Data-analysis and Retrieval
Ordinal Classification

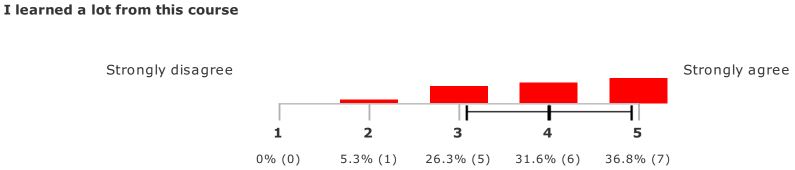
Ad Feelders

Universiteit Utrecht

Ordinal Classification

- When a variable is ordinal, its categories can be ranked from low to high, but the distances between adjacent categories are unknown.
- In ordinal classification the class variable is ordinal.

Example: Likert scale



Logistic Regression Revisited

Consider the linear regression model

$$y^* = \beta^\top \mathbf{x} + \varepsilon, \qquad E[\varepsilon \mid \mathbf{x}] = 0$$

where y^* is an unobserved (latent) numeric variable.
We only observe whether y^* is bigger than a given threshold:

$$y = \begin{cases} 1 & \text{if } y^* > 0 \\ 0 & \text{if } y^* \leq 0 \end{cases}$$

Note the vector notation: $\mathbf{x} = (1, x_1, \ldots, x_p)^\top$ and $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^\top$, so

$$\beta^\top \mathbf{x} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

Logistic Regression Revisited

According to this model, the probability that $y = 1$ is

$$\begin{aligned} P(y = 1) &= P(y^* > 0) \\ &= P(\beta^\top \mathbf{x} + \varepsilon > 0) \\ &= P(\varepsilon > -\beta^\top \mathbf{x}) \end{aligned}$$

If the distribution of ε is symmetric around zero, then $P(\varepsilon > a) = P(\varepsilon < -a)$, so

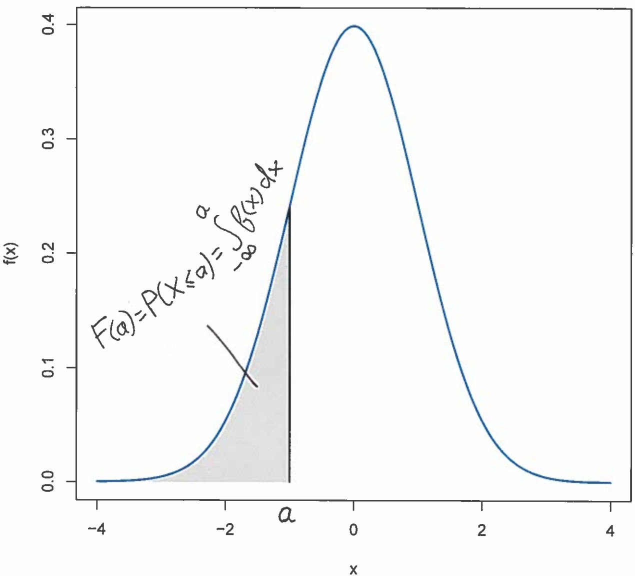
$$P(\varepsilon > -\beta^\top \mathbf{x}) = P(\varepsilon < \beta^\top \mathbf{x}) \equiv F(\beta^\top \mathbf{x})$$

Here F is the cumulative density function (cdf) of ε.
The cdf is defined as

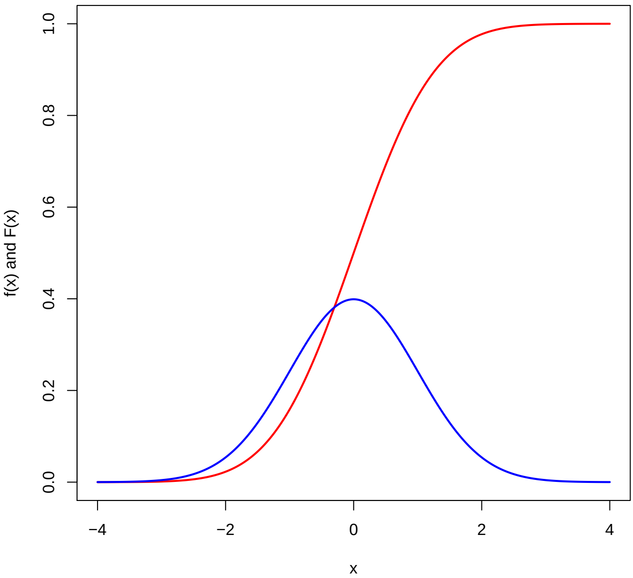
$$F(z) = P(Z \leq z) = \int_{-\infty}^{z} f(Z)dZ$$

where f is the probability density function (pdf) of Z.

