Item Analysis

```r
library(tidyverse)
library(modelsummary)  # for summarizing data
```

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
```

```
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

```
library(TAM)
library(psych)  # for factor analysis
library(performance)  # for item-total correlation
library(boot)  # for bootstrapping
```

Cognitive Items

We will use a sample data set from the `TAM` package, which contains 143 students on 30 items. There were actually two versions of the data set, which `data.mc$raw` showing the raw responses (i.e., the response selected), and the scored responses (i.e., 0 = incorrect, 1 = correct).

```
data(data.mc)
```

Descriptives

```
mc_raw <- data.mc$raw  # get the raw choices portion
# Frequency table for the first two items
datasummary_skim(mc_raw[, 1:2], type = "categorical")
```

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
```

```
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
```

		N	%
I01	9	2	1.4
	A	130	90.9
	B	4	2.8
	C	3	2.1
	D	3	2.1
	NA	1	0.7
I02	8	1	0.7
	9	1	0.7
	A	5	3.5
	B	126	88.1
	C	4	2.8
	D	5	3.5
	NA	1	0.7

```
See help("Deprecated")
```

```
Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

```
Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

> **Q1**
>
> Which options do you think are likely to be the correct option for I10 and I20?
> Answer:

Here are the scored version

```
mc_scored <- data.mc$scored  # get the scored responses
# Frequency table for all items, with two digits
datasummary(All(mc_scored) ~ N + Mean + Median + SD, data = mc_scored)
```

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
```

```
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
```

```
See help("Deprecated")
```

```
Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

```
Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

> **Q2**
>
> Why are the medians and sd redundant in the above table?
> Answer:

Item Difficulty

This is the proportion of "1"s divided by the total number of responses. Note this is somewhat a misnomer, as an item with a higher "difficulty" level is easier.

```
colSums(mc_scored, na.rm = TRUE) / nrow(mc_scored)
```

```
      I01       I02       I03       I04       I05       I06       I07       I08
0.9090909 0.8811189 0.7762238 0.9020979 0.7622378 0.8671329 0.6573427 0.6153846
      I09       I10       I11       I12       I13       I14       I15       I16
0.4125874 0.2167832 0.3986014 0.4405594 0.7622378 0.3916084 0.2587413 0.5804196
      I17       I18       I19       I20       I21       I22       I23       I24
0.6293706 0.5804196 0.5594406 0.6293706 0.6573427 0.2867133 0.5874126 0.4685315
      I25       I26       I27       I28       I29       I30
0.5034965 0.3286713 0.1958042 0.3426573 0.2377622 0.5454545
```

> **Q3**
>
> Which item is the most difficult? Which is the least difficult?
> Answer:

> **Q4**
>
> Observe the item means and the standard deviations. What is the relation between the mean and the standard deviation?
> Answer:

	N	Mean	Median	SD
I01	142	0.92	1.00	0.28
I02	142	0.89	1.00	0.32
I03	142	0.78	1.00	0.41
I04	141	0.91	1.00	0.28
I05	141	0.77	1.00	0.42
I06	141	0.88	1.00	0.33
I07	140	0.67	1.00	0.47
I08	140	0.63	1.00	0.48
I09	140	0.42	0.00	0.50
I10	140	0.22	0.00	0.42
I11	140	0.41	0.00	0.49
I12	139	0.45	0.00	0.50
I13	139	0.78	1.00	0.41
I14	138	0.41	0.00	0.49
I15	138	0.27	0.00	0.44
I16	138	0.60	1.00	0.49
I17	138	0.65	1.00	0.48
I18	136	0.61	1.00	0.49
I19	136	0.59	1.00	0.49
I20	135	0.67	1.00	0.47
I21	135	0.70	1.00	0.46
I22	135	0.30	0.00	0.46
I23	134	0.63	1.00	0.49
I24	131	0.51	1.00	0.50
I25	131	0.55	1.00	0.50
I26	131	0.36	0.00	0.48
I27	129	0.22	0.00	0.41
I28	129	0.38	0.00	0.49
I29	128	0.27	0.00	0.44
I30	127	0.61	1.00	0.49

Item Discrimination

> The degree to which an item can distinguish respondents of different level of knowledge or skill, such that those with a higher level of knowledge or skill is more likely to answer an item correctly.

