Empathy and Action

Katharina Hellmund & Nadia Hajighassem

2023-12-09

Contents

Pre-processing

Empathy-Questionnaire

Loading in Data and Cleaning

```r
#No 9 or 11 :) the data jumps.
#Unknown data could be: A1, A6
personality_questionnaire <- read_csv('data/PercAct23_Mocap_personality_questionnaires.csv')%>%
  dplyr::filter(`I was a participant in the Mocap workshop`=="Yep") %>%
  mutate(`Participant ID` = case_when(
    `Participant ID` == "7A" ~ "A7",
    `Participant ID` == "b12" ~ "B12",
    `Participant ID` == "6b" ~ "B6",
    `Participant ID` == "12 A" ~ "A12",
    `Participant ID` == "B01" ~ "B1",
```

```
    `Participant ID` == "A" ~ "A6",
    `Participant ID` == "nnn" ~ "A1",
    `Participant ID` == "H7" ~ "A13",
    TRUE ~ `Participant ID`  # If no match, keep the original value
  ))

empathy_questionnaire <- personality_questionnaire %>%
  select(
    "Participant ID",
    "Gender",
    "Age",
    starts_with("Question")
  ) %>%
  rename_with(~paste0("question:", str_replace_all(.x, "Question|\\[|\\]", "")), starts_with("Question")

empathy_questionnaire <- empathy_questionnaire %>%
  mutate(across(starts_with("question:"), ~as.numeric(str_replace_all(., "[^0-9]", ""))))
```

Dance experience Survey

In order to account for experience with dance we score each participant according to their answers on the survey. We select all questions from Factor 1 and factor 2, and extract only the numeric values (discard characters such as "completely agree" and leave the numeric codes corresponding to such characters).

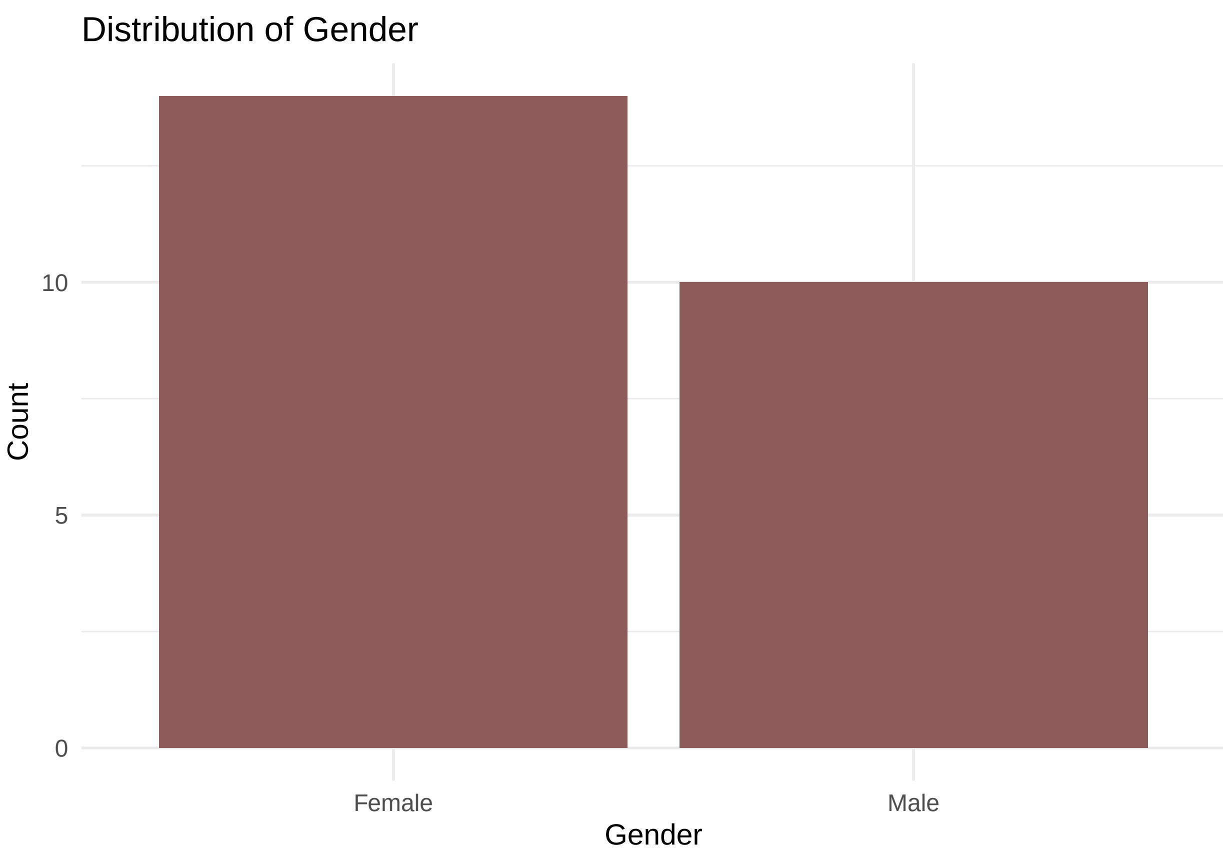
```
dance_survey <- read_csv('data/PercAct23_Mocap_personality_questionnaires.csv')%>%
  dplyr::filter(`I was a participant in the Mocap workshop`=="Yep") %>%
  mutate(`Participant ID` = case_when(
    `Participant ID` == "7A" ~ "A7",
    `Participant ID` == "b12" ~ "B12",
    `Participant ID` == "6b" ~ "B6",
    `Participant ID` == "12 A" ~ "A12",
    `Participant ID` == "B01" ~ "B1",
    `Participant ID` == "A" ~ "A6",
    `Participant ID` == "nnn" ~ "A1",
    `Participant ID` == "H7" ~ "A13",
    TRUE ~ `Participant ID`  # If no match, keep the original value
  ))

dance_survey <- dance_survey %>%
  select(
    `Participant ID`,
    Gender,
    Age,
    starts_with("Factor")
  )

dance_survey <- dance_survey %>%
  mutate(across(starts_with("Factor"), ~as.numeric(str_replace_all(., "[^0-9]", ""))))
```

Motion Capture

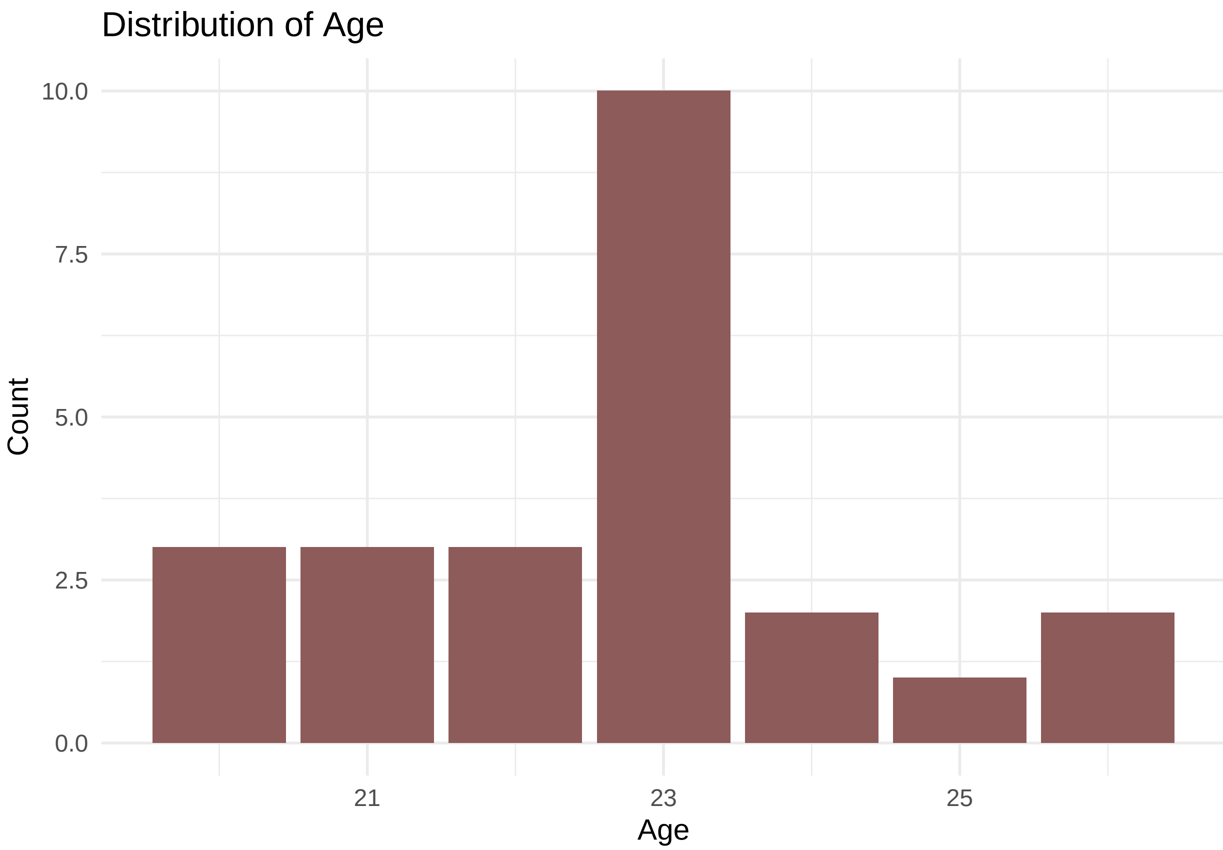
```
synchrony_scores_PCA <- read_csv("results/dtw_results_PCA.csv")
synchrony_scores_ICA <- read_csv("results/dtw_results_ICA.csv")
```

Exploratory data-analysis

```r
#Gender plot
personality_questionnaire %>%
  ggplot(aes(x=Gender))+
  geom_bar(fill= global_fill_colour)+
  labs(title = "Distribution of Gender",
       x = "Gender",
       y = "Count")
```




```r
#Age plot
personality_questionnaire %>%
  ggplot(aes(x=Age))+
  geom_bar(fill = global_fill_colour)+
  labs(title = "Distribution of Age",
       x = "Age",
       y = "Count")
```



Empathy scoring

We start out by making a database for each question, its weight and whether it is reversed or not (counts negatively towards the final score. The EQ-22 short does not provide weights for each question, therefore we assume each question to be weighted the same.

```
empathy_question_weights <- data.frame(
  original_index = c("1",
                     "3",
                     "4",
                     "8",
                     "9",
                     "11",
                     "12",
                     "13",
                     "14",
                     "15",
                     "18",
                     "21",
                     "22",
                     "26",
                     "28",
                     "29",
                     "31",
                     "34",
                     "35",
```

