Analysis Report: Motivated Responses to a Masculinity Threat in a German Cultural Context

2025-12-18

```
df <- read.csv(here("data", "fragile_masculinity_motivation_anonymized.csv"))
df <- preprocessData(df, all_items)
```

```
## Number of Observations: 196
## Condition noThreat: 95 participants
## Condition threat: 101 participants
```

1. Study Inclusion Criteria

1.1 Debrief Consent

Participants that withdrew their consent after the debrief are excluded

```
df <- exclude_participants(
  df,
  DEBRIEFCONSENT == "Y",
  vars = "DEBRIEFCONSENT",
  description = "Debrief consent"
)
```

```
## Debrief consent: Excluded 8 participants (8 of 196).
## Remaining: 188
## Excluded responses summary for `DEBRIEFCONSENT`:  DEBRIEFCONSENT n
## 1              N 8
```

1.2 Demographic criteria

Participants have to be older than 18 years, self-identify as male and native level German skills

```
df <- exclude_participants(df, AGE >= 18, vars = "AGE", description = "Age")
```

```
## Age: Excluded 3 participants (3 of 188).
## Remaining: 185
## Excluded responses summary for `AGE`:  AGE n
## 1   0 1
## 2   3 1
## 3  12 1
```

```
df <- exclude_participants(df, SEX == "A001", vars = "SEX", description = "Gender")
```

```
## Gender: Excluded 9 participants (9 of 185).
## Remaining: 176
## Excluded responses summary for `SEX`:   SEX n
## 1 A002 7
```

```
## 2 A004 2
```

```
df <- exclude_participants(df, GER %in% c("A004", "A005"), vars = "GER", description =
↪  "German skills")
```

```
## German skills: Excluded 1 participants (1 of 176).
## Remaining: 175
## Excluded responses summary for `GER`:   GER n
## 1 A003 1
```

1.3 Suspicion

Participants may not indicate a strong suspicion about the study (coded in column SUSPICIONEXCLUSION, free text are excluded in anonymous data set)

```
df <- exclude_participants(
  df,
  SUSPICIONEXCLUSION == FALSE,
  vars = "SUSPICIONEXCLUSION",
  description = "Suspicion"
)
```

```
## Suspicion: Excluded 9 participants (9 of 175).
## Remaining: 166
## Excluded responses summary for `SUSPICIONEXCLUSION`:  SUSPICIONEXCLUSION n
## 1                TRUE 9
```

1.4 WFCT comprehension

Participants should complete more than 50% of word fragments with existing words

```
df <- exclude_participants(
  df,
  validWordCompletionScore >= 0.5,
  vars = "validWordCompletionScore",
  description = "WFCT comprehension"
)
```

```
## WFCT comprehension: Excluded 11 participants (11 of 166).
## Remaining: 155
## Excluded responses summary for `validWordCompletionScore`:  validWordCompletionScore n
## 1                     0.00 7
## 2                     0.40 2
## 3                     0.05 1
## 4                     0.15 1
```

1.5 Responses to Motivation for Masculine Behaviour

Participants should respond to at least one item of each MMB scale

```
df <- exclude_participants(
  df,
  rowSums(!is.na(df[mmb_pressured_items])) >= 1,
  vars = mmb_pressured_items,
  description = "Missing all Pressured items"
)
```

```
## Missing all Pressured items: Excluded 1 participants (1 of 155).
## Remaining: 154
## Excluded responses summary for `MMBi1`:  MMBi1 n
## 1    NA 1
##
## Excluded responses summary for `MMBi2`:  MMBi2 n
## 1    NA 1
##
## Excluded responses summary for `MMBi3`:  MMBi3 n
## 1    NA 1
##
## Excluded responses summary for `MMBi4`:  MMBi4 n
## 1    NA 1
##
## Excluded responses summary for `MMBi5`:  MMBi5 n
## 1    NA 1
```

```r
df <- exclude_participants(
  df,
  rowSums(!is.na(df[mmb_autonomous_items])) >= 1,
  vars = mmb_autonomous_items,
  description = "Missing all Autonomous items"
)
```

```
## Missing all Autonomous items: Excluded 0 participants (0 of 154).
## Remaining: 154
```

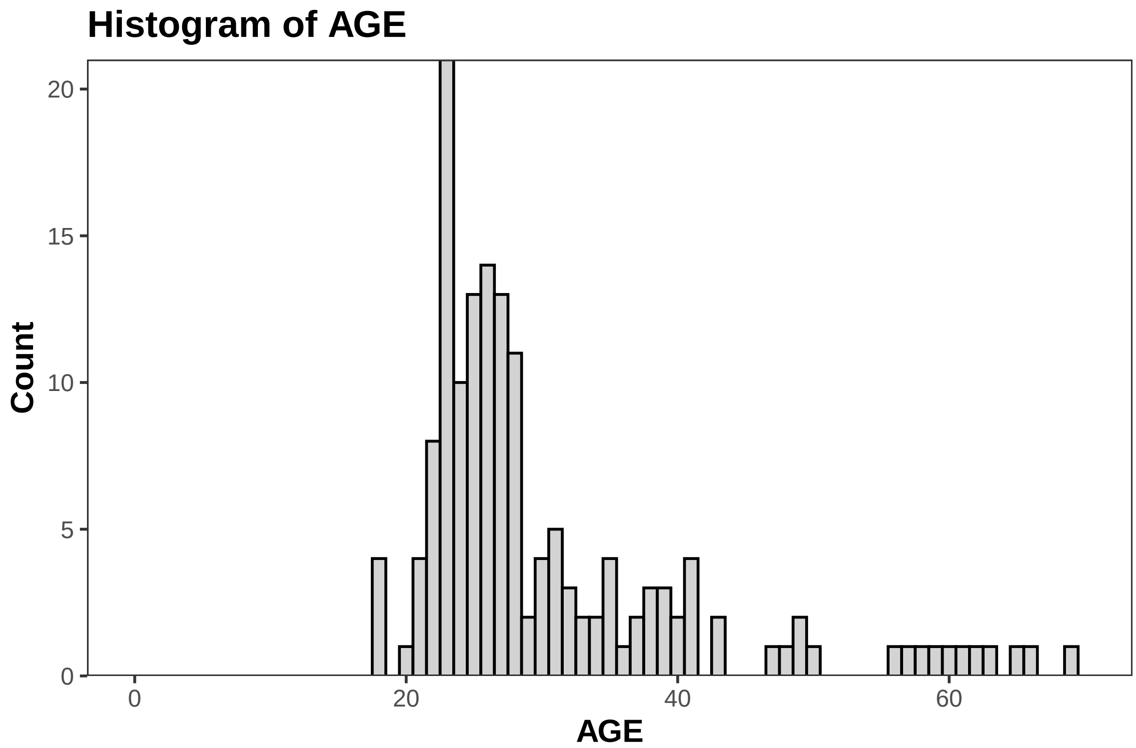
2. Demographics

2.2 Age

```r
summary_mean_sd(df, "AGE")
```

```
## # A tibble: 1 x 5
##   variable  mean    sd     n missing
##   <chr>    <dbl> <dbl> <dbl>   <dbl>
## 1 AGE       30.6  10.9   154       0
```

```r
plotHist("AGE", c(0, 70), condition = NULL, binwidth = 1)
```

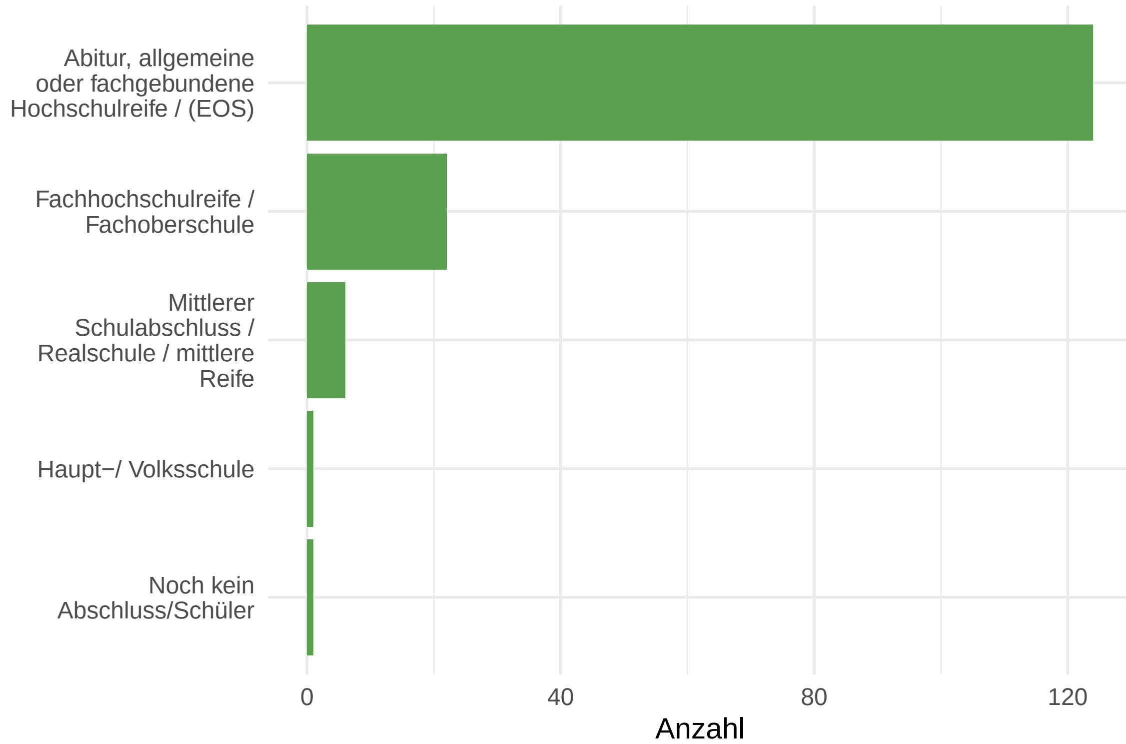


2.3 Educational Degree

```r
freq_table(df, EDU_label)
```

```
## # A tibble: 5 x 3
##   EDU_label                                                              Count Percent
##   <chr>                                                                  <int>   <dbl>
## 1 "Abitur, allgemeine\noder fachgebundene\nHochschulreife / (EOS)"        124    80.5
## 2 "Fachhochschulreife /\nFachoberschule"                                   22    14.3
## 3 "Mittlerer\nSchulabschluss /\nRealschule / mittlere\nReife"               6     3.9
## 4 "Haupt-/ Volksschule"                                                     1     0.6
## 5 "Noch kein\nAbschluss/Schüler"                                            1     0.6
```

```r
plotBarHorizontal("EDU_label") +
  labs(
    x = NULL,
    y = "Anzahl"
  )
```




```r
freq_table(df, EDUPUPIL_label)
```

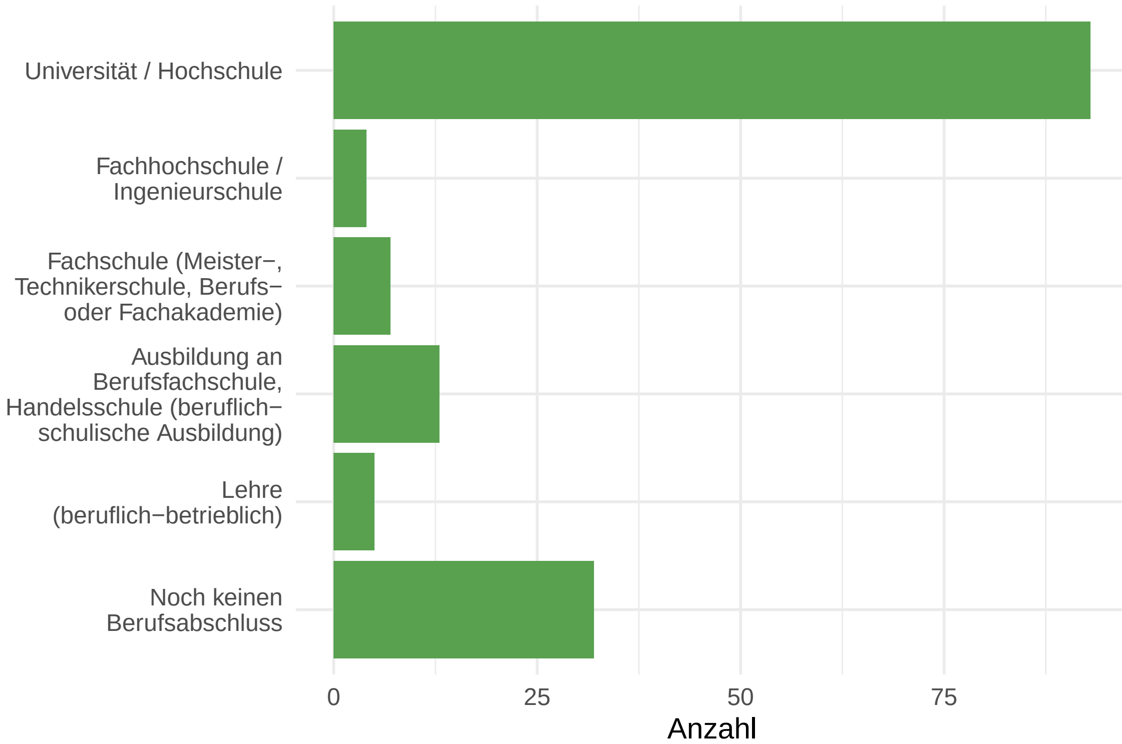
```
## # A tibble: 2 x 3
##   EDUPUPIL_label      Count Percent
##   <chr>               <int>   <dbl>
## 1 <NA>                  153    99.4
## 2 Haupt-/ Volksschule     1     0.6
```

2.4 Occupational Degree

```r
freq_table(df, OCC_label)
```

```
## # A tibble: 6 x 3
##   OCC_label                                                        Count Percent
##   <chr>                                                            <int>   <dbl>
## 1 "Universität / Hochschule"                                          93    60.4
## 2 "Noch keinen\nBerufsabschluss"                                      32    20.8
## 3 "Ausbildung an\nBerufsfachschule,\nHandelsschule (beruflich-\ns~    13     8.4
## 4 "Fachschule (Meister-,\nTechnikerschule, Berufs-\noder Fachakad~     7     4.5
## 5 "Lehre\n(beruflich-betrieblich)"                                     5     3.2
## 6 "Fachhochschule /\nIngenieurschule"                                  4     2.6
```

```r
plotBarHorizontal("OCC_label") +
  labs(
    x = NULL,
    y = "Anzahl"
  )
```



3. Experimental Manipulation Check

3.1 Conditions

```r
printNumberOfParticipants(df, "threatCondition")
```

```
## Number of Observations: 154
## Condition noThreat: 79 participants
## Condition threat: 75 participants
```

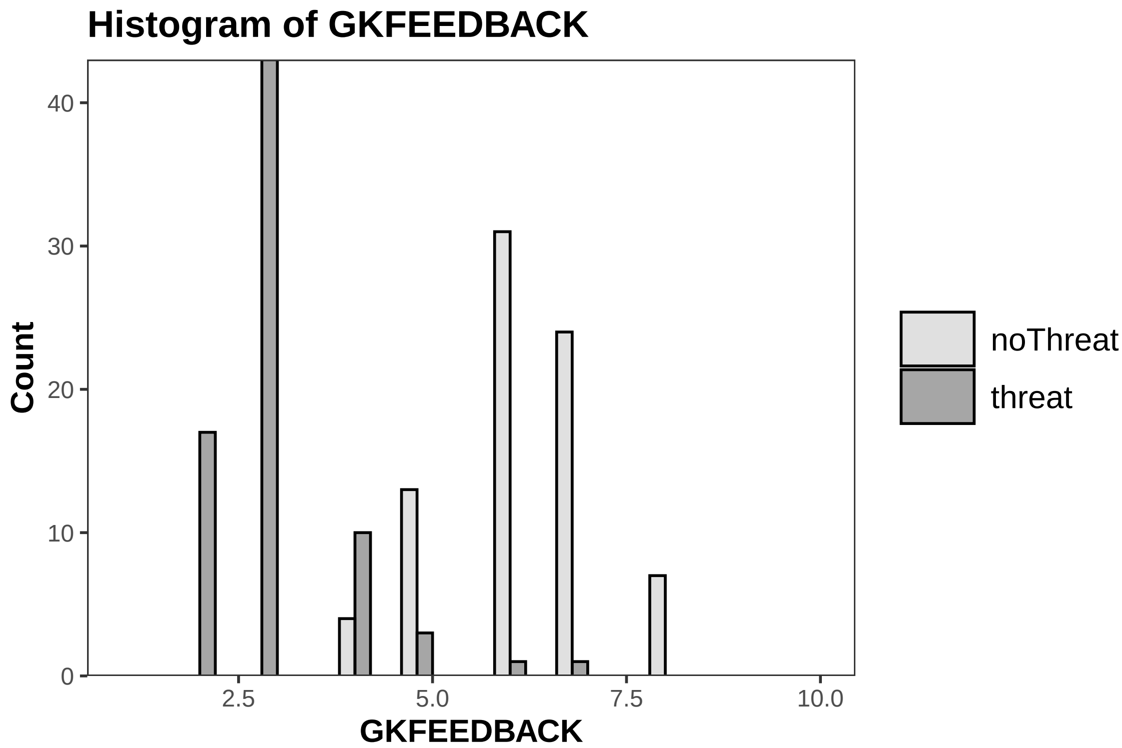
3.2 Gender Knowledge Feedback

[GKFEEDBACK] Welche Rückmeldung haben Sie im Verlauf der Studie zu Ihrem Wissen in geschlechtsspezifischen Themen erhalten? Scale 1 to 10

```r
summary_gkfeedback <- summary_mean_sd(df, "GKFEEDBACK", "threatCondition")
summary_gkfeedback
```

```
## # A tibble: 2 x 6
##   threatCondition variable    mean    sd     n missing
##   <fct>           <chr>      <dbl> <dbl> <dbl>   <dbl>
## 1 noThreat        GKFEEDBACK  6.22  1       79       0
## 2 threat          GKFEEDBACK  3.08  0.93    75       0
```

```r
plotHist("GKFEEDBACK", c(1, 10), "threatCondition")
```



Interpretation

- There is a clar split between the two conditions, meaning that participants paid attention to the feedback on the GenderKnoledge task and that the feedback was comprehensible

3.3 Self evaluation of Gender Knowledge

Objectives:

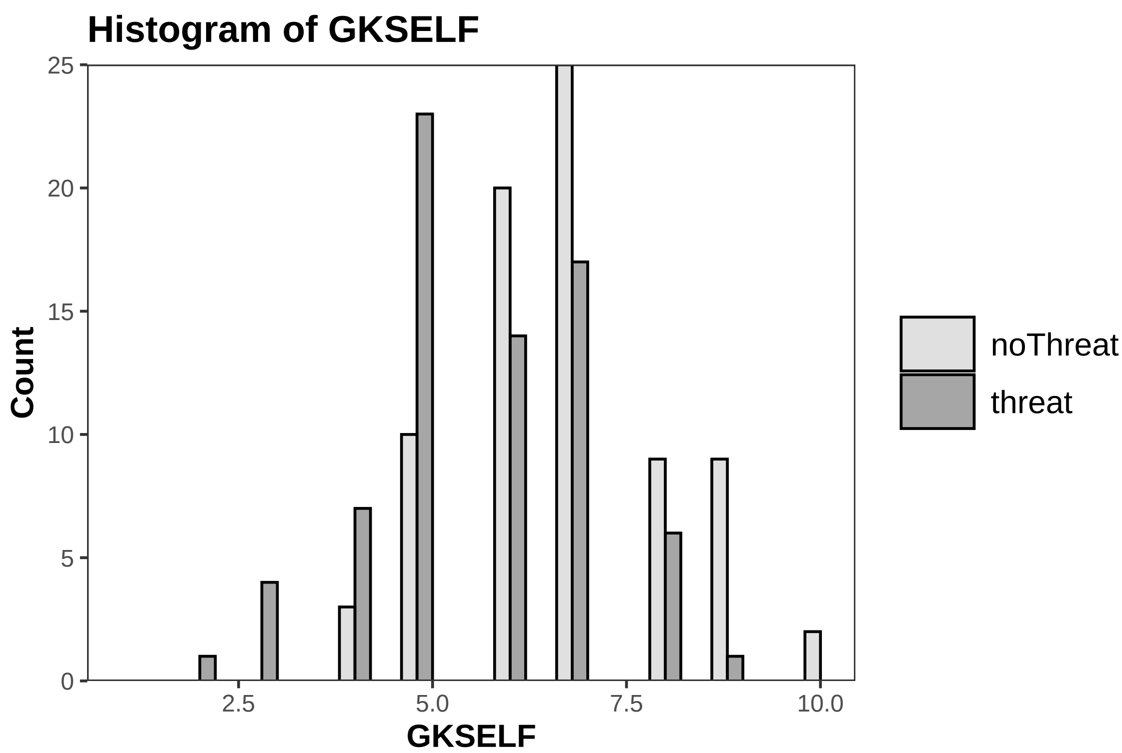
- Testing if self evaluated gender knowledge is significantly less in the threat Condition

Methodology:

- Visualization & Descriptives
- Prerequisites for inference statistics (homogeneity, normal distribution)
- Inference Statistics (t-test)

