R Notebook

This notebook is for recreating the relevant plots from wordbank: Source: https://wordbank-book.stanford.edu/psychometrics.html Code adapted from: https://github.com/langcog/wordbank-book

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#install.packages("wordbankr")
library(wordbankr)
library(ggstance)
```

```
##
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

Load items from wordbank

```
items <- get_item_data(language = "English (American)", form = "WS")
```

Load data from psychometrics

```
base::load("~/GitHub_C/wordbank-book/data/psychometrics/eng_ws_raw_data.Rds")

d_ws <- eng_ws %>%
  mutate(produces = value == "produces") %>%
  filter(!is.na(category)) %>%
  select(data_id, produces, age, production, sex, definition)

base::load("~/GitHub_C/wordbank-book/data/psychometrics/eng_ws_mods_2pl.Rds")
```

```
## Warning: namespace 'mirt' is not available and has been replaced
## by .GlobalEnv when processing object 'mod_2pl'
```

```
d_ws_summary <- d_ws %>%
  group_by(data_id, sex, age) %>%
  summarise(production = production[1]) %>%
  right_join(fscores_2pl %>%
               mutate(data_id = as.numeric(data_id))) %>%
  filter(!is.na(sex))
```

```
## `summarise()` has grouped output by 'data_id', 'sex'. You can override using the `.groups` argument.
```

```
## Joining, by = "data_id"
```

Figure 4.5: Item characteristic curves for a set of individual items from the English WS sample.

```
thetas <- seq(-6,6,.1)
irt4pl <- function(a, d, g, u, theta = seq(-6,6,.1)) {
  p = g + (u - g) * boot::inv.logit(a * (theta + d))
  return(p)
}
irt2pl <- function(a, d, theta = seq(-6,6,.1)) {
  p = boot::inv.logit(a * (theta + d))
  return(p)
}

examples <- c("table","mommy*","trash","yesterday")
iccs <- coefs_2pl %>%
  filter(definition %in% examples) %>%
  split(.$definition) %>%
  map_df(function(d) {
    return(data_frame(definition = d$definition,
                      theta = thetas,
                      p = irt2pl(d$a1, d$d, thetas)))
  })
```

```
## Warning: `data_frame()` is deprecated as of tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
ggplot(iccs,
       aes(x = theta, y = p)) +
  geom_line() +
  facet_wrap(~definition) +
  xlab("Ability") +
  ylab("Probability of production")
```

