

# Logistic Regression: Mixed-Effects Models

2019-04-28

In this set of notes, you will learn about generalized mixed-effects models and how we can extend the logistic regression model to account for non-independence.

## Dataset and Research Question

In this set of notes, we will use data from the file *math.csv*. These data include the responses (correct = 1; incorrect = 0) for 10 items on a standardized math test for  $n = 30$  examinees.

```
# Load libraries
library(AICcmodavg)
library(broom)
library(corr)
library(dplyr)
library(ggplot2)
library(lme4)
library(readr)
library(sm)
library(tidyr)

# Read in data
math = read_csv(file = "~/Documents/github/epsey-8252/data/math.csv")

# View data
head(math)
```

```
# A tibble: 6 x 11
  examinee item_01 item_02 item_03 item_04 item_05 item_06 item_07 item_08
  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1       1       1       0       0       0       0       1       0       0
2       2       2       1       1       1       1       1       1       0
3       3       3       0       0       1       0       1       0       1
4       4       4       0       0       0       1       1       0       1
5       5       5       0       1       0       0       1       0       0
6       6       6       0       1       1       1       0       1       1
# ... with 2 more variables: item_09 <dbl>, item_10 <dbl>
```

Item response theory (IRT) is the dominant psychometric paradigm for analysis of assessment data. It also is the primary vehicle by which assessments are designed and scored. It is based on the simple idea that the probability of an examinee providing a correct response to an assessment item is a function of the examinee's "ability" and properties of the item. IRT has many useful characteristics including:

- Examinee's "ability" estimates are invariant with respect to the assessment.
- Item characteristics are estimated to be independent of the sample of examinees.
- It allows for variation in the reliability across the entire range of ability estimates

## Rasch Model

In the 1950's, Georg Rasch was working on how to measure reading ability for the Danish military. He developed a theory of measurement (Rasch measurement) that laid out a probabilistic model for observing an examinee's response. In this model, the probability of an observed response (e.g. correct/incorrect answer) is modeled as a function of the examinee's ability (on some latent trait) and the test items' parameters. Specifically, the probability of a correct response is modeled as a logistic function of the difference between the examinee's ability and the item's difficulty parameter.

$$\pi_{ij} = \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}}$$

where,  $\pi_{ij}$  is the probability of examinee  $i$  responding correctly to item  $j$ ,  $\theta_i$  is examinee  $i$ 's ability level on the latent trait, and  $\beta_j$  is the difficulty parameter of item  $j$ . Note this statistical model is often referred to as the *one-parameter logistic model* (1PL-model) to differentiate it from the more general philosophical approach that is "Rasch modeling".

### Log-Odds of a Correct Response

We can also write this as log-odds,

$$\begin{aligned} \ln \left[ \frac{\pi_{ij}}{1 - \pi_{ij}} \right] &= \ln \left[ \frac{\frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}}}{1 - \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}}} \right] \\ &= \ln \left[ \frac{\frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}}}{\frac{1 + e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}} - \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}}} \right] \\ &= \ln \left[ \frac{\frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}}}{\frac{1}{1 + e^{\theta_i - \beta_j}}} \right] \\ &= \ln \left[ e^{\theta_i - \beta_j} \right] \\ &= \theta_i - \beta_j \end{aligned}$$

Thus, the log-odds of answering an item correctly is the difference between the examinee's ability level and the difficulty of the item.

## Expressing the Rasch Model using a Mixed-Effects Model

Consider a generalized mixed-effects model that includes a set of dummy variables indicating the item (no fixed-effect of intercept) and a random-effect of intercept. This model will use a binomial error structure and the logit link function to predict log-odds of answering the item correctly. For example, the model for three items will include three dummy variables:

- `item_01` = 1 if the item is the first item and 0 otherwise
- `item_02` = 1 if the item is the second item and 0 otherwise

- `item_03 = 1` if the item is the third item and 0 otherwise

We can include effects for all three dummy variables in the model if we omit the intercept from the model. Mathematically, we can write the link function as:

$$\ln \left[ \frac{\pi_{ij}}{1 - \pi_{ij}} \right] = \beta_1(\text{Item\_01}_{ij}) + \beta_2(\text{Item\_02}_{ij}) + \beta_3(\text{Item\_03}_{ij}) + b_{0j}$$

where  $i$  represents the  $i$ th item and  $j$  is the  $j$ th examinee. (Here item responses are clustered in examinees.)