3.3.1 Visualization & Descriptives

```
plotHist("GKSELF", c(1, 10), "threatCondition")
```



[GKSELF] Wie würden Sie Ihr Wissen in geschlechtsspezifischen Themen einschätzen?

```
summary_gkself <- summary_mean_sd(df, "GKSELF", "threatCondition")
summary_gkself
```

```
## # A tibble: 2 x 6
##   threatCondition variable  mean    sd     n missing
##   <fct>           <chr>    <dbl> <dbl> <dbl>   <dbl>
## 1 noThreat        GKSELF    6.79  1.38    78       1
## 2 threat          GKSELF    5.71  1.43    73       2
```

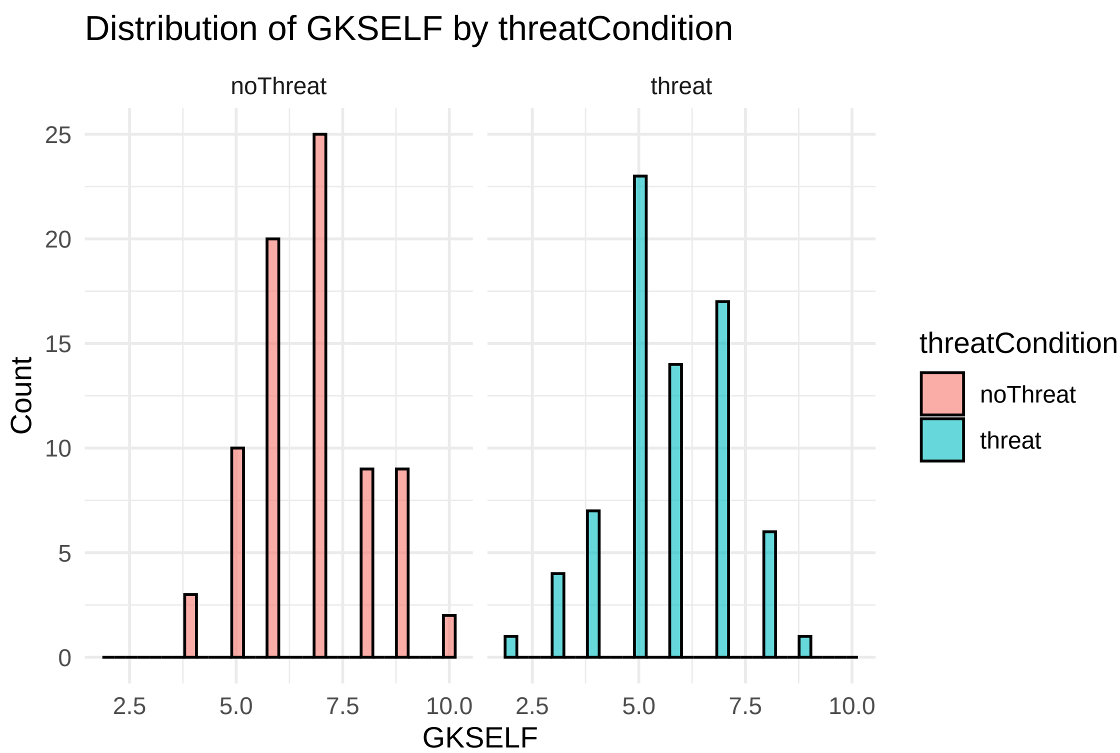
Interpretation

- The mean of self evalutated gender knowledge is lower in the threat condition, than noThreat condition

3.3.2 Inference Statistical Comparison

Normality of distribution

```
check_normality_hist(
  df, GKSELF,
  threatCondition
)
```



Homogenity of Variance (Levene Test)

```
check_homogeneity(
  df, GKSELF,
  threatCondition
)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1    0.61  0.436
##       149
```

p > .05 → equal variances (use Student's t-test: var.equal = TRUE)

Interpretation:

- A visual inspection confirms normally distributed data for both conditions
- The Levene test confirms homogeneity of variance

```
run_ttest(df, GKSELF, threatCondition,
  alternative = "greater", var_equal = TRUE
)
```

```
##
##  Two Sample t-test
##
## data:  GKSELF by threatCondition
## t = 4.735, df = 149, p-value = 2.527e-06
## alternative hypothesis: true difference in means between group noThreat and group threat is greater t
## 95 percent confidence interval:
##  0.7041376       Inf
## sample estimates:
```

```
## mean in group noThreat   mean in group threat
##                6.794872               5.712329
```

Interpretation:

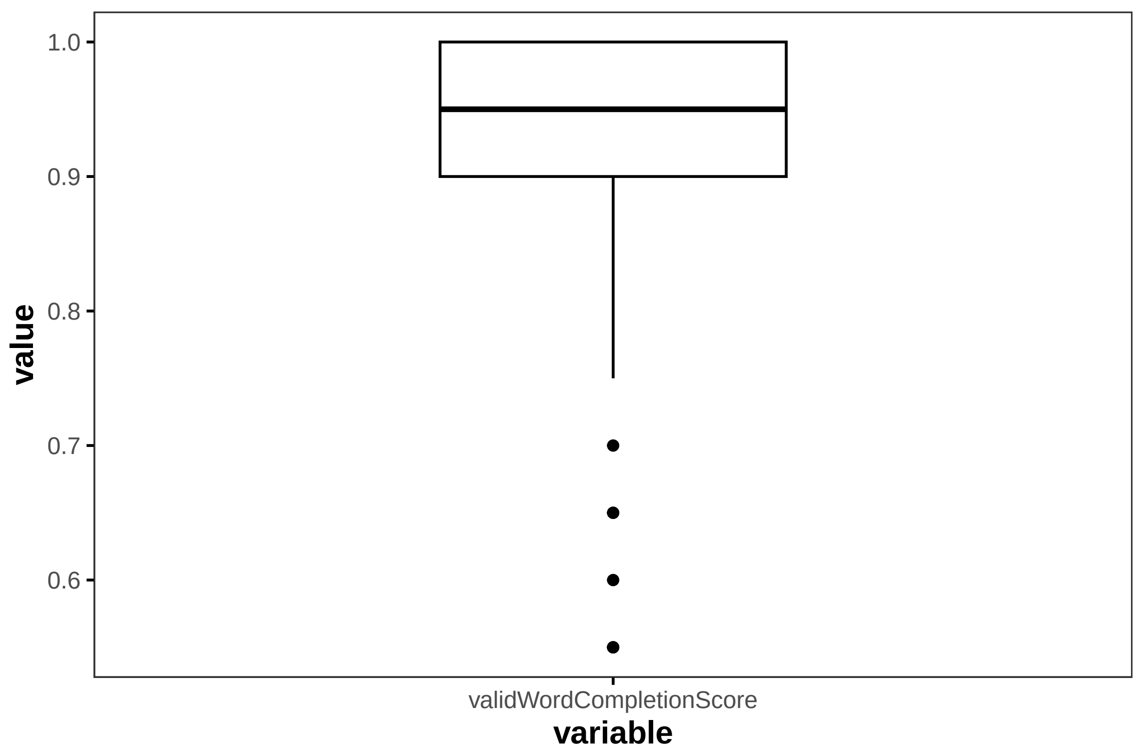
- The p-value is highly significant ($p < 0.01$) showing that the threat condition had a significant effect on the self evaluation of gender specific knowledge (i.e. lower self evaluation)

3.4 Valid Word Completions

```r
summary_mean_sd(df, "validWordCompletionScore")
```

```
## # A tibble: 1 x 5
##   variable                  mean    sd     n missing
##   <chr>                    <dbl> <dbl> <dbl>   <dbl>
## 1 validWordCompletionScore  0.92   0.1   154       0
```

```r
plotBox("validWordCompletionScore")
```



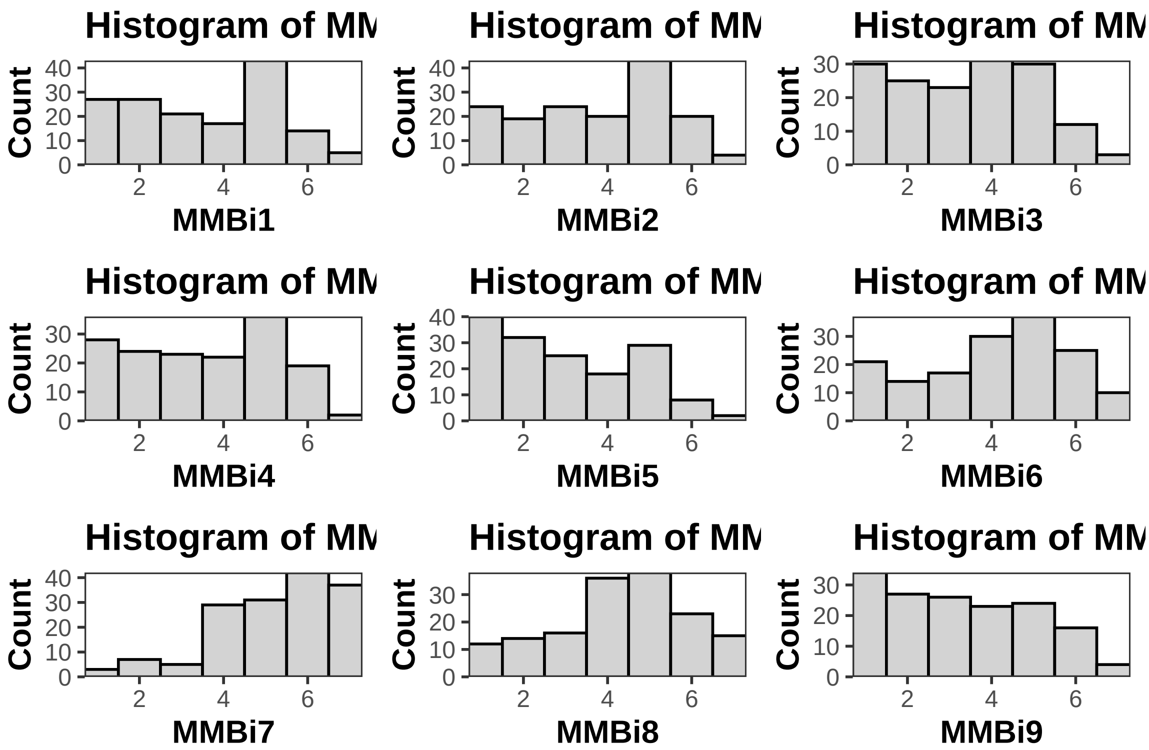
Interpretation

- Participants with a score lower than 50% were excluded
- Generally participants were able to complete the WCFT

3.5 Motivation for Masculine Behaviour

```r
plots <- lapply(MMB_item_codes, function(item) {
  plotHist(item, c(1, 7), condition = NULL, binwidth = 1)
})
```

```r
do.call(grid.arrange, c(plots, ncol = 3))
```

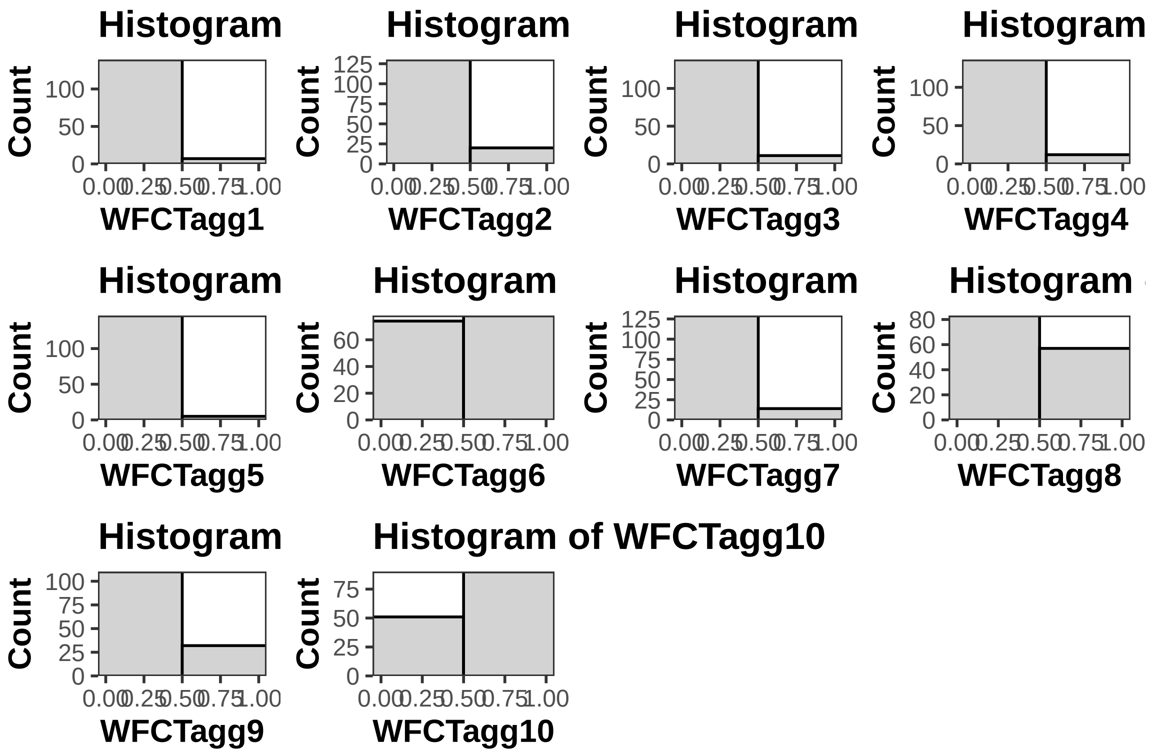



```r
summary_mean_sd(df, MMB_item_codes)
```

```
## # A tibble: 9 x 5
##   variable  mean    sd     n missing
##   <chr>    <dbl> <dbl> <dbl>   <dbl>
## 1 MMBi1     3.55  1.78   154       0
## 2 MMBi2     3.75  1.74   154       0
## 3 MMBi3     3.35  1.68   154       0
## 4 MMBi4     3.51  1.74   154       0
## 5 MMBi5     2.97  1.68   154       0
## 6 MMBi6     4.06  1.79   154       0
## 7 MMBi7     5.29  1.49   154       0
## 8 MMBi8     4.32  1.68   154       0
## 9 MMBi9     3.26  1.77   154       0
```

3.6 Aggressive Cognition

```r
plots <- lapply(WFCT_aggression_items, function(item) {
  plotHist(item, c(0, 1), condition = NULL, binwidth = 1)
})

do.call(grid.arrange, c(plots, ncol = 4))
```

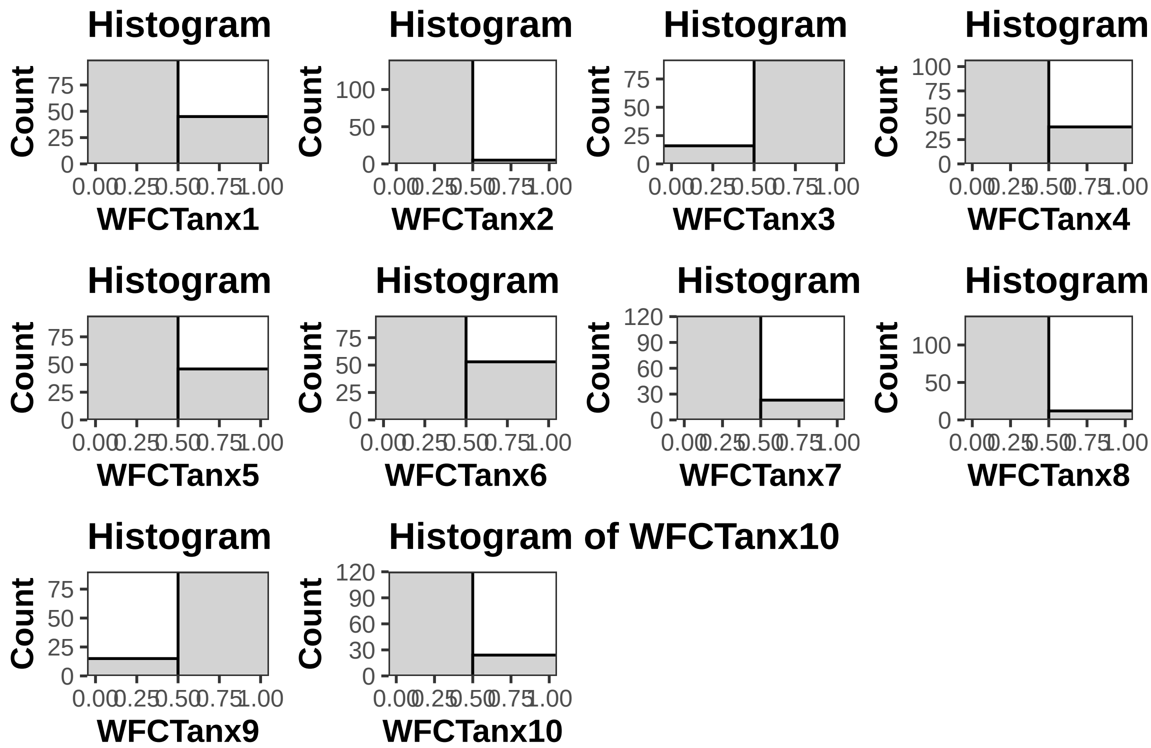



```
summary_mean_sd(df, WFCT_aggression_items)
```

```
## # A tibble: 10 x 5
##    variable  mean    sd     n missing
##    <chr>    <dbl> <dbl> <dbl>   <dbl>
##  1 WFCTagg1  0.05  0.21   146       8
##  2 WFCTagg2  0.13  0.34   150       4
##  3 WFCTagg3  0.07  0.26   149       5
##  4 WFCTagg4  0.08  0.27   148       6
##  5 WFCTagg5  0.03  0.18   151       3
##  6 WFCTagg6  0.51  0.5    152       2
##  7 WFCTagg7  0.1   0.3    143      11
##  8 WFCTagg8  0.41  0.49   140      14
##  9 WFCTagg9  0.23  0.42   142      12
## 10 WFCTagg10 0.64  0.48   141      13
```

3.7 Anxious Cognition

```
plots <- lapply(WFCT_anxiety_items, function(item) {
  plotHist(item, c(0, 1), condition = NULL, binwidth = 1)
})

do.call(grid.arrange, c(plots, ncol = 4))
```




```r
summary_mean_sd(df, WFCT_anxiety_items)
```

```
## # A tibble: 10 x 5
##    variable  mean    sd     n missing
##    <chr>    <dbl> <dbl> <dbl>   <dbl>
##  1 WFCTanx1  0.31  0.47   144      10
##  2 WFCTanx2  0.03  0.18   145       9
##  3 WFCTanx3  0.85  0.36   108      46
##  4 WFCTanx4  0.26  0.44   145       9
##  5 WFCTanx5  0.33  0.47   140      14
##  6 WFCTanx6  0.36  0.48   148       6
##  7 WFCTanx7  0.16  0.37   144      10
##  8 WFCTanx8  0.08  0.27   151       3
##  9 WFCTanx9  0.86  0.35   105      49
## 10 WFCTanx10 0.17  0.37   144      10
```

4. Test Quality Criteria: Motivation for Masculine Behaviour

Objectives:

- Assess the *validity* and *reliability* of the scale Motivation for Masculine Behaviour (MMB)
- Compute participant latent trait scores

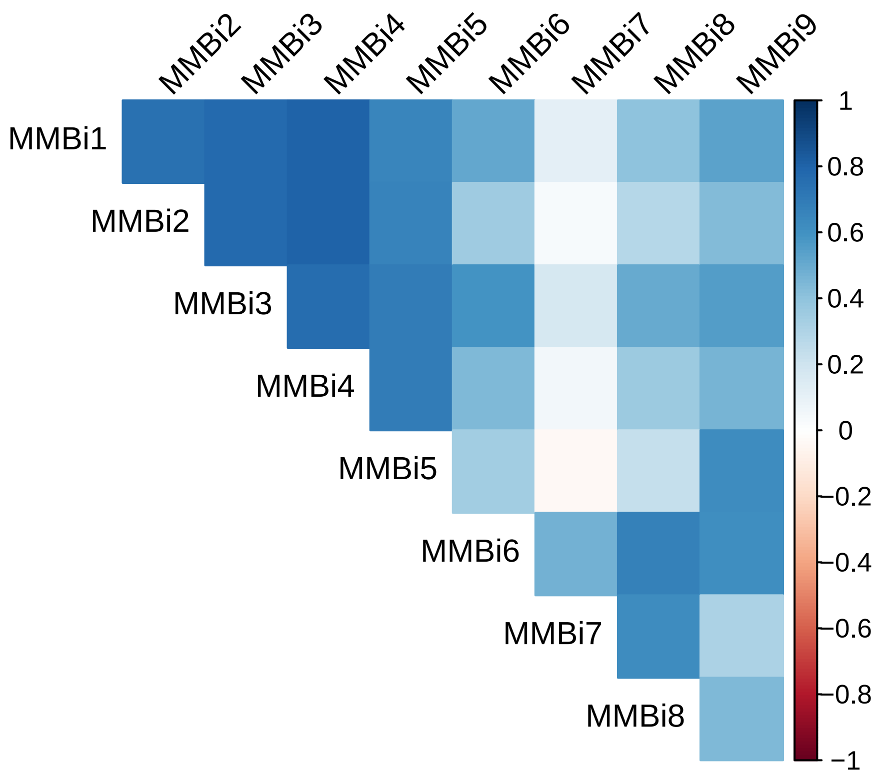
Methodology:

- Visualization & Descriptives
- CFA and EFA to => construct validity
- McDonald's Omega => Reliabiltiy
- Regression scoring => latent trait scores

4.1 Construct Validity

4.1.1 Visualization & Descriptives

```
cor_mmb <- psych::polychoric(df[MMB_item_codes])$rho

corrplot(cor_mmb,
  method = "color",
  type = "upper",
  order = "original",
  tl.col = "black",
  tl.srt = 45,
  diag = FALSE
)
```



Interpretation

- The correlation plit shows a very clear block for MMB 1-5 items (pressured scale)
- The block for for autonomous items (MMB 6 - 9) is way less clear

```
summary_mean_sd(df, MMB_item_codes)
```

```
## # A tibble: 9 x 5
##   variable  mean    sd     n missing
##   <chr>    <dbl> <dbl> <dbl>   <dbl>
## 1 MMBi1     3.55  1.78   154       0
## 2 MMBi2     3.75  1.74   154       0
## 3 MMBi3     3.35  1.68   154       0
## 4 MMBi4     3.51  1.74   154       0
## 5 MMBi5     2.97  1.68   154       0
## 6 MMBi6     4.06  1.79   154       0
```

```
## 7 MMBi7     5.29  1.49   154       0
## 8 MMBi8     4.32  1.68   154       0
## 9 MMBi9     3.26  1.77   154       0
```

Interpretation

- The item scores ranged between 1 and 7
- Most items show a mean in the center of the scale
- Item MMBi7 shows a very high mean (5.27). "Ich bin gerne männlich"

- Generally, the items do not show bottom or ceiling effects, and there is some variance to the responses

4.1.2 Check sampling adequacy for Factory Analysis

```r
KMO(cor_mmb)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = cor_mmb)
## Overall MSA =  0.85
## MSA for each item =
## MMBi1 MMBi2 MMBi3 MMBi4 MMBi5 MMBi6 MMBi7 MMBi8 MMBi9
##  0.92  0.88  0.89  0.88  0.83  0.83  0.69  0.81  0.80
```

```r
cortest.bartlett(cor_mmb, n = nrow(df))
```

```
## $chisq
## [1] 1034.431
##
## $p.value
## [1] 9.376131e-194
##
## $df
## [1] 36
```

KMO (Kaiser-Meyer-Olkin) should be > 0.6 (ideally > 0.8) Bartlett's test should be significant ($p < .05$)

Interpretation

- The overall KMO is 0.86 showing a good smapling adequacy for factor analysis
- Individual scores of items are good as well, except MMBi7, which has a sufficient, yet lower KMO value (0.72) due to it's ceiling effect
- The Bartlett test is significant as well, showing a sampling adequacy for factor analysis

4.1.3 Two-Factor Uncorrelated CFA

Testing the proposed 2 factor structure by Stanaland & Gaither (2021)

- 2 factors
- no covariance between factors

```r
model_twof <- "
  # latent variables
  Pressured =~ MMBi1 + MMBi2 + MMBi3 + MMBi4 + MMBi5
  Autonomous =~ MMBi6 + MMBi7 + MMBi8 + MMBi9

  # constrain covariance to 0
  Pressured ~~ 0*Autonomous
"
```

```r
results_twof <- run_cfa_model(model_twof, df, model_name = "Two-Factor Uncorrelated",
  show_mi = TRUE)
```

```
##
## -------------------------
## MODEL SUMMARY
## -------------------------
## lavaan 0.6-20 ended normally after 22 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        18
##
##   Number of observations                           154
##
## Model Test User Model:
##
##   Test statistic                               159.497
##   Degrees of freedom                                27
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                               923.528
##   Degrees of freedom                                36
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.851
##   Tucker-Lewis Index (TLI)                       0.801
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2317.331
##   Loglikelihood unrestricted model (H1)      -2237.582
##
##   Akaike (AIC)                                4670.661
##   Bayesian (BIC)                              4725.326
##   Sample-size adjusted Bayesian (SABIC)       4668.354
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.179
##   90 Percent confidence interval - lower         0.152
##   90 Percent confidence interval - upper         0.206
##   P-value H_0: RMSEA <= 0.050                    0.000
##   P-value H_0: RMSEA >= 0.080                    1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.248
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Pressured =~
##     MMBi1             1.529    0.116   13.203    0.000    1.529    0.862
##     MMBi2             1.491    0.114   13.101    0.000    1.491    0.858
##     MMBi3             1.439    0.110   13.104    0.000    1.439    0.858
##     MMBi4             1.553    0.111   14.012    0.000    1.553    0.894
##     MMBi5             1.246    0.117   10.617    0.000    1.246    0.746
##   Autonomous =~
##     MMBi6             1.458    0.134   10.848    0.000    1.458    0.818
##     MMBi7             0.893    0.119    7.523    0.000    0.893    0.603
##     MMBi8             1.304    0.127   10.254    0.000    1.304    0.781
##     MMBi9             1.064    0.142    7.520    0.000    1.064    0.603
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Pressured ~~
##     Autonomous        0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .MMBi1             0.807    0.118    6.857    0.000    0.807    0.257
##    .MMBi2             0.797    0.115    6.929    0.000    0.797    0.264
##    .MMBi3             0.741    0.107    6.927    0.000    0.741    0.264
##    .MMBi4             0.606    0.099    6.131    0.000    0.606    0.201
##    .MMBi5             1.239    0.156    7.961    0.000    1.239    0.444
##    .MMBi6             1.048    0.226    4.629    0.000    1.048    0.330
##    .MMBi7             1.394    0.182    7.670    0.000    1.394    0.636
##    .MMBi8             1.088    0.199    5.457    0.000    1.088    0.390
##    .MMBi9             1.982    0.258    7.671    0.000    1.982    0.636
##     Pressured         1.000                               1.000    1.000
##     Autonomous        1.000                               1.000    1.000
##
##
## -------------------------
## MODEL FIT INDICES
## -------------------------
##       Index Value Criteria
## cfi     cfi 0.851     0.95
## tli     tli 0.801     0.95
## rmsea rmsea 0.179     0.06
## srmr   srmr 0.248     0.08
##
## -------------------------
## STANDARDIZED FACTOR LOADINGS
## -------------------------
##       Factor  Item Std_Loading
```

```
## 1  Pressured MMBi1       0.862
## 2  Pressured MMBi2       0.858
## 3  Pressured MMBi3       0.858
## 4  Pressured MMBi4       0.894
## 5  Pressured MMBi5       0.746
## 6 Autonomous MMBi6       0.818
## 7 Autonomous MMBi7       0.603
## 8 Autonomous MMBi8       0.781
## 9 Autonomous MMBi9       0.603
##
## -------------------------
## TOP MODIFICATION INDICES
## -------------------------
##           lhs op        rhs     mi   epc sepc.lv sepc.all sepc.nox
## 10  Pressured ~~ Autonomous 37.076 0.559   0.559    0.559    0.559
## 60      MMBi5 ~~      MMBi9 30.371 0.767   0.767    0.489    0.489
## 64      MMBi7 ~~      MMBi8 19.500 0.745   0.745    0.605    0.605
## 63      MMBi6 ~~      MMBi9 19.500 0.992   0.992    0.689    0.689
## 28 Autonomous =~      MMBi3 17.903 0.359   0.359    0.214    0.214
```

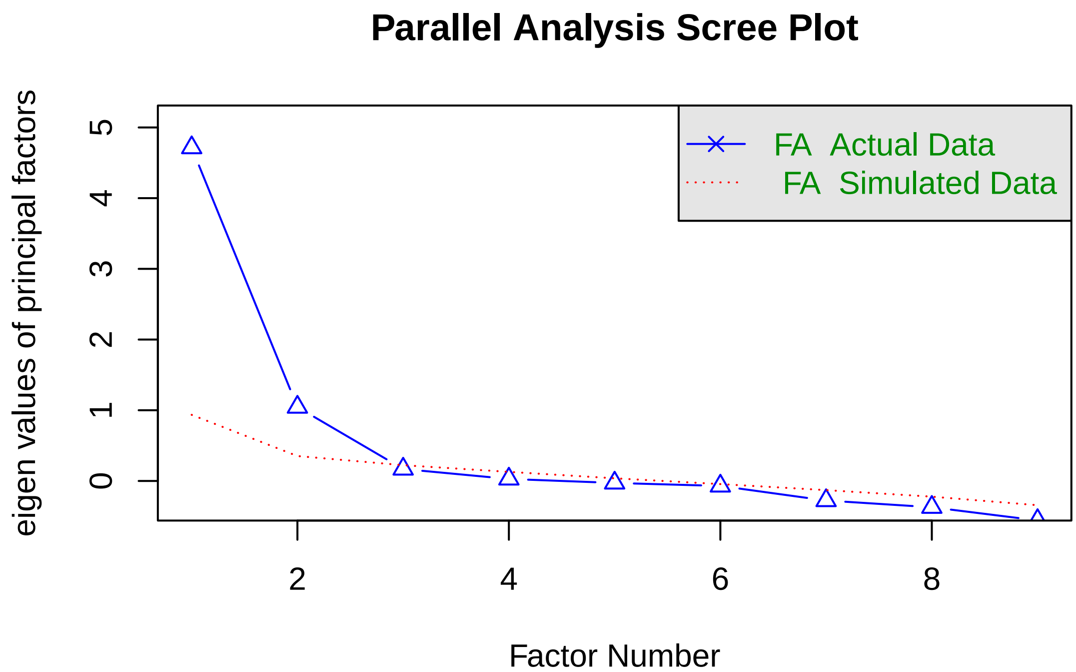
Interpretation

- None of the model fit indices is satisfying
- Mod Indices suggest a correlation between latent pressured and autonomous factors

4.1.4 Exploratory Factor Analysis

```r
fa.parallel(cor_mmb,
  fm = "ml",
  fa = "fa", n.iter = 100, show.legend = TRUE,
  main = "Parallel Analysis Scree Plot"
)
```

Determine the number of latent factors




```
## Parallel analysis suggests that the number of factors =  2  and the number of components =  NA
```

Interpretation:

- The "elbow" of the scree plot suggest a two factor model

```r
efa_result_twof <- fa(df[MMB_item_codes], nfactors = 2, rotate = "oblimin", fm = "ml")
print(efa_result_twof, cut = 0.3) # only loadings > .30
```

Compute EFA

```
## Factor Analysis using method =  ml
## Call: fa(r = df[MMB_item_codes], nfactors = 2, rotate = "oblimin",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##         ML1   ML2   h2   u2 com
## MMBi1  0.83       0.75 0.25 1.0
## MMBi2  0.90       0.75 0.25 1.0
## MMBi3  0.79       0.78 0.22 1.1
## MMBi4  0.89       0.79 0.21 1.0
## MMBi5  0.78       0.58 0.42 1.0
## MMBi6        0.69 0.64 0.36 1.2
## MMBi7        0.74 0.47 0.53 1.2
## MMBi8        0.80 0.67 0.33 1.0
## MMBi9  0.39  0.39 0.42 0.58 2.0
##
##                        ML1  ML2
## SS loadings           3.87 1.96
## Proportion Var        0.43 0.22
## Cumulative Var        0.43 0.65
```

```
## Proportion Explained  0.66 0.34
## Cumulative Proportion 0.66 1.00
##
##  With factor correlations of
##     ML1 ML2
## ML1 1.0 0.4
## ML2 0.4 1.0
##
## Mean item complexity =  1.2
## Test of the hypothesis that 2 factors are sufficient.
##
## df null model =  36  with the objective function =  6 with Chi Square =  894.54
## df of  the model are 19  and the objective function was  0.38
##
## The root mean square of the residuals (RMSR) is  0.04
## The df corrected root mean square of the residuals is  0.06
##
## The harmonic n.obs is  154 with the empirical chi square  20.92  with prob <  0.34
## The total n.obs was  154  with Likelihood Chi Square =  56.32  with prob <  1.5e-05
##
## Tucker Lewis Index of factoring reliability =  0.917
## RMSEA index =  0.113  and the 90 % confidence intervals are  0.08 0.148
## BIC =  -39.38
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy
##                                                    ML1  ML2
## Correlation of (regression) scores with factors   0.97 0.91
## Multiple R square of scores with factors          0.94 0.83
## Minimum correlation of possible factor scores     0.87 0.66
```

```r
# Create loadings matrix
loadings_matrix <- as.data.frame(efa_result_twof$loadings[1:length(mmb_labels), ])

# Format table
loading_table <- data.frame(
  Item = MMB_item_codes,
  Label = mmb_labels,
  Factor1 = round(loadings_matrix$ML1, 2),
  Factor2 = round(loadings_matrix$ML2, 2)
)

# Hide loadings < .30
loading_table <- loading_table %>%
  mutate(
    Factor1 = ifelse(abs(Factor1) < 0.30, "", as.character(Factor1)),
    Factor2 = ifelse(abs(Factor2) < 0.30, "", as.character(Factor2))
  )

# Render nice APA-style table
loading_table %>%
  kbl(
    caption = "EFA Loadings for MMB Items (2-Factor Solution)",
    align = "lccc",
    booktabs = TRUE
```

```r
  ) %>%
  kable_styling(
    full_width = FALSE,
    position = "center",
    bootstrap_options = c("striped", "hover", "condensed"),
    latex_options = "hold_position"
  ) %>%
  column_spec(2, width = "10cm") %>%
  row_spec(0, bold = TRUE, background = "#f2f2f2") %>%
  row_spec(1:nrow(loading_table), background = "white", color = "black")
```

Table 1: EFA Loadings for MMB Items (2-Factor Solution)

Item	Label	Factor1	Factor2
MMBi1	Allgemein verhalte ich mich männlich, weil ich die Akzeptanz und Anerkennung anderer möchte	0.83	
MMBi2	Allgemein bin ich männlich, weil das von mir erwartet wird	0.9	
MMBi3	Ich verhalte mich männlich, weil ich möchte, dass man mich mag	0.79	
MMBi4	Ich verhalte mich in Gegenwart anderer männlich, um ihre Erwartungen zu erfüllen	0.89	
MMBi5	Ich verhalte mich nicht weiblich, weil ich glaube, dass mich die Leute sonst nicht mögen würden	0.78	
MMBi6	Es ist mir wichtig, männlich zu sein		0.69
MMBi7	Ich bin gerne männlich		0.74
MMBi8	Es macht mich glücklich, mich männlich zu verhalten		0.8
MMBi9	Es ist mir wichtig, mich nicht weiblich zu verhalten	0.39	0.39

Interpretation

- EFA sows a very distinct factor for pressured motivation
- The factor for autonomous motivation is less clear and MMBi9 shows loadings on both factors

4.1.5 Two-Factor Correlated CFA

- 2 latent factors,
- correlation allowed between factors

```r
model_twof_cor <- "
  # latent variables
  Pressured =~ MMBi1 + MMBi2 + MMBi3 + MMBi4 + MMBi5
  Autonomous =~ MMBi6 + MMBi7 + MMBi8 + MMBi9

  # allow correlation between factors
  Pressured ~~ Autonomous
"

results_twof_cor <- run_cfa_model(model_twof_cor, df, model_name = "Two-Factor
 ↪  Correlated", show_mi = TRUE)
```

```
##
## -------------------------
## MODEL SUMMARY
## -------------------------
```

```
## lavaan 0.6-20 ended normally after 21 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        19
##
##   Number of observations                           154
##
## Model Test User Model:
##
##   Test statistic                               113.810
##   Degrees of freedom                                26
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                               923.528
##   Degrees of freedom                                36
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.901
##   Tucker-Lewis Index (TLI)                       0.863
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2294.487
##   Loglikelihood unrestricted model (H1)      -2237.582
##
##   Akaike (AIC)                                4626.974
##   Bayesian (BIC)                              4684.676
##   Sample-size adjusted Bayesian (SABIC)       4624.539
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.148
##   90 Percent confidence interval - lower         0.121
##   90 Percent confidence interval - upper         0.176
##   P-value H_0: RMSEA <= 0.050                    0.000
##   P-value H_0: RMSEA >= 0.080                    1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.099
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
```

```
##   Pressured =~
##     MMBi1             1.537    0.115   13.329    0.000    1.537    0.867
##     MMBi2             1.474    0.115   12.868    0.000    1.474    0.848
##     MMBi3             1.460    0.109   13.421    0.000    1.460    0.870
##     MMBi4             1.541    0.111   13.857    0.000    1.541    0.887
##     MMBi5             1.244    0.117   10.599    0.000    1.244    0.745
##   Autonomous =~
##     MMBi6             1.524    0.128   11.898    0.000    1.524    0.856
##     MMBi7             0.790    0.120    6.593    0.000    0.790    0.533
##     MMBi8             1.229    0.126    9.787    0.000    1.229    0.736
##     MMBi9             1.161    0.137    8.494    0.000    1.161    0.658
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Pressured ~~
##     Autonomous        0.587    0.064    9.126    0.000    0.587    0.587
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .MMBi1             0.782    0.114    6.830    0.000    0.782    0.249
##    .MMBi2             0.849    0.119    7.134    0.000    0.849    0.281
##    .MMBi3             0.682    0.101    6.762    0.000    0.682    0.242
##    .MMBi4             0.641    0.100    6.382    0.000    0.641    0.212
##    .MMBi5             1.244    0.156    7.989    0.000    1.244    0.446
##    .MMBi6             0.849    0.201    4.222    0.000    0.849    0.268
##    .MMBi7             1.568    0.193    8.123    0.000    1.568    0.715
##    .MMBi8             1.277    0.192    6.662    0.000    1.277    0.458
##    .MMBi9             1.766    0.236    7.477    0.000    1.766    0.567
##     Pressured         1.000                               1.000    1.000
##     Autonomous        1.000                               1.000    1.000
##
##
## -------------------------
## MODEL FIT INDICES
## -------------------------
##       Index Value Criteria
## cfi     cfi 0.901     0.95
## tli     tli 0.863     0.95
## rmsea rmsea 0.148     0.06
## srmr   srmr 0.099     0.08
##
## -------------------------
## STANDARDIZED FACTOR LOADINGS
## -------------------------
##       Factor  Item Std_Loading
## 1  Pressured MMBi1       0.867
## 2  Pressured MMBi2       0.848
## 3  Pressured MMBi3       0.870
## 4  Pressured MMBi4       0.887
## 5  Pressured MMBi5       0.745
## 6 Autonomous MMBi6       0.856
## 7 Autonomous MMBi7       0.533
## 8 Autonomous MMBi8       0.736
## 9 Autonomous MMBi9       0.658
```

```
##
## -------------------------
## TOP MODIFICATION INDICES
## -------------------------
##           lhs op   rhs     mi    epc sepc.lv sepc.all sepc.nox
## 60      MMBi5 ~~ MMBi9 31.894  0.750   0.750    0.506    0.506
## 64      MMBi7 ~~ MMBi8 24.886  0.717   0.717    0.506    0.506
## 28 Autonomous =~ MMBi3 15.682  0.434   0.434    0.259    0.259
## 23  Pressured =~ MMBi7 14.810 -0.560  -0.560   -0.378   -0.378
## 25  Pressured =~ MMBi9 13.735  0.610   0.610    0.345    0.345
```

Interpretation

- Allowing for a correlation between latent factors creates a better fit but still not sufficient
- Mod indices suggest A correlation between Item MMBi5 and MMBi9, This makes conceptually sense, because they both deal with anti femininity.

4.1.6 Two-Factor Correlated + remove MMBi9 + Residuals CFA

- 2 latent factors
- correlation allowed between factors
- remove item MMBi9, because it poorly fits into the concept.
- allow correlated residuals item 7 and 8

```r
model_twof_cor_mod <- "
  # latent variables
  Pressured =~ MMBi1 + MMBi2 + MMBi3 + MMBi4 + MMBi5
  Autonomous =~ MMBi6 + MMBi7 + MMBi8

  Pressured ~~ Autonomous

  MMBi7 ~~ MMBi8
"

results_twof_cor_mod <- run_cfa_model(model_twof_cor_mod, df, model_name = "Two-Factor
 ↪  Correlated Modified", show_mi = TRUE)
```

```
##
## -------------------------
## MODEL SUMMARY
## -------------------------
## lavaan 0.6-20 ended normally after 26 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        18
##
##   Number of observations                           154
##
## Model Test User Model:
##
##   Test statistic                                44.676
##   Degrees of freedom                                18
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
```

```
##
##   Test statistic                               805.861
##   Degrees of freedom                                28
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.966
##   Tucker-Lewis Index (TLI)                       0.947
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2012.763
##   Loglikelihood unrestricted model (H1)      -1990.425
##
##   Akaike (AIC)                                4061.527
##   Bayesian (BIC)                              4116.192
##   Sample-size adjusted Bayesian (SABIC)       4059.219
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.098
##   90 Percent confidence interval - lower         0.062
##   90 Percent confidence interval - upper         0.135
##   P-value H_0: RMSEA <= 0.050                    0.017
##   P-value H_0: RMSEA >= 0.080                    0.813
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.066
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Pressured =~
##     MMBi1             1.537    0.115   13.329    0.000    1.537    0.867
##     MMBi2             1.475    0.114   12.885    0.000    1.475    0.849
##     MMBi3             1.458    0.109   13.400    0.000    1.458    0.870
##     MMBi4             1.543    0.111   13.883    0.000    1.543    0.889
##     MMBi5             1.241    0.117   10.564    0.000    1.241    0.743
##   Autonomous =~
##     MMBi6             1.637    0.165    9.940    0.000    1.637    0.919
##     MMBi7             0.680    0.129    5.262    0.000    0.680    0.459
##     MMBi8             1.140    0.146    7.811    0.000    1.140    0.683
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Pressured ~~
##     Autonomous        0.551    0.072    7.633    0.000    0.551    0.551
```

```
##   .MMBi7 ~~
##     .MMBi8            0.647    0.186    3.482    0.000    0.647    0.403
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     .MMBi1            0.782    0.115    6.825    0.000    0.782    0.249
##     .MMBi2            0.845    0.119    7.120    0.000    0.845    0.280
##     .MMBi3            0.686    0.101    6.772    0.000    0.686    0.244
##     .MMBi4            0.635    0.100    6.349    0.000    0.635    0.210
##     .MMBi5            1.251    0.157    7.995    0.000    1.251    0.448
##     .MMBi6            0.493    0.408    1.211    0.226    0.493    0.156
##     .MMBi7            1.729    0.215    8.045    0.000    1.729    0.789
##     .MMBi8            1.488    0.260    5.733    0.000    1.488    0.534
##      Pressured        1.000                               1.000    1.000
##      Autonomous       1.000                               1.000    1.000
##
##
## -------------------------
## MODEL FIT INDICES
## -------------------------
##       Index Value Criteria
## cfi     cfi 0.966     0.95
## tli     tli 0.947     0.95
## rmsea rmsea 0.098     0.06
## srmr   srmr 0.066     0.08
##
## -------------------------
## STANDARDIZED FACTOR LOADINGS
## -------------------------
##       Factor  Item Std_Loading
## 1  Pressured MMBi1       0.867
## 2  Pressured MMBi2       0.849
## 3  Pressured MMBi3       0.870
## 4  Pressured MMBi4       0.889
## 5  Pressured MMBi5       0.743
## 6 Autonomous MMBi6       0.919
## 7 Autonomous MMBi7       0.459
## 8 Autonomous MMBi8       0.683
##
## -------------------------
## TOP MODIFICATION INDICES
## -------------------------
##           lhs op   rhs     mi    epc sepc.lv sepc.all sepc.nox
## 26 Autonomous =~ MMBi3 16.068  0.413   0.413    0.246    0.246
## 25 Autonomous =~ MMBi2 11.319 -0.375  -0.375   -0.216   -0.216
## 23  Pressured =~ MMBi8  7.872  0.603   0.603    0.361    0.361
## 22  Pressured =~ MMBi7  7.872 -0.360  -0.360   -0.243   -0.243
## 55      MMBi6 ~~ MMBi8  7.872 -1.247  -1.247   -1.455   -1.455
```

Interpretation

- …

4.2 Reliability

```
fit_twof_cor_mod <- cfa(model_twof_cor_mod, data = df, std.lv = TRUE)


# Compute McDonald's Omega for each factor using compRelSEM
omega <- compRelSEM(fit_twof_cor_mod, tau.eq = FALSE, ord.scale = FALSE, return.df =
 ↪ TRUE)

cat("McDonald's Omega Reliability Coefficients:\n")
```

```
## McDonald's Omega Reliability Coefficients:
```

```
print(omega)
```

```
##  Pressured Autonomous
##      0.926      0.702
```

Interpretation

- Pressured scale shows an excellent reliability (> .9)
- Autonomous scale shows an acceptable reliability (> .7) => reliability is sufficient, so we use the proposed factor structure to compute factor scored for autonomous and pressured motivation

Omega values .70 (acceptable), .80 (good). => Alpha assumes tau-equivalence (all items same loading), which isn't true here, so we trust omega more

4.3 Latent Trait Scores

4.3.1 Compute Scores

Use regression scoring from the chosen model, to compute latent trait scores

```
fs_reg <- lavPredict(fit_twof_cor_mod, method = "regression")

df$regPressured <- scale(fs_reg[, "Pressured"])
df$regAutonomous <- scale(fs_reg[, "Autonomous"])
```

Also compute Row means for visualizations

```
df$rowmeansPressured <- rowMeans(
  df[mmb_pressured_items],
  na.rm = TRUE
)

df$rowmeansAutonomous <- rowMeans(
  df[setdiff(mmb_autonomous_items, "MMBi9")], # exclude item 9
  na.rm = TRUE
)
```

Decide which computation to use for later analysis

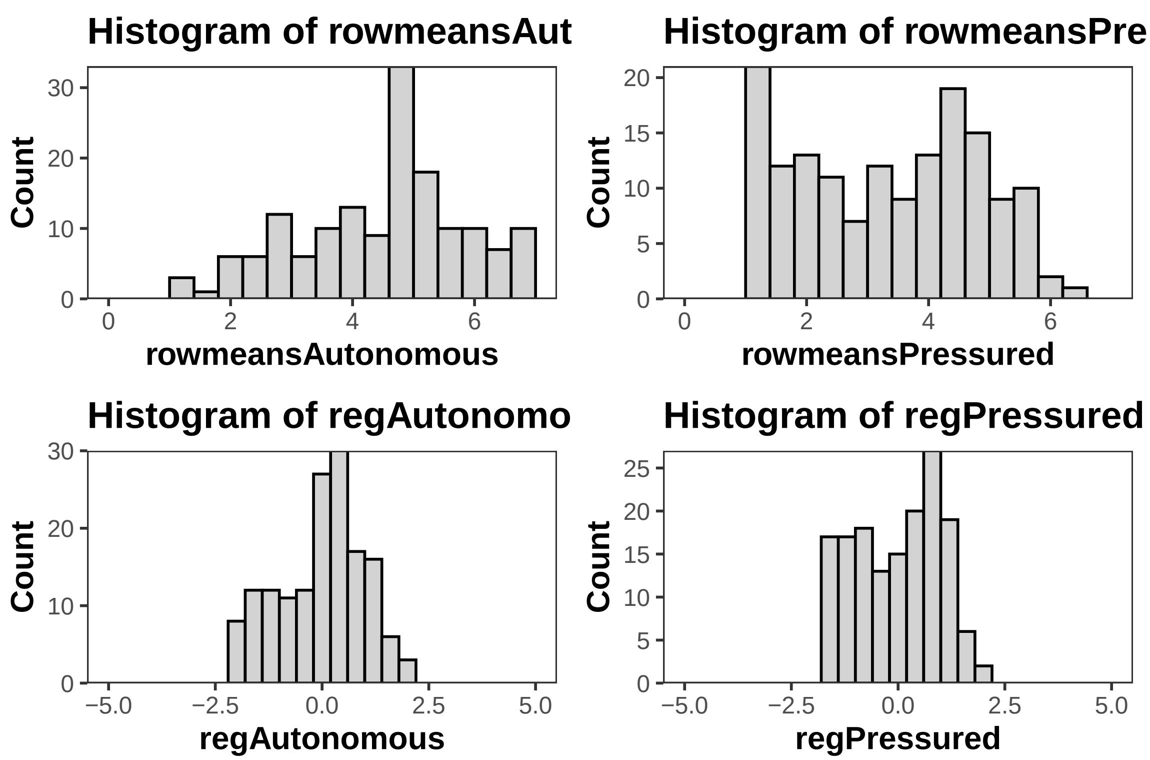
```
df$pressuredMotivation = df$rowmeansPressured
df$autonomousMotivation = df$rowmeansAutonomous
```

4.3.2 Visualization & Descriptives

```
summary_mean_sd(df, c("rowmeansAutonomous", "rowmeansPressured", "regAutonomous",
↪  "regPressured"))
```

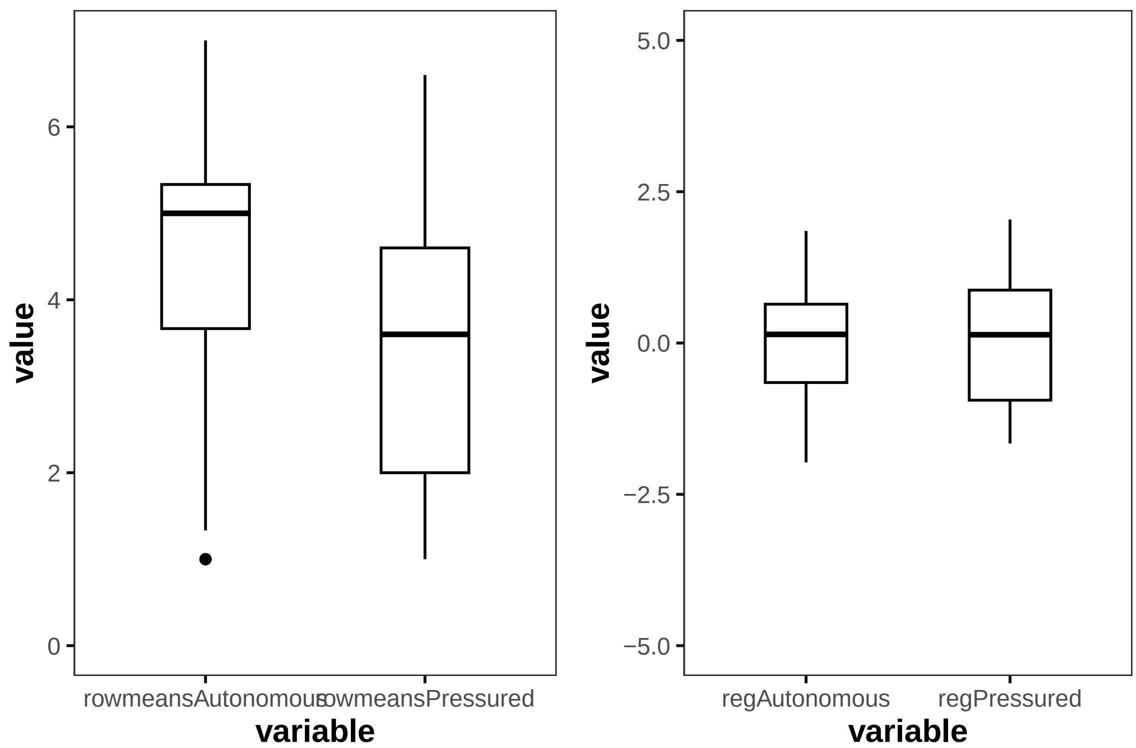
```
## # A tibble: 4 x 5
##   variable            mean    sd     n missing
##   <chr>              <dbl> <dbl> <dbl>   <dbl>
## 1 rowmeansAutonomous  4.55  1.38   154       0
## 2 rowmeansPressured   3.43  1.51   154       0
## 3 regAutonomous       0     1      154       0
## 4 regPressured        0     1      154       0
```

```
rm_aut_hist <- plotHist("rowmeansAutonomous", c(0, 7))
rm_pres_hist <- plotHist("rowmeansPressured", c(0, 7))
reg_aut_hist <- plotHist("regAutonomous", c(-5, 5))
reg_pres_hist <- plotHist("regPressured", c(-5, 5))

grid.arrange(rm_aut_hist, rm_pres_hist, reg_aut_hist, reg_pres_hist, ncol = 2)
```



-> regression scoring reduces measurement error at the price of losing variance

```
rm_box <- plotBox(c("rowmeansAutonomous", "rowmeansPressured"))
reg_box <- plotBox(c("regAutonomous", "regPressured"))

grid.arrange(ggpar(rm_box, ylim = c(0, 7)), ggpar(reg_box, ylim = c(-5, 5)), ncol = 2)
```




```
cor(df$regAutonomous, df$regPressured)
```

```
##            [,1]
## [1,] 0.6037161
```

```
cor(df$rowmeansAutonomous, df$rowmeansPressured)
```

```
## [1] 0.3916663
```

Interpretation

- moderate to high correlation between pressured and autonomous factor
- the correlation increases with the regression scores (because factors are modeled as correlative)

5. Test Quality Criteria: Aggressive Cognition

Objectives:

- Assess the *validity* and *reliability* of the scale for aggressive cognition
- Compute participant latent trait scores

Methodology:

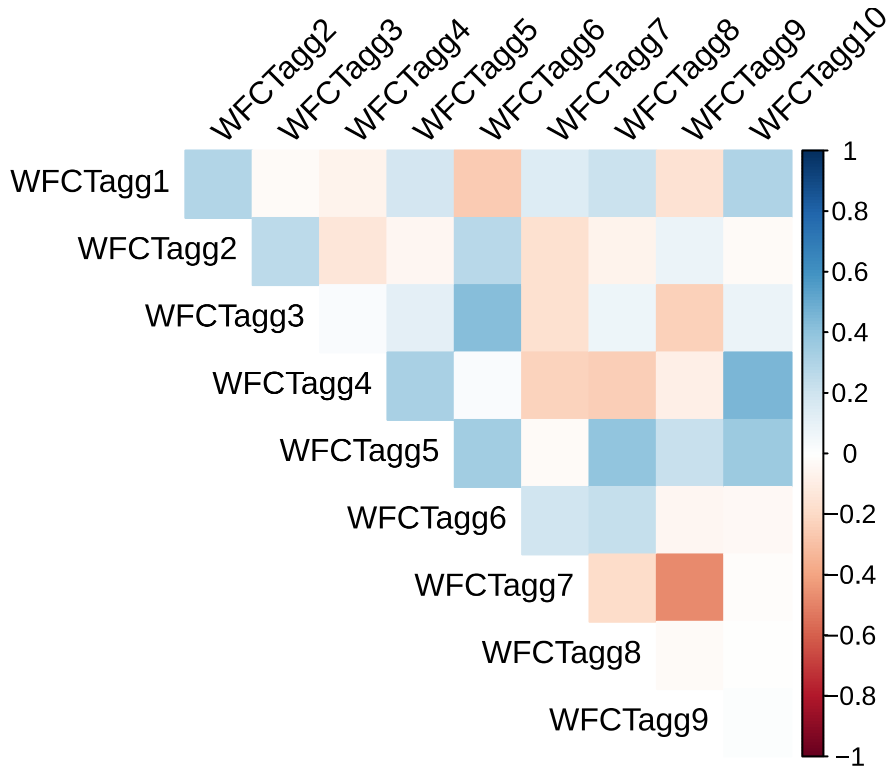
- Visualization & Descriptives

5.1 Construct Validity

5.1.1 Visualization & Descriptives

```
cor_aggr <- psych::tetrachoric(df[, WFCT_aggression_items])$rho
```

```r
corrplot::corrplot(cor_aggr,
  method = "color",
  type = "upper",
  order = "original",
  tl.col = "black",
  tl.srt = 45,
  diag = FALSE
)
```



-> a tetrachoric correlation is used, because responses are always binary (aggressive, vs. non-aggressive response)

Interpretation

- The correlation plot shows no clear pattern between the items
- If all items were to measure the same latent construct (aggressive cognition) we would expect (somewhat) positive correlations between all the items

```r
summary_binary_counts(df, WFCT_aggression_items)
```

```
##     variable negative positive pct_positive missing
## 1   WFCTagg1      139        7          4.8       8
## 2   WFCTagg2      130       20         13.3       4
## 3   WFCTagg3      138       11          7.4       5
## 4   WFCTagg4      136       12          8.1       6
## 5   WFCTagg5      146        5          3.3       3
## 6   WFCTagg6       74       78         51.3       2
## 7   WFCTagg7      129       14          9.8      11
## 8   WFCTagg8       83       57         40.7      14
## 9   WFCTagg9      110       32         22.5      12
```

```
## 10 WFCTagg10       51       90          63.8       13
```

Interpretation

- *Low Positive Response Rates:* Several items show low positive (aggressive) completion rates
- *Missing Values:* Some missings on items 7, 8, 9, 10

5.1.2 Check sampling adequacy for Factor Analysis

```r
KMO(cor_aggr)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = cor_aggr)
## Overall MSA =  0.16
## MSA for each item =
##  WFCTagg1  WFCTagg2  WFCTagg3  WFCTagg4  WFCTagg5  WFCTagg6  WFCTagg7  WFCTagg8
##      0.20      0.14      0.14      0.13      0.32      0.17      0.11      0.10
##  WFCTagg9 WFCTagg10
##      0.12      0.50
```

```r
cortest.bartlett(cor_aggr, n = nrow(df))
```

```
## $chisq
## [1] 623.5155
##
## $p.value
## [1] 7.520443e-103
##
## $df
## [1] 45
```

KMO (Kaiser-Meyer-Olkin) should be > 0.6 (ideally > 0.8) Bartlett's test should be significant (p < .05)

Interpretation

- The KMO is extremely low, indicating that the items are not suited for a factor analysis
- This is because the items share very little variance (are not correlating)

5.2 Reliability

```r
psych::alpha(cor_aggr, check.keys = FALSE)
```

```
## Some items ( WFCTagg7 ) were negatively correlated with the first principal component and
## probably should be reversed.
## To do this, run the function again with the 'check.keys=TRUE' option

##
## Reliability analysis
## Call: psych::alpha(x = cor_aggr, check.keys = FALSE)
##
##   raw_alpha std.alpha G6(smc) average_r  S/N median_r
##       0.36      0.36    0.78     0.053 0.55     0.01
##
##     95% confidence boundaries
##       lower alpha upper
## Feldt -0.46  0.36  0.81
##
```

```
##  Reliability if an item is dropped:
##           raw_alpha std.alpha G6(smc) average_r  S/N var.r   med.r
## WFCTagg1       0.31      0.31    0.72     0.048 0.45 0.047  0.0025
## WFCTagg2       0.33      0.33    0.72     0.052 0.50 0.049  0.0180
## WFCTagg3       0.32      0.32    0.71     0.050 0.47 0.047 -0.0106
## WFCTagg4       0.38      0.38    0.68     0.063 0.61 0.043  0.0409
## WFCTagg5       0.11      0.11    0.67     0.014 0.12 0.043 -0.0197
## WFCTagg6       0.24      0.24    0.62     0.033 0.31 0.044 -0.0106
## WFCTagg7       0.47      0.47    0.67     0.091 0.90 0.040  0.0497
## WFCTagg8       0.34      0.34    0.65     0.055 0.52 0.047  0.0180
## WFCTagg9       0.46      0.46    0.72     0.086 0.84 0.042  0.0497
## WFCTagg10      0.24      0.24    0.76     0.034 0.32 0.047  0.0011
##
##  Item statistics
##               r   r.cor r.drop
## WFCTagg1  0.426  0.3756  0.180
## WFCTagg2  0.385  0.3209  0.134
## WFCTagg3  0.410  0.3519  0.162
## WFCTagg4  0.281  0.2471  0.022
## WFCTagg5  0.750  0.7454  0.595
## WFCTagg6  0.566  0.5742  0.347
## WFCTagg7  0.024 -0.0200 -0.230
## WFCTagg8  0.365  0.3440  0.111
## WFCTagg9  0.075  0.0081 -0.183
## WFCTagg10 0.557  0.4518  0.336
```

Interpretation

5.3 Latent Trait Scores

5.3.1 Compute Scores

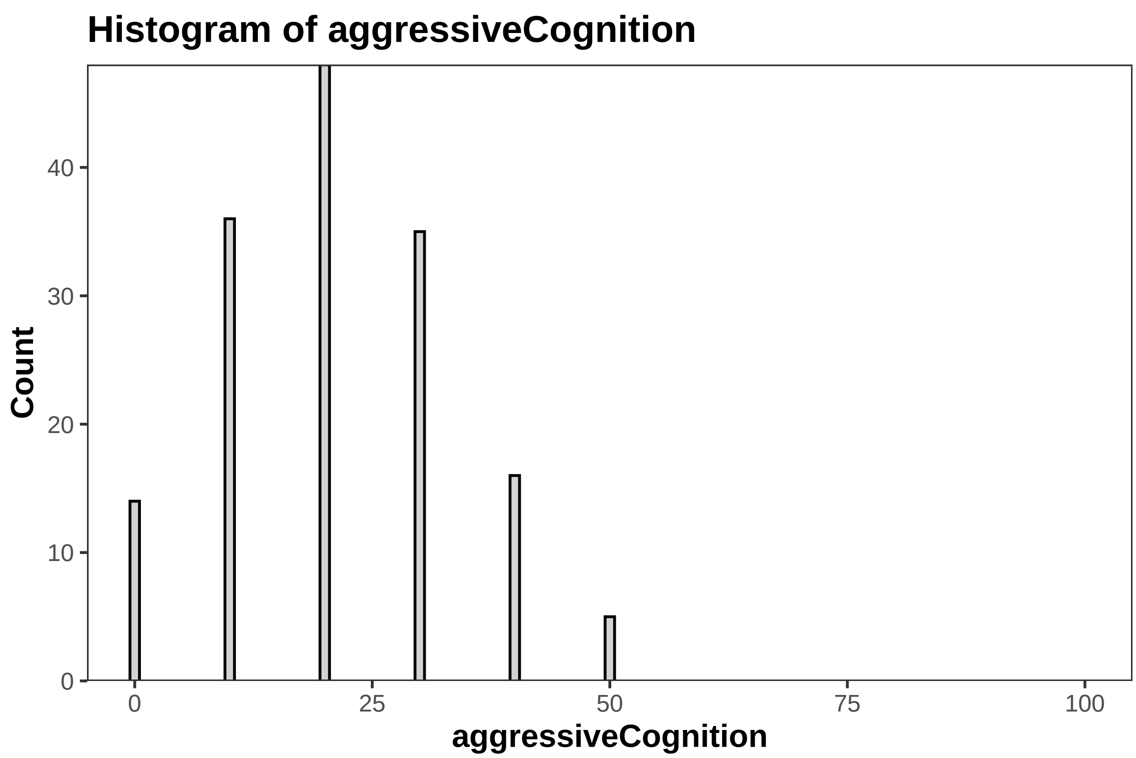
```
df$aggressiveWordCompletionScore <- rowMeans(
  replace(df[WFCT_aggression_items], is.na(df[WFCT_aggression_items]), 0)
)
df$aggressiveCognition <- df$aggressiveWordCompletionScore * 100
```

5.3.2 Visualization & Descriptives

```
summary_mean_sd(df, "aggressiveCognition")
```

```
## # A tibble: 1 x 5
##   variable             mean    sd     n missing
##   <chr>               <dbl> <dbl> <dbl>   <dbl>
## 1 aggressiveCognition  21.2  12.4   154       0
```

```
plotHist("aggressiveCognition", c(0, 100), condition = NULL, binwidth = 1)
```



Summary

…

6. Test Quality Criteria: Anxious Cognition

Objectives:

- Assess the *validity* and *reliability* of the scale for anxious cognition
- Compute participant latent trait scores

Methodology:

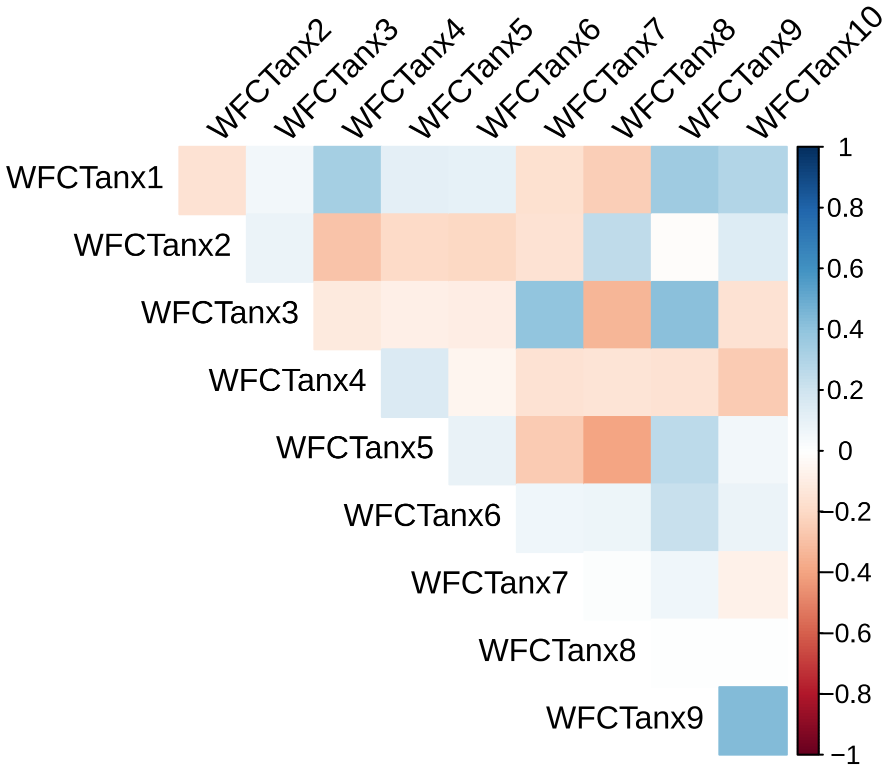
- Descriptives & Visualization

6.1 Construct Validity

6.1.1 Visualization & Descriptives

```
cor_anx <- psych::tetrachoric(df[, WFCT_anxiety_items])$rho
```

```
corrplot::corrplot(cor_anx,
  method = "color",
  type = "upper",
  order = "original",
  tl.col = "black",
  tl.srt = 45,
  diag = FALSE
)
```



-> a tetrachoric correlation is used, because responses are always binary (aggressive, vs. non-aggressive)

```
summary_binary_counts(df, WFCT_anxiety_items)
```

```
##     variable negative positive pct_positive missing
## 1   WFCTanx1       99       45         31.2      10
## 2   WFCTanx2      140        5          3.4       9
## 3   WFCTanx3       16       92         85.2      46
## 4   WFCTanx4      107       38         26.2       9
## 5   WFCTanx5       94       46         32.9      14
## 6   WFCTanx6       95       53         35.8       6
## 7   WFCTanx7      121       23         16.0      10
## 8   WFCTanx8      139       12          7.9       3
## 9   WFCTanx9       15       90         85.7      49
## 10 WFCTanx10      120       24         16.7      10
```

Interpretation

- *Low Positive Response Rates:* Several items show low positive (anxious) completion rates
- *Missing Values:* …

Interpretation

- …

6.1.2 Check sampling adequacy for Factor Analysis

```
KMO(cor_anx)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = cor_anx)
```

```
## Overall MSA =  0.3
## MSA for each item =
##  WFCTanx1  WFCTanx2  WFCTanx3  WFCTanx4  WFCTanx5  WFCTanx6  WFCTanx7  WFCTanx8
##      0.48      0.35      0.22      0.53      0.27      0.47      0.43      0.22
##  WFCTanx9 WFCTanx10
##      0.26      0.27
```

```r
cortest.bartlett(cor_anx, n = nrow(df))
```

```
## $chisq
## [1] 432.2395
##
## $p.value
## [1] 1.010417e-64
##
## $df
## [1] 45
```

KMO (Kaiser-Meyer-Olkin) should be > 0.6 (ideally > 0.8) Bartlett's test should be significant (p < .05)

Interpretation

- The KMO is extremely low, indicating that the items are not suited for a factor analysis
- This is because the items share very little variance …

6.2 Reliability

```r
psych::alpha(cor_anx, check.keys = FALSE)
```

```
## Some items ( WFCTanx2 WFCTanx7 WFCTanx8 ) were negatively correlated with the first principal compone
## probably should be reversed.
## To do this, run the function again with the 'check.keys=TRUE' option
```

```
##
## Reliability analysis
## Call: psych::alpha(x = cor_anx, check.keys = FALSE)
##
##   raw_alpha std.alpha G6(smc) average_r   S/N median_r
##      0.086     0.086    0.51    0.0093 0.094   0.0075
##
##     95% confidence boundaries
##       lower alpha upper
## Feldt -1.08  0.09  0.73
##
##  Reliability if an item is dropped:
##           raw_alpha std.alpha G6(smc) average_r    S/N var.r   med.r
## WFCTanx1     -0.075    -0.075   0.405   -0.0078 -0.069 0.043 -0.0051
## WFCTanx2      0.195     0.195   0.548    0.0263  0.243 0.047  0.0345
## WFCTanx3      0.058     0.058   0.338    0.0068  0.062 0.042  0.0090
## WFCTanx4      0.222     0.222   0.566    0.0307  0.285 0.045  0.0589
## WFCTanx5      0.145     0.145   0.487    0.0185  0.170 0.045  0.0074
## WFCTanx6      0.029     0.029   0.509    0.0033  0.030 0.052 -0.0051
## WFCTanx7      0.146     0.146   0.531    0.0186  0.171 0.047  0.0329
## WFCTanx8      0.235     0.235   0.494    0.0330  0.307 0.043  0.0589
## WFCTanx9     -0.403    -0.403   0.063   -0.0329 -0.287 0.037 -0.0644
## WFCTanx10    -0.034    -0.034   0.386   -0.0036 -0.032 0.045 -0.0050
```

```
##
##  Item statistics
##                 r  r.cor r.drop
## WFCTanx1  0.516  0.480  0.240
## WFCTanx2  0.144 -0.081 -0.160
## WFCTanx3  0.356  0.381  0.056
## WFCTanx4  0.095 -0.145 -0.206
## WFCTanx5  0.228  0.105 -0.077
## WFCTanx6  0.394  0.195  0.098
## WFCTanx7  0.227  0.039 -0.079
## WFCTanx8  0.070 -0.073 -0.228
## WFCTanx9  0.791  0.991  0.623
## WFCTanx10 0.470  0.461  0.185
```

6.3 Latent Trait Scores

6.3.1 Compute Scores

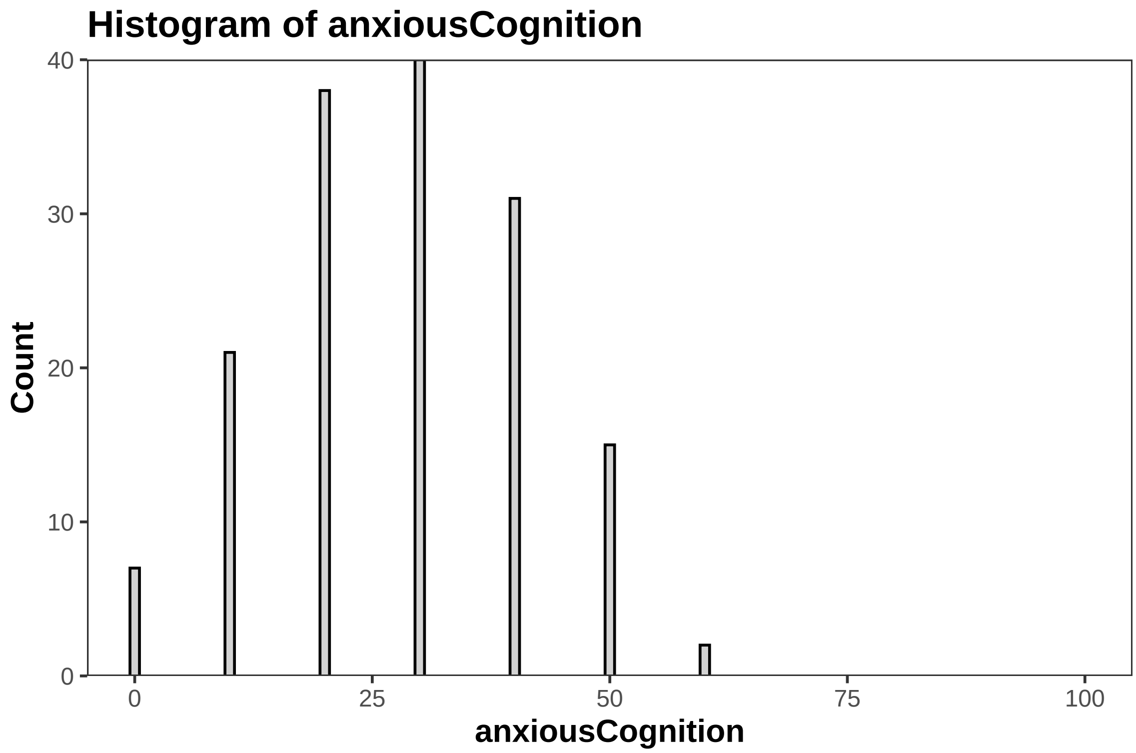
```r
df$anxiousWordCompletionScore <- rowMeans(
  replace(df[WFCT_anxiety_items], is.na(df[WFCT_anxiety_items]), 0)
)
df$anxiousCognition <-df$anxiousWordCompletionScore * 100
```

6.3.2 Visualization & Descriptives

```r
summary_mean_sd(df, c("anxiousCognition"))
```

```
## # A tibble: 1 x 5
##   variable          mean    sd     n missing
##   <chr>            <dbl> <dbl> <dbl>   <dbl>
## 1 anxiousCognition  27.8  13.7   154       0
```

```r
plotHist("anxiousCognition", c(0, 100), condition = NULL, binwidth = 1)
```



6.4 Summary

- lorem ipsum
- dolor sit amet

7. Effect of Threat Condition on Aggressive Cognition (Hypothesis 1a)

Objectives:

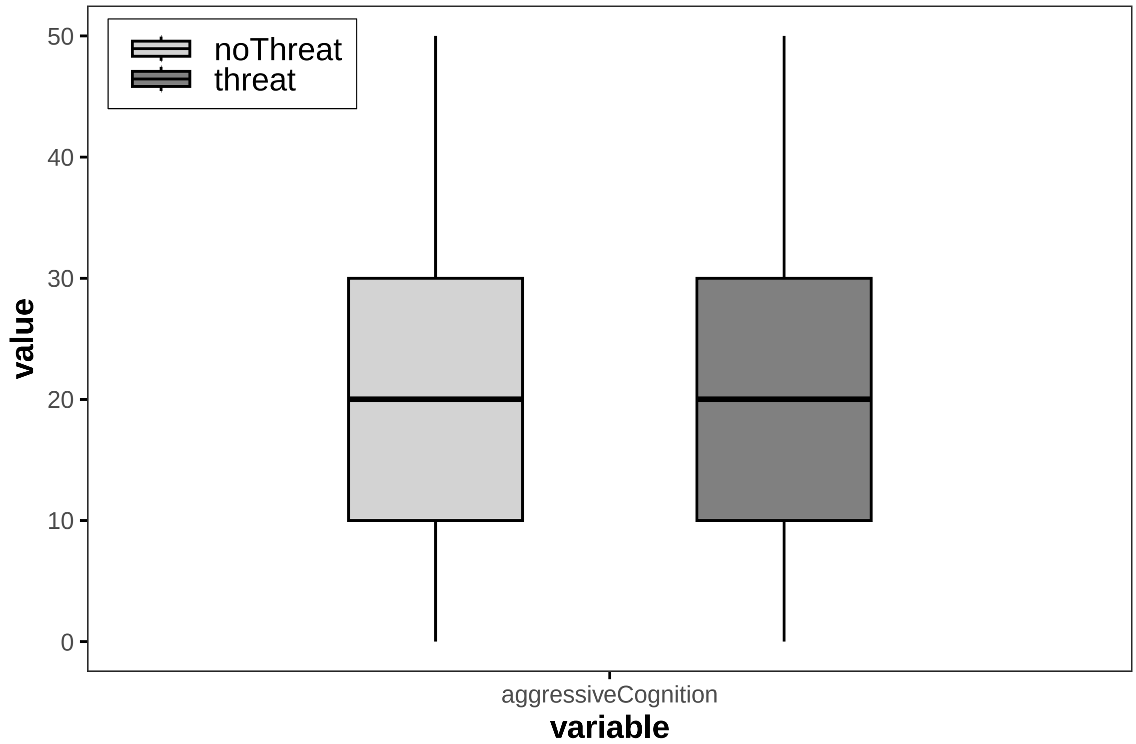
- Assess if there is a statistically significant effect of threat condition on aggressive cognition

Methodology:

- Visualization & Descriptives
- Check assumptions for t-test (normality, homogeneity)
- Since assumptions are violated: Binomial GLM on proportion data

7.1 Visualization & Descriptives

```r
plotBox("aggressiveCognition", "threatCondition")
```




```
summary_mean_sd(df, "aggressiveCognition", "threatCondition")
```

```
## # A tibble: 2 x 6
##   threatCondition variable             mean    sd     n missing
##   <fct>           <chr>               <dbl> <dbl> <dbl>   <dbl>
## 1 noThreat        aggressiveCognition  22.8  12.3    79       0
## 2 threat          aggressiveCognition  19.5  12.3    75       0
```

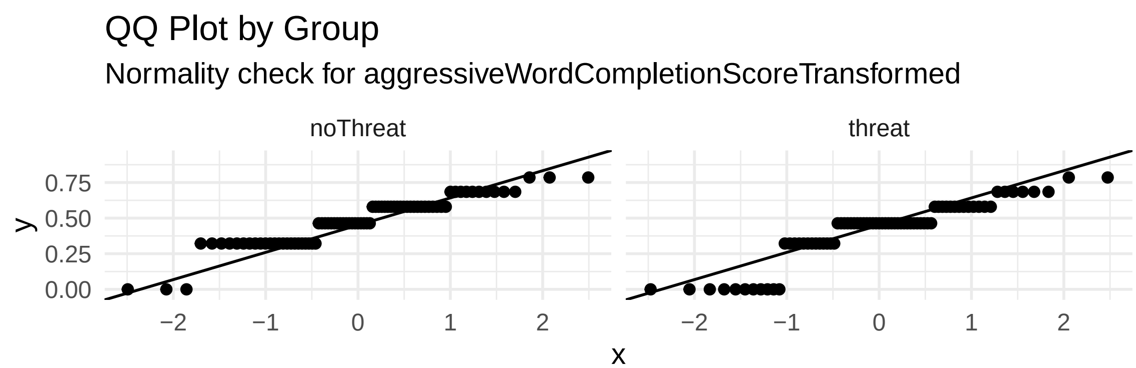
7.2 Check T-Test Assumptions

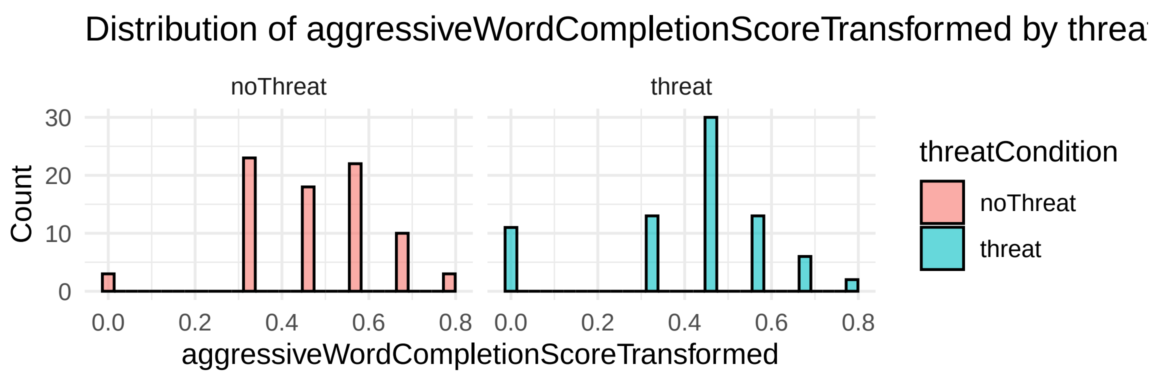
To ensure independence between value and variance (as pre-registered), we apply an arcsine square root transformation to the proportion scores before checking assumptions.

```
df$aggressiveWordCompletionScoreTransformed <-
↪  asin(sqrt(df$aggressiveWordCompletionScore))
```

7.2.1 Normality within each group

```
agg_qq <- check_normality_qq(df, aggressiveWordCompletionScoreTransformed,
↪  threatCondition)
agg_hist <- check_normality_hist(df, aggressiveWordCompletionScoreTransformed,
↪  threatCondition)
grid.arrange(agg_qq, agg_hist, ncol = 1)
```





Interpretation:

- Even after arcsine-sqrt transformation, the distribution shows zero-inflation (many participants with 0 aggressive completions)
- Normality assumption is violated, therefore a t-test is not appropriate

7.2.2 Homogeneity of Variance

```
check_homogeneity(df, aggressiveWordCompletionScoreTransformed, threatCondition)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  0.1237 0.7256
##       152
```

7.3 Inference Statistics: Binomial GLM

Since the outcome is a proportion (number of aggressive completions out of 10 trials) and assumptions for parametric tests are violated, we use a binomial GLM.

