Score TDS Qualtrics Data

John Flournoy

2018-02-22

Contents

This presents both a walk-through for how you could go about adding a new questionnaire to be scored, and also should provide up-to-date descriptives and scored scales every time you recompile this Rmd document.

This is a living document, meaning that you can and should edit it as you need to (and git commit often)! If you need a hand understanding something, or get stuck, email flournoy@uoregon.edu.

It would be nice if this were something completely automated. Unfortunately, the nature of data that is still under collection, and the fact that there are a lot of moving pieces means that something totally automated is likely to break pretty easily (at least, something automated that has been coded by me). For that reason this document will take a highly modular approach so that hopefully if something goes wrong, you can see more exactly where that happens. I'll try to explain every step pretty verbosely too. You should probably be reading this document in R Studio, running each chunk as you go. Make sure you're up to date with upgrades for R, R Studio, and packages.

Setting options

If you're reading the compiled HTML, you won't see the options below, because we've written `warning=F,echo=F,message=F,error=F` in the head of the chunk. You can change this by replacing the "F"s with "T"s. This chunk sets variables that will be reused to define where output goes, how it's named, what codes are used for gender in the PDS and so on.

Install the `scorequaltrics` package

We need to have the scorequaltrics package installed. I wrote this so that I can maintain helpful functions related to scoring.

```
#this chunk won't evaluate. If you need to
#install the package, run this by hand.
devtools::install_github('jflournoy/qualtrics')
```

Accessing qualtrics data

You also need to have a token in a YAML formatted file for accessing Qualtrics via the API. It's formatted like:

```
user: username
token: apitoken
```

Once we've loaded this, we can get a list of questionnaires.

```
library(scorequaltrics)
library(ggplot2)
library(dplyr)
library(tidyr)

credentials <- scorequaltrics::creds_from_file(cred_file_location)

rawSurveys <- scorequaltrics::get_surveys(credentials)
rawSurveysTDS <- filter(rawSurveys, grepl('.*(TDS1|TDS2|TDS3).*',SurveyName))

knitr::kable(arrange(select(rawSurveysTDS, SurveyName), SurveyName))
```

SurveyName
Qsupport TDS1, Session 3 - Child - Copy
QSupport TDS2 Session 3 - Parent - Copy - Copy
TDS1 AND TDS3 PSQI

SurveyName
TDS1 CBCL - Post
TDS1 CBCL - Pre
TDS1 PDS (Session2) - Post
TDS1 Saliva Questions - Post
TDS1 Saliva Questions - Pre
TDS1 Session 1 & 2 Makeups
TDS1 Session 1 - Post
TDS1 Session 1 - Pre
TDS1 Session 2 A - Post
TDS1 Session 2 A - Pre
TDS1 Session 2 B - Post
TDS1 Session 2 B - Pre
TDS1 Session 2 B - SHORTENED_A
TDS1 Session 2 B - SHORTENED_B
TDS1, Session 3 - Child
TDS1 Session 3 Child *A*
TDS1 Session 3 Child *B*
TDS1, Session 3 - Parent
TDS1, Session 3 - Parent part 2
TDS1 Spinning Wheel Game - New
TDS1 Spinning Wheel Game - Post
TDS2 CBCL
TDS2 - PDS (Session 2)
TDS2 Saliva Questions
TDS2 Session 1
TDS2 Session 2 A
TDS2 Session 2 B
TDS2 Session 2 B - SHORTENED_A
TDS2 Session 2 B - SHORTENED_B
TDS2 Session 3 - Child
TDS2 Session 3 - KSRQ & SAQ
TDS2 Session 3 - Parent
TDS2 Session 3 - Parent - Copy
TDS2 Spinning Wheel Game

We have a lot of different questionnaires from different samples and different sessions. For simplicity, and to aid in diagnosing any problems, we can proceed through each sample and each wave of data collection. *Note*, however, that if we ensured that naming conventions were consistent across all questionnaires and rubrics, and if we had accurate session dates attached, we could score everything in one fell swoop.

Cleaning and scoring data

As an example, I'll walk through step-by-step how to clean and score data using the TDS2, Wave 1 sample.

This is the first sample collected, so we can begin here. I'll demonstrate in this how to do a single massive data scoring. Following that, I'll demonstrate how you can get more information about scales that have been constructed in a psychometric tradition, and that therefore are easy to evaluate using standard reliability metrics.

First, we download the data for the surveys we want.

```
tds2_wave1_surveys <- rawSurveysTDS %>%
    filter(grepl('TDS2 (Session [12]|CBCL)', SurveyName))

print(tds2_wave1_surveys$SurveyName)
```

```
## [1] "TDS2 CBCL"                       "TDS2 Session 2 B"
## [3] "TDS2 Session 2 B - SHORTENED_B" "TDS2 Session 2 A"
## [5] "TDS2 Session 1"                  "TDS2 Session 2 B - SHORTENED_A"
```

```
tds2_wave1_long <- scorequaltrics::get_survey_data(tds2_wave1_surveys,
                                                   credentials,
                                                   pid_col = pid_column_name)
dim(tds2_wave1_long)
```

```
## [1] 85919     4
```

```
names(tds2_wave1_long)
```

```
## [1] "SID"         "item"        "value"       "survey_name"
```

The resulting data frame should have a lot of rows (the first part of the output of `dim`) and 4 columns.

```
## It looks like all is in order here. Note that the PID column is named "SID".
```

Note that some participants re-did their PDS questionnaires at wave 1, session 2, if it had been a long time since wave 1, session 1. We'll have to deal with these participants' PDS scores separately, and so we'll download that session 2 PDS data separately, and then replace the PDS in the main data file for PDS session 1 for anyone who has it from session 2.

```
tds2_wave1_surveys_pds2 <- rawSurveysTDS %>%
    filter(grepl('TDS2 - PDS \\(Session 2\\)', SurveyName))
tds2_wave1_long_pds2 <- scorequaltrics::get_survey_data(tds2_wave1_surveys_pds2,
                                                        credentials,
                                                        pid_col = pid_column_name) %>%
    filter(grepl('PDS', item))

tds2_wave1_sid_with_pds2 <- tds2_wave1_long_pds2 %>%
    group_by(SID) %>%
    summarize(n_items = sum(!is.na(as.numeric(value)))) %>%
    filter(n_items > 0) %>%
    select(SID) %>%
    unlist
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
tds2_wave1_long <- tds2_wave1_long %>%
    filter(!(grepl('PDS', item) &
                 SID %in% tds2_wave1_sid_with_pds2)) %>% # remove PDS items from pds
                                                         # session 2 participants
    bind_rows(filter(tds2_wave1_long_pds2,
                     SID %in% tds2_wave1_sid_with_pds2)) # Add rows from PDS session 2
                                                         # participants, given that they're
                                                         # in the list of SIDs with >0 items
```

Before doing any scoring, we should take care of all the complex response recoding that may be specified. So We'll load all the response recoding rubrics and apply those. It's important that you pass the full path of the file to the next function, so if you use `dir` to collect filenames as I do below, make sure you set `full.names = TRUE`.

```
dir(file.path(tds2_wave1_rubric_dir), pattern = '.*response_recoding.*.csv')
```

```
## [1] "PAL2_response_recoding.csv"
## [2] "SES_response_recoding.csv"
## [3] "YRBS_response_recoding_TDS2_session_2.csv"
```

```
#You should see a result below -- if not, the path is likely wrong.

tds2_wave1_recoding_rubrics <- data.frame(file = dir(file.path(tds2_wave1_rubric_dir),
                                                     pattern = '.*response_recoding.*.csv',
                                                     full.names = TRUE))

tds2_wave1_recoding_data_long <- scorequaltrics::get_rubrics(tds2_wave1_recoding_rubrics,
                                                             type = 'recoding')

tds2_wave1_long_recoded <- scorequaltrics::recode_responses(tds2_wave1_long,
                                                            tds2_wave1_recoding_data_long)
```

Now let's load in the scoring rubrics.

```
tds2_wave1_scoring_rubrics <- data.frame(file = dir(file.path(tds2_wave1_rubric_dir),
                                                    pattern = '.*scoring_rubric.*.csv',
                                                    full.names = TRUE))

tds2_wave1_scoring_data_long <- scorequaltrics::get_rubrics(tds2_wave1_scoring_rubrics,
                                                            type = 'scoring')

head(tds2_wave1_scoring_data_long[, -1])
```

```
## # A tibble: 6 x 9
##   data_file_name scale_name column_name transform reverse min   max
##   <chr>          <chr>      <chr>       <chr>     <chr>   <chr> <chr>
## 1 TDS2_Session_1 ACE        ACE_1       0         0       0     0
## 2 TDS2_Session_1 ACE        ACE_2       0         0       0     0
## 3 TDS2_Session_1 ACE        ACE_3       0         0       0     0
## 4 TDS2_Session_1 ACE        ACE_4       0         0       0     0
## 5 TDS2_Session_1 ACE        ACE_5       0         0       0     0
## 6 TDS2_Session_1 ACE        ACE_6       0         0       0     0
## # ... with 2 more variables: scored_scale <chr>, include <chr>
```

Cleaning

We can make sure we clean out duplicate responses which will help later with ensuring that scale scores are calculated from the correct subset of items. This is a point at which, if there is something funky going on, you'll want to investigate it and make a decision. For example, if a participant has two conflicting answers to the same question for the same wave, it's likely that a small investigation should commence.

Before we do that, we can ensure that we're only keeping the data in the scoring rubrics in the first place, using `scorequaltrics::get_items_in_rubric(tds2_wave1_scoring_data_long)`.

We want to make sure that certain items coded as missing, such as when someone picks "decline to answer". We do this with the lines

```
...
mutate(value = ifelse(grepl('SES', item) & as.numeric(value) < 0,
                      NA,
```

```
                        value)) %>%
...
```

We can also ensure that we're only scoring data for participants with the correct ID numbers. The line in the middle of the first call, `filter(grepl('[1234]\\d\\d', SID))`, ensures we only keep people with ID's starting with "1".

```
tds2_wave1_long_recoded_nodupes <- tds2_wave1_long_recoded %>%
    scorequaltrics::get_items_in_rubric(tds2_wave1_scoring_data_long) %>%
    mutate(value = ifelse(grepl('SES', item) & as.numeric(value) < 0,
                          NA,
                          value)) %>%
    filter(grepl('[14]\\d\\d', SID)) %>%
    scorequaltrics::clean_dupes(pid_col = 'SID')
```

If you get "NAs introduced by coercion" it probably means that one of the rubrics references a column that has text input that is not transformable into a number. For example, if the questionnaire asks for ethnicity and someone writes in "White" it is not possible to turn that into a score to be used in a scale calculation (but there's a rubric that thinks it can). We can check that by using the function `scorequaltrics::get_uncoercibles()`.

```
tds2_wave1_uncoer <- tds2_wave1_long_recoded %>%
    get_items_in_rubric(tds2_wave1_scoring_data_long) %>%
    filter(grepl('[1234]\\d\\d', SID)) %>%
    scorequaltrics::get_uncoercibles() %>%
    distinct(item, value)

head(tds2_wave1_uncoer, 10)
```

```
## Empty data.table (0 rows) of 2 cols: item,value
```

```
unique(tds2_wave1_uncoer$item)
```

```
## character(0)
```

Now we can look at what rubrics have those items, if any.

```
tds2_wave1_scoring_data_long %>%
    filter(column_name %in% unique(tds2_wave1_uncoer$item),
           include %in% c(1, "1", "sum", "prod")) %>%
    ungroup() %>%
    select(scale_name, scored_scale, column_name, include)
```

```
## # A tibble: 0 x 4
## # ... with 4 variables: scale_name <chr>, scored_scale <chr>,
## #   column_name <chr>, include <chr>
```

If the above two chunks didn't result in output, we're good!

Now we can see which items had conflicting responses on duplicate questionnaires.

```
#Check that dropped values weren't ambiguous
tds2_wave1_long_recoded_nodupes %>%
  filter(dropped) %>%
  group_by(SID, item) %>%
  summarize(noinfo = all(length(unlist(old.value)) < 1)) %>%
  ungroup() %>%
  summarize(n_with_info = sum(!noinfo))
```

```
## # A tibble: 1 x 1
##   n_with_info
##         <int>
## 1          55
```

```
tds2_wave1_long_recoded_nodupes %>%
    filter(dropped) %>%
    group_by(SID, item) %>%
    filter(!all(length(unlist(old.value)) < 1)) %>%
    mutate(old.value = paste(old.value, collaps = ' ')) %>%
    knitr::kable()
```

SID	item	value	survey_name	old.value	dropped
109	YRBS_10	NA	TDS2 Session 2 B	c(1, 0)	TRUE
124	PDS_F3	NA	TDS2 Session 1	c(4, 3)	TRUE
125	YRBS_10	NA	TDS2 Session 2 B	c(1, 0)	TRUE
159	YRBS_10	NA	TDS2 Session 2 B	c(1, 0)	TRUE
189	YRBS_10	NA	TDS2 Session 2 B	c(1, 0)	TRUE
190	YRBS_10	NA	TDS2 Session 2 B	c(1, 0)	TRUE
196	BIS_1	NA	TDS2 Session 2 A	c(3, 2)	TRUE
196	BIS_10	NA	TDS2 Session 2 A	c(4, 2)	TRUE
196	BIS_14	NA	TDS2 Session 2 A	c(2, 3)	TRUE
196	BIS_2	NA	TDS2 Session 2 A	c(3, 2)	TRUE
196	BIS_6	NA	TDS2 Session 2 A	c(2, 3)	TRUE
196	BIS_7	NA	TDS2 Session 2 A	c(4, 3)	TRUE
196	SSS_3	NA	TDS2 Session 2 A	c(1, 0)	TRUE
196	SSS_4	NA	TDS2 Session 2 A	c(1, 0)	TRUE
401	BFNE_1	NA	TDS2 Session 2 B	c(1, 2, 1)	TRUE
401	BFNE_10	NA	TDS2 Session 2 B	c(2, 3, 1)	TRUE
401	BFNE_11	NA	TDS2 Session 2 B	c(1, 2, 1)	TRUE
401	BFNE_12	NA	TDS2 Session 2 B	c(1, 2, 1)	TRUE
401	BFNE_2	NA	TDS2 Session 2 B	c(3, 4, 4)	TRUE
401	BFNE_3	NA	TDS2 Session 2 B	c(1, 2, 1)	TRUE
401	BFNE_4	NA	TDS2 Session 2 B	c(4, 5, 4)	TRUE
401	BFNE_5	NA	TDS2 Session 2 B	c(1, 2, 1)	TRUE
401	BFNE_6	NA	TDS2 Session 2 B	c(0, 1, 1)	TRUE
401	BFNE_7	NA	TDS2 Session 2 B	c(2, 3, 1)	TRUE
401	BFNE_8	NA	TDS2 Session 2 B	c(1, 2, 1)	TRUE
401	BFNE_9	NA	TDS2 Session 2 B	c(0, 1, 1)	TRUE
401	NTS_1	NA	TDS2 Session 2 B	c(2, 2, 1)	TRUE
401	NTS_10	NA	TDS2 Session 2 B	c(2, 2, 1)	TRUE
401	NTS_2	NA	TDS2 Session 2 B	c(2, 2, 1)	TRUE
401	NTS_4	NA	TDS2 Session 2 B	c(4, 4, 1)	TRUE
401	NTS_5	NA	TDS2 Session 2 B	c(5, 5, 1)	TRUE
401	NTS_6	NA	TDS2 Session 2 B	c(3, 3, 1)	TRUE
401	NTS_7	NA	TDS2 Session 2 B	c(3, 3, 1)	TRUE
401	NTS_8	NA	TDS2 Session 2 B	c(2, 2, 1)	TRUE
401	NTS_9	NA	TDS2 Session 2 B	c(2, 2, 1)	TRUE
401	PAL2_11A	NA	TDS2 Session 2 B	c(2, 1)	TRUE
401	PAL2_11B	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_11C	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_11E	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_14A	NA	TDS2 Session 2 B	c(2, 1)	TRUE
401	PAL2_14B	NA	TDS2 Session 2 B	c(2, 1)	TRUE

SID	item	value	survey_name	old.value	dropped
401	PAL2_14C	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_14D	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_14E	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_14F	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_14G	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_14H	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_14I	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_14J	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_14K	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	PAL2_8	NA	TDS2 Session 2 B	c(4, 3)	TRUE
401	PAL2_9C	NA	TDS2 Session 2 B	c(1, 0)	TRUE
401	YRBS_33	NA	TDS2 Session 2 B	c(0, 1)	TRUE
401	YRBS_36	NA	TDS2 Session 2 B	c(0, 1)	TRUE
401	YRBS_37	NA	TDS2 Session 2 B	c(0, 1)	TRUE

For now, if there are ambiguous entries, I'm going to ignore them. The value to be used for scoring is set to NA – we have to treat that data as missing since the responses are in conflict.

Scoring almost all at once

There are a few different options for scoring questionnaires. First, we can provide a rubric and data to `scorequaltrics::score_questionnaire(dataDF, rubricsDF, psych = TRUE)`, which will use the `psych` package to do the scoring. This has the advantage that you get back a lot of information about the measurement quality of the scale, but it only works for scales that follow certain psychometric principles (e.g., each item is rated on a continuous scale, and is an indicator of a latent construct). It won't work well for other kinds of data (like scales where you want to know the number of risky behaviors, for example).

The second option is to use `scorequaltrics::score_step_one_and_two(dataDF, rubricsDF)` which was created to take care of several special cases for the TDS project questionnaires. The RPI, and RSQ both require special handling because of their idiosyncratic questionnaire design.

```
tds2_wave1_scored <- scorequaltrics::score_step_one_and_two(tds2_wave1_long_recoded_nodupes,
                                                            tds2_wave1_scoring_data_long)

set.seed(9567)
tds2_wave1_scored %>% ungroup %>%
    dplyr::sample_n(size = 10) %>%
    select(-SID)
```

```
## # A tibble: 10 x 6
##    scale_name        scored_scale       score     n_items n_missing method
##    <chr>             <chr>              <chr>       <int>     <int> <chr>
##  1 CBCL              somatic_complaints 0.1            10         1 1
##  2 CARE-R Expected ~ risky_car          1               2         0 1
##  3 RSQ_part2         rsq_mean_anxious_~ 10.3           10         0 1
##  4 PAL-2             pal2_antisocial    1.181818~      11         0 1
##  5 CARE-R Willingne~ risky_sex_regular~ 1               5         0 1
##  6 SPSRQ-S           sensitivity_rewar~ 0.3            10         0 1
##  7 UPPS-P            pos_urgency        1.285714~      14         0 1
##  8 PEQ-R             prosocial_to_me    2.6             5         0 1
##  9 RPI_part2         rpi_mean           4              10         0 1
## 10 CARE-R Social     care_soc_not_risk~ 1               3         0 1
```

One thing missing still is the Pubertal Development Scale scored via the Shirtcliff method, so I'll calculate that now with the special function `scorequaltrics::score_pdss()`

```
tds2_wave1_scored_pdss <- scorequaltrics::score_pdss(tds2_wave1_long_recoded_nodupes,
                          gender_mix = pdss_gender_mix,
                          gendercode = pdss_gender_code)
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

That hopefully went off without a hitch. Now we can examine the scales we've scored, and along the way, see how we can get more information about these scales when we are able to use the `psych` package scoring function.

Scoring a single scale using `score_questionnaire`

It is possible, and maybe most convenient in many cases, to score a single questionnaire at a time, and write it to a CSV file. I'll demonstrate with the Barratt's Impulsiveness Scale Version 15. This scale has just a single rubric – no recoding rubric, and no part 2 scoring rubric.

All we have to do is load the rubric using `get_rubrics` and then score the questionnaire using `score_questionnaire`.

```
tds2_wave1_bis_scoring <- scorequaltrics::get_rubrics(
    rubric_filenames = data.frame(
        file = file.path(tds2_wave1_rubric_dir, 'BIS_scoring_rubric.csv')))

#using the data we've already cleaned and recoded -- note that BIS does not require recoding
tds2_wave1_bis_scored <- scorequaltrics::score_questionnaire(tds2_wave1_long_recoded_nodupes,
                                                             tds2_wave1_bis_scoring)

#display a random set of 10 rows to check out the data.
set.seed(4210)
sample_n(select(ungroup(tds2_wave1_bis_scored), -SID), size = 10)
```

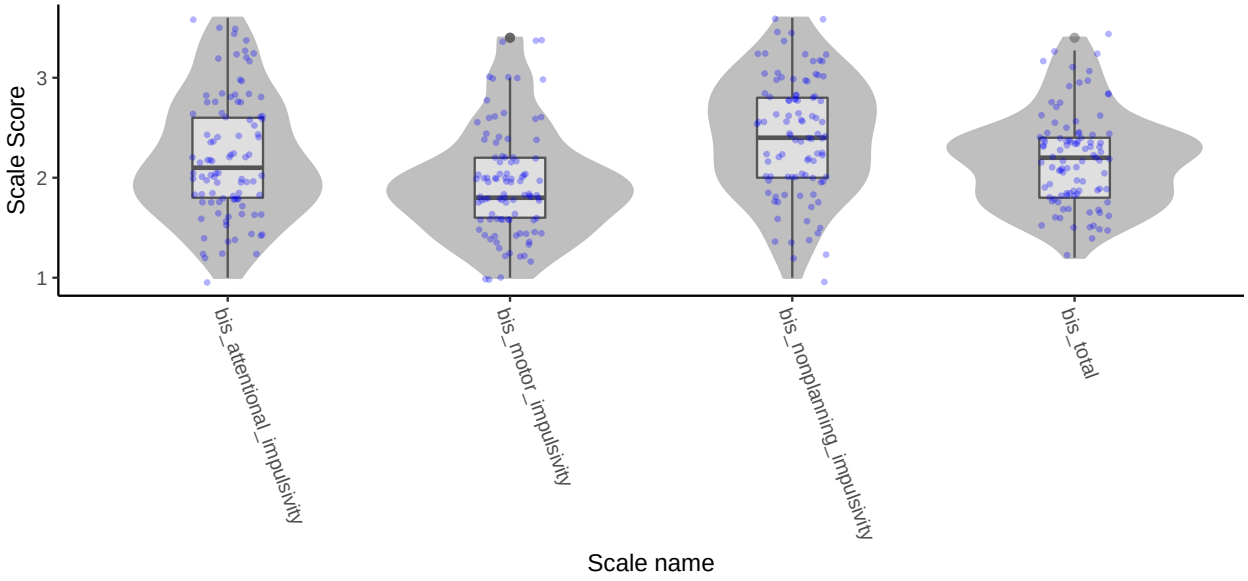
```
## # A tibble: 10 x 6
##    scale_name scored_scale                score   n_items n_missing method
##    <chr>      <chr>                       <chr>     <int>     <int> <chr>
##  1 BIS-15     bis_nonplanning_impulsivity 1.8           5         0 1
##  2 BIS-15     bis_nonplanning_impulsivity 3             5         0 1
##  3 BIS-15     bis_motor_impulsivity       2.4           5         0 1
##  4 BIS-15     bis_total                   2.3333~      15         0 1
##  5 BIS-15     bis_nonplanning_impulsivity 2.4           5         0 1
##  6 BIS-15     bis_total                   2.2666~      15         0 1
##  7 BIS-15     bis_motor_impulsivity       1.75          4         1 1
##  8 BIS-15     bis_motor_impulsivity       1.8           5         0 1
##  9 BIS-15     bis_total                   3.2          15         0 1
```

```
## 10 BIS-15     bis_total                    2.5333~      15         0 1
```

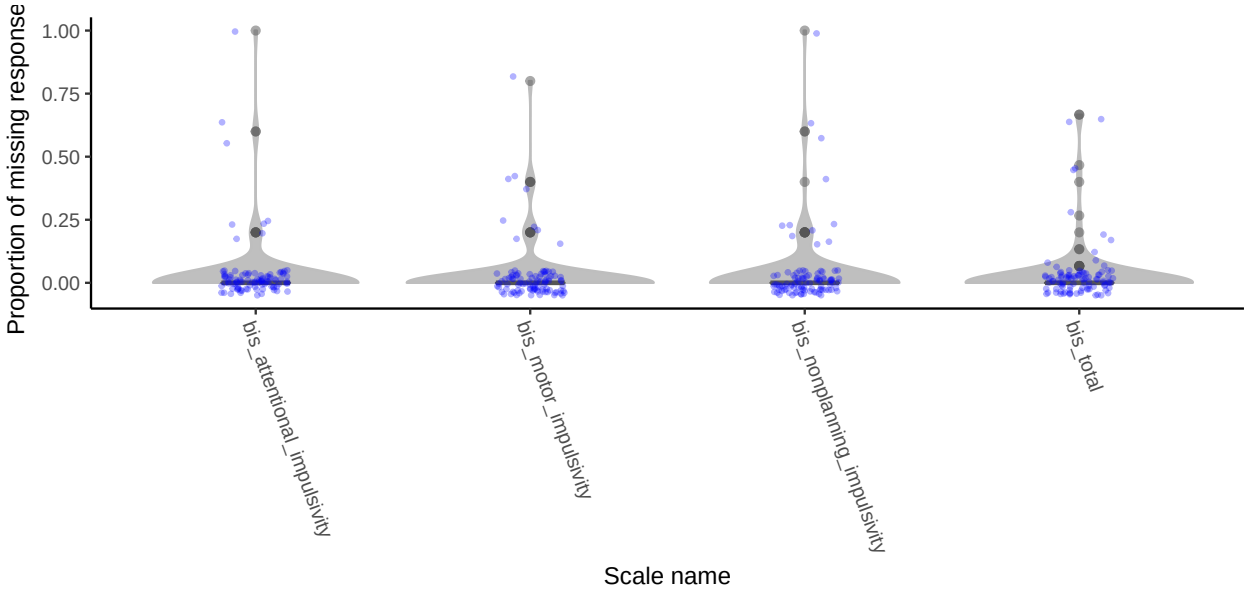
I can plot these data now using `plot_scored_scale`, which has options to select the scale name (not subscale) using regular expressions (defaults to all), or to plot number or proportion of items missing.

```
scorequaltrics::plot_scored_scale(tds2_wave1_bis_scored)
```

```
## Warning: Removed 2 rows containing non-finite values (stat_ydensity).
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
## Warning: Removed 2 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_bis_scored, type = 'p_missing')
```



I can also widen this data frame for ease of use in regressions and exporting, using `widen_qualtrics_long`. Note that even with a single scored scale, you need to specify the scale name to be matched exactly. In this case, we can see that `unique(tds2_wave1_bis_scored$scale_name)` gives us: BIS-15. Note that the return value of the `widen_qualtrics_long` function is a list that contains a data frame of scores at `...$scores`

and of missing responses at `...$data_quality`.

```
tds2_wave1_bis_scored_wide <- scorequaltrics::widen_qualtrics_long(tds2_wave1_bis_scored,
                                                                   scale_names = 'BIS-15')

set.seed(322415)
sample_n(select(tds2_wave1_bis_scored_wide$scores, -SID), size = 3)
```

```
## # A tibble: 3 x 4
##   bis_attentional_impulsivity bis_motor_impuls~ bis_nonplanning~ bis_total
##   <chr>                       <chr>             <chr>            <chr>
## 1 1.4                         1.6               1.8              1.6
## 2 2.4                         2.2               2.4              2.333333~
## 3 2.8                         1.8               2.6              2.4
```

```
sample_n(select(tds2_wave1_bis_scored_wide$data_quality, -SID), size = 3)
```

```
## # A tibble: 3 x 8
##   bis_attentional_i~ bis_attentional_i~ bis_motor_impul~ bis_motor_impuls~
##                <int>              <int>            <int>             <int>
## 1                  5                  0                5                 0
## 2                  5                  0                5                 0
## 3                  4                  1                5                 0
## # ... with 4 more variables: bis_nonplanning_impulsivity_n_items <int>,
## #   bis_nonplanning_impulsivity_n_missing <int>, bis_total_n_items <int>,
## #   bis_total_n_missing <int>
```

We can easily write these tables to a file now:

```
write.csv(tds2_wave1_bis_scored_wide$scores, '/location/to/save/csv/file')
```

Scoring a single scale using `score_questionnaire` with `psych=TRUE`

There is a subset of scales that we can examine using tools from the psychometrics literature. That is, scales that consist of many items attempting to measure a theoretical latent construct can be examined using reliability measures to see how much consistency there is, from person to person, among these items that are ostensibly measuring the same thing. This doesn't make sense to look at in a scale like the Adverse Childhood Events or Youth Risk Behavior Survey because these are indices that are supposed to measure the number of certain events or behaviors that together are thought to convey risk. In other words, there is not theoretical basis for assuming that all adverse events in childhood are caused by the same latent construct. So even though it may be the case that certain adverse events tend to correlate, it is not necessary that they *do* correlate.

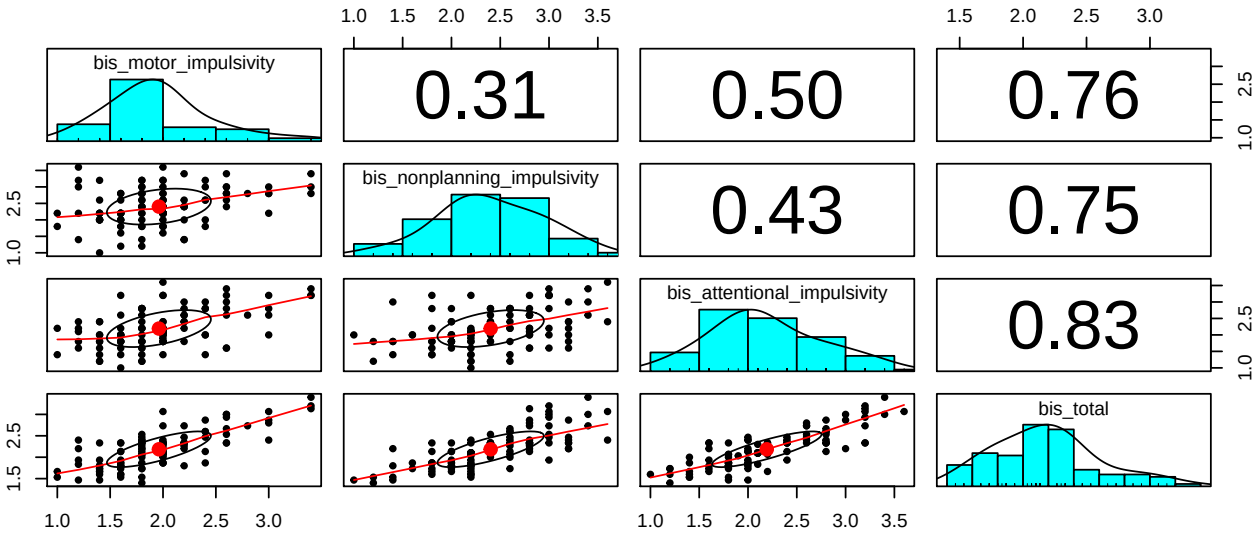
To investigate a scale using these tools, all we have to do is set it to be scored by the `psych` package, rather than the `scorequaltrics` back-end. To do so, we merely set the option `psych=TRUE`. We can then apply the `print` function to get information about reliability, and the `pairs.panels` function to the object's `...$scores`.

```
#using the rubric we've alread loaded:
tds2_wave1_bis_psych <- scorequaltrics::score_questionnaire(
    tds2_wave1_long_recoded_nodupes,
    tds2_wave1_bis_scoring,
    psych = T)

print(tds2_wave1_bis_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
```

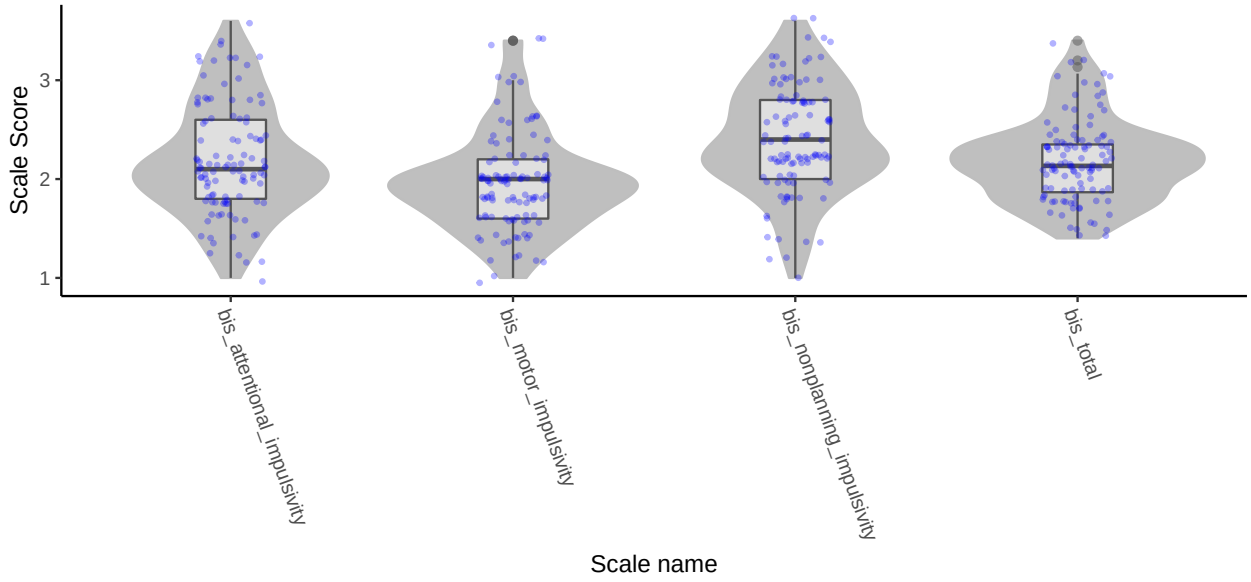
```
##
## (Unstandardized) Alpha:
##       bis_motor_impulsivity bis_nonplanning_impulsivity
## alpha                  0.76                        0.71
##       bis_attentional_impulsivity bis_total
## alpha                        0.73      0.83
##
## Standard errors of unstandardized Alpha:
##       bis_motor_impulsivity bis_nonplanning_impulsivity
## ASE                   0.065                       0.072
##       bis_attentional_impulsivity bis_total
## ASE                         0.068     0.033
##
## Average item correlation:
##           bis_motor_impulsivity bis_nonplanning_impulsivity
## average.r                  0.39                        0.32
##           bis_attentional_impulsivity bis_total
## average.r                        0.35      0.24
##
##  Guttman 6* reliability:
##          bis_motor_impulsivity bis_nonplanning_impulsivity
## Lambda.6                  0.78                        0.75
##          bis_attentional_impulsivity bis_total
## Lambda.6                        0.78      0.87
##
## Signal/Noise based upon av.r :
##              bis_motor_impulsivity bis_nonplanning_impulsivity
## Signal/Noise                   3.2                         2.4
##              bis_attentional_impulsivity bis_total
## Signal/Noise                         2.7       4.8
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                            bis_motor_impulsivity
## bis_motor_impulsivity                       0.76
## bis_nonplanning_impulsivity                 0.31
## bis_attentional_impulsivity                 0.50
## bis_total                                   0.76
##                            bis_nonplanning_impulsivity
## bis_motor_impulsivity                             0.43
## bis_nonplanning_impulsivity                       0.71
## bis_attentional_impulsivity                       0.43
## bis_total                                         0.75
##                            bis_attentional_impulsivity bis_total
## bis_motor_impulsivity                             0.67      0.95
## bis_nonplanning_impulsivity                       0.60      0.99
## bis_attentional_impulsivity                       0.73      1.07
## bis_total                                         0.83      0.83
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

```
pairs.panels(tds2_wave1_bis_psych$scores)
```



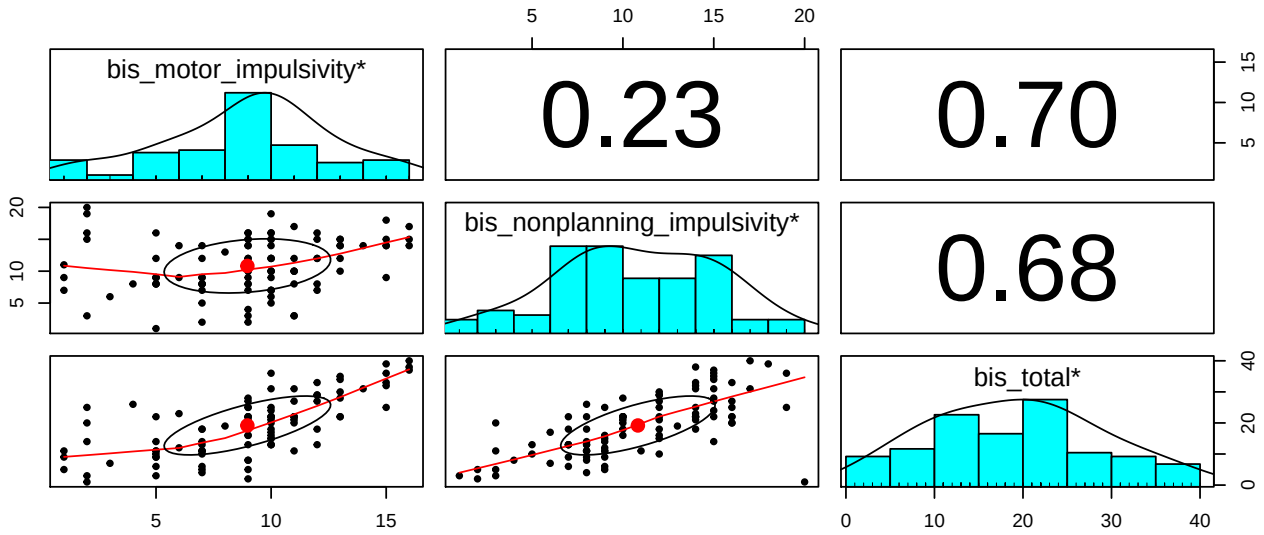
If we want to apply the `scorequaltrics` plot function, we need to transform the data to long-form using the `longen_psych_wide` function.

```
tds2_wave1_bis_score_long <- scorequaltrics::longen_psych_wide(tds2_wave1_bis_psych$scores)

scorequaltrics::plot_scored_scale(tds2_wave1_bis_score_long)
```



Finally, we can also use the `pairs.plot` function with data scored using the `scorequaltrics` back-end using the `widen_qualtrics_long` function again. This time I'll take the BIS-15 data from the main scored data set.

```
tds2_wave1_bis15_wide <- scorequaltrics::widen_qualtrics_long(tds2_wave1_scored, scale_names = 'BIS-15')

library(psych)
#remove the first two columns that contain the ID and scale name
pairs.panels(tds2_wave1_bis15_wide$scores[, -(1:2)])
```



Putting it all together

In the next section, I will simply walk through the scoring and display of descriptives for the data we have using the tools reviewed above.

TDS 2, Wave 1 Scores

Available scales:

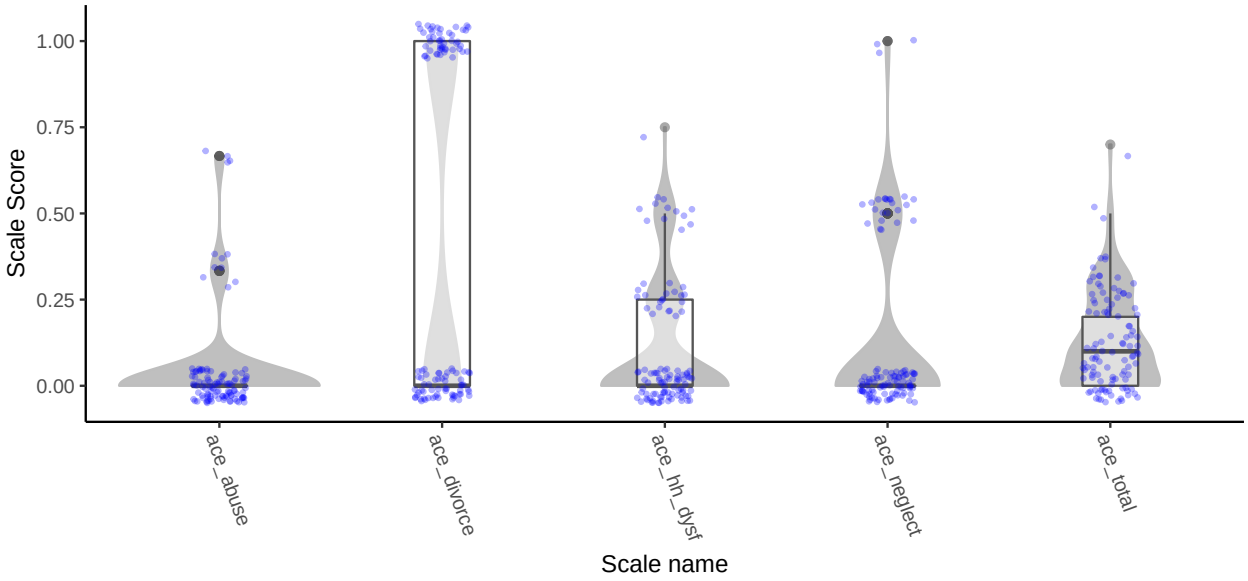
```
tds2_wave1_scored %>%
    ungroup() %>%
    distinct(scale_name) %>%
    knitr::kable()
```

scale_name
ACE
BFNE
BIS-15
Brief SCARED
BSSS
CARE-R Expected Involvement
CARE-R Social
CARE-R Willingness to Engage
CBCL
CES-DC
MSSSS
NTS
PAL-2
PDS
PEQ-R
SES
SPSRQ-S
UPPS-P
YRBS
RPI_part2

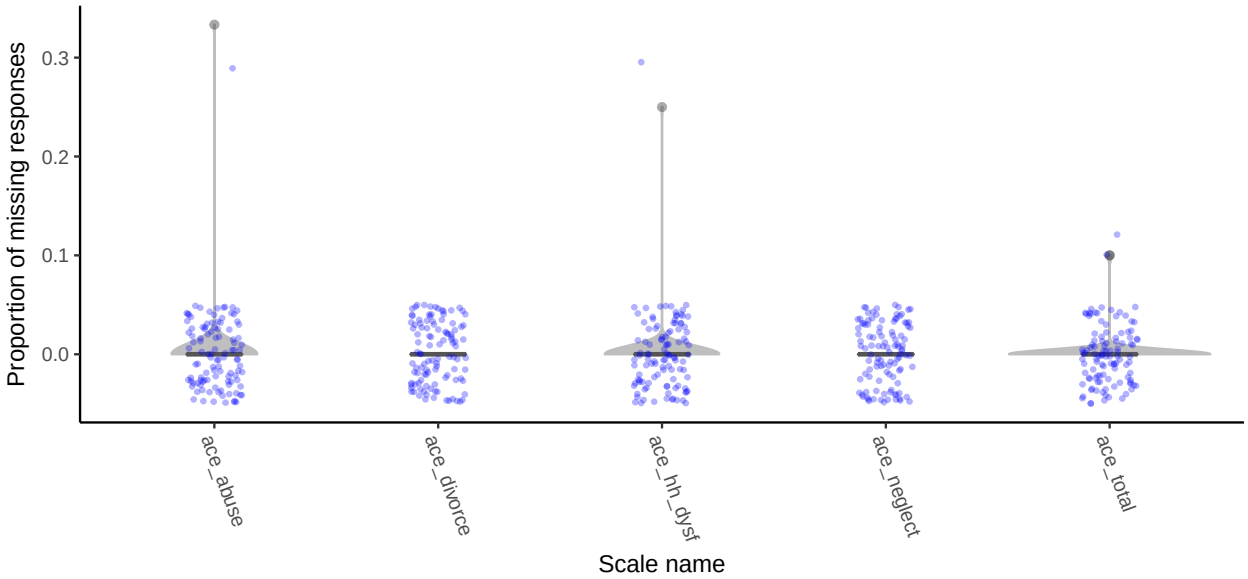
scale_name
RSQ_part2

ACE

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^ACE$',
                                  type = 'score')
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^ACE$',
                                  type = 'p_missing')
```



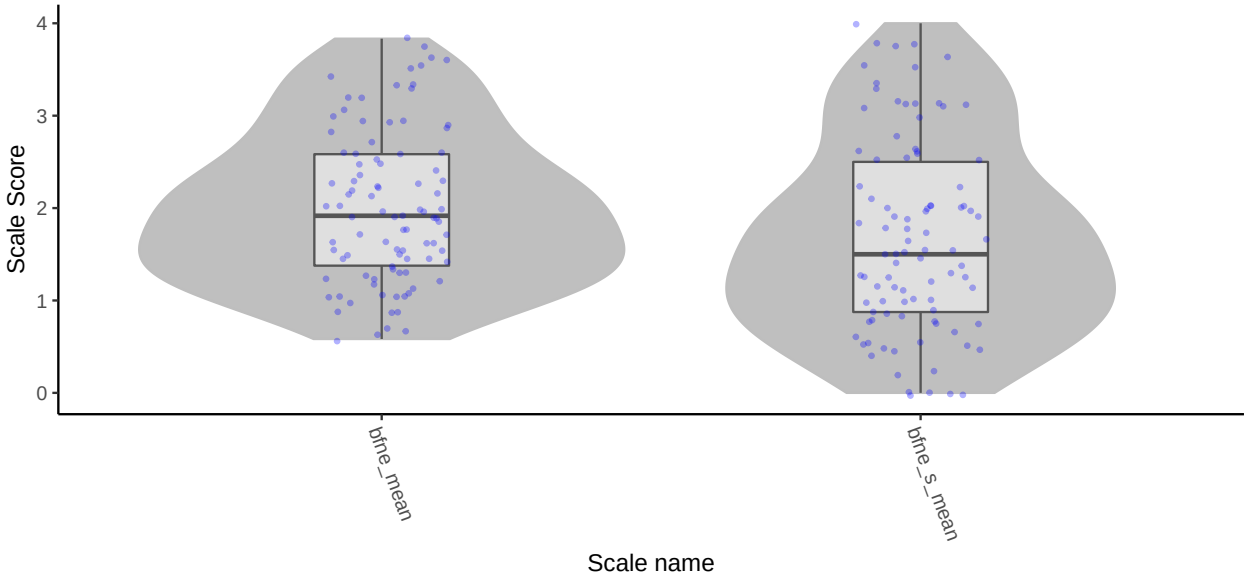
BFNE

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^BFNE$',
                                  type = 'score')
```

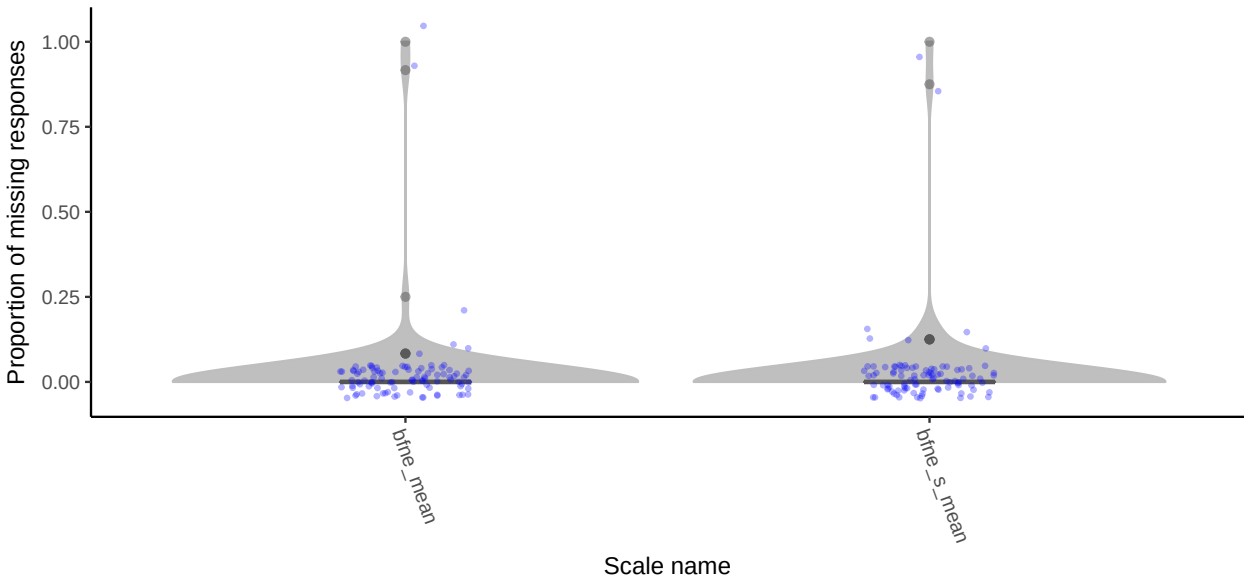
```
## Warning: Removed 2 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^BFNE$',
                                  type = 'p_missing')
```

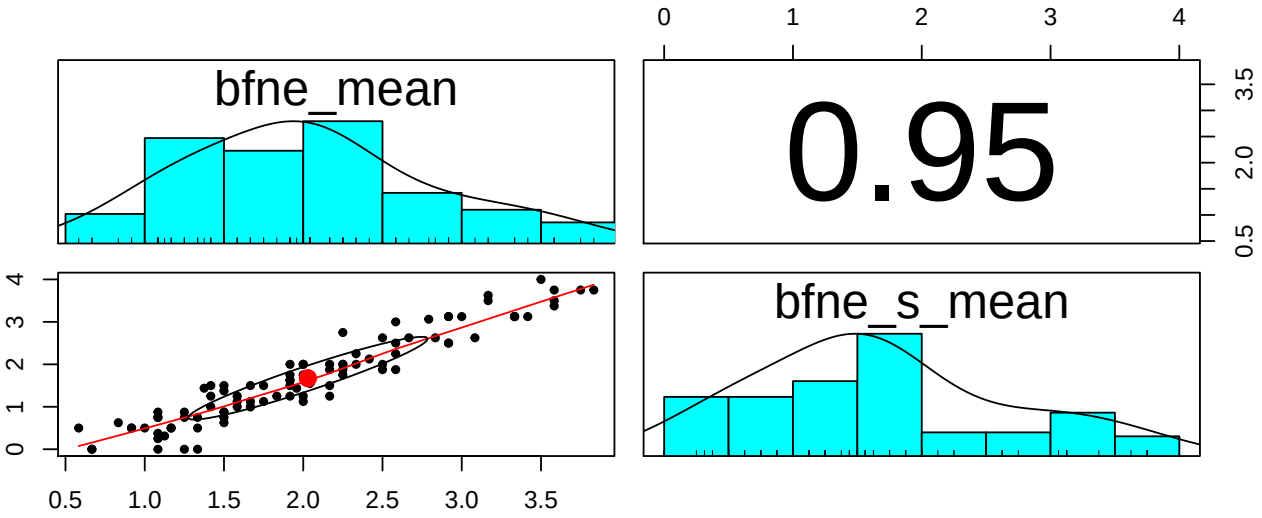



```
tds2w1_BFNERubric <- tds2_wave1_scoring_data_long %>%
    filter(scale_name == 'BFNE')
```

```r
tds2_wave1_bfne_psych <- scorequaltrics::score_questionnaire(tds2_wave1_long_recoded_nodupes,
                                    tds2w1_BFNERubric,
                                    psych = T)
print(tds2_wave1_bfne_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       bfne_mean bfne_s_mean
## alpha      0.89        0.94
##
## Standard errors of unstandardized Alpha:
##       bfne_mean bfne_s_mean
## ASE       0.027       0.027
##
## Average item correlation:
##           bfne_mean bfne_s_mean
## average.r       0.4        0.66
##
##  Guttman 6* reliability:
##          bfne_mean bfne_s_mean
## Lambda.6      0.92        0.95
##
## Signal/Noise based upon av.r :
##              bfne_mean bfne_s_mean
## Signal/Noise         8          16
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##             bfne_mean bfne_s_mean
## bfne_mean        0.89        1.04
## bfne_s_mean      0.95        0.94
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

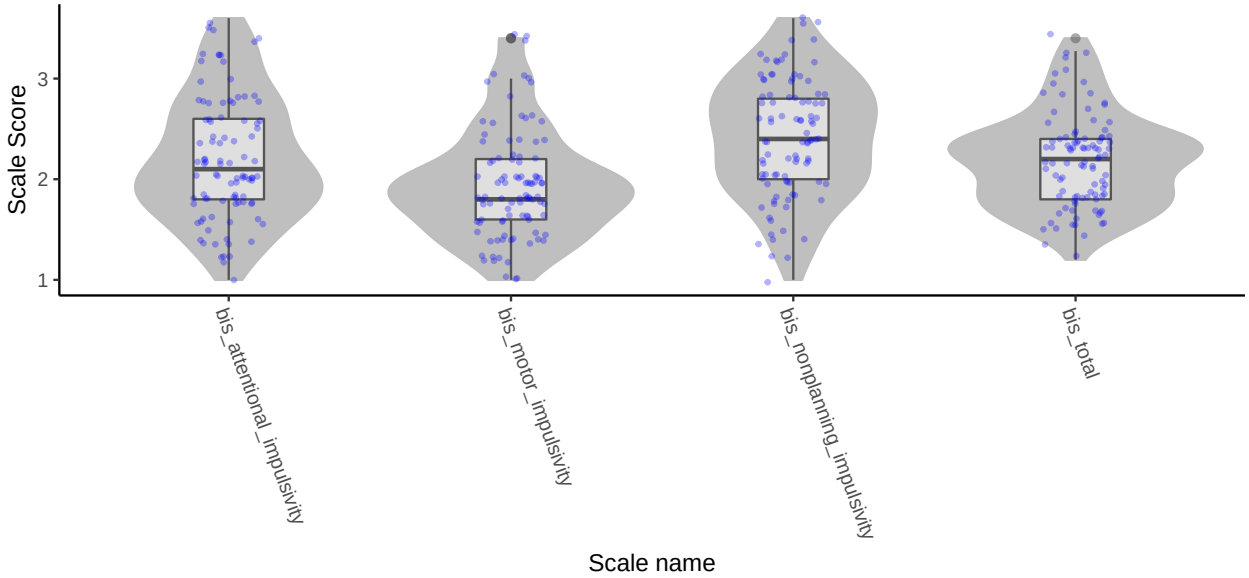
```r
pairs.panels(tds2_wave1_bfne_psych$scores)
```



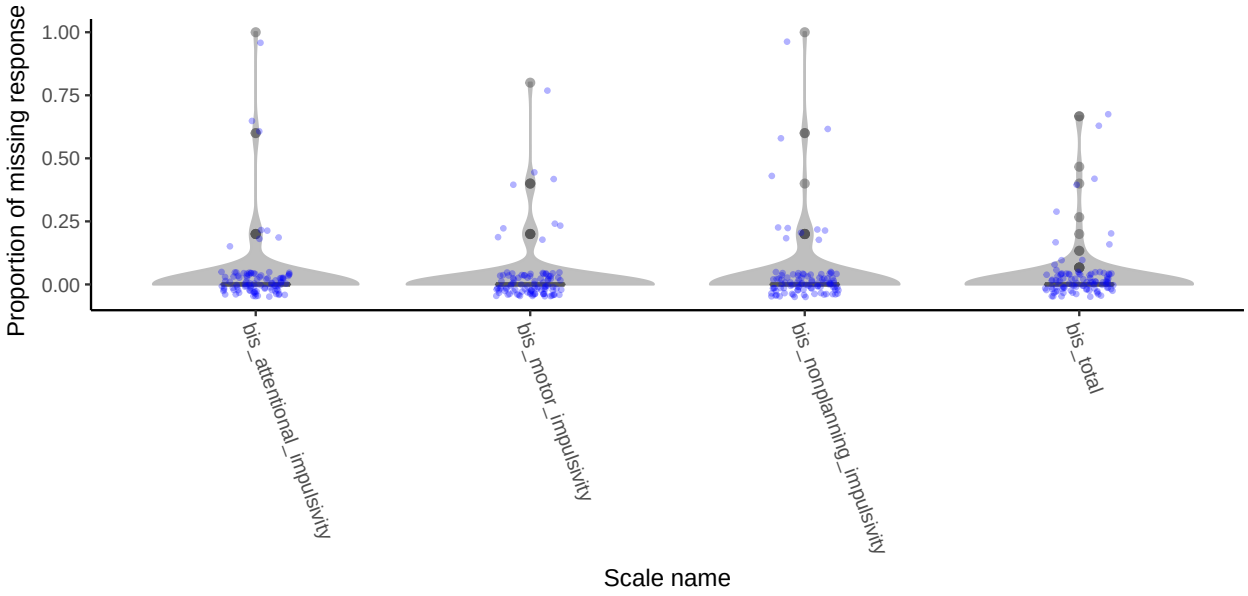
BIS-15

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^BIS-15$',
                                  type = 'score')
```

```
## Warning: Removed 2 rows containing non-finite values (stat_ydensity).
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
## Warning: Removed 2 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^BIS-15$',
                                  type = 'p_missing')
```

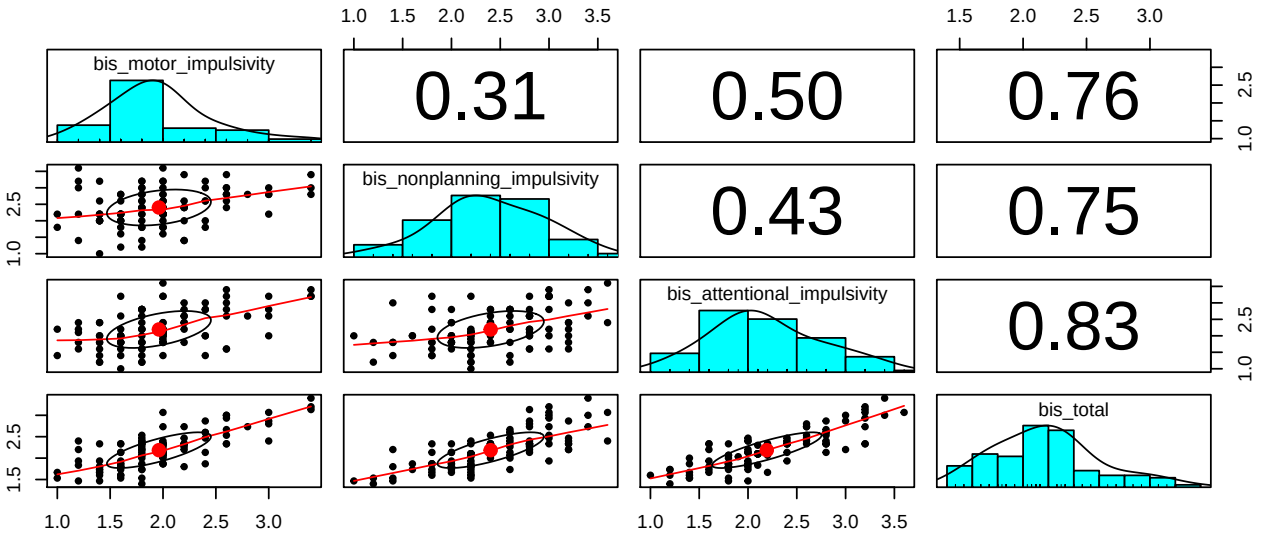



```
tds2w1_BISRubric <- tds2_wave1_scoring_data_long %>%
    filter(scale_name == 'BIS-15')

tds2_wave1_bis_psych <- scorequaltrics::score_questionnaire(tds2_wave1_long_recoded_nodupes,
                                    tds2w1_BISRubric,
                                    psych = T)
print(tds2_wave1_bis_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       bis_motor_impulsivity bis_nonplanning_impulsivity
## alpha                  0.76                        0.71
##       bis_attentional_impulsivity bis_total
## alpha                        0.73      0.83
##
## Standard errors of unstandardized Alpha:
##       bis_motor_impulsivity bis_nonplanning_impulsivity
## ASE                   0.065                       0.072
##       bis_attentional_impulsivity bis_total
## ASE                       0.068     0.033
##
## Average item correlation:
##           bis_motor_impulsivity bis_nonplanning_impulsivity
## average.r                  0.39                        0.32
##           bis_attentional_impulsivity bis_total
## average.r                        0.35      0.24
##
##  Guttman 6* reliability:
##          bis_motor_impulsivity bis_nonplanning_impulsivity
## Lambda.6                  0.78                        0.75
##          bis_attentional_impulsivity bis_total
## Lambda.6                        0.78      0.87
##
## Signal/Noise based upon av.r :
```

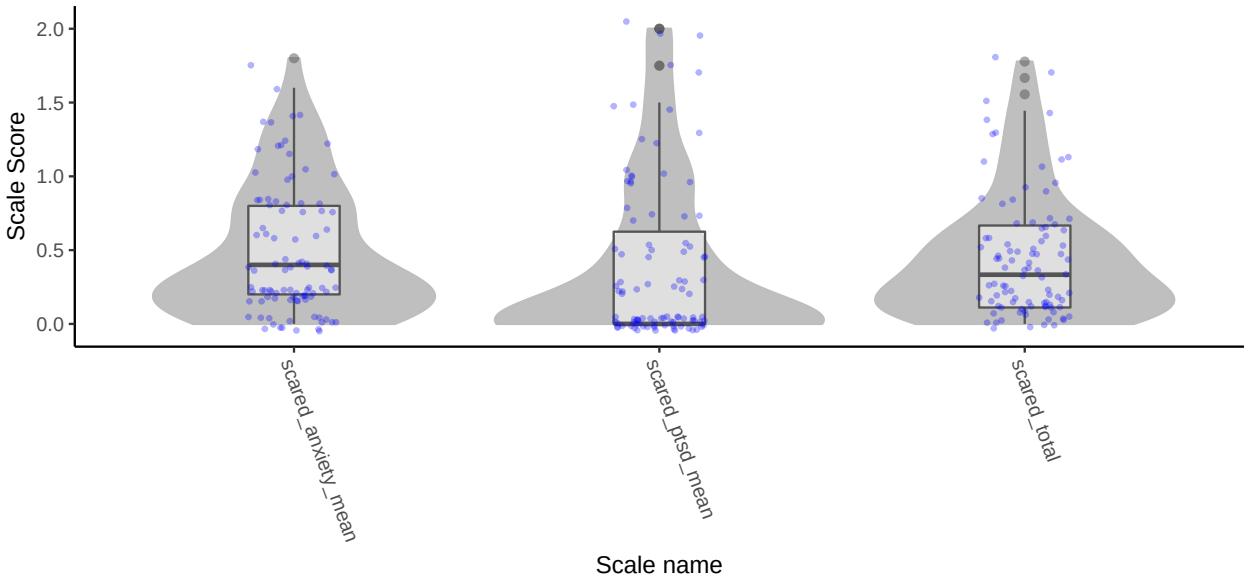
```
##              bis_motor_impulsivity bis_nonplanning_impulsivity
## Signal/Noise                   3.2                         2.4
##              bis_attentional_impulsivity bis_total
## Signal/Noise                         2.7       4.8
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                              bis_motor_impulsivity
## bis_motor_impulsivity                         0.76
## bis_nonplanning_impulsivity                   0.31
## bis_attentional_impulsivity                   0.50
## bis_total                                     0.76
##                              bis_nonplanning_impulsivity
## bis_motor_impulsivity                               0.43
## bis_nonplanning_impulsivity                         0.71
## bis_attentional_impulsivity                         0.43
## bis_total                                           0.75
##                              bis_attentional_impulsivity bis_total
## bis_motor_impulsivity                               0.67      0.95
## bis_nonplanning_impulsivity                         0.60      0.99
## bis_attentional_impulsivity                         0.73      1.07
## bis_total                                           0.83      0.83
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

```
pairs.panels(tds2_wave1_bis_psych$scores)
```

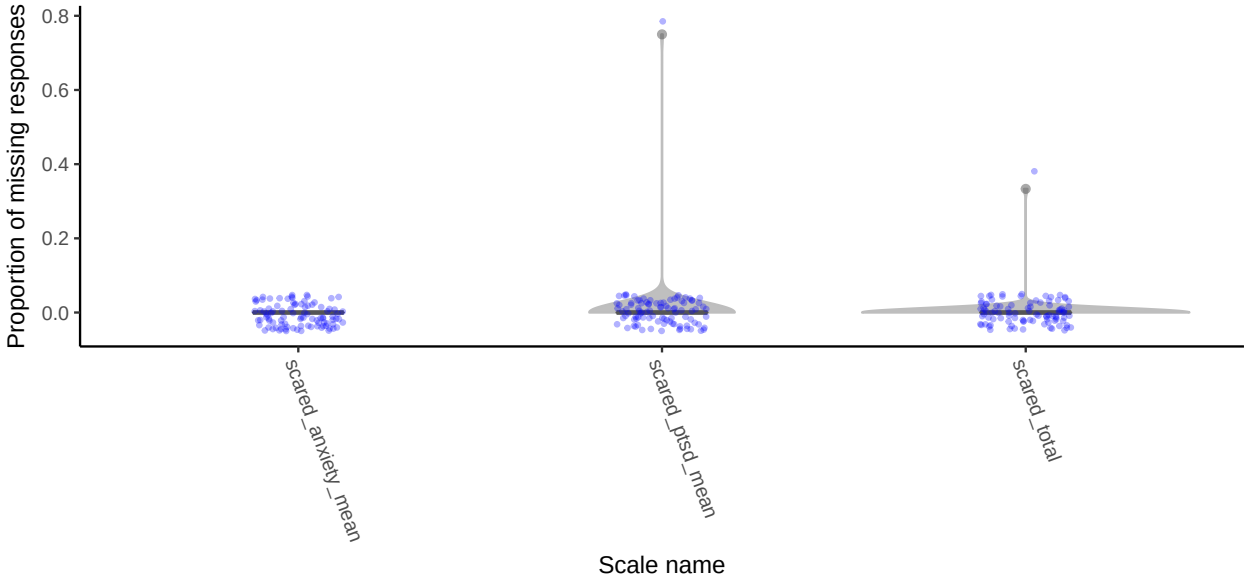


Brief SCARED

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^Brief SCARED$',
                                  type = 'score')
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^Brief SCARED$',
                                  type = 'p_missing')
```

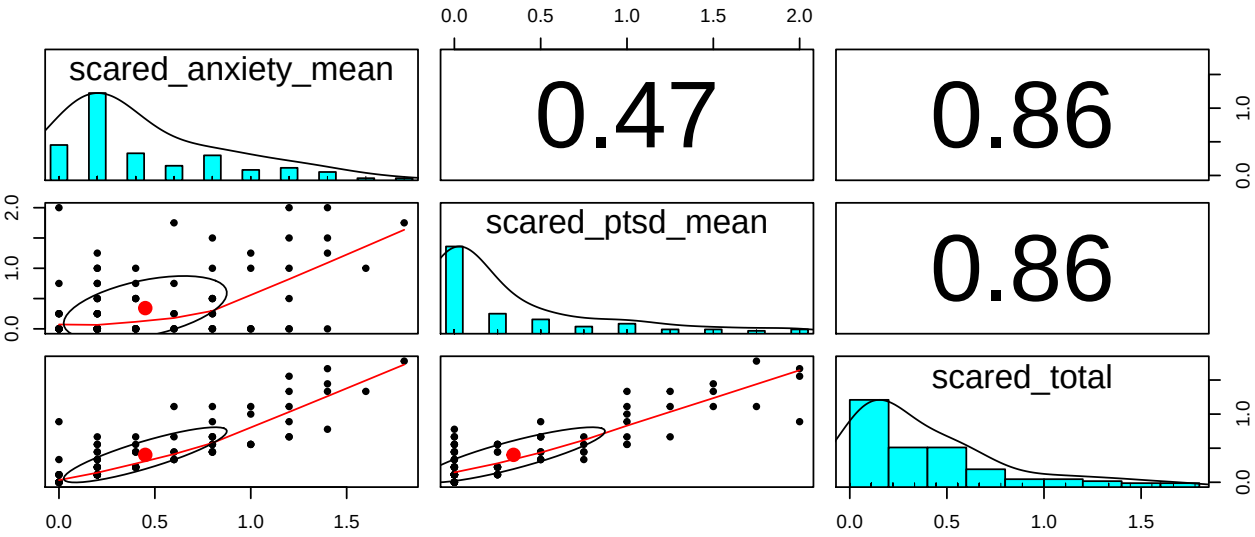



```
tds2w1_BscaredRubric <- tds2_wave1_scoring_data_long %>%
    filter(scale_name == 'Brief SCARED')

tds2_wave1_bscared_psych <- scorequaltrics::score_questionnaire(tds2_wave1_long_recoded_nodupes,
                                  tds2w1_BscaredRubric,
                                  psych = T)
print(tds2_wave1_bscared_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       scared_anxiety_mean scared_ptsd_mean scared_total
## alpha                0.73             0.88         0.84
```

```
##
## Standard errors of unstandardized Alpha:
##       scared_anxiety_mean scared_ptsd_mean scared_total
## ASE                 0.069            0.057        0.039
##
## Average item correlation:
##           scared_anxiety_mean scared_ptsd_mean scared_total
## average.r                0.35             0.66         0.36
##
##  Guttman 6* reliability:
##          scared_anxiety_mean scared_ptsd_mean scared_total
## Lambda.6                0.74             0.88         0.87
##
## Signal/Noise based upon av.r :
##              scared_anxiety_mean scared_ptsd_mean scared_total
## Signal/Noise                 2.6              7.7          5.1
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                     scared_anxiety_mean scared_ptsd_mean scared_total
## scared_anxiety_mean                0.73             0.59         1.10
## scared_ptsd_mean                   0.47             0.88         1.00
## scared_total                       0.86             0.86         0.84
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

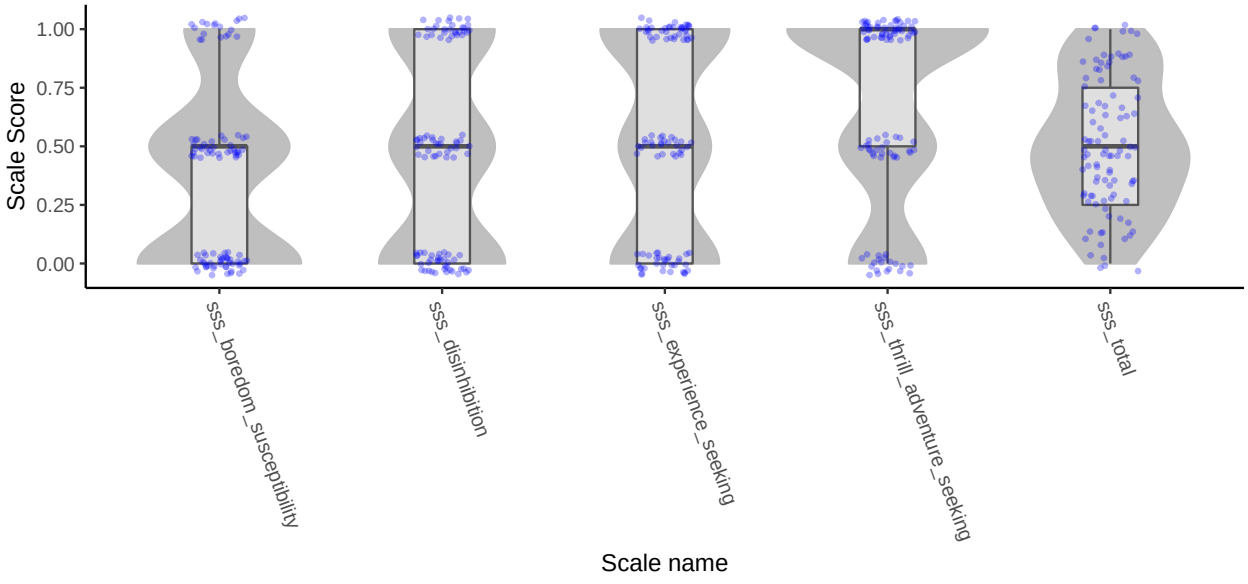
```
pairs.panels(tds2_wave1_bscared_psych$scores)
```



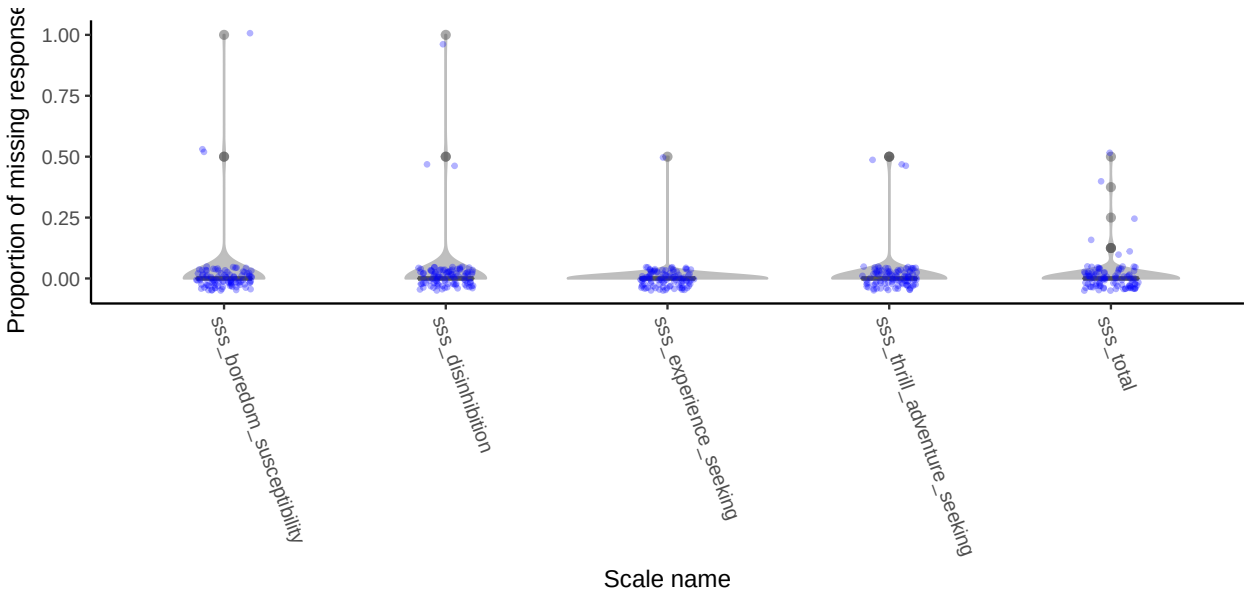
BSSS

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^BSSS$',
                                  type = 'score')
```

```
## Warning: Removed 2 rows containing non-finite values (stat_ydensity).
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
## Warning: Removed 2 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^BSSS$',
                                  type = 'p_missing')
```

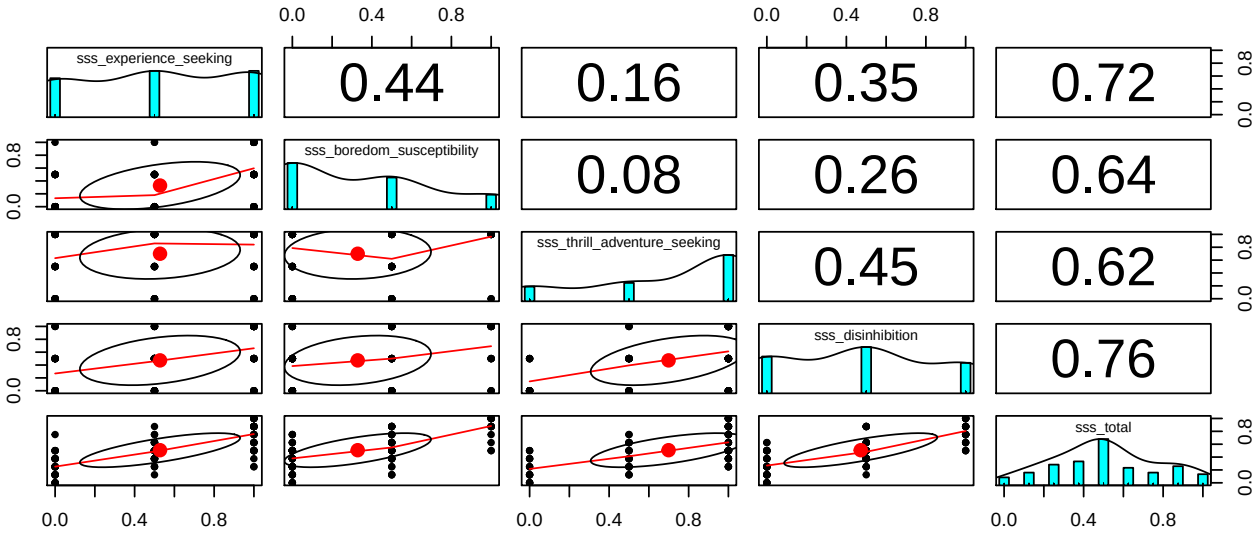



```
tds2w1_BSSSRubric <- tds2_wave1_scoring_data_long %>%
    filter(scale_name == 'BSSS')

tds2_wave1_bsss_psych <- scorequaltrics::score_questionnaire(tds2_wave1_long_recoded_nodupes,
                                  tds2w1_BSSSRubric,
                                  psych = T)
print(tds2_wave1_bsss_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       sss_experience_seeking sss_boredom_susceptibility
## alpha                   0.53                       0.37
##       sss_thrill_adventure_seeking sss_disinhibition sss_total
## alpha                         0.62              0.42      0.68
##
## Standard errors of unstandardized Alpha:
##       sss_experience_seeking sss_boredom_susceptibility
## ASE                     0.16                       0.18
##       sss_thrill_adventure_seeking sss_disinhibition sss_total
## ASE                           0.15              0.17     0.062
##
## Average item correlation:
##           sss_experience_seeking sss_boredom_susceptibility
## average.r                   0.36                       0.22
##           sss_thrill_adventure_seeking sss_disinhibition sss_total
## average.r                         0.45              0.27      0.21
##
##  Guttman 6* reliability:
##          sss_experience_seeking sss_boredom_susceptibility
## Lambda.6                    0.5                        0.4
##          sss_thrill_adventure_seeking sss_disinhibition sss_total
## Lambda.6                         0.55              0.43      0.72
##
## Signal/Noise based upon av.r :
##              sss_experience_seeking sss_boredom_susceptibility
## Signal/Noise                    1.1                       0.58
##              sss_thrill_adventure_seeking sss_disinhibition sss_total
## Signal/Noise                          1.6              0.73       2.1
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                             sss_experience_seeking
## sss_experience_seeking                        0.53
## sss_boredom_susceptibility                    0.44
## sss_thrill_adventure_seeking                  0.16
## sss_disinhibition                             0.35
## sss_total                                     0.72
##                             sss_boredom_susceptibility
## sss_experience_seeking                           0.999
## sss_boredom_susceptibility                       0.366
## sss_thrill_adventure_seeking                     0.076
## sss_disinhibition                                0.262
## sss_total                                        0.639
##                             sss_thrill_adventure_seeking
## sss_experience_seeking                              0.27
## sss_boredom_susceptibility                          0.16
## sss_thrill_adventure_seeking                        0.62
## sss_disinhibition                                   0.45
## sss_total                                           0.62
##                             sss_disinhibition sss_total
```

```
## sss_experience_seeking                0.74      1.19
## sss_boredom_susceptibility            0.67      1.28
## sss_thrill_adventure_seeking          0.89      0.95
## sss_disinhibition                     0.42      1.41
## sss_total                             0.76      0.68
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

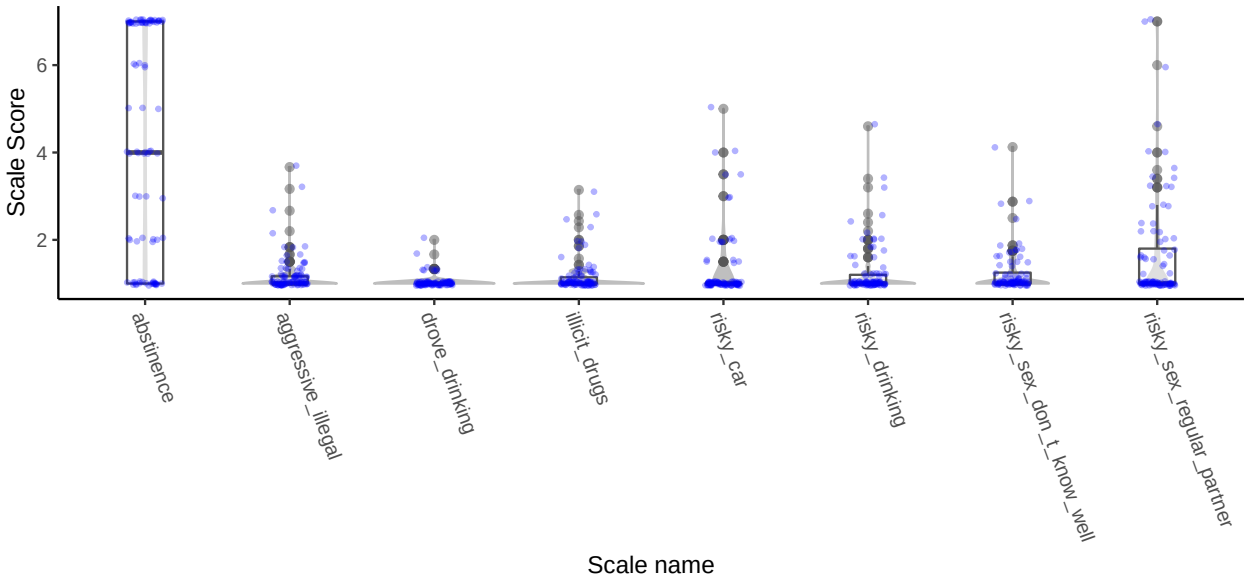
```
pairs.panels(tds2_wave1_bsss_psych$scores)
```



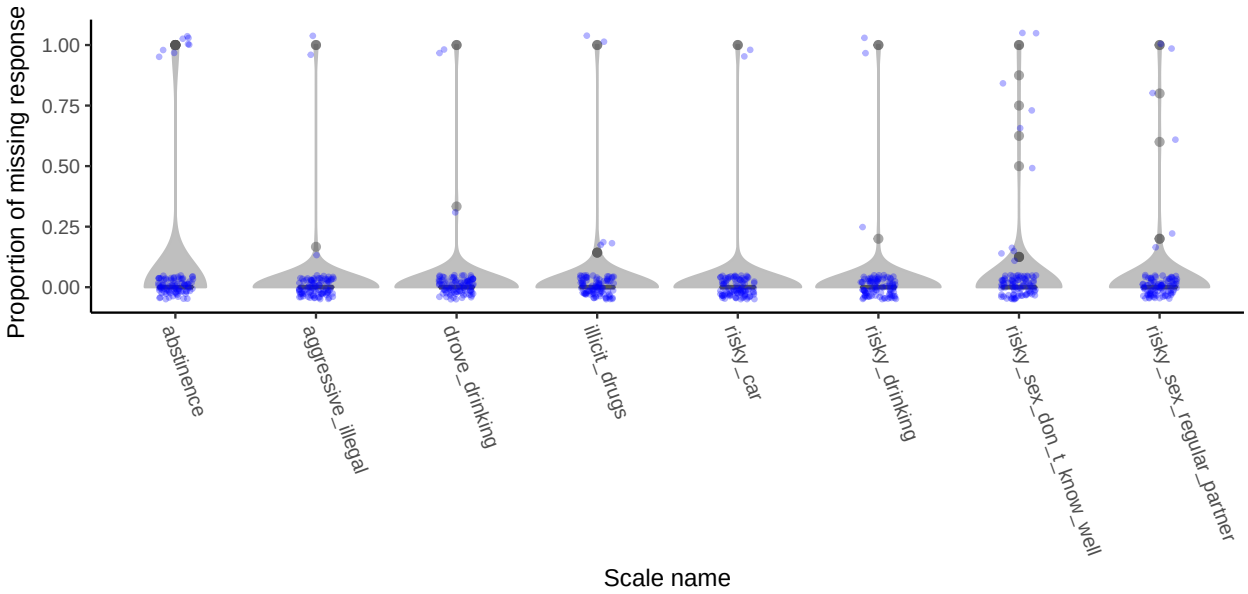
CARE-R Expected Involvement

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^CARE-R Expected Involvement$',
                                  type = 'score')
```

```
## Warning: Removed 22 rows containing non-finite values (stat_ydensity).
## Warning: Removed 22 rows containing non-finite values (stat_boxplot).
## Warning: Removed 22 rows containing missing values (geom_point).
```



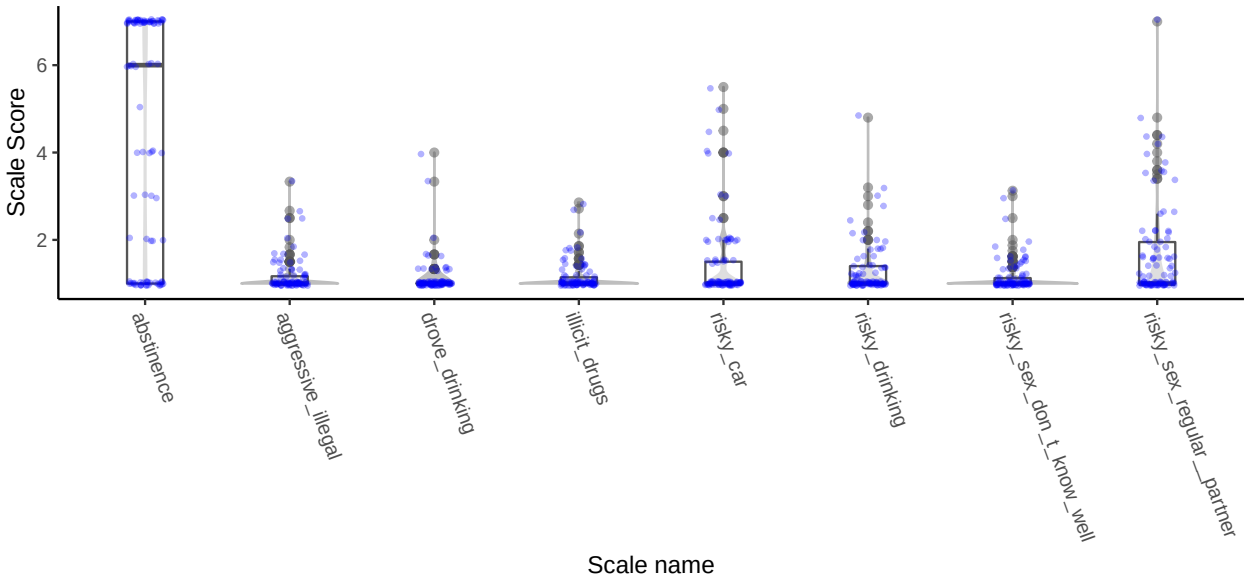

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^CARE-R Expected Involvement$',
                                  type = 'p_missing')
```



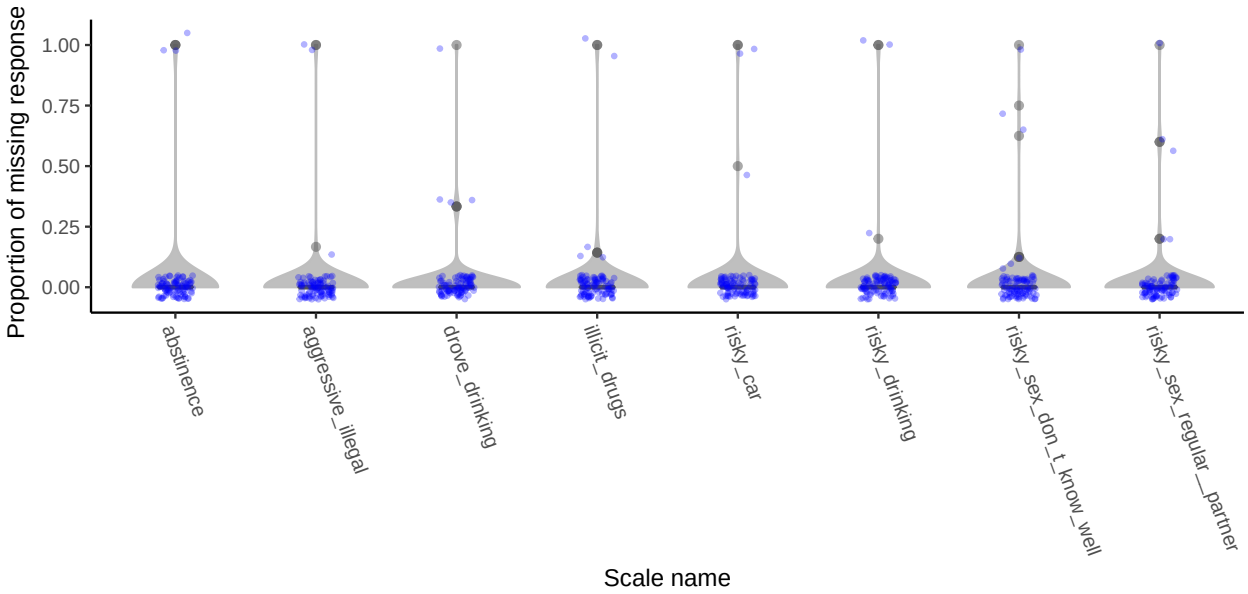
CARE-R Willingness to Engage

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^CARE-R Willingness to Engage$',
                                  type = 'score')
```

```
## Warning: Removed 14 rows containing non-finite values (stat_ydensity).
## Warning: Removed 14 rows containing non-finite values (stat_boxplot).
## Warning: Removed 14 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^CARE-R Willingness to Engage$',
                                  type = 'p_missing')
```



CARE-R Social

Note: Lower scores on "not doing this" items indicate that someone thinks not engaging in the behavior will make people like them *more*, and thus, like the "doing this" items, a lower score corresponds to a downward influence on that behavior.

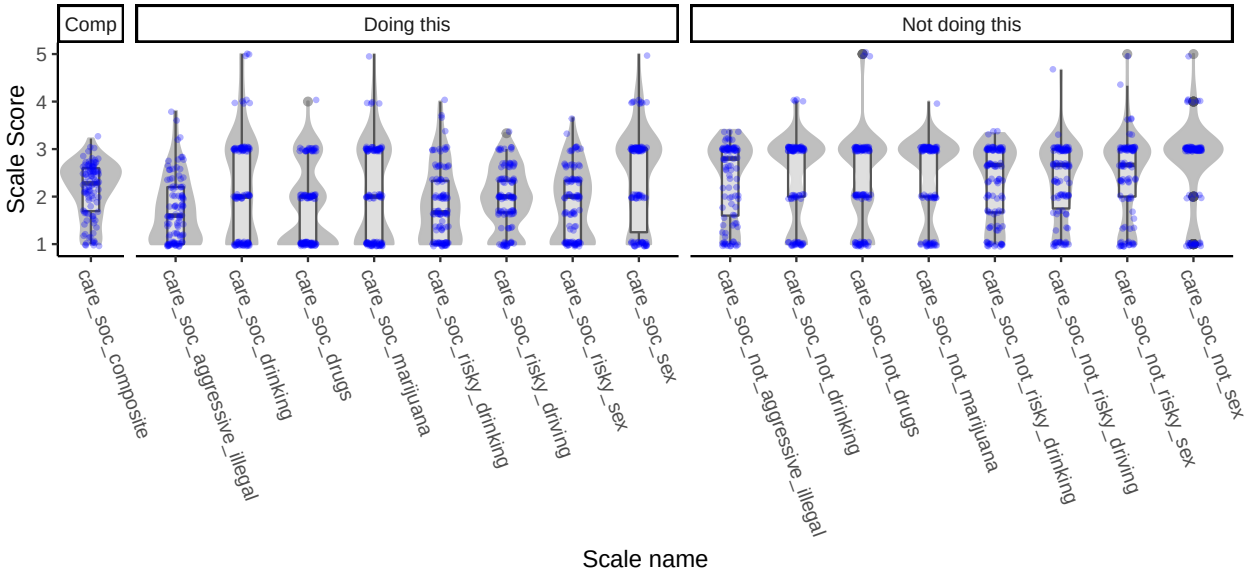
```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^CARE-R Social$',
                                  type = 'score') +
    facet_grid(~factor(grepl('not', scored_scale) - grepl('composite', scored_scale),
                       levels = c(-1, 0, 1),
                       labels = c('Comp', 'Doing this', 'Not doing this')),
```

```
                scales = 'free_x',
                space = 'free_x')
```

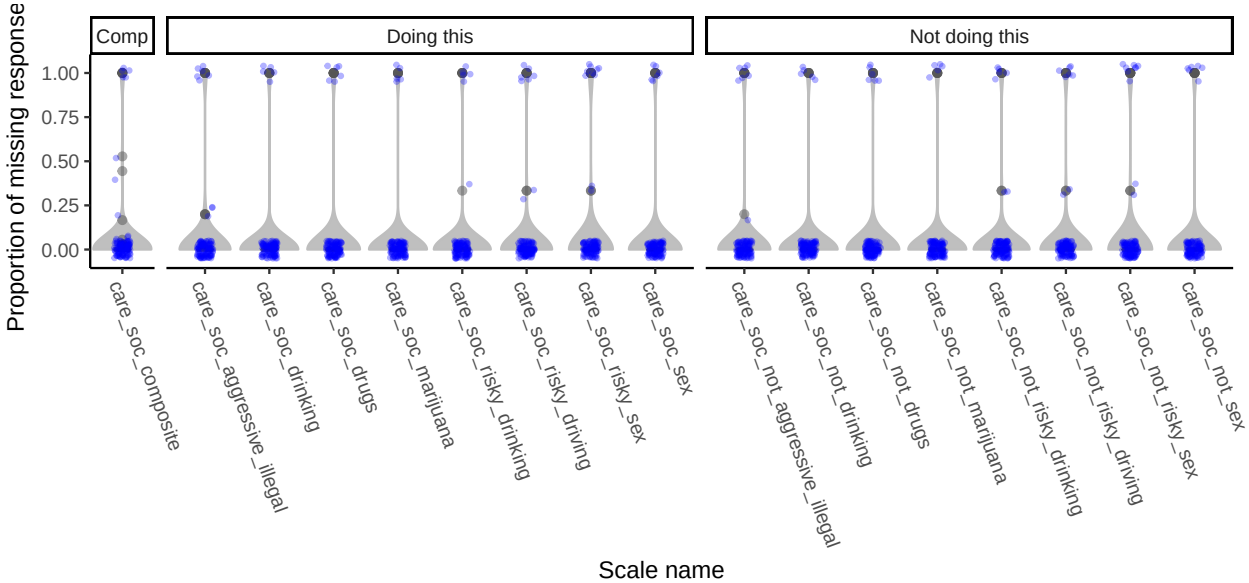
```
## Warning: Removed 99 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 99 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 99 rows containing missing values (geom_point).
```

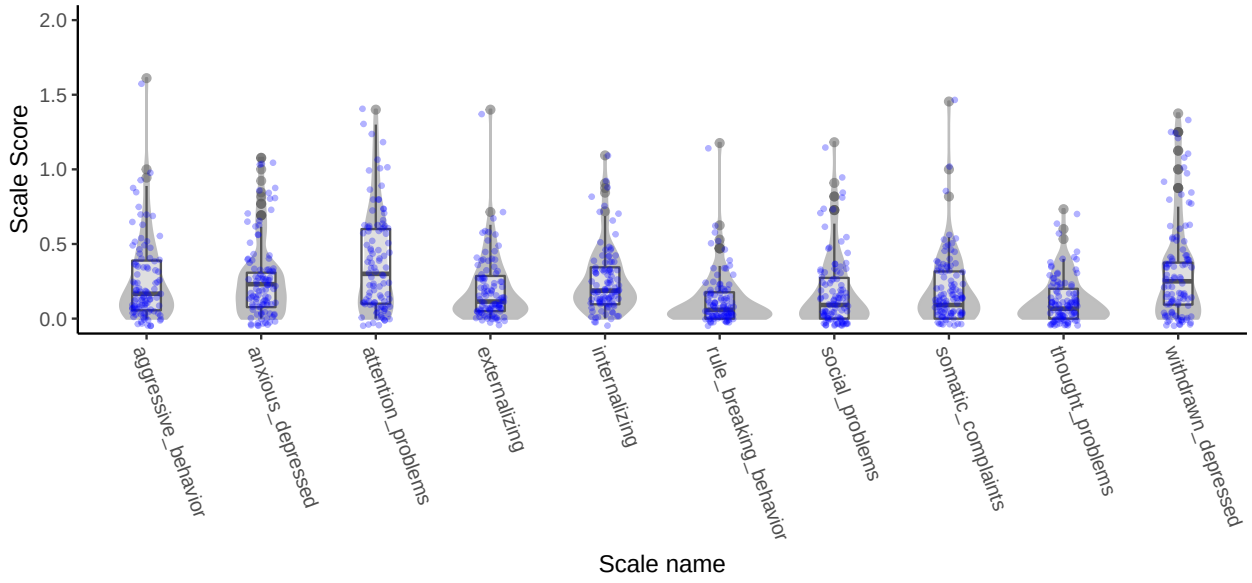



```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^CARE-R Social$',
                                  type = 'p_missing') +
    facet_grid(~factor(grepl('not', scored_scale) - grepl('composite', scored_scale),
                       levels = c(-1, 0, 1),
                       labels = c('Comp', 'Doing this', 'Not doing this')),
               scales = 'free_x',
               space = 'free_x')
```

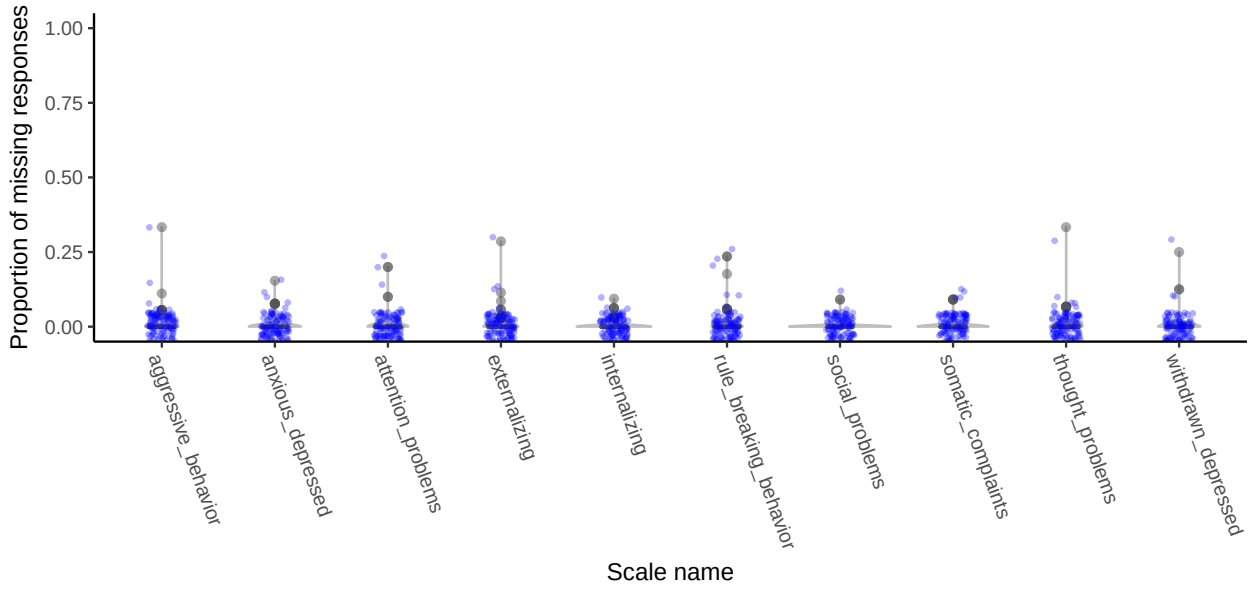


CBCL

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^CBCL$',
                                  type = 'score')+
    coord_cartesian(y = c(0,2))
```



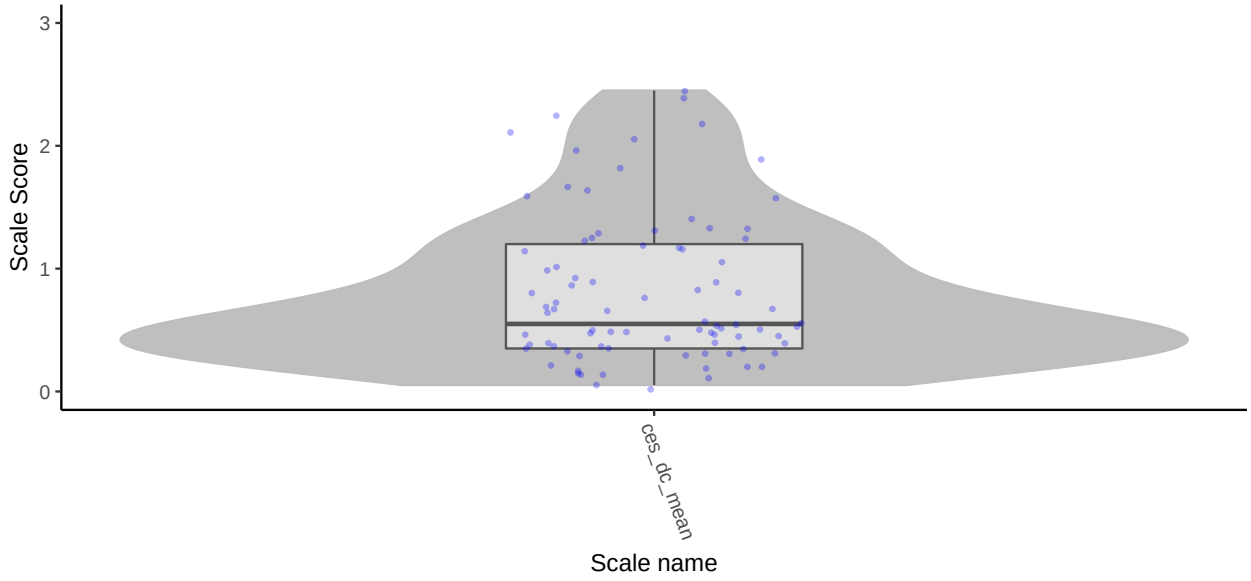

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^CBCL$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```



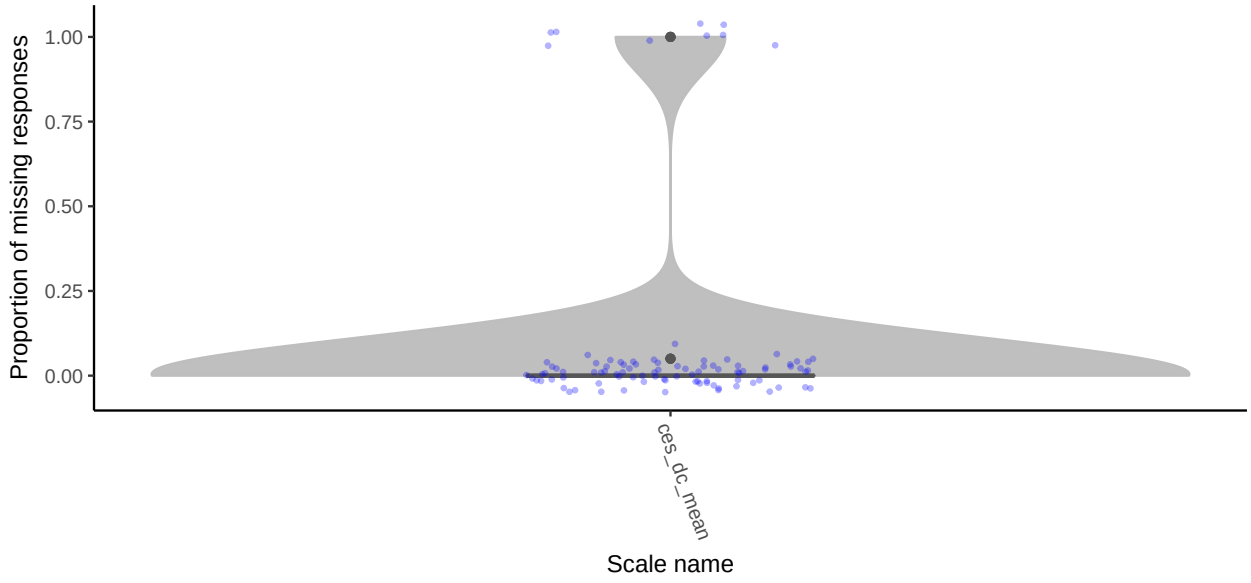
CES-DC

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^CES-DC$',
                                  type = 'score')+
    coord_cartesian(y = c(0,3))
```

```
## Warning: Removed 9 rows containing non-finite values (stat_ydensity).
## Warning: Removed 9 rows containing non-finite values (stat_boxplot).
## Warning: Removed 9 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^CES-DC$',
                                  type = 'p_missing')
```



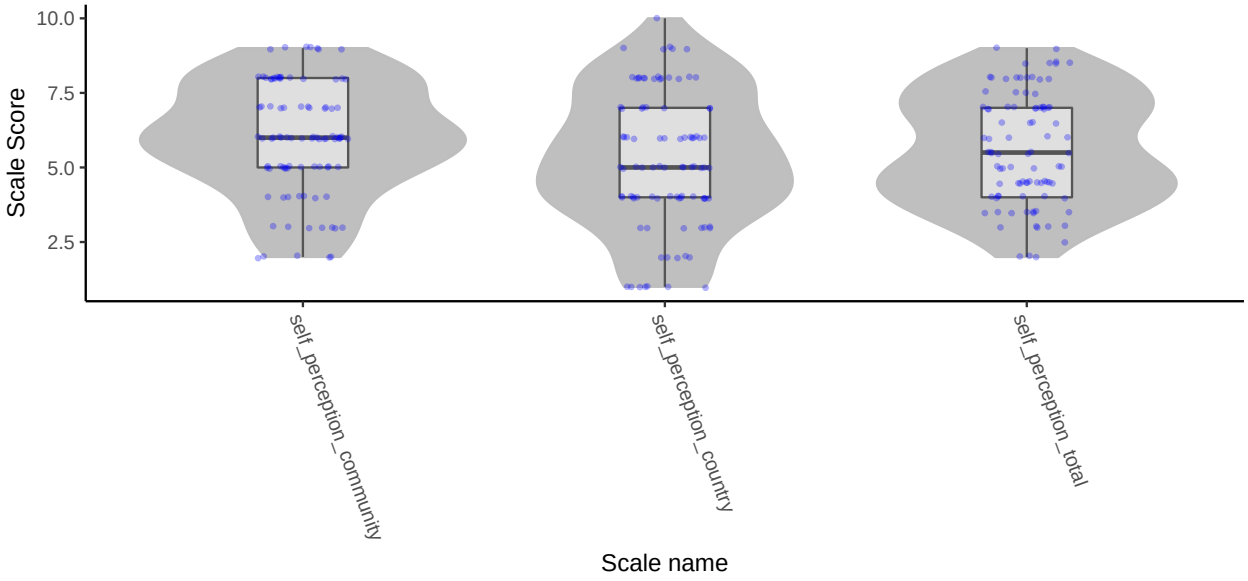
MSSSS

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^MSSSS$',
                                  type = 'score')
```

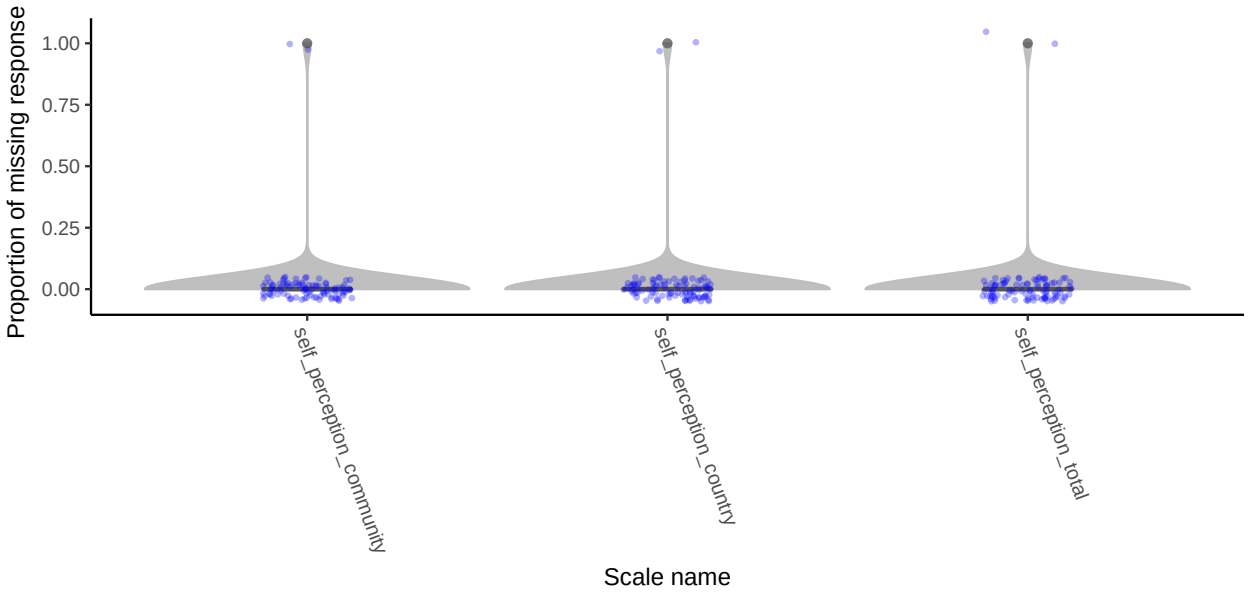
```
## Warning: Removed 6 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 6 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^MSSSS$',
                                  type = 'p_missing')
```



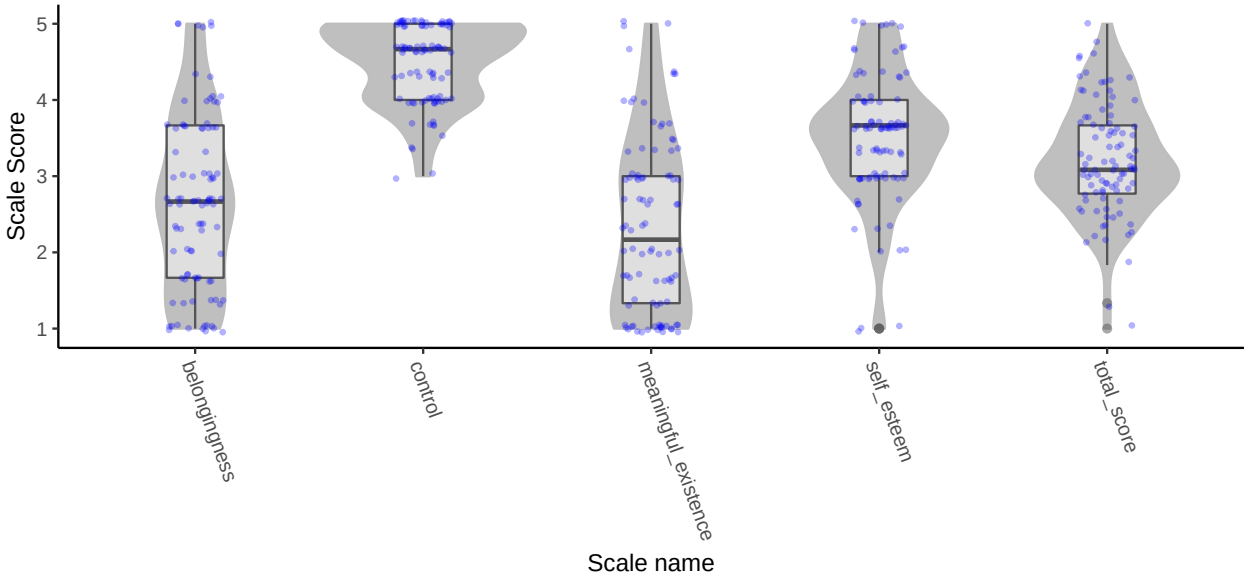
NTS

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^NTS$',
                                  type = 'score')
```

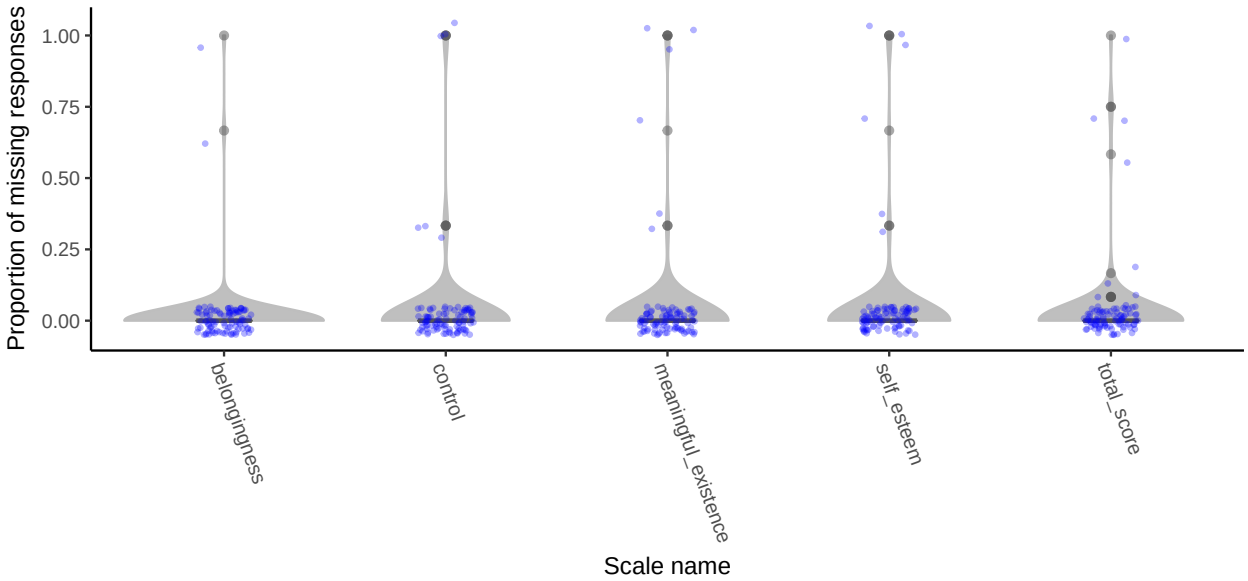
```
## Warning: Removed 11 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 11 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^NTS$',
                                  type = 'p_missing')
```

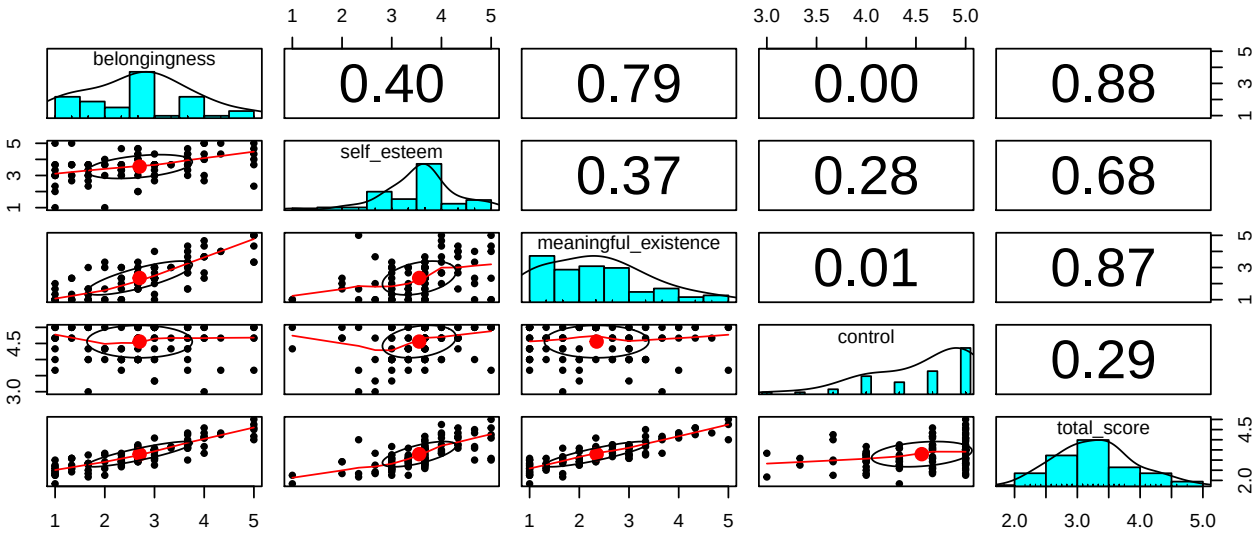



```
tds2w1_NTSRubric <- tds2_wave1_scoring_data_long %>%
    filter(scale_name == 'NTS')
```

```
tds2_wave1_nts_psych <- scorequaltrics::score_questionnaire(tds2_wave1_long_recoded_nodupes,
                                    tds2w1_NTSRubric,
                                    psych = T)
print(tds2_wave1_nts_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       belongingness self_esteem meaningful_existence control total_score
## alpha           0.9        0.72                 0.86    0.63        0.85
##
## Standard errors of unstandardized Alpha:
##       belongingness self_esteem meaningful_existence control total_score
## ASE           0.072       0.097                0.077    0.11       0.032
##
## Average item correlation:
##           belongingness self_esteem meaningful_existence control
## average.r          0.74        0.46                 0.67    0.36
##           total_score
## average.r        0.32
##
##  Guttman 6* reliability:
##          belongingness self_esteem meaningful_existence control
## Lambda.6          0.89        0.71                 0.86     0.6
##          total_score
## Lambda.6        0.91
##
## Signal/Noise based upon av.r :
##              belongingness self_esteem meaningful_existence control
## Signal/Noise           8.7         2.6                  6.2     1.7
##              total_score
## Signal/Noise         5.8
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                      belongingness self_esteem meaningful_existence
## belongingness               0.8966        0.49               0.8952
## self_esteem                 0.3952        0.72               0.4655
## meaningful_existence        0.7868        0.37               0.8616
## control                    -0.0028        0.28               0.0051
## total_score                 0.8758        0.68               0.8681
##                      control total_score
## belongingness        -0.0037        1.00
## self_esteem           0.4117        0.87
## meaningful_existence  0.0069        1.01
## control               0.6329        0.39
## total_score           0.2867        0.85
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

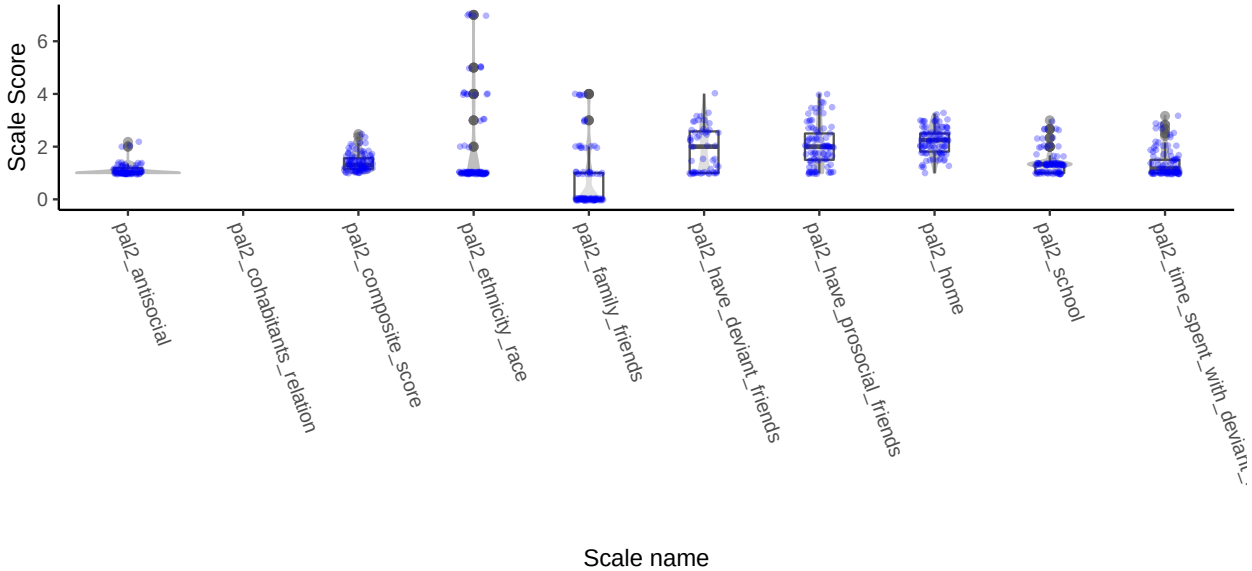
```
pairs.panels(tds2_wave1_nts_psych$scores)
```



PAL-2

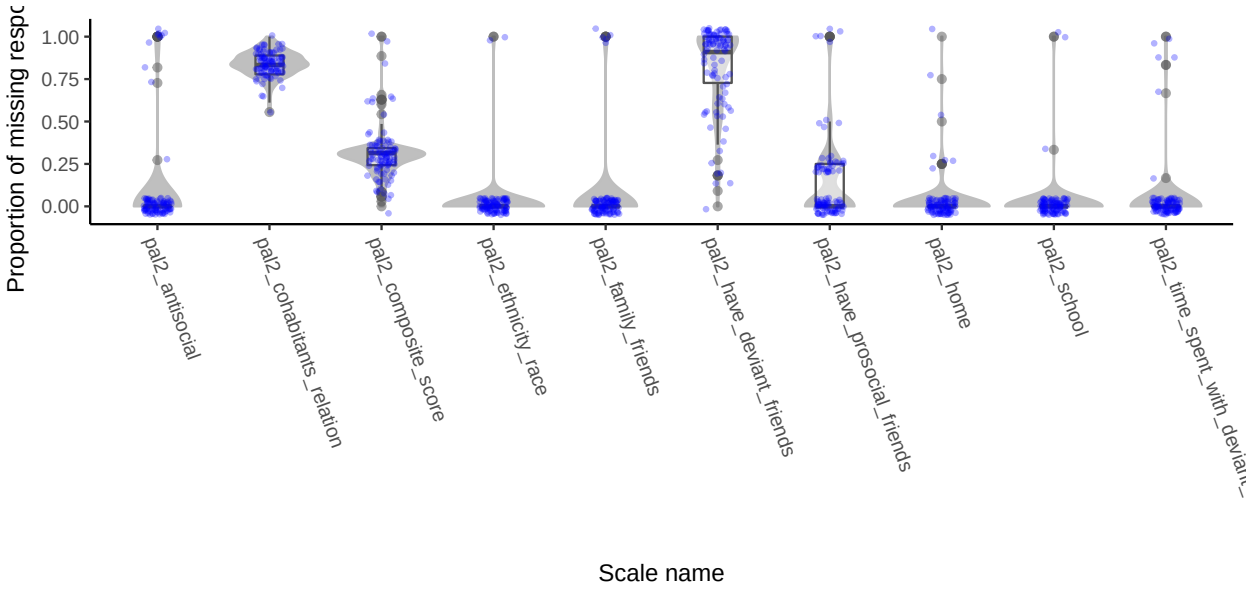
```
scorequaltrics::plot_scored_scale(filter(tds2_wave1_scored, scored_scale != 'pal2_age'),
                                  scale_regx = '^PAL-2$',
                                  type = 'score')
```

```
## Warning: Removed 164 rows containing non-finite values (stat_ydensity).
## Warning: Removed 164 rows containing non-finite values (stat_boxplot).
## Warning: Removed 164 rows containing missing values (geom_point).
```



Scale name

```
scorequaltrics::plot_scored_scale(filter(tds2_wave1_scored, scored_scale != 'pal2_age'),
                                  scale_regx = '^PAL-2$',
                                  type = 'p_missing')
```



PDS

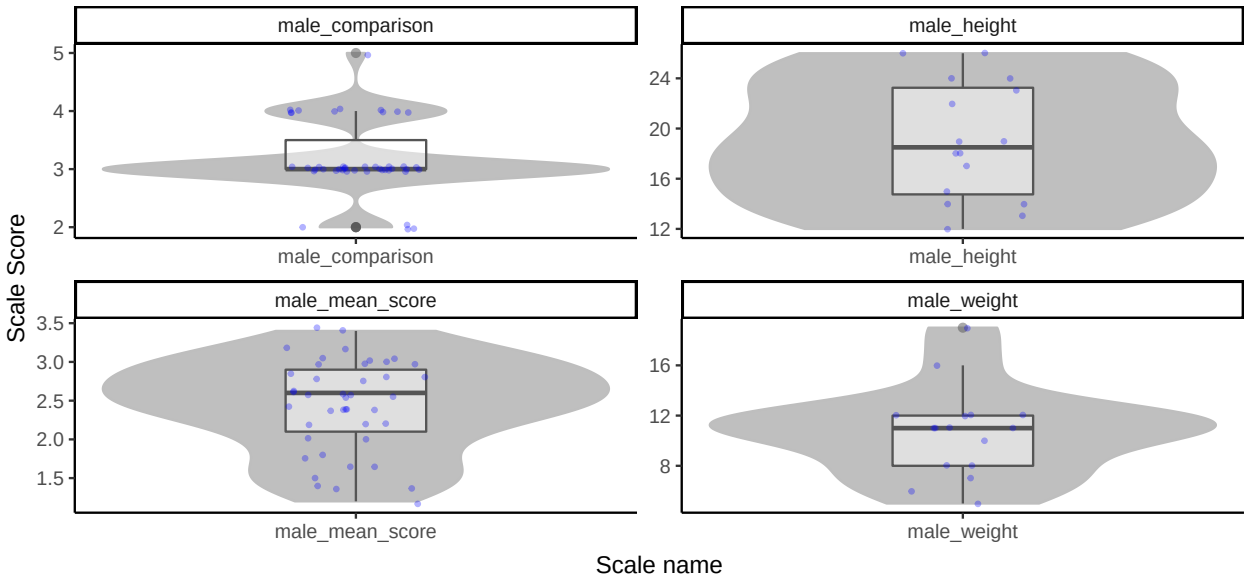
NEED TO GET PEOPLE WHO DID SESSION 2 PDS

```
#We need a way to separate female from male responses
male_sids <- tds2_wave1_scored %>%
    filter(scored_scale == 'pds_gender', score =='0') %>%
    ungroup() %>%
    select(SID)

female_sids <- tds2_wave1_scored %>%
    filter(scored_scale == 'pds_gender', score =='1') %>%
    ungroup() %>%
    select(SID)

scorequaltrics::plot_scored_scale(filter(tds2_wave1_scored,
                                         SID %in% male_sids$SID,
                                         grepl('^male', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'score') +
    facet_wrap(~scored_scale, scales = 'free') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

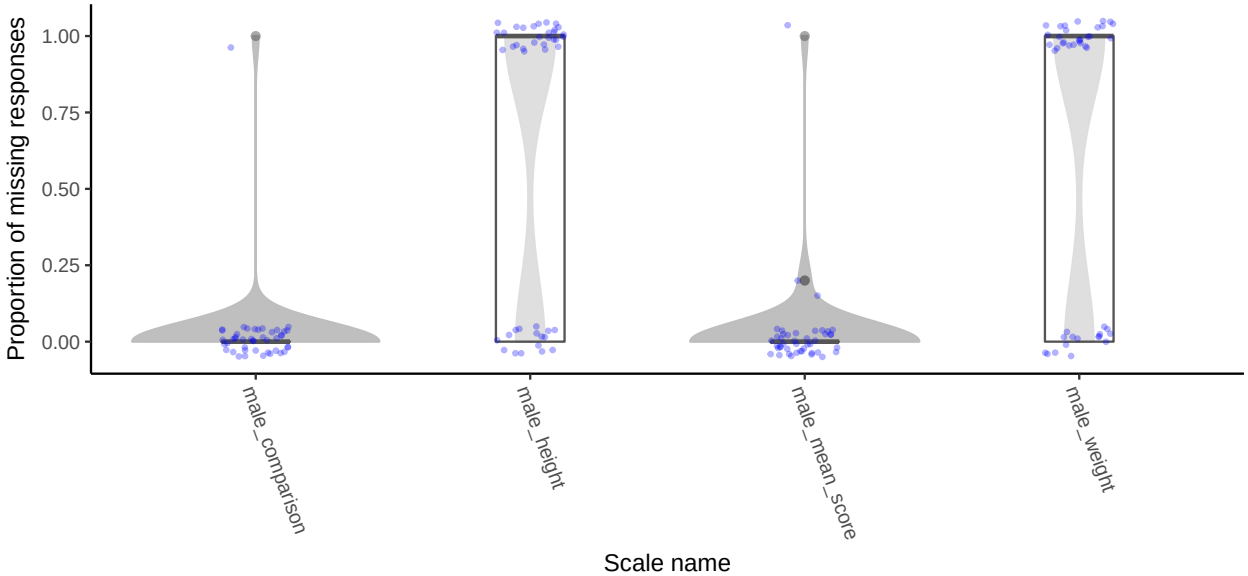
```
## Warning: Removed 58 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 58 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 58 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(filter(tds2_wave1_scored,
                                         SID %in% male_sids$SID,
                                         grepl('^male', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'p_missing')
```

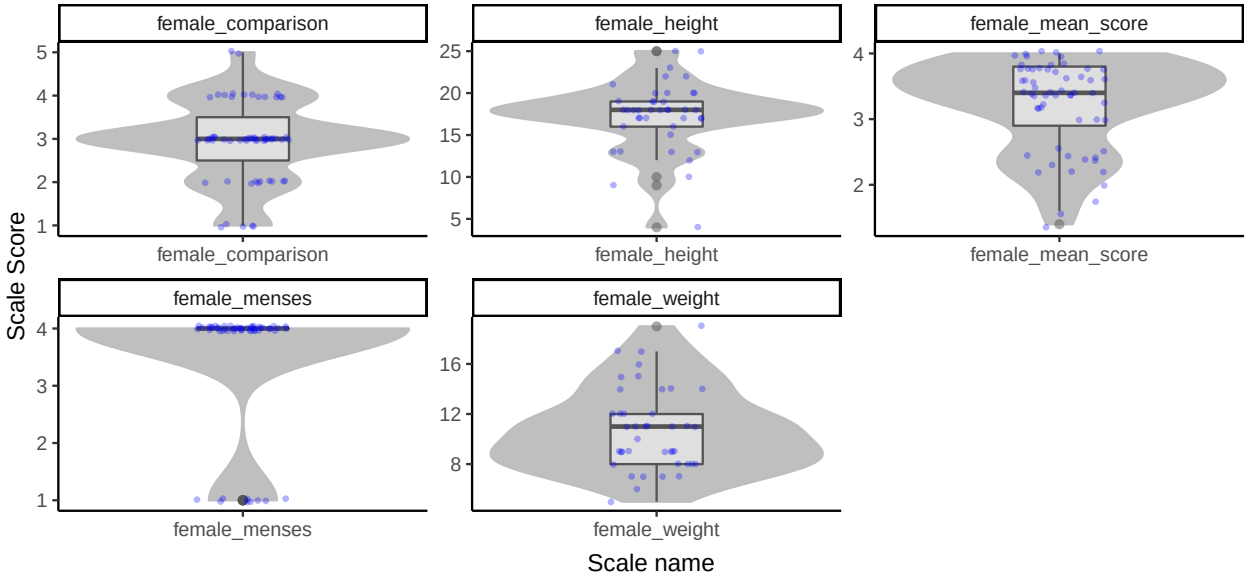



```
scorequaltrics::plot_scored_scale(filter(tds2_wave1_scored,
                                         SID %in% female_sids$SID,
                                         grepl('^female', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'score') +
    facet_wrap(~scored_scale, scales = 'free') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

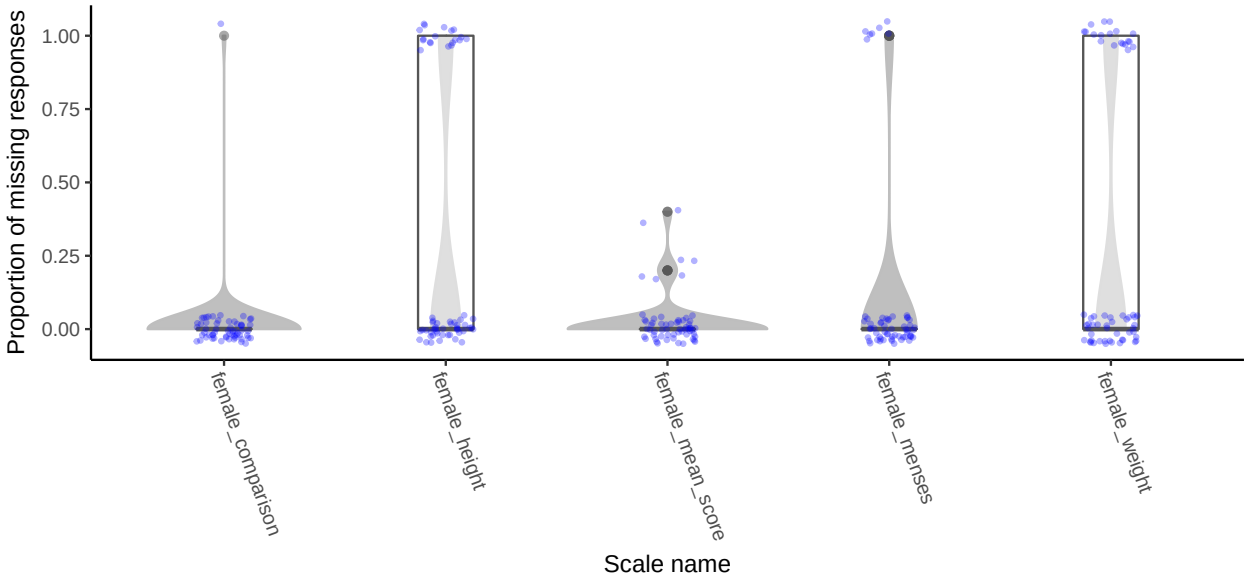
```
## Warning: Removed 46 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 46 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 46 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(filter(tds2_wave1_scored,
                                         SID %in% female_sids$SID,
                                         grepl('^female', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'p_missing')
```



PDSS

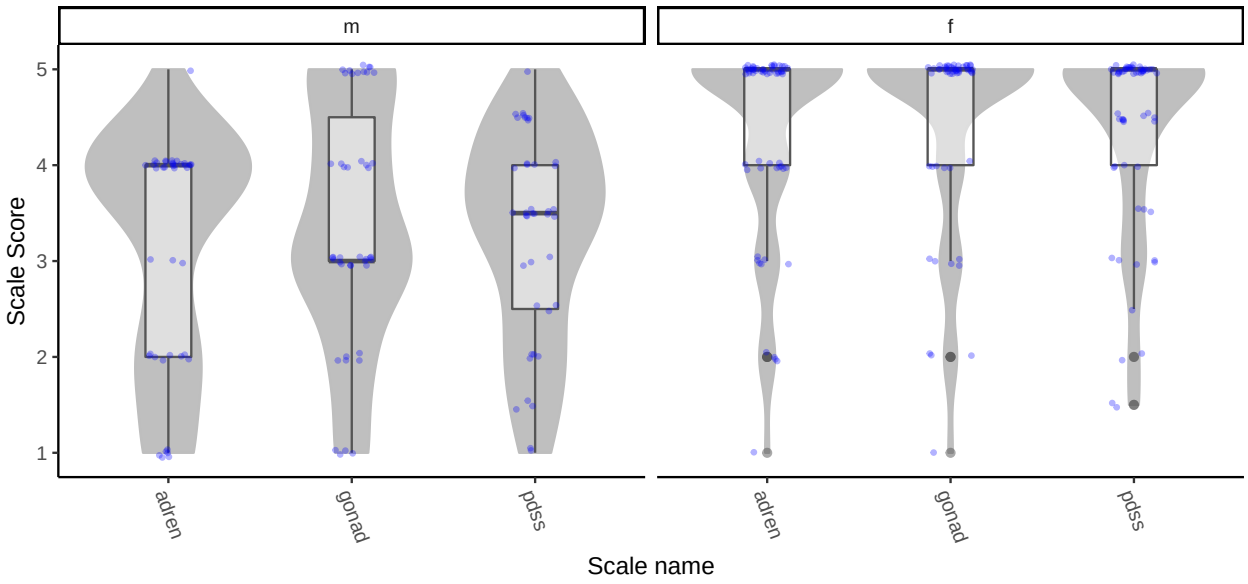
```
tds2_wave1_scored_pdss_gender <- tds2_wave1_scored_pdss %>%
    filter(scored_scale == 'gender') %>%
    spread(scored_scale, score) %>%
    select(SID, gender)
```

```
tds2_wave1_scored_pdss_gender_wide <- tds2_wave1_scored_pdss %>%
    filter(scored_scale != 'gender') %>%
    left_join(tds2_wave1_scored_pdss_gender)

scorequaltrics::plot_scored_scale(tds2_wave1_scored_pdss_gender_wide,
                                  scale_regx = '^PDSS$',
                                  type = 'score') +
    facet_wrap(~factor(gender,
                       levels = pdss_gender_code,
                       labels = names(pdss_gender_code)))
```

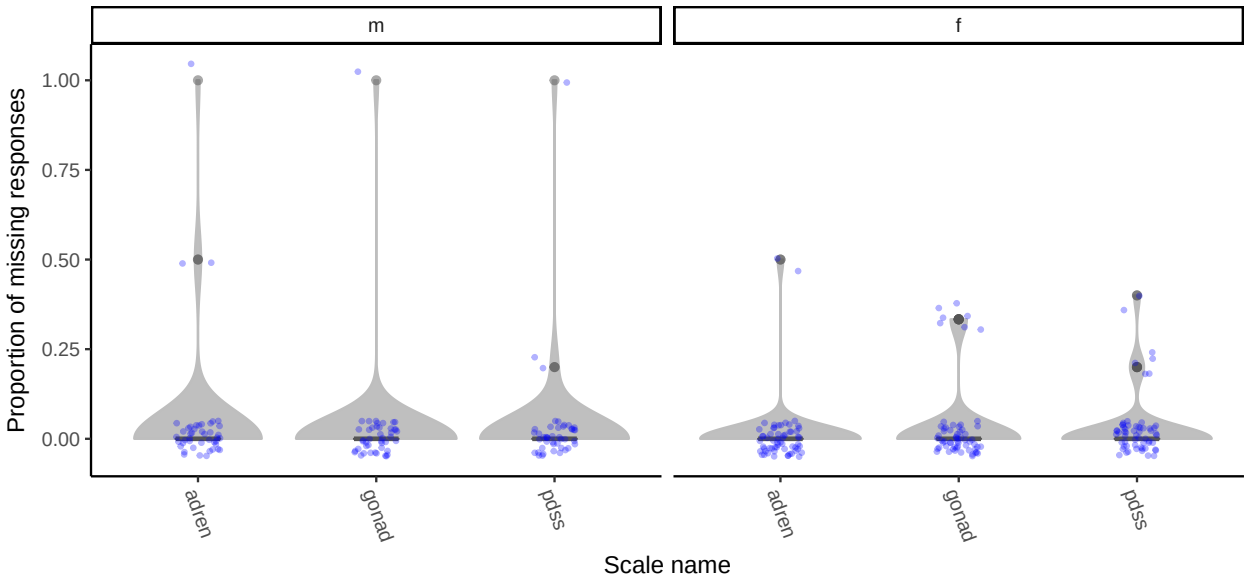
```
## Warning: Removed 14 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 14 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 14 rows containing missing values (geom_point).
```



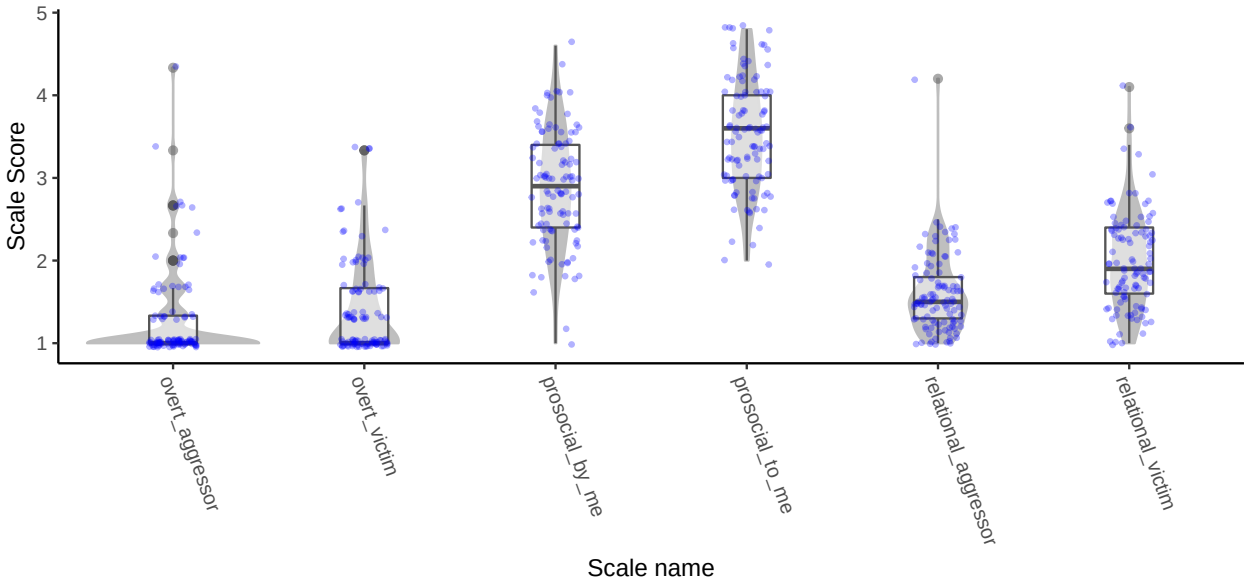

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored_pdss_gender_wide,
                                  scale_regx = '^PDSS$',
                                  type = 'p_missing') +
    facet_wrap(~factor(gender,
                       levels = pdss_gender_code,
                       labels = names(pdss_gender_code)))
```



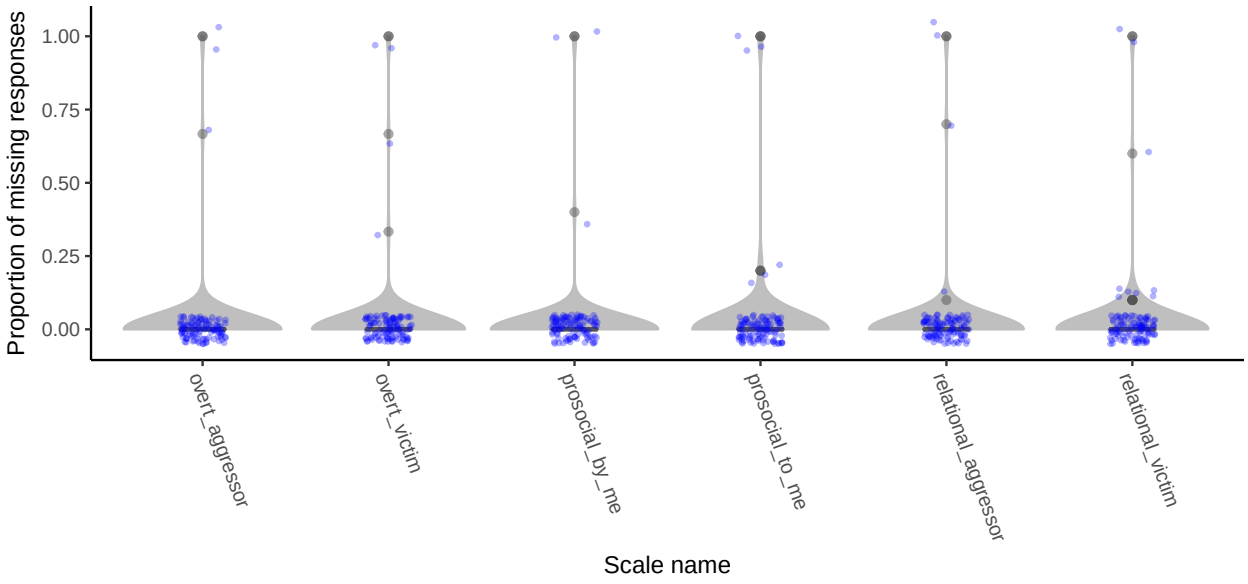
PEQ-R

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^PEQ-R$',
                                  type = 'score')
```

```
## Warning: Removed 13 rows containing non-finite values (stat_ydensity).
## Warning: Removed 13 rows containing non-finite values (stat_boxplot).
## Warning: Removed 13 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^PEQ-R$',
                                  type = 'p_missing')
```

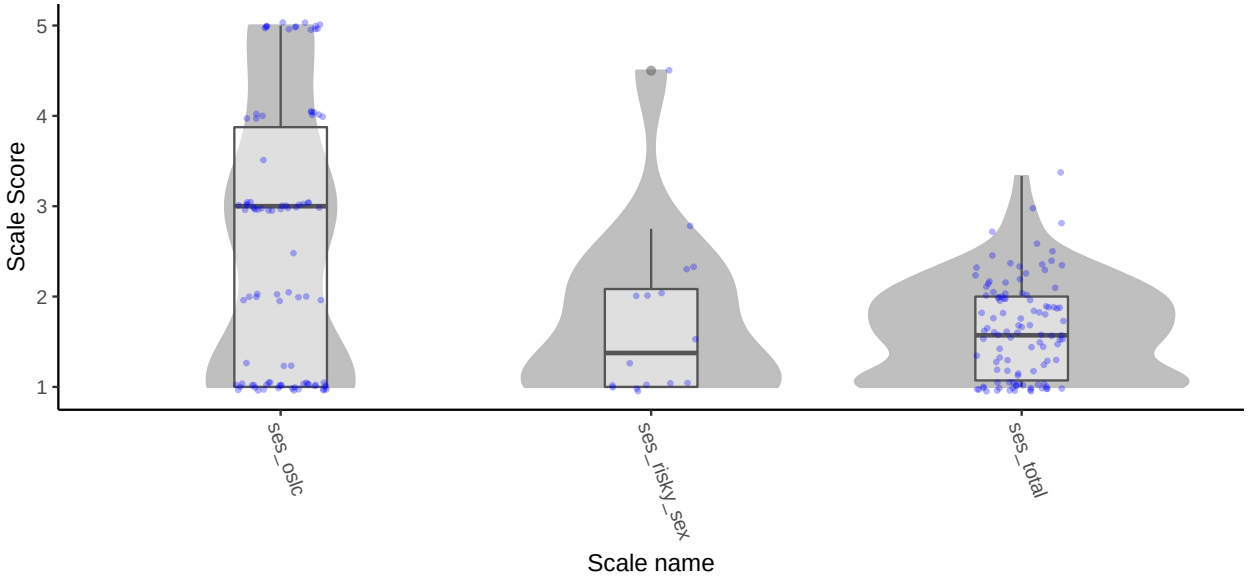


SES

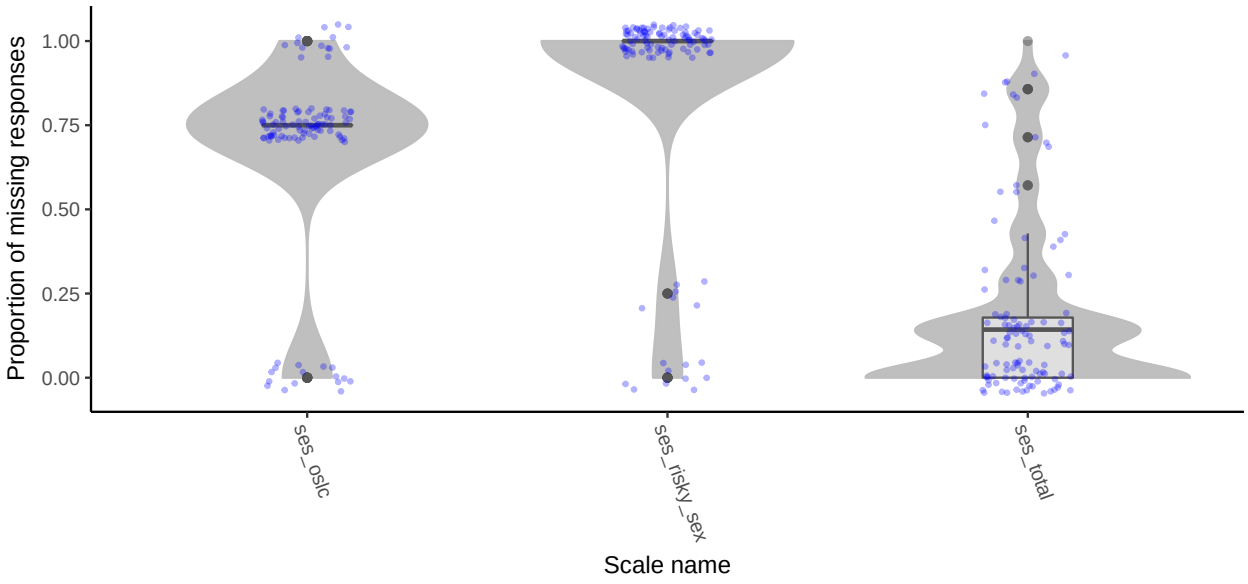
Needs a checkup

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^SES$',
                                  type = 'score')
```

```
## Warning: Removed 107 rows containing non-finite values (stat_ydensity).
## Warning: Removed 107 rows containing non-finite values (stat_boxplot).
## Warning: Removed 107 rows containing missing values (geom_point).
```

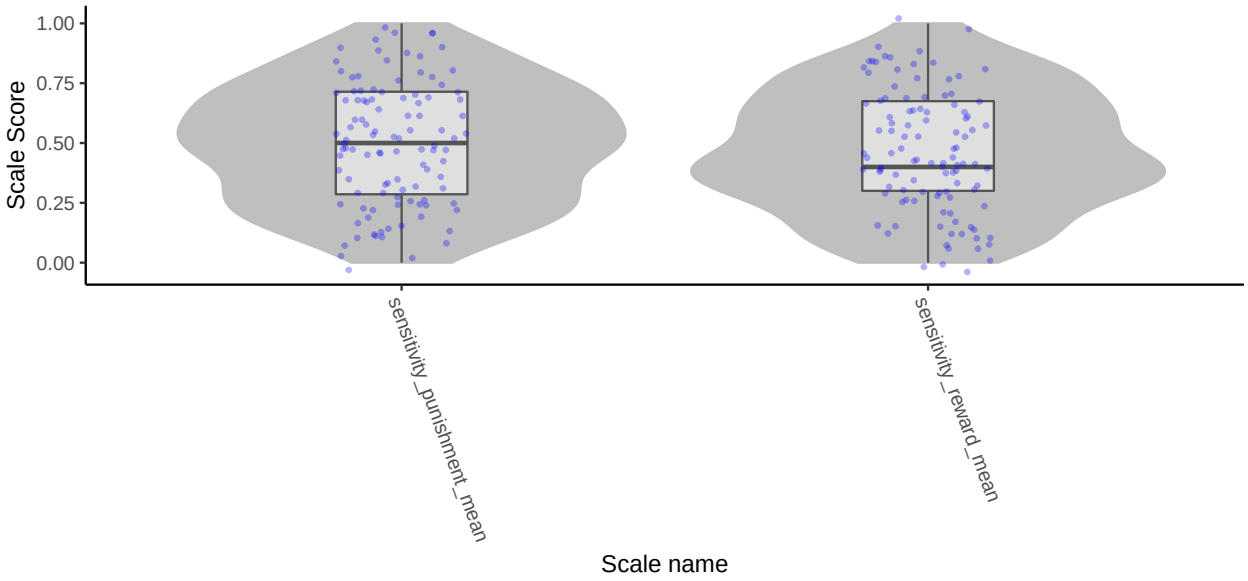



```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^SES$',
                                  type = 'p_missing')
```

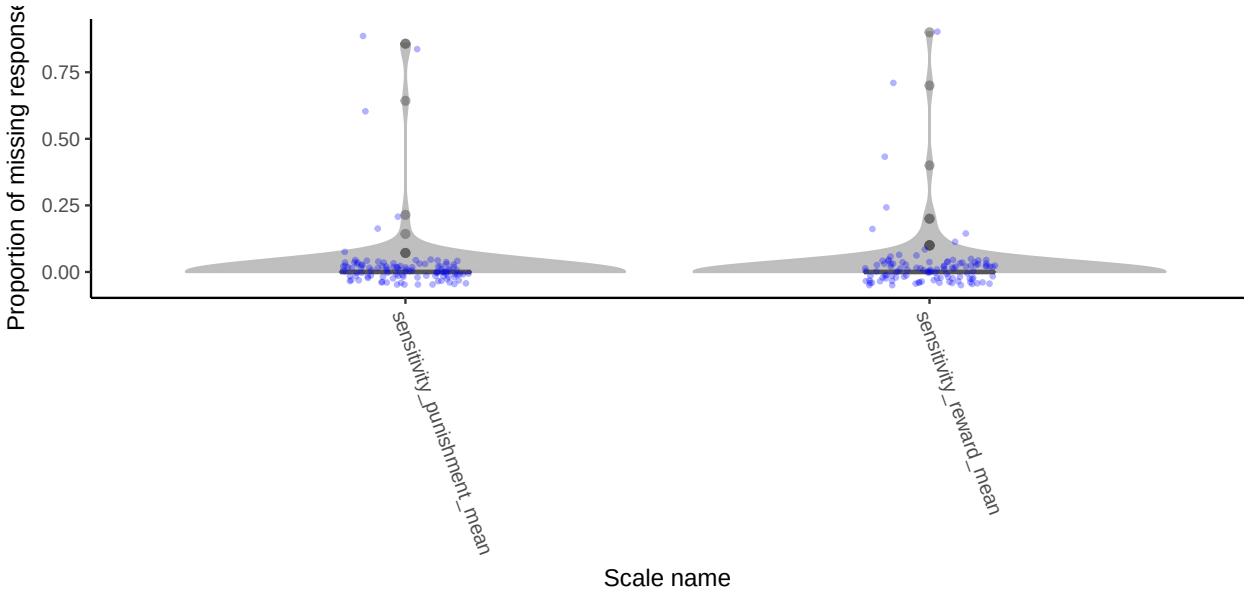


SPSRQ-S

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^SPSRQ-S$',
                                  type = 'score')
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^SPSRQ-S$',
                                  type = 'p_missing')
```

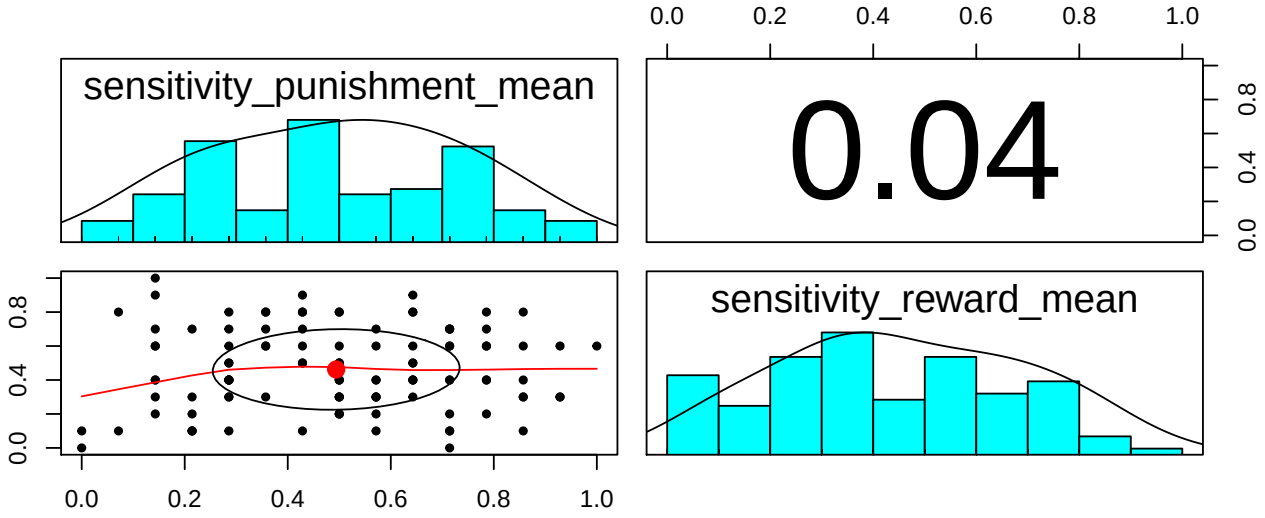



```
tds2w1_SPSRQSubric <- tds2_wave1_scoring_data_long %>%
    filter(scale_name == 'SPSRQ-S')

tds2_wave1_spsrqs_psych <- scorequaltrics::score_questionnaire(tds2_wave1_long_recoded_nodupes,
                                    tds2w1_SPSRQSubric,
                                    psych = T)
print(tds2_wave1_spsrqs_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       sensitivity_punishment_mean sensitivity_reward_mean
## alpha                        0.75                    0.67
##
## Standard errors of unstandardized Alpha:
##       sensitivity_punishment_mean sensitivity_reward_mean
## ASE                        0.044                   0.059
##
## Average item correlation:
##           sensitivity_punishment_mean sensitivity_reward_mean
## average.r                        0.18                    0.17
##
##  Guttman 6* reliability:
##          sensitivity_punishment_mean sensitivity_reward_mean
## Lambda.6                        0.82                    0.75
##
## Signal/Noise based upon av.r :
##              sensitivity_punishment_mean sensitivity_reward_mean
## Signal/Noise                         3.1                     2.1
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                             sensitivity_punishment_mean
## sensitivity_punishment_mean                       0.754
```

```
## sensitivity_reward_mean                           0.039
##                              sensitivity_reward_mean
## sensitivity_punishment_mean                   0.055
## sensitivity_reward_mean                       0.673
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

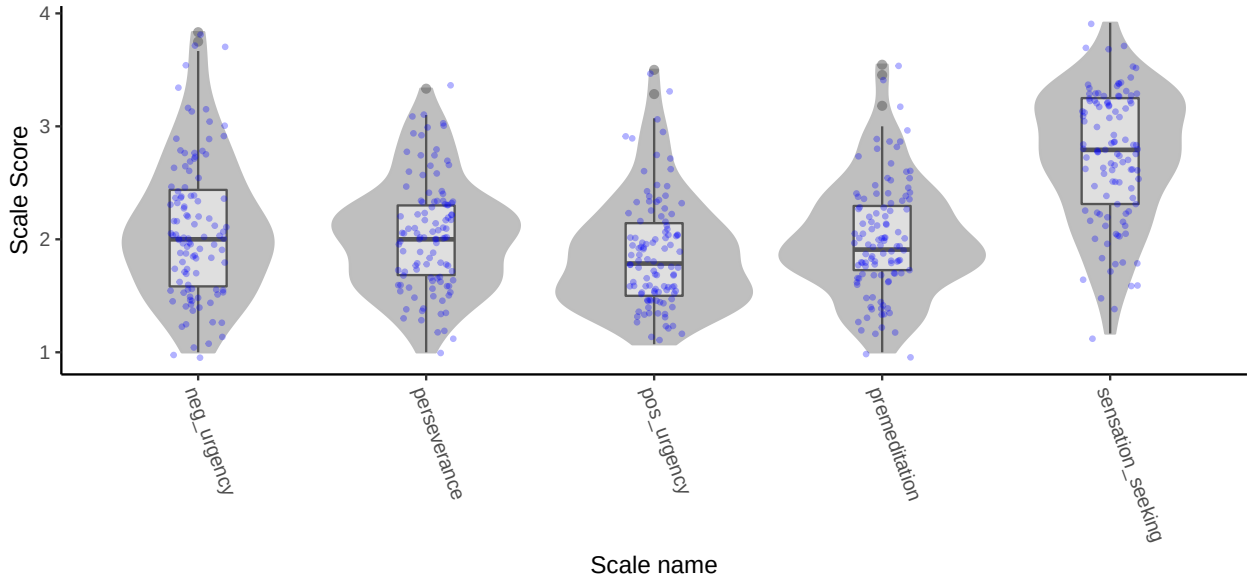
```
pairs.panels(tds2_wave1_spsrqs_psych$scores)
```



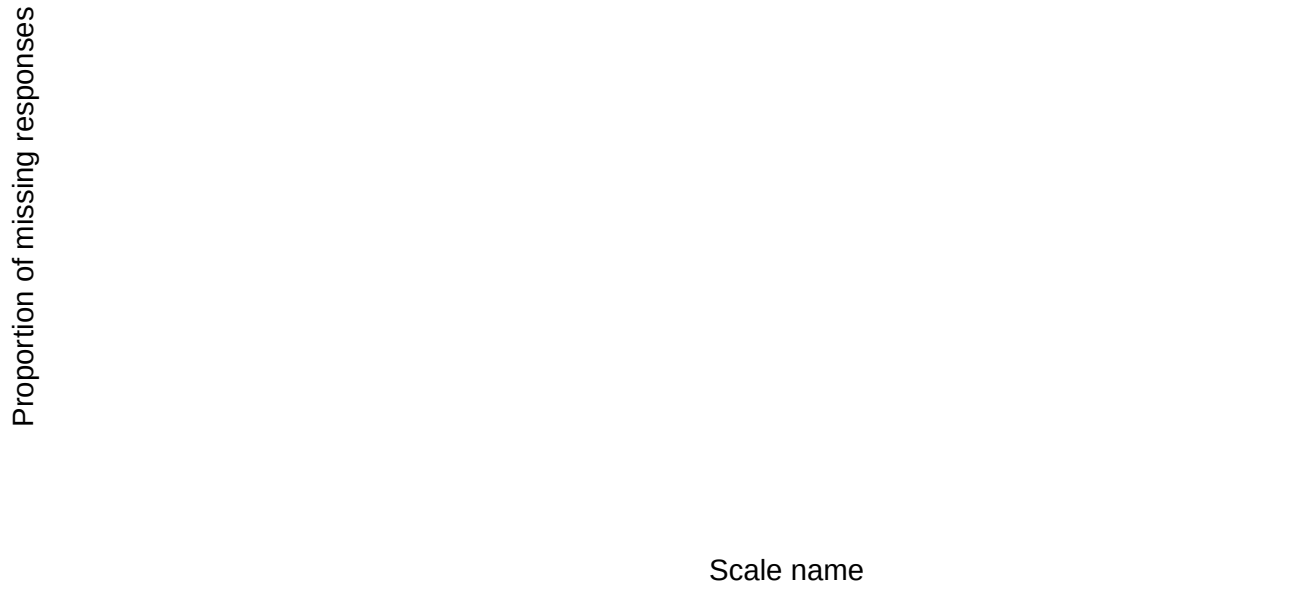
UPPS-P

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^UPPS-P$',
                                  type = 'score')
```

```
## Warning: Removed 1 rows containing non-finite values (stat_ydensity).
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^SUPPS-P$',
                                  type = 'p_missing')
```

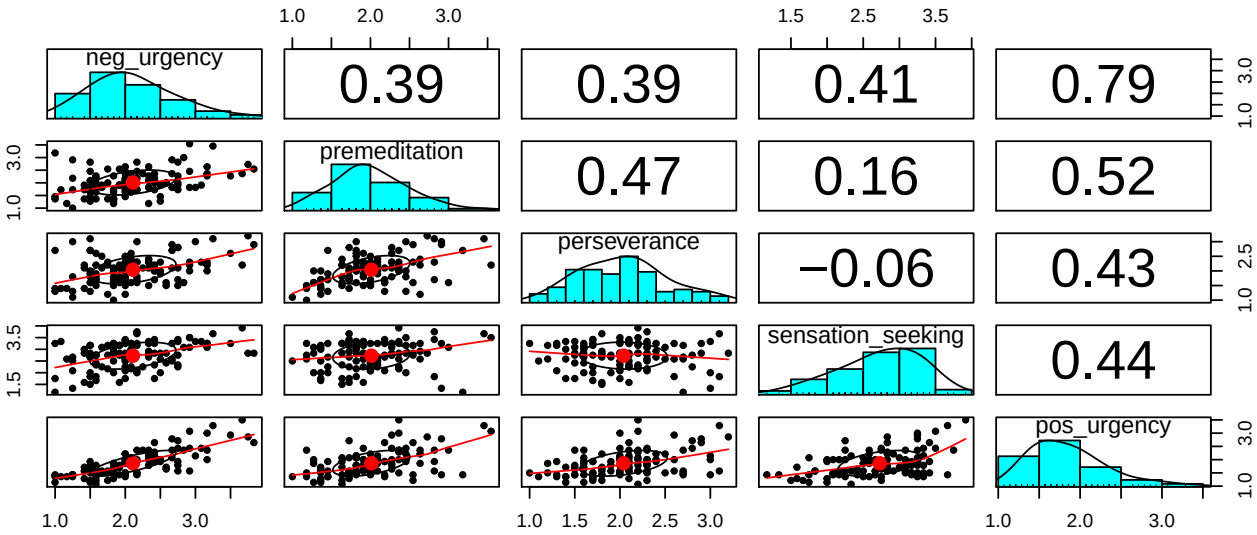



```
tds2w1_UPPSPRubric <- tds2_wave1_scoring_data_long %>%
    filter(scale_name == 'UPPS-P')

tds2_wave1_uppsp_psych <- scorequaltrics::score_questionnaire(tds2_wave1_long_recoded_nodupes,
                                  tds2w1_UPPSPRubric,
                                  psych = T)
print(tds2_wave1_uppsp_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       neg_urgency premeditation perseverance sensation_seeking pos_urgency
## alpha        0.89          0.83          0.8              0.82        0.87
```

```
##
## Standard errors of unstandardized Alpha:
##       neg_urgency premeditation perseverance sensation_seeking pos_urgency
## ASE         0.027         0.037        0.042             0.037       0.029
##
## Average item correlation:
##           neg_urgency premeditation perseverance sensation_seeking
## average.r        0.41           0.3         0.29              0.27
##           pos_urgency
## average.r        0.32
##
##  Guttman 6* reliability:
##          neg_urgency premeditation perseverance sensation_seeking
## Lambda.6        0.96          0.93         0.92              0.94
##          pos_urgency
## Lambda.6        0.95
##
## Signal/Noise based upon av.r :
##              neg_urgency premeditation perseverance sensation_seeking
## Signal/Noise         8.4           4.7            4               4.5
##              pos_urgency
## Signal/Noise         6.5
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                   neg_urgency premeditation perseverance sensation_seeking
## neg_urgency              0.89          0.45         0.46             0.480
## premeditation            0.39          0.83         0.58             0.191
## perseverance             0.39          0.47         0.80            -0.075
## sensation_seeking        0.41          0.16        -0.06             0.819
## pos_urgency              0.79          0.52         0.43             0.442
##                   pos_urgency
## neg_urgency              0.90
## premeditation            0.61
## perseverance             0.52
## sensation_seeking        0.52
## pos_urgency              0.87
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

```
pairs.panels(tds2_wave1_uppsp_psych$scores)
```



YRBS

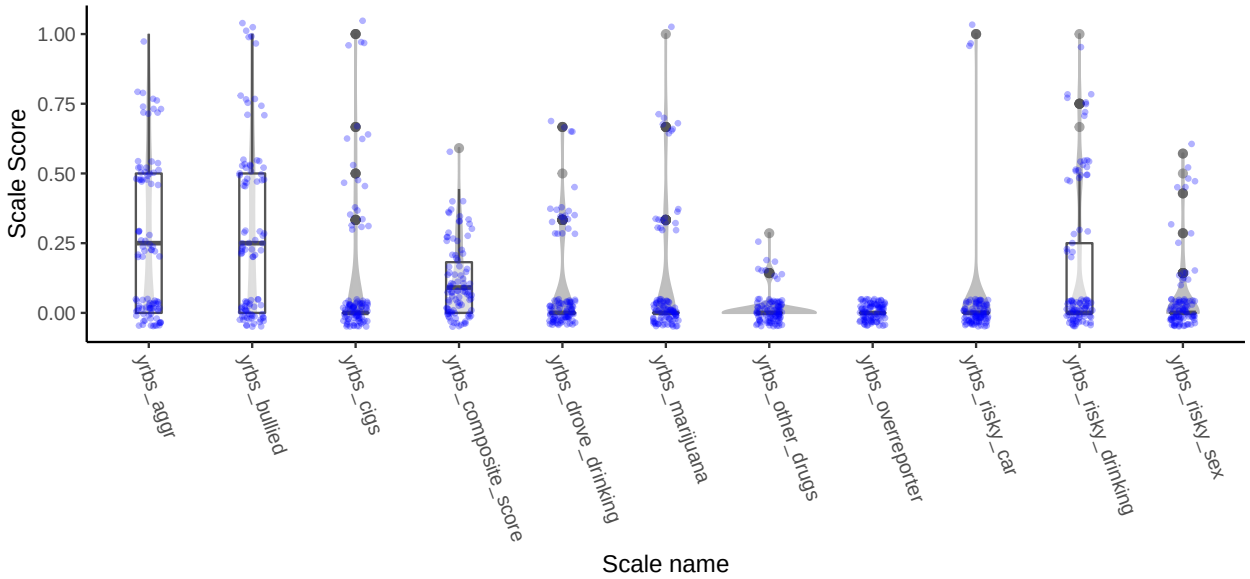
This has been scored according to the CDC manual. Items have been scored with either 1 or 0, with the mean calculated for each subscale. This gives you the proportion of endorsed items in that scale. To convert to a sum of endorsed items, simply multiply the scaled score by the number of total items (`n_missing` + `n_items`).

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^YRBS$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 360-70))
```

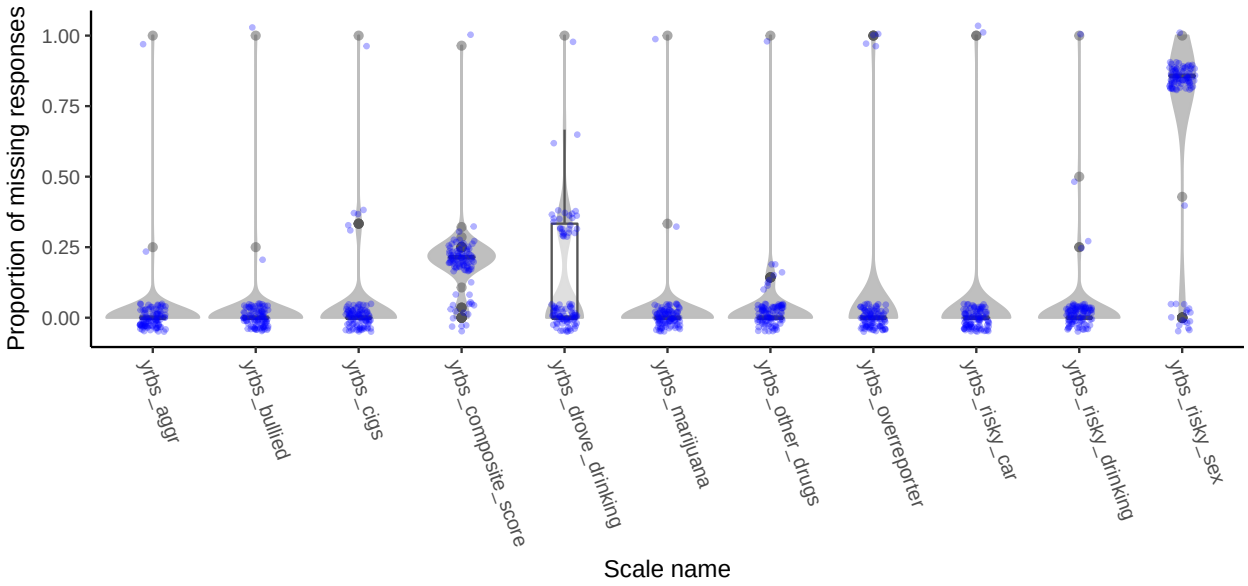
```
## Warning: Removed 14 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 14 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 14 rows containing missing values (geom_point).
```

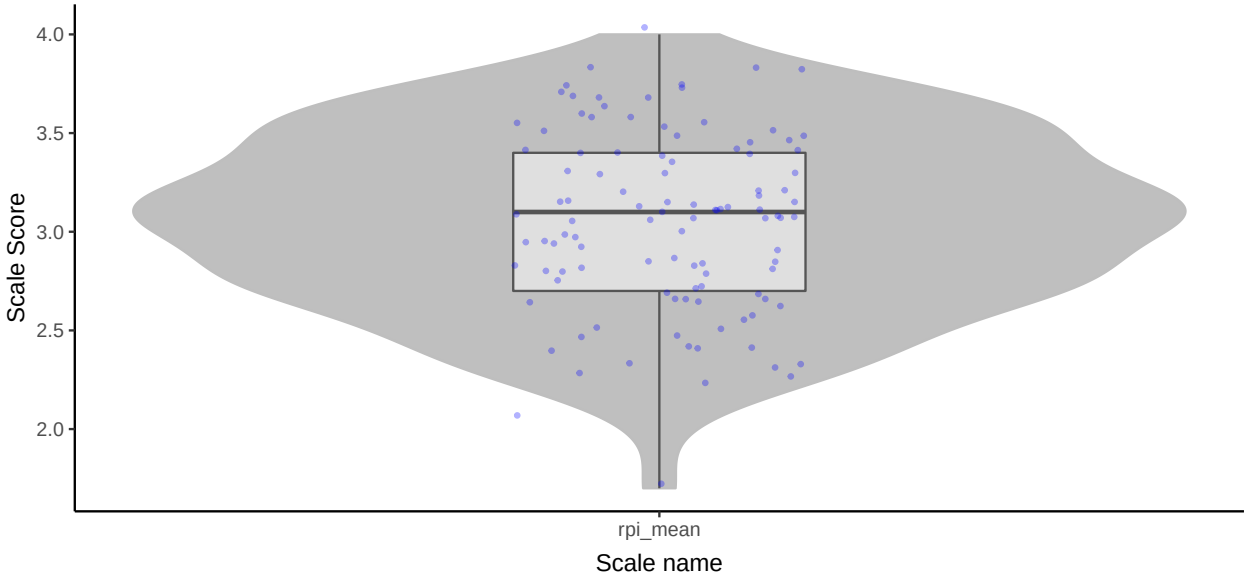


```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^YRBS$',
                                  type = 'p_missing')
```



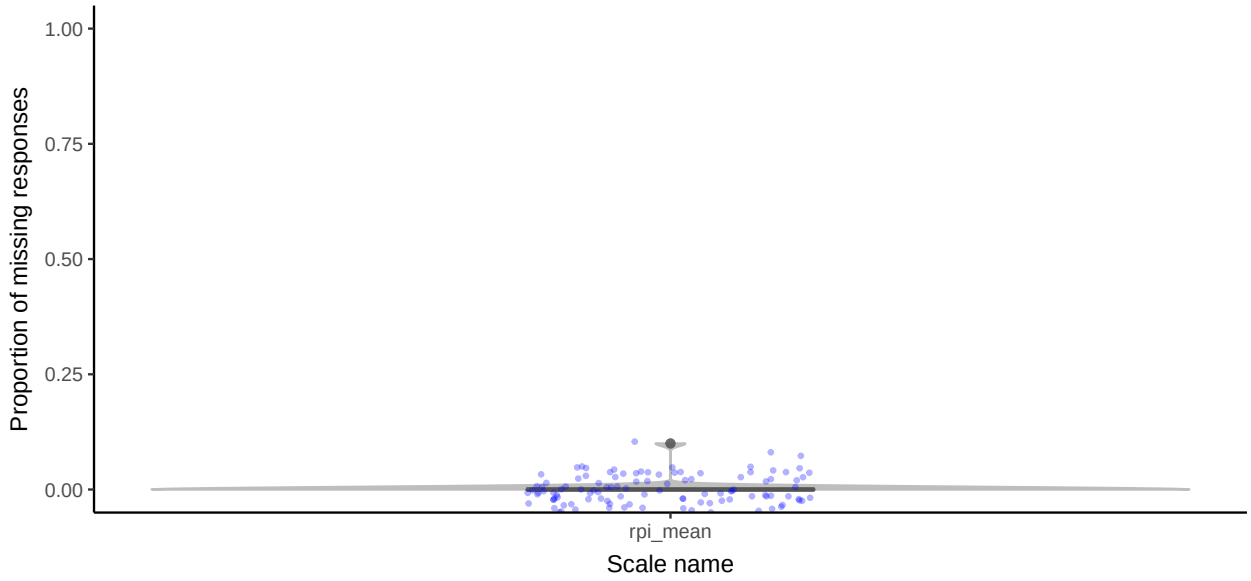
RPI

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^RPI_part2$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```




```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^RPI_part2$',
                                  type = 'p_missing') +
    coord_cartesian(y = c(0,1)) +
```

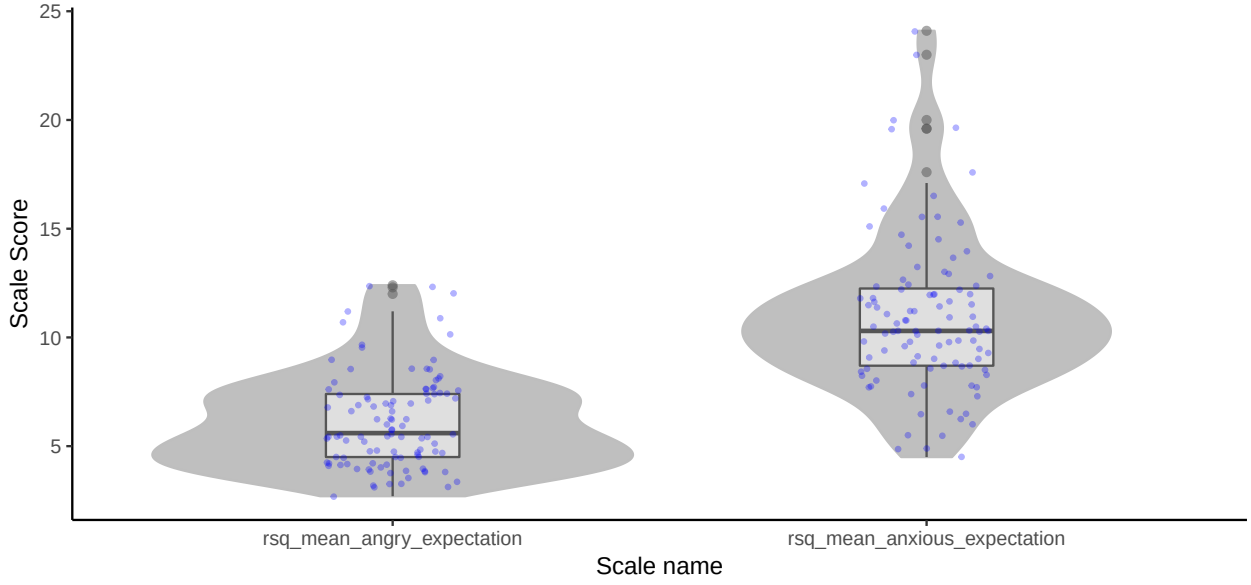
```
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```



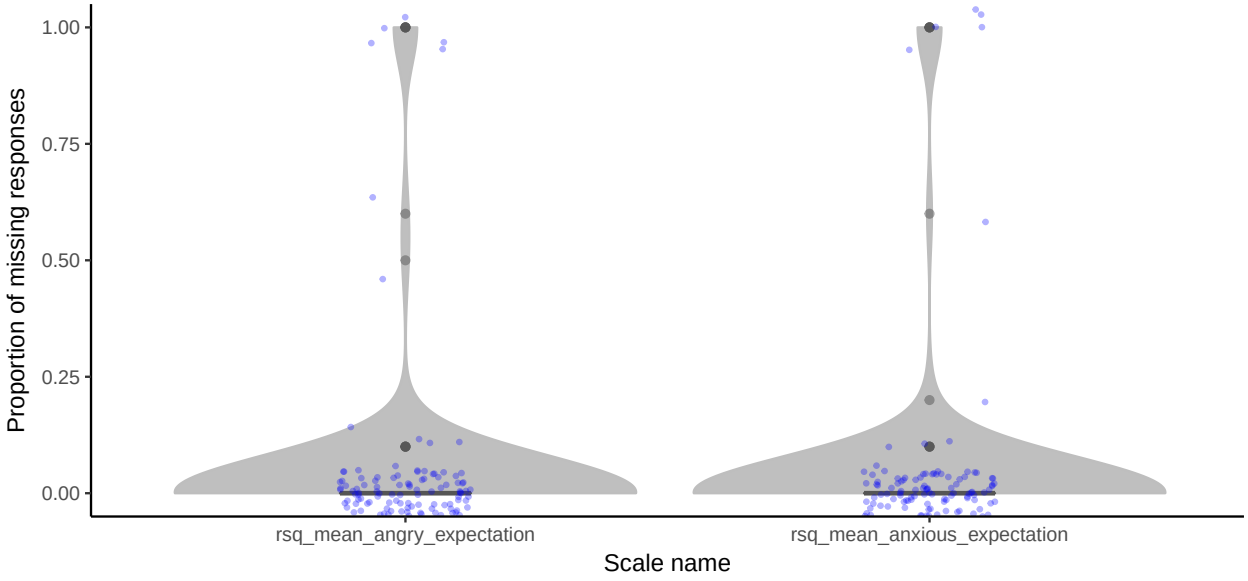
RSQ

```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^RSQ_part2$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

```
## Warning: Removed 10 rows containing non-finite values (stat_ydensity).
## Warning: Removed 10 rows containing non-finite values (stat_boxplot).
## Warning: Removed 10 rows containing missing values (geom_point).
```



```
scorequaltrics::plot_scored_scale(tds2_wave1_scored,
                                  scale_regx = '^RSQ_part2$',
                                  type = 'p_missing') +
    coord_cartesian(y = c(0,1)) +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```



Write data

We'll write out each data frame in wide format for ease of use in other contexts. We can use `lapply` on a list of unique scale names to write each one out using `widen_qualtrics_long` and `write.csv`. We treat the PDSS separately.

```
list_of_scales <- as.list(unique(tds2_wave1_scored$scale_name))
tds2_wave1_metadata <- list(wave = 1, questionnaires = 'TDS2')

throwaway <- lapply(list_of_scales,
                    function(aScale){
                        file_name <- paste0(TDS2_name, '-',
                                            wave1_name, '-',
                                            scorequaltrics::make_nice_scale_fname(aScale),
                                            '.csv')
                        scorequaltrics::write_widened_scored_scale(
                            tds2_wave1_scored,
                            scale_names = aScale,
                            dir_name = output_file_dir,
                            file_name = file_name,
                            metadata = tds2_wave1_metadata)
                    })


tds2_wave1_pdss_filename <-  paste0(TDS2_name, '-',
                                    wave1_name, '-',
                                    'PDSS.csv')
```

```
scorequaltrics::write_widened_scored_scale(
    tds2_wave1_scored_pdss,
    scale_names = 'PDSS',
    dir_name = output_file_dir,
    file_name = tds2_wave1_pdss_filename,
    metadata = tds2_wave1_metadata)
```

TDS 2 Wave 2

Get the data and clean it

Get data, rubrics, recode, and clean. Note that for wave 2 (aka, session 3), we need to separate out parent from child questionnaires, since the PMQ items are identical.

```
#Get surveys
tds2_wave2_surveys <- rawSurveysTDS %>%
    filter(grepl('TDS2 Session 3 - (Child|Parent)$', SurveyName))
knitr::kable(select(tds2_wave2_surveys, SurveyName))
```

SurveyName
TDS2 Session 3 - Parent
TDS2 Session 3 - Child

```
#Get data from those surveys
tds2_wave2_long <- scorequaltrics::get_survey_data(tds2_wave2_surveys,
                                                   credentials,
                                                   pid_col = pid_column_name)

#Get recoding rubrics
tds2_wave2_recoding_rubrics <- data.frame(file = dir(file.path(tds2_wave2_rubric_dir),
                                                     pattern = '.*response_recoding.*.csv',
                                                     full.names = TRUE))
tds2_wave2_recoding_data_long <- scorequaltrics::get_rubrics(tds2_wave2_recoding_rubrics,
                                                             type = 'recoding')

#Recode data
tds2_wave2_long_recoded <- scorequaltrics::recode_responses(tds2_wave2_long,
                                                            tds2_wave2_recoding_data_long)

#Code -99 as NA
tds2_wave2_long_recoded <- tds2_wave2_long_recoded %>%
    mutate(value = ifelse(grepl('SES', item) & as.numeric(value) < 0,
                          NA,
                          value))
```

```
## Warning in ifelse(grepl("SES", item) & as.numeric(value) < 0, NA, value):
## NAs introduced by coercion
```

```
#Get scoring rubrics
tds2_wave2_scoring_rubrics <- data.frame(file = dir(file.path(tds2_wave2_rubric_dir),
                                                    pattern = '.*scoring_rubric.*.csv',
                                                    full.names = TRUE))
```

```
tds2_wave2_scoring_data_long <- scorequaltrics::get_rubrics(tds2_wave2_scoring_rubrics,
                                                            type = 'scoring')

#Split scoring into parent and child
tds2_wave2_scoring_data_long_p <- tds2_wave2_scoring_data_long %>%
    filter(grepl('[pP]arent', data_file_name))
tds2_wave2_scoring_data_long_c <- tds2_wave2_scoring_data_long %>%
    filter(!grepl('[pP]arent', data_file_name))

#Split data into parent and child
tds2_wave2_long_recoded_p <- tds2_wave2_long_recoded %>%
    filter(grepl('[pP]arent', survey_name))
tds2_wave2_long_recoded_c <- tds2_wave2_long_recoded %>%
    filter(!grepl('[pP]arent', survey_name))

#Clean and de-dupe parent and child data
tds2_wave2_long_recoded_p_nodupes <- tds2_wave2_long_recoded_p %>%
    get_items_in_rubric(tds2_wave2_scoring_data_long_p) %>%
    filter(grepl('[14]\\d\\d', SID)) %>%
    scorequaltrics::clean_dupes(pid_col = 'SID')
tds2_wave2_long_recoded_c_nodupes <- tds2_wave2_long_recoded_c %>%
    get_items_in_rubric(tds2_wave2_scoring_data_long_c) %>%
    filter(grepl('[14]\\d\\d', SID)) %>%
    scorequaltrics::clean_dupes(pid_col = 'SID')
```

After cleaning duplicates, it's possible that there were some responses that have conflicting answers. We can look at those now.

Now we can see which items had conflicting responses on duplicate questionnaires.

```
#Check that dropped values weren't ambiguous
tds2_wave2_long_recoded_p_nodupes %>%
    filter(dropped) %>%
    group_by(SID, item) %>%
    filter(!all(length(unlist(old.value)) < 1)) %>%
    mutate(old.value = paste(old.value, collaps = ' ')) %>%
    knitr::kable(caption = "Parent questionnaire dupes")
```

Table 5: Parent questionnaire dupes

SID	item	value	survey_name	old.value	dropped
168	VIII_100	NA	TDS2 Session 3 - Parent	c(1, 0)	TRUE
168	VIII_101	NA	TDS2 Session 3 - Parent	c(0, 2)	TRUE
168	VIII_104	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE
168	VIII_105	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE
168	VIII_12	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE
168	VIII_3	NA	TDS2 Session 3 - Parent	c(1, 0)	TRUE
168	VIII_32	NA	TDS2 Session 3 - Parent	c(1, 2)	TRUE
168	VIII_42	NA	TDS2 Session 3 - Parent	c(0, 2)	TRUE
168	VIII_45	NA	TDS2 Session 3 - Parent	c(0, 2)	TRUE
168	VIII_46	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE
168	VIII_49	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE
168	VIII_50	NA	TDS2 Session 3 - Parent	c(0, 2)	TRUE
168	VIII_51	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE

SID	item	value	survey_name	old.value	dropped
168	VIII_52	NA	TDS2 Session 3 - Parent	c(1, 0)	TRUE
168	VIII_54	NA	TDS2 Session 3 - Parent	c(0, 2)	TRUE
168	VIII_56_b	NA	TDS2 Session 3 - Parent	c(2, 0)	TRUE
168	VIII_56_c	NA	TDS2 Session 3 - Parent	c(1, 2)	TRUE
168	VIII_56_d	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE
168	VIII_56_e	NA	TDS2 Session 3 - Parent	c(1, 0)	TRUE
168	VIII_56_f	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE
168	VIII_59	NA	TDS2 Session 3 - Parent	c(0, 2)	TRUE
168	VIII_64	NA	TDS2 Session 3 - Parent	c(0, 2)	TRUE
168	VIII_70	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE
168	VIII_87	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE
168	VIII_88	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE
168	VIII_89	NA	TDS2 Session 3 - Parent	c(1, 0)	TRUE
168	VIII_9	NA	TDS2 Session 3 - Parent	c(1, 2)	TRUE
168	VIII_91	NA	TDS2 Session 3 - Parent	c(0, 1)	TRUE

```
tds2_wave2_long_recoded_c_nodupes %>%
    filter(dropped) %>%
    group_by(SID, item) %>%
    filter(!all(length(unlist(old.value)) < 1)) %>%
    mutate(old.value = paste(old.value, collaps = ' ')) %>%
    knitr::kable(caption = "Child questionnaire dupes")
```

Table: Child questionnaire dupes

SID item value survey_name old.value dropped —- —— —— ———— ———- ———

Scoring

```
tds2_wave2_scored_p <- scorequaltrics::score_step_one_and_two(tds2_wave2_long_recoded_p_nodupes,
                                                              tds2_wave2_scoring_data_long_p)
tds2_wave2_scored_c <- scorequaltrics::score_step_one_and_two(tds2_wave2_long_recoded_c_nodupes,
                                                              tds2_wave2_scoring_data_long_c)
tds2_wave2_scored_pdss <- scorequaltrics::score_pdss(tds2_wave2_long_recoded_c_nodupes,
                          gender_mix = pdss_gender_mix,
                          gendercode = pdss_gender_code)
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
#PSQI requires a bit more finesse
tds2_wave2_scored_psqi <- scorequaltrics::score_psqi(
    filter(tds2_wave2_long_recoded_c,
           grepl('PSQI', item),
           grepl('[14]\\d\\d', SID)),
    pid_col = 'SID') %>%
    mutate(score = as.character(score))
```

```
## Warning in scorequaltrics:::score_psqi_2(psqi_data$PSQI_2_1_TEXT): NAs
## introduced by coercion
```

```
## Warning: All formats failed to parse. No formats found.
```

```
## Warning: All formats failed to parse. No formats found.
```

```
tds2_wave2_scored_c <- bind_rows(tds2_wave2_scored_c, tds2_wave2_scored_psqi)
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
tds2_wave2_scored_p %>%
    ungroup() %>%
    distinct(scale_name) %>%
    knitr::kable()
```

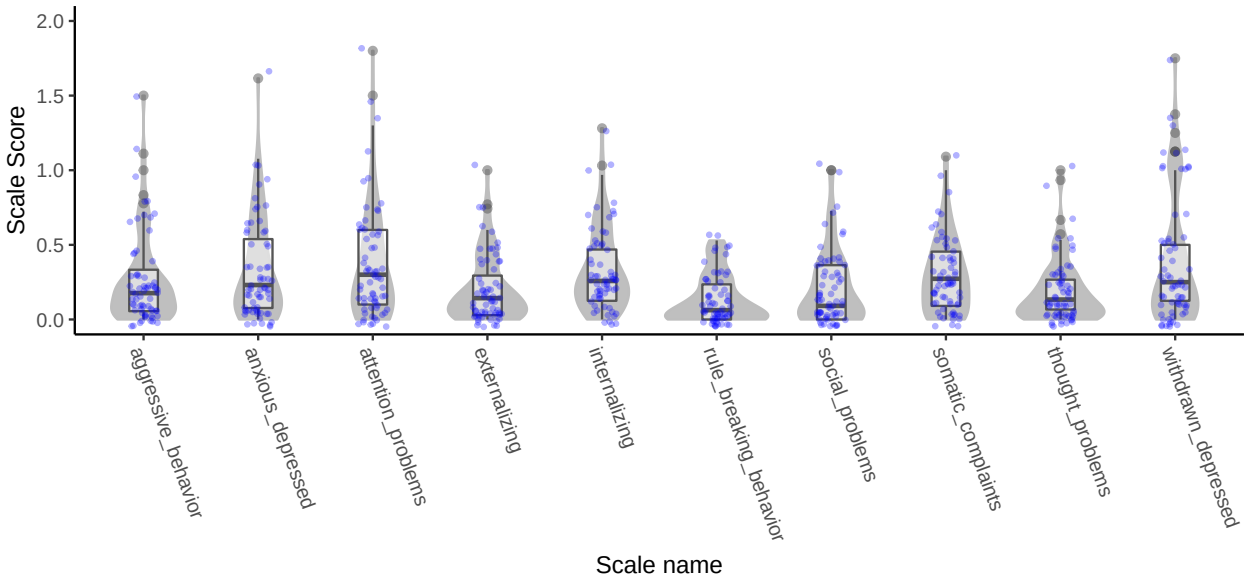
scale_name
CBCL
PMQ_Parent

```
tds2_wave2_scored_c %>%
    ungroup() %>%
    distinct(scale_name) %>%
    knitr::kable()
```

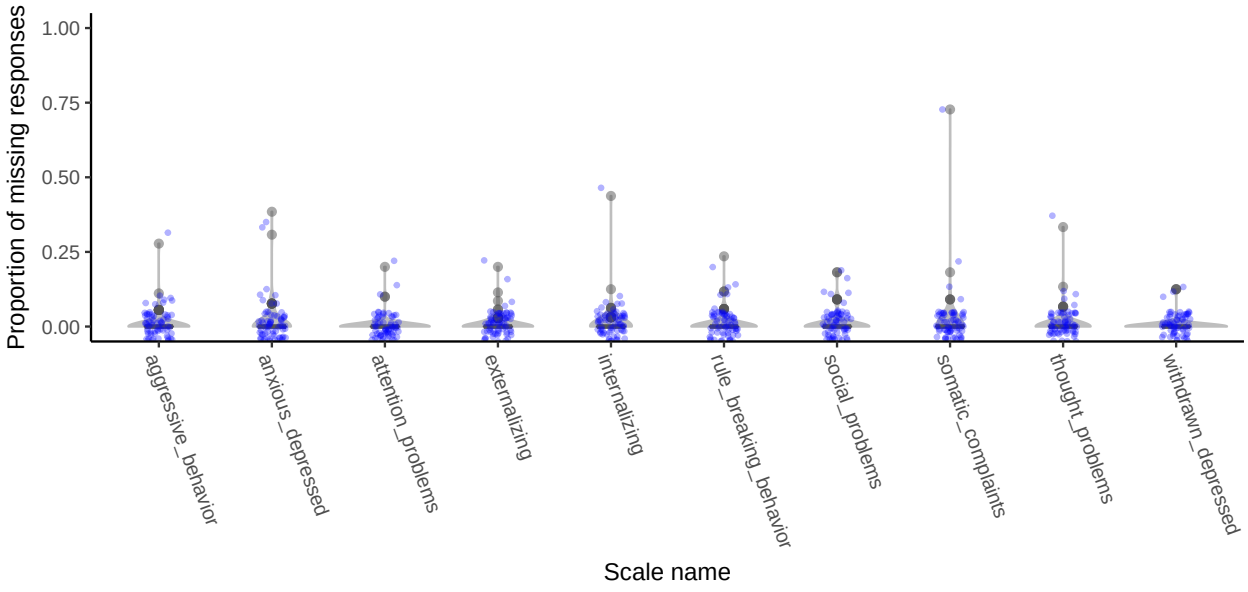
scale_name
BFI-10
CARE-R Social
PAL-2
PDS
PMQ_Child
SES
YRBS
RPI_part2
PSQI

CBCL

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_p,
                                  scale_regx = '^CBCL$',
                                  type = 'score')+
    coord_cartesian(y = c(0,2))
```



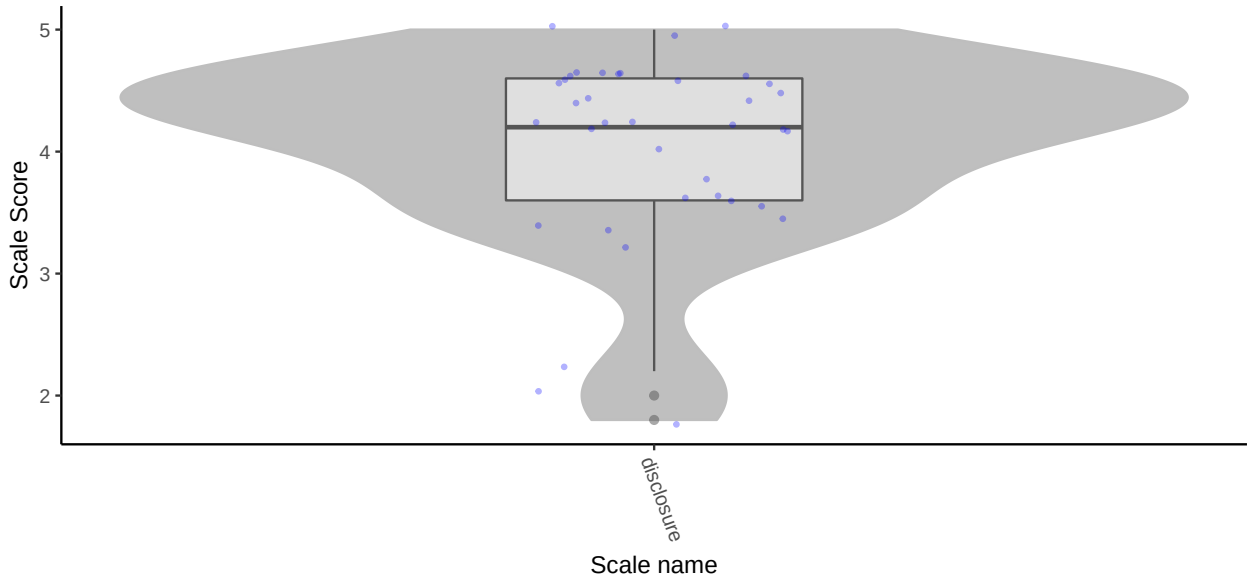

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_p,
                                  scale_regx = '^CBCL$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```



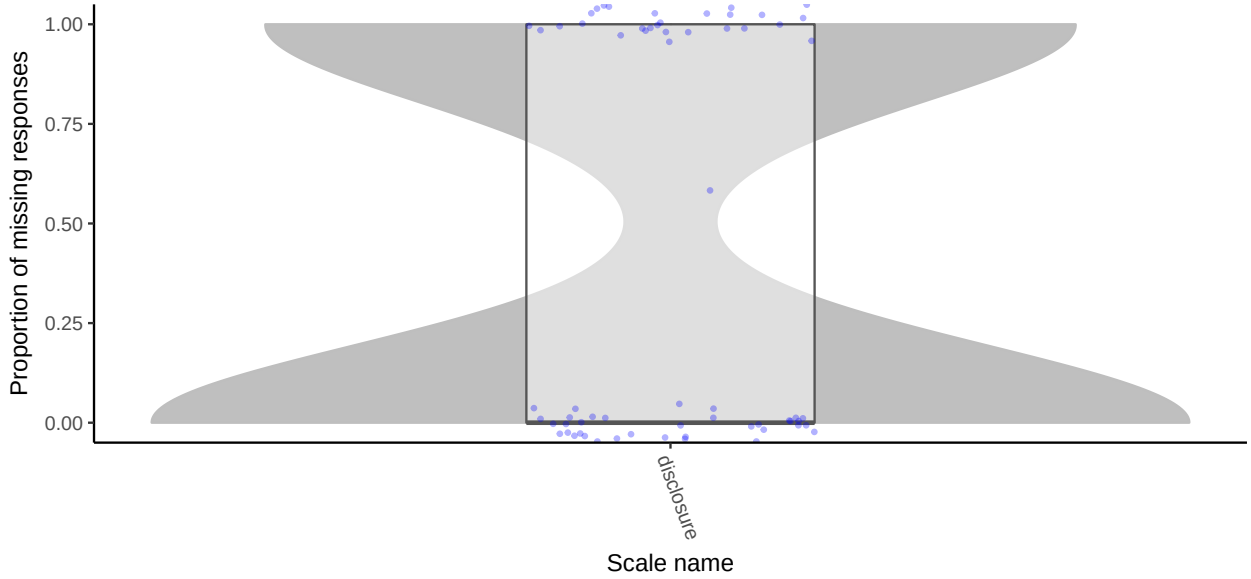
PMQ Parent

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_p,
                                  scale_regx = '^PMQ_Parent$',
                                  type = 'score')
```

```
## Warning: Removed 28 rows containing non-finite values (stat_ydensity).
## Warning: Removed 28 rows containing non-finite values (stat_boxplot).
## Warning: Removed 28 rows containing missing values (geom_point).
```



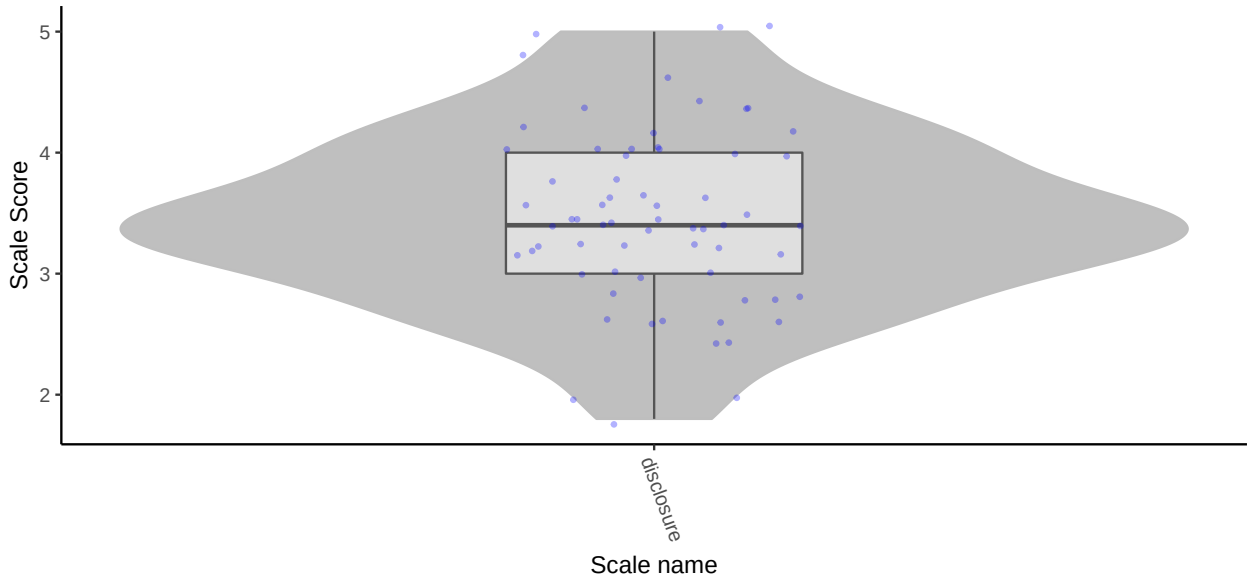

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_p,
                                  scale_regx = '^PMQ_Parent$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```



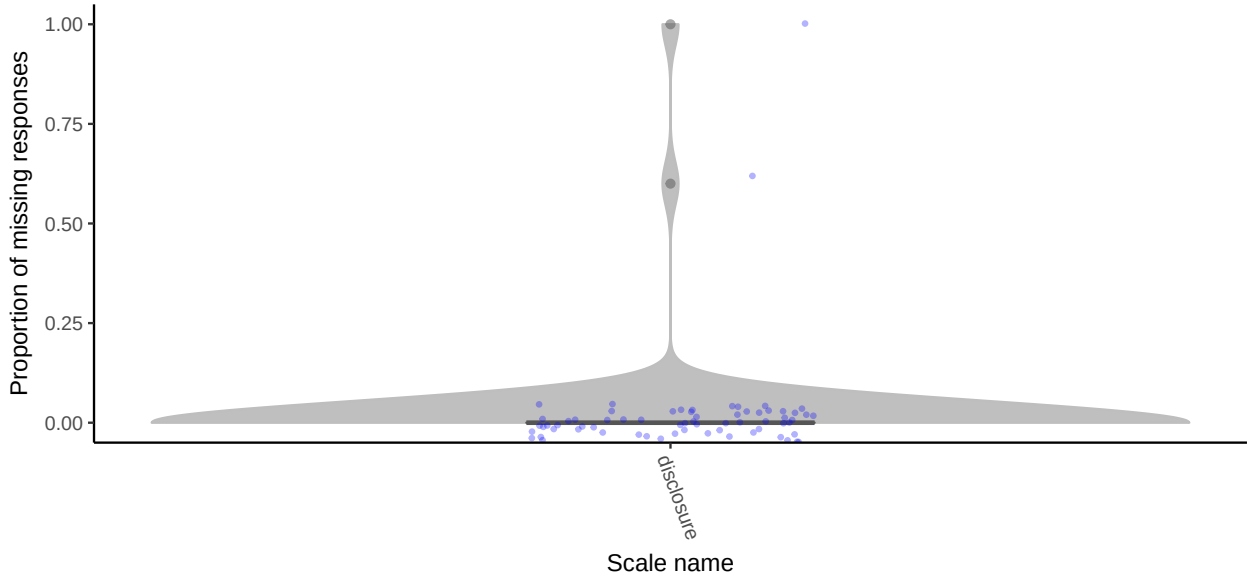
PMQ Child

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^PMQ_Child$',
                                  type = 'score')
```

```
## Warning: Removed 1 rows containing non-finite values (stat_ydensity).
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
## Warning: Removed 1 rows containing missing values (geom_point).
```



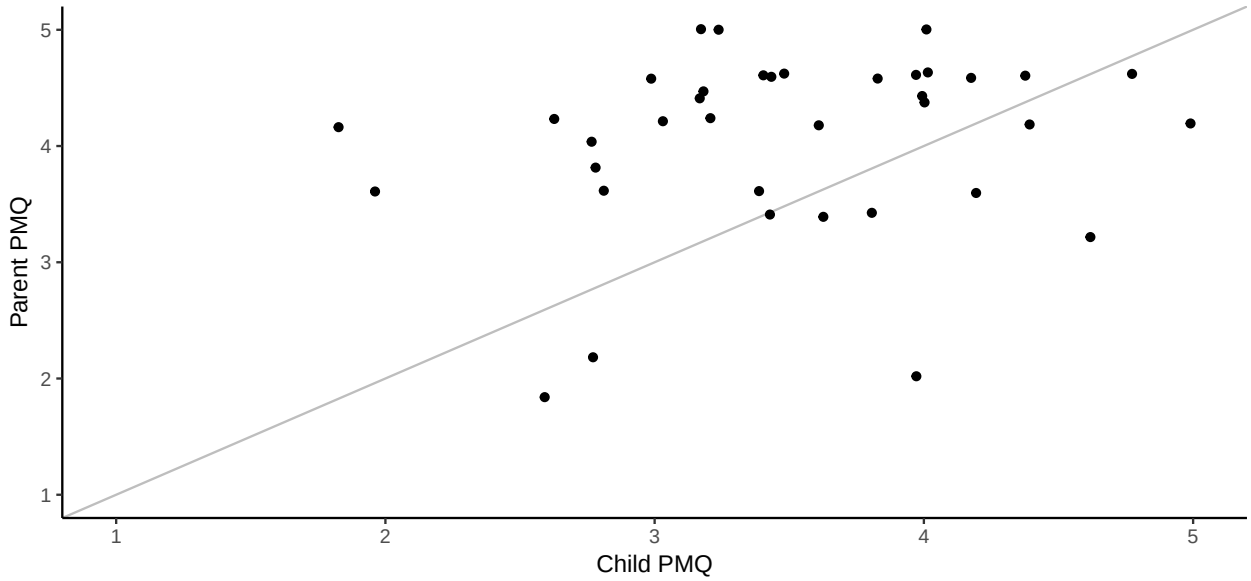

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^PMQ_Child$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```




```
tds2_wave2_disclosure_c <- tds2_wave2_scored_c %>%
    ungroup() %>%
    filter(scored_scale == 'disclosure') %>%
    select(scored_scale, SID, score) %>%
    spread(scored_scale, score) %>%
    rename('Child PMQ' = disclosure)

tds2_wave2_disclosure_p <- tds2_wave2_scored_p %>%
    ungroup() %>%
    filter(scored_scale == 'disclosure') %>%
    select(scored_scale, SID, score) %>%
    spread(scored_scale, score) %>%
```

```
    rename('Parent PMQ' = disclosure)

left_join(tds2_wave2_disclosure_c,
          tds2_wave2_disclosure_p) %>%
    mutate_at(vars(`Child PMQ`, `Parent PMQ`), as.numeric) %>%
    ggplot(aes(x = `Child PMQ`, y = `Parent PMQ`)) +
    geom_abline(intercept = 0, slope = 1, color = 'gray') +
    geom_point(position = position_jitter(), alpha = 1) +
    coord_cartesian(x = c(1, 5), y = c(1, 5)) +
    theme_classic()
```

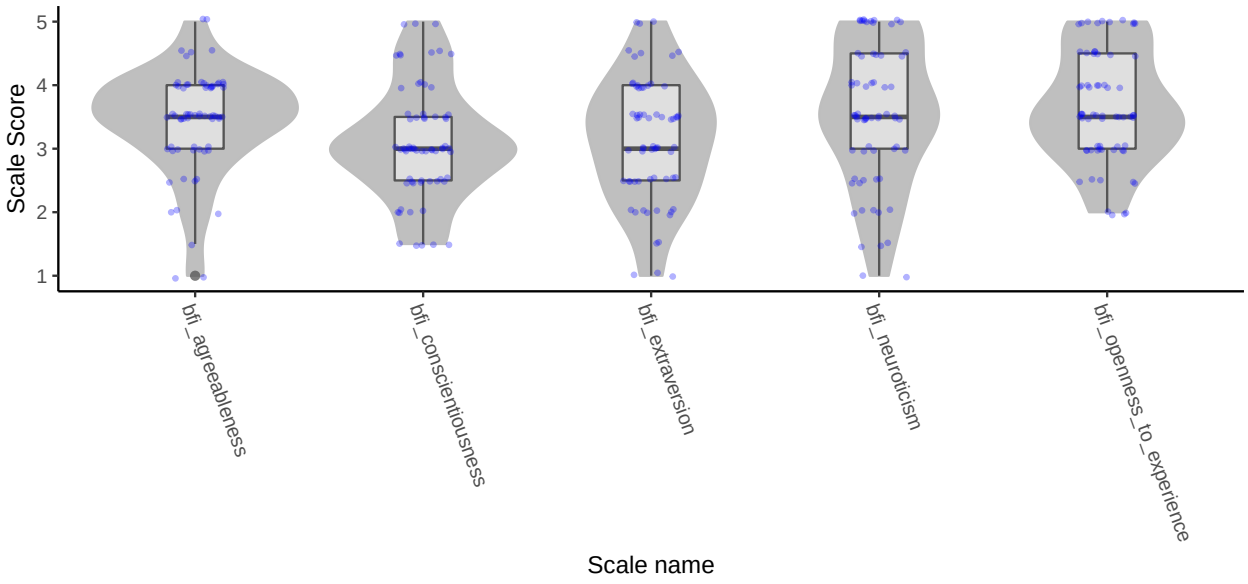
```
## Warning: Removed 30 rows containing missing values (geom_point).
```



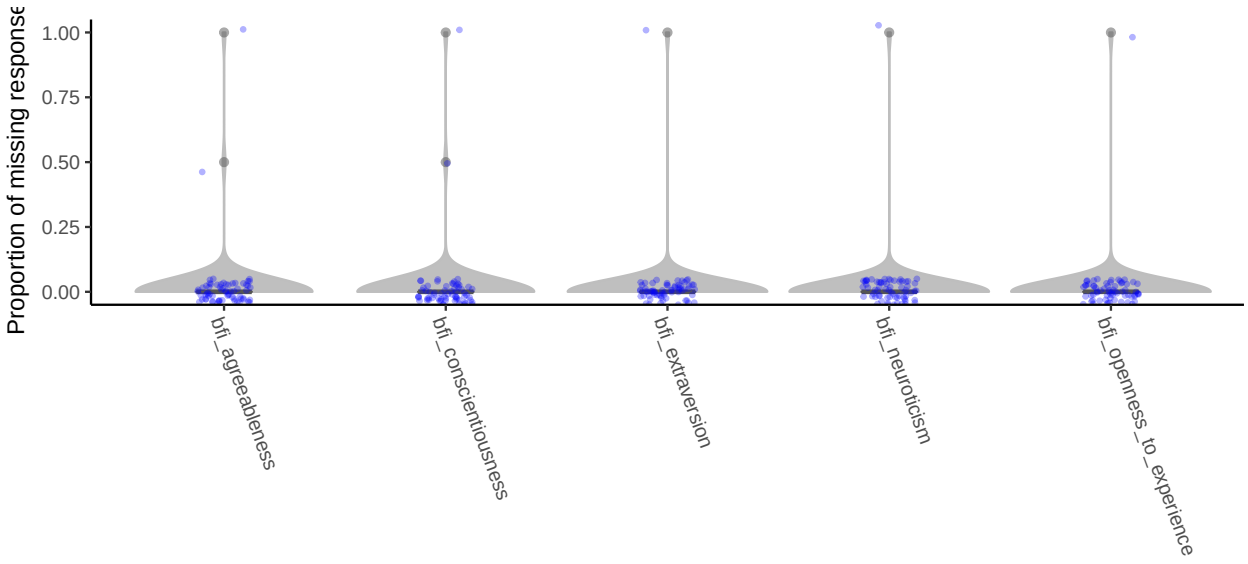
BFI-10

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^BFI-10$',
                                  type = 'score')
```

```
## Warning: Removed 5 rows containing non-finite values (stat_ydensity).
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
## Warning: Removed 5 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^BFI-10$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```



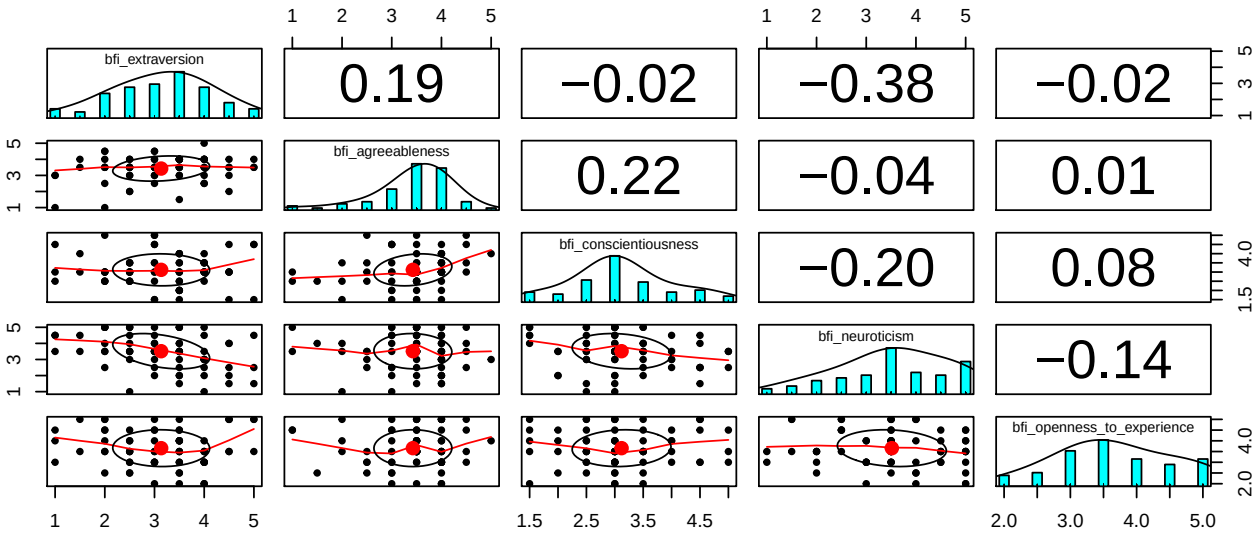

```
tds2w2_BFIRubric <- tds2_wave2_scoring_data_long_c %>%
    filter(scale_name == 'BFI-10')

tds2_wave2_bfi_psych <- scorequaltrics::score_questionnaire(tds2_wave2_long_recoded_c_nodupes,
                                  tds2w2_BFIRubric,
                                  psych = T)
print(tds2_wave2_bfi_psych, short = F)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
```

```
##       bfi_extraversion bfi_agreeableness bfi_conscientiousness
## alpha             0.66              0.23                  0.64
##       bfi_neuroticism bfi_openness_to_experience
## alpha            0.67                     0.008
##
## Standard errors of unstandardized Alpha:
##       bfi_extraversion bfi_agreeableness bfi_conscientiousness
## ASE               0.19              0.23                  0.19
##       bfi_neuroticism bfi_openness_to_experience
## ASE              0.19                      0.23
##
## Average item correlation:
##           bfi_extraversion bfi_agreeableness bfi_conscientiousness
## average.r             0.49              0.13                  0.47
##           bfi_neuroticism bfi_openness_to_experience
## average.r             0.5                     0.004
##
##  Guttman 6* reliability:
##          bfi_extraversion bfi_agreeableness bfi_conscientiousness
## Lambda.6              0.6              0.25                   0.6
##          bfi_neuroticism bfi_openness_to_experience
## Lambda.6             0.6                      0.17
##
## Signal/Noise based upon av.r :
##              bfi_extraversion bfi_agreeableness bfi_conscientiousness
## Signal/Noise              1.9              0.29                   1.8
##              bfi_neuroticism bfi_openness_to_experience
## Signal/Noise               2                     0.008
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                            bfi_extraversion bfi_agreeableness
## bfi_extraversion                      0.655            0.4981
## bfi_agreeableness                     0.191            0.2256
## bfi_conscientiousness                -0.024            0.2184
## bfi_neuroticism                      -0.381           -0.0427
## bfi_openness_to_experience           -0.016            0.0064
##                            bfi_conscientiousness bfi_neuroticism
## bfi_extraversion                          -0.037           -0.57
## bfi_agreeableness                          0.576           -0.11
## bfi_conscientiousness                      0.637           -0.31
## bfi_neuroticism                           -0.204            0.67
## bfi_openness_to_experience                 0.076           -0.14
##                            bfi_openness_to_experience
## bfi_extraversion                             -0.228
## bfi_agreeableness                             0.150
## bfi_conscientiousness                         1.065
## bfi_neuroticism                              -1.905
## bfi_openness_to_experience                    0.008
##
## Item by scale correlations:
##  corrected for item overlap and scale reliability
##        bfi_extraversion bfi_agreeableness bfi_conscientiousness
```

```
## BFI_1              -0.66              -0.27                    0.25
## BFI_6               0.68               0.38                    0.16
## BFI_2               0.24               0.35                    0.06
## BFI_7              -0.12              -0.41                   -0.40
## BFI_3              -0.06              -0.41                   -0.66
## BFI_8              -0.14               0.34                    0.69
## BFI_4               0.36               0.09                    0.25
## BFI_9              -0.50              -0.05                   -0.21
## BFI_5               0.06              -0.02                   -0.01
## BFI_10              0.05               0.00                    0.15
##        bfi_neuroticism bfi_openness_to_experience
## BFI_1             0.30                      -0.23
## BFI_6            -0.54                      -0.26
## BFI_2            -0.03                      -0.20
## BFI_7             0.06                      -0.27
## BFI_3             0.30                      -0.32
## BFI_8            -0.14                      -0.06
## BFI_4            -0.67                       0.37
## BFI_9             0.67                      -0.21
## BFI_5            -0.04                      -0.26
## BFI_10           -0.36                       0.34
##
## Non missing response frequency for each item
##            1    2    3    4    5 miss
## BFI_1   0.06 0.12 0.36 0.33 0.12 0.01
## BFI_6   0.05 0.18 0.21 0.26 0.30 0.01
## BFI_2   0.08 0.08 0.08 0.48 0.29 0.01
## BFI_7   0.06 0.23 0.40 0.28 0.03 0.03
## BFI_3   0.08 0.12 0.11 0.48 0.21 0.01
## BFI_8   0.02 0.08 0.11 0.62 0.18 0.03
## BFI_4   0.21 0.21 0.21 0.24 0.12 0.01
## BFI_9   0.06 0.09 0.15 0.30 0.39 0.01
## BFI_5   0.26 0.14 0.20 0.30 0.11 0.01
## BFI_10  0.02 0.06 0.18 0.23 0.52 0.01
```

```
pairs.panels(tds2_wave2_bfi_psych$scores)
```



CARE-R Social

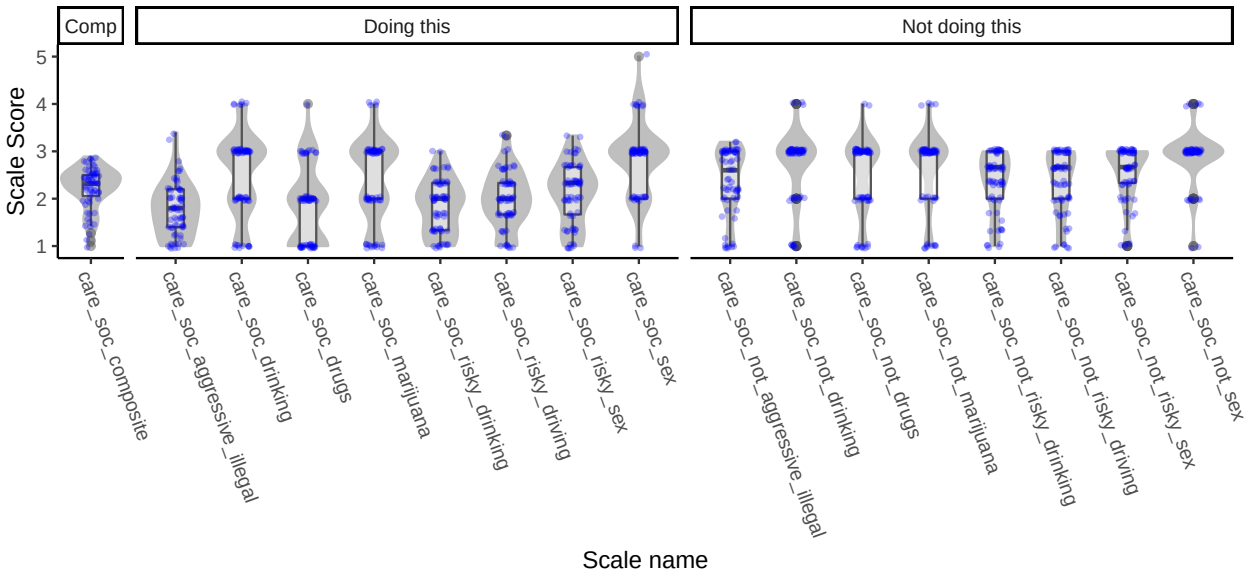
Note: Lower scores on "not doing this" items indicate that someone thinks not engaging in the behavior will make people like them *more*, and thus, like the "doing this" items, a lower score corresponds to a downward influence on that behavior.

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^CARE-R Social$',
                                  type = 'score') +
    facet_grid(~factor(grepl('not', scored_scale) - grepl('composite', scored_scale),
                       levels = c(-1, 0, 1),
                       labels = c('Comp', 'Doing this', 'Not doing this')),
               scales = 'free_x',
               space = 'free_x')
```

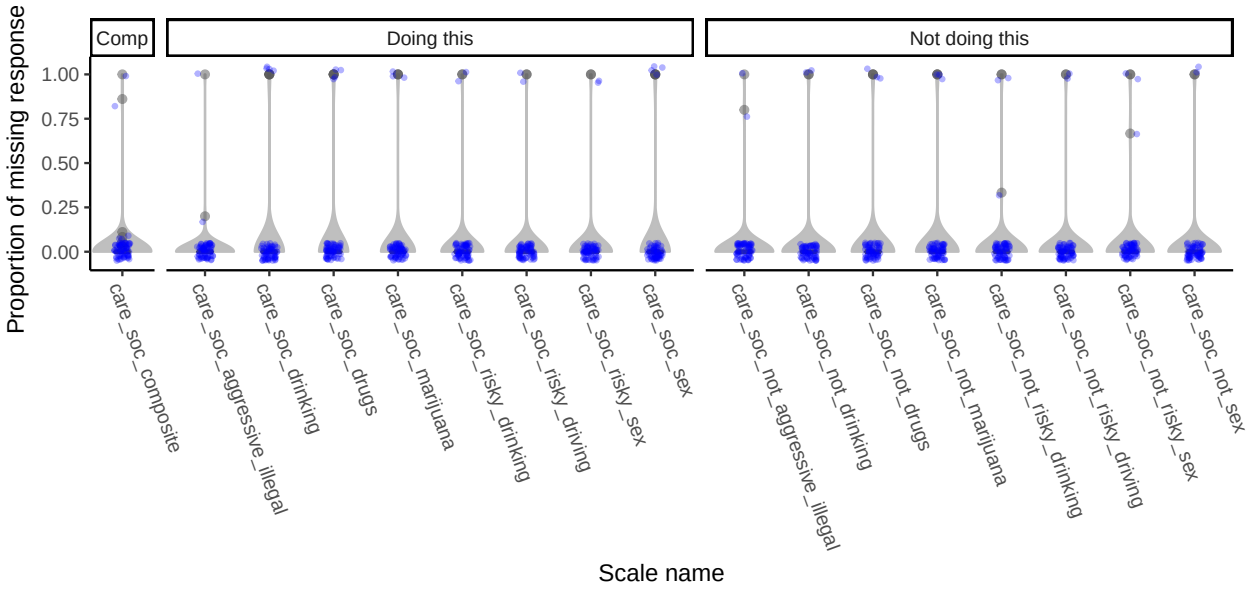
```
## Warning: Removed 40 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 40 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 40 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^CARE-R Social$',
                                  type = 'p_missing') +
    facet_grid(~factor(grepl('not', scored_scale) - grepl('composite', scored_scale),
                       levels = c(-1, 0, 1),
                       labels = c('Comp', 'Doing this', 'Not doing this')),
               scales = 'free_x',
               space = 'free_x')
```



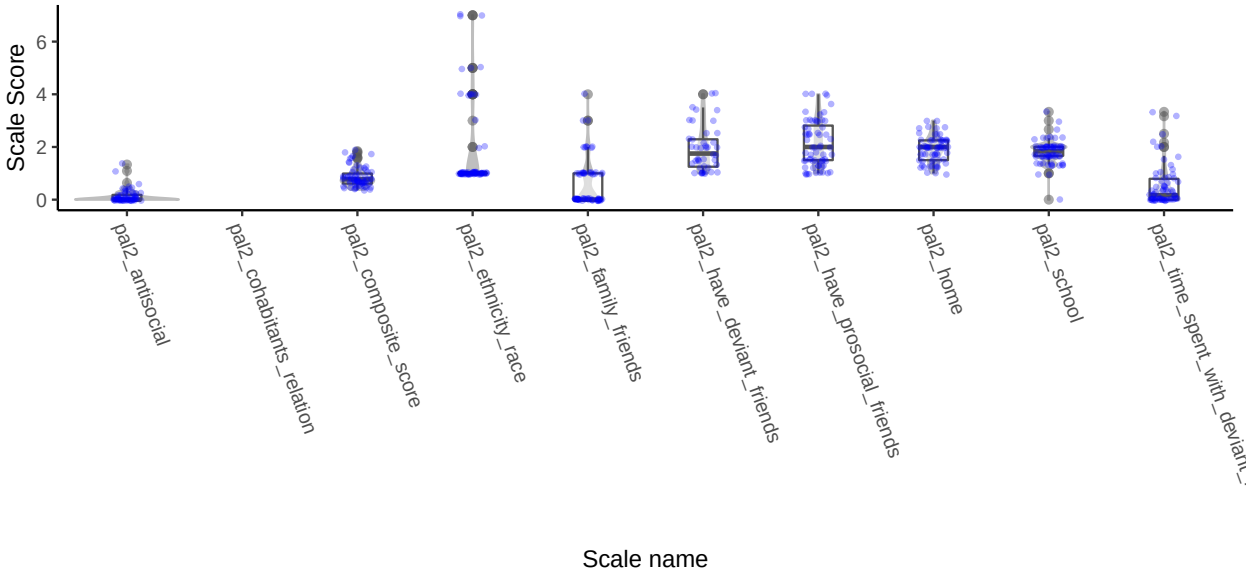
PAL-2

```
scorequaltrics::plot_scored_scale(filter(tds2_wave2_scored_c, scored_scale != 'pal2_age'),
                                  scale_regx = '^PAL-2$',
                                  type = 'score')
```

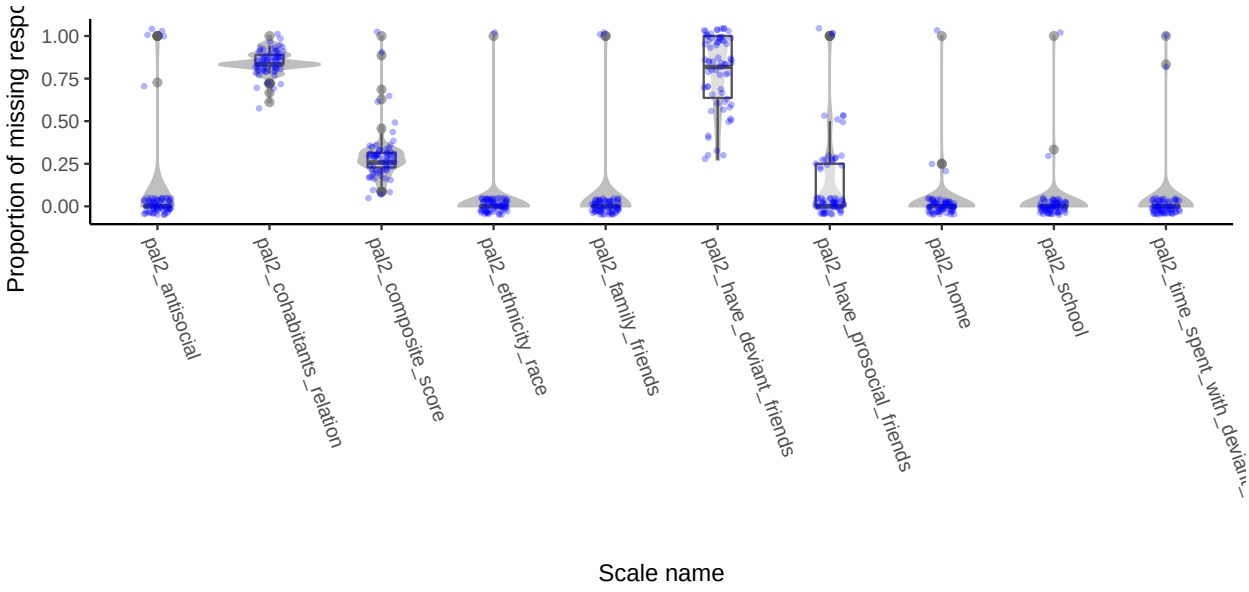
```
## Warning: Removed 105 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 105 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 105 rows containing missing values (geom_point).
```

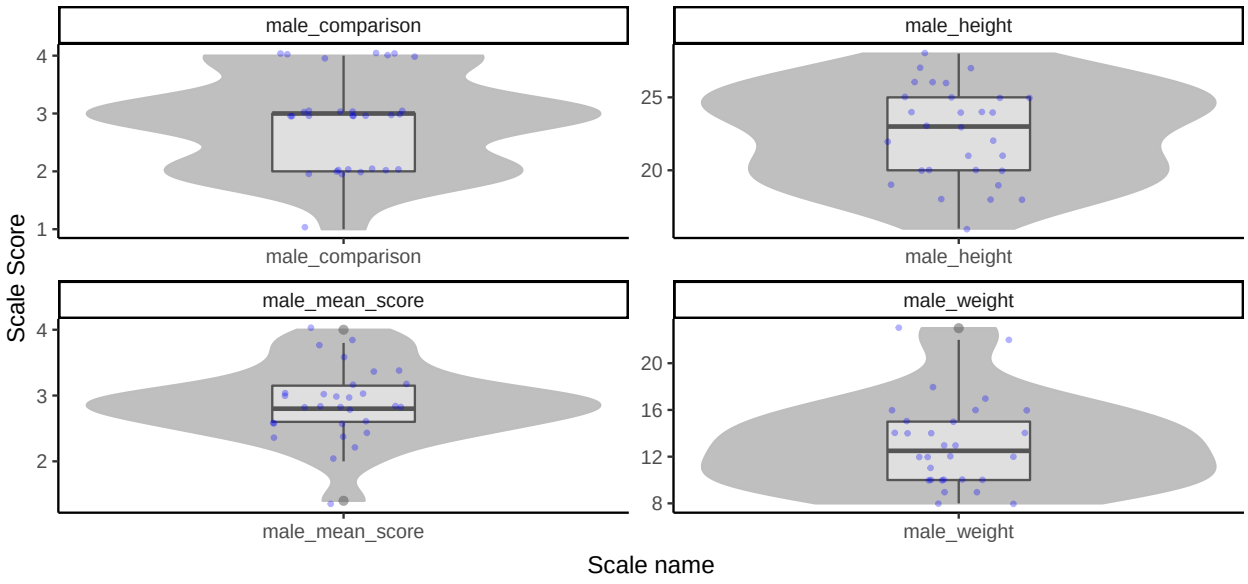



```
scorequaltrics::plot_scored_scale(filter(tds2_wave2_scored_c, scored_scale != 'pal2_age'),
                                  scale_regx = '^PAL-2$',
                                  type = 'p_missing')
```

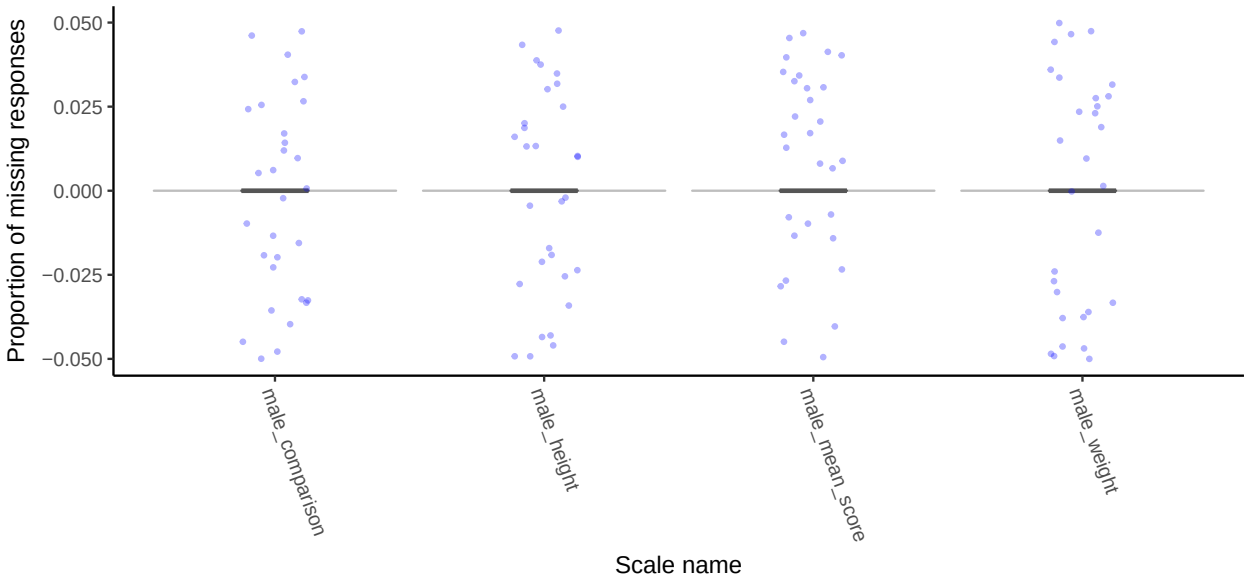


PDS

```
#We need a way to separate female from male responses
male_sids <- tds2_wave2_scored_c %>%
    filter(scored_scale == 'pds_gender', score =='0') %>%
    ungroup() %>%
    select(SID)

female_sids <- tds2_wave2_scored_c %>%
    filter(scored_scale == 'pds_gender', score =='1') %>%
    ungroup() %>%
    select(SID)

scorequaltrics::plot_scored_scale(filter(tds2_wave2_scored_c,
                                         SID %in% male_sids$SID,
                                         grepl('^male', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'score') +
    facet_wrap(~scored_scale, scales = 'free') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```




```
scorequaltrics::plot_scored_scale(filter(tds2_wave2_scored_c,
                                         SID %in% male_sids$SID,
                                         grepl('^male', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'p_missing')
```

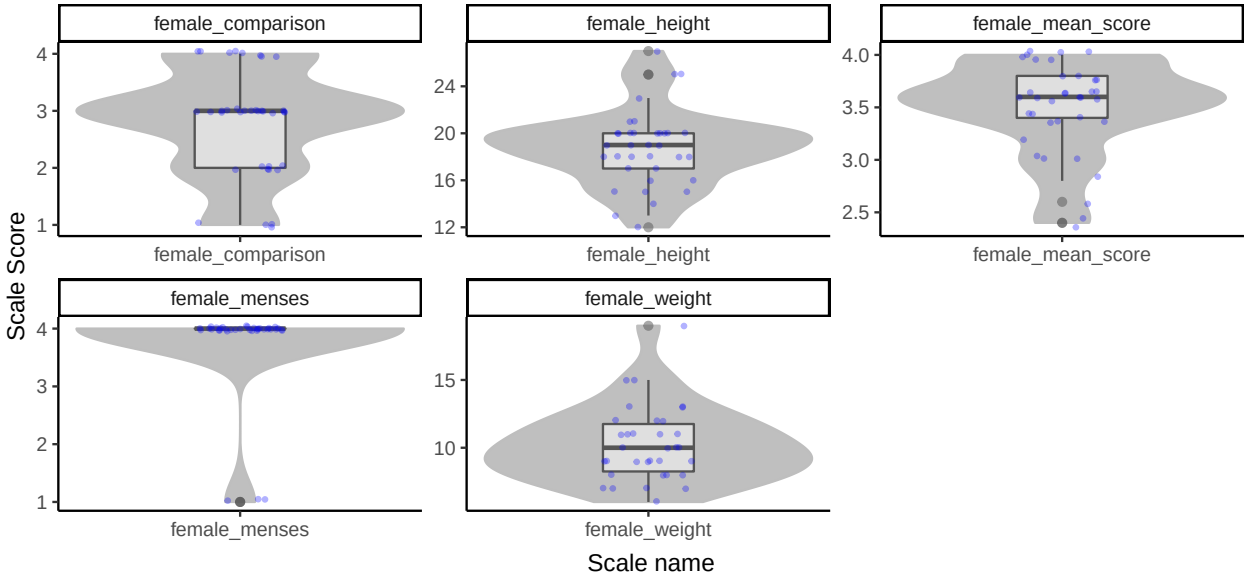



```
scorequaltrics::plot_scored_scale(filter(tds2_wave2_scored_c,
                                         SID %in% female_sids$SID,
                                         grepl('^female', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'score') +
    facet_wrap(~scored_scale, scales = 'free') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

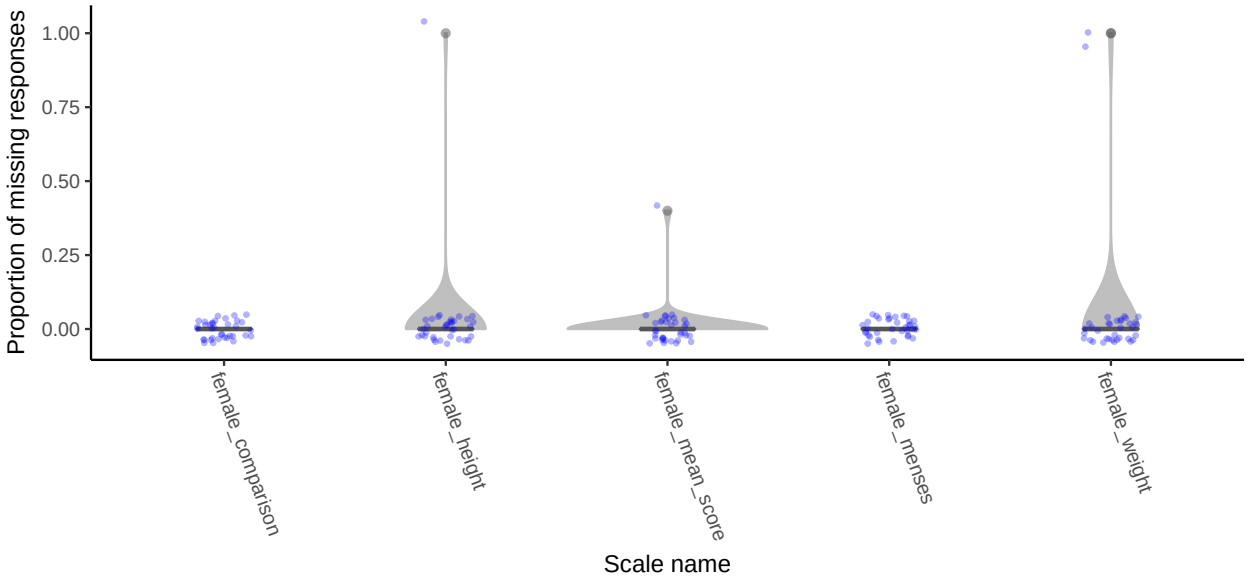
```
## Warning: Removed 3 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 3 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



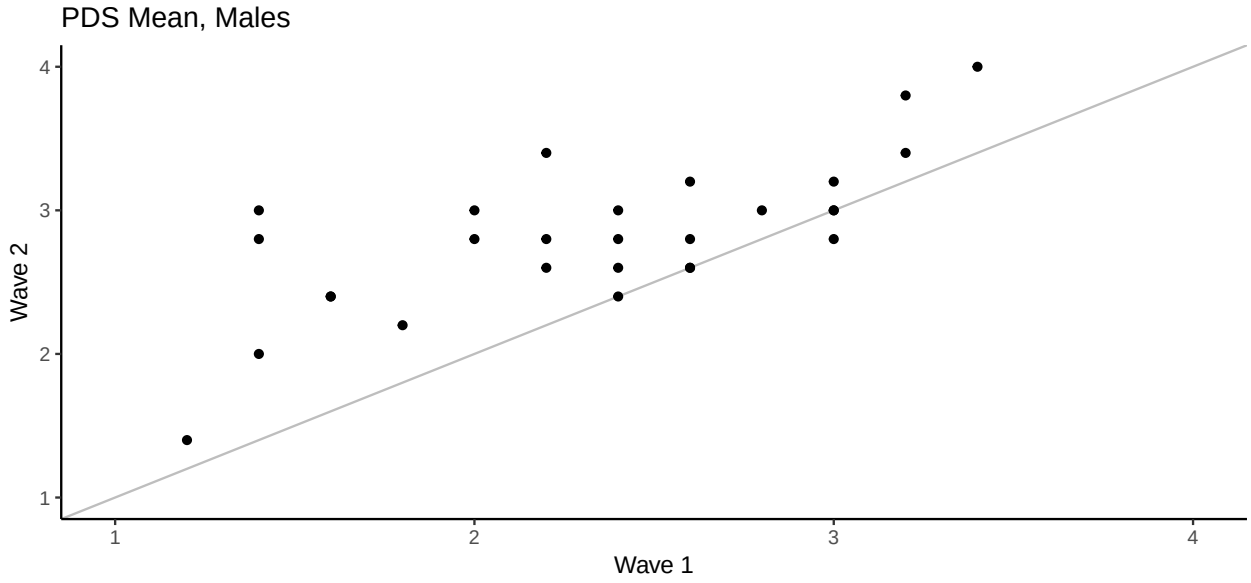

```
scorequaltrics::plot_scored_scale(filter(tds2_wave2_scored_c,
                                         SID %in% female_sids$SID,
                                         grepl('^female', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'p_missing')
```



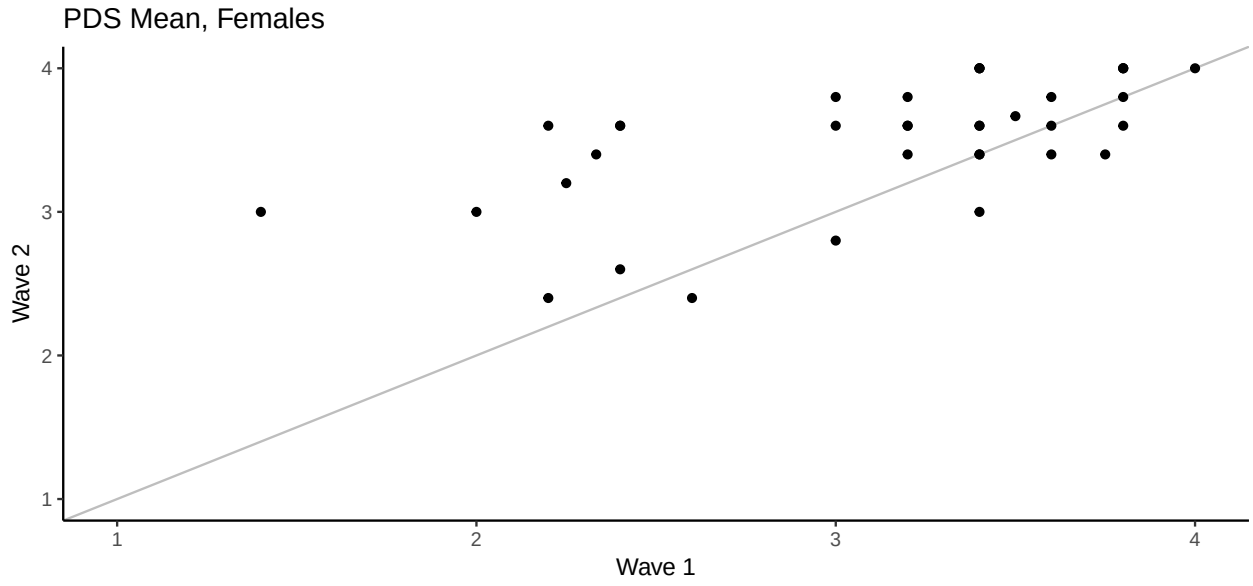

```
tds2_wave2_scored_c_pds <- tds2_wave2_scored_c %>%
    filter(scale_name == 'PDS',
           grepl('mean_score', scored_scale))
tds2_wave1_scored_pds <- tds2_wave1_scored %>%
    filter(scale_name == 'PDS',
           grepl('mean_score', scored_scale))

tds2_wave12_pds <- bind_rows("Wave 1" = tds2_wave1_scored_pds,
                             "Wave 2" = tds2_wave2_scored_c_pds,
```

```
                              .id = 'wave') %>%
    select(wave, scored_scale, SID, score)

tds2_wave12_pds %>%
    filter(SID %in% male_sids$SID,
           grepl('^male', scored_scale)) %>%
    spread(wave, score) %>%
    mutate_at(vars(`Wave 1`, `Wave 2`), .funs = as.numeric) %>%
    ggplot(aes(x = `Wave 1`, y = `Wave 2`)) +
    geom_abline(intercept = 0, slope = 1, color = 'gray') +
    geom_point() +
    coord_cartesian(x = c(1,4), y = c(1, 4)) +
    theme_classic() +
    labs(title = 'PDS Mean, Males')
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



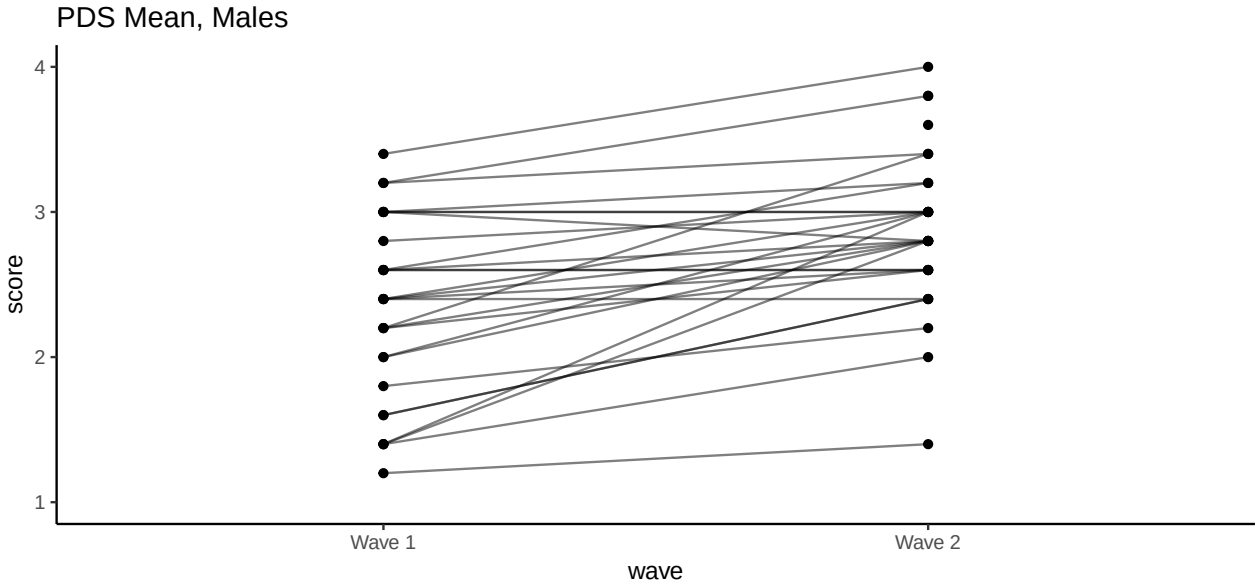

```
tds2_wave12_pds %>%
    filter(SID %in% female_sids$SID,
           grepl('^female', scored_scale)) %>%
    spread(wave, score) %>%
    mutate_at(vars(`Wave 1`, `Wave 2`), .funs = as.numeric) %>%
    ggplot(aes(x = `Wave 1`, y = `Wave 2`)) +
    geom_abline(intercept = 0, slope = 1, color = 'gray') +
    geom_point() +
    coord_cartesian(x = c(1,4), y = c(1, 4)) +
    theme_classic() +
    labs(title = 'PDS Mean, Females')
```




```
tds2_wave12_pds %>%
    filter(SID %in% male_sids$SID,
           grepl('^male', scored_scale)) %>%
    mutate(score = as.numeric(score)) %>%
    ggplot(aes(x = wave, y = score, group = SID)) +
    geom_line(alpha = .5) +
    geom_point() +
    coord_cartesian(y = c(1, 4)) +
    theme_classic() +
    labs(title = 'PDS Mean, Males')
```

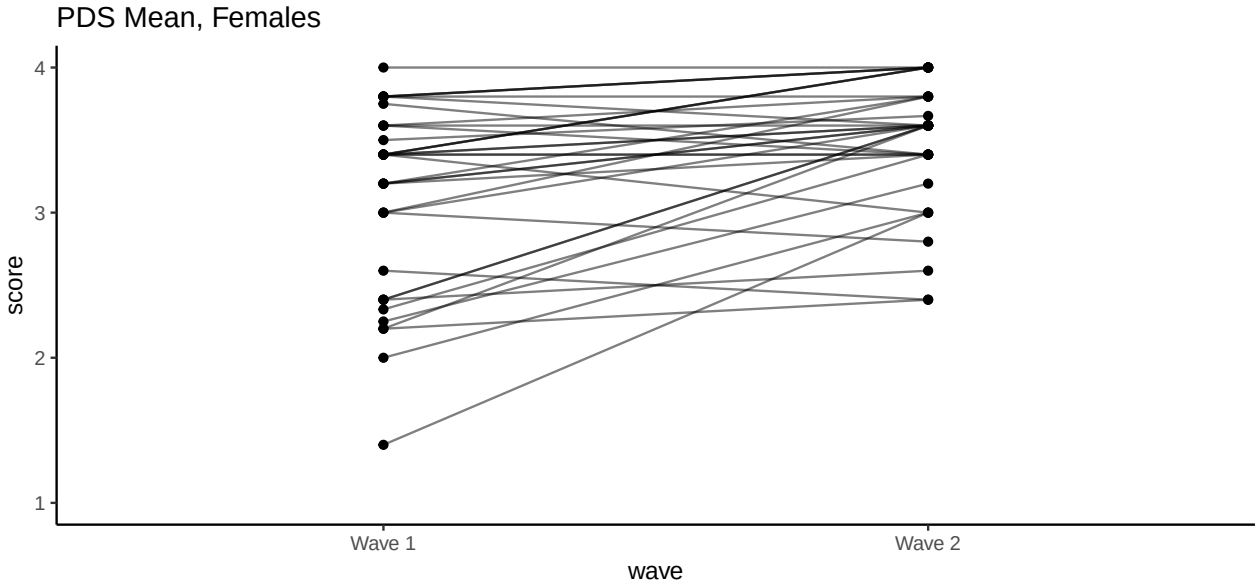
```
## Warning: Removed 2 rows containing missing values (geom_path).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



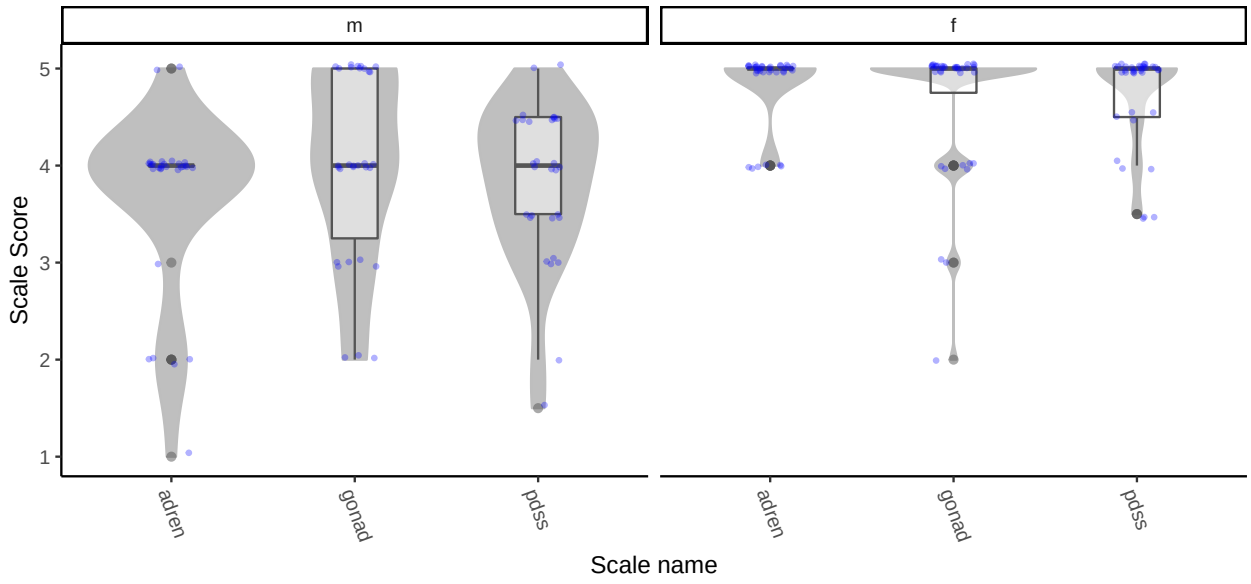

```
tds2_wave12_pds %>%
    filter(SID %in% female_sids$SID,
           grepl('^female', scored_scale)) %>%
```

```
    mutate(score = as.numeric(score)) %>%
    ggplot(aes(x = wave, y = score, group = SID)) +
    geom_line(alpha = .5) +
    geom_point() +
    coord_cartesian(y = c(1, 4)) +
    theme_classic() +
    labs(title = 'PDS Mean, Females')
```

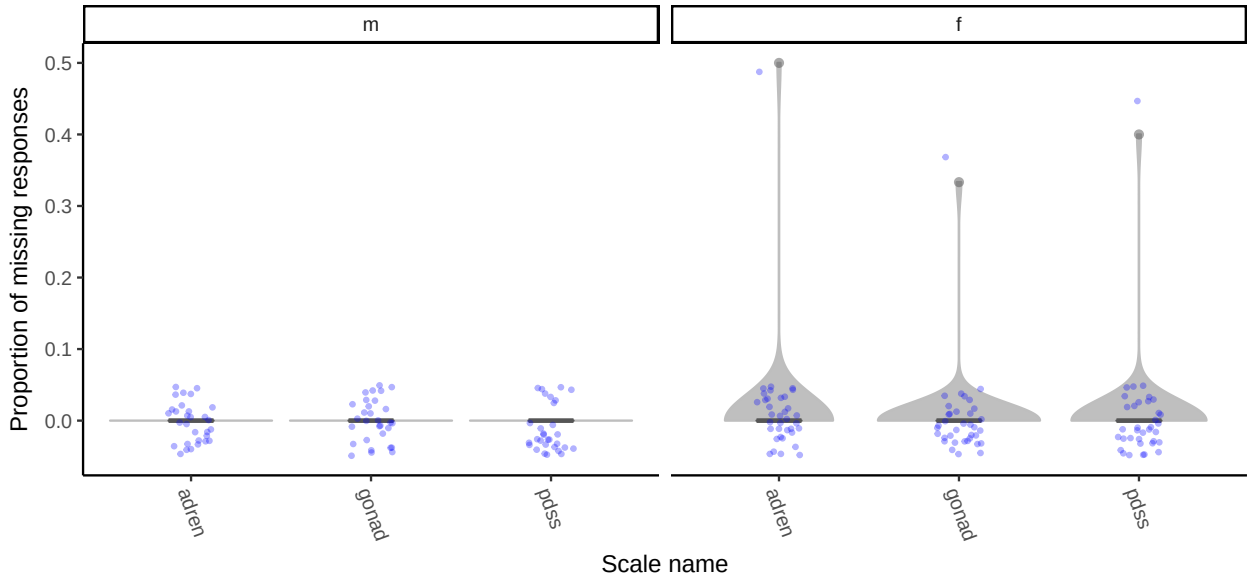


PDSS

```
tds2_wave2_scored_pdss_gender <- tds2_wave2_scored_pdss %>%
    filter(scored_scale == 'gender') %>%
    spread(scored_scale, score) %>%
    select(SID, gender)

tds2_wave2_scored_pdss_gender_wide <- tds2_wave2_scored_pdss %>%
    filter(scored_scale != 'gender') %>%
    left_join(tds2_wave2_scored_pdss_gender)

scorequaltrics::plot_scored_scale(tds2_wave2_scored_pdss_gender_wide,
                                  scale_regx = '^PDSS$',
                                  type = 'score') +
    facet_wrap(~factor(gender,
                       levels = pdss_gender_code,
                       labels = names(pdss_gender_code)))
```




```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_pdss_gender_wide,
                                  scale_regx = '^PDSS$',
                                  type = 'p_missing') +
    facet_wrap(~factor(gender,
                       levels = pdss_gender_code,
                       labels = names(pdss_gender_code)))
```




```
tds2_wave12_pdss <- bind_rows(wave_1 = tds2_wave1_scored_pdss_gender_wide,
                              wave_2 = tds2_wave2_scored_pdss_gender_wide,
                              .id = 'wave')

tds2_wave12_pdss %>%
    select(SID, wave, gender) %>%
    distinct(SID, wave, gender) %>%
    mutate(gender = factor(gender,
                           levels = pdss_gender_code,
                           labels = names(pdss_gender_code))) %>%
```

```
    spread(wave, gender) %>%
    select(wave_1, wave_2) %>%
    table(useNA = 'ifany') %>%
    knitr::kable(caption = 'Gender consistency across waves')
```

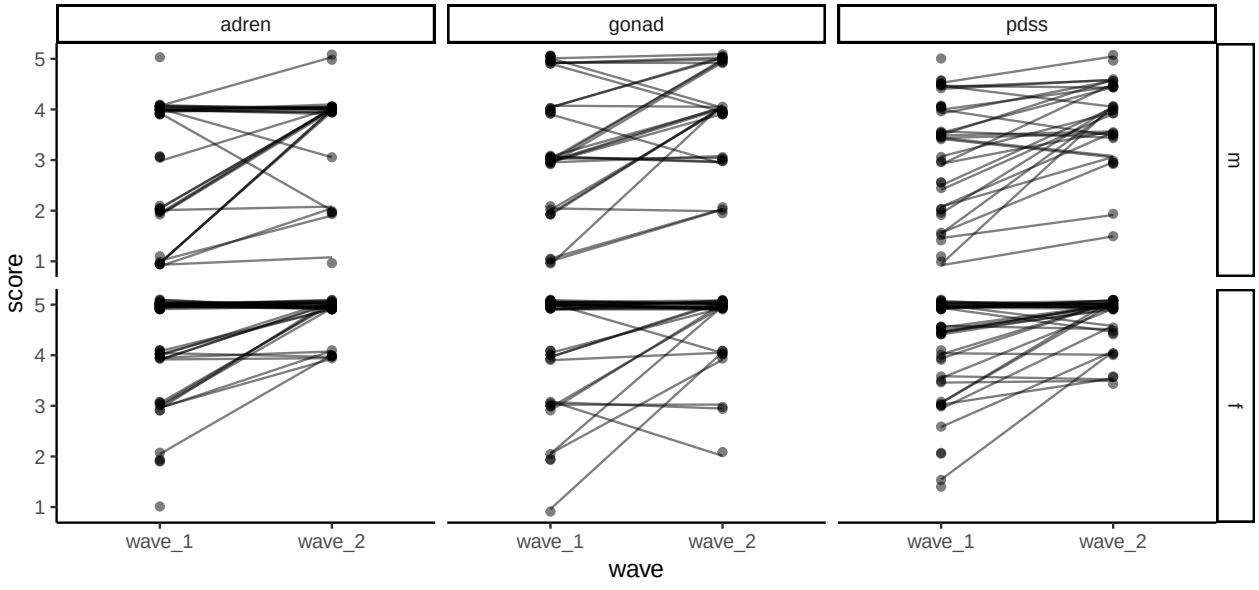
Table 8: Gender consistency across waves

	m	f	NA
m	29	0	15
f	0	36	24
NA	1	0	0

```
ajitter <- position_jitter(w = 0, h = .1)
tds2_wave12_pdss %>%
    mutate(score = as.numeric(score),
           gender = factor(gender,
                           levels = pdss_gender_code,
                           labels = names(pdss_gender_code))) %>%
    ggplot(aes(x = wave, y = score, group = SID)) +
    geom_point(alpha = .5, position = ajitter) +
    geom_line(alpha = .5, position = ajitter) +
    facet_grid(gender~scored_scale) +
    theme_classic()
```

```
## Warning: Removed 14 rows containing missing values (geom_point).
```

```
## Warning: Removed 5 rows containing missing values (geom_path).
```

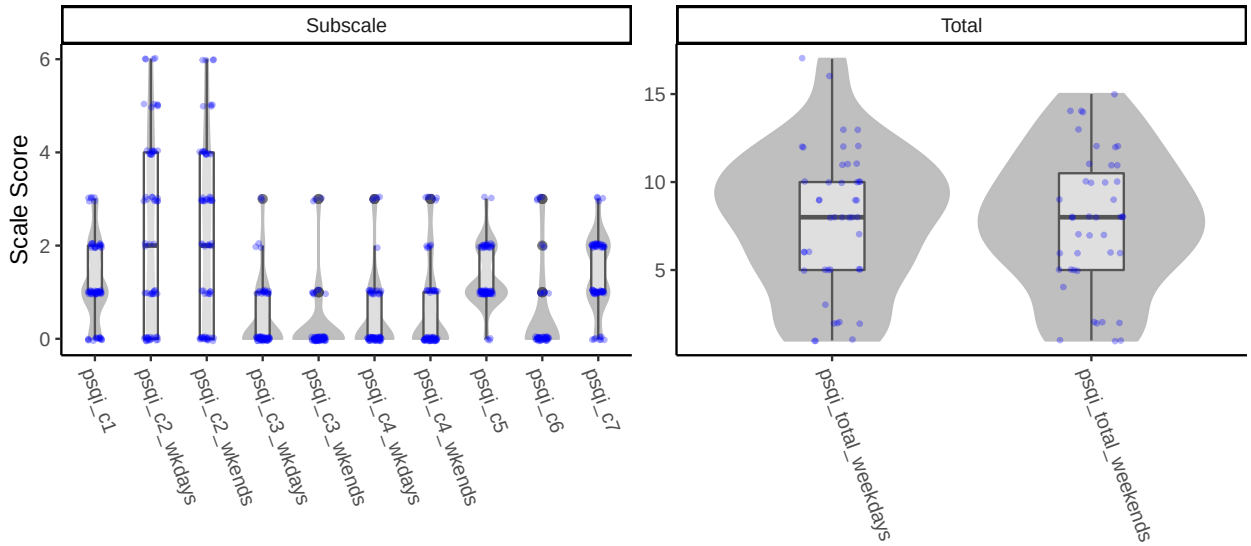


PSQI

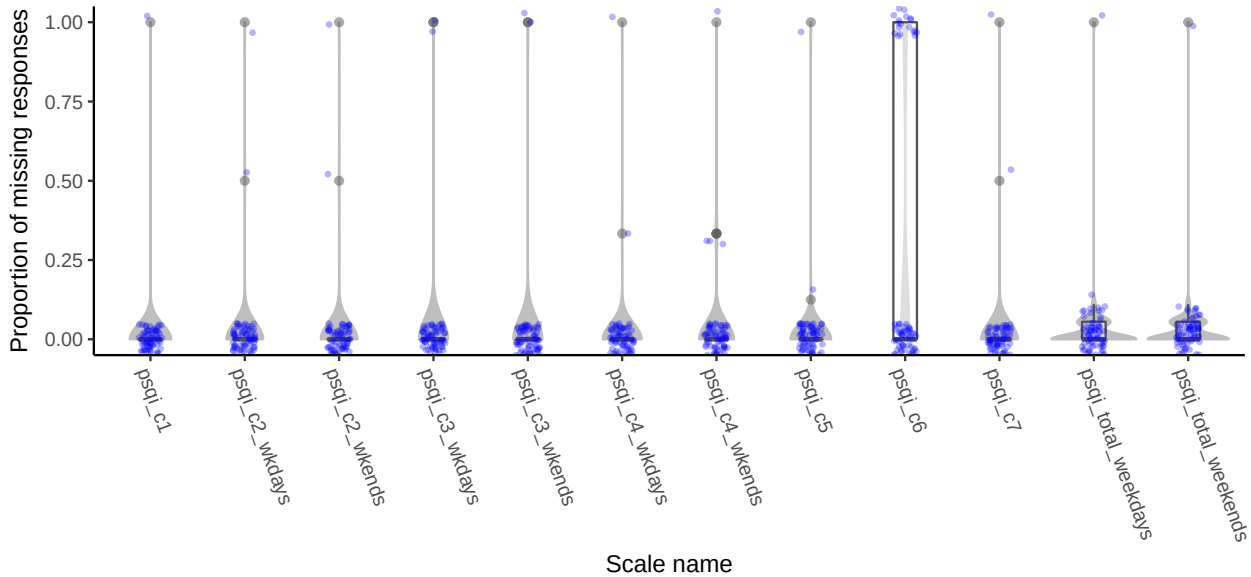
```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_psqi,
                                  scale_regx = '^PSQI$',
                                  type = 'score')+
```

```
    facet_wrap(~factor(as.numeric(grepl('total', scored_scale)),
                       levels = c(0, 1),
                       labels = c('Subscale', 'Total')),
               scales = 'free')
```

```
## Warning: Removed 86 rows containing non-finite values (stat_ydensity).
## Warning: Removed 86 rows containing non-finite values (stat_boxplot).
## Warning: Removed 86 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_psqi,
                                  scale_regx = '^PSQI$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```



SES

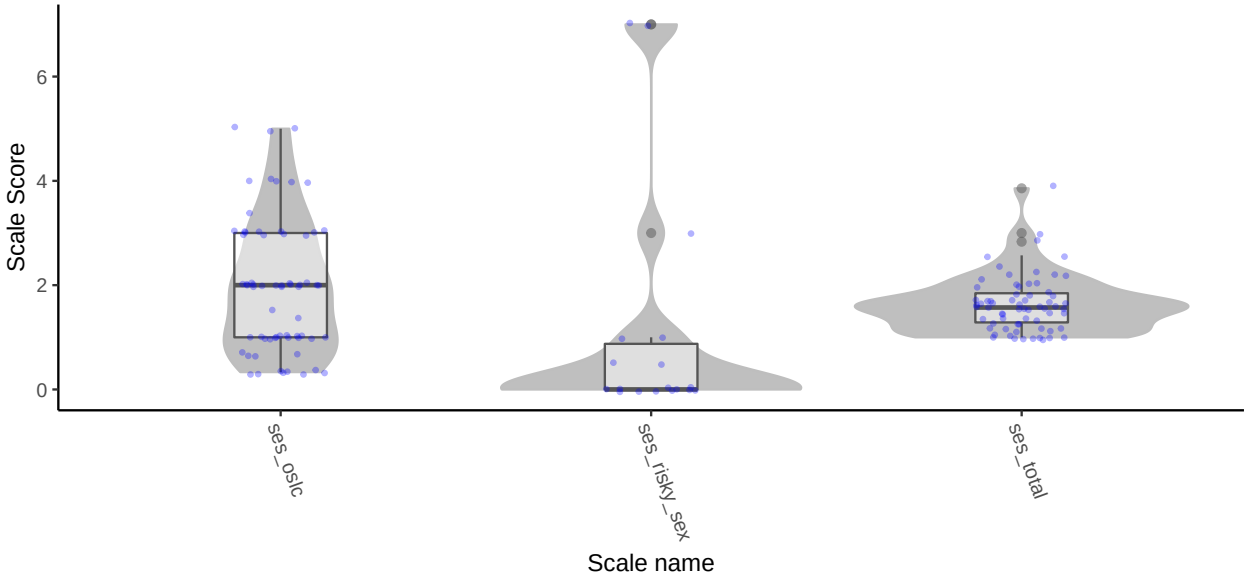
Needs a checkup

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^SES$',
                                  type = 'score')
```

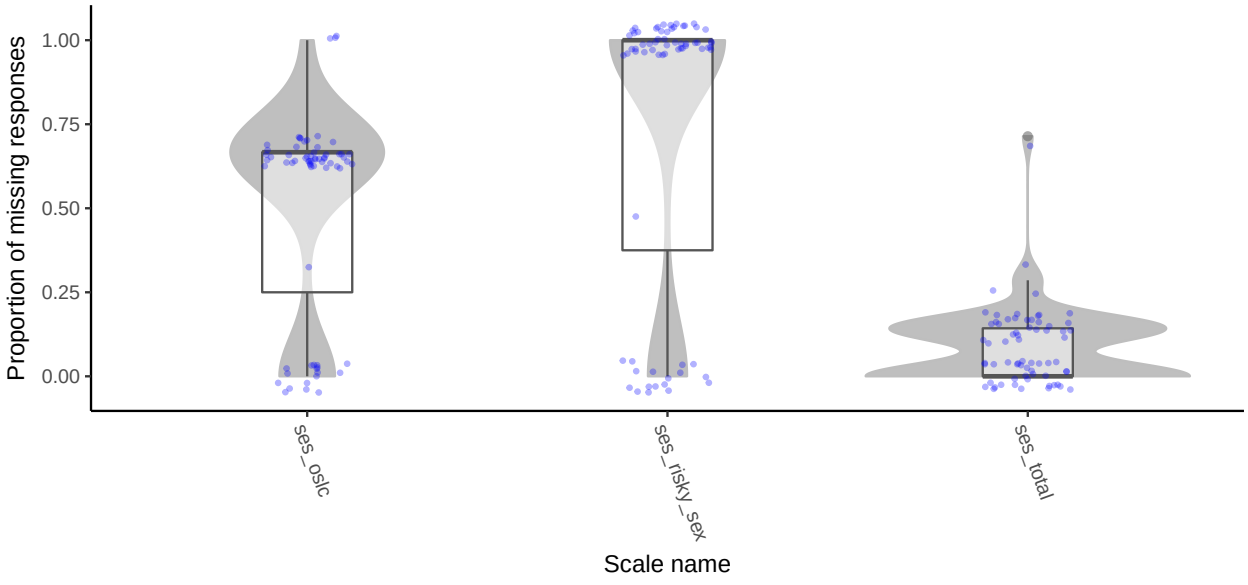
```
## Warning: Removed 53 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 53 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 53 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^SES$',
                                  type = 'p_missing')
```



YRBS

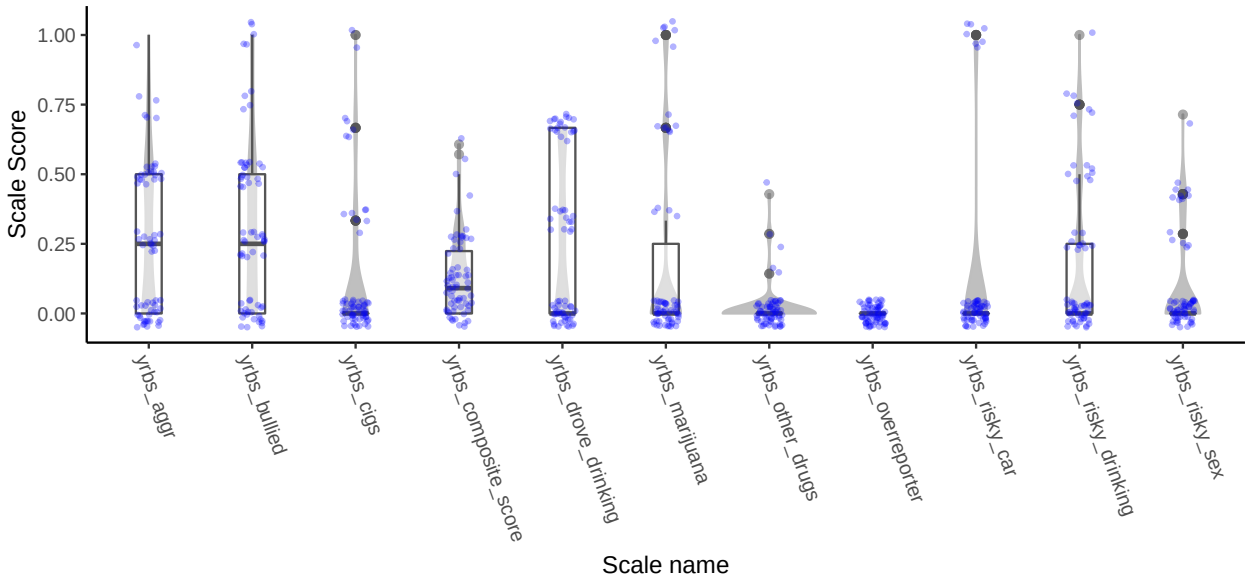
This has been scored according to the CDC manual. Items have been scored with either 1 or 0, with the mean calculated for each subscale. This gives you the proportion of endorsed items in that scale. To convert to a sum of endorsed items, simply multiply the scaled score by the number of total items (`n_missing` + `n_items`).

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^YRBS$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 360-70))
```

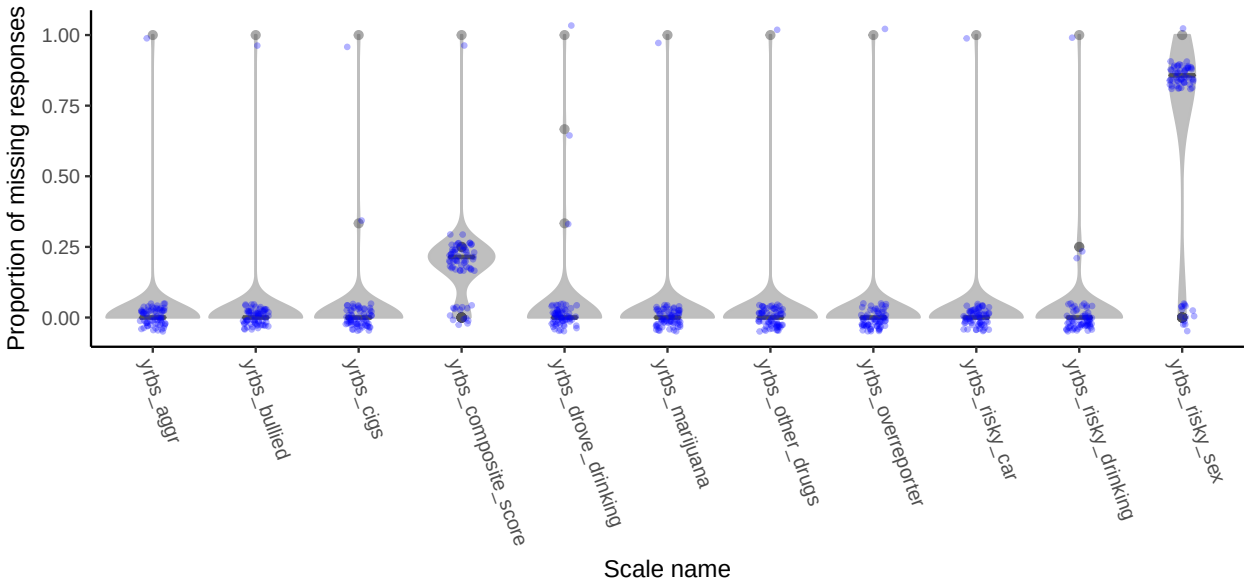
```
## Warning: Removed 11 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 11 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```

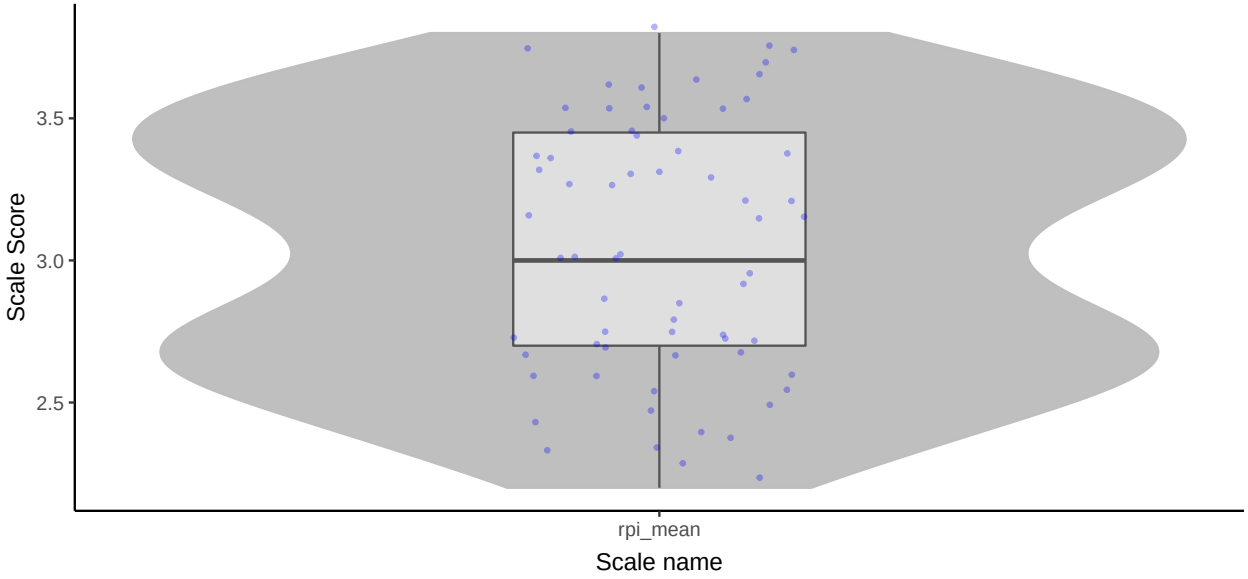



```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^YRBS$',
                                  type = 'p_missing')
```

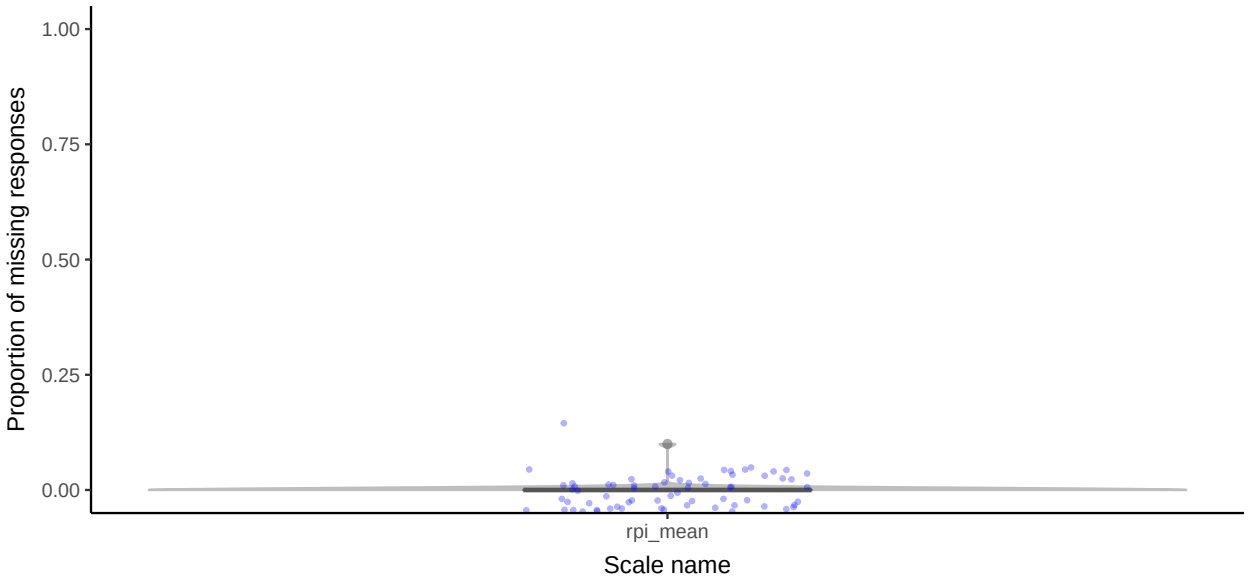


RPI

```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^RPI_part2$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```




```
scorequaltrics::plot_scored_scale(tds2_wave2_scored_c,
                                  scale_regx = '^RPI_part2$',
                                  type = 'p_missing') +
    coord_cartesian(y = c(0,1)) +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```



Write the data

```
list_of_scales_c <- as.list(unique(tds2_wave2_scored_c$scale_name))
list_of_scales_p <- as.list(unique(tds2_wave2_scored_p$scale_name))
tds2_wave2_metadata <- list(wave = 2, questionnaires = 'TDS2')

throwaway <- lapply(list_of_scales_c,
                    function(aScale){
                        file_name <- paste0(TDS2_name, '-',
                                            wave2_name, '-',
                                            scorequaltrics::make_nice_scale_fname(aScale),
                                            '.csv')
                        scorequaltrics::write_widened_scored_scale(
                            tds2_wave2_scored_c,
                            scale_names = aScale,
                            dir_name = output_file_dir,
                            file_name = file_name,
                            metadata = tds2_wave2_metadata)
                    })

throwaway <- lapply(list_of_scales_p,
                    function(aScale){
                        file_name <- paste0(TDS2_name, '-',
                                            wave2_name, '-',
                                            scorequaltrics::make_nice_scale_fname(aScale),
                                            '.csv')
                        scorequaltrics::write_widened_scored_scale(
                            tds2_wave2_scored_p,
                            scale_names = aScale,
                            dir_name = output_file_dir,
                            file_name = file_name,
                            metadata = tds2_wave2_metadata)
                    })
```

```
tds2_wave2_pdss_filename <-  paste0(TDS2_name, '-',
                                    wave2_name, '-',
                                    'PDSS.csv')

scorequaltrics::write_widened_scored_scale(
    tds2_wave2_scored_pdss,
    scale_names = 'PDSS',
    dir_name = output_file_dir,
    file_name = tds2_wave2_pdss_filename,
    metadata = tds2_wave2_metadata)

tds2_wave2_psqi_filename <-  paste0(TDS2_name, '-',
                                    wave2_name, '-',
                                    'PSQI.csv')

scorequaltrics::write_widened_scored_scale(
    tds2_wave2_scored_psqi,
    scale_names = 'PSQI',
    dir_name = output_file_dir,
    file_name = tds2_wave2_psqi_filename,
    metadata = tds2_wave2_metadata)
```

TDS 1, Wave 1 Scores

Get the data and clean it

Get data, rubrics, recode, and clean. The following questionnaires should be exactly the same between pre and post, so I'll start with these. Also, it's important to note that all response recoding comes from the tds1/wave1/post folder.

Same across pre and post:

- ACE
- BFNE
- BIS-15
- BSSS
- CARE-R EI
- CARE-R WE
- CBCL
- MSSS
- NTS
- PEQ
- RPI
- SPRSQ-S
- UPPSP

Unique in pre:

- CARE-R PF (Past Frequency scale only in TDS1 Wave 1 Pre)
- RSQ (12 items pre, 10 items post)
- RSQ Video & Grocery (questionnaire exists only in TDS1 Wave 1 pre)

```
#separate out the surveys we want
tds1_wave1_surveys <- rawSurveysTDS %>%
```

```
    filter(grepl('TDS1 (Session [12]|CBCL).*(Post|Pre)$', SurveyName))
knitr::kable(select(tds1_wave1_surveys, SurveyName))
```

SurveyName
TDS1 CBCL - Pre
TDS1 Session 1 - Post
TDS1 Session 2 B - Post
TDS1 Session 2 A - Pre
TDS1 CBCL - Post
TDS1 Session 2 B - Pre
TDS1 Session 1 - Pre
TDS1 Session 2 A - Post

```
#Get data from those surveys
tds1_wave1_long <- scorequaltrics::get_survey_data(tds1_wave1_surveys,
                                                   credentials,
                                                   pid_col = pid_column_name)

#separate out the unique pre questions
pre_item_regex <- '.*(CARE_PF|RSQ).*'
tds1_wave1_long_pre <- tds1_wave1_long %>%
    filter(grepl('Pre', survey_name)) %>%
    filter(grepl(pre_item_regex, item))

tds1_wave1_long_prepost <- tds1_wave1_long %>%
    filter(!(grepl('Pre', survey_name) & grepl(pre_item_regex, item)))

#Add in PDS for those who redid it during session 2
tds1_wave1_surveys_pds2 <- rawSurveysTDS %>%
    filter(grepl('TDS1 PDS \\(Session2\\) - Post', SurveyName))
tds1_wave1_long_pds2 <- scorequaltrics::get_survey_data(tds1_wave1_surveys_pds2,
                                                        credentials,
                                                        pid_col = pid_column_name) %>%
    filter(grepl('PDS', item))

tds1_wave1_sid_with_pds2 <- tds1_wave1_long_pds2 %>%
    group_by(SID) %>%
    summarize(n_items = sum(!is.na(as.numeric(value)))) %>%
    filter(n_items > 0) %>%
    select(SID) %>%
    unlist
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
tds1_wave1_long_prepost <- tds1_wave1_long_prepost %>%
    filter(!(grepl('PDS', item) &
                 SID %in% tds1_wave1_sid_with_pds2)) %>% # remove PDS items from pds
                                                         # session 2 participants
    bind_rows(filter(tds1_wave1_long_pds2,
                     SID %in% tds1_wave1_sid_with_pds2)) # Add rows from PDS session 2
                                                         # participants, given that they're
                                                         # in the list of SIDs with >0 items
```

```
#Get recoding rubrics for all data from the post folder
tds1_wave1_recoding_rubrics <- data.frame(file = dir(file.path(tds1_wave1_rubric_dir),
                                                     pattern = '.*response_recoding.*.csv',
                                                     full.names = TRUE))
tds1_wave1_recoding_data_long <- scorequaltrics::get_rubrics(tds1_wave1_recoding_rubrics,
                                                             type = 'recoding')

#Recode data for both pre unique and pre/post
tds1_wave1_long_pre_recoded <- scorequaltrics::recode_responses(tds1_wave1_long_pre,
                                                                tds1_wave1_recoding_data_long)
tds1_wave1_long_prepost_recoded <- scorequaltrics::recode_responses(tds1_wave1_long_prepost,
                                                                    tds1_wave1_recoding_data_long)

#Code -99 as NA
tds1_wave1_long_prepost_recoded <- tds1_wave1_long_prepost_recoded %>%
    mutate(value = ifelse(grepl('(SES|YRBS)', item) & as.numeric(value) < 0,
                          NA,
                          value))
```

```
## Warning in ifelse(grepl("(SES|YRBS)", item) & as.numeric(value) < 0, NA, :
## NAs introduced by coercion
```

```
#Get scoring rubrics for pre and pre/post
tds1_wave1_scoring_rubrics_pre <- data.frame(file = dir(file.path(tds1_wave1_rubric_dir_pre),
                                                        pattern = '.*scoring_rubric.*.csv',
                                                        full.names = TRUE))
tds1_wave1_scoring_data_long_pre <- scorequaltrics::get_rubrics(tds1_wave1_scoring_rubrics_pre,
                                                                type = 'scoring')

tds1_wave1_scoring_rubrics <- data.frame(file = dir(file.path(tds1_wave1_rubric_dir),
                                                    pattern = '.*scoring_rubric.*.csv',
                                                    full.names = TRUE))
tds1_wave1_scoring_data_long <- scorequaltrics::get_rubrics(tds1_wave1_scoring_rubrics,
                                                            type = 'scoring')
```

```
## Warning: Unknown variables: `transform`
```

```
#Clean and de-dupe pre and pre/post data
tds1_wave1_long_pre_recoded_nodupes <- tds1_wave1_long_pre_recoded %>%
    get_items_in_rubric(tds1_wave1_scoring_data_long_pre) %>%
    filter(grepl('3\\d\\d', SID)) %>%
    scorequaltrics::clean_dupes(pid_col = 'SID')
tds1_wave1_long_prepost_recoded_nodupes <- tds1_wave1_long_prepost_recoded %>%
    get_items_in_rubric(tds1_wave1_scoring_data_long) %>%
    filter(grepl('3\\d\\d', SID)) %>%
    scorequaltrics::clean_dupes(pid_col = 'SID')
```

```
## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion
```

After cleaning duplicates, it's possible that there were some responses that have conflicting answers. We can look at those now.

Now we can see which items had conflicting responses on duplicate questionnaires.

```
#Check that dropped values weren't ambiguous
tds1_wave1_long_pre_recoded_nodupes %>%
    filter(dropped) %>%
    group_by(SID, item) %>%
    filter(!all(length(unlist(old.value)) < 1)) %>%
    mutate(old.value = paste(old.value, collaps = ' ')) %>%
    knitr::kable(caption = "Pre questionnaire dupes")
```

Table: Pre questionnaire dupes

SID item value survey_name old.value dropped —- —— —— ———— ———- ———

```
tds1_wave1_long_prepost_recoded_nodupes %>%
    filter(dropped) %>%
    group_by(SID, item) %>%
    filter(!all(length(unlist(old.value)) < 1)) %>%
    mutate(old.value = paste(old.value, collaps = ' ')) %>%
    knitr::kable(caption = "Pre/Post questionnaire dupes")
```

Table 10: Pre/Post questionnaire dupes

SID	item	value	survey_name	old.value	dropped
336	YRBS_10	NA	TDS1 Session 2 B - Post	c(1, 0)	TRUE
348	BIS_11	NA	TDS1 Session 2 A - Post	c(1, 2)	TRUE
348	BIS_12	NA	TDS1 Session 2 A - Post	c(2, 1)	TRUE
348	BIS_13	NA	TDS1 Session 2 A - Post	c(2, 1)	TRUE
348	BIS_15	NA	TDS1 Session 2 A - Post	c(2, 4)	TRUE
348	BIS_3	NA	TDS1 Session 2 A - Post	c(3, 1)	TRUE
348	BIS_4	NA	TDS1 Session 2 A - Post	c(2, 1)	TRUE
348	BIS_5	NA	TDS1 Session 2 A - Post	c(4, 3)	TRUE
348	BIS_9	NA	TDS1 Session 2 A - Post	c(1, 3)	TRUE
348	SSS_3	NA	TDS1 Session 2 A - Post	c(1, 0)	TRUE
348	SSS_6	NA	TDS1 Session 2 A - Post	c(0, 1)	TRUE
362	BIS_8	NA	TDS1 Session 2 A - Post	c(4, 3)	TRUE
362	SSS_8	NA	TDS1 Session 2 A - Post	c(0, 1)	TRUE
372	YRBS_10	NA	TDS1 Session 2 B - Post	c(1, 0)	TRUE

Scoring

```
tds1_wave1_scored_pre <- scorequaltrics::score_step_one_and_two(
    tds1_wave1_long_pre_recoded_nodupes,
    tds1_wave1_scoring_data_long_pre)
tds1_wave1_scored_prepost <- scorequaltrics::score_step_one_and_two(
    tds1_wave1_long_prepost_recoded_nodupes,
    tds1_wave1_scoring_data_long)

tds1_wave1_scored <- bind_rows(tds1_wave1_scored_prepost, tds1_wave1_scored_pre)

tds1_wave1_scored_pdss <- scorequaltrics::score_pdss(tds1_wave1_long_prepost_recoded_nodupes,
                          gender_mix = pdss_gender_mix,
                          gendercode = pdss_gender_code)
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
```

```
## coercing into character vector
```

```
tds1_wave1_scored %>%
    ungroup() %>%
    distinct(scale_name) %>%
    knitr::kable()
```

scale_name
ACE
BFNE
BIS-15
Brief SCARED
BSSS
CARE-R Expected Involvement
CARE-R Social
CARE-R Willingness to Engage
CBCL
CES-DC
MSSSS
NTS
NTS, Cyberball 1
NTS, Cyberball 2
PAL-2
PDS
PEQ-R
SPSRQ-S
UPPS-P
YRBS
RPI_part2
RSQMod_part2
CARE-R_Past_Frequency
RSQ_2
RSQ1_part2

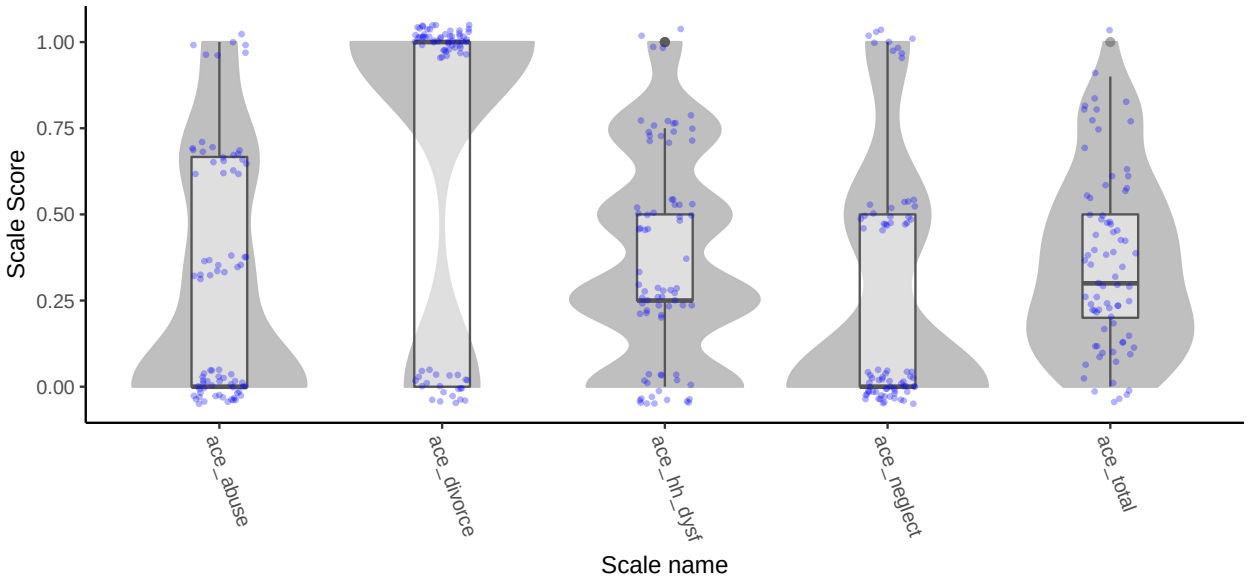
ACE

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^ACE$',
                                  type = 'score')
```

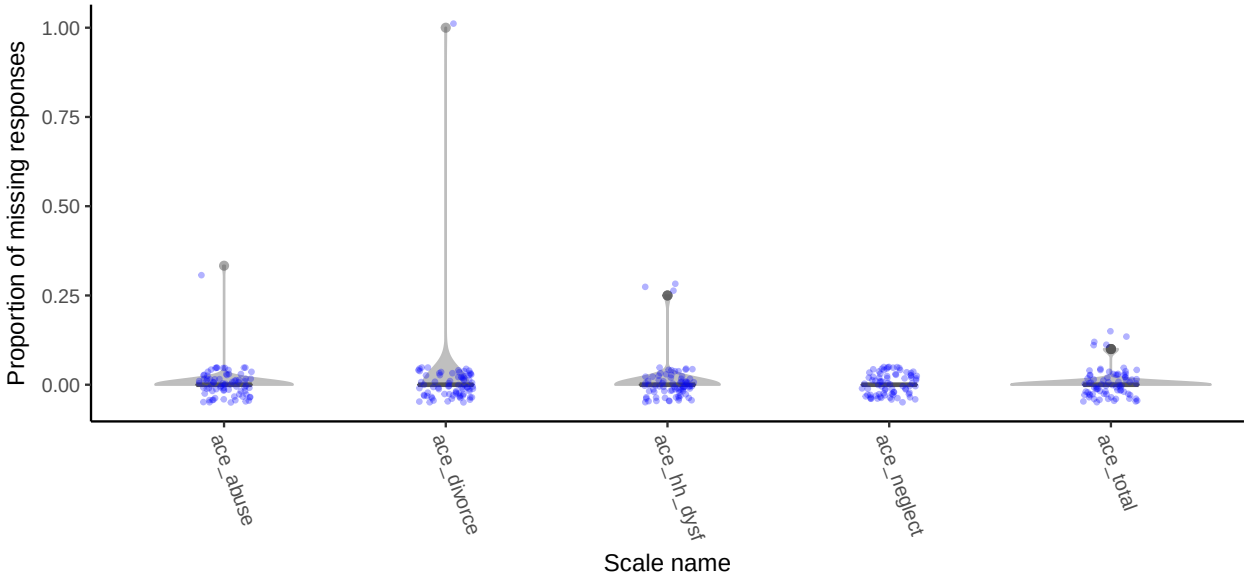
```
## Warning: Removed 1 rows containing non-finite values (stat_ydensity).
```

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

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



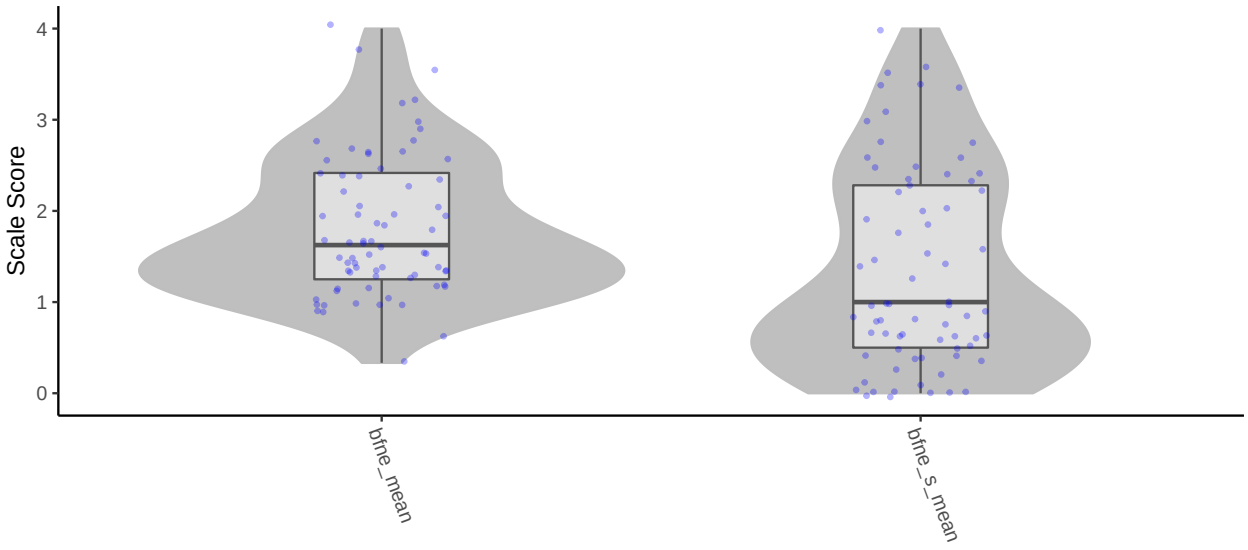

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^ACE$',
                                  type = 'p_missing')
```



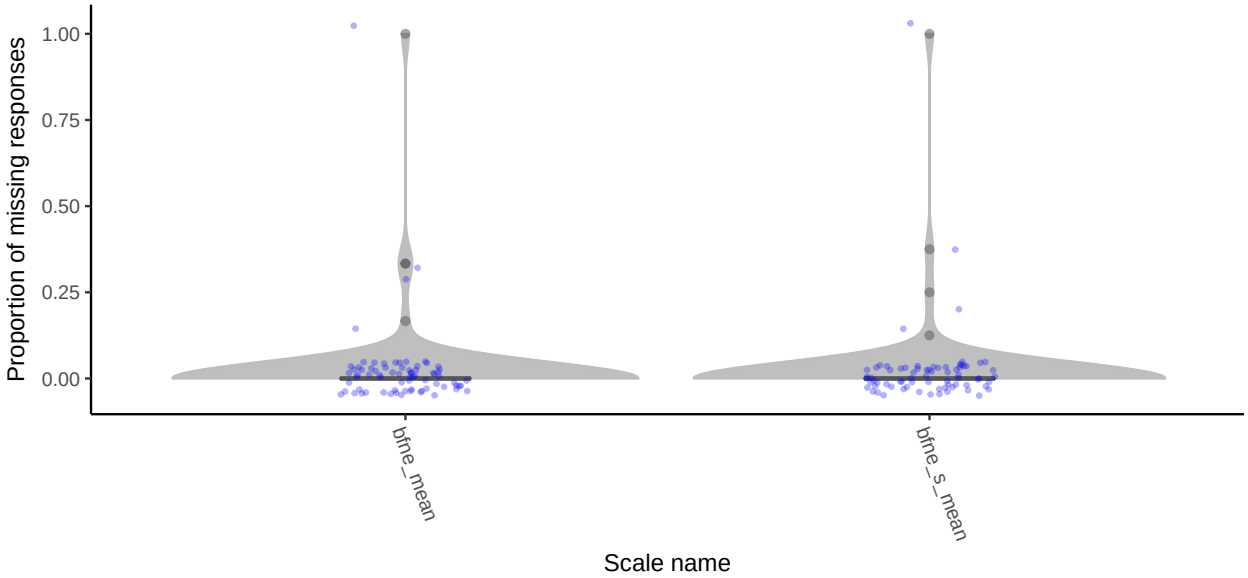
BFNE

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^BFNE$',
                                  type = 'score')
```

```
## Warning: Removed 2 rows containing non-finite values (stat_ydensity).
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
## Warning: Removed 2 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^BFNE$',
                                  type = 'p_missing')
```

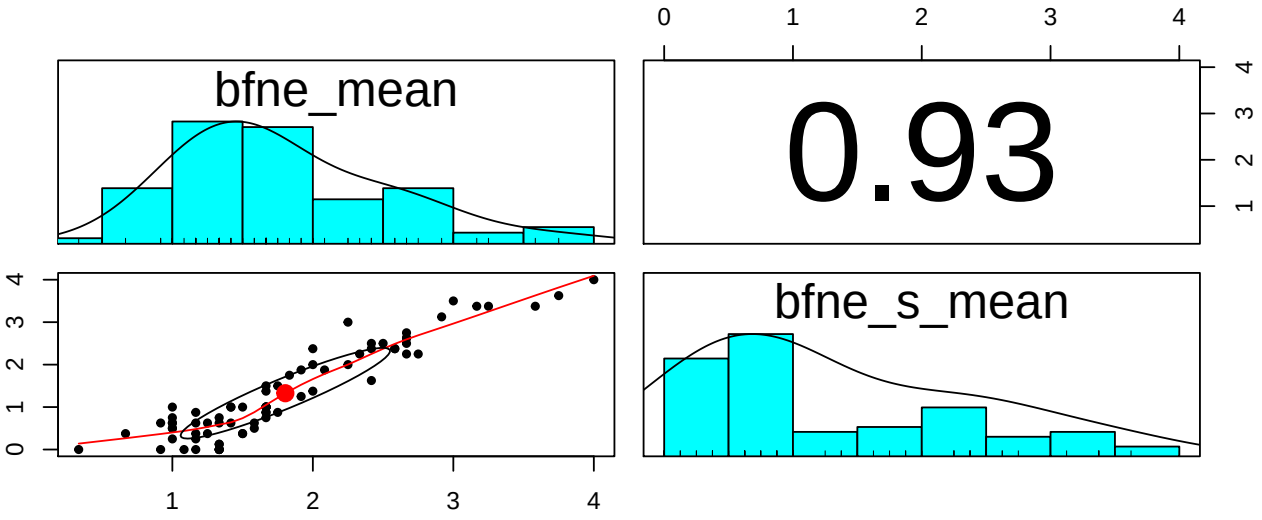



```
tds1w1_BFNERubric <- tds1_wave1_scoring_data_long %>%
    filter(scale_name == 'BFNE')

tds1_wave1_bfne_psych <- scorequaltrics::score_questionnaire(
    tds1_wave1_long_prepost_recoded_nodupes,
    tds1w1_BFNERubric,
    psych = T)
print(tds1_wave1_bfne_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       bfne_mean bfne_s_mean
```

```
## alpha      0.84        0.94
##
## Standard errors of unstandardized Alpha:
##       bfne_mean bfne_s_mean
## ASE        0.04       0.032
##
## Average item correlation:
##           bfne_mean bfne_s_mean
## average.r       0.3        0.65
##
##  Guttman 6* reliability:
##          bfne_mean bfne_s_mean
## Lambda.6      0.91        0.95
##
## Signal/Noise based upon av.r :
##              bfne_mean bfne_s_mean
## Signal/Noise       5.1          15
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##             bfne_mean bfne_s_mean
## bfne_mean        0.84        1.05
## bfne_s_mean      0.93        0.94
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

```
pairs.panels(tds1_wave1_bfne_psych$scores)
```

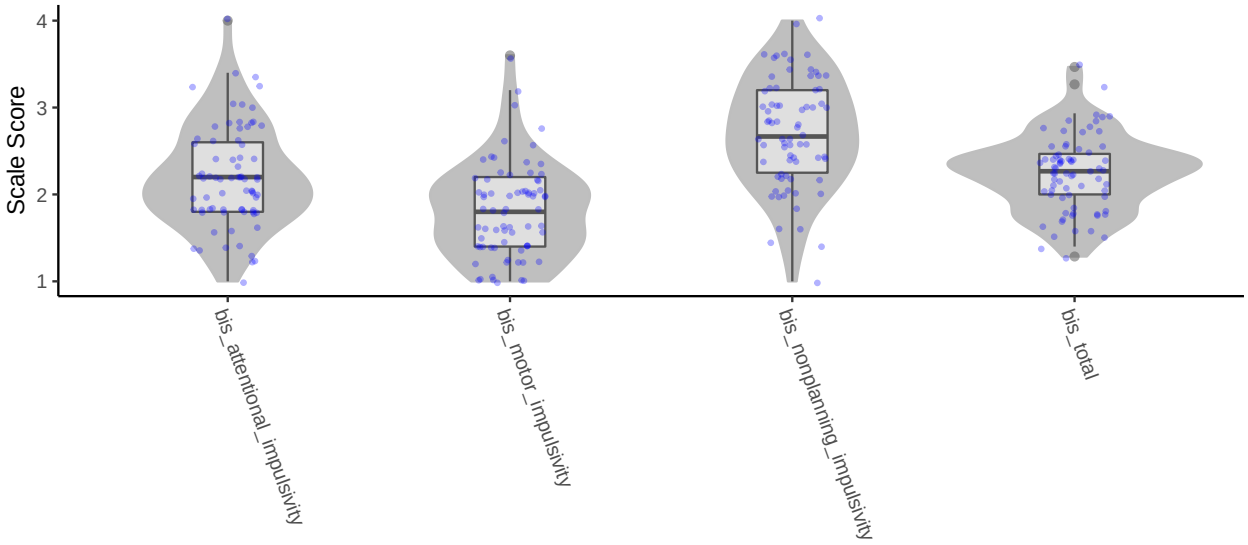


BIS-15

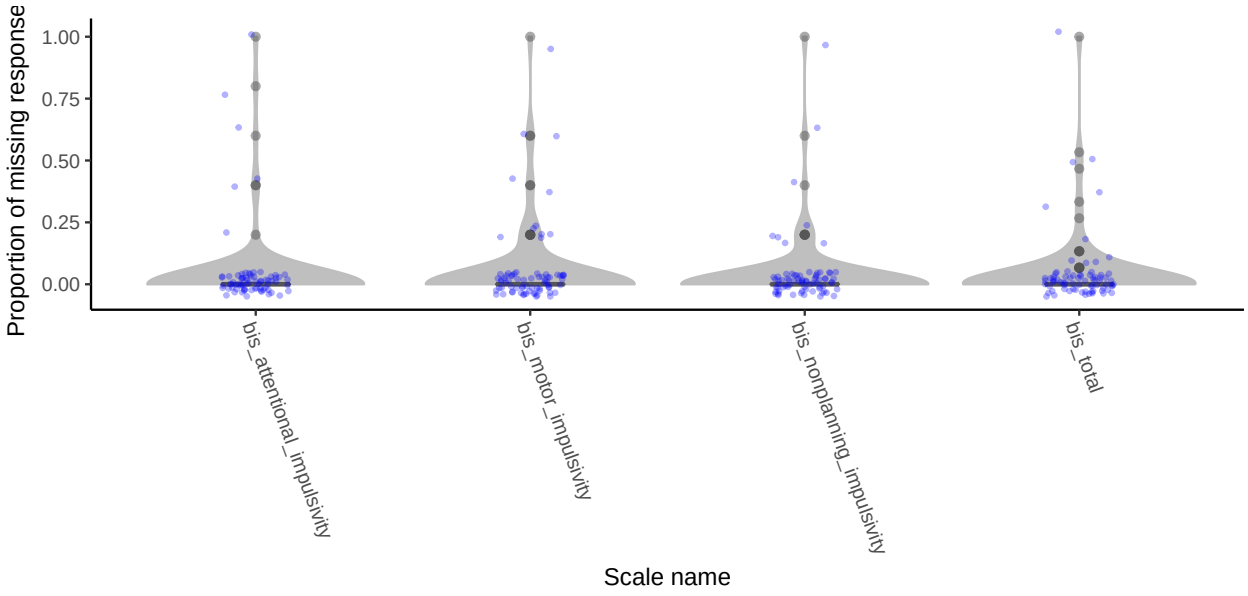
```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^BIS-15$',
                                  type = 'score')
```

```
## Warning: Removed 4 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 4 rows containing non-finite values (stat_boxplot).
## Warning: Removed 4 rows containing missing values (geom_point).
```



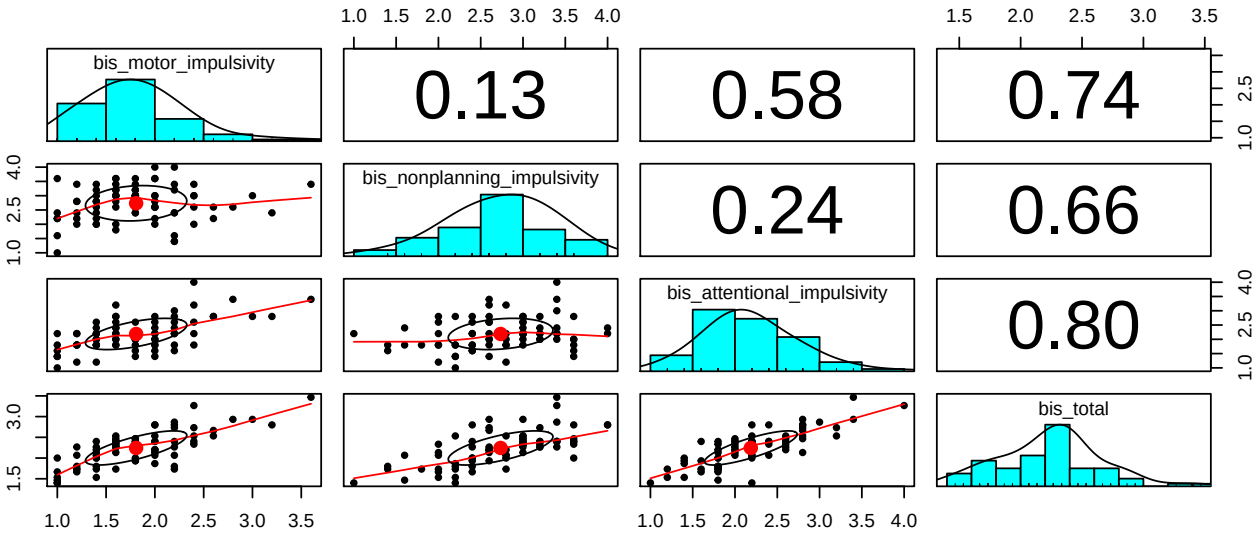

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^BIS-15$',
                                  type = 'p_missing')
```




```
tds1w1_BISRubric <- tds1_wave1_scoring_data_long %>%
    filter(scale_name == 'BIS-15')

tds1_wave1_bis_psych <- scorequaltrics::score_questionnaire(
    tds1_wave1_long_prepost_recoded_nodupes,
    tds1w1_BISRubric,
    psych = T)
print(tds1_wave1_bis_psych)
```

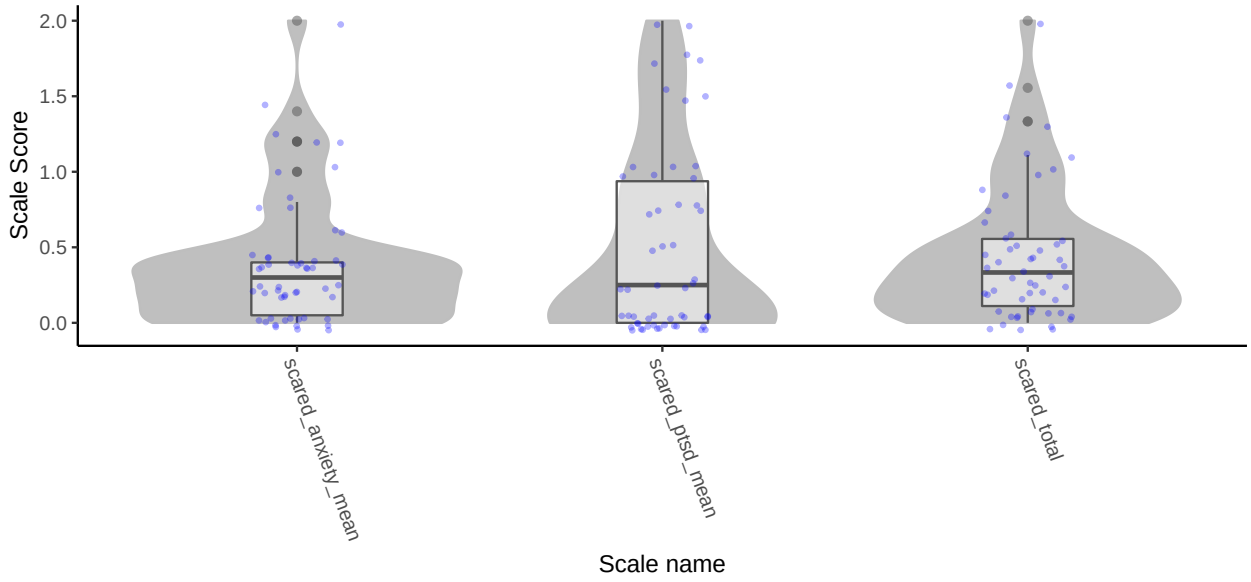
```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       bis_motor_impulsivity bis_nonplanning_impulsivity
## alpha                  0.75                       0.77
##       bis_attentional_impulsivity bis_total
## alpha                        0.69       0.8
##
## Standard errors of unstandardized Alpha:
##       bis_motor_impulsivity bis_nonplanning_impulsivity
## ASE                   0.077                      0.075
##       bis_attentional_impulsivity bis_total
## ASE                         0.086     0.044
##
## Average item correlation:
##           bis_motor_impulsivity bis_nonplanning_impulsivity
## average.r                  0.38                        0.4
##           bis_attentional_impulsivity bis_total
## average.r                        0.31      0.21
##
##  Guttman 6* reliability:
##          bis_motor_impulsivity bis_nonplanning_impulsivity
## Lambda.6                  0.82                       0.81
##          bis_attentional_impulsivity bis_total
## Lambda.6                        0.81      0.88
##
## Signal/Noise based upon av.r :
##              bis_motor_impulsivity bis_nonplanning_impulsivity
## Signal/Noise                   3.1                         3.3
##              bis_attentional_impulsivity bis_total
## Signal/Noise                         2.2         4
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                            bis_motor_impulsivity
## bis_motor_impulsivity                       0.75
## bis_nonplanning_impulsivity                 0.13
## bis_attentional_impulsivity                 0.58
## bis_total                                   0.74
##                            bis_nonplanning_impulsivity
## bis_motor_impulsivity                             0.17
## bis_nonplanning_impulsivity                       0.77
## bis_attentional_impulsivity                       0.24
## bis_total                                         0.66
##                            bis_attentional_impulsivity bis_total
## bis_motor_impulsivity                             0.80      0.96
## bis_nonplanning_impulsivity                       0.33      0.84
## bis_attentional_impulsivity                       0.69      1.08
## bis_total                                         0.80      0.80
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

```
pairs.panels(tds1_wave1_bis_psych$scores)
```

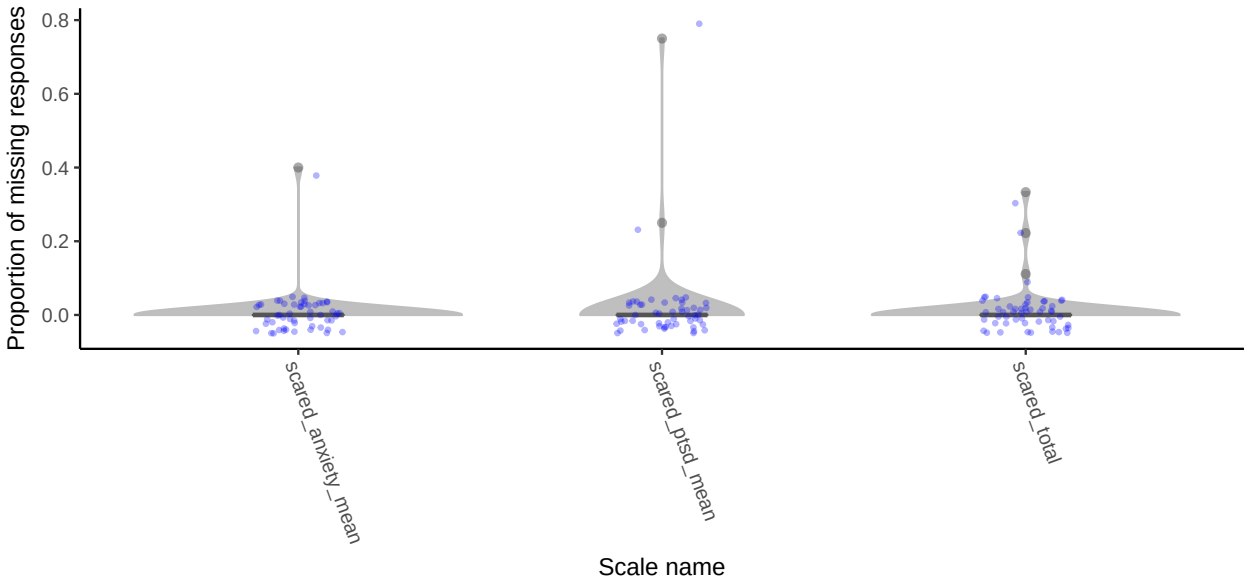


Brief SCARED

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^Brief SCARED$',
                                  type = 'score')
```




```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^Brief SCARED$',
                                  type = 'p_missing')
```

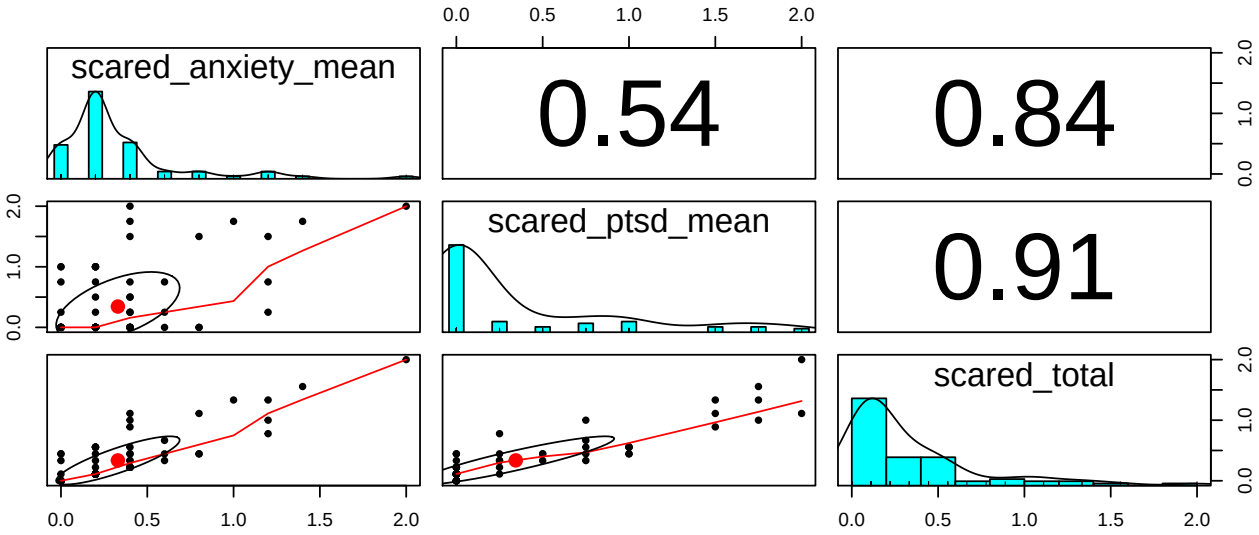



```
tds1w1_BscaredRubric <- tds1_wave1_scoring_data_long %>%
    filter(scale_name == 'Brief SCARED')

tds1_wave1_bscared_psych <- scorequaltrics::score_questionnaire(
    tds1_wave1_long_prepost_recoded_nodupes,
    tds1w1_BscaredRubric,
    psych = T)
print(tds1_wave1_bscared_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       scared_anxiety_mean scared_ptsd_mean scared_total
## alpha                0.71              0.9         0.86
##
## Standard errors of unstandardized Alpha:
##       scared_anxiety_mean scared_ptsd_mean scared_total
## ASE                 0.084            0.065        0.043
##
## Average item correlation:
##           scared_anxiety_mean scared_ptsd_mean scared_total
## average.r                0.33             0.69          0.4
##
##  Guttman 6* reliability:
##          scared_anxiety_mean scared_ptsd_mean scared_total
## Lambda.6                0.75             0.89         0.89
##
## Signal/Noise based upon av.r :
##              scared_anxiety_mean scared_ptsd_mean scared_total
## Signal/Noise                 2.5                9          5.9
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                      scared_anxiety_mean scared_ptsd_mean scared_total
```

```
## scared_anxiety_mean                0.71            0.67         1.08
## scared_ptsd_mean                   0.54            0.90         1.03
## scared_total                       0.84            0.91         0.86
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

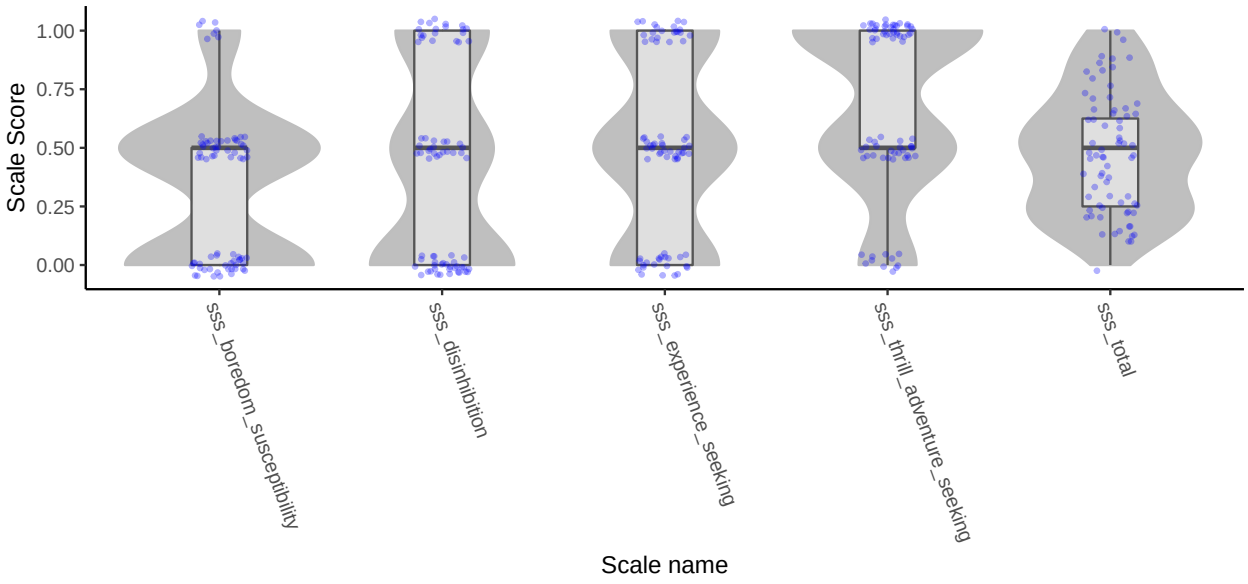
```
pairs.panels(tds1_wave1_bscared_psych$scores)
```



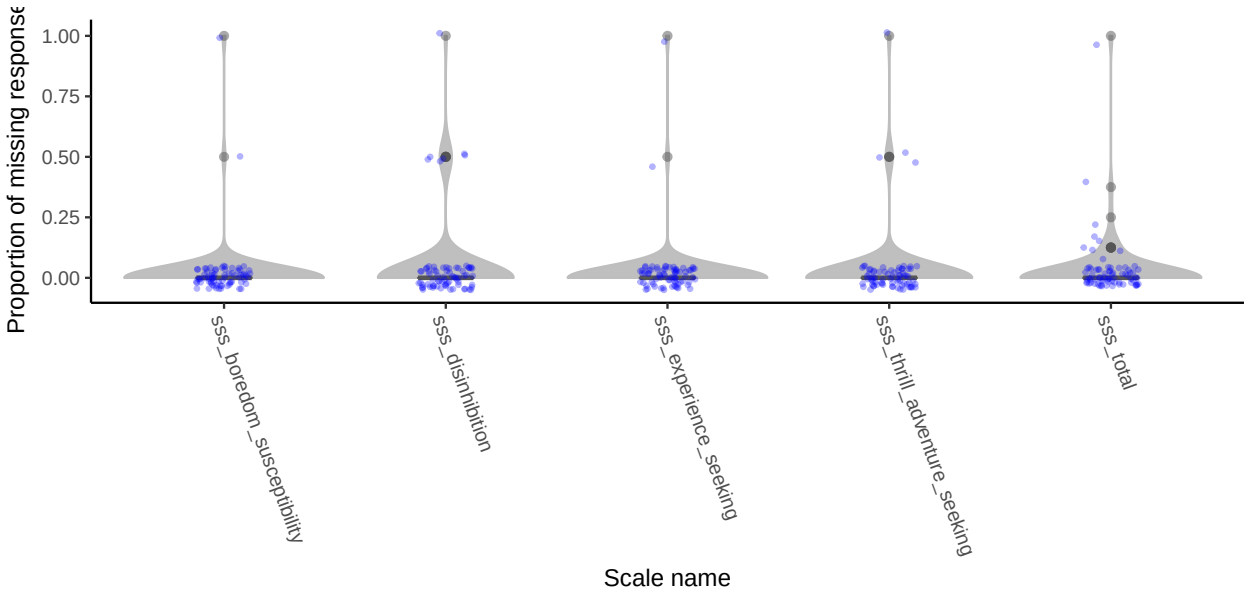
BSSS

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^BSSS$',
                                  type = 'score')
```

```
## Warning: Removed 5 rows containing non-finite values (stat_ydensity).
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
## Warning: Removed 5 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^BSSS$',
                                  type = 'p_missing')
```




```
tds1w1_BSSSRubric <- tds1_wave1_scoring_data_long %>%
    filter(scale_name == 'BSSS')

tds1_wave1_bsss_psych <- scorequaltrics::score_questionnaire(
    tds1_wave1_long_prepost_recoded_nodupes,
    tds1w1_BSSSRubric,
    psych = T)
```

```
## Warning in sqrt(y[i] * y[j]): NaNs produced
## Warning in sqrt(y[i] * y[j]): NaNs produced
## Warning in sqrt(y[i] * y[j]): NaNs produced
```

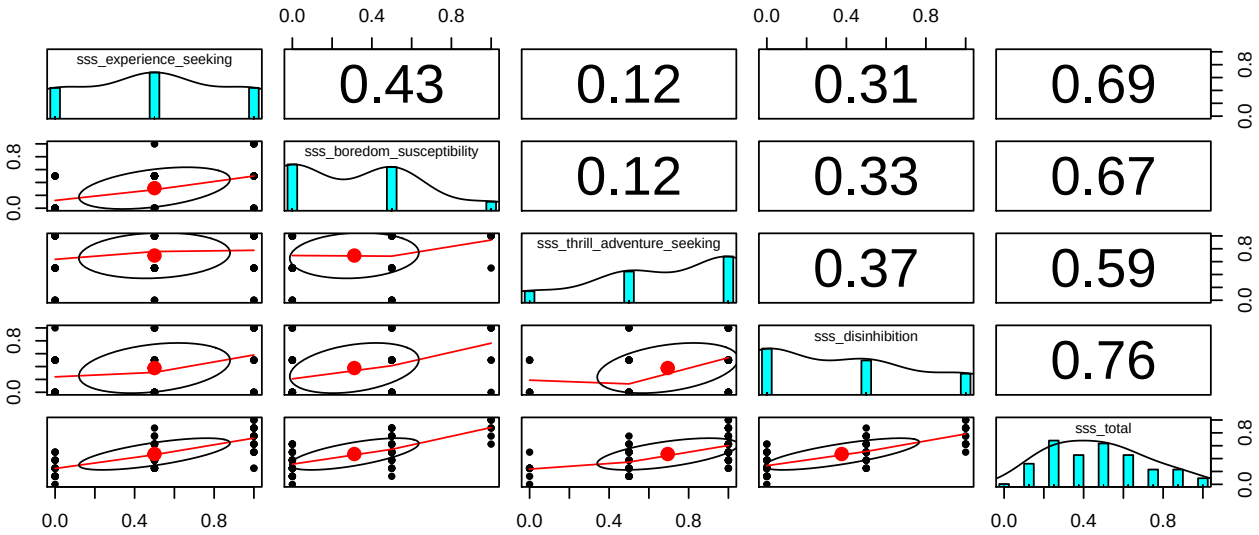
```
## Warning in sqrt(y[i] * y[j]): NaNs produced
```

```
print(tds1_wave1_bsss_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       sss_experience_seeking sss_boredom_susceptibility
## alpha                   0.29                     -0.034
##       sss_thrill_adventure_seeking sss_disinhibition sss_total
## alpha                          0.3              0.44      0.61
##
## Standard errors of unstandardized Alpha:
##       sss_experience_seeking sss_boredom_susceptibility
## ASE                     0.22                      0.23
##       sss_thrill_adventure_seeking sss_disinhibition sss_total
## ASE                           0.22               0.2     0.085
##
## Average item correlation:
##           sss_experience_seeking sss_boredom_susceptibility
## average.r                   0.17                     -0.017
##           sss_thrill_adventure_seeking sss_disinhibition sss_total
## average.r                         0.18              0.28      0.16
##
##  Guttman 6* reliability:
##          sss_experience_seeking sss_boredom_susceptibility
## Lambda.6                    0.3                      0.17
##          sss_thrill_adventure_seeking sss_disinhibition sss_total
## Lambda.6                         0.32               0.4      0.62
##
## Signal/Noise based upon av.r :
##              sss_experience_seeking sss_boredom_susceptibility
## Signal/Noise                    0.4                     -0.033
##              sss_thrill_adventure_seeking sss_disinhibition sss_total
## Signal/Noise                         0.44              0.78       1.5
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                              sss_experience_seeking
## sss_experience_seeking                         0.29
## sss_boredom_susceptibility                     0.43
## sss_thrill_adventure_seeking                   0.12
## sss_disinhibition                              0.31
## sss_total                                      0.69
##                              sss_boredom_susceptibility
## sss_experience_seeking                              NaN
## sss_boredom_susceptibility                       -0.034
## sss_thrill_adventure_seeking                      0.123
## sss_disinhibition                                 0.333
## sss_total                                         0.670
##                              sss_thrill_adventure_seeking
## sss_experience_seeking                               0.41
## sss_boredom_susceptibility                            NaN
```

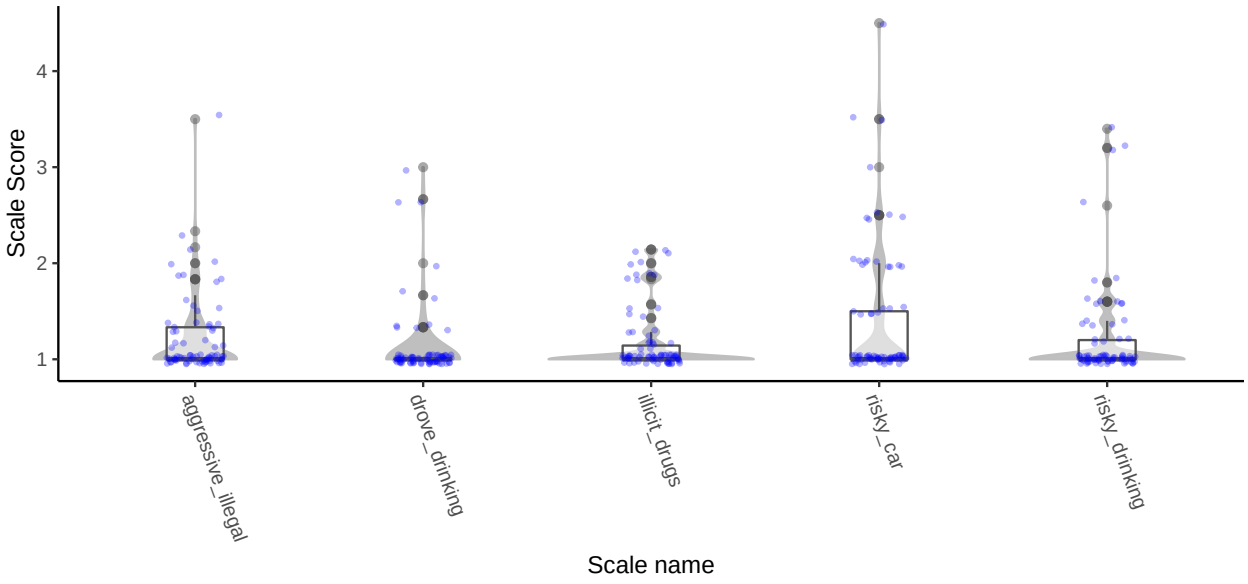
```
## sss_thrill_adventure_seeking                          0.30
## sss_disinhibition                                     0.37
## sss_total                                             0.59
##                                sss_disinhibition sss_total
## sss_experience_seeking                      0.87      1.66
## sss_boredom_susceptibility                   NaN       NaN
## sss_thrill_adventure_seeking                1.00      1.38
## sss_disinhibition                           0.44      1.47
## sss_total                                   0.76      0.61
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

```
pairs.panels(tds1_wave1_bsss_psych$scores)
```

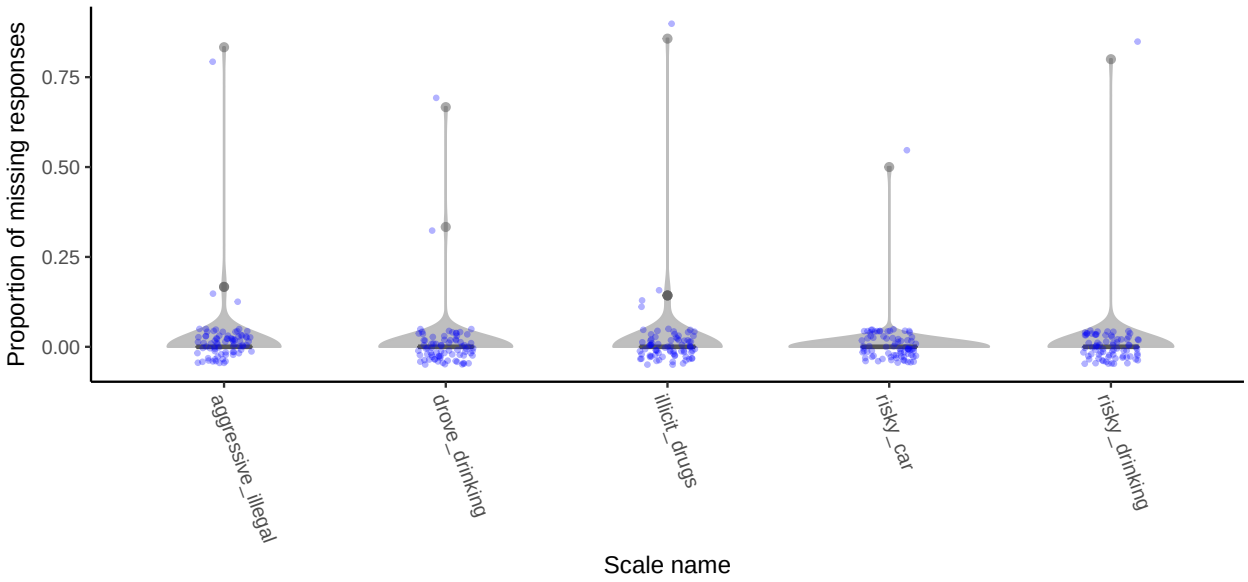


CARE-R Expected Involvement

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^CARE-R Expected Involvement$',
                                  type = 'score')
```



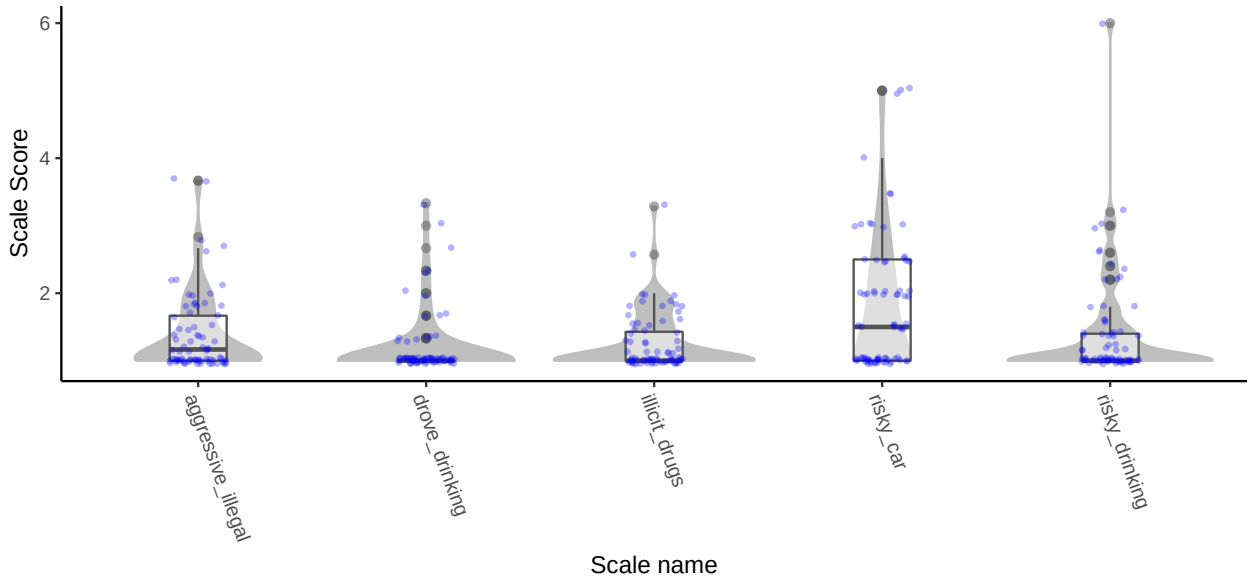

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^CARE-R Expected Involvement$',
                                  type = 'p_missing')
```



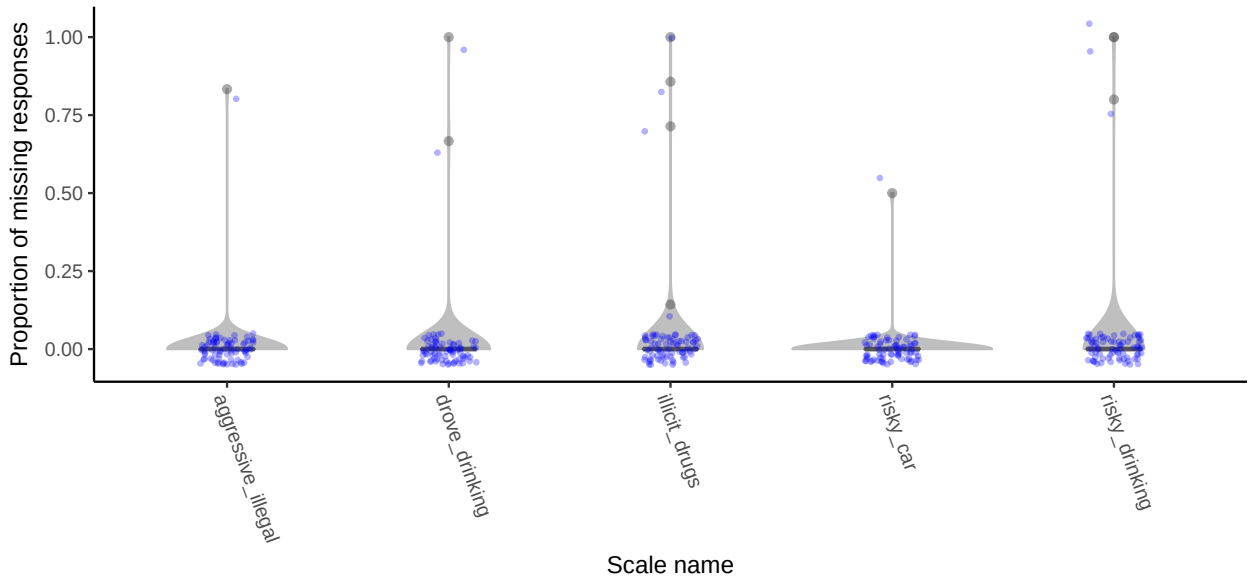
CARE-R Willingness to Engage

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^CARE-R Willingness to Engage$',
                                  type = 'score')
```

```
## Warning: Removed 4 rows containing non-finite values (stat_ydensity).
## Warning: Removed 4 rows containing non-finite values (stat_boxplot).
## Warning: Removed 4 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^CARE-R Willingness to Engage$',
                                  type = 'p_missing')
```



CARE-R Social

Note: Lower scores on "not doing this" items indicate that someone thinks not engaging in the behavior will make people like them *more*, and thus, like the "doing this" items, a lower score corresponds to a downward influence on that behavior.

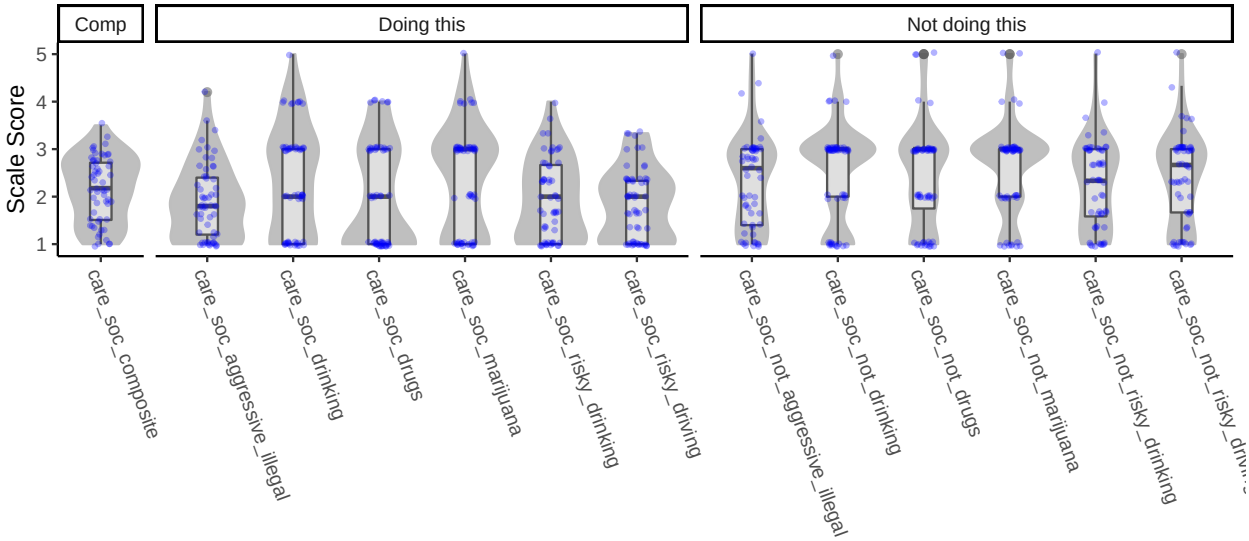
```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^CARE-R Social$',
                                  type = 'score') +
    facet_grid(~factor(grepl('not', scored_scale) - grepl('composite', scored_scale),
                       levels = c(-1, 0, 1),
                       labels = c('Comp', 'Doing this', 'Not doing this')),
```

```
                scales = 'free_x',
                space = 'free_x')
```

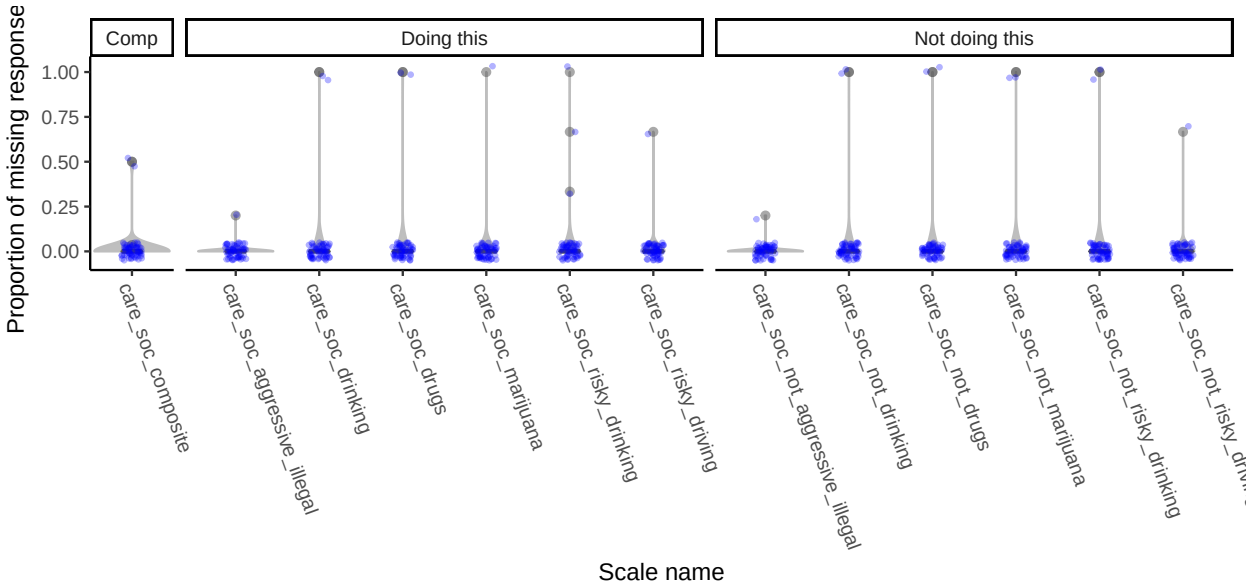
```
## Warning: Removed 14 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 14 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 14 rows containing missing values (geom_point).
```

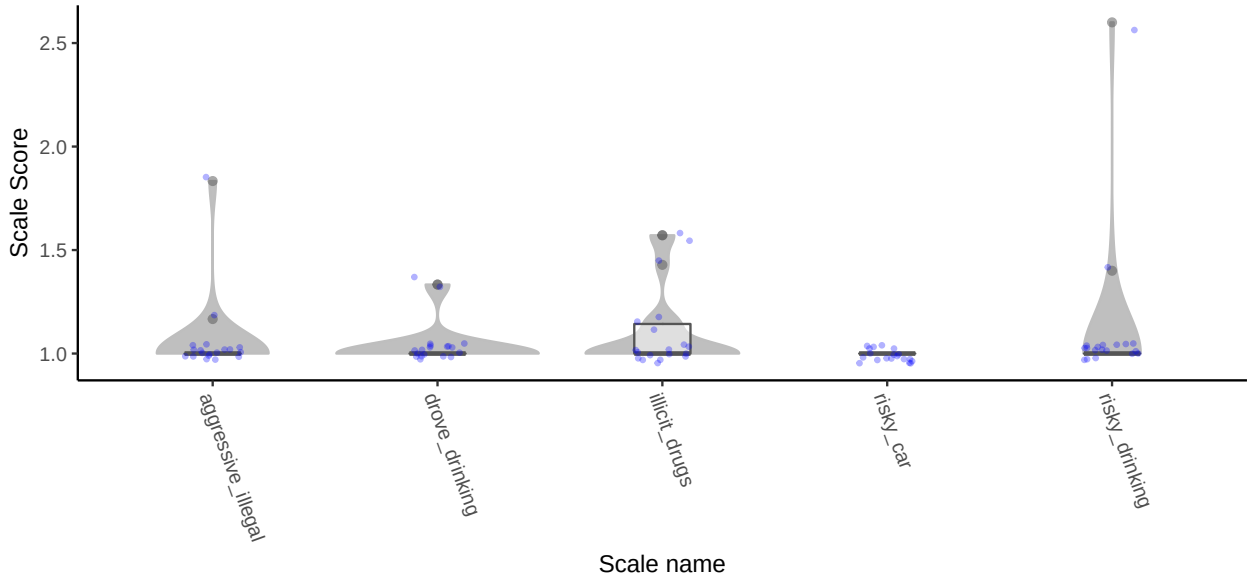



```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^CARE-R Social$',
                                  type = 'p_missing') +
    facet_grid(~factor(grepl('not', scored_scale) - grepl('composite', scored_scale),
                       levels = c(-1, 0, 1),
                       labels = c('Comp', 'Doing this', 'Not doing this')),
               scales = 'free_x',
               space = 'free_x')
```

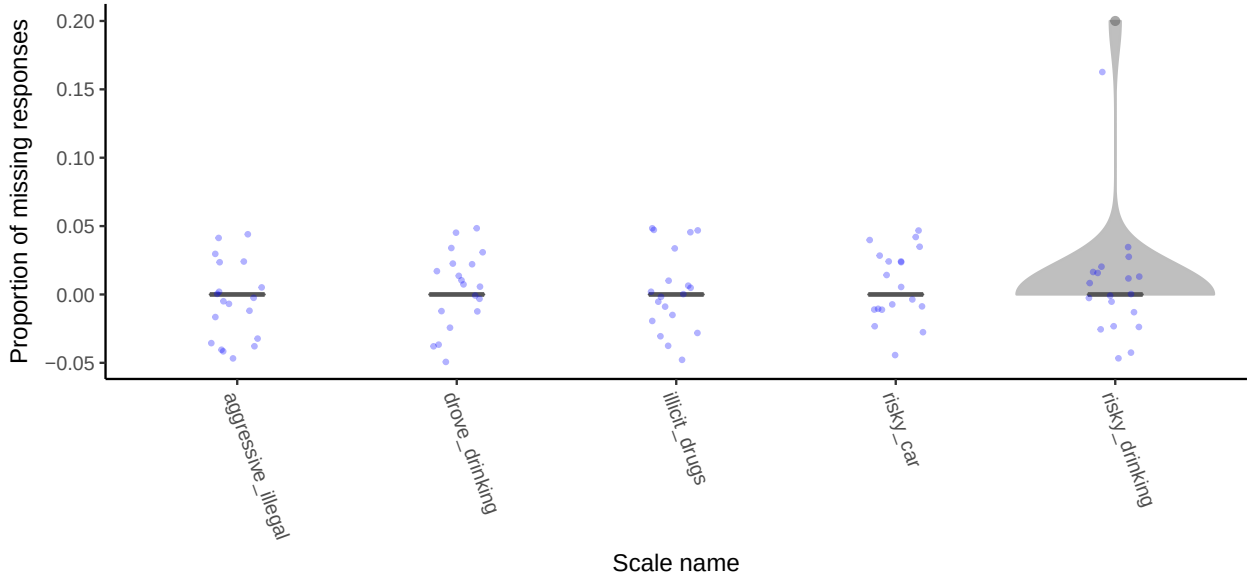


CARE-R Past Frequency

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^CARE-R_Past_Frequency$',
                                  type = 'score')
```



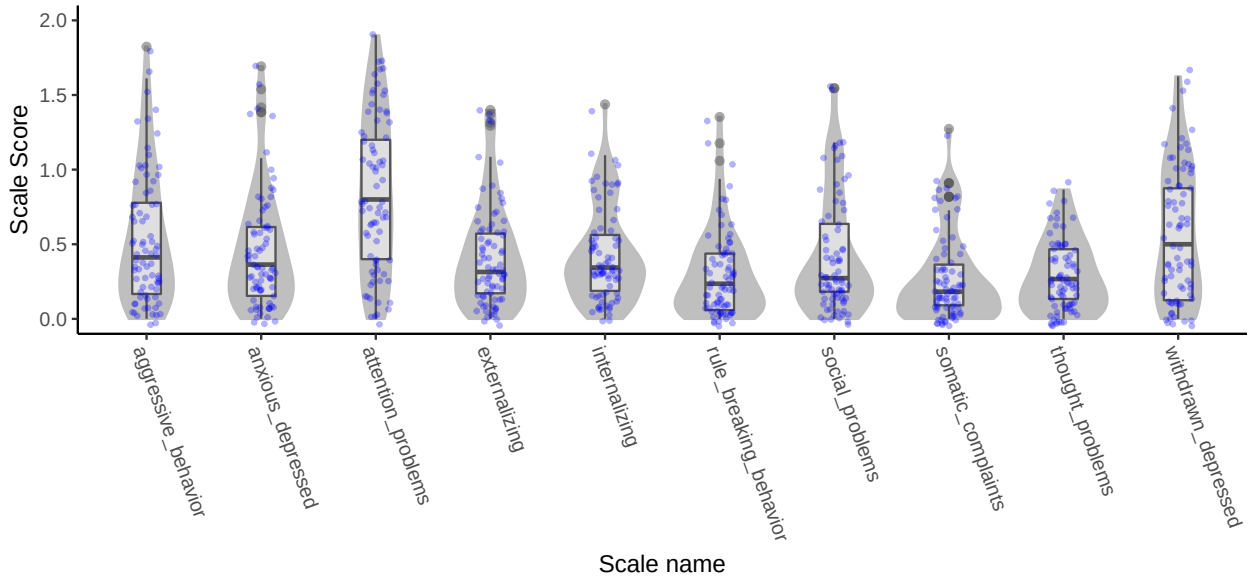

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^CARE-R_Past_Frequency$',
                                  type = 'p_missing')
```



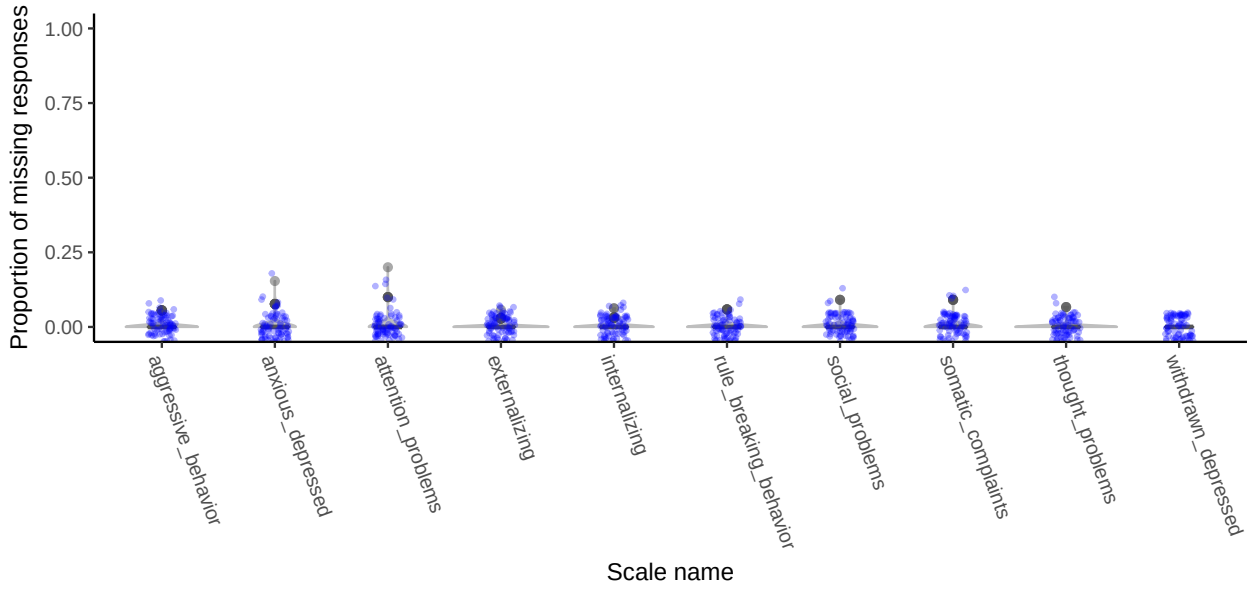
CBCL

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^CBCL$',
```

```
                                type = 'score')+
    coord_cartesian(y = c(0,2))
```

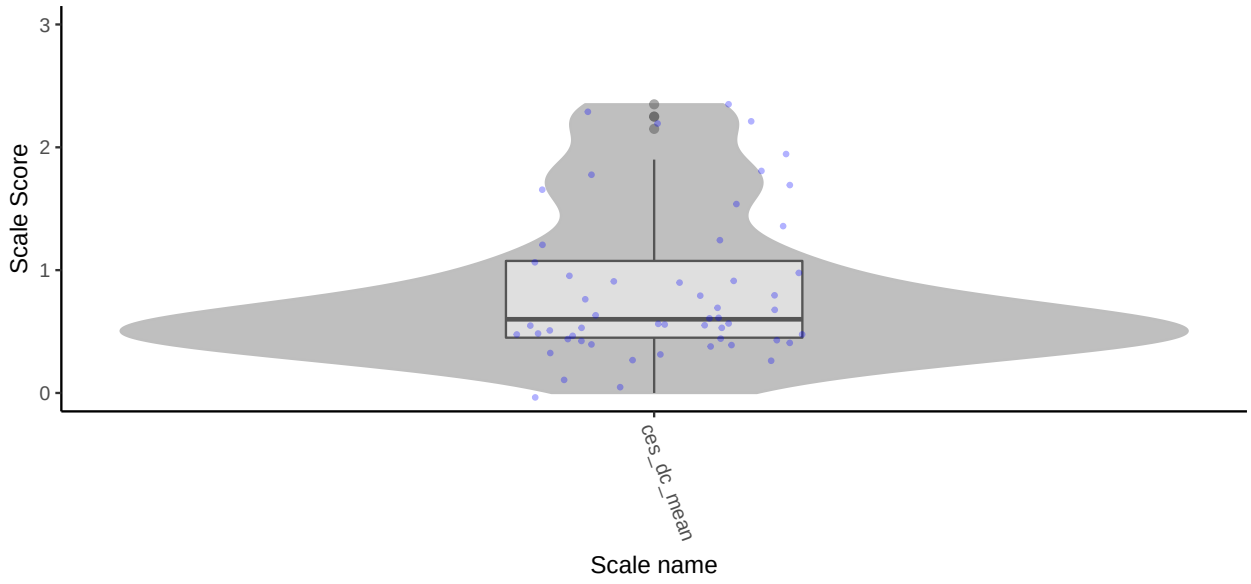



```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                scale_regx = '^CBCL$',
                                type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```

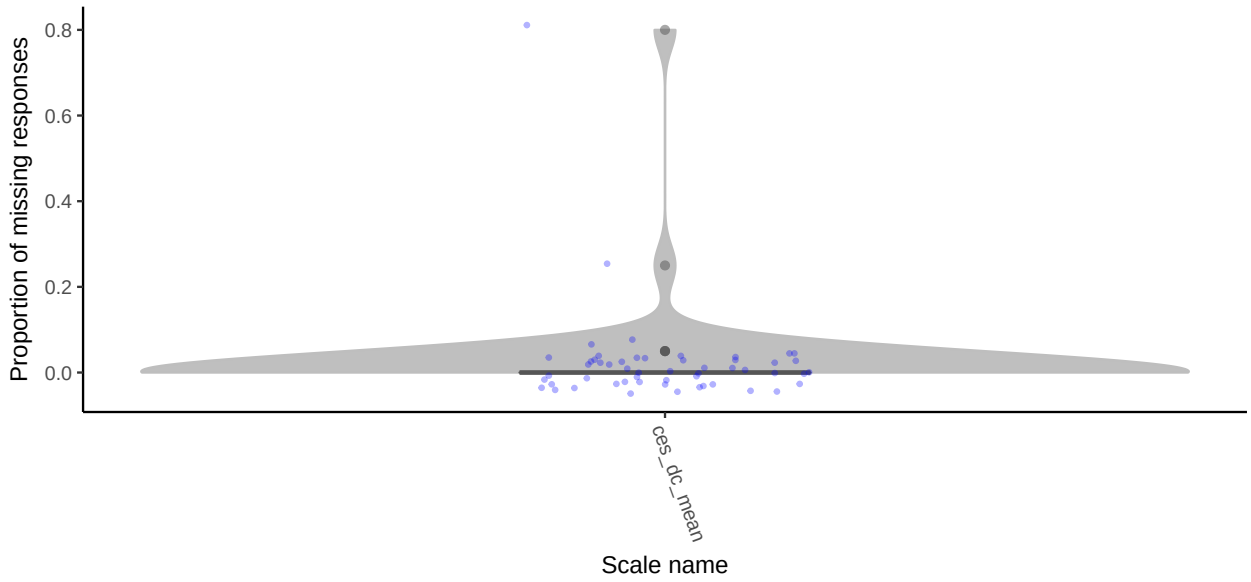


CES-DC

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                scale_regx = '^CES-DC$',
                                type = 'score')+
    coord_cartesian(y = c(0,3))
```



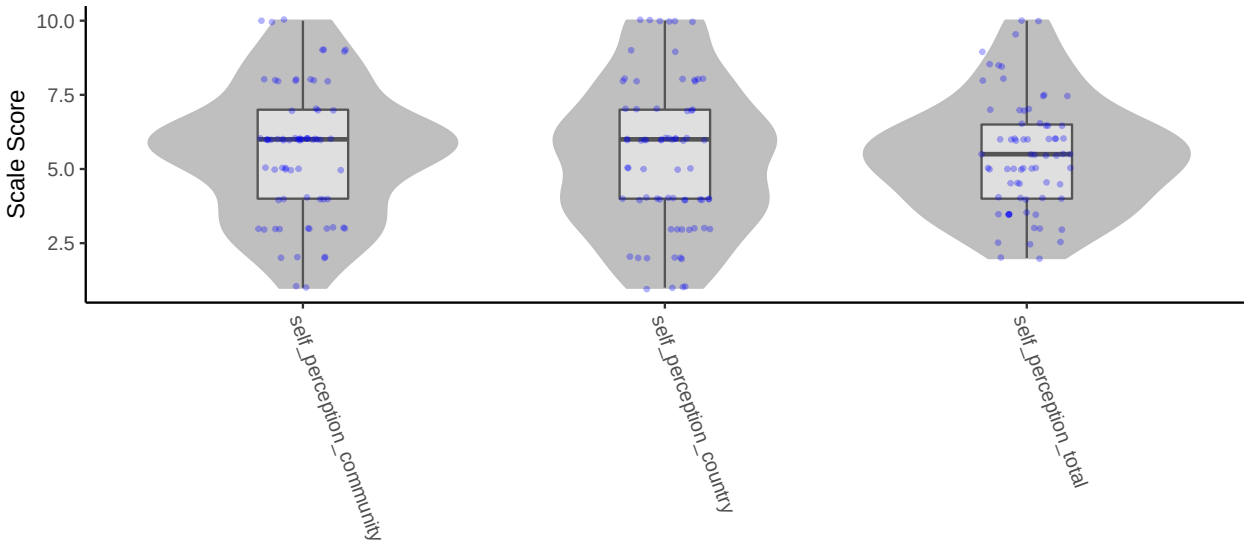

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^CES-DC$',
                                  type = 'p_missing')
```



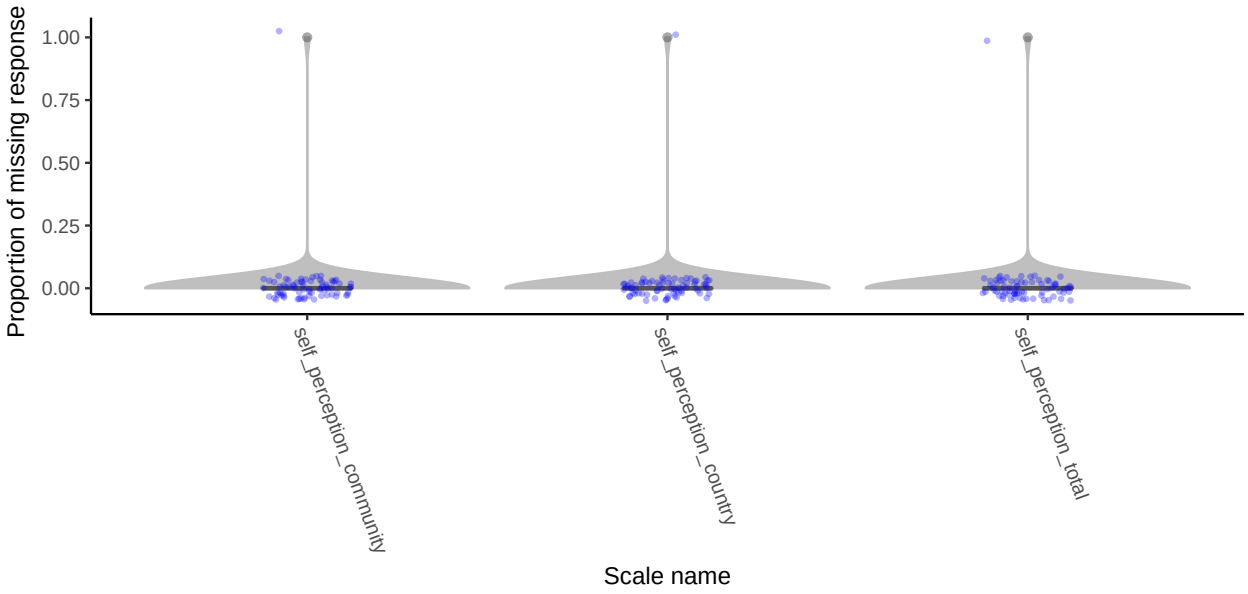
MSSSS

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^MSSSS$',
                                  type = 'score')
```

```
## Warning: Removed 3 rows containing non-finite values (stat_ydensity).
## Warning: Removed 3 rows containing non-finite values (stat_boxplot).
## Warning: Removed 3 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^MSSSS$',
                                  type = 'p_missing')
```



NTS

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^NTS$',
                                  type = 'score')

scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^NTS$',
                                  type = 'p_missing')

tds1w1_NTSRubric <- tds1_wave1_scoring_data_long %>%
```

```
    filter(scale_name == 'NTS')

tds1_wave1_nts_psych <- scorequaltrics::score_questionnaire(
    tds1_wave1_long_prepost_recoded_nodupes,
    tds1w1_NTSRubric,
    psych = T)
print(tds1_wave1_nts_psych)
pairs.panels(tds1_wave1_nts_psych$scores)
```

NTS Cyberball 1

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^NTS, Cyberball 1$',
                                  type = 'score')

scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^NTS, Cyberball 1$',
                                  type = 'p_missing')

tds1w1_NTSc1Rubric <- tds1_wave1_scoring_data_long %>%
    filter(scale_name == 'NTS, Cyberball 1')

tds1_wave1_ntsc1_psych <- scorequaltrics::score_questionnaire(
    tds1_wave1_long_prepost_recoded_nodupes,
    tds1w1_NTSc1Rubric,
    psych = T)
print(tds1_wave1_ntsc1_psych)
pairs.panels(tds1_wave1_ntsc1_psych$scores)
```

NTS Cyberball 2

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^NTS, Cyberball 2$',
                                  type = 'score')

scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^NTS, Cyberball 2$',
                                  type = 'p_missing')

tds1w1_NTSc2Rubric <- tds1_wave1_scoring_data_long %>%
    filter(scale_name == 'NTS, Cyberball 2')

tds1_wave1_ntsc2_psych <- scorequaltrics::score_questionnaire(
    tds1_wave1_long_prepost_recoded_nodupes,
    tds1w1_NTSc2Rubric,
    psych = T)
print(tds1_wave1_ntsc2_psych)
pairs.panels(tds1_wave1_ntsc2_psych$scores)
```

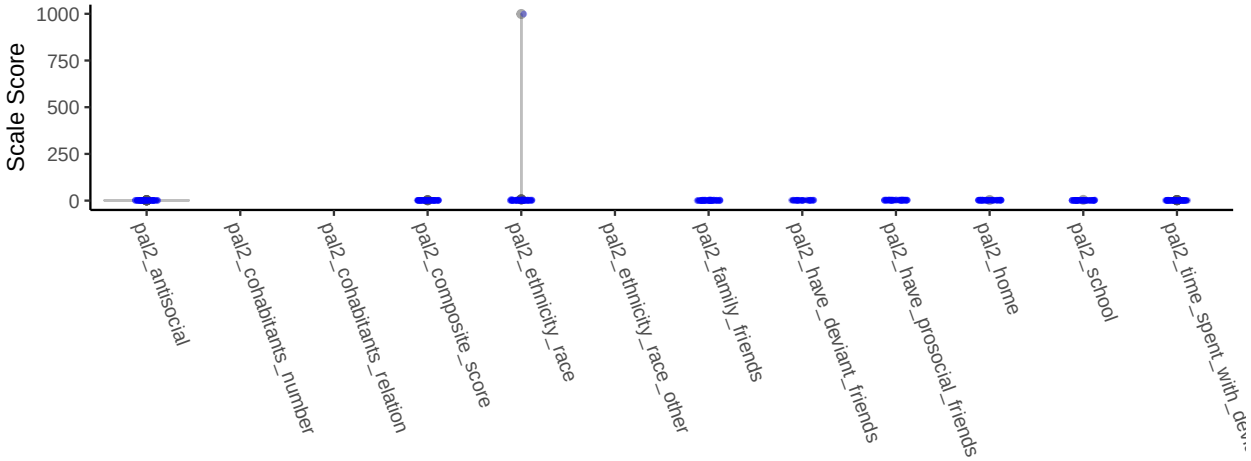
PAL-2

```
scorequaltrics::plot_scored_scale(filter(tds1_wave1_scored, scored_scale != 'pal2_age'),
                                  scale_regx = '^PAL-2$',
                                  type = 'score')
```

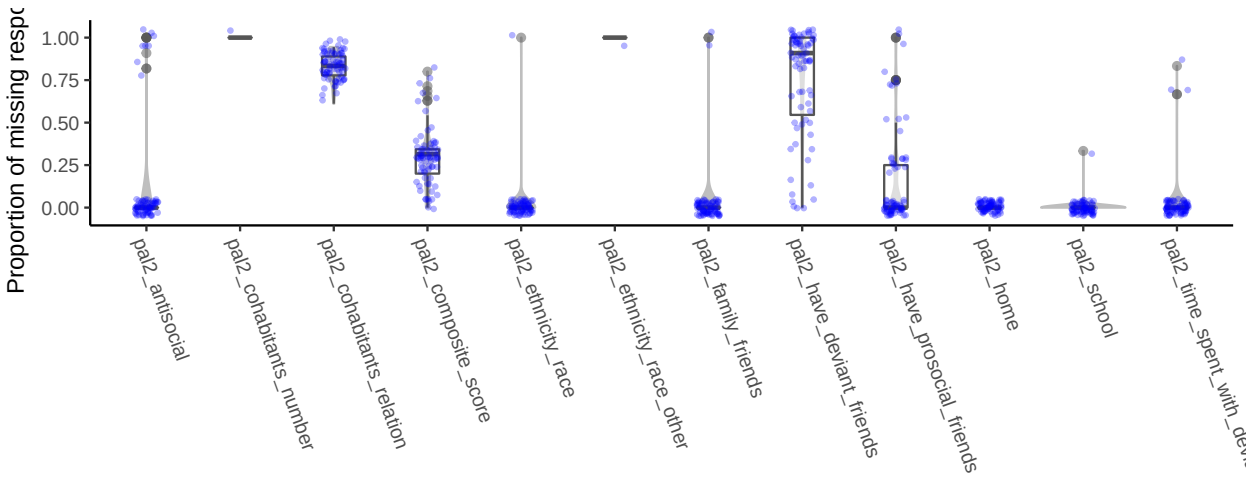
```
## Warning: Removed 114 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 114 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 114 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(filter(tds1_wave1_scored, scored_scale != 'pal2_age'),
                                  scale_regx = '^PAL-2$',
                                  type = 'p_missing')
```



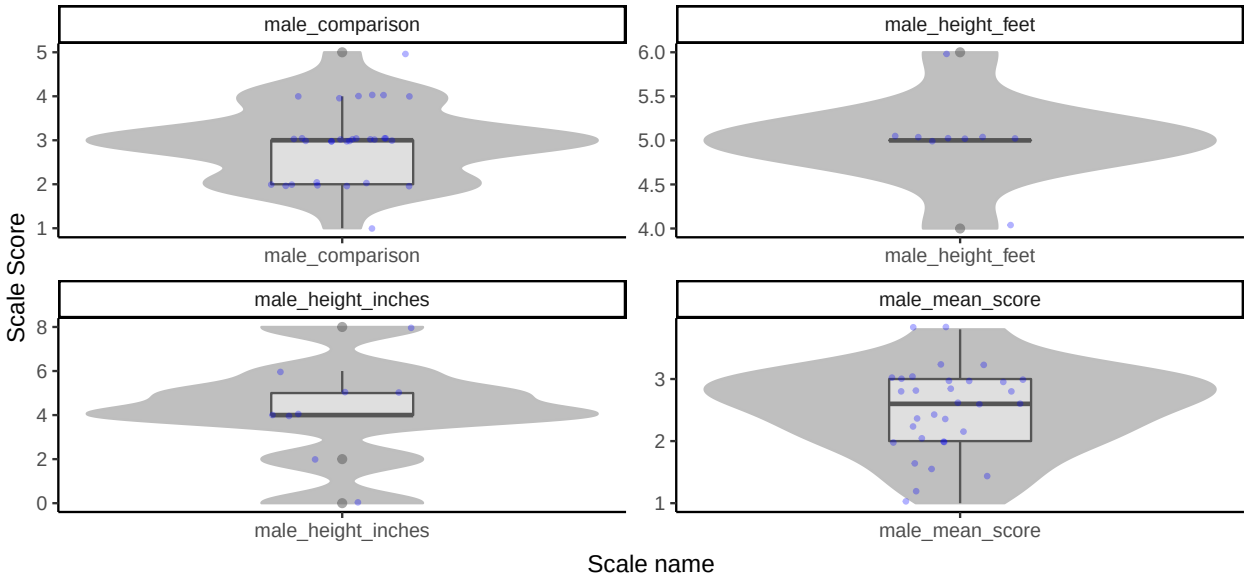
PDS

```
#We need a way to separate female from male responses
male_sids <- tds1_wave1_scored %>%
    filter(scored_scale == 'pds_gender', score =='0') %>%
    ungroup() %>%
    select(SID)

female_sids <- tds1_wave1_scored %>%
    filter(scored_scale == 'pds_gender', score =='1') %>%
    ungroup() %>%
    select(SID)

scorequaltrics::plot_scored_scale(filter(tds1_wave1_scored,
                                         SID %in% male_sids$SID,
                                         grepl('^male', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'score') +
    facet_wrap(~scored_scale, scales = 'free') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

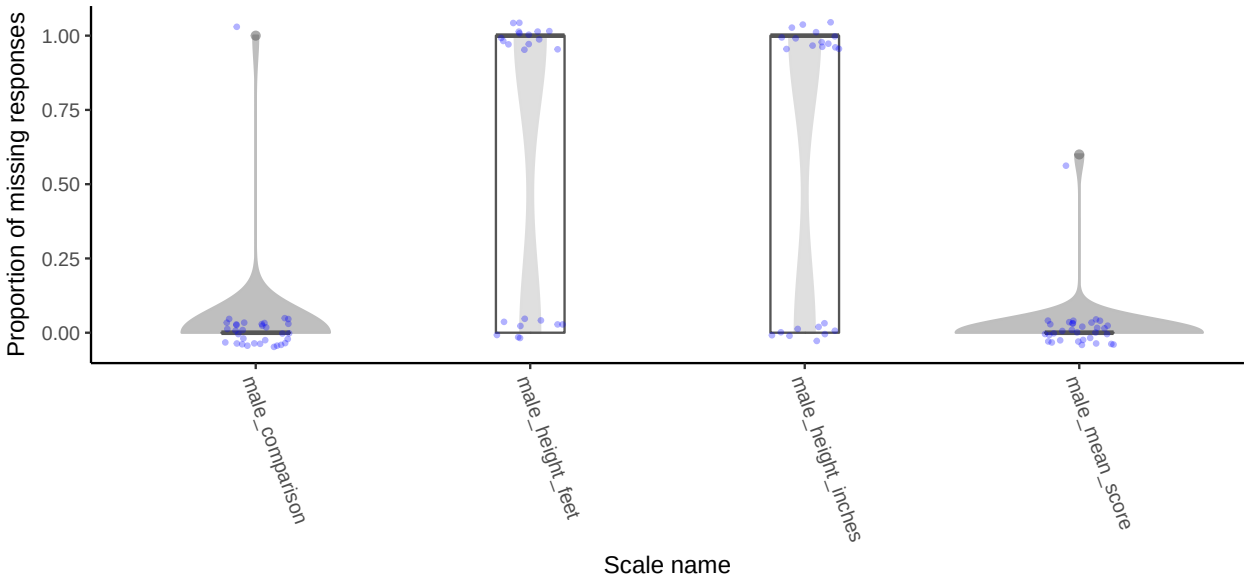
```
## Warning: Removed 29 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 29 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 29 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(filter(tds1_wave1_scored,
                                         SID %in% male_sids$SID,
                                         grepl('^male', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'p_missing')
```

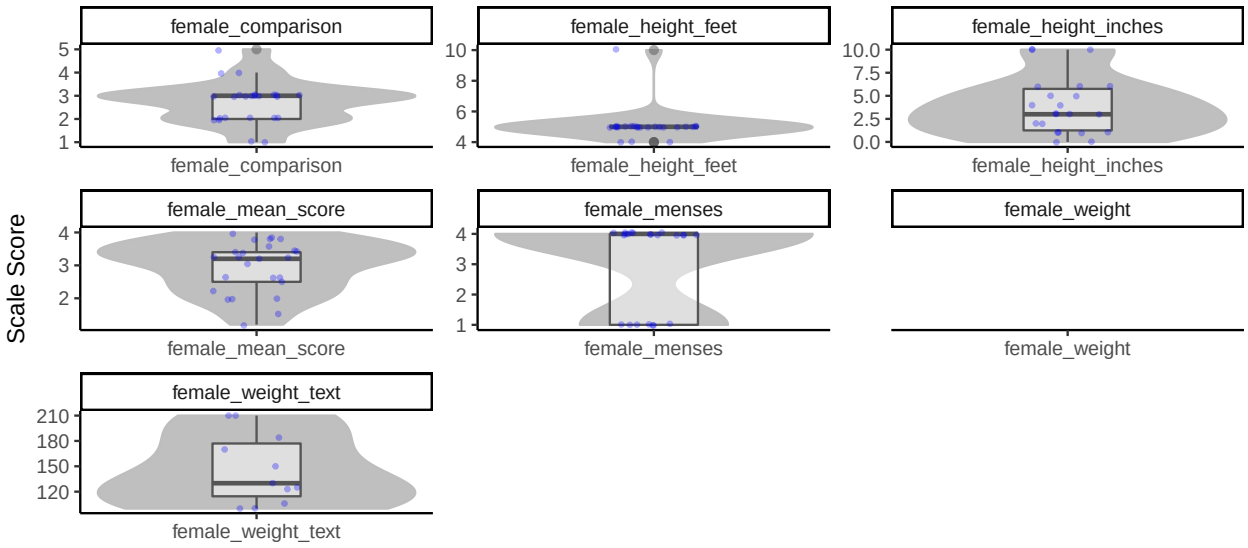



```
scorequaltrics::plot_scored_scale(filter(tds1_wave1_scored,
                                         SID %in% female_sids$SID,
                                         grepl('^female', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'score') +
    facet_wrap(~scored_scale, scales = 'free') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

```
## Warning: Removed 22 rows containing non-finite values (stat_ydensity).
```

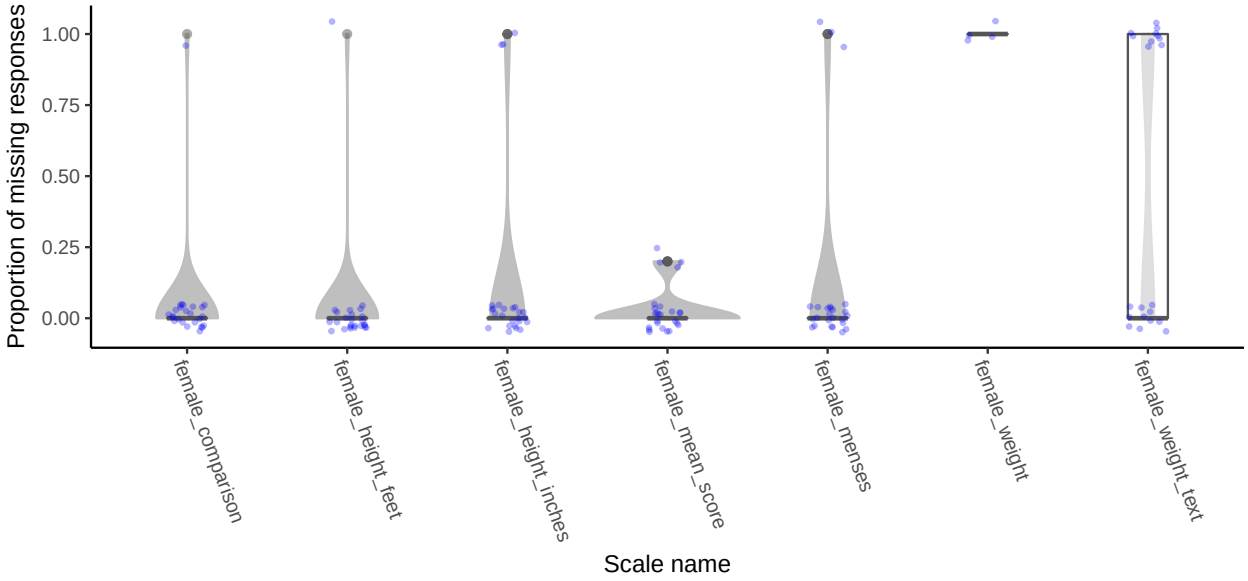
```
## Warning: Removed 22 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 22 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(filter(tds1_wave1_scored,
                                         SID %in% female_sids$SID,
                                         grepl('^female', scored_scale)),
                                  scale_regx = '^PDS$',
```

```
                                type = 'p_missing')
```



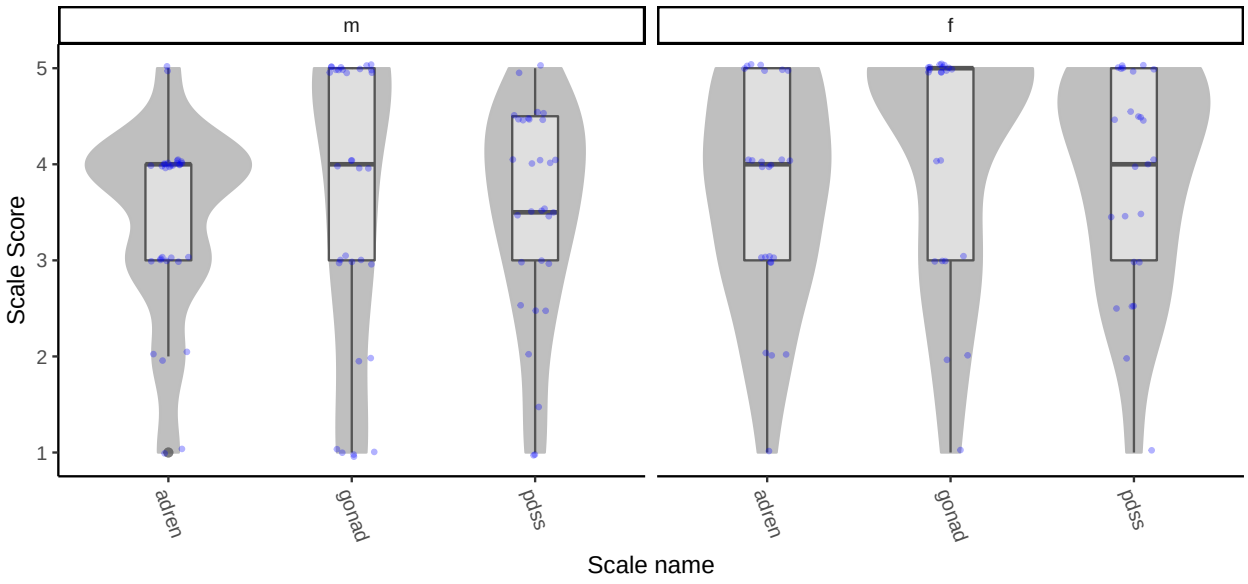
PDSS

```
tds1_wave1_scored_pdss_gender <- tds1_wave1_scored_pdss %>%
    filter(scored_scale == 'gender') %>%
    spread(scored_scale, score) %>%
    select(SID, gender)

tds1_wave1_scored_pdss_gender_wide <- tds1_wave1_scored_pdss %>%
    filter(scored_scale != 'gender') %>%
    left_join(tds1_wave1_scored_pdss_gender)

scorequaltrics::plot_scored_scale(tds1_wave1_scored_pdss_gender_wide,
                                  scale_regx = '^PDSS$',
                                  type = 'score') +
    facet_wrap(~factor(gender,
                       levels = pdss_gender_code,
                       labels = names(pdss_gender_code)))
```

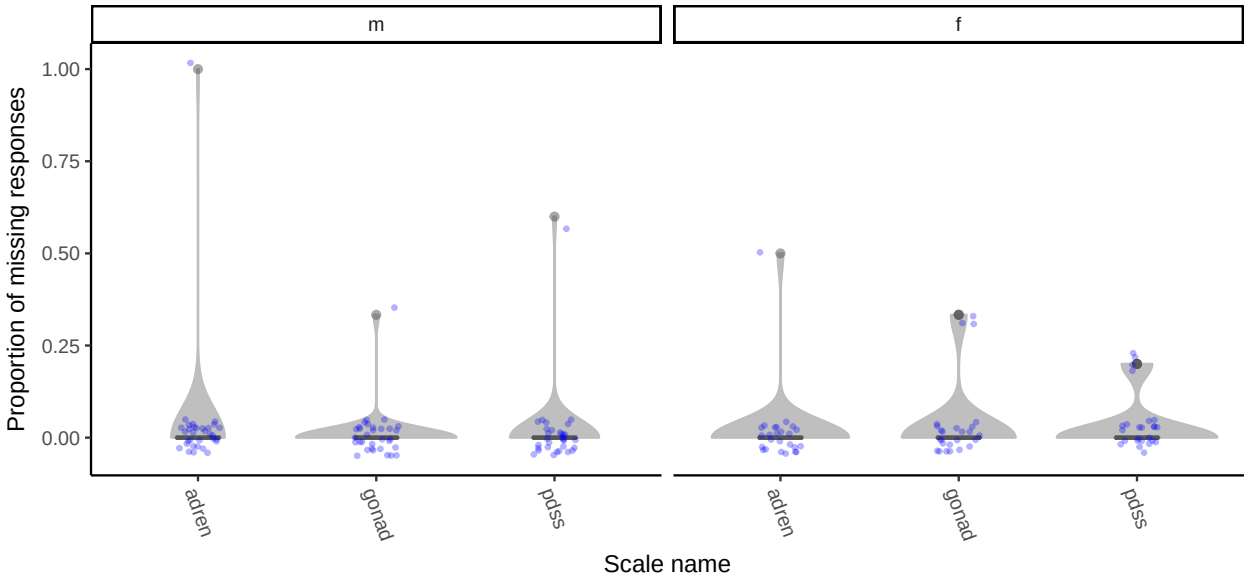
```
## Warning: Removed 6 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 6 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds1_wave1_scored_pdss_gender_wide,
                                  scale_regx = '^PDSS$',
                                  type = 'p_missing') +
    facet_wrap(~factor(gender,
                       levels = pdss_gender_code,
                       labels = names(pdss_gender_code)))
```



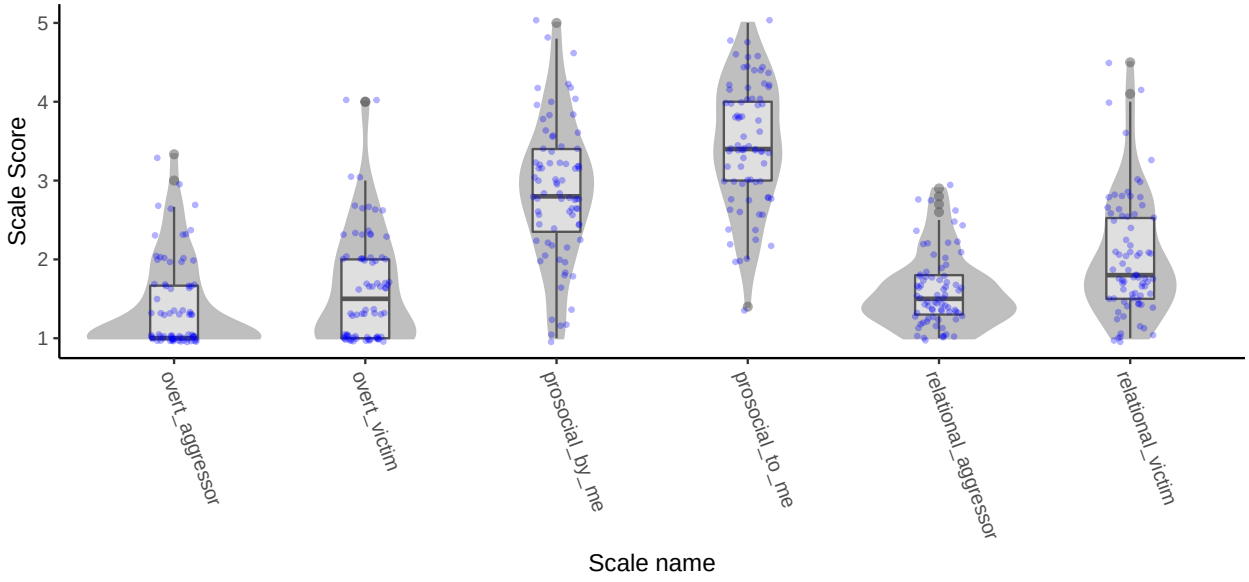
PEQ-R

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^PEQ-R$',
                                  type = 'score')
```

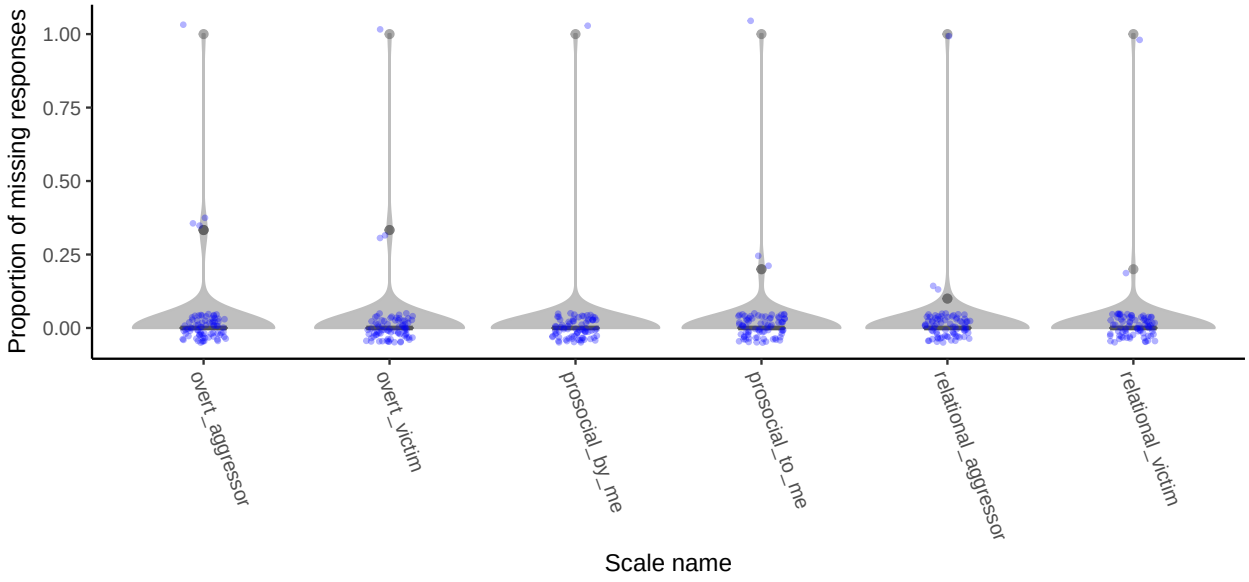
```
## Warning: Removed 6 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 6 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```



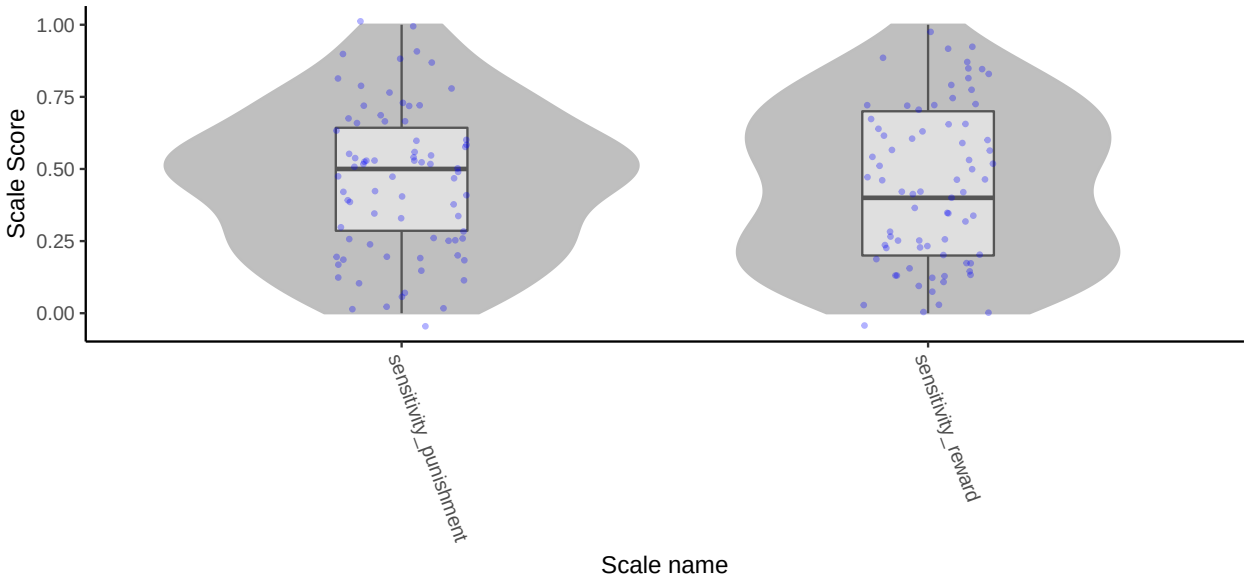

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^PEQ-R$',
                                  type = 'p_missing')
```



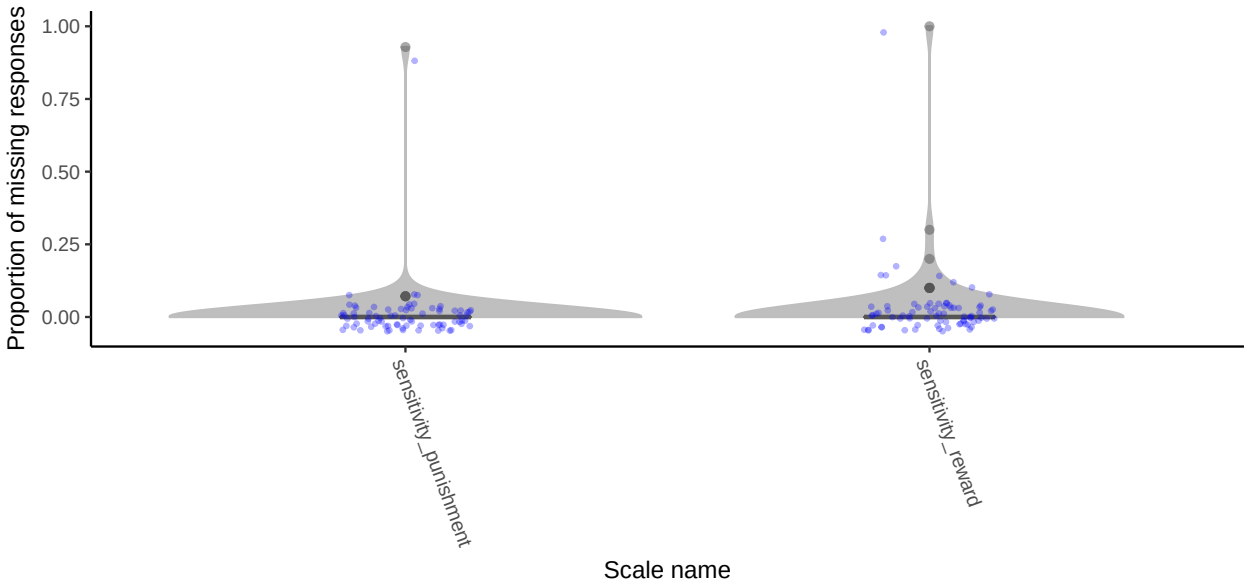
SPSRQ-S

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^SPSRQ-S$',
                                  type = 'score')
```

```
## Warning: Removed 1 rows containing non-finite values (stat_ydensity).
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^SPSRQ-S$',
                                  type = 'p_missing')
```

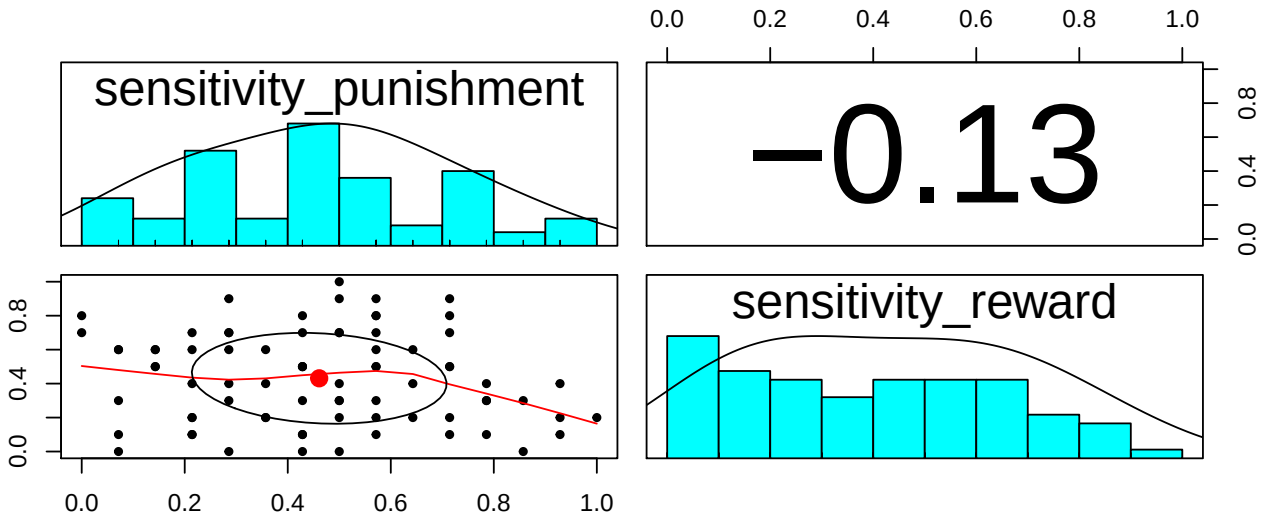



```
tds1w1_SPSRQSubric <- tds1_wave1_scoring_data_long %>%
    filter(scale_name == 'SPSRQ-S')

tds1_wave1_spsrqs_psych <- scorequaltrics::score_questionnaire(
    tds1_wave1_long_prepost_recoded_nodupes,
    tds1w1_SPSRQSubric,
    psych = T)
print(tds1_wave1_spsrqs_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       sensitivity_punishment sensitivity_reward
```

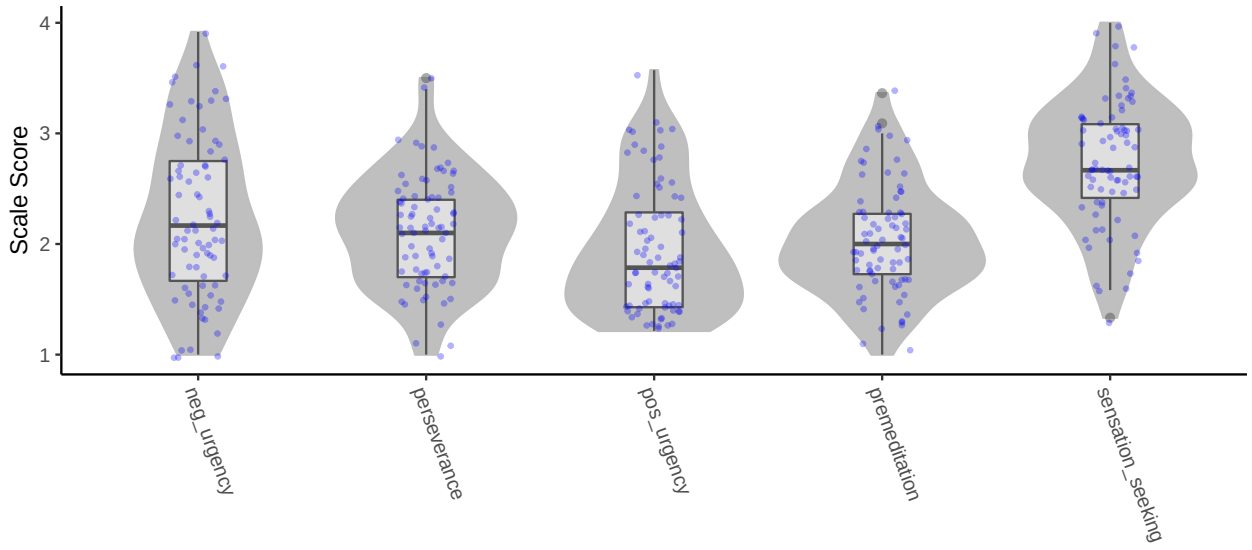
```
## alpha                   0.77                0.76
##
## Standard errors of unstandardized Alpha:
##       sensitivity_punishment sensitivity_reward
## ASE                    0.049               0.056
##
## Average item correlation:
##           sensitivity_punishment sensitivity_reward
## average.r                   0.19                0.24
##
##  Guttman 6* reliability:
##          sensitivity_punishment sensitivity_reward
## Lambda.6                   0.84                0.83
##
## Signal/Noise based upon av.r :
##              sensitivity_punishment sensitivity_reward
## Signal/Noise                    3.3                 3.2
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                         sensitivity_punishment sensitivity_reward
## sensitivity_punishment                   0.77               -0.16
## sensitivity_reward                      -0.13                0.76
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

```
pairs.panels(tds1_wave1_spsrqs_psych$scores)
```



UPPS-P

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^UPPS-P$',
                                  type = 'score')
```



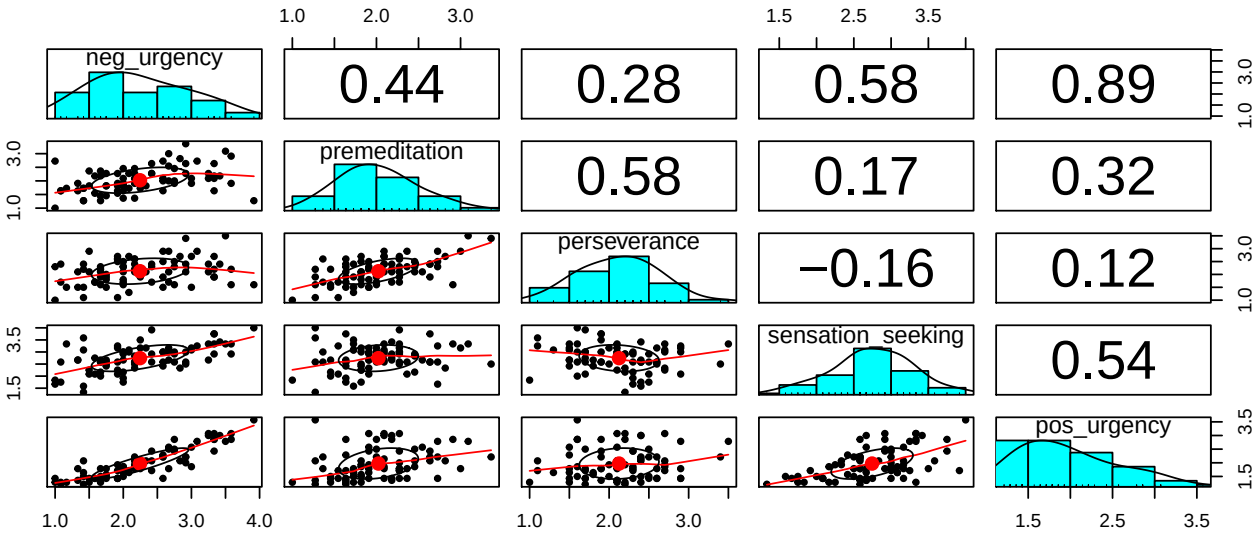

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^SUPPS-P$',
                                  type = 'p_missing')
```


```
tds1w1_UPPSPRubric <- tds1_wave1_scoring_data_long %>%
    filter(scale_name == 'UPPS-P')

tds1_wave1_uppsp_psych <- scorequaltrics::score_questionnaire(
    tds1_wave1_long_prepost_recoded_nodupes,
    tds1w1_UPPSPRubric,
    psych = T)
print(tds1_wave1_uppsp_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       neg_urgency premeditation perseverance sensation_seeking pos_urgency
```

```
## alpha        0.92           0.8         0.79              0.81        0.89
##
## Standard errors of unstandardized Alpha:
##       neg_urgency premeditation perseverance sensation_seeking pos_urgency
## ASE         0.028         0.048        0.052             0.046       0.029
##
## Average item correlation:
##           neg_urgency premeditation perseverance sensation_seeking
## average.r        0.49          0.27         0.27              0.26
##           pos_urgency
## average.r        0.37
##
##  Guttman 6* reliability:
##          neg_urgency premeditation perseverance sensation_seeking
## Lambda.6        0.98          0.97         0.95              0.96
##          pos_urgency
## Lambda.6        0.98
##
## Signal/Noise based upon av.r :
##              neg_urgency premeditation perseverance sensation_seeking
## Signal/Noise          11           4.1          3.7               4.1
##              pos_urgency
## Signal/Noise         8.3
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                   neg_urgency premeditation perseverance sensation_seeking
## neg_urgency              0.92          0.51         0.33              0.67
## premeditation            0.44          0.80         0.73              0.21
## perseverance             0.28          0.58         0.79             -0.20
## sensation_seeking        0.58          0.17        -0.16              0.81
## pos_urgency              0.89          0.32         0.12              0.54
##                   pos_urgency
## neg_urgency              0.98
## premeditation            0.38
## perseverance             0.14
## sensation_seeking        0.64
## pos_urgency              0.89
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

```
pairs.panels(tds1_wave1_uppsp_psych$scores)
```



YRBS

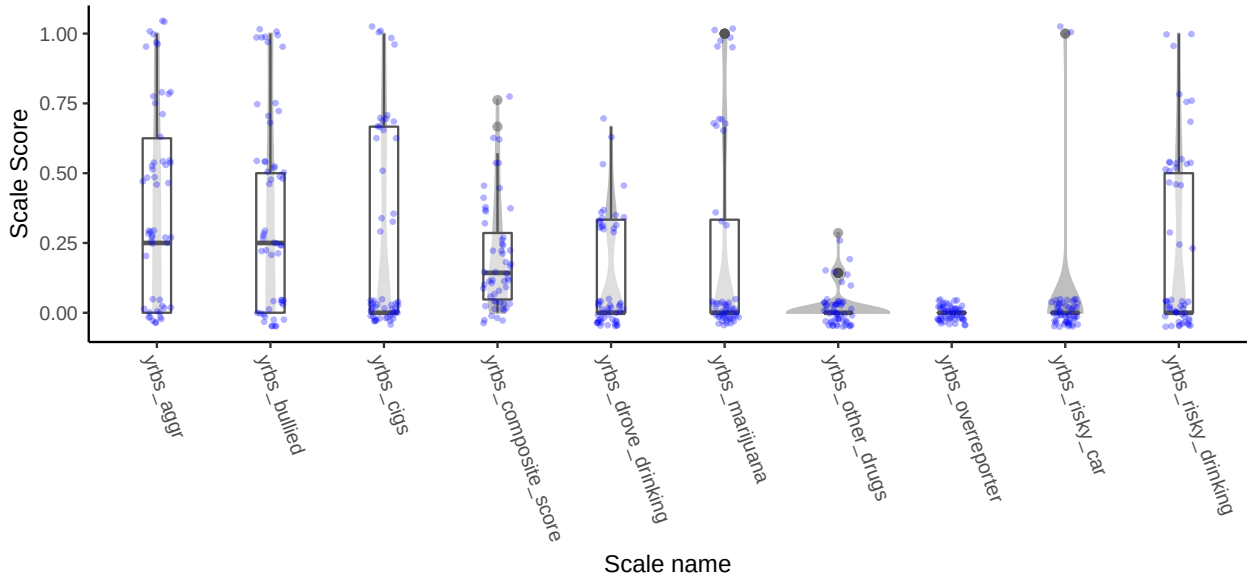
This has been scored according to the CDC manual. Items have been scored with either 1 or 0, with the mean calculated for each subscale. This gives you the proportion of endorsed items in that scale. To convert to a sum of endorsed items, simply multiply the scaled score by the number of total items (`n_missing` + `n_items`).

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^YRBS$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 360-70))
```

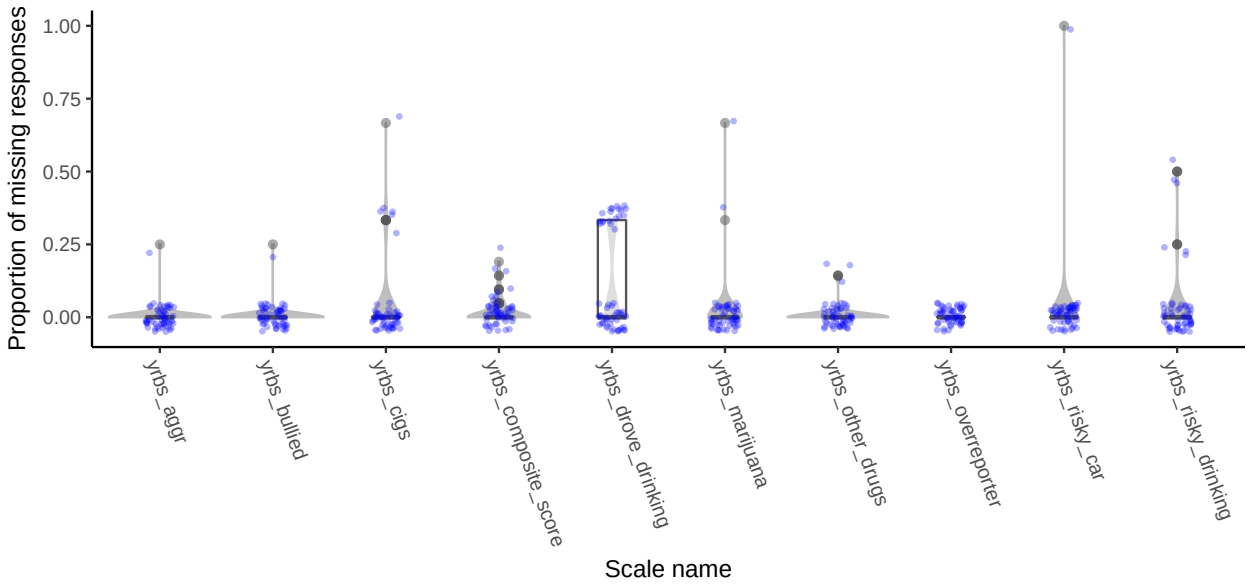
```
## Warning: Removed 1 rows containing non-finite values (stat_ydensity).
```

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

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

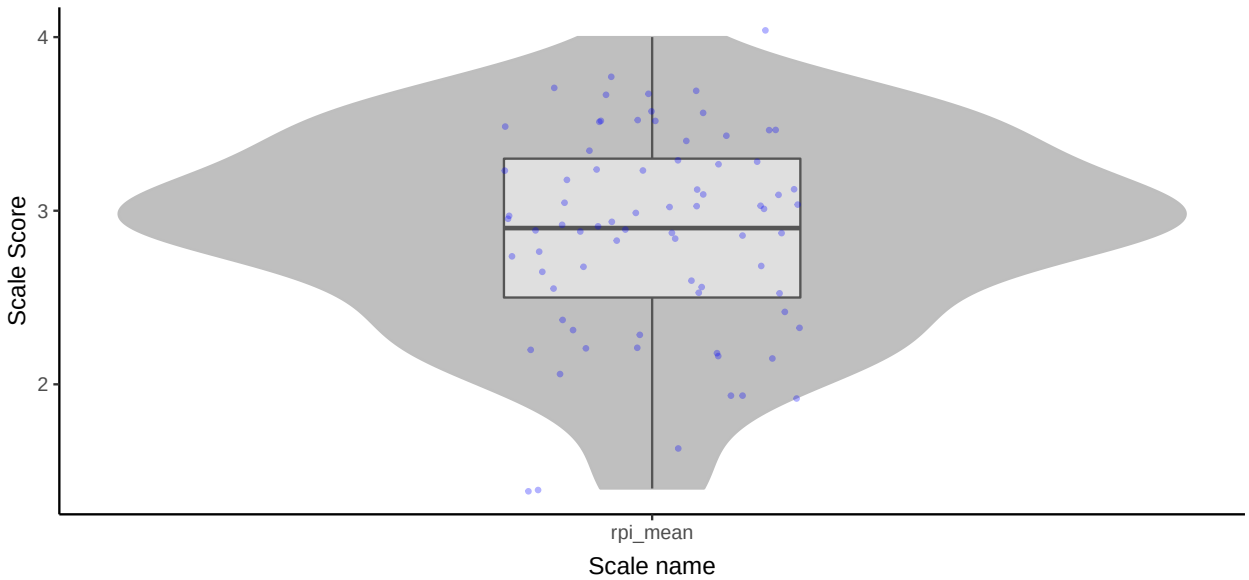


```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^YRBS$',
                                  type = 'p_missing')
```



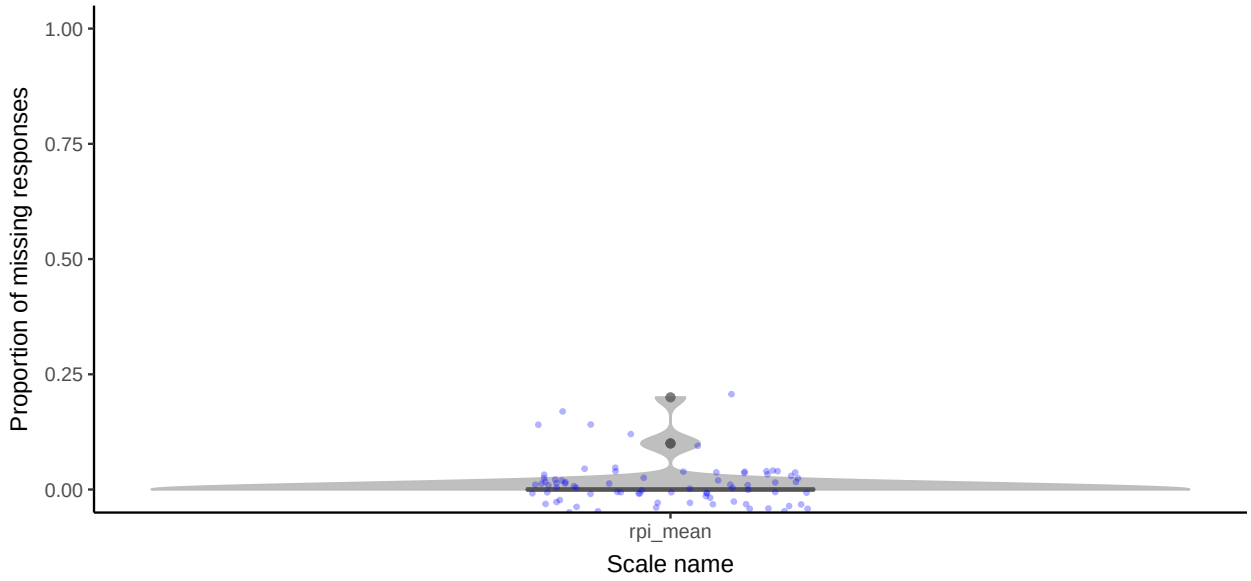
RPI

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^RPI_part2$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```




```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^RPI_part2$',
                                  type = 'p_missing') +
    coord_cartesian(y = c(0,1)) +
```

```
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

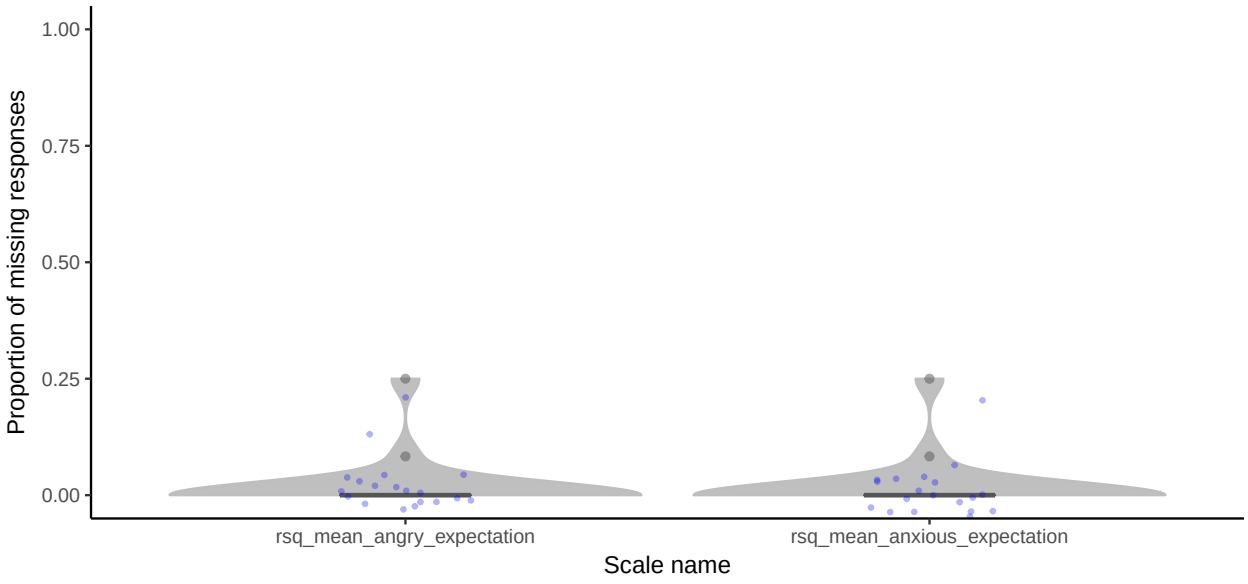


RSQ - 1

From TDS1-Pre participants only.

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^RSQ1_part2$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

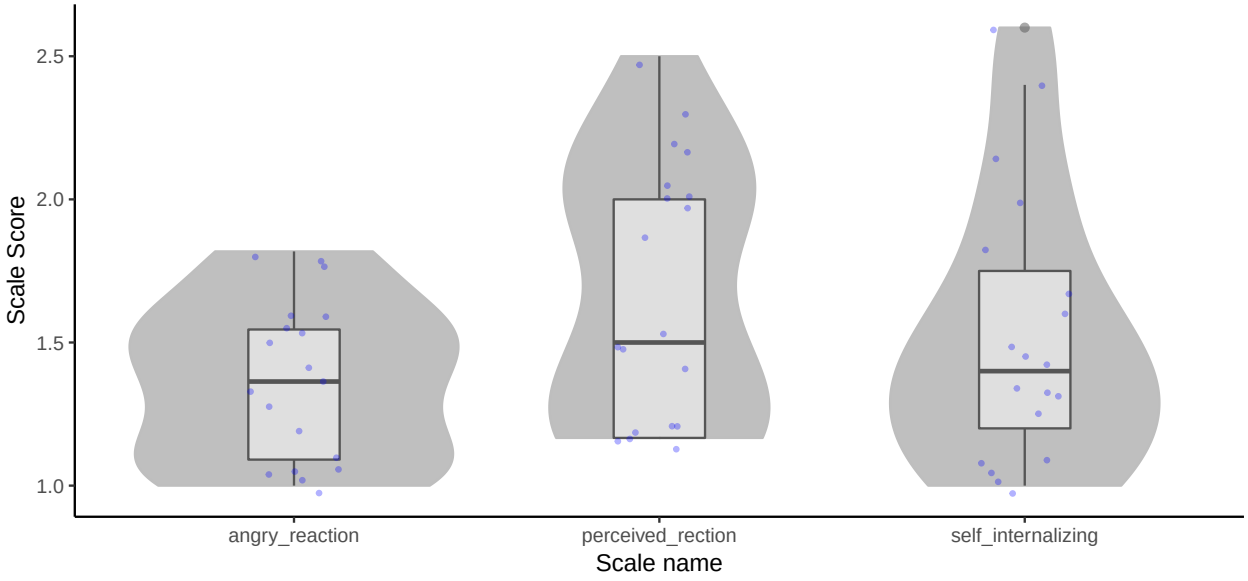

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^RSQ1_part2$',
                                  type = 'p_missing') +
    coord_cartesian(y = c(0,1)) +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```



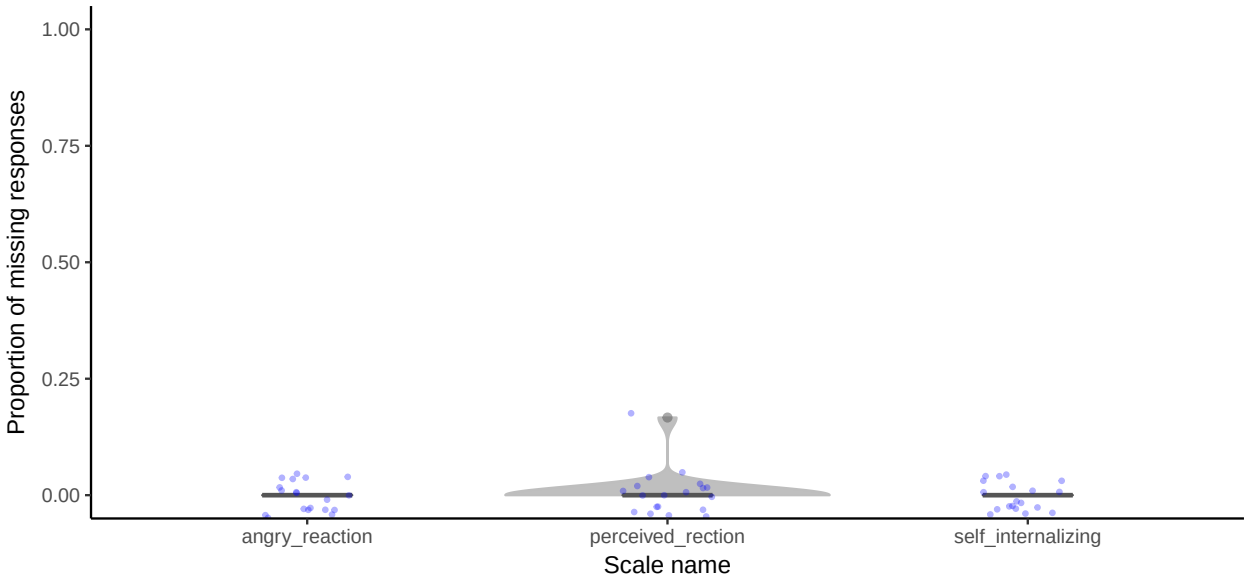
RSQ - 2

From TDS1-Pre participants only.

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^RSQ_2$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```




```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^RSQ_2$',
                                  type = 'p_missing') +
    coord_cartesian(y = c(0,1)) +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

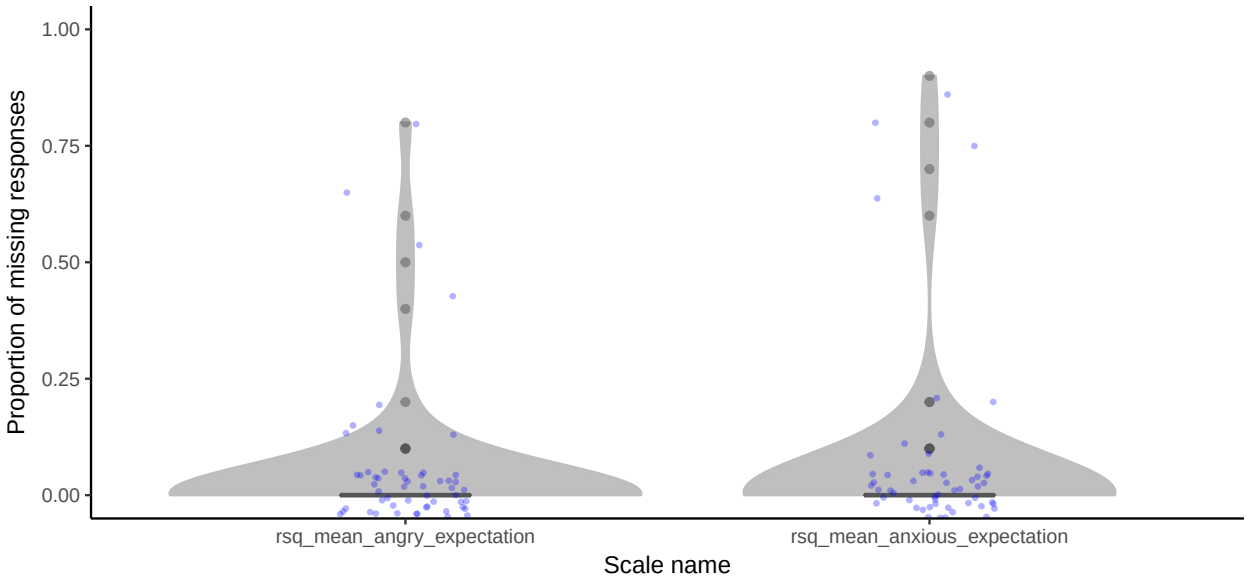


RSQ - Modified

From TDS1-Post participants only.

```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^RSQMod_part2$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```


```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^RSQMod_part2$',
                                  type = 'p_missing') +
    coord_cartesian(y = c(0,1)) +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```



Write data

We'll write out each data frame in wide format for ease of use in other contexts. We can use `lapply` on a list of unique scale names to write each one out using `widen_qualtrics_long` and `write.csv`. We treat the PDSS separately.

```
list_of_scales <- as.list(unique(tds1_wave1_scored$scale_name))
tds1_wave1_metadata <- list(wave = 1, questionnaires = 'TDS1')

throwaway <- lapply(list_of_scales,
                    function(aScale){
                        file_name <- paste0(TDS1_name, '-',
                                            wave1_name, '-',
                                            scorequaltrics::make_nice_scale_fname(aScale),
                                            '.csv')
                        scorequaltrics::write_widened_scored_scale(
                            tds1_wave1_scored,
                            scale_names = aScale,
                            dir_name = output_file_dir,
                            file_name = file_name,
                            metadata = tds1_wave1_metadata)
                    })


tds1_wave1_pdss_filename <-  paste0(TDS1_name, '-',
                                    wave1_name, '-',
                                    'PDSS.csv')

scorequaltrics::write_widened_scored_scale(
    tds1_wave1_scored_pdss,
    scale_names = 'PDSS',
    dir_name = output_file_dir,
    file_name = tds1_wave1_pdss_filename,
    metadata = tds1_wave1_metadata)
```

TDS 1 Wave 2

Get the data and clean it

Get data, rubrics, recode, and clean. Note that for wave 2 (aka, session 3), we need to separate out parent from child questionnaires, since the PMQ items are identical.

```
#Get surveys
tds1_wave2_surveys <- rawSurveysTDS %>%
    filter(grepl('(TDS1, Session 3 - (Child|Parent)$|TDS1 AND TDS3 PSQI)', SurveyName))
knitr::kable(select(tds1_wave2_surveys, SurveyName))
```

SurveyName
TDS1, Session 3 - Parent
TDS1, Session 3 - Child
TDS1 AND TDS3 PSQI

```
#Get data from those surveys
tds1_wave2_long <- scorequaltrics::get_survey_data(tds1_wave2_surveys,
                                                   credentials,
                                                   pid_col = pid_column_name)

#Get recoding rubrics
tds1_wave2_recoding_rubrics <- data.frame(file = dir(file.path(tds1_wave2_rubric_dir),
                                                     pattern = '.*response_recoding.*.csv',
                                                     full.names = TRUE))
tds1_wave2_recoding_data_long <- scorequaltrics::get_rubrics(tds1_wave2_recoding_rubrics,
                                                             type = 'recoding')

#Recode data
tds1_wave2_long_recoded <- scorequaltrics::recode_responses(tds1_wave2_long,
                                                            tds1_wave2_recoding_data_long)

#Code -99 as NA
tds1_wave2_long_recoded <- tds1_wave2_long_recoded %>%
    mutate(value = ifelse(grepl('SES', item) & as.numeric(value) < 0,
                          NA,
                          value))
```

```
## Warning in ifelse(grepl("SES", item) & as.numeric(value) < 0, NA, value):
## NAs introduced by coercion
```

```
#Get scoring rubrics
tds1_wave2_scoring_rubrics <- data.frame(file = dir(file.path(tds1_wave2_rubric_dir),
                                                    pattern = '.*scoring_rubric.*.csv',
                                                    full.names = TRUE))
tds1_wave2_scoring_data_long <- scorequaltrics::get_rubrics(tds1_wave2_scoring_rubrics,
                                                            type = 'scoring')

#Split scoring into parent and child
tds1_wave2_scoring_data_long_p <- tds1_wave2_scoring_data_long %>%
    filter(grepl('[pP]arent', data_file_name))
tds1_wave2_scoring_data_long_c <- tds1_wave2_scoring_data_long %>%
    filter(!grepl('[pP]arent', data_file_name))
```

```r
#Split data into parent and child
tds1_wave2_long_recoded_p <- tds1_wave2_long_recoded %>%
    filter(grepl('[pP]arent', survey_name))
tds1_wave2_long_recoded_c <- tds1_wave2_long_recoded %>%
    filter(!grepl('[pP]arent', survey_name))

#Clean and de-dupe parent and child data
tds1_wave2_long_recoded_p_nodupes <- tds1_wave2_long_recoded_p %>%
    get_items_in_rubric(tds1_wave2_scoring_data_long_p) %>%
    filter(grepl('3\\d\\d', SID)) %>%
    scorequaltrics::clean_dupes(pid_col = 'SID')
```

```
## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion
## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion
```

```
## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion
```

```
tds1_wave2_long_recoded_c_nodupes <- tds1_wave2_long_recoded_c %>%
    get_items_in_rubric(tds1_wave2_scoring_data_long_c) %>%
    filter(grepl('3\\d\\d', SID)) %>%
    scorequaltrics::clean_dupes(pid_col = 'SID')
```

```
## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion

## Warning in na.exclude(as.numeric(.$value)): NAs introduced by coercion
```

```
#Need to clean PSQI data a bit differently
tds1_wave2_long_recoded_c_nodupes_psqi <- filter(tds1_wave2_long_recoded_c,
                                                  grepl('PSQI', item),
                                                  grepl('3\\d\\d', SID)) %>%
    scorequaltrics::clean_dupes(pid_col = 'SID', keep_text = T)
```

After cleaning duplicates, it's possible that there were some responses that have conflicting answers. We can look at those now.

Now we can see which items had conflicting responses on duplicate questionnaires.

```
#Check that dropped values weren't ambiguous
tds1_wave2_long_recoded_p_nodupes %>%
```

```
    filter(dropped) %>%
    group_by(SID, item) %>%
    filter(!all(length(unlist(old.value)) < 1)) %>%
    mutate(old.value = paste(old.value, collaps = ' ')) %>%
    knitr::kable(caption = "Parent questionnaire dupes")
```

Table: Parent questionnaire dupes

SID item value survey_name old.value dropped —- —— —— ———— ———- ———

```
tds1_wave2_long_recoded_c_nodupes %>%
    filter(dropped) %>%
    group_by(SID, item) %>%
    filter(!all(length(unlist(old.value)) < 1)) %>%
    mutate(old.value = paste(old.value, collaps = ' ')) %>%
    knitr::kable(caption = "Child questionnaire dupes")
```

Table 13: Child questionnaire dupes

SID	item	value	survey_name	old.value	dropped
323	YRBS_10	NA	TDS1, Session 3 - Child	c(1, 0)	TRUE

```
tds1_wave2_long_recoded_c_nodupes_psqi %>%
    filter(dropped) %>%
    group_by(SID, item) %>%
    filter(!all(length(unlist(old.value)) < 1)) %>%
    mutate(old.value = paste(old.value, collaps = ' ')) %>%
    knitr::kable(caption = "PSQI questionnaire dupes")
```

Table: PSQI questionnaire dupes

SID item value survey_name old.value dropped —- —— —— ———— ———- ———

Scoring

```
tds1_wave2_scored_p <- scorequaltrics::score_step_one_and_two(tds1_wave2_long_recoded_p_nodupes,
                                                              tds1_wave2_scoring_data_long_p)
tds1_wave2_scored_c <- scorequaltrics::score_step_one_and_two(tds1_wave2_long_recoded_c_nodupes,
                                                              tds1_wave2_scoring_data_long_c)
tds1_wave2_scored_pdss <- scorequaltrics::score_pdss(tds1_wave2_long_recoded_c_nodupes,
                                                     gender_mix = pdss_gender_mix,
                                                     gendercode = pdss_gender_code)
tds1_wave2_scored_psqi <- scorequaltrics::score_psqi(
    tds1_wave2_long_recoded_c_nodupes_psqi,
    pid_col = 'SID') %>%
    mutate(score = as.character(score))
```

```
## Warning in scorequaltrics:::score_psqi_2(psqi_data$PSQI_2_2_TEXT): NAs
## introduced by coercion
```

```
## Warning in scorequaltrics:::score_psqi_4(psqi_data$PSQI_4_2_TEXT): NAs
## introduced by coercion
```

```
## Warning in scorequaltrics:::score_psqi_4(psqi_data$PSQI_4_2_TEXT,
```

```
## return_value = "hours"): NAs introduced by coercion
```

```
tds1_wave2_scored_p %>%
    ungroup() %>%
    distinct(scale_name) %>%
    knitr::kable()
```

scale_name
CBCL
PMQ_Parent

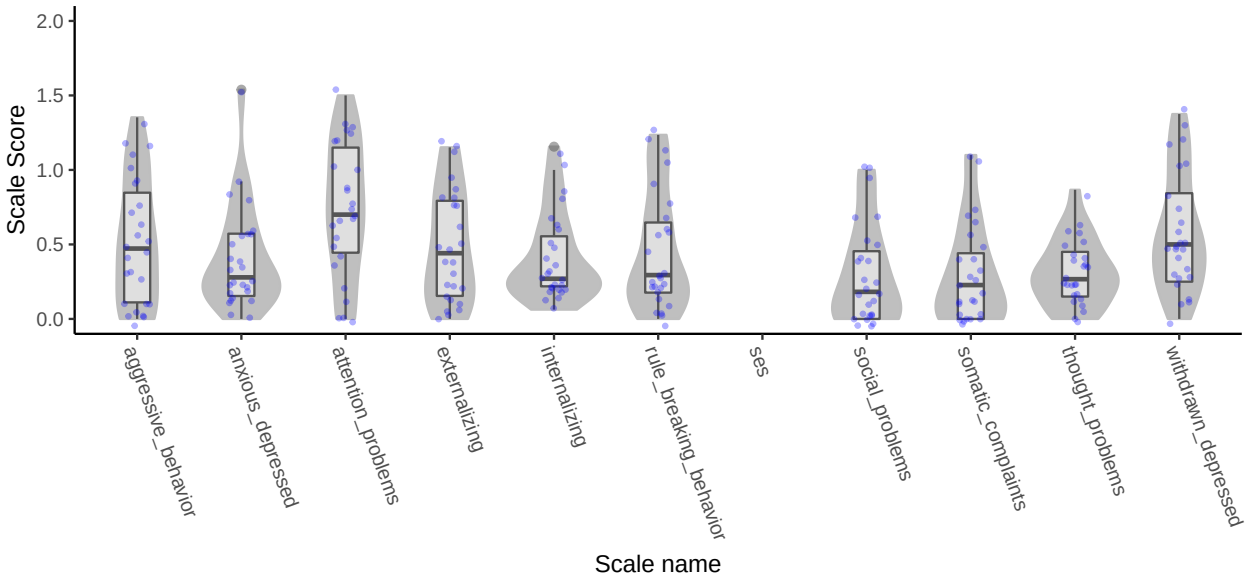
```
tds1_wave2_scored_c %>%
    ungroup() %>%
    distinct(scale_name) %>%
    knitr::kable()
```

scale_name
BFI-10
BIS-15
BSSS
CARE-R Social
CES-DC
K-SRQ
PAL-2
PDS
PMQ_Child
SAQ
YRBS
RPI_part2
RSQMod_part2

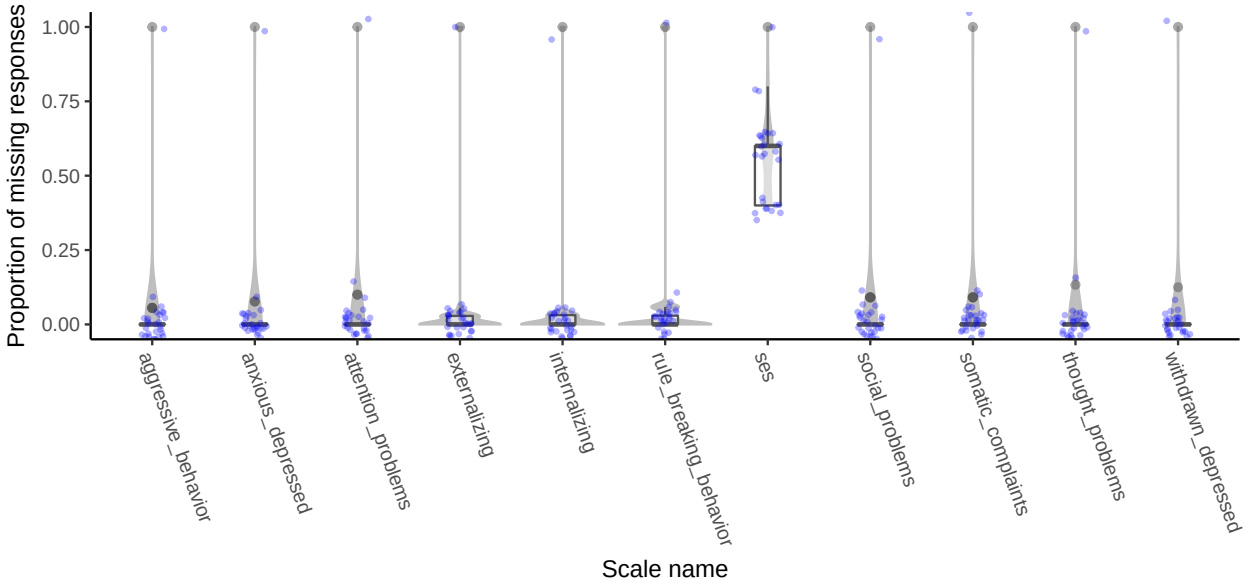
CBCL

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_p,
                                  scale_regx = '^CBCL$',
                                  type = 'score')+
    coord_cartesian(y = c(0,2))
```

```
## Warning: Removed 11 rows containing non-finite values (stat_ydensity).
## Warning: Removed 11 rows containing non-finite values (stat_boxplot).
## Warning: Removed 11 rows containing missing values (geom_point).
```



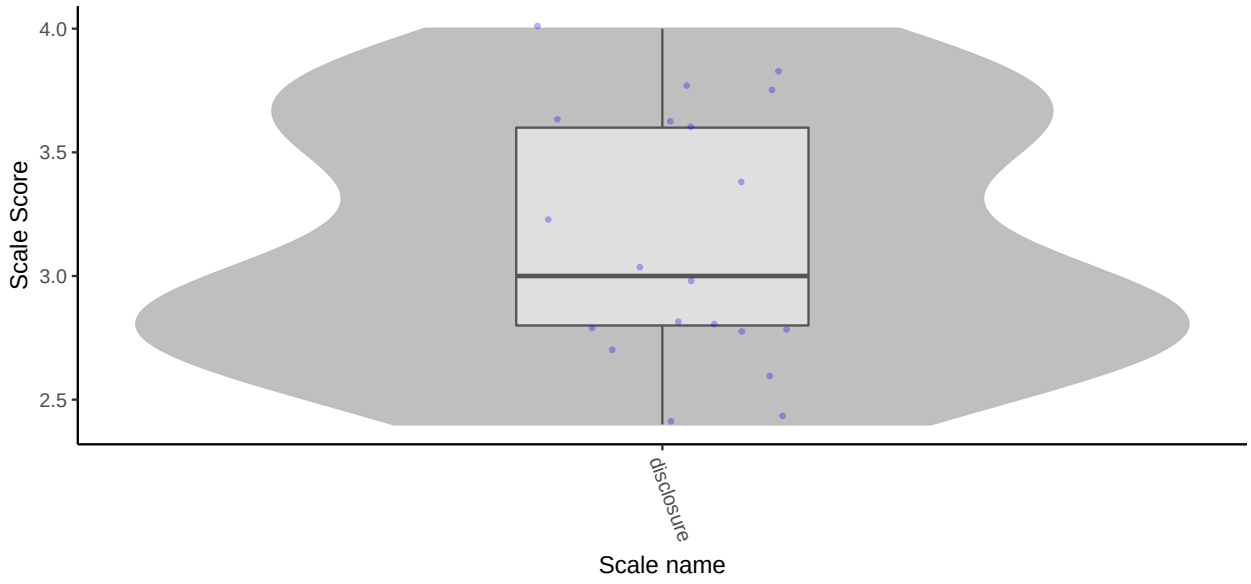

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_p,
                                  scale_regx = '^CBCL$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```



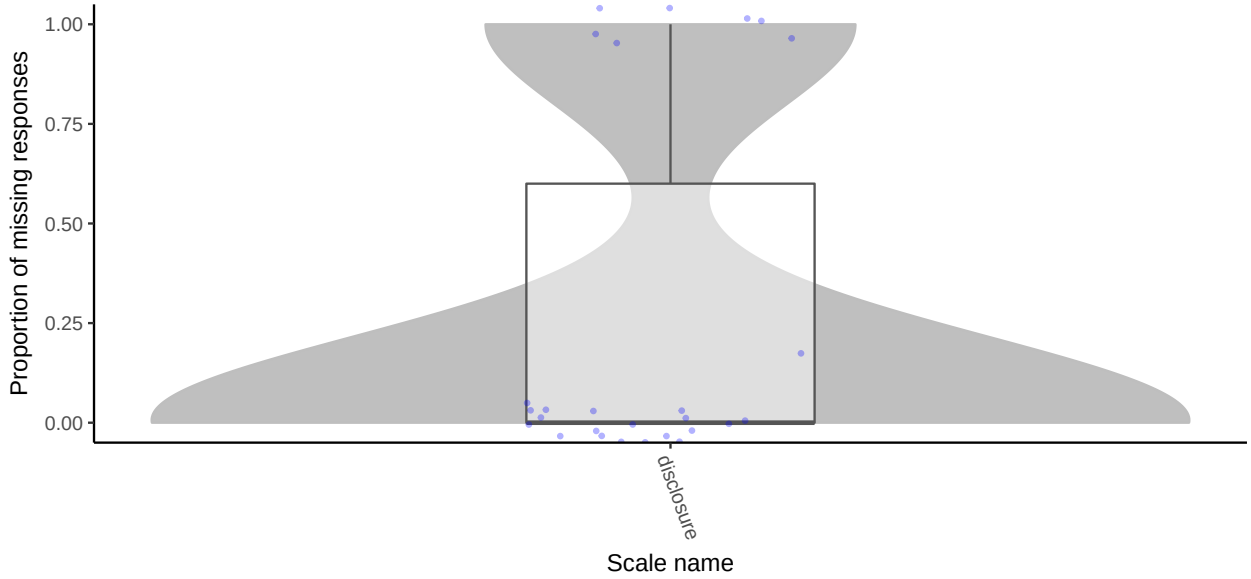
PMQ Parent

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_p,
                                  scale_regx = '^PMQ_Parent$',
                                  type = 'score')
```

```
## Warning: Removed 7 rows containing non-finite values (stat_ydensity).
## Warning: Removed 7 rows containing non-finite values (stat_boxplot).
## Warning: Removed 7 rows containing missing values (geom_point).
```



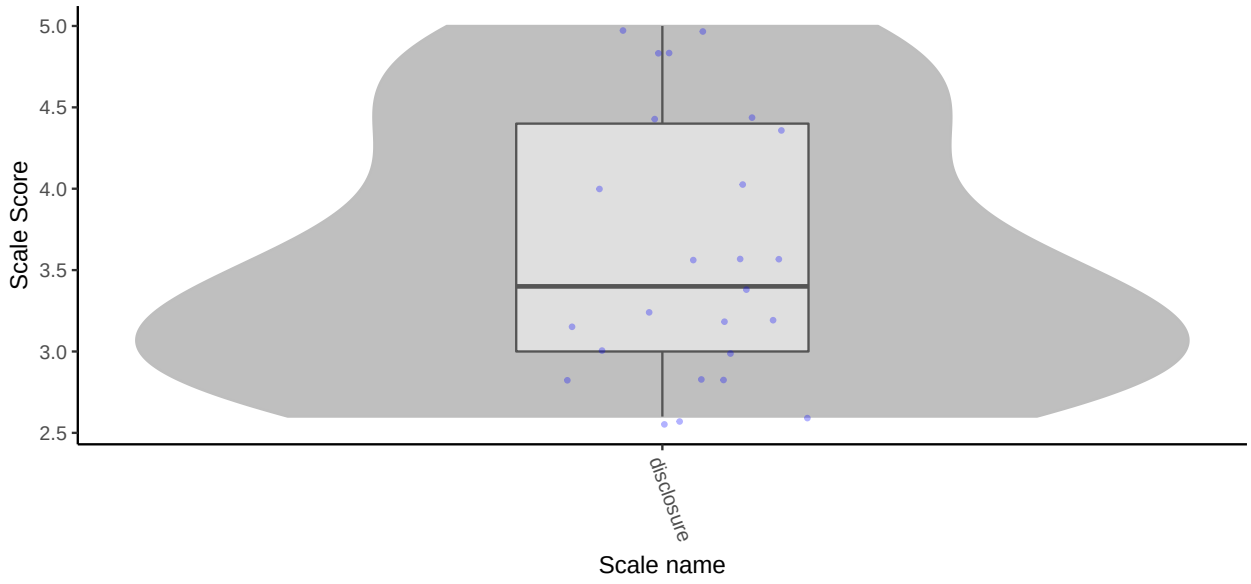

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_p,
                                  scale_regx = '^PMQ_Parent$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```



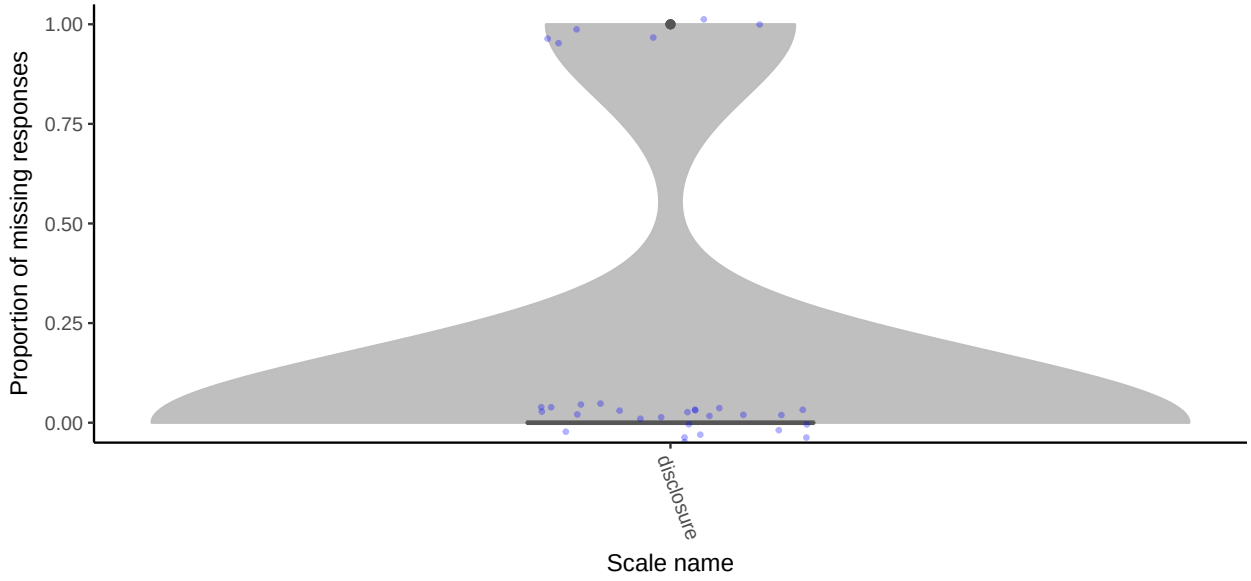
PMQ Child

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^PMQ_Child$',
                                  type = 'score')
```

```
## Warning: Removed 6 rows containing non-finite values (stat_ydensity).
## Warning: Removed 6 rows containing non-finite values (stat_boxplot).
## Warning: Removed 6 rows containing missing values (geom_point).
```



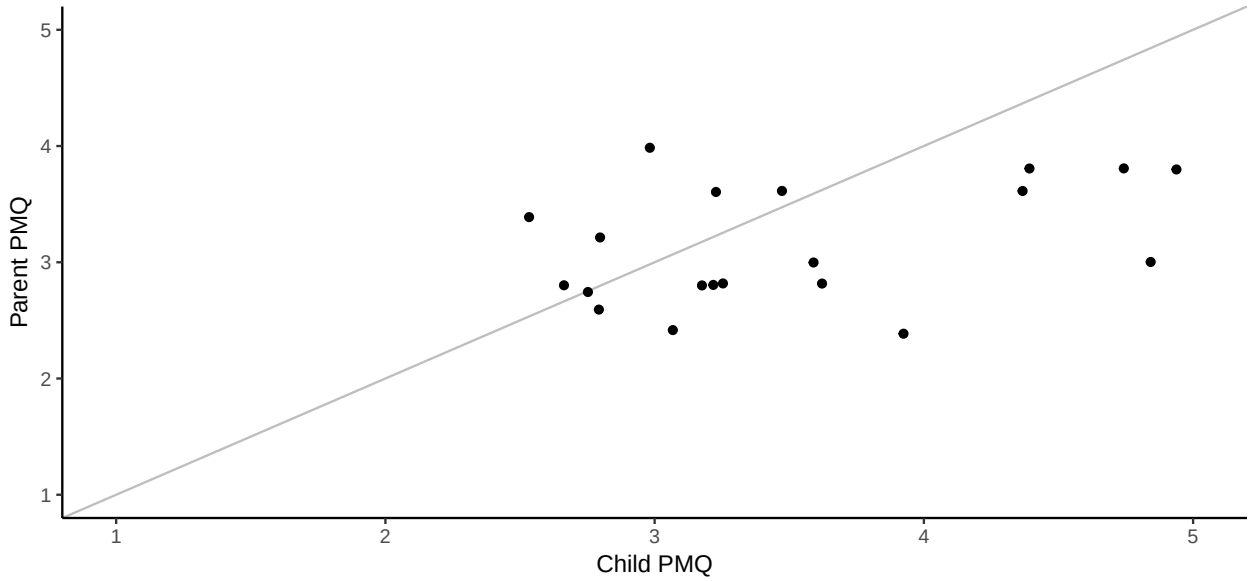

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^PMQ_Child$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```




```
tds1_wave2_disclosure_c <- tds1_wave2_scored_c %>%
    ungroup() %>%
    filter(scored_scale == 'disclosure') %>%
    select(scored_scale, SID, score) %>%
    spread(scored_scale, score) %>%
    rename('Child PMQ' = disclosure)

tds1_wave2_disclosure_p <- tds1_wave2_scored_p %>%
    ungroup() %>%
    filter(scored_scale == 'disclosure') %>%
    select(scored_scale, SID, score) %>%
    spread(scored_scale, score) %>%
```

```
    rename('Parent PMQ' = disclosure)

left_join(tds1_wave2_disclosure_c,
          tds1_wave2_disclosure_p) %>%
    mutate_at(vars(`Child PMQ`, `Parent PMQ`), as.numeric) %>%
    ggplot(aes(x = `Child PMQ`, y = `Parent PMQ`)) +
    geom_abline(intercept = 0, slope = 1, color = 'gray') +
    geom_point(position = position_jitter(), alpha = 1) +
    coord_cartesian(x = c(1, 5), y = c(1, 5)) +
    theme_classic()
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```



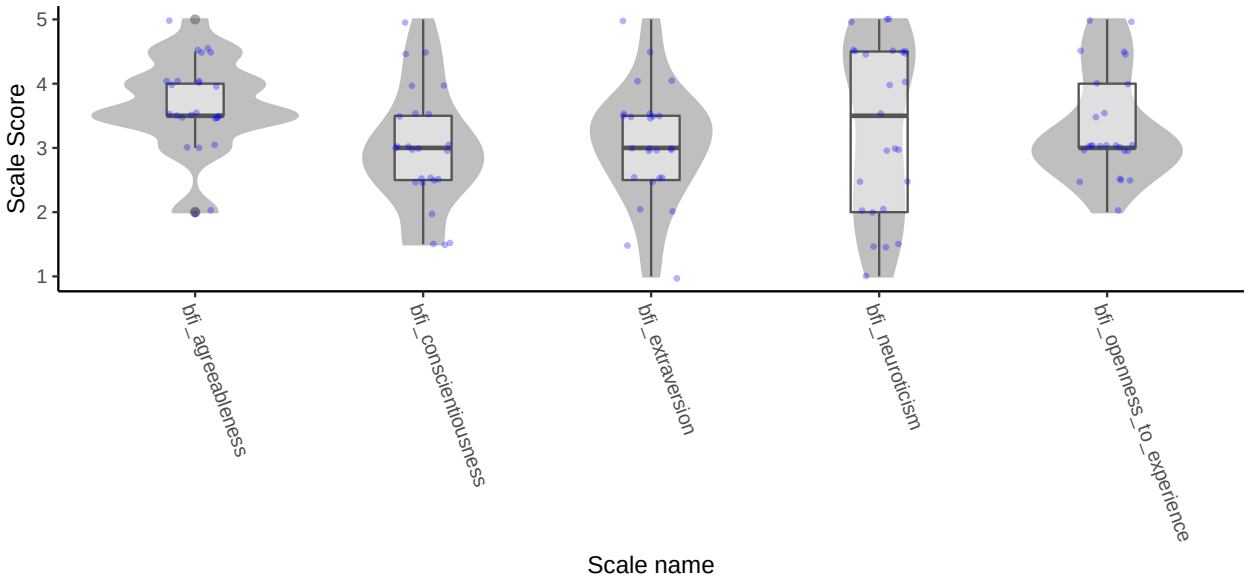
BFI-10

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^BFI-10$',
                                  type = 'score')
```

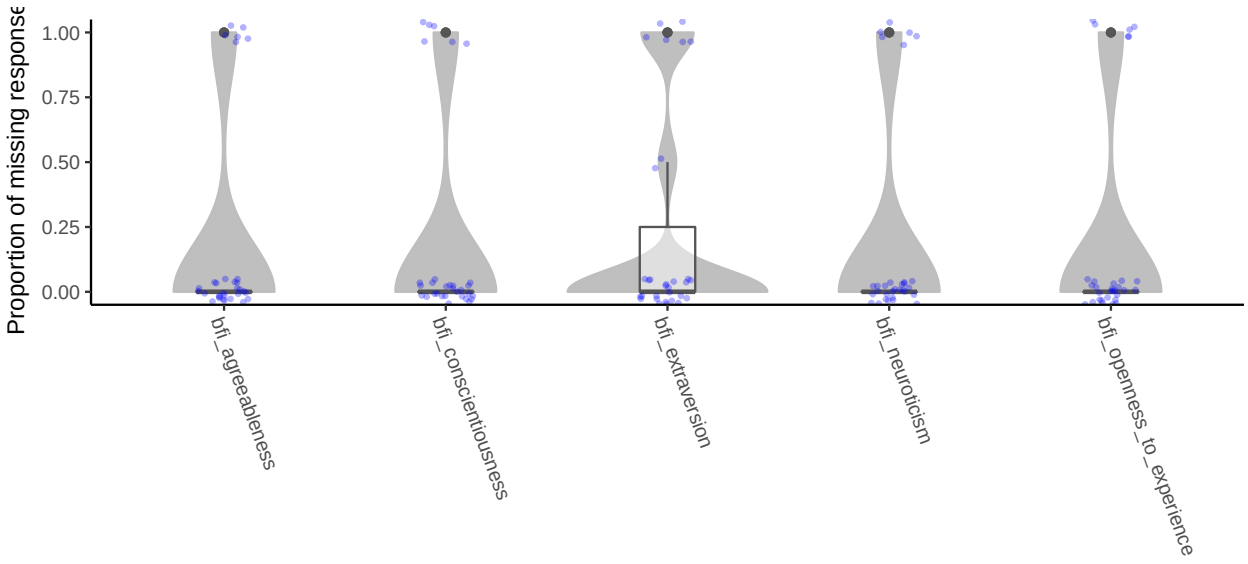
```
## Warning: Removed 30 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 30 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 30 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^BFI-10$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```




```
tds1w2_BFIRubric <- tds1_wave2_scoring_data_long_c %>%
    filter(scale_name == 'BFI-10')

tds1_wave2_bfi_psych <- scorequaltrics::score_questionnaire(tds1_wave2_long_recoded_c_nodupes,
                                  tds1w2_BFIRubric,
                                  psych = T)
```

```
## Warning in sqrt(y[i] * y[j]): NaNs produced
## Warning in sqrt(y[i] * y[j]): NaNs produced

## Warning in sqrt(y[i] * y[j]): NaNs produced
```

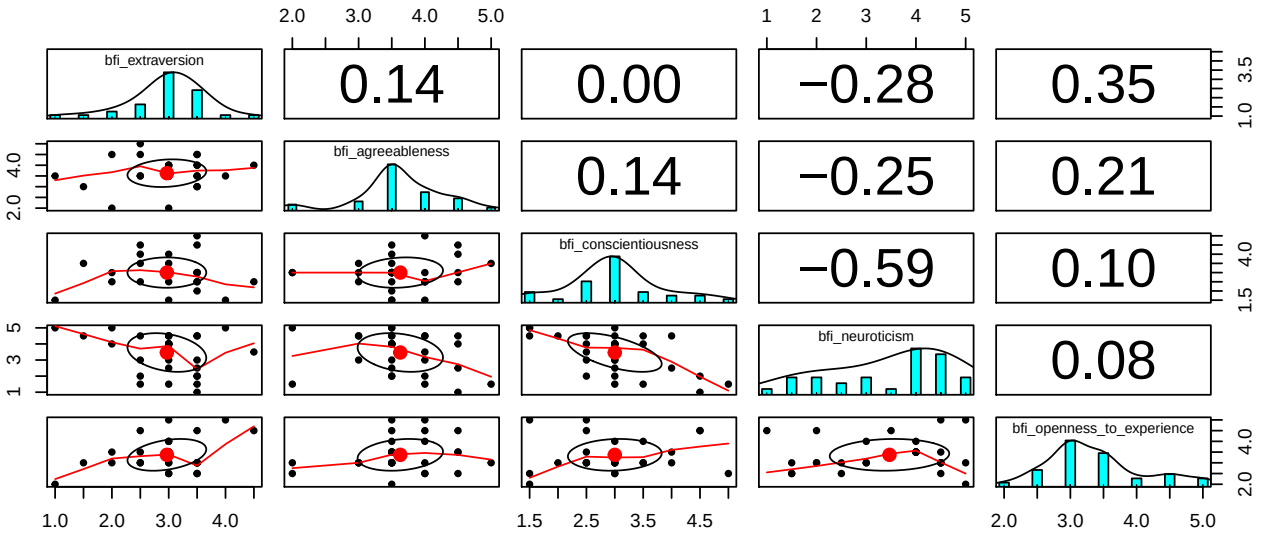
```
## Warning in sqrt(y[i] * y[j]): NaNs produced

## Warning in sqrt(y[i] * y[j]): NaNs produced

## Warning in sqrt(y[i] * y[j]): NaNs produced
```

```
print(tds1_wave2_bfi_psych, short = F)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       bfi_extraversion bfi_agreeableness bfi_conscientiousness
## alpha             0.24             -0.11                  0.71
##       bfi_neuroticism bfi_openness_to_experience
## alpha            0.75                      -0.57
##
## Standard errors of unstandardized Alpha:
##       bfi_extraversion bfi_agreeableness bfi_conscientiousness
## ASE               0.34              0.36                  0.26
##       bfi_neuroticism bfi_openness_to_experience
## ASE              0.26                     0.034
##
## Average item correlation:
##           bfi_extraversion bfi_agreeableness bfi_conscientiousness
## average.r             0.14            -0.053                  0.55
##           bfi_neuroticism bfi_openness_to_experience
## average.r             0.6                     -0.22
##
##  Guttman 6* reliability:
##          bfi_extraversion bfi_agreeableness bfi_conscientiousness
## Lambda.6             0.55              0.53                  0.73
##          bfi_neuroticism bfi_openness_to_experience
## Lambda.6            0.83                      0.37
##
## Signal/Noise based upon av.r :
##              bfi_extraversion bfi_agreeableness bfi_conscientiousness
## Signal/Noise             0.32              -0.1                   2.5
##              bfi_neuroticism bfi_openness_to_experience
## Signal/Noise               3                     -0.36
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                           bfi_extraversion bfi_agreeableness
## bfi_extraversion                   2.4e-01               NaN
## bfi_agreeableness                  1.4e-01             -0.11
## bfi_conscientiousness              1.6e-21              0.14
## bfi_neuroticism                   -2.8e-01             -0.25
## bfi_openness_to_experience         3.5e-01              0.21
##                           bfi_conscientiousness bfi_neuroticism
## bfi_extraversion                        3.8e-21          -0.660
## bfi_agreeableness                           NaN             NaN
## bfi_conscientiousness                   7.1e-01          -0.803
## bfi_neuroticism                        -5.9e-01           0.749
```

```
## bfi_openness_to_experience               9.5e-02          0.079
##                            bfi_openness_to_experience
## bfi_extraversion                                  NaN
## bfi_agreeableness                                0.84
## bfi_conscientiousness                             NaN
## bfi_neuroticism                                   NaN
## bfi_openness_to_experience                      -0.57
##
## Item by scale correlations:
##  corrected for item overlap and scale reliability
##        bfi_extraversion bfi_agreeableness bfi_conscientiousness
## BFI_1             -0.58             -0.39                  0.45
## BFI_6              0.55             -0.06                  0.38
## BFI_2             -0.16              0.46                  0.08
## BFI_7             -0.41             -0.54                 -0.15
## BFI_3             -0.01             -0.34                 -0.78
## BFI_8             -0.02             -0.05                  0.73
## BFI_4              0.21              0.30                  0.67
## BFI_9             -0.50             -0.32                 -0.55
## BFI_5             -0.33             -0.66                 -0.08
## BFI_10             0.25             -0.58                  0.05
##        bfi_neuroticism bfi_openness_to_experience
## BFI_1            -0.13                      -0.61
## BFI_6            -0.54                       0.28
## BFI_2             0.01                       0.27
## BFI_7             0.38                      -0.21
## BFI_3             0.56                      -0.34
## BFI_8            -0.59                      -0.13
## BFI_4            -0.83                      -0.11
## BFI_9             0.81                       0.12
## BFI_5            -0.10                      -0.53
## BFI_10           -0.02                       0.16
##
## Non missing response frequency for each item
##           1    2    3    4    5 miss
## BFI_1  0.00 0.09 0.26 0.35 0.30 0.26
## BFI_6  0.04 0.12 0.12 0.44 0.28 0.19
## BFI_2  0.04 0.04 0.04 0.40 0.48 0.19
## BFI_7  0.16 0.12 0.36 0.36 0.00 0.19
## BFI_3  0.08 0.08 0.12 0.48 0.24 0.19
## BFI_8  0.00 0.12 0.20 0.52 0.16 0.19
## BFI_4  0.20 0.32 0.04 0.16 0.28 0.19
## BFI_9  0.08 0.16 0.12 0.28 0.36 0.19
## BFI_5  0.24 0.08 0.20 0.16 0.32 0.19
## BFI_10 0.00 0.12 0.16 0.40 0.32 0.19
```

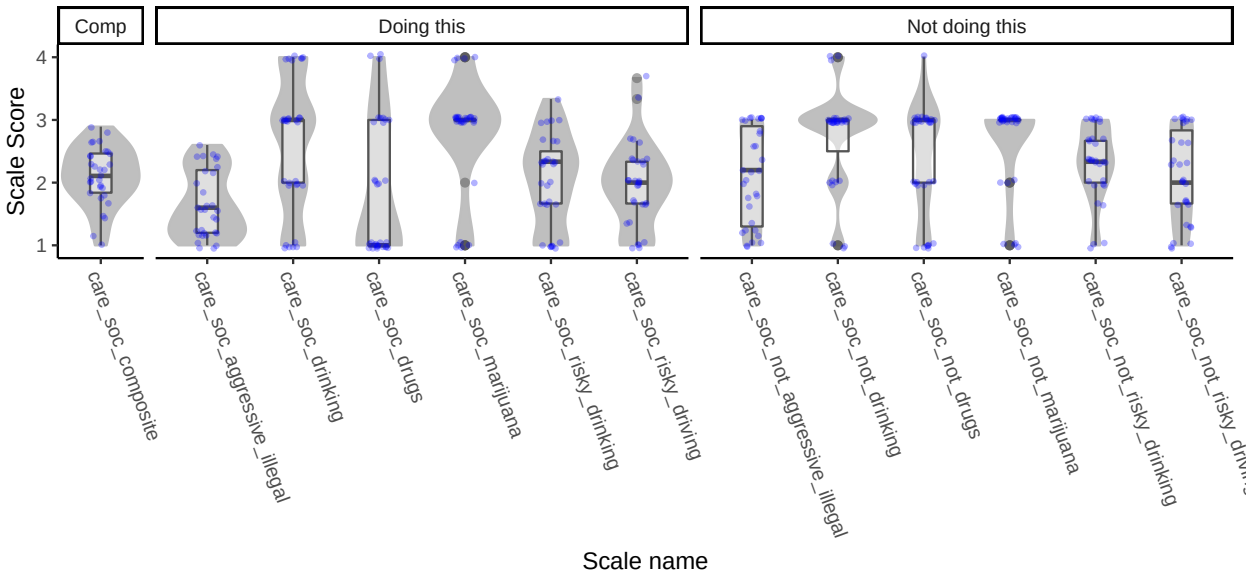
```
pairs.panels(tds1_wave2_bfi_psych$scores)
```



CARE-R Social

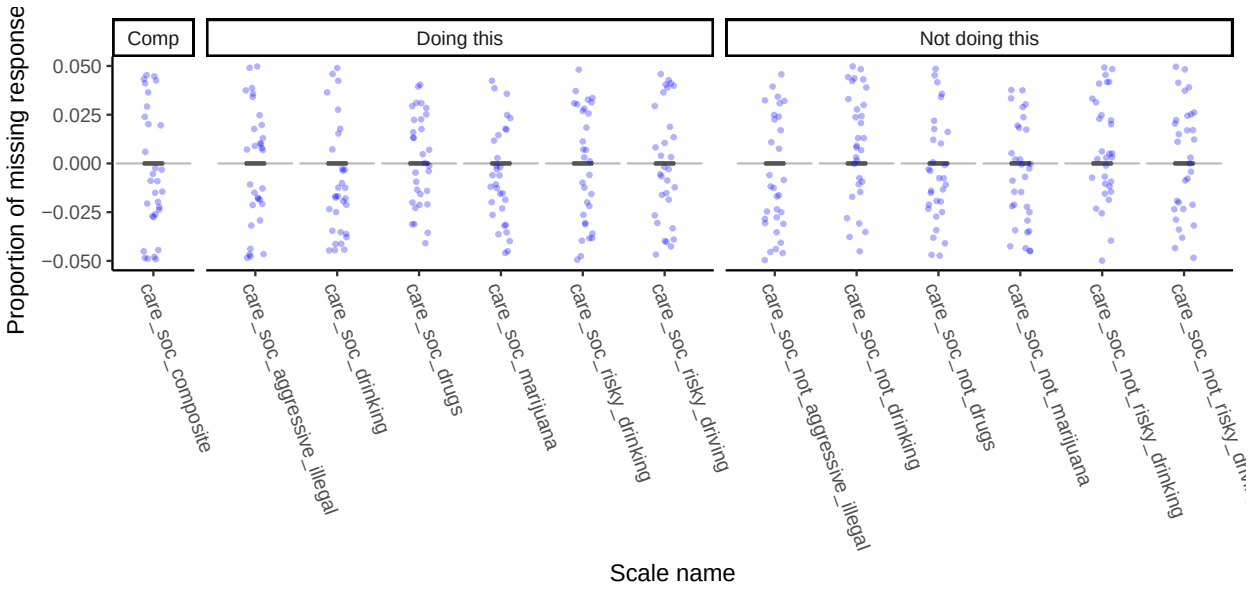
Note: Lower scores on "not doing this" items indicate that someone thinks not engaging in the behavior will make people like them *more*, and thus, like the "doing this" items, a lower score corresponds to a downward influence on that behavior.

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^CARE-R Social$',
                                  type = 'score') +
    facet_grid(~factor(grepl('not', scored_scale) - grepl('composite', scored_scale),
                       levels = c(-1, 0, 1),
                       labels = c('Comp', 'Doing this', 'Not doing this')),
               scales = 'free_x',
               space = 'free_x')
```




```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^CARE-R Social$',
                                  type = 'p_missing') +
```

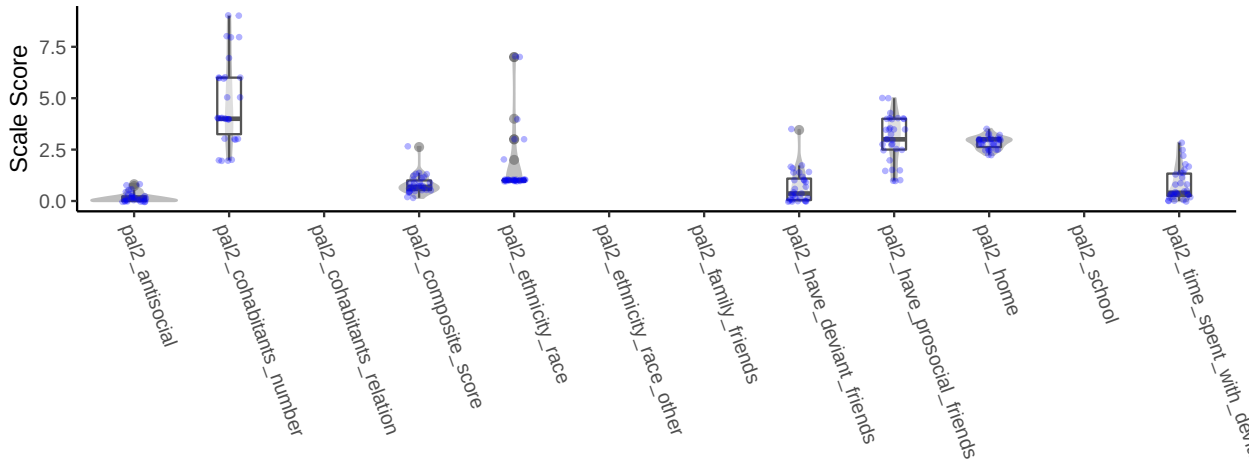
```
    facet_grid(~factor(grepl('not', scored_scale) - grepl('composite', scored_scale),
                       levels = c(-1, 0, 1),
                       labels = c('Comp', 'Doing this', 'Not doing this')),
               scales = 'free_x',
               space = 'free_x')
```



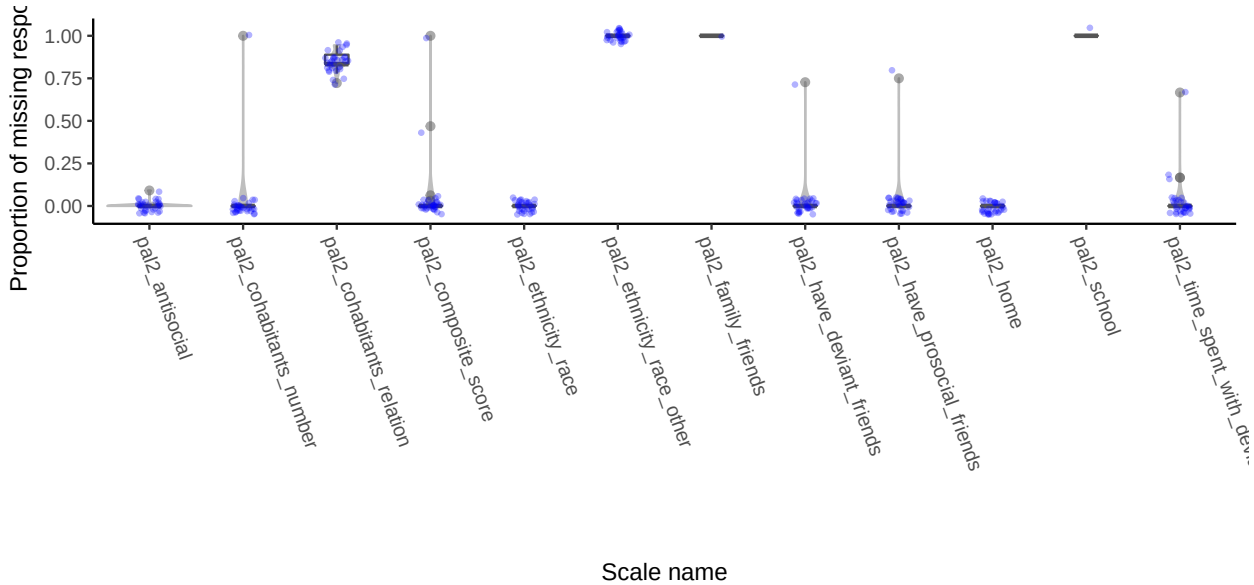
PAL-2

```
scorequaltrics::plot_scored_scale(filter(tds1_wave2_scored_c, scored_scale != 'pal2_age'),
                                  scale_regx = '^PAL-2$',
                                  type = 'score')
```

```
## Warning: Removed 66 rows containing non-finite values (stat_ydensity).
## Warning: Removed 66 rows containing non-finite values (stat_boxplot).
## Warning: Removed 66 rows containing missing values (geom_point).
```



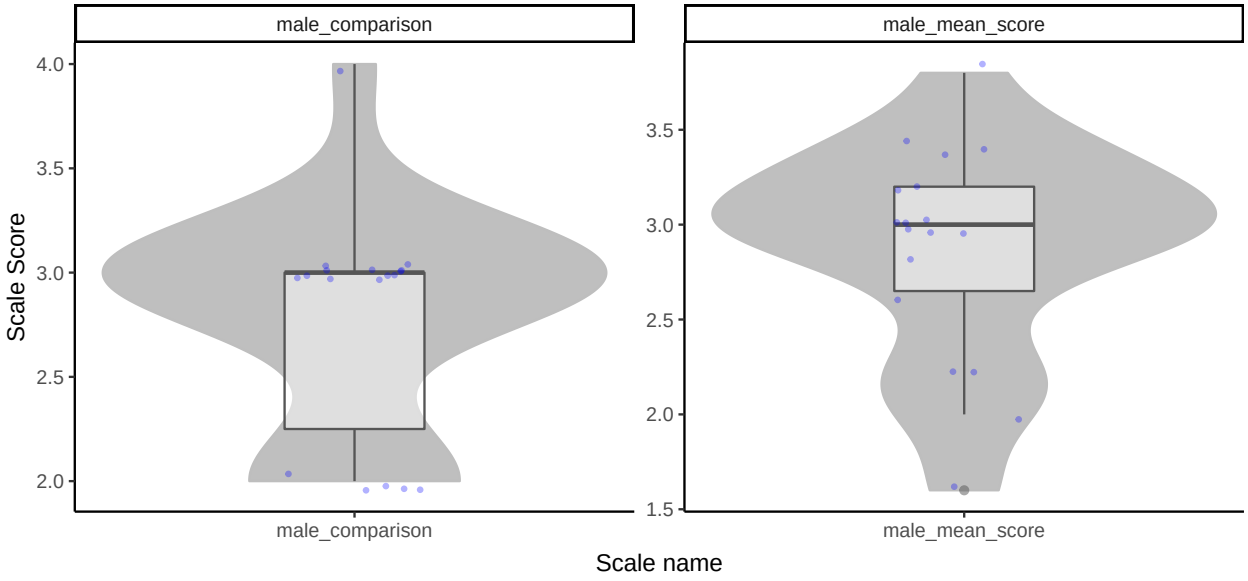

```
scorequaltrics::plot_scored_scale(filter(tds1_wave2_scored_c, scored_scale != 'pal2_age'),
                                  scale_regx = '^PAL-2$',
                                  type = 'p_missing')
```



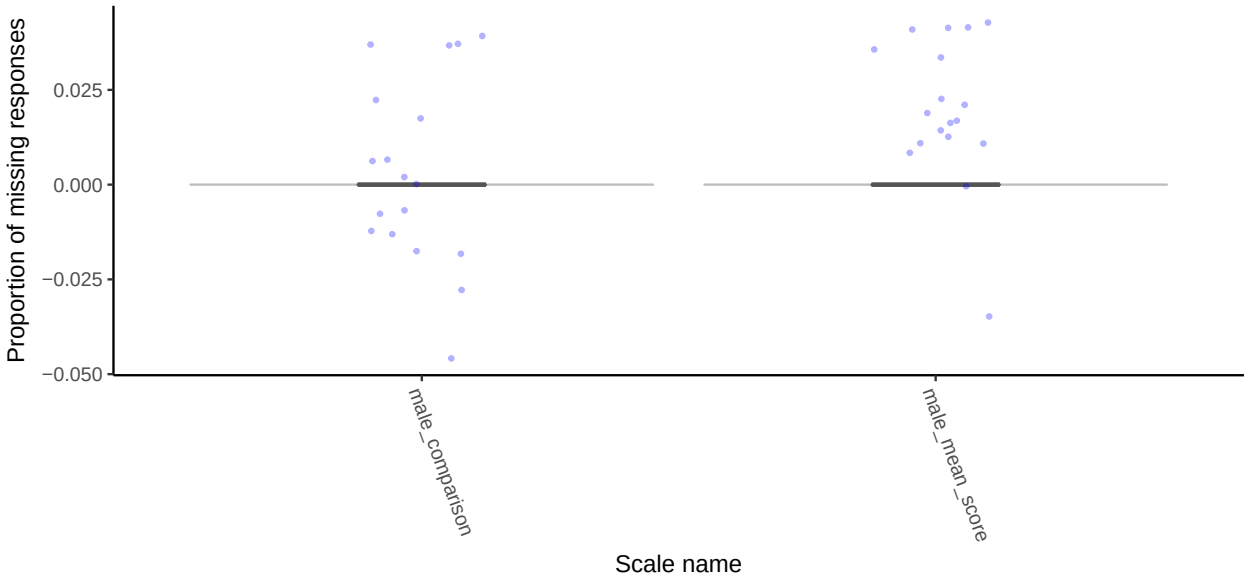
PDS

```
#We need a way to separate female from male responses
male_sids <- tds1_wave2_scored_c %>%
    filter(scored_scale == 'pds_gender', score =='0') %>%
    ungroup() %>%
    select(SID)

female_sids <- tds1_wave2_scored_c %>%
    filter(scored_scale == 'pds_gender', score =='1') %>%
    ungroup() %>%
```

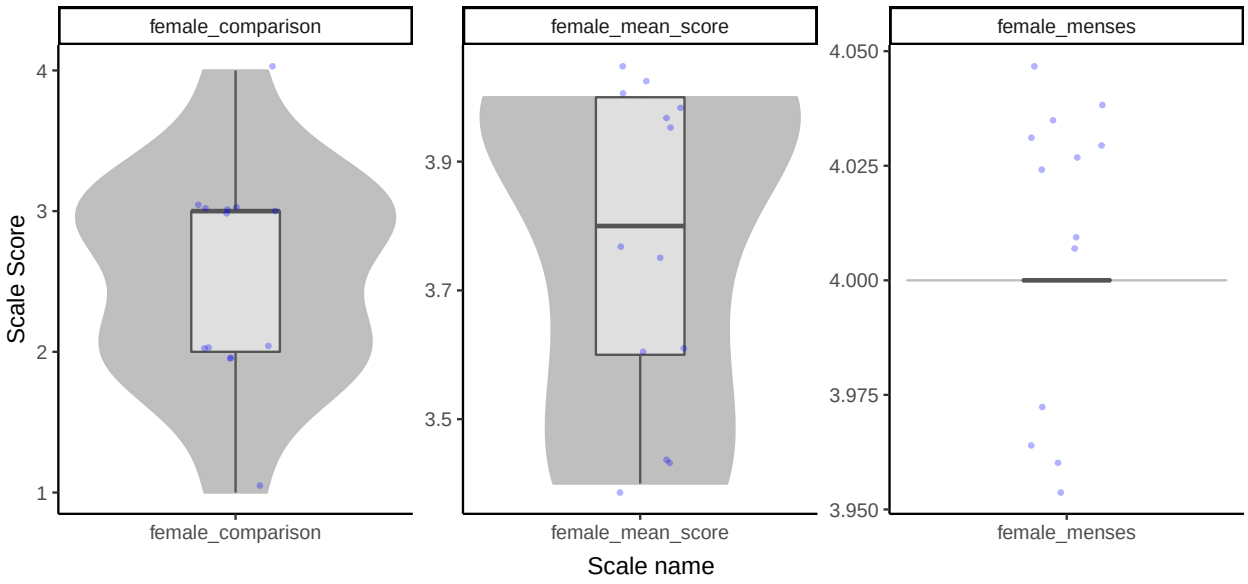
```
    select(SID)

scorequaltrics::plot_scored_scale(filter(tds1_wave2_scored_c,
                                         SID %in% male_sids$SID,
                                         grepl('^male', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'score') +
    facet_wrap(~scored_scale, scales = 'free') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```



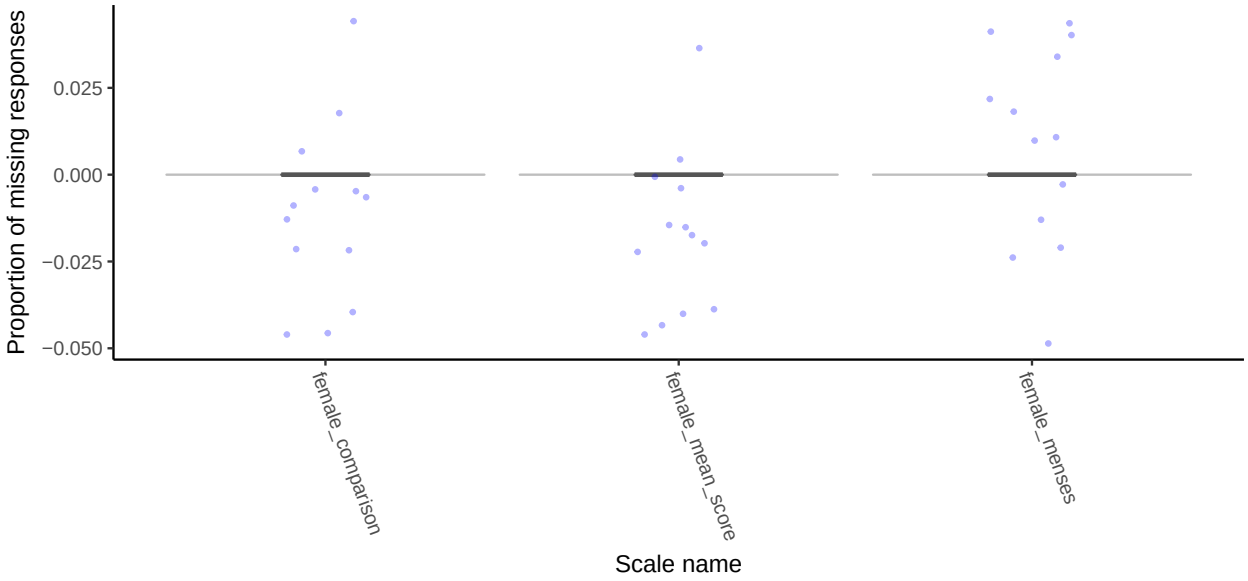

```
scorequaltrics::plot_scored_scale(filter(tds1_wave2_scored_c,
                                         SID %in% male_sids$SID,
                                         grepl('^male', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'p_missing')
```



```
scorequaltrics::plot_scored_scale(filter(tds1_wave2_scored_c,
                                         SID %in% female_sids$SID,
                                         grepl('^female', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'score') +
    facet_wrap(~scored_scale, scales = 'free') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```



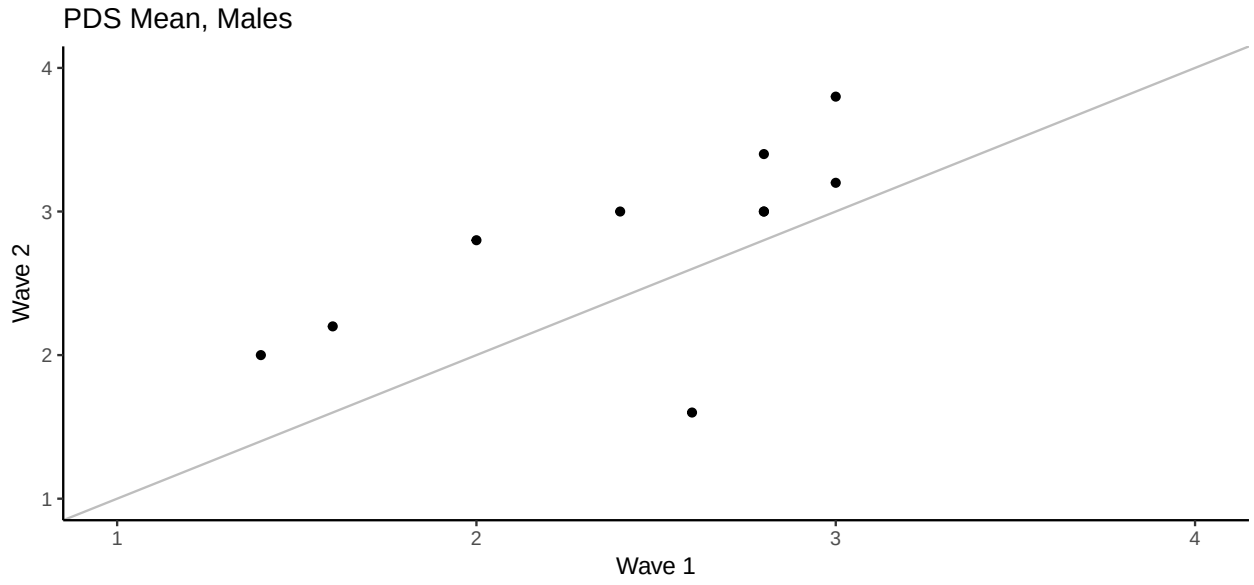

```
scorequaltrics::plot_scored_scale(filter(tds1_wave2_scored_c,
                                         SID %in% female_sids$SID,
                                         grepl('^female', scored_scale)),
                                  scale_regx = '^PDS$',
                                  type = 'p_missing')
```




```
tds1_wave2_scored_c_pds <- tds1_wave2_scored_c %>%
    filter(scale_name == 'PDS',
```

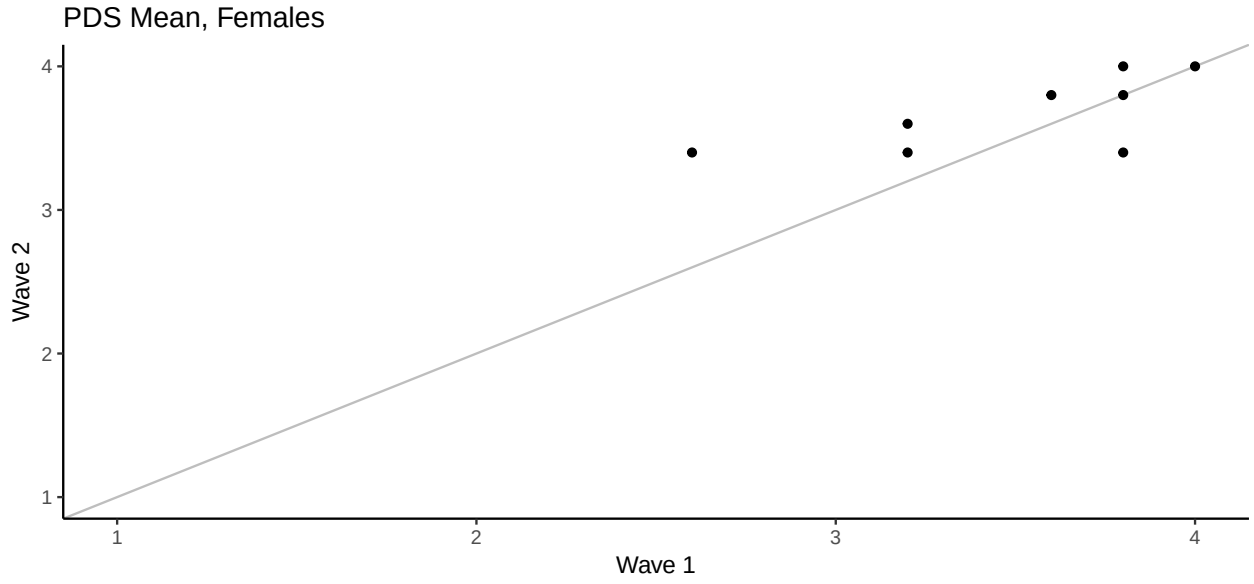
```
           grepl('mean_score', scored_scale))
tds1_wave1_scored_pds <- tds1_wave1_scored %>%
    filter(scale_name == 'PDS',
           grepl('mean_score', scored_scale))

tds1_wave12_pds <- bind_rows("Wave 1" = tds1_wave1_scored_pds,
                             "Wave 2" = tds1_wave2_scored_c_pds,
                             .id = 'wave') %>%
    select(wave, scored_scale, SID, score)

tds1_wave12_pds %>%
    filter(SID %in% male_sids$SID,
           grepl('^male', scored_scale)) %>%
    spread(wave, score) %>%
    mutate_at(vars(`Wave 1`, `Wave 2`), .funs = as.numeric) %>%
    ggplot(aes(x = `Wave 1`, y = `Wave 2`)) +
    geom_abline(intercept = 0, slope = 1, color = 'gray') +
    geom_point() +
    coord_cartesian(x = c(1,4), y = c(1, 4)) +
    theme_classic() +
    labs(title = 'PDS Mean, Males')
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```

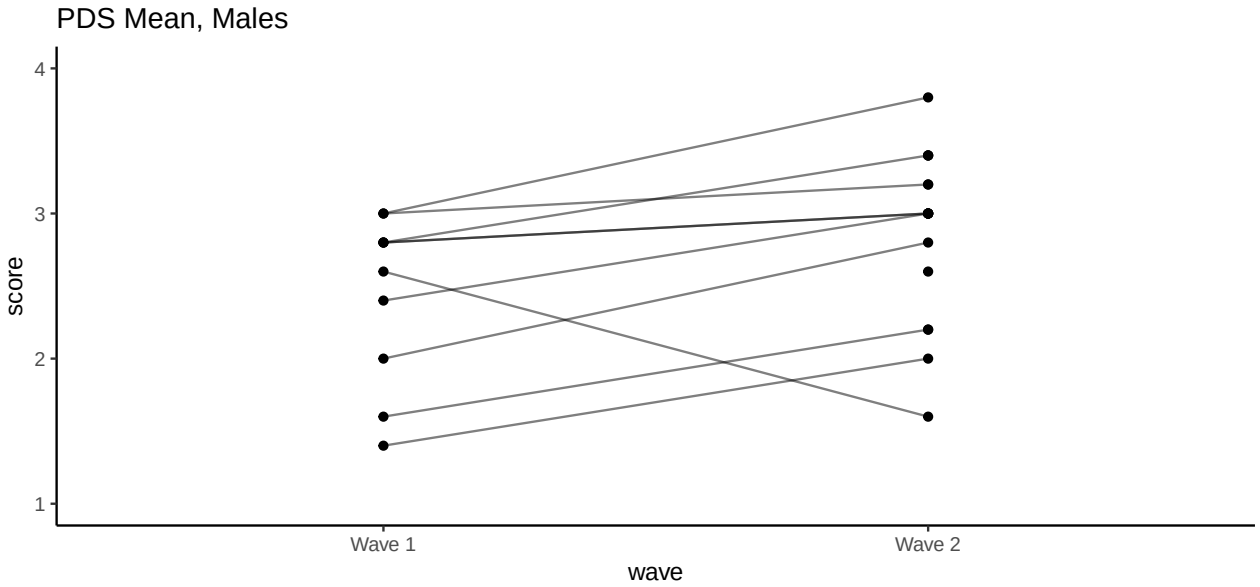



```
tds1_wave12_pds %>%
    filter(SID %in% female_sids$SID,
           grepl('^female', scored_scale)) %>%
    spread(wave, score) %>%
    mutate_at(vars(`Wave 1`, `Wave 2`), .funs = as.numeric) %>%
    ggplot(aes(x = `Wave 1`, y = `Wave 2`)) +
    geom_abline(intercept = 0, slope = 1, color = 'gray') +
    geom_point() +
    coord_cartesian(x = c(1,4), y = c(1, 4)) +
    theme_classic() +
    labs(title = 'PDS Mean, Females')
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

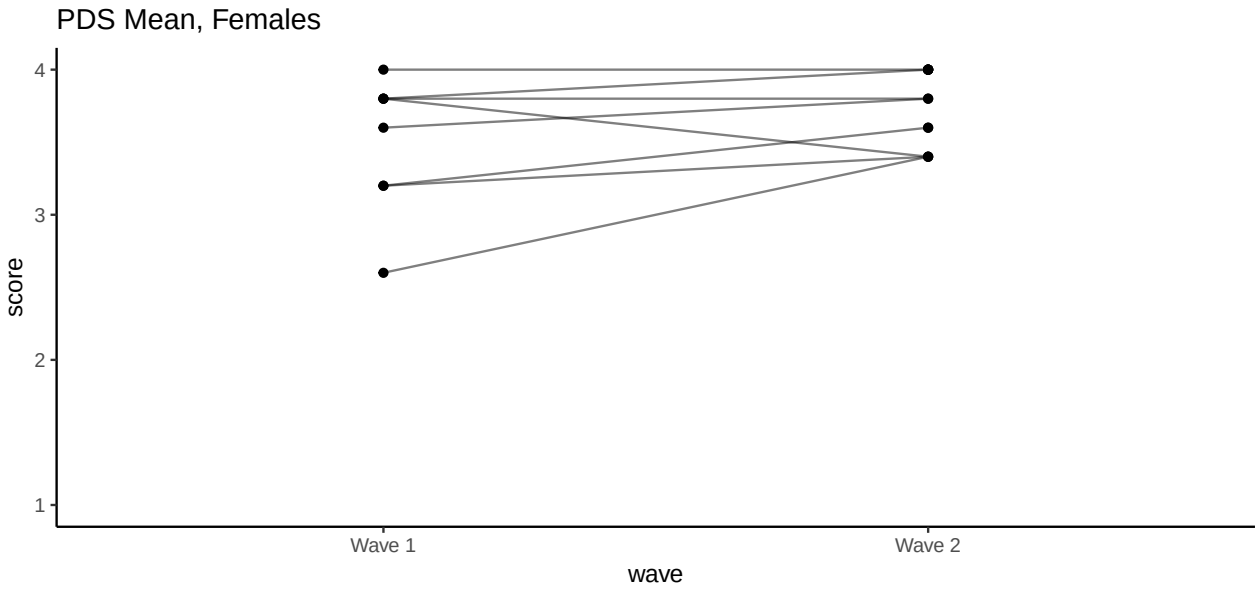



```
tds1_wave12_pds %>%
    filter(SID %in% male_sids$SID,
           grepl('^male', scored_scale)) %>%
    mutate(score = as.numeric(score)) %>%
    ggplot(aes(x = wave, y = score, group = SID)) +
    geom_line(alpha = .5) +
    geom_point() +
    coord_cartesian(y = c(1, 4)) +
    theme_classic() +
    labs(title = 'PDS Mean, Males')
```



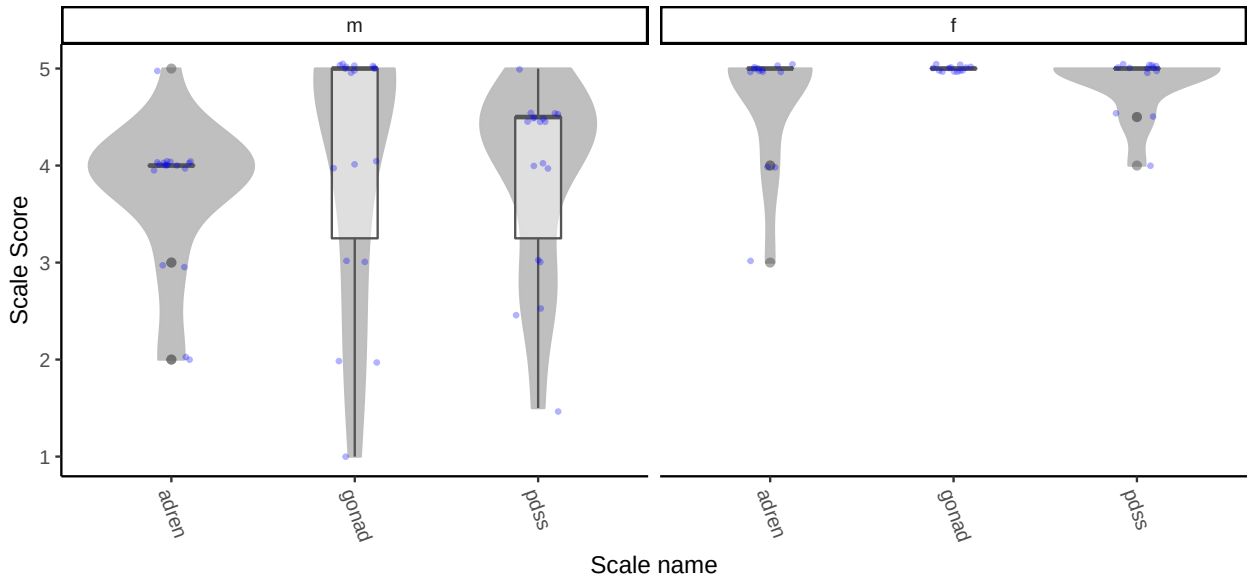

```
tds1_wave12_pds %>%
    filter(SID %in% female_sids$SID,
           grepl('^female', scored_scale)) %>%
    mutate(score = as.numeric(score)) %>%
```

```
    ggplot(aes(x = wave, y = score, group = SID)) +
    geom_line(alpha = .5) +
    geom_point() +
    coord_cartesian(y = c(1, 4)) +
    theme_classic() +
    labs(title = 'PDS Mean, Females')
```

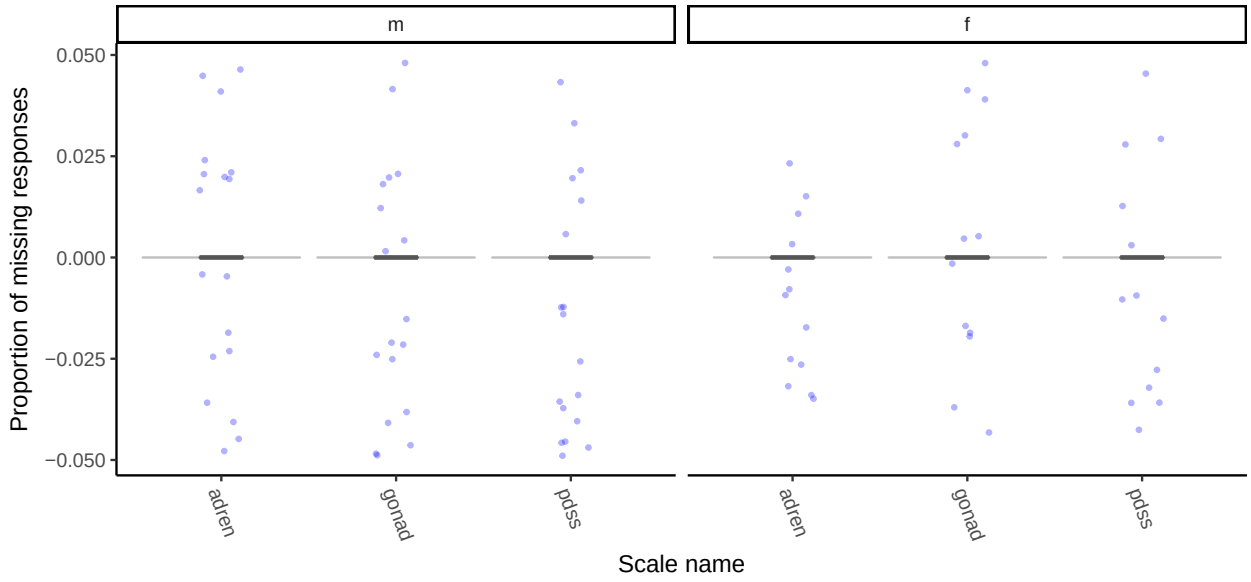


PDSS

```
tds1_wave2_scored_pdss_gender <- tds1_wave2_scored_pdss %>%
    filter(scored_scale == 'gender') %>%
    spread(scored_scale, score) %>%
    select(SID, gender)

tds1_wave2_scored_pdss_gender_wide <- tds1_wave2_scored_pdss %>%
    filter(scored_scale != 'gender') %>%
    left_join(tds1_wave2_scored_pdss_gender)

scorequaltrics::plot_scored_scale(tds1_wave2_scored_pdss_gender_wide,
                                  scale_regx = '^PDSS$',
                                  type = 'score') +
    facet_wrap(~factor(gender,
                       levels = pdss_gender_code,
                       labels = names(pdss_gender_code)))
```




```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_pdss_gender_wide,
                                  scale_regx = '^PDSS$',
                                  type = 'p_missing') +
    facet_wrap(~factor(gender,
                       levels = pdss_gender_code,
                       labels = names(pdss_gender_code)))
```




```
tds1_wave12_pdss <- bind_rows(wave_1 = tds1_wave1_scored_pdss_gender_wide,
                              wave_2 = tds1_wave2_scored_pdss_gender_wide,
                              .id = 'wave')

tds1_wave12_pdss %>%
    select(SID, wave, gender) %>%
    distinct(SID, wave, gender) %>%
    mutate(gender = factor(gender,
                           levels = pdss_gender_code,
                           labels = names(pdss_gender_code))) %>%
```

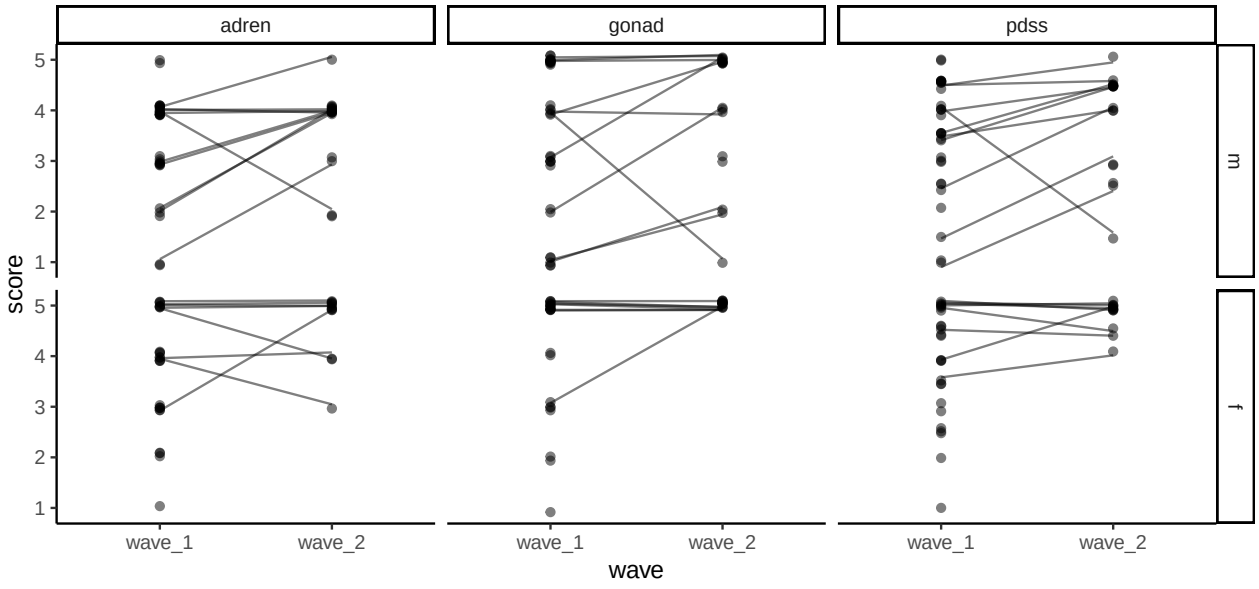
```
    spread(wave, gender) %>%
    select(wave_1, wave_2) %>%
    table(useNA = 'ifany') %>%
    knitr::kable(caption = 'Gender consistency across waves')
```

Table 16: Gender consistency across waves

	m	f	NA
m	10	0	22
f	0	8	17
NA	8	5	0

```
ajitter <- position_jitter(w = 0, h = .1)
tds1_wave12_pdss %>%
    mutate(score = as.numeric(score),
           gender = factor(gender,
                           levels = pdss_gender_code,
                           labels = names(pdss_gender_code))) %>%
    ggplot(aes(x = wave, y = score, group = SID)) +
    geom_point(alpha = .5, position = ajitter) +
    geom_line(alpha = .5, position = ajitter) +
    facet_grid(gender~scored_scale) +
    theme_classic()
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```



PSQI

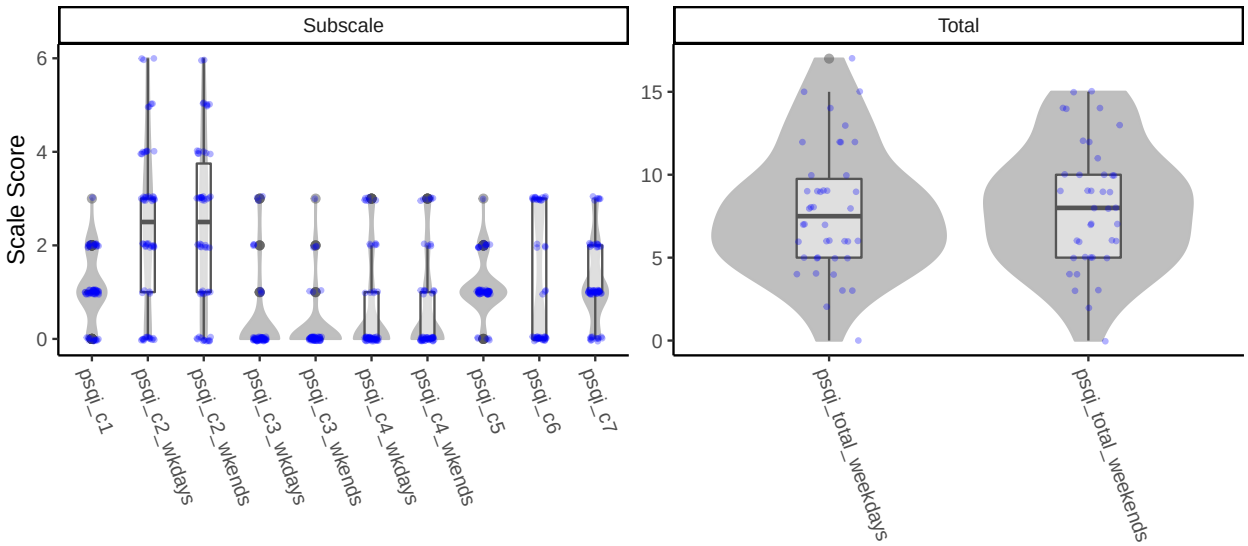
```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_psqi,
                                  scale_regx = '^PSQI$',
                                  type = 'score')+
    facet_wrap(~factor(as.numeric(grepl('total', scored_scale)),
```

```
                      levels = c(0, 1),
                      labels = c('Subscale', 'Total')),
               scales = 'free')
```

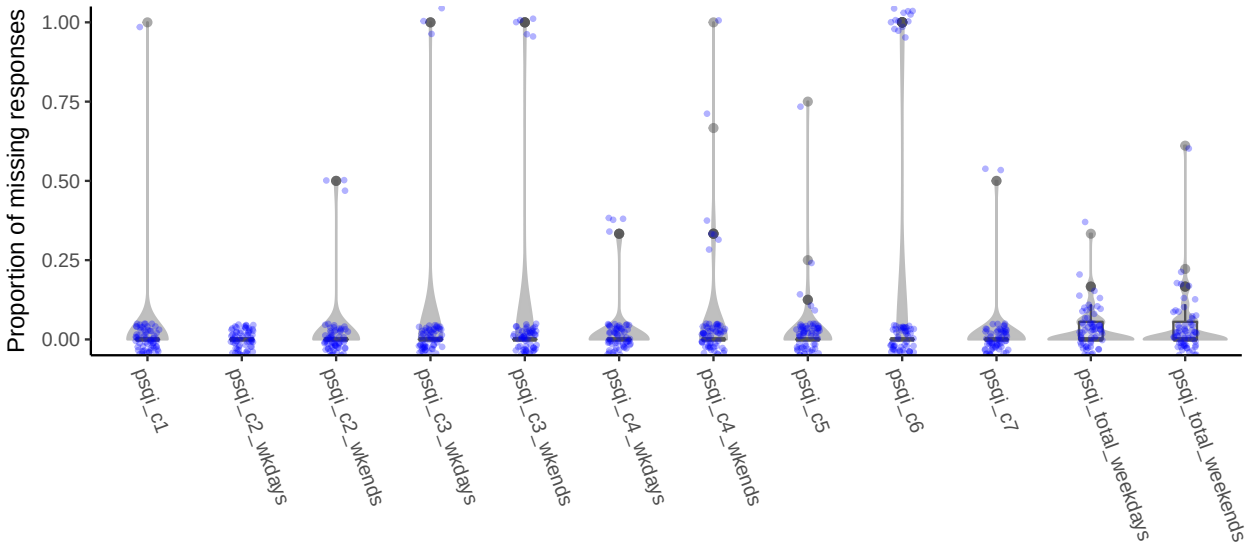
```
## Warning: Removed 88 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 88 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 88 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_psqi,
                                  scale_regx = '^PSQI$',
                                  type = 'p_missing')+
    coord_cartesian(y = c(0,1))
```



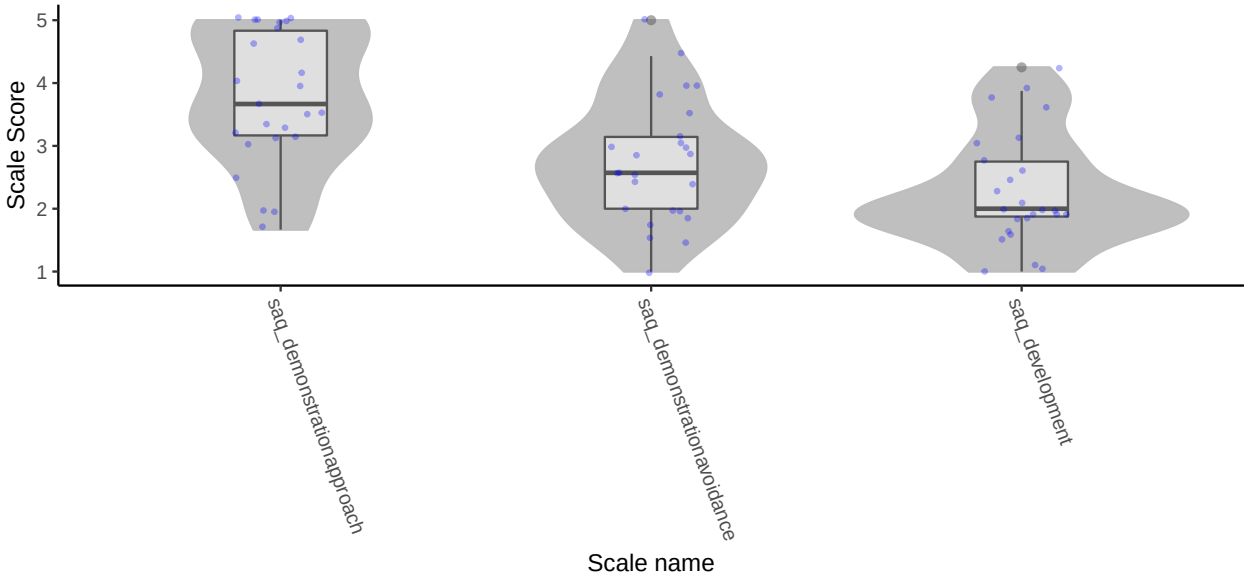
SAQ

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^SAQ$',
                                  type = 'score')
```

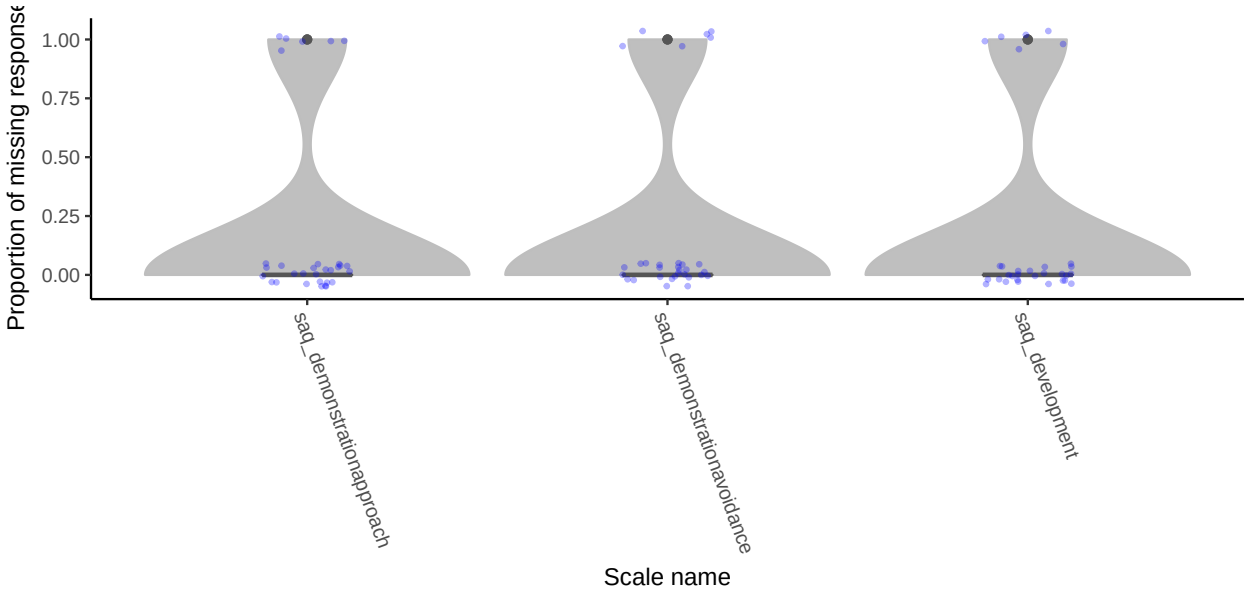
```
## Warning: Removed 18 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 18 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 18 rows containing missing values (geom_point).
```



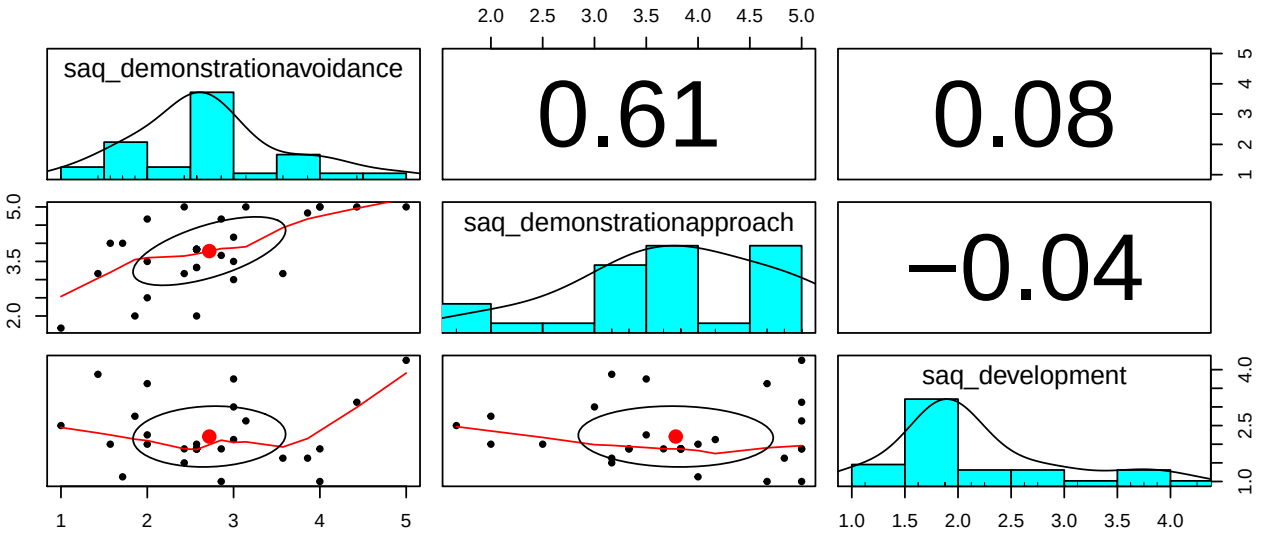

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^SAQ$',
                                  type = 'p_missing')
```



```r
tds1w2_SAQRubric <- tds1_wave2_scoring_data_long_c %>%
    filter(scale_name == 'SAQ')

tds1_wave2_saq_psych <- scorequaltrics::score_questionnaire(
    tds1_wave2_long_recoded_c_nodupes,
    tds1w2_SAQRubric,
    psych = T)
print(tds1_wave2_saq_psych)
```

```
## Call: scoreItems(keys = key_list, items = dataDF_w)
##
## (Unstandardized) Alpha:
##       saq_demonstrationavoidance saq_demonstrationapproach saq_development
## alpha                       0.87                       0.9             0.9
##
## Standard errors of unstandardized Alpha:
##       saq_demonstrationavoidance saq_demonstrationapproach saq_development
## ASE                        0.074                     0.074            0.06
##
## Average item correlation:
##           saq_demonstrationavoidance saq_demonstrationapproach
## average.r                       0.49                       0.6
##           saq_development
## average.r            0.53
##
##  Guttman 6* reliability:
##          saq_demonstrationavoidance saq_demonstrationapproach
## Lambda.6                       0.98                      0.99
##          saq_development
## Lambda.6            0.99
##
## Signal/Noise based upon av.r :
##              saq_demonstrationavoidance saq_demonstrationapproach
## Signal/Noise                        6.8                       8.8
##              saq_development
## Signal/Noise             9.2
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##                            saq_demonstrationavoidance
## saq_demonstrationavoidance                      0.871
## saq_demonstrationapproach                       0.610
## saq_development                                 0.079
##                            saq_demonstrationapproach saq_development
## saq_demonstrationavoidance                      0.69           0.090
## saq_demonstrationapproach                       0.90          -0.044
## saq_development                                -0.04           0.902
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

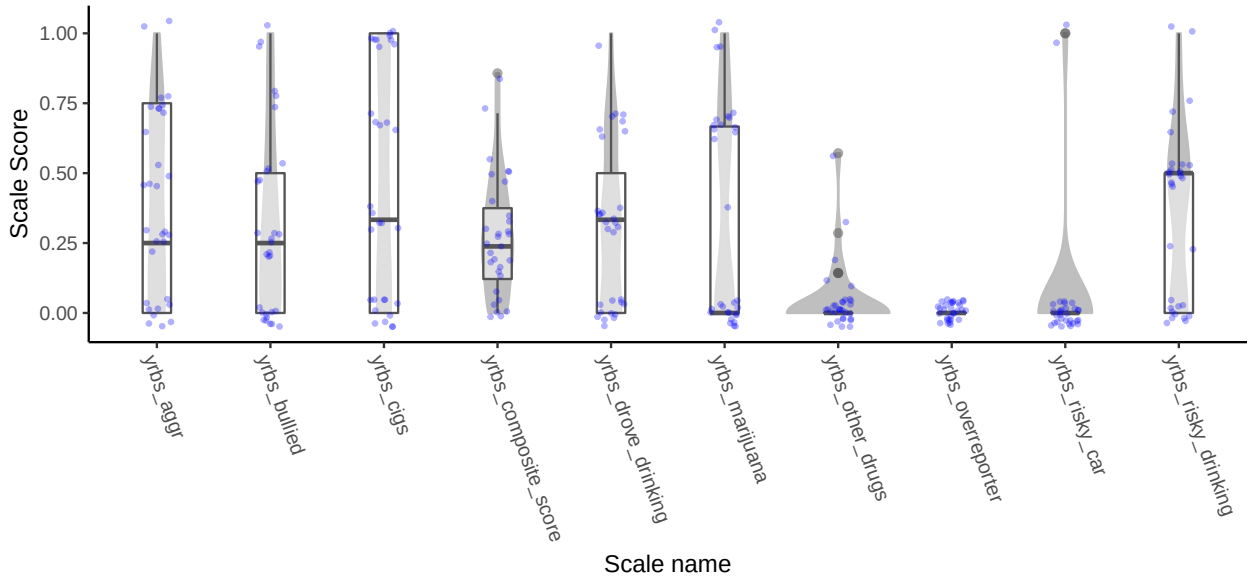
```
pairs.panels(tds1_wave2_saq_psych$scores)
```



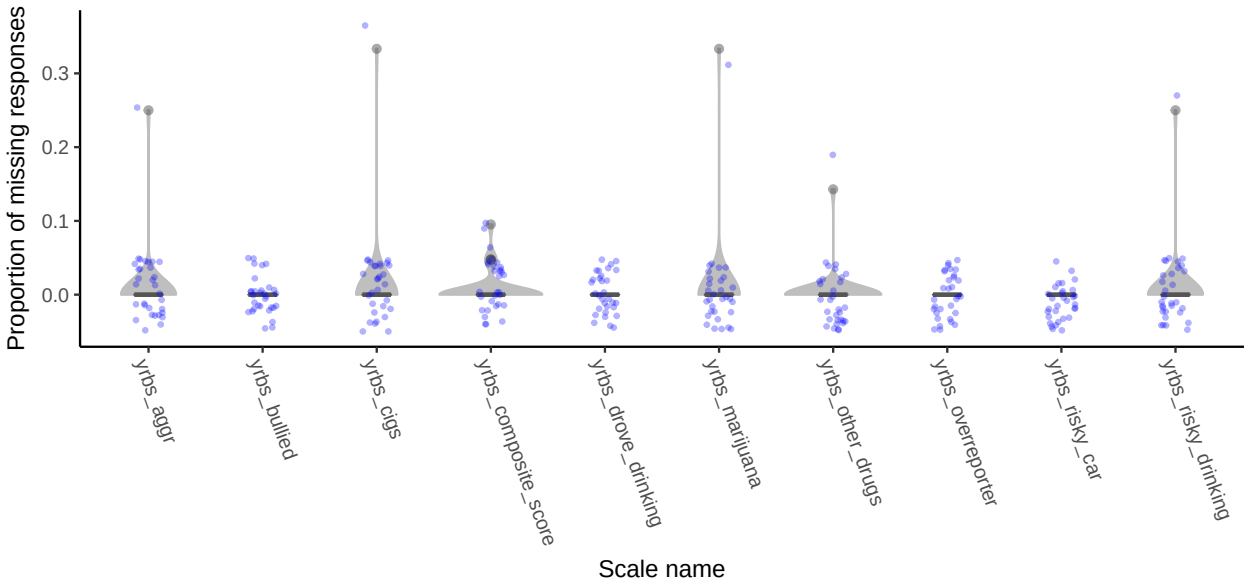
YRBS

This has been scored according to the CDC manual. Items have been scored with either 1 or 0, with the mean calculated for each subscale. This gives you the proportion of endorsed items in that scale. To convert to a sum of endorsed items, simply multiply the scaled score by the number of total items (`n_missing` + `n_items`).

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^YRBS$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 360-70))
```




```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^YRBS$',
                                  type = 'p_missing')
```



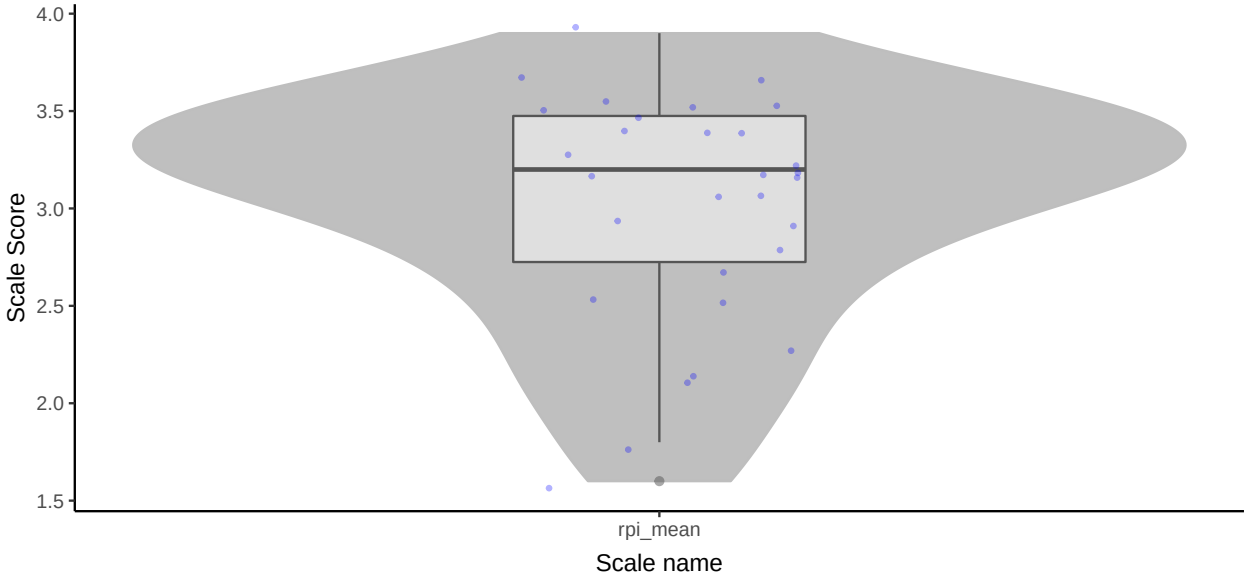
RPI

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^RPI_part2$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

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

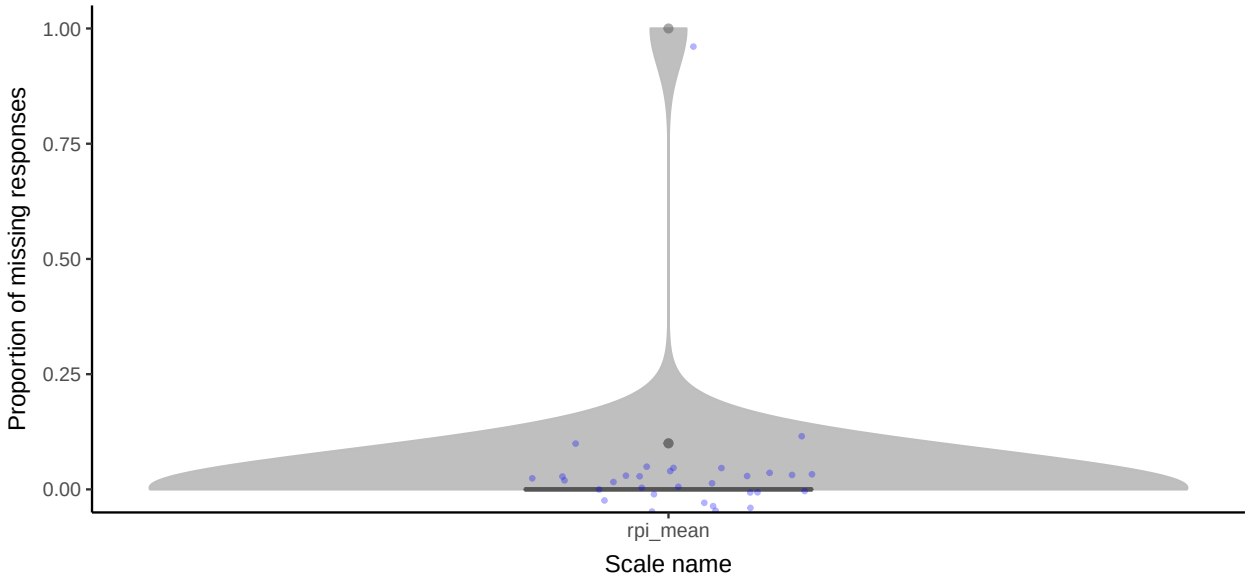
```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^RPI_part2$',
                                  type = 'p_missing') +
```

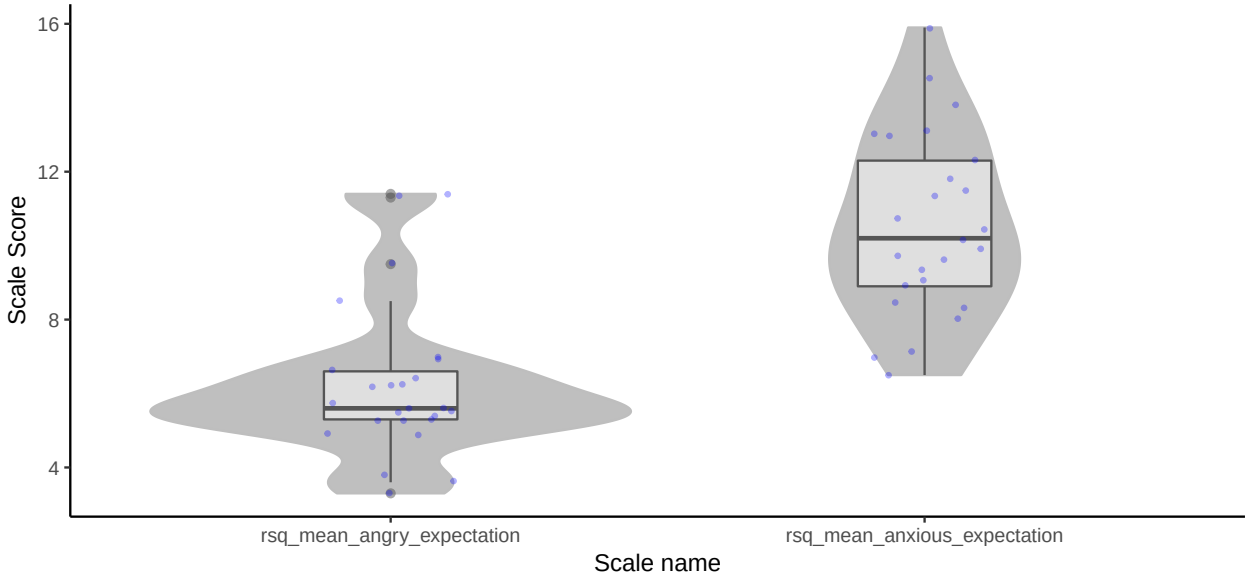
```
    coord_cartesian(y = c(0,1)) +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```



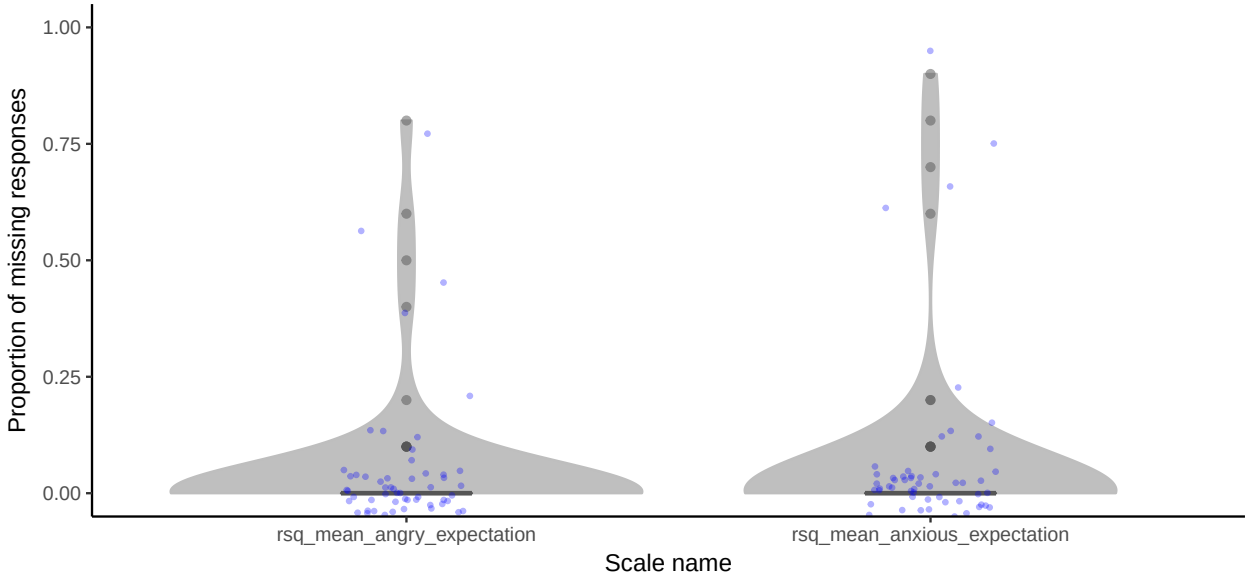
RSQ - Modified

```
scorequaltrics::plot_scored_scale(tds1_wave2_scored_c,
                                  scale_regx = '^RSQMod_part2$',
                                  type = 'score') +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```

```
## Warning: Removed 12 rows containing non-finite values (stat_ydensity).
## Warning: Removed 12 rows containing non-finite values (stat_boxplot).
## Warning: Removed 12 rows containing missing values (geom_point).
```



```
scorequaltrics::plot_scored_scale(tds1_wave1_scored,
                                  scale_regx = '^RSQMod_part2$',
                                  type = 'p_missing') +
    coord_cartesian(y = c(0,1)) +
    theme(axis.text.x = element_text(angle = 0, hjust = .5))
```



Write the data

```
list_of_scales_c <- as.list(unique(tds1_wave2_scored_c$scale_name))
list_of_scales_p <- as.list(unique(tds1_wave2_scored_p$scale_name))
tds1_wave2_metadata <- list(wave = 2, questionnaires = 'TDS1')


throwaway <- lapply(list_of_scales_c,
                    function(aScale){
                        file_name <- paste0(TDS1_name, '-',
                                            wave2_name, '-',
                                            scorequaltrics::make_nice_scale_fname(aScale),
                                            '.csv')
                        scorequaltrics::write_widened_scored_scale(
                            tds1_wave2_scored_c,
                            scale_names = aScale,
                            dir_name = output_file_dir,
                            file_name = file_name,
                            metadata = tds1_wave2_metadata)
                    })

throwaway <- lapply(list_of_scales_p,
                    function(aScale){
                        file_name <- paste0(TDS1_name, '-',
                                            wave2_name, '-',
                                            scorequaltrics::make_nice_scale_fname(aScale),
                                            '.csv')
```

```
                        scorequaltrics::write_widened_scored_scale(
                            tds1_wave2_scored_p,
                            scale_names = aScale,
                            dir_name = output_file_dir,
                            file_name = file_name,
                            metadata = tds1_wave2_metadata)
                    })

tds1_wave2_pdss_filename <-  paste0(TDS1_name, '-',
                                    wave2_name, '-',
                                    'PDSS.csv')

scorequaltrics::write_widened_scored_scale(
    tds1_wave2_scored_pdss,
    scale_names = 'PDSS',
    dir_name = output_file_dir,
    file_name = tds1_wave2_pdss_filename,
    metadata = tds1_wave2_metadata)

tds1_wave2_psqi_filename <-  paste0(TDS1_name, '-',
                                    wave2_name, '-',
                                    'PSQI.csv')

scorequaltrics::write_widened_scored_scale(
    tds1_wave2_scored_psqi,
    scale_names = 'PSQI',
    dir_name = output_file_dir,
    file_name = tds1_wave2_psqi_filename,
    metadata = tds1_wave2_metadata)
```