Density and cumulative density



Standard Normal density and cumulative density function



The Probit Model

We have established that (under certain assumptions):

$$P(y = 1) = F(\beta^\top \mathbf{x})$$

Depending on the choice of F (or f) we get different models.

- If we choose $\varepsilon \sim N(0, 1)$, then we get the so-called probit model:

  $$P(y = 1) = \Phi(\beta^\top \mathbf{x})$$

  where $\Phi(\cdot)$ denotes the standard normal cumulative density function.
- The assumption of unit variance is a harmless normalization.

$\varepsilon \sim N(0, 1)$ is a harmless normalization.

Suppose we assume instead that $\varepsilon \sim N(0, \sigma^2)$, as is common in linear regression. First of all, note that

$$P(y = 1 \mid \mathbf{x}) = P(\varepsilon < \beta^\top \mathbf{x}) = P\left(\frac{\varepsilon}{\sigma} < \frac{\beta^\top \mathbf{x}}{\sigma}\right)$$

Define $u = \frac{\varepsilon}{\sigma}$. Then $u \sim N(0, 1)$. Furthermore, let $\alpha_j = \frac{\beta_j}{\sigma}$.

The model with coefficients α_j and error term u is "observationally equivalent" to the model with coefficients β_j and error term ε. They are "observationally equivalent" because they produce the exact same probabilities for the different Y values. Since Y is all we observe (not Y^*), the two models cannot be distinguished from each other on the basis of observations.

The Logit Model (Logistic Regression)

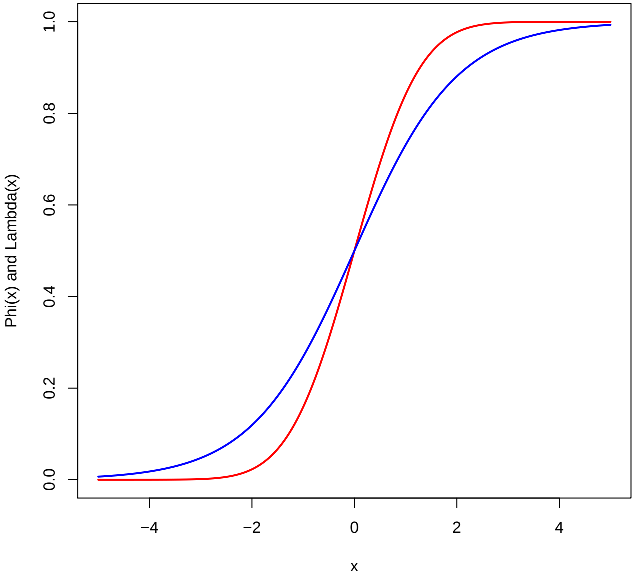
For the logit (logistic regression) model

$$P(y = 1) = \Lambda(\beta^\top \mathbf{x}) = \frac{e^{\beta^\top \mathbf{x}}}{1 + e^{\beta^\top \mathbf{x}}}$$

where $\Lambda(\cdot)$ denotes the logistic cumulative density function.

Note that this is the logistic response function we have already seen in one of the previous lectures.

Normal (red) and logistic (blue) cumulative density



Alternative Parametrization

Instead of fixing the threshold at zero, we can also remove the intercept β_0 from the model and make the threshold an unknown parameter. Then we get the model:

$$y^* = \sum_{j=1}^{p} \beta_j x_j + \varepsilon, \qquad E[\varepsilon \mid \mathbf{x}] = 0$$

where y^* is still an unobserved (latent) numeric variable.
We only observe whether y^* is bigger than a threshold t:

$$y = \begin{cases} 1 & \text{if } y^* > t \\ 0 & \text{if } y^* \leq t \end{cases}$$

Generalization to Ordinal Classification

Let m denote the number of classes, where the classes are labeled $\{1, 2, \ldots, m\}$. Then y is defined as follows:

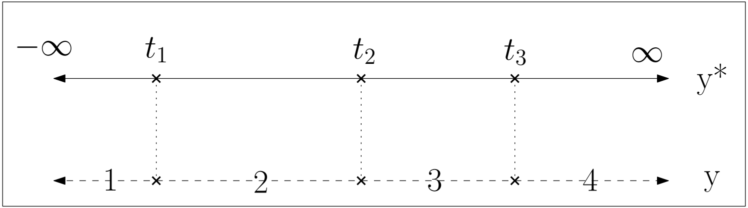
$$y = \begin{cases} 1 & \text{if } -\infty < y^* \leq t_1 \\ 2 & \text{if } t_1 < y^* \leq t_2 \\ \vdots & \vdots \\ m & \text{if } t_{m-1} < y^* < \infty \end{cases}$$

We only observe between which thresholds y^* falls.

Here $t_1, \ldots, t_{m-1}$ are unknown thresholds that have to be estimated from the data (together with the coefficient vector β).

Discretization of y^*

We only observe y, which indicates the interval y^* falls into.



Class Probabilities

We observe $y = 1$ when y^* falls between $t_0 = -\infty$ and t_1. Hence

$$P(y_i = 1 \mid \mathbf{x}_i) = P(t_0 \leq y_i^* < t_1 \mid \mathbf{x}_i)$$

Substituting $y_i^* = \beta^\top \mathbf{x}_i + \varepsilon_i$, (suppressing condition on $\mathbf{x}_i$) we get

$$P(y_i = 1) = P(t_0 \leq \beta^\top \mathbf{x}_i + \varepsilon_i < t_1)$$

Now we subtract $\beta^\top \mathbf{x}_i$ from all terms in the inequality to get

$$P(y_i = 1) = P(t_0 - \beta^\top \mathbf{x}_i \leq \varepsilon_i < t_1 - \beta^\top \mathbf{x}_i)$$

Class Probabilities

Continuing from the previous slide:

$$
\begin{aligned}
P(y_i = 1) &= P(t_0 - \beta^\top \mathbf{x}_i \leq \varepsilon_i < t_1 - \beta^\top \mathbf{x}_i) \\
&= P(\varepsilon_i < t_1 - \beta^\top \mathbf{x}_i) - P(\varepsilon_i < t_0 - \beta^\top \mathbf{x}_i) \\
&= F(t_1 - \beta^\top \mathbf{x}_i) - F(t_0 - \beta^\top \mathbf{x}_i),
\end{aligned}
$$

because $P(a \leq Z \leq b) = P(Z \leq b) - P(Z \leq a) = F(b) - F(a)$.

This derivation can be generalized to compute the probability of any observed outcome $y_i = j$ given $\mathbf{x}_i$:

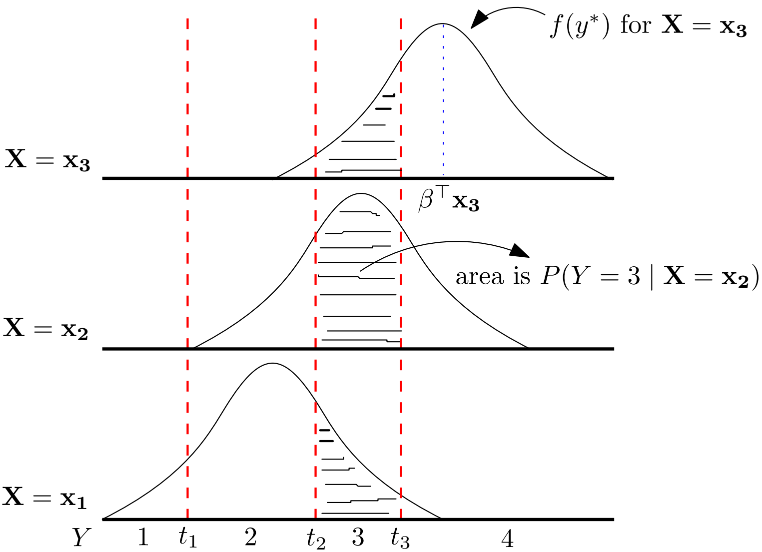
$$
P(y_i = j \mid \mathbf{x}_i) = F(t_j - \beta^\top \mathbf{x}_i) - F(t_{j-1} - \beta^\top \mathbf{x}_i), \qquad j = 1, \ldots, m.
$$