Substituting the appropriate 0/1 values into the equation we end up with three equations, one for each item:

$$\text{Item 1 : } \ln \left[ \frac{\pi_{ij}}{1 - \pi_{ij}} \right] = b_{0j} + \beta_1$$

$$\text{Item 2 : } \ln \left[ \frac{\pi_{ij}}{1 - \pi_{ij}} \right] = b_{0j} + \beta_2$$

$$\text{Item 3 : } \ln \left[ \frac{\pi_{ij}}{1 - \pi_{ij}} \right] = b_{0j} + \beta_3$$

Notice that this is essentially the same equation as is produced by Rasch. The ability parameter is equivalent to the random-effect in the mixed-effect model. The  $\beta$ -terms are related to the difficulty parameters in the Rasch model. The only difference is that Rasch expressed his model as a *difference* and the mixed-effects model is expressed as a *sum*. In the mixed-effects models, the  $\beta$ -terms represent **item easiness** rather than item difficulty.

We can easily re-express the mixed-effects model using a difference rather than a sum:

$$\text{Item 1 : } \ln \left[ \frac{\pi_{ij}}{1 - \pi_{ij}} \right] = b_{0j} - (-\beta_1)$$

$$\text{Item 2 : } \ln \left[ \frac{\pi_{ij}}{1 - \pi_{ij}} \right] = b_{0j} - (-\beta_2)$$

$$\text{Item 3 : } \ln \left[ \frac{\pi_{ij}}{1 - \pi_{ij}} \right] = b_{0j} - (-\beta_3)$$

Thus, if we want item difficulties, we simply need to take the negative of the  $\beta$ -terms.

## Fitting the Rasch Model using `glmer()`

Prior to fitting the mixed-effects model we need to convert our data from the wide to the long/tidy format.

```
# Re-structure to long format
math_long = math %>%
  gather(key = "item", value = "correct", item_01:item_10) %>%
  arrange(examinee, item)

# View data
head(math_long, 15)
```

```
# A tibble: 15 x 3
  examinee item    correct
  <dbl> <chr>    <dbl>
1       1 item_01      0
2       1 item_02      0
3       1 item_03      0
4       1 item_04      0
5       1 item_05      0
6       1 item_06      1
7       1 item_07      0
8       1 item_08      0
9       1 item_09      0
10      1 item_10      0
11      2 item_01      1
12      2 item_02      1
13      2 item_03      1
14      2 item_04      1
15      2 item_05      1
```

Now, we fit a model using the `glmer()` function that includes fixed-effects (dummy variables) for each of the five items and a random-effect of intercept. Importantly, we also omit the fixed-effect of intercept from the model. To do this we use `-1` rather than `1` in the fixed-effects part of the model formula. We define this model to incorporate a binomial error structure and use the logit link function.

```
# The -1 omits the intercept
glmer.1 = glmer(correct ~ -1 + item + (1|examinee), data = math_long, family = binomial(link = "logit"))
summary(glmer.1)
```

Generalized linear mixed model fit by maximum likelihood (Laplace

Approximation) [glmerMod]

Family: binomial ( logit )

Formula: correct ~ -1 + item + (1 | examinee)

Data: math\_long

AIC	BIC	logLik	deviance	df.resid
359	400	-168	337	289

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.709	-0.645	0.320	0.596	2.770

Random effects:

Groups	Name	Variance	Std.Dev.
examinee	(Intercept)	0.756	0.87

Number of obs: 300, groups: examinee, 30

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z )
itemitem_01	-1.378	0.492	-2.80	0.00508 **
itemitem_02	-1.176	0.473	-2.48	0.01297 *
itemitem_03	0.155	0.426	0.36	0.71712
itemitem_04	0.637	0.439	1.45	0.14666
itemitem_05	1.596	0.514	3.11	0.00190 **

```

itemitem_06    1.597    0.514    3.11  0.00190 **
itemitem_07   -0.159    0.427   -0.37  0.71012
itemitem_08   -0.641    0.440   -1.46  0.14495
itemitem_09    1.375    0.490    2.81  0.00503 **
itemitem_10    1.844    0.546    3.38  0.00073 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Correlation of Fixed Effects:

```

      itm_01 itm_02 itm_03 itm_04 itm_05 itm_06 itm_07 itm_08 itm_09
itemitem_02 0.130
itemitem_03 0.117 0.122
itemitem_04 0.104 0.110 0.137
itemitem_05 0.074 0.081 0.118 0.125
itemitem_06 0.074 0.081 0.118 0.125 0.122
itemitem_07 0.124 0.128 0.139 0.133 0.111 0.111
itemitem_08 0.130 0.133 0.133 0.124 0.097 0.097 0.137
itemitem_09 0.081 0.088 0.124 0.129 0.125 0.125 0.117 0.104
itemitem_10 0.067 0.073 0.111 0.119 0.118 0.118 0.104 0.090 0.120

```

Here the coefficient estimates represent item-easiness. To obtain the difficulty values we take the negative of each estimate.

```
-fixef(glmer.1)
```

```

itemitem_01 itemitem_02 itemitem_03 itemitem_04 itemitem_05 itemitem_06
    1.378      1.176     -0.155     -0.637     -1.596     -1.597
itemitem_07 itemitem_08 itemitem_09 itemitem_10
    0.159      0.641     -1.375     -1.844

```

In Rasch's interpretation of item-difficulties, these represent the ability level of the examinee needed to have a greater than 50/50 chance (odds = 1) of answering the item correctly. (Remember that ability is equivalent to the random-effect in the model and is therefore assumed to be normally distributed with a mean of 0; average ability = 0)

For example, interpreting the item-difficulty estimate associated with Item 01:

An examinee with ability of 1.38 (higher than average ability) would have a probability of 0.5 of answering Item 01 correctly. Examinees with ability levels below 1.38 would have a probability of less than 0.5 of answering the item correctly, and those with ability levels above 1.38 would have a probability higher than 0.5 of answering the item correctly.

Based on the difficulty levels, Items 03, 04, 05, 06, 09, and 10 would be considered “easy” as the item-difficulty values are negative, indicating that an examinee of average ability (=0) would all have a greater than 50/50 chance of answering them correctly. Items 01, 02, 07, and 08 would be considered “difficult” as the item-difficulty values are positive; an examinee of average ability (=0) would all have a less than 50/50 chance of answering them correctly.

## Examinee Ability-Levels

To obtain the ability-levels for the examinees, we need to output the estimated random-effects.

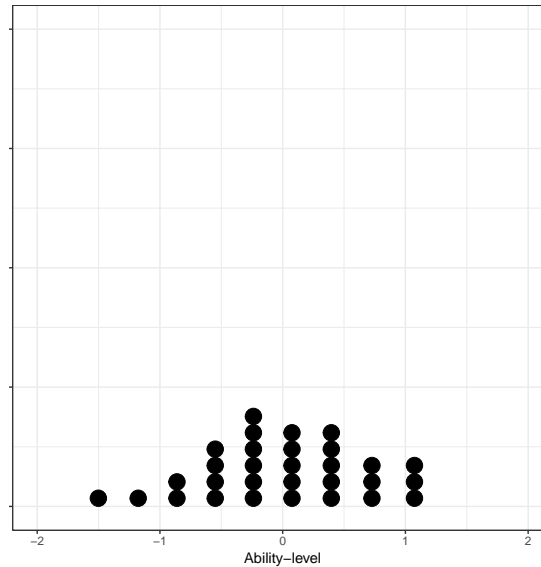


Figure 1: Dotplot showing the ability-levels for the 30 examinees.

```
# Extract the ability-levels
abilities = ranef(glmer.1)$examinee[ , 1]

# Output the first six examinees' ability-levels
head(abilities)
```

```
[1] -1.5016  1.0739  0.0759  0.0759 -0.8612  0.3964
```

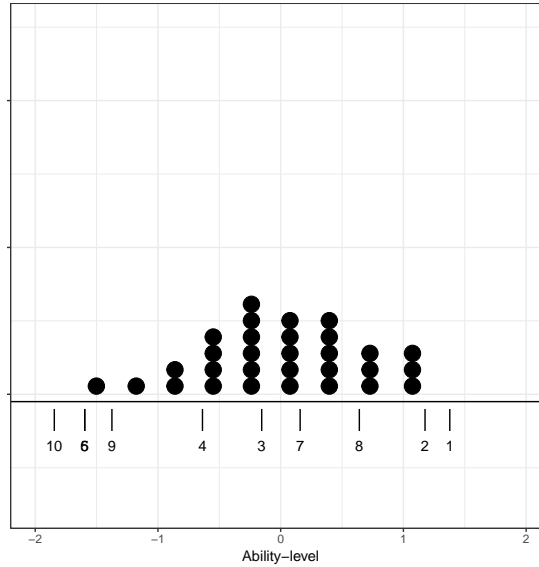
Negative values for ability indicate the examinee is below average on the latent trait (math ability) and a positive ability value indicates the examinee has higher than average math ability. Below is a dotplot of the 30 examinee's ability levels:

## Wright Map

One really great feature of using IRT is that the item difficulties and the examinee abilities are on the same scale. Because of that, we can display both sets of estimates in the same plot.