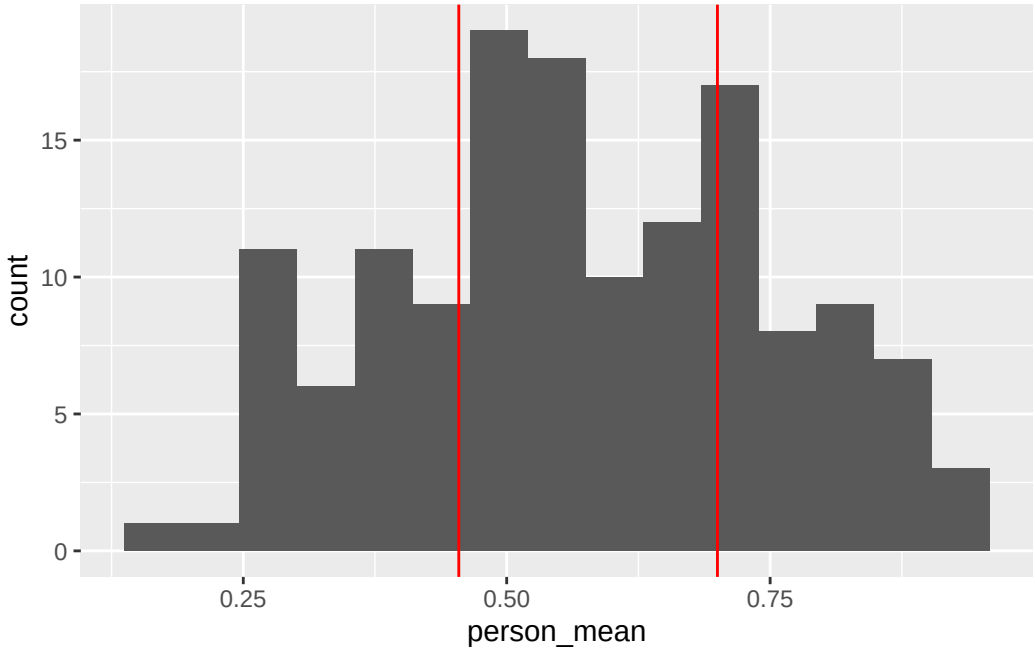
Internal criterion: Performance in the current test

```
# Create mean score
mc_scored$person_mean <- rowMeans(mc_scored[1:30], na.rm = TRUE)
# Top and bottom 27%
nrow(mc_scored) * .27  # 39 people
```

```
[1] 38.61
```

```
# Corresponding scores
bounds <- sort(mc_scored$person_mean)[c(39, 143 - 39)]
mc_scored <- mc_scored |>
    mutate(group = case_when(
        person_mean > bounds[2] ~ "upper",
        person_mean <= bounds[1] ~ "lower"
    ))
# Plot the distributions
ggplot(mc_scored, aes(x = person_mean)) +
    geom_histogram(bins = 15) +
    geom_vline(
        xintercept = bounds,
        col = "red"
    )
```

```
Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```



We can see the choices by the two groups, for example Item 5:

```
table(mc_scored$group, mc_raw[ , "I05"])
```

```
       A  B  C  D
lower  9  4 11 14
upper  0  0  0 35
```

which shows that the upper group all selects "D". Assuming D is the right choice (and it is), the item has a reasonable discrimination. On the other hand, consider Item 10:

```
table(mc_scored$group, mc_raw[ , "I10"])
```

```
       9  A  B  C  D
lower  2 11 10  7  8
upper  0 23  0  2 10
```

which seems more mixed.

We can compute the discrimination for item i as

$$D_i = P_{ui} - P_{li},$$

where P_u is the proportion correct in the upper group and P_l is that in the lower group. We can do this for all items:

```
mc_p <- mc_scored |>
    filter(!is.na(group)) |>
    group_by(group) |>
    summarize(across(I01:I30, .fns = list(P = ~ mean(.x, na.rm = TRUE))))
mc_p
```