Class Probabilities

So for a model with four possible classes, the formula's for the different outcomes are:

$$P(y_i = 1 \mid \mathbf{x}_i) = F(t_1 - \beta^\top \mathbf{x}_i)$$
$$P(y_i = 2 \mid \mathbf{x}_i) = F(t_2 - \beta^\top \mathbf{x}_i) - F(t_1 - \beta^\top \mathbf{x}_i)$$
$$P(y_i = 3 \mid \mathbf{x}_i) = F(t_3 - \beta^\top \mathbf{x}_i) - F(t_2 - \beta^\top \mathbf{x}_i)$$
$$P(y_i = 4 \mid \mathbf{x}_i) = 1 - F(t_3 - \beta^\top \mathbf{x}_i)$$

Class Probabilities



Class Probabilities

Verbal description:

- Depending on the value of $\mathbf{x}$, the distribution of y^* is shifted.
- The expected value of y^* is $\beta^\top \mathbf{x}$.
- The class probabilities are defined by the area under $f(y^*)$ between the different thresholds.
- In this way, the class probabilities depend on the value of $\mathbf{x}$.

Maximum Likelihood Estimation

The likelihood function is

$$
\begin{aligned}
L(\beta, t\ ;\ \mathbf{X}, \mathbf{y}) &= \prod_{j=1}^{m} \prod_{i:y_i=j} P(y_i = j \mid \mathbf{x}_i, \beta, t) \\
&= \prod_{j=1}^{m} \prod_{i:y_i=j} \left[F(t_j - \beta^\top \mathbf{x}_i) - F(t_{j-1} - \beta^\top \mathbf{x}_i) \right],
\end{aligned}
$$

where $\prod_{i:y_i=j}$ indicates we multiply over all cases where y is observed to have value j.

Taking logs, the log likelihood is equal to

$$
\log L(\beta, t\ ;\ \mathbf{X}, \mathbf{y}) = \sum_{j=1}^{m} \sum_{i:y_i=j} \log \left[F(t_j - \beta^\top \mathbf{x}_i) - F(t_{j-1} - \beta^\top \mathbf{x}_i) \right].
$$

This expression can be maximized with numerical methods to estimate the thresholds t_j and vector of coefficients β.

Maximum Likelihood Estimation

The likelihood function is

$$L(\beta, t\,;\,\mathbf{X}, \mathbf{y}) = \prod_{j=1}^{m} \prod_{i:y_i=j} P(y_i = j \mid \mathbf{x}_i, \beta, t)$$

Note that the likelihood score of a model (choice of t, β) only depends on the probability that it assigns to the correct class.

Homework: Can you think of an argument against using MLE in *ordinal* classification?

Cumulative Class Probabilities

Also, note that:

$$
\begin{aligned}
P(y_i \leq 1 \mid \mathbf{x}_i) &= F(t_1 - \beta^\top \mathbf{x}_i) \\
P(y_i \leq 2 \mid \mathbf{x}_i) &= F(t_2 - \beta^\top \mathbf{x}_i) \\
P(y_i \leq 3 \mid \mathbf{x}_i) &= F(t_3 - \beta^\top \mathbf{x}_i) \\
P(y_i \leq 4 \mid \mathbf{x}_i) &= 1
\end{aligned}
$$

In general we have $P(y_i \leq j \mid \mathbf{x}_i) = F(t_j - \beta^\top \mathbf{x}_i)$.

Cumulative Class Probabilities

We have seen that:

$$P(y \leq j \mid \mathbf{x}) = F(t_j - \beta^\top \mathbf{x}).$$

In logistic regression we choose for F the logistic cdf

$$\Lambda(z) = \frac{\exp(z)}{1 + \exp(z)},$$

so we get

$$P(y \leq j \mid \mathbf{x}) = \frac{\exp(t_j - \beta^\top \mathbf{x})}{1 + \exp(t_j - \beta^\top \mathbf{x})}.$$

Set of *parallel* logistic regression models for $y \leq j$ against $y > j$:

$$\log \left[\frac{P(y \leq j \mid \mathbf{x})}{P(y > j \mid \mathbf{x})} \right] = t_j - \beta^\top \mathbf{x}$$

