Data Analysis

0 - Info about the data and code

The following code will load in raw data from the Parenting Across Cultures data set. After some data wrangling, it will combine items that are common among the different rater versions into subscales for internalizing and externalizing symptoms. Lastly, we will analyze the rater disagreement and age of the child as a predictor of rater disagreement with four different statistical approaches: difference scores, Pearson correlation, intraclass correlation, and the tri-factor model. With that, we can explore whether different statistical approaches lead to divergent conclusions regarding rater disagreement and age as a predictor of rater disagreement.

1 - Loading libraries

```
# Loading Tidyverse for data handling
library(tidyverse)
```

```
Warning: package 'ggplot2' was built under R version 4.3.3

Warning: package 'tidyr' was built under R version 4.3.3

Warning: package 'readr' was built under R version 4.3.3

Warning: package 'purrr' was built under R version 4.3.3

Warning: package 'lubridate' was built under R version 4.3.3

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
  dplyr     1.1.4     readr     2.1.5
  forcats   1.0.0     stringr   1.5.1
  ggplot2   3.5.1     tibble    3.2.1
  lubridate 1.9.4     tidyr     1.3.1
  purrr     1.0.4
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
  dplyr::filter() masks stats::filter()
  dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# Loading corrr for calculating correlations in a way that is more easily compatible with Tidyverse
library(corrr)
```

```
Warning: package 'corrr' was built under R version 4.3.3
```

```
# Loading correlation for caluclating correlations with confidence intervals
library(correlation)
```

```
Warning: package 'correlation' was built under R version 4.3.3
```

```
# Loading irr for calculating intraclass correlations
```

```r
library(irr)
```

```
Warning: package 'irr' was built under R version 4.3.3

Loading required package: lpSolve

Warning: package 'lpSolve' was built under R version 4.3.3
```

```r
# Loading lavaan to run SEMs
library(lavaan)
```

```
Warning: package 'lavaan' was built under R version 4.3.3

This is lavaan 0.6-19
lavaan is FREE software! Please report any bugs.
```

```r
# Load MplusAutomation to interact with Mplus through R
library("MplusAutomation")
```

```
Warning: package 'MplusAutomation' was built under R version 4.3.3

Version:  1.1.1
We work hard to write this free software. Please help us get credit by citing:

Hallquist, M. N. & Wiley, J. F. (2018). MplusAutomation: An R Package for Facilitating Large-Scale
Latent Variable Analyses in Mplus. Structural Equation Modeling, 25, 621-638. doi:
10.1080/10705511.2017.1402334.

-- see citation("MplusAutomation").
```

```r
# Loading psych to calculate Chronbach's alpha
library(psych)
```

```
Warning: package 'psych' was built under R version 4.3.3


Attaching package: 'psych'

The following object is masked from 'package:lavaan':

    cor2cov

The following objects are masked from 'package:ggplot2':

    %+%, alpha
```

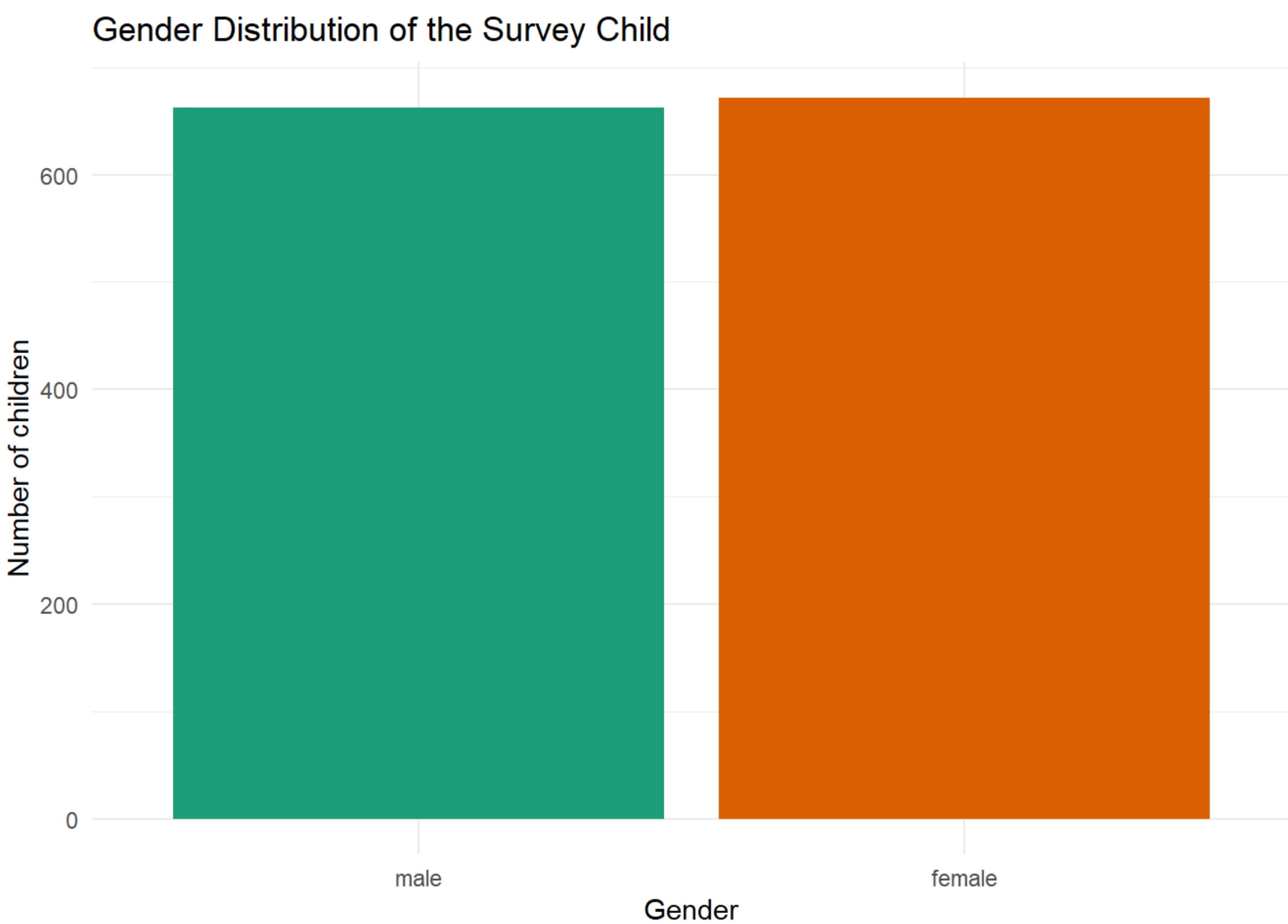
2 - Loading the data

```
data_df_long <- readRDS(file = "data_df_long.rds")
data_df_wide <- readRDS(file = "data_df_wide.rds")
```

3 - Getting to know the data

Gender distribution:

```
data_df_wide %>%
  filter(!is.na(ChGender)) %>%
  ggplot(aes(x = ChGender, fill = ChGender)) +
  geom_bar(show.legend = FALSE) +
  scale_fill_brewer(palette = "Dark2") +
  theme_minimal() +
  labs(title = "Gender Distribution of the Survey Child",
       x = "Gender",
       y = "Number of children")
```



The data set has 663 males and 672 females:

```
data_df_wide %>% count(ChGender)
```

```
# A tibble: 3 × 2
```

```
  ChGender     n
  <fct>    <int>
1 male       663
2 female     672
3 <NA>       119
```

Age ranges at wave 1:

```r
data_df_wide %>% count(round(y1Age))
```

```
# A tibble: 6 × 2
  `round(y1Age)`     n
           <dbl> <int>
1              7    50
2              8   570
3              9   518
4             10   127
5             11     5
6             NA   184
```

The youngest children were 7 years old at wave 1 and the oldest were 11 years old.

Age ranges at wave 13:

```r
data_df_wide %>% count(round(y13Age))
```

```
# A tibble: 6 × 2
  `round(y13Age)`     n
            <dbl> <int>
1              21    14
2              22    92
3              23    95
4              24    49
5              25     6
6              NA  1198
```

The oldest children were 25 at the last wave, wave 13. The youngest were 20 years old.

Distribution of data collection locations:

```r
data_df_wide %>% count(IDN)
```

```
# A tibble: 14 × 2
   IDN                 n
   <fct>           <int>
 1 China-Shanghai    123
 2 Italy-Naples      102
 3 Italy-Rome        111
 4 Kenya             100
 5 Philippines       120
 6 Thailand          120
 7 Sweden            129
```

```
 8 US-AA             102
 9 US-EA             110
10 US-H               99
11 Colombia          108
12 Jordan            114
13 China-Chongqing   114
14 <NA>                2
```

Looking at the average number of years with formal education in our sample:

```
data_df_wide %>%
  # reframe() works like summarize, but allows for multiple rows, here once for mothers and one for
fathers
  reframe(
    Parent = c("Mothers", "Fathers"),
    mean_edu = c(mean(m1educ_m, na.rm = TRUE), mean(f1educ_f, na.rm = TRUE)),
    min_edu = c(min(m1educ_m, na.rm = TRUE), min(f1educ_f, na.rm = TRUE)),
    max_edu = c(max(m1educ_m, na.rm = TRUE), max(f1educ_f, na.rm = TRUE))
    )
```

```
# A tibble: 2 × 4
  Parent  mean_edu min_edu max_edu
  <chr>      <dbl>   <dbl>   <dbl>
1 Mothers     12.8       0      31
2 Fathers     12.9       0      31
```

The average number of years with formal schooling for the parents in the data set were almost 13 years for both mothers and fathers. This corresponds to completing high school (if you take pre-school into account). The minimum values are 0 years of formal schooling for both fathers and mothers, while the maximum years of schooling are 31 for mothers and fathers.

Looking at the average number of years with formal education in our sample per location:

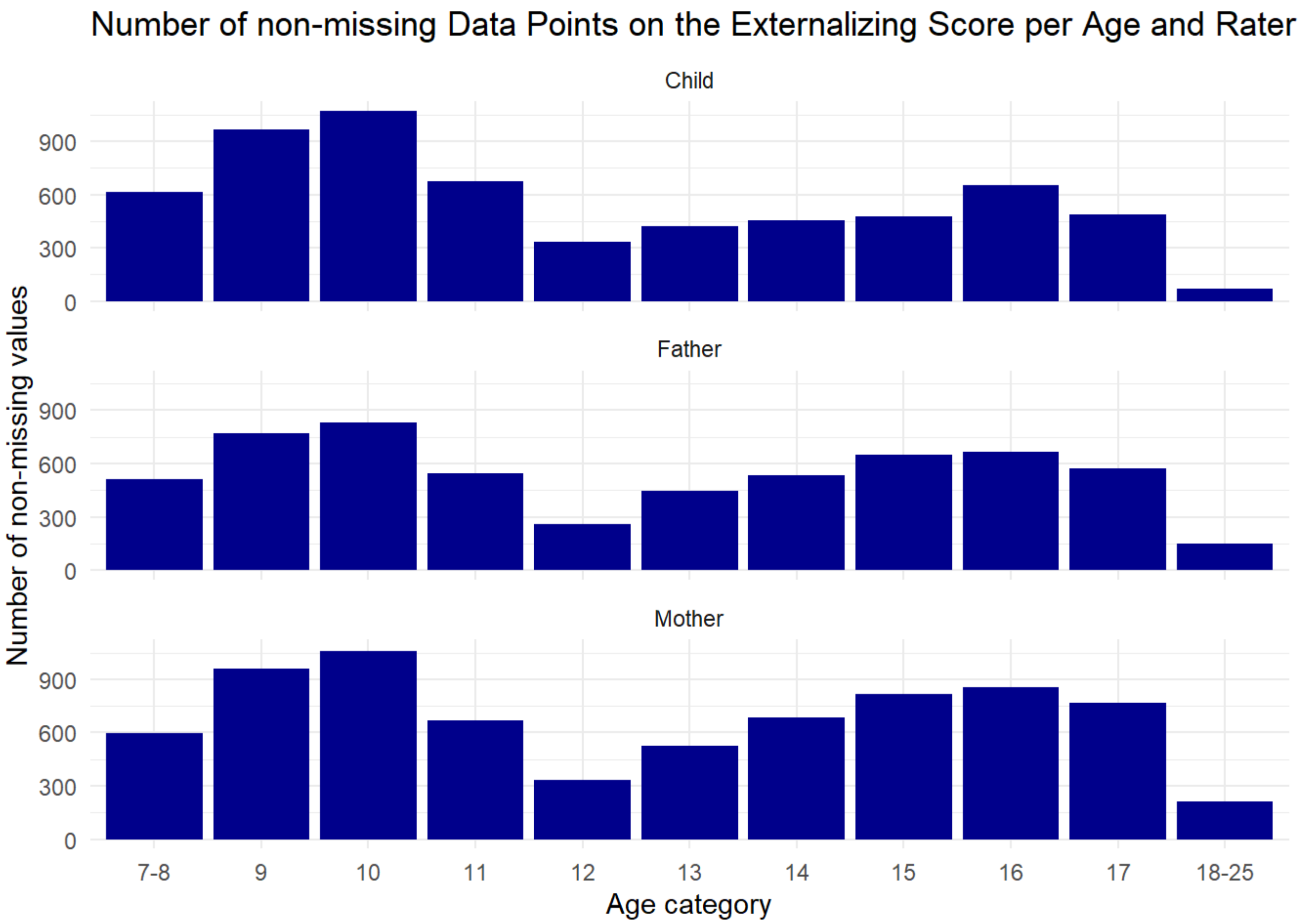
```
data_df_wide %>%
  # reframe() works like summarize, but allows for multiple rows, here once for mothers and one for
fathers
  filter(!is.na(IDN)) %>%
  group_by(IDN) %>%
  reframe(
    Parent = c("Mothers", "Fathers"),
    mean_edu = c(mean(m1educ_m, na.rm = TRUE), mean(f1educ_f, na.rm = TRUE)),
    min_edu = c(min(m1educ_m, na.rm = TRUE), min(f1educ_f, na.rm = TRUE)),
    max_edu = c(max(m1educ_m, na.rm = TRUE), max(f1educ_f, na.rm = TRUE))
    ) %>%
  ungroup()
```

```
Warning: There were 4 warnings in `reframe()`.
The first warning was:
ℹ In argument: `min_edu = c(min(m1educ_m, na.rm = TRUE), min(f1educ_f, na.rm =
  TRUE))`.
ℹ In group 13: `IDN = China-Chongqing`.
Caused by warning in `min()`:
! no non-missing arguments to min; returning Inf
```

```
ℹ Run `dplyr::last_dplyr_warnings()` to see the 3 remaining warnings.

# A tibble: 26 × 5
   IDN            Parent  mean_edu min_edu max_edu
   <fct>          <chr>      <dbl>   <dbl>   <dbl>
 1 China-Shanghai Mothers     13.6       5      19
 2 China-Shanghai Fathers     14         8      24
 3 Italy-Naples   Mothers     10.1       0      22
 4 Italy-Naples   Fathers     10.7       3      24
 5 Italy-Rome     Mothers     14.1       5      25
 6 Italy-Rome     Fathers     13.8       5      31
 7 Kenya          Mothers     10.7       0      20
 8 Kenya          Fathers     12.3       3      29
 9 Philippines    Mothers     13.6       5      27
10 Philippines    Fathers     13.9       5      25
# ℹ 16 more rows
```

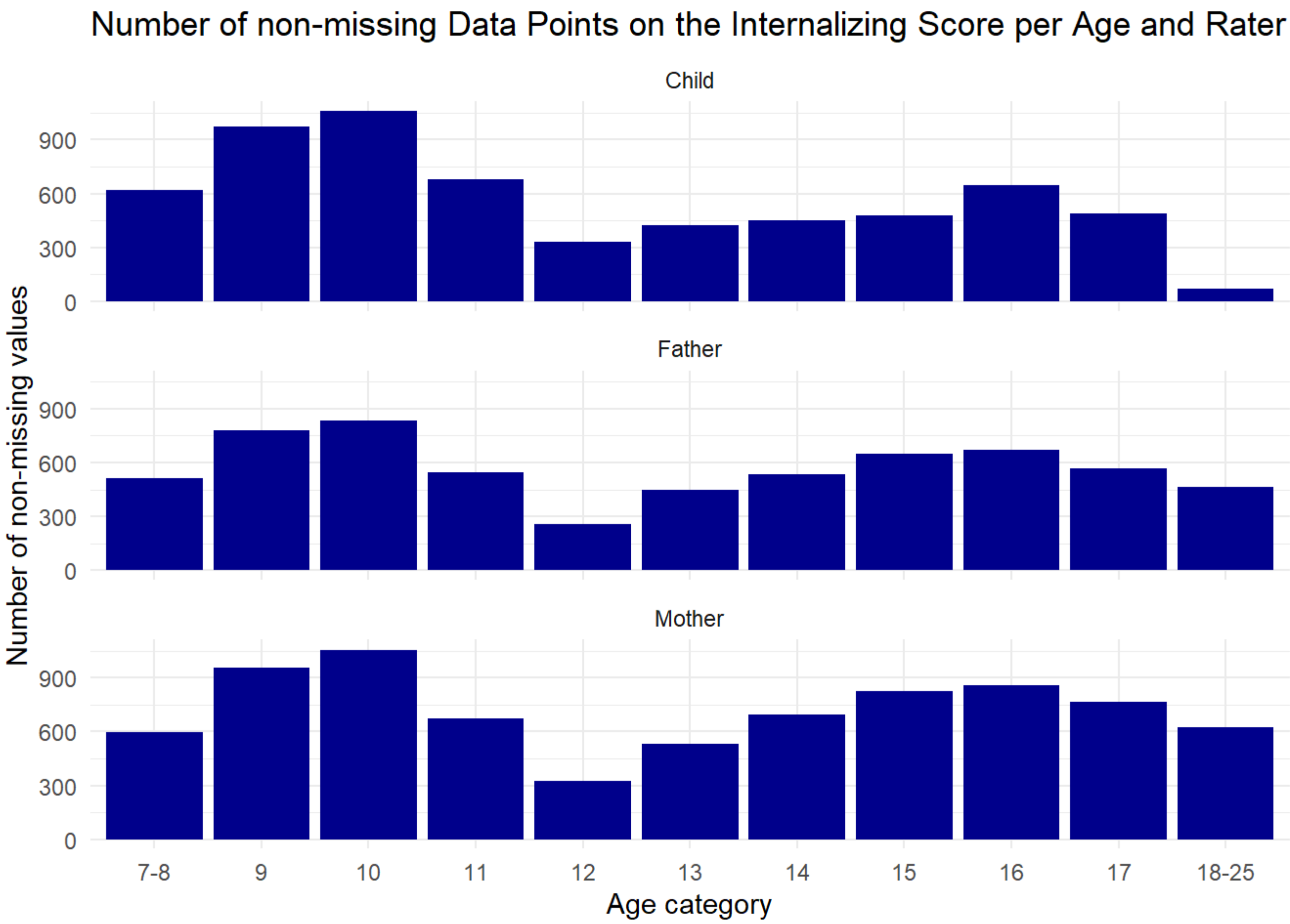
Looking at the number of non-missing data points on the externalizing score per age and rater:

```
data_df_long %>%
    select(ext_scale_f, ext_scale_m, ext_scale_c, Age_cat) %>%
    pivot_longer(cols = c(ext_scale_f, ext_scale_m, ext_scale_c), names_to = "rater", values_to =
"rating") %>%
  mutate(
        rater = case_when(
            rater == "ext_scale_f" ~ "Father",
            rater == "ext_scale_m" ~ "Mother",
            rater == "ext_scale_c" ~ "Child"
        )) %>%
  filter(!is.na(rating)) %>%
  filter(!is.na(Age_cat)) %>%
  group_by(rater, Age_cat) %>%
  count() %>%
  ggplot(aes(x = Age_cat, y = n)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  facet_wrap(vars(rater), nrow = 3) +
  theme_minimal() +
  labs(title = "Number of non-missing Data Points on the Externalizing Score per Age and Rater",
       x = "Age category",
       y = "Number of non-missing values")
```



Looking at the number of non-missing data points on the internalizing score per age and rater:

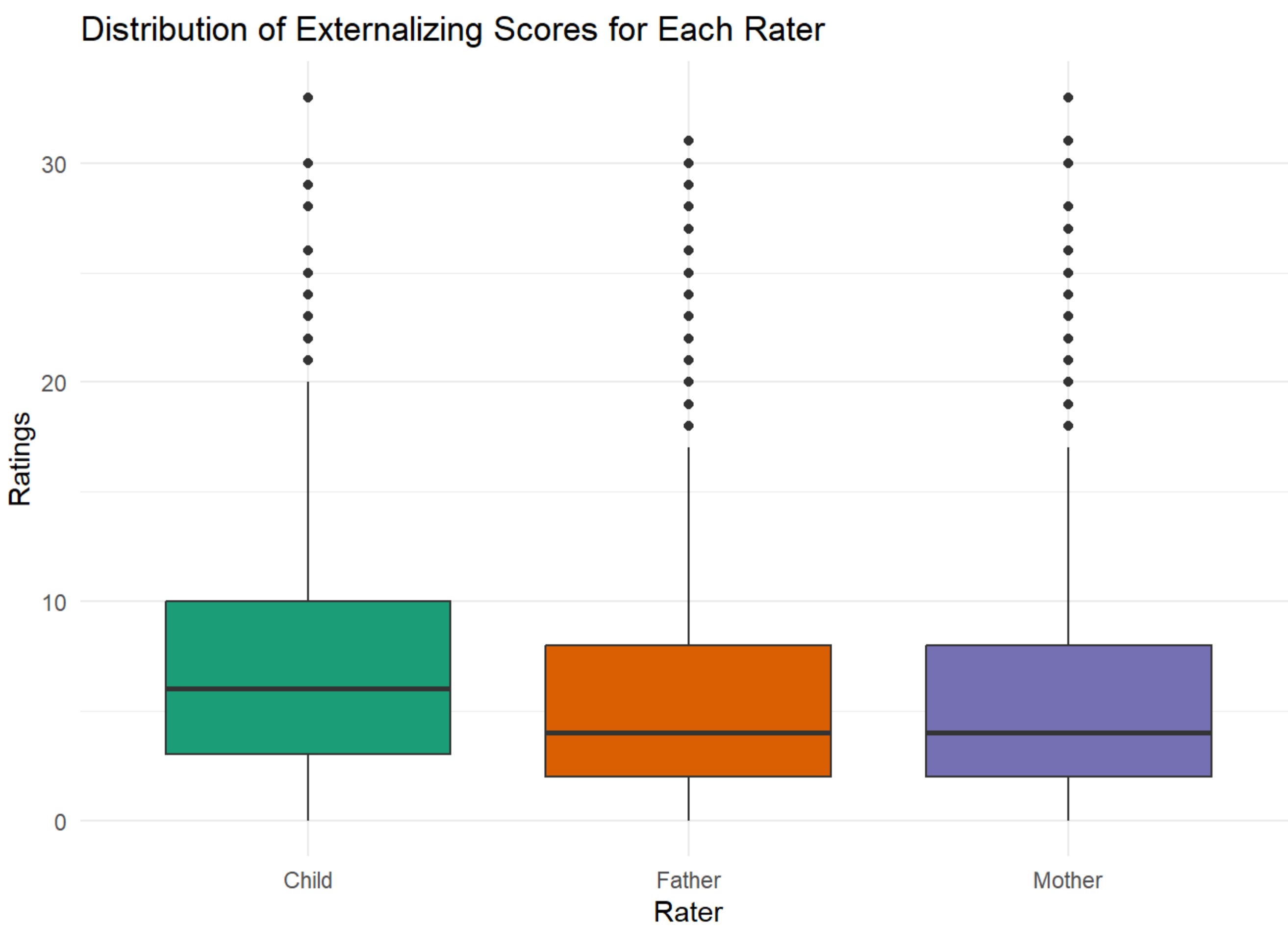
```
data_df_long %>%
    select(int_scale_f, int_scale_m, int_scale_c, Age_cat) %>%
    pivot_longer(cols = c(int_scale_f, int_scale_m, int_scale_c), names_to = "rater", values_to =
"rating") %>%
  mutate(
        rater = case_when(
            rater == "int_scale_f" ~ "Father",
            rater == "int_scale_m" ~ "Mother",
            rater == "int_scale_c" ~ "Child"
        )) %>%
  filter(!is.na(rating)) %>%
  filter(!is.na(Age_cat)) %>%
  group_by(rater, Age_cat) %>%
  count() %>%
  ggplot(aes(x = Age_cat, y = n)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  facet_wrap(vars(rater), nrow = 3) +
  theme_minimal() +
  labs(title = "Number of non-missing Data Points on the Internalizing Score per Age and Rater",
       x = "Age category",
       y = "Number of non-missing values")
```



Box plot of the distribution of externalizing scores for the different raters:

```
data_df_long %>%
  select(ext_scale_f, ext_scale_m, ext_scale_c) %>%
  pivot_longer(cols = c(ext_scale_f, ext_scale_m, ext_scale_c), names_to = "rater", values_to =
"rating") %>%
  mutate(
    rater = case_when(
      rater == "ext_scale_f" ~ "Father",
      rater == "ext_scale_m" ~ "Mother",
      rater == "ext_scale_c" ~ "Child"
    )
  ) %>%
  ggplot(aes(x = rater, y = rating, fill = rater)) +
  geom_boxplot(show.legend = FALSE) +
  scale_fill_brewer(palette = "Dark2") +
  theme_minimal() +
  labs(
    title = "Distribution of Externalizing Scores for Each Rater",
    x = "Rater",
    y = "Ratings"
  )
```

```
Warning: Removed 11498 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

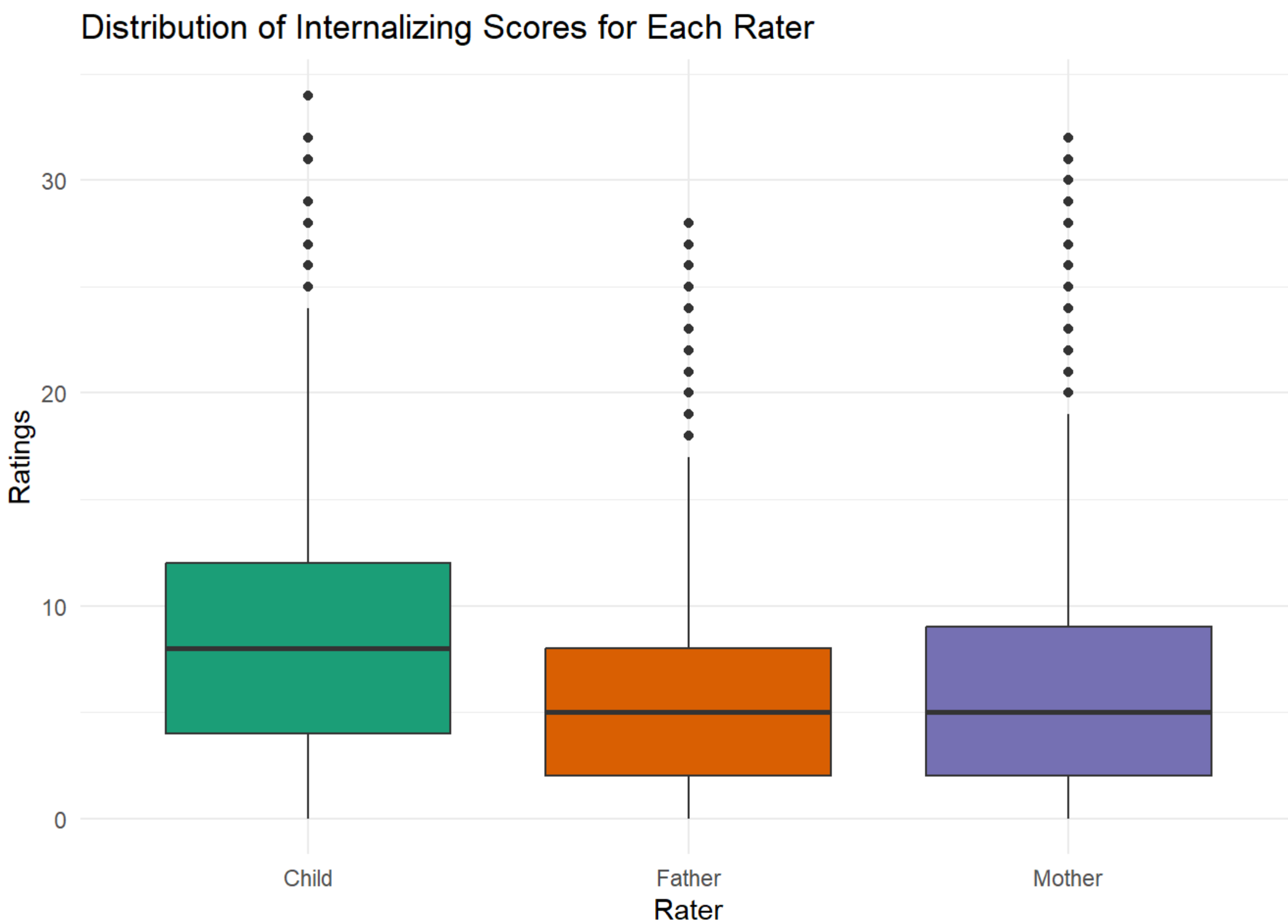


Box plot of the distribution of internalizing scores for the different raters:

```
data_df_long %>%
  select(int_scale_f, int_scale_m, int_scale_c) %>%
  pivot_longer(cols = c(int_scale_f, int_scale_m, int_scale_c), names_to = "rater", values_to =
"rating") %>%
  mutate(
    rater = case_when(
      rater == "int_scale_f" ~ "Father",
      rater == "int_scale_m" ~ "Mother",
      rater == "int_scale_c" ~ "Child"
    )
  ) %>%
  ggplot(aes(x = rater, y = rating, fill = rater)) +
  geom_boxplot(show.legend = FALSE) +
  scale_fill_brewer(palette = "Dark2") +
  theme_minimal() +
  labs(
    title = "Distribution of Internalizing Scores for Each Rater",
    x = "Rater",
    y = "Ratings"
  )
```

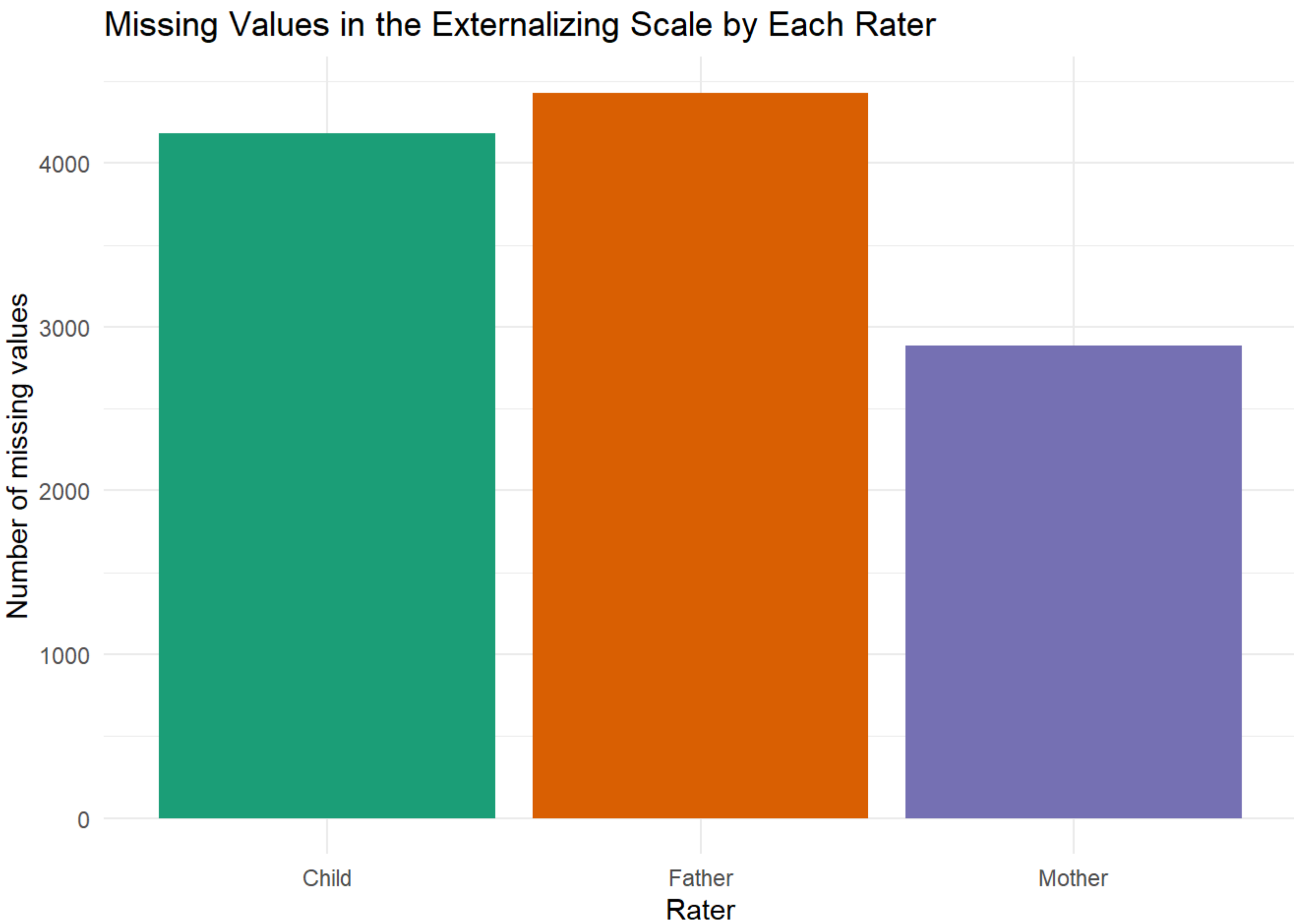
```
Warning: Removed 10685 rows containing non-finite outside the scale range
```

```
(`stat_boxplot()`).
```



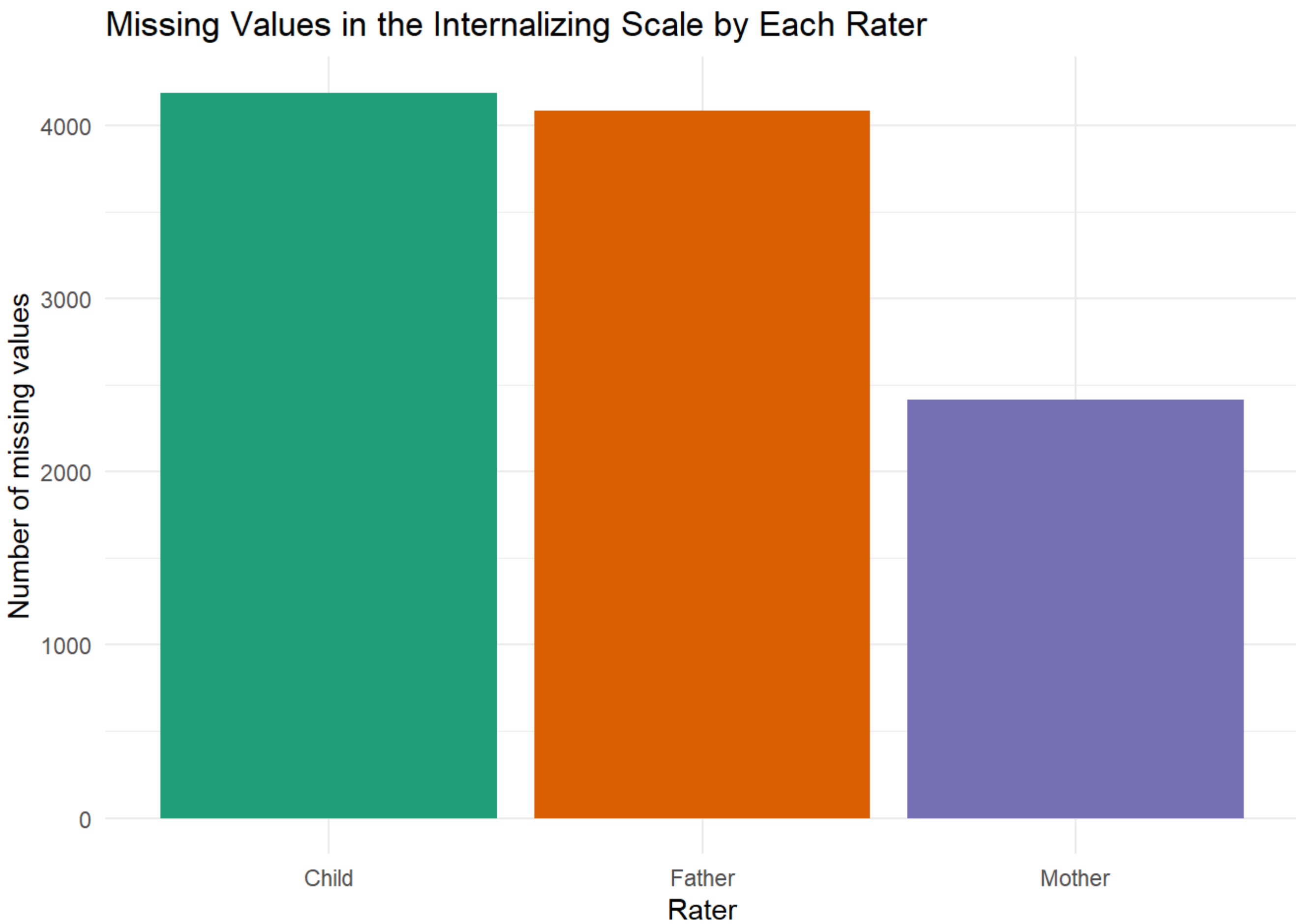
Looking at the number of missing values for each rater for externalizing scores:

```
data_df_long %>%
  select(ext_scale_f, ext_scale_m, ext_scale_c) %>%
  pivot_longer(cols = c(ext_scale_f, ext_scale_m, ext_scale_c), names_to = "rater", values_to =
"rating") %>%
  mutate(
    rater = case_when(
      rater == "ext_scale_f" ~ "Father",
      rater == "ext_scale_m" ~ "Mother",
      rater == "ext_scale_c" ~ "Child"
    )) %>%
  group_by(rater) %>%
  summarise(num_na = sum(is.na(rating))) %>%
  ggplot(aes(x = rater, y = num_na, fill = rater)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  scale_fill_brewer(palette = "Dark2") +
  theme_minimal() +
  labs(title = "Missing Values in the Externalizing Scale by Each Rater",
       x = "Rater",
       y = "Number of missing values")
```



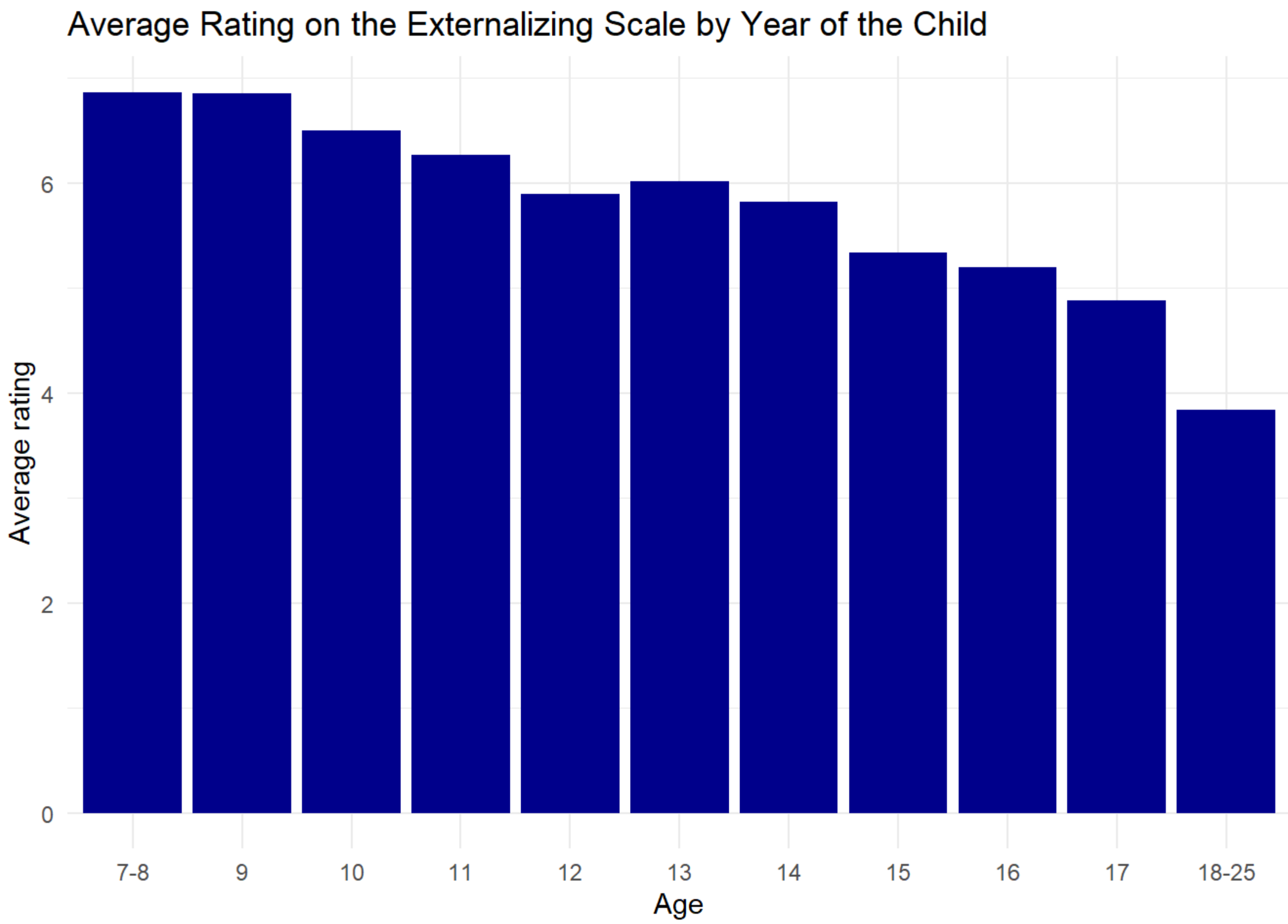
Looking at the number of missing values for each rater for internalizing scores:

```
data_df_long %>%
  select(int_scale_f, int_scale_m, int_scale_c) %>%
  pivot_longer(cols = c(int_scale_f, int_scale_m, int_scale_c), names_to = "rater", values_to =
"rating") %>%
  mutate(
    rater = case_when(
      rater == "int_scale_f" ~ "Father",
      rater == "int_scale_m" ~ "Mother",
      rater == "int_scale_c" ~ "Child"
    )) %>%
  group_by(rater) %>%
  summarise(num_na = sum(is.na(rating))) %>%
  ggplot(aes(x = rater, y = num_na, fill = rater)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  scale_fill_brewer(palette = "Dark2") +
  theme_minimal() +
  labs(title = "Missing Values in the Internalizing Scale by Each Rater",
       x = "Rater",
       y = "Number of missing values")
```



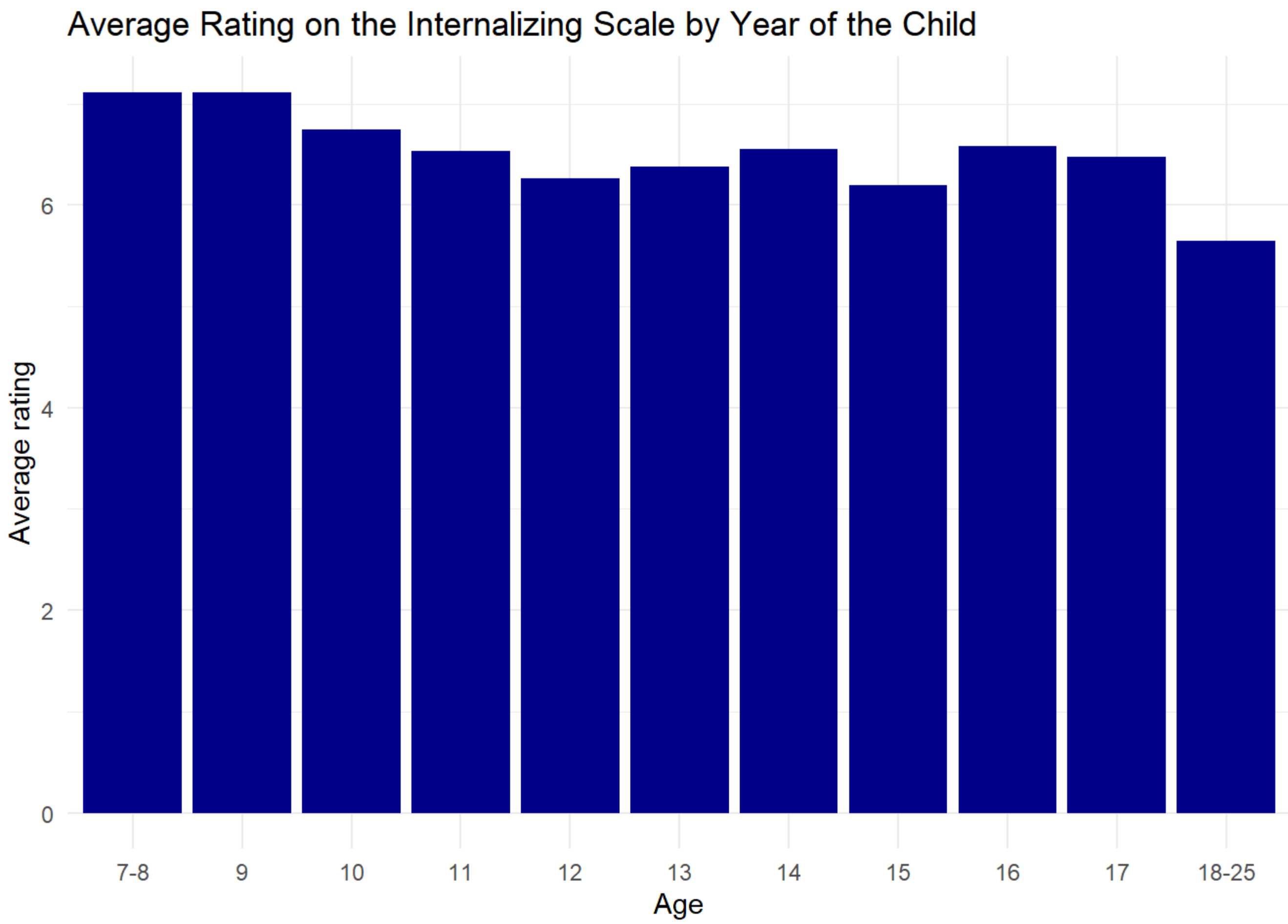
Average score for externalizing scale by age of the study child:

```
data_df_long %>%
  select(ext_scale_f, ext_scale_m, ext_scale_c, Age_cat) %>%
  pivot_longer(cols = c(ext_scale_f, ext_scale_m, ext_scale_c), names_to = "rater", values_to =
"rating") %>%
  filter(!is.na(Age_cat)) %>%
  group_by(Age_cat) %>%
  summarize(avg_rating = mean(rating, na.rm = TRUE)) %>%
  ggplot(aes(x = Age_cat, y = avg_rating)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  theme_minimal() +
  labs(title = "Average Rating on the Externalizing Scale by Year of the Child",
       x = "Age",
       y = "Average rating")
```



Average score for internalizing scale by age of the study child:

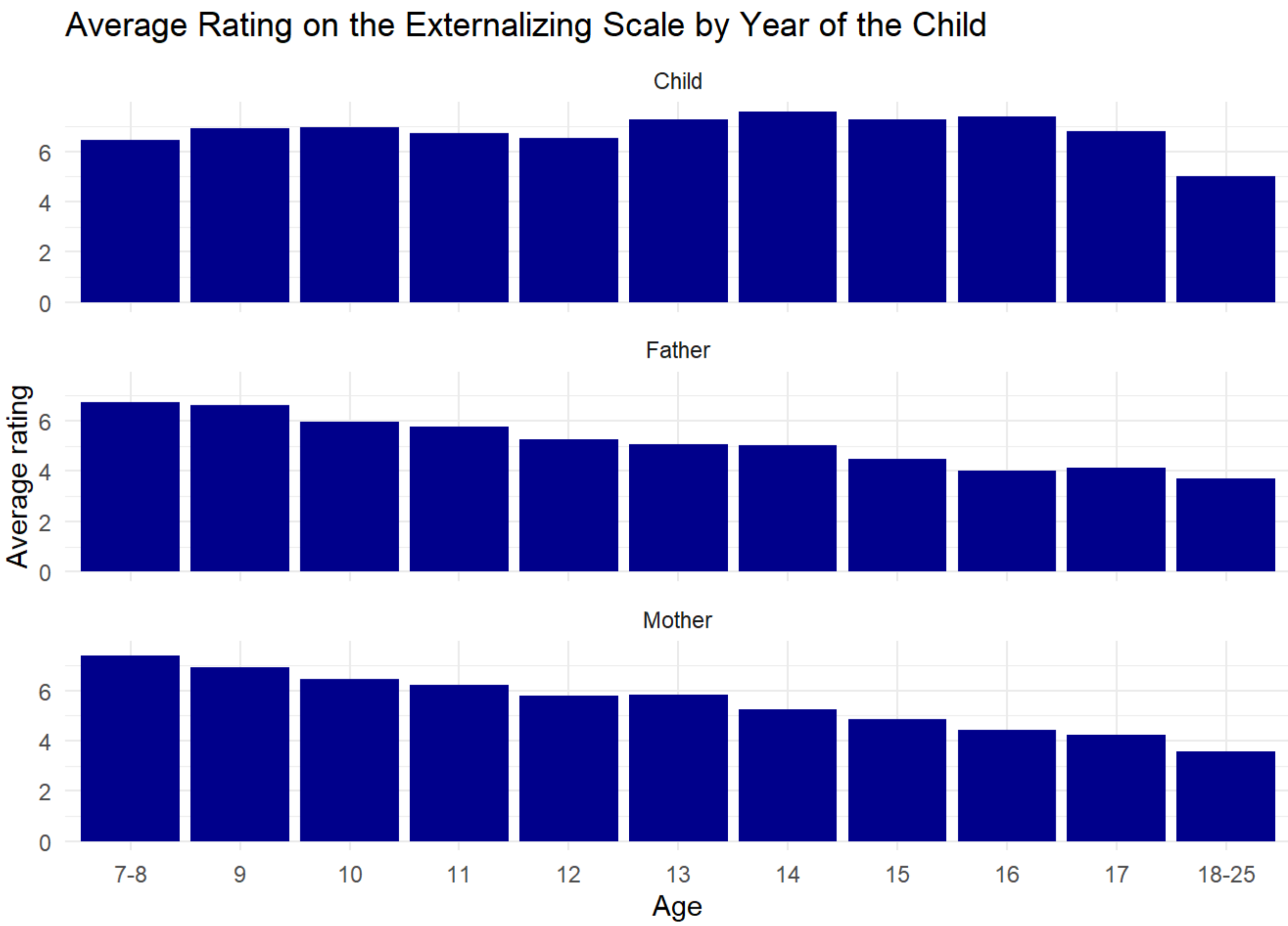
```
data_df_long %>%
  select(int_scale_f, int_scale_m, int_scale_c, Age_cat) %>%
  pivot_longer(cols = c(int_scale_f, int_scale_m, int_scale_c), names_to = "rater", values_to =
"rating") %>%
  filter(!is.na(Age_cat)) %>%
  group_by(Age_cat) %>%
  summarize(avg_rating = mean(rating, na.rm = TRUE)) %>%
  ggplot(aes(x = Age_cat, y = avg_rating)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  theme_minimal() +
  labs(title = "Average Rating on the Internalizing Scale by Year of the Child",
       x = "Age",
       y = "Average rating")
```



Average score for externalizing scale by age of the study child, faceted by rater:

```
data_df_long %>%
  select(ext_scale_f, ext_scale_m, ext_scale_c, Age_cat) %>%
  pivot_longer(cols = c(ext_scale_f, ext_scale_m, ext_scale_c), names_to = "rater", values_to =
"rating") %>%
  mutate(
    rater = case_when(
      rater == "ext_scale_f" ~ "Father",
      rater == "ext_scale_m" ~ "Mother",
      rater == "ext_scale_c" ~ "Child"
    )
  ) %>%
  filter(!is.na(Age_cat)) %>%
  group_by(Age_cat, rater) %>%
  summarize(avg_rating = mean(rating, na.rm = TRUE)) %>%
  ungroup() %>%
  ggplot(aes(x = Age_cat, y = avg_rating)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  facet_wrap(vars(rater), nrow = 3) +
  theme_minimal() +
  labs(title = "Average Rating on the Externalizing Scale by Year of the Child",
       x = "Age",
       y = "Average rating")
```

```
`summarise()` has grouped output by 'Age_cat'. You can override using the
`.groups` argument.
```

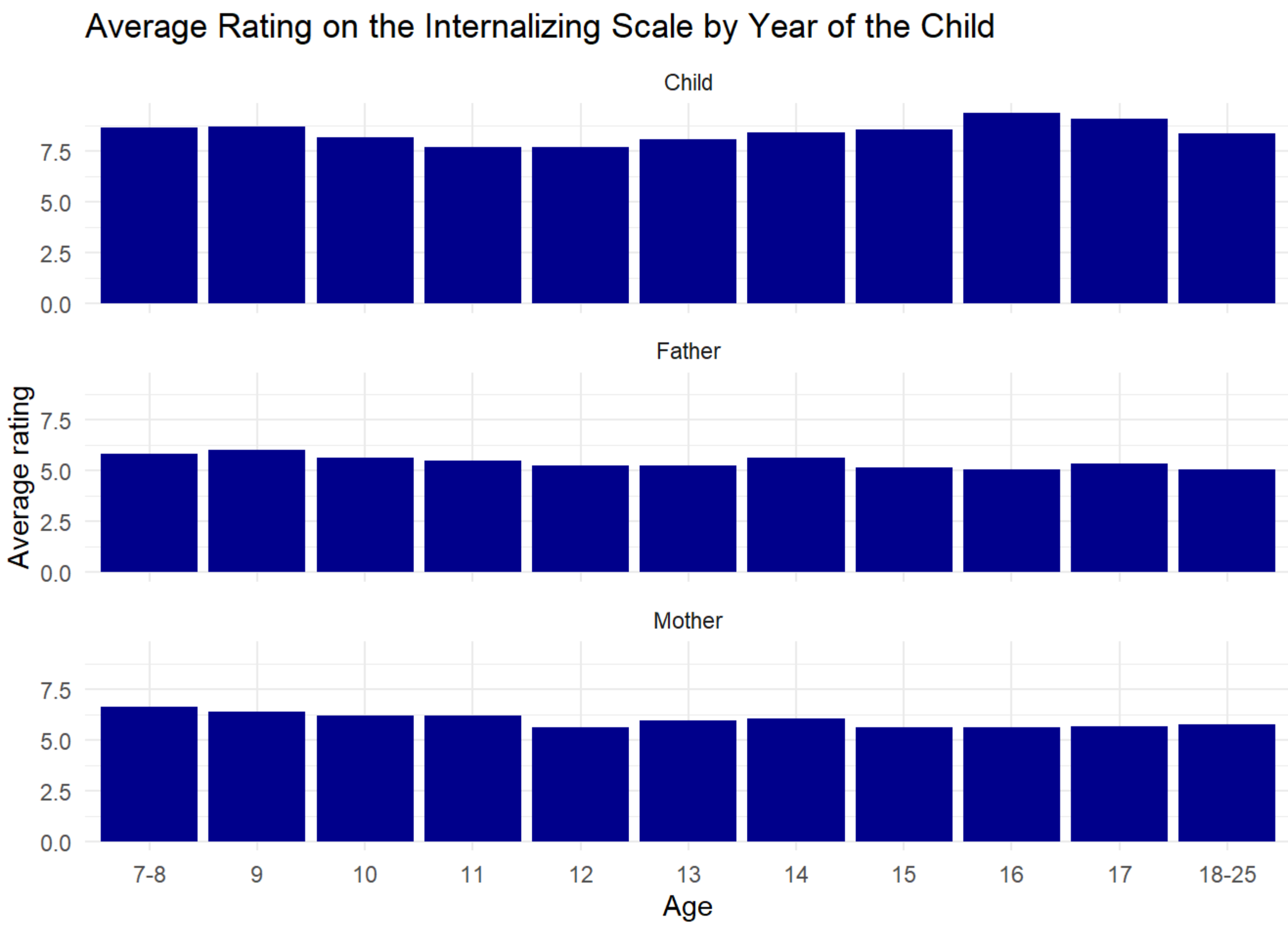


Average score for internalizing scale by age of the study child, faceted by rater:

```
data_df_long %>%
  select(int_scale_f, int_scale_m, int_scale_c, Age_cat) %>%
  pivot_longer(cols = c(int_scale_f, int_scale_m, int_scale_c), names_to = "rater", values_to =
"rating") %>%
  mutate(
    rater = case_when(
      rater == "int_scale_f" ~ "Father",
      rater == "int_scale_m" ~ "Mother",
      rater == "int_scale_c" ~ "Child"
    )
  ) %>%
  filter(!is.na(Age_cat)) %>%
  group_by(Age_cat, rater) %>%
  summarize(avg_rating = mean(rating, na.rm = TRUE)) %>%
  ungroup() %>%
  ggplot(aes(x = Age_cat, y = avg_rating)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  facet_wrap(vars(rater), nrow = 3) +
  theme_minimal() +
  labs(title = "Average Rating on the Internalizing Scale by Year of the Child",
```

```
       x = "Age",
       y = "Average rating")
```

```
`summarise()` has grouped output by 'Age_cat'. You can override using the
`.groups` argument.
```



4 - Creating variables for common items

In order to more easily compare the score by different raters on the common items, we will create a variable for each item that has the score for that item from each rater. For example, the item "threatens people" exist in the parent, child, and youth version of the survey. However, it is item number 50 for parents, 46 for child, and 45 for young adult. Instead, we will make a variable called e.g. ext_16 which has the score for this item by each rater.

This is especially going to make the switch from the child to young adult survey easier. We analyze the data based on age, while the switch was done based on wave (i.e. children were of different ages when switching to the young adult version). It is therefore easier to just have one variable per item instead of using if statements based on waves when doing the analyses based on age categories.

```
# Vectors with the child and youth items ordered so that the same item between the version are
indexed by the same index number (i.e. in order)
child_item_vect_ext = c("YSR1", "YSR2", "YSR5", "YSR6", "YSR8", "YSR17","YSR18", "YSR20",  "YSR27",
"YSR31", "YSR35", "YSR_max_37_38", "YSR40", "YSR43", "YSR44", "YSR45", "YSR46", "YSR50")
young_item_vect_ext = c("YASR1", "YASR3", "YASR7", "YASR8", "YASR9", "YASR19", "YASR21", "YASR23",
```

```
"YASR29", "YASR32", "YASR35", "YASR37", "YASR39", "YASR42", "YASR43", "YASR44", "YASR45", "YASR47")

# Creating a function to make a new variable for each item
create_com_var <- function(data_frame, child_item_vect, young_item_vect, item, symptom) {
  data_frame %>%
    mutate(!!paste0(symptom, "_cya_", item) := case_when(
      Wave < 9 ~ pull(data_frame, child_item_vect[item]),
      Wave > 9 ~ pull(data_frame, young_item_vect[item]))
    )
}

for (i in 1:length(child_item_vect_ext)){
  data_df_long <- create_com_var(data_df_long, child_item_vect_ext, young_item_vect_ext, i, "ext")
}
```

Creating the same variables, but for internalizing symptoms:

```
# Vectors with the child and youth items ordered so that the same item between the version are
indexed by the same index number (i.e. in order)
child_item_vect_int = c("YSR3", "YSR4", "YSR12", "YSR13", "YSR14", "YSR15", "YSR16", "YSR21",
"YSR22", "YSR24", "YSR29", "YSR32", "YSR33", "YSR36", "YSR49", "YSR52", "YSR53")
young_item_vect_int = c("YASR4", "YASR6", "YASR14", "YASR15", "YASR16", "YASR17", "YASR18",
"YASR24", "YASR27", "YASR28", "YASR30", "YASR33", "YASR34", "YASR36", "YASR46", "YASR48", "YASR49")

for (i in 1:length(child_item_vect_int)){
  data_df_long <- create_com_var(data_df_long, child_item_vect_int, young_item_vect_int, i, "int")
}
```

For ease of reading the output from the tri-factor model, I will also similarly create copies of the variables in the data frames that are named with the same symptom_rater_itemnum convention as when creating the variables for the child/young adult items:

```
father_item_vect_ext <- c("DCBC1", "DCBC2", "DCBC5", "DCBC6", "DCBC8", "DCBC18", "DCBC19",
"DCBC21", "DCBC28", "DCBC32", "DCBC36",  "DCBC_max_39_40", "DCBC42", "DCBC46", "DCBC47", "DCBC48",
"DCBC50", "DCBC54")
mother_item_vect_ext <- c("MCBC1", "MCBC2", "MCBC5", "MCBC6", "MCBC8", "MCBC18", "MCBC19",
"MCBC21", "MCBC28", "MCBC32", "MCBC36", "MCBC_max_39_40", "MCBC42", "MCBC46", "MCBC47", "MCBC48",
"MCBC50", "MCBC54")

for (i in 1:length(father_item_vect_ext)) {

  data_df_long[[paste0("ext_f_", i)]] <-    data_df_long[[father_item_vect_ext[i]]]

  data_df_long[[paste0("ext_m_", i)]] <- data_df_long[[mother_item_vect_ext[i]]]
}
```

For internalizing symptoms:

```
father_item_vect_int <- c("DCBC3", "DCBC4", "DCBC13", "DCBC14", "DCBC15", "DCBC16", "DCBC17",
"DCBC22", "DCBC23", "DCBC25", "DCBC30", "DCBC33", "DCBC34", "DCBC37", "DCBC53", "DCBC57", "DCBC58")
mother_item_vect_int <- c("MCBC3", "MCBC4", "MCBC13", "MCBC14", "MCBC15", "MCBC16", "MCBC17",
"MCBC22", "MCBC23", "MCBC25", "MCBC30", "MCBC33", "MCBC34", "MCBC37", "MCBC53", "MCBC57", "MCBC58")
```

```
for (i in 1:length(father_item_vect_int)) {

  data_df_long[[paste0("int_f_", i)]] <-    data_df_long[[father_item_vect_int[i]]]

  data_df_long[[paste0("int_m_", i)]] <- data_df_long[[mother_item_vect_int[i]]]
}
```

5 - Creating a Data Frame with Only One Observation per Child

For the analyzes that looks at just rater disagreement, regardless of the age of the child, we will need a data frame that has just one observation per child, to get rid of the nesting the wave creates. Therefore, we will create a data frame that randomly draws one observation per child that can be used in further analyzes.

Making the data_df_long_unique data frame from drawing one observation per child from the data_df_long data frame:

```
# Setting a seed for reproducibility
set.seed(1814)

# Creating a new data frame, data_df_long_unique from choosing one random observation per child
from data_df_long
data_df_long_unique <- data_df_long %>%
  # Filtering out any observation that has a missing value in one or more of the internalizing and
externalizing scales for any of the raters
  filter(!if_any(c(ext_scale_f, ext_scale_m, ext_scale_c, int_scale_f, int_scale_m,
int_scale_c),is.na)) %>%
  # Group by ID to get one observation per child
  group_by(ID) %>%
  # Chose one random observation based on the grouping variable
  slice_sample() %>%
  # Ungroup the grouping by ID
  ungroup()
```

6 - Difference scores

We will calculate difference scores between each combination of rater - father, mother, and child. First, we will do it for rater disagreement in general, using the data_df_long_unique data frame, then we will do is separately for each age. In addition, we will do all of these operations separately for externalizing and internalizing symptoms, since our analyzes are separated by the type of symptom

6.1 - In general

Calculating difference scores for each combination of rater, regardless of age of the child:

```
data_df_long_unique <- data_df_long_unique %>%
  mutate(
    diff_sc_ext_fm = ext_scale_f - ext_scale_m,
    diff_sc_ext_fc = ext_scale_f - ext_scale_c,
    diff_sc_ext_mc = ext_scale_m - ext_scale_c,
    diff_sc_int_fm = int_scale_f - int_scale_m,
```

```
    diff_sc_int_fc = int_scale_f - int_scale_c,
    diff_sc_int_mc = int_scale_m - int_scale_c,
  )
```

Finding the mean and sd of the difference scores between the raters:

```
diff_scores_stats_df <- data_df_long_unique %>%
  summarize(across(starts_with("diff_sc_"), list(mean = mean, sd = sd), na.rm = TRUE))
```

```
Warning: There was 1 warning in `summarize()`.
ℹ In argument: `across(starts_with("diff_sc_"), list(mean = mean, sd = sd),
  na.rm = TRUE)`.
Caused by warning:
! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
Supply arguments directly to `.fns` through an anonymous function instead.