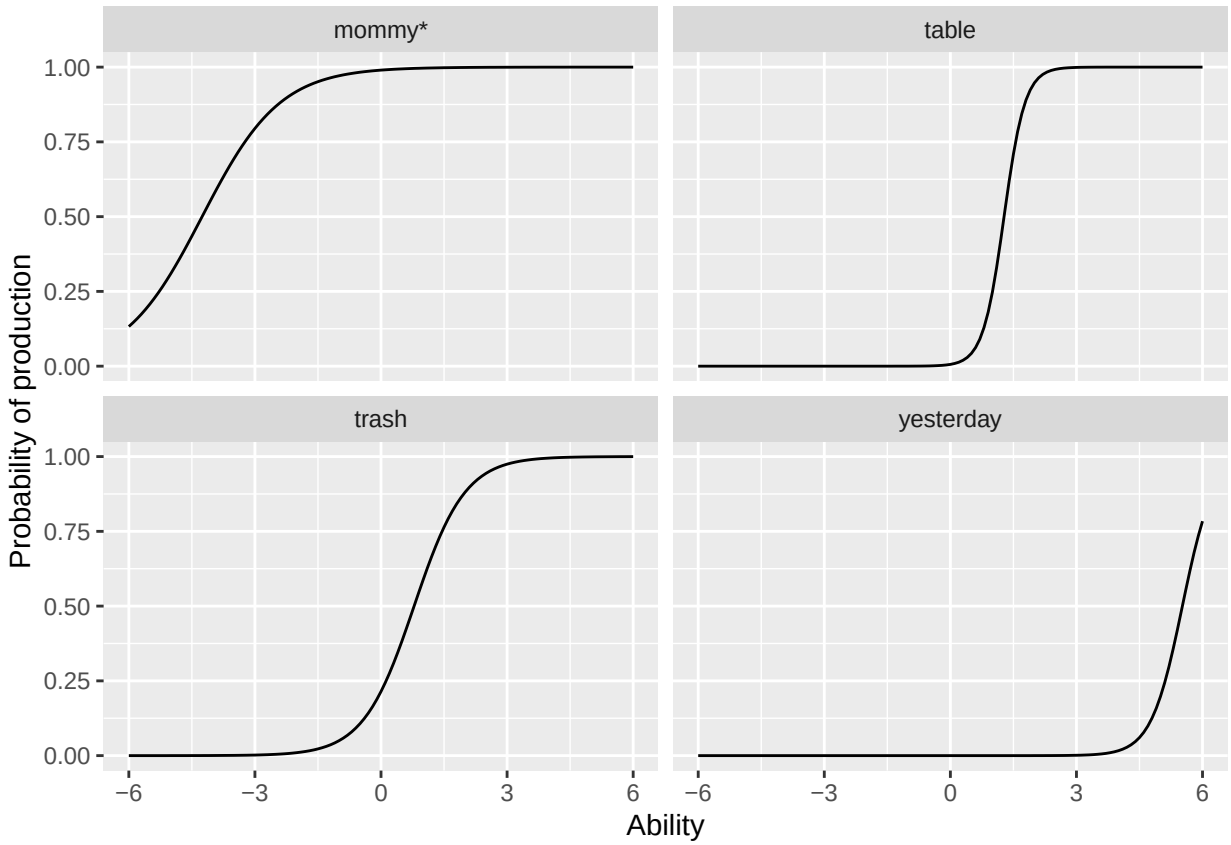


Figure 4.6: Words (points), plotted by their difficulty and discrimination parameters, as recovered by the 2-parameter IRT model (see text). Outliers are labeled.

```
ggplot(coefs_2pl,
       aes(x = a1, y = -d)) +
  geom_point(alpha = .3) +
  ggrepel::geom_text_repel(data = filter(coefs_2pl,
                                         -d < -3.8 | -d > 5.3 | a1 > 4 | a1 < 1),
                           aes(label = definition), size = 3) +
  xlab("Discrimination") +
  ylab("Difficulty")
```