Interpretation: effect of increase in x_k

We have

$$P(y \leq j \mid \mathbf{x}) = F(t_j - \beta^\top \mathbf{x}).$$

Hence

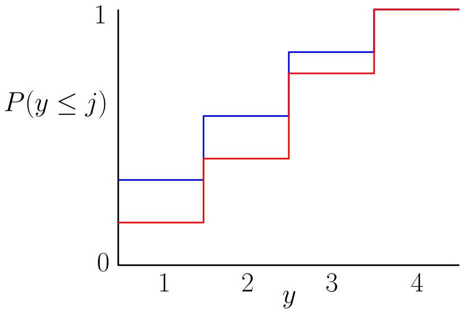
$$\begin{aligned}\frac{\partial P(y \leq j \mid \mathbf{x})}{\partial x_k} &= \frac{\partial F(t_j - \beta^\top \mathbf{x})}{\partial x_k} = \frac{\partial F(z)}{\partial z}\frac{\partial z}{\partial x_k} \\ &= f(z) \times -\beta_k = -\beta_k f(t_j - \beta^\top \mathbf{x}).\end{aligned}$$

$f(t_j - \beta^\top \mathbf{x})$ is always positive, since f is a probability density function.
So if β_k is positive, an increase in x_k will lead to a decrease in $P(y \leq j)$ for all $j = 1, \ldots, m-1$.

Or (same thing), an increase in x_k will lead to an increase in $P(y \geq j)$ for all $j = 2, \ldots, m$.

In this specific sense, one can say that if x_k increases, higher values of y become more likely.

Interpretation for β_k positive



Blue: $P(y \leq j)$ for x_k.
Red: $P(y \leq j)$ for $x_k + 1$.

The cumulative distribution of y for $x_k + 1$ is *entirely below* the cumulative distribution of y for x_k.

Summarizing the Differences

How exactly are the ordered and unordered (= multinomial) logistic regression model different?

- The ordinal model has a single coefficient vector β for all classes, whereas the multinomial model has a coefficient vector β_k for each class k (except one).
- As a consequence the decision boundaries are restricted to be parallel to each other in the ordinal model. This is quite a strong constraint!
- In the ordinal model the relation between predictor and class label is monotone, either increasing or decreasing.
- For example: if β_k is positive, then (all else equal) an increase in x_k makes the higher classes more likely and a decrease in x_k makes the lower classes more likely.

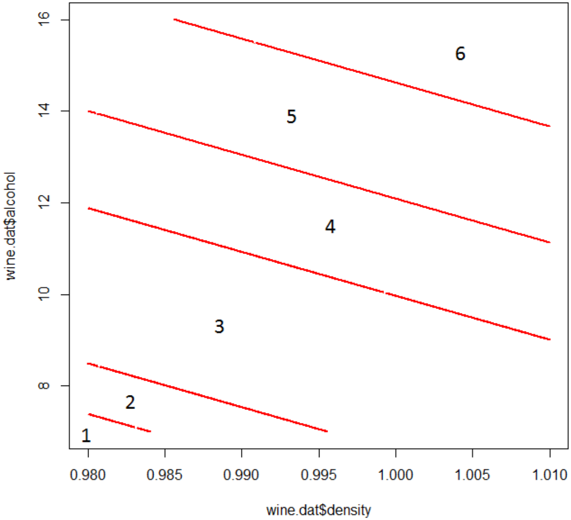
Fitting the Ordinal Logistic Regression Model in R

```
> library(MASS)
# fit proportional odds logistic regression model
> wine.polr2 <- polr(quality~density+alcohol,data=wine.dat, Hess=T)
> summary(wine.polr2)

Coefficients:
         Value Std. Error t value
density 106.27    0.30531  348.08
alcohol   1.11    0.05267   21.07

Intercepts:
    Value    Std. Error t value
1|2 111.9811   0.3032   369.3438
2|3 113.8616   0.3057   372.4595
3|4 117.2274   0.3253   360.3388
4|5 119.7431   0.3667   326.5169
5|6 122.6179   0.4476   273.9202
```

Prediction Accuracy

```
# predict class labels
> wine.pred <- predict(wine.polr,wine.dat,type="class")
# construct confusion matrix
> wine.confmat <- table(wine.dat[,12],wine.pred)
> wine.confmat
   wine.pred
      1   2   3   4   5   6
  1   0   0   9   1   0   0
  2   0   0  36  17   0   0
  3   0   2 503 173   2   1
  4   0   0 213 392  33   0
  5   0   0   7 138  54   0
  6   0   0   0  10   8   0

# compute accuracy
> sum(diag(wine.confmat))/1599
[1] 0.5934959
> summary(wine.dat[,12])
  1   2   3   4   5   6
 10  53 681 638 199  18
> 681/1599
[1] 0.4258912
```