```
# A tibble: 2 x 31
  group I01_P I02_P I03_P I04_P I05_P I06_P I07_P I08_P I09_P I10_P I11_P I12_P
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 lower 0.795 0.718 0.513 0.763 0.368 0.763 0.316 0.263 0.211 0.211 0.132 0.216
2 upper 1     0.971 0.971 1     1     0.943 0.943 0.857 0.771 0.286 0.743 0.857
# i 18 more variables: I13_P <dbl>, I14_P <dbl>, I15_P <dbl>, I16_P <dbl>,
#   I17_P <dbl>, I18_P <dbl>, I19_P <dbl>, I20_P <dbl>, I21_P <dbl>,
#   I22_P <dbl>, I23_P <dbl>, I24_P <dbl>, I25_P <dbl>, I26_P <dbl>,
#   I27_P <dbl>, I28_P <dbl>, I29_P <dbl>, I30_P <dbl>
```

And get the discrimination using the difference

```
mc_p[2, 2:31] - mc_p[1, 2:31]
```

```
      I01_P     I02_P     I03_P     I04_P     I05_P     I06_P     I07_P
1 0.2051282 0.2534799 0.4586081 0.2368421 0.6315789 0.1796992 0.6270677
     I08_P     I09_P      I10_P     I11_P     I12_P     I13_P     I14_P
1 0.593985 0.5609023 0.07518797 0.6112782 0.6409266 0.4594595 0.3552124
      I15_P     I16_P     I17_P     I18_P     I19_P     I20_P     I21_P
1 0.6030888 0.7552124 0.5583012 0.7728758 0.6617647 0.4395425 0.5539216
      I22_P    I23_P    I24_P     I25_P     I26_P     I27_P     I28_P
1 0.1944444 0.307563 0.701426 0.6426025 0.3137255 0.1672794 0.6599265
       I29_P     I30_P
1 0.09582543 0.5825427
```

This shows that Item 10 has a lower discrimination than Item 5, as we expect.

Q5

What are the maximum and minimum values of D? And when are those values achieved?
Answer:

External criterion: Get upper and lower groups based on another test. This is usually preferred when feasible.

Using correlations

> Tip
>
> *Point-biserial correlation*: Pearson correlation between a dichotomous and a continuous variable.

For example, we can compute the item-total correlation for Item 1:

```
cor(mc_scored$I01, mc_scored$person_mean, use = "complete")
```

```
[1] 0.3354805
```

Because the item is included in the calculation of the total/mean score, another common approach is to compute the total/mean of all other items, *excluding* the one of interest:

```
# Exclude Item 1
person_mean_no1 <- rowMeans(mc_scored[2:30], na.rm = TRUE)
# Correlation
cor(mc_scored$I01, person_mean_no1, use = "complete")
```

```
[1] 0.2725089
```

We can compute discrimination for all items:

```
compute_discrimination <- function(i, data = mc_scored[1:30]) {
    # Exclude Item i
    person_mean_noi <- rowMeans(data[, -i], na.rm = TRUE)
    # Correlation
    cor(data[[i]], person_mean_noi, use = "complete")
}
# Apply to all items
```

```r
item_discrimination <- sapply(1:30, FUN = compute_discrimination)
# Add to table
datasummary(
    # Same table as before
    All(mc_scored[1:30]) ~ N + Mean + Median + SD, data = mc_scored[1:30],
    # Add an additional column with data.frame input
    add_columns = data.frame(Discrimination = item_discrimination),
    # center-aligned columns
    align = "lccccc")
```

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
```

```
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

```
Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

```
Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

Evaluating the Distractors

The discrimination for the distractors should be negative. Let's consider Item 17:

```r
table(mc_scored$group, mc_raw[ , "I17"])
```

```
        9  A  B  C  D
  lower  2  3 11 10 11
  upper  1  1  4 29  0
```

Here "C" is the correct option, and "A", "B", and "D" are distractors. We can see that those options are endorsed more often by the lower group. Also, "A" is relatively ineffective, as it was barely chosen by either groups.

	N	Mean	Median	SD	Discrimination
I01	142	0.92	1.00	0.28	0.27
I02	142	0.89	1.00	0.32	0.28
I03	142	0.78	1.00	0.41	0.40
I04	141	0.91	1.00	0.28	0.26
I05	141	0.77	1.00	0.42	0.47
I06	141	0.88	1.00	0.33	0.25
I07	140	0.67	1.00	0.47	0.44
I08	140	0.63	1.00	0.48	0.40
I09	140	0.42	0.00	0.50	0.35
I10	140	0.22	0.00	0.42	−0.03
I11	140	0.41	0.00	0.49	0.36
I12	139	0.45	0.00	0.50	0.40
I13	139	0.78	1.00	0.41	0.30
I14	138	0.41	0.00	0.49	0.13
I15	138	0.27	0.00	0.44	0.42
I16	138	0.60	1.00	0.49	0.56
I17	138	0.65	1.00	0.48	0.39
I18	136	0.61	1.00	0.49	0.52
I19	136	0.59	1.00	0.49	0.46
I20	135	0.67	1.00	0.47	0.33
I21	135	0.70	1.00	0.46	0.41
I22	135	0.30	0.00	0.46	0.11
I23	134	0.63	1.00	0.49	0.28
I24	131	0.51	1.00	0.50	0.51
I25	131	0.55	1.00	0.50	0.46
I26	131	0.36	0.00	0.48	0.13
I27	129	0.22	0.00	0.41	0.14
I28	129	0.38	0.00	0.49	0.47
I29	128	0.27	0.00	0.44	−0.02
I30	127	0.61	1.00	0.49	0.38

Noncognitive Items

We will use the `bfi` data set again.

```
data(bfi, package = "psych")
```

Note: Generally Likert-type items are not normal, but using a normal distribution is probably most defensible when there were at least five scale points and when the item distributions are relatively symmetric and unimodal (Rhemtulla et al., 2012, https://doi.org/10.1037/a0029315).

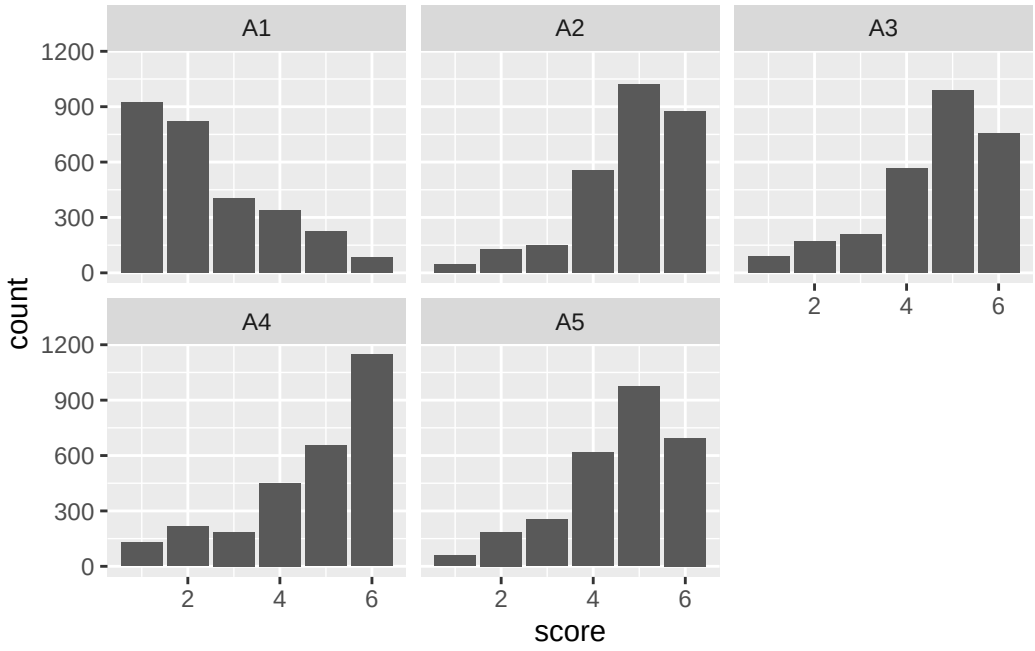
Frequency and Descriptives

While one can compute the usual descriptives like mean, median, and *SD*, for scale items that have only a few options, I generally recommend using actual bar plots.

You can see that the Agreeableness items are relatively skewed:

```
library(tidyr)
bfi |>
    select(A1:A5) |>
    pivot_longer(cols = A1:A5, names_to = "item", values_to = "score") |>
    ggplot(aes(x = score)) +
    geom_bar() +
    facet_wrap(~ item)
```

```
Warning: Removed 104 rows containing non-finite values (`stat_count()`).
```



Q5

Why do you think the Agreeableness items are skewed?
Answer:

When items show non-normal distributions, consider the reason for it and the purpose of the scale. For example, some skewed items are needed to assess extreme levels of a construct.

Recoding

An important step is to recode the negatively oriented variables.