  # Previously
  across(a:b, mean, na.rm = TRUE)

  # Now
  across(a:b, \(x) mean(x, na.rm = TRUE))
```

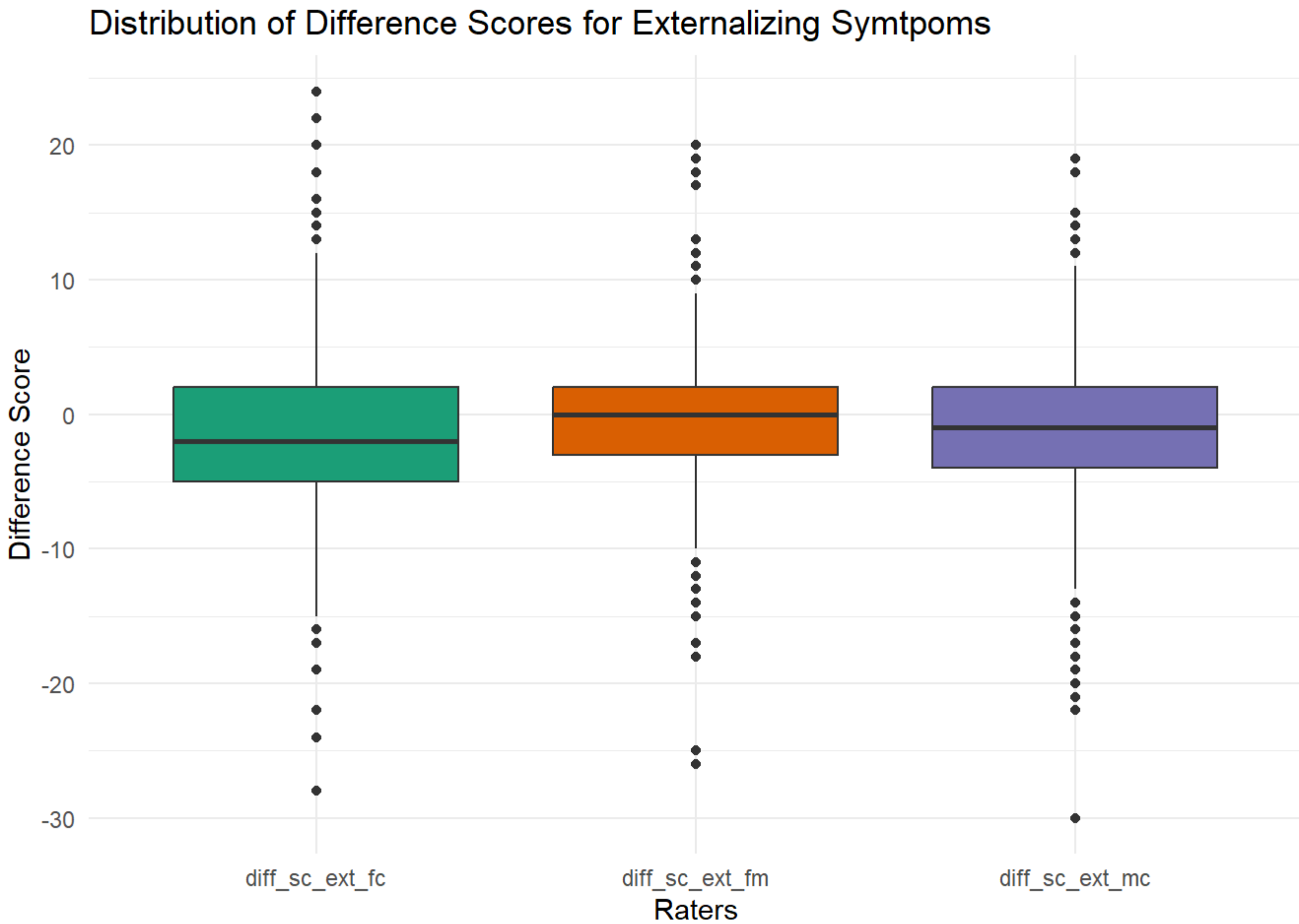
```
print(diff_scores_stats_df)
```

```
# A tibble: 1 × 12
  diff_sc_ext_fm_mean diff_sc_ext_fm_sd diff_sc_ext_fc_mean diff_sc_ext_fc_sd
                <dbl>             <dbl>               <dbl>             <dbl>
1              -0.443              4.41               -1.49              5.54
# ℹ 8 more variables: diff_sc_ext_mc_mean <dbl>, diff_sc_ext_mc_sd <dbl>,
#   diff_sc_int_fm_mean <dbl>, diff_sc_int_fm_sd <dbl>,
#   diff_sc_int_fc_mean <dbl>, diff_sc_int_fc_sd <dbl>,
#   diff_sc_int_mc_mean <dbl>, diff_sc_int_mc_sd <dbl>
```

Plotting the diff scores for externalizing symptoms:

```
data_df_long_unique %>%
  select(diff_sc_ext_fm, diff_sc_ext_fc, diff_sc_ext_mc) %>%
  pivot_longer(cols = c(diff_sc_ext_fm, diff_sc_ext_fc, diff_sc_ext_mc), names_to = "raters",
values_to = "difference") %>%
  mutate(
    rater = case_when(
      raters == "diff_sc_ext_fm" ~ "Father - Mother",
      raters == "diff_sc_ext_fc" ~ "Father - Child",
      raters == "diff_sc_ext_mc" ~ "Mother - Child"
    )
  ) %>%
  ggplot(aes(x = raters, y = difference, fill = raters)) +
  geom_boxplot(show.legend = FALSE) +
  scale_fill_brewer(palette = "Dark2") +
  theme_minimal() +
  labs(
```

```
    title = "Distribution of Difference Scores for Externalizing Symtpoms",
    x = "Raters",
    y = "Difference Score"
  )
```

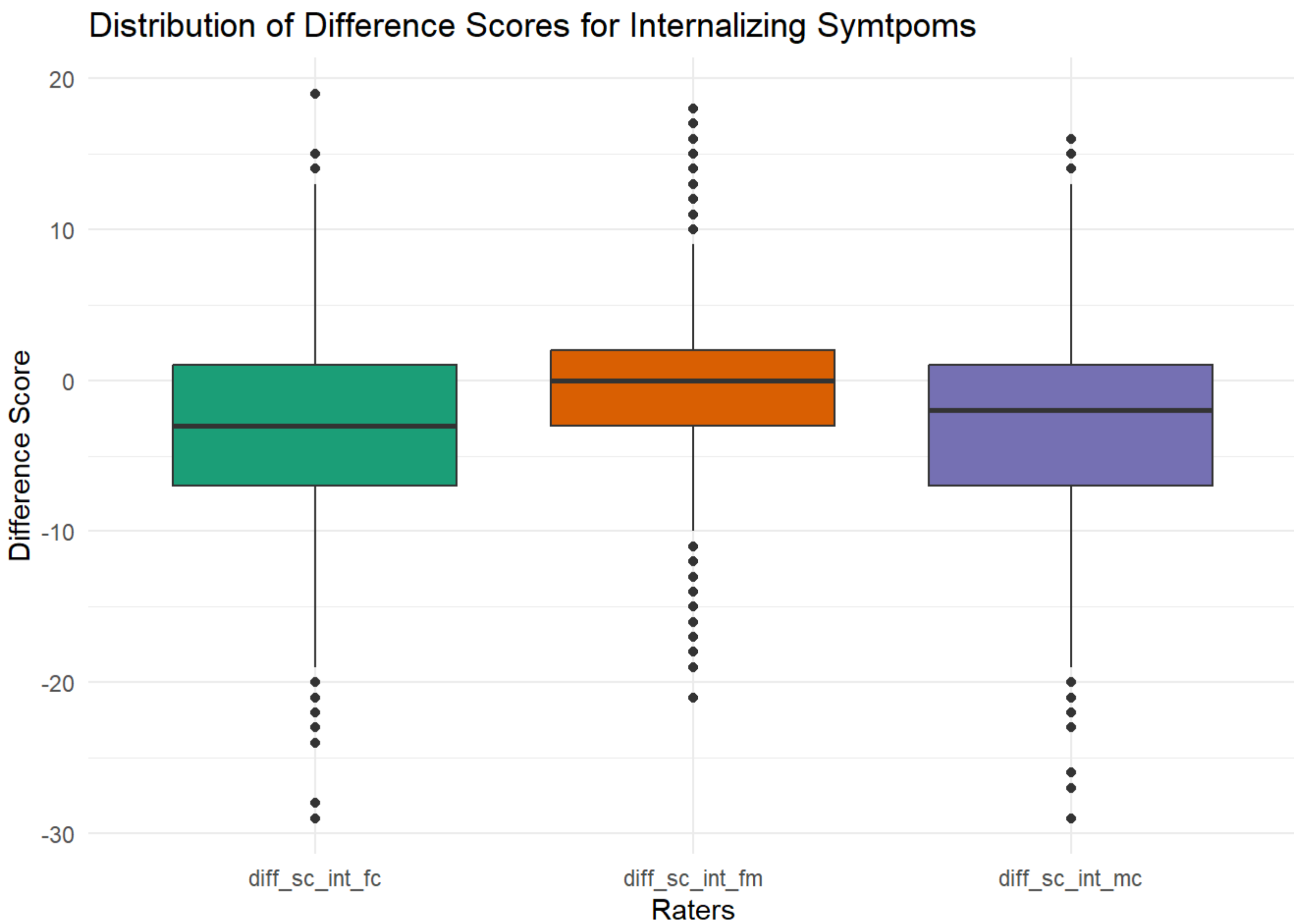


You can see that in general, fathers and mothers tend to have fairly similar ratings, since the mean is close to 0. Otherwise, the children tends to score themselves higher on externalizing symptoms than mothers or fathers.

Plotting the diff scores for internalizing symptoms:

```
data_df_long_unique %>%
  select(diff_sc_int_fm, diff_sc_int_fc, diff_sc_int_mc) %>%
  pivot_longer(cols = c(diff_sc_int_fm, diff_sc_int_fc, diff_sc_int_mc), names_to = "raters",
values_to = "difference") %>%
  mutate(
    rater = case_when(
      raters == "diff_sc_int_fm" ~ "Father - Mother",
      raters == "diff_sc_int_fc" ~ "Father - Child",
      raters == "diff_sc_int_mc" ~ "Mother - Child"
    )
  ) %>%
  ggplot(aes(x = raters, y = difference, fill = raters)) +
  geom_boxplot(show.legend = FALSE) +
  scale_fill_brewer(palette = "Dark2") +
  theme_minimal() +
```

```
  labs(
    title = "Distribution of Difference Scores for Internalizing Symtpoms",
    x = "Raters",
    y = "Difference Score"
  )
```

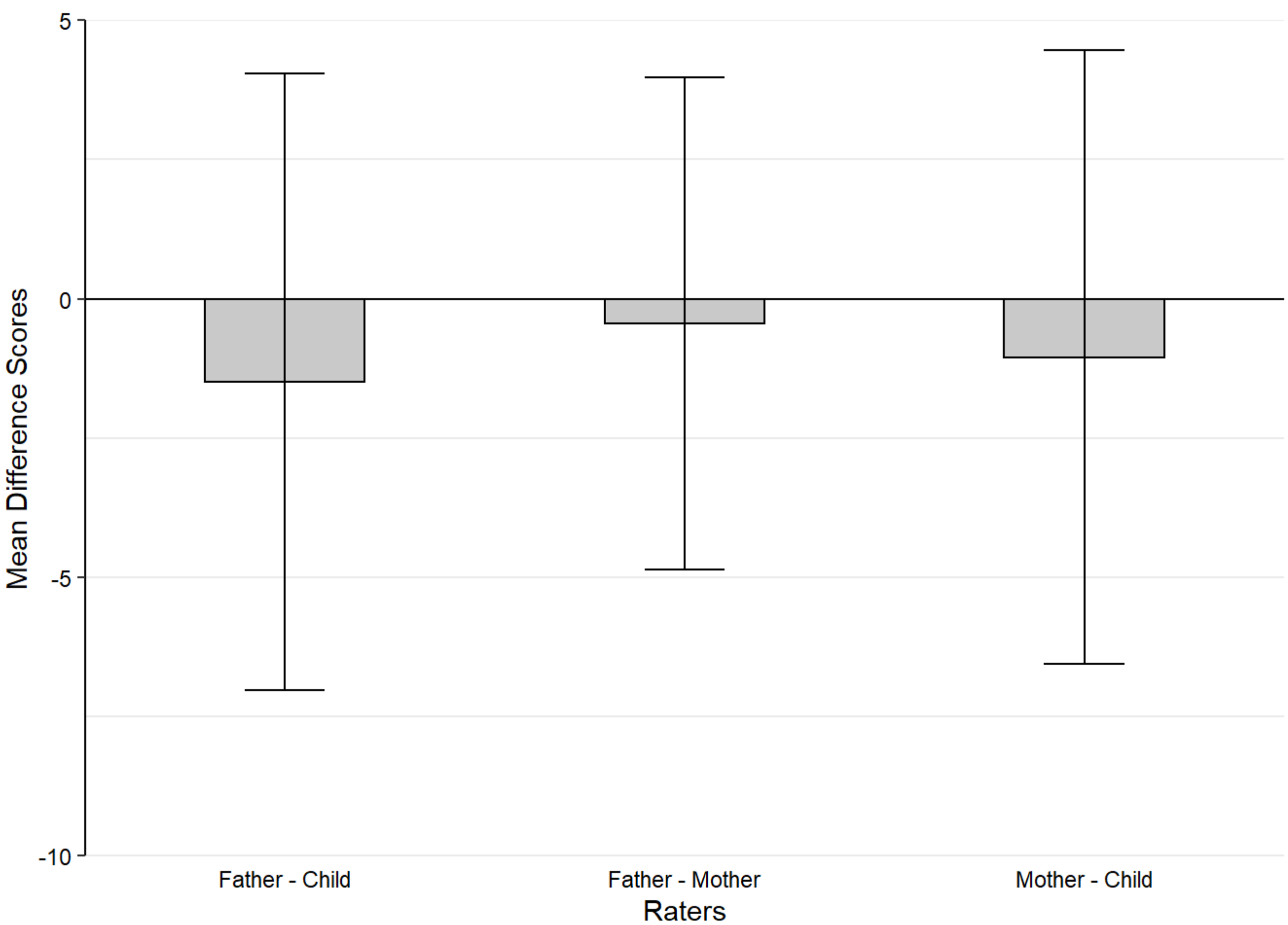


You can see that in general, fathers and mothers tend to have fairly similar ratings, since the mean is close to 0. Otherwise, the children tends to score themselves higher on internalizing symptoms than mothers or fathers. This pattern was very similar to that of externalizing symptoms.

Creating a bar graph for the results section in the thesis for externalizing symptoms:

```
diff_scores_stats_df %>%
  select(diff_sc_ext_fm_mean, diff_sc_ext_fm_sd, diff_sc_ext_fc_mean, diff_sc_ext_fc_sd,
diff_sc_ext_mc_mean, diff_sc_ext_mc_sd) %>%
  pivot_longer(cols = c(diff_sc_ext_fm_mean, diff_sc_ext_mc_mean, diff_sc_ext_fc_mean,
diff_sc_ext_fm_sd, diff_sc_ext_mc_sd, diff_sc_ext_fc_sd), names_to = c("raters", ".value"),
names_pattern = "[[:alpha:]]+_sc_ext_([[:alpha:]]+)_([[:alpha:]]+)") %>%
  mutate(
    raters = case_when(
      raters == "fm" ~ "Father - Mother",
      raters == "fc" ~ "Father - Child",
      raters == "mc" ~ "Mother - Child"
    )
  ) %>%
```

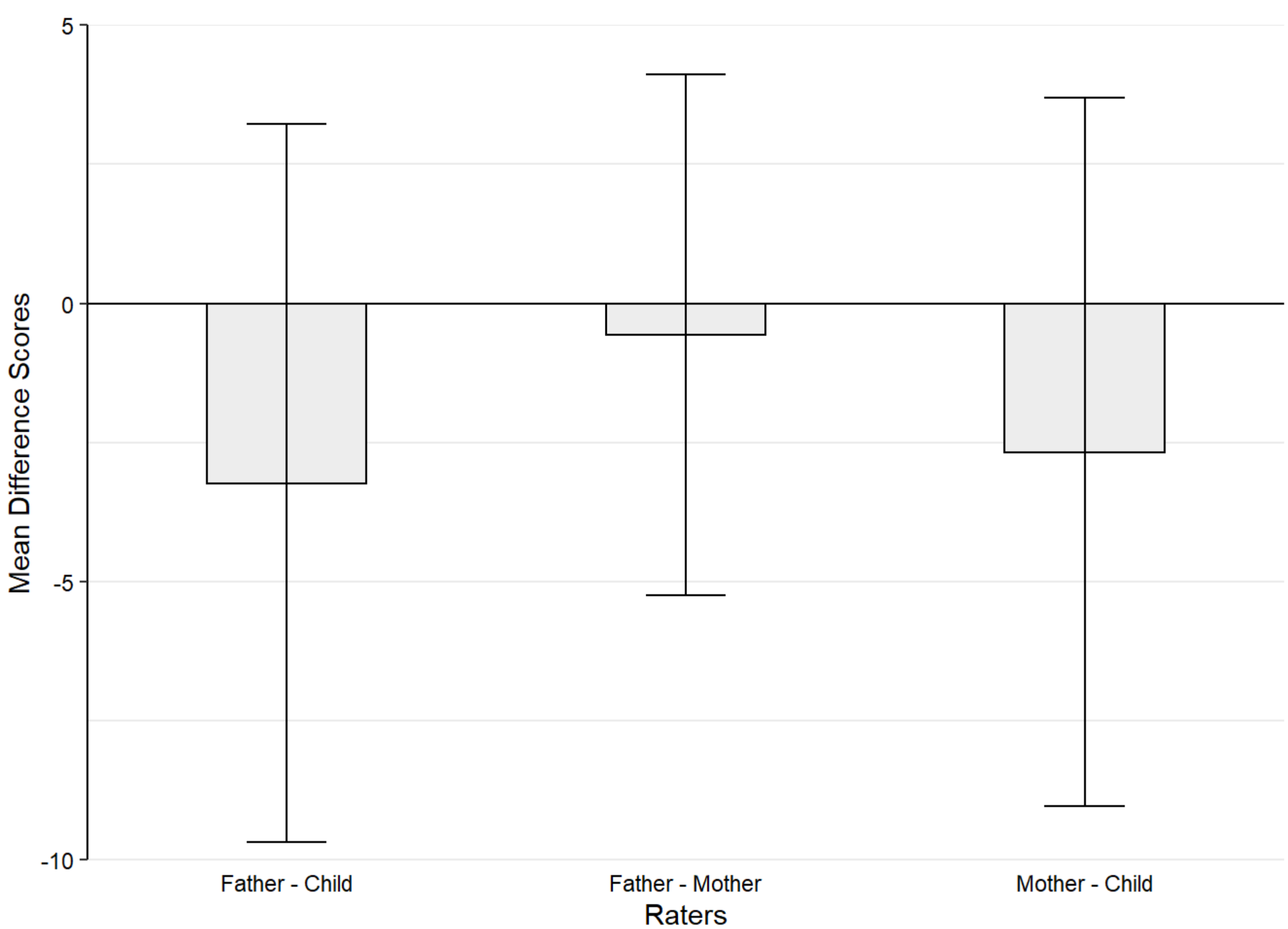
```
  ggplot(aes(x = raters, y = mean)) +
  geom_bar(stat = "identity", color = "black", fill = "grey79", width = 0.4) +
  geom_errorbar(aes(ymin = mean - sd, ymax = mean + sd), width = 0.20) +
  ylim(-10, 5) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line()) +
  coord_cartesian(expand = FALSE, xlim = c(0.5, 3.5), ylim = c(-10, 5)) +
  geom_hline(aes(yintercept = 0)) +
  labs(
    x = "Raters",
    y = "Mean Difference Scores"
  )
```



Creating a bar graph for the results section in the thesis for internalizing symptoms:

```
diff_scores_stats_df %>%
  select(diff_sc_int_fm_mean, diff_sc_int_fm_sd, diff_sc_int_fc_mean, diff_sc_int_fc_sd,
diff_sc_int_mc_mean, diff_sc_int_mc_sd) %>%
  pivot_longer(cols = c(diff_sc_int_fm_mean, diff_sc_int_mc_mean, diff_sc_int_fc_mean,
diff_sc_int_fm_sd, diff_sc_int_mc_sd, diff_sc_int_fc_sd), names_to = c("raters", ".value"),
```

```
names_pattern = "[[:alpha:]]+_sc_int_([[:alpha:]]+)_([[:alpha:]]+)") %>%
  mutate(
    raters = case_when(
      raters == "fm" ~ "Father - Mother",
      raters == "fc" ~ "Father - Child",
      raters == "mc" ~ "Mother - Child"
    )
  ) %>%
  ggplot(aes(x = raters, y = mean)) +
  geom_bar(stat = "identity", color = "black", fill = "grey93", width = 0.4) +
  geom_errorbar(aes(ymin = mean - sd, ymax = mean + sd), width = 0.20) +
  ylim(-10, 5) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line()) +
  coord_cartesian(expand = FALSE, xlim = c(0.5, 3.5), ylim = c(-10, 5)) +
  geom_hline(aes(yintercept = 0)) +
  labs(
    x = "Raters",
    y = "Mean Difference Scores"
  )
```



6.2 - At each Age

Calculating difference scores for each combination of rater for each age of the child:

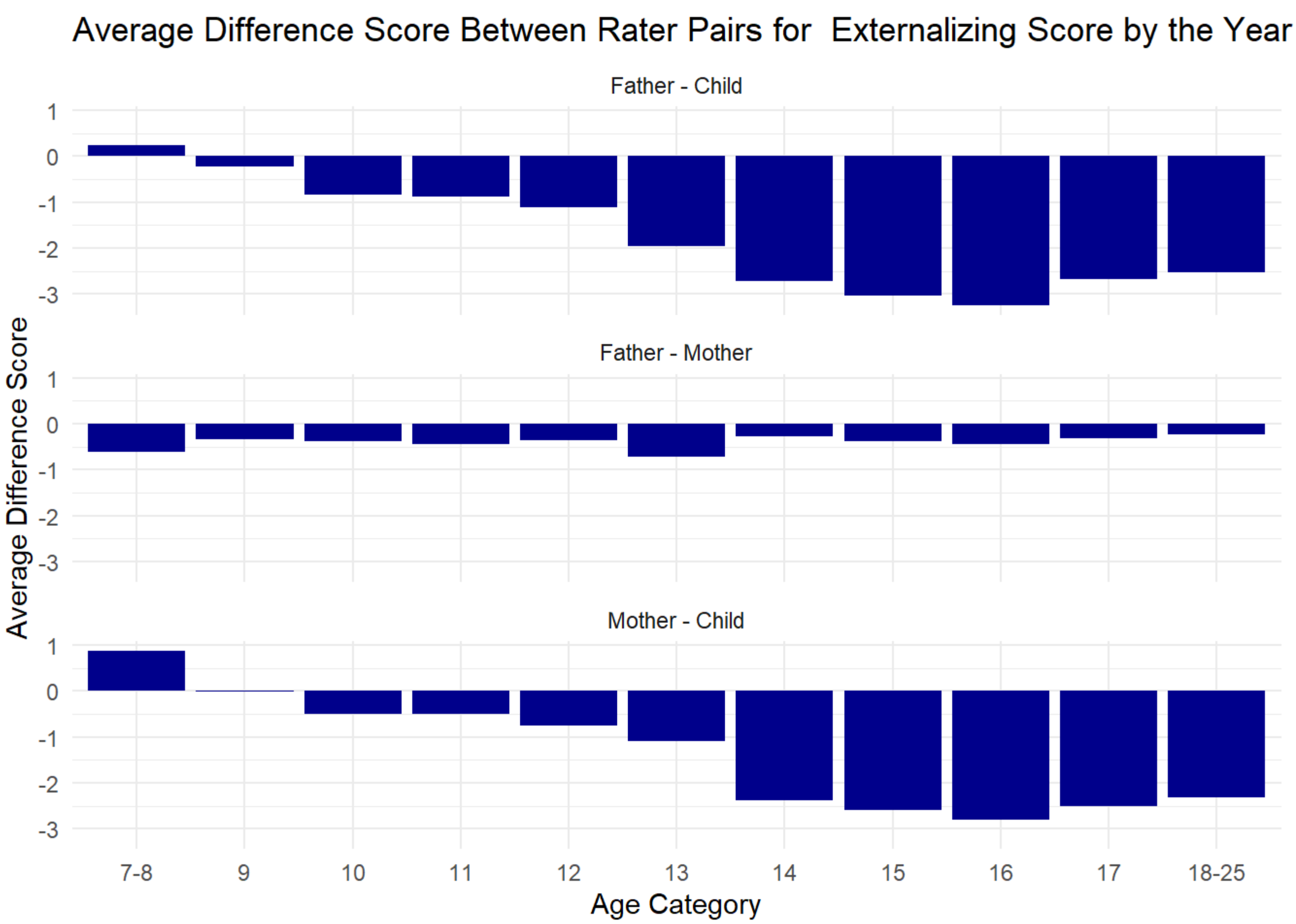
```
data_df_long <- data_df_long %>%
  mutate(
    diff_sc_ext_fm = ext_scale_f - ext_scale_m,
    diff_sc_ext_fc = ext_scale_f - ext_scale_c,
    diff_sc_ext_mc = ext_scale_m - ext_scale_c,
    diff_sc_int_fm = int_scale_f - int_scale_m,
    diff_sc_int_fc = int_scale_f - int_scale_c,
    diff_sc_int_mc = int_scale_m - int_scale_c,
  )

# Creating a data frame with the average difference scores
avg_diff_score_df <- data_df_long %>%
  filter(!is.na(Age_cat)) %>%
  group_by(Age_cat) %>%
  summarize(across(starts_with("diff_sc_"), list(mean = mean, sd = sd), na.rm = TRUE)) %>%
  ungroup()

print(avg_diff_score_df)
```

```
# A tibble: 11 × 13
   Age_cat diff_sc_ext_fm_mean diff_sc_ext_fm_sd diff_sc_ext_fc_mean
   <fct>                 <dbl>             <dbl>               <dbl>
 1 7-8                  -0.625              4.47               0.233
 2 9                    -0.342              4.90              -0.219
 3 10                   -0.39               4.60              -0.825
 4 11                   -0.451              4.82              -0.884
 5 12                   -0.352              4.51              -1.12
 6 13                   -0.713              4.20              -1.95
 7 14                   -0.287              4.56              -2.71
 8 15                   -0.379              4.00              -3.03
 9 16                   -0.436              4.05              -3.24
10 17                   -0.309              4.03              -2.67
11 18-25                -0.243              3.61              -2.52
# ℹ 9 more variables: diff_sc_ext_fc_sd <dbl>, diff_sc_ext_mc_mean <dbl>,
#   diff_sc_ext_mc_sd <dbl>, diff_sc_int_fm_mean <dbl>,
#   diff_sc_int_fm_sd <dbl>, diff_sc_int_fc_mean <dbl>,
#   diff_sc_int_fc_sd <dbl>, diff_sc_int_mc_mean <dbl>, diff_sc_int_mc_sd <dbl>
```

Plotting the average difference scores for externalizing symptoms by age:

```
avg_diff_score_df %>%
  pivot_longer(cols = c(diff_sc_ext_fm_mean, diff_sc_ext_fc_mean, diff_sc_ext_mc_mean), names_to =
"raters", values_to = "diff_score") %>%
  mutate(
    raters = case_when(
      raters == "diff_sc_ext_fm_mean" ~ "Father - Mother",
      raters == "diff_sc_ext_fc_mean" ~ "Father - Child",
```

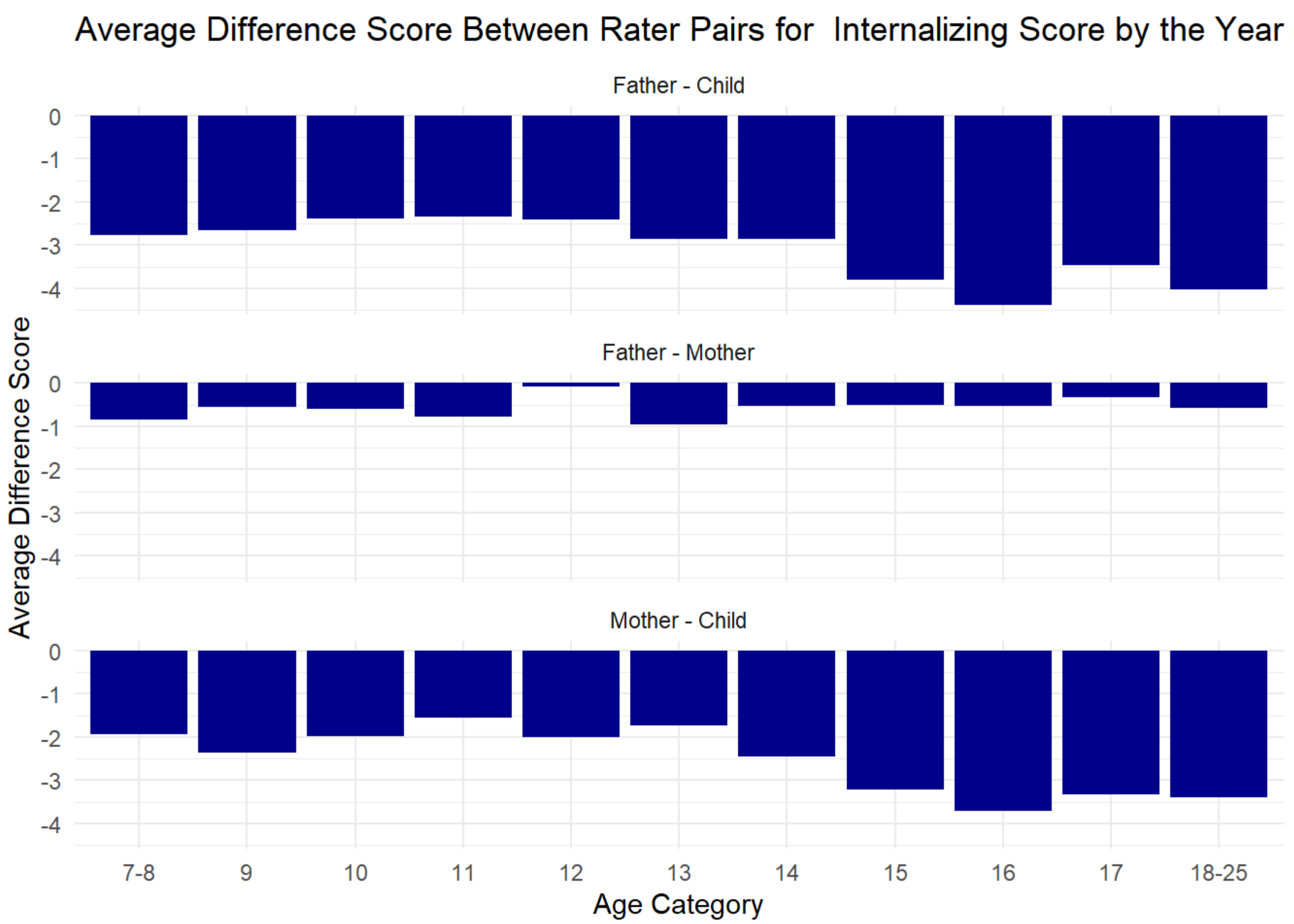
```
      raters == "diff_sc_ext_mc_mean" ~ "Mother - Child"
    )
  ) %>%
  ggplot(aes(x = Age_cat, y = diff_score)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  facet_wrap(vars(raters), nrow = 3) +
  theme_minimal() +
  labs(title = "Average Difference Score Between Rater Pairs for  Externalizing Score by the Year
of the Child",
       x = "Age Category",
       y = "Average Difference Score")
```



Plotting the average difference scores for internalizing symptoms by age:

```
avg_diff_score_df %>%
  pivot_longer(cols = c(diff_sc_int_fm_mean, diff_sc_int_fc_mean, diff_sc_int_mc_mean), names_to =
"raters", values_to = "diff_score") %>%
  mutate(
    raters = case_when(
      raters == "diff_sc_int_fm_mean" ~ "Father - Mother",
      raters == "diff_sc_int_fc_mean" ~ "Father - Child",
      raters == "diff_sc_int_mc_mean" ~ "Mother - Child"
    )
  ) %>%
  ggplot(aes(x = Age_cat, y = diff_score)) +
```

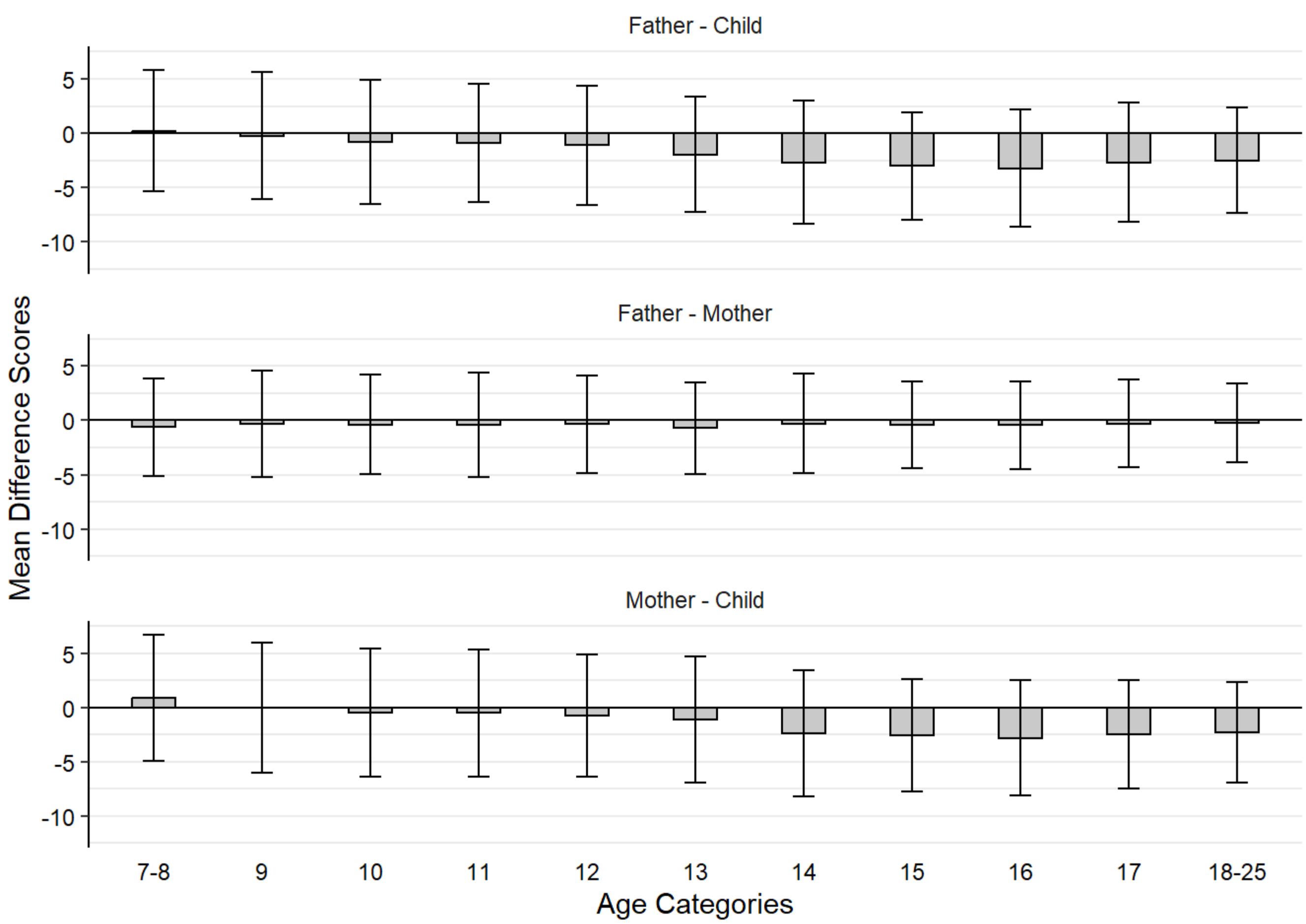
```
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  facet_wrap(vars(raters), nrow = 3) +
  theme_minimal() +
  labs(title = "Average Difference Score Between Rater Pairs for  Internalizing Score by the Year
of the Child",
       x = "Age Category",
       y = "Average Difference Score")
```



Creating a bar graph for the results section in the thesis for externalizing symptoms at each age category:

```
avg_diff_score_df %>%
   select(diff_sc_ext_fm_mean, diff_sc_ext_fm_sd, diff_sc_ext_fc_mean, diff_sc_ext_fc_sd,
diff_sc_ext_mc_mean, diff_sc_ext_mc_sd, Age_cat) %>%
  pivot_longer(cols = -Age_cat, names_to = c("raters", ".value"), names_pattern = "
[[:alpha:]]+_sc_ext_([[:alpha:]]+)_([[:alpha:]]+)") %>%
  mutate(
    raters = case_when(
      raters == "fm" ~ "Father - Mother",
      raters == "fc" ~ "Father - Child",
      raters == "mc" ~ "Mother - Child"
    )
  ) %>%
  ggplot(aes(x = Age_cat, y = mean)) +
  geom_bar(stat = "identity", color = "black", fill = "grey79", width = 0.4) +
  geom_errorbar(aes(ymin = mean - sd, ymax = mean + sd), width = 0.20) +
```

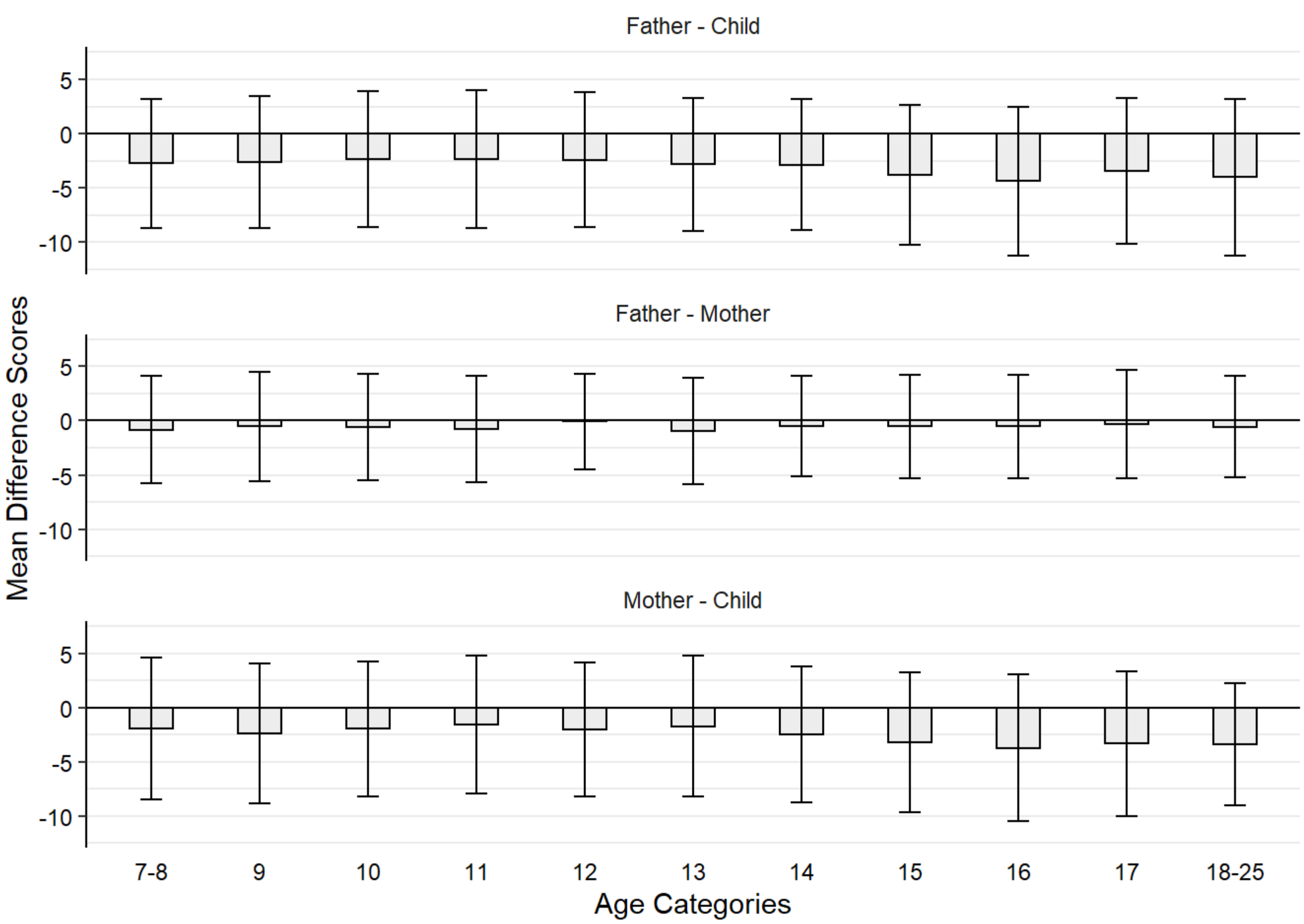
```
  facet_wrap(vars(raters), nrow = 3) +
  ylim(-12, 7) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  geom_hline(aes(yintercept = 0)) +
  labs(
    x = "Age Categories",
    y = "Mean Difference Scores"
  )
```



Internalizing scores:

```
avg_diff_score_df %>%
   select(diff_sc_int_fm_mean, diff_sc_int_fm_sd, diff_sc_int_fc_mean, diff_sc_int_fc_sd,
diff_sc_int_mc_mean, diff_sc_int_mc_sd, Age_cat) %>%
  pivot_longer(cols = -Age_cat, names_to = c("raters", ".value"), names_pattern = "
[[:alpha:]]+_sc_int_([[:alpha:]]+)_([[:alpha:]]+)") %>%
  mutate(
    raters = case_when(
```

```
    raters == "fm" ~ "Father - Mother",
    raters == "fc" ~ "Father - Child",
    raters == "mc" ~ "Mother - Child"
  )
) %>%
ggplot(aes(x = Age_cat, y = mean)) +
geom_bar(stat = "identity", color = "black", fill = "grey93", width = 0.4) +
geom_errorbar(aes(ymin = mean - sd, ymax = mean + sd), width = 0.20) +
facet_wrap(vars(raters), nrow = 3) +
ylim(-12, 7) +
theme_classic() +
theme(axis.text.x = element_text(colour = "black"),
      axis.text.y = element_text(colour = "black"),
      axis.line.x = element_blank(),
      axis.ticks.x = element_blank(),
      panel.grid.major.y = element_line(),
      panel.grid.minor.y = element_line(),
      strip.background = element_blank()) +
geom_hline(aes(yintercept = 0)) +
labs(
  x = "Age Categories",
  y = "Mean Difference Scores"
)
```



6.3 - Adult Attachment

This is the easiest to do if the data set is in a wide format. There is a data frame called data_df_wide that is in wide format. However, this data frame doesn't have the difference scores between each combination of two raters calculated. Therefore, we will convert from the current data_df_long to data_df_wide_diff:

```
data_df_wide_diff <- data_df_long %>%
  pivot_wider(id_cols = c(ID,  IDN, m1educ_m, f1educ_f, ChGender), names_from = Wave, values_from =
setdiff(names(data_df_long), c("ID", "IDN", "m1educ_m", "f1educ_f", "ChGender", "Wave")))

# Changing the column names to have the same format as they originally had, where yx where x is a
number designates the wave number. The following code finds column names that have some symbols (in
a capture group) and ends with "_x" where x is a number in a capture group. It then adds the prefix
"y", then the number in the second capture group, then the original column name. The "_" is deleted
from the column name.
colnames(data_df_wide_diff) <- gsub("^(.*)_(\\d+)$", "y\\2\\1", colnames(data_df_wide_diff))
```

Creating a data frame with the average difference scores for father:

```
avg_diff_score_aaa_f_df <- data_df_wide_diff %>%
  filter(!is.na(y10aaa_scale_ya_f_dich)) %>%
  group_by(y10aaa_scale_ya_f_dich) %>%
  summarize(across(c(y8diff_sc_ext_fc, y8diff_sc_int_fc), list(mean = mean, sd = sd), na.rm =
TRUE)) %>%
  ungroup() %>%
  mutate(raters = "Father - Child") %>%
  pivot_longer(cols = starts_with("y8diff"), names_to = c("Type", ".value"), names_pattern =
"^y8diff_sc_([[:alpha:]]+)_[[:alpha:]]+_([[:alpha:]]+)$") %>% rename(Attachment =
y10aaa_scale_ya_f_dich, Mean = mean)

print(avg_diff_score_aaa_f_df)
```

```
# A tibble: 4 × 5
  Attachment raters         Type   Mean    sd
  <chr>      <chr>          <chr> <dbl> <dbl>
1 High       Father - Child ext   -2.52  5.08
2 High       Father - Child int   -4.80  7.25
3 Low        Father - Child ext   -3.13  4.43
4 Low        Father - Child int   -3.53  5.76
```

Creating a data frame with the average difference scores for mother:

```
avg_diff_score_aaa_m_df <- data_df_wide_diff %>%
  filter(!is.na(y10aaa_scale_ya_m_dich)) %>%
  group_by(y10aaa_scale_ya_m_dich) %>%
  summarize(across(c(y8diff_sc_ext_mc, y8diff_sc_int_mc), list(mean = mean, sd = sd), na.rm =
TRUE)) %>%
  ungroup() %>%
  mutate(raters = "Mother - Child") %>%
  pivot_longer(cols = starts_with("y8diff"), names_to = c("Type", ".value"), names_pattern =
"^y8diff_sc_([[:alpha:]]+)_[[:alpha:]]+_([[:alpha:]]+)$") %>%
  rename(Attachment = y10aaa_scale_ya_m_dich, Mean = mean)
```

```
print(avg_diff_score_aaa_m_df)
```

```
# A tibble: 4 × 5
  Attachment raters         Type   Mean    sd
  <chr>      <chr>          <chr> <dbl> <dbl>
1 High       Mother - Child ext   -2.73  4.46
2 High       Mother - Child int   -4.15  6.35
3 Low        Mother - Child ext   -2.93  4.65
4 Low        Mother - Child int   -2.74  6.19
```

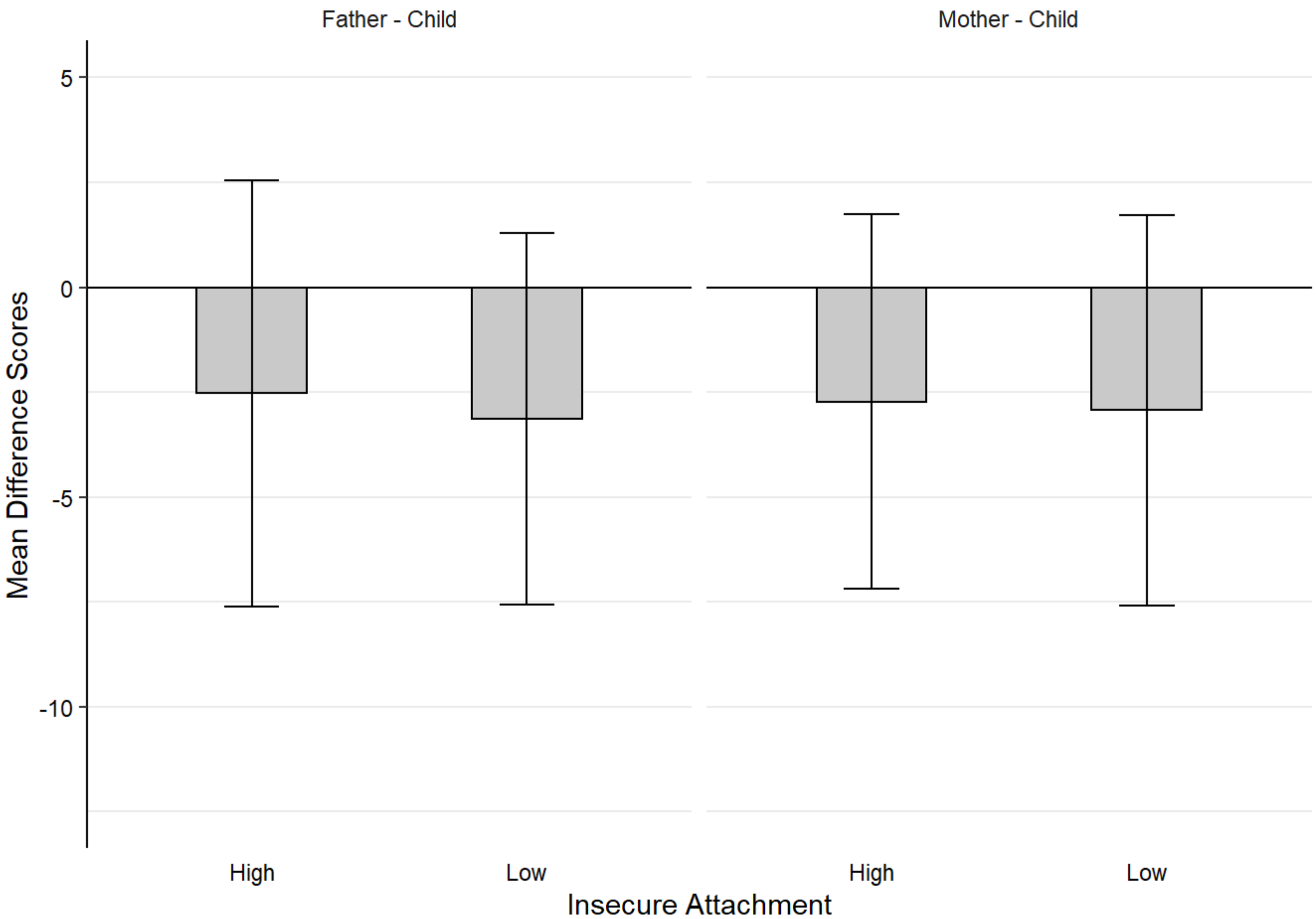
Combining the data sets for mother and father difference scores. They were calculated separately, so that the statistics could be calculated grouping by high or low attachment for father and mother separately

```
avg_diff_score_aaa_df <- avg_diff_score_aaa_f_df %>%
  rbind(avg_diff_score_aaa_m_df)

avg_diff_score_aaa_df
```

```
# A tibble: 8 × 5
  Attachment raters         Type   Mean    sd
  <chr>      <chr>          <chr> <dbl> <dbl>
1 High       Father - Child ext   -2.52  5.08
2 High       Father - Child int   -4.80  7.25
3 Low        Father - Child ext   -3.13  4.43
4 Low        Father - Child int   -3.53  5.76
5 High       Mother - Child ext   -2.73  4.46
6 High       Mother - Child int   -4.15  6.35
7 Low        Mother - Child ext   -2.93  4.65
8 Low        Mother - Child int   -2.74  6.19
```

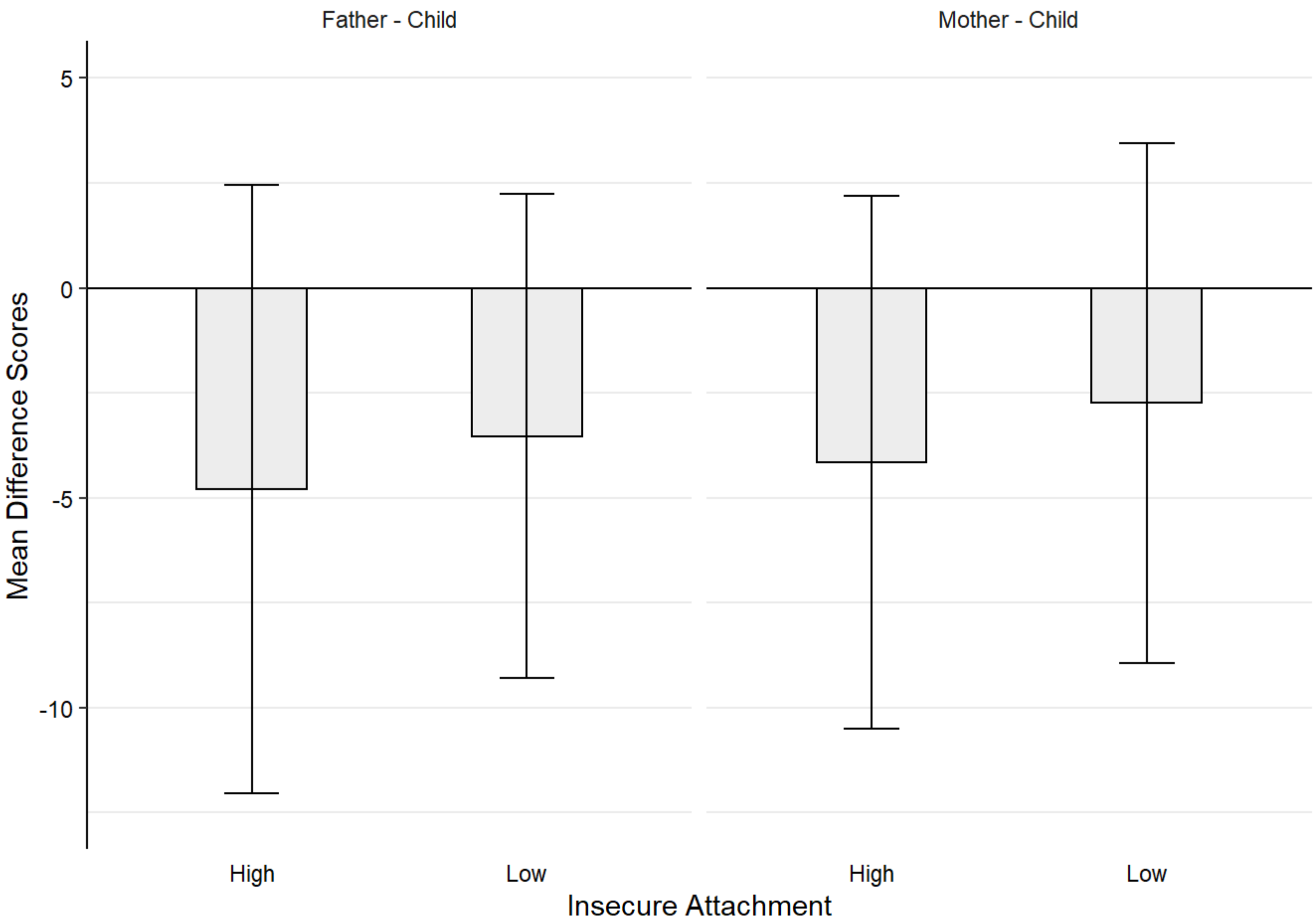
Creating a bar graph for the thesis with externalizing symptoms and adult attachment:

```
avg_diff_score_aaa_df %>%
  filter(Type == "ext") %>%
  ggplot(aes(x = Attachment, y = Mean)) +
  geom_bar(stat = "identity", color = "black", fill = "grey79", width = 0.4) +
  geom_errorbar(aes(ymin = Mean - sd, ymax = Mean + sd), width = 0.20) +
  facet_wrap(vars(raters)) +
  ylim(-12.5, 5) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  geom_hline(aes(yintercept = 0)) +
  labs(
    x = "Insecure Attachment",
    y = "Mean Difference Scores"
  )
```



Creating a bar graph for the thesis with internalizing symptoms and adult attachment:

```
avg_diff_score_aaa_df %>%
  filter(Type == "int") %>%
  ggplot(aes(x = Attachment, y = Mean)) +
  geom_bar(stat = "identity", color = "black", fill = "grey93", width = 0.4) +
  geom_errorbar(aes(ymin = Mean - sd, ymax = Mean + sd), width = 0.20) +
  facet_wrap(vars(raters)) +
  ylim(-12.5, 5) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  geom_hline(aes(yintercept = 0)) +
  labs(
    x = "Insecure Attachment",
    y = "Mean Difference Scores"
  )
```



Correlating difference scores with attachment:

```
diff_corr_ext_f <- data_df_wide_diff %>%
  select(y8diff_sc_ext_fc, y10aaa_scale_ya_f_dich) %>%
  # Convert to numerical variable
  mutate(y10aaa_scale_ya_f_dich = case_when(
    y10aaa_scale_ya_f_dich == "Low" ~ 0,
    y10aaa_scale_ya_f_dich == "High" ~ 1)
    ) %>%
  summarize(correlation_matrix = list(correlation(across(everything()))))

diff_corr_ext_m <- data_df_wide_diff %>%
  select(y8diff_sc_ext_mc, y10aaa_scale_ya_m_dich) %>%
  # Convert to numerical variable
  mutate(y10aaa_scale_ya_m_dich = case_when(
    y10aaa_scale_ya_m_dich == "Low" ~ 0,
    y10aaa_scale_ya_m_dich == "High" ~ 1)
    ) %>%
  summarize(correlation_matrix = list(correlation(across(everything()))))

diff_corr_int_f <- data_df_wide_diff %>%
  select(y8diff_sc_int_fc, y10aaa_scale_ya_f_dich) %>%
  # Convert to numerical variable
  mutate(y10aaa_scale_ya_f_dich = case_when(
    y10aaa_scale_ya_f_dich == "Low" ~ 0,
```

```
    y10aaa_scale_ya_f_dich == "High" ~ 1)
    ) %>%
  summarize(correlation_matrix = list(correlation(across(everything()))))

diff_corr_int_m <- data_df_wide_diff %>%
  select(y8diff_sc_int_mc, y10aaa_scale_ya_m_dich) %>%
  # Convert to numerical variable
  mutate(y10aaa_scale_ya_m_dich = case_when(
    y10aaa_scale_ya_m_dich == "Low" ~ 0,
    y10aaa_scale_ya_m_dich == "High" ~ 1)
    ) %>%
  summarize(correlation_matrix = list(correlation(across(everything()))))

print(diff_corr_ext_f[[1]])
```

```
[[1]]
# Correlation Matrix (pearson-method)

Parameter1       |             Parameter2 |    r |        95% CI | t(218) |     p
---------------------------------------------------------------------------------
y8diff_sc_ext_fc | y10aaa_scale_ya_f_dich | 0.06 | [-0.07, 0.19] |   0.95 | 0.343

p-value adjustment method: Holm (1979)
Observations: 220
```

```
print(diff_corr_ext_m[[1]])
```

```
[[1]]
# Correlation Matrix (pearson-method)

Parameter1       |             Parameter2 |    r |        95% CI | t(277) |     p
---------------------------------------------------------------------------------
y8diff_sc_ext_mc | y10aaa_scale_ya_m_dich | 0.02 | [-0.10, 0.14] |   0.37 | 0.712

p-value adjustment method: Holm (1979)
Observations: 279
```

```
print(diff_corr_int_f[[1]])
```

```
[[1]]
# Correlation Matrix (pearson-method)

Parameter1       |             Parameter2 |     r |        95% CI | t(217) |     p
----------------------------------------------------------------------------------
y8diff_sc_int_fc | y10aaa_scale_ya_f_dich | -0.10 | [-0.23, 0.04] |  -1.44 | 0.151

p-value adjustment method: Holm (1979)
Observations: 219
```

```
print(diff_corr_int_m[[1]])
```

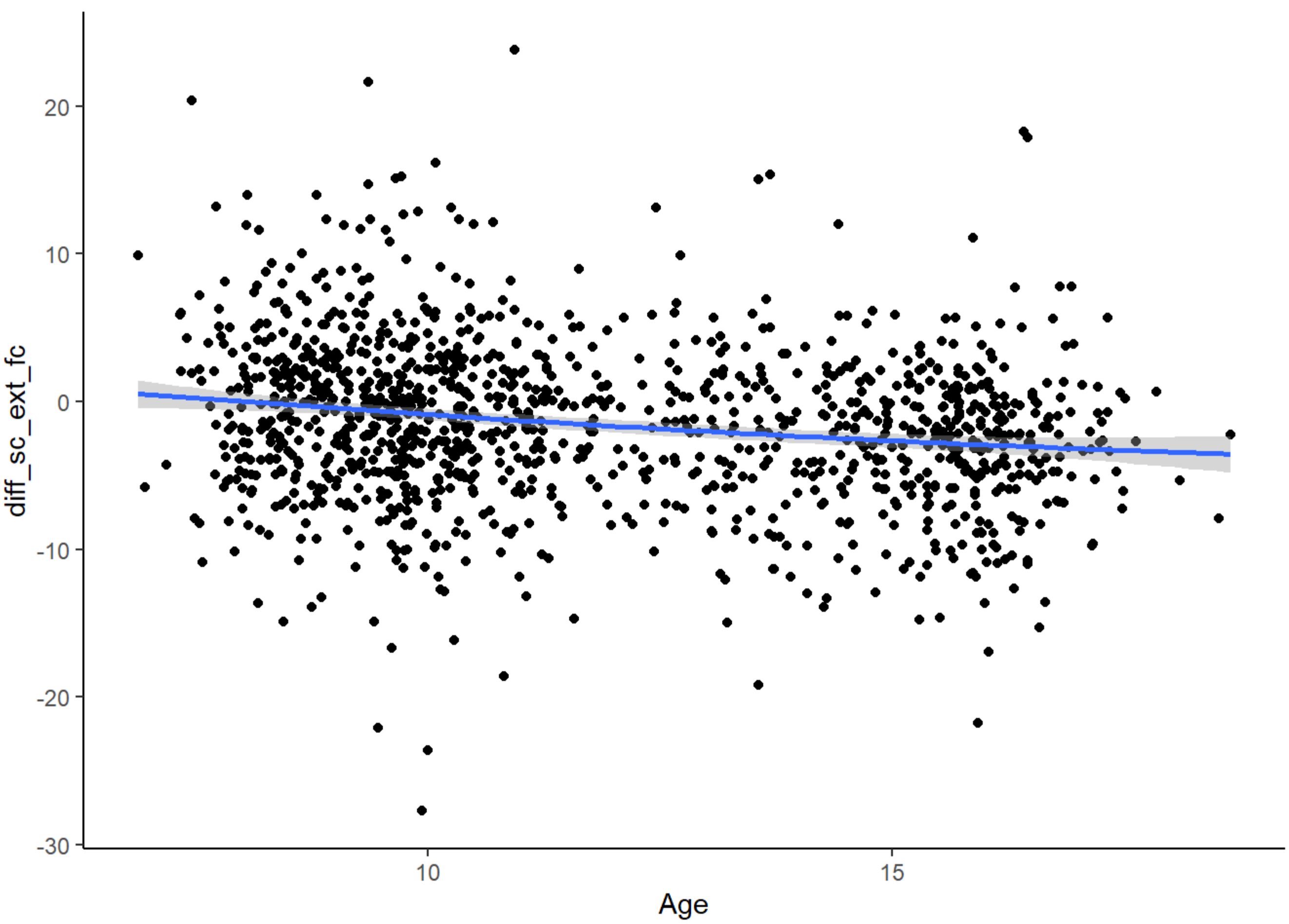
```
[[1]]
# Correlation Matrix (pearson-method)

Parameter1       |             Parameter2 |     r |        95% CI | t(274) |     p
---------------------------------------------------------------------------------
y8diff_sc_int_mc | y10aaa_scale_ya_m_dich | -0.11 | [-0.23, 0.01] |  -1.87 | 0.063

p-value adjustment method: Holm (1979)
Observations: 276
```

6.4 - Visualizing Patterns

Creating a scatterplot between difference scores and age of the child:

```
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = diff_sc_ext_fc)) +
  geom_jitter() +
  geom_smooth(method = "auto") +
  theme_classic()
```

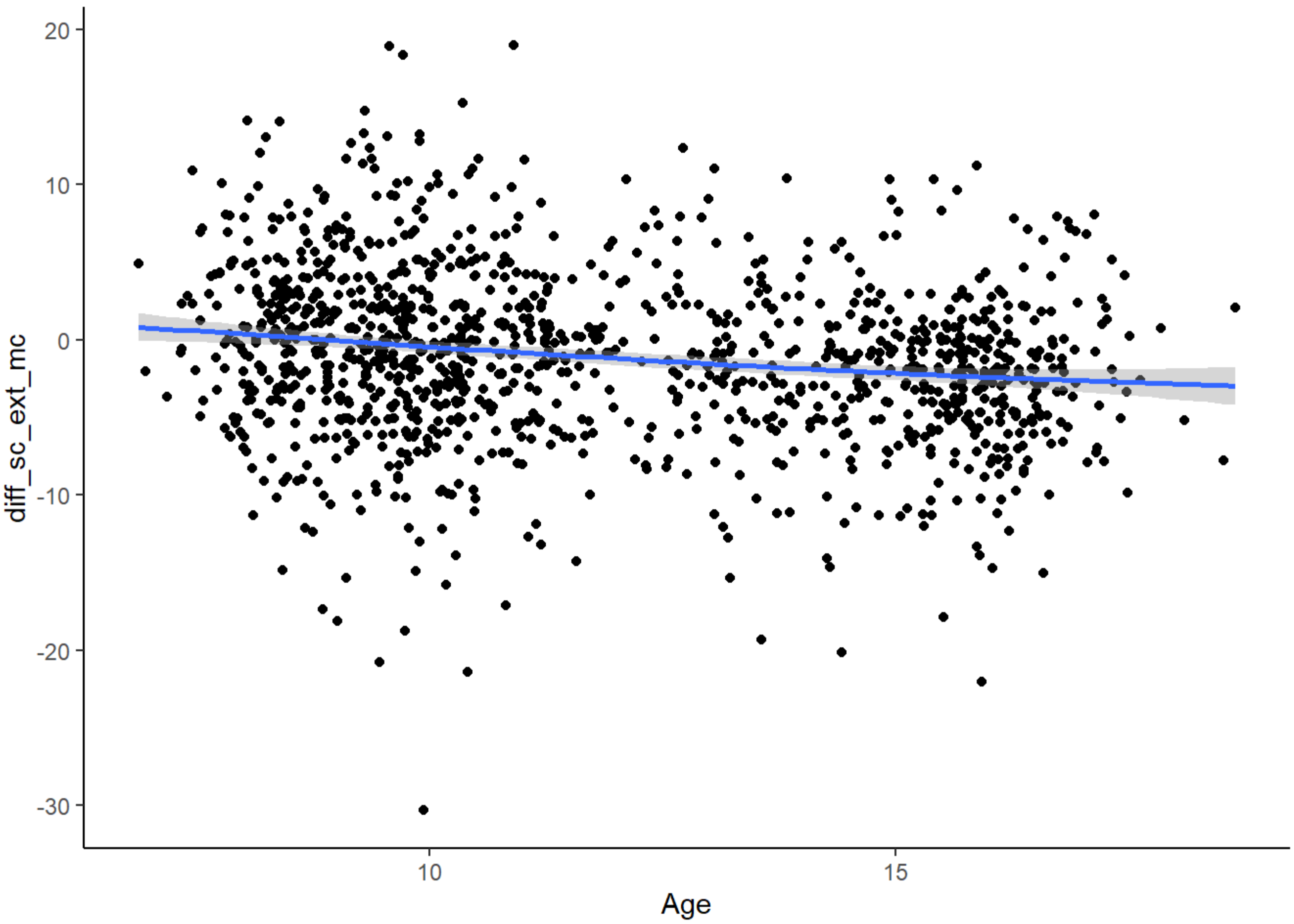
```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
data_df_long_unique %>%
```

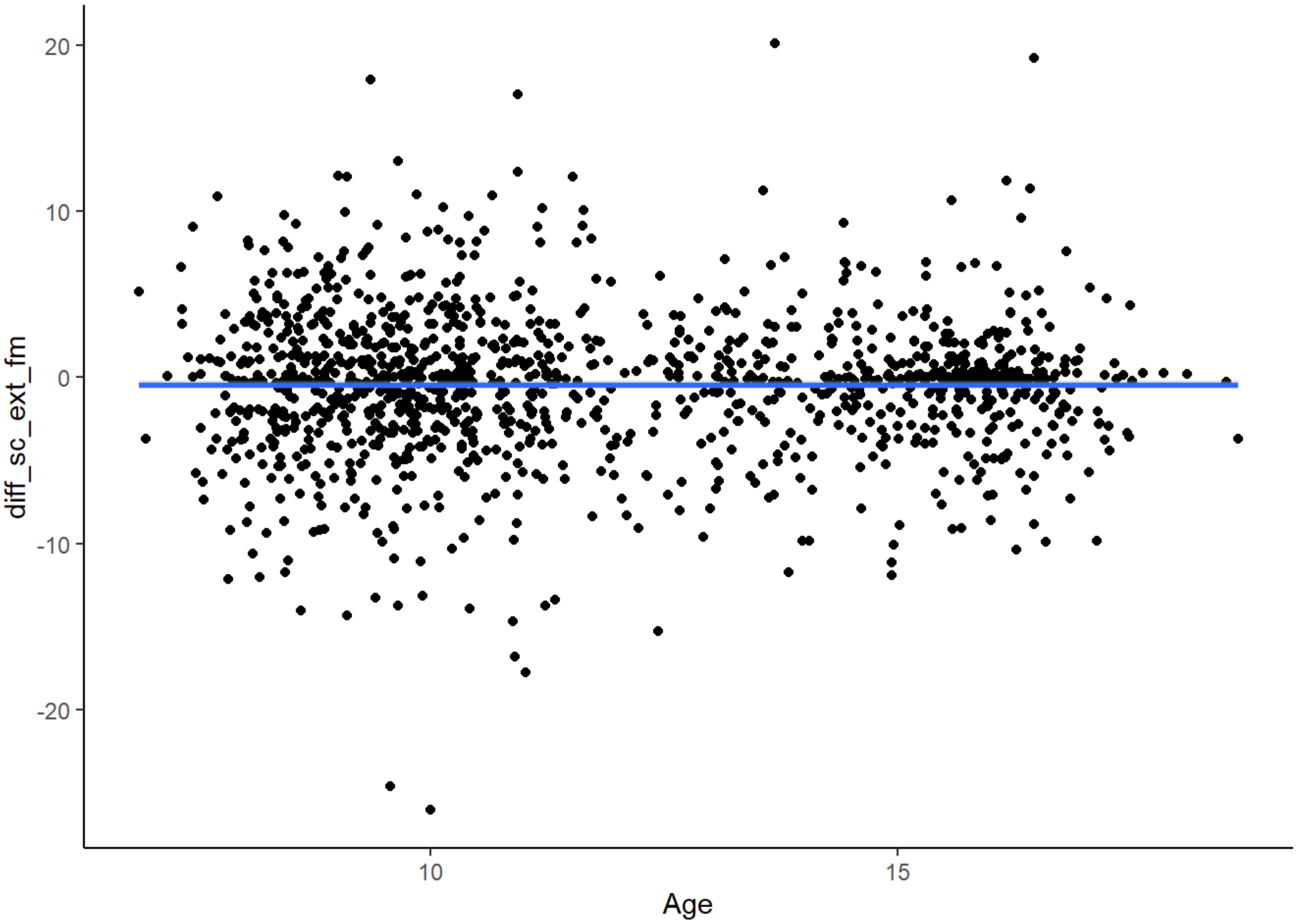
```
  ggplot(aes(x = Age ,y = diff_sc_ext_mc)) +
  geom_jitter() +
  geom_smooth(method = "auto") +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



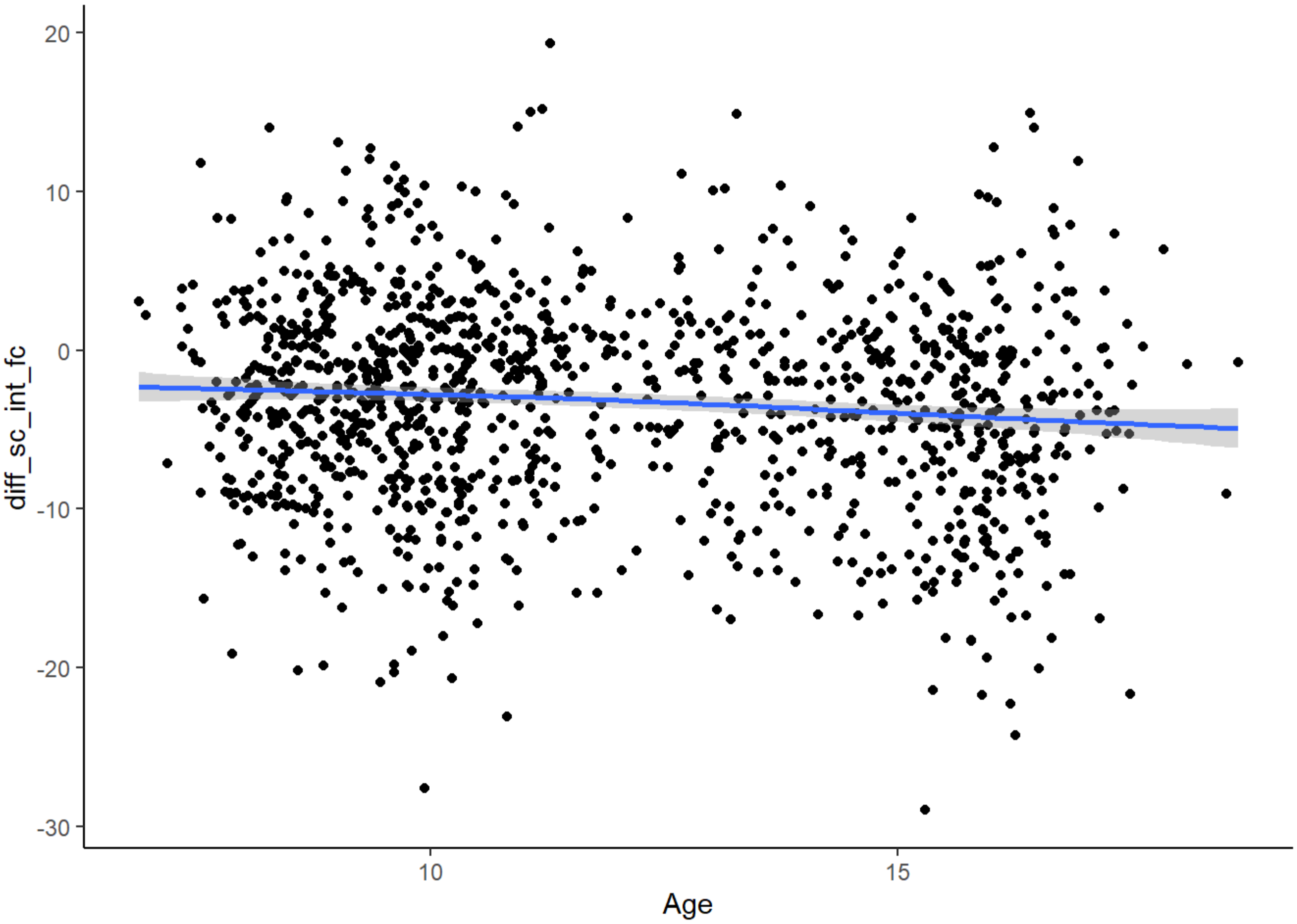

```
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = diff_sc_ext_fm)) +
  geom_jitter() +
  geom_smooth(method = "auto") +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



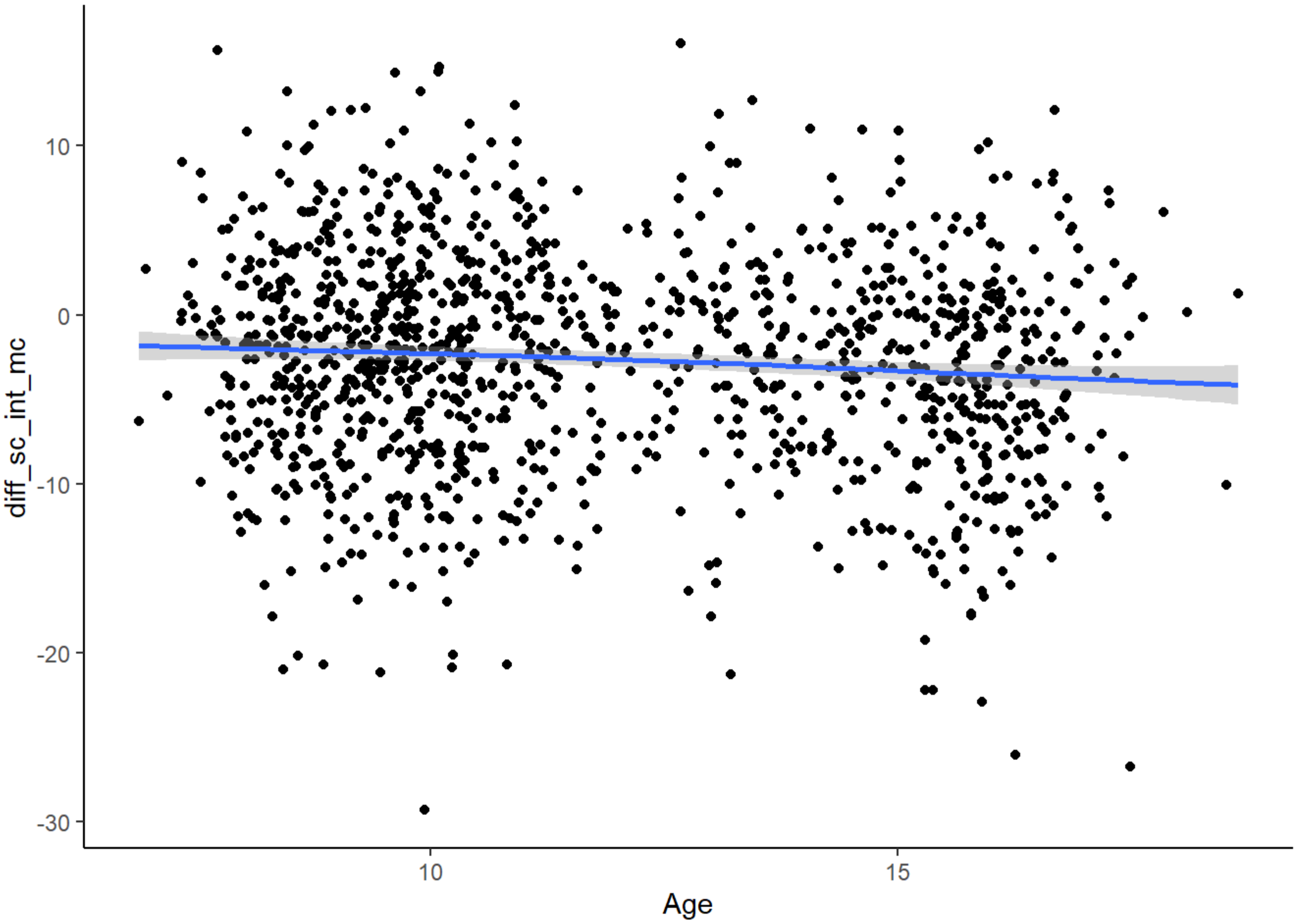

```
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = diff_sc_int_fc)) +
  geom_jitter() +
  geom_smooth(method = "auto") +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



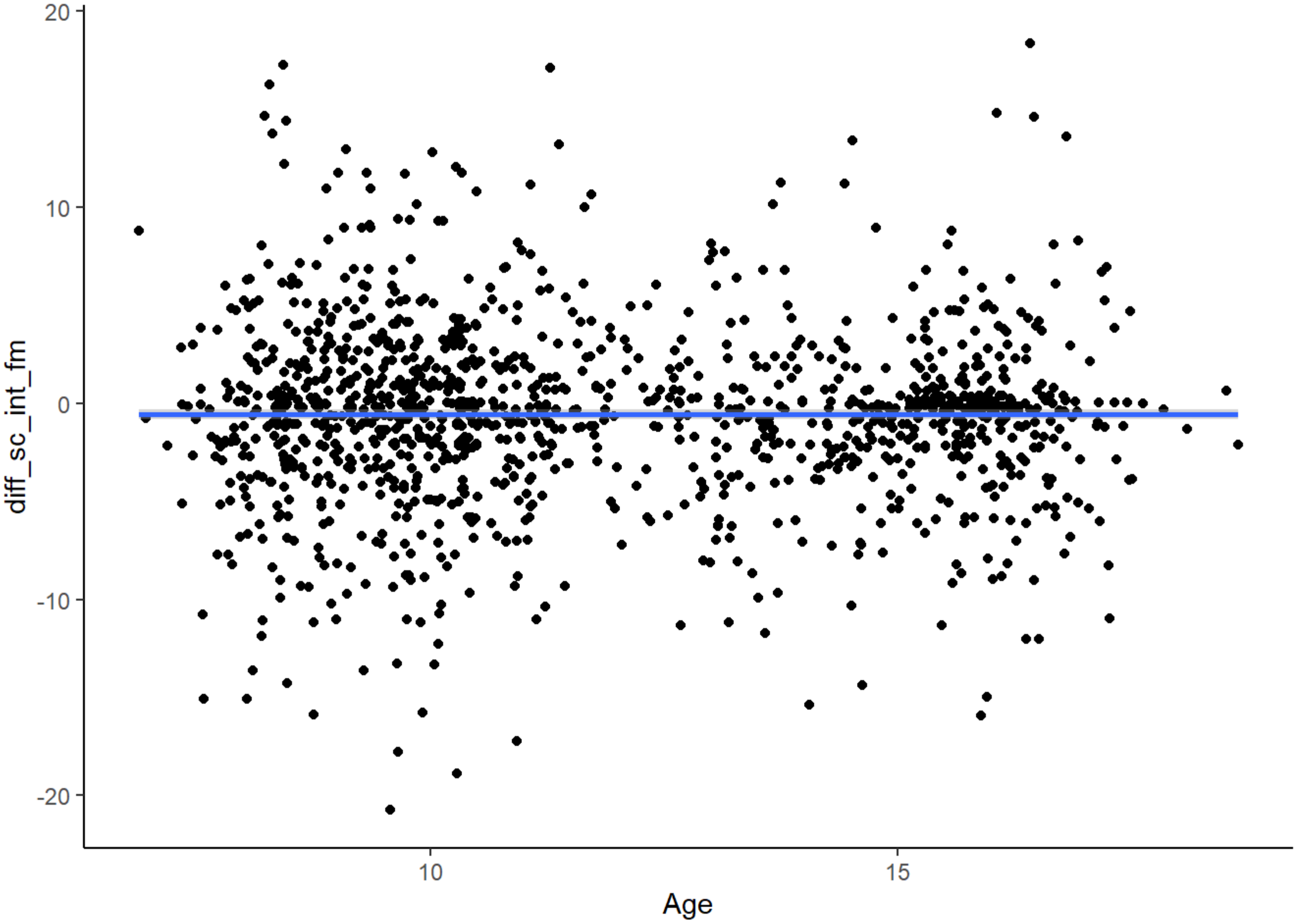

```r
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = diff_sc_int_mc)) +
  geom_jitter() +
  geom_smooth(method = "auto") +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



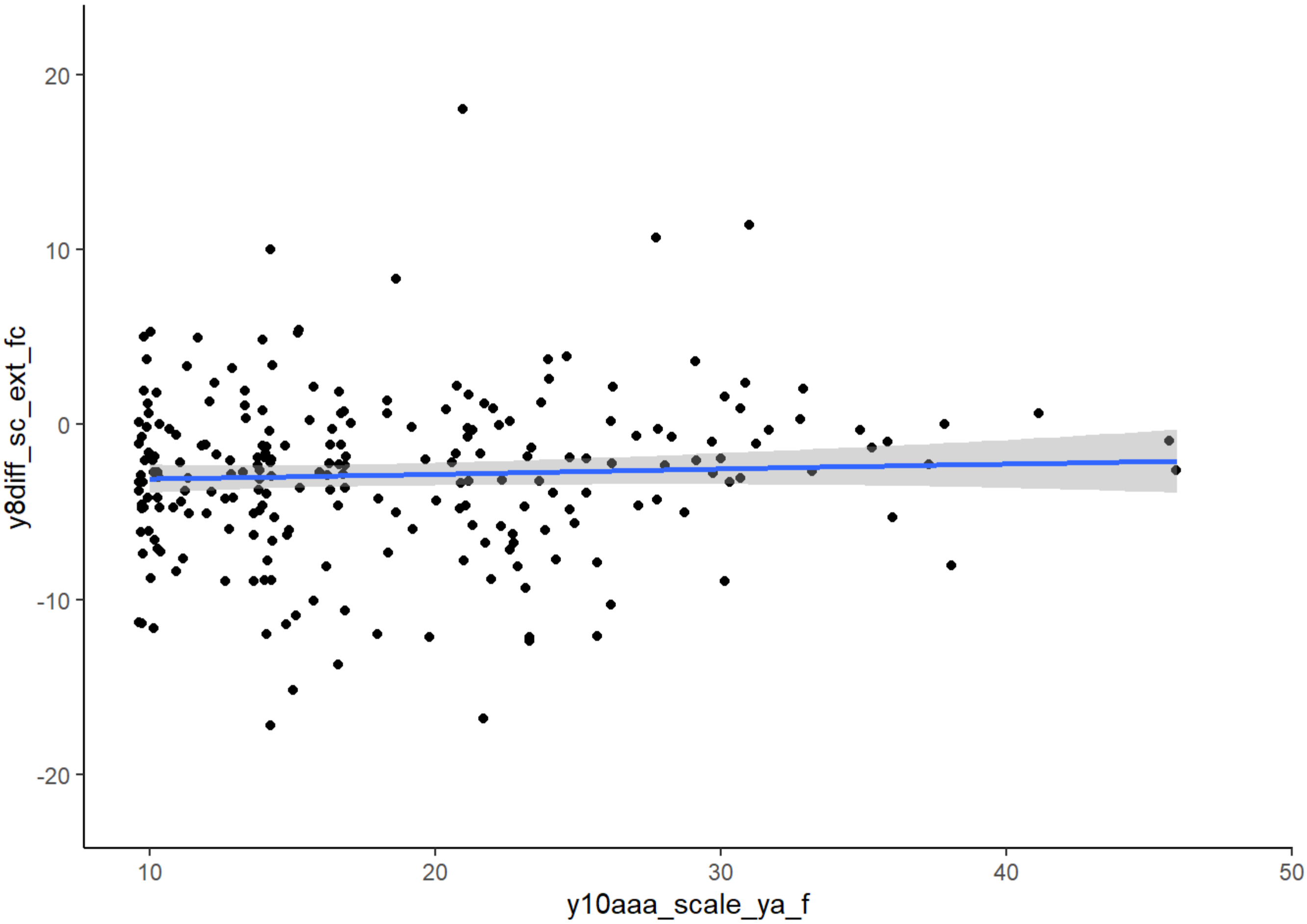

```
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = diff_sc_int_fm)) +
  geom_jitter() +
  geom_smooth(method = "auto") +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



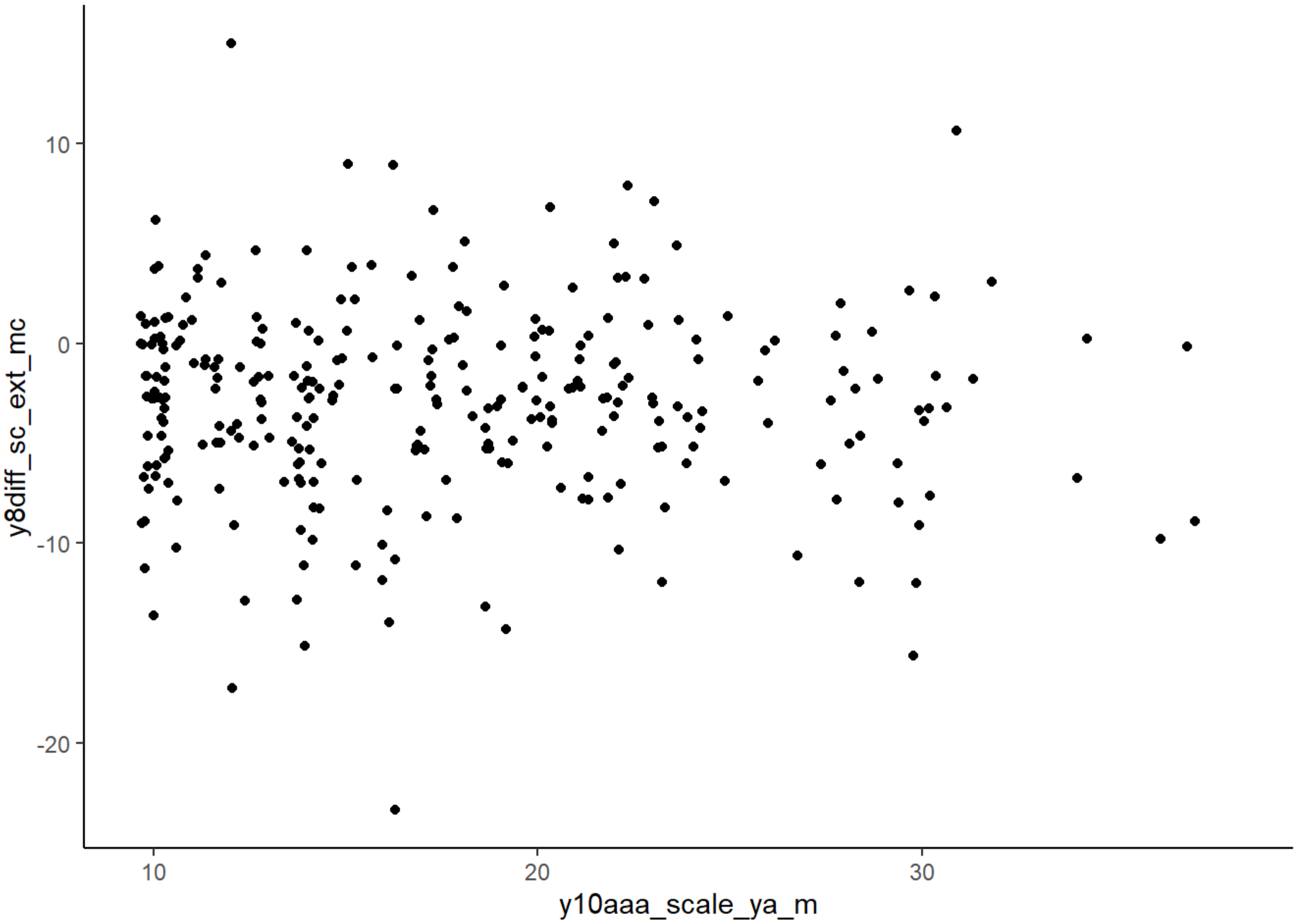

```
data_df_wide_diff %>%
  ggplot(aes(x = y10aaa_scale_ya_f ,y = y8diff_sc_ext_fc)) +
  geom_jitter() +
  geom_smooth(method = "auto") +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'

Warning: Removed 1221 rows containing non-finite outside the scale range
(`stat_smooth()`).

Warning: Removed 1221 rows containing missing values or values outside the scale range
(`geom_point()`).
```



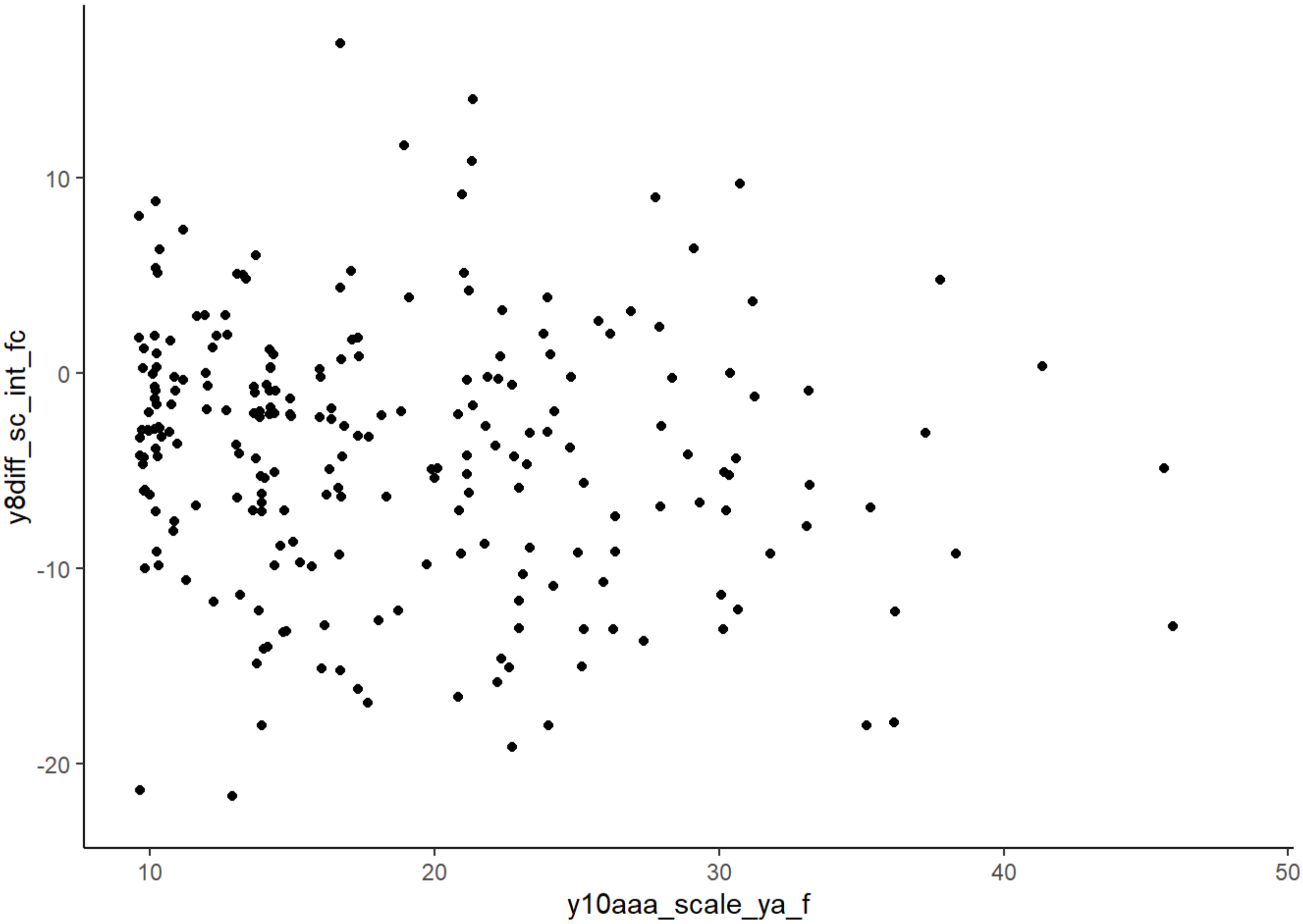

```
data_df_wide_diff %>%
  ggplot(aes(x = y10aaa_scale_ya_m ,y = y8diff_sc_ext_mc)) +
  geom_jitter() +
  theme_classic()
```

```
Warning: Removed 1163 rows containing missing values or values outside the scale range
(`geom_point()`).
```



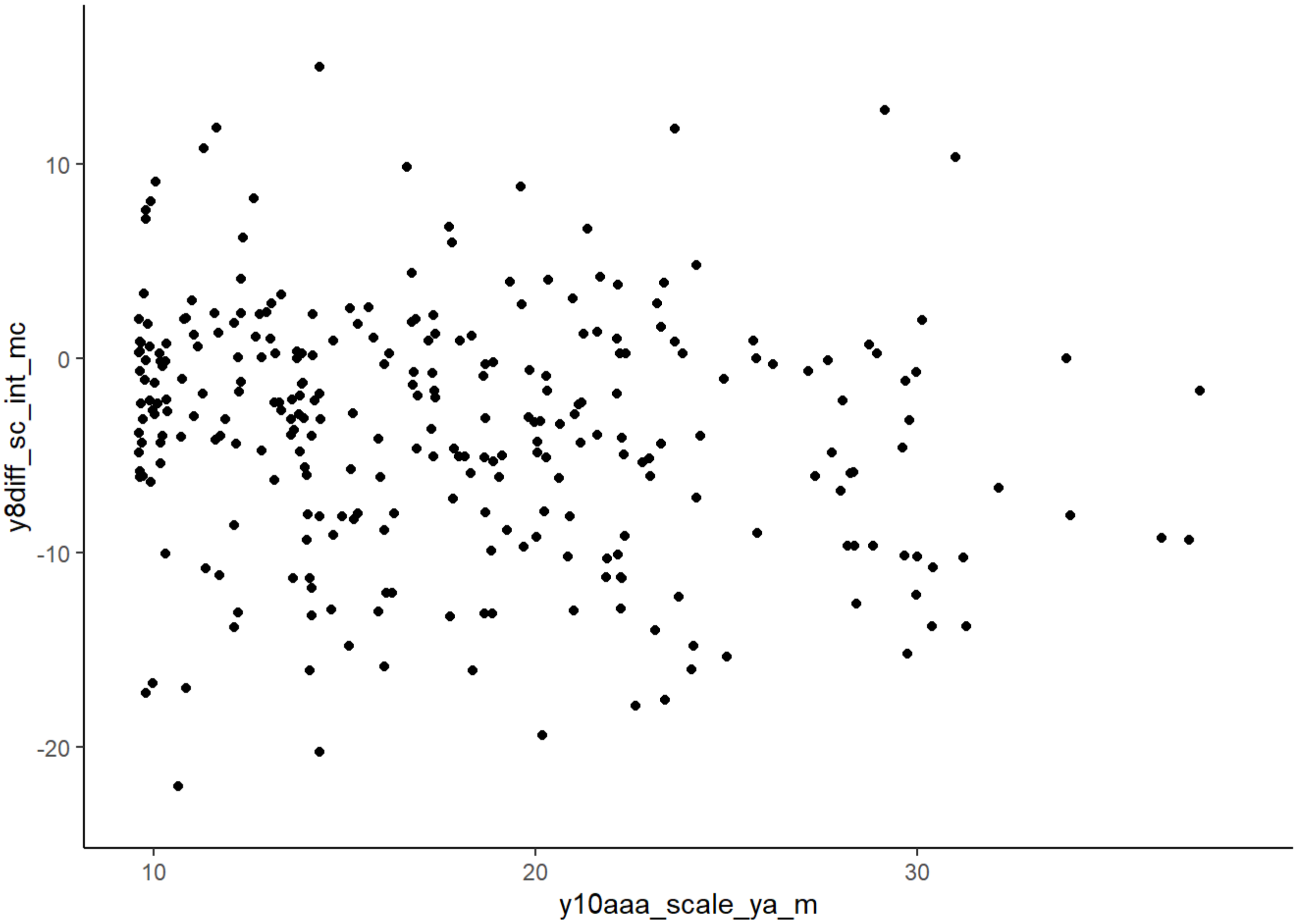

```
data_df_wide_diff %>%
  ggplot(aes(x = y10aaa_scale_ya_f ,y = y8diff_sc_int_fc)) +
  geom_jitter() +
  theme_classic()
```

```
Warning: Removed 1220 rows containing missing values or values outside the scale range
(`geom_point()`).
```



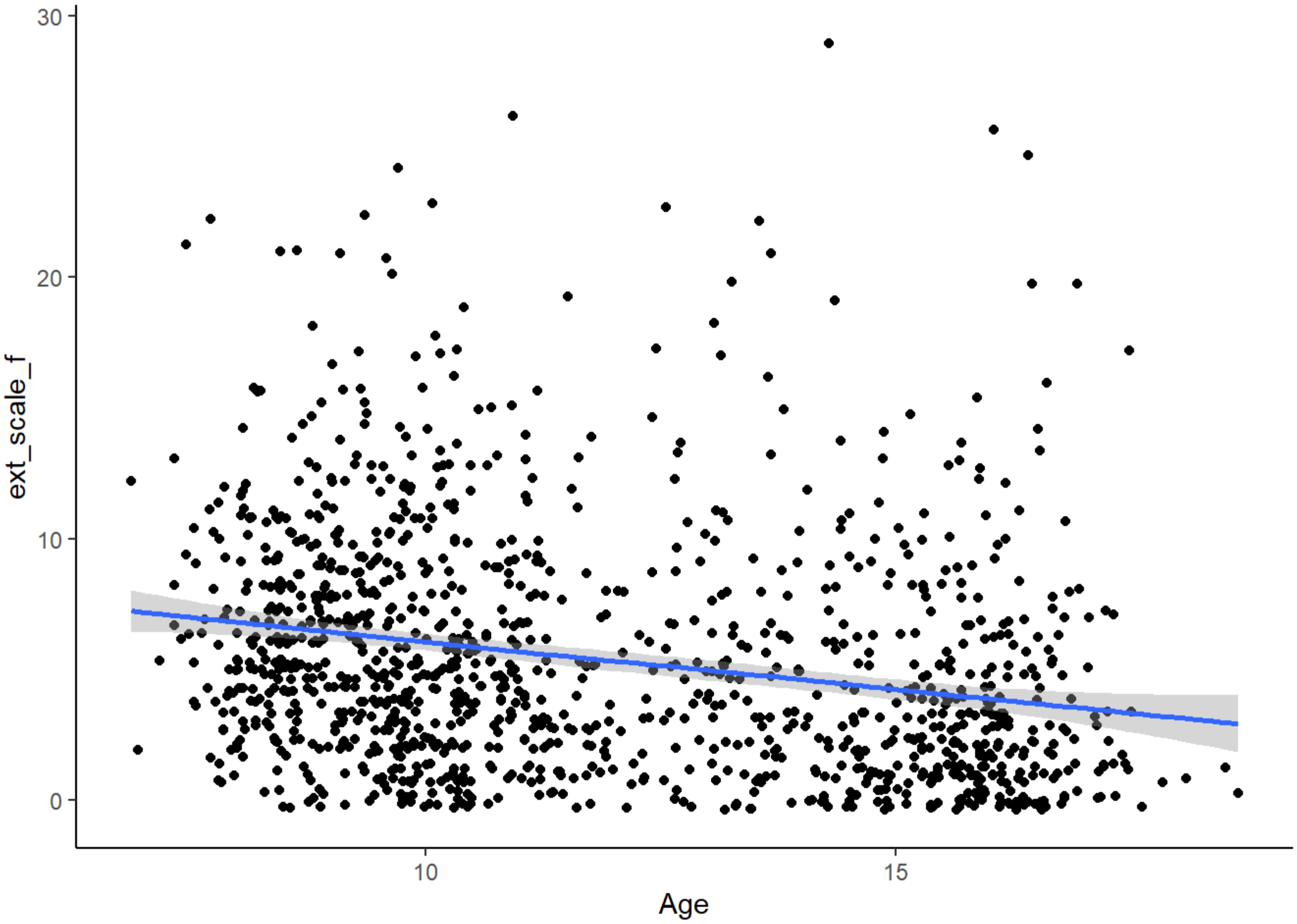

```
data_df_wide_diff %>%
  ggplot(aes(x = y10aaa_scale_ya_m ,y = y8diff_sc_int_mc)) +
  geom_jitter() +
  theme_classic()
```

```
Warning: Removed 1166 rows containing missing values or values outside the scale range
(`geom_point()`).
```



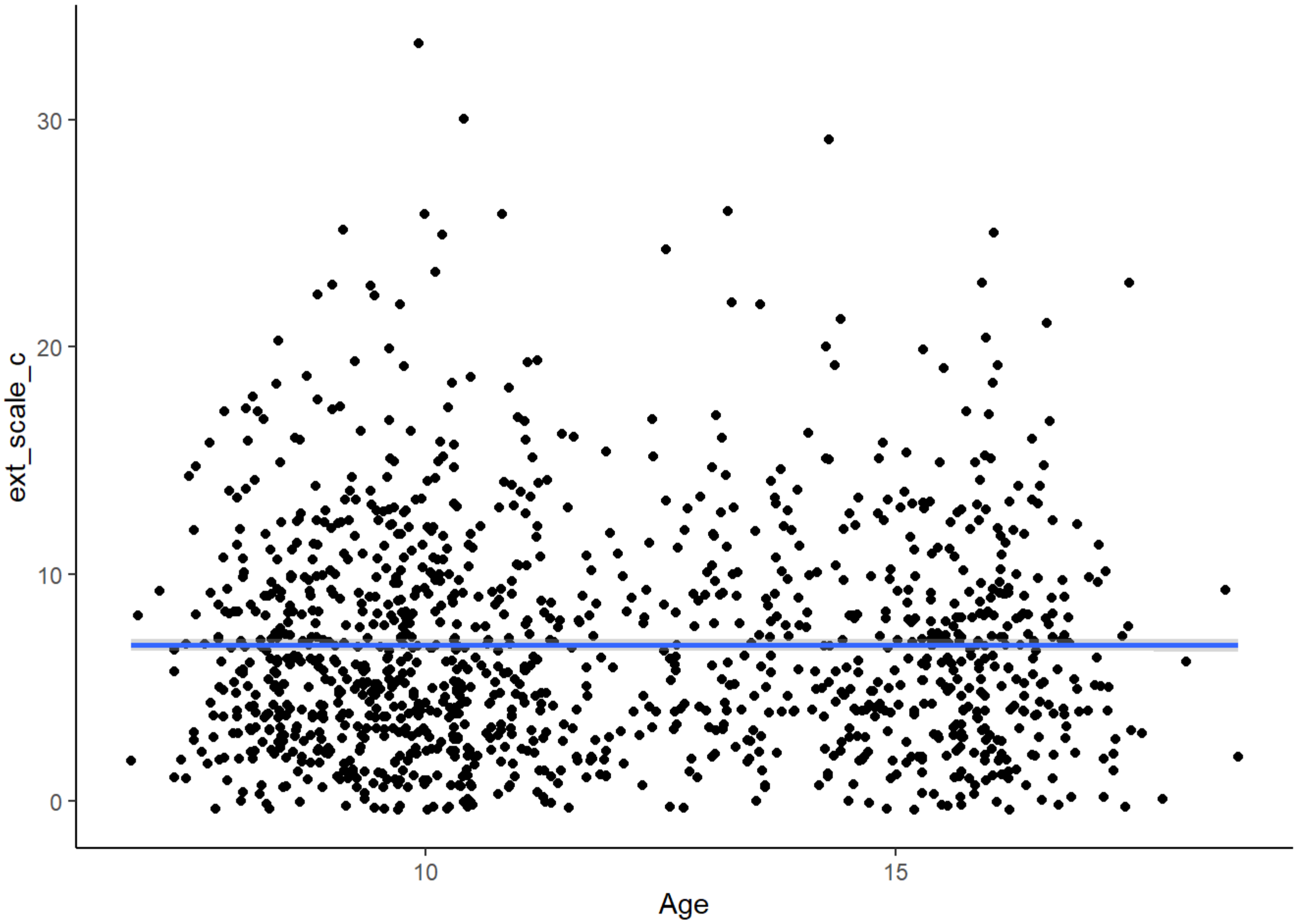

```
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = ext_scale_f)) +
  geom_jitter() +
  geom_smooth() +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



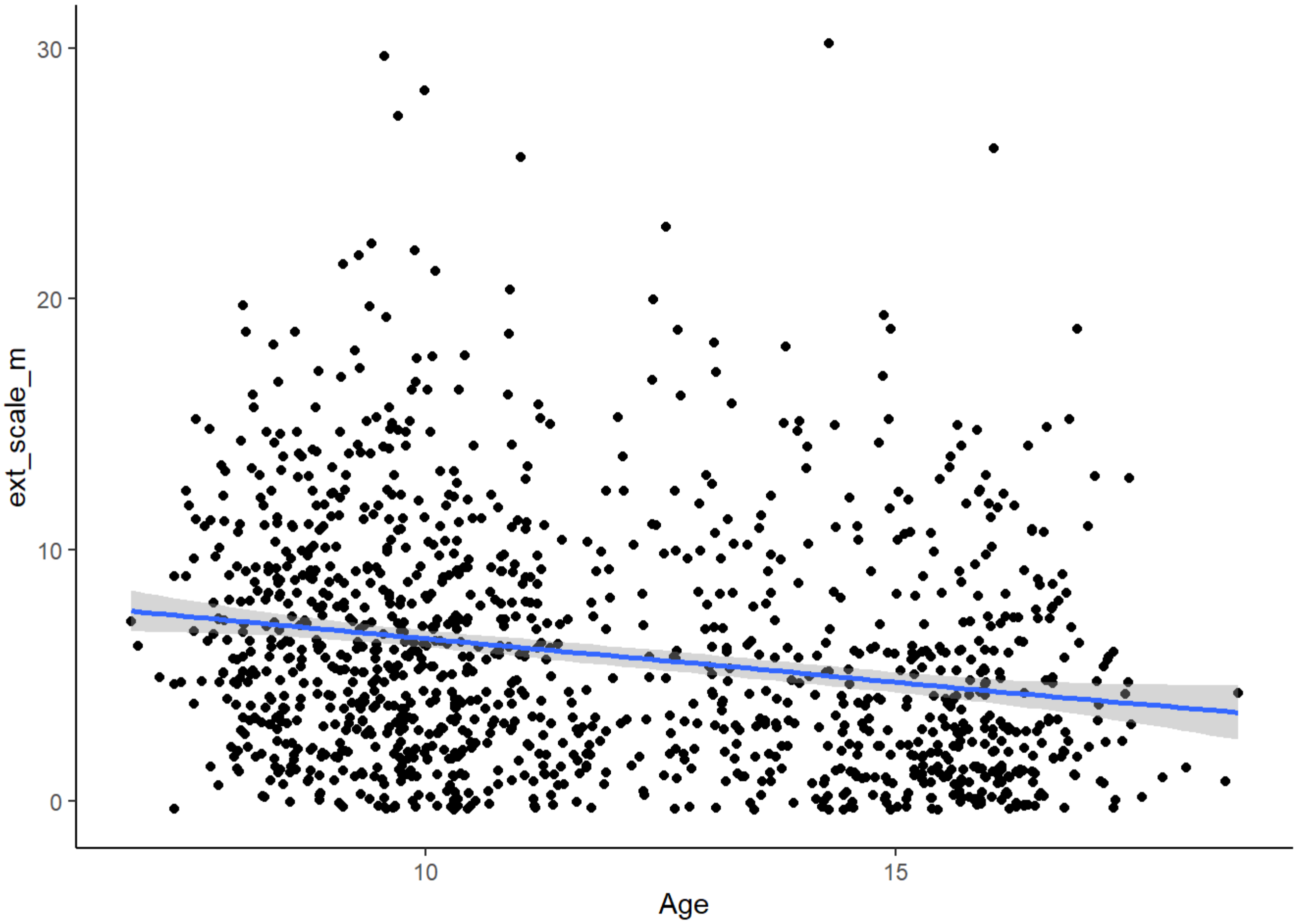

```
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = ext_scale_c)) +
  geom_jitter() +
  geom_smooth() +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



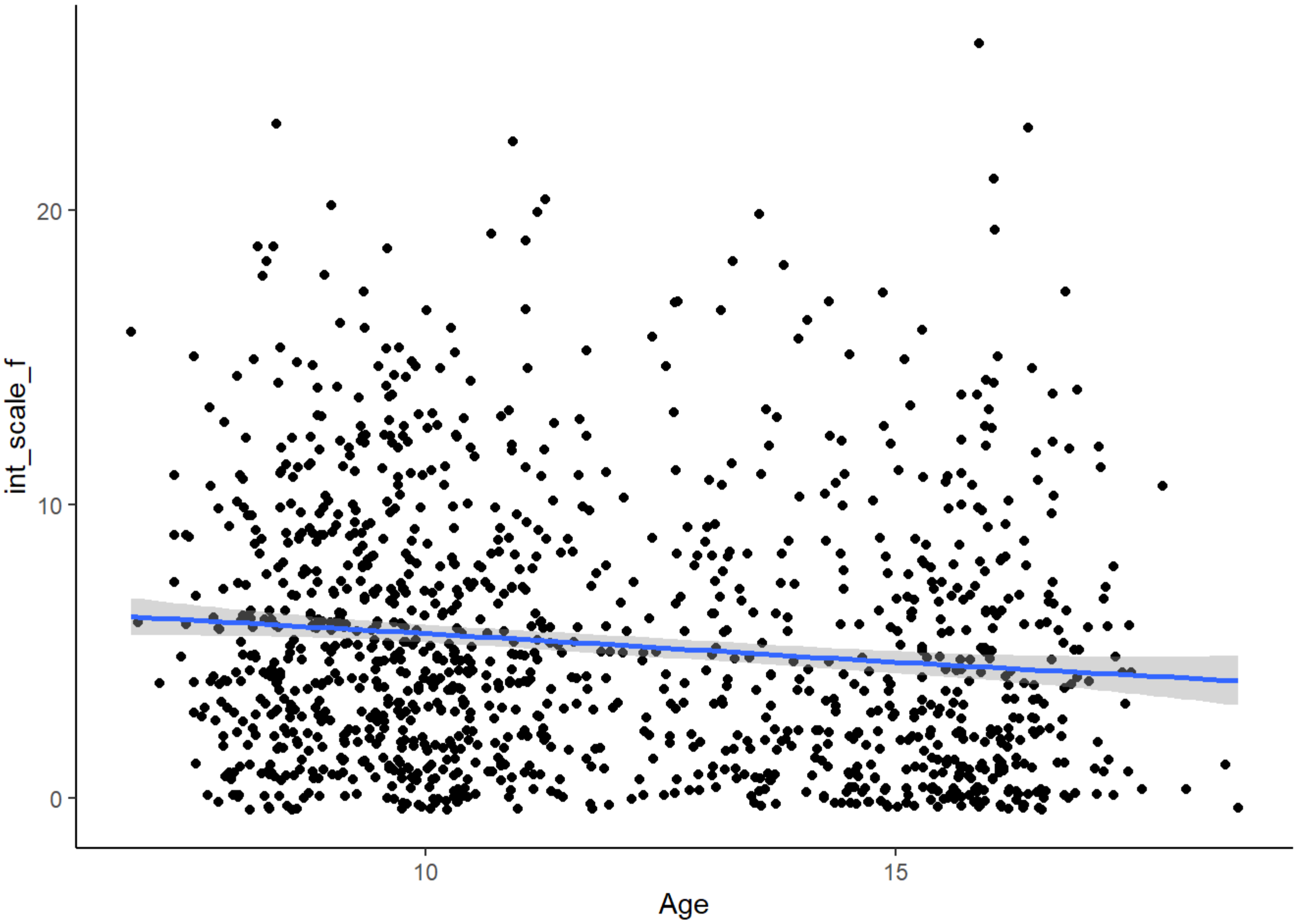

```r
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = ext_scale_m)) +
  geom_jitter() +
  geom_smooth() +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



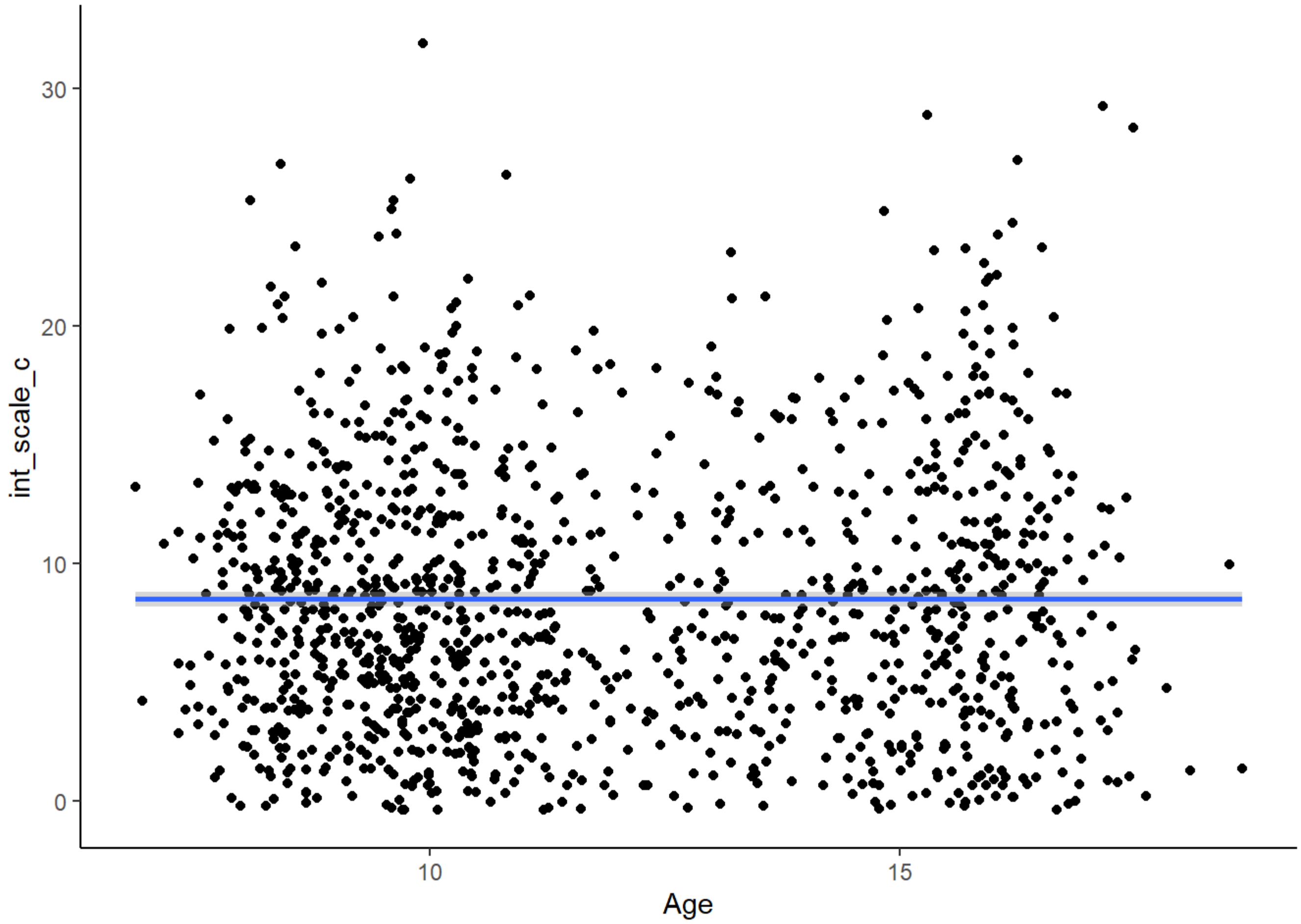

```
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = int_scale_f)) +
  geom_jitter() +
  geom_smooth() +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



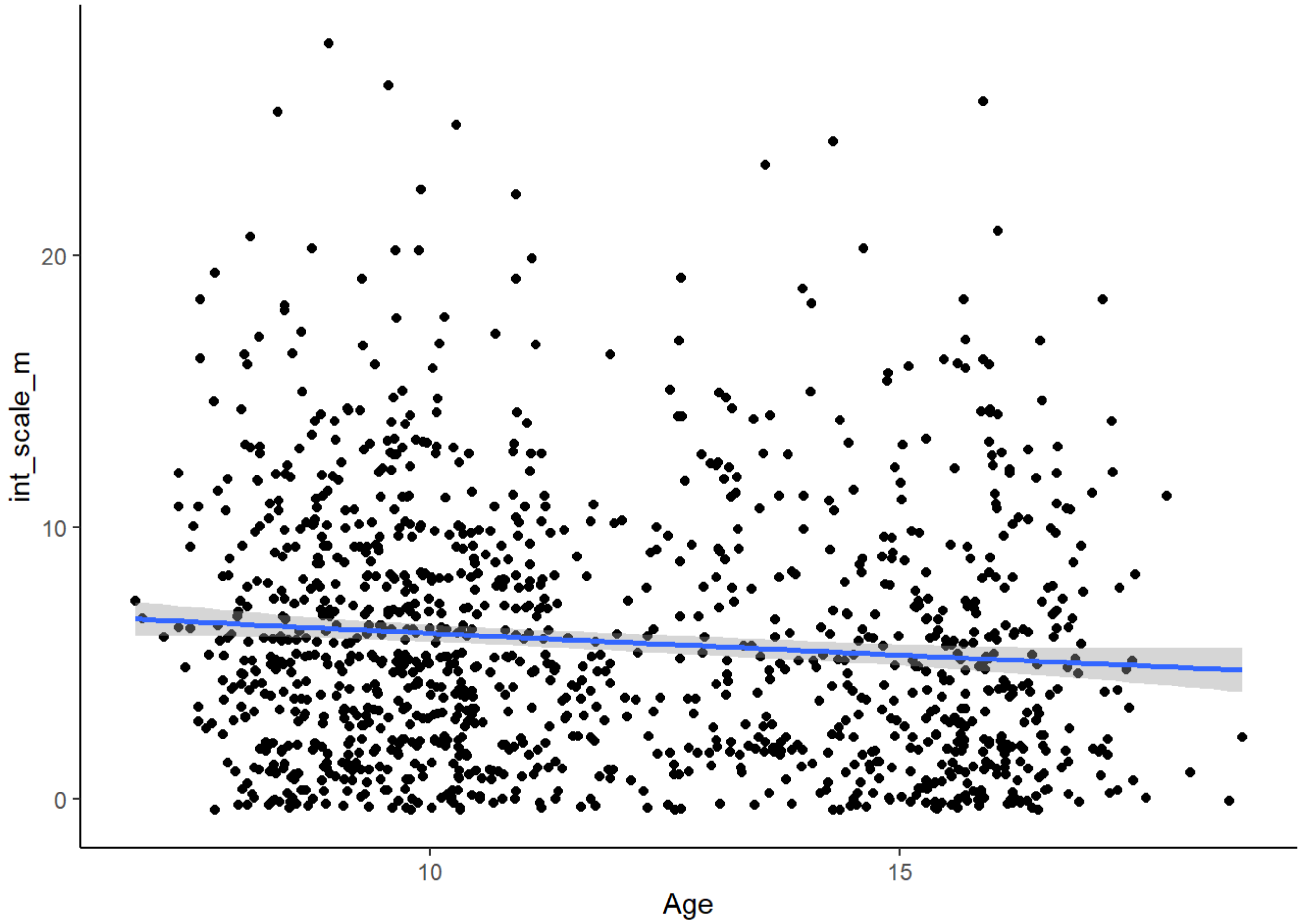

```
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = int_scale_c)) +
  geom_jitter() +
  geom_smooth() +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
data_df_long_unique %>%
  ggplot(aes(x = Age ,y = int_scale_m)) +
  geom_jitter() +
  geom_smooth() +
  theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



7 - Pearson Correlation

We will calculate Pearson correlation for the ratings between the different raters for both rater disagreement in general, as well as divided up by age.

The results were checked to match with the case r, cor function. correlate() from corr was used instead of cor() from base r, since this resturns the output as a tibble, which is easier to work with than the matrix cor() returns.

7.1 - In general

```
data_df_long_unique %>%
  select(ext_scale_f, ext_scale_m, ext_scale_c) %>%
  correlation()
```

```
# Correlation Matrix (pearson-method)

Parameter1  |  Parameter2 |    r |       95% CI | t(1246) |         p
---------------------------------------------------------------------
ext_scale_f | ext_scale_m | 0.57 | [0.53, 0.60] |   24.24 | < .001***
ext_scale_f | ext_scale_c | 0.33 | [0.28, 0.38] |   12.32 | < .001***
ext_scale_m | ext_scale_c | 0.37 | [0.32, 0.42] |   14.12 | < .001***
```

```
p-value adjustment method: Holm (1979)
Observations: 1248
```

Plotting the results:

```
# Expanded on a solution by @rosie-betzler on Stackoverflow
https://stackoverflow.com/questions/3735286/create-a-matrix-of-scatterplots-pairs-equivalent-in-
ggplot2

x_df <- data_df_long_unique %>%
  select(ID, ext_scale_f, ext_scale_m, ext_scale_c) %>%
  pivot_longer(c(ext_scale_f, ext_scale_m, ext_scale_c), names_to = "rater_x", values_to = "val_x")
%>%
  mutate(
        rater_x = case_when(
            rater_x == "ext_scale_f" ~ "Father",
            rater_x == "ext_scale_m" ~ "Mother",
            rater_x == "ext_scale_c" ~ "Child"
        ))

y_df <- x_df %>%
  rename(rater_y = "rater_x", val_y = "val_x")

xy_df <- full_join(x_df, y_df,
          by = c("ID"),
          relationship = "many-to-many")

xy_df %>%
  unite("combo", rater_x, rater_y, sep = " & ") %>%
  filter(combo %in% c("Father & Mother", "Father & Child", "Mother & Child")) %>%
  ggplot(aes(x = val_x, y = val_y)) +
  geom_point(position = "jitter") +
  geom_smooth(method = lm, se = FALSE, linewidth = 1.3) +
  facet_wrap(vars(combo)) +
  theme_minimal() +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank()
  ) +
  labs(title = "Relationship Between Raters and Their Correlations for Externalizing Symptoms")
```

```
`geom_smooth()` using formula = 'y ~ x'
```

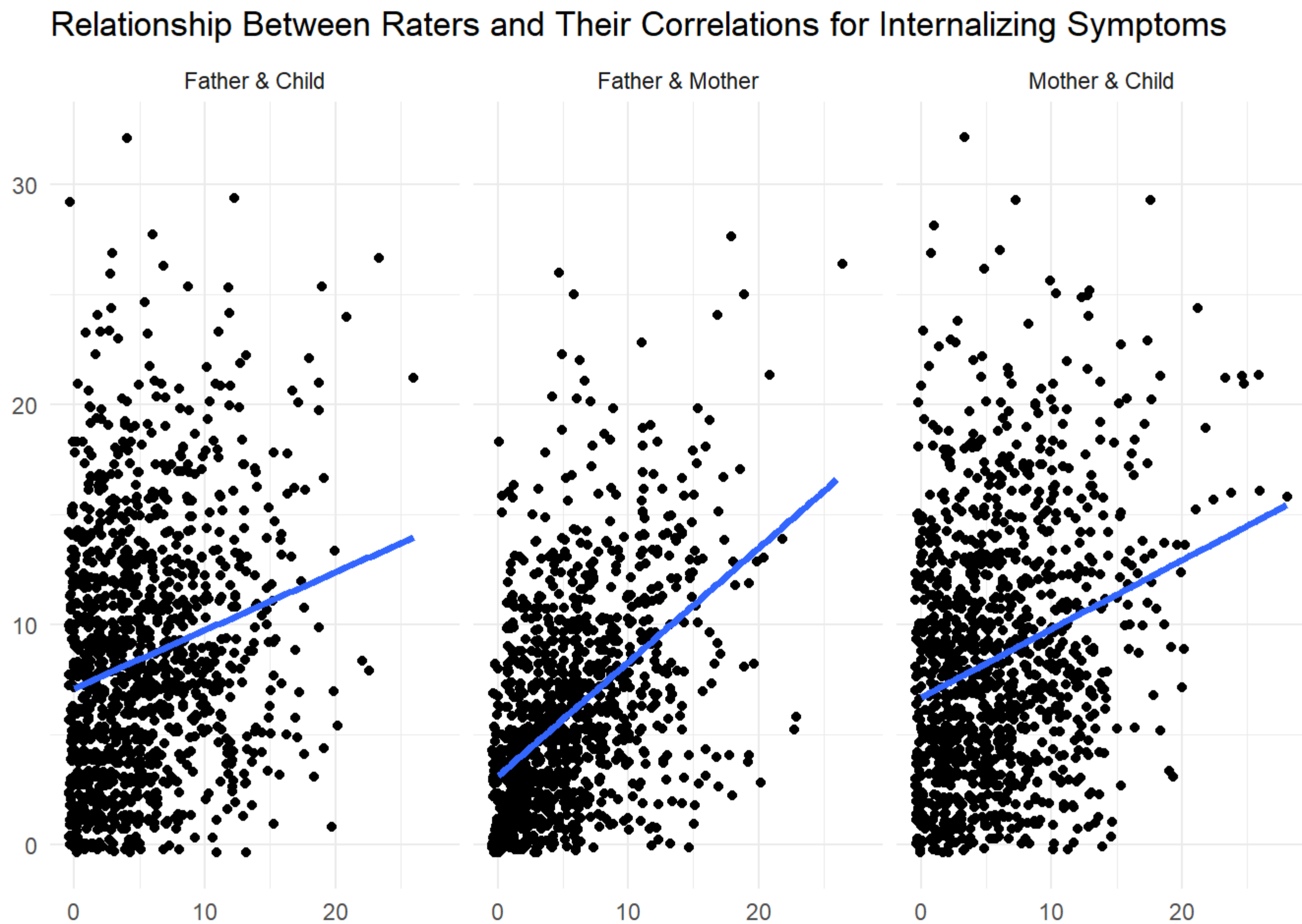

```
data_df_long_unique %>%
  select(int_scale_f, int_scale_m, int_scale_c) %>%
  correlation()
```

```
# Correlation Matrix (pearson-method)