Decision Boundary Ordinal LR on Wine Data



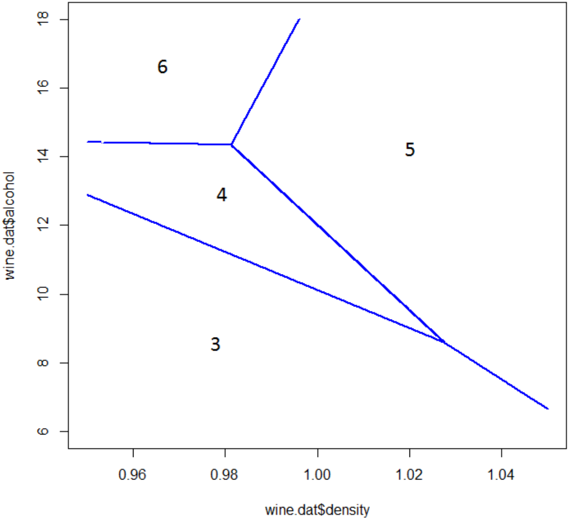
Fitting the Multinomial Logistic Regression Model

```
> library(nnet)
# fit multinomial logistic regression model
> wine.multi2 <- multinom(quality~density+alcohol,data=wine.dat)

> summary(wine.multi2)

Coefficients:
  (Intercept)    density    alcohol
2   43.003814 -45.879145  0.4365809
3   68.378492 -62.949698 -0.1396656
4    2.385932  -7.137190  0.8667334
5 -108.347472  93.833526  1.6793355
6  -17.906887  -4.108423  2.0746746

Std. Errors:
  (Intercept)  density   alcohol
2    2.260309 2.274173 0.4522562
3    2.137517 2.146860 0.4289871
4    2.139384 2.141155 0.4282659
5    2.166738 2.182500 0.4334476
6    2.507561 2.531655 0.4810792
```

Prediction Accuracy with Multinomial Logit

```
# predict class labels
> wine.pred.m <- predict(wine.multi2,wine.dat,type="class")
# construct confusion matrix
> wine.confmat.m <- table(wine.dat[,12],wine.pred.m)
> wine.confmat.m
   wine.pred.m
      1   2   3   4   5   6
  1   0   0   7   3   0   0
  2   0   0  30  22   1   0
  3   0   0 514 161   6   0
  4   0   0 267 347  24   0
  5   0   0  23 159  17   0
  6   0   0   2  10   6   0

# compute accuracy of predictions
> sum(diag(wine.confmat.m))/1599
[1] 0.5490932
```

Decision Boundary Multinomial LR on Wine Data



Comparison of Ordinal and Multinomial Model

```
> wine.polr.corr <- as.numeric(wine.pred==wine.dat[,12])
> wine.multi.corr <- as.numeric(wine.pred.m==wine.dat[,12])

> wine.comp <- table(wine.polr.corr,wine.multi.corr)
> wine.comp
              wine.multi.corr
wine.polr.corr   0   1
             0 546 104
             1 175 774

# is the difference in error rate (= 1-accuracy) significant?
```

Null hypothesis:

$$H_0 : e_{\text{polr}} = e_{\text{multi}}, \qquad H_a : e_{\text{polr}} \neq e_{\text{multi}}$$

If the null hypothesis is correct then $P(\text{cell } (1{,}0)) = P(\text{cell } (0{,}1)) = \frac{1}{2}$ (the other two cells are ignored).

Comparison of Ordinal and Multinomial Model

Hence the p-value is

$$P(X \leq 104) + P(X \geq 175), \text{ where } X \sim \text{Binom}(\pi = \tfrac{1}{2}, n = 279)$$

In R we can compute this as

```
> 2*pbinom(104,175+104,prob=0.5)
[1] 2.531092e-05

# yes, the p-value is smaller than 0.01, which is
# already a very strict significance level
```

The p-value is very small, so we conclude that the ordinal model has "significantly higher" accuracy than the multinomial model.

Exploiting dependence of predictions

If predictions were independent, we would have gotten the following table:

```
> wine.comp
             wine.multi.corr
wine.polr.corr   0   1
             0 293 357
             1 428 521

# is the difference in error rate (= 1-accuracy) significant?

> 2*pbinom(357,357+428,prob=0.5)
[1] 0.0124263
```

Now the p-value is much higher for the same difference in accuracy!