

R Code Full Analysis Picture Naming Accuracy

Elise Oosterhuis

Last compiled on 03/08/22

Analysis Picture Naming Accuracy GBG online study

```
##### Read files #####
PNobjects <- read.csv("../Data/Tidy/PNobjects_complete_final.csv") %>%
  filter(Age.Category != "")

PNactions <- read.csv("../Data/Tidy/PNactions_complete_final.csv") %>%
  filter(Age.Category != "")

# head(PNobjects[1:6,1:4]) tail(PNobjects[1:6,1:4])
```

Read in data

Descriptives

Mean and standard deviations picture naming tasks

```
#Suppress summarise message (`summarise()` has grouped output by
#'Task.Name', 'Age.Category'. You can override using the `.groups`
#argument.) in output
options(dplyr.summarise.inform = FALSE)

# Combine both picture-naming datasets
PN_all <- rbind(PNobjects, PNactions) %>%
  convert(chr(type)) %>%
  #Change task names
  dplyr::mutate(Task.Name=dplyr::recode(Task.Name, 'Picture Naming Task - Actions'
    = "Actions", 'Picture Naming Task - Objects'
    = "Objects")) %>%

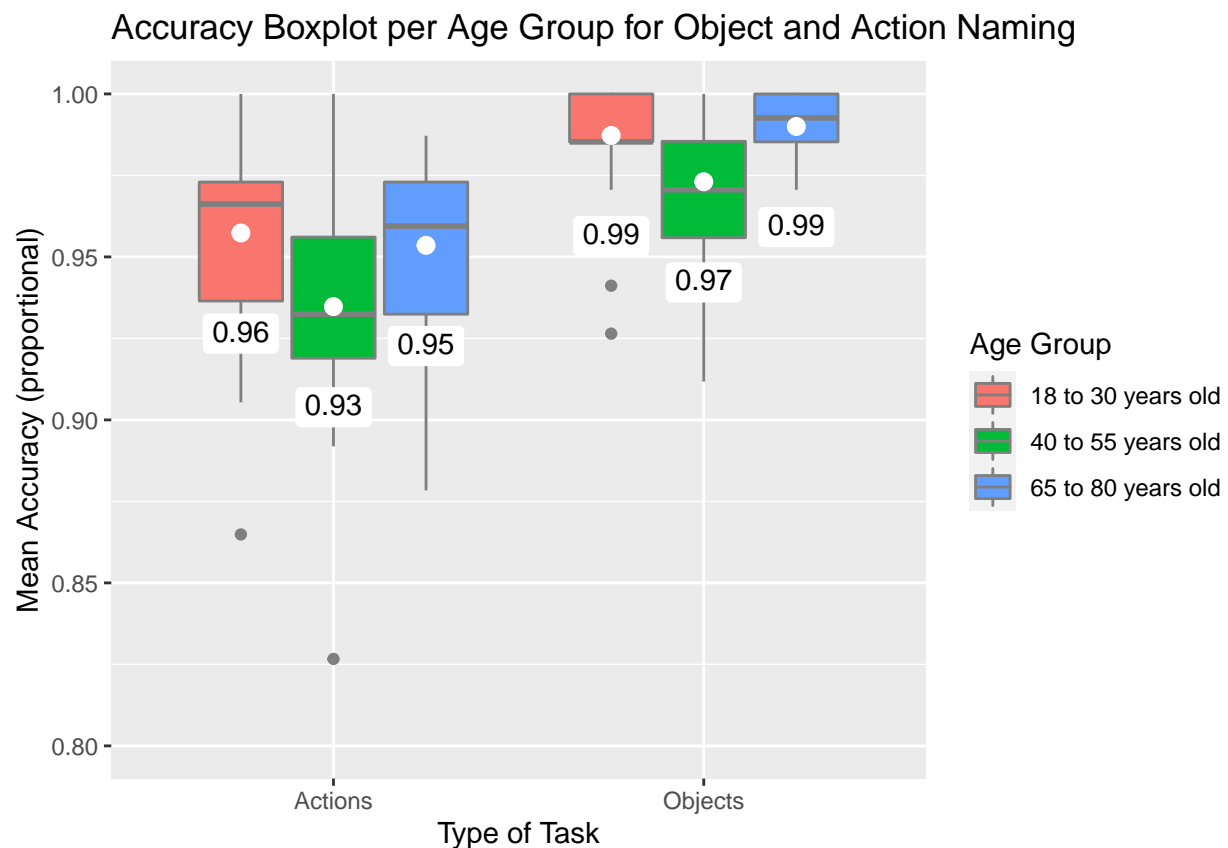
#Recode age groups
dplyr::mutate(Age.Category=as.factor(dplyr::recode(Age.Category,
  '18 to 30 years old'="Younger",
  '40 to 55 years old'="Middle-Aged",
  '65 to 80 years old'="Older")))
```

```
# Mean and SD (in %) for picture naming - Accuracy
(PNacc_sum <- PN_all %>%
  group_by(Task.Name, Age.Category) %>%
  #Obtain mean and standard deviation for reaction time per task
  # (actions and objects separately) and Age Group
  dplyr::summarise(mean_Acc = round((mean(Acc, na.rm=T)*100), 2), #in percentages
    sd_Acc = round((sd(Acc, na.rm = T)*100), 2))) #in percentages
```

```
## # A tibble: 6 x 4
## # Groups:   Task.Name [2]
##   Task.Name Age.Category mean_Acc sd_Acc
##   <chr>      <fct>      <dbl> <dbl>
## 1 Actions   Middle-Aged    95.4  20.9
## 2 Actions   Older         93.5  24.7
## 3 Actions   Younger       94.7  22.4
## 4 Objects   Middle-Aged    98.7  11.2
## 5 Objects   Older         97.3  16.1
## 6 Objects   Younger       99.0   9.86
```

```
## Save output table as .csv file
# write.csv(PNacc_sum, "./Figures and Tables/Descriptives_PNacc.csv", row.names = F)
```

