for the logistic regression model. Hence, if we have more than one variable we will have to choose levels at which the other variables are held constant.

$$\hat{Pr}(y = m|x) = \Lambda(\hat{\tau_m} - x\hat{\beta}) - \Lambda(\hat{\tau_{m-1}} - x\hat{\beta}) \tag{32}$$

Interpreting the Model

The effect of education

Let us see what the effect of education on support for democracy is, holding the other variables constant at the mean/median values.

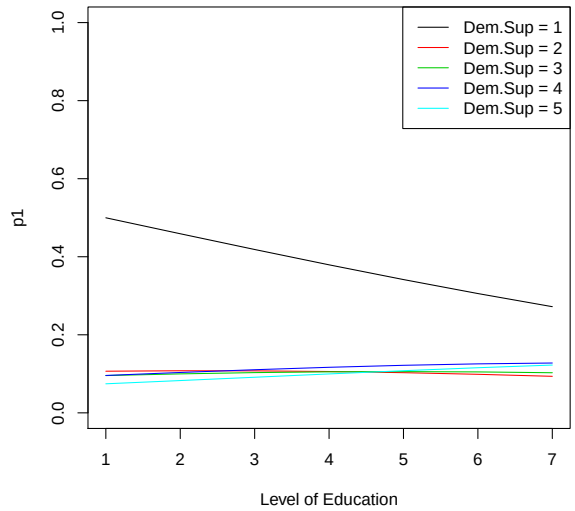
Setting up the calculation

```
> beta <- coef(model1)
> tau <- model1$zeta
> X <- cbind(0, 1, 0, 0, 1:7, 3, 0.49, 7.49,
+     0, 1)
> logit.prob <- function(eta) {
+     exp(eta)/(1 + exp(eta))
+ }
```

Interpreting the Model

```
> p1 <- logit.prob(tau[1] - X %*% beta)
> p2 <- logit.prob(tau[2] - X %*% beta) -
+     logit.prob(tau[1] - X %*% beta)
> p3 <- logit.prob(tau[3] - X %*% beta) -
+     logit.prob(tau[2] - X %*% beta)
> p4 <- logit.prob(tau[4] - X %*% beta) -
+     logit.prob(tau[3] - X %*% beta)
> p5 <- logit.prob(tau[5] - X %*% beta) -
+     logit.prob(tau[4] - X %*% beta)

> plot(1:7, p1, type = "l", col = 1, ylim = c(0,
+     1), xlab = "Level of Education")
> lines(1:7, p2, col = 2)
> lines(1:7, p3, col = 3)
> lines(1:7, p4, col = 4)
> lines(1:7, p5, col = 5)
> legend("topright", paste("Dem.Sup =",
+     1:5), lty = 1, col = 1:5)
```