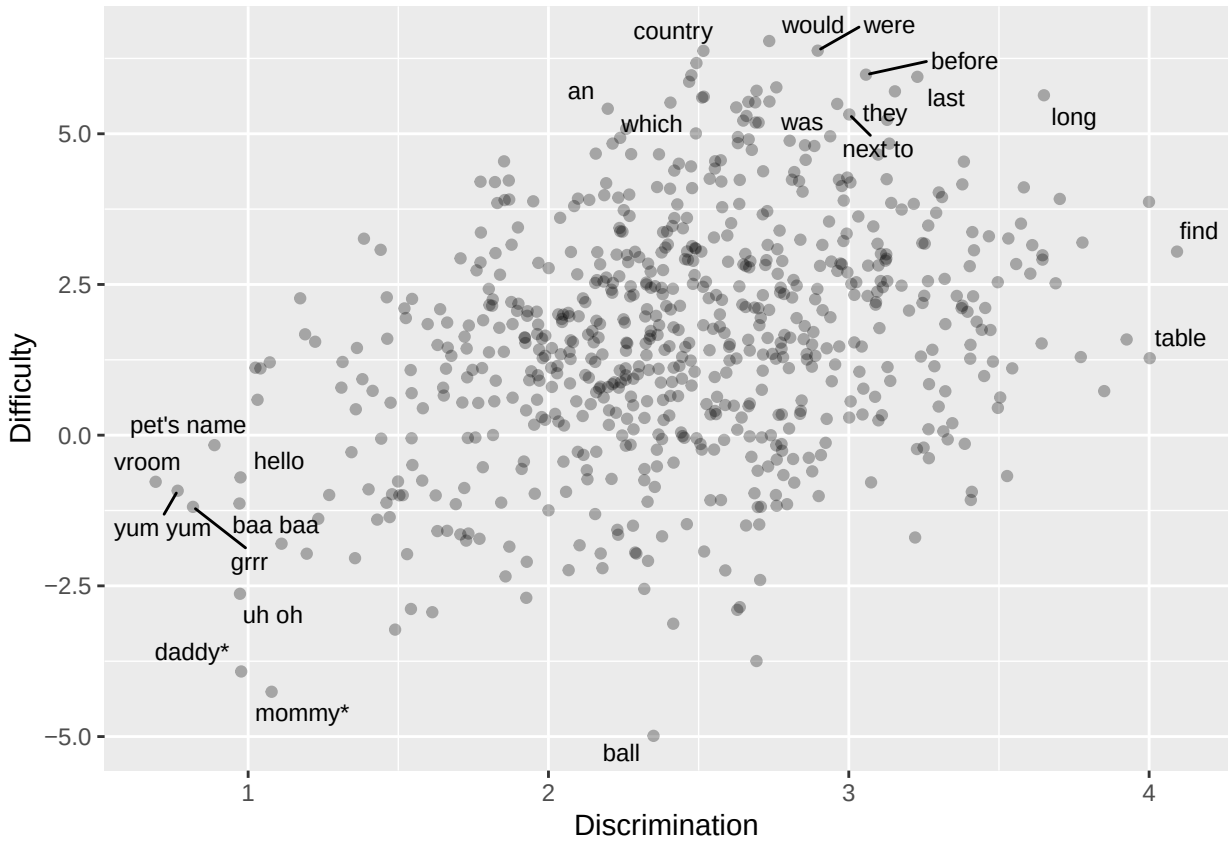


Figure 4.7: Words (points), plotted now by their lower and upper bound parameters from the 4-parameter IRT model.

```
base::load("~/GitHub_C/wordbank-book/data/psychometrics/eng_ws_mods_4pl.Rds")

ggplot(coefs_4pl, aes(x = g, y = u)) +
  geom_point(alpha = .3) +
  ggrepel::geom_text_repel(data = filter(coefs_4pl,
                                         abs(g) > .4 | u < .75),
                           aes(label = definition), size = 3) +
  xlab("Lower bound (high base rate)") +
  ylab("Upper bound (not known by many)")
```