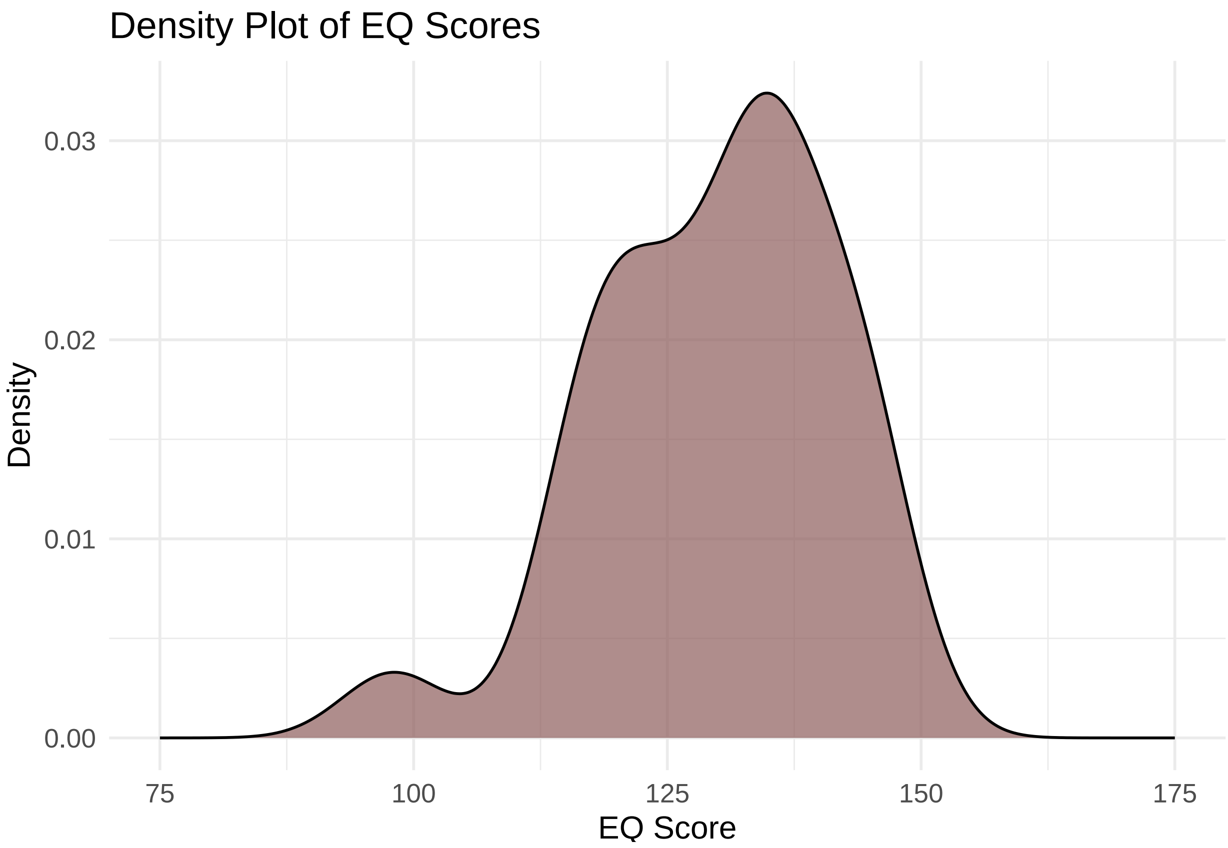
```
            "36",
            "38",
            "39"),
  question = c("I can easily tell if someone else wants to enter a conversation.",
               "I really enjoy caring for other people.",
               "I find it hard to know what to do in a social situation.",
               "I often find it difficult to judge if something is rude or polite.",
               "In a conversation, I tend to focus on my own thoughts rather than on what my listener mi

thinking.",
               "I can pick up quickly if someone says one thing but means another.",
               "It is hard for me to see why some things upset people so much.",
               "I find it easy to put myself in somebody else's shoes.",
               "I am good at predicting how someone will feel.",
               "I am quick to spot when someone in a group is feeling awkward or uncomfortable.",
               "I can't always see why someone should have felt offended by a remark.",
               "I don't tend to find social situations confusing.",
               "Other people tell me I am good at understanding how they are feeling and what they are

thinking.",
               "I can easily tell if someone else is interested or bored with what I am saying.",
               "Friends usually talk to me about their problems as they say that I am very understandin
               "I can sense if I am intruding, even if the other person doesn't tell me.",
               "Other people often say that I am insensitive, though I don't always see why.",
               "I can tune into how someone else feels rapidly and intuitively.",
               "I can easily work out what another person might want to talk about.",
               "I can tell if someone is masking their true emotion.",
               "I am good at predicting what someone will do.",
               "I tend to get emotionally involved with a friend's problems."),
  reverse = c("no",
              "no",
              "yes",
              "yes",
              "yes",
              "no",
              "yes",
              "no",
              "no",
              "no",
              "yes",
              "no",
              "no",
              "no",
              "no",
              "no",
              "yes",
              "no",
              "no",
              "no",
              "no",
              "no")
)
```

Now we make a function for giving each participant an EQ score

```r
# Select columns that start with "question:"
question_cols <- empathy_questionnaire %>%
                 select(starts_with("question:"))

# Assuming empathy_question_weights has the same column names as question_cols and a 'reverse' column
columns_to_reverse <- empathy_question_weights$reverse == "yes"
columns_to_reverse <- names(question_cols)[columns_to_reverse]

# Copy the original dataframe
empathy_questionnaire_reversed <- empathy_questionnaire

# Apply the reversal only to the specified columns
empathy_questionnaire_reversed[columns_to_reverse] <- 11 - empathy_questionnaire_reversed[columns_to_rev

# Calculate the sum of the "question:" columns per row and add it as a new column
empathy_questionnaire_reversed <- empathy_questionnaire_reversed %>%
                                  rowwise() %>%
                                  mutate(EQ_score = sum(c_across(starts_with("question:")), na.rm = TRUE
                                  ungroup()

empathy_questionnaire$EQ_score <- empathy_questionnaire_reversed$EQ_score

write_csv(empathy_questionnaire, "data/EmpathyQuestionnaireScored.csv")

#Removing things from environment to decrease clutter :)
rm(empathy_questionnaire_reversed, empathy_question_weights, question_cols, columns_to_reverse)
```

Plotting the scores

```r
empathy_questionnaire %>%
  ggplot(aes(x=EQ_score)) +
  geom_density(fill = global_fill_colour, alpha=0.7) +  # Adds color to the density plot
  xlim(75,175)+
  labs(title="Density Plot of EQ Scores",
       x="EQ Score",
       y="Density") +  # Add labels and title
  theme(text = element_text(size=12))  # Adjusts the text size for readability
```



Dance Sophistication Scoring

We give each participant a score for body awareness and a score for social dancing according to the GOLD-dsi (The Goldsmiths Dance Sophistication Index) P1 and P2. In similar fashion to EQ-22, some questions count reversed, so we begin by defining those in accordance to the appendix in GOLD-dsi

```
dance_sophistication_weights <- data.frame(
  original_index = c(
    "P1.1",
    "P1.2",
    "P1.3",
    "P1.4",
    "P1.5",
    "P1.6",
    "P2.1",
    "P2.2",
    "P2.3",
    "P2.4",
    "P2.5",
    "P2.6"),
  question = c(
    "I find it easy to learn new movements.",
    "I feel like I have two left feet.",
    "I find it easy to control my movements.",
    "I am not very coordinated.",
    "I am aware of my body and how I hold myself.",
    "I find it easy to learn or imitate other people’s move-
```

```
ments.",
    "If someone asks me to dance, I usually say yes.",
    "I would rather go to a pub than a club so that I do not have
to dance.",
    "I like dancing in front of people.",
    "I find dancing really embarrassing.",
    "Dancing with other people is a great night out as far as
I'm concerned.",
    "You normally have to drag me onto the dance floor
because I'm not really sure what to do. "
  ),
  reverse = c(
    "no",
    "yes",
    "no",
    "yes",
    "no",
    "no",
    "no",
    "yes",
    "no",
    "yes",
    "no",
    "yes"
  )
)
```