```
# Put abilities into a data frame for plotting
difficulties = data.frame(
  item = 1:10,
  beta = -fixef(glmer.1)
)

ggplot(data = examinees, aes(x = theta)) +
  geom_dotplot() +
  theme_bw() +
  scale_y_continuous(name = "", labels = NULL, limits = c(-0.3, 1)) +
  scale_x_continuous(name = "Ability-level", limits = c(-2, 2)) +
  geom_hline(yintercept = -0.02) +
  geom_segment(data = difficulties, aes(x = beta, xend = beta, y = -.04, yend = -0.1)) +
  geom_text(data = difficulties, aes(x = beta, y = -0.14, label = item))
```



This display is called a *Wright Map* and gives us information about the ability levels of our sample of examinees and also which ability levels the items on the assessment are tapping. For example, we would expect that the six highest ability students would have a greater than 50% chance of responding correctly to all of the items except Item 01 and Item 02. While the lowest ability student would have a greater than 50% chance of responding correctly to only Items 10 and 06.

In thinking about assessment design, the first two items are the most difficult; it may be better to start the assessment with easier items. It may also be beneficial to write (1) some more difficult items (there do not seem to be any items that tap ability levels higher than 1 SD above the average ability) and (2) some easy items that fill in the gaps around ability-levels at 1 SD below average. Writing items for a particular ability-level is easier said than done.

## Item Characteristic Curve (ICC)

It can be useful to plot the probability of answering each item correctly for a range of examinee ability levels. This plot is referred to as an *item characteristic curve*. Below we outline the steps to create such a plot for Item 01.

- Compute the log-odds of answering Item 01 correctly for a range of ability levels using the item equation:

$$\ln \left[ \frac{\hat{\pi}_{ij}}{1 - \hat{\pi}_{ij}} \right] = b_{0j} - \hat{\beta}_1$$

where  $\hat{\beta}_1$  is the estimated item *difficulty* level. For Item 01, since the difficulty level is  $-(-1.378) = +1.378$ ,

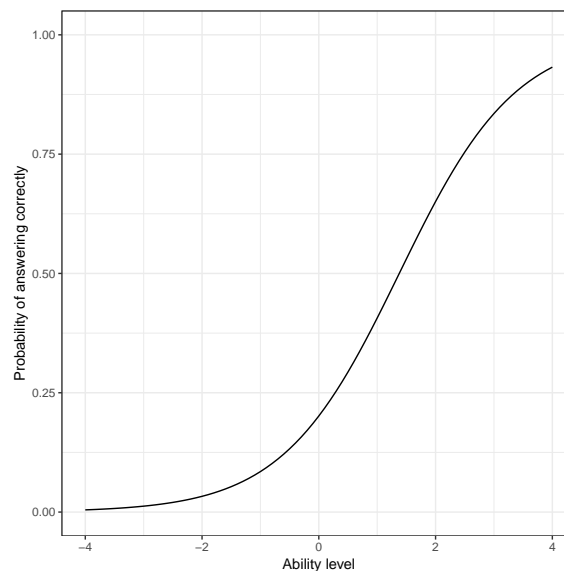
$$\ln \left[ \frac{\hat{\pi}_{ij}}{1 - \hat{\pi}_{ij}} \right] = b_{0j} - 1.378$$

Then we,

- Convert the predicted log-odds to odds and then to probability.
- Plot the predicted probabilities versus the ability levels.

Here is some syntax to do this for Item 01.