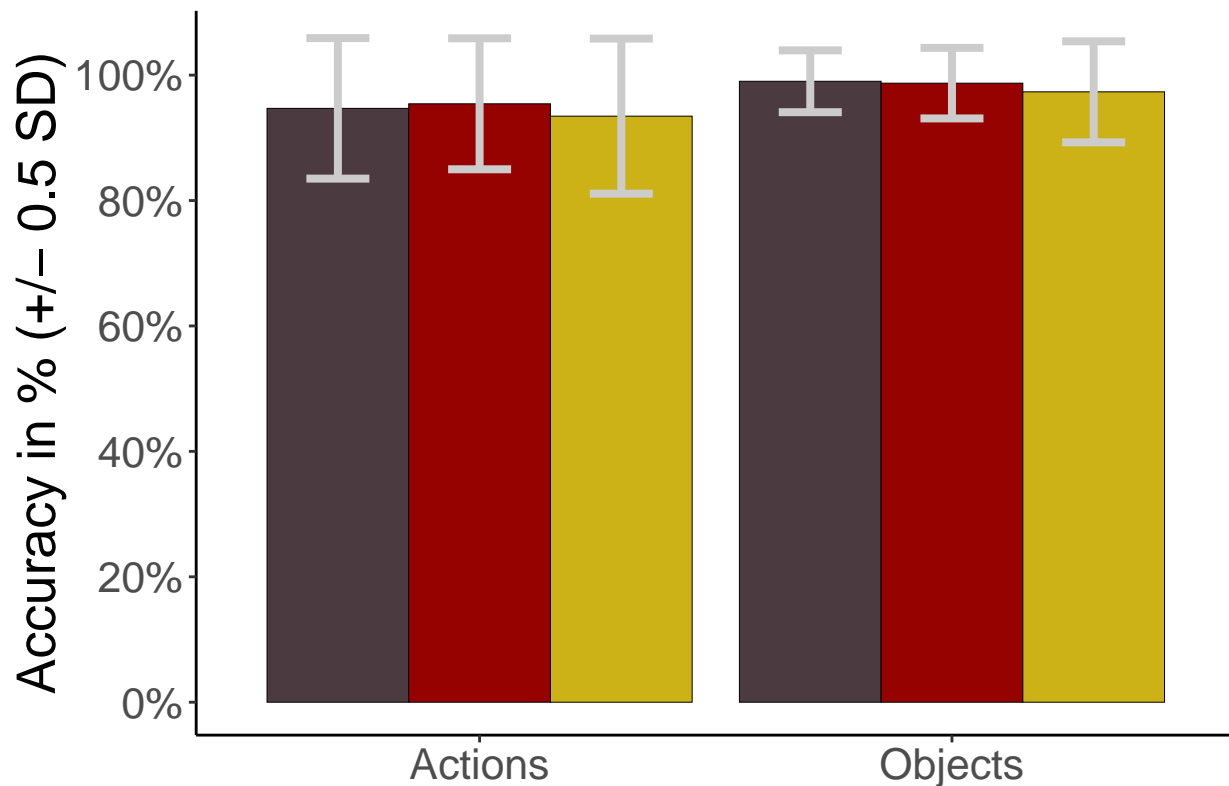
Plots for Accuracy



```
# Barplot for Accuracy

# tiff(file='../Figures and Tables/PNacc_AgeGroups.tiff',
# width=800, height=700)

(Barplot_PNacc <- PN_all %>%
  mutate(Age.Category_ordered = factor(Age.Category, levels = c("Younger",
    "Middle-Aged", "Older")))) %>%
  ggplot(aes(x = Task.Name, y = Acc, fill = as.factor(Age.Category_ordered))) +
  stat_summary(geom = "bar", fun = mean, position = "dodge",
    colour = "black", show.legend = FALSE, size = 0) + geom_errorbar(stat = "summary",
  fun.data = mean_sdl, fun.args = list(mult = 0.5), width = 0.4,
  size = 1.4, position = position_dodge(0.9), colour = "#CCCCCC") +
  coord_cartesian(ylim = c(0, 1.05)) + scale_y_continuous(labels = scales::label_percent(accuracy = 1),
  minor_breaks = seq(0, 1, 0.2), breaks = seq(0, 1, 0.2)) +
  labs(x = "", y = "Accuracy in % (+/- 0.5 SD)") + theme(text = element_text(size = 20),
  panel.background = element_rect(fill = "white"), plot.background = element_rect(fill = "white"),
  strip.background = element_rect(fill = "white"), axis.line.x = element_line(color = "black"),
  axis.line.y = element_line(color = "black"), axis.text.x = element_text(size = 16)) +
  scale_fill_manual(values = confPalette, guide = guide_legend(title = "Age Group")))
```



```
# dev.off()
```

Statistical Analysis

Generalised Linear Mixed Models - Action naming

```
# Only include trials for action naming (i.e., type==1)
PNactions_Acc <- PN_all %>%
  dplyr::filter(type == 1)
# Check unique(PNactions_Acc$Task.Name) #actions
```

Create user-defined contrasts for the Age Category variable We will use Reverse Helmert coding where the first contrast in the model will reflect the difference between Middle-Aged and Younger adults, and the second contrast will reflect the difference between the Older adults and the mean of the Middle-Aged and Younger adults.

```
PNactions_Acc <- mutate(PNactions_Acc, Age.Category = factor(Age.Category,
  levels = c("Middle-Aged", "Younger", "Older")))

PNactions_Acc_coded <- PNactions_Acc
contrasts(PNactions_Acc_coded$Age.Category) <- contr.helmert(3)
contrasts(PNactions_Acc_coded$Age.Category)
```

```
##           [,1] [,2]
## Middle-Aged  -1  -1
## Younger      1   -1
## Older        0    2
```

Should we include random effects for ID and Trial.Number?

```
# Base model with Accuracy as outcome variable. Family
# binomial.
Mbase_act <- glm(Acc ~ 1, family = "binomial", data = PNactions_Acc_coded)

# Base model with only ID/individual variability as random
# effect
Mrand.ID_act <- glmer(Acc ~ 1 + (1 | ID), family = binomial(link = "cloglog"),
  control = glmerControl(optimizer = "bobyqa"), data = PNactions_Acc_coded)
# Base model with only Trial.Number/trial variability as
# random effect
Mrand.Trial_act <- glmer(Acc ~ 1 + (1 | Trial.Number), family = binomial(link = "cloglog"),
  data = PNactions_Acc_coded)
# Base model with both random effects
Mrand.All_act <- glmer(Acc ~ 1 + (1 | ID) + (1 | Trial.Number),
  family = binomial(link = "cloglog"), data = PNactions_Acc_coded)
```

AIC base model

```
# Obtain AIC values for each model
(AIC.base <- AIC(logLik(Mbase_act)))
```

```
## [1] 2994.411
```

AIC - only ID as random effect

```
(AIC.reID <- AIC(logLik(Mrandom.ID_act)))
```

```
## [1] 2965.09
```

AIC - only trial as random effect

```
(AIC.reTrial <- AIC(logLik(Mrandom.Trial_act)))
```

```
## [1] 2806.116
```

AIC - both ID and trial as random effect

```
(AIC.reBoth <- AIC(logLik(Mrandom.All_act)))
```

```
## [1] 2771.468
```

The AIC for the model including both random effects is lowest → we justified inclusion of both Trial and Subject as random effects.

Null model of accuracy for action naming with random effects included cloglog was used because accuracy is a binomial variable with a highly skewed distribution (i.e., more 1's than 0's)

```
MO_PNactAcc <- lme4::glmer(Acc ~ 1 + (1 | ID) + (1 | Trial.Number),  
  data = PNactions_Acc_coded, family = binomial(link = "cloglog"))  
summary(MO_PNactAcc)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( cloglog )  
## Formula: Acc ~ 1 + (1 | ID) + (1 | Trial.Number)  
## Data: PNactions_Acc_coded  
##  
##      AIC      BIC    logLik deviance df.resid  
## 2771.5    2792.0  -1382.7   2765.5     7041  
##  
## Scaled residuals:  
##      Min       1Q   Median       3Q      Max  
## -9.9207  0.1000  0.1695  0.2606  0.6968  
##  
## Random effects:  
## Groups      Name                Variance Std.Dev.  
## ID          (Intercept) 0.03698  0.1923  
## Trial.Number (Intercept) 0.13522  0.3677  
## Number of obs: 7044, groups: ID, 90; Trial.Number, 79  
##  
## Fixed effects:  
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  1.25418    0.05398   23.23  <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Unconditional model, i.e., the model without covariates/control measures

```
Muncond_PNactAcc <- glmer(Acc ~ Age.Category * CR.composite.before +
  (1 | ID) + (1 | Trial.Number), data = PNactions_Acc_coded,
  family = binomial(link = "cloglog"))
```

Full model, i.e., model with covariates/control measures.

```
Mfull_PNactAcc <- lme4::glmer(Acc ~ Age.Category * CR.composite.before +
  GenCogProc.composite + (1 | ID) + (1 | Trial.Number), data = PNactions_Acc_coded,
  family = binomial(link = "cloglog"))
summary(Mfull_PNactAcc)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( cloglog )
## Formula: Acc ~ Age.Category * CR.composite.before + GenCogProc.composite +
## (1 | ID) + (1 | Trial.Number)
## Data: PNactions_Acc_coded
##
##      AIC      BIC   logLik deviance df.resid
## 2765.2   2826.9 -1373.6  2747.2     7035
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -9.5229  0.0984  0.1693  0.2622  0.6927
##
## Random effects:
## Groups      Name                Variance Std.Dev.
## ID          (Intercept)  0.0246     0.1568
## Trial.Number (Intercept)  0.1348     0.3672
## Number of obs: 7044, groups: ID, 90; Trial.Number, 79
##
## Fixed effects:
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.255022    0.052675  23.826 < 2e-16 ***
## Age.Category1     -0.037472    0.034619  -1.082  0.279070
## Age.Category2     -0.052885    0.020914  -2.529  0.011449 *
## CR.composite.before  0.023107    0.026734   0.864  0.387404
## GenCogProc.composite -0.021384    0.059306  -0.361  0.718428
## Age.Category1:CR.composite.before -0.112703    0.033349  -3.380  0.000726 ***
## Age.Category2:CR.composite.before -0.006726    0.018507  -0.363  0.716267
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) Ag.Ct1 Ag.Ct2 CR.cm. GnCgP. A.C1:C
## Age.Catgry1 -0.031
## Age.Catgry2 -0.047 -0.129
## CR.cmpst.bf  0.033 -0.109  0.002
## GnCgPrc.cmp -0.007 -0.320  0.492  0.064
## Ag.Ct1:CR.. -0.060  0.089  0.001 -0.083 -0.118
## Ag.Ct2:CR.. -0.019  0.073 -0.003 -0.051 -0.035  0.058
```

```
# Look at pairwise comparisons between contrasts
esm.act <- entrends(Mfull_PNactAcc, ~Age.Category, var = "CR.composite.before")
pairs(esm.act)
```

```
## contrast          estimate      SE  df z.ratio p.value
## (Middle-Aged) - Younger    0.2254 0.0667 Inf   3.380  0.0021
## (Middle-Aged) - Older     0.1329 0.0664 Inf   2.001  0.1120
## Younger - Older          -0.0925 0.0631 Inf  -1.467  0.3070
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

Checking Assumptions GLMER Actions *The chosen link function is appropriate*

```
## [1] 2765.201
```

```
## [1] 2765.244
```

We chose for the cloglog link as it's a better link for highly skewed binomial distributions.

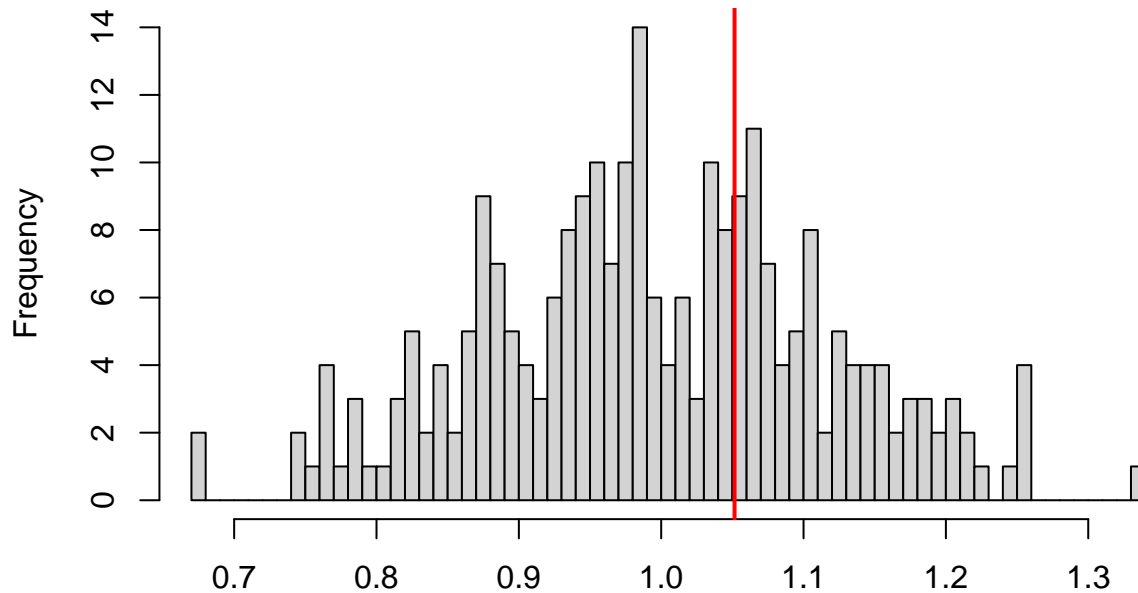
Appropriate estimation of variance (i.e. no over- or underdispersion)

Non-significant p-value so it seems to be ok to approach with this model.

Using a more formal test with the DHARMA package, which is based on simulations:

```
sim_Mfull_PNactAcc <- simulateResiduals(fittedModel = Mfull_PNactAcc)
testDispersion(simulationOutput = sim_Mfull_PNactAcc, alternative = "greater",
  plot = TRUE) # 'less' to indicate we're testing for underdispersion
```

DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated



Simulated values, red line = fitted model. p-value (greater) = 0.332

```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 1.0564, p-value = 0.332
## alternative hypothesis: greater
```

According to this method, there is no under- or overdispersion (value close to 1; 1.0522). In general, glmers with binomial data seem to be under-/overdispersed often but this model fit seems to be ok. (non-significant p-value)

Checking variance explained by random factors:

```
0.0246/(0.0246 + 0.1348)
```

```
## [1] 0.1543287
```

```
# ~15.4% of variance that's left over after the variance
# explained by our predictor variables is explained by ID
# (i.e. subject)
```

```
0.1348/(0.0246 + 0.1348)
```

```
## [1] 0.8456713
```



```
# ~84.6% of variance that's left over after the variance
# explained by our predictor variables is explained by
# trial (i.e. stimuli)
```

Model comparison Acc action naming

```
# Quick overview full model outcome
broom.mixed::glance(M0_PNactAcc)
```

```
## # A tibble: 1 x 6
##   sigma logLik   AIC   BIC deviance df.residual
##   <dbl>  <dbl> <dbl> <dbl>   <dbl>      <int>
## 1      1 -1383. 2771. 2792.   2468.        7041
```

```
broom.mixed::glance(Mfull_PNactAcc)
```

```
## # A tibble: 1 x 6
##   sigma logLik   AIC   BIC deviance df.residual
##   <dbl>  <dbl> <dbl> <dbl>   <dbl>      <int>
## 1      1 -1374. 2765. 2827.   2476.        7035
```

```
# AIC null model = 2771.056; AIC full model = 2768.112 -->
# the full model fits the data slightly better than the
# null model
```

```
# Tidy model summary
(tidyMfull_PNactAcc <- broom.mixed::tidy(Mfull_PNactAcc, effects = "fixed",
  conf.int = T, conf.level = 0.95) %>%
  mutate_if(is.numeric, round, 3))
```

```
## # A tibble: 7 x 8
##   effect term          estimate std.error statistic p.value conf.low conf.high
##   <chr>  <chr>          <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>
## 1 fixed (Intercept)      1.25     0.053     23.8     0        1.15     1.36
## 2 fixed Age.Category1    -0.037    0.035     -1.08    0.279    -0.105    0.03
## 3 fixed Age.Category2    -0.053    0.021     -2.53    0.011    -0.094   -0.012
## 4 fixed CR.composite.b~  0.023    0.027     0.864    0.387    -0.029    0.076
## 5 fixed GenCogProc.com~ -0.021    0.059    -0.361    0.718    -0.138    0.095
## 6 fixed Age.Category1::~ -0.113    0.033     -3.38    0.001    -0.178   -0.047
## 7 fixed Age.Category2::~ -0.007    0.019     -0.363    0.716    -0.043    0.03
```

```
## Write tidy table to .csv file
## write.csv(tidyMfull_PNactAcc, '../Figures and
## Tables/PNactAcc_glmFull.csv', row.names = F)
```

Concordance and Somer's D to assess predictive performance of the model

```
# Calculate probability of fitted effects full model
probs.PNactAcc = binomial()$linkinv(fitted(Mfull_PNactAcc))
# Calculate Concordance Somer's D
somers2(probs.PNactAcc, as.numeric(PNactions_Acc_coded$Acc))
```

```
##           C           Dxy           n           Missing
##    0.824284    0.648568 7044.000000    0.000000
```

Concordance = .824 → When C takes the value 0.5, the predictions are random, when it is 1, prediction is perfect. A value above 0.8 indicates that the model may have some real predictive capacity (Baayen 2008, 204). Somer's D = .649 → Predicted probabilities and observed responses ranges between 0 (randomness) and 1 (perfect prediction). The value should be higher than .5 for the model to be meaningful (Baayen, 2008; p.204)

The larger the values, the better the predictive performance of the model. The values for Concordance and Somer's D seem to indicate that the model has some meaningful predictive power.

Effect Size - R squared

```
MuMin::r.squaredGLMM(object = Mfull_PNactAcc, null = Mrandom.All_act)
```

```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.
```

```
##           R2m           R2c
## theoretical 0.008163226 0.09579715
## delta      0.005661864 0.06644316
```

Marginal R2(R2m) is the variance explained by fixed effects. Conditional R2 (R2c) is the variance explained by the whole model. Theoretical is for binomial distributions.

0.82% of the variance in the data is explained by the fixed effects only. 9.6% of the variance is explained by the whole model.

From: <https://www.investopedia.com/terms/r/r-squared.asp> R-squared will give you an estimate of the relationship between movements of a dependent variable based on an independent variable's movements. It doesn't tell you whether your chosen model is good or bad, nor will it tell you whether the data and predictions are biased. A high or low R-square isn't necessarily good or bad, as it doesn't convey the reliability of the model, nor whether you've chosen the right regression. You can get a low R-squared for a good model, or a high R-square for a poorly fitted model, and vice versa.

Generalised Linear Mixed Models - Object naming

```
PNobjects_Acc <- PN_all %>%
  # Only include trials for object naming (i.e., type==2)
  dplyr::filter(type == 2)
# Check unique(PNobjects_Acc$Task.Name) #objects
```

Contrast coding

```
PNobjects_Acc <- mutate(PNobjects_Acc, Age.Category = factor(Age.Category,
  levels = c("Middle-Aged", "Younger", "Older")))
```

```
PNobjects_Acc_coded <- PNobjects_Acc
contrasts(PNobjects_Acc_coded$Age.Category) <- contr.helmert(3)
contrasts(PNobjects_Acc_coded$Age.Category)
```

```
##           [,1] [,2]
## Middle-Aged -1  -1
## Younger      1  -1
## Older        0   2
```

Should we include random effects for ID and Trial.Number?

```
# Base model with Accuracy as outcome variable. Family =
# binomial
Mbase_obj <- glm(Acc ~ 1, family = "binomial", data = PNobjects_Acc_coded)

# Base model with only ID/individual variability as random
# effect
Mrandom.ID_obj <- glmer(Acc ~ 1 + (1 | ID), family = binomial(link = "cloglog"),
  control = glmerControl(optimizer = "bobyqa"), data = PNobjects_Acc_coded)
# Base model with only Trial.Number/trial variability as
# random effect
Mrandom.Trial_obj <- glmer(Acc ~ 1 + (1 | Trial.Number), family = binomial(link = "cloglog"),
  control = glmerControl(optimizer = "bobyqa"), data = PNobjects_Acc_coded)
# Base model with both random effects
Mrandom.All_obj <- glmer(Acc ~ 1 + (1 | ID) + (1 | Trial.Number),
  family = binomial(link = "cloglog"), data = PNobjects_Acc_coded)
```

AIC base model

```
# Obtain AIC values for each model
(AIC.base_obj <- AIC(logLik(Mbase_obj)))
```

```
## [1] 1070.891
```

AIC - only ID as random effect

```
(AIC.reID_obj <- AIC(logLik(Mrandom.ID_obj)))
```

```
## [1] 1064.13
```

AIC - only trial as random effect

```
(AIC.reTrial_obj <- AIC(logLik(Mrandom.Trial_obj)))
```

```
## [1] 1019.809
```

AIC - both ID and trial as random effect

```
(AIC.reBoth_obj <- AIC(logLik(Mrandom.All_obj)))
```

```
## [1] 1012.498
```

The AIC for the model including both random effects is lowest → we justified inclusion of both Trial and Subject as random effects.

Null model of reaction times for object naming with random effects included

```
# Null model
M0_PNobjAcc <- lme4::glmer(Acc ~ 1 + (1 | ID) + (1 | Trial.Number),
  data = PNobjects_Acc_coded, family = binomial(link = "cloglog"))
summary(M0_PNobjAcc)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( cloglog )
## Formula: Acc ~ 1 + (1 | ID) + (1 | Trial.Number)
## Data: PNobjects_Acc_coded
##
##      AIC      BIC    logLik deviance df.resid
##  1012.5   1032.8   -503.2   1006.5     6322
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0839   0.0512   0.0762   0.1239   0.4385
##
## Random effects:
##  Groups      Name      Variance Std.Dev.
##  ID          (Intercept) 0.03639  0.1908
##  Trial.Number (Intercept) 0.09721  0.3118
## Number of obs: 6325, groups: ID, 88; Trial.Number, 70
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.63109    0.07063   23.09  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Unconditional model, i.e., the model without covariates/control measures

```
# Model without covariates
Muncond_PNobjAcc <- glmer(Acc ~ Age.Category * CR.composite.before +
  (1 | ID) + (1 | Trial.Number), data = PNobjects_Acc_coded,
  family = binomial(link = "cloglog"))
```

Full model, i.e., model with covariates/control measures

```
Mfull_PNobjAcc <- lme4::glmer(Acc ~ Age.Category * CR.composite.before +
  GenCogProc.composite + (1 | ID) + (1 | Trial.Number), data = PNobjects_Acc_coded,
  family = binomial(link = "cloglog"))
summary(Mfull_PNobjAcc)
```

```

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( cloglog )
## Formula: Acc ~ Age.Category * CR.composite.before + GenCogProc.composite +
## (1 | ID) + (1 | Trial.Number)
## Data: PObjects_Acc_coded
##
##      AIC      BIC    logLik deviance df.resid
## 1006.7   1067.5   -494.3    988.7     6316
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -13.0139   0.0477   0.0812   0.1277   0.4231
##
## Random effects:
## Groups      Name                Variance Std.Dev.
## ID          (Intercept) 0.02100   0.1449
## Trial.Number (Intercept) 0.09512   0.3084
## Number of obs: 6325, groups: ID, 88; Trial.Number, 70
##
## Fixed effects:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        1.63312     0.06867  23.781 < 2e-16 ***
## Age.Category1                      0.03110     0.04767   0.652  0.51414
## Age.Category2                     -0.07386     0.02595  -2.846  0.00443 **
## CR.composite.before                -0.01291     0.03359  -0.384  0.70077
## GenCogProc.composite                0.04670     0.07255   0.644  0.51976
## Age.Category1:CR.composite.before -0.10568     0.04464  -2.368  0.01791 *
## Age.Category2:CR.composite.before  0.01107     0.02201   0.503  0.61511
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr) Ag.Ct1 Ag.Ct2 CR.cm. GnCgP. A.C1:C
## Age.Catgry1  0.045
## Age.Catgry2 -0.165 -0.227
## CR.cmpst.bf -0.047 -0.178  0.057
## GnCgPrc.cmp  0.029 -0.344  0.485  0.061
## Ag.Ct1:CR.. -0.167  0.001  0.064  0.095 -0.127
## Ag.Ct2:CR..  0.046  0.141 -0.056 -0.220 -0.051 -0.075

# Look at pairwise comparisons between contrasts
esm.obj <- emtrends(Mfull_PNobjAcc, ~Age.Category, var = "CR.composite.before")
pairs(esm.obj)

## contrast      estimate      SE df z.ratio p.value
## (Middle-Aged) - Younger    0.2114 0.0893 Inf   2.368  0.0471
## (Middle-Aged) - Older     0.0725 0.0769 Inf   0.943  0.6132
## Younger - Older          -0.1389 0.0824 Inf  -1.685  0.2108
##
## P value adjustment: tukey method for comparing a family of 3 estimates

```

Checking Assumptions GLMER

The chosen link function is appropriate

```
## [1] 1006.697
```

```
## [1] 1001.556
```

The logit link resulted in convergence errors. Hence, we chose for the cloglog link as it resulted in a better model fit + it's a better link for highly skewed binomial distributions.

Appropriate estimation of variance

(i.e. no over- or underdispersion)

```
# Function to calculate overdispersion of residuals
overdisp_fun <- function(model) {
  rdf <- df.residual(model)
  rp <- residuals(model, type = "pearson")
  Pearson.chisq <- sum(rp^2)
  prat <- Pearson.chisq/rdf
  pval <- pchisq(Pearson.chisq, df = rdf, lower.tail = FALSE)
  c(chisq = Pearson.chisq, ratio = prat, rdf = rdf, p = pval)
}

overdisp_fun(Mfull_PNobjAcc)
```

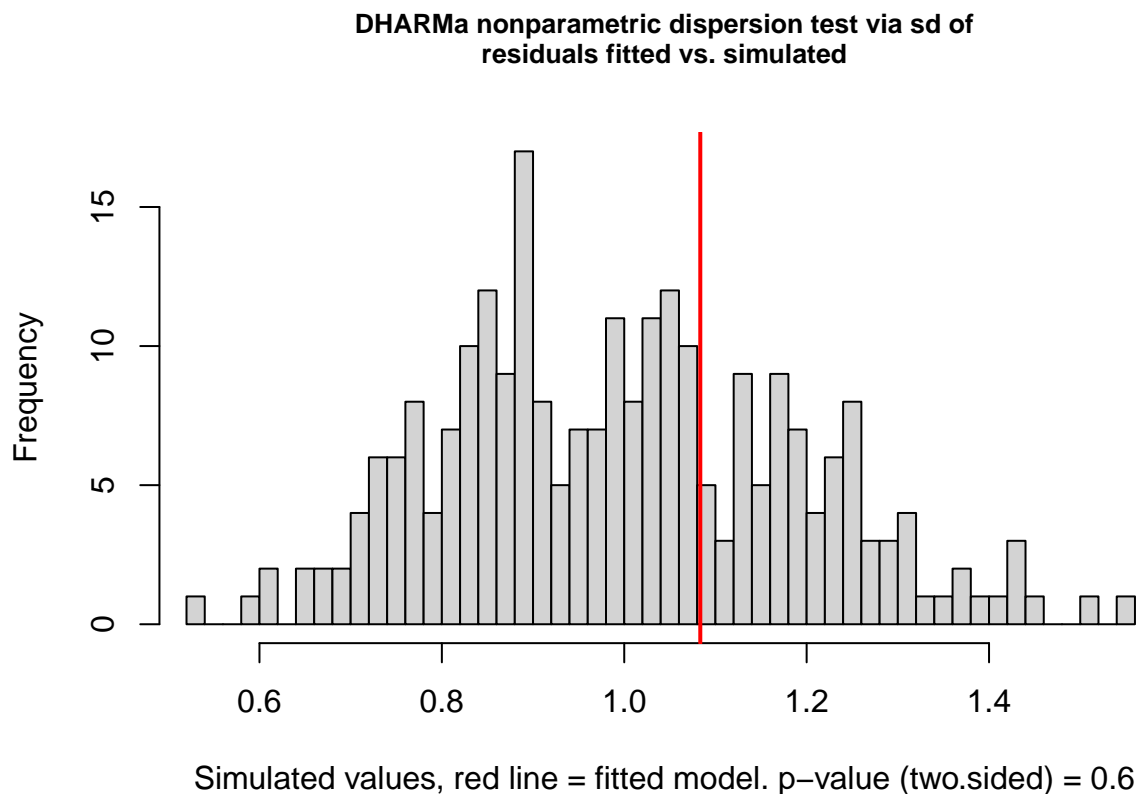
```
##          chisq          ratio          rdf          p
## 3518.2192967    0.5570328 6316.0000000    1.0000000
```

P value is non-significant so we don't assume overdispersion

Using a more formal test with the DHARMA package, which is based on simulations:

```
sim_Mfull_PNobjAcc <- simulateResiduals(fittedModel = Mfull_PNobjAcc)

testDispersion(simulationOutput = sim_Mfull_PNobjAcc, plot = TRUE) #two sided (both under- and over di.
```



```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 1.0871, p-value = 0.6
## alternative hypothesis: two.sided
```

According to this method, there is no over- or underdispersion (value close to 1; 1.0743). This model fit seems to be ok.

Checking variance explained by random factors:

```
0.01453/(0.01453 + 0.09485) #~13.3% of variance that's left over after the variance explained by our p
```

```
## [1] 0.1328396
```

```
0.09485/(0.01453 + 0.09485) #~86.7% of variance that's left over after the variance explained by our p
```

```
## [1] 0.8671604
```

Model comparison Acc object naming

```
# Quick overview full model outcome
broom.mixed::glance(M0_PNobjAcc) #AIC null model = 1012.498
```

```
## # A tibble: 1 x 6
##   sigma logLik   AIC   BIC deviance df.residual
##   <dbl> <dbl> <dbl> <dbl>   <dbl>      <int>
## 1     1    -503. 1012. 1033.    837.        6322
```

```
broom.mixed::glance(Mfull_PNobjAcc) #AIC full model = 1006.697
```

```
## # A tibble: 1 x 6
##   sigma logLik   AIC   BIC deviance df.residual
##   <dbl> <dbl> <dbl> <dbl>   <dbl>      <int>
## 1     1   -494. 1007. 1067.    843.        6316
```

```
# The full model fits the data better than the null model
```

```
# Tidy model summary
(tidyMfull_PNobjAcc <- broom.mixed::tidy(Mfull_PNobjAcc, effects = "fixed",
  conf.int = T, conf.level = 0.95) %>%
  mutate_if(is.numeric, round, 3))
```

```
## # A tibble: 7 x 8
##   effect term          estimate std.error statistic p.value conf.low conf.high
##   <chr> <chr>          <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>
## 1 fixed (Intercept)      1.63     0.069     23.8     0       1.50     1.77
## 2 fixed Age.Category1      0.031    0.048     0.652  0.514    -0.062    0.125
## 3 fixed Age.Category2     -0.074    0.026    -2.85   0.004    -0.125   -0.023
## 4 fixed CR.composite.b~   -0.013    0.034    -0.384  0.701    -0.079    0.053
## 5 fixed GenCogProc.com~    0.047    0.073     0.644  0.52     -0.095    0.189
## 6 fixed Age.Category1::~ -0.106    0.045    -2.37   0.018    -0.193   -0.018
## 7 fixed Age.Category2::~  0.011    0.022     0.503  0.615    -0.032    0.054
```

```
## Write tidy table to .csv file
## write.csv(tidyMfull2_PNobjAcc, '../Figures and
## Tables/PNobjAcc_glmFull.csv', row.names = F)
```

Concordance and Somer's D to assess predictive performance of the model

```
# Calculate probability of fitted effects full model
probs.PNobjAcc = binomial()$linkinv(fitted(Mfull_PNobjAcc))
# Calculate Concordance Somer's D
somers2(probs.PNobjAcc, as.numeric(PNobjects_Acc_coded$Acc))
```

```
##           C           Dxy           n           Missing
## 0.8896884 0.7793768 6325.0000000 0.0000000
```

Concordance = .890 → When C takes the value 0.5, the predictions are random, when it is 1, prediction is perfect. A value above 0.8 indicates that the model may have some real predictive capacity (Baayen 2008, 204). Somer's D = .779 → Predicted probabilities and observed responses ranges between 0 (randomness)

and 1 (perfect prediction). The value should be higher than .5 for the model to be meaningful (Baayen, 2008; p.204)

The larger the values, the better the predictive performance of the model.

Effect Size - R squared

```
MuMIn::r.squaredGLMM(object = Mfull_PNobjAcc, null = Mrandom.All_obj)
```

```
##                R2m          R2c
## theoretical 0.012432630 0.07755089
## delta      0.003916285 0.02442857
```

Marginal R2 is the variance explained by fixed effects. Conditional R2 is the variance explained by the whole model. Theoretical is for binomial distributions.

1.24% of the variance in the data is explained by the fixed effects only. 7.8% of the variance is explained by the whole model.

Visualising significant predictors

```
# Create labels for legend per age group
Ages = as_labeller(c(Younger = "Younger Adults", `Middle-Aged` = "Middle-Aged Adults",
  Older = "Older Adults"))
```

```
# Colour Palette - colourblind friendly
cbbPalette <- c("#999999", "#E69F00", "#56B4E9")
windowsFonts("Times New Roman")
```

```
## $<NA>
## NULL
```

```
# Create means per participant for Acc and CR Action Naming
meanAcc_act <- PNactions_Acc_coded %>%
  dplyr::group_by(Age.Category, ID) %>%
  summarise(meanAcc = mean(Acc, na.rm = T), CR = mean(CR.composite.before)) %>%
  mutate(Age.Category = fct_relevel(Age.Category, c("Younger",
    "Middle-Aged", "Older")))

# meanAcc_act <-
# relevel(meanAcc_act$Age.Category, 'Younger')

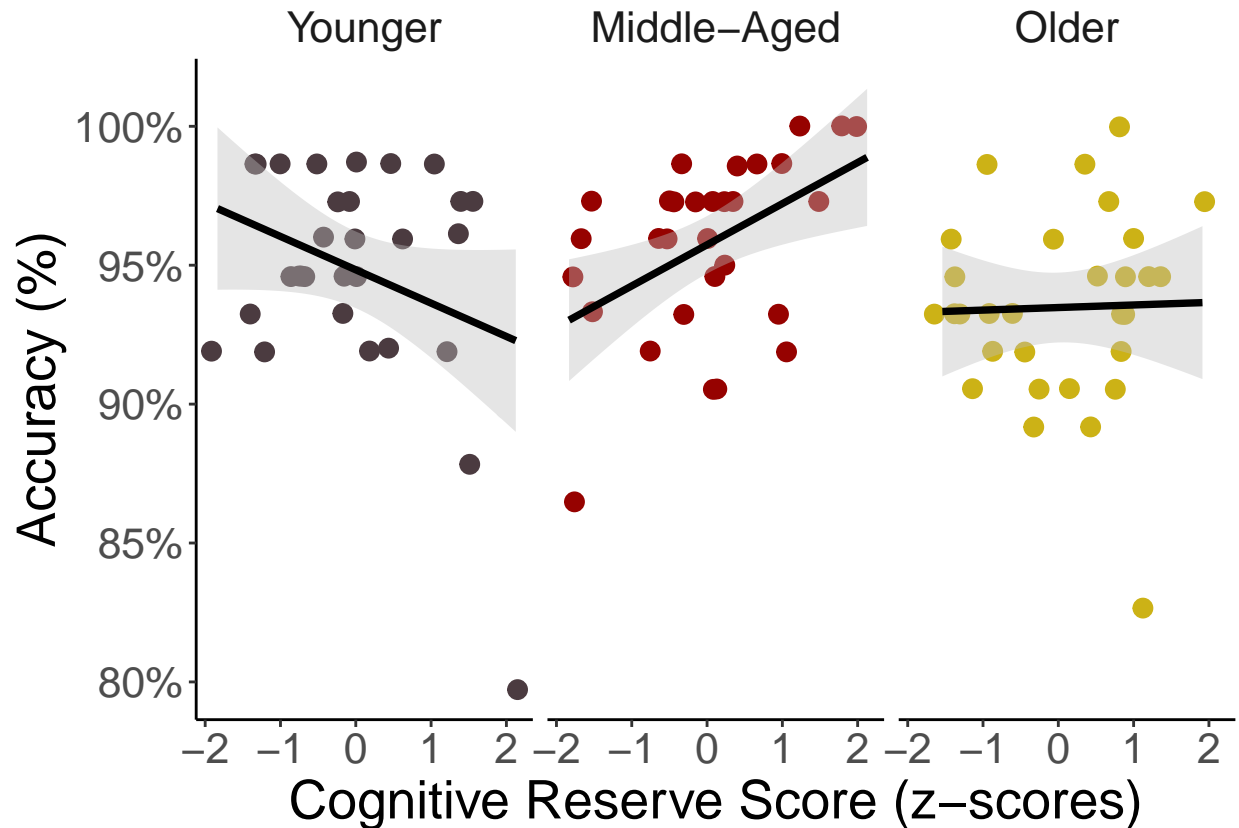
# Save figure as tiff file tiff(file='../Figures and
# Tables/RelationCR-PNactAcc.tiff', res = 500, family =
# 'sans', width = 12, height=4.5, units='in')

# Relationship Acc and CR pre-pandemic of picture naming
# Actions
(plot.PNactACC_CR <- meanAcc_act %>%
  ggplot(aes(x = CR, y = meanAcc, colour = as.factor(Age.Category))) +
  geom_jitter(width = 0.25, size = 3, show.legend = F) + geom_smooth(method = "glm",
```

```

formula = y ~ x, fill = "grey", colour = "black", show.legend = F,
size = 1.3) + labs(x = "Cognitive Reserve Score (z-scores)",
y = "Accuracy (%)") + scale_x_continuous(breaks = seq(-3,
3, 1)) + scale_y_continuous(labels = scales::label_percent(accuracy = 1),
minor_breaks = seq(0.8, 1, 0.05), breaks = seq(0.8, 1, 0.05)) +
facet_grid(~Age.Category) + theme(text = element_text(size = 20),
panel.background = element_rect(fill = "white"), plot.background = element_rect(fill = "white"),
strip.background = element_rect(fill = "white"), axis.line.x = element_line(color = "black"),
axis.line.y = element_line(color = "black")) + scale_colour_manual(values = confPalette))

```



```
# dev.off()
```

```

# Create means per participant for Acc and CR Object Naming
meanAcc_obj <- PNOjects_Acc %>%
  dplyr::group_by(Age.Category, ID) %>%
  dplyr::summarise(meanAcc = mean(Acc, na.rm = T), CR = mean(CR.composite.before)) %>%
  mutate(Age.Category = fct_relevel(Age.Category, c("Younger",
    "Middle-Aged", "Older")))

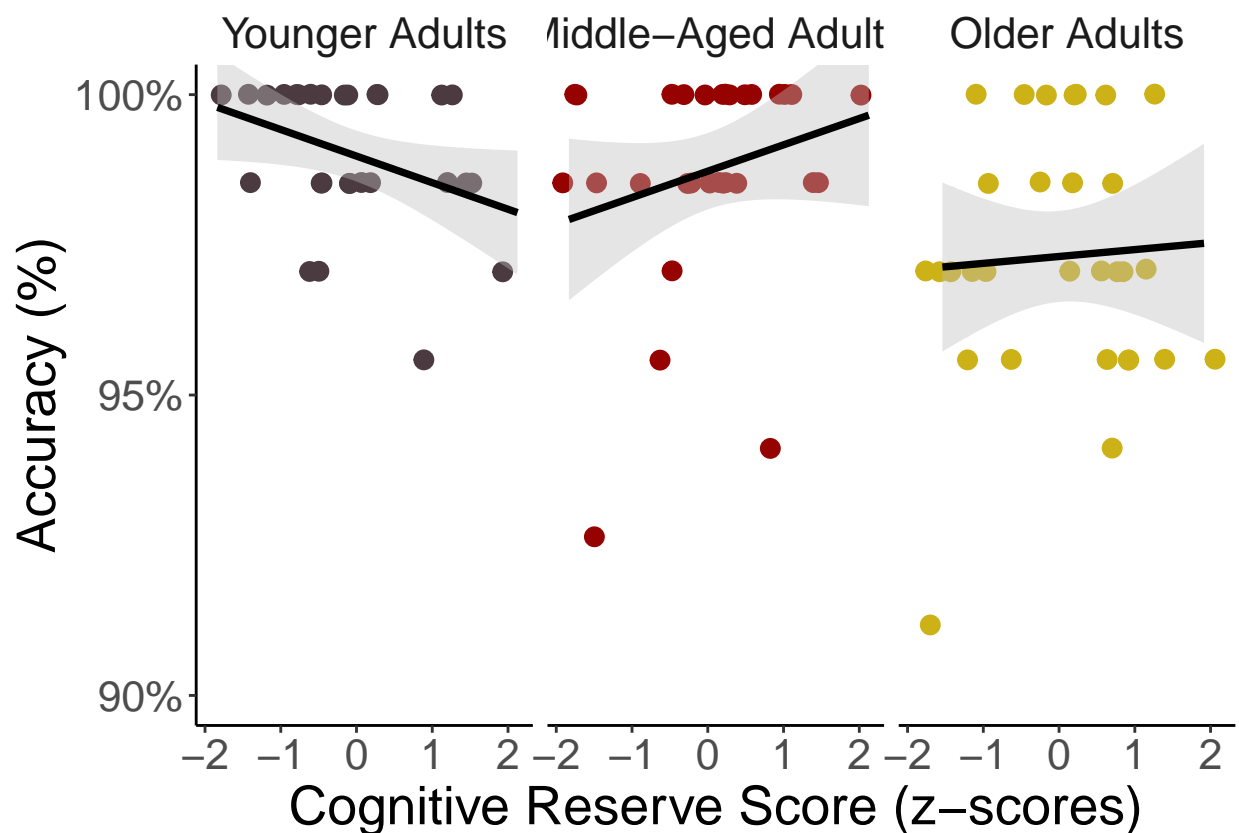
# tiff(file='../Figures and
# Tables/RelationCR-PNobjAcc.tiff', res = 500, family =
# 'sans', width = 12, height=4.5, units='in')

# Relationship Acc and CR pre-pandemic of picture naming

```

```
# Actions
```

```
(plot.PNobjACC_CR <- meanAcc_obj %>%
  ggplot(aes(x = CR, y = meanAcc, colour = as.factor(Age.Category))) +
  geom_jitter(width = 0.25, size = 3, show.legend = F) + geom_smooth(method = "glm",
  formula = y ~ x, fill = "grey", colour = "black", show.legend = F,
  size = 1.3) + labs(x = "Cognitive Reserve Score (z-scores)",
  y = "Accuracy (%)") + coord_cartesian(ylim = c(0.9, 1)) +
  scale_x_continuous(breaks = seq(-3, 3, 1)) + scale_y_continuous(labels = scales::label_percent(accur
  minor_breaks = seq(0.8, 1, 0.05), breaks = seq(0.8, 1, 0.05)) +
  facet_grid(~Age.Category, labeller = labeller(Age.Category = Ages)) +
  theme(text = element_text(size = 20), panel.background = element_rect(fill = "white"),
  plot.background = element_rect(fill = "white"), strip.background = element_rect(fill = "white")
  axis.line.x = element_line(color = "black"), axis.line.y = element_line(color = "black")) +
  scale_colour_manual(values = confPalette))
```



```
# dev.off()
```

Model comparisons for the CR measure preceding and coinciding with the COVID-19 pandemic

The effect of the COVID-19 pandemic on CR and subsequent behavioural performance

```
# Action Naming
```

```
Mfull_PNactAcc.during <- lme4::glmer(Acc ~ Age.Category * CR.composite.during +
  GenCogProc.composite + (1 | ID) + (1 | Trial.Number), data = PNactions_Acc_coded,
  family = binomial(link = "cloglog"))
```

No convergence errors . To be able to compare the models better, we will restart the model as done with the model that has the CR composite before as predictor

The significant predictive effect of CR has disappeared. That is, the CR composite during Covid does not significantly predict Accuracy for Action naming

Comparing the models

```
anova(Mfull_PNactAcc, Mfull_PNactAcc.during)
```

```
## Data: PNactions_Acc_coded
## Models:
## Mfull_PNactAcc: Acc ~ Age.Category * CR.composite.before + GenCogProc.composite +
## Mfull_PNactAcc:      (1 | ID) + (1 | Trial.Number)
## Mfull_PNactAcc.during: Acc ~ Age.Category * CR.composite.during + GenCogProc.composite +
## Mfull_PNactAcc.during:      (1 | ID) + (1 | Trial.Number)
##
##          npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## Mfull_PNactAcc      9 2765.2 2826.9 -1373.6   2747.2
## Mfull_PNactAcc.during 9 2774.1 2835.9 -1378.1   2756.1      0  0
```

There are barely any differences between the two models. The AIC is a tiny bit lower for the CR composite score before the pandemic, meaning that that model fits the data slightly better. This model is also the one with the significant effect of CR.

Object Naming

```
Mfull_PNobjAcc.during <- lme4::glmer(Acc ~ Age.Category * CR.composite.during +
  GenCogProc.composite + (1 | ID) + (1 | Trial.Number), data = PNobjects_Acc_coded,
  family = binomial(link = "cloglog")) #converged
```

```
anova(Mfull_PNobjAcc, Mfull_PNobjAcc.during)
```

```
## Data: PNobjects_Acc_coded
## Models:
## Mfull_PNobjAcc: Acc ~ Age.Category * CR.composite.before + GenCogProc.composite +
## Mfull_PNobjAcc:      (1 | ID) + (1 | Trial.Number)
## Mfull_PNobjAcc.during: Acc ~ Age.Category * CR.composite.during + GenCogProc.composite +
## Mfull_PNobjAcc.during:      (1 | ID) + (1 | Trial.Number)
##
##          npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## Mfull_PNobjAcc      9 1006.7 1067.5 -494.35   988.7
## Mfull_PNobjAcc.during 9 1010.2 1071.0 -496.10   992.2      0  0
```

There are barely any differences between the two models. The AIC is a bit lower for the CR composite score before the pandemic, meaning that that model fits the data slightly better. In the model with CR during, the interaction term between CR score and being a middle-aged adult has also disappeared.

References

- Baayen, Harald. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics Using r*. Cambridge: Cambridge University Press.
- Barton, Kamil. 2020. *MuMIn: Multi-Model Inference*. <https://CRAN.R-project.org/package=MuMIn>.

- Bates, Douglas, and Martin Maechler. 2019. *Matrix: Sparse and Dense Matrix Classes and Methods*. <http://Matrix.R-forge.R-project.org/>.
- Bates, Douglas, Martin Maechler, Ben Bolker, and Steven Walker. 2020. *Lme4: Linear Mixed-Effects Models Using Eigen and S4*. <https://github.com/lme4/lme4/>.
- Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. “Fitting Linear Mixed-Effects Models Using lme4.” *Journal of Statistical Software* 67 (1): 1–48. <https://doi.org/10.18637/jss.v067.i01>.
- Chan, Chung-hong, and Thomas J. Leeper. 2021. *Rio: A Swiss-Army Knife for Data i/o*. <https://github.com/leeper/rio>.
- Fernando, Jason. 2021. “What Is r-Squared?” *Investopedia*. Investopedia. <https://www.investopedia.com/terms/r/r-squared.asp>.
- Fox, John, and Sanford Weisberg. 2019. *An R Companion to Applied Regression*. Third. Thousand Oaks CA: Sage. <https://socialsciences.mcmaster.ca/jfox/Books/Companion/>.
- Fox, John, Sanford Weisberg, and Brad Price. 2020. *carData: Companion to Applied Regression Data Sets*. <https://CRAN.R-project.org/package=carData>.
- . 2021. *Car: Companion to Applied Regression*. <https://CRAN.R-project.org/package=car>.
- Harrell, Frank E, Jr. 2021. *Hmisc: Harrell Miscellaneous*. <https://CRAN.R-project.org/package=Hmisc>.
- Hartig, Florian. 2021. *DHARMa: Residual Diagnostics for Hierarchical (Multi-Level / Mixed) Regression Models*. <http://florianhartig.github.io/DHARMa/>.
- Henry, Lionel, and Hadley Wickham. 2020. *Purrr: Functional Programming Tools*. <https://CRAN.R-project.org/package=purrr>.
- Hester, Jim, and Hadley Wickham. 2020. *Fs: Cross-Platform File System Operations Based on Libuv*. <https://CRAN.R-project.org/package=fs>.
- Kuznetsova, Alexandra, Per B. Brockhoff, and Rune H. B. Christensen. 2017. “lmerTest Package: Tests in Linear Mixed Effects Models.” *Journal of Statistical Software* 82 (13): 1–26. <https://doi.org/10.18637/jss.v082.i13>.
- Kuznetsova, Alexandra, Per Bruun Brockhoff, and Rune Haubo Bojesen Christensen. 2020. *lmerTest: Tests in Linear Mixed Effects Models*. <https://github.com/runehaubo/lmerTestR>.
- Lenth, Russell V. 2021. *Emmeans: Estimated Marginal Means, Aka Least-Squares Means*. <https://github.com/rvleenth/emmeans>.
- Lüdtke, Daniel, Mattan S. Ben-Shachar, Indrajeet Patil, Philip Waggoner, and Dominique Makowski. 2021. “performance: An R Package for Assessment, Comparison and Testing of Statistical Models.” *Journal of Open Source Software* 6 (60): 3139. <https://doi.org/10.21105/joss.03139>.
- Lüdtke, Daniel, Dominique Makowski, Mattan S. Ben-Shachar, Indrajeet Patil, Philip Waggoner, and Brenton M. Wiernik. 2021. *Performance: Assessment of Regression Models Performance*. <https://easystats.github.io/performance/>.
- Meyer, David, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. 2021. *E1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. <https://CRAN.R-project.org/package=e1071>.
- Müller, Kirill, and Hadley Wickham. 2021. *Tibble: Simple Data Frames*. <https://CRAN.R-project.org/package=tibble>.
- R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.
- Robinson, David, Alex Hayes, and Simon Couch. 2021. *Broom: Convert Statistical Objects into Tidy Tibbles*. <https://CRAN.R-project.org/package=broom>.

- Sarkar, Deepayan. 2008. *Lattice: Multivariate Data Visualization with r*. New York: Springer. <http://lmdvr.r-forge.r-project.org>.
- . 2021. *Lattice: Trellis Graphics for r*. <http://lattice.r-forge.r-project.org/>.
- Sjoberg, David. 2020. *Hablar: Non-Astonishing Results in r*. <https://davidsjoberg.github.io/>.
- Terry M. Therneau, and Patricia M. Grambsch. 2000. *Modeling Survival Data: Extending the Cox Model*. New York: Springer.
- Therneau, Terry M. 2021. *Survival: Survival Analysis*. <https://github.com/therneau/survival>.
- Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>.
- . 2019. *Stringr: Simple, Consistent Wrappers for Common String Operations*. <https://CRAN.R-project.org/package=stringr>.
- . 2021a. *Forcats: Tools for Working with Categorical Variables (Factors)*. <https://CRAN.R-project.org/package=forcats>.
- . 2021b. *Tidyr: Tidy Messy Data*. <https://CRAN.R-project.org/package=tidyr>.
- . 2021c. *Tidyverse: Easily Install and Load the Tidyverse*. <https://CRAN.R-project.org/package=tidyverse>.
- Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D’Agostino McGowan, Romain François, Garrett Grolemond, et al. 2019. “Welcome to the tidyverse.” *Journal of Open Source Software* 4 (43): 1686. <https://doi.org/10.21105/joss.01686>.
- Wickham, Hadley, and Jennifer Bryan. 2019. *Readxl: Read Excel Files*. <https://CRAN.R-project.org/package=readxl>.
- Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2021. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. <https://CRAN.R-project.org/package=ggplot2>.
- Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. <https://CRAN.R-project.org/package=dplyr>.
- Wickham, Hadley, and Jim Hester. 2021. *Readr: Read Rectangular Text Data*. <https://CRAN.R-project.org/package=readr>.
- Xie, Yihui. 2014. “Knitr: A Comprehensive Tool for Reproducible Research in R.” In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. <http://www.crcpress.com/product/isbn/9781466561595>.
- . 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. <https://yihui.org/knitr/>.
- . 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. <https://yihui.org/knitr/>.
- . 2022. *formatR: Format r Code Automatically*. <https://github.com/yihui/formatR>.
- Zeileis, Achim, and Yves Croissant. 2010. “Extended Model Formulas in R: Multiple Parts and Multiple Responses.” *Journal of Statistical Software* 34 (1): 1–13. <https://doi.org/10.18637/jss.v034.i01>.
- . 2020. *Formula: Extended Model Formulas*. <https://CRAN.R-project.org/package=Formula>.