Parameter1  |  Parameter2 |    r |        95% CI | t(1246) |         p
---------------------------------------------------------------------
int_scale_f | int_scale_m | 0.49 | [0.45, 0.53] |   19.92 | < .001***
int_scale_f | int_scale_c | 0.21 | [0.16, 0.26] |    7.59 | < .001***
int_scale_m | int_scale_c | 0.26 | [0.21, 0.31] |    9.63 | < .001***

p-value adjustment method: Holm (1979)
Observations: 1248
```

```
# Expanded on a solution by @rosie-betzler on Stackoverflow
https://stackoverflow.com/questions/3735286/create-a-matrix-of-scatterplots-pairs-equivalent-in-ggplot2

x_df <- data_df_long_unique %>%
  select(ID, int_scale_f, int_scale_m, int_scale_c) %>%
  pivot_longer(c(int_scale_f, int_scale_m, int_scale_c), names_to = "rater_x", values_to = "val_x")
%>%
```

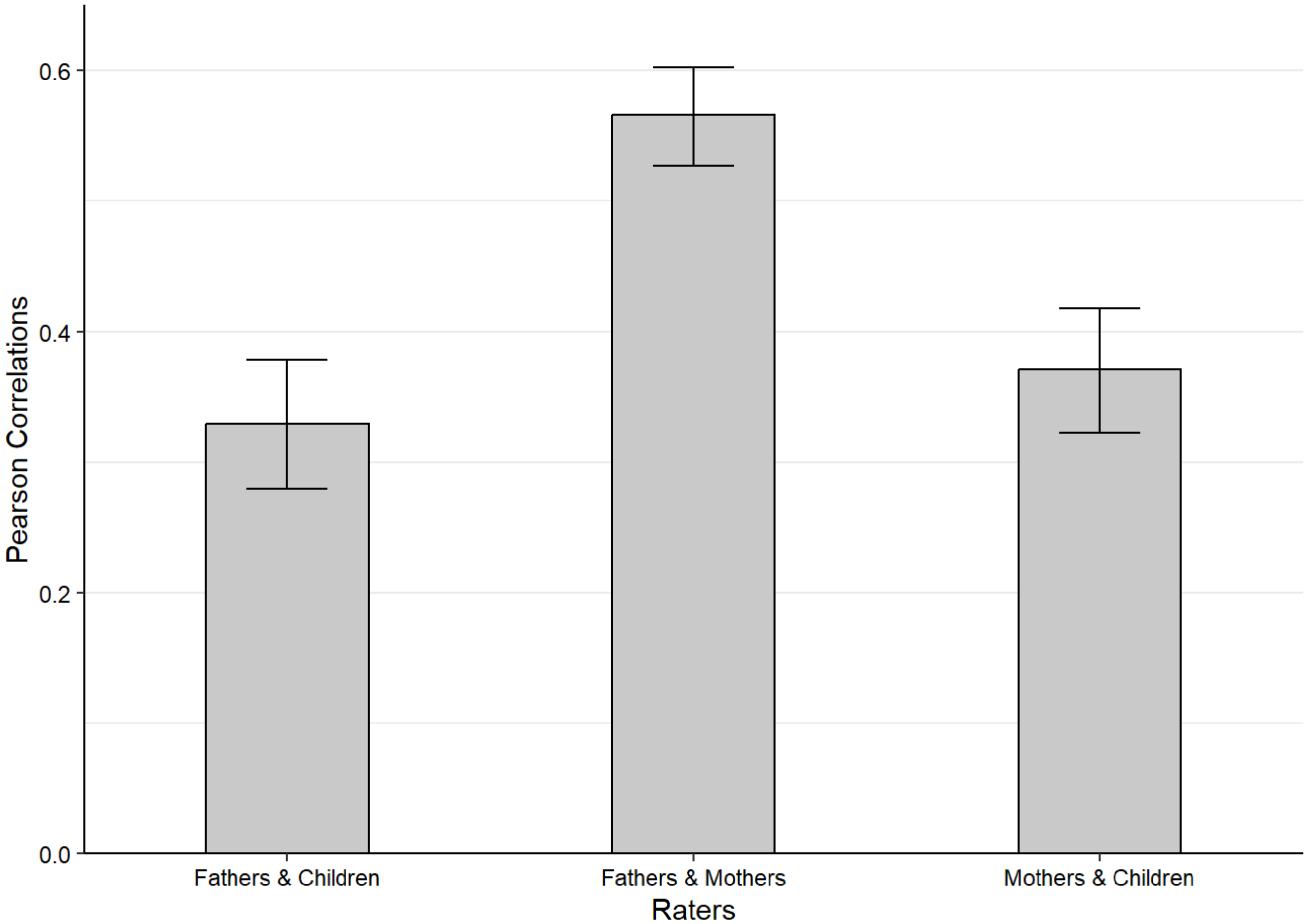
```
  mutate(
        rater_x = case_when(
            rater_x == "int_scale_f" ~ "Father",
            rater_x == "int_scale_m" ~ "Mother",
            rater_x == "int_scale_c" ~ "Child"
        ))

y_df <- x_df %>%
  rename(rater_y = "rater_x", val_y = "val_x")

xy_df <- full_join(x_df, y_df,
          by = c("ID"),
          relationship = "many-to-many")

xy_df %>%
  unite("combo", rater_x, rater_y, sep = " & ") %>%
  filter(combo %in% c("Father & Mother", "Father & Child", "Mother & Child")) %>%
  ggplot(aes(x = val_x, y = val_y)) +
  geom_point(position = "jitter") +
  geom_smooth(method = lm, se = FALSE, linewidth = 1.3) +
  facet_wrap(vars(combo)) +
  theme_minimal() +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank()
  ) +
  labs(title = "Relationship Between Raters and Their Correlations for Internalizing Symptoms")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



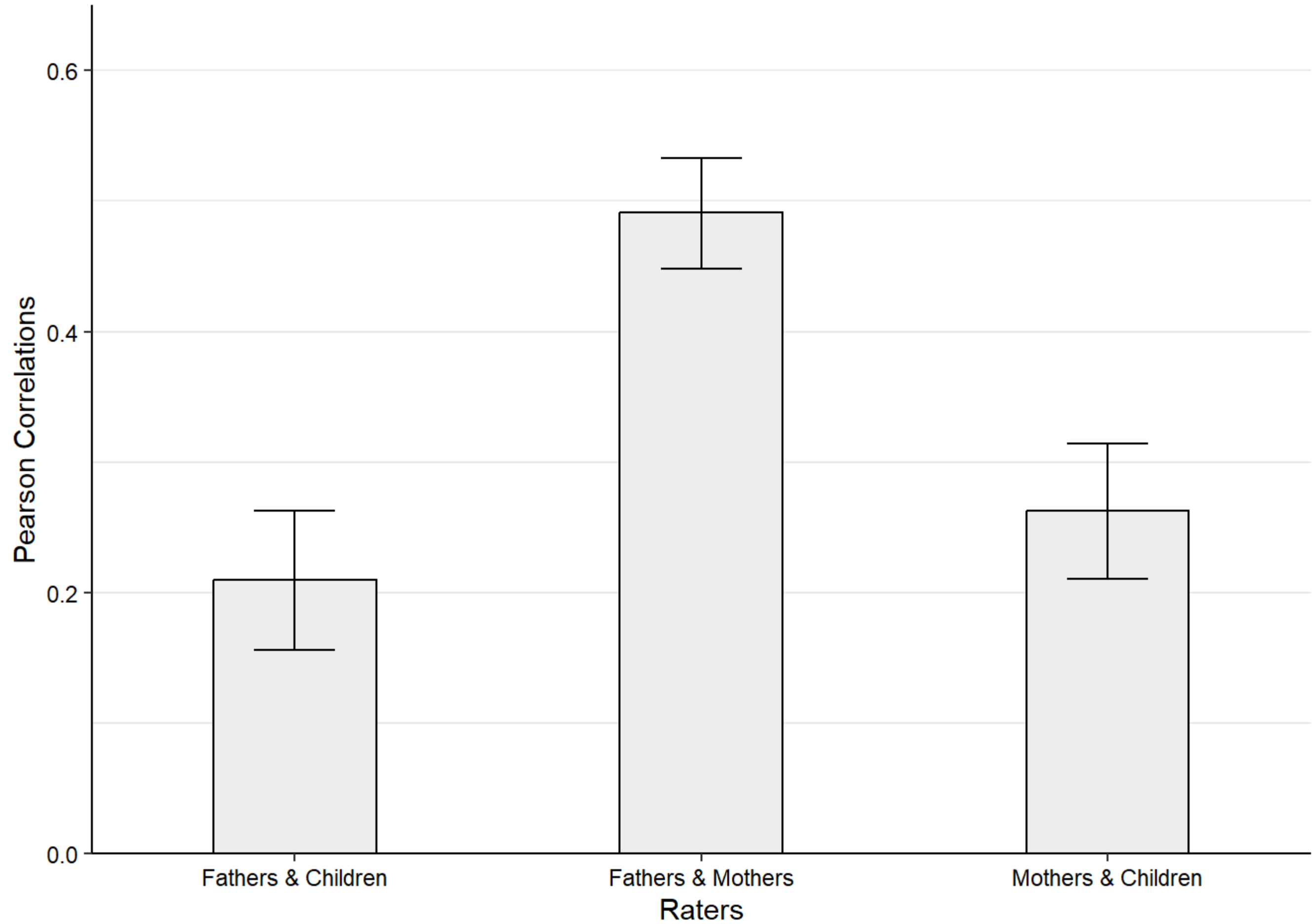
Making a plot for the thesis with the Pearson correlation for externalizing symptoms:

```
data_df_long_unique %>%
  select(Fathers = ext_scale_f, Mothers = ext_scale_m, Children = ext_scale_c) %>%
  correlation() %>%
  as_tibble() %>%
  mutate(
    raters = paste(Parameter1, Parameter2, sep = " & ")
  ) %>%
  ggplot(aes(x = raters, y = r)) +
  geom_bar(stat = "identity", fill = "grey79", color = "black", width = 0.4) +
  geom_errorbar(aes(ymin = CI_low, ymax = CI_high), width = 0.20) +
  ylim(0, 0.65) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line()) +
  coord_cartesian(expand = FALSE, xlim = c(0.5, 3.5), ylim = c(0, 0.65)) +
  labs(
    x = "Raters",
    y = "Pearson Correlations"
  )
```



Making a plot for the thesis with the Pearson correlation for internalizing symptoms:

```
data_df_long_unique %>%
  select(Fathers = int_scale_f, Mothers = int_scale_m, Children = int_scale_c) %>%
  correlation() %>%
  as_tibble() %>%
  mutate(
    raters = paste(Parameter1, Parameter2, sep = " & ")
  ) %>%
  ggplot(aes(x = raters, y = r)) +
  geom_bar(stat = "identity", fill = "grey93", color = "black", width = 0.4) +
  geom_errorbar(aes(ymin = CI_low, ymax = CI_high), width = 0.20) +
  ylim(0, 0.65) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line()) +
  coord_cartesian(expand = FALSE, xlim = c(0.5, 3.5), ylim = c(0, 0.65)) +
  labs(
    x = "Raters",
    y = "Pearson Correlations"
  )
```



7.2 - At each Age

Calculating Pearson correlation for externalizing symptoms at each age:

```
cor_df_ext_temp <- data_df_long %>%
  select(fathers = ext_scale_f, mothers = ext_scale_m, children = ext_scale_c, Age_cat) %>%
  group_by(Age_cat) %>%
  summarize(correlation_matrix = list(correlate(across(everything()),
                                              method = "pearson",
                                              use = "pairwise.complete.obs")),.groups = "drop")
```

```
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
```

```
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
```

```r
conv_cor_matrix <- function(i, df) {
  df[[2]][[i]] %>%
  pivot_wider(names_from = term, values_from = c(fathers, mothers, children)) %>%
  select(fathers_mothers, fathers_children, mothers_children)}

cor_df_ext <- do.call(rbind, lapply(1:12, conv_cor_matrix, cor_df_ext_temp))

# Creating a tibble with the age_cat levels that can be used to add to the data frames we create
with stats later:
age_cat_vect <- cor_df_ext_temp$Age_cat %>% as.vector()

cor_df_ext <- cor_df_ext %>%
  mutate(Age_cat = age_cat_vect) %>%
  filter(!is.na(Age_cat))

cor_df_ext$Age_cat <- cor_df_ext$Age_cat %>% as_factor()

print(cor_df_ext)
```

```
# A tibble: 11 × 4
   fathers_mothers fathers_children mothers_children Age_cat
             <dbl>            <dbl>            <dbl> <fct>
 1           0.521            0.246            0.259 7-8
 2           0.480            0.267            0.274 9
 3           0.545            0.300            0.348 10
 4           0.464            0.300            0.291 11
 5           0.542            0.295            0.382 12
 6           0.595            0.413            0.361 13
 7           0.502            0.303            0.311 14
```

```
 8          0.591           0.417           0.383 15
 9          0.570           0.295           0.366 16
10          0.569           0.312           0.383 17
11          0.635           0.257           0.260 18-25
```

Doing the same with correlation() to get confidence intervals:

```
cor_df_ext_temp <- data_df_long %>%
  select(Fathers = ext_scale_f, Mothers = ext_scale_m, Children = ext_scale_c, Age_cat) %>%
  group_by(Age_cat) %>%
  summarize(correlation_matrix = list(correlation(across(everything()),)),.groups = "drop")

conv_cor_matrix_ci <- function(i, df) {
  df[[2]][[i]] %>%
  select(Parameter1, Parameter2, r, CI_low, CI_high)}

cor_df_ext_ci <- do.call(rbind, lapply(1:12, conv_cor_matrix_ci, cor_df_ext_temp))

# Creating a vector with the age_cat levels that can be used to add to the data frames we create
with stats later:
age_cat_3_vect <- cor_df_ext_temp$Age_cat %>% as.tibble %>% uncount(3) %>% as.vector()
```

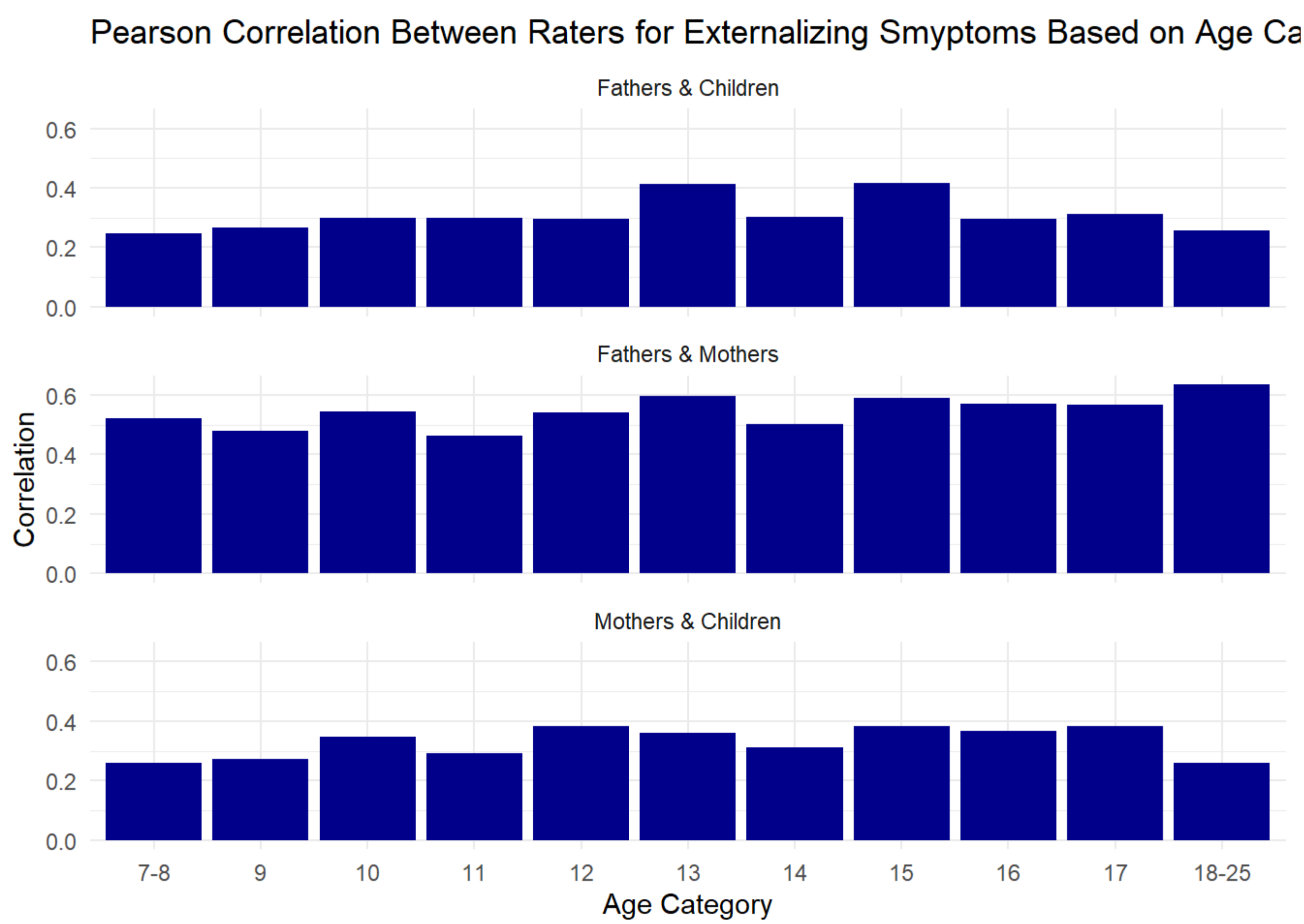
```
Warning: `as.tibble()` was deprecated in tibble 2.0.0.
ℹ Please use `as_tibble()` instead.
ℹ The signature and semantics have changed, see `?as_tibble`.
```

```
cor_df_ext_ci <- cor_df_ext_ci %>% as_tibble() %>%
  bind_cols(age_cat_3_vect) %>%
  rename(Age_cat = "value") %>%
  filter(!is.na(Age_cat))

cor_df_ext_ci$Age_cat <- cor_df_ext_ci$Age_cat %>% as_factor()

print(cor_df_ext_ci)
```

```
# A tibble: 33 × 6
   Parameter1 Parameter2     r CI_low CI_high Age_cat
   <chr>      <chr>      <dbl>  <dbl>   <dbl> <fct>
 1 Fathers    Mothers    0.521  0.454   0.583 7-8
 2 Fathers    Children   0.246  0.162   0.327 7-8
 3 Mothers    Children   0.259  0.182   0.333 7-8
 4 Fathers    Mothers    0.480  0.422   0.533 9
 5 Fathers    Children   0.267  0.199   0.332 9
 6 Mothers    Children   0.274  0.213   0.332 9
 7 Fathers    Mothers    0.545  0.494   0.592 10
 8 Fathers    Children   0.300  0.236   0.361 10
 9 Mothers    Children   0.348  0.294   0.401 10
10 Fathers    Mothers    0.464  0.394   0.529 11
# ℹ 23 more rows
```

Plotting the Pearson correlation between raters for externalizing symptoms across ages:

```
cor_df_ext %>%
  pivot_longer(cols = c(fathers_mothers, fathers_children, mothers_children), names_to = "raters",
values_to = "correlation") %>%
  mutate(raters = case_when(
    raters == "fathers_mothers" ~ "Fathers & Mothers",
    raters == "fathers_children" ~ "Fathers & Children",
    raters == "mothers_children" ~ "Mothers & Children"
  )) %>%
  ggplot(aes(x = Age_cat, y = correlation)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  facet_wrap(vars(raters), nrow = 3) +
  theme_minimal() +
  labs(title = "Pearson Correlation Between Raters for Externalizing Smyptoms Based on Age
Category",
       x = "Age Category",
       y = "Correlation")
```



Calculating Pearson correlation for internalizing symptoms at each age:

```
cor_df_int_temp <- data_df_long %>%
  select(fathers = int_scale_f, mothers = int_scale_m, children = int_scale_c, Age_cat) %>%
  group_by(Age_cat) %>%
  summarize(correlation_matrix = list(correlate(across(everything()),
                                              method = "pearson",
                                              use = "pairwise.complete.obs")),.groups = "drop")
```

```
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
```

```r
conv_cor_matrix <- function(i, df) {
  df[[2]][[i]] %>%
  pivot_wider(names_from = term, values_from = c(fathers, mothers, children)) %>%
  select(fathers_mothers, fathers_children, mothers_children)}

cor_df_int <- do.call(rbind, lapply(1:12, conv_cor_matrix, cor_df_int_temp))

cor_df_int <- cor_df_int %>%
  mutate(Age_cat = age_cat_vect) %>%
  filter(!is.na(Age_cat))

cor_df_int$Age_cat <- cor_df_int$Age_cat %>% as_factor()

print(cor_df_int)
```

```
# A tibble: 11 × 4
   fathers_mothers fathers_children mothers_children Age_cat
             <dbl>            <dbl>            <dbl> <fct>
 1           0.402            0.236            0.141 7-8
 2           0.419            0.247            0.227 9
 3           0.424            0.193            0.263 10
 4           0.429            0.133            0.247 11
 5           0.448            0.275            0.299 12
 6           0.408            0.290            0.229 13
 7           0.538            0.266            0.357 14
 8           0.455            0.252            0.287 15
 9           0.554            0.292            0.361 16
10           0.467            0.248            0.344 17
11           0.536            0.376            0.586 18-25
```

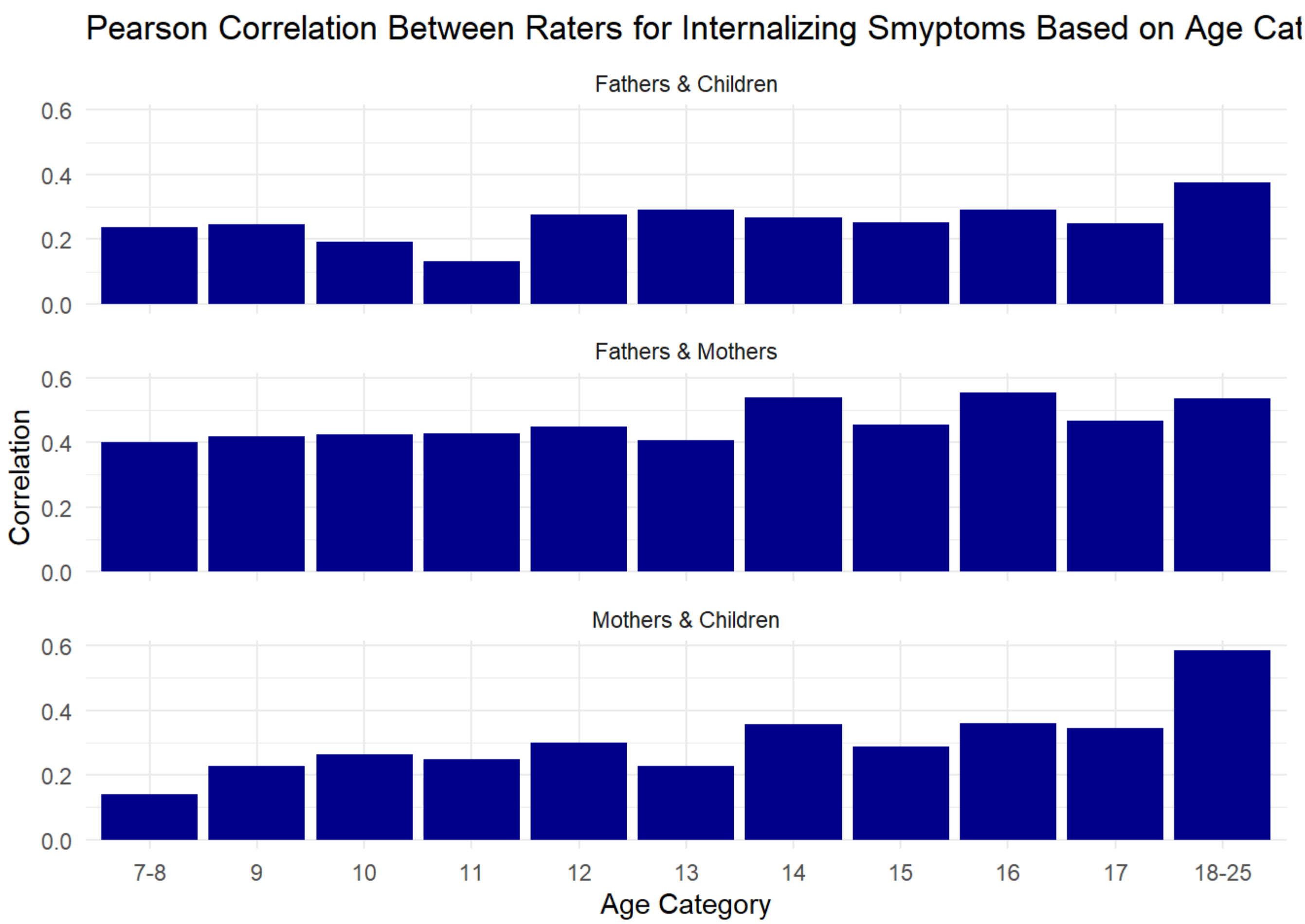
Doing the same with correlation() to get confidence intervals:

```
cor_df_int_temp <- data_df_long %>%
  select(Fathers = int_scale_f, Mothers = int_scale_m, Children = int_scale_c, Age_cat) %>%
  group_by(Age_cat) %>%
  summarize(correlation_matrix = list(correlation(across(everything()),)),.groups = "drop")

conv_cor_matrix_ci <- function(i, df) {
  df[[2]][[i]] %>%
  select(Parameter1, Parameter2, r, CI_low, CI_high)}

cor_df_int_ci <- do.call(rbind, lapply(1:12, conv_cor_matrix_ci, cor_df_int_temp))

cor_df_int_ci <- cor_df_int_ci %>% as_tibble() %>%
  bind_cols(age_cat_3_vect) %>%
  rename(Age_cat = "value") %>%
  filter(!is.na(Age_cat))

cor_df_int_ci$Age_cat <- cor_df_int_ci$Age_cat %>% as_factor()

print(cor_df_int_ci)
```

```
# A tibble: 33 × 6
   Parameter1 Parameter2     r CI_low CI_high Age_cat
   <chr>      <chr>      <dbl>  <dbl>   <dbl> <fct>
 1 Fathers    Mothers    0.402 0.325    0.474 7-8
 2 Fathers    Children   0.236 0.152    0.317 7-8
 3 Mothers    Children   0.141 0.0614   0.219 7-8
 4 Fathers    Mothers    0.419 0.358    0.476 9
 5 Fathers    Children   0.247 0.179    0.313 9
 6 Mothers    Children   0.227 0.166    0.287 9
 7 Fathers    Mothers    0.424 0.366    0.479 10
 8 Fathers    Children   0.193 0.126    0.259 10
 9 Mothers    Children   0.263 0.205    0.319 10
10 Fathers    Mothers    0.429 0.356    0.496 11
# ℹ 23 more rows
```

Plotting the Pearson correlation between raters for internalizing symptoms across ages:

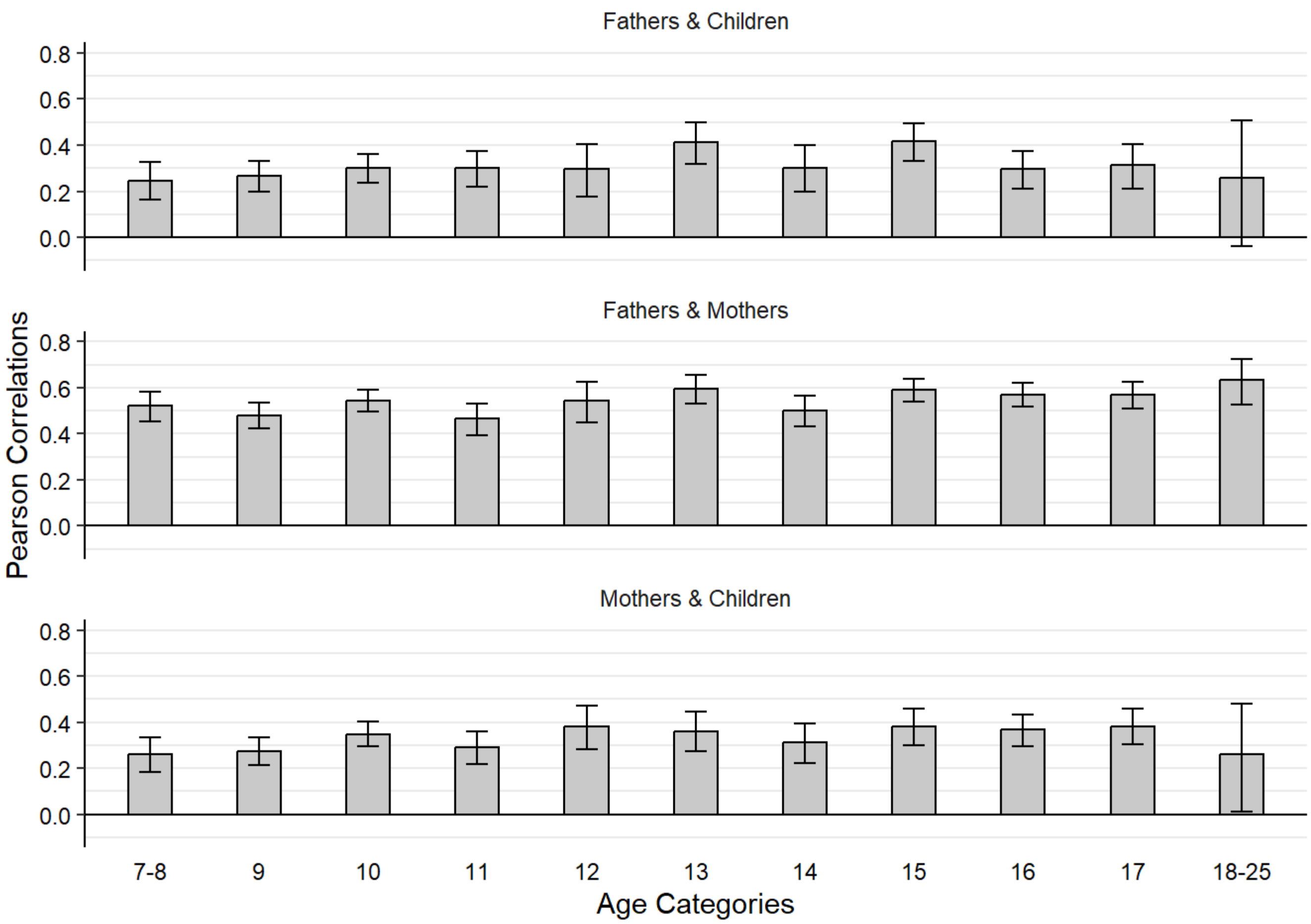
```
cor_df_int %>%
  pivot_longer(cols = c(fathers_mothers, fathers_children, mothers_children), names_to = "raters",
values_to = "correlation") %>%
  mutate(raters = case_when(
    raters == "fathers_mothers" ~ "Fathers & Mothers",
    raters == "fathers_children" ~ "Fathers & Children",
    raters == "mothers_children" ~ "Mothers & Children"
  )) %>%
  ggplot(aes(x = Age_cat, y = correlation)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  facet_wrap(vars(raters), nrow = 3) +
  theme_minimal() +
  labs(title = "Pearson Correlation Between Raters for Internalizing Smyptoms Based on Age
Category",
       x = "Age Category",
       y = "Correlation")
```



Making a thesis plor of the Pearson correlation between raters for externalizing symptoms across ages:

```
cor_df_ext_ci %>%
  mutate(
    raters = paste(Parameter1, Parameter2, sep = " & ")
  ) %>%
  ggplot(aes(x = Age_cat, y = r)) +
```

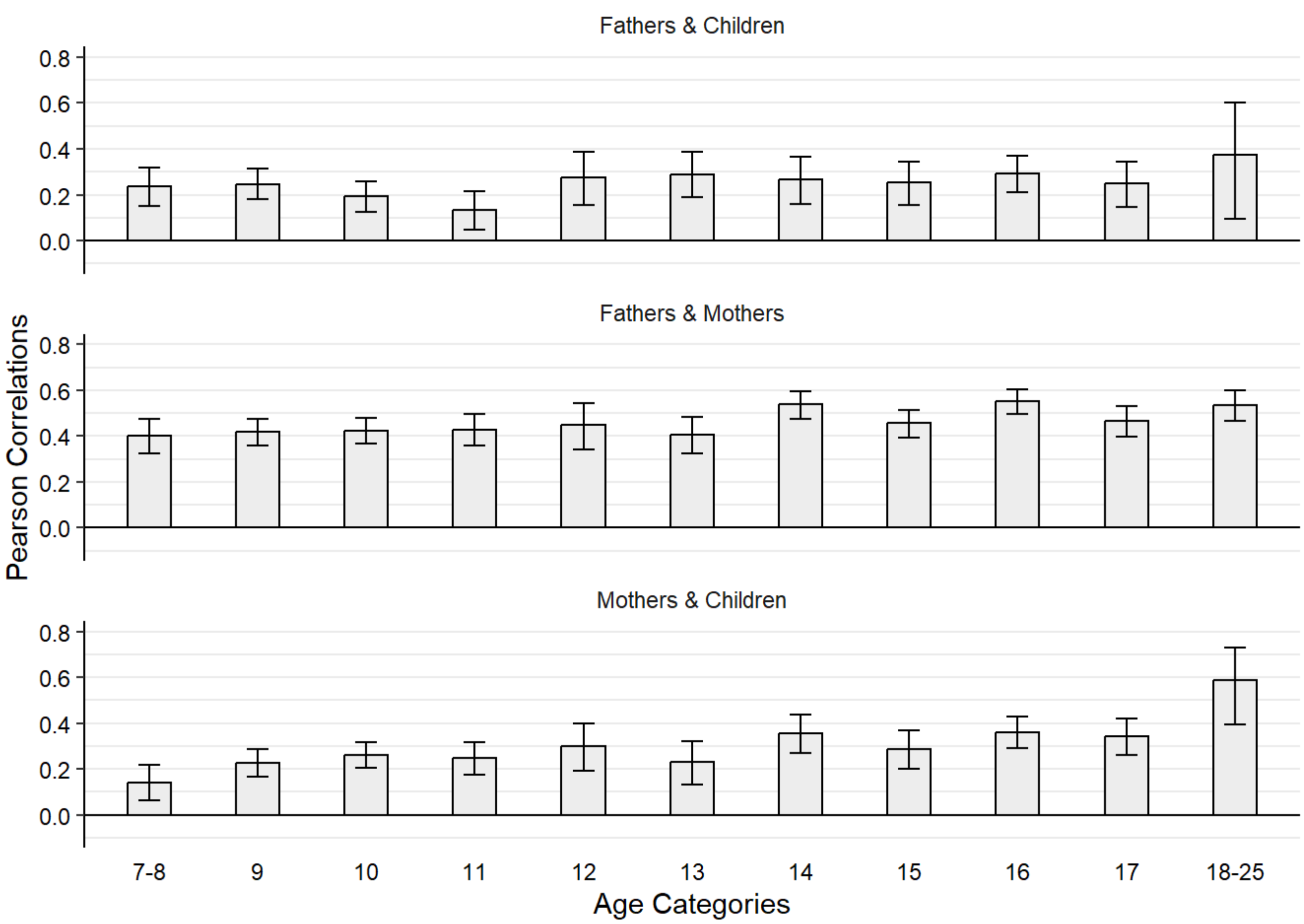
```
  geom_bar(stat = "identity", color = "black", fill = "grey79", width = 0.4) +
  geom_errorbar(aes(ymin = CI_low, ymax = CI_high), width = 0.20) +
  facet_wrap(vars(raters), nrow = 3) +
  ylim(-0.1, 0.8) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  geom_hline(aes(yintercept = 0)) +
  labs(x = "Age Categories",
       y = "Pearson Correlations")
```



Making a thesis plor of the Pearson correlation between raters for externalizing symptoms across ages:

```
cor_df_int_ci %>%
  mutate(
    raters = paste(Parameter1, Parameter2, sep = " & ")
  ) %>%
  ggplot(aes(x = Age_cat, y = r)) +
  geom_bar(stat = "identity", color = "black", fill = "grey93", width = 0.4) +
  geom_errorbar(aes(ymin = CI_low, ymax = CI_high), width = 0.20) +
```

```
  facet_wrap(vars(raters), nrow = 3) +
  ylim(-0.1, 0.8) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  geom_hline(aes(yintercept = 0)) +
  labs(x = "Age Categories",
       y = "Pearson Correlations")
```



7.3 - Adult Attachment

Calculating Pearson Correlation for externalizing symptoms by adult attachment for fathers and mothers separately, then combining it:

```
# Calculating Pearson correlation and creating a data frame for relationship with fathers
cor_df_ext_f_temp <- data_df_wide_diff %>%
  select(Fathers = y8ext_scale_f, Children = y8ext_scale_c, y10aaa_scale_ya_f_dich) %>%
  group_by(y10aaa_scale_ya_f_dich) %>%
  summarize(correlation_matrix = list(correlation(across(everything()),)),.groups = "drop")
```

```
conv_cor_matrix_ci <- function(i, df) {
  df[[2]][[i]] %>%
  select(Parameter1, Parameter2, r, CI_low, CI_high)}

cor_df_ext_f_ci <- do.call(rbind, lapply(1:3, conv_cor_matrix_ci, cor_df_ext_f_temp))

# Creating a vector with the age_cat levels that can be used to add to the data frames we create
with stats later:
attachment_vect <- cor_df_ext_f_temp$y10aaa_scale_ya_f_dich  %>% as.tibble %>% as.vector()

cor_df_ext_f_ci <- cor_df_ext_f_ci %>% as_tibble() %>%
  bind_cols(attachment_vect) %>%
  rename(Attachment = "value") %>%
  filter(!is.na(Attachment))


# Calculating Pearson correlation and creating a data frame for relationship with mothers
cor_df_ext_m_temp <- data_df_wide_diff %>%
  select(Mothers = y8ext_scale_m, Children = y8ext_scale_c, y10aaa_scale_ya_m_dich) %>%
  group_by(y10aaa_scale_ya_m_dich) %>%
  summarize(correlation_matrix = list(correlation(across(everything()),)),.groups = "drop")

conv_cor_matrix_ci <- function(i, df) {
  df[[2]][[i]] %>%
  select(Parameter1, Parameter2, r, CI_low, CI_high)}

cor_df_ext_m_ci <- do.call(rbind, lapply(1:3, conv_cor_matrix_ci, cor_df_ext_m_temp))

# Creating a vector with the age_cat levels that can be used to add to the data frames we create
with stats later:
attachment_vect <- cor_df_ext_m_temp$y10aaa_scale_ya_m_dich  %>% as.tibble %>% as.vector()

cor_df_ext_m_ci <- cor_df_ext_m_ci %>% as_tibble() %>%
  bind_cols(attachment_vect) %>%
  rename(Attachment = "value") %>%
  filter(!is.na(Attachment))

cor_df_ext_all_ci <- cor_df_ext_f_ci %>%
  rbind(cor_df_ext_m_ci)

cor_df_ext_all_ci$Attachment <- cor_df_ext_all_ci$Attachment %>% as_factor()

cor_df_ext_all_ci
```

```
# A tibble: 4 × 6
  Parameter1 Parameter2     r CI_low CI_high Attachment
  <chr>      <chr>      <dbl>  <dbl>   <dbl> <fct>
1 Fathers    Children   0.342  0.157   0.504 High
2 Fathers    Children   0.480  0.328   0.607 Low
3 Mothers    Children   0.370  0.220   0.502 High
4 Mothers    Children   0.435  0.286   0.563 Low
```

Calculating Pearson Correlation for internalizing symptoms by adult attachment for fathers and mothers separately, then combining it:

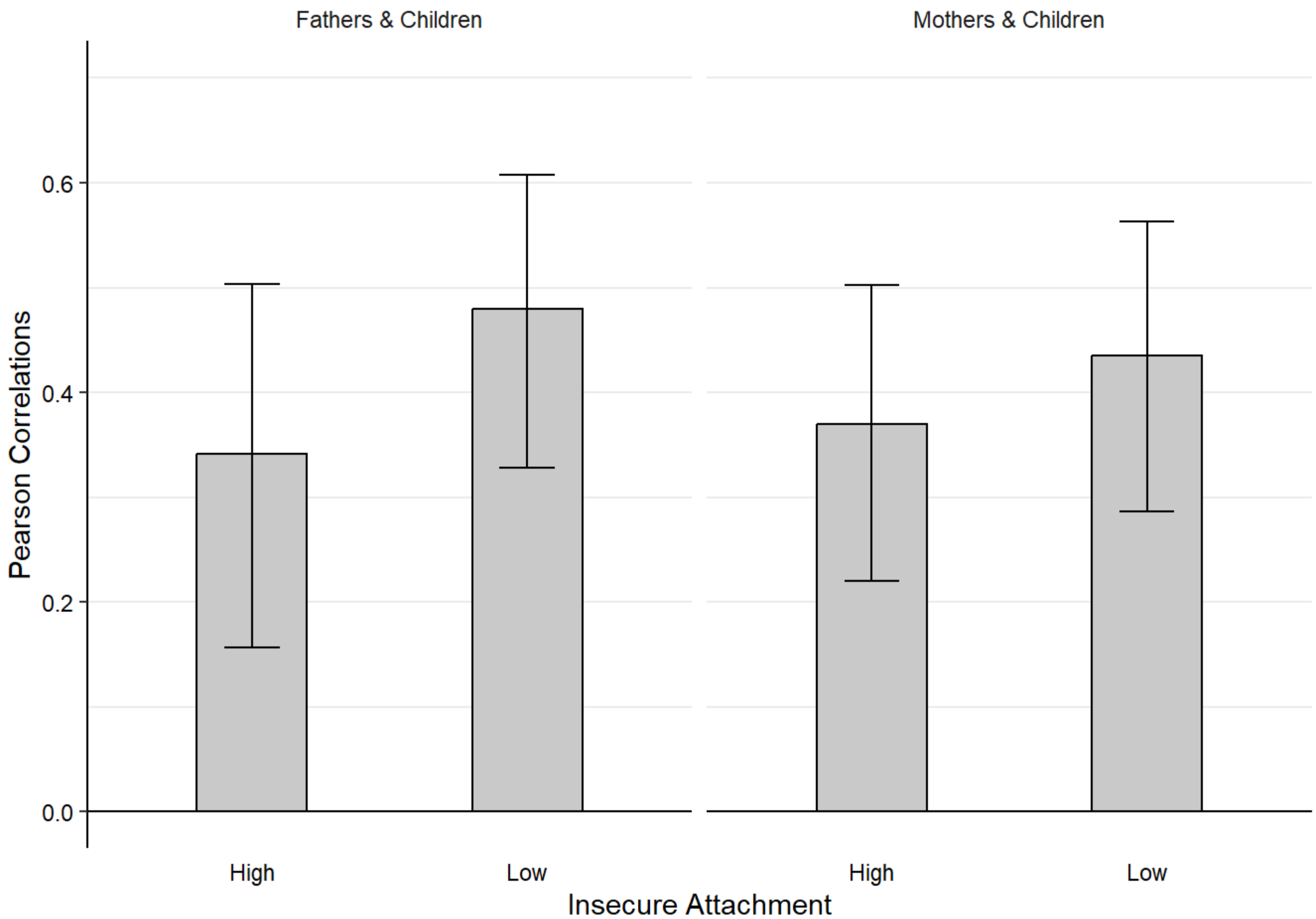
```
# Calculating Pearson correlation and creating a data frame for relationship with fathers
cor_df_int_f_temp <- data_df_wide_diff %>%
  select(Fathers = y8int_scale_f, Children = y8int_scale_c, y10aaa_scale_ya_f_dich) %>%
  group_by(y10aaa_scale_ya_f_dich) %>%
  summarize(correlation_matrix = list(correlation(across(everything()),)),.groups = "drop")

conv_cor_matrix_ci <- function(i, df) {
  df[[2]][[i]] %>%
  select(Parameter1, Parameter2, r, CI_low, CI_high)}

cor_df_int_f_ci <- do.call(rbind, lapply(1:3, conv_cor_matrix_ci, cor_df_int_f_temp))

# Creating a vector with the age_cat levels that can be used to add to the data frames we create
with stats later:
attachment_vect <- cor_df_int_f_temp$y10aaa_scale_ya_f_dich  %>% as.tibble %>% as.vector()

cor_df_int_f_ci <- cor_df_int_f_ci %>% as_tibble() %>%
  bind_cols(attachment_vect) %>%
  rename(Attachment = "value") %>%
  filter(!is.na(Attachment))


# Calculating Pearson correlation and creating a data frame for relationship with mothers
cor_df_int_m_temp <- data_df_wide_diff %>%
  select(Mothers = y8int_scale_m, Children = y8int_scale_c, y10aaa_scale_ya_m_dich) %>%
  group_by(y10aaa_scale_ya_m_dich) %>%
  summarize(correlation_matrix = list(correlation(across(everything()),)),.groups = "drop")

conv_cor_matrix_ci <- function(i, df) {
  df[[2]][[i]] %>%
  select(Parameter1, Parameter2, r, CI_low, CI_high)}

cor_df_int_m_ci <- do.call(rbind, lapply(1:3, conv_cor_matrix_ci, cor_df_int_m_temp))

# Creating a vector with the age_cat levels that can be used to add to the data frames we create
with stats later:
attachment_vect <- cor_df_int_m_temp$y10aaa_scale_ya_m_dich  %>% as.tibble %>% as.vector()

cor_df_int_m_ci <- cor_df_int_m_ci %>% as_tibble() %>%
  bind_cols(attachment_vect) %>%
  rename(Attachment = "value") %>%
  filter(!is.na(Attachment))

cor_df_int_all_ci <- cor_df_int_f_ci %>%
  rbind(cor_df_int_m_ci)

cor_df_int_all_ci$Attachment <- cor_df_int_all_ci$Attachment %>% as_factor()

cor_df_int_all_ci
```

```
# A tibble: 4 × 6
```

```
  Parameter1 Parameter2     r CI_low CI_high Attachment
  <chr>      <chr>      <dbl>  <dbl>   <dbl> <fct>
1 Fathers    Children   0.280 0.0890   0.450 High
2 Fathers    Children   0.435 0.276    0.571 Low
3 Mothers    Children   0.391 0.244    0.521 High
4 Mothers    Children   0.389 0.232    0.526 Low
```

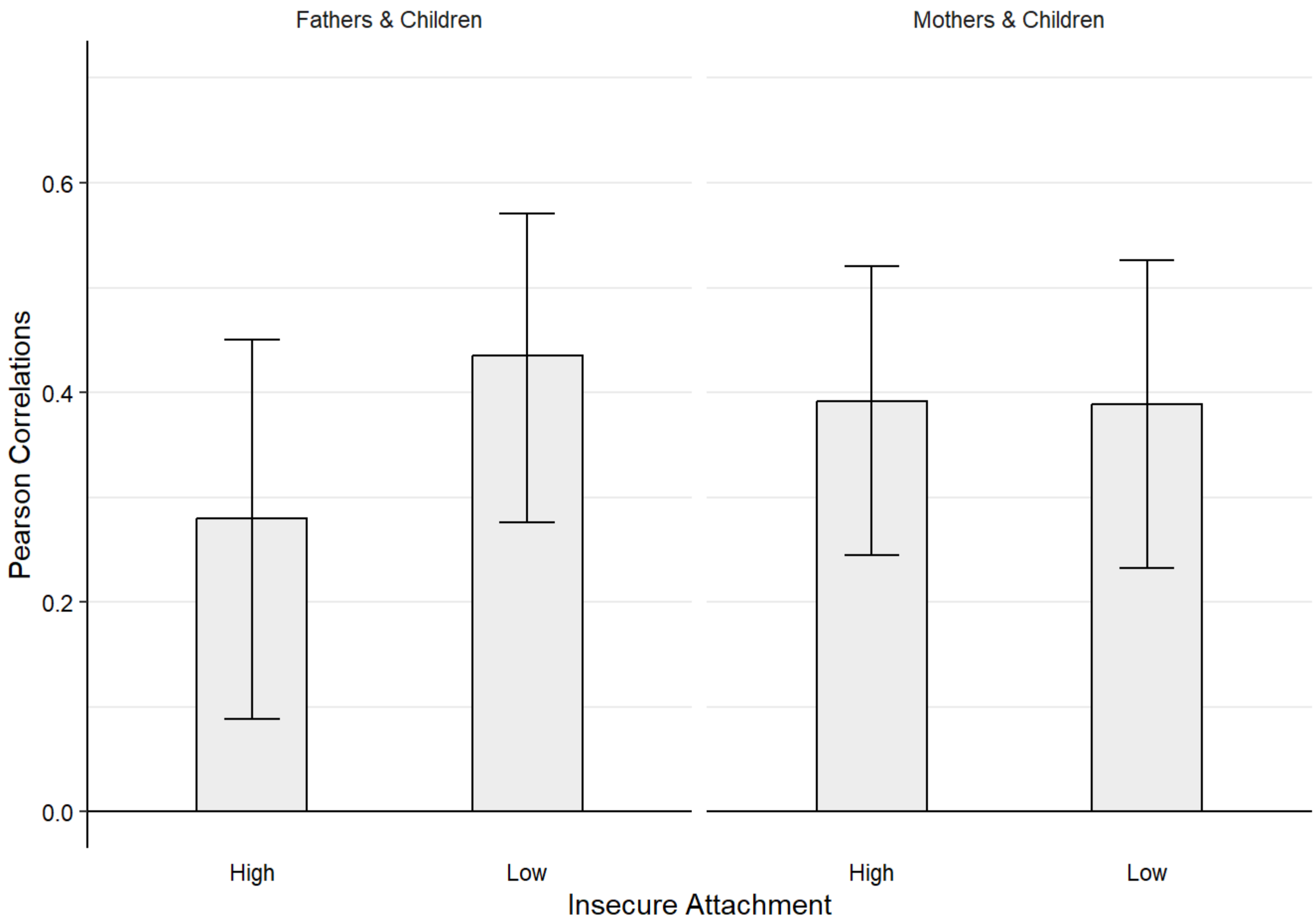
Making a thesis plot of the Pearson correlation between raters for externalizing symptoms across high and low adult attachment:

```
cor_df_ext_all_ci %>%
  mutate(
    raters = paste(Parameter1, Parameter2, sep = " & ")
  ) %>%
  ggplot(aes(x = Attachment, y = r)) +
  geom_bar(stat = "identity", color = "black", fill = "grey79", width = 0.4) +
  geom_errorbar(aes(ymin = CI_low, ymax = CI_high), width = 0.20) +
  facet_wrap(vars(raters)) +
  ylim(0, 0.7) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  geom_hline(aes(yintercept = 0)) +
  labs(x = "Insecure Attachment",
       y = "Pearson Correlations")
```



Making a thesis plot of the Pearson correlation between raters for internalizing symptoms across high and low adult attachment:

```
cor_df_int_all_ci %>%
  mutate(
    raters = paste(Parameter1, Parameter2, sep = " & ")
  ) %>%
  ggplot(aes(x = Attachment, y = r)) +
  geom_bar(stat = "identity", color = "black", fill = "grey93", width = 0.4) +
  geom_errorbar(aes(ymin = CI_low, ymax = CI_high), width = 0.20) +
  facet_wrap(vars(raters)) +
  ylim(0, 0.7) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  geom_hline(aes(yintercept = 0)) +
  labs(x = "Insecure Attachment",
       y = "Pearson Correlations")
```



8 - Intraclass Correlation

8.1 - In general

Calculating intraclass correlation for externalizing symptoms between the raters, comparing dyads, as well as all three raters regardless of age:

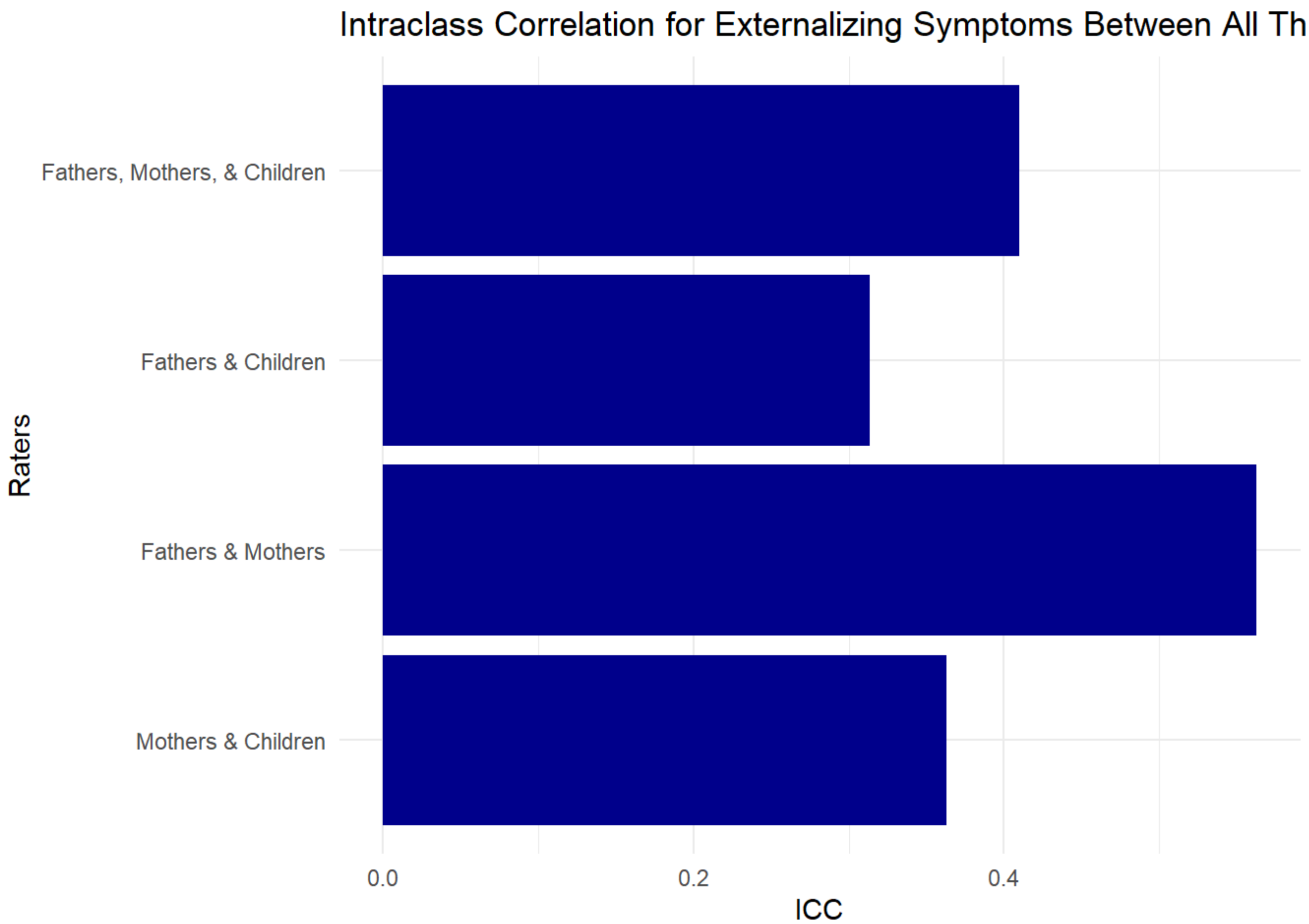
```
calc_icc <- function(cols, data) {
  icc_results <- data %>%
  select(all_of(cols)) %>%
icc(model = "twoway", type = "agreement", unit = "single")

   tibble(
   raters = paste0(cols, collapse = "_"),
   icc_value = icc_results$value,
   ci_lower_boundary = icc_results$lbound,
   ci_upper_boundary = icc_results$ubound
   )
}

# Create a nested vector for running the ICC at with a vector of all three raters in a nested vector
column_comb_ext <- combn(c("ext_scale_f", "ext_scale_c", "ext_scale_m"), 3, simplify = FALSE)
```

```
# Add items for each combination of two raters, based on combining these column names, two at a time
column_comb_ext <- append(column_comb_ext, combn(c("ext_scale_f", "ext_scale_c", "ext_scale_m"), 2, simplify = FALSE))

icc_df_ext <- do.call(rbind, lapply(column_comb_ext, calc_icc, data_df_long_unique))


icc_df_ext
```

```
# A tibble: 4 × 4
  raters                            icc_value ci_lower_boundary ci_upper_boundary
  <chr>                                 <dbl>             <dbl>             <dbl>
1 ext_scale_f_ext_scale_c_ext_sca…      0.410             0.370             0.448
2 ext_scale_f_ext_scale_c               0.314             0.249             0.374
3 ext_scale_f_ext_scale_m               0.563             0.523             0.600
4 ext_scale_c_ext_scale_m               0.363             0.310             0.414
```

Plotting the intraclass correlation for externalizing symptoms for all three raters and dyads:

```
icc_df_ext %>%
  # To rearrange the variable order for the plot
  arrange(-row_number()) %>%
  mutate(raters = case_when(
    raters == "ext_scale_f_ext_scale_c_ext_scale_m" ~ "Fathers, Mothers, & Children",
    raters == "ext_scale_f_ext_scale_m" ~ "Fathers & Mothers",
    raters == "ext_scale_f_ext_scale_c" ~ "Fathers & Children",
    raters == "ext_scale_c_ext_scale_m" ~ "Mothers & Children",
  )) %>%
  ggplot(aes(x = icc_value, y = fct_inorder(raters))) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  theme_minimal() +
  labs(title = "Intraclass Correlation for Externalizing Symptoms Between All Three Raters and Dyads",
       x = "ICC",
       y = "Raters")
```



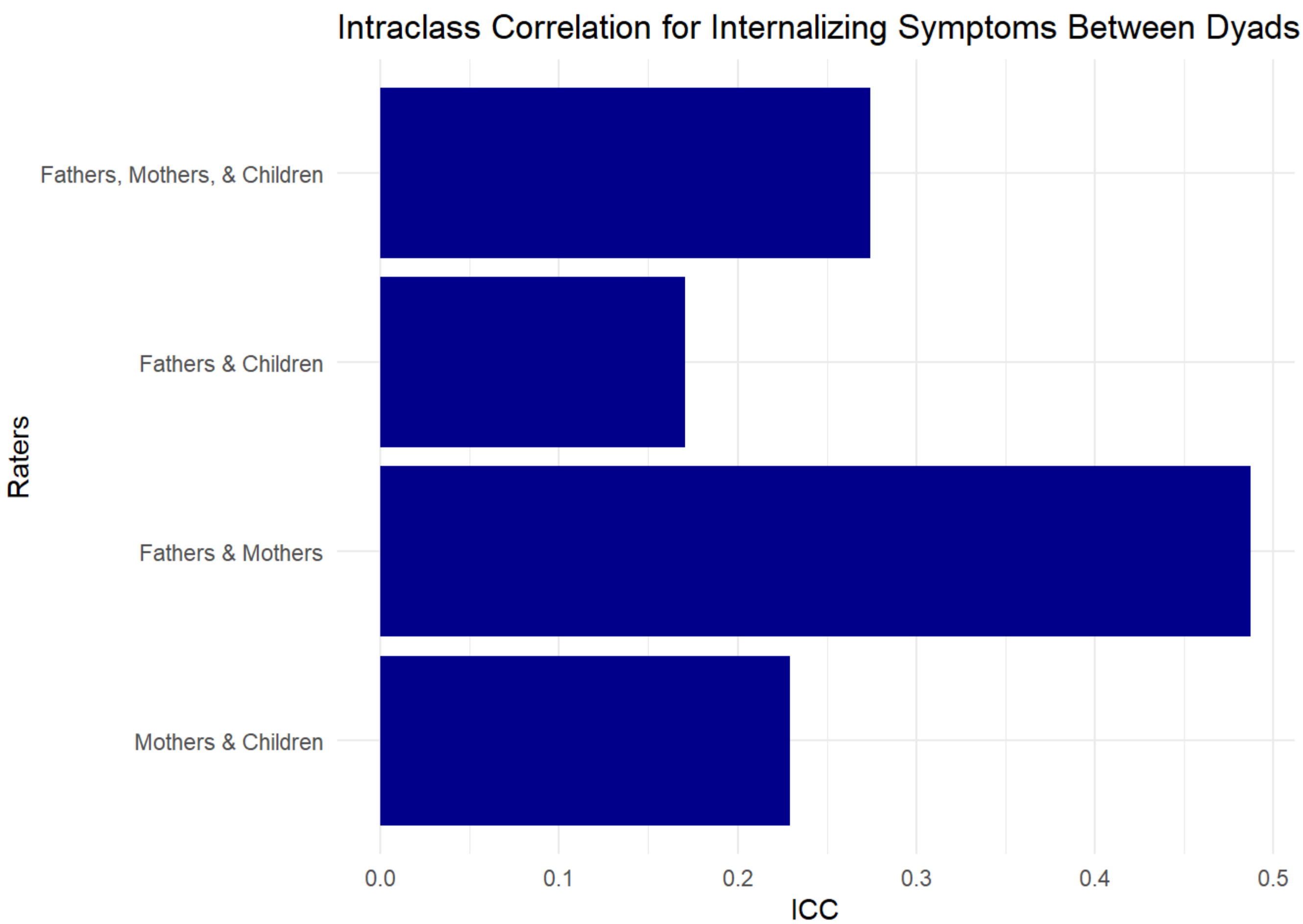
Calculating intraclass correlation for internalizing symptoms between the raters, comparing dyads, regardless of age:

```
# Create a nested vector for running the ICC at with a vector of all three raters in a nested
vector
column_comb_int <- combn(c("int_scale_f", "int_scale_c", "int_scale_m"), 3, simplify = FALSE)
# Add items for each combination of two raters, based on combining these column names, two at a
time
column_comb_int <- append(column_comb_int, combn(c("int_scale_f", "int_scale_c", "int_scale_m"), 2,
simplify = FALSE))

icc_df_int <- do.call(rbind, lapply(column_comb_int, calc_icc, data_df_long_unique))


icc_df_int
```

```
# A tibble: 4 × 4
  raters                            icc_value ci_lower_boundary ci_upper_boundary
  <chr>                                 <dbl>             <dbl>             <dbl>
1 int_scale_f_int_scale_c_int_sca…      0.274            0.206              0.338
2 int_scale_f_int_scale_c               0.171            0.0765             0.258
3 int_scale_f_int_scale_m               0.487            0.443              0.529
4 int_scale_c_int_scale_m               0.229            0.139              0.312
```

Plotting the intraclass correlation for internalizing symptoms for dyads:

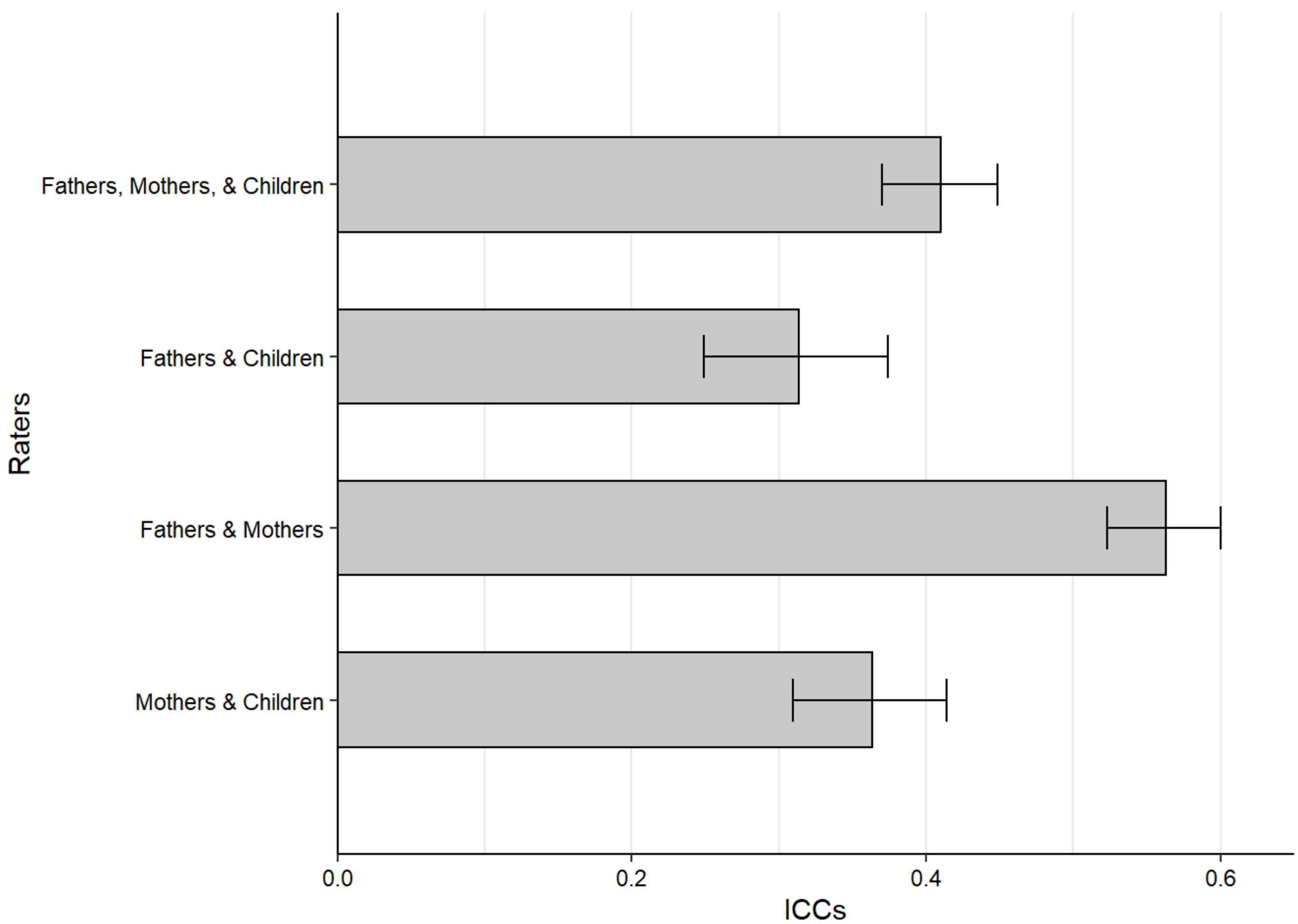
```r
icc_df_int %>%
  # To rearrange the variable order for the plot
  arrange(-row_number()) %>%
  mutate(raters = case_when(
    raters == "int_scale_f_int_scale_c_int_scale_m" ~ "Fathers, Mothers, & Children",
    raters == "int_scale_f_int_scale_m" ~ "Fathers & Mothers",
    raters == "int_scale_f_int_scale_c" ~ "Fathers & Children",
    raters == "int_scale_c_int_scale_m" ~ "Mothers & Children"
  )) %>%
  ggplot(aes(x = icc_value, y = fct_inorder(raters))) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  theme_minimal() +
  labs(title = "Intraclass Correlation for Internalizing Symptoms Between Dyads",
       x = "ICC",
       y = "Raters")
```