```
data.frame(  
  b = seq(from = -4, to = 4, by = 0.1)  
) %>%  
  mutate(  
    log_odds = b - 1.378,  
    odds = exp(log_odds),  
    prob = odds / (1 + odds)  
  ) %>%  
  ggplot(aes(x = b, y = prob)) +  
    geom_line() +  
    theme_bw() +  
    xlab("Ability level") +  
    ylab("Probability of answering correctly") +  
    ylim(0, 1)
```



```
ggtitle("Item 01")
```

```
$title  
[1] "Item 01"  
  
attr(,"class")  
[1] "labels"
```

From this plot we see,

- The ability-level that corresponds to answering Item 01 correctly 50% of the time (probability = 0.5) is 1.378; the item's difficulty level.
- The “slope” of the curve represents how well the item can differentiate between examinees of different ability levels; the steeper the “slope” the better it can differentiate. This is referred to as *item discrimination*. In the Rasch model this is constrained to the value 1 for all items.



## ICC for all Items

Typically we want to show the ICC for each item on the assessment. To do this we need to create these estimates for each item.

```
# Create individual vectors
item = unique(math_long$item)
beta = -fixef(glmer.1)
b = seq(from = -4, to = 4, by = 0.1)

# Cross abilities with item names/difficulties
all_items = crossing(b, nesting(item, beta))

# Obtain probability of answering correctly for each item and ability
all_items = all_items %>%
  mutate(
    log_odds = b - beta,
    odds = exp(log_odds),
    prob = odds / (1 + odds)
  )

# View data
head(all_items)
```

```
# A tibble: 6 x 6
      b item      beta log_odds   odds   prob
<dbl> <chr>   <dbl>   <dbl>   <dbl> <dbl>
1    -4 item_01  1.38    -5.38  0.00462 0.00460
2    -4 item_02  1.18    -5.18  0.00565 0.00562
3    -4 item_03 -0.155   -3.85  0.0214  0.0209
4    -4 item_04 -0.637   -3.36  0.0346  0.0335
5    -4 item_05 -1.60    -2.40  0.0904  0.0829
6    -4 item_06 -1.60    -2.40  0.0904  0.0829
```

Now we show two plots of all 10 item ICCs. The first, which is more difficult to differentiate the items, places all of the ICCs on the same plot. This is pedagogically useful as you can see that the discrimination (“slope”) for all 10 curves are the same. The difference is how far to the left or right each curve is on the plot. Since the horizontal axis is ability-level, that corresponds to how difficult each item is; what is the ability-level need to have a probability of 0.5 of answering the item correctly.

```
ggplot(data = all_items, aes(x = b, y = prob, color = item)) +
  geom_line() +
  theme_bw() +
  xlab("Ability level") +
  ylab("Probability of answering correctly") +
  ylim(0, 1)
```

The second plot shows each of the different ICC curves in a separate panel. This makes it easier to explore the separate items.

```
ggplot(data = all_items, aes(x = b, y = prob)) +
  geom_line() +
  theme_bw() +
```

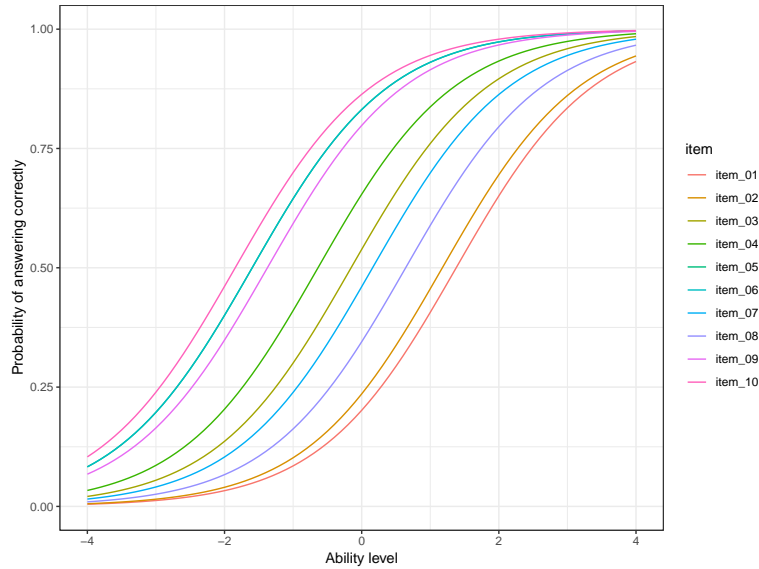


Figure 2: ICC for each of the 10 items.