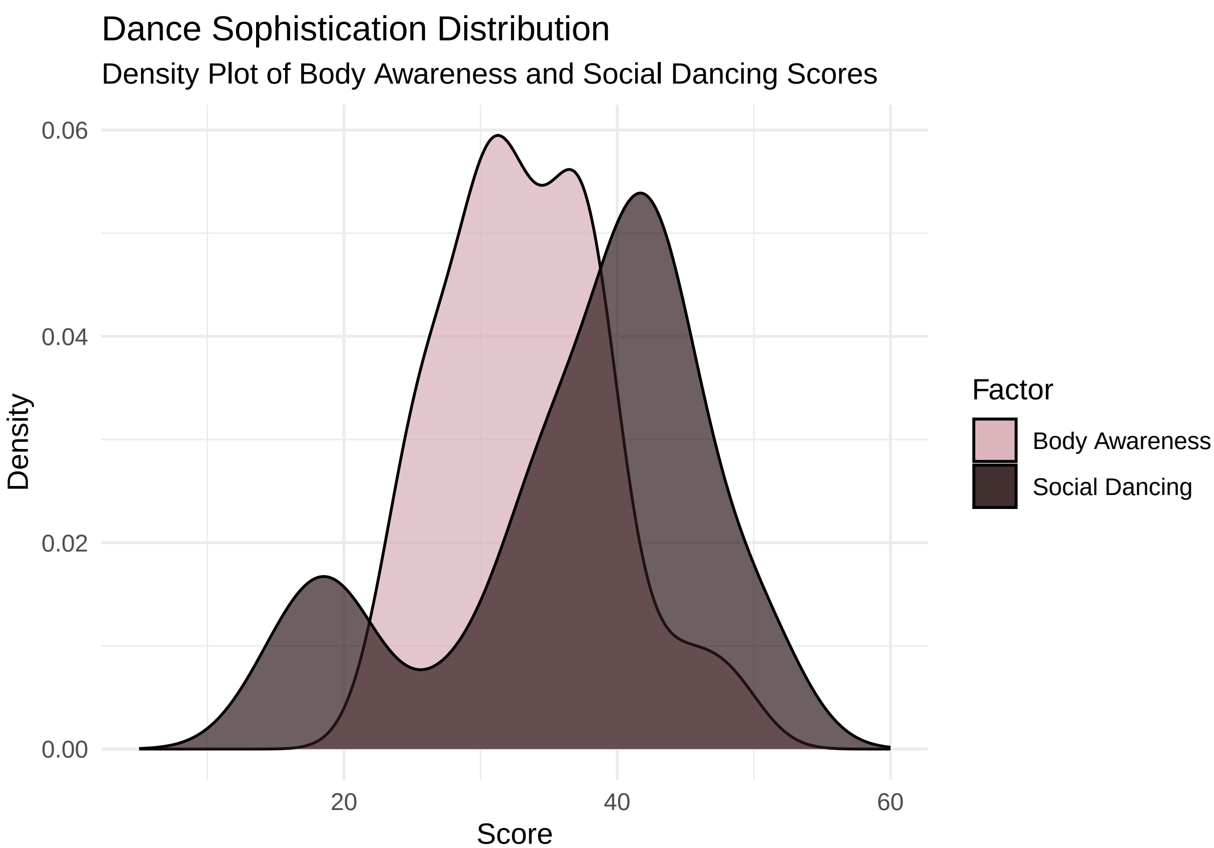
Now we reverse the values for the relevant columns.

```
# Select columns that start with "question:"
question_cols <- dance_survey %>%
                 select(starts_with("Factor"))

# Assuming empathy_question_weights has the same column names as question_cols and a 'reverse' column
columns_to_reverse <- dance_sophistication_weights$reverse == "yes"
columns_to_reverse <- names(question_cols)[columns_to_reverse]

# Copy the original dataframe
dance_survey_reversed <- dance_survey

# Apply the reversal only to the specified columns
dance_survey_reversed[columns_to_reverse] <- 11 - dance_survey_reversed[columns_to_reverse]

# Calculate the sum of the "Factor 1" columns per row and add it as a new column
dance_survey_reversed <- dance_survey_reversed %>%
                                  rowwise() %>%
                                  mutate(Factor1_BodyAwareness_score = sum(c_across(starts_with("Factor
                                  ungroup()

dance_survey$Factor1_BodyAwareness_score <- dance_survey_reversed$Factor1_BodyAwareness_score

# Calculate the sum of the "Factor 2" columns per row and add it as a new column
dance_survey_reversed <- dance_survey_reversed %>%
                                  rowwise() %>%
```

```
                                  mutate(Factor2_SocialDancing_score = sum(c_across(starts_with("Factor
                                  ungroup()

dance_survey$Factor2_SocialDancing_score <- dance_survey_reversed$Factor2_SocialDancing_score

write_csv(dance_survey, "data/dance_survey_scored.csv")

#Removing things from environment to decrease clutter :)
rm(dance_survey_reversed, dance_sophistication_weights, question_cols, columns_to_reverse)
```

Plotting the scores:

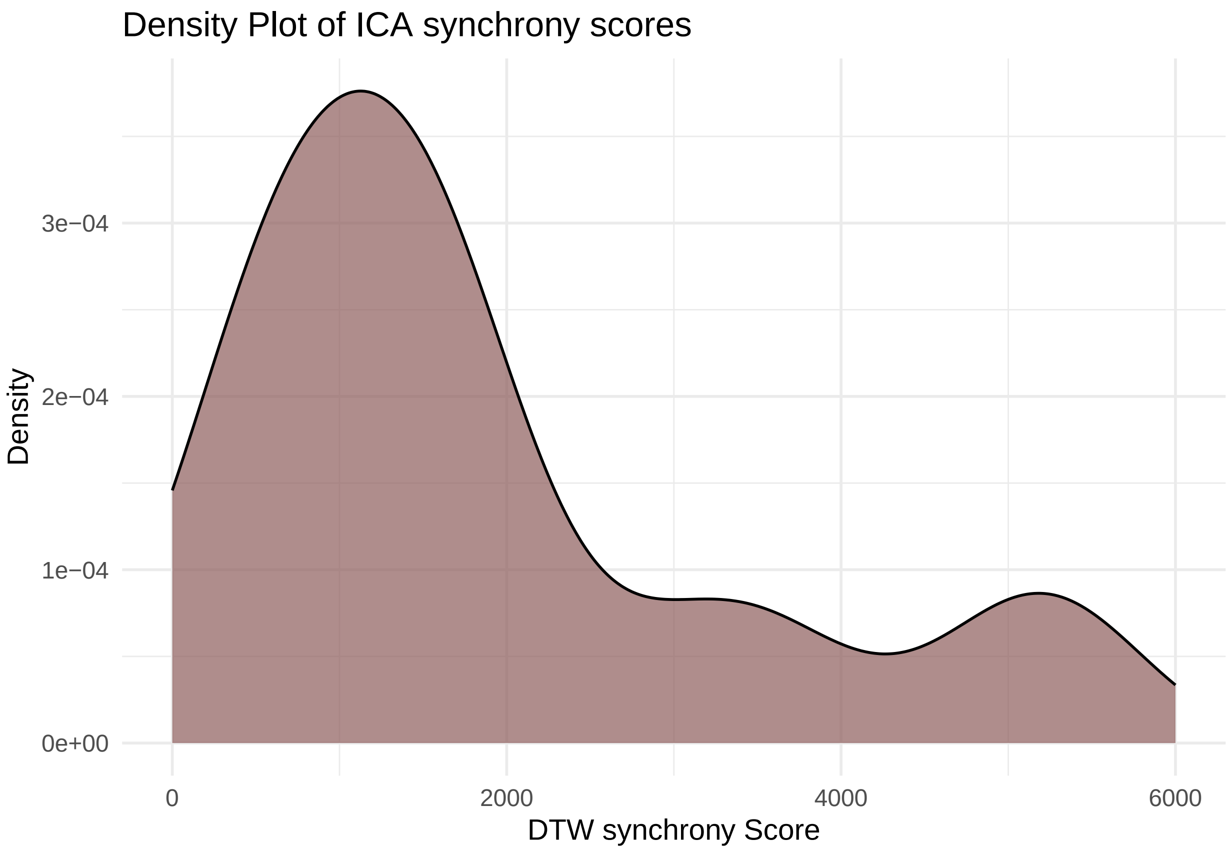
```
dance_survey %>%
  ggplot() +
  geom_density(aes(x = Factor1_BodyAwareness_score, fill = "Body Awareness"), alpha = 0.7) +
  geom_density(aes(x = Factor2_SocialDancing_score, fill = "Social Dancing"), alpha = 0.7) +
  xlim(5, 60) +
  labs(title = "Dance Sophistication Distribution",
       subtitle = "Density Plot of Body Awareness and Social Dancing Scores",
       x = "Score",
       y = "Density",
       fill = "Factor") +
  scale_fill_manual(values = aesthetic_highlight_difference_palette)
```



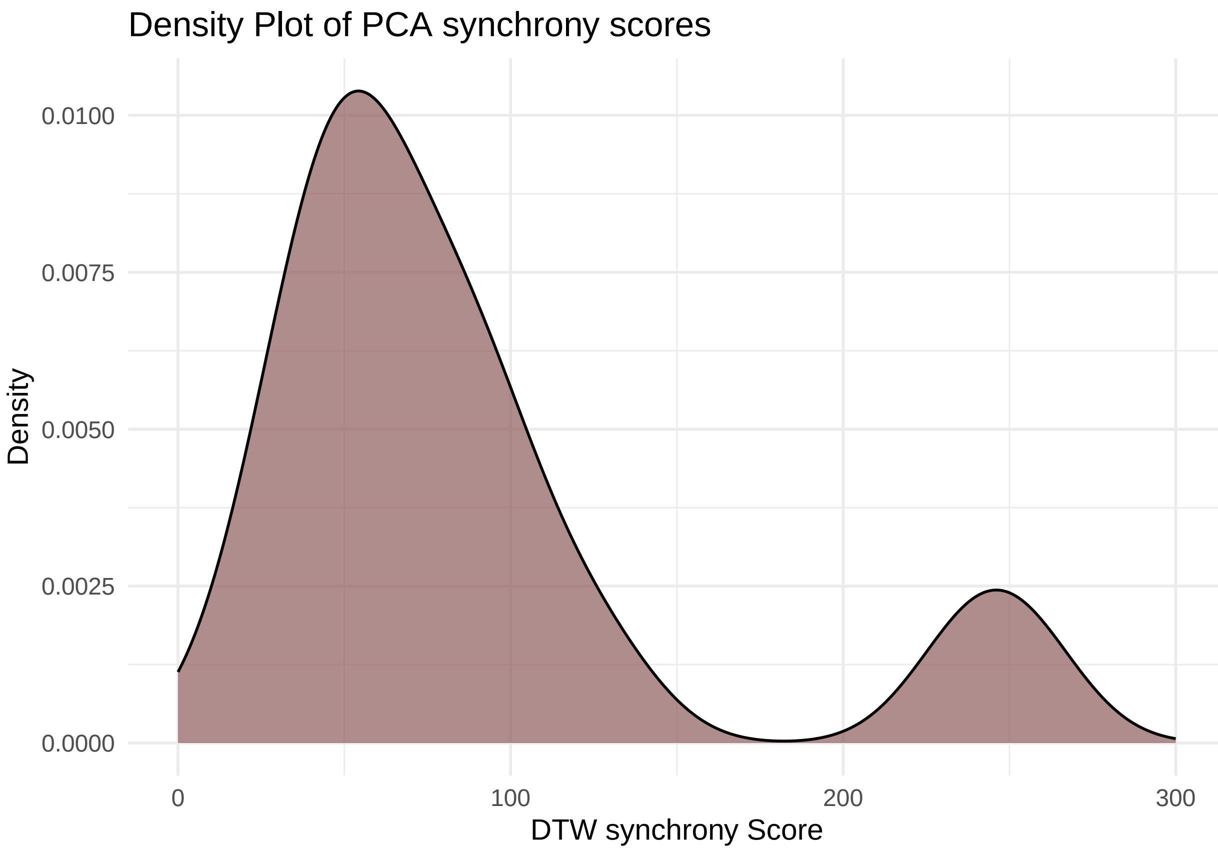
Synchrony

Lets model both synchrony scores for ICA and PCA

```r
synchrony_scores_ICA %>%
  ggplot(aes(x=DTW_Distance)) +
  geom_density(fill = global_fill_colour, alpha=0.7) +
    xlim(0, 6000) +
  labs(title="Density Plot of ICA synchrony scores",
       x="DTW synchrony Score",
       y="Density")
```




```r
synchrony_scores_PCA %>%
  ggplot(aes(x=DTW_Distance)) +
  geom_density(fill = global_fill_colour, alpha=0.7) +
    xlim(0, 300) +
  labs(title="Density Plot of PCA synchrony scores",
       x="DTW synchrony Score",
       y="Density")
```



Going forward, we choose to work with PCA synchrony scores

```
synchrony_scores <- synchrony_scores_ICA
```

Empathy, Dance sophistication, and Synchrony

Before plotting we must unify the dataframes into something usable. There is only one synchrony score per pair, so the A and B pair will have the same score. However, they will each have two scores stemming from the two conditions: Joint Lead and lead follower.