```
# Recode A1
bfi$A1r <- 7 - bfi$A1
```

Interitem Correlations

Q7

Use the `modelsummary::datasummary_correlation()` to compute the interitem correlations among the Agreeableness items.

```
# Your code
```

Corrected Item-Total Correlations

To see if a person scoring high on one item also scores high on the other items. Computationally, this is the same as the item discrimination for cognitive items.

```r
# Create a data subset for A
bfi_a <- bfi |>
    select(A1r, A2:A5)
# Use the same function as before
item_total_cor <- sapply(1:5, FUN = compute_discrimination, data = bfi_a)
# Generate a table
datasummary(
    All(bfi_a) ~ N + Mean + SD + Min + Median + Max + Histogram * Arguments(bins = 6), data =
    add_columns = data.frame(`Item-Total` = item_total_cor))
```

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

	N	Mean	SD	Min	Median	Max	Histogram	Item.Total
A1r	2784	4.59	1.41	1.00	5.00	6.00		0.31
A2	2773	4.80	1.17	1.00	5.00	6.00		0.56
A3	2774	4.60	1.30	1.00	5.00	6.00		0.58
A4	2781	4.70	1.48	1.00	5.00	6.00		0.39
A5	2784	4.56	1.26	1.00	5.00	6.00		0.49

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")

Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")

Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

Factor Analysis

Sorting out the items based on the relative magnitude of their intercorrelations.

> **!** Important
>
> *Factors*: based on shared variance among items.
> *Factor loading*: the degree to which an item share variance with the common factor.

```
fa(bfi[1:25], nfactors = 5)
```

```
Loading required namespace: GPArotation

Factor Analysis using method =  minres
Call: fa(r = bfi[1:25], nfactors = 5)
Standardized loadings (pattern matrix) based upon correlation matrix
     MR2   MR1   MR3   MR5   MR4   h2   u2 com
A1  0.21  0.17  0.07 -0.41 -0.06 0.19 0.81 2.0
A2 -0.02  0.00  0.08  0.64  0.03 0.45 0.55 1.0
A3 -0.03  0.12  0.02  0.66  0.03 0.52 0.48 1.1
A4 -0.06  0.06  0.19  0.43 -0.15 0.28 0.72 1.7
A5 -0.11  0.23  0.01  0.53  0.04 0.46 0.54 1.5
C1  0.07 -0.03  0.55 -0.02  0.15 0.33 0.67 1.2
C2  0.15 -0.09  0.67  0.08  0.04 0.45 0.55 1.2
C3  0.03 -0.06  0.57  0.09 -0.07 0.32 0.68 1.1
C4  0.17  0.00 -0.61  0.04 -0.05 0.45 0.55 1.2
C5  0.19 -0.14 -0.55  0.02  0.09 0.43 0.57 1.4
E1 -0.06 -0.56  0.11 -0.08 -0.10 0.35 0.65 1.2
E2  0.10 -0.68 -0.02 -0.05 -0.06 0.54 0.46 1.1
E3  0.08  0.42  0.00  0.25  0.28 0.44 0.56 2.6
E4  0.01  0.59  0.02  0.29 -0.08 0.53 0.47 1.5
E5  0.15  0.42  0.27  0.05  0.21 0.40 0.60 2.6
N1  0.81  0.10  0.00 -0.11 -0.05 0.65 0.35 1.1
N2  0.78  0.04  0.01 -0.09  0.01 0.60 0.40 1.0
N3  0.71 -0.10 -0.04  0.08  0.02 0.55 0.45 1.1
N4  0.47 -0.39 -0.14  0.09  0.08 0.49 0.51 2.3
N5  0.49 -0.20  0.00  0.21 -0.15 0.35 0.65 2.0
O1  0.02  0.10  0.07  0.02  0.51 0.31 0.69 1.1
O2  0.19  0.06 -0.08  0.16 -0.46 0.26 0.74 1.7
O3  0.03  0.15  0.02  0.08  0.61 0.46 0.54 1.2
O4  0.13 -0.32 -0.02  0.17  0.37 0.25 0.75 2.7
O5  0.13  0.10 -0.03  0.04 -0.54 0.30 0.70 1.2