```

xlab("Ability level") +
ylab("Probability of answering correctly") +
ylim(0, 1) +
facet_wrap(~item)

```

## Item Information

Another common plot in Rasch analysis is the item information plot. Information is akin to reliability in the classical test theory paradigm. The advantage is that with IRT we obtain information at the item-level rather than only at the test-level. And moreover, we can obtain information at each ability level to see if the test is “reliable” for examinees of varying ability-levels.

Information for Item  $j$  at a particular ability-level ( $\theta$ ) is simply computed as

$$I_j(\theta) = p_{j\theta}(1 - p_{j\theta})$$

where  $p_{j\theta}$  is the probability of an examinee with ability-level  $\theta$  responding correctly to Item  $j$ . Since we have the probability of responding correctly to each of the different items already, it is easy to compute the information as well.

```

# Obtain item information
all_items = all_items %>%
  mutate(
    info = prob * (1 - prob)
  )

```

Below we plot the information curves for each of the 10 items

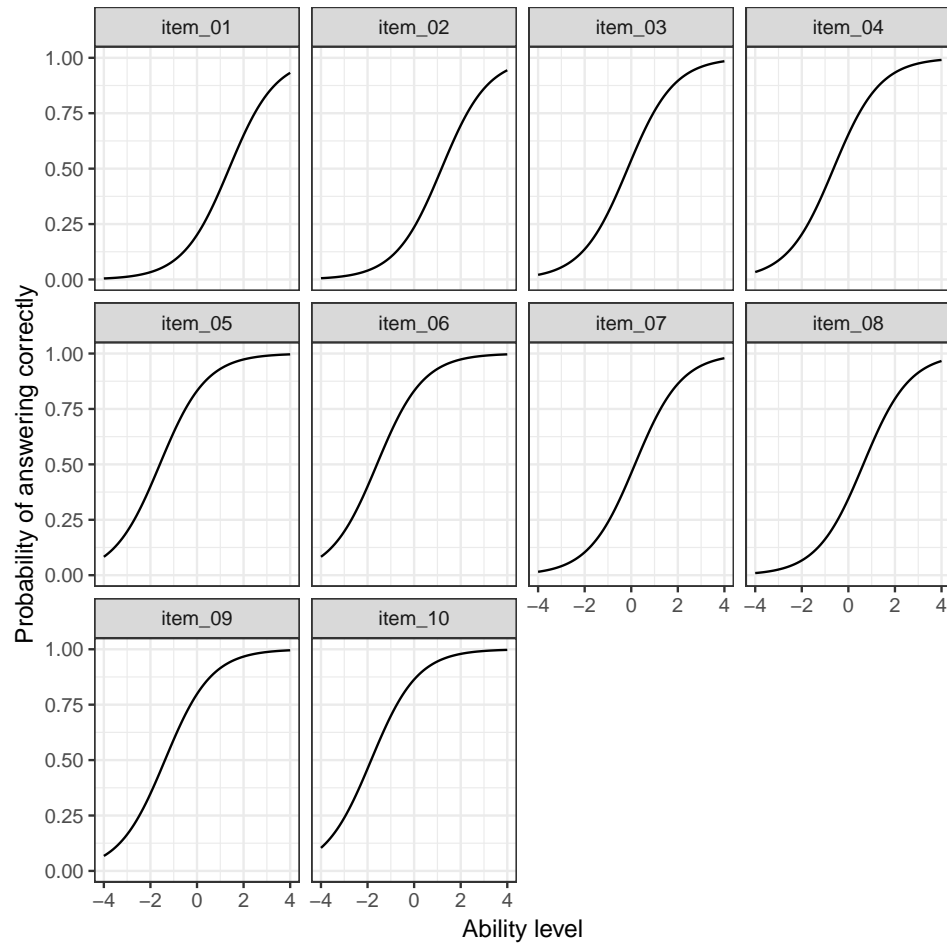


Figure 3: ICC for each of the 10 items.

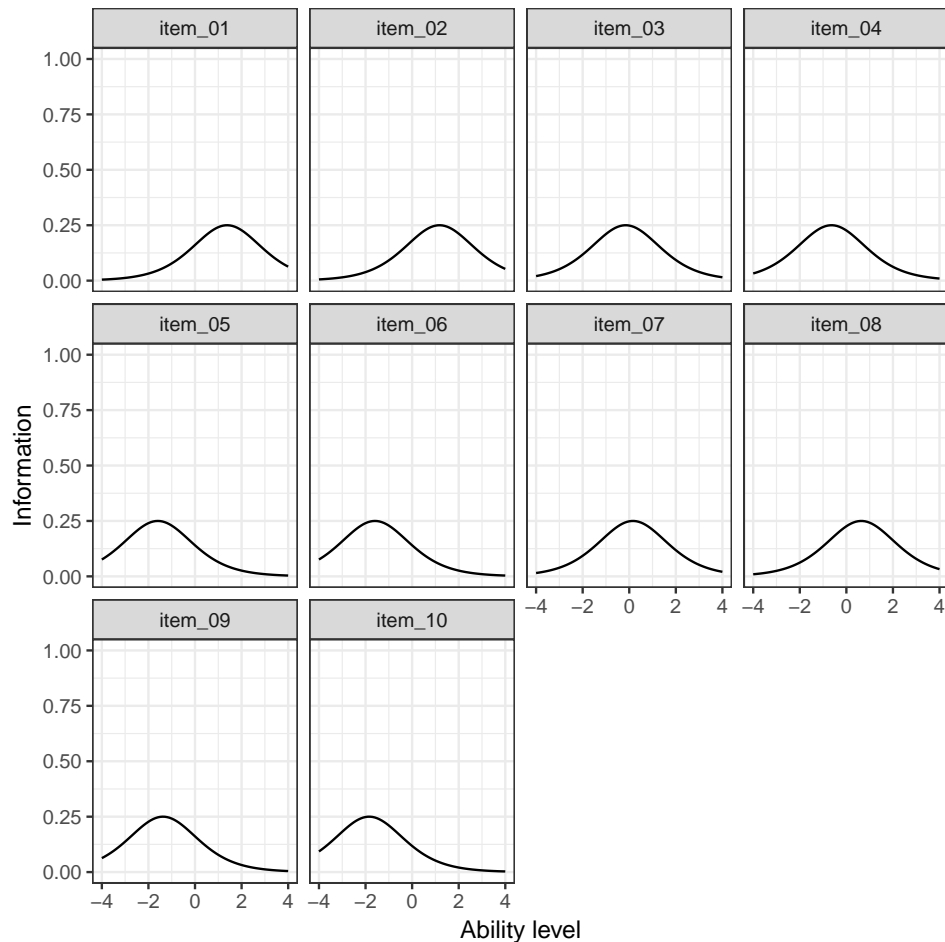


Figure 4: Item information curve for each of the 10 items.