We start by reshaping into a wide format.

```
library(tidyr)

# Reshaping the dataframe
synchrony_scores <- synchrony_scores %>%
  pivot_wider(
    names_from = Condition,
    values_from = DTW_Distance,
    names_prefix = "DTW_"
  )
```

We now combine the synchrony, empathy and dance sophistication scores into one dataframe to be used for analysis.

```
# Selecting and mutating from empathy_questionnaire
eq_df <- empathy_questionnaire %>%
  select(`Participant ID`, Gender, Age, EQ_score) %>%
  mutate(Group = as.numeric(str_extract(`Participant ID`, "\\d+")))
```

```r
# Selecting and mutating from dance_survey
ds_df <- dance_survey %>%
  select(`Participant ID`, Gender, Age, Factor1_BodyAwareness_score, Factor2_SocialDancing_score) %>%
  mutate(Group = as.numeric(str_extract(`Participant ID`, "\\d+")))

# Merging both data frames
complete_df <- full_join(eq_df, ds_df, by = c("Participant ID", "Gender", "Age", "Group"))

# Merge with synchrony_scores
complete_df <- complete_df %>%
  left_join(synchrony_scores, by = "Group")

# Reshape into long-format
complete_df <- complete_df %>%
  pivot_longer(
    cols = starts_with("DTW_"),
    names_to = "DTW_Type",
    values_to = "DTW_Value"
  )

# View the combined dataframe
print(complete_df)
```

```
## # A tibble: 48 x 9
##    `Participant ID` Gender   Age EQ_score Group Factor1_BodyAwareness_score
##    <chr>            <chr>  <dbl>    <dbl> <dbl>                       <dbl>
##  1 A10              Female    23      146    10                          48
##  2 A10              Female    23      146    10                          48
##  3 A3               Male      24      124     3                          37
##  4 A3               Male      24      124     3                          37
##  5 B8               Female    21      136     8                          30
##  6 B8               Female    21      136     8                          30
##  7 A7               Male      22      132     7                          38
##  8 A7               Male      22      132     7                          38
##  9 B1               Female    23      145     1                          32
## 10 B1               Female    23      145     1                          32
## # i 38 more rows
## # i 3 more variables: Factor2_SocialDancing_score <dbl>, DTW_Type <chr>,
## #   DTW_Value <dbl>
```

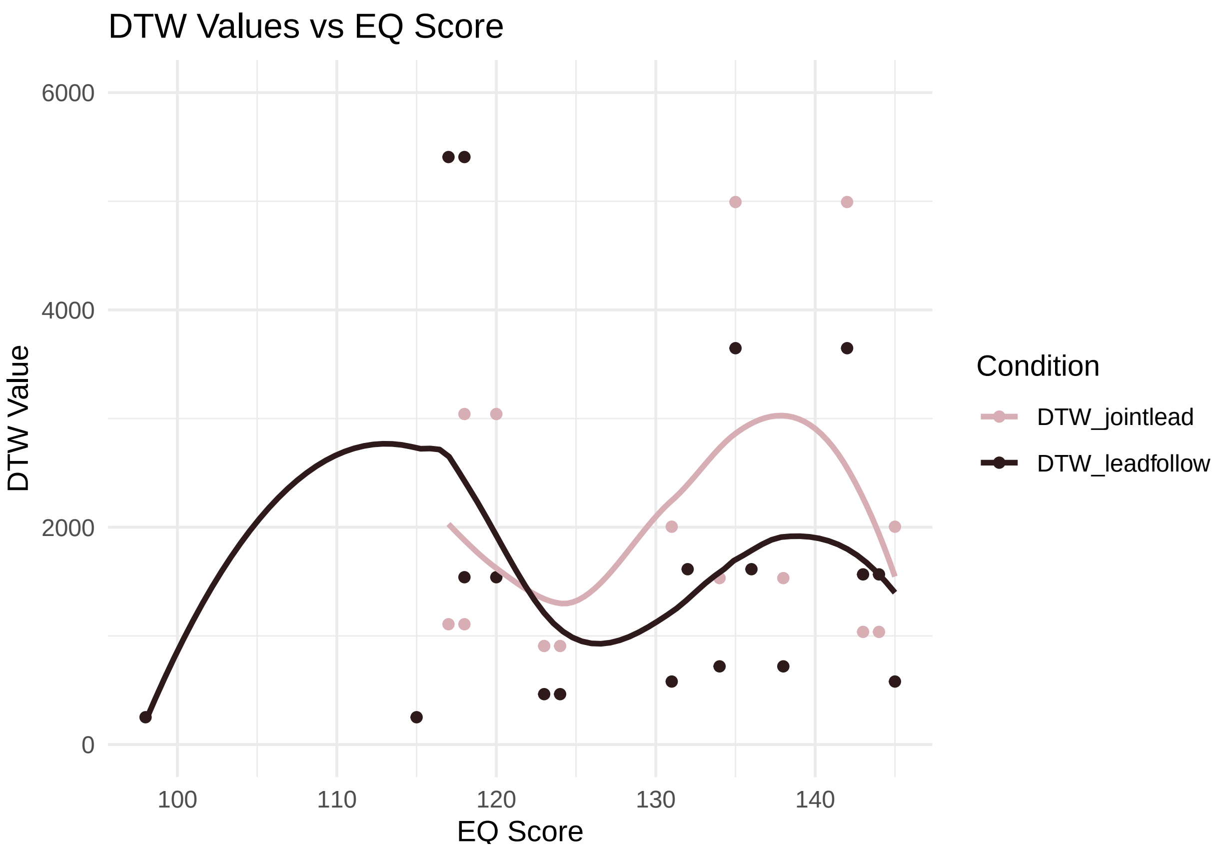
```r
rm(ds_df, eq_df, dance_survey, empathy_questionnaire, personality_questionnaire, synchrony_scores, sync
```

We save our complete dataframe for backup

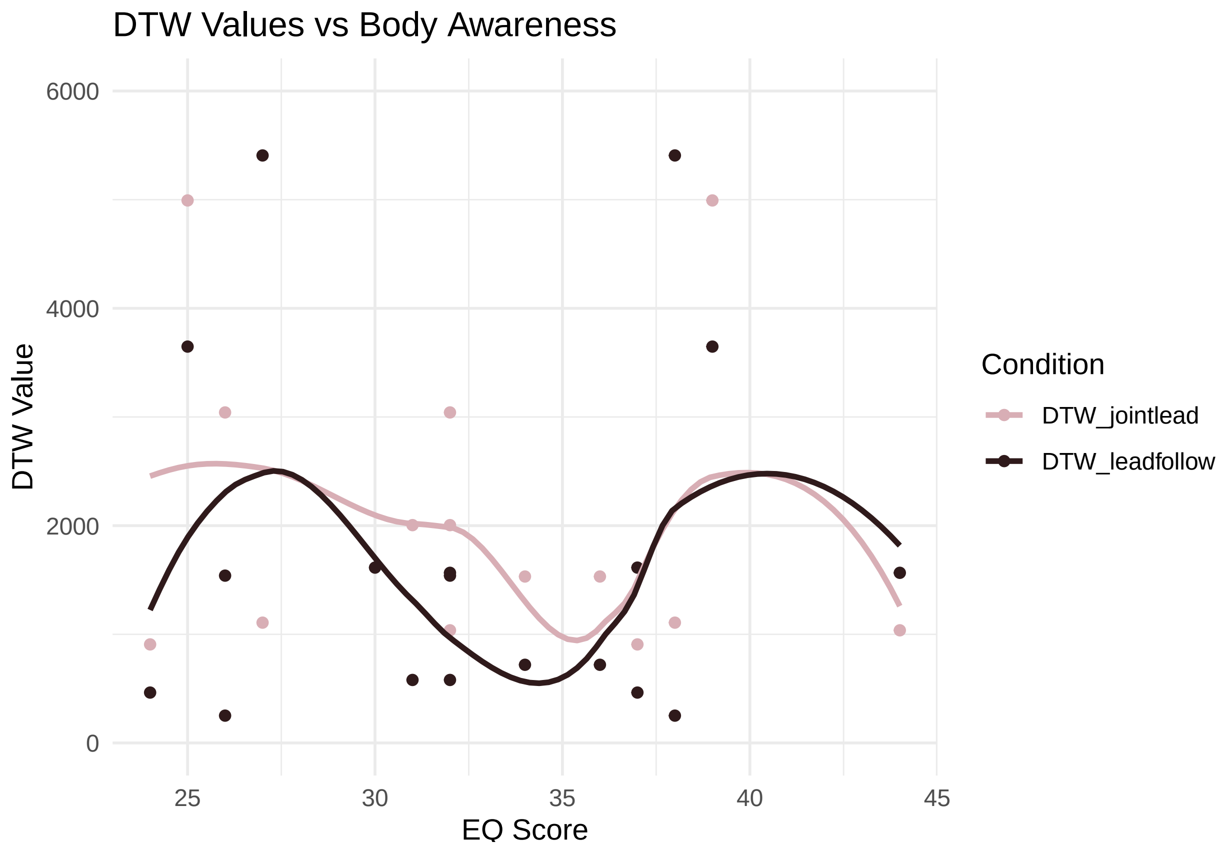
```r
complete_df <- complete_df %>%
  select(-Group)
# Check if the 'results' directory exists, create it if it doesn't
if (!dir.exists("results")) {
  dir.create("results")
}

# Write the dataframe to a CSV file
write.csv(complete_df, "results/actionandempathy_dataset.csv", row.names = FALSE)
```

Now we can plot them to see if there is any relationship:

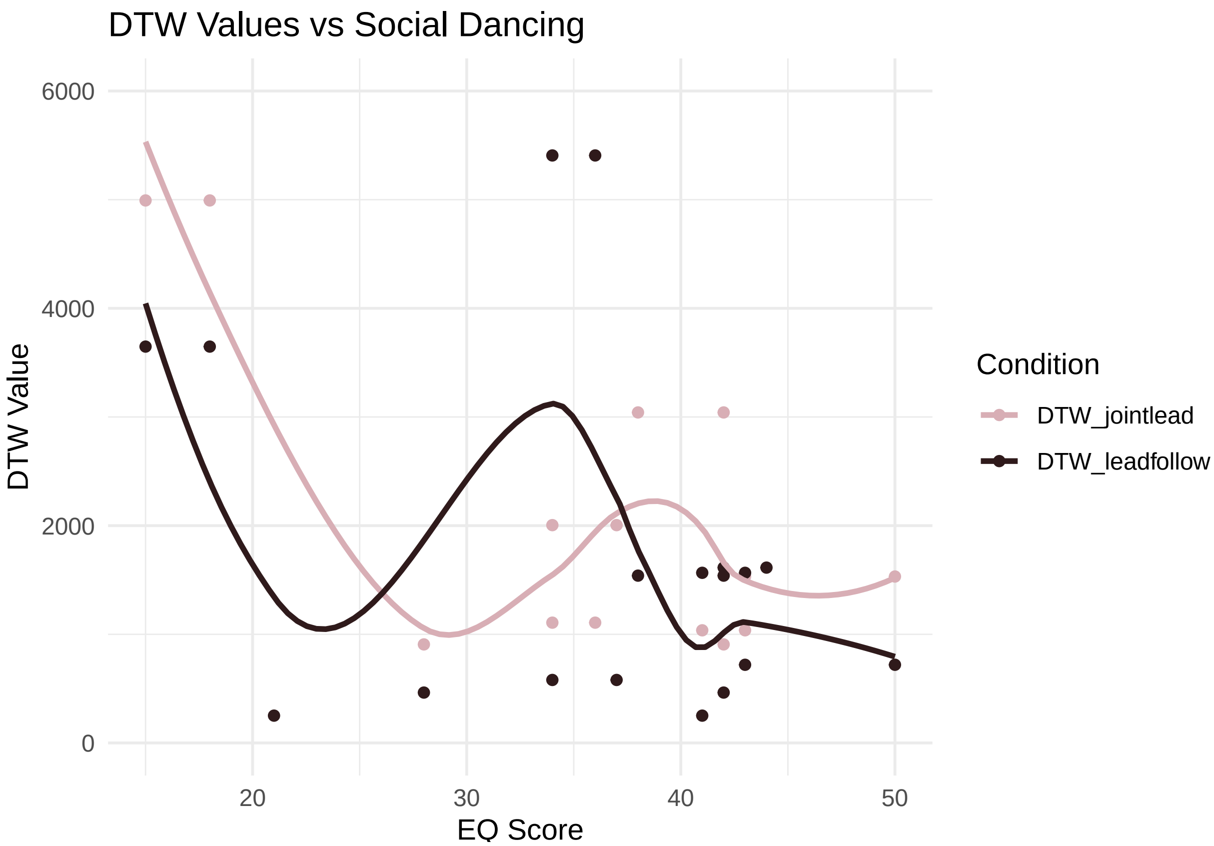
```r
complete_df <- na.omit(complete_df) #OMITTING ALL NAS!!!! SOME DONT HAVE A DTW SCORE
#EQ DTW
ggplot(complete_df, aes(x = EQ_score, y = DTW_Value, color = DTW_Type)) +
  geom_point() +
  geom_smooth(se=FALSE)+
  ylim(0, 6000)+
  labs(
    title = "DTW Values vs EQ Score",
    x = "EQ Score",
    y = "DTW Value",
    color = "Condition"
  )+
  scale_color_manual(values = aesthetic_highlight_difference_palette)
```




```r
#F1 DTW
ggplot(complete_df, aes(x = Factor1_BodyAwareness_score, y = DTW_Value, color = DTW_Type)) +
  geom_point() +
  geom_smooth(se=FALSE)+
  ylim(0, 6000)+
  labs(
    title = "DTW Values vs Body Awareness",
    x = "EQ Score",
    y = "DTW Value",
    color = "Condition"
  )+
  scale_color_manual(values = aesthetic_highlight_difference_palette)
```




```
#F2 DTW
ggplot(complete_df, aes(x = Factor2_SocialDancing_score, y = DTW_Value, color = DTW_Type)) +
  geom_point() +
  geom_smooth(se=FALSE)+
  ylim(0, 6000)+
  labs(
    title = "DTW Values vs Social Dancing",
    x = "EQ Score",
    y = "DTW Value",
    color = "Condition"
  )+
  scale_color_manual(values = aesthetic_highlight_difference_palette)
```



Modelling

We wish to model to account for each participants degree of dance-expertise. We create a linear mixed effects model with following variables:

Dependent variable
DTW_Value

We expect the effect of age and gender to vary across the different clusters, and are not of interest on their own to our model, therefore they are included as random effects.

Independent variables:

Table 3: Table: variables in the mixed-effects model

Fixed effects	Random Effects
EQ_score	Participant ID
Factor1_BodyAwareness_score	(Age)
Factor2_SocialDancing_score	(Gender)
DTW_type	

However, due to the size (or lack thereof) of the dataset, including both age and gender as random effects would lead to overfitting. Therefore, they won't be included in our model, and are only mentioned as in an ideal setting with a large set, they should be included.

Due to the extremely small dataset, we wish to keep our model as simple as possible.

```
library(Matrix)
library(lme4)

model <- lme4::lmer(DTW_Value ~ EQ_score + DTW_Type + Factor1_BodyAwareness_score + Factor2_SocialDanci
```

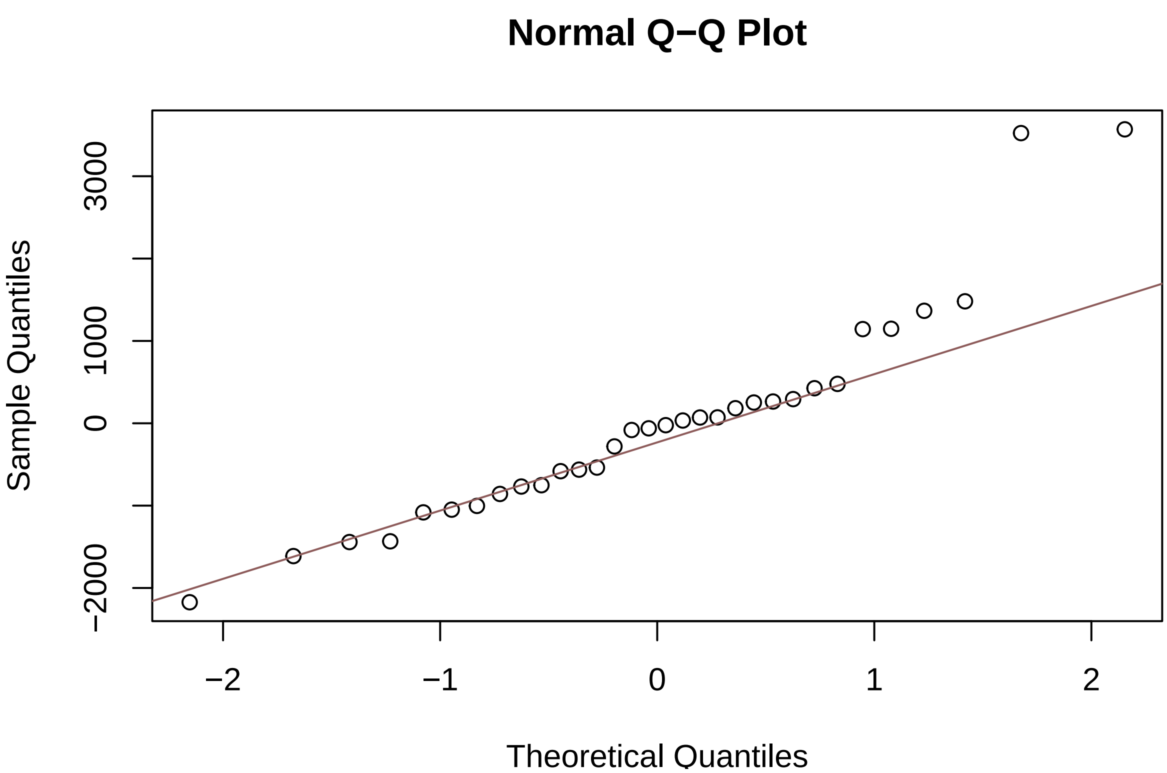
Checking assumptions

In order to verify the validity of our models results, we must check if it upholds the assumptions that mixed effects models are based on.

Normality of residuals:

To check for normality of residuals, we make a Q-Q plot to compare the quantiles of the residuals to the quantiles of a normal distribution.

```
qqnorm(resid(model))
qqline(resid(model), col = global_fill_colour)
```




```
#IF WE WANT TO USE GGPLOT. IF NOT COMMENT OUT

residuals_df <- data.frame(Residuals = resid(model))
ggplot(residuals_df, aes(sample = Residuals)) +
    geom_qq() +
    geom_qq_line(colour = global_fill_colour) +
    labs(title = "Q-Q Plot of Model Residuals",
```

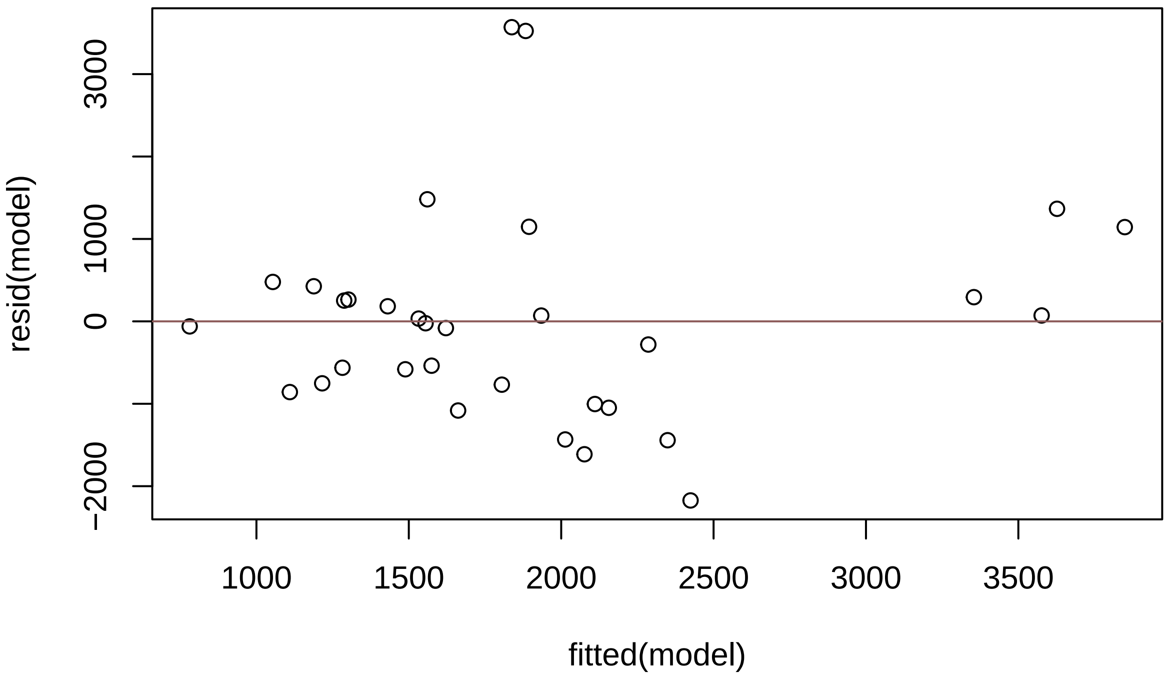
```
         x = "Theoretical Quantiles",
         y = "Sample Quantiles")
```

There is a slight departure from the line towards the tail which indicates a departure from normality.