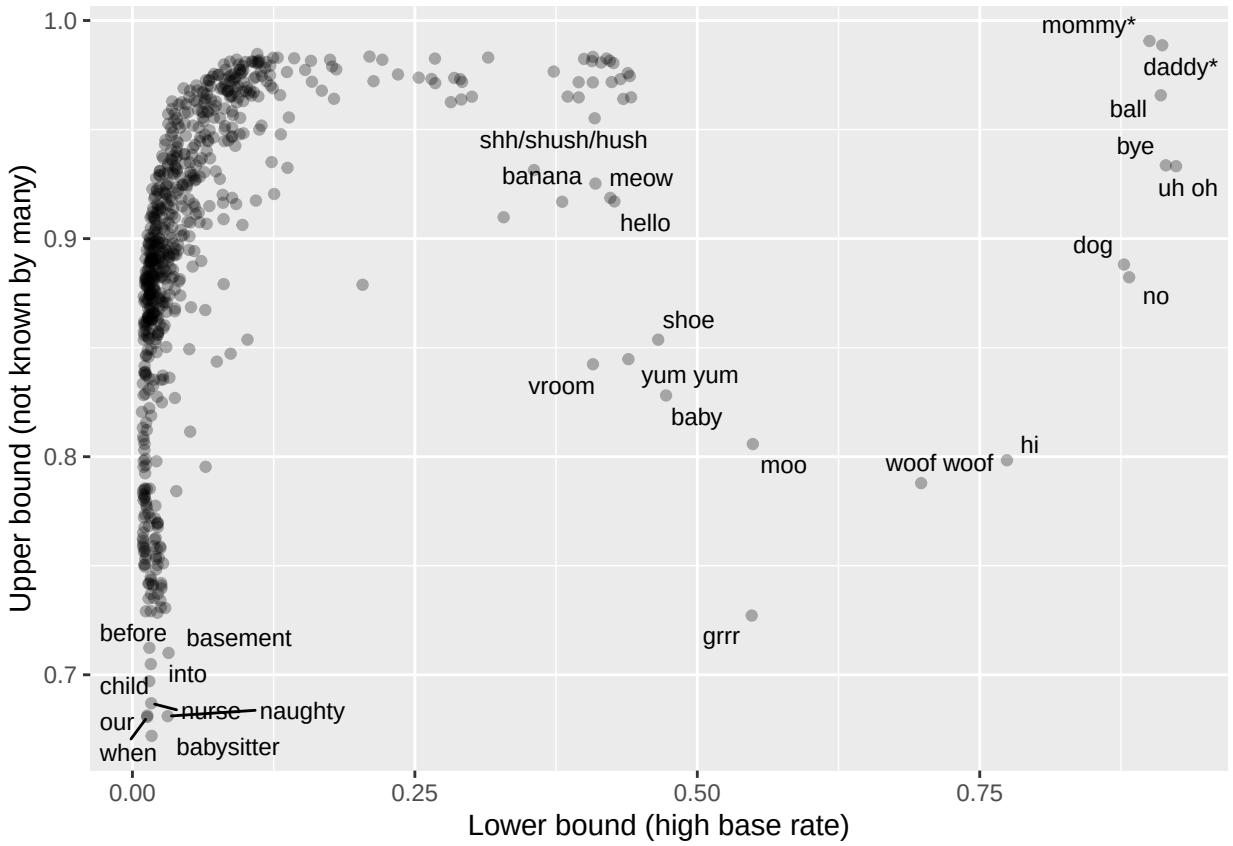


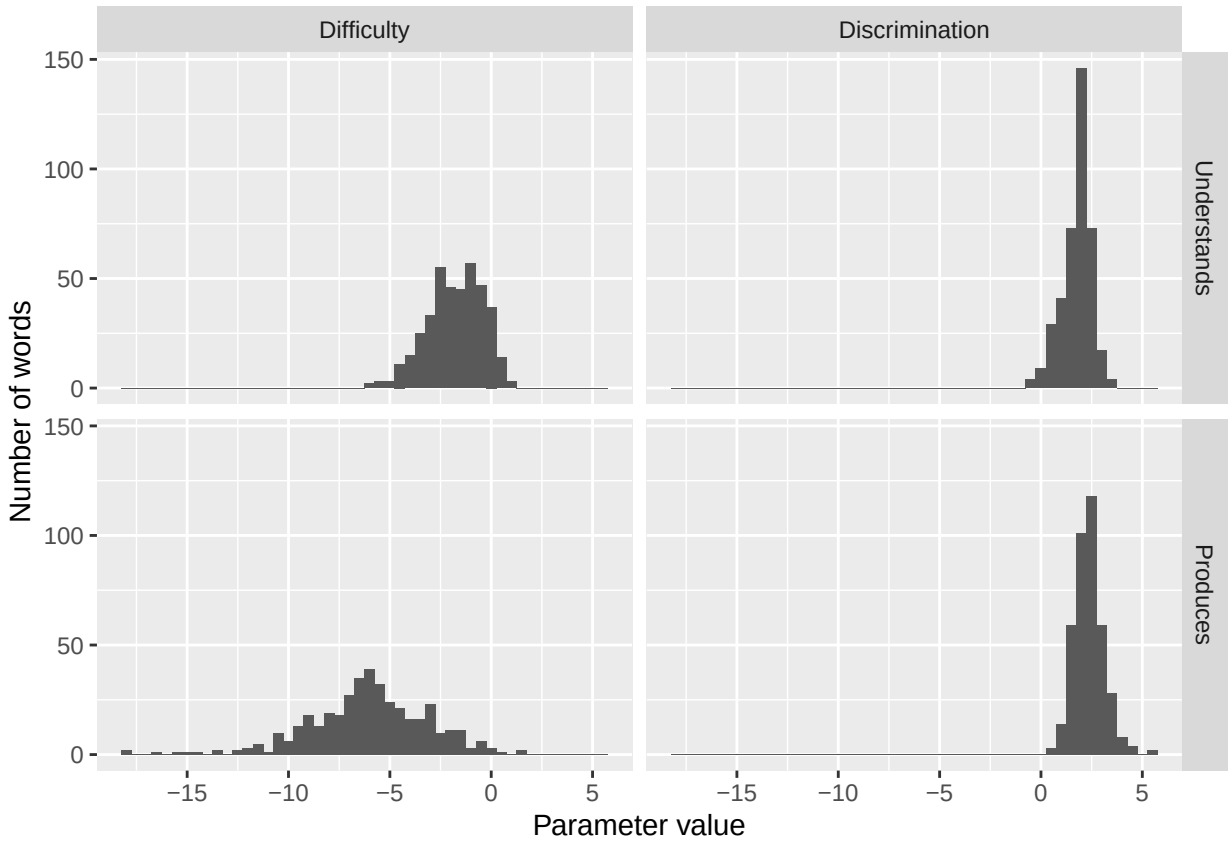
Figure 4.8: Histograms of words' difficulty and discrimination parameters, for comprehension and production.

```
base::load("~/GitHub_C/wordbank-book/data/psychometrics/eng_wg_mods_2pl.Rds")

coefs_2pl_wg <- bind_rows(coefs_2pl_wg_produces %>%
                            mutate(measure = "Produces"),
                          coefs_2pl_wg_understands %>%
                            mutate(measure = "Understands"))

wg_comp_prod <-
  coefs_2pl_wg %>%
  select(a1, d, measure) %>%
  gather(parameter, value, a1, d) %>%
  mutate(parameter = fct_recode(parameter,
                                Discrimination = "a1",
                                Difficulty = "d") %>%
           relevel("Difficulty"),
         measure = fct_relevel(measure, "Understands"))

ggplot(wg_comp_prod,
       aes(x = value)) +
  geom_histogram(binwidth = .5) +
  facet_grid(measure ~ parameter) +
  # xlim(-5,5) +
  xlab("Parameter value") +
  ylab("Number of words")
```




```
wg_comp_prod_summary <- wg_comp_prod %>%
  group_by(measure, parameter) %>%
  summarise(value = mean(value))