```
ggplot(data = all_items, aes(x = b, y = info)) +
  geom_line() +
  theme_bw() +
  xlab("Ability level") +
  ylab("Information") +
  ylim(0, 1) +
  facet_wrap(~item)
```

Examining the information curve for Item 01 we see that this item is giving us the most information when examinees have an ability-level near 1.378 (the item's difficulty level). It is not giving us a lot of information when the examinee has an ability-level that is really low. Item 10, on the other hand, gives us more information about low-ability examinees and not a lot of information about high-ability examinees.

## Test-Level Information

We can sum the item-level information at each ability-level to get an indication of the total information encompassed in the assessment.

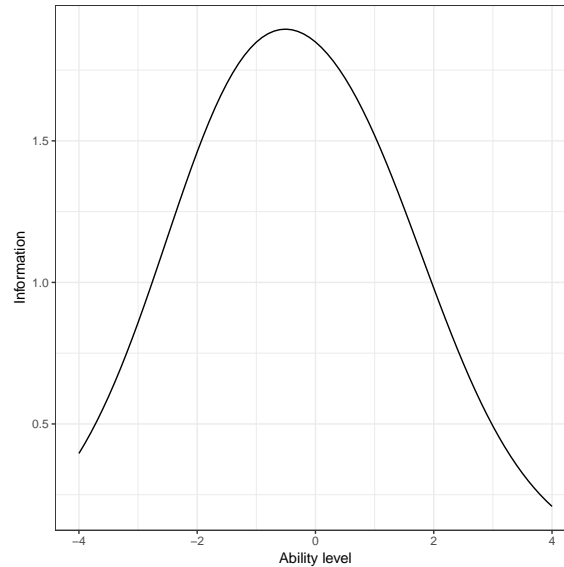


Figure 5: Total information curve for all 10 items.