7.3.1 Fit Model

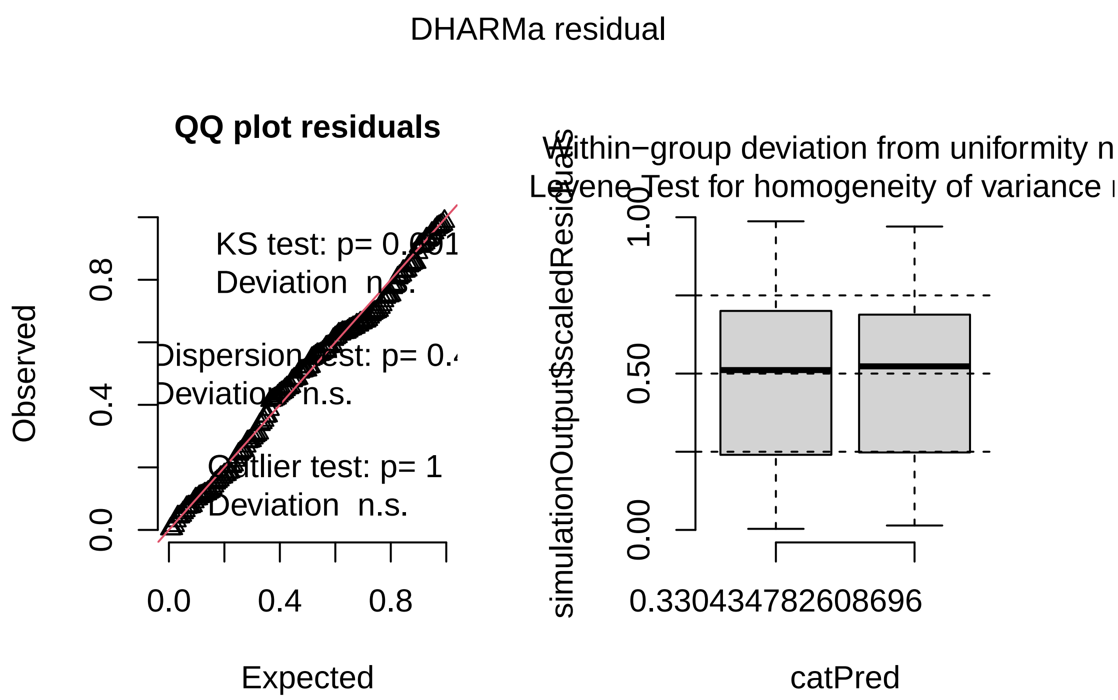
```
N_TRIALS_AGGRESSION <- 10

glm_agg <- run_binomial_glm(
  data = df,
  score_col = "aggressiveWordCompletionScore",
  predictor = "threatCondition",
  n_trials = N_TRIALS_AGGRESSION
```

```r
)

summary(glm_agg$model)
```

```
##
## Call:
## glm(formula = formula, family = glm_family, data = data)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.22050    0.08482 -14.389   <2e-16 ***
## threatConditionthreat -0.19947    0.12530  -1.592    0.111
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 163.12  on 153  degrees of freedom
## Residual deviance: 160.58  on 152  degrees of freedom
## AIC: 498.55
##
## Number of Fisher Scoring iterations: 4
```

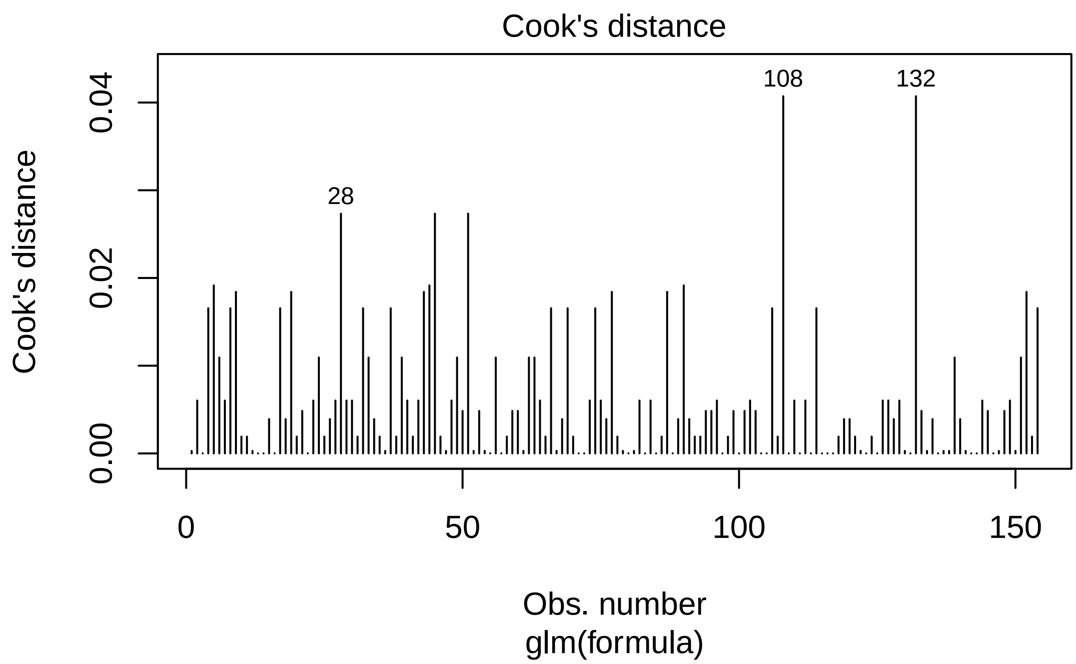
7.3.2 Model Diagnostics: Residuals

```r
check_glm_residual_normality(glm_agg$model)
```



7.3.3 Model Diagnostics: Cook's Distance

```r
plot(glm_agg$model, which = 4)
```




```r
diag_agg <- glm_cooks_diagnostics(glm_agg$model)
print_cooks_summary(diag_agg)
```

```
## Cook's Distance Threshold: 0.0263
## Total observations: 154
## Influential observations: 5
##
## Influential observation indices: 28 45 51 108 132
## Cook's Distance values:
##     28     45     51    108    132
## 0.0273 0.0273 0.0273 0.0407 0.0407
```

7.3.4 Sensitivity Analysis: Model without Influential Observations

```r
glm_agg_clean <- glm_remove_influential(glm_agg, diag_agg, "threatCondition")
```

```
## --- Data Cleaning Summary ---
## Original observations: 154
## Removed (influential): 5
## Remaining observations: 149
```

```r
summary_mean_sd(glm_agg_clean$data_clean, "aggressiveWordCompletionScore",
↪  "threatCondition")
```

```
## # A tibble: 2 x 6
##   threatCondition variable                       mean    sd     n missing
```

Table 2: GLM Results: Effect of Threat Condition on Aggressive Cognition

Term	Estimate	SE	z	p	Sig	Estimate_sens	SE_sens	z_sens	p_sens	Sig_
Intercept (No Threat)	-1.221	0.085	-14.39	$<$ .001	***	-1.283	0.088	-14.58	$<$ .001	***
Threat Condition	-0.199	0.125	-1.59	0.944		-0.192	0.130	-1.48	0.930	

```
##   <fct>          <chr>                          <dbl> <dbl> <dbl>   <dbl>
## 1 noThreat       aggressiveWordCompletionScore  0.22  0.11    76       0
## 2 threat         aggressiveWordCompletionScore  0.19  0.11    73       0
```

```r
summary(glm_agg_clean$model_clean)
```

```
##
## Call:
## glm(formula = formula, family = glm_family, data = data_clean)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.28262    0.08798 -14.578   <2e-16 ***
## threatConditionthreat -0.19161    0.12953  -1.479    0.139
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 142.06  on 148  degrees of freedom
## Residual deviance: 139.87  on 147  degrees of freedom
## AIC: 463.83
##
## Number of Fisher Scoring iterations: 4
```

7.3.5 Results Summary

```r
glm_summary_table(
  model = glm_agg$model,
  model_clean = glm_agg_clean$model_clean,
  labels = c(
    "(Intercept)" = "Intercept (No Threat)",
    "threatConditionthreat" = "Threat Condition"
  ),
  one_tailed = c("threatConditionthreat" = "positive")
) %>%
  kbl(caption = "GLM Results: Effect of Threat Condition on Aggressive Cognition") %>%
  kable_styling(bootstrap_options = c("striped", "hover"), full_width = FALSE)
```

Note on p-value correction: The hypothesis predicts a directional effect (threat condition increases aggressive cognition). The GLM produces two-tailed p-values by default. For the threat condition coefficient, we apply a one-tailed correction: if the effect is in the expected direction (positive coefficient), p_one-tailed = p_two-tailed / 2; if opposite, p_one-tailed = 1 - (p_two-tailed / 2).

Interpretation:

- …

8. Effect of Threat Condition on Anxious Cognition (Hypothesis 1b)

Objectives:

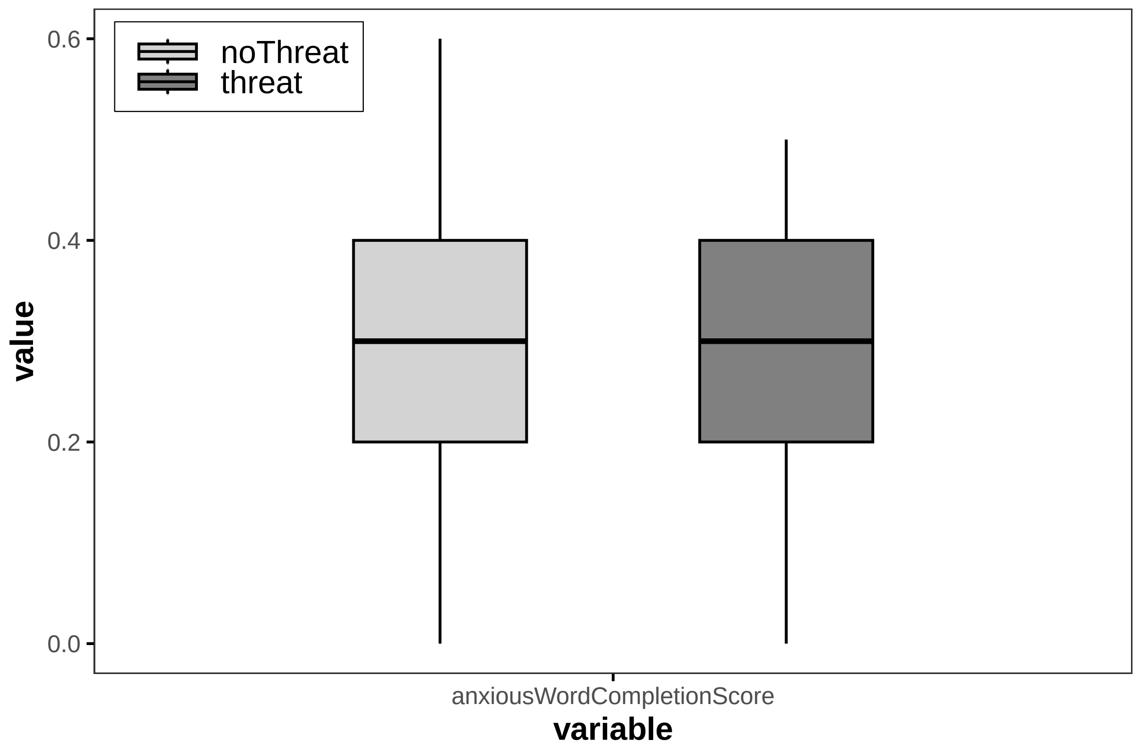
- Assess if there is a statistically significant effect of threat condition on anxious cognition

Methodology:

- Visualization & Descriptives
- Check assumptions for t-test (normality, homogeneity)
- Since assumptions are violated: Binomial GLM on proportion data

8.1 Visualization & Descriptives

```
plotBox("anxiousWordCompletionScore", "threatCondition")
```




```
summary_mean_sd(df, "anxiousWordCompletionScore", "threatCondition")
```

```
## # A tibble: 2 x 6
##   threatCondition variable                    mean    sd     n missing
##   <fct>           <chr>                      <dbl> <dbl> <dbl>   <dbl>
## 1 noThreat        anxiousWordCompletionScore  0.28  0.15    79       0
## 2 threat          anxiousWordCompletionScore  0.28  0.13    75       0
```

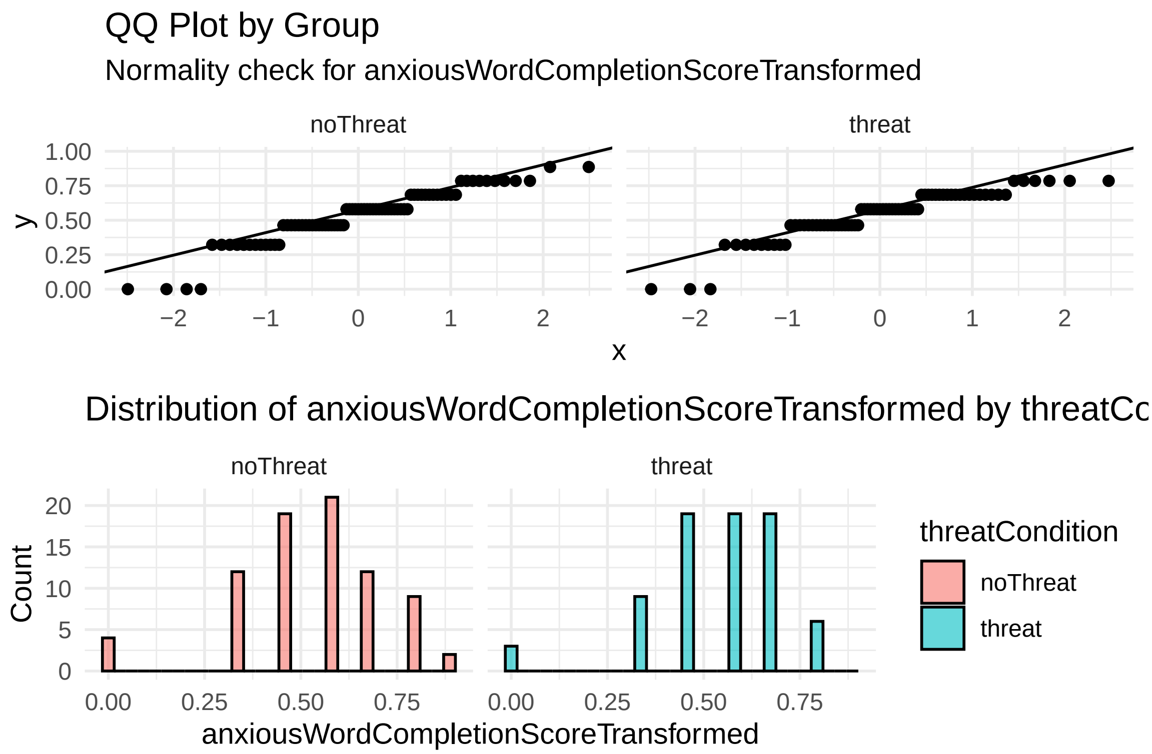
8.2 Check T-Test Assumptions

To ensure independence between value and variance (as pre-registered), we apply an arcsine square root transformation to the proportion scores before checking assumptions.

```r
df$anxiousWordCompletionScoreTransformed <- asin(sqrt(df$anxiousWordCompletionScore))
```

8.2.1 Normality within each group

```r
anx_qq <- check_normality_qq(df, anxiousWordCompletionScoreTransformed, threatCondition)
anx_hist <- check_normality_hist(df, anxiousWordCompletionScoreTransformed,
  threatCondition)
grid.arrange(anx_qq, anx_hist, ncol = 1)
```



Interpretation:

- Even after arcsine-sqrt transformation, the distribution shows zero-inflation
- Normality assumption is violated, therefore a t-test is not appropriate

8.2.2 Homogeneity of Variance

```r
check_homogeneity(df, anxiousWordCompletionScoreTransformed, threatCondition)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1    0.65 0.4214
##       152
```

8.3 Inference Statistics: Binomial GLM

Since the outcome is a proportion (number of anxious completions out of 10 trials) and assumptions for parametric tests are violated, we use a binomial GLM.

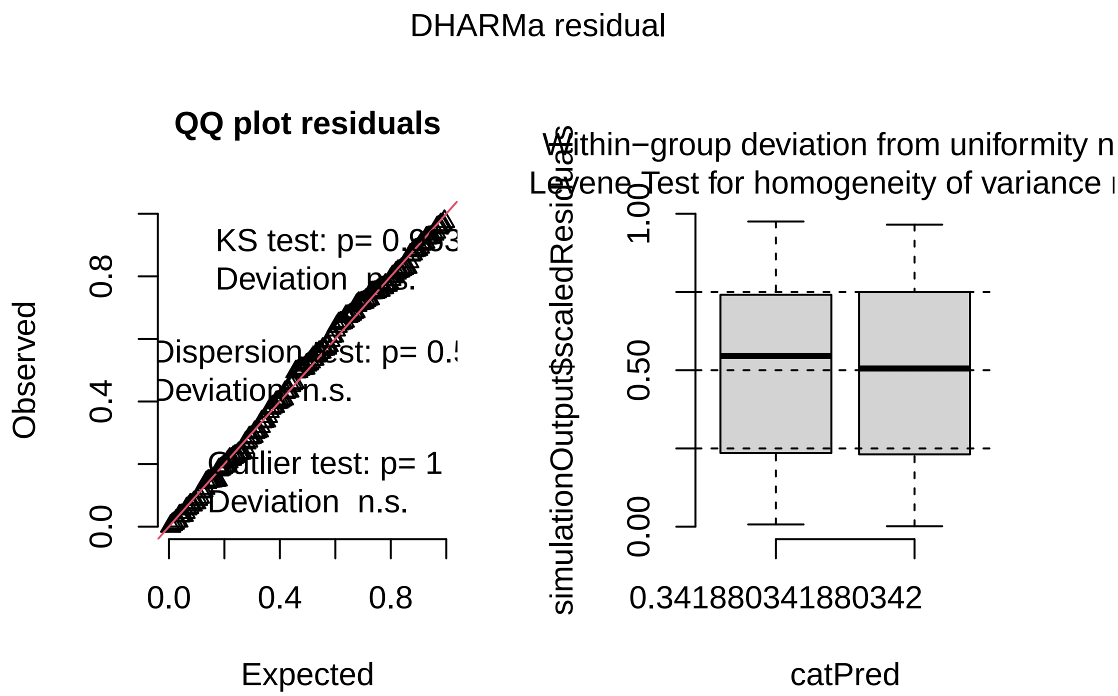
8.3.1 Fit Model

```
N_TRIALS_ANXIETY <- 10

glm_anx <- run_binomial_glm(
  data = df,
  score_col = "anxiousWordCompletionScore",
  predictor = "threatCondition",
  n_trials = N_TRIALS_ANXIETY
)

summary(glm_anx$model)
```

```
##
## Call:
## glm(formula = formula, family = glm_family, data = data)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -0.96464    0.07960 -12.119   <2e-16 ***
## threatConditionthreat    0.02018    0.11379   0.177    0.859
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 162.55  on 153  degrees of freedom
## Residual deviance: 162.52  on 152  degrees of freedom
## AIC: 536.38
##
## Number of Fisher Scoring iterations: 4
```

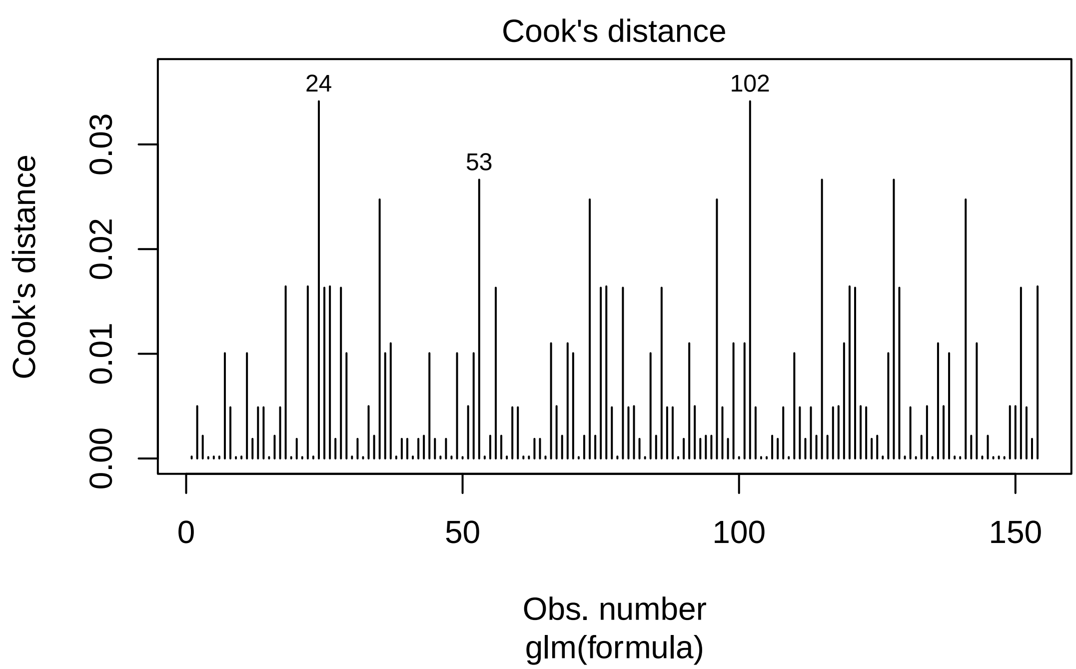
8.3.2 Model Diagnostics: Residuals

```
check_glm_residual_normality(glm_anx$model)
```



8.3.3 Model Diagnostics: Cook's Distance

```
plot(glm_anx$model, which = 4)
```



```r
diag_anx <- glm_cooks_diagnostics(glm_anx$model)
print_cooks_summary(diag_anx)
```

```
## Cook's Distance Threshold: 0.0263
## Total observations: 154
## Influential observations: 5
##
## Influential observation indices: 24 53 102 115 128
## Cook's Distance values:
##     24     53    102    115    128
## 0.0341 0.0266 0.0341 0.0266 0.0266
```

8.3.4 Sensitivity Analysis: Model without Influential Observations

```r
glm_anx_clean <- glm_remove_influential(glm_anx, diag_anx, "threatCondition")
```

```
## --- Data Cleaning Summary ---
## Original observations: 154
## Removed (influential): 5
## Remaining observations: 149
```

```r
summary_mean_sd(glm_anx_clean$data_clean, "anxiousWordCompletionScore",
 ↪  "threatCondition")
```

```
## # A tibble: 2 x 6
##   threatCondition variable                    mean    sd     n missing
##   <fct>           <chr>                      <dbl> <dbl> <dbl>   <dbl>
## 1 noThreat        anxiousWordCompletionScore  0.27  0.14    77       0
## 2 threat          anxiousWordCompletionScore  0.29  0.12    72       0
```

```r
summary(glm_anx_clean$model_clean)
```

```
##
## Call:
## glm(formula = formula, family = glm_family, data = data_clean)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.00718    0.08141 -12.372   <2e-16 ***
## threatConditionthreat  0.11987    0.11554   1.037      0.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 133.99  on 148  degrees of freedom
## Residual deviance: 132.91  on 147  degrees of freedom
## AIC: 501.24
##
## Number of Fisher Scoring iterations: 4
```

Table 3: GLM Results: Effect of Threat Condition on Anxious Cognition

Term	Estimate	SE	z	p	Sig	Estimate_sens	SE_sens	z_sens	p_sens	Sig_
Intercept (No Threat)	-0.965	0.080	-12.12	$< .001$	***	-1.007	0.081	-12.37	$< .001$	***
Threat Condition	0.020	0.114	0.18	0.430		0.120	0.116	1.04	0.150	

8.3.5 Results Summary

```
glm_summary_table(
  model = glm_anx$model,
  model_clean = glm_anx_clean$model_clean,
  labels = c(
    "(Intercept)" = "Intercept (No Threat)",
    "threatConditionthreat" = "Threat Condition"
  ),
  one_tailed = c("threatConditionthreat" = "positive")
) %>%
  kbl(caption = "GLM Results: Effect of Threat Condition on Anxious Cognition") %>%
  kable_styling(bootstrap_options = c("striped", "hover"), full_width = FALSE)
```

Note on p-value correction: The hypothesis predicts a directional effect (threat condition increases anxious cognition). One-tailed p-values are reported for the threat condition coefficient.

Interpretation:

- …

9. Moderation: Pressured Motivation on Aggressive Cognition (Hypothesis 2a)

Objectives:

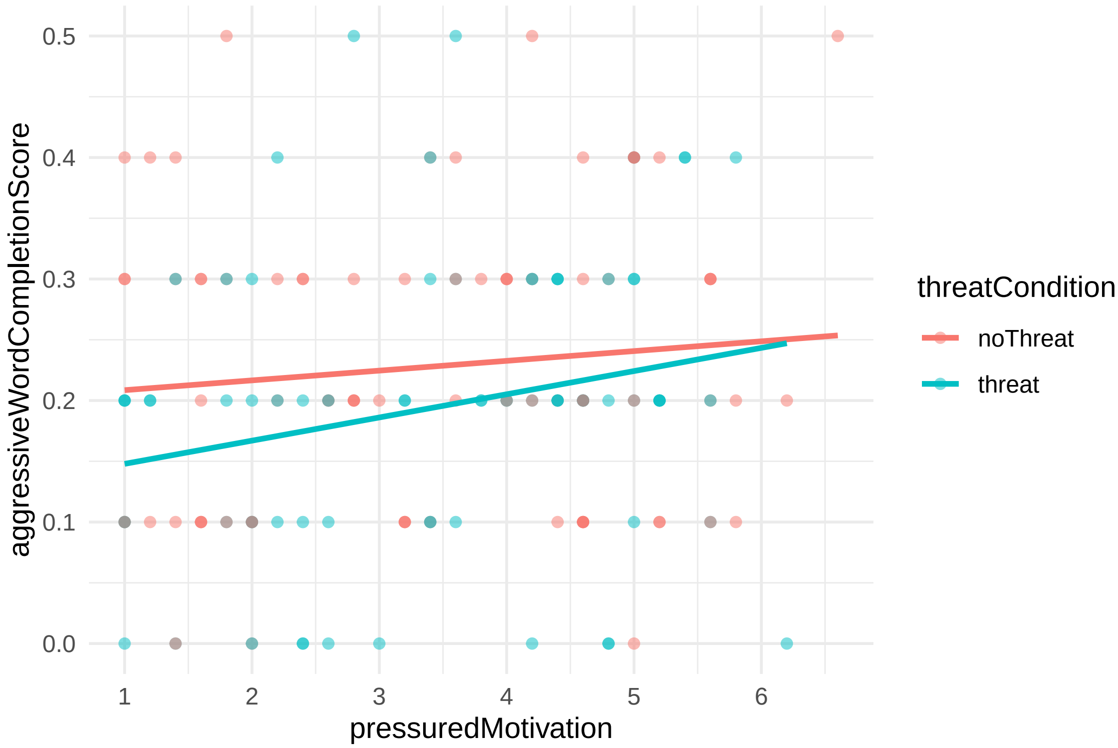
- Assess if pressured motivation for masculine behaviour moderates the effect of threat condition on aggressive cognition

Methodology:

- Visualization & Descriptives
- Check assumptions for multiple linear regression
- Since assumptions are violated: Binomial GLM with interaction term

9.1 Visualization

```
plotLine("pressuredMotivation", "aggressiveWordCompletionScore",
  condition = "threatCondition", df = df
)
```



9.2 Check Linear Regression Assumptions

Using the arcsine-sqrt transformed scores (as pre-registered) to check assumptions.

9.2.1 Multicollinearity

```
check_multicollinearity(df, "threatCondition", "pressuredMotivation")
```

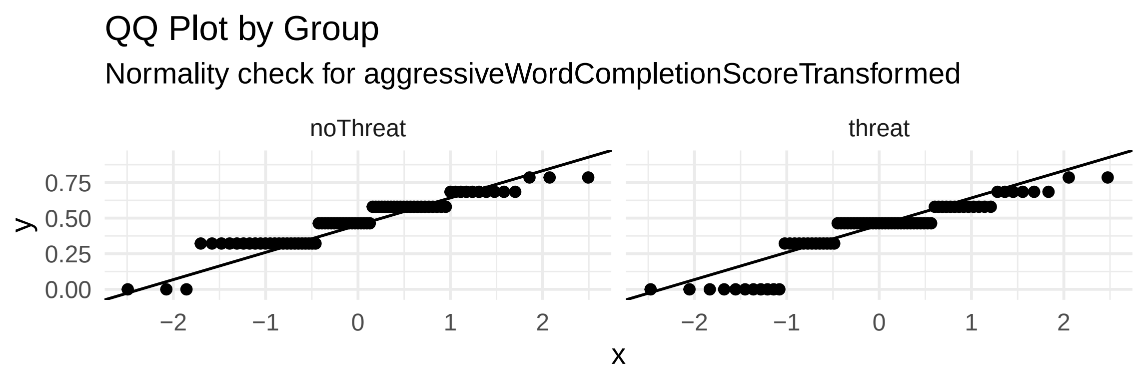
```
## Correlation between threatCondition and pressuredMotivation : 0.016
## Interpretation: Low correlation - no multicollinearity concern
```

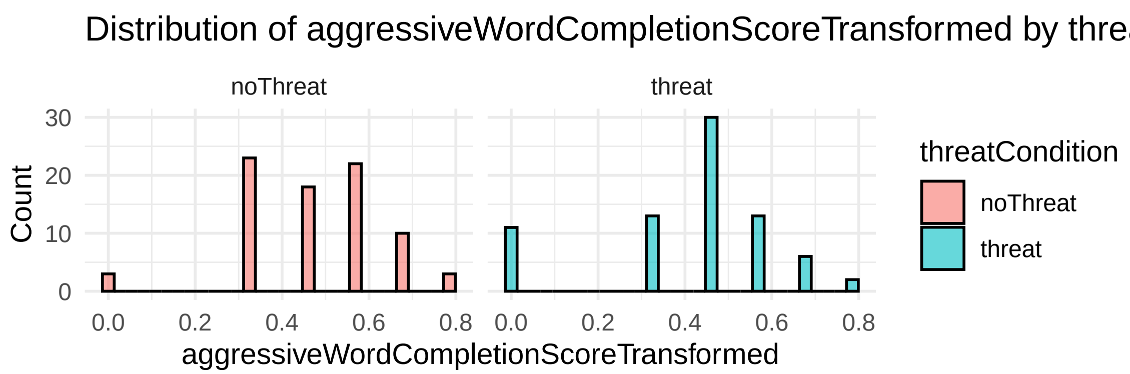
Interpretation:

- Since threat condition is randomly assigned, we expect no correlation with the moderator

9.2.2 Normality of Outcome

```
agg_mod_qq <- check_normality_qq(df, aggressiveWordCompletionScoreTransformed,
 ↪  threatCondition)
agg_mod_hist <- check_normality_hist(df, aggressiveWordCompletionScoreTransformed,
 ↪  threatCondition)
grid.arrange(agg_mod_qq, agg_mod_hist, ncol = 1)
```





Interpretation:

- Even after arcsine-sqrt transformation, the outcome shows zero-inflation
- Normality assumption for linear regression is violated
- Therefore we use a binomial GLM with interaction term

9.3 Inference Statistics: Binomial GLM with Moderation

We model the interaction between threat condition and pressured motivation to test whether pressured motivation moderates the effect of threat on aggressive cognition.

9.3.1 Fit Model

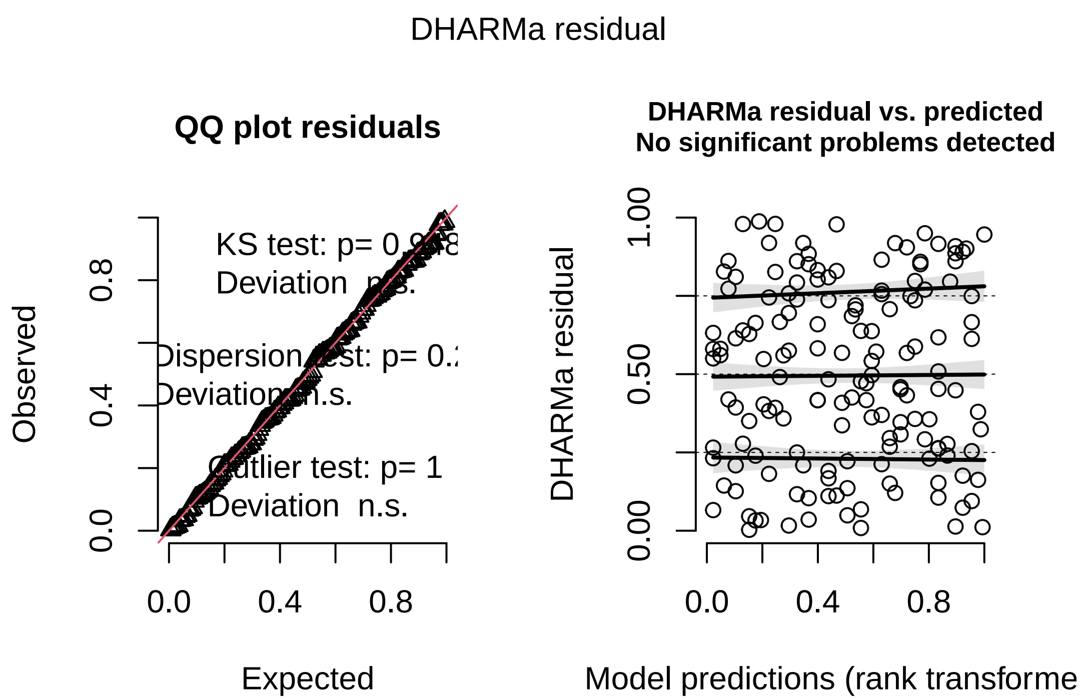
```
glm_agg_mod <- run_binomial_glm_moderation(
  data = df,
  score_col = "aggressiveWordCompletionScore",
  predictor = "threatCondition",
  moderator = "pressuredMotivation",
  n_trials = N_TRIALS_AGGRESSION
)

summary(glm_agg_mod$model)
```

```
##
## Call:
## glm(formula = formula, family = glm_family, data = data)
##
## Coefficients:
##                                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                                 -1.37743    0.20728  -6.645 3.03e-11
```

```
## threatConditionthreat                         -0.47882    0.32492  -1.474    0.141
## pressuredMotivation                            0.04572    0.05462   0.837    0.403
## threatConditionthreat:pressuredMotivation      0.07786    0.08445   0.922    0.357
##
## (Intercept)                                   ***
## threatConditionthreat
## pressuredMotivation
## threatConditionthreat:pressuredMotivation
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 163.12  on 153  degrees of freedom
## Residual deviance: 156.14  on 150  degrees of freedom
## AIC: 498.12
##
## Number of Fisher Scoring iterations: 4
```

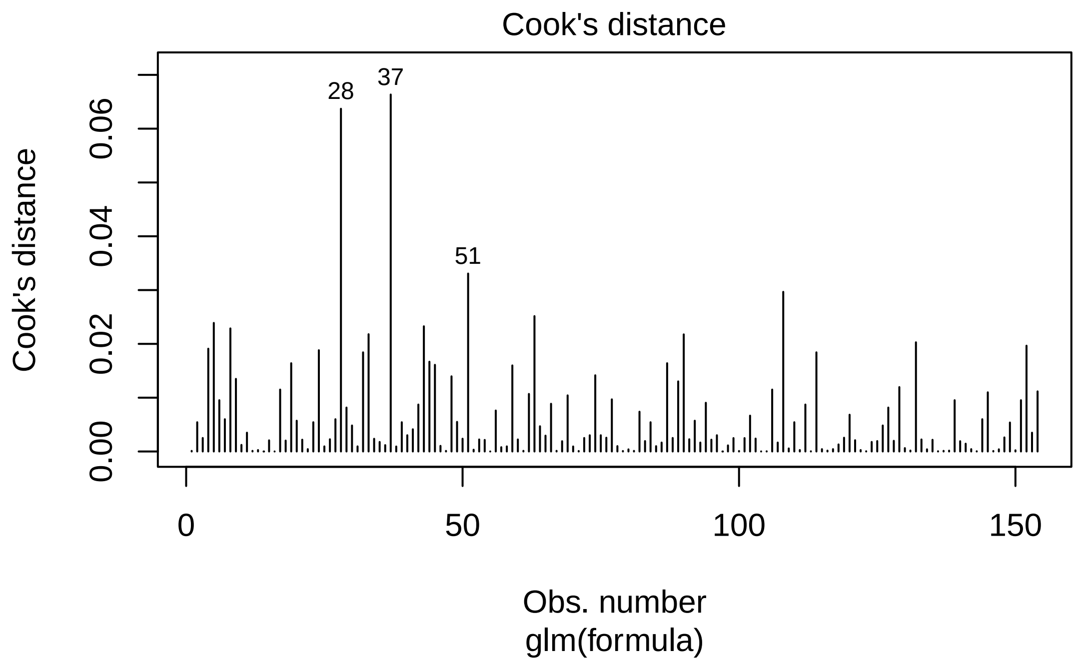
9.3.2 Model Diagnostics: Residuals

```
check_glm_residual_normality(glm_agg_mod$model)
```



9.3.3 Model Diagnostics: Cook's Distance

```
plot(glm_agg_mod$model, which = 4)
```




```
diag_agg_mod <- glm_cooks_diagnostics(glm_agg_mod$model)
print_cooks_summary(diag_agg_mod)
```

```
## Cook's Distance Threshold: 0.0267
## Total observations: 154
## Influential observations: 4
##
## Influential observation indices: 28 37 51 108
## Cook's Distance values:
##     28     37     51    108
## 0.0637 0.0663 0.0331 0.0297
```

9.3.4 Sensitivity Analysis: Model without Influential Observations

```
glm_agg_mod_clean <- glm_remove_influential_moderation(
  glm_agg_mod, diag_agg_mod,
  "threatCondition", "pressuredMotivation"
)
```

```
## --- Data Cleaning Summary ---
## Original observations: 154
## Removed (influential): 4
## Remaining observations: 150
```

```
summary(glm_agg_mod_clean$model_clean)
```

```
##
## Call:
## glm(formula = formula, family = glm_family, data = data_clean)
```

Table 4: GLM Results: Moderation of Threat Effect by Pressured Motivation on Aggressive Cognition

Term	Estimate	SE	z	p	Sig	Estimate_sens	SE_sens	z_sens	p_sens
Intercept	-1.377	0.207	-6.65	< .001	***	-1.382	0.215	-6.44	< .001
Threat Condition	-0.479	0.325	-1.47	0.930		-0.626	0.338	-1.85	0.968
Pressured Motivation	0.046	0.055	0.84	0.403		0.036	0.057	0.62	0.534
Threat x Pressured Motivation	0.078	0.084	0.92	0.178		0.128	0.088	1.46	0.073

```
##
## Coefficients:
##                                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                                -1.38234    0.21451  -6.444 1.16e-10
## threatConditionthreat                      -0.62571    0.33790  -1.852   0.0641
## pressuredMotivation                         0.03560    0.05722   0.622   0.5338
## threatConditionthreat:pressuredMotivation  0.12845    0.08819   1.457   0.1453
##
## (Intercept)                                ***
## threatConditionthreat                      .
## pressuredMotivation
## threatConditionthreat:pressuredMotivation
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 146.07  on 149  degrees of freedom
## Residual deviance: 137.81  on 146  degrees of freedom
## AIC: 471.38
##
## Number of Fisher Scoring iterations: 4
```

9.3.5 Results Summary

```
glm_summary_table(
  model = glm_agg_mod$model,
  model_clean = glm_agg_mod_clean$model_clean,
  labels = c(
    "(Intercept)" = "Intercept",
    "threatConditionthreat" = "Threat Condition",
    "pressuredMotivation" = "Pressured Motivation",
    "threatConditionthreat:pressuredMotivation" = "Threat x Pressured Motivation"
  ),
  one_tailed = c(
    "threatConditionthreat" = "positive",
    "threatConditionthreat:pressuredMotivation" = "positive"
  )
) %>%
  kbl(caption = "GLM Results: Moderation of Threat Effect by Pressured Motivation on
↪  Aggressive Cognition") %>%
  kable_styling(bootstrap_options = c("striped", "hover"), full_width = FALSE)
```

Note on p-value correction: The hypothesis predicts that pressured motivation positively moderates the effect of threat on aggressive cognition (i.e., higher pressured motivation strengthens the threat effect). One-tailed p-values are reported for the threat condition and interaction coefficients.

Interpretation:

- The interaction term (threatCondition:pressuredMotivation) tests whether pressured motivation moderates the effect of threat condition on aggressive cognition
- …

10. Moderation: Autonomous Motivation on Anxious Cognition (Hypothesis 2b)

Objectives:

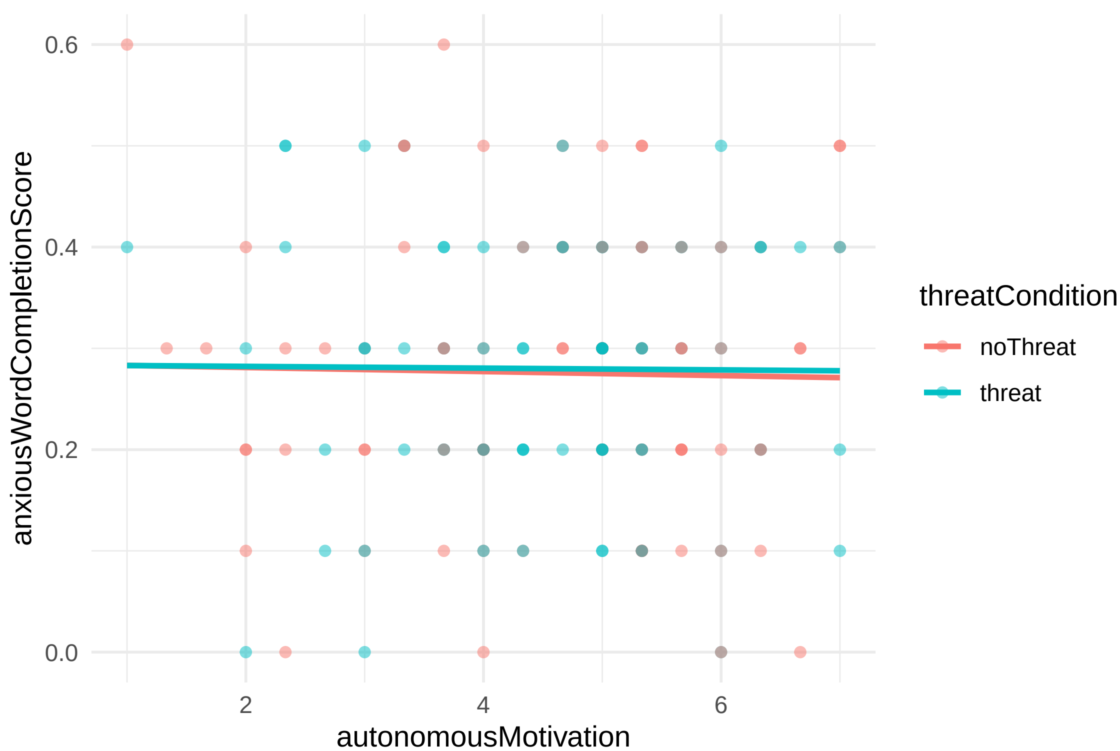
- Assess if autonomous motivation for masculine behaviour moderates the effect of threat condition on anxious cognition

Methodology:

- Visualization & Descriptives
- Check assumptions for multiple linear regression
- Since assumptions are violated: Binomial GLM with interaction term

10.1 Visualization

```
plotLine("autonomousMotivation", "anxiousWordCompletionScore",
  condition = "threatCondition", df = df
)
```



10.2 Check Linear Regression Assumptions

Using the arcsine-sqrt transformed scores (as pre-registered) to check assumptions.

10.2.1 Multicollinearity

```
check_multicollinearity(df, "threatCondition", "autonomousMotivation")
```

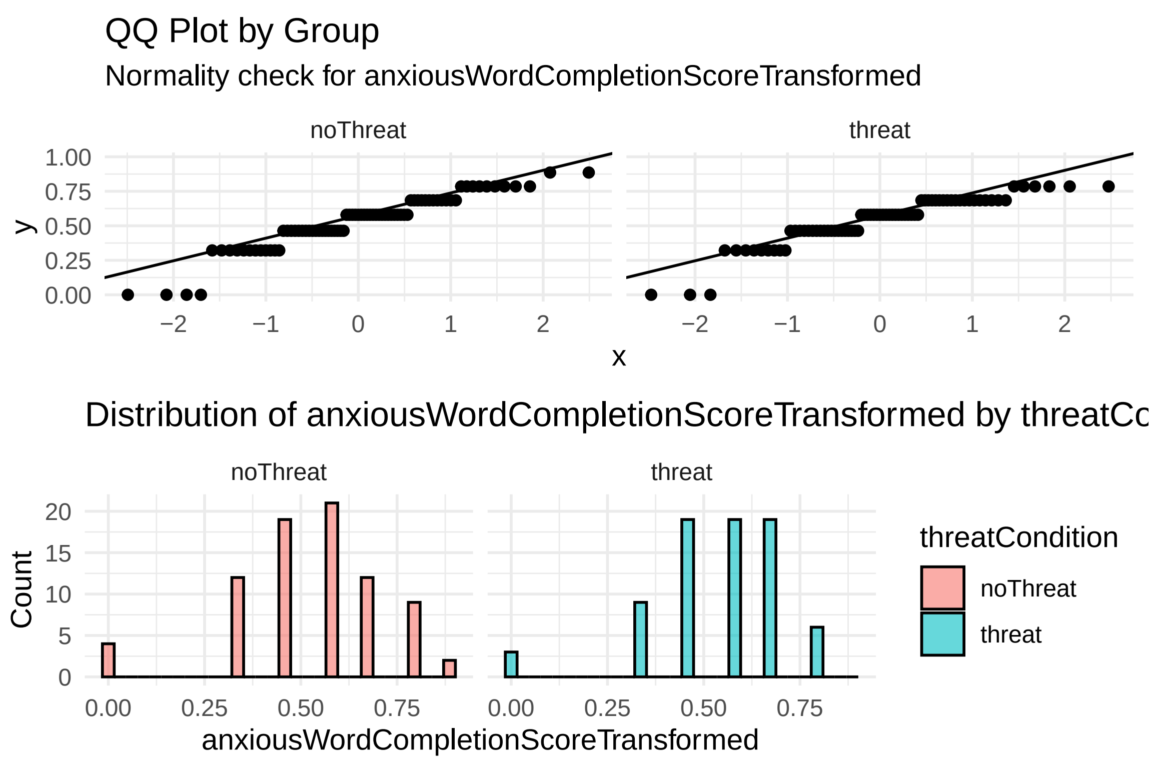
```
## Correlation between threatCondition and autonomousMotivation : -0.0179
## Interpretation: Low correlation - no multicollinearity concern
```

Interpretation:

- Since threat condition is randomly assigned, we expect no correlation with the moderator

10.2.2 Normality of Outcome

```
anx_mod_qq <- check_normality_qq(df, anxiousWordCompletionScoreTransformed,
↪  threatCondition)
anx_mod_hist <- check_normality_hist(df, anxiousWordCompletionScoreTransformed,
↪  threatCondition)
grid.arrange(anx_mod_qq, anx_mod_hist, ncol = 1)
```



Interpretation:

- Even after arcsine-sqrt transformation, the outcome shows zero-inflation
- Normality assumption for linear regression is violated
- Therefore we use a binomial GLM with interaction term

10.3 Inference Statistics: Binomial GLM with Moderation

We model the interaction between threat condition and autonomous motivation to test whether autonomous motivation moderates the effect of threat on anxious cognition.

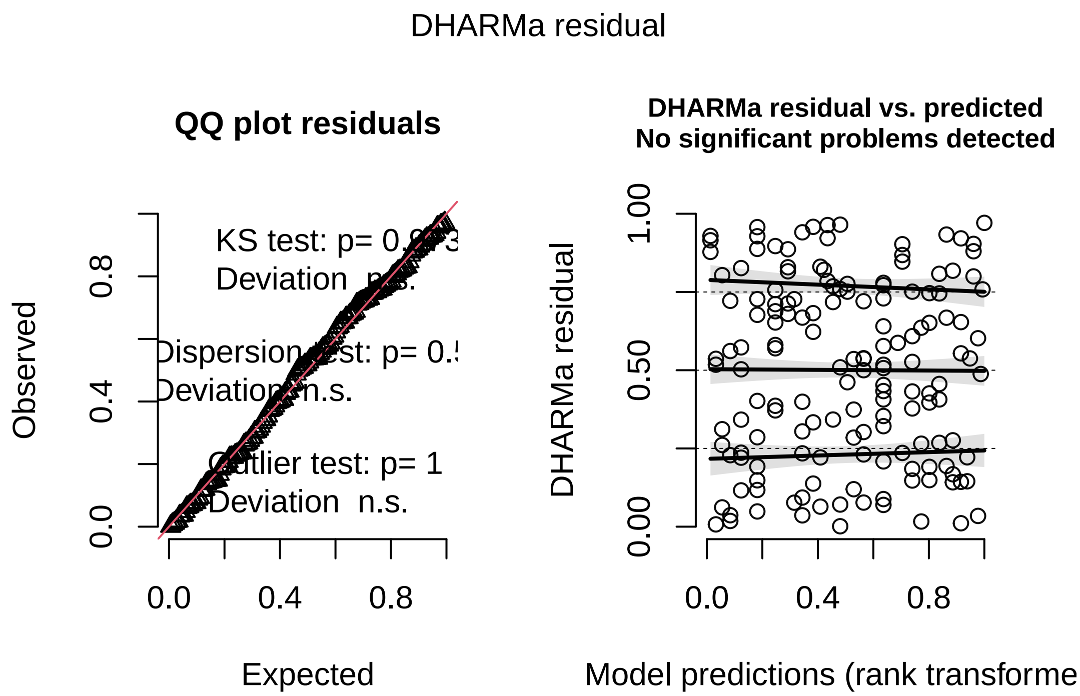
10.3.1 Fit Model

```
glm_anx_mod <- run_binomial_glm_moderation(
  data = df,
  score_col = "anxiousWordCompletionScore",
  predictor = "threatCondition",
  moderator = "autonomousMotivation",
  n_trials = N_TRIALS_ANXIETY
)

summary(glm_anx_mod$model)
```

```
##
## Call:
## glm(formula = formula, family = glm_family, data = data)
##
## Coefficients:
##                                              Estimate Std. Error z value
## (Intercept)                                 -0.918992   0.261592  -3.513
## threatConditionthreat                       -0.006093   0.396485  -0.015
## autonomousMotivation                        -0.009982   0.054544  -0.183
## threatConditionthreat:autonomousMotivation   0.005702   0.083582   0.068
##                                             Pr(>|z|)
## (Intercept)                                 0.000443 ***
## threatConditionthreat                       0.987740
## autonomousMotivation                        0.854788
## threatConditionthreat:autonomousMotivation 0.945608
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 162.55  on 153  degrees of freedom
## Residual deviance: 162.49  on 150  degrees of freedom
## AIC: 540.34
##
## Number of Fisher Scoring iterations: 4
```

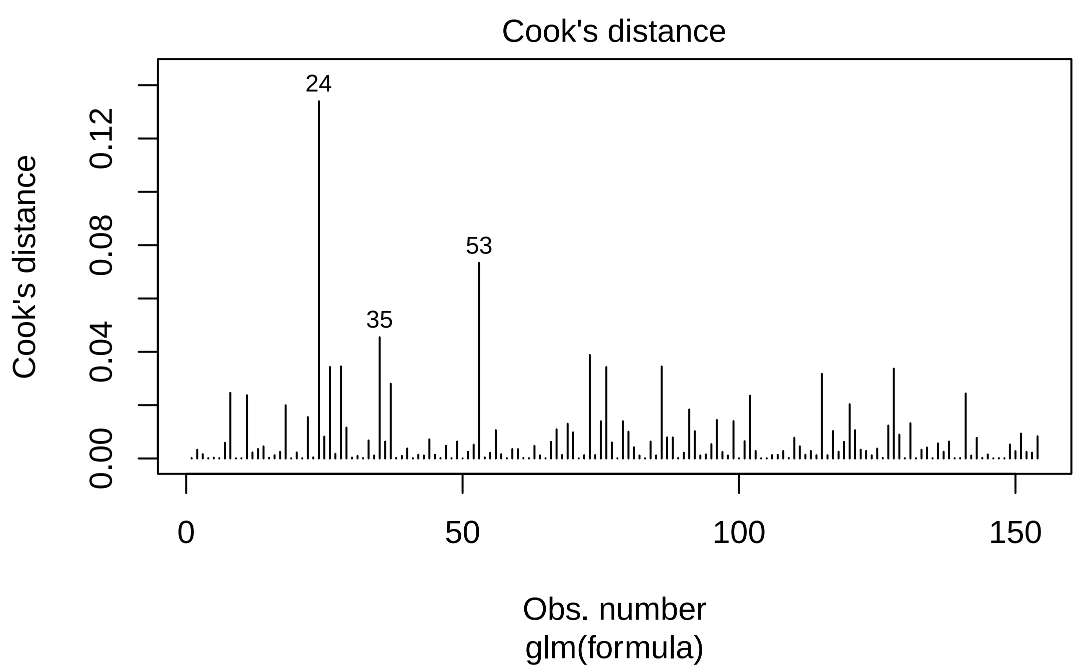
10.3.2 Model Diagnostics: Residuals

```
check_glm_residual_normality(glm_anx_mod$model)
```



10.3.3 Model Diagnostics: Cook's Distance

```
plot(glm_anx_mod$model, which = 4)
```



```r
diag_anx_mod <- glm_cooks_diagnostics(glm_anx_mod$model)
print_cooks_summary(diag_anx_mod)
```

```
## Cook's Distance Threshold: 0.0267
## Total observations: 154
## Influential observations: 11
##
## Influential observation indices: 24 26 28 35 37 53 73 76 86 115 128
## Cook's Distance values:
##     24     26     28     35     37     53     73     76     86    115    128
## 0.1340 0.0343 0.0345 0.0455 0.0281 0.0734 0.0388 0.0343 0.0345 0.0317 0.0337
```

10.3.4 Sensitivity Analysis: Model without Influential Observations

```r
glm_anx_mod_clean <- glm_remove_influential_moderation(
  glm_anx_mod, diag_anx_mod,
  "threatCondition", "autonomousMotivation"
)
```

```
## --- Data Cleaning Summary ---
## Original observations: 154
## Removed (influential): 11
## Remaining observations: 143
```

```r
summary(glm_anx_mod_clean$model_clean)
```

```
##
## Call:
## glm(formula = formula, family = glm_family, data = data_clean)
##
## Coefficients:
##                                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                                -0.92693    0.29093  -3.186  0.00144
## threatConditionthreat                      -0.10236    0.44671  -0.229  0.81875
## autonomousMotivation                       -0.01155    0.06127  -0.189  0.85045
## threatConditionthreat:autonomousMotivation  0.03896    0.09395   0.415  0.67835
##
## (Intercept)                                **
## threatConditionthreat
## autonomousMotivation
## threatConditionthreat:autonomousMotivation
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 114.73  on 142  degrees of freedom
## Residual deviance: 114.12  on 139  degrees of freedom
## AIC: 476.1
##
## Number of Fisher Scoring iterations: 4
```

Table 5: GLM Results: Moderation of Threat Effect by Autonomous Motivation on Anxious Cognition

Term	Estimate	SE	z	p	Sig	Estimate_sens	SE_sens	z_sens	p_s
Intercept	-0.919	0.262	-3.51	< .001	***	-0.927	0.291	-3.19	0.00
Threat Condition	-0.006	0.396	-0.02	0.506		-0.102	0.447	-0.23	0.59
Autonomous Motivation	-0.010	0.055	-0.18	0.855		-0.012	0.061	-0.19	0.85
Threat x Autonomous Motivation	0.006	0.084	0.07	0.473		0.039	0.094	0.41	0.33

10.3.5 Results Summary

```r
glm_summary_table(
  model = glm_anx_mod$model,
  model_clean = glm_anx_mod_clean$model_clean,
  labels = c(
    "(Intercept)" = "Intercept",
    "threatConditionthreat" = "Threat Condition",
    "autonomousMotivation" = "Autonomous Motivation",
    "threatConditionthreat:autonomousMotivation" = "Threat x Autonomous Motivation"
  ),
  one_tailed = c(
    "threatConditionthreat" = "positive",
    "threatConditionthreat:autonomousMotivation" = "positive"
  )
) %>%
  kbl(caption = "GLM Results: Moderation of Threat Effect by Autonomous Motivation on
↪  Anxious Cognition") %>%
  kable_styling(bootstrap_options = c("striped", "hover"), full_width = FALSE)
```

Note on p-value correction: The hypothesis predicts that autonomous motivation positively moderates the effect of threat on anxious cognition (i.e., higher autonomous motivation strengthens the threat effect). One-tailed p-values are reported for the threat condition and interaction coefficients.

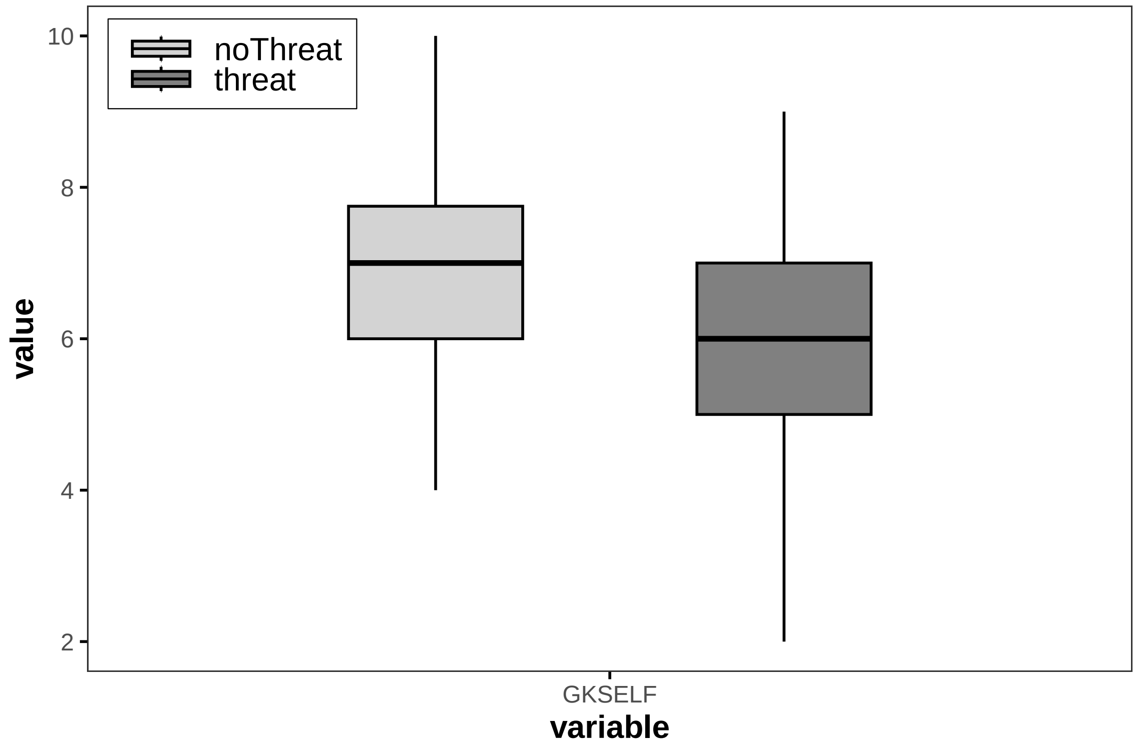
Interpretation:

- The interaction term (threatCondition:autonomousMotivation) tests whether autonomous motivation moderates the effect of threat condition on anxious cognition
- …

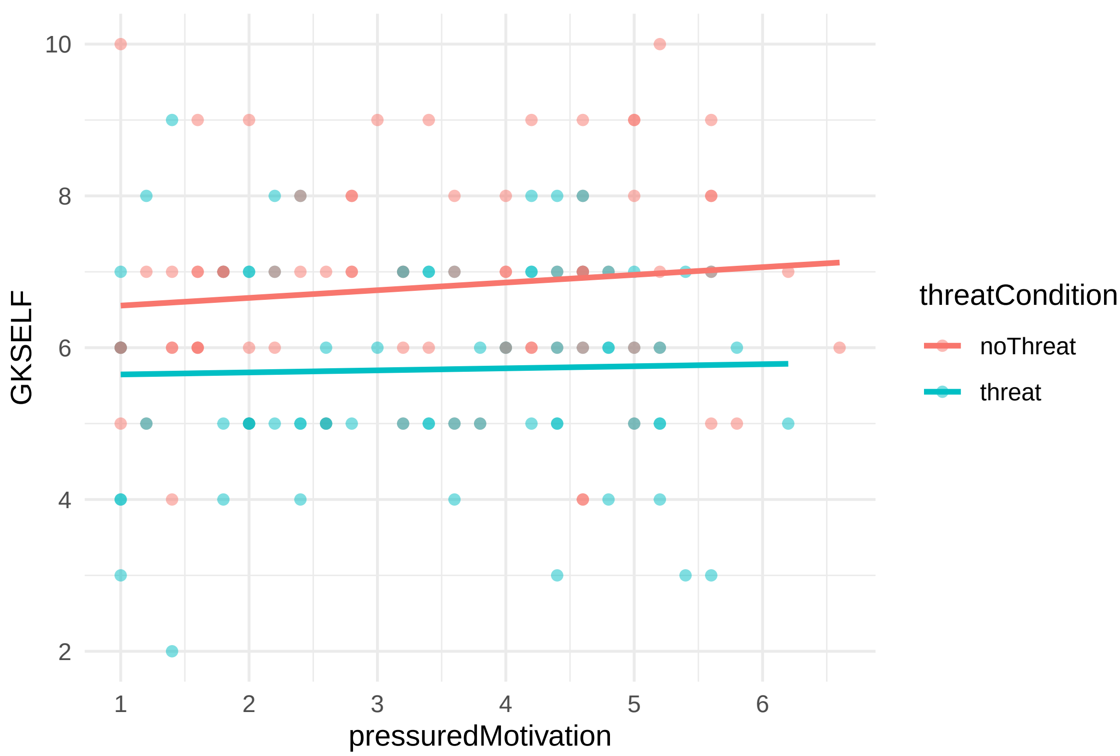
Post Hoc Exploratory Analysis

using GKSELF as a dependent variable, can we check if MMB moderates the impact of threatCondition on GKSELF

```r
plotBox("GKSELF", "threatCondition")
```




```
plotLine("pressuredMotivation", "GKSELF",
  condition = "threatCondition", df = df
)
```



```r
model_gkself <- lm(GKSELF ~
  threatCondition * rowmeansAutonomous, data = df)

summary(model_gkself)
```

```
##
## Call:
## lm(formula = GKSELF ~ threatCondition * rowmeansAutonomous, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4590 -0.9098  0.0579  1.0740  3.4720
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          6.11711    0.51771  11.816   <2e-16
## threatConditionthreat               -1.34831    0.79208  -1.702   0.0908
## rowmeansAutonomous                   0.14864    0.10815   1.374   0.1714
## threatConditionthreat:rowmeansAutonomous  0.05841    0.16650   0.351   0.7262
##
## (Intercept)                         ***
## threatConditionthreat               .
## rowmeansAutonomous
## threatConditionthreat:rowmeansAutonomous
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.392 on 147 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.157,  Adjusted R-squared:  0.1398
## F-statistic: 9.123 on 3 and 147 DF,  p-value: 1.413e-05
```

…

Additional Visualization & Descriptives

Corrplots

```r
save_corrplot_apa(cor_mmb, "corrplot-mmb", cluster_groups = c(1, 5, 6, 9))
save_corrplot_apa(cor_aggr, "corrplot-aggressive-cognition")
save_corrplot_apa(cor_anx, "corrplot-anxious-cognition")
```

Masculinity Threat Effects

Aggressive and anxious word completions by threat condition

```r
agg_anx_by_threat_boxplot <-  plotBox(
    c("aggressiveCognition", "anxiousCognition"),
    "threatCondition",
    x_label = "Cognition Type",
    y_label = "Word Completion Score (%)",
    var_labels = c(
      "aggressiveCognition" = "Aggressive",
      "anxiousCognition" = "Anxious"
```

```r
    ),
    fill_labels = c(
      "noThreat" = "No Threat",
      "threat" = "Threat"
    ),
    legend_title = "Condition",
    ylim = c(0, 70),
    legend_position = "top.left"
  )
agg_anx_by_threat_boxplot
save_plot(agg_anx_by_threat_boxplot, "aggressive-anxious-cognition-by-threat-boxplot")
```

```r
summary_mean_sd(df, c("aggressiveWordCompletionScore", "anxiousWordCompletionScore"),
↪  "threatCondition")
```

Modertation of Masculinity Threat Effects

```r
agg_moderation_pressured_motivation <- plotModeration(
    x = "pressuredMotivation",
    y = "aggressiveCognition",
    condition = "threatCondition",
    df = df,
    x_label = "MMB Pressured Score",
    y_label = "Agg. Word Completion Score (%)",
    condition_labels = c(
      "noThreat" = "No Threat",
      "threat" = "Threat"
    ),
    legend_title = "Condition",
    ylim = c(10, 30),
    legend_position = "top.left"
  )
agg_moderation_pressured_motivation
save_plot(agg_moderation_pressured_motivation,
↪  "aggressive-cognition-pressured-motivation-line-chart")
```

```r
anx_moderation_autonomous_motivation <- plotModeration(
    x = "autonomousMotivation",
    y = "anxiousCognition",
    condition = "threatCondition",
    df = df,
    x_label = "MMB Autonomous Score",
    y_label = "Anx. Word Completion Score (%)",
    condition_labels = c(
      "noThreat" = "No Threat",
      "threat" = "Threat"
    ),
    legend_title = "Condition",
    ylim = c(20, 40),
    legend_position = "top.left"
  )
anx_moderation_autonomous_motivation
save_plot(anx_moderation_autonomous_motivation,
↪  "anxious-cognition-autonomous-motivation-line-chart")
```