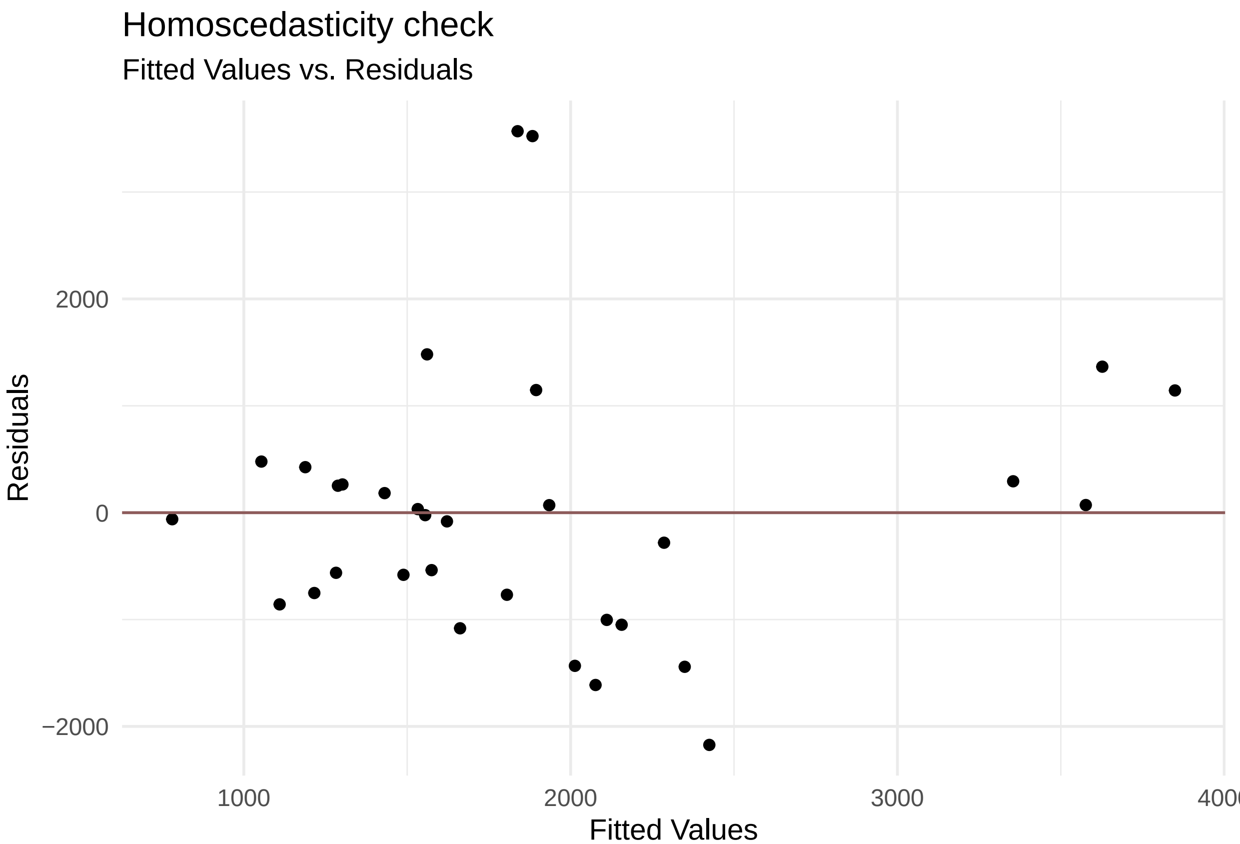
Homoscedasticity (Homogeneity of variance):

We plot the residuals against fitted values, to see if the variance is constant throughout all levels of the independent variables

```
plot(fitted(model), resid(model))
abline(h = 0, col = global_fill_colour)
```




```r
#IF WE WANT TO USE GGPLOT. IF NOT COMMENT OUT
model_data <- data.frame(Fitted = fitted(model), Residuals = resid(model))
ggplot(model_data, aes(x = Fitted, y = Residuals)) +
    geom_point() +
    geom_hline(yintercept = 0, color = global_fill_colour) +
    labs(title="Homoscedasticity check",
         subtitle = "Fitted Values vs. Residuals",
         x = "Fitted Values",
         y = "Residuals")
```



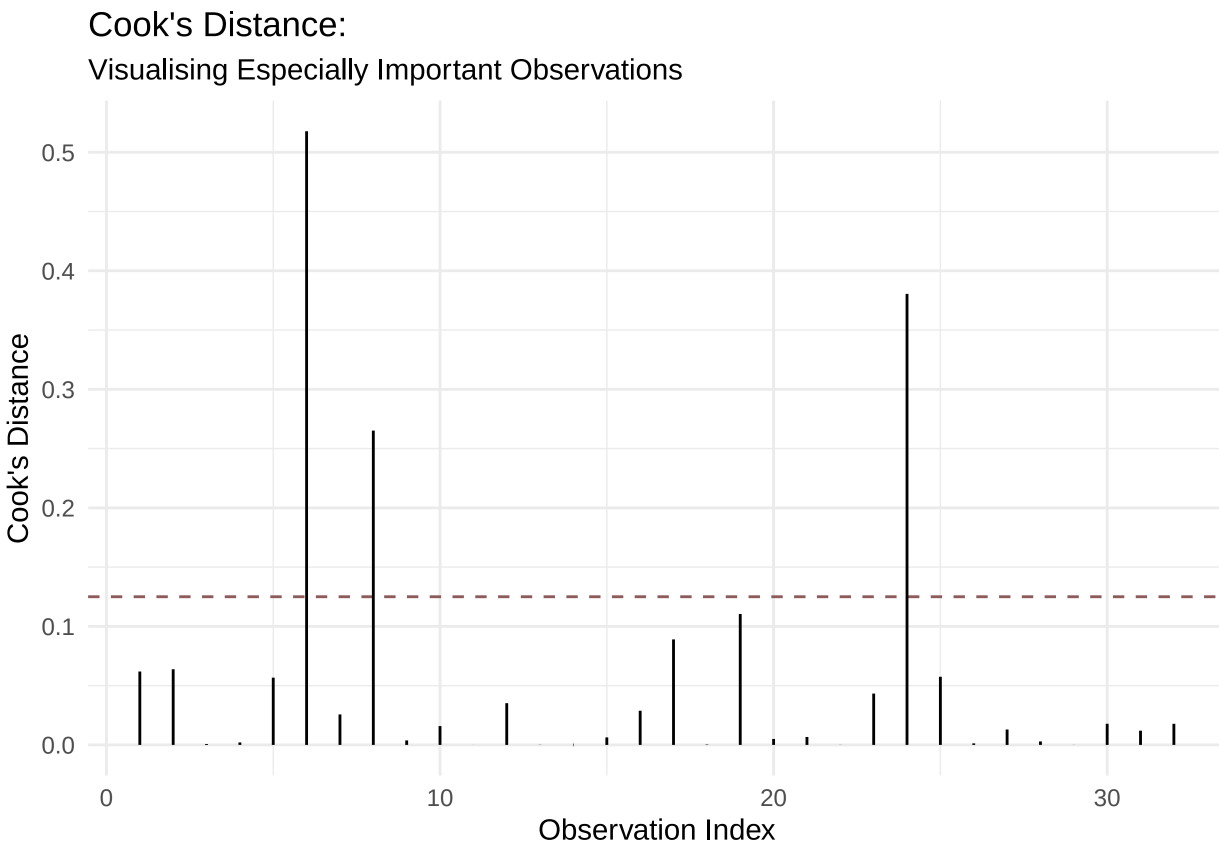
The variance appears somewhat random, but with a small gap between clusters. For larger datasets this could be caused by an underlying bimodal distribution or that we are missing an important predictor, but since our dataset is so small, it is difficult to draw a conclusion.

Though the assumptions are somewhat shaky, we continue with assessing the model fit.

Cook's distance

We wish to know if any observations have a particularily high influence on our model. For that we use Cook's distance and check if any observations exceed the threshold

```r
# Calculate Cook's distance
cooks_dist <- cooks.distance(model)

# Calculate the threshold
threshold <- 4 / length(cooks_dist)

# Create a data frame for plotting
cooks_df <- data.frame(Index = 1:length(cooks_dist), CooksDistance = cooks_dist)

# Plot Cook's distance using ggplot2
ggplot(cooks_df, aes(x = Index, y = CooksDistance)) +
    geom_hline(yintercept = threshold, linetype = "dashed", color = global_fill_colour) +
    geom_segment(aes(x = Index, xend = Index, y = 0, yend = CooksDistance)) +
    labs(title = "Cook's Distance: ",
         subtitle = "Visualising Especially Important Observations",
         y = "Cook's Distance",
         x = "Observation Index")
```




```r
# Find observations where Cook's distance exceeds the threshold
influential_obs <- which(cooks_dist > threshold)

# Print the indices of these observations
print(influential_obs)
```

```
##  6  8 24
##  6  8 24
```

According to Cook's distance, observation 6, 8 and 24 have especially large influence over the model. These observations correspond to the index in complete_df.

OBS: SUBJECT TO CHANGE ONCE WE SWITCH FROM ICA TO PCA :D SHOULD WE REMOVE OUTLIERS?