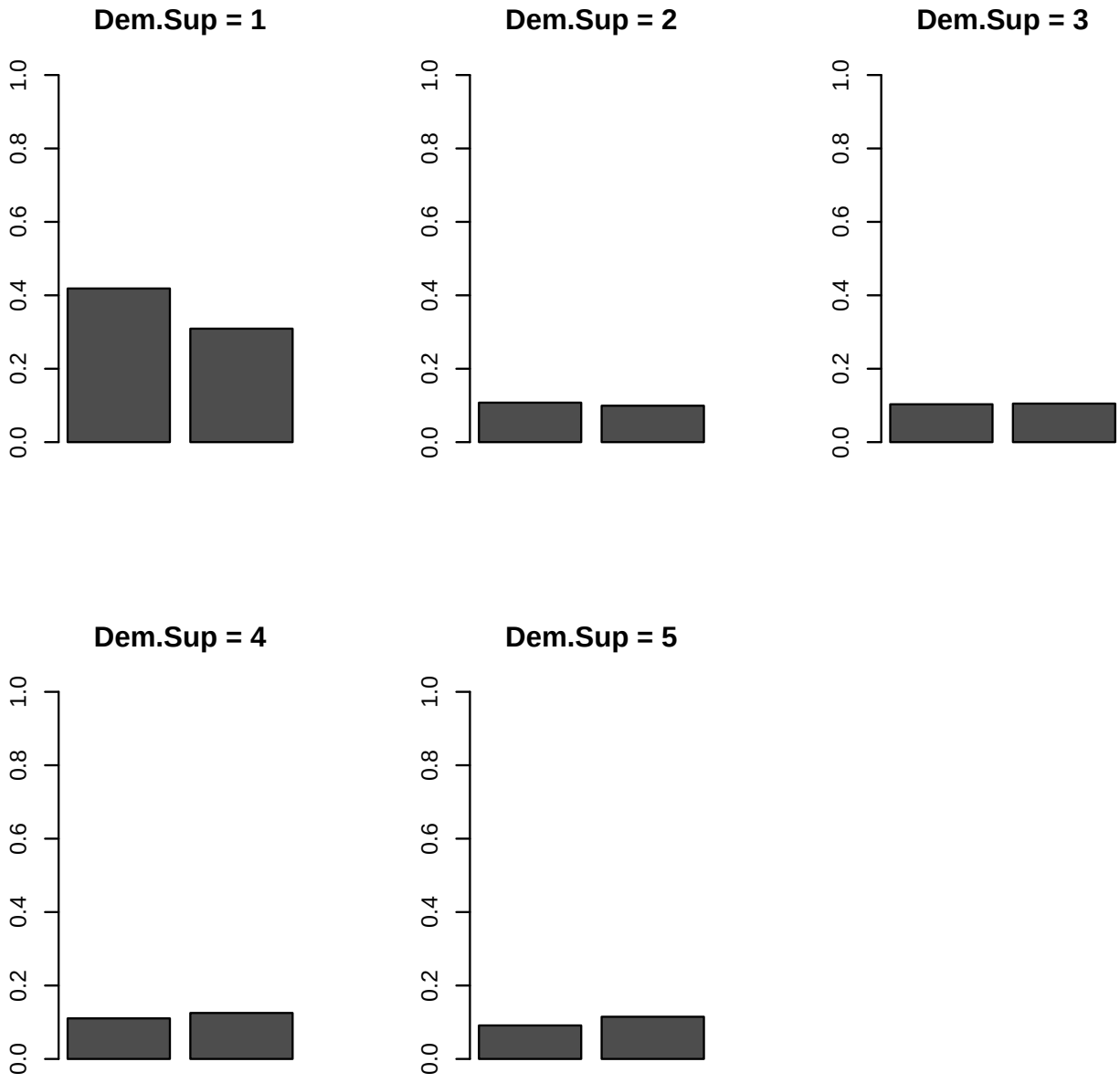
Interpreting the Model

The effect of EU support

Let us see what the effect of supporting the EU is, holding the other variables constant at their means/medians

Setting up the model

```
> X <- cbind(0:1, 1, 0, 0, 3, 3, 0.49, 7.49,
+     0, 1)
> p1 <- logit.prob(tau[1] - X %*% beta)
> p2 <- logit.prob(tau[2] - X %*% beta) -
+     logit.prob(tau[1] - X %*% beta)
> p3 <- logit.prob(tau[3] - X %*% beta) -
+     logit.prob(tau[2] - X %*% beta)
> p4 <- logit.prob(tau[4] - X %*% beta) -
+     logit.prob(tau[3] - X %*% beta)
> p5 <- logit.prob(tau[5] - X %*% beta) -
+     logit.prob(tau[4] - X %*% beta)
> par(mfrow = c(2, 3))
> barplot(t(p1), main = "Dem.Sup = 1", ylim = c(0,
+     1))
> barplot(t(p2), main = "Dem.Sup = 2", ylim = c(0,
+     1))
> barplot(t(p3), main = "Dem.Sup = 3", ylim = c(0,
+     1))
> barplot(t(p4), main = "Dem.Sup = 4", ylim = c(0,
+     1))
> barplot(t(p5), main = "Dem.Sup = 5", ylim = c(0,
+     1))
```


Dem.Sup = 1
Dem.Sup = 2
Dem.Sup = 3
Dem.Sup = 4
Dem.Sup = 5
0.0
0.2
0.4
0.6
0.8
1.0

Interpreting the Model

Kennedys Effect

Kennedy estimates the effect of a respondent supporting democracy at a level of 3 or more, given his/her support for the EU. To do this we calculate the probability of a respondent supporting democracy at a level of 2 or less (the CDF evaluated at τ_2) and subtract this from 1:

$$Pr(y > 2) = 1 - \Lambda(\hat{\tau_2} - x\hat{\beta}) \tag{33}$$

```
> pkennedy <- 1 - logit.prob(tau[2] - X %*%
+     beta)
> barplot(t(pkennedy), main = "Pr(Dem.Sup > 2)",
+     ylim = c(0, 1))
```