```

4.3.3 Lexical category effects on item performance

```
coefs_2pl <-
  coefs_2pl %>%
  left_join(
    items %>%
    filter(language == "English (American)", form == "WS")
  ) %>%
  mutate(
    lexical_class_label =
      lexical_class %>% factor() %>% fct_relabel(~.x %>% as.character())
  )

class_summary <- coefs_2pl %>%
  group_by(lexical_class, lexical_class_label) %>%
  summarise(sd_a1 = sd(a1, na.rm=TRUE),
            a1 = mean(a1))

a <- ggplot(coefs_2pl,
       aes(x = a1, y = -d, col = lexical_class_label)) +
  geom_point(alpha = .3) +
  ggrepel::geom_text_repel(data = filter(coefs_2pl,
                                 a1 < 1 | a1 > 3.8 | -d > 5 | -d < -2.5),
                   aes(label = definition), size = 2,
```

```
                  show.legend = FALSE) +
  scale_colour_discrete(name = "Lexical class") +
  xlab("Discrimination") +
  ylab("Difficulty")

b <- ggplot(coefs_2pl,
       aes(x = a1, fill = lexical_class_label)) +
  geom_histogram() +
  scale_fill_discrete(name = "Lexical class") +
  xlab("Discrimination") +
  ylab("Number of words") +
  xlim(0,4)

gridExtra::grid.arrange(a, b)
```