```
# Obtain total information at each ability-level
test_info = all_items %>%
  group_by(b) %>%
  summarize(
    total_info = sum(info)
  )

# View data
head(test_info)
```

```
# A tibble: 6 x 2
      b total_info
  <dbl>     <dbl>
1  -4      0.396
2 -3.9      0.431
3 -3.8      0.468
4 -3.7      0.508
5 -3.6      0.551
6 -3.5      0.596
```

Plotting the test information curve

```
ggplot(data = test_info, aes(x = b, y = total_info)) +
  geom_line() +
  theme_bw() +
  xlab("Ability level") +
  ylab("Information")
```

From this curve we see that the test is giving us more information about the low ability-level examinees than the high ability-level examinees.

## Standard Error of Ability

Information has an inverse relationship with the uncertainty associated with estimating examinees' ability-levels. If we wanted to give an uncertainty estimate we could compute the SE associated with each ability-estimate using,

$$SE = \sqrt{\frac{1}{I(\theta)}}$$

where  $I(\theta)$  is the total information for a given ability-level  $\theta$ .

```
# Compute uncertainty for each ability-level
test_info = test_info %>%
  mutate(
    se = sqrt(1 / total_info)
  )

# View data
head(test_info)
```

```
# A tibble: 6 x 3
      b total_info    se
  <dbl>    <dbl> <dbl>
1   -4      0.396  1.59
2  -3.9      0.431  1.52
3  -3.8      0.468  1.46
4  -3.7      0.508  1.40
5  -3.6      0.551  1.35
6  -3.5      0.596  1.30
```

Plotting this we can see that the uncertainty varies by ability estimate. The least amount of uncertainty is for slightly lower than average ability-level examinees (where we have the most information). We have much more uncertainty for examinees that have really low or really high ability-levels.

```
ggplot(data = test_info, aes(x = b, y = se)) +
  geom_line() +
  theme_bw() +
  xlab("Ability level") +
  ylab("Uncertainty/Standard Error of Measurement")
```

In application, a test score (or ability estimate) is a point estimate of the examinee's true ability level. Because of the information in the test, we are more certain about some of these scores or ability estimates than others. Just like in statistics, we can indicate this uncertainty by computing a confidence interval for the score/ability level. Based on the plot, the CIs for examinees with high ability estimates will be wider (more uncertainty) than the CIs for examinees with low ability estimates.

## Other Resources

In addition to the notes and what we cover in class, there are many other resources for learning about using mixed-effects binomial logistic regression models for analyzing test data. Here are some resources that may be helpful in that endeavor:

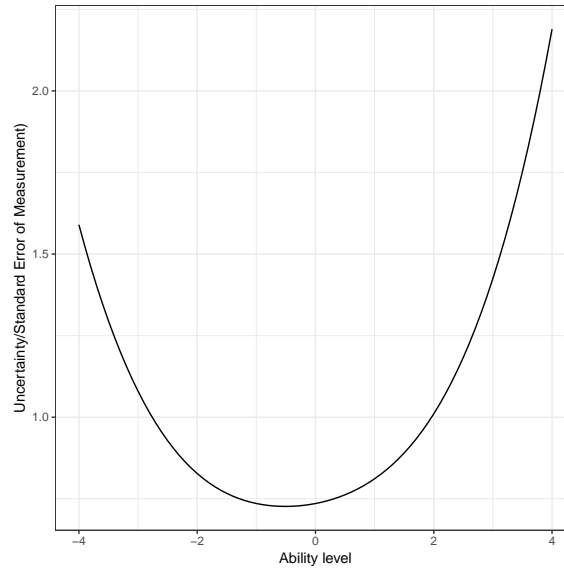


Figure 6: Uncertainty by ability-level.

- De Boeck, P., Bakker, M., Zwitser, R., Nivard, M., Hofman, A., Tuerlinckz, F., & Partchev, I. (2011). [The estimation of item response models with the lmer Function from the lme4 package in R.](#) *Journal of Statistical Software*, 39(12), 1–28.
- Rijmen, F., Tuerlinckx, F., De Boeck, P., & Kuppens, P. (2003). [A nonlinear mixed model framework for item response theory.](#) *Psychological Methods*, 8(2), 185–205.