Plotting the intraclass correlation for externalizing symptoms for all three raters and dyads for the thesis:

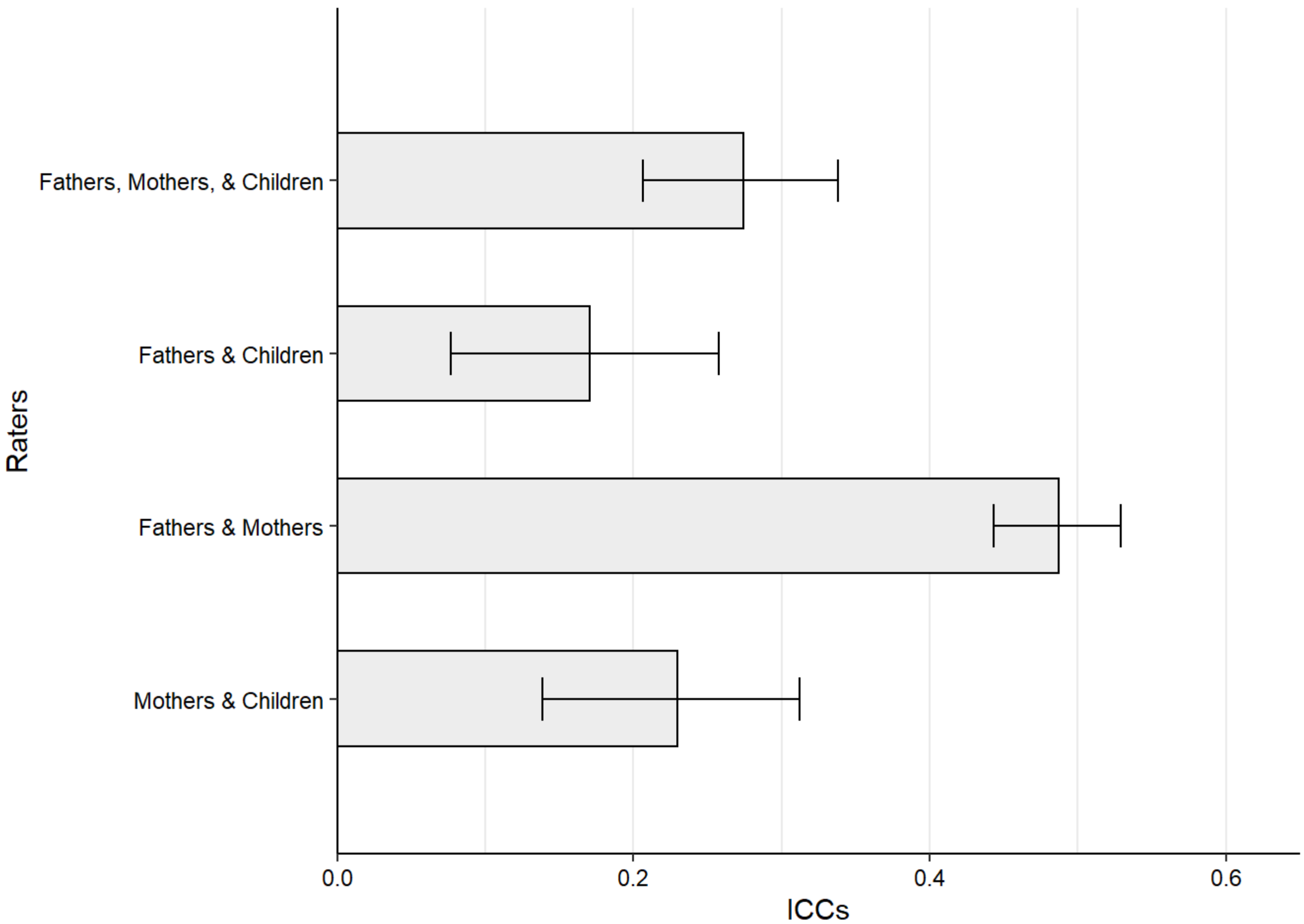
```r
icc_df_ext %>%
  # To rearrange the variable order for the plot
  arrange(-row_number()) %>%
  mutate(raters = case_when(
    raters == "ext_scale_f_ext_scale_c_ext_scale_m" ~ "Fathers, Mothers, & Children",
    raters == "ext_scale_f_ext_scale_m" ~ "Fathers & Mothers",
    raters == "ext_scale_f_ext_scale_c" ~ "Fathers & Children",
```

```
    raters == "ext_scale_c_ext_scale_m" ~ "Mothers & Children",
  )) %>%
  ggplot(aes(x = icc_value, y = fct_inorder(raters))) +
  geom_bar(stat = "identity", fill = "grey79", color = "black", width = 0.55) +
  geom_errorbar(aes(xmin = ci_lower_boundary, xmax = ci_upper_boundary), width = 0.25) +
  xlim(0, 0.60) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        panel.grid.major.x = element_line(),
        panel.grid.minor.x = element_line()) +
 coord_cartesian(expand = FALSE, xlim = c(0, 0.65), ylim = c(0.1, 5)) +
  labs(x = "ICCs",
       y = "Raters")
```




```
icc_df_int %>%
  # To rearrange the variable order for the plot
  arrange(-row_number()) %>%
  mutate(raters = case_when(
    raters == "int_scale_f_int_scale_c_int_scale_m" ~ "Fathers, Mothers, & Children",
    raters == "int_scale_f_int_scale_m" ~ "Fathers & Mothers",
    raters == "int_scale_f_int_scale_c" ~ "Fathers & Children",
    raters == "int_scale_c_int_scale_m" ~ "Mothers & Children",
  )) %>%
```

```
  ggplot(aes(x = icc_value, y = fct_inorder(raters))) +
  geom_bar(stat = "identity", fill = "grey93", color = "black", width = 0.55) +
  geom_errorbar(aes(xmin = ci_lower_boundary, xmax = ci_upper_boundary), width = 0.25) +
  xlim(0, 0.60) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        panel.grid.major.x = element_line(),
        panel.grid.minor.x = element_line()) +
 coord_cartesian(expand = FALSE, xlim = c(0, 0.65), ylim = c(0.1, 5)) +
  labs(x = "ICCs",
       y = "Raters")
```



8.2. - At each Age

Calculating intraclass correlation for externalizing symptoms between the raters at each age:

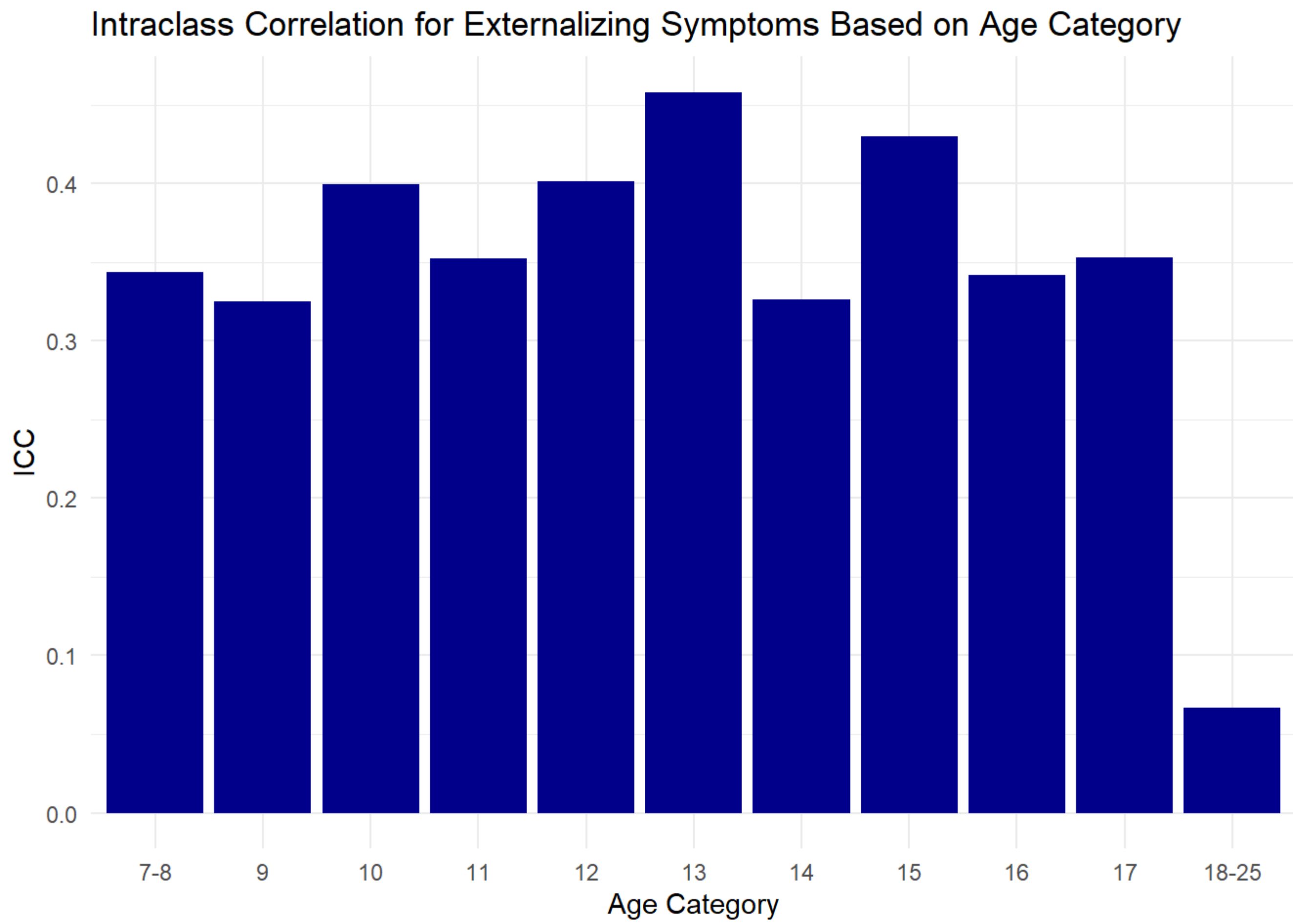
```
calc_icc <- function(cols, data) {
  icc_results <- data %>%
  select(all_of(cols)) %>%
icc(model = "twoway", type = "agreement", unit = "single")

  tibble(
   raters = paste0(cols, collapse = "_"),
```

```
    icc_value = icc_results$value,
    ci_lower_boundary = icc_results$lbound,
    ci_upper_boundary = icc_results$ubound
    )
}

icc_df_ext_age <- data_df_long %>%
  select(ext_scale_f, ext_scale_m, ext_scale_c, Age_cat) %>%
  group_split(Age_cat) %>%
  map(calc_icc, cols = c("ext_scale_f", "ext_scale_m", "ext_scale_c")) %>%
  bind_rows()

icc_df_ext_age <- icc_df_ext_age %>%
  mutate(Age_cat = age_cat_vect) %>%
  filter(!is.na(Age_cat))

icc_df_ext_age$Age_cat <- icc_df_ext_age$Age_cat %>% as_factor()

icc_df_ext_age
```

```
# A tibble: 11 × 5
   raters                  icc_value ci_lower_boundary ci_upper_boundary Age_cat
   <chr>                       <dbl>             <dbl>             <dbl> <fct>
 1 ext_scale_f_ext_scale_…    0.344             0.287             0.401 7-8
 2 ext_scale_f_ext_scale_…    0.325             0.278             0.372 9
 3 ext_scale_f_ext_scale_…    0.399             0.356             0.443 10
 4 ext_scale_f_ext_scale_…    0.352             0.297             0.408 11
 5 ext_scale_f_ext_scale_…    0.401             0.323             0.480 12
 6 ext_scale_f_ext_scale_…    0.458             0.385             0.528 13
 7 ext_scale_f_ext_scale_…    0.327             0.236             0.413 14
 8 ext_scale_f_ext_scale_…    0.430             0.314             0.530 15
 9 ext_scale_f_ext_scale_…    0.342             0.238             0.436 16
10 ext_scale_f_ext_scale_…    0.353             0.262             0.440 17
11 ext_scale_f_ext_scale_…    0.0670           -0.0745            0.252 18-25
```

Plotting the intraclass correlation for externalizing symptoms across ages:

```
icc_df_ext_age %>%
  ggplot(aes(x = Age_cat, y = icc_value)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  theme_minimal() +
  labs(title = "Intraclass Correlation for Externalizing Symptoms Based on Age Category",
       x = "Age Category",
       y = "ICC")
```



Calculating intraclass correlation for internalizing symptoms between the raters at each age:

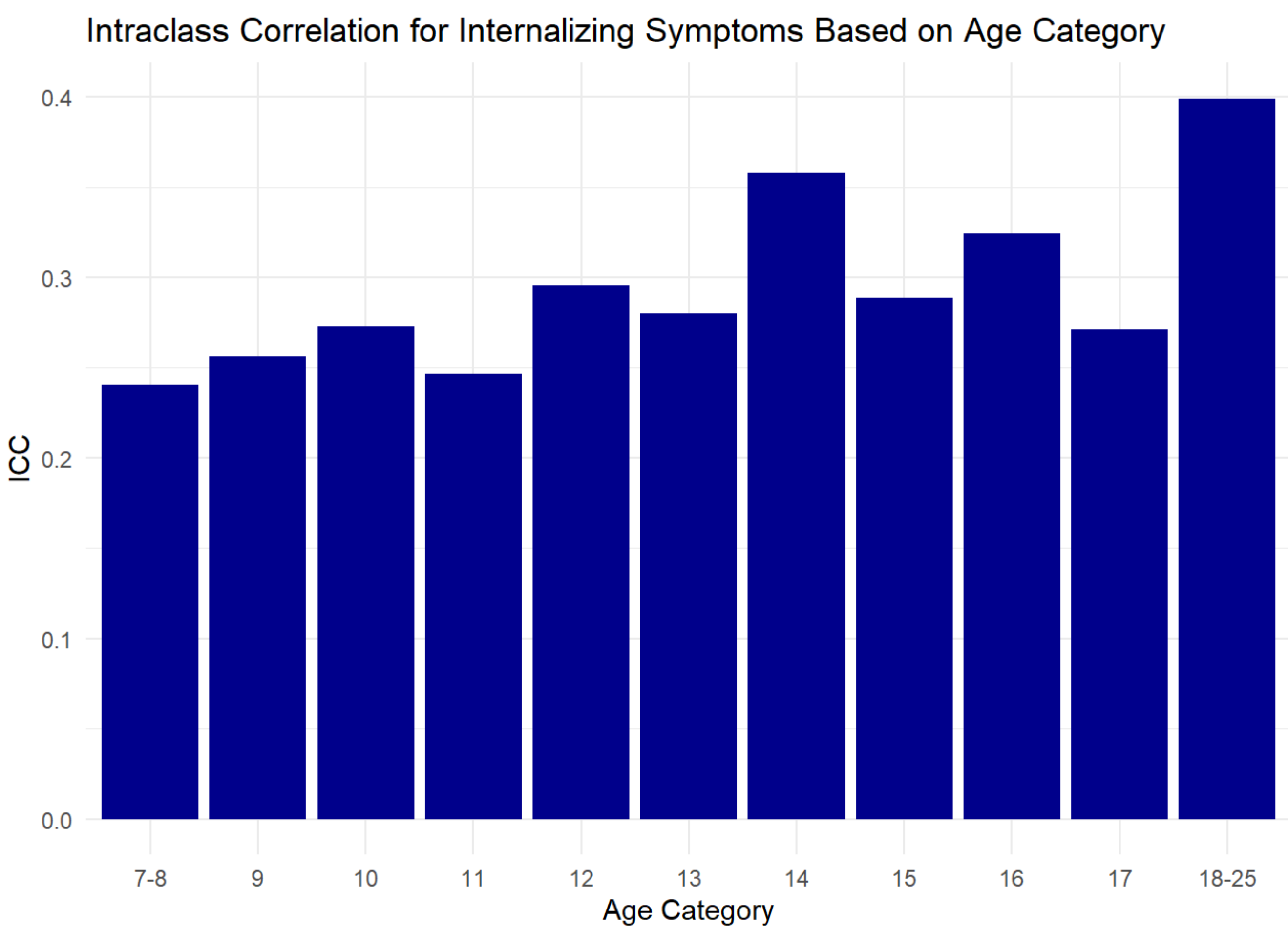
```
icc_df_int_age <- data_df_long %>%
  select(int_scale_f, int_scale_m, int_scale_c, Age_cat) %>%
  group_split(Age_cat) %>%
  map(calc_icc, cols = c("int_scale_f", "int_scale_m", "int_scale_c")) %>%
  bind_rows()

icc_df_int_age <- icc_df_int_age %>%
  mutate(Age_cat = cor_df_ext_temp$Age_cat) %>%
  filter(!is.na(Age_cat))

icc_df_int_age$Age_cat <- icc_df_int_age$Age_cat %>% as_factor()

icc_df_int_age
```

```
# A tibble: 11 × 5
   raters                 icc_value ci_lower_boundary ci_upper_boundary Age_cat
   <chr>                      <dbl>             <dbl>             <dbl> <fct>
 1 int_scale_f_int_scale_…    0.240             0.177             0.304 7-8
 2 int_scale_f_int_scale_…    0.256             0.198             0.314 9
 3 int_scale_f_int_scale_…    0.273             0.219             0.327 10
 4 int_scale_f_int_scale_…    0.247             0.186             0.308 11
 5 int_scale_f_int_scale_…    0.296             0.207             0.385 12
 6 int_scale_f_int_scale_…    0.280             0.202             0.358 13
```

```
 7 int_scale_f_int_scale_…     0.358             0.264             0.446 14
 8 int_scale_f_int_scale_…     0.289             0.187             0.384 15
 9 int_scale_f_int_scale_…     0.324             0.210             0.427 16
10 int_scale_f_int_scale_…     0.272             0.183             0.358 17
11 int_scale_f_int_scale_…     0.399             0.196             0.594 18-25
```

Plotting the intraclass correlation for internalizing symptoms across ages:

```r
icc_df_int_age %>%
  ggplot(aes(x = Age_cat, y = icc_value)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  theme_minimal() +
  labs(title = "Intraclass Correlation for Internalizing Symptoms Based on Age Category",
       x = "Age Category",
       y = "ICC")
```



Calculating intraclass correlation for externalizing symptoms between the raters at each age:

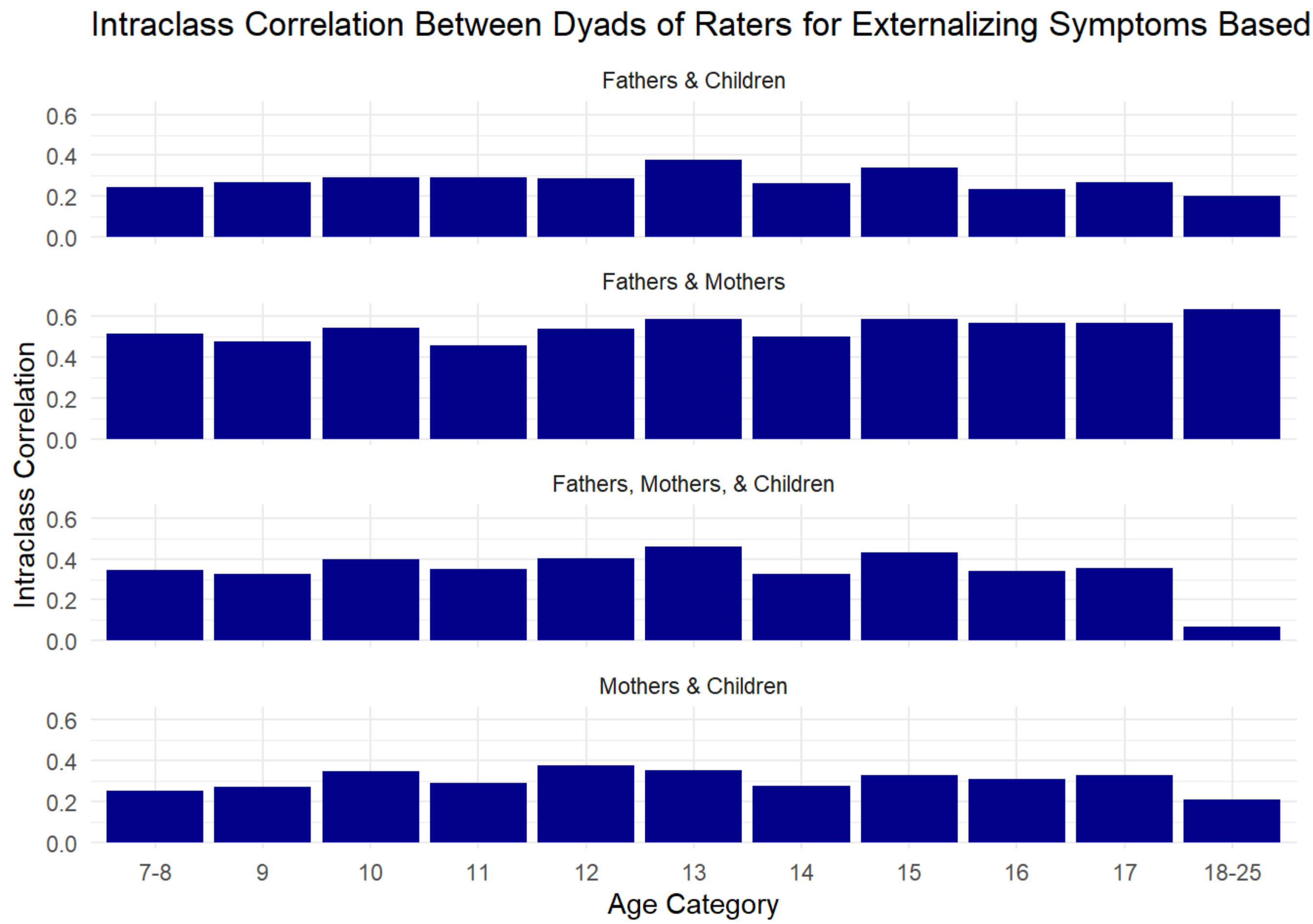
```r
icc_df_ext_age_rater <- data_df_long %>%
  select(ext_scale_f, ext_scale_m, ext_scale_c, Age_cat) %>%
  group_split(Age_cat, .keep = TRUE) %>%
  lapply(function(group_data) {
    # Iterate over column combinations and apply calc_icc for each using do.call
    result_list <- do.call("rbind", lapply(column_comb_ext, function(cols) calc_icc(cols,
group_data)))
```

```
    result_list  # Return the results for this group
  }) %>%
  bind_rows()

age_cat_4_vect <- cor_df_ext_temp$Age_cat %>% as.tibble %>% uncount(4) %>% as.vector()

icc_df_ext_age_rater <- icc_df_ext_age_rater %>%
  bind_cols(age_cat_4_vect) %>%
  rename(Age_cat = "value") %>%
  filter(!is.na(Age_cat))

icc_df_ext_age_rater$Age_cat <- icc_df_ext_age_rater$Age_cat %>% as_factor()

icc_df_ext_age_rater
```

```
# A tibble: 44 × 5
   raters                     icc_value ci_lower_boundary ci_upper_boundary Age_cat
   <chr>                          <dbl>             <dbl>             <dbl> <fct>
 1 ext_scale_f_ext_scale_…        0.344             0.287             0.401 7-8
 2 ext_scale_f_ext_scale_c        0.246             0.162             0.326 7-8
 3 ext_scale_f_ext_scale_m        0.517             0.448             0.579 7-8
 4 ext_scale_c_ext_scale_m        0.255             0.178             0.328 7-8
 5 ext_scale_f_ext_scale_…        0.325             0.278             0.372 9
 6 ext_scale_f_ext_scale_c        0.266             0.198             0.331 9
 7 ext_scale_f_ext_scale_m        0.477             0.420             0.531 9
 8 ext_scale_c_ext_scale_m        0.274             0.214             0.332 9
 9 ext_scale_f_ext_scale_…        0.399             0.356             0.443 10
10 ext_scale_f_ext_scale_c        0.295             0.230             0.356 10
# ℹ 34 more rows
```

Plotting the intraclass correlation between dyads of raters for externalizing symptoms across ages:

```
icc_df_ext_age_rater %>%
  mutate(raters = case_when(
    raters == "ext_scale_f_ext_scale_c_ext_scale_m" ~ "Fathers, Mothers, & Children",
    raters == "ext_scale_f_ext_scale_m" ~ "Fathers & Mothers",
    raters == "ext_scale_f_ext_scale_c" ~ "Fathers & Children",
    raters == "ext_scale_c_ext_scale_m" ~ "Mothers & Children"
  )) %>%
  ggplot(aes(x = Age_cat, y = icc_value)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  facet_wrap(vars(raters), nrow = 4) +
  theme_minimal() +
  labs(title = "Intraclass Correlation Between Dyads of Raters for Externalizing Symptoms Based on
Age Category",
       x = "Age Category",
       y = "Intraclass Correlation")
```



Calculating intraclass correlation for internalizing symptoms between the raters, as dyads, at each age:

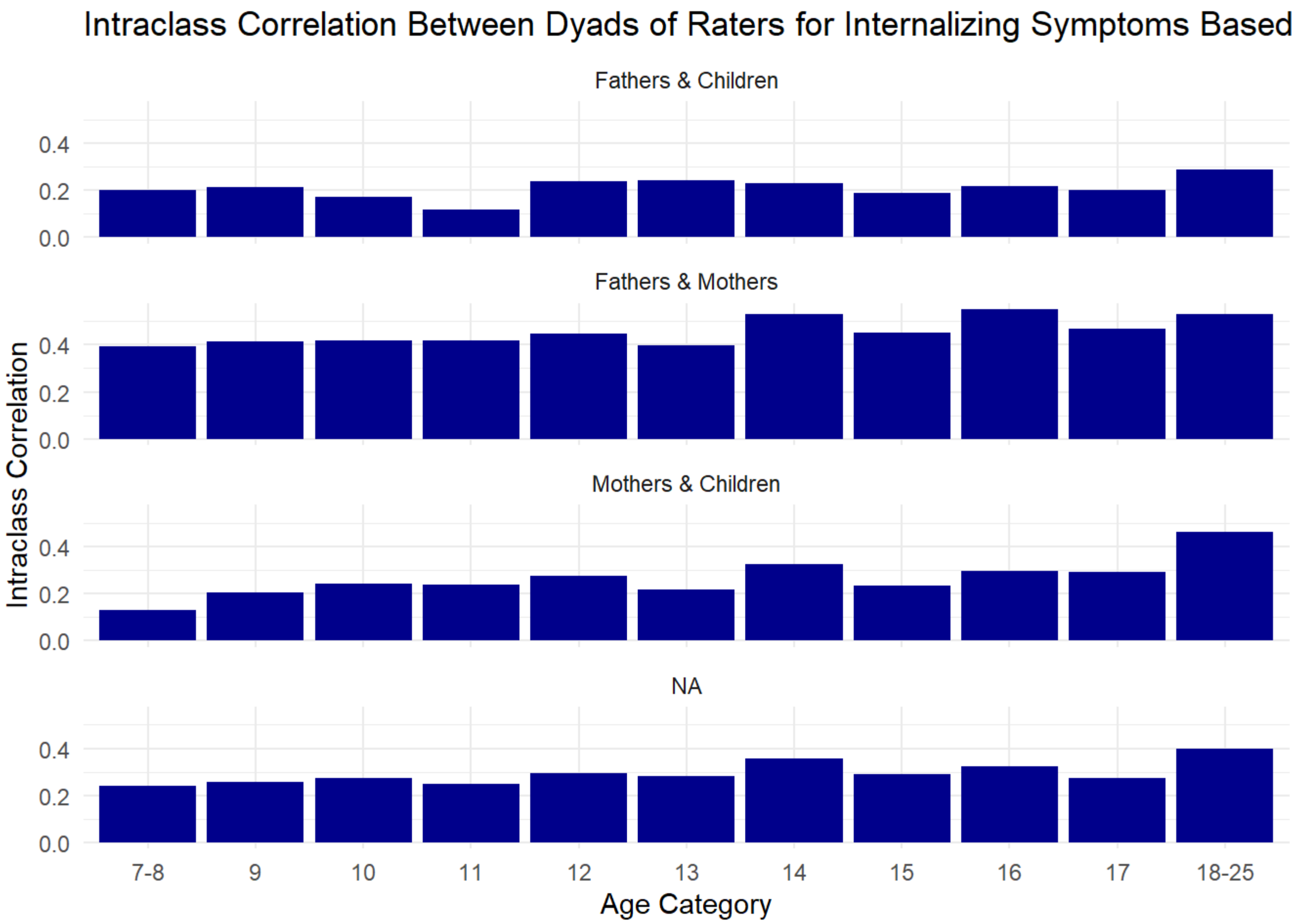
```
icc_df_int_age_rater <- data_df_long %>%
  select(int_scale_f, int_scale_m, int_scale_c, Age_cat) %>%
  group_split(Age_cat, .keep = TRUE) %>%
  lapply(function(group_data) {
    # Iterate over column combinations and apply calc_icc for each using do.call
    result_list <- do.call("rbind", lapply(column_comb_int, function(cols) calc_icc(cols, 
group_data)))
    result_list  # Return the results for this group
  }) %>%
  bind_rows()

icc_df_int_age_rater <- icc_df_int_age_rater %>%
  bind_cols(age_cat_4_vect) %>%
  rename(Age_cat = "value") %>%
  filter(!is.na(Age_cat))

icc_df_int_age_rater$Age_cat <- icc_df_int_age_rater$Age_cat %>% as_factor()

icc_df_int_age_rater
```

```
# A tibble: 44 × 5
   raters                       icc_value ci_lower_boundary ci_upper_boundary Age_cat
   <chr>                            <dbl>             <dbl>             <dbl> <fct>
```

```
 1 int_scale_f_int_scale_…     0.240            0.177             0.304 7-8
 2 int_scale_f_int_scale_c     0.199            0.0896            0.301 7-8
 3 int_scale_f_int_scale_m     0.394            0.315             0.467 7-8
 4 int_scale_c_int_scale_m     0.131            0.0523            0.208 7-8
 5 int_scale_f_int_scale_…     0.256            0.198             0.314 9
 6 int_scale_f_int_scale_c     0.212            0.113             0.302 9
 7 int_scale_f_int_scale_m     0.414            0.353             0.472 9
 8 int_scale_c_int_scale_m     0.205            0.125             0.279 9
 9 int_scale_f_int_scale_…     0.273            0.219             0.327 10
10 int_scale_f_int_scale_c     0.170            0.0904            0.245 10
# ℹ 34 more rows
```

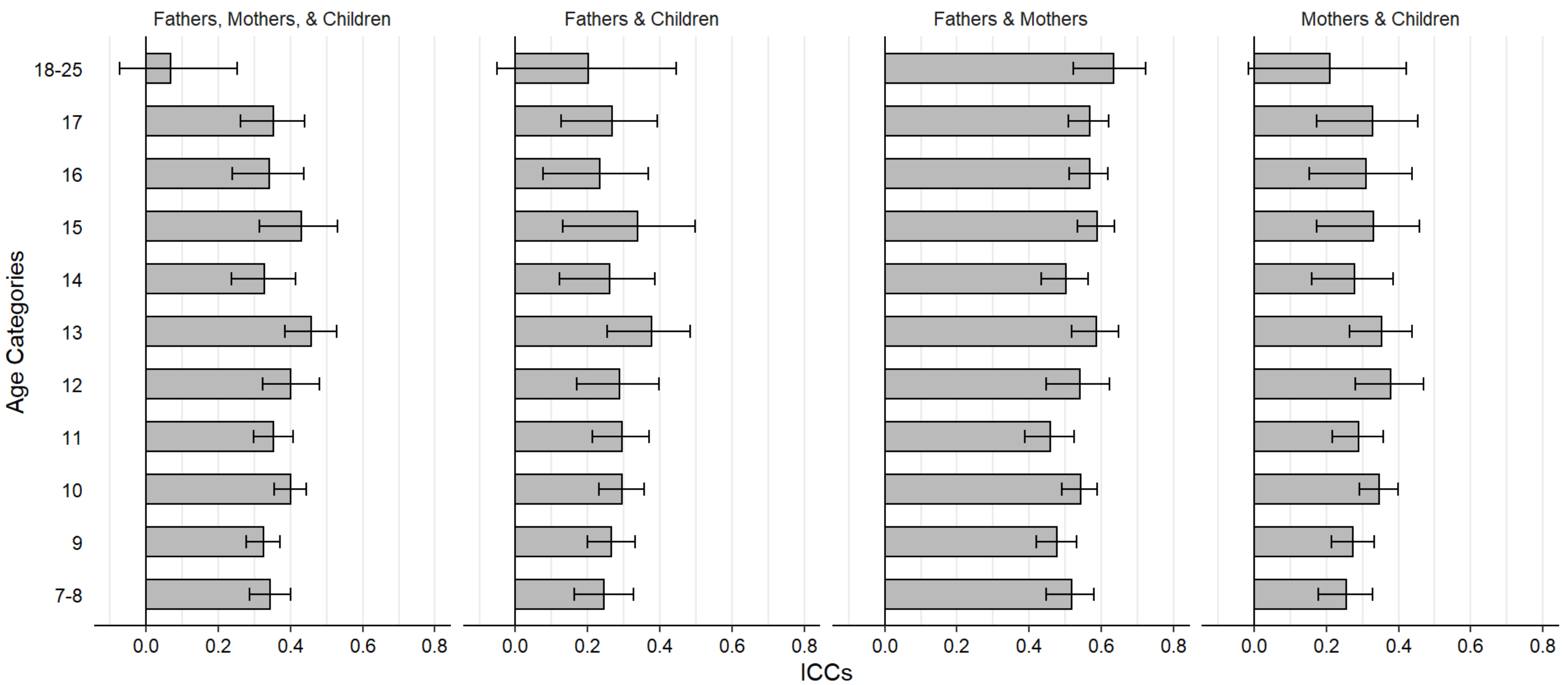
Plotting the intraclass correlation between dyads of raters for internalizing symptoms across ages:

```
icc_df_int_age_rater %>%
  mutate(raters = case_when(
    raters == "int_scale_f_int_scale_m" ~ "Fathers & Mothers",
    raters == "int_scale_f_int_scale_c" ~ "Fathers & Children",
    raters == "int_scale_c_int_scale_m" ~ "Mothers & Children"
  )) %>%
  ggplot(aes(x = Age_cat, y = icc_value)) +
  geom_bar(stat = "identity", show.legend = FALSE, fill = "darkblue") +
  facet_wrap(vars(raters), nrow = 4) +
  theme_minimal() +
  labs(title = "Intraclass Correlation Between Dyads of Raters for Internalizing Symptoms Based on
Age Category",
       x = "Age Category",
       y = "Intraclass Correlation")
```



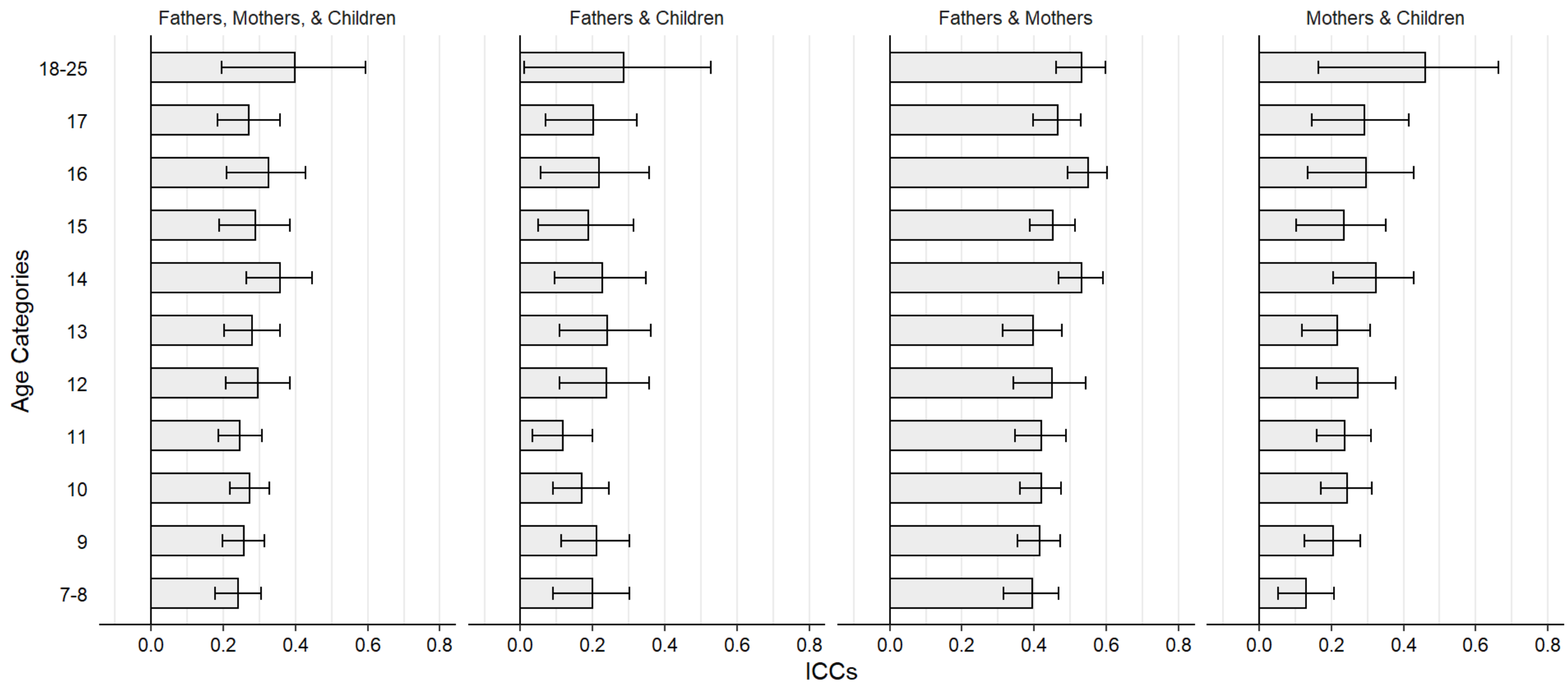
Making a plot for the thesis for ICC for externalizing symptoms with age:

```
icc_df_ext_age_rater %>%
  mutate(raters = case_when(
    raters == "ext_scale_f_ext_scale_c_ext_scale_m" ~ "Fathers, Mothers, & Children",
    raters == "ext_scale_f_ext_scale_c" ~ "Fathers & Children",
    raters == "ext_scale_f_ext_scale_m" ~ "Fathers & Mothers",
    raters == "ext_scale_c_ext_scale_m" ~ "Mothers & Children"
  )) %>%
  ggplot(aes(x = icc_value, y = Age_cat)) +
  geom_bar(stat = "identity", fill = "grey73", color = "black", width = 0.55) +
  facet_wrap(vars(fct_inorder(raters)), ncol = 4) +
  geom_errorbar(aes(xmin = ci_lower_boundary, xmax = ci_upper_boundary), width = 0.25) +
  xlim(-0.1, 0.8) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.y = element_blank(),
        axis.ticks.y = element_blank(),
        panel.grid.major.x = element_line(),
        panel.grid.minor.x = element_line(),
        strip.background = element_blank()) +
  geom_vline(aes(xintercept = 0)) +
  labs(x = "ICCs",
     y = "Age Categories")
```



Making a plot for the thesis for ICC for internalizing symptoms with age:

```
icc_df_int_age_rater %>%
  mutate(raters = case_when(
    raters == "int_scale_f_int_scale_c_int_scale_m" ~ "Fathers, Mothers, & Children",
    raters == "int_scale_f_int_scale_m" ~ "Fathers & Mothers",
    raters == "int_scale_f_int_scale_c" ~ "Fathers & Children",
    raters == "int_scale_c_int_scale_m" ~ "Mothers & Children"
  )) %>%
  ggplot(aes(x = icc_value, y = Age_cat)) +
  geom_bar(stat = "identity", fill = "grey93", color = "black", width = 0.55) +
  facet_wrap(vars(fct_inorder(raters)), ncol = 4) +
  geom_errorbar(aes(xmin = ci_lower_boundary, xmax = ci_upper_boundary), width = 0.25) +
  xlim(-0.1, 0.8) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.y = element_blank(),
        axis.ticks.y = element_blank(),
        panel.grid.major.x = element_line(),
        panel.grid.minor.x = element_line(),
        strip.background = element_blank()) +
  geom_vline(aes(xintercept = 0)) +
  labs(x = "ICCs",
       y = "Age Categories")
```



8.3 - Adult Attachment

Calculating intraclass correlation for externalizing symptoms between the raters at high and low attachment:

```
# This is not an elegant solution, but I don't want to redo my function calc_icc to work with
grouping variables, so since there are only two groups, I'll just do the grouping manually

icc_df_ext_aaa_f_h <- data_df_wide_diff %>%
  select(y8ext_scale_f, y8ext_scale_c, y10aaa_scale_ya_f_dich) %>% filter(y10aaa_scale_ya_f_dich ==
"High") %>% calc_icc(c("y8ext_scale_f", "y8ext_scale_c"), .) %>% mutate(Attachment = "High")

icc_df_ext_aaa_f_l <- data_df_wide_diff %>%
  select(y8ext_scale_f, y8ext_scale_c, y10aaa_scale_ya_f_dich) %>% filter(y10aaa_scale_ya_f_dich ==
"Low") %>% calc_icc(c("y8ext_scale_f", "y8ext_scale_c"), .) %>% mutate(Attachment = "Low")

icc_df_ext_aaa_m_h <- data_df_wide_diff %>%
  select(y8ext_scale_m, y8ext_scale_c, y10aaa_scale_ya_m_dich) %>% filter(y10aaa_scale_ya_m_dich ==
"High") %>% calc_icc(c("y8ext_scale_m", "y8ext_scale_c"), .) %>% mutate(Attachment = "High")

icc_df_ext_aaa_m_l <- data_df_wide_diff %>%
  select(y8ext_scale_m, y8ext_scale_c, y10aaa_scale_ya_m_dich) %>% filter(y10aaa_scale_ya_m_dich ==
"Low") %>% calc_icc(c("y8ext_scale_m", "y8ext_scale_c"), .) %>% mutate(Attachment = "Low")

icc_df_ext_aaa <- rbind(icc_df_ext_aaa_f_h, icc_df_ext_aaa_f_l, icc_df_ext_aaa_m_h,
icc_df_ext_aaa_m_l) %>% mutate(raters = case_when(
  raters == "y8ext_scale_f_y8ext_scale_c" ~ "Fathers & Children",
  raters == "y8ext_scale_m_y8ext_scale_c" ~ "Mothers & Children"
))

icc_df_ext_aaa
```

```
# A tibble: 4 × 5
  raters             icc_value ci_lower_boundary ci_upper_boundary Attachment
  <chr>                  <dbl>             <dbl>             <dbl> <chr>
1 Fathers & Children     0.295            0.0918             0.472 High
2 Fathers & Children     0.376            0.0854             0.581 Low
3 Mothers & Children     0.298            0.0813             0.475 High
4 Mothers & Children     0.348            0.104              0.534 Low
```

Calculating intraclass correlation for internalizing symptoms between the raters at high and low attachment:

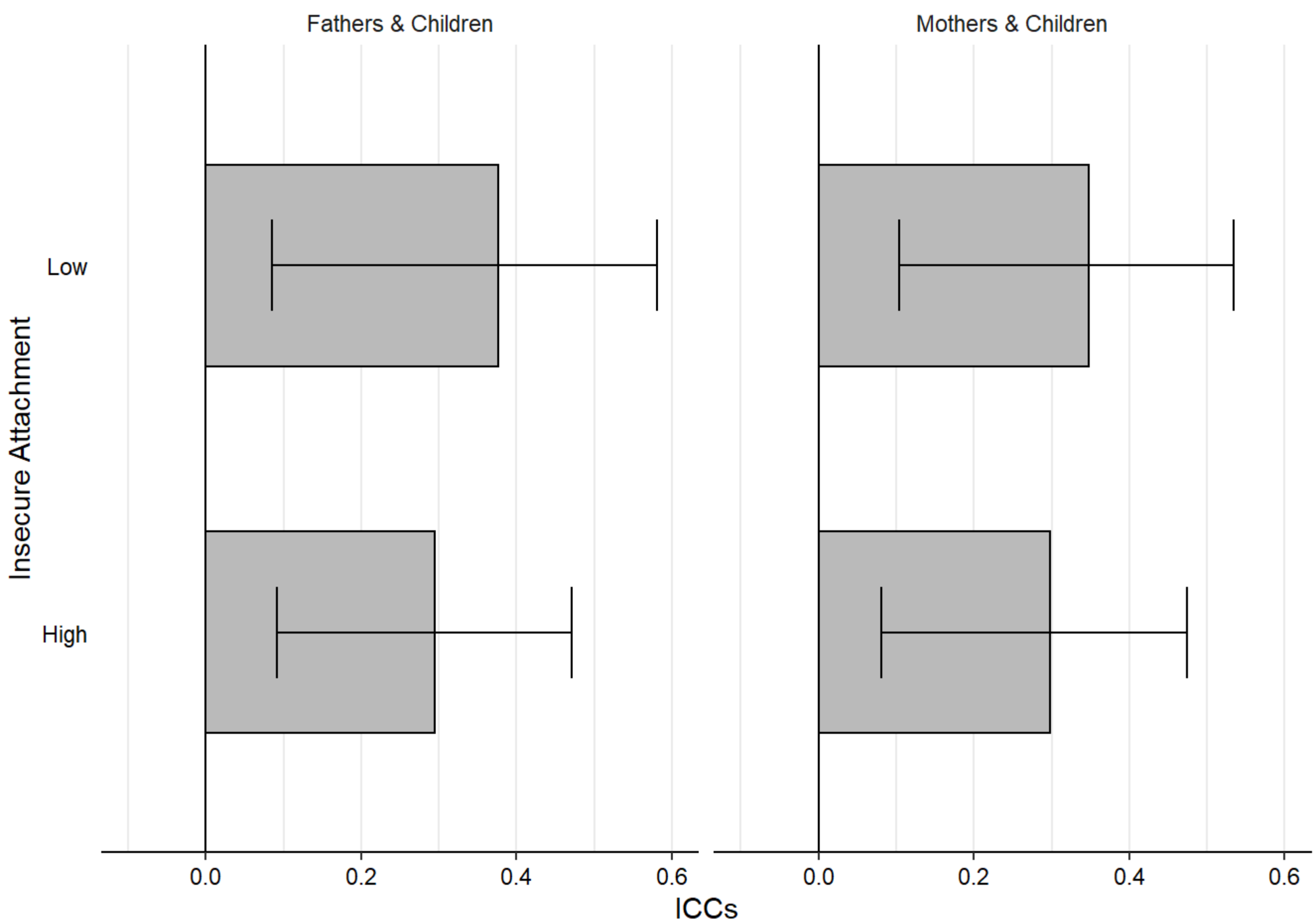
```
# This is not an elegant solution, but I don't want to redo my function calc_icc to work with grouping variables, so since there are only two groups, I'll just do the grouping manually

icc_df_int_aaa_f_h <- data_df_wide_diff %>%
  select(y8int_scale_f, y8int_scale_c, y10aaa_scale_ya_f_dich) %>% filter(y10aaa_scale_ya_f_dich == "High") %>% calc_icc(c("y8int_scale_f", "y8int_scale_c"), .) %>% mutate(Attachment = "High")

icc_df_int_aaa_f_l <- data_df_wide_diff %>%
  select(y8int_scale_f, y8int_scale_c, y10aaa_scale_ya_f_dich) %>% filter(y10aaa_scale_ya_f_dich == "Low") %>% calc_icc(c("y8int_scale_f", "y8int_scale_c"), .) %>% mutate(Attachment = "Low")

icc_df_int_aaa_m_h <- data_df_wide_diff %>%
  select(y8int_scale_m, y8int_scale_c, y10aaa_scale_ya_m_dich) %>% filter(y10aaa_scale_ya_m_dich == "High") %>% calc_icc(c("y8int_scale_m", "y8int_scale_c"), .) %>% mutate(Attachment = "High")

icc_df_int_aaa_m_l <- data_df_wide_diff %>%
  select(y8int_scale_m, y8int_scale_c, y10aaa_scale_ya_m_dich) %>% filter(y10aaa_scale_ya_m_dich == "Low") %>% calc_icc(c("y8int_scale_m", "y8int_scale_c"), .) %>% mutate(Attachment = "Low")

icc_df_int_aaa <- rbind(icc_df_int_aaa_f_h, icc_df_int_aaa_f_l, icc_df_int_aaa_m_h, icc_df_int_aaa_m_l) %>% mutate(raters = case_when(
  raters == "y8int_scale_f_y8int_scale_c" ~ "Fathers & Children",
  raters == "y8int_scale_m_y8int_scale_c" ~ "Mothers & Children"
))

icc_df_int_aaa
```

```
# A tibble: 4 × 5
  raters             icc_value ci_lower_boundary ci_upper_boundary Attachment
  <chr>                  <dbl>             <dbl>             <dbl> <chr>
1 Fathers & Children     0.209           0.00306             0.397 High
2 Fathers & Children     0.346           0.105               0.533 Low
3 Mothers & Children     0.307           0.0732              0.493 High
4 Mothers & Children     0.342           0.160               0.498 Low
```

Making a plot for the thesis for ICC for externalizing symptoms with aaa:

```
icc_df_ext_aaa %>%
  ggplot(aes(x = icc_value, y = Attachment)) +
  geom_bar(stat = "identity", fill = "grey73", color = "black", width = 0.55) +
  facet_wrap(vars(raters)) +
  geom_errorbar(aes(xmin = ci_lower_boundary, xmax = ci_upper_boundary), width = 0.25) +
  xlim(-0.1, 0.6) +
```

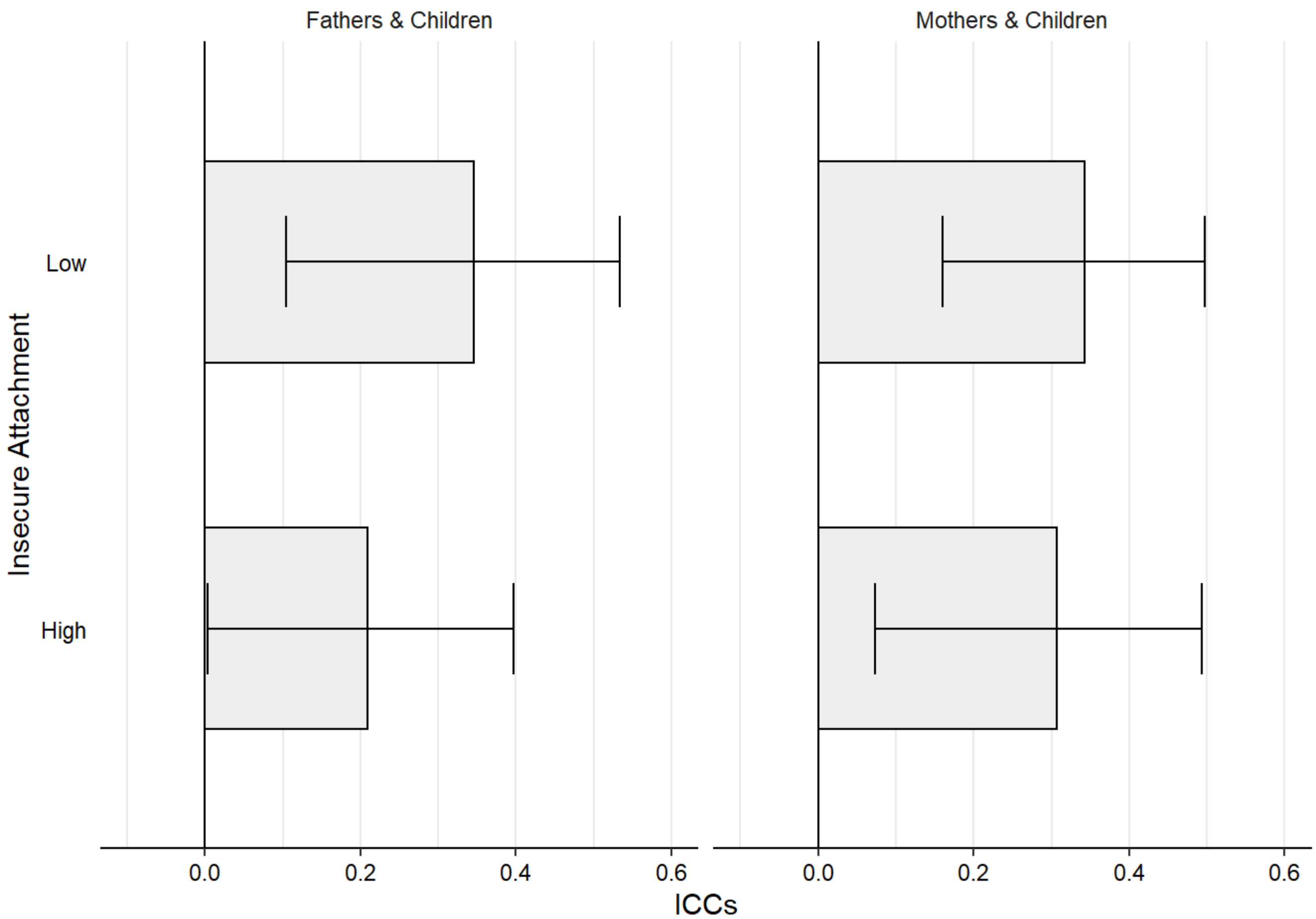
```r
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.y = element_blank(),
        axis.ticks.y = element_blank(),
        panel.grid.major.x = element_line(),
        panel.grid.minor.x = element_line(),
        strip.background = element_blank()) +
  geom_vline(aes(xintercept = 0)) +
  labs(x = "ICCs",
       y = "Insecure Attachment")
```



Making a plot for the thesis for ICC for internalizing symptoms with aaa:

```r
icc_df_int_aaa %>%
  ggplot(aes(x = icc_value, y = Attachment)) +
  geom_bar(stat = "identity", fill = "grey93", color = "black", width = 0.55) +
  facet_wrap(vars(raters)) +
  geom_errorbar(aes(xmin = ci_lower_boundary, xmax = ci_upper_boundary), width = 0.25) +
  xlim(-0.1, 0.6) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.y = element_blank(),
        axis.ticks.y = element_blank(),
```

```
        panel.grid.major.x = element_line(),
        panel.grid.minor.x = element_line(),
        strip.background = element_blank()) +
  geom_vline(aes(xintercept = 0)) +
  labs(x = "ICCs",
       y = "Insecure Attachment")
```



9 - Tri-factor model

9.1 - Checking Uniform Assumptions Before Fitting the Tri-Factor Model

Converting the data_df_long_unique df from wide to long in regards to rater, converting the symptom_rater_itemnum variables to long format:

```
# First, I will rename the columns to have the rater information at the end, which will make it easier to to the wide to long conversion for raters. The variables will now go from being named symptom_rater_item to symtpom_item_rater, which will make it easier to separate out the rater portion for the wide to long conversion
colnames(data_df_long_unique) <- gsub("^(ext|int)_([[:alpha:]]+)_(\\d+)", "\\1_\\3_\\2", colnames(data_df_long_unique))

# Wide to long conversion
```

```
data_df_longer_rater_unique <- data_df_long_unique %>%
  pivot_longer(cols = matches("^(ext|int)_(\\d)+_[[:alpha:]]+"), names_to = c(".value", "rater"),
names_pattern = "^((?:ext|int)_(?:\\d)+)_([[:alpha:]]+)")
```

Converting the rater variable to factor:

```
data_df_longer_rater_unique$rater <- data_df_longer_rater_unique$rater %>% factor()
```

Doing the same for data_df_long (not the unique version):

```
# First, I will rename the columns to have the rater information at the end, which will make it
easier to to the wide to long conversion for raters. The variables will now go from being named
symptom_rater_item to symtpom_item_rater, which will make it easier to separate out the rater
portion for the wide to long conversion
colnames(data_df_long) <- gsub("^(ext|int)_([[:alpha:]]+)_(\\d+)", "\\1_\\3_\\2",
colnames(data_df_long))

# Wide to long conversion
data_df_longer_rater <- data_df_long %>%
  pivot_longer(cols = matches("^(ext|int)_(\\d)+_[[:alpha:]]+"), names_to = c(".value", "rater"),
names_pattern = "^((?:ext|int)_(?:\\d)+)_([[:alpha:]]+)")
```

Converting the rater variable to factor:

```
data_df_longer_rater$rater <- data_df_longer_rater$rater %>% factor()
```

9.1.1 - In general

Collapsing the two highest response options for externalizing symptoms in data_df_longer_rater_unique as they had too low of a distribution, so that the model will fit correlatly:

```
data_df_longer_rater_unique <- data_df_longer_rater_unique %>%
  mutate(across(starts_with("ext_"),
              ~ case_when(
                . == "Not" ~ "Not",
                . %in% c("Somewhat/sometimes", "Very/often") ~ "Somewhat/sometimes-Very/often",
                TRUE ~ as.character(.)
              )))
```

Checking the assumption for further model fitting to the tri-factor model by looking at uniform distributions for all combinations of raters for externalizing symptoms:

```
model_ext <- 'f1 =~ NA*ext_1 + ext_2 + ext_3 + ext_4 + ext_5 + ext_6 + ext_7 + ext_8 + ext_9 +
ext_10 + ext_11 + ext_12 + ext_13 + ext_14 + ext_15 + ext_16 + ext_17 + ext_18
              f1 ~~ 1*f1'

# Item num is different for the externalizing and internalizing scales
test_uniform_ass <- function(data, model, item_num) {
  data %>%
    # Group everything by rater
```

```
    group_by(rater) %>%
    # Define what to do while being grouped
    group_modify( ~{
      # Fit the model
      fit <- cfa(model, data = .x, ordered = TRUE)
      # Saving the relevant estimated values in a vector
      est_values_vect <- parameterEstimates(fit)
      # Saving a data frame with the relevant fit statistics
      temp_df <- tibble(
        rmsea = fitMeasures(fit)["rmsea"],
        cfi = fitMeasures(fit)["cfi"],
        tli = fitMeasures(fit)["tli"])
      # Saving a data frame with the relevant estimated values
        # Using t() to get each value as a separate column instead of a separate row
      est_df <- est_values_vect[[4]][1:item_num] %>% t() %>% as_tibble()
      # Binding the estiamted values and fit statistics into the same data frame
      temp_df <- bind_cols(temp_df, est_df)
    }) %>%
    ungroup()
}

uniform_ass_df_ext <- test_uniform_ass(data_df_longer_rater_unique, model_ext, 18)
```

```
Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
`.name_repair` is omitted as of tibble 2.0.0.
ℹ Using compatibility `.name_repair`.
```

```
colnames(uniform_ass_df_ext) <- gsub("^V(\\d)", "est_\\1", colnames(uniform_ass_df_ext))

print(uniform_ass_df_ext)
```

```
# A tibble: 3 × 22
  rater  rmsea   cfi   tli est_1 est_2 est_3 est_4 est_5 est_6 est_7 est_8 est_9
  <fct>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 cya   0.0383 0.968 0.964 0.553 0.600 0.636 0.359 0.564 0.642 0.469 0.566 0.708
2 f     0.0412 0.974 0.970 0.576 0.551 0.670 0.566 0.541 0.718 0.636 0.635 0.776
3 m     0.0463 0.967 0.963 0.558 0.528 0.731 0.565 0.568 0.715 0.601 0.561 0.714
# ℹ 9 more variables: est_10 <dbl>, est_11 <dbl>, est_12 <dbl>, est_13 <dbl>,
#   est_14 <dbl>, est_15 <dbl>, est_16 <dbl>, est_17 <dbl>, est_18 <dbl>
```

I get a warning stating “Warning message: lavaan->lav_model_vcov():
The variance-covariance matrix of the estimated parameters (vcov) does not appear to be positive definite! The smallest eigenvalue (= -2.342213e-17) is smaller than zero. This may be a symptom that the model is not identified.” from running the above code. I get output and it looks good, but even when collapsing the two highest categories, I get this warning

TESTING THE CODE AS ABOVE, BUT DOING IT MANUALLY WITH LOOPS. TO CHECK THAT EVERYTHING WORKS AS IT SHOULD:

```
model_ext <- 'f1 =~ NA*ext_1 + ext_2 + ext_3 + ext_4 + ext_5 + ext_6 + ext_7 + ext_8 + ext_9 +
ext_10 + ext_11 + ext_12 + ext_13 + ext_14 + ext_15 + ext_16 + ext_17 + ext_18
                f1 ~~ 1*f1'
```

```
# Item num is different for the externalizing and internalizing scales
test_uniform_ass_looping <- function(data, model, item_num, rater_var) {
  data <- data %>%
    filter(rater == rater_var)
      # Fit the model
      fit <- cfa(model, data = data, ordered = TRUE)
      # Saving the relevant estimated values in a vector
      est_values_vect <- parameterEstimates(fit)
      # Saving a data frame with the relevant fit statistics
      temp_df <- tibble(
        rater = rater_var,
        rmsea = fitMeasures(fit)["rmsea"],
        cfi = fitMeasures(fit)["cfi"],
        tli = fitMeasures(fit)["tli"])
      # Saving a data frame with the relevant estimated values
        # Using t() to get each value as a separate column instead of a separate row
      est_df <- est_values_vect[[4]][1:item_num] %>% t() %>% as_tibble()
      # Binding the estiamted values and fit statistics into the same data frame
      temp_df <- bind_cols(temp_df, est_df)
}

rater_vect_test <- c("cya", "f", "m")


for (i in 1:length(rater_vect_test)){
  if (i == 1){
    uniform_ext_loop_df <- test_uniform_ass_looping(data_df_longer_rater_unique, model_ext, 18,
rater_vect_test[i])
  } else {
    temp_df <- test_uniform_ass_looping(data_df_longer_rater_unique, model_ext, 18,
rater_vect_test[i])
    uniform_ext_loop_df <- rbind(uniform_ext_loop_df, temp_df)
  }
}

colnames(uniform_ext_loop_df) <- gsub("^V(\\d)", "est_\\1", colnames(uniform_ext_loop_df))

print(uniform_ext_loop_df)
```

```
# A tibble: 3 × 22
  rater  rmsea   cfi   tli est_1 est_2 est_3 est_4 est_5 est_6 est_7 est_8 est_9
  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 cya   0.0383 0.968 0.964 0.553 0.600 0.636 0.359 0.564 0.642 0.469 0.566 0.708
2 f     0.0412 0.974 0.970 0.576 0.551 0.670 0.566 0.541 0.718 0.636 0.635 0.776
3 m     0.0463 0.967 0.963 0.558 0.528 0.731 0.565 0.568 0.715 0.601 0.561 0.714
# ℹ 9 more variables: est_10 <dbl>, est_11 <dbl>, est_12 <dbl>, est_13 <dbl>,
#   est_14 <dbl>, est_15 <dbl>, est_16 <dbl>, est_17 <dbl>, est_18 <dbl>
```

It looks like these numbers are the same as for the code above. So that code works well.

Checking for out-of-bounds combinations:

We are looking at the three following fit indices: RMSEA, CFI, and TLI We have defined RMSEA > .1 and CFI and TLI < .9 as being out of bounds

```
# Filtering rows with fit indices that are out of bounds
uniform_ass_df_ext %>% filter(rmsea > .1 | cfi < .9 | tli < .9)
```

```
# A tibble: 0 × 22
# ℹ 22 variables: rater <fct>, rmsea <dbl>, cfi <dbl>, tli <dbl>, est_1 <dbl>,
#   est_2 <dbl>, est_3 <dbl>, est_4 <dbl>, est_5 <dbl>, est_6 <dbl>,
#   est_7 <dbl>, est_8 <dbl>, est_9 <dbl>, est_10 <dbl>, est_11 <dbl>,
#   est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>, est_16 <dbl>,
#   est_17 <dbl>, est_18 <dbl>
```

Checking for Non-ideal combinations:

We defined RMSEA > .05 and CFI and TLI < .95 as being ok, but not ideal

```
# Filtering rows with fit indices with non-ideal fit indices
uniform_ass_df_ext %>% filter(rmsea > .05 | cfi < .95 | tli < .95)
```

```
# A tibble: 0 × 22
# ℹ 22 variables: rater <fct>, rmsea <dbl>, cfi <dbl>, tli <dbl>, est_1 <dbl>,
#   est_2 <dbl>, est_3 <dbl>, est_4 <dbl>, est_5 <dbl>, est_6 <dbl>,
#   est_7 <dbl>, est_8 <dbl>, est_9 <dbl>, est_10 <dbl>, est_11 <dbl>,
#   est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>, est_16 <dbl>,
#   est_17 <dbl>, est_18 <dbl>
```

Checking for perfect combinations:

```
# Filtering rows with fit indices that are perfect
uniform_ass_df_ext %>% filter(rmsea == 0 | cfi == 1 | tli == 1)
```

```
# A tibble: 0 × 22
# ℹ 22 variables: rater <fct>, rmsea <dbl>, cfi <dbl>, tli <dbl>, est_1 <dbl>,
#   est_2 <dbl>, est_3 <dbl>, est_4 <dbl>, est_5 <dbl>, est_6 <dbl>,
#   est_7 <dbl>, est_8 <dbl>, est_9 <dbl>, est_10 <dbl>, est_11 <dbl>,
#   est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>, est_16 <dbl>,
#   est_17 <dbl>, est_18 <dbl>
```

We are not collapsing the scores for internalizing symptoms like we did with externalizing symptoms, as there was more variability in the scores for internalizing symptoms than externalizing symptoms.