                       MR2  MR1  MR3  MR5  MR4
SS loadings           2.57 2.20 2.03 1.99 1.59
```

```
Proportion Var        0.10 0.09 0.08 0.08 0.06
Cumulative Var        0.10 0.19 0.27 0.35 0.41
Proportion Explained  0.25 0.21 0.20 0.19 0.15
Cumulative Proportion 0.25 0.46 0.66 0.85 1.00

 With factor correlations of
      MR2   MR1   MR3   MR5   MR4
MR2  1.00 -0.21 -0.19 -0.04 -0.01
MR1 -0.21  1.00  0.23  0.33  0.17
MR3 -0.19  0.23  1.00  0.20  0.19
MR5 -0.04  0.33  0.20  1.00  0.19
MR4 -0.01  0.17  0.19  0.19  1.00

Mean item complexity =  1.5
Test of the hypothesis that 5 factors are sufficient.

df null model =  300  with the objective function =  7.23 with Chi Square =  20163.79
df of  the model are 185  and the objective function was  0.65

The root mean square of the residuals (RMSR) is  0.03
The df corrected root mean square of the residuals is  0.04

The harmonic n.obs is  2762 with the empirical chi square  1392.16  with prob <  5.6e-184
The total n.obs was  2800  with Likelihood Chi Square =  1808.94  with prob <  4.3e-264

Tucker Lewis Index of factoring reliability =  0.867
RMSEA index =  0.056  and the 90 % confidence intervals are  0.054 0.058
BIC =  340.53
Fit based upon off diagonal values = 0.98
Measures of factor score adequacy
                                                 MR2  MR1  MR3  MR5  MR4
Correlation of (regression) scores with factors  0.92 0.89 0.88 0.88 0.84
Multiple R square of scores with factors         0.85 0.79 0.77 0.77 0.71
Minimum correlation of possible factor scores    0.70 0.59 0.54 0.54 0.42
```

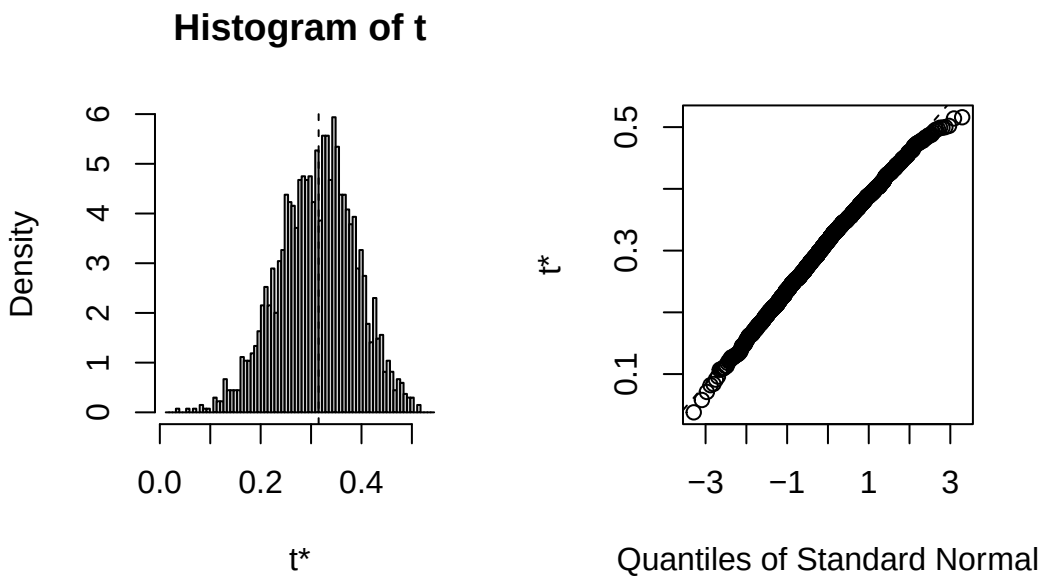
There are two caveats if choosing items solely based on their predictive and diagnostic utility: (a) it may result in nonsensical scales, and (b) it may result in low internal consistency reliability.

Considering Uncertainty in Statistics

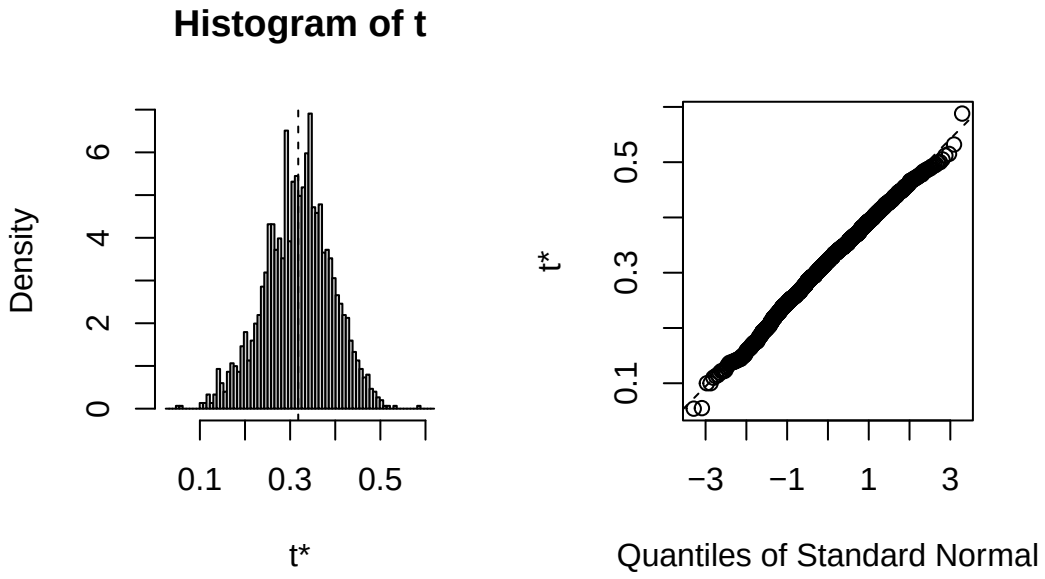
- Statistical information only provides one source of information when selecting or deleting items.
- An item showing suboptimal performance may nevertheless be important conceptually.
- An item showing a poor performance in one's sample may perform adequately in previous and subsequent samples.

To the last point, it is important to consider the uncertainty in the statistics we use for making decisions, especially when we have a small and/or non-representative sample. While software packages seldom report standard errors or confidence intervals for statistics like item-total correlation, it is relatively straightforward to use the *bootstrap* to quantify uncertainty for a lot of the statistics. Below is an example, using a small subset of the `bfi` data.