Model fit and interpretation

```r
print("SUMMARY RESULTS")
```

```
## [1] "SUMMARY RESULTS"
```

```r
summary(model) #GIVE ME P-VALUES!!!! WHY WONT U! NOW I HAVE TO RUN ANOVA!
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: DTW_Value ~ EQ_score + DTW_Type + Factor1_BodyAwareness_score +
##     Factor2_SocialDancing_score + (1 | `Participant ID`)
##    Data: complete_df
##
## REML criterion at convergence: 497.9
```

```
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.55152 -0.56458 -0.03017  0.23326  2.54836
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  Participant ID (Intercept)  127499   357.1
##  Residual                   1961556  1400.6
## Number of obs: 32, groups:  Participant ID, 18
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)                 3402.191   3134.800   1.085
## EQ_score                      12.647     24.399   0.518
## DTW_TypeDTW_leadfollow      -272.805    503.987  -0.541
## Factor1_BodyAwareness_score   -7.986     51.188  -0.156
## Factor2_SocialDancing_score  -75.973     28.281  -2.686
##
## Correlation of Fixed Effects:
##             (Intr) EQ_scr DTW_TD F1_BA_
## EQ_score    -0.825
## DTW_TypDTW_ -0.175  0.108
## Fctr1_BdyA_ -0.126 -0.356 -0.038
## Fctr2_SclD_ -0.241 -0.008 -0.015 -0.139
```

```r
print("ANOVA RESULTS")
```

```
## [1] "ANOVA RESULTS"
```

```r
anova(model)
```

```
## Analysis of Variance Table
##                             npar   Sum Sq  Mean Sq F value
## EQ_score                       1   297569   297569  0.1517
## DTW_Type                       1   713825   713825  0.3639
## Factor1_BodyAwareness_score    1   560076   560076  0.2855
## Factor2_SocialDancing_score    1 14155841 14155841  7.2166
```

Formula
DTW_Value ~ EQ_score + DTW_Type + Factor1_BodyAwareness_score + Factor2_SocialDancing_score + (1 | Participant ID)

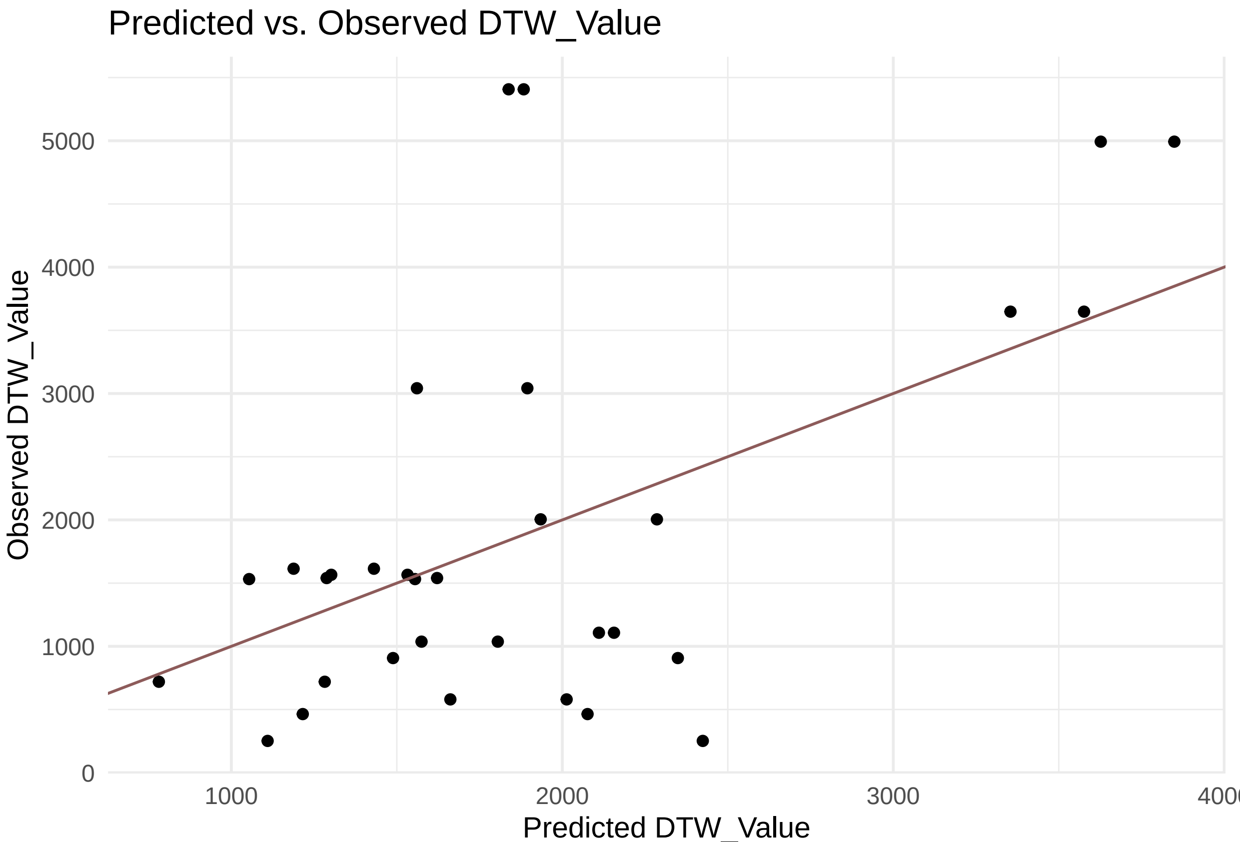
ADD INTERPRETATION AFTER PCA

Visualising

Is the model decent?

```r
predicted_values <- predict(model)
ggplot(complete_df, aes(x = predicted_values, y = DTW_Value)) +
    geom_point() +
    geom_abline(color = global_fill_colour) +
    labs(title = "Predicted vs. Observed DTW_Value",
```

```
         x = "Predicted DTW_Value",
         y = "Observed DTW_Value")
```

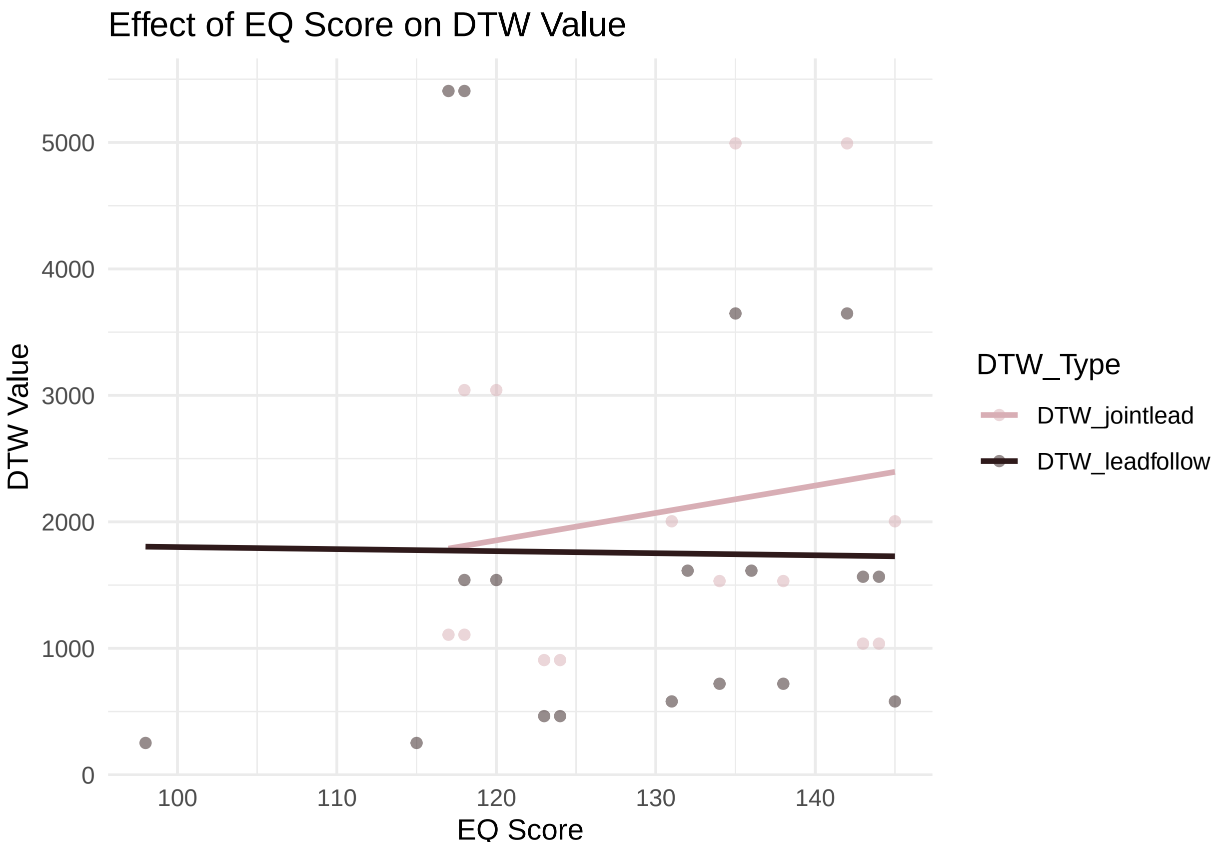


Fixed effects:

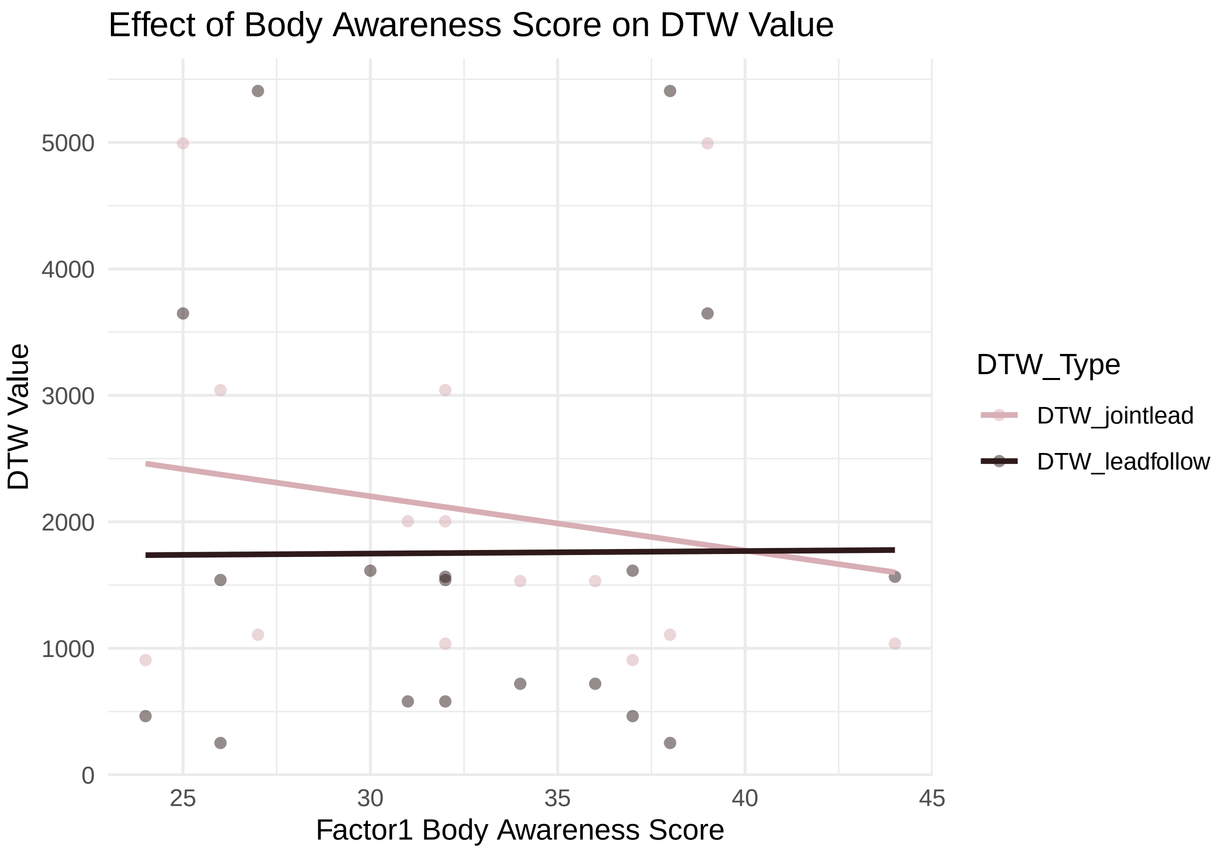
We plot how each fixed effect affects DTW according to our model. However it is important to not that these plots do not show the significance of this prediction!!!

```
complete_df$predicted_DTW_Value <- predict(model, re.form = NA)  # re.form = NA to exclude random effec
```

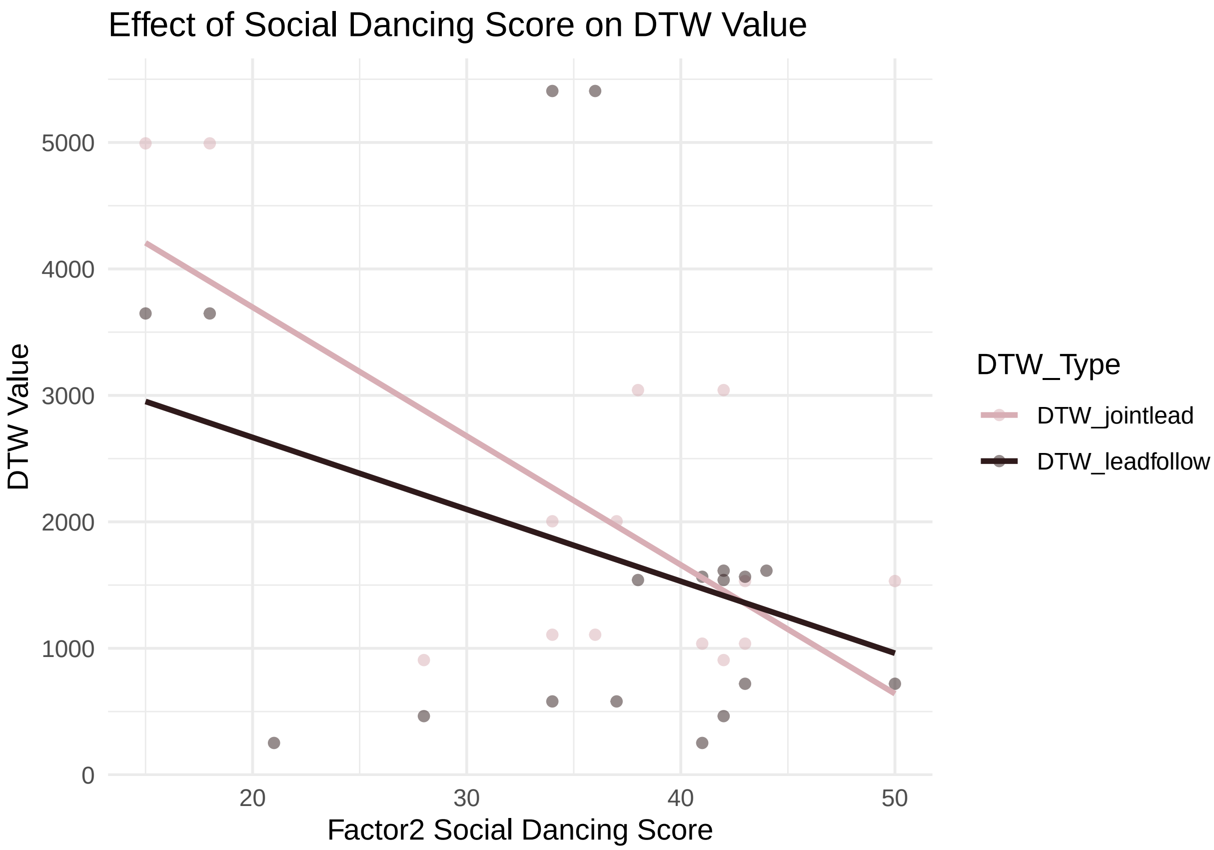
```
ggplot(complete_df, aes(x = EQ_score, y = DTW_Value, color = DTW_Type)) +
    geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", formula = y ~ x, se = FALSE, aes(color = DTW_Type)) +
    labs(title = "Effect of EQ Score on DTW Value",
         x = "EQ Score",
         y = "DTW Value")+
  scale_color_manual(values= aesthetic_highlight_difference_palette)
```




```r
ggplot(complete_df, aes(x = Factor1_BodyAwareness_score, y = DTW_Value, color = DTW_Type)) +
    geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", formula = y ~ x, se = FALSE, aes(color = DTW_Type)) +
    labs(title = "Effect of Body Awareness Score on DTW Value",
         x = "Factor1 Body Awareness Score",
         y = "DTW Value")+
  scale_color_manual(values= aesthetic_highlight_difference_palette)
```

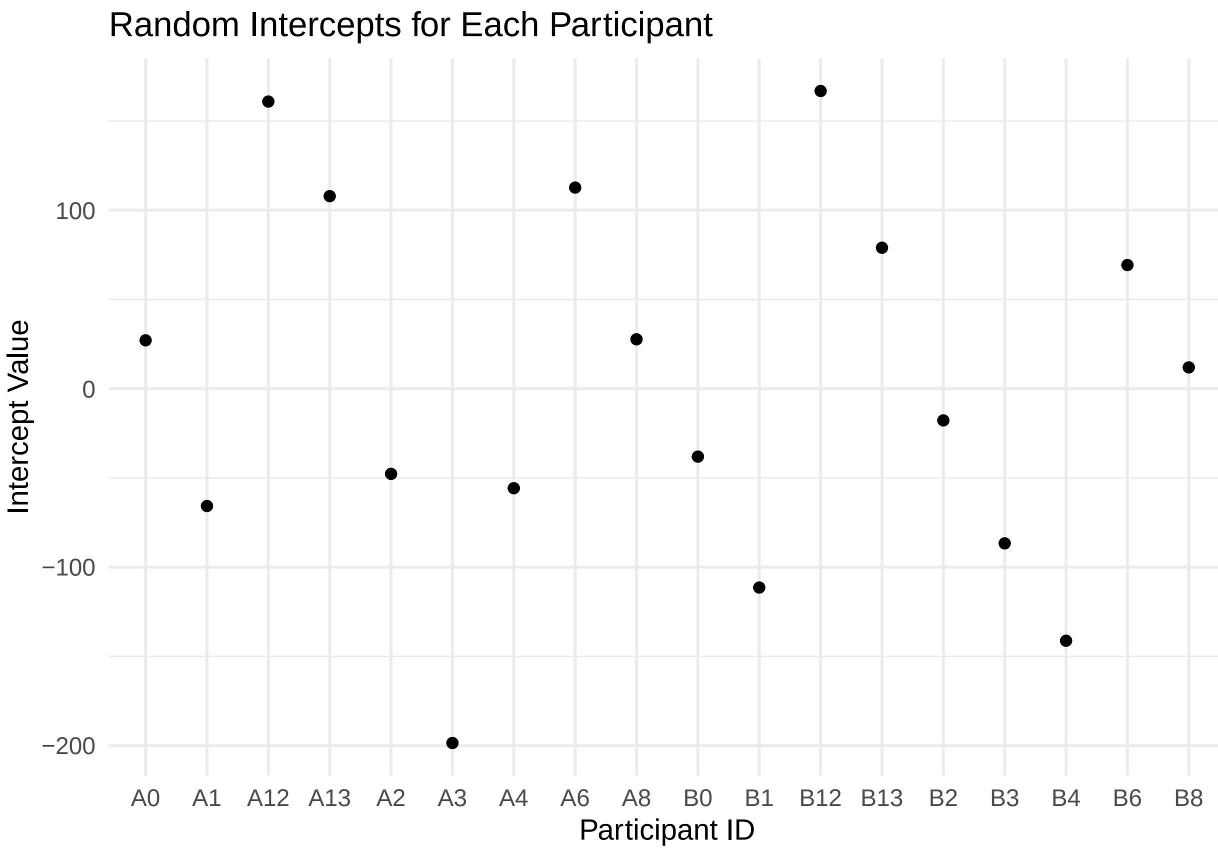



```
ggplot(complete_df, aes(x = Factor2_SocialDancing_score, y = DTW_Value, color = DTW_Type)) +
    geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", formula = y ~ x, se = FALSE, aes(color = DTW_Type)) +
    labs(title = "Effect of Social Dancing Score on DTW Value",
         x = "Factor2 Social Dancing Score",
         y = "DTW Value")+
  scale_color_manual(values= aesthetic_highlight_difference_palette)
```



Random effects:

```
library(lme4)
# Extracting random effects
ran_ef <- ranef(model)$`Participant ID`
ran_ef <- as.data.frame(ran_ef)

# Plotting random intercepts
ggplot(ran_ef, aes(x = rownames(ran_ef), y = `(Intercept)`)) +
    geom_point() +
    labs(title = "Random Intercepts for Each Participant",
         x = "Participant ID",
         y = "Intercept Value")
```



Final interpretation