Checking the assumption for further model fitting to the tri-factor model by looking at uniform distributions for all combinations of raters for internalizing symptoms:

```
model_int <- 'f1 =~ NA*int_1 + int_2 + int_3 + int_4 + int_5 + int_6 + int_7 + int_8 + int_9 +
int_10 + int_11 + int_12 + int_13 + int_14 + int_15 + int_16 + int_17
                f1 ~~ 1*f1'

uniform_ass_df_int <- test_uniform_ass(data_df_longer_rater_unique, model_int, 17)

colnames(uniform_ass_df_int) <- gsub("^V(\\d)", "est_\\1", colnames(uniform_ass_df_int))

print(uniform_ass_df_int)
```

```
# A tibble: 3 × 21
  rater  rmsea   cfi   tli est_1 est_2 est_3 est_4 est_5 est_6 est_7 est_8 est_9
  <fct>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 cya   0.0545 0.969 0.965 0.596 0.522 0.528 0.292 0.587 0.546 0.706 0.669 0.714
2 f     0.0554 0.964 0.959 0.550 0.514 0.546 0.412 0.623 0.629 0.688 0.754 0.815
3 m     0.0608 0.954 0.947 0.536 0.465 0.503 0.449 0.631 0.609 0.640 0.730 0.811
# ℹ 8 more variables: est_10 <dbl>, est_11 <dbl>, est_12 <dbl>, est_13 <dbl>,
#   est_14 <dbl>, est_15 <dbl>, est_16 <dbl>, est_17 <dbl>
```

Fitting for internalizing symptoms does not give any warnings or errors.

Checking for out-of-bounds combinations:

We are looking at the three following fit indices: RMSEA, CFI, and TLI We have defined RMSEA > .1 and CFI and TLI < .9 as being out of bounds

```
# Filtering rows with fit indices that are out of bounds
uniform_ass_df_int %>% filter(rmsea > .1 | cfi < .9 | tli < .9)
```

```
# A tibble: 0 × 21
# ℹ 21 variables: rater <fct>, rmsea <dbl>, cfi <dbl>, tli <dbl>, est_1 <dbl>,
#   est_2 <dbl>, est_3 <dbl>, est_4 <dbl>, est_5 <dbl>, est_6 <dbl>,
#   est_7 <dbl>, est_8 <dbl>, est_9 <dbl>, est_10 <dbl>, est_11 <dbl>,
#   est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>, est_16 <dbl>,
#   est_17 <dbl>
```

Checking for Non-ideal combinations:

We defined RMSEA > .05 and CFI and TLI < .95 as being ok, but not ideal

```
# Filtering rows with fit indices with non-ideal fit indices
uniform_ass_df_int %>% filter(rmsea > .05 | cfi < .95 | tli < .95)
```

```
# A tibble: 3 × 21
  rater  rmsea   cfi   tli est_1 est_2 est_3 est_4 est_5 est_6 est_7 est_8 est_9
  <fct>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 cya   0.0545 0.969 0.965 0.596 0.522 0.528 0.292 0.587 0.546 0.706 0.669 0.714
2 f     0.0554 0.964 0.959 0.550 0.514 0.546 0.412 0.623 0.629 0.688 0.754 0.815
3 m     0.0608 0.954 0.947 0.536 0.465 0.503 0.449 0.631 0.609 0.640 0.730 0.811
# ℹ 8 more variables: est_10 <dbl>, est_11 <dbl>, est_12 <dbl>, est_13 <dbl>,
#   est_14 <dbl>, est_15 <dbl>, est_16 <dbl>, est_17 <dbl>
```

Checking for perfect combinations:

```
# Filtering rows with fit indices that are perfect
uniform_ass_df_int %>% filter(rmsea == 0 | cfi == 1 | tli == 1)
```

```
# A tibble: 0 × 21
# ℹ 21 variables: rater <fct>, rmsea <dbl>, cfi <dbl>, tli <dbl>, est_1 <dbl>,
#   est_2 <dbl>, est_3 <dbl>, est_4 <dbl>, est_5 <dbl>, est_6 <dbl>,
#   est_7 <dbl>, est_8 <dbl>, est_9 <dbl>, est_10 <dbl>, est_11 <dbl>,
```

```
#   est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>, est_16 <dbl>,
#   est_17 <dbl>
```

9.1.2 - At each Age

Collapsing the two highest response options for externalizing symptoms in data_df_longer_rater_unique as they had too low of a distribution, so that the model will fit correlatly:

```
data_df_longer_rater <- data_df_longer_rater %>%
  mutate(across(starts_with("ext_"),
                ~ case_when(
                  . == "Not" ~ "Not",
                  . %in% c("Somewhat/sometimes", "Very/often") ~ "Somewhat/sometimes-Very/often",
                  TRUE ~ as.character(.)
                )))
```

Checking the assumption for further model fitting to the tri-factor model by looking at uniform distributions for all combinations of raters for externalizing symptoms:

This code doesn't run, as there is a problem fitting the model to one of the categories. The next few code blocks is different variations of trying to troubleshoot this, including the last one with the loop, which is to make sure the problem wasn't with how this function is written and to see more easily what combination of rater and age category failed.

```
# Item num is different for the externalizing and internalizing scales
test_uniform_ass_age <- function(data, model, item_num) {
  data %>%
    # Group everything by rater and age category
    group_by(rater, Age_cat) %>%
    # Define what to do while being grouped
    group_modify( ~{
      # Fit the model
      fit <- cfa(model, data = .x, ordered = TRUE)
      # Saving the relevant estimated values in a vector
      est_values_vect <- parameterEstimates(fit)
      # Saving a data frame with the relevant fit statistics
      temp_df <- tibble(
        rmsea = fitMeasures(fit)["rmsea"],
        cfi = fitMeasures(fit)["cfi"],
        tli = fitMeasures(fit)["tli"])
      # Saving a data frame with the relevant estimated values
        # Using t() to get each value as a separate column instead of a separate row
      est_df <- est_values_vect[[4]][1:item_num] %>% t() %>% as_tibble()
      # Binding the estimated values and fit statistics into the same data frame
      temp_df <- bind_cols(temp_df, est_df)
    }) %>%
    ungroup()
}

# The data_df_longer_rater data contains several values in Age_cat that are NA. This leads the code
to try to fit a model to when Age_cat = NA. Therefore, we will filter out the responses that have
missing age values before fitting the model
data_df_longer_rater_no_na <- data_df_longer_rater %>% filter(!is.na(Age_cat))
```

```r
uniform_ass_df_ext_age <- test_uniform_ass_age(data_df_longer_rater_no_na, model_ext, 18)

colnames(uniform_ass_df_ext_age) <- gsub("^V(\\d)", "est_\\1", colnames(uniform_ass_df_ext_age))

print(uniform_ass_df_ext_age)
```

```
# A tibble: 33 × 23
   rater Age_cat  rmsea   cfi   tli  est_1  est_2  est_3  est_4  est_5  est_6
   <fct> <fct>    <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
 1 cya   7-8     0.0228 0.985 0.983  0.507  0.587  0.615  0.207  0.343  0.624
 2 cya   9       0.0380 0.971 0.967  0.478  0.571  0.688  0.382  0.563  0.652
 3 cya   10      0.0330 0.981 0.978  0.507  0.631  0.699  0.395  0.662  0.608
 4 cya   11      0.0333 0.978 0.975  0.551  0.571  0.652  0.426  0.547  0.613
 5 cya   12      0      1     1.00  -0.433 -0.702 -0.621 -0.445 -0.560 -0.695
 6 cya   13      0.0343 0.983 0.981  0.445  0.657  0.685  0.561  0.604  0.761
 7 cya   14      0.0399 0.966 0.962  0.481  0.552  0.601  0.436  0.508  0.601
 8 cya   15      0.0426 0.966 0.961  0.678  0.640  0.573  0.417  0.520  0.666
 9 cya   16      0.0483 0.956 0.951  0.637  0.509  0.592  0.349  0.631  0.744
10 cya   17      0.0459 0.962 0.957  0.655  0.596  0.657  0.417  0.626  0.756
# ℹ 23 more rows
# ℹ 12 more variables: est_7 <dbl>, est_8 <dbl>, est_9 <dbl>, est_10 <dbl>,
#   est_11 <dbl>, est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>,
#   est_16 <dbl>, est_17 <dbl>, est_18 <dbl>
```

The code works, but two combinations give a perfect fit. This is the case for cya at 12 years old and mother at 18-25 years old.

Checking for out-of-bounds combinations:

We are looking at the three following fit indices: RMSEA, CFI, and TLI We have defined RMSEA > .1 and CFI and TLI < .9 as being out of bounds

```r
# Filtering rows with fit indices that are out of bounds
uniform_ass_df_ext_age %>% filter(rmsea > .1 | cfi < .9 | tli < .9)
```

```
# A tibble: 0 × 23
# ℹ 23 variables: rater <fct>, Age_cat <fct>, rmsea <dbl>, cfi <dbl>,
#   tli <dbl>, est_1 <dbl>, est_2 <dbl>, est_3 <dbl>, est_4 <dbl>, est_5 <dbl>,
#   est_6 <dbl>, est_7 <dbl>, est_8 <dbl>, est_9 <dbl>, est_10 <dbl>,
#   est_11 <dbl>, est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>,
#   est_16 <dbl>, est_17 <dbl>, est_18 <dbl>
```

Checking for Non-ideal combinations:

We defined RMSEA > .05 and CFI and TLI < .95 as being ok, but not ideal

```r
# Filtering rows with fit indices with non-ideal fit indices
uniform_ass_df_ext_age %>% filter(rmsea > .05 | cfi < .95 | tli < .95)
```

```
# A tibble: 2 × 23
  rater Age_cat  rmsea   cfi   tli est_1 est_2 est_3 est_4 est_5 est_6 est_7
  <fct> <fct>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
```

```
1 f     16      0.0517 0.970 0.966 0.637 0.572 0.779 0.622 0.752 0.838 0.733
2 m     16      0.0545 0.964 0.959 0.694 0.538 0.807 0.649 0.659 0.778 0.690
# ℹ 11 more variables: est_8 <dbl>, est_9 <dbl>, est_10 <dbl>, est_11 <dbl>,
#   est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>, est_16 <dbl>,
#   est_17 <dbl>, est_18 <dbl>
```

Checking for perfect combinations:

```
# Filtering rows with fit indices that are perfect
uniform_ass_df_ext_age %>% filter(rmsea == 0 | cfi == 1 | tli == 1)
```

```
# A tibble: 2 × 23
  rater Age_cat rmsea   cfi   tli  est_1  est_2  est_3  est_4  est_5  est_6
  <fct> <fct>   <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1 cya   12          0     1  1.00 -0.433 -0.702 -0.621 -0.445 -0.560 -0.695
2 m     18-25       0     1  1.00  0.650  0.686  0.908  0.760  0.523  0.676
# ℹ 12 more variables: est_7 <dbl>, est_8 <dbl>, est_9 <dbl>, est_10 <dbl>,
#   est_11 <dbl>, est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>,
#   est_16 <dbl>, est_17 <dbl>, est_18 <dbl>
```

Fitting the models for internalizing symptoms with age categories:

```
# We still use the data_df_longer_rater_no_na data frame which is filtered to have no data points
with missing age categories values

uniform_ass_df_int_age <- test_uniform_ass_age(data_df_longer_rater_no_na, model_int, 17)

colnames(uniform_ass_df_int_age) <- gsub("^V(\\d)", "est_\\1", colnames(uniform_ass_df_int_age))

print(uniform_ass_df_int_age)
```

```
# A tibble: 33 × 22
   rater Age_cat  rmsea   cfi   tli est_1 est_2 est_3 est_4 est_5 est_6 est_7
   <fct> <fct>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1 cya   7-8     0.0433 0.964 0.959 0.441 0.426 0.583 0.256 0.494 0.512 0.507
 2 cya   9       0.0434 0.972 0.968 0.486 0.519 0.493 0.309 0.497 0.544 0.642
 3 cya   10      0.0506 0.968 0.963 0.583 0.506 0.450 0.311 0.596 0.623 0.689
 4 cya   11      0.0598 0.960 0.955 0.602 0.522 0.471 0.309 0.663 0.564 0.718
 5 cya   12      0.0476 0.982 0.980 0.694 0.583 0.552 0.365 0.599 0.659 0.714
 6 cya   13      0.0429 0.984 0.982 0.748 0.585 0.494 0.352 0.735 0.603 0.695
 7 cya   14      0.0609 0.970 0.966 0.697 0.620 0.480 0.431 0.687 0.608 0.750
 8 cya   15      0.0637 0.976 0.973 0.744 0.652 0.465 0.343 0.773 0.631 0.835
 9 cya   16      0.0645 0.979 0.976 0.744 0.626 0.467 0.351 0.745 0.607 0.835
10 cya   17      0.0603 0.983 0.981 0.779 0.646 0.623 0.466 0.847 0.597 0.834
# ℹ 23 more rows
# ℹ 10 more variables: est_8 <dbl>, est_9 <dbl>, est_10 <dbl>, est_11 <dbl>,
#   est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>, est_16 <dbl>,
#   est_17 <dbl>
```

Checking for out-of-bounds combinations:

We are looking at the three following fit indices: RMSEA, CFI, and TLI We have defined RMSEA > .1 and CFI and TLI

< .9 as being out of bounds

```
# Filtering rows with fit indices that are out of bounds
uniform_ass_df_int_age %>% filter(rmsea > .1 | cfi < .9 | tli < .9)
```

```
# A tibble: 1 × 22
  rater Age_cat  rmsea   cfi   tli est_1 est_2 est_3 est_4 est_5 est_6 est_7
  <fct> <fct>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 m     7-8     0.0790 0.909 0.895 0.482 0.482 0.425 0.401 0.569 0.522 0.551
# ℹ 10 more variables: est_8 <dbl>, est_9 <dbl>, est_10 <dbl>, est_11 <dbl>,
#   est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>, est_16 <dbl>,
#   est_17 <dbl>
```

Checking for Non-ideal combinations:

We defined RMSEA > .05 and CFI and TLI < .95 as being ok, but not ideal

```
# Filtering rows with fit indices with non-ideal fit indices
uniform_ass_df_int_age %>% filter(rmsea > .05 | cfi < .95 | tli < .95)
```

```
# A tibble: 28 × 22
   rater Age_cat  rmsea   cfi   tli est_1 est_2 est_3 est_4 est_5 est_6 est_7
   <fct> <fct>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1 cya   10      0.0506 0.968 0.963 0.583 0.506 0.450 0.311 0.596 0.623 0.689
 2 cya   11      0.0598 0.960 0.955 0.602 0.522 0.471 0.309 0.663 0.564 0.718
 3 cya   14      0.0609 0.970 0.966 0.697 0.620 0.480 0.431 0.687 0.608 0.750
 4 cya   15      0.0637 0.976 0.973 0.744 0.652 0.465 0.343 0.773 0.631 0.835
 5 cya   16      0.0645 0.979 0.976 0.744 0.626 0.467 0.351 0.745 0.607 0.835
 6 cya   17      0.0603 0.983 0.981 0.779 0.646 0.623 0.466 0.847 0.597 0.834
 7 cya   18-25   0.0571 0.987 0.985 0.768 0.604 0.499 0.330 0.766 0.648 0.845
 8 f     7-8     0.0583 0.940 0.931 0.442 0.379 0.461 0.393 0.526 0.556 0.625
 9 f     9       0.0648 0.932 0.922 0.483 0.471 0.380 0.273 0.550 0.618 0.605
10 f     11      0.0618 0.950 0.943 0.586 0.382 0.569 0.366 0.683 0.670 0.642
# ℹ 18 more rows
# ℹ 10 more variables: est_8 <dbl>, est_9 <dbl>, est_10 <dbl>, est_11 <dbl>,
#   est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>, est_16 <dbl>,
#   est_17 <dbl>
```

Checking for perfect combinations:

```
# Filtering rows with fit indices that are perfect
uniform_ass_df_int_age %>% filter(rmsea == 0 | cfi == 1 | tli == 1)
```

```
# A tibble: 0 × 22
# ℹ 22 variables: rater <fct>, Age_cat <fct>, rmsea <dbl>, cfi <dbl>,
#   tli <dbl>, est_1 <dbl>, est_2 <dbl>, est_3 <dbl>, est_4 <dbl>, est_5 <dbl>,
#   est_6 <dbl>, est_7 <dbl>, est_8 <dbl>, est_9 <dbl>, est_10 <dbl>,
#   est_11 <dbl>, est_12 <dbl>, est_13 <dbl>, est_14 <dbl>, est_15 <dbl>,
#   est_16 <dbl>, est_17 <dbl>
```

9.2 - Making mplus compatible code

The data has to be in a different format for the tri-factor model than for the uniform testing. For the uniform testing, we needed it to be in long format for both age and rater, while for the tri-factor model, we need it to be in long format only for age. Therefore, we will use the data_df_long and data_df_long_unique data to fit the tri-factor model. We haven't collapsed the two highest response alternatives for this data set yet, so we will make a copy of the data_df_long(_unique) data and collapse the two highest response alterantives. We only do this for the externalizing items:

```
data_df_long_collapsed <- data_df_long %>%
  mutate(across(starts_with("ext_"),
                ~ case_when(
                  . == "Not" ~ "Not",
                  . %in% c("Somewhat/sometimes", "Very/often") ~ "Somewhat/sometimes-Very/often",
                  TRUE ~ as.character(.)
                ))) %>%
  # Filtering for age not being missing since age is used as a predictor for hte tri-factor model.
I don't know whether I have to do this here or whether Mplus does it automatically, but I have it
here to be safe. I do not do this for the unique data set as we don't use age as a predictor for
that data set. Removes around 800 responses
  filter(!is.na(Age))
```

```
data_df_long_unique_collapsed <- data_df_long_unique %>%
  mutate(across(starts_with("ext_"),
                ~ case_when(
                  . == "Not" ~ "Not",
                  . %in% c("Somewhat/sometimes", "Very/often") ~ "Somewhat/sometimes-Very/often",
                  TRUE ~ as.character(.)
                )))
```

In Mplus, all the columns for the data frames has to be named explicitly. Therefore, I will filter the data_df_long_collapsed and data_df_long_unique_collapsed data frames so that they only have the variables we will need to fit the tri-factor model:

```
# Making a copy of the data frame to make mplus compatible
data_df_long_unique_collapsed_mplus <- data_df_long_unique_collapsed

data_df_long_unique_collapsed_mplus <- data_df_long_unique_collapsed_mplus %>% select(ID, Age,
ext_1_f, ext_1_m, ext_1_cya, ext_2_f, ext_2_m, ext_2_cya, ext_3_f, ext_3_m, ext_3_cya, ext_4_f,
ext_4_m, ext_4_cya, ext_5_f, ext_5_m, ext_5_cya, ext_6_f, ext_6_m, ext_6_cya, ext_7_f, ext_7_m,
ext_7_cya, ext_8_f, ext_8_m, ext_8_cya, ext_9_f, ext_9_m, ext_9_cya, ext_10_f, ext_10_m,
ext_10_cya, ext_11_f, ext_11_m, ext_11_cya, ext_12_f, ext_12_m, ext_12_cya, ext_13_f, ext_13_m,
ext_13_cya, ext_14_f, ext_14_m, ext_14_cya, ext_15_f, ext_15_m, ext_15_cya, ext_16_f, ext_16_m,
ext_16_cya, ext_17_f, ext_17_m, ext_17_cya, ext_18_f, ext_18_m, ext_18_cya, int_1_f, int_1_m,
int_1_cya, int_2_f, int_2_m, int_2_cya, int_3_f, int_3_m, int_3_cya, int_4_f, int_4_m, int_4_cya,
int_5_f, int_5_m, int_5_cya, int_6_f, int_6_m, int_6_cya, int_7_f, int_7_m, int_7_cya, int_8_f,
int_8_m, int_8_cya, int_9_f, int_9_m, int_9_cya, int_10_f, int_10_m, int_10_cya, int_11_f,
int_11_m, int_11_cya, int_12_f, int_12_m, int_12_cya, int_13_f, int_13_m, int_13_cya, int_14_f,
int_14_m, int_14_cya, int_15_f, int_15_m, int_15_cya, int_16_f, int_16_m, int_16_cya, int_17_f,
int_17_m, int_17_cya)
```

Mplus has a very specific format that the data has to be in for it to read it correctly. The following section will therefore make a copy of the data frames we will be using that are made compatible with mplus:

```
# Code to make it only for the first 100 rows, to have a more manageable file to inspect with
```

```
notebook
#data_df_long_unique_collapsed_mplus <- data_df_long_unique_collapsed_mplus[1:100,]

# Make free form data file and save it as a dat. file in a Mplus folder
prepareMplusData(df = data_df_long_unique_collapsed_mplus, filename =
"Mplus/data_df_long_collapsed_unique_mplus.dat")
```

```
-----
Factor: ext_1_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_1_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_1_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_2_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_2_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_2_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_3_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_3_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_3_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_4_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_4_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_4_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_5_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_5_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_5_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_6_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_6_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_6_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_7_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_7_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_7_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_8_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_8_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_8_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_9_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_9_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_9_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_10_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_10_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_10_cya
Conversion:
                            level number
                              Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_11_f
Conversion:
                            level number
                              Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_11_m
Conversion:
                            level number
                              Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_11_cya
Conversion:
                            level number
                              Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_12_f
Conversion:
                            level number
                              Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_12_m
Conversion:
                            level number
                              Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_12_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_13_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_13_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_13_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_14_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_14_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_14_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----


-----
Factor: ext_15_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----


-----
Factor: ext_15_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----


-----
Factor: ext_15_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----


-----
Factor: ext_16_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----


-----
Factor: ext_16_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_16_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_17_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_17_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_17_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_18_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_18_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_18_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: int_1_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_1_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_1_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_2_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_2_m
Conversion:
              level number
                Not      1
```

```
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_2_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_3_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_3_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_3_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_4_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_4_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_4_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_5_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_5_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_5_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_6_f
Conversion:
```

```
             level number
               Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_6_m
Conversion:
             level number
               Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_6_cya
Conversion:
             level number
               Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_7_f
Conversion:
             level number
               Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_7_m
Conversion:
             level number
               Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_7_cya
Conversion:
             level number
               Not      1
 Somewhat/sometimes      2
         Very/often      3
```

```
-----



-----
Factor: int_8_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_8_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_8_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_9_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_9_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
```

```
Factor: int_9_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_10_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_10_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_10_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_11_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_11_m
Conversion:
              level number
                Not      1
```

```
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_11_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_12_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_12_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_12_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_13_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_13_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_13_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_14_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_14_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_14_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_15_f
Conversion:
```

```
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_15_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_15_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_16_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_16_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_16_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
```

```
-----



-----
Factor: int_17_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_17_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_17_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----

The file(s)
 'data_df_long_collapsed_unique_mplus.dat'
currently exist(s) and will be overwritten

TITLE: Your title goes here
DATA: FILE = "Mplus/data_df_long_collapsed_unique_mplus.dat";
VARIABLE:
NAMES = ID Age ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya ext_3_f ext_3_m
     ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m ext_5_cya ext_6_f ext_6_m
     ext_6_cya ext_7_f ext_7_m ext_7_cya ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m
     ext_9_cya ext_10_f ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
     ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f ext_14_m ext_14_cya
     ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m ext_16_cya ext_17_f ext_17_m
     ext_17_cya ext_18_f ext_18_m ext_18_cya int_1_f int_1_m int_1_cya int_2_f
     int_2_m int_2_cya int_3_f int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f
     int_5_m int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m int_7_cya int_8_f
     int_8_m int_8_cya int_9_f int_9_m int_9_cya int_10_f int_10_m int_10_cya
     int_11_f int_11_m int_11_cya int_12_f int_12_m int_12_cya int_13_f int_13_m
     int_13_cya int_14_f int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f
     int_16_m int_16_cya int_17_f int_17_m int_17_cya;
MISSING=.;
```

```
# For Veronica
#prepareMplusData(df = data_df_long_unique_collapsed_mplus, filename =
"mplusdata/data_df_long_collapsed_unique_mplus.dat")

# The above code also gives an output that can be used as a basis for an inp file. However, note
that the file path is too long, so you need to add a line shift somewhere in it so that Mplus won't
cut it off
```

```
# Making a copy of the data frame to make mplus compatible
data_df_long_collapsed_mplus <- data_df_long_collapsed

data_df_long_collapsed_mplus <- data_df_long_collapsed_mplus %>% select(ID, Age_cat, ext_1_f,
ext_1_m, ext_1_cya, ext_2_f, ext_2_m, ext_2_cya, ext_3_f, ext_3_m, ext_3_cya, ext_4_f, ext_4_m,
ext_4_cya, ext_5_f, ext_5_m, ext_5_cya, ext_6_f, ext_6_m, ext_6_cya, ext_7_f, ext_7_m, ext_7_cya,
ext_8_f, ext_8_m, ext_8_cya, ext_9_f, ext_9_m, ext_9_cya, ext_10_f, ext_10_m, ext_10_cya, ext_11_f,
ext_11_m, ext_11_cya, ext_12_f, ext_12_m, ext_12_cya, ext_13_f, ext_13_m, ext_13_cya, ext_14_f,
ext_14_m, ext_14_cya, ext_15_f, ext_15_m, ext_15_cya, ext_16_f, ext_16_m, ext_16_cya, ext_17_f,
ext_17_m, ext_17_cya, ext_18_f, ext_18_m, ext_18_cya, int_1_f, int_1_m, int_1_cya, int_2_f,
int_2_m, int_2_cya, int_3_f, int_3_m, int_3_cya, int_4_f, int_4_m, int_4_cya, int_5_f, int_5_m,
int_5_cya, int_6_f, int_6_m, int_6_cya, int_7_f, int_7_m, int_7_cya, int_8_f, int_8_m, int_8_cya,
int_9_f, int_9_m, int_9_cya, int_10_f, int_10_m, int_10_cya, int_11_f, int_11_m, int_11_cya,
int_12_f, int_12_m, int_12_cya, int_13_f, int_13_m, int_13_cya, int_14_f, int_14_m, int_14_cya,
int_15_f, int_15_m, int_15_cya, int_16_f, int_16_m, int_16_cya, int_17_f, int_17_m, int_17_cya)
```

```
# Code to make it only for the first 100 rows, to have a more manageable file to inspect with
notebook
  # data_df_long_collapsed_mplus <- data_df_long_collapsed_mplus[1:100,]

# Make free form data file and save it as a dat. file in a Mplus folder
prepareMplusData(df = data_df_long_collapsed_mplus, filename =
"Mplus/data_df_long_collapsed_mplus.dat")
```

```
-----
Factor: Age_cat
Conversion:
 level number
   7-8      1
     9      2
    10      3
    11      4
    12      5
    13      6
    14      7
    15      8
    16      9
    17     10
 18-25     11
-----
```

```
-----
Factor: ext_1_f
Conversion:
                              level number
                                Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_1_m
Conversion:
                              level number
                                Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_1_cya
Conversion:
                              level number
                                Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_2_f
Conversion:
                              level number
                                Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_2_m
Conversion:
                              level number
                                Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_2_cya
Conversion:
                              level number
                                Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_3_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_3_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_3_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_4_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_4_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_4_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_5_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_5_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_5_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_6_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_6_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_6_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_7_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_7_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_7_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_8_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_8_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_8_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_9_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_9_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_9_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_10_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_10_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_10_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_11_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_11_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_11_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_12_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_12_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_12_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_13_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_13_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_13_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_14_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_14_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_14_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_15_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_15_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_15_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_16_f
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_16_m
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_16_cya
Conversion:
                          level number
                            Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: ext_17_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_17_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_17_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_18_f
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_18_m
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----



-----
Factor: ext_18_cya
Conversion:
                         level number
                           Not      1
 Somewhat/sometimes-Very/often      2
-----
```

```
-----
Factor: int_1_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_1_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_1_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_2_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_2_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_2_cya
Conversion:
              level number
```

```
               Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_3_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_3_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_3_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_4_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_4_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_4_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_5_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_5_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_5_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_6_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_6_m
```

```
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_6_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_7_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_7_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_7_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_8_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
```

```
        Very/often      3
-----



-----
Factor: int_8_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: int_8_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: int_9_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: int_9_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: int_9_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----
```

```
-----
Factor: int_10_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_10_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_10_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_11_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_11_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: int_11_cya
Conversion:
              level number
```

```
               Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: int_12_f
Conversion:
             level number
               Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: int_12_m
Conversion:
             level number
               Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: int_12_cya
Conversion:
             level number
               Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: int_13_f
Conversion:
             level number
               Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: int_13_m
Conversion:
             level number
               Not      1
 Somewhat/sometimes      2
        Very/often      3
-----
```

```
-----
Factor: int_13_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_14_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_14_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_14_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_15_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: int_15_m
```

```
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----


-----
Factor: int_15_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----


-----
Factor: int_16_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----


-----
Factor: int_16_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----


-----
Factor: int_16_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----


-----
Factor: int_17_f
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
```

```
        Very/often      3
-----



-----
Factor: int_17_m
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: int_17_cya
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----

The file(s)
 'data_df_long_collapsed_mplus.dat'
currently exist(s) and will be overwritten

TITLE: Your title goes here
DATA: FILE = "Mplus/data_df_long_collapsed_mplus.dat";
VARIABLE:
NAMES = ID Age_cat ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya ext_3_f
     ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m ext_5_cya ext_6_f
     ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya ext_8_f ext_8_m ext_8_cya ext_9_f
     ext_9_m ext_9_cya ext_10_f ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya
     ext_12_f ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f ext_14_m
     ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m ext_16_cya ext_17_f
     ext_17_m ext_17_cya ext_18_f ext_18_m ext_18_cya int_1_f int_1_m int_1_cya
     int_2_f int_2_m int_2_cya int_3_f int_3_m int_3_cya int_4_f int_4_m int_4_cya
     int_5_f int_5_m int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m int_7_cya
     int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya int_10_f int_10_m
     int_10_cya int_11_f int_11_m int_11_cya int_12_f int_12_m int_12_cya int_13_f
     int_13_m int_13_cya int_14_f int_14_m int_14_cya int_15_f int_15_m int_15_cya
     int_16_f int_16_m int_16_cya int_17_f int_17_m int_17_cya;
MISSING=.;
```

```
# For Veronica
#prepareMplusData(df = data_df_long_collapsed_mplus, filename =
"mplusdata/data_df_long_collapsed_mplus.dat")

# The above code also gives an output that can be used as a basis for an inp file. However, note
that the file path is too long, so you need to add a line shift somewhere in it so that Mplus won't
cut it off
```

9.3 - Running a Constrained Conditional Model with Age

The code blocks for the tri-factor models all have eval = FALSE to not have to run through all the models every time the document is knitted. They therefore have to be run through manually to update them.

```
# Takes a few minutes to run through

# Creating the code for the input file in Mplus, which defines the data set and the model.
tri_factor_fmc_ext_code  <- '
  TITLE: Constrained conditional tri-factor model for externalizing symptoms;

  DATA: FILE = "Mplus/data_df_long_collapsed_unique_mplus.dat";

  VARIABLE:

      NAMES=
      ID Age ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya
      ext_3_f ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
      ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
      ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
      ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
      ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
      ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
      ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
      ext_18_cya int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya
      int_3_f int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f
      int_5_m int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m
      int_7_cya int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya
      int_10_f int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
      int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
      int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f
      int_16_m int_16_cya int_17_f int_17_m int_17_cya;

      USEVAR=
      Age ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya ext_3_f
      ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
      ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
      ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
      ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
      ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
      ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
      ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
      ext_18_cya;

      CATEGORICAL=
      ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya ext_3_f
      ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
      ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
      ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
      ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
      ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
      ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
      ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
```

```
    ext_18_cya;

    IDVARIABLE= ID;

    MISSING=.;

ANALYSIS:

    ESTIMATOR=wlsmv;
    PARAMETERIZATION=theta;
    PROCESSORS=8;

MODEL:

       !Equality constraints in this model impose invariance
         !across reporters in thresholds and loadings;

       !Item thresholds, subject to equality constraints across reporters;
       [ext_1_f$1 ext_1_m$1 ext_1_cya$1] (it11);
       [ext_2_f$1 ext_2_m$1 ext_2_cya$1] (it12);
       [ext_3_f$1 ext_3_m$1 ext_3_cya$1] (it13);
       [ext_4_f$1 ext_4_m$1 ext_4_cya$1] (it14);
       [ext_5_f$1 ext_5_m$1 ext_5_cya$1] (it15);
       [ext_6_f$1 ext_6_m$1 ext_6_cya$1] (it16);
       [ext_7_f$1 ext_7_m$1 ext_7_cya$1] (it17);
       [ext_8_f$1 ext_8_m$1 ext_8_cya$1] (it18);
       [ext_9_f$1 ext_9_m$1 ext_9_cya$1] (it19);
       [ext_10_f$1 ext_10_m$1 ext_10_cya$1] (it110);
       [ext_11_f$1 ext_11_m$1 ext_11_cya$1] (it111);
       [ext_12_f$1 ext_12_m$1 ext_12_cya$1] (it112);
       [ext_13_f$1 ext_13_m$1 ext_13_cya$1] (it113);
       [ext_14_f$1 ext_14_m$1 ext_14_cya$1] (it114);
       [ext_15_f$1 ext_15_m$1 ext_15_cya$1] (it115);
       [ext_16_f$1 ext_16_m$1 ext_16_cya$1] (it116);
       [ext_17_f$1 ext_17_m$1 ext_17_cya$1] (it117);
       [ext_18_f$1 ext_18_m$1 ext_18_cya$1] (it118);

       !general factor;
       na BY ext_1_f* (in1);
       na BY ext_2_f* (in2);
       na BY ext_3_f* (in3);
       na BY ext_4_f* (in4);
       na BY ext_5_f* (in5);
       na BY ext_6_f* (in6);
       na BY ext_7_f* (in7);
       na BY ext_8_f* (in8);
       na BY ext_9_f* (in9);
       na BY ext_10_f* (in10);
       na BY ext_11_f* (in11);
       na BY ext_12_f* (in12);
       na BY ext_13_f* (in13);
       na BY ext_14_f* (in14);
       na BY ext_15_f* (in15);
       na BY ext_16_f* (in16);
```

```
        na BY ext_17_f* (in17);
        na BY ext_18_f* (in18);

        na BY ext_1_m* (in1);
        na BY ext_2_m* (in2);
        na BY ext_3_m* (in3);
        na BY ext_4_m* (in4);
        na BY ext_5_m* (in5);
        na BY ext_6_m* (in6);
        na BY ext_7_m* (in7);
        na BY ext_8_m* (in8);
        na BY ext_9_m* (in9);
        na BY ext_10_m* (in10);
        na BY ext_11_m* (in11);
        na BY ext_12_m* (in12);
        na BY ext_13_m* (in13);
        na BY ext_14_m* (in14);
        na BY ext_15_m* (in15);
        na BY ext_16_m* (in16);
        na BY ext_17_m* (in17);
        na BY ext_18_m* (in18);

        na BY ext_1_cya* (in1);
        na BY ext_2_cya* (in2);
        na BY ext_3_cya* (in3);
        na BY ext_4_cya* (in4);
        na BY ext_5_cya* (in5);
        na BY ext_6_cya* (in6);
        na BY ext_7_cya* (in7);
        na BY ext_8_cya* (in8);
        na BY ext_9_cya* (in9);
        na BY ext_10_cya* (in10);
        na BY ext_11_cya* (in11);
        na BY ext_12_cya* (in12);
        na BY ext_13_cya* (in13);
        na BY ext_14_cya* (in14);
        na BY ext_15_cya* (in15);
        na BY ext_16_cya* (in16);
        na BY ext_17_cya* (in17);
        na BY ext_18_cya* (in18);

        !father specific factor;
        father BY ext_1_f* (ip1);
        father BY ext_2_f* (ip2);
        father BY ext_3_f* (ip3);
        father BY ext_4_f* (ip4);
        father BY ext_5_f* (ip5);
        father BY ext_6_f* (ip6);
        father BY ext_7_f* (ip7);
        father BY ext_8_f* (ip8);
        father BY ext_9_f* (ip9);
        father BY ext_10_f* (ip10);
        father BY ext_11_f* (ip11);
        father BY ext_12_f* (ip12);
```

```
        father BY ext_13_f* (ip13);
        father BY ext_14_f* (ip14);
        father BY ext_15_f* (ip15);
        father BY ext_16_f* (ip16);
        father BY ext_17_f* (ip17);
        father BY ext_18_f* (ip18);

        !mom-specific factor;
        mother BY ext_1_m* (ip1);
        mother BY ext_2_m* (ip2);
        mother BY ext_3_m* (ip3);
        mother BY ext_4_m* (ip4);
        mother BY ext_5_m* (ip5);
        mother BY ext_6_m* (ip6);
        mother BY ext_7_m* (ip7);
        mother BY ext_8_m* (ip8);
        mother BY ext_9_m* (ip9);
        mother BY ext_10_m* (ip10);
        mother BY ext_11_m* (ip11);
        mother BY ext_12_m* (ip12);
        mother BY ext_13_m* (ip13);
        mother BY ext_14_m* (ip14);
        mother BY ext_15_m* (ip15);
        mother BY ext_16_m* (ip16);
        mother BY ext_17_m* (ip17);
        mother BY ext_18_m* (ip18);

        !child-specific factor;
        child BY ext_1_cya* (ip1);
        child BY ext_2_cya* (ip2);
        child BY ext_3_cya* (ip3);
        child BY ext_4_cya* (ip4);
        child BY ext_5_cya* (ip5);
        child BY ext_6_cya* (ip6);
        child BY ext_7_cya* (ip7);
        child BY ext_8_cya* (ip8);
        child BY ext_9_cya* (ip9);
        child BY ext_10_cya* (ip10);
        child BY ext_11_cya* (ip11);
        child BY ext_12_cya* (ip12);
        child BY ext_13_cya* (ip13);
        child BY ext_14_cya* (ip14);
        child BY ext_15_cya* (ip15);
        child BY ext_16_cya* (ip16);
        child BY ext_17_cya* (ip17);
        child BY ext_18_cya* (ip18);

        !item-specific factors;
        i1 BY ext_1_f* ext_1_m* ext_1_cya* (if1); [i1@0]; i1@1;
        i2 BY ext_2_f* ext_2_m* ext_2_cya* (if2); [i2@0]; i2@1;
        i3 BY ext_3_f* ext_3_m* ext_3_cya* (if3); [i3@0]; i3@1;
        i4 BY ext_4_f* ext_4_m* ext_4_cya* (if4); [i4@0]; i4@1;
        i5 BY ext_5_f* ext_5_m* ext_5_cya* (if5); [i5@0]; i5@1;
        i6 BY ext_6_f* ext_6_m* ext_6_cya* (if6); [i6@0]; i6@1;
```

```
        i7 BY ext_7_f* ext_7_m* ext_7_cya* (if7); [i7@0]; i7@1;
        i8 BY ext_8_f* ext_8_m* ext_8_cya* (if8); [i8@0]; i8@1;
        i9 BY ext_9_f* ext_9_m* ext_9_cya* (if9); [i9@0]; i9@1;
        i10 BY ext_10_f* ext_10_m* ext_10_cya* (if10); [i10@0]; i10@1;
        i11 BY ext_11_f* ext_11_m* ext_11_cya* (if11); [i11@0]; i11@1;
        i12 BY ext_12_f* ext_12_m* ext_12_cya* (if12); [i12@0]; i12@1;
        i13 BY ext_13_f* ext_13_m* ext_13_cya* (if13); [i13@0]; i13@1;
        i14 BY ext_14_f* ext_14_m* ext_14_cya* (if14); [i14@0]; i14@1;
        i15 BY ext_15_f* ext_15_m* ext_15_cya* (if15); [i15@0]; i15@1;
        i16 BY ext_16_f* ext_16_m* ext_16_cya* (if16); [i16@0]; i16@1;
        i17 BY ext_17_f* ext_17_m* ext_17_cya* (if17); [i17@0]; i17@1;
        i18 BY ext_18_f* ext_18_m* ext_18_cya* (if18); [i18@0]; i18@1;

        !identification/scaling constraints;
        [ na@0 ];
        [ father@0 ]; !Father is reference;
        [ mother* ];  !Mom is estimated;
        [ child* ];  !Child is estimated;

        na@1;
        father@1;  !Father is reference;
        mother*;   !Mom is estimated;
        child*;   !Child is estimated;

        !Imposing orthogonality of factors
        na WITH father@0;
        na WITH mother@0;
        na WITH child@0;
        father WITH mother@0;
        father WITH child@0;
        mother WITH child@0;

        !Imposing orthogonality between negative affect and item-specific factors;
        na WITH i1@0;
        na WITH i2@0;
        na WITH i3@0;
        na WITH i4@0;
        na WITH i5@0;
        na WITH i6@0;
        na WITH i7@0;
        na WITH i8@0;
        na WITH i9@0;
        na WITH i10@0;
        na WITH i11@0;
        na WITH i12@0;
        na WITH i13@0;
        na WITH i14@0;
        na WITH i15@0;
        na WITH i16@0;
        na WITH i17@0;
        na WITH i18@0;

        i1 WITH i2@0 i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0
        i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
```

```
        i2 WITH i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0
        i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i3 WITH i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0
        i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i4 WITH i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i5 WITH i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i6 WITH i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0
        i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i7 WITH i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0
        i17@0 i18@0 father@0 mother@0 child@0;
        i8 WITH i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        i18@0 father@0 mother@0 child@0;
        i9 WITH i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0
        father@0 mother@0 child@0;
        i10 WITH i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0
        father@0 mother@0 child@0;
        i11 WITH i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0
        mother@0 child@0;
        i12 WITH i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0
        mother@0 child@0;
        i13 WITH i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i14 WITH i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i15 WITH i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i16 WITH i17@0 i18@0 father@0 mother@0 child@0;
        i17 WITH i18@0 father@0 mother@0 child@0;
        i18 WITH father@0 mother@0 child@0;

        !Adding age as a predictor
        mother ON Age;
        child ON Age;


  OUTPUT: modindices (0);

  SAVEDATA: file=tri_factor_fmc_ext_constrained_conditional.dat;'

  # Create a file that is stored in the Mplus folder from the text specified in the mplus_model
variable above
writeLines(tri_factor_fmc_ext_code, "Mplus/tri-factor model - fmc - ext - conditional - age -
constrained.inp")

# Running the model through the input Mplus file using MplusAutomation
runModels("Mplus/tri-factor model - fmc - ext - conditional - age - constrained.inp")
  # An output file will then be saved automatically with the results
```

Extracting the fit statistics from the conditional model: This code has to be run through manually, as it is eval = FALSE

```
# Reading in the data from the output file of the tri-factor model
data <- readModels("Mplus/tri-factor model - fmc - ext - conditional - age - constrained.out")

    # Assign a starting value of NA for all the fit indices
```

```
    chi_sq <- chi_sq_df <- chi_sq_p <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting chi-square stats from data$summaries form the output file
      chi_sq <- data$summaries$ChiSqM_Value
      chi_sq_df <- data$summaries$ChiSqM_DF
      chi_sq_p <- data$summaries$ChiSqM_PValue
      # Provides the number of warnings in the output file
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model
didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error
block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The
values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # Assign a starting value of NA for the warning variable
    warn_list <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting warnings from data$warnings form the output file
      # List of all the warnings in the output file
      warn_list <- data$warnings[1]
      # Use [1] after because of some weird nesting of the values
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model
didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error
block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The
values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # List of all the errors in the output file
    error_list <- data$errors[1]
      # Use [1] after because of some weird nesting of the values

    # Create a temporary data file with all the values gathered above for the current iteration of
the subscale to add to the overall data frame.
    tri_fac_fit_ext_df <- data.frame(i, chi_sq, chi_sq_df, chi_sq_p, I(warn_list), I(error_list))
      # I() means to treat the object as it is, so R keeps the list as a list when adding it to the
data frame
    # Changing the column names so that they are always the same, regardless of whether the
variable names are used (when there are no fit indices) or the name of the extracted value is (when
there are fit indices). This is to facilitate binding the data frame with the overall data frame
without getting any errors matching up the columns.
    colnames(tri_fac_fit_ext_df) <- c("Item", "chi_sq", "chi_sq_df", "chi_sq_p", "warnings",
"errors")


# Print the data frame with the results of the tri-factor models
print(tri_fac_fit_ext_df)
```

Looking at any errors from fitting the model:

```
tri_fac_fit_ext_df$errors
```

Looking at any warnings from fitting the model:

```
tri_fac_fit_ext_df$warnings
```

I ran into a problem later with the capacity by sprintf(), so I will try a different approaches using a function and sprintf() for subparts of the function to make the model template automatically. I am testing it out on the above unconditional model so that I know what the values should be for the model if it is done correctly. After testing several versions, I found that I code make two separate templates with sprintf() which are then pasted together as a workaround.

Internalizing symptoms: Run through manually as eval = FALSE

```
# Takes a few minutes to run through

# Creating the code for the input file in Mplus, which defines the data set and the model.
tri_factor_fmc_int_code  <- '
  TITLE: Constrained conditional tri-factor model for internalizing symptoms;

  DATA: FILE = "Mplus/data_df_long_collapsed_unique_mplus.dat";

  VARIABLE:

      NAMES=
      ID Age ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya
      ext_3_f ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
      ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
      ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
      ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
      ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
      ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
      ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
      ext_18_cya int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya
      int_3_f int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f
      int_5_m int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m
      int_7_cya int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya
      int_10_f int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
      int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
      int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f
      int_16_m int_16_cya int_17_f int_17_m int_17_cya;

      USEVAR=
      Age int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya int_3_f
      int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f int_5_m
      int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m int_7_cya
      int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya int_10_f
      int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
      int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
      int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f int_16_m
      int_16_cya int_17_f int_17_m int_17_cya;

      CATEGORICAL=
```

```
    int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya int_3_f
    int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f int_5_m
    int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m int_7_cya
    int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya int_10_f
    int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
    int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
    int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f int_16_m
    int_16_cya int_17_f int_17_m int_17_cya;

    IDVARIABLE= ID;

    MISSING=.;

ANALYSIS:

    ESTIMATOR=wlsmv;
    PARAMETERIZATION=theta;
    PROCESSORS=8;

MODEL:

       !Equality constraints in this model impose invariance
         !across reporters in thresholds and loadings;

       !Item thresholds, subject to equality constraints across reporters;
       [int_1_f$1 int_1_m$1 int_1_cya$1] (it11);
       [int_2_f$1 int_2_m$1 int_2_cya$1] (it12);
       [int_3_f$1 int_3_m$1 int_3_cya$1] (it13);
       [int_4_f$1 int_4_m$1 int_4_cya$1] (it14);
       [int_5_f$1 int_5_m$1 int_5_cya$1] (it15);
       [int_6_f$1 int_6_m$1 int_6_cya$1] (it16);
       [int_7_f$1 int_7_m$1 int_7_cya$1] (it17);
       [int_8_f$1 int_8_m$1 int_8_cya$1] (it18);
       [int_9_f$1 int_9_m$1 int_9_cya$1] (it19);
       [int_10_f$1 int_10_m$1 int_10_cya$1] (it110);
       [int_11_f$1 int_11_m$1 int_11_cya$1] (it111);
       [int_12_f$1 int_12_m$1 int_12_cya$1] (it112);
       [int_13_f$1 int_13_m$1 int_13_cya$1] (it113);
       [int_14_f$1 int_14_m$1 int_14_cya$1] (it114);
       [int_15_f$1 int_15_m$1 int_15_cya$1] (it115);
       [int_16_f$1 int_16_m$1 int_16_cya$1] (it116);
       [int_17_f$1 int_17_m$1 int_17_cya$1] (it117);

       [int_1_f$2 int_1_m$2 int_1_cya$2] (it21);
       [int_2_f$2 int_2_m$2 int_2_cya$2] (it22);
       [int_3_f$2 int_3_m$2 int_3_cya$2] (it23);
       [int_4_f$2 int_4_m$2 int_4_cya$2] (it24);
       [int_5_f$2 int_5_m$2 int_5_cya$2] (it25);
       [int_6_f$2 int_6_m$2 int_6_cya$2] (it26);
       [int_7_f$2 int_7_m$2 int_7_cya$2] (it27);
       [int_8_f$2 int_8_m$2 int_8_cya$2] (it28);
       [int_9_f$2 int_9_m$2 int_9_cya$2] (it29);
       [int_10_f$2 int_10_m$2 int_10_cya$2] (it210);
       [int_11_f$2 int_11_m$2 int_11_cya$2] (it211);
```

```
        [int_12_f$2 int_12_m$2 int_12_cya$2] (it212);
        [int_13_f$2 int_13_m$2 int_13_cya$2] (it213);
        [int_14_f$2 int_14_m$2 int_14_cya$2] (it214);
        [int_15_f$2 int_15_m$2 int_15_cya$2] (it215);
        [int_16_f$2 int_16_m$2 int_16_cya$2] (it216);
        [int_17_f$2 int_17_m$2 int_17_cya$2] (it217);

        !general factor;
        na BY int_1_f* (in1);
        na BY int_2_f* (in2);
        na BY int_3_f* (in3);
        na BY int_4_f* (in4);
        na BY int_5_f* (in5);
        na BY int_6_f* (in6);
        na BY int_7_f* (in7);
        na BY int_8_f* (in8);
        na BY int_9_f* (in9);
        na BY int_10_f* (in10);
        na BY int_11_f* (in11);
        na BY int_12_f* (in12);
        na BY int_13_f* (in13);
        na BY int_14_f* (in14);
        na BY int_15_f* (in15);
        na BY int_16_f* (in16);
        na BY int_17_f* (in17);

        na BY int_1_m* (in1);
        na BY int_2_m* (in2);
        na BY int_3_m* (in3);
        na BY int_4_m* (in4);
        na BY int_5_m* (in5);
        na BY int_6_m* (in6);
        na BY int_7_m* (in7);
        na BY int_8_m* (in8);
        na BY int_9_m* (in9);
        na BY int_10_m* (in10);
        na BY int_11_m* (in11);
        na BY int_12_m* (in12);
        na BY int_13_m* (in13);
        na BY int_14_m* (in14);
        na BY int_15_m* (in15);
        na BY int_16_m* (in16);
        na BY int_17_m* (in17);

        na BY int_1_cya* (in1);
        na BY int_2_cya* (in2);
        na BY int_3_cya* (in3);
        na BY int_4_cya* (in4);
        na BY int_5_cya* (in5);
        na BY int_6_cya* (in6);
        na BY int_7_cya* (in7);
        na BY int_8_cya* (in8);
        na BY int_9_cya* (in9);
        na BY int_10_cya* (in10);
```

```
na BY int_11_cya* (in11);
na BY int_12_cya* (in12);
na BY int_13_cya* (in13);
na BY int_14_cya* (in14);
na BY int_15_cya* (in15);
na BY int_16_cya* (in16);
na BY int_17_cya* (in17);

!father specific factor;
father BY int_1_f* (ip1);
father BY int_2_f* (ip2);
father BY int_3_f* (ip3);
father BY int_4_f* (ip4);
father BY int_5_f* (ip5);
father BY int_6_f* (ip6);
father BY int_7_f* (ip7);
father BY int_8_f* (ip8);
father BY int_9_f* (ip9);
father BY int_10_f* (ip10);
father BY int_11_f* (ip11);
father BY int_12_f* (ip12);
father BY int_13_f* (ip13);
father BY int_14_f* (ip14);
father BY int_15_f* (ip15);
father BY int_16_f* (ip16);
father BY int_17_f* (ip17);

!mom-specific factor;
mother BY int_1_m* (ip1);
mother BY int_2_m* (ip2);
mother BY int_3_m* (ip3);
mother BY int_4_m* (ip4);
mother BY int_5_m* (ip5);
mother BY int_6_m* (ip6);
mother BY int_7_m* (ip7);
mother BY int_8_m* (ip8);
mother BY int_9_m* (ip9);
mother BY int_10_m* (ip10);
mother BY int_11_m* (ip11);
mother BY int_12_m* (ip12);
mother BY int_13_m* (ip13);
mother BY int_14_m* (ip14);
mother BY int_15_m* (ip15);
mother BY int_16_m* (ip16);
mother BY int_17_m* (ip17);

!child-specific factor;
child BY int_1_cya* (ip1);
child BY int_2_cya* (ip2);
child BY int_3_cya* (ip3);
child BY int_4_cya* (ip4);
child BY int_5_cya* (ip5);
child BY int_6_cya* (ip6);
child BY int_7_cya* (ip7);
```

```
        child BY int_8_cya* (ip8);
        child BY int_9_cya* (ip9);
        child BY int_10_cya* (ip10);
        child BY int_11_cya* (ip11);
        child BY int_12_cya* (ip12);
        child BY int_13_cya* (ip13);
        child BY int_14_cya* (ip14);
        child BY int_15_cya* (ip15);
        child BY int_16_cya* (ip16);
        child BY int_17_cya* (ip17);

        !item-specific factors;
        i1 BY int_1_f* int_1_m* int_1_cya* (if1); [i1@0]; i1@1;
        i2 BY int_2_f* int_2_m* int_2_cya* (if2); [i2@0]; i2@1;
        i3 BY int_3_f* int_3_m* int_3_cya* (if3); [i3@0]; i3@1;
        i4 BY int_4_f* int_4_m* int_4_cya* (if4); [i4@0]; i4@1;
        i5 BY int_5_f* int_5_m* int_5_cya* (if5); [i5@0]; i5@1;
        i6 BY int_6_f* int_6_m* int_6_cya* (if6); [i6@0]; i6@1;
        i7 BY int_7_f* int_7_m* int_7_cya* (if7); [i7@0]; i7@1;
        i8 BY int_8_f* int_8_m* int_8_cya* (if8); [i8@0]; i8@1;
        i9 BY int_9_f* int_9_m* int_9_cya* (if9); [i9@0]; i9@1;
        i10 BY int_10_f* int_10_m* int_10_cya* (if10); [i10@0]; i10@1;
        i11 BY int_11_f* int_11_m* int_11_cya* (if11); [i11@0]; i11@1;
        i12 BY int_12_f* int_12_m* int_12_cya* (if12); [i12@0]; i12@1;
        i13 BY int_13_f* int_13_m* int_13_cya* (if13); [i13@0]; i13@1;
        i14 BY int_14_f* int_14_m* int_14_cya* (if14); [i14@0]; i14@1;
        i15 BY int_15_f* int_15_m* int_15_cya* (if15); [i15@0]; i15@1;
        i16 BY int_16_f* int_16_m* int_16_cya* (if16); [i16@0]; i16@1;
        i17 BY int_17_f* int_17_m* int_17_cya* (if17); [i17@0]; i17@1;

        !identification/scaling constraints;
        [ na@0 ];
        [ father@0 ]; !Father is reference;
        [ mother* ];  !Mom is estimated;
        [ child* ];  !Child is estimated;

        na@1;
        father@1;  !Father is reference;
        mother*;   !Mom is estimated;
        child*;   !Child is estimated;

        !Imposing orthogonality of factors
        na WITH father@0;
        na WITH mother@0;
        na WITH child@0;
        father WITH mother@0;
        father WITH child@0;
        mother WITH child@0;

        !Imposing orthogonality between negative affect and item-specific factors;
        na WITH i1@0;
        na WITH i2@0;
        na WITH i3@0;
        na WITH i4@0;
```

```
        na WITH i5@0;
        na WITH i6@0;
        na WITH i7@0;
        na WITH i8@0;
        na WITH i9@0;
        na WITH i10@0;
        na WITH i11@0;
        na WITH i12@0;
        na WITH i13@0;
        na WITH i14@0;
        na WITH i15@0;
        na WITH i16@0;
        na WITH i17@0;

        i1 WITH i2@0 i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0
        i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i2 WITH i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0
        i13@0 i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i3 WITH i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0
        i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i4 WITH i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i5 WITH i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i6 WITH i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0
        i16@0 i17@0 father@0 mother@0 child@0;
        i7 WITH i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0
        i17@0 father@0 mother@0 child@0;
        i8 WITH i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 mother@0 child@0;
        i9 WITH i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 mother@0 child@0;
        i10 WITH i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 mother@0 child@0;
        i11 WITH i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 father@0
        mother@0 child@0;
        i12 WITH i13@0 i14@0 i15@0 i16@0 i17@0 father@0
        mother@0 child@0;
        i13 WITH i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i14 WITH i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i15 WITH i16@0 i17@0 father@0 mother@0 child@0;
        i16 WITH i17@0 father@0 mother@0 child@0;
        i17 WITH father@0 mother@0 child@0;

        !Adding age as a predictor
        mother ON Age;
        child ON Age;


 OUTPUT: modindices (0);

 SAVEDATA: file=tri_factor_fmc_int_constrained_conditional.dat;'

 # Create a file that is stored in the Mplus folder from the text specified in the mplus_model
```

```
variable above
writeLines(tri_factor_fmc_int_code, "Mplus/tri-factor model - fmc - int - conditional - age -
constrained.inp")

# Running the model through the input Mplus file using MplusAutomation
runModels("Mplus/tri-factor model - fmc - int - conditional - age - constrained.inp")
  # An output file will then be saved automatically with the results
```

Extracting the fit statistics from the conditional model: This code has to be run through manually, as it is eval = FALSE

```
# Reading in the data from the output file of the tri-factor model
data <- readModels("Mplus/tri-factor model - fmc - int - conditional - age - constrained.out")

    # Assign a starting value of NA for all the fit indices
    chi_sq <- chi_sq_df <- chi_sq_p <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting chi-square stats from data$summaries form the output file
      chi_sq <- data$summaries$ChiSqM_Value
      chi_sq_df <- data$summaries$ChiSqM_DF
      chi_sq_p <- data$summaries$ChiSqM_PValue
      # Provides the number of warnings in the output file
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model
didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error
block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The
values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # Assign a starting value of NA for the warning variable
    warn_list <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting warnings from data$warnings form the output file
      # List of all the warnings in the output file
      warn_list <- data$warnings[1]
      # Use [1] after because of some weird nesting of the values
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model
didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error
block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The
values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # List of all the errors in the output file
    error_list <- data$errors[1]
      # Use [1] after because of some weird nesting of the values

    # Create a temporary data file with all the values gathered above for the current iteration of
the subscale to add to the overall data frame.
    tri_fac_fit_int_df <- data.frame(i, chi_sq, chi_sq_df, chi_sq_p, I(warn_list), I(error_list))
```

```
      # I() means to treat the object as it is, so R keeps the list as a list when adding it to the data frame
    # Changing the column names so that they are always the same, regardless of whether the variable names are used (when there are no fit indices) or the name of the extracted value is (when there are fit indices). This is to facilitate binding the data frame with the overall data frame without getting any errors matching up the columns.
    colnames(tri_fac_fit_int_df) <- c("Item", "chi_sq", "chi_sq_df", "chi_sq_p", "warnings", "errors")


# Print the data frame with the results of the tri-factor models
print(tri_fac_fit_int_df)
```

Looking at any errors from fitting the model:

```
tri_fac_fit_int_df$errors
```

Looking at any warnings from fitting the model:

```
tri_fac_fit_int_df$errors
```

9.4 - Likelihood testing of the Conditional Model with Age

Freeing general and rater parameters per item Running likelihood testing in Mplus with the Difftest function. Running five difftests: one test for each item The constrained conditional model from 9.3 was compared to an unconstrained model with one item freed for the raters and the general factor All the p-values for the chi-square test was .0000, indicating that the restrictions are worsening the model fit significantly - Check this with PAC data

The nuts and bolts: The unrestricted model for each item was saved with save and difftest function. Then five different files were ran with the restricted model (the general model from 9.3) and the difftest option under analysis for each unrestricted model (per item) from earlier. The unrestricted model files are stored as “unconditional tri-factor model - fmc - likelihood testing dataset - free item x” and the files for the restricted model with the difftests are saved as “unconditional tri-factor model - fmc - likelihood testing - freeing item x”

Running the likelihood testing in Mplus with Mplusautomation:

9.4.1 - Saving Data from Running Unconstrained Conditional Models with Age to use for Later Diff Tests

Prepping for the likelihood testing: Creating five different vectors, all having every fifth element as empty. The vectors have the different constraints that are used in the tri-factor model for the items, with each vector having one item freed. This will be used in the sprintf() function when making the likelihood testing data sets with one item freed

```
# Creating a vector with values "(in1)", "(in2)", "(in3)", "(in4)", "(in5)"... up until 18 for a constrain on each item
in_vect <- paste0("(in", 1:18, ")")
# Creating a vector with values "(ip1)", "(ip2)", "(ip3)", "(ip4)", "(ip5)"... up until 18 for a constrain on each item
ip_vect <- paste0("(ip", 1:18, ")")

# Making an empty list to add the vectors freeing one item each to
  # 3 is for the number of raters
```

```
freed_item_each_list_3 <- list()

# Iterating through number to the number of items there are
for (i in 1:18){
  # Adding the two vectors to each other
  freed_item_vect_3 <- c(rep(in_vect, times = 3), rep(ip_vect, times = 3))
    # 4 as there are 3 raters
  # Changing the items that should be freed to ""
  freed_item_vect_3[freed_item_vect_3 == in_vect[i] | freed_item_vect_3 == ip_vect[i]] <- ""
  # Adding the current vector to the list
  freed_item_each_list_3[[i]] <- freed_item_vect_3
}
```

Saving data sets with one item freed: As preparation for running the likelihood testing with, to compare these data sets with one item freed to the ones from the restricted tri-factor model

The code for the likelihood testing originally used sprintf() with a variable for each factor that is changed for the 18 items. However, it ended up giving sprintf() 110 arguments, when the max limit is 100. Therefore, I had to switch out bigger chunks at a time in the model. It makes it a little harder to read, but it is the easiest way of working around the max number of arguments problem. I tried switching out a part of it as one variable, but then I got an error about characters.

This code fits a data set with the fit of the unconstrained models, having one item freed. This data set will then be used to do the likelihood testing. I have divided up the sprintf() in two and pasting the code for the model together. That way, we get around the item and character limits. The code has to be run manually as it is eval = FALSE

```
# It takes about 35 minutes to run through this code and fit the 18 models

# Creating the code for the input file in Mplus, which defines the data set and the model. The %s
and %d are placeholders that will be switched out with values by the sprintf() function. In
addition, the template will be divided into two sections and pasted together, since sprintf()
doesn't have the capacity for more than 100 parameters

tri_factor_likelihood_fmc_ext_code_1  <- '
  TITLE: Conditional tri-factor model test freeing up item %d
(unrestricted model) for likelihood testing for externalizing symptoms;

DATA: FILE = "Mplus/data_df_long_collapsed_unique_mplus.dat";

  VARIABLE:

      NAMES=
      ID Age ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya
      ext_3_f ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
      ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
      ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
      ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
      ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
      ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
      ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
      ext_18_cya int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya
      int_3_f int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f
      int_5_m int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m
      int_7_cya int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya
      int_10_f int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
```

```
      int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
      int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f
      int_16_m int_16_cya int_17_f int_17_m int_17_cya;

      USEVAR=
      Age ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya ext_3_f
      ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
      ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
      ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
      ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
      ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
      ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
      ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
      ext_18_cya;

      CATEGORICAL=
      ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya ext_3_f
      ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
      ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
      ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
      ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
      ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
      ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
      ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
      ext_18_cya;

      IDVARIABLE= ID;

      MISSING=.;

ANALYSIS:

    ESTIMATOR=wlsmv;
    PARAMETERIZATION=theta;
    PROCESSORS=8;

MODEL:

      !Equality constraints in this model impose invariance
        !across reporters in thresholds and loadings;

      !Item thresholds, subject to equality constraints across reporters;
      [ext_1_f$1 ext_1_m$1 ext_1_cya$1] (it11);
      [ext_2_f$1 ext_2_m$1 ext_2_cya$1] (it12);
      [ext_3_f$1 ext_3_m$1 ext_3_cya$1] (it13);
      [ext_4_f$1 ext_4_m$1 ext_4_cya$1] (it14);
      [ext_5_f$1 ext_5_m$1 ext_5_cya$1] (it15);
      [ext_6_f$1 ext_6_m$1 ext_6_cya$1] (it16);
      [ext_7_f$1 ext_7_m$1 ext_7_cya$1] (it17);
      [ext_8_f$1 ext_8_m$1 ext_8_cya$1] (it18);
      [ext_9_f$1 ext_9_m$1 ext_9_cya$1] (it19);
      [ext_10_f$1 ext_10_m$1 ext_10_cya$1] (it110);
      [ext_11_f$1 ext_11_m$1 ext_11_cya$1] (it111);
      [ext_12_f$1 ext_12_m$1 ext_12_cya$1] (it112);
```

```
    [ext_13_f$1 ext_13_m$1 ext_13_cya$1] (it113);
    [ext_14_f$1 ext_14_m$1 ext_14_cya$1] (it114);
    [ext_15_f$1 ext_15_m$1 ext_15_cya$1] (it115);
    [ext_16_f$1 ext_16_m$1 ext_16_cya$1] (it116);
    [ext_17_f$1 ext_17_m$1 ext_17_cya$1] (it117);
    [ext_18_f$1 ext_18_m$1 ext_18_cya$1] (it118);

    !general factor;
       na BY ext_1_f* %s;
       na BY ext_2_f* %s;
       na BY ext_3_f* %s;
       na BY ext_4_f* %s;
       na BY ext_5_f* %s;
       na BY ext_6_f* %s;
       na BY ext_7_f* %s;
       na BY ext_8_f* %s;
       na BY ext_9_f* %s;
       na BY ext_10_f* %s;
       na BY ext_11_f* %s;
       na BY ext_12_f* %s;
       na BY ext_13_f* %s;
       na BY ext_14_f* %s;
       na BY ext_15_f* %s;
       na BY ext_16_f* %s;
       na BY ext_17_f* %s;
       na BY ext_18_f* %s;

       na BY ext_1_m* %s;
       na BY ext_2_m* %s;
       na BY ext_3_m* %s;
       na BY ext_4_m* %s;
       na BY ext_5_m* %s;
       na BY ext_6_m* %s;
       na BY ext_7_m* %s;
       na BY ext_8_m* %s;
       na BY ext_9_m* %s;
       na BY ext_10_m* %s;
       na BY ext_11_m* %s;
       na BY ext_12_m* %s;
       na BY ext_13_m* %s;
       na BY ext_14_m* %s;
       na BY ext_15_m* %s;
       na BY ext_16_m* %s;
       na BY ext_17_m* %s;
       na BY ext_18_m* %s;

       na BY ext_1_cya* %s;
       na BY ext_2_cya* %s;
       na BY ext_3_cya* %s;
       na BY ext_4_cya* %s;
       na BY ext_5_cya* %s;
       na BY ext_6_cya* %s;
       na BY ext_7_cya* %s;
       na BY ext_8_cya* %s;
```

```
        na BY ext_9_cya* %s;
        na BY ext_10_cya* %s;
        na BY ext_11_cya* %s;
        na BY ext_12_cya* %s;
        na BY ext_13_cya* %s;
        na BY ext_14_cya* %s;
        na BY ext_15_cya* %s;
        na BY ext_16_cya* %s;
        na BY ext_17_cya* %s;
        na BY ext_18_cya* %s;'

tri_factor_likelihood_fmc_ext_code_2 <- '

        !father specific factor;
        father BY ext_1_f* %s;
        father BY ext_2_f* %s;
        father BY ext_3_f* %s;
        father BY ext_4_f* %s;
        father BY ext_5_f* %s;
        father BY ext_6_f* %s;
        father BY ext_7_f* %s;
        father BY ext_8_f* %s;
        father BY ext_9_f* %s;
        father BY ext_10_f* %s;
        father BY ext_11_f* %s;
        father BY ext_12_f* %s;
        father BY ext_13_f* %s;
        father BY ext_14_f* %s;
        father BY ext_15_f* %s;
        father BY ext_16_f* %s;
        father BY ext_17_f* %s;
        father BY ext_18_f* %s;

        !mom-specific factor;
        mother BY ext_1_m* %s;
        mother BY ext_2_m* %s;
        mother BY ext_3_m* %s;
        mother BY ext_4_m* %s;
        mother BY ext_5_m* %s;
        mother BY ext_6_m* %s;
        mother BY ext_7_m* %s;
        mother BY ext_8_m* %s;
        mother BY ext_9_m* %s;
        mother BY ext_10_m* %s;
        mother BY ext_11_m* %s;
        mother BY ext_12_m* %s;
        mother BY ext_13_m* %s;
        mother BY ext_14_m* %s;
        mother BY ext_15_m* %s;
        mother BY ext_16_m* %s;
        mother BY ext_17_m* %s;
        mother BY ext_18_m* %s;

        !child-specific factor;
```

```
        child BY ext_1_cya* %s;
        child BY ext_2_cya* %s;
        child BY ext_3_cya* %s;
        child BY ext_4_cya* %s;
        child BY ext_5_cya* %s;
        child BY ext_6_cya* %s;
        child BY ext_7_cya* %s;
        child BY ext_8_cya* %s;
        child BY ext_9_cya* %s;
        child BY ext_10_cya* %s;
        child BY ext_11_cya* %s;
        child BY ext_12_cya* %s;
        child BY ext_13_cya* %s;
        child BY ext_14_cya* %s;
        child BY ext_15_cya* %s;
        child BY ext_16_cya* %s;
        child BY ext_17_cya* %s;
        child BY ext_18_cya* %s;

        !item-specific factors;
        i1 BY ext_1_f* ext_1_m* ext_1_cya* (if1); [i1@0]; i1@1;
        i2 BY ext_2_f* ext_2_m* ext_2_cya* (if2); [i2@0]; i2@1;
        i3 BY ext_3_f* ext_3_m* ext_3_cya* (if3); [i3@0]; i3@1;
        i4 BY ext_4_f* ext_4_m* ext_4_cya* (if4); [i4@0]; i4@1;
        i5 BY ext_5_f* ext_5_m* ext_5_cya* (if5); [i5@0]; i5@1;
        i6 BY ext_6_f* ext_6_m* ext_6_cya* (if6); [i6@0]; i6@1;
        i7 BY ext_7_f* ext_7_m* ext_7_cya* (if7); [i7@0]; i7@1;
        i8 BY ext_8_f* ext_8_m* ext_8_cya* (if8); [i8@0]; i8@1;
        i9 BY ext_9_f* ext_9_m* ext_9_cya* (if9); [i9@0]; i9@1;
        i10 BY ext_10_f* ext_10_m* ext_10_cya* (if10); [i10@0]; i10@1;
        i11 BY ext_11_f* ext_11_m* ext_11_cya* (if11); [i11@0]; i11@1;
        i12 BY ext_12_f* ext_12_m* ext_12_cya* (if12); [i12@0]; i12@1;
        i13 BY ext_13_f* ext_13_m* ext_13_cya* (if13); [i13@0]; i13@1;
        i14 BY ext_14_f* ext_14_m* ext_14_cya* (if14); [i14@0]; i14@1;
        i15 BY ext_15_f* ext_15_m* ext_15_cya* (if15); [i15@0]; i15@1;
        i16 BY ext_16_f* ext_16_m* ext_16_cya* (if16); [i16@0]; i16@1;
        i17 BY ext_17_f* ext_17_m* ext_17_cya* (if17); [i17@0]; i17@1;
        i18 BY ext_18_f* ext_18_m* ext_18_cya* (if18); [i18@0]; i18@1;

        !identification/scaling constraints;
        [ na@0 ];
        [ father@0 ]; !Father is reference;
        [ mother* ];  !Mom is estimated;
        [ child* ];  !Child is estimated;

        na@1;
        father@1;  !Father is reference;
        mother*;   !Mom is estimated;
        child*;   !Child is estimated;

        !Imposing orthogonality of factors
        na WITH father@0;
        na WITH mother@0;
        na WITH child@0;
```

```
    father WITH mother@0;
    father WITH child@0;
    mother WITH child@0;

   !Imposing orthogonality between negative affect and item-specific factors;
    na WITH i1@0;
    na WITH i2@0;
    na WITH i3@0;
    na WITH i4@0;
    na WITH i5@0;
    na WITH i6@0;
    na WITH i7@0;
    na WITH i8@0;
    na WITH i9@0;
    na WITH i10@0;
    na WITH i11@0;
    na WITH i12@0;
    na WITH i13@0;
    na WITH i14@0;
    na WITH i15@0;
    na WITH i16@0;
    na WITH i17@0;
    na WITH i18@0;

    i1 WITH i2@0 i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0
    i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
    i2 WITH i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0
    i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
    i3 WITH i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0
    i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
    i4 WITH i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
    i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
    i5 WITH i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
    i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
    i6 WITH i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0
    i16@0 i17@0 i18@0 father@0 mother@0 child@0;
    i7 WITH i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0
    i17@0 i18@0 father@0 mother@0 child@0;
    i8 WITH i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
    i18@0 father@0 mother@0 child@0;
    i9 WITH i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0
    father@0 mother@0 child@0;
    i10 WITH i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0
    father@0 mother@0 child@0;
    i11 WITH i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0
    mother@0 child@0;
    i12 WITH i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0
    mother@0 child@0;
    i13 WITH i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
    i14 WITH i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
    i15 WITH i16@0 i17@0 i18@0 father@0 mother@0 child@0;
    i16 WITH i17@0 i18@0 father@0 mother@0 child@0;
    i17 WITH i18@0 father@0 mother@0 child@0;
    i18 WITH father@0 mother@0 child@0;
```

```
         !Adding age as a predictor
         mother ON Age;
         child ON Age;
         i%s ON Age;


OUTPUT: modindices (0);

SAVEDATA: DIFFTEST=tri_factor_fmc_ext_diff_test_freeing_item%d.dat;'

# Iterating through each item
for (i in 1:18) {
    # Creating the text specifying the data set and tri-factor model for the current item (by
changing out the placeholders to match the current item).
  # Putting in values for the placeholders for the first part of the model template
    mplus_model_1 <- sprintf(tri_factor_likelihood_fmc_ext_code_1, i, freed_item_each_list_3[[i]]
[1], freed_item_each_list_3[[i]][2], freed_item_each_list_3[[i]][3], freed_item_each_list_3[[i]]
[4], freed_item_each_list_3[[i]][5], freed_item_each_list_3[[i]][6], freed_item_each_list_3[[i]]
[7], freed_item_each_list_3[[i]][8], freed_item_each_list_3[[i]][9], freed_item_each_list_3[[i]]
[10], freed_item_each_list_3[[i]][11], freed_item_each_list_3[[i]][12], freed_item_each_list_3[[i]]
[13], freed_item_each_list_3[[i]][14], freed_item_each_list_3[[i]][15], freed_item_each_list_3[[i]]
[16], freed_item_each_list_3[[i]][17], freed_item_each_list_3[[i]][18], freed_item_each_list_3[[i]]
[19], freed_item_each_list_3[[i]][20], freed_item_each_list_3[[i]][21], freed_item_each_list_3[[i]]
[22], freed_item_each_list_3[[i]][23], freed_item_each_list_3[[i]][24], freed_item_each_list_3[[i]]
[25], freed_item_each_list_3[[i]][26], freed_item_each_list_3[[i]][27], freed_item_each_list_3[[i]]
[28], freed_item_each_list_3[[i]][29], freed_item_each_list_3[[i]][30], freed_item_each_list_3[[i]]
[31], freed_item_each_list_3[[i]][32], freed_item_each_list_3[[i]][33], freed_item_each_list_3[[i]]
[34], freed_item_each_list_3[[i]][35], freed_item_each_list_3[[i]][36], freed_item_each_list_3[[i]]
[37], freed_item_each_list_3[[i]][38], freed_item_each_list_3[[i]][39], freed_item_each_list_3[[i]]
[40], freed_item_each_list_3[[i]][41], freed_item_each_list_3[[i]][42], freed_item_each_list_3[[i]]
[43], freed_item_each_list_3[[i]][44], freed_item_each_list_3[[i]][45], freed_item_each_list_3[[i]]
[46], freed_item_each_list_3[[i]][47], freed_item_each_list_3[[i]][48], freed_item_each_list_3[[i]]
[49], freed_item_each_list_3[[i]][50], freed_item_each_list_3[[i]][51], freed_item_each_list_3[[i]]
[52], freed_item_each_list_3[[i]][53], freed_item_each_list_3[[i]][54])
      # Used the following code to make the indexed variables with freed_item_each_list
paste0("freed_item_each_list_3[[i]][", 1:108, "]")

    # Putting in values for the placeholders for the second part of the model template
    mplus_model_2 <- sprintf(tri_factor_likelihood_fmc_ext_code_2, freed_item_each_list_3[[i]][55],
freed_item_each_list_3[[i]][56], freed_item_each_list_3[[i]][57], freed_item_each_list_3[[i]][58],
freed_item_each_list_3[[i]][59], freed_item_each_list_3[[i]][60], freed_item_each_list_3[[i]][61],
freed_item_each_list_3[[i]][62], freed_item_each_list_3[[i]][63], freed_item_each_list_3[[i]][64],
freed_item_each_list_3[[i]][65], freed_item_each_list_3[[i]][66], freed_item_each_list_3[[i]][67],
freed_item_each_list_3[[i]][68], freed_item_each_list_3[[i]][69], freed_item_each_list_3[[i]][70],
freed_item_each_list_3[[i]][71], freed_item_each_list_3[[i]][72], freed_item_each_list_3[[i]][73],
freed_item_each_list_3[[i]][74], freed_item_each_list_3[[i]][75], freed_item_each_list_3[[i]][76],
freed_item_each_list_3[[i]][77], freed_item_each_list_3[[i]][78], freed_item_each_list_3[[i]][79],
freed_item_each_list_3[[i]][80], freed_item_each_list_3[[i]][81], freed_item_each_list_3[[i]][82],
freed_item_each_list_3[[i]][83], freed_item_each_list_3[[i]][84], freed_item_each_list_3[[i]][85],
freed_item_each_list_3[[i]][86], freed_item_each_list_3[[i]][87], freed_item_each_list_3[[i]][88],
freed_item_each_list_3[[i]][89], freed_item_each_list_3[[i]][90], freed_item_each_list_3[[i]][91],
freed_item_each_list_3[[i]][92], freed_item_each_list_3[[i]][93], freed_item_each_list_3[[i]][94],
freed_item_each_list_3[[i]][95], freed_item_each_list_3[[i]][96], freed_item_each_list_3[[i]][97],
```

```
freed_item_each_list_3[[i]][98], freed_item_each_list_3[[i]][99], freed_item_each_list_3[[i]][100],
freed_item_each_list_3[[i]][101], freed_item_each_list_3[[i]][102], freed_item_each_list_3[[i]]
[103], freed_item_each_list_3[[i]][104], freed_item_each_list_3[[i]][105],
freed_item_each_list_3[[i]][106], freed_item_each_list_3[[i]][107], freed_item_each_list_3[[i]]
[108], i, i)

    # Pasting together the two parts of the model template, after having used sprintf()
    mplus_model <- paste0(mplus_model_1, mplus_model_2)

    # Create a file that is stored in the Mplus folder from the text specified in the mplus_model
variable above
    writeLines(mplus_model, sprintf("Mplus/tri-factor model - fmc - ext - likelihood testing
dataset - free item %d.inp", i))

    # Running the model through the input Mplus file using MplusAutomation
    runModels(sprintf("Mplus/tri-factor model - fmc - ext - likelihood testing dataset - free item
%d.inp", i))
      # An output file will then be saved automatically with the results.
      # Furthermore, as specified in the model text, the data from the model will be saved in a dat
file in the Mplus folder with the file name "tri_factor_fmc_ext_diff_test_freeing_item%d.dat" where
%d is the item number
  }
```

Extracting the results from likelihood test data set: This code has to be run through manually, as it is eval = FALSE

```
# Takes a few minutes to run through

# Creating an empty data frame to add the data from each iteration to
tri_fac_diff_test_data_ext_df <- data.frame(matrix(nrow = 0, ncol = 6))

# Iterating through each subscale
for (i in 1:18) {
  # Storing the name of the file path with the current subscale to a variable names file_path
    file_path <- sprintf("Mplus/tri-factor model - fmc - ext - likelihood testing dataset - free
item %d.out", i)
      # The sprintf() function takes text with a placeholder (for example %s) and adds in a value
that is provided as another argument, for example "scale_vect[i].

    # Reading in the data from the output file of the tri-factor model for the current subscale,
using the file path specified at the beginning of the code.
    data <- readModels(file_path)

    # Assign a starting value of NA for all the fit indices
    chi_sq <- chi_sq_df <- chi_sq_p <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting chi-square stats from data$summaries form the output file
      chi_sq <- data$summaries$ChiSqM_Value
      chi_sq_df <- data$summaries$ChiSqM_DF
      chi_sq_p <- data$summaries$ChiSqM_PValue
      # Provides the number of warnings in the output file
    }, error = function(e) {
```

```
      # If there is an error trying to extract the values above (which there will be if the model didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # Assign a starting value of NA for the warning variable
    warn_list <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and leave the variables with the initial value of NA
    tryCatch({
      # Extracting warnings from data$warnings form the output file
      # List of all the warnings in the output file
      warn_list <- data$warnings[1]
      # Use [1] after because of some weird nesting of the values
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # List of all the errors in the output file
    error_list <- data$errors[1]
      # Use [1] after because of some weird nesting of the values

    # Create a temporary data file with all the values gathered above for the current iteration of the subscale to add to the overall data frame.
    temp_df <- data.frame(i, chi_sq, chi_sq_df, chi_sq_p, I(warn_list), I(error_list))
      # I() means to treat the object as it is, so R keeps the list as a list when adding it to the data frame
    # Changing the column names so that they are always the same, regardless of whether the variable names are used (when there are no fit indices) or the name of the extracted value is (when there are fit indices). This is to facilitate binding the data frame with the overall data frame without getting any errors matching up the columns.
    colnames(temp_df) <- c("Item", "chi_sq", "chi_sq_df", "chi_sq_p", "warnings", "errors")

    # Bind the data frame from the current model to the overall sdq_tri_fac_stat_df data frame
    tri_fac_diff_test_data_ext_df <- rbind(tri_fac_diff_test_data_ext_df, temp_df)
  }

# Print the data frame with the results of the tri-factor models
print(tri_fac_diff_test_data_ext_df)
```

Check the errors from fitting the model:

```
print(tri_fac_diff_test_data_ext_df$errors)
```

Check the warnings from fitting the model:

```
print(tri_fac_diff_test_data_ext_df$warnings)
```

Getting parameters for each item freed: Extracting the parameters from the conditinoal tri-factor model with one item freed for externalizing symptoms: This code has to be run through manually, as it is eval = FALSE

```
# Takes a few minutes to run through

# Creating an empty data frame to add the data from each iteration to
tri_fac_diff_test_parameters_ext_df <- data.frame(matrix(nrow = 0, ncol = 7))

# Iterating through each subscale
for (i in 1:18) {
  # Storing the name of the file path with the current subscale to a variable names file_path
    file_path <- sprintf("Mplus/tri-factor model - fmc - ext - likelihood testing dataset - free item %d.out", i)
      # The sprintf() function takes text with a placeholder (for example %s) and adds in a value that is provided as another argument, for example "scale_vect[i].


    # Reading in the data from the output file of the tri-factor model for the current subscale, using the file path specified at the beginning of the code.
    data <- readModels(file_path)

    # Extracting the estimates of raters on age from the model
    temp_df <- data$parameters$unstandardized %>% filter(paramHeader == "Intercepts")
    # Extracting the estimates of rater intercepts from the model
    temp_df <- rbind(temp_df, data$parameters$unstandardized %>% filter(param == "AGE"))

    # Extracting every fifth row, starting on the current item
    #temp_df <- temp_df[seq(i, nrow(temp_df), 18), ]
      # Provides the parameter stats for only the item currently freed
    # Checking whether there are 6 columns. If not, Mplpus hasn't been able to calculate standard error estimates, which leaves out 3 columns
   # if (ncol(temp_df) != 6){
      # If the standard estimates are absent, add those three columns with a value of NA
      #temp_df$se <- temp_df$est_se <- temp_df$pval <- rep(NA, nrow(temp_df))
        # So that the data frames will have similar dimensions and therefore be able to combine them
   # }
    temp_df$item <- rep(i, nrow(temp_df))
    # Extracting the rater in question for the parameter estimate (differ per row)
    #temp_df$rater <- str_extract(temp_df$param, "[[:alpha:]]+$")
    # Reordering the columns so that the model information comes before the parameter statistics
    temp_df <- temp_df[, c("item", "paramHeader", "param", "est", "se", "est_se", "pval")]

    # Bind the data frame from the current model to the overall sdq_tri_fac_parameters_df data frame
    tri_fac_diff_test_parameters_ext_df <- rbind(tri_fac_diff_test_parameters_ext_df, temp_df)
  }

# Changing the misleading name of the est_se column to est_div_se
colnames(temp_df)[colnames(temp_df) == 'est_se'] <- 'est_div_se'

# Saving the data frame with the parameters for later reference
save(tri_fac_diff_test_parameters_ext_df, file = "tri_fac_diff_test_parameters_ext_df.Rdata")
```

```
# Print the data frame with the results of the tri-factor models
print(tri_fac_diff_test_parameters_ext_df)
```

Filtering for the intercepts:

```
tri_fac_diff_test_parameters_ext_df %>% filter(paramHeader == "Intercepts")
```

Making a plot of the intercepts for externalizing symptoms for the thesis:

```
tri_fac_diff_test_parameters_ext_df %>% filter(paramHeader == "Intercepts" & (param == "MOTHER" |
param == "CHILD")) %>%
  mutate(param = case_when(
    param == "CHILD" ~ "Children compared to fathers",
    param == "MOTHER" ~ "Mothers compared to fathers")
    ) %>%
  ggplot(aes(x = item, y = est)) +
  geom_bar(stat = "identity", fill = "grey73", color = "black", width = 0.55) +
  facet_wrap(vars(param), nrow = 2) +
  geom_errorbar(aes(ymin = est - se, ymax = est + se), width = 0.25) +
  ylim(-1.50, 0.3) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  scale_x_continuous(breaks = 1:18) +
  geom_hline(aes(yintercept = 0)) +
  labs(x = "Items",
       y = "Intercepts")
```

Filtering for the estimates on age:

```
tri_fac_diff_test_parameters_ext_df %>% filter(param == "AGE")
```

Making a plot of the estimates on age for externalizing symptoms for the thesis:

```
tri_fac_diff_test_parameters_ext_df %>% filter(param == "AGE" & (paramHeader == "MOTHER.ON" |
paramHeader == "CHILD.ON")) %>%
  mutate(paramHeader = case_when(
    paramHeader == "CHILD.ON" ~ "Children compared to fathers",
    paramHeader == "MOTHER.ON" ~ "Mothers compared to fathers")
    ) %>%
  ggplot(aes(x = item, y = est)) +
  geom_bar(stat = "identity", fill = "grey73", color = "black", width = 0.55) +
  facet_wrap(vars(paramHeader), nrow = 2) +
  geom_errorbar(aes(ymin = est - se, ymax = est + se), width = 0.25) +
  ylim(-0.15, 0.05) +
  theme_classic() +
```

```
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  scale_x_continuous(breaks = 1:18) +
  geom_hline(aes(yintercept = 0)) +
  labs(x = "Items",
       y = "Estimates on Age")
```

Internalizing symptoms:

```
# Creating a vector with values "(in1)", "(in2)", "(in3)", "(in4)", "(in5)"... up until 18 for a
constrain on each item
in_vect <- paste0("(in", 1:17, ")")
# Creating a vector with values "(ip1)", "(ip2)", "(ip3)", "(ip4)", "(ip5)"... up until 18 for a
constrain on each item
ip_vect <- paste0("(ip", 1:17, ")")

# Making an empty list to add the vectors freeing one item each to
  # 3 is for the number of raters
freed_item_each_list_3 <- list()

# Iterating through number to the number of items there are
for (i in 1:17){
  # Adding the two vectors to each other
  freed_item_vect_3 <- c(rep(in_vect, times = 3), rep(ip_vect, times = 3))
    # 4 as there are 3 raters
  # Changing the items that should be freed to ""
  freed_item_vect_3[freed_item_vect_3 == in_vect[i] | freed_item_vect_3 == ip_vect[i]] <- ""
  # Adding the current vector to the list
  freed_item_each_list_3[[i]] <- freed_item_vect_3
}
```

Likelihood testing for internalizing symptoms: This code has to be run through manually, as it is eval = FALSE

```
# It takes about 35 minutes to run through this code and fit the 17 models

# Creating the code for the input file in Mplus, which defines the data set and the model. The %s
and %d are placeholders that will be switched out with values by the sprintf() function. In
addition, the template will be divided into two sections and pasted together, since sprintf()
doesn't have the capacity for more than 100 parameters

tri_factor_likelihood_fmc_int_code_1  <- '
  TITLE: Unconditional tri-factor model test freeing up item %d
(unrestricted model) for likelihood testing for internalizing symptoms;

DATA: FILE = "Mplus/data_df_long_collapsed_unique_mplus.dat";

  VARIABLE:
```

```
      NAMES=
      ID Age ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya
      ext_3_f ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
      ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
      ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
      ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
      ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
      ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
      ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
      ext_18_cya int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya
      int_3_f int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f
      int_5_m int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m
      int_7_cya int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya
      int_10_f int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
      int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
      int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f
      int_16_m int_16_cya int_17_f int_17_m int_17_cya;

      USEVAR=
      Age int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya int_3_f
      int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f int_5_m
      int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m int_7_cya
      int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya int_10_f
      int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
      int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
      int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f int_16_m
      int_16_cya int_17_f int_17_m int_17_cya;

      CATEGORICAL=
      int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya int_3_f
      int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f int_5_m
      int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m int_7_cya
      int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya int_10_f
      int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
      int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
      int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f int_16_m
      int_16_cya int_17_f int_17_m int_17_cya;

      IDVARIABLE= ID;

      MISSING=.;

ANALYSIS:

    ESTIMATOR=wlsmv;
    PARAMETERIZATION=theta;
    PROCESSORS=8;

MODEL:

      !Equality constraints in this model impose invariance
        !across reporters in thresholds and loadings;
```

```
    !Item thresholds, subject to equality constraints across reporters;
      [int_1_f$1 int_1_m$1 int_1_cya$1] (it11);
      [int_2_f$1 int_2_m$1 int_2_cya$1] (it12);
      [int_3_f$1 int_3_m$1 int_3_cya$1] (it13);
      [int_4_f$1 int_4_m$1 int_4_cya$1] (it14);
      [int_5_f$1 int_5_m$1 int_5_cya$1] (it15);
      [int_6_f$1 int_6_m$1 int_6_cya$1] (it16);
      [int_7_f$1 int_7_m$1 int_7_cya$1] (it17);
      [int_8_f$1 int_8_m$1 int_8_cya$1] (it18);
      [int_9_f$1 int_9_m$1 int_9_cya$1] (it19);
      [int_10_f$1 int_10_m$1 int_10_cya$1] (it110);
      [int_11_f$1 int_11_m$1 int_11_cya$1] (it111);
      [int_12_f$1 int_12_m$1 int_12_cya$1] (it112);
      [int_13_f$1 int_13_m$1 int_13_cya$1] (it113);
      [int_14_f$1 int_14_m$1 int_14_cya$1] (it114);
      [int_15_f$1 int_15_m$1 int_15_cya$1] (it115);
      [int_16_f$1 int_16_m$1 int_16_cya$1] (it116);
      [int_17_f$1 int_17_m$1 int_17_cya$1] (it117);

      [int_1_f$2 int_1_m$2 int_1_cya$2] (it21);
      [int_2_f$2 int_2_m$2 int_2_cya$2] (it22);
      [int_3_f$2 int_3_m$2 int_3_cya$2] (it23);
      [int_4_f$2 int_4_m$2 int_4_cya$2] (it24);
      [int_5_f$2 int_5_m$2 int_5_cya$2] (it25);
      [int_6_f$2 int_6_m$2 int_6_cya$2] (it26);
      [int_7_f$2 int_7_m$2 int_7_cya$2] (it27);
      [int_8_f$2 int_8_m$2 int_8_cya$2] (it28);
      [int_9_f$2 int_9_m$2 int_9_cya$2] (it29);
      [int_10_f$2 int_10_m$2 int_10_cya$2] (it210);
      [int_11_f$2 int_11_m$2 int_11_cya$2] (it211);
      [int_12_f$2 int_12_m$2 int_12_cya$2] (it212);
      [int_13_f$2 int_13_m$2 int_13_cya$2] (it213);
      [int_14_f$2 int_14_m$2 int_14_cya$2] (it214);
      [int_15_f$2 int_15_m$2 int_15_cya$2] (it215);
      [int_16_f$2 int_16_m$2 int_16_cya$2] (it216);
      [int_17_f$2 int_17_m$2 int_17_cya$2] (it217);

    !general factor;
      na BY int_1_f* %s;
      na BY int_2_f* %s;
      na BY int_3_f* %s;
      na BY int_4_f* %s;
      na BY int_5_f* %s;
      na BY int_6_f* %s;
      na BY int_7_f* %s;
      na BY int_8_f* %s;
      na BY int_9_f* %s;
      na BY int_10_f* %s;
      na BY int_11_f* %s;
      na BY int_12_f* %s;
      na BY int_13_f* %s;
      na BY int_14_f* %s;
      na BY int_15_f* %s;
      na BY int_16_f* %s;
```

```
        na BY int_17_f* %s;

        na BY int_1_m* %s;
        na BY int_2_m* %s;
        na BY int_3_m* %s;
        na BY int_4_m* %s;
        na BY int_5_m* %s;
        na BY int_6_m* %s;
        na BY int_7_m* %s;
        na BY int_8_m* %s;
        na BY int_9_m* %s;
        na BY int_10_m* %s;
        na BY int_11_m* %s;
        na BY int_12_m* %s;
        na BY int_13_m* %s;
        na BY int_14_m* %s;
        na BY int_15_m* %s;
        na BY int_16_m* %s;
        na BY int_17_m* %s;

        na BY int_1_cya* %s;
        na BY int_2_cya* %s;
        na BY int_3_cya* %s;
        na BY int_4_cya* %s;
        na BY int_5_cya* %s;
        na BY int_6_cya* %s;
        na BY int_7_cya* %s;
        na BY int_8_cya* %s;
        na BY int_9_cya* %s;
        na BY int_10_cya* %s;
        na BY int_11_cya* %s;
        na BY int_12_cya* %s;
        na BY int_13_cya* %s;
        na BY int_14_cya* %s;
        na BY int_15_cya* %s;
        na BY int_16_cya* %s;
        na BY int_17_cya* %s;'

tri_factor_likelihood_fmc_int_code_2 <- '

        !father specific factor;
        father BY int_1_f* %s;
        father BY int_2_f* %s;
        father BY int_3_f* %s;
        father BY int_4_f* %s;
        father BY int_5_f* %s;
        father BY int_6_f* %s;
        father BY int_7_f* %s;
        father BY int_8_f* %s;
        father BY int_9_f* %s;
        father BY int_10_f* %s;
        father BY int_11_f* %s;
        father BY int_12_f* %s;
        father BY int_13_f* %s;
```

```
        father BY int_14_f* %s;
        father BY int_15_f* %s;
        father BY int_16_f* %s;
        father BY int_17_f* %s;

        !mom-specific factor;
        mother BY int_1_m* %s;
        mother BY int_2_m* %s;
        mother BY int_3_m* %s;
        mother BY int_4_m* %s;
        mother BY int_5_m* %s;
        mother BY int_6_m* %s;
        mother BY int_7_m* %s;
        mother BY int_8_m* %s;
        mother BY int_9_m* %s;
        mother BY int_10_m* %s;
        mother BY int_11_m* %s;
        mother BY int_12_m* %s;
        mother BY int_13_m* %s;
        mother BY int_14_m* %s;
        mother BY int_15_m* %s;
        mother BY int_16_m* %s;
        mother BY int_17_m* %s;

        !child-specific factor;
        child BY int_1_cya* %s;
        child BY int_2_cya* %s;
        child BY int_3_cya* %s;
        child BY int_4_cya* %s;
        child BY int_5_cya* %s;
        child BY int_6_cya* %s;
        child BY int_7_cya* %s;
        child BY int_8_cya* %s;
        child BY int_9_cya* %s;
        child BY int_10_cya* %s;
        child BY int_11_cya* %s;
        child BY int_12_cya* %s;
        child BY int_13_cya* %s;
        child BY int_14_cya* %s;
        child BY int_15_cya* %s;
        child BY int_16_cya* %s;
        child BY int_17_cya* %s;

        !item-specific factors;
        i1 BY int_1_f* int_1_m* int_1_cya* (if1); [i1@0]; i1@1;
        i2 BY int_2_f* int_2_m* int_2_cya* (if2); [i2@0]; i2@1;
        i3 BY int_3_f* int_3_m* int_3_cya* (if3); [i3@0]; i3@1;
        i4 BY int_4_f* int_4_m* int_4_cya* (if4); [i4@0]; i4@1;
        i5 BY int_5_f* int_5_m* int_5_cya* (if5); [i5@0]; i5@1;
        i6 BY int_6_f* int_6_m* int_6_cya* (if6); [i6@0]; i6@1;
        i7 BY int_7_f* int_7_m* int_7_cya* (if7); [i7@0]; i7@1;
        i8 BY int_8_f* int_8_m* int_8_cya* (if8); [i8@0]; i8@1;
        i9 BY int_9_f* int_9_m* int_9_cya* (if9); [i9@0]; i9@1;
        i10 BY int_10_f* int_10_m* int_10_cya* (if10); [i10@0]; i10@1;
```

```
        i11 BY int_11_f* int_11_m* int_11_cya* (if11); [i11@0]; i11@1;
        i12 BY int_12_f* int_12_m* int_12_cya* (if12); [i12@0]; i12@1;
        i13 BY int_13_f* int_13_m* int_13_cya* (if13); [i13@0]; i13@1;
        i14 BY int_14_f* int_14_m* int_14_cya* (if14); [i14@0]; i14@1;
        i15 BY int_15_f* int_15_m* int_15_cya* (if15); [i15@0]; i15@1;
        i16 BY int_16_f* int_16_m* int_16_cya* (if16); [i16@0]; i16@1;
        i17 BY int_17_f* int_17_m* int_17_cya* (if17); [i17@0]; i17@1;


        !identification/scaling constraints;
        [ na@0 ];
        [ father@0 ]; !Father is reference;
        [ mother* ];  !Mom is estimated;
        [ child* ];  !Child is estimated;

        na@1;
        father@1;  !Father is reference;
        mother*;   !Mom is estimated;
        child*;   !Child is estimated;

        !Imposing orthogonality of factors
        na WITH father@0;
        na WITH mother@0;
        na WITH child@0;
        father WITH mother@0;
        father WITH child@0;
        mother WITH child@0;

       !Imposing orthogonality between negative affect and item-specific factors;
        na WITH i1@0;
        na WITH i2@0;
        na WITH i3@0;
        na WITH i4@0;
        na WITH i5@0;
        na WITH i6@0;
        na WITH i7@0;
        na WITH i8@0;
        na WITH i9@0;
        na WITH i10@0;
        na WITH i11@0;
        na WITH i12@0;
        na WITH i13@0;
        na WITH i14@0;
        na WITH i15@0;
        na WITH i16@0;
        na WITH i17@0;

        i1 WITH i2@0 i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0
        i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i2 WITH i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0
        i13@0 i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i3 WITH i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0
        i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i4 WITH i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
```

```
        i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i5 WITH i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i6 WITH i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0
        i16@0 i17@0 father@0 mother@0 child@0;
        i7 WITH i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0
        i17@0 father@0 mother@0 child@0;
        i8 WITH i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 mother@0 child@0;
        i9 WITH i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 mother@0 child@0;
        i10 WITH i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 mother@0 child@0;
        i11 WITH i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 father@0
        mother@0 child@0;
        i12 WITH i13@0 i14@0 i15@0 i16@0 i17@0 father@0
        mother@0 child@0;
        i13 WITH i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i14 WITH i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i15 WITH i16@0 i17@0 father@0 mother@0 child@0;
        i16 WITH i17@0 father@0 mother@0 child@0;
        i17 WITH father@0 mother@0 child@0;

        !Adding age as a predictor
        mother ON Age;
        child ON Age;
        i%s ON Age;


OUTPUT: modindices (0);

SAVEDATA: DIFFTEST=tri_factor_fmc_int_diff_test_freeing_item%d.dat;'

# Iterating through each item
for (i in 1:17) {
    # Creating the text specifying the data set and tri-factor model for the current item (by
changing out the placeholders to match the current item).
  # Putting in values for the placeholders for the first part of the model template
    mplus_model_1 <- sprintf(tri_factor_likelihood_fmc_int_code_1, i, freed_item_each_list_3[[i]]
[1], freed_item_each_list_3[[i]][2], freed_item_each_list_3[[i]][3], freed_item_each_list_3[[i]]
[4], freed_item_each_list_3[[i]][5], freed_item_each_list_3[[i]][6], freed_item_each_list_3[[i]]
[7], freed_item_each_list_3[[i]][8], freed_item_each_list_3[[i]][9], freed_item_each_list_3[[i]]
[10], freed_item_each_list_3[[i]][11], freed_item_each_list_3[[i]][12], freed_item_each_list_3[[i]]
[13], freed_item_each_list_3[[i]][14], freed_item_each_list_3[[i]][15], freed_item_each_list_3[[i]]
[16], freed_item_each_list_3[[i]][17], freed_item_each_list_3[[i]][18], freed_item_each_list_3[[i]]
[19], freed_item_each_list_3[[i]][20], freed_item_each_list_3[[i]][21], freed_item_each_list_3[[i]]
[22], freed_item_each_list_3[[i]][23], freed_item_each_list_3[[i]][24], freed_item_each_list_3[[i]]
[25], freed_item_each_list_3[[i]][26], freed_item_each_list_3[[i]][27], freed_item_each_list_3[[i]]
[28], freed_item_each_list_3[[i]][29], freed_item_each_list_3[[i]][30], freed_item_each_list_3[[i]]
[31], freed_item_each_list_3[[i]][32], freed_item_each_list_3[[i]][33], freed_item_each_list_3[[i]]
[34], freed_item_each_list_3[[i]][35], freed_item_each_list_3[[i]][36], freed_item_each_list_3[[i]]
[37], freed_item_each_list_3[[i]][38], freed_item_each_list_3[[i]][39], freed_item_each_list_3[[i]]
[40], freed_item_each_list_3[[i]][41], freed_item_each_list_3[[i]][42], freed_item_each_list_3[[i]]
[43], freed_item_each_list_3[[i]][44], freed_item_each_list_3[[i]][45], freed_item_each_list_3[[i]]
```

```
[46], freed_item_each_list_3[[i]][47], freed_item_each_list_3[[i]][48], freed_item_each_list_3[[i]]
[49], freed_item_each_list_3[[i]][50], freed_item_each_list_3[[i]][51])
      # Used the following code to make the indexed variables with freed_item_each_list
paste0("freed_item_each_list_3[[i]][", 1:108, "]")

    # Putting in values for the placeholders for the second part of the model template
    mplus_model_2 <- sprintf(tri_factor_likelihood_fmc_int_code_2, freed_item_each_list_3[[i]][52],
freed_item_each_list_3[[i]][53], freed_item_each_list_3[[i]][54], freed_item_each_list_3[[i]][55],
freed_item_each_list_3[[i]][56], freed_item_each_list_3[[i]][57], freed_item_each_list_3[[i]][58],
freed_item_each_list_3[[i]][59], freed_item_each_list_3[[i]][60], freed_item_each_list_3[[i]][61],
freed_item_each_list_3[[i]][62], freed_item_each_list_3[[i]][63], freed_item_each_list_3[[i]][64],
freed_item_each_list_3[[i]][65], freed_item_each_list_3[[i]][66], freed_item_each_list_3[[i]][67],
freed_item_each_list_3[[i]][68], freed_item_each_list_3[[i]][69], freed_item_each_list_3[[i]][70],
freed_item_each_list_3[[i]][71], freed_item_each_list_3[[i]][72], freed_item_each_list_3[[i]][73],
freed_item_each_list_3[[i]][74], freed_item_each_list_3[[i]][75], freed_item_each_list_3[[i]][76],
freed_item_each_list_3[[i]][77], freed_item_each_list_3[[i]][78], freed_item_each_list_3[[i]][79],
freed_item_each_list_3[[i]][80], freed_item_each_list_3[[i]][81], freed_item_each_list_3[[i]][82],
freed_item_each_list_3[[i]][83], freed_item_each_list_3[[i]][84], freed_item_each_list_3[[i]][85],
freed_item_each_list_3[[i]][86], freed_item_each_list_3[[i]][87], freed_item_each_list_3[[i]][88],
freed_item_each_list_3[[i]][89], freed_item_each_list_3[[i]][90], freed_item_each_list_3[[i]][91],
freed_item_each_list_3[[i]][92], freed_item_each_list_3[[i]][93], freed_item_each_list_3[[i]][94],
freed_item_each_list_3[[i]][95], freed_item_each_list_3[[i]][96], freed_item_each_list_3[[i]][97],
freed_item_each_list_3[[i]][98], freed_item_each_list_3[[i]][99], freed_item_each_list_3[[i]][100],
freed_item_each_list_3[[i]][101], freed_item_each_list_3[[i]][102], i, i)

    # Pasting together the two parts of the model template, after having used sprintf()
    mplus_model <- paste0(mplus_model_1, mplus_model_2)

    # Create a file that is stored in the Mplus folder from the text specified in the mplus_model
variable above
    writeLines(mplus_model, sprintf("Mplus/tri-factor model - fmc - int - likelihood testing
dataset - free item %d.inp", i))

    # Running the model through the input Mplus file using MplusAutomation
    runModels(sprintf("Mplus/tri-factor model - fmc - int - likelihood testing dataset - free item
%d.inp", i))
      # An output file will then be saved automatically with the results.
      # Furthermore, as specified in the model text, the data from the model will be saved in a dat
file in the Mplus folder with the file name "tri_factor_fmc_int_diff_test_freeing_item%d.dat" where
%d is the item number
  }
```

Extracting the results from likelihood test data set: This code has to be run through manually, as it is eval = FALSE

```
# Takes a few minutes to run through

# Creating an empty data frame to add the data from each iteration to
tri_fac_diff_test_data_int_df <- data.frame(matrix(nrow = 0, ncol = 6))

# Iterating through each subscale
for (i in 1:17) {
  # Storing the name of the file path with the current subscale to a variable names file_path
    file_path <- sprintf("Mplus/tri-factor model - fmc - int - likelihood testing dataset - free
```

```
item %d.out", i)
      # The sprintf() function takes text with a placeholder (for example %s) and adds in a value
that is provided as another argument, for example "scale_vect[i].

    # Reading in the data from the output file of the tri-factor model for the current subscale,
using the file path specified at the beginning of the code.
    data <- readModels(file_path)

    # Assign a starting value of NA for all the fit indices
    chi_sq <- chi_sq_df <- chi_sq_p <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting chi-square stats from data$summaries form the output file
      chi_sq <- data$summaries$ChiSqM_Value
      chi_sq_df <- data$summaries$ChiSqM_DF
      chi_sq_p <- data$summaries$ChiSqM_PValue
      # Provides the number of warnings in the output file
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model
didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error
block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The
values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # Assign a starting value of NA for the warning variable
    warn_list <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting warnings from data$warnings form the output file
      # List of all the warnings in the output file
      warn_list <- data$warnings[1]
      # Use [1] after because of some weird nesting of the values
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model
didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error
block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The
values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # List of all the errors in the output file
    error_list <- data$errors[1]
      # Use [1] after because of some weird nesting of the values

    # Create a temporary data file with all the values gathered above for the current iteration of
the subscale to add to the overall data frame.
    temp_df <- data.frame(i, chi_sq, chi_sq_df, chi_sq_p, I(warn_list), I(error_list))
      # I() means to treat the object as it is, so R keeps the list as a list when adding it to the
data frame
    # Changing the column names so that they are always the same, regardless of whether the
variable names are used (when there are no fit indices) or the name of the extracted value is (when
there are fit indices). This is to facilitate binding the data frame with the overall data frame
without getting any errors matching up the columns.
```

```
    colnames(temp_df) <- c("Item", "chi_sq", "chi_sq_df", "chi_sq_p", "warnings", "errors")

    # Bind the data frame from the current model to the overall sdq_tri_fac_stat_df data frame
    tri_fac_diff_test_data_int_df <- rbind(tri_fac_diff_test_data_int_df, temp_df)
  }

# Print the data frame with the results of the tri-factor models
print(tri_fac_diff_test_data_int_df)
```

Check the errors from fitting the model:

```
print(tri_fac_diff_test_data_int_df$errors)
```

Check the warnings from fitting the model:

```
print(tri_fac_diff_test_data_int_df$warnings)
```

Getting parameters for each item freed: Extracting the parameters from the conditional tri-factor model with one item freed for internalizing symptoms: This code has to be run through manually, as it is eval = FALSE

```
# Takes a few minutes to run through

# Creating an empty data frame to add the data from each iteration to
tri_fac_diff_test_parameters_int_df <- data.frame(matrix(nrow = 0, ncol = 7))

# Iterating through each item
for (i in 1:17) {
  # Storing the name of the file path with the current subscale to a variable names file_path
    file_path <- sprintf("Mplus/tri-factor model - fmc - int - likelihood testing dataset - free
item %d.out", i)
      # The sprintf() function takes text with a placeholder (for example %s) and adds in a value
that is provided as another argument, for example "scale_vect[i].


    # Reading in the data from the output file of the tri-factor model for the current subscale,
using the file path specified at the beginning of the code.
    data <- readModels(file_path)

    # Extracting the estimates of raters on age from the model
    temp_df <- data$parameters$unstandardized %>% filter(paramHeader == "Intercepts")
    # Extracting the estimates of rater intercepts from the model
    temp_df <- rbind(temp_df, data$parameters$unstandardized %>% filter(param == "AGE"))

    # Extracting every fifth row, starting on the current item
    #temp_df <- temp_df[seq(i, nrow(temp_df), 18), ]
      # Provides the parameter stats for only the item currently freed
    # Checking whether there are 6 columns. If not, Mplpus hasn't been able to calculate standard
error estimates, which leaves out 3 columns
   # if (ncol(temp_df) != 6){
      # If the standard estimates are absent, add those three columns with a value of NA
      #temp_df$se <- temp_df$est_se <- temp_df$pval <- rep(NA, nrow(temp_df))
        # So that the data frames will have similar dimensions and therefore be able to combine
```

```
them
   # }
    temp_df$item <- rep(i, nrow(temp_df))
    # Extracting the rater in question for the parameter estimate (differ per row)
    #temp_df$rater <- str_extract(temp_df$param, "[[:alpha:]]+$")
    # Reordering the columns so that the model information comes before the parameter statistics
    temp_df <- temp_df[, c("item", "paramHeader", "param", "est", "se", "est_se", "pval")]

    # Bind the data frame from the current model to the overall sdq_tri_fac_parameters_df data
frame
    tri_fac_diff_test_parameters_int_df <- rbind(tri_fac_diff_test_parameters_int_df, temp_df)
  }

# Changing the misleading name of the est_se column to est_div_se
colnames(temp_df)[colnames(temp_df) == 'est_se'] <- 'est_div_se'

# Saving the data frame with the parameters for later reference
save(tri_fac_diff_test_parameters_int_df, file = "tri_fac_diff_test_parameters_int_df.Rdata")

# Print the data frame with the results of the tri-factor models
print(tri_fac_diff_test_parameters_int_df)
```

Filtering for the intercepts:

```
tri_fac_diff_test_parameters_int_df %>% filter(paramHeader == "Intercepts")
```

Making a plot of the intercepts for internalizing symptoms for the thesis:

```
tri_fac_diff_test_parameters_int_df %>% filter(paramHeader == "Intercepts" & (param == "MOTHER" |
param == "CHILD")) %>%
  mutate(param = case_when(
    param == "CHILD" ~ "Children compared to fathers",
    param == "MOTHER" ~ "Mothers compared to fathers")
    ) %>%
  ggplot(aes(x = item, y = est)) +
  geom_bar(stat = "identity", fill = "grey93", color = "black", width = 0.55) +
  facet_wrap(vars(param), nrow = 2) +
  geom_errorbar(aes(ymin = est - se, ymax = est + se), width = 0.25) +
  ylim(-1.50, 0.3) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  scale_x_continuous(breaks = 1:17) +
  geom_hline(aes(yintercept = 0)) +
  labs(x = "Items",
       y = "Intercepts")
```

Filtering for the estimates on age:

```
tri_fac_diff_test_parameters_int_df %>% filter(param == "AGE")
```

Making a plot of the estimates on age for internalizing symptoms for the thesis:

```
tri_fac_diff_test_parameters_int_df %>% filter(param == "AGE" & (paramHeader == "MOTHER.ON" |
paramHeader == "CHILD.ON")) %>%
  mutate(paramHeader = case_when(
    paramHeader == "CHILD.ON" ~ "Children compared to fathers",
    paramHeader == "MOTHER.ON" ~ "Mothers compared to fathers")
    ) %>%
  ggplot(aes(x = item, y = est)) +
  geom_bar(stat = "identity", fill = "grey93", color = "black", width = 0.55) +
  facet_wrap(vars(paramHeader), nrow = 2) +
  geom_errorbar(aes(ymin = est - se, ymax = est + se), width = 0.25) +
  ylim(-0.15, 0.05) +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  scale_x_continuous(breaks = 1:17) +
  geom_hline(aes(yintercept = 0)) +
  labs(x = "Items",
       y = "Estimates on Age")
```

9.4.2 - Running Likelihood Tests

Eval = FALSE, so run through this code manually to update it.

```
# Takes around 40 minutes to run through

# Creating the code for the input file in Mplus, which defines the data set and the model.
tri_factor_fmc_ext_code_1  <- '
  TITLE: Likelihood testing a conditional tri-factor model for age on externalizing symptoms;

  DATA: FILE = "Mplus/data_df_long_collapsed_unique_mplus.dat";

  VARIABLE:

      NAMES=
      ID Age ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya
      ext_3_f ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
      ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
      ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
      ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
      ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
      ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
```

```
    ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
    ext_18_cya int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya
    int_3_f int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f
    int_5_m int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m
    int_7_cya int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya
    int_10_f int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
    int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
    int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f
    int_16_m int_16_cya int_17_f int_17_m int_17_cya;

    USEVAR=
    Age ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya ext_3_f
    ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
    ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
    ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
    ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
    ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
    ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
    ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
    ext_18_cya;

    CATEGORICAL=
    ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya ext_3_f
    ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
    ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
    ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
    ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
    ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
    ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
    ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
    ext_18_cya;

    IDVARIABLE= ID;

    MISSING=.;

ANALYSIS:

    ESTIMATOR=wlsmv;
    PARAMETERIZATION=theta;
    PROCESSORS=8;
    DIFFTEST = tri_factor_fmc_ext_diff_test_freeing_item%d.dat;

MODEL:

       !Equality constraints in this model impose invariance
         !across reporters in thresholds and loadings;

       !Item thresholds, subject to equality constraints across reporters;
       [ext_1_f$1 ext_1_m$1 ext_1_cya$1] (it11);
       [ext_2_f$1 ext_2_m$1 ext_2_cya$1] (it12);
       [ext_3_f$1 ext_3_m$1 ext_3_cya$1] (it13);
       [ext_4_f$1 ext_4_m$1 ext_4_cya$1] (it14);
       [ext_5_f$1 ext_5_m$1 ext_5_cya$1] (it15);
```

```
    [ext_6_f$1 ext_6_m$1 ext_6_cya$1] (it16);
    [ext_7_f$1 ext_7_m$1 ext_7_cya$1] (it17);
    [ext_8_f$1 ext_8_m$1 ext_8_cya$1] (it18);
    [ext_9_f$1 ext_9_m$1 ext_9_cya$1] (it19);
    [ext_10_f$1 ext_10_m$1 ext_10_cya$1] (it110);
    [ext_11_f$1 ext_11_m$1 ext_11_cya$1] (it111);
    [ext_12_f$1 ext_12_m$1 ext_12_cya$1] (it112);
    [ext_13_f$1 ext_13_m$1 ext_13_cya$1] (it113);
    [ext_14_f$1 ext_14_m$1 ext_14_cya$1] (it114);
    [ext_15_f$1 ext_15_m$1 ext_15_cya$1] (it115);
    [ext_16_f$1 ext_16_m$1 ext_16_cya$1] (it116);
    [ext_17_f$1 ext_17_m$1 ext_17_cya$1] (it117);
    [ext_18_f$1 ext_18_m$1 ext_18_cya$1] (it118);

    !general factor;
    na BY ext_1_f* (in1);
    na BY ext_2_f* (in2);
    na BY ext_3_f* (in3);
    na BY ext_4_f* (in4);
    na BY ext_5_f* (in5);
    na BY ext_6_f* (in6);
    na BY ext_7_f* (in7);
    na BY ext_8_f* (in8);
    na BY ext_9_f* (in9);
    na BY ext_10_f* (in10);
    na BY ext_11_f* (in11);
    na BY ext_12_f* (in12);
    na BY ext_13_f* (in13);
    na BY ext_14_f* (in14);
    na BY ext_15_f* (in15);
    na BY ext_16_f* (in16);
    na BY ext_17_f* (in17);
    na BY ext_18_f* (in18);

    na BY ext_1_m* (in1);
    na BY ext_2_m* (in2);
    na BY ext_3_m* (in3);
    na BY ext_4_m* (in4);
    na BY ext_5_m* (in5);
    na BY ext_6_m* (in6);
    na BY ext_7_m* (in7);
    na BY ext_8_m* (in8);
    na BY ext_9_m* (in9);
    na BY ext_10_m* (in10);
    na BY ext_11_m* (in11);
    na BY ext_12_m* (in12);
    na BY ext_13_m* (in13);
    na BY ext_14_m* (in14);
    na BY ext_15_m* (in15);
    na BY ext_16_m* (in16);
    na BY ext_17_m* (in17);
    na BY ext_18_m* (in18);

    na BY ext_1_cya* (in1);
```

```
        na BY ext_2_cya* (in2);
        na BY ext_3_cya* (in3);
        na BY ext_4_cya* (in4);
        na BY ext_5_cya* (in5);
        na BY ext_6_cya* (in6);
        na BY ext_7_cya* (in7);
        na BY ext_8_cya* (in8);
        na BY ext_9_cya* (in9);
        na BY ext_10_cya* (in10);
        na BY ext_11_cya* (in11);
        na BY ext_12_cya* (in12);
        na BY ext_13_cya* (in13);
        na BY ext_14_cya* (in14);
        na BY ext_15_cya* (in15);
        na BY ext_16_cya* (in16);
        na BY ext_17_cya* (in17);
        na BY ext_18_cya* (in18);'

tri_factor_fmc_ext_code_2 <- '

        !father specific factor;
        father BY ext_1_f* (ip1);
        father BY ext_2_f* (ip2);
        father BY ext_3_f* (ip3);
        father BY ext_4_f* (ip4);
        father BY ext_5_f* (ip5);
        father BY ext_6_f* (ip6);
        father BY ext_7_f* (ip7);
        father BY ext_8_f* (ip8);
        father BY ext_9_f* (ip9);
        father BY ext_10_f* (ip10);
        father BY ext_11_f* (ip11);
        father BY ext_12_f* (ip12);
        father BY ext_13_f* (ip13);
        father BY ext_14_f* (ip14);
        father BY ext_15_f* (ip15);
        father BY ext_16_f* (ip16);
        father BY ext_17_f* (ip17);
        father BY ext_18_f* (ip18);

        !mom-specific factor;
        mother BY ext_1_m* (ip1);
        mother BY ext_2_m* (ip2);
        mother BY ext_3_m* (ip3);
        mother BY ext_4_m* (ip4);
        mother BY ext_5_m* (ip5);
        mother BY ext_6_m* (ip6);
        mother BY ext_7_m* (ip7);
        mother BY ext_8_m* (ip8);
        mother BY ext_9_m* (ip9);
        mother BY ext_10_m* (ip10);
        mother BY ext_11_m* (ip11);
        mother BY ext_12_m* (ip12);
        mother BY ext_13_m* (ip13);
```

```
        mother BY ext_14_m* (ip14);
        mother BY ext_15_m* (ip15);
        mother BY ext_16_m* (ip16);
        mother BY ext_17_m* (ip17);
        mother BY ext_18_m* (ip18);

        !child-specific factor;
        child BY ext_1_cya* (ip1);
        child BY ext_2_cya* (ip2);
        child BY ext_3_cya* (ip3);
        child BY ext_4_cya* (ip4);
        child BY ext_5_cya* (ip5);
        child BY ext_6_cya* (ip6);
        child BY ext_7_cya* (ip7);
        child BY ext_8_cya* (ip8);
        child BY ext_9_cya* (ip9);
        child BY ext_10_cya* (ip10);
        child BY ext_11_cya* (ip11);
        child BY ext_12_cya* (ip12);
        child BY ext_13_cya* (ip13);
        child BY ext_14_cya* (ip14);
        child BY ext_15_cya* (ip15);
        child BY ext_16_cya* (ip16);
        child BY ext_17_cya* (ip17);
        child BY ext_18_cya* (ip18);

        !item-specific factors;
        i1 BY ext_1_f* ext_1_m* ext_1_cya* (if1); [i1@0]; i1@1;
        i2 BY ext_2_f* ext_2_m* ext_2_cya* (if2); [i2@0]; i2@1;
        i3 BY ext_3_f* ext_3_m* ext_3_cya* (if3); [i3@0]; i3@1;
        i4 BY ext_4_f* ext_4_m* ext_4_cya* (if4); [i4@0]; i4@1;
        i5 BY ext_5_f* ext_5_m* ext_5_cya* (if5); [i5@0]; i5@1;
        i6 BY ext_6_f* ext_6_m* ext_6_cya* (if6); [i6@0]; i6@1;
        i7 BY ext_7_f* ext_7_m* ext_7_cya* (if7); [i7@0]; i7@1;
        i8 BY ext_8_f* ext_8_m* ext_8_cya* (if8); [i8@0]; i8@1;
        i9 BY ext_9_f* ext_9_m* ext_9_cya* (if9); [i9@0]; i9@1;
        i10 BY ext_10_f* ext_10_m* ext_10_cya* (if10); [i10@0]; i10@1;
        i11 BY ext_11_f* ext_11_m* ext_11_cya* (if11); [i11@0]; i11@1;
        i12 BY ext_12_f* ext_12_m* ext_12_cya* (if12); [i12@0]; i12@1;
        i13 BY ext_13_f* ext_13_m* ext_13_cya* (if13); [i13@0]; i13@1;
        i14 BY ext_14_f* ext_14_m* ext_14_cya* (if14); [i14@0]; i14@1;
        i15 BY ext_15_f* ext_15_m* ext_15_cya* (if15); [i15@0]; i15@1;
        i16 BY ext_16_f* ext_16_m* ext_16_cya* (if16); [i16@0]; i16@1;
        i17 BY ext_17_f* ext_17_m* ext_17_cya* (if17); [i17@0]; i17@1;
        i18 BY ext_18_f* ext_18_m* ext_18_cya* (if18); [i18@0]; i18@1;

        !identification/scaling constraints;
        [ na@0 ];
        [ father@0 ]; !Father is reference;
        [ mother* ];  !Mom is estimated;
        [ child* ];  !Child is estimated;

        na@1;
        father@1;  !Father is reference;
```

```
        mother*;   !Mom is estimated;
        child*;   !Child is estimated;

        !Imposing orthogonality of factors
        na WITH father@0;
        na WITH mother@0;
        na WITH child@0;
        father WITH mother@0;
        father WITH child@0;
        mother WITH child@0;

        !Imposing orthogonality between negative affect and item-specific factors;
        na WITH i1@0;
        na WITH i2@0;
        na WITH i3@0;
        na WITH i4@0;
        na WITH i5@0;
        na WITH i6@0;
        na WITH i7@0;
        na WITH i8@0;
        na WITH i9@0;
        na WITH i10@0;
        na WITH i11@0;
        na WITH i12@0;
        na WITH i13@0;
        na WITH i14@0;
        na WITH i15@0;
        na WITH i16@0;
        na WITH i17@0;
        na WITH i18@0;

        i1 WITH i2@0 i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0
        i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i2 WITH i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0
        i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i3 WITH i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0
        i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i4 WITH i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i5 WITH i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i6 WITH i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0
        i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i7 WITH i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0
        i17@0 i18@0 father@0 mother@0 child@0;
        i8 WITH i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        i18@0 father@0 mother@0 child@0;
        i9 WITH i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0
        father@0 mother@0 child@0;
        i10 WITH i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0
        father@0 mother@0 child@0;
        i11 WITH i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0
        mother@0 child@0;
        i12 WITH i13@0 i14@0 i15@0 i16@0 i17@0 i18@0 father@0
```

```
        mother@0 child@0;
        i13 WITH i14@0 i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i14 WITH i15@0 i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i15 WITH i16@0 i17@0 i18@0 father@0 mother@0 child@0;
        i16 WITH i17@0 i18@0 father@0 mother@0 child@0;
        i17 WITH i18@0 father@0 mother@0 child@0;
        i18 WITH father@0 mother@0 child@0;

        !Adding age as a predictor
        mother ON Age;
        child ON Age;


  OUTPUT: modindices (0);

  SAVEDATA: file=tri_factor_fmc_ext_conditional_likelihood_test_item%d.dat;'

# Iterating through each item
for (i in 1:18) {
    # Creating the text specifying the data set and tri-factor model for the current item (by
changing out the placeholders to match the current item).
  mplus_model_1 <- sprintf(tri_factor_fmc_ext_code_1, i)

  mplus_model_2 <- sprintf(tri_factor_fmc_ext_code_2, i)

  # Pasting together the two parts of the model template, after having used sprintf()
    mplus_model <- paste0(mplus_model_1, mplus_model_2)

    # Create a file that is stored in the Mplus folder from the text specified in the mplus_model
variable above
    writeLines(mplus_model, sprintf("Mplus/tri-factor model - fmc - ext - conditional - age -
likelihood-testing - freeing item %d.inp", i))

    # Running the model through the input Mplus file using MplusAutomation
    runModels(sprintf("Mplus/tri-factor model - fmc - ext - conditional - age - likelihood-testing
- freeing item %d.inp", i))
      # An output file will then be saved automatically with the results from the likelihood
testing
  }
```

Extracting the results from likelihood test data set: This code has to be run through manually, as it is eval = FALSE

```
# Takes three minutes to run through

# Creating an empty data frame to add the data from each iteration to
tri_fac_diff_test_stat_ext_df <- data.frame(matrix(nrow = 0, ncol = 6))

# Iterating through each subscale
for (i in 1:18) {
  # Storing the name of the file path with the current subscale to a variable names file_path
    file_path <- sprintf("Mplus/tri-factor model - fmc - ext - conditional - age - likelihood-
testing - freeing item %d.out", i)
      # The sprintf() function takes text with a placeholder (for example %s) and adds in a value
```

```
that is provided as another argument, for example "scale_vect[i].

    # Reading in the data from the output file of the tri-factor model for the current subscale, using the file path specified at the beginning of the code.
    data <- readModels(file_path)

    # Assign a starting value of NA for all the fit indices
    chi_sq <- chi_sq_df <- chi_sq_p <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and leave the variables with the initial value of NA
    tryCatch({
      # Extracting chi-square stats from data$summaries form the output file
      chi_sq <- data$summaries$ChiSqDiffTest_Value
      chi_sq_df <- data$summaries$ChiSqDiffTest_DF
      chi_sq_p <- data$summaries$ChiSqDiffTest_PValue
      # Provides the number of warnings in the output file
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # Assign a starting value of NA for the warning variable
    warn_list <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and leave the variables with the initial value of NA
    tryCatch({
      # Extracting warnings from data$warnings form the output file
      # List of all the warnings in the output file
      warn_list <- data$warnings[1]
      # Use [1] after because of some weird nesting of the values
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # List of all the errors in the output file
    error_list <- data$errors[1]
      # Use [1] after because of some weird nesting of the values

    # Create a temporary data file with all the values gathered above for the current iteration of the subscale to add to the overall data frame.
    temp_df <- data.frame(i, chi_sq, chi_sq_df, chi_sq_p, I(warn_list), I(error_list))
      # I() means to treat the object as it is, so R keeps the list as a list when adding it to the data frame
    # Changing the column names so that they are always the same, regardless of whether the variable names are used (when there are no fit indices) or the name of the extracted value is (when there are fit indices). This is to facilitate binding the data frame with the overall data frame without getting any errors matching up the columns.
    colnames(temp_df) <- c("Item", "chi_sq", "chi_sq_df", "chi_sq_p", "warnings", "errors")
```

```
    # Bind the data frame from the current model to the overall sdq_tri_fac_stat_df data frame
    tri_fac_diff_test_stat_ext_df <- rbind(tri_fac_diff_test_stat_ext_df, temp_df)
  }

# Sacing the data frame as a Rdata file
save(tri_fac_diff_test_stat_ext_df, file = "tri_fac_diff_test_stat_ext_df.Rdata")

# Print the data frame with the results of the tri-factor models
print(tri_fac_diff_test_stat_ext_df)
```

Filter for items that did not show a significant worse fit for the constrained model:

```
tri_fac_diff_test_stat_ext_df %>% filter(chi_sq_p >= 0.05)
```

Printing the errors from fitting the models:

```
print(tri_fac_diff_test_stat_ext_df$errors)
```

Printing the warnings from fitting the likelihood models:

```
print(tri_fac_diff_test_stat_ext_df$warnings)
```

Fitting the models for internalizing symptoms: Eval = FALSE, so run through this code manually to update it.

```
# Takes about 20 minutes to run through

# Creating the code for the input file in Mplus, which defines the data set and the model.
tri_factor_fmc_int_code_1  <- '
  TITLE: Likelihood testing a conditional tri-factor model for age on internalizing symptoms;

  DATA: FILE = "Mplus/data_df_long_collapsed_unique_mplus.dat";

  VARIABLE:

      NAMES=
      ID Age ext_1_f ext_1_m ext_1_cya ext_2_f ext_2_m ext_2_cya
      ext_3_f ext_3_m ext_3_cya ext_4_f ext_4_m ext_4_cya ext_5_f ext_5_m
      ext_5_cya ext_6_f ext_6_m ext_6_cya ext_7_f ext_7_m ext_7_cya
      ext_8_f ext_8_m ext_8_cya ext_9_f ext_9_m ext_9_cya ext_10_f
      ext_10_m ext_10_cya ext_11_f ext_11_m ext_11_cya ext_12_f
      ext_12_m ext_12_cya ext_13_f ext_13_m ext_13_cya ext_14_f
      ext_14_m ext_14_cya ext_15_f ext_15_m ext_15_cya ext_16_f ext_16_m
      ext_16_cya ext_17_f ext_17_m ext_17_cya ext_18_f ext_18_m
      ext_18_cya int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya
      int_3_f int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f
      int_5_m int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m
      int_7_cya int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya
      int_10_f int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
      int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
      int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f
      int_16_m int_16_cya int_17_f int_17_m int_17_cya;
```

```
    USEVAR=
    Age int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya int_3_f
    int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f int_5_m
    int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m int_7_cya
    int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya int_10_f
    int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
    int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
    int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f int_16_m
    int_16_cya int_17_f int_17_m int_17_cya;

    CATEGORICAL=
    int_1_f int_1_m int_1_cya int_2_f int_2_m int_2_cya int_3_f
    int_3_m int_3_cya int_4_f int_4_m int_4_cya int_5_f int_5_m
    int_5_cya int_6_f int_6_m int_6_cya int_7_f int_7_m int_7_cya
    int_8_f int_8_m int_8_cya int_9_f int_9_m int_9_cya int_10_f
    int_10_m int_10_cya int_11_f int_11_m int_11_cya int_12_f
    int_12_m int_12_cya int_13_f int_13_m int_13_cya int_14_f
    int_14_m int_14_cya int_15_f int_15_m int_15_cya int_16_f int_16_m
    int_16_cya int_17_f int_17_m int_17_cya;

    IDVARIABLE= ID;

    MISSING=.;

ANALYSIS:

    ESTIMATOR=wlsmv;
    PARAMETERIZATION=theta;
    PROCESSORS=8;
    DIFFTEST = tri_factor_fmc_int_diff_test_freeing_item%d.dat;

MODEL:

       !Equality constraints in this model impose invariance
         !across reporters in thresholds and loadings;

       !Item thresholds, subject to equality constraints across reporters;
       [int_1_f$1 int_1_m$1 int_1_cya$1] (it11);
       [int_2_f$1 int_2_m$1 int_2_cya$1] (it12);
       [int_3_f$1 int_3_m$1 int_3_cya$1] (it13);
       [int_4_f$1 int_4_m$1 int_4_cya$1] (it14);
       [int_5_f$1 int_5_m$1 int_5_cya$1] (it15);
       [int_6_f$1 int_6_m$1 int_6_cya$1] (it16);
       [int_7_f$1 int_7_m$1 int_7_cya$1] (it17);
       [int_8_f$1 int_8_m$1 int_8_cya$1] (it18);
       [int_9_f$1 int_9_m$1 int_9_cya$1] (it19);
       [int_10_f$1 int_10_m$1 int_10_cya$1] (it110);
       [int_11_f$1 int_11_m$1 int_11_cya$1] (it111);
       [int_12_f$1 int_12_m$1 int_12_cya$1] (it112);
       [int_13_f$1 int_13_m$1 int_13_cya$1] (it113);
       [int_14_f$1 int_14_m$1 int_14_cya$1] (it114);
       [int_15_f$1 int_15_m$1 int_15_cya$1] (it115);
       [int_16_f$1 int_16_m$1 int_16_cya$1] (it116);
```

```
        [int_17_f$1 int_17_m$1 int_17_cya$1] (it117);

        [int_1_f$2 int_1_m$2 int_1_cya$2] (it21);
        [int_2_f$2 int_2_m$2 int_2_cya$2] (it22);
        [int_3_f$2 int_3_m$2 int_3_cya$2] (it23);
        [int_4_f$2 int_4_m$2 int_4_cya$2] (it24);
        [int_5_f$2 int_5_m$2 int_5_cya$2] (it25);
        [int_6_f$2 int_6_m$2 int_6_cya$2] (it26);
        [int_7_f$2 int_7_m$2 int_7_cya$2] (it27);
        [int_8_f$2 int_8_m$2 int_8_cya$2] (it28);
        [int_9_f$2 int_9_m$2 int_9_cya$2] (it29);
        [int_10_f$2 int_10_m$2 int_10_cya$2] (it210);
        [int_11_f$2 int_11_m$2 int_11_cya$2] (it211);
        [int_12_f$2 int_12_m$2 int_12_cya$2] (it212);
        [int_13_f$2 int_13_m$2 int_13_cya$2] (it213);
        [int_14_f$2 int_14_m$2 int_14_cya$2] (it214);
        [int_15_f$2 int_15_m$2 int_15_cya$2] (it215);
        [int_16_f$2 int_16_m$2 int_16_cya$2] (it216);
        [int_17_f$2 int_17_m$2 int_17_cya$2] (it217);

        !general factor;
        na BY int_1_f* (in1);
        na BY int_2_f* (in2);
        na BY int_3_f* (in3);
        na BY int_4_f* (in4);
        na BY int_5_f* (in5);
        na BY int_6_f* (in6);
        na BY int_7_f* (in7);
        na BY int_8_f* (in8);
        na BY int_9_f* (in9);
        na BY int_10_f* (in10);
        na BY int_11_f* (in11);
        na BY int_12_f* (in12);
        na BY int_13_f* (in13);
        na BY int_14_f* (in14);
        na BY int_15_f* (in15);
        na BY int_16_f* (in16);
        na BY int_17_f* (in17);

        na BY int_1_m* (in1);
        na BY int_2_m* (in2);
        na BY int_3_m* (in3);
        na BY int_4_m* (in4);
        na BY int_5_m* (in5);
        na BY int_6_m* (in6);
        na BY int_7_m* (in7);
        na BY int_8_m* (in8);
        na BY int_9_m* (in9);
        na BY int_10_m* (in10);
        na BY int_11_m* (in11);
        na BY int_12_m* (in12);
        na BY int_13_m* (in13);
        na BY int_14_m* (in14);
        na BY int_15_m* (in15);
```

```
        na BY int_16_m* (in16);
        na BY int_17_m* (in17);

        na BY int_1_cya* (in1);
        na BY int_2_cya* (in2);
        na BY int_3_cya* (in3);
        na BY int_4_cya* (in4);
        na BY int_5_cya* (in5);
        na BY int_6_cya* (in6);
        na BY int_7_cya* (in7);
        na BY int_8_cya* (in8);
        na BY int_9_cya* (in9);
        na BY int_10_cya* (in10);
        na BY int_11_cya* (in11);
        na BY int_12_cya* (in12);
        na BY int_13_cya* (in13);
        na BY int_14_cya* (in14);
        na BY int_15_cya* (in15);
        na BY int_16_cya* (in16);
        na BY int_17_cya* (in17);'

tri_factor_fmc_int_code_2 <- '

        !father specific factor;
        father BY int_1_f* (ip1);
        father BY int_2_f* (ip2);
        father BY int_3_f* (ip3);
        father BY int_4_f* (ip4);
        father BY int_5_f* (ip5);
        father BY int_6_f* (ip6);
        father BY int_7_f* (ip7);
        father BY int_8_f* (ip8);
        father BY int_9_f* (ip9);
        father BY int_10_f* (ip10);
        father BY int_11_f* (ip11);
        father BY int_12_f* (ip12);
        father BY int_13_f* (ip13);
        father BY int_14_f* (ip14);
        father BY int_15_f* (ip15);
        father BY int_16_f* (ip16);
        father BY int_17_f* (ip17);

        !mom-specific factor;
        mother BY int_1_m* (ip1);
        mother BY int_2_m* (ip2);
        mother BY int_3_m* (ip3);
        mother BY int_4_m* (ip4);
        mother BY int_5_m* (ip5);
        mother BY int_6_m* (ip6);
        mother BY int_7_m* (ip7);
        mother BY int_8_m* (ip8);
        mother BY int_9_m* (ip9);
        mother BY int_10_m* (ip10);
        mother BY int_11_m* (ip11);
```

```
        mother BY int_12_m* (ip12);
        mother BY int_13_m* (ip13);
        mother BY int_14_m* (ip14);
        mother BY int_15_m* (ip15);
        mother BY int_16_m* (ip16);
        mother BY int_17_m* (ip17);

        !child-specific factor;
        child BY int_1_cya* (ip1);
        child BY int_2_cya* (ip2);
        child BY int_3_cya* (ip3);
        child BY int_4_cya* (ip4);
        child BY int_5_cya* (ip5);
        child BY int_6_cya* (ip6);
        child BY int_7_cya* (ip7);
        child BY int_8_cya* (ip8);
        child BY int_9_cya* (ip9);
        child BY int_10_cya* (ip10);
        child BY int_11_cya* (ip11);
        child BY int_12_cya* (ip12);
        child BY int_13_cya* (ip13);
        child BY int_14_cya* (ip14);
        child BY int_15_cya* (ip15);
        child BY int_16_cya* (ip16);
        child BY int_17_cya* (ip17);

        !item-specific factors;
        i1 BY int_1_f* int_1_m* int_1_cya* (if1); [i1@0]; i1@1;
        i2 BY int_2_f* int_2_m* int_2_cya* (if2); [i2@0]; i2@1;
        i3 BY int_3_f* int_3_m* int_3_cya* (if3); [i3@0]; i3@1;
        i4 BY int_4_f* int_4_m* int_4_cya* (if4); [i4@0]; i4@1;
        i5 BY int_5_f* int_5_m* int_5_cya* (if5); [i5@0]; i5@1;
        i6 BY int_6_f* int_6_m* int_6_cya* (if6); [i6@0]; i6@1;
        i7 BY int_7_f* int_7_m* int_7_cya* (if7); [i7@0]; i7@1;
        i8 BY int_8_f* int_8_m* int_8_cya* (if8); [i8@0]; i8@1;
        i9 BY int_9_f* int_9_m* int_9_cya* (if9); [i9@0]; i9@1;
        i10 BY int_10_f* int_10_m* int_10_cya* (if10); [i10@0]; i10@1;
        i11 BY int_11_f* int_11_m* int_11_cya* (if11); [i11@0]; i11@1;
        i12 BY int_12_f* int_12_m* int_12_cya* (if12); [i12@0]; i12@1;
        i13 BY int_13_f* int_13_m* int_13_cya* (if13); [i13@0]; i13@1;
        i14 BY int_14_f* int_14_m* int_14_cya* (if14); [i14@0]; i14@1;
        i15 BY int_15_f* int_15_m* int_15_cya* (if15); [i15@0]; i15@1;
        i16 BY int_16_f* int_16_m* int_16_cya* (if16); [i16@0]; i16@1;
        i17 BY int_17_f* int_17_m* int_17_cya* (if17); [i17@0]; i17@1;

        !identification/scaling constraints;
        [ na@0 ];
        [ father@0 ]; !Father is reference;
        [ mother* ];  !Mom is estimated;
        [ child* ];  !Child is estimated;

        na@1;
        father@1;  !Father is reference;
        mother*;   !Mom is estimated;
```

```
        child*;   !Child is estimated;

        !Imposing orthogonality of factors
        na WITH father@0;
        na WITH mother@0;
        na WITH child@0;
        father WITH mother@0;
        father WITH child@0;
        mother WITH child@0;

        !Imposing orthogonality between negative affect and item-specific factors;
        na WITH i1@0;
        na WITH i2@0;
        na WITH i3@0;
        na WITH i4@0;
        na WITH i5@0;
        na WITH i6@0;
        na WITH i7@0;
        na WITH i8@0;
        na WITH i9@0;
        na WITH i10@0;
        na WITH i11@0;
        na WITH i12@0;
        na WITH i13@0;
        na WITH i14@0;
        na WITH i15@0;
        na WITH i16@0;
        na WITH i17@0;

        i1 WITH i2@0 i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0
        i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i2 WITH i3@0 i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0
        i13@0 i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i3 WITH i4@0 i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0
        i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i4 WITH i5@0 i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i5 WITH i6@0 i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i6 WITH i7@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0
        i16@0 i17@0 father@0 mother@0 child@0;
        i7 WITH i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0
        i17@0 father@0 mother@0 child@0;
        i8 WITH i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 mother@0 child@0;
        i9 WITH i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 mother@0 child@0;
        i10 WITH i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 mother@0 child@0;
        i11 WITH i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 father@0
        mother@0 child@0;
        i12 WITH i13@0 i14@0 i15@0 i16@0 i17@0 father@0
        mother@0 child@0;
        i13 WITH i14@0 i15@0 i16@0 i17@0 father@0 mother@0 child@0;
```

```
        i14 WITH i15@0 i16@0 i17@0 father@0 mother@0 child@0;
        i15 WITH i16@0 i17@0 father@0 mother@0 child@0;
        i16 WITH i17@0 father@0 mother@0 child@0;
        i17 WITH father@0 mother@0 child@0;

        !Adding age as a predictor
        mother ON Age;
        child ON Age;


  OUTPUT: modindices (0);

  SAVEDATA: file=tri_factor_fmc_int_conditional_likelihood_test_item%d.dat;'

# Iterating through each item
for (i in 1:17) {
    # Creating the text specifying the data set and tri-factor model for the current item (by
changing out the placeholders to match the current item).
  mplus_model_1 <- sprintf(tri_factor_fmc_int_code_1, i)

  mplus_model_2 <- sprintf(tri_factor_fmc_int_code_2, i)

  # Pasting together the two parts of the model template, after having used sprintf()
    mplus_model <- paste0(mplus_model_1, mplus_model_2)

    # Create a file that is stored in the Mplus folder from the text specified in the mplus_model
variable above
    writeLines(mplus_model, sprintf("Mplus/tri-factor model - fmc - int - conditional - age -
likelihood-testing - freeing item %d.inp", i))

    # Running the model through the input Mplus file using MplusAutomation
    runModels(sprintf("Mplus/tri-factor model - fmc - int - conditional - age - likelihood-testing
- freeing item %d.inp", i))
      # An output file will then be saved automatically with the results from the likelihood
testing
  }
```

Extracting the results from likelihood test data set: This code has to be run through manually, as it is eval = FALSE

```
# Takes three minutes to run through

# Creating an empty data frame to add the data from each iteration to
tri_fac_diff_test_stat_int_df <- data.frame(matrix(nrow = 0, ncol = 6))

# Iterating through each subscale
for (i in 1:17) {
  # Storing the name of the file path with the current subscale to a variable names file_path
    file_path <- sprintf("Mplus/tri-factor model - fmc - int - conditional - age - likelihood-
testing - freeing item %d.out", i)
      # The sprintf() function takes text with a placeholder (for example %s) and adds in a value
that is provided as another argument, for example "scale_vect[i].

    # Reading in the data from the output file of the tri-factor model for the current subscale,
```

```
using the file path specified at the beginning of the code.
    data <- readModels(file_path)

    # Assign a starting value of NA for all the fit indices
    chi_sq <- chi_sq_df <- chi_sq_p <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting chi-square stats from data$summaries form the output file
      chi_sq <- data$summaries$ChiSqDiffTest_Value
      chi_sq_df <- data$summaries$ChiSqDiffTest_DF
      chi_sq_p <- data$summaries$ChiSqDiffTest_PValue
      # Provides the number of warnings in the output file
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model
didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error
block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The
values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # Assign a starting value of NA for the warning variable
    warn_list <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting warnings from data$warnings form the output file
      # List of all the warnings in the output file
      warn_list <- data$warnings[1]
      # Use [1] after because of some weird nesting of the values
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model
didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error
block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The
values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # List of all the errors in the output file
    error_list <- data$errors[1]
      # Use [1] after because of some weird nesting of the values

    # Create a temporary data file with all the values gathered above for the current iteration of
the subscale to add to the overall data frame.
    temp_df <- data.frame(i, chi_sq, chi_sq_df, chi_sq_p, I(warn_list), I(error_list))
      # I() means to treat the object as it is, so R keeps the list as a list when adding it to the
data frame
    # Changing the column names so that they are always the same, regardless of whether the
variable names are used (when there are no fit indices) or the name of the extracted value is (when
there are fit indices). This is to facilitate binding the data frame with the overall data frame
without getting any errors matching up the columns.
    colnames(temp_df) <- c("Item", "chi_sq", "chi_sq_df", "chi_sq_p", "warnings", "errors")

    # Bind the data frame from the current model to the overall sdq_tri_fac_stat_df data frame
    tri_fac_diff_test_stat_int_df <- rbind(tri_fac_diff_test_stat_int_df, temp_df)
  }
```

```
# Sacing the data frame as a Rdata file
save(tri_fac_diff_test_stat_int_df, file = "tri_fac_diff_test_stat_int_df.Rdata")

# Print the data frame with the results of the tri-factor models
print(tri_fac_diff_test_stat_int_df)
```

Filter for items that did not show a significant worse fit for the constrained model:

```
tri_fac_diff_test_stat_int_df %>% filter(chi_sq_p >= 0.05)
```

Printing the errors from fitting the models:

```
print(tri_fac_diff_test_stat_int_df$errors)
```

Printing the warnings from fitting the models:

```
print(tri_fac_diff_test_stat_int_df$warnings)
```

9.5. Running a Constrained tri-factor model with Insecure attachment as a predictor

In Mplus, all the columns for the data frames has to be named explicitly. Therefore, I will filter the data_df_long_collapsed and data_df_long_unique_collapsed data frames so that they only have the variables we will need to fit the tri-factor model:

```
# Making a copy of the data frame to make mplus compatible
data_df_wide_diff_mplus <- data_df_wide_diff

data_df_wide_diff_mplus <- data_df_wide_diff_mplus %>% select(ID, y10aaa_scale_ya_f,
y10aaa_scale_ya_m, y8ext_f_1, y8ext_m_1, y8ext_cya_1, y8ext_f_2, y8ext_m_2, y8ext_cya_2, y8ext_f_3,
y8ext_m_3, y8ext_cya_3, y8ext_f_4, y8ext_m_4, y8ext_cya_4, y8ext_f_5, y8ext_m_5, y8ext_cya_5,
y8ext_f_6, y8ext_m_6, y8ext_cya_6, y8ext_f_7, y8ext_m_7, y8ext_cya_7, y8ext_f_8, y8ext_m_8,
y8ext_cya_8, y8ext_f_9, y8ext_m_9, y8ext_cya_9, y8ext_f_10, y8ext_m_10, y8ext_cya_10, y8ext_f_11,
y8ext_m_11, y8ext_cya_11, y8ext_f_12, y8ext_m_12, y8ext_cya_12, y8ext_f_13, y8ext_m_13,
y8ext_cya_13, y8ext_f_14, y8ext_m_14, y8ext_cya_14, y8ext_f_15, y8ext_m_15, y8ext_cya_15,
y8ext_f_16, y8ext_m_16, y8ext_cya_16, y8ext_f_17, y8ext_m_17, y8ext_cya_17, y8ext_f_18, y8ext_m_18,
y8ext_cya_18, y8int_f_1, y8int_m_1, y8int_cya_1, y8int_f_2, y8int_m_2, y8int_cya_2, y8int_f_3,
y8int_m_3, y8int_cya_3, y8int_f_4, y8int_m_4, y8int_cya_4, y8int_f_5, y8int_m_5, y8int_cya_5,
y8int_f_6, y8int_m_6, y8int_cya_6, y8int_f_7, y8int_m_7, y8int_cya_7, y8int_f_8, y8int_m_8,
y8int_cya_8, y8int_f_9, y8int_m_9, y8int_cya_9, y8int_f_10, y8int_m_10, y8int_cya_10, y8int_f_11,
y8int_m_11, y8int_cya_11, y8int_f_12, y8int_m_12, y8int_cya_12, y8int_f_13, y8int_m_13,
y8int_cya_13, y8int_f_14, y8int_m_14, y8int_cya_14, y8int_f_15, y8int_m_15, y8int_cya_15,
y8int_f_16, y8int_m_16, y8int_cya_16, y8int_f_17, y8int_m_17, y8int_cya_17
)

# The variable names are going to be too long, as mplus only takes 8 characters in a variable name.
We will therefore rename the variables to be shorter
colnames(data_df_wide_diff_mplus) <- gsub("^y8(i|e).*_([[:alpha:]]+)_(\\d+)$", "\\1_\\2_\\3",
colnames(data_df_wide_diff_mplus))
  # This changes the variable names for the ext and int items from the format "y8int_f_1" to
"i_f_1"
```

```
colnames(data_df_wide_diff_mplus) <- gsub("^y10aaa_scale_ya_(f|m)$", "aaa_\\1",
colnames(data_df_wide_diff_mplus))
  # This changes the attachment scale from the format "y10aaa_scale_ya_f" to "aaa_f"
```

Mplus has a very specific format that the data has to be in for it to read it correctly. The following section will therefore make a copy of the data frames we will be using that are made compatible with mplus:

```
# Code to make it only for the first 100 rows, to have a more manageable file to inspect with
notebook
#data_df_long_unique_attachment_mplus <- data_df_long_unique_attachment_mplus[1:100,]

# Make free form data file and save it as a dat. file in a Mplus folder
prepareMplusData(df = data_df_wide_diff_mplus, filename = "Mplus/data_df_wide_diff_mplus.dat")
```

```
-----
Factor: e_f_1
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_1
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_1
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_2
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
```

```
-----



-----
Factor: e_m_2
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_2
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_3
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_3
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_3
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
```

```
Factor: e_f_4
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_4
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_4
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_5
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_5
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_5
Conversion:
              level number
                Not      1
```

```
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_6
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_6
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_6
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_7
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_7
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: e_cya_7
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
          Very/often     3
-----



-----
Factor: e_f_8
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
          Very/often     3
-----



-----
Factor: e_m_8
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
          Very/often     3
-----



-----
Factor: e_cya_8
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
          Very/often     3
-----



-----
Factor: e_f_9
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
          Very/often     3
-----



-----
Factor: e_m_9
Conversion:
```

```
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_9
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_10
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_10
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_10
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_11
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
```

```
-----



-----
Factor: e_m_11
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_11
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_12
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_12
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_12
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
```

```
Factor: e_f_13
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_13
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_13
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_14
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_14
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_14
Conversion:
              level number
                Not      1
```

```
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_15
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_15
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_15
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_16
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_16
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: e_cya_16
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_17
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_17
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_17
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_f_18
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_m_18
Conversion:
```

```
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: e_cya_18
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_1
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_1
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_1
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_2
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
```

```
-----



-----
Factor: i_m_2
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_2
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_3
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_3
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_3
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
```

```
Factor: i_f_4
Conversion:
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_4
Conversion:
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_4
Conversion:
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_5
Conversion:
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_5
Conversion:
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_5
Conversion:
               level number
                 Not      1
```

```
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_6
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_6
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_6
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_7
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_7
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: i_cya_7
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_8
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_8
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_8
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_9
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_9
Conversion:
```

```
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_9
Conversion:
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_10
Conversion:
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_10
Conversion:
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_10
Conversion:
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_11
Conversion:
               level number
                 Not      1
 Somewhat/sometimes      2
         Very/often      3
```

```
-----



-----
Factor: i_m_11
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_11
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_12
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_12
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_12
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
```

```
Factor: i_f_13
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_13
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_13
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_14
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_14
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_14
Conversion:
              level number
                Not      1
```

```
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_15
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_15
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_cya_15
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_f_16
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----



-----
Factor: i_m_16
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
         Very/often      3
-----
```

```
-----
Factor: i_cya_16
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: i_f_17
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: i_m_17
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----



-----
Factor: i_cya_17
Conversion:
              level number
                Not      1
 Somewhat/sometimes      2
        Very/often      3
-----

The file(s)
 'data_df_wide_diff_mplus.dat'
currently exist(s) and will be overwritten

TITLE: Your title goes here
DATA: FILE = "Mplus/data_df_wide_diff_mplus.dat";
VARIABLE:
NAMES = ID aaa_f aaa_m e_f_1 e_m_1 e_cya_1 e_f_2 e_m_2 e_cya_2 e_f_3 e_m_3 e_cya_3
     e_f_4 e_m_4 e_cya_4 e_f_5 e_m_5 e_cya_5 e_f_6 e_m_6 e_cya_6 e_f_7 e_m_7 e_cya_7
     e_f_8 e_m_8 e_cya_8 e_f_9 e_m_9 e_cya_9 e_f_10 e_m_10 e_cya_10 e_f_11 e_m_11
     e_cya_11 e_f_12 e_m_12 e_cya_12 e_f_13 e_m_13 e_cya_13 e_f_14 e_m_14 e_cya_14
     e_f_15 e_m_15 e_cya_15 e_f_16 e_m_16 e_cya_16 e_f_17 e_m_17 e_cya_17 e_f_18
     e_m_18 e_cya_18 i_f_1 i_m_1 i_cya_1 i_f_2 i_m_2 i_cya_2 i_f_3 i_m_3 i_cya_3
     i_f_4 i_m_4 i_cya_4 i_f_5 i_m_5 i_cya_5 i_f_6 i_m_6 i_cya_6 i_f_7 i_m_7 i_cya_7
```

```
     i_f_8 i_m_8 i_cya_8 i_f_9 i_m_9 i_cya_9 i_f_10 i_m_10 i_cya_10 i_f_11 i_m_11
     i_cya_11 i_f_12 i_m_12 i_cya_12 i_f_13 i_m_13 i_cya_13 i_f_14 i_m_14 i_cya_14
     i_f_15 i_m_15 i_cya_15 i_f_16 i_m_16 i_cya_16 i_f_17 i_m_17 i_cya_17;
MISSING=.;
```

```
# For Veronica
#prepareMplusData(df = data_df_wide_diff_mplus, filename = "mplusdata/data_df_wide_diff_mplus.dat")

# The above code also gives an output that can be used as a basis for an inp file. However, note
that the file path is too long, so you need to add a line shift somewhere in it so that Mplus won't
cut it off
```

Internalizing symptoms: There are two different attachment variables, one for attachment with fathers and one with mothers. I will therefore run two tri-factor models where one is fit with attachment for fathers and one with mothers Run through manually as eval = FALSE

Attachment with father: We had to take out item 7, as it didn't have enough responses in all of the possible response categories

```
# Takes a few minutes to run through

# Creating the code for the input file in Mplus, which defines the data set and the model.
tri_factor_fmc_int_attachment_code  <- '
  TITLE: Constrained conditional tri-factor model for internalizing symptoms;

  DATA: FILE = "Mplus/data_df_wide_diff_mplus.dat";

  VARIABLE:

      NAMES=
      ID aaa_f aaa_m e_1_f e_1_m e_1_cya e_2_f e_2_m e_2_cya e_3_f
      e_3_m e_3_cya e_4_f e_4_m e_4_cya e_5_f e_5_m e_5_cya e_6_f
      e_6_m e_6_cya e_7_f e_7_m e_7_cya e_8_f e_8_m e_8_cya e_9_f
      e_9_m e_9_cya e_10_f e_10_m e_10_cya e_11_f e_11_m e_11_cya
      e_12_f e_12_m e_12_cya e_13_f e_13_m e_13_cya e_14_f e_14_m
      e_14_cya e_15_f e_15_m e_15_cya e_16_f e_16_m e_16_cya
      e_17_f e_17_m e_17_cya e_18_f e_18_m e_18_cya i_1_f i_1_m
      i_1_cya i_2_f i_2_m i_2_cya i_3_f i_3_m i_3_cya i_4_f i_4_m
      i_4_cya i_5_f i_5_m i_5_cya i_6_f i_6_m i_6_cya i_7_f i_7_m
      i_7_cya i_8_f i_8_m i_8_cya i_9_f i_9_m i_9_cya i_10_f i_10_m
      i_10_cya i_11_f i_11_m i_11_cya i_12_f i_12_m i_12_cya i_13_f
      i_13_m i_13_cya i_14_f i_14_m i_14_cya i_15_f i_15_m i_15_cya
      i_16_f i_16_m i_16_cya i_17_f i_17_m i_17_cya;


      USEVAR=
      aaa_f i_1_f i_1_cya i_2_f i_2_cya i_3_f i_3_cya i_4_f i_4_cya
      i_5_f i_5_cya i_6_f i_6_cya i_8_f i_8_cya i_9_f
      i_9_cya i_10_f i_10_cya i_11_f i_11_cya i_12_f i_12_cya i_13_f
      i_13_cya i_14_f i_14_cya i_15_f i_15_cya i_16_f i_16_cya i_17_f
      i_17_cya;

      CATEGORICAL=
```

```
    i_1_f i_1_cya i_2_f i_2_cya i_3_f i_3_cya i_4_f i_4_cya
    i_5_f i_5_cya i_6_f i_6_cya i_8_f i_8_cya i_9_f
    i_9_cya i_10_f i_10_cya i_11_f i_11_cya i_12_f i_12_cya i_13_f
    i_13_cya i_14_f i_14_cya i_15_f i_15_cya i_16_f i_16_cya i_17_f
    i_17_cya;

    IDVARIABLE= ID;

    MISSING=.;

ANALYSIS:

    ESTIMATOR=wlsmv;
    PARAMETERIZATION=theta;
    PROCESSORS=8;

MODEL:

       !Equality constraints in this model impose invariance
         !across reporters in thresholds and loadings;

       !Item thresholds, subject to equality constraints across reporters;
       [i_1_f$1 i_1_cya$1] (it11);
       [i_2_f$1 i_2_cya$1] (it12);
       [i_3_f$1 i_3_cya$1] (it13);
       [i_4_f$1 i_4_cya$1] (it14);
       [i_5_f$1 i_5_cya$1] (it15);
       [i_6_f$1 i_6_cya$1] (it16);
       [i_8_f$1 i_8_cya$1] (it18);
       [i_9_f$1 i_9_cya$1] (it19);
       [i_10_f$1 i_10_cya$1] (it110);
       [i_11_f$1 i_11_cya$1] (it111);
       [i_12_f$1 i_12_cya$1] (it112);
       [i_13_f$1 i_13_cya$1] (it113);
       [i_14_f$1 i_14_cya$1] (it114);
       [i_15_f$1 i_15_cya$1] (it115);
       [i_16_f$1 i_16_cya$1] (it116);
       [i_17_f$1 i_17_cya$1] (it117);


       [i_1_f$2 i_1_cya$2] (it21);
       [i_2_f$2 i_2_cya$2] (it22);
       [i_3_f$2 i_3_cya$2] (it23);
       [i_4_f$2 i_4_cya$2] (it24);
       [i_5_f$2 i_5_cya$2] (it25);
       [i_6_f$2 i_6_cya$2] (it26);
       [i_8_f$2 i_8_cya$2] (it28);
       [i_9_f$2 i_9_cya$2] (it29);
       [i_10_f$2 i_10_cya$2] (it210);
       [i_11_f$2 i_11_cya$2] (it211);
       [i_12_f$2 i_12_cya$2] (it212);
       [i_13_f$2 i_13_cya$2] (it213);
       [i_14_f$2 i_14_cya$2] (it214);
       [i_15_f$2 i_15_cya$2] (it215);
```

```
        [i_16_f$2 i_16_cya$2] (it216);
        [i_17_f$2 i_17_cya$2] (it217);

        !general factor;
        na BY i_1_f* (in1);
        na BY i_2_f* (in2);
        na BY i_3_f* (in3);
        na BY i_4_f* (in4);
        na BY i_5_f* (in5);
        na BY i_6_f* (in6);
        na BY i_8_f* (in8);
        na BY i_9_f* (in9);
        na BY i_10_f* (in10);
        na BY i_11_f* (in11);
        na BY i_12_f* (in12);
        na BY i_13_f* (in13);
        na BY i_14_f* (in14);
        na BY i_15_f* (in15);
        na BY i_16_f* (in16);
        na BY i_17_f* (in17);

        na BY i_1_cya* (in1);
        na BY i_2_cya* (in2);
        na BY i_3_cya* (in3);
        na BY i_4_cya* (in4);
        na BY i_5_cya* (in5);
        na BY i_6_cya* (in6);
        na BY i_8_cya* (in8);
        na BY i_9_cya* (in9);
        na BY i_10_cya* (in10);
        na BY i_11_cya* (in11);
        na BY i_12_cya* (in12);
        na BY i_13_cya* (in13);
        na BY i_14_cya* (in14);
        na BY i_15_cya* (in15);
        na BY i_16_cya* (in16);
        na BY i_17_cya* (in17);

        !father specific factor;
        father BY i_1_f* (ip1);
        father BY i_2_f* (ip2);
        father BY i_3_f* (ip3);
        father BY i_4_f* (ip4);
        father BY i_5_f* (ip5);
        father BY i_6_f* (ip6);
        father BY i_8_f* (ip8);
        father BY i_9_f* (ip9);
        father BY i_10_f* (ip10);
        father BY i_11_f* (ip11);
        father BY i_12_f* (ip12);
        father BY i_13_f* (ip13);
        father BY i_14_f* (ip14);
        father BY i_15_f* (ip15);
        father BY i_16_f* (ip16);
```

```
        father BY i_17_f* (ip17);

        !child-specific factor;
        child BY i_1_cya* (ip1);
        child BY i_2_cya* (ip2);
        child BY i_3_cya* (ip3);
        child BY i_4_cya* (ip4);
        child BY i_5_cya* (ip5);
        child BY i_6_cya* (ip6);
        child BY i_8_cya* (ip8);
        child BY i_9_cya* (ip9);
        child BY i_10_cya* (ip10);
        child BY i_11_cya* (ip11);
        child BY i_12_cya* (ip12);
        child BY i_13_cya* (ip13);
        child BY i_14_cya* (ip14);
        child BY i_15_cya* (ip15);
        child BY i_16_cya* (ip16);
        child BY i_17_cya* (ip17);

        !item-specific factors;
        i1 BY i_1_cya* i_1_f* (if1); [i1@0]; i1@1;
        i2 BY i_2_cya* i_2_f* (if2); [i2@0]; i2@1;
        i3 BY i_3_cya* i_3_f* (if3); [i3@0]; i3@1;
        i4 BY i_4_cya* i_4_f* (if4); [i4@0]; i4@1;
        i5 BY i_5_cya* i_5_f* (if5); [i5@0]; i5@1;
        i6 BY i_6_cya* i_6_f* (if6); [i6@0]; i6@1;
        i8 BY i_8_cya* i_8_f* (if8); [i8@0]; i8@1;
        i9 BY i_9_cya* i_9_f* (if9); [i9@0]; i9@1;
        i10 BY i_10_cya* i_10_f* (if10); [i10@0]; i10@1;
        i11 BY i_11_cya* i_11_f* (if11); [i11@0]; i11@1;
        i12 BY i_12_cya* i_12_f* (if12); [i12@0]; i12@1;
        i13 BY i_13_cya* i_13_f* (if13); [i13@0]; i13@1;
        i14 BY i_14_cya* i_14_f* (if14); [i14@0]; i14@1;
        i15 BY i_15_cya* i_15_f* (if15); [i15@0]; i15@1;
        i16 BY i_16_cya* i_16_f* (if16); [i16@0]; i16@1;
        i17 BY i_17_cya* i_17_f* (if17); [i17@0]; i17@1;

        !identification/scaling constraints;
        [ na@0 ];
        [ child@0 ];  !Child is the reference;
        [ father* ]; !Father is estimated;

        na@1;
        child@1;  !Child is reference;
        father*;   !Father is estimated;

        !Imposing orthogonality of factors
        na WITH father@0;
        na WITH child@0;

        !Imposing orthogonality between negative affect and item-specific factors;
        na WITH i1@0;
        na WITH i2@0;
```

```
        na WITH i3@0;
        na WITH i4@0;
        na WITH i5@0;
        na WITH i6@0;
        na WITH i8@0;
        na WITH i9@0;
        na WITH i10@0;
        na WITH i11@0;
        na WITH i12@0;
        na WITH i13@0;
        na WITH i14@0;
        na WITH i15@0;
        na WITH i16@0;
        na WITH i17@0;

        i1 WITH i2@0 i3@0 i4@0 i5@0 i6@0 i8@0 i9@0 i10@0 i11@0
        i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 father@0 child@0;
        i2 WITH i3@0 i4@0 i5@0 i6@0 i8@0 i9@0 i10@0 i11@0 i12@0
        i13@0 i14@0 i15@0 i16@0 i17@0 father@0 child@0;
        i3 WITH i4@0 i5@0 i6@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0
        i14@0 i15@0 i16@0 i17@0 father@0 child@0;
        i4 WITH i5@0 i6@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 father@0 child@0;
        i5 WITH i6@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 father@0 child@0;
        i6 WITH i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0
        i16@0 i17@0 father@0 child@0;
        i8 WITH i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 child@0;
        i9 WITH i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 child@0;
        i10 WITH i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        father@0 child@0;
        i11 WITH i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 father@0
        child@0;
        i12 WITH i13@0 i14@0 i15@0 i16@0 i17@0 father@0
        child@0;
        i13 WITH i14@0 i15@0 i16@0 i17@0 father@0 child@0;
        i14 WITH i15@0 i16@0 i17@0 father@0 child@0;
        i15 WITH i16@0 i17@0 father@0 child@0;
        i16 WITH i17@0 father@0 child@0;
        i17 WITH father@0 child@0;

        !Adding attachment as a predictor
        father ON aaa_f;


  OUTPUT: modindices (0);

  SAVEDATA: file=tri_factor_fmc_int_constrained_attachment_father_conditional.dat;'

  # Create a file that is stored in the Mplus folder from the text specified in the mplus_model
variable above
writeLines(tri_factor_fmc_int_attachment_code, "Mplus/tri-factor model - fmc - int - conditional -
```

```
attachment - father - constrained - without 7.inp")

# Running the model through the input Mplus file using MplusAutomation
runModels("Mplus/tri-factor model - fmc - int - conditional - attachment - father - constrained -
without 7.inp")
  # An output file will then be saved automatically with the results
```

Extracting the fit statistics from the conditional model: This code has to be run through manually, as it is eval = FALSE

```
# Reading in the data from the output file of the tri-factor model
data <- readModels("Mplus/tri-factor model - fmc - int - conditional - attachment - father -
constrained - without 7.out")

    chi_sq <- chi_sq_df <- chi_sq_p <- "N/A"

    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    # tryCatch({
    #   # Extracting chi-square stats from data$summaries form the output file
    #   chi_sq <- data$summaries$ChiSqM_Value
    #   chi_sq_df <- data$summaries$ChiSqM_DF
    #   chi_sq_p <- data$summaries$ChiSqM_PValue
    #   # Provides the number of warnings in the output file
    # }, error = function(e) {
    # })

    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting warnings from data$warnings form the output file
      # List of all the warnings in the output file
      warn_list <- data$warnings[1]
      # Use [1] after because of some weird nesting of the values
    }, error = function(e) {
        warn_list <- "N/A"
    })

    # List of all the errors in the output file
    error_list <- data$errors[1]
      # Use [1] after because of some weird nesting of the values

    # Create a temporary data file with all the values gathered above for the current iteration of
the subscale to add to the overall data frame.
    tri_fac_fit_int_attach_f_df <- data.frame(chi_sq, chi_sq_df, chi_sq_p, I(warn_list),
I(error_list))
      # I() means to treat the object as it is, so R keeps the list as a list when adding it to the
data frame
    # Changing the column names so that they are always the same, regardless of whether the
variable names are used (when there are no fit indices) or the name of the extracted value is (when
there are fit indices). This is to facilitate binding the data frame with the overall data frame
without getting any errors matching up the columns.
    colnames(tri_fac_fit_int_attach_f_df) <- c("chi_sq", "chi_sq_df", "chi_sq_p", "warnings",
"errors")
```

```
# Print the data frame with the results of the tri-factor models
print(tri_fac_fit_int_attach_f_df)
```

The model converged, but didn't fit properly, since it isn't identified. I will therefore only use the model with the mothers, which does fit the data successfully.

Looking at any errors from fitting the model:

```
tri_fac_fit_int_attach_f_df$errors
```

Looking at any warnings from fitting the model:

```
tri_fac_fit_int_attach_f_df$warnings
```

Getting parameters for each item freed: Extracting the parameters from the conditional tri-factor model with one item freed for internalizing symptoms: This code has to be run through manually, as it is eval = FALSE

```
# Takes a few minutes to run through

# Creating an empty data frame to add the data from each iteration to
tri_fac_parameters_int_attach_f_df <- data.frame(matrix(nrow = 0, ncol = 6))

# Reading in the data from the output file of the tri-factor model
    data <- readModels("Mplus/tri-factor model - fmc - int - conditional - attachment - father - constrained - without 7.out")

    # Extracting the estimates of raters on age from the model
    temp_df <- data$parameters$unstandardized %>% filter(paramHeader == "Intercepts")
    # Extracting the estimates of rater intercepts from the model
    temp_df <- rbind(temp_df, data$parameters$unstandardized %>% filter(param == "AAA_F"))


    # Bind the data frame from the current model to the overall sdq_tri_fac_parameters_df data frame
    tri_fac_parameters_int_attach_f_df <- rbind(tri_fac_parameters_int_attach_f_df, temp_df)


# Changing the misleading name of the est_se column to est_div_se
colnames(temp_df)[colnames(temp_df) == 'est_se'] <- 'est_div_se'

# Saving the data frame with the parameters for later reference
save(tri_fac_parameters_int_attach_f_df, file = "tri_fac_parameters_int_attach_f_df.Rdata")

# Print the data frame with the results of the tri-factor models
print(tri_fac_parameters_int_attach_f_df)
```

These estimates are not interpretable, since the model didn't provide any fit statistics.

Attachment with mother: We had to take out item 7, as it didn't have enough responses in all of the possible response categories

```
# Takes a few minutes to run through

# Creating the code for the input file in Mplus, which defines the data set and the model.
tri_factor_fmc_int_attachment_code  <- '
  TITLE: Constrained conditional tri-factor model for internalizing symptoms;

  DATA: FILE = "Mplus/data_df_wide_diff_mplus.dat";

  VARIABLE:

      NAMES=
      ID aaa_f aaa_m e_1_f e_1_m e_1_cya e_2_f e_2_m e_2_cya e_3_f
      e_3_m e_3_cya e_4_f e_4_m e_4_cya e_5_f e_5_m e_5_cya e_6_f
      e_6_m e_6_cya e_7_f e_7_m e_7_cya e_8_f e_8_m e_8_cya e_9_f
      e_9_m e_9_cya e_10_f e_10_m e_10_cya e_11_f e_11_m e_11_cya
      e_12_f e_12_m e_12_cya e_13_f e_13_m e_13_cya e_14_f e_14_m
      e_14_cya e_15_f e_15_m e_15_cya e_16_f e_16_m e_16_cya
      e_17_f e_17_m e_17_cya e_18_f e_18_m e_18_cya i_1_f i_1_m
      i_1_cya i_2_f i_2_m i_2_cya i_3_f i_3_m i_3_cya i_4_f i_4_m
      i_4_cya i_5_f i_5_m i_5_cya i_6_f i_6_m i_6_cya i_7_f i_7_m
      i_7_cya i_8_f i_8_m i_8_cya i_9_f i_9_m i_9_cya i_10_f i_10_m
      i_10_cya i_11_f i_11_m i_11_cya i_12_f i_12_m i_12_cya i_13_f
      i_13_m i_13_cya i_14_f i_14_m i_14_cya i_15_f i_15_m i_15_cya
      i_16_f i_16_m i_16_cya i_17_f i_17_m i_17_cya;


      USEVAR=
      aaa_m i_1_m i_1_cya i_2_m i_2_cya i_3_m i_3_cya i_4_m i_4_cya
      i_5_m i_5_cya i_6_m i_6_cya i_8_m i_8_cya i_9_m
      i_9_cya i_10_m i_10_cya i_11_m i_11_cya i_12_m i_12_cya i_13_m
      i_13_cya i_14_m i_14_cya i_15_m i_15_cya i_16_m i_16_cya i_17_m
      i_17_cya;

      CATEGORICAL=
      i_1_m i_1_cya i_2_m i_2_cya i_3_m i_3_cya i_4_m i_4_cya
      i_5_m i_5_cya i_6_m i_6_cya i_8_m i_8_cya i_9_m
      i_9_cya i_10_m i_10_cya i_11_m i_11_cya i_12_m i_12_cya i_13_m
      i_13_cya i_14_m i_14_cya i_15_m i_15_cya i_16_m i_16_cya i_17_m
      i_17_cya;

      IDVARIABLE= ID;

      MISSING=.;

  ANALYSIS:

      ESTIMATOR=wlsmv;
      PARAMETERIZATION=theta;
      PROCESSORS=8;

  MODEL:

         !Equality constraints in this model impose invariance
```

```
  !across reporters in thresholds and loadings;

!Item thresholds, subject to equality constraints across reporters;
[i_1_m$1 i_1_cya$1] (it11);
[i_2_m$1 i_2_cya$1] (it12);
[i_3_m$1 i_3_cya$1] (it13);
[i_4_m$1 i_4_cya$1] (it14);
[i_5_m$1 i_5_cya$1] (it15);
[i_6_m$1 i_6_cya$1] (it16);
[i_8_m$1 i_8_cya$1] (it18);
[i_9_m$1 i_9_cya$1] (it19);
[i_10_m$1 i_10_cya$1] (it110);
[i_11_m$1 i_11_cya$1] (it111);
[i_12_m$1 i_12_cya$1] (it112);
[i_13_m$1 i_13_cya$1] (it113);
[i_14_m$1 i_14_cya$1] (it114);
[i_15_m$1 i_15_cya$1] (it115);
[i_16_m$1 i_16_cya$1] (it116);
[i_17_m$1 i_17_cya$1] (it117);


[i_1_m$2 i_1_cya$2] (it21);
[i_2_m$2 i_2_cya$2] (it22);
[i_3_m$2 i_3_cya$2] (it23);
[i_4_m$2 i_4_cya$2] (it24);
[i_5_m$2 i_5_cya$2] (it25);
[i_6_m$2 i_6_cya$2] (it26);
[i_8_m$2 i_8_cya$2] (it28);
[i_9_m$2 i_9_cya$2] (it29);
[i_10_m$2 i_10_cya$2] (it210);
[i_11_m$2 i_11_cya$2] (it211);
[i_12_m$2 i_12_cya$2] (it212);
[i_13_m$2 i_13_cya$2] (it213);
[i_14_m$2 i_14_cya$2] (it214);
[i_15_m$2 i_15_cya$2] (it215);
[i_16_m$2 i_16_cya$2] (it216);
[i_17_m$2 i_17_cya$2] (it217);

!general factor;
na BY i_1_m* (in1);
na BY i_2_m* (in2);
na BY i_3_m* (in3);
na BY i_4_m* (in4);
na BY i_5_m* (in5);
na BY i_6_m* (in6);
na BY i_8_m* (in8);
na BY i_9_m* (in9);
na BY i_10_m* (in10);
na BY i_11_m* (in11);
na BY i_12_m* (in12);
na BY i_13_m* (in13);
na BY i_14_m* (in14);
na BY i_15_m* (in15);
na BY i_16_m* (in16);
```

```
        na BY i_17_m* (in17);

        na BY i_1_cya* (in1);
        na BY i_2_cya* (in2);
        na BY i_3_cya* (in3);
        na BY i_4_cya* (in4);
        na BY i_5_cya* (in5);
        na BY i_6_cya* (in6);
        na BY i_8_cya* (in8);
        na BY i_9_cya* (in9);
        na BY i_10_cya* (in10);
        na BY i_11_cya* (in11);
        na BY i_12_cya* (in12);
        na BY i_13_cya* (in13);
        na BY i_14_cya* (in14);
        na BY i_15_cya* (in15);
        na BY i_16_cya* (in16);
        na BY i_17_cya* (in17);

        !mother specific factor;
        mother BY i_1_m* (ip1);
        mother BY i_2_m* (ip2);
        mother BY i_3_m* (ip3);
        mother BY i_4_m* (ip4);
        mother BY i_5_m* (ip5);
        mother BY i_6_m* (ip6);
        mother BY i_8_m* (ip8);
        mother BY i_9_m* (ip9);
        mother BY i_10_m* (ip10);
        mother BY i_11_m* (ip11);
        mother BY i_12_m* (ip12);
        mother BY i_13_m* (ip13);
        mother BY i_14_m* (ip14);
        mother BY i_15_m* (ip15);
        mother BY i_16_m* (ip16);
        mother BY i_17_m* (ip17);

        !child-specific factor;
        child BY i_1_cya* (ip1);
        child BY i_2_cya* (ip2);
        child BY i_3_cya* (ip3);
        child BY i_4_cya* (ip4);
        child BY i_5_cya* (ip5);
        child BY i_6_cya* (ip6);
        child BY i_8_cya* (ip8);
        child BY i_9_cya* (ip9);
        child BY i_10_cya* (ip10);
        child BY i_11_cya* (ip11);
        child BY i_12_cya* (ip12);
        child BY i_13_cya* (ip13);
        child BY i_14_cya* (ip14);
        child BY i_15_cya* (ip15);
        child BY i_16_cya* (ip16);
        child BY i_17_cya* (ip17);
```

```
!item-specific factors;
i1 BY i_1_cya* i_1_m* (if1); [i1@0]; i1@1;
i2 BY i_2_cya* i_2_m* (if2); [i2@0]; i2@1;
i3 BY i_3_cya* i_3_m* (if3); [i3@0]; i3@1;
i4 BY i_4_cya* i_4_m* (if4); [i4@0]; i4@1;
i5 BY i_5_cya* i_5_m* (if5); [i5@0]; i5@1;
i6 BY i_6_cya* i_6_m* (if6); [i6@0]; i6@1;
i8 BY i_8_cya* i_8_m* (if8); [i8@0]; i8@1;
i9 BY i_9_cya* i_9_m* (if9); [i9@0]; i9@1;
i10 BY i_10_cya* i_10_m* (if10); [i10@0]; i10@1;
i11 BY i_11_cya* i_11_m* (if11); [i11@0]; i11@1;
i12 BY i_12_cya* i_12_m* (if12); [i12@0]; i12@1;
i13 BY i_13_cya* i_13_m* (if13); [i13@0]; i13@1;
i14 BY i_14_cya* i_14_m* (if14); [i14@0]; i14@1;
i15 BY i_15_cya* i_15_m* (if15); [i15@0]; i15@1;
i16 BY i_16_cya* i_16_m* (if16); [i16@0]; i16@1;
i17 BY i_17_cya* i_17_m* (if17); [i17@0]; i17@1;

!identification/scaling constraints;
[ na@0 ];
[ child@0 ];  !Child is the reference;
[ mother* ]; !Mother is estimated;

na@1;
child@1;  !Child is reference;
mother*;   !Mother is estimated;

!Imposing orthogonality of factors
na WITH mother@0;
na WITH child@0;

!Imposing orthogonality between negative affect and item-specific factors;
na WITH i1@0;
na WITH i2@0;
na WITH i3@0;
na WITH i4@0;
na WITH i5@0;
na WITH i6@0;
na WITH i8@0;
na WITH i9@0;
na WITH i10@0;
na WITH i11@0;
na WITH i12@0;
na WITH i13@0;
na WITH i14@0;
na WITH i15@0;
na WITH i16@0;
na WITH i17@0;

i1 WITH i2@0 i3@0 i4@0 i5@0 i6@0 i8@0 i9@0 i10@0 i11@0
i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 mother@0 child@0;
i2 WITH i3@0 i4@0 i5@0 i6@0 i8@0 i9@0 i10@0 i11@0 i12@0
i13@0 i14@0 i15@0 i16@0 i17@0 mother@0 child@0;
```

```
        i3 WITH i4@0 i5@0 i6@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0
        i14@0 i15@0 i16@0 i17@0 mother@0 child@0;
        i4 WITH i5@0 i6@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 mother@0 child@0;
        i5 WITH i6@0 i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0
        i15@0 i16@0 i17@0 mother@0 child@0;
        i6 WITH i8@0 i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0
        i16@0 i17@0 mother@0 child@0;
        i8 WITH i9@0 i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        mother@0 child@0;
        i9 WITH i10@0 i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        mother@0 child@0;
        i10 WITH i11@0 i12@0 i13@0 i14@0 i15@0 i16@0 i17@0
        mother@0 child@0;
        i11 WITH i12@0 i13@0 i14@0 i15@0 i16@0 i17@0 mother@0
        child@0;
        i12 WITH i13@0 i14@0 i15@0 i16@0 i17@0 mother@0
        child@0;
        i13 WITH i14@0 i15@0 i16@0 i17@0 mother@0 child@0;
        i14 WITH i15@0 i16@0 i17@0 mother@0 child@0;
        i15 WITH i16@0 i17@0 mother@0 child@0;
        i16 WITH i17@0 mother@0 child@0;
        i17 WITH mother@0 child@0;

        !Adding attachment as a predictor
        mother ON aaa_m;


  OUTPUT: modindices (0);

  SAVEDATA: file=tri_factor_fmc_int_constrained_attachment_mother_conditional.dat;'

  # Create a file that is stored in the Mplus folder from the text specified in the mplus_model
variable above
writeLines(tri_factor_fmc_int_attachment_code, "Mplus/tri-factor model - fmc - int - conditional -
attachment - mother - constrained - without 7.inp")

# Running the model through the input Mplus file using MplusAutomation
runModels("Mplus/tri-factor model - fmc - int - conditional - attachment - mother - constrained -
without 7.inp")
  # An output file will then be saved automatically with the results
```

Extracting the fit statistics from the conditional model: This code has to be run through manually, as it is eval = FALSE

```
# Reading in the data from the output file of the tri-factor model
data <- readModels("Mplus/tri-factor model - fmc - int - conditional - attachment - mother -
constrained - without 7.out")

    # Assign a starting value of NA for all the fit indices
    chi_sq <- chi_sq_df <- chi_sq_p <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
```

```
      # Extracting chi-square stats from data$summaries form the output file
      chi_sq <- data$summaries$ChiSqM_Value
      chi_sq_df <- data$summaries$ChiSqM_DF
      chi_sq_p <- data$summaries$ChiSqM_PValue
      # Provides the number of warnings in the output file
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model
didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error
block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The
values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # Assign a starting value of NA for the warning variable
    warn_list <- NA
    # Trying to extract the fit indices. If it fails, the tryCatch() block will catch the error and
leave the variables with the initial value of NA
    tryCatch({
      # Extracting warnings from data$warnings form the output file
      # List of all the warnings in the output file
      warn_list <- data$warnings[1]
      # Use [1] after because of some weird nesting of the values
    }, error = function(e) {
      # If there is an error trying to extract the values above (which there will be if the model
didn't fit properly), the tryCatch() block will catch it. There is nothing specified in the error
block, so it will just ignore the error and skip the rest of the code in the tryCatch() block. The
values will therefore remain as "NA", which was specified in the code above the tryCatch() block.
    })

    # List of all the errors in the output file
    error_list <- data$errors[1]
      # Use [1] after because of some weird nesting of the values

    # Create a temporary data file with all the values gathered above for the current iteration of
the subscale to add to the overall data frame.
    tri_fac_fit_int_attach_m_df <- data.frame(chi_sq, chi_sq_df, chi_sq_p, I(warn_list),
I(error_list))
      # I() means to treat the object as it is, so R keeps the list as a list when adding it to the
data frame
    # Changing the column names so that they are always the same, regardless of whether the
variable names are used (when there are no fit indices) or the name of the extracted value is (when
there are fit indices). This is to facilitate binding the data frame with the overall data frame
without getting any errors matching up the columns.
    colnames(tri_fac_fit_int_attach_m_df) <- c("chi_sq", "chi_sq_df", "chi_sq_p", "warnings",
"errors")


# Print the data frame with the results of the tri-factor models
print(tri_fac_fit_int_attach_m_df)
```

Looking at any errors from fitting the model:

```
tri_fac_fit_int_attach_m_df$errors
```

Looking at any warnings from fitting the model:

```
tri_fac_fit_int_attach_m_df$warnings
```

Getting parameters for each item freed: Extracting the parameters from the conditional tri-factor model with one item freed for internalizing symptoms: This code has to be run through manually, as it is eval = FALSE

```
# Takes a few minutes to run through

# Creating an empty data frame to add the data from each iteration to
tri_fac_parameters_int_attach_m_df <- data.frame(matrix(nrow = 0, ncol = 6))

# Reading in the data from the output file of the tri-factor model
    data <- readModels("Mplus/tri-factor model - fmc - int - conditional - attachment - mother -
constrained - without 7.out")

    # Extracting the estimates of raters on age from the model
    temp_df <- data$parameters$unstandardized %>% filter(paramHeader == "Intercepts")
    # Extracting the estimates of rater intercepts from the model
    temp_df <- rbind(temp_df, data$parameters$unstandardized %>% filter(param == "AAA_M"))


    # Bind the data frame from the current model to the overall sdq_tri_fac_parameters_df data
frame
    tri_fac_parameters_int_attach_m_df <- rbind(tri_fac_parameters_int_attach_m_df, temp_df)


# Changing the misleading name of the est_se column to est_div_se
colnames(temp_df)[colnames(temp_df) == 'est_se'] <- 'est_div_se'

# Saving the data frame with the parameters for later reference
save(tri_fac_parameters_int_attach_m_df, file = "tri_fac_parameters_int_attach_m_df.Rdata")

# Print the data frame with the results of the tri-factor models
print(tri_fac_parameters_int_attach_m_df)
```

Making a plot of the estimates on attachment for the thesis:

```
# Bind the mother attachment and father attachment data together into one data frame
tri_fac_parameters_int_attach_m_df %>%
  # Filter for rows with estimate on attachment
filter(param == "AAA_F" | param == "AAA_M") %>%
  mutate(param = case_when(
    param == "AAA_F" ~ "Fathers",
    param == "AAA_M" ~ "Mothers")
    ) %>%
  ggplot(aes(x = param, y = est)) +
  geom_bar(stat = "identity", fill = "grey93", color = "black", width = 0.55) +
  geom_errorbar(aes(ymin = est - se, ymax = est + se), width = 0.25) +
  theme_classic() +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_text(colour = "black"),
```

```
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major.y = element_line(),
        panel.grid.minor.y = element_line(),
        strip.background = element_blank()) +
  geom_hline(aes(yintercept = 0)) +
  labs(x = "Mothers compared to children",
       y = "Estimate on Attachment")
```

10 - Chronbach's alpha

Calculating the alpha for externalizing symptoms for father:

```
alpha_res <- data_df_long %>% select(ext_1_f, ext_2_f, ext_3_f, ext_4_f, ext_5_f, ext_6_f, ext_7_f,
ext_8_f, ext_9_f, ext_10_f, ext_11_f, ext_12_f, ext_13_f, ext_14_f, ext_15_f, ext_16_f, ext_17_f,
ext_18_f) %>% mutate(across(everything(), as.numeric)) %>% psych::alpha()

alpha_res$total
```

```
 raw_alpha std.alpha   G6(smc) average_r      S/N        ase     mean        sd
 0.8379993 0.8493015 0.8558537 0.2384423 5.635766 0.00219602 1.291742 0.2561077
  median_r
 0.2341331
```

Calculating the alpha for externalizing symptoms for mother:

```
alpha_res <- data_df_long %>% select(ext_1_m, ext_2_m, ext_3_m, ext_4_m, ext_5_m, ext_6_m, ext_7_m,
ext_8_m, ext_9_m, ext_10_m, ext_11_m, ext_12_m, ext_13_m, ext_14_m, ext_15_m, ext_16_m, ext_17_m,
ext_18_m) %>% mutate(across(everything(), as.numeric)) %>% psych::alpha()

alpha_res$total
```

```
 raw_alpha std.alpha   G6(smc) average_r      S/N         ase     mean       sd
  0.844834 0.8529657 0.8597845 0.2437335 5.801135 0.002085332 1.309708 0.273293
 median_r
  0.23471
```

Calculating the alpha for externalizing symptoms for child and young adult:

```
alpha_res <- data_df_long %>% select(ext_1_cya, ext_2_cya, ext_3_cya, ext_4_cya, ext_5_cya,
ext_6_cya, ext_7_cya, ext_8_cya, ext_9_cya, ext_10_cya, ext_11_cya, ext_12_cya, ext_13_cya,
ext_14_cya, ext_15_cya, ext_16_cya, ext_17_cya, ext_18_cya) %>% mutate(across(everything(),
as.numeric)) %>% psych::alpha()

alpha_res$total
```

```
 raw_alpha std.alpha   G6(smc) average_r      S/N        ase     mean        sd
 0.8135485  0.822269 0.8252061  0.204472 4.626482 0.00258688 1.383147 0.2739391
  median_r
```

```
0.1985912
```

Calculating the alpha for internalizing symptoms for father:

```
alpha_res <- data_df_long %>% select(int_1_f, int_2_f, int_3_f, int_4_f, int_5_f, int_6_f, int_7_f,
int_8_f, int_9_f, int_10_f, int_11_f, int_12_f, int_13_f, int_14_f, int_15_f, int_16_f, int_17_f)
%>% mutate(across(everything(), as.numeric)) %>% psych::alpha()

alpha_res$total
```

```
raw_alpha std.alpha   G6(smc) average_r      S/N         ase     mean
0.8212408 0.8308517 0.8384799 0.2241684 4.911971 0.002515318 1.323793
       sd median_r
0.2705294 0.214422
```

Calculating the alpha for internalizing symptoms for mother:

```
alpha_res <- data_df_long %>% select(int_1_m, int_2_m, int_3_m, int_4_m, int_5_m, int_6_m, int_7_m,
int_8_m, int_9_m, int_10_m, int_11_m, int_12_m, int_13_m, int_14_m, int_15_m, int_16_m, int_17_m)
%>% mutate(across(everything(), as.numeric)) %>% psych::alpha()

alpha_res$total
```

```
raw_alpha std.alpha   G6(smc) average_r      S/N         ase     mean
0.8348866 0.8403942 0.8486614 0.2364847 5.265436 0.002315471 1.353543
       sd  median_r
0.2944323 0.2272747
```

Calculating the alpha for internalizing symptoms for child and young adult:

```
alpha_res <- data_df_long %>% select(int_1_cya, int_2_cya, int_3_cya, int_4_cya, int_5_cya,
int_6_cya, int_7_cya, int_8_cya, int_9_cya, int_10_cya, int_11_cya, int_12_cya, int_13_cya,
int_14_cya, int_15_cya, int_16_cya, int_17_cya) %>% mutate(across(everything(), as.numeric)) %>%
psych::alpha()

alpha_res$total
```

```
raw_alpha std.alpha   G6(smc) average_r      S/N         ase     mean
0.8479929  0.851835 0.8550595 0.2527221 5.749234 0.002151586 1.508601
       sd  median_r
0.3460224 0.2441082
```

Calculating the alpha for insecure attachment to father:

```
alpha_res <- data_df_wide %>% select(y10AAA1RF_Y, y10AAA2RF_Y, y10AAA3RF_Y, y10AAA4RF_Y_R,
y10AAA5RF_Y, y10AAA6RF_Y, y10AAA7RF_Y, y10AAA8RF_Y, y10AAA9RF_Y, y10AAA10RF_Y) %>%
mutate(across(everything(), as.numeric)) %>% psych::alpha()

alpha_res$total
```

```
raw_alpha std.alpha   G6(smc) average_r      S/N         ase     mean        sd
```

```
0.9101679 0.9144861 0.9234827 0.5167684 10.69401 0.003520616 1.920759 0.832951
 median_r
0.5064688
```

Calculating the alpha for insecure attachment to mother:

```
alpha_res <- data_df_wide %>% select(y10AAA1RM_Y, y10AAA2RM_Y, y10AAA3RM_Y, y10AAA4RM_Y_R,
y10AAA5RM_Y, y10AAA6RM_Y, y10AAA7RM_Y, y10AAA8RM_Y, y10AAA9RM_Y, y10AAA10RM_Y) %>%
mutate(across(everything(), as.numeric)) %>% psych::alpha()

alpha_res$total
```

```
raw_alpha std.alpha   G6(smc) average_r      S/N         ase     mean
 0.864424 0.8741252 0.8815218 0.4098345 6.944401 0.005352134 1.773947
       sd  median_r
0.6605988 0.3858522
```