```
# Obtain a subset
bfi_sub <- bfi[1:200, ]
# Bootstrap CI
# 1. Define function for item-total correlations. The second argument needs to an index for r
ff <- function(d, inds) {
    performance::item_reliability(d[inds, ])$item_discrimination
}
# 2. Bootstrap
boo <- boot(bfi_sub[, c("A1r", "A2", "A3", "A4", "A5")],
     statistic = ff,
     R = 1999)
# 3. Summarize bootstrap distributions
plot(boo, index = 1)  # for A1r
```




```
plot(boo, index = 4)  # for A4
```



```r
# Number of bootstrap samples where item-total correlations < .20
# (A1r, A2:A5)
colMeans(boo$t < .20)
```

```
[1] 0.07253627 0.00000000 0.00000000 0.06653327 0.00000000
```

```r
# 4. Bootstrap confidence intervals for the first statistic
# (i.e., item-total correlation for A1r)
boot.ci(boo, index = 1)
```

```
Warning in boot.ci(boo, index = 1): bootstrap variances needed for studentized
intervals

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1999 bootstrap replicates

CALL :
boot.ci(boot.out = boo, index = 1)

Intervals :
Level      Normal              Basic
95%   ( 0.1683,  0.4669 )   ( 0.1750,  0.4710 )

Level     Percentile            BCa
95%   ( 0.1590,  0.4550 )   ( 0.1567,  0.4540 )
Calculations and Intervals on Original Scale
```

Q8

From the bootstrap samples above, what is the proportion of replications where Item A1r has the lowest item-total correlation? Show your code, and **discuss what this implies.**

```r
# Your code
```

Answer: