R Code Full Analysis Verbal Fluency Number of Correctly Produced Words

Elise Oosterhuis

Last compiled on 03/08/22

Analysis Verbal Fluency Data - Number of Correctly Produced Words

```
## Semantic Fluency
VFcat <- read.csv("../Data/Tidy/VFcat_complete_final.csv")
# head(VFcat[1:6,1:11]) #shows top 6 rows and the 1st to
# 11th column.  tail(VFcat[1:6,1:11]) #shows bottom 6 rows
# and the 1st to 11th column.

## Letter Fluency
VFlet <- read.csv("../Data/Tidy/VFlet_complete_final.csv")
# head(VFlet[1:6,1:9]) tail(VFlet[1:6,1:9])

## Action Fluency
VFact <- read.csv("../Data/Tidy/VFact_complete_final.csv")
# head(VFact[1:6,1:8]) tail(VFact[1:6,1:8])
```

Read in data

Verbal Fluency - Semantic/Categories

Descriptive Statistics

```
## Create dataset that only includes data of the Number of Correct words Produced
VFcat_Ncorrect <- VFcat %>%
  dplyr::filter(Measures=="Ncorrect") %>%
  dplyr::select(-Measures) %>%
      #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"))) %>%
  #Use the z scores to filter out outliers (i.e., exclude values +/-2.5 SD per trial )
  filter(between(zComp.Cat, -2.5, +2.5)) %>% #No outliers
  ungroup()
```

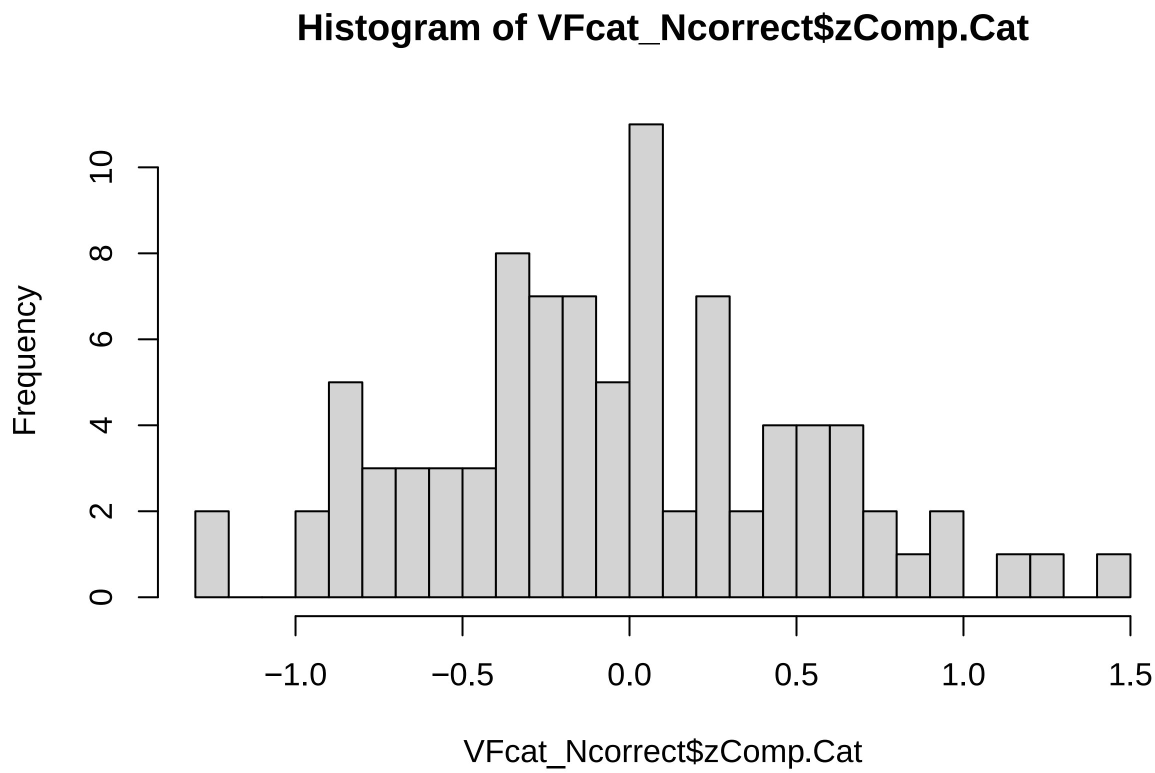
```r
# head(VFcat_Ncorrect_coded, L=6)
```

```r
#Summarise data to present descriptives in a table
(Descr_VFcat <- VFcat_Ncorrect %>%
#Per Age Group
  group_by(Age.Category) %>%
  summarise(Nppt = length(unique(ID)), #Number of participants
            total = round(mean(Total, na.rm=T),2),
            sdtotal = round(sd(Total, na.rm=T),2),#Total words correctly produced
            # ztotal = mean(zComp.Cat, na.rm=T), #Z-score per age group of total words correctly produc
            animals = round(mean(Animals, na.rm=T),2), #Total words correctly produced in category Anim
            vehicles = round(mean(Vehicles, na.rm=T),2), #Total words correctly produced in category ve
            vandF = round(mean(Fruits.and.Vegetables, na.rm=T),2), #Total words correctly produced in c
            fluid = round(mean(Fluid, na.rm=T),2), #Total words correctly produced in category fluid
            writing = round(mean(Writing.Utensils, na.rm=T),2))) #Total words correctly produced in cat
```

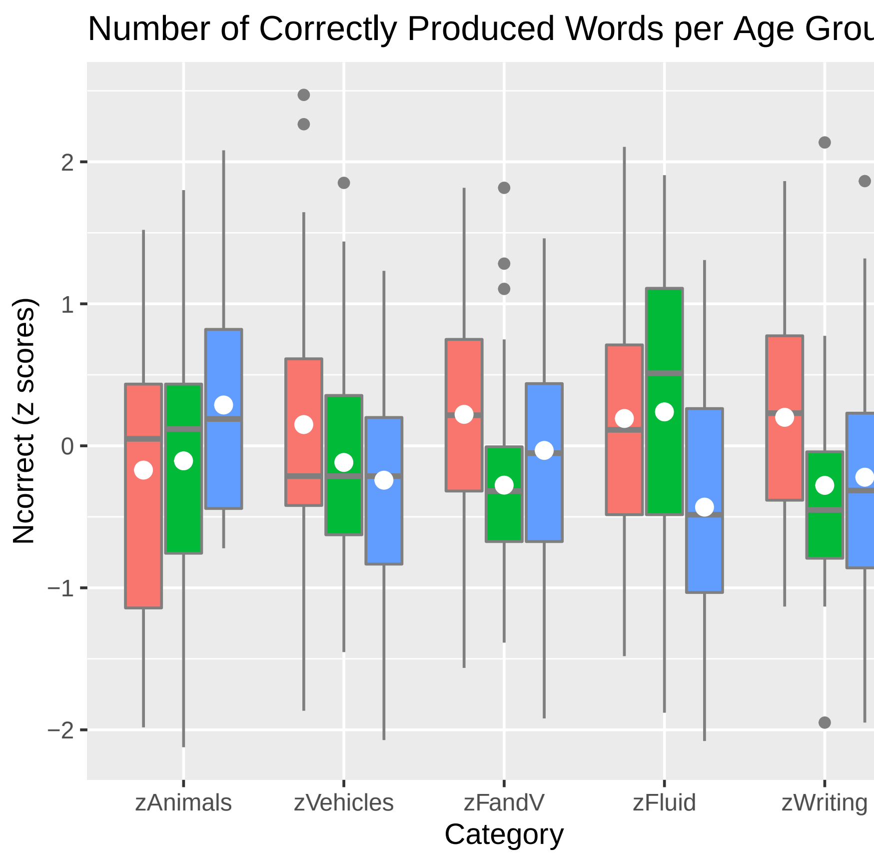
```
## # A tibble: 3 x 9
##   Age.Category  Nppt total sdtotal animals vehicles vandF fluid writing
##   <fct>        <int> <dbl>   <dbl>   <dbl>    <dbl> <dbl> <dbl>   <dbl>
## 1 Middle-Aged     30  93.5    18.8    22.8     17.2  26.6  14.4    12.5
## 2 Older           30  85.3    13.9    23.4     15.5  23.2  14.6    10.1
## 3 Younger         30  88.2    19.2    26.2     15.4  24.5  11.3    10.8
```

```r
#Save table
# write.csv(Descr_VFcat, "./Figures and Tables/Descr_VFcat_Ncorrect_coded.csv", row.names = F)
```

```r
# z-distribution of composite score for semantic fluency
hist(VFcat_Ncorrect$zComp.Cat, breaks = 20)  #Composite z-score of Semantic Fluency
```



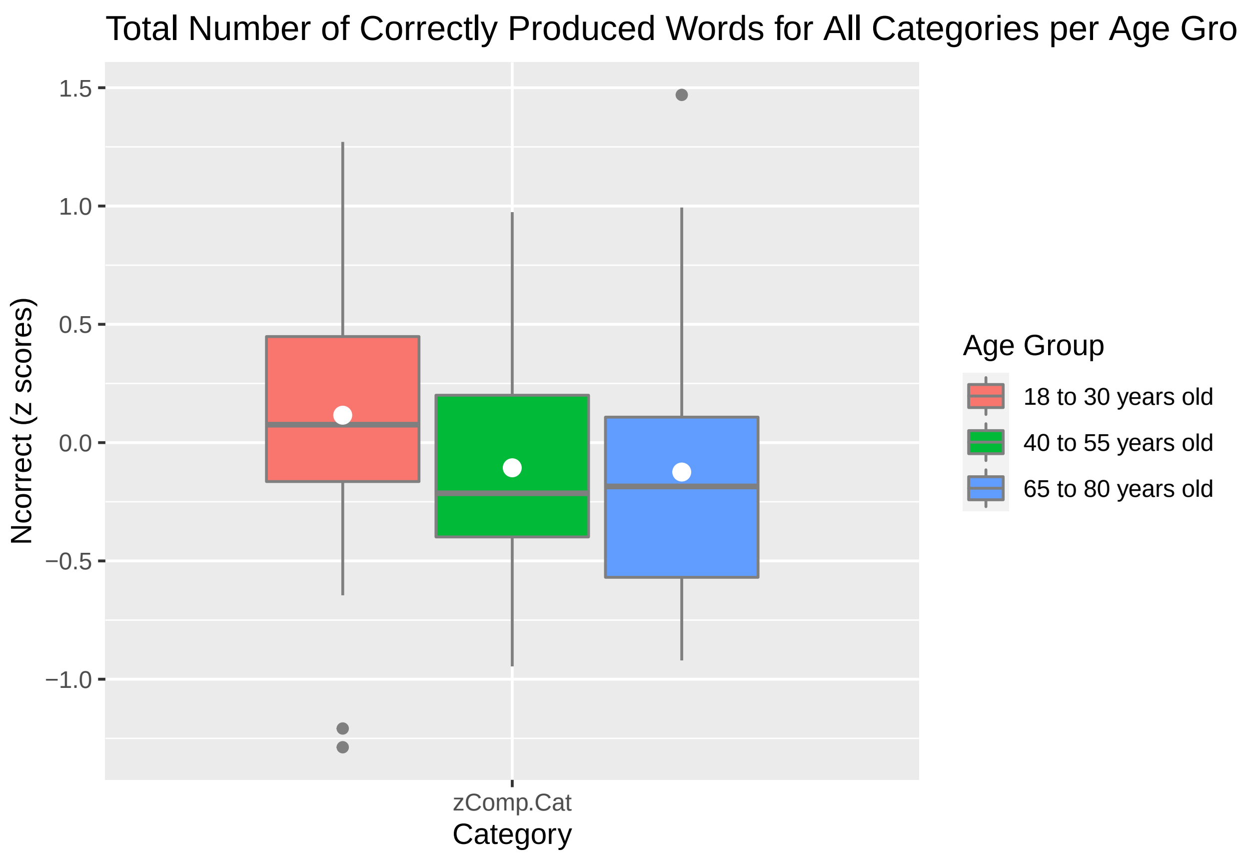

```
## Convert wide to long format for visualisation of data
VFcat_Ncorrect.longz.scores <- VFcat_Ncorrect %>%
  pivot_longer(cols=zComp.Cat:zWriting, names_to = "zcategory", values_to = "zNcorrect")

# Boxplot VF cat Ncorrect per category
# png(file="./Figures and Tables/Boxplot_VFcatNcorrect_zscores.png",
# width=600, height=350) #writes boxplot below to a .png file

(Boxplot_VF <- VFcat_Ncorrect.longz.scores %>%
    dplyr::filter(zcategory!="zComp.Cat") %>%
    ggplot(aes(x=factor(zcategory, levels=c("zAnimals", "zVehicles", "zFandV", "zFluid", "zWriting")), y
                                               fill = as.factor(Age.Category))) +
    geom_boxplot(colour="grey50")+
      stat_summary(fun = "mean", position = position_dodge(.75),
               show.legend=F, colour="white")+ #Mean as white dot
    labs(x = "Category",
         y = "Ncorrect (z scores)",
         title = "Number of Correctly Produced Words per Age Group and Category")+
        scale_fill_discrete(guide=guide_legend(title = "Age Group"), labels=c("18 to 30 years old","40 t
```



Visualisation Ncorrect Semantic Fluency

```
# dev.off() #close png() function
```

```
# Boxplot VF cat N correct for total score
# png(file="./Figures and Tables/Boxplot_VFcatNcorrect_RawTotal_zscores.png",
# width=600, height=350)

(Boxplot_VF <- VFcat_Ncorrect.longz.scores %>%
    dplyr::filter(zcategory=="zComp.Cat") %>%
    ggplot(aes(x=zcategory, y=zNcorrect,fill = as.factor(Age.Category))) +
    geom_boxplot(colour="grey50")+
      stat_summary(fun = "mean", position = position_dodge(.75),
               show.legend=F, colour="white")+ #Mean as white dot
    labs(x = "Category",
         y = "Ncorrect (z scores)",
         title = "Total Number of Correctly Produced Words for All Categories per Age Group")+
        scale_fill_discrete(guide=guide_legend(title = "Age Group"), labels=c("18 to 30 years old","40 
```




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

Multiple Linear Regression - Semantic Fluency

Create user-defined contrasts for the Age Category variable

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

VFcat_Ncorrect_coded <- VFcat_Ncorrect
contrasts(VFcat_Ncorrect_coded$Age.Category) <- contr.helmert(3)
contrasts(VFcat_Ncorrect_coded$Age.Category)
```

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

Unconditional model, i.e. without covariates

```r
lmUncond.VFcat.Ncorrect <- lm(zComp.Cat ~ Age.Category * CR.composite.before,
    data = VFcat_Ncorrect_coded)
# broom::tidy(lmUncond.VFcat.Ncorrect, conf.int=T)
```

Full model including the covariates; outcome variable in z-distribution

```r
lmFull.VFcat.Ncorrect <- lm(zComp.Cat ~ Age.Category * CR.composite.before +
    GenCogProc.composite, data = VFcat_Ncorrect_coded)
summary(lmFull.VFcat.Ncorrect)
```

```
##
## Call:
## lm(formula = zComp.Cat ~ Age.Category * CR.composite.before +
##     GenCogProc.composite, data = VFcat_Ncorrect_coded)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31478 -0.32685 -0.04433  0.32026  1.64077
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -0.03807    0.05858  -0.650   0.5175
## Age.Category1                     -0.12893    0.07579  -1.701   0.0927 .
## Age.Category2                     -0.02548    0.04777  -0.533   0.5951
## CR.composite.before                0.03853    0.05966   0.646   0.5202
## GenCogProc.composite               0.04897    0.13588   0.360   0.7195
## Age.Category1:CR.composite.before -0.07573    0.07320  -1.035   0.3039
## Age.Category2:CR.composite.before  0.04072    0.04213   0.967   0.3365
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5557 on 83 degrees of freedom
## Multiple R-squared:  0.06702,    Adjusted R-squared:  -0.0004276
## F-statistic: 0.9937 on 6 and 83 DF,  p-value: 0.4351
```

```r
# Tidy table output
(tidylmFull.VFcat.Ncorrect <- broom::tidy(lmFull.VFcat.Ncorrect,
    conf.int = T) %>%
    mutate_if(is.numeric, round, 3))
```

```
## # A tibble: 7 x 7
##   term                    estimate std.error statistic p.value conf.low conf.high
##   <chr>                      <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)               -0.038     0.059    -0.65    0.518   -0.155     0.078
## 2 Age.Category1             -0.129     0.076    -1.70    0.093   -0.28      0.022
## 3 Age.Category2             -0.025     0.048    -0.533   0.595   -0.12      0.07
## 4 CR.composite.before        0.039     0.06      0.646   0.52    -0.08      0.157
## 5 GenCogProc.composite       0.049     0.136     0.36    0.719   -0.221     0.319
## 6 Age.Category1:CR.comp~    -0.076     0.073    -1.03    0.304   -0.221     0.07
## 7 Age.Category2:CR.comp~     0.041     0.042     0.967   0.337   -0.043     0.125
```

```r
# write.csv(tidylmFull.VFcat.Ncorrect, './Figures and
# tables/VFcat_zNcorrect_lmFull.csv', row.names = F) #write
# tidy output table to file
```

```
# Look at pairwise comparisons between contrasts
lmFull.VFcat.Ncorrect.emmeans <- emmeans::emtrends(lmFull.VFcat.Ncorrect,
    ~Age.Category, var = "CR.composite.before")
pairs(lmFull.VFcat.Ncorrect.emmeans)
```

```
##  contrast                estimate    SE df t.ratio p.value
##  (Middle-Aged) - Younger   0.1515 0.146 83   1.035  0.5573
##  (Middle-Aged) - Older    -0.0464 0.146 83  -0.318  0.9459
##  Younger - Older          -0.1979 0.146 83  -1.355  0.3690
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

Full model including the covariates; outcome variable as raw score

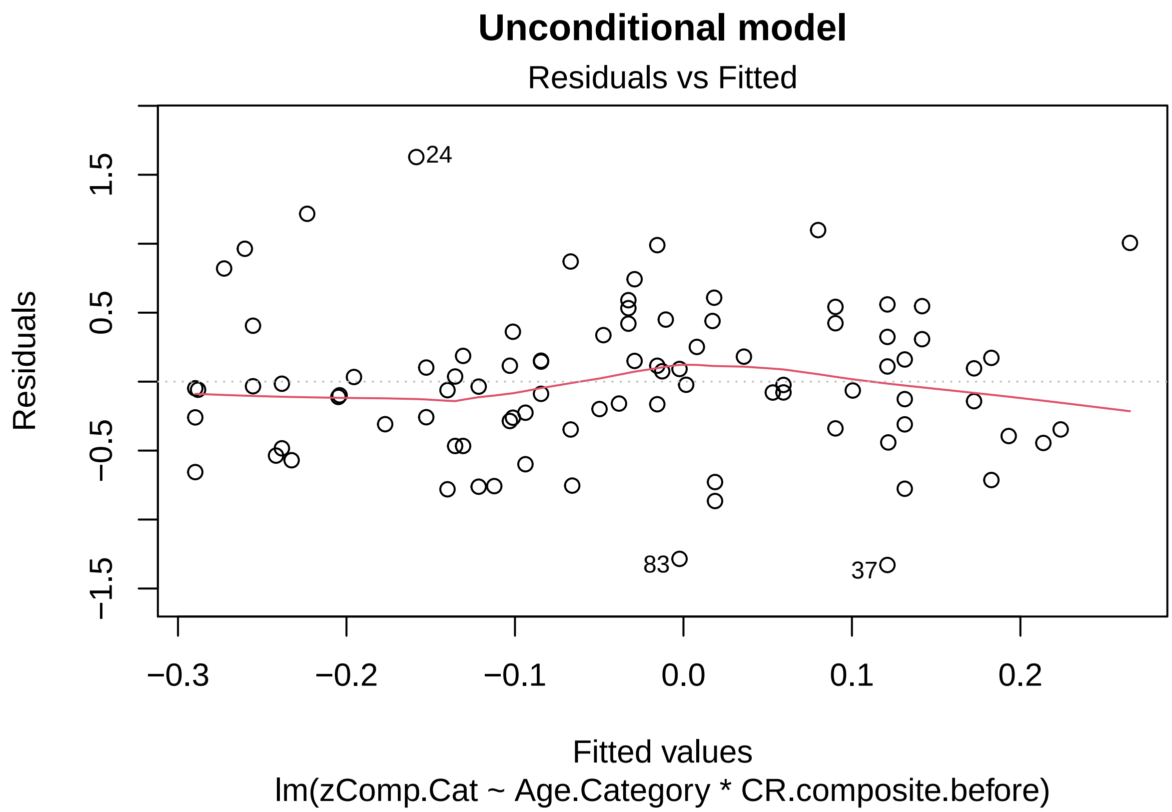
```
lmFull.VFcat.Ncorrect.raw <- lm(Total ~ Age.Category * CR.composite.before +
    GenCogProc.composite, data = VFcat_Ncorrect_coded)
# Tidy table output
broom::tidy(lmFull.VFcat.Ncorrect.raw, conf.int = T) %>%
    mutate_if(is.numeric, round, 3)
```

```
## # A tibble: 7 x 7
##   term                    estimate std.error statistic p.value conf.low conf.high
##   <chr>                      <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)               89.0        1.83    48.8     0         85.4     92.7
## 2 Age.Category1             -3.52       2.36    -1.49    0.14      -8.22     1.18
## 3 Age.Category2             -1.02       1.49    -0.684   0.496     -3.98     1.94
## 4 CR.composite.before        0.024      1.86     0.013   0.99      -3.68     3.72
## 5 GenCogProc.composite       4.82       4.24     1.14    0.259     -3.61    13.2
## 6 Age.Category1:CR.comp~    -4.38       2.28    -1.92    0.058     -8.92     0.161
## 7 Age.Category2:CR.comp~     1.02       1.31     0.781   0.437     -1.59     3.64
```

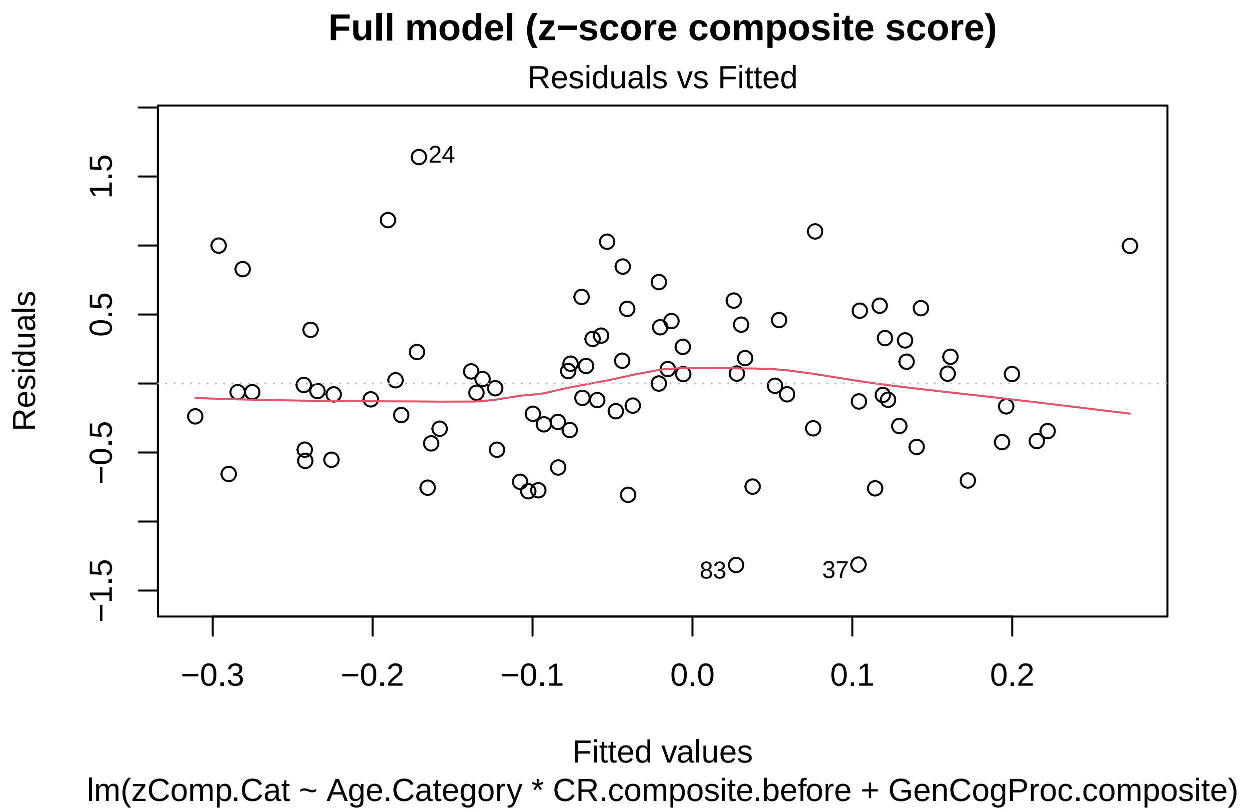
The model doesn't seem to predict the composite score (z-distribution) for Verbal Fluency Categories. However, the interaction between middle age (40-55 years) and CR before seems to show a small significance. Let's check the model assumptions + fit.

Assumption 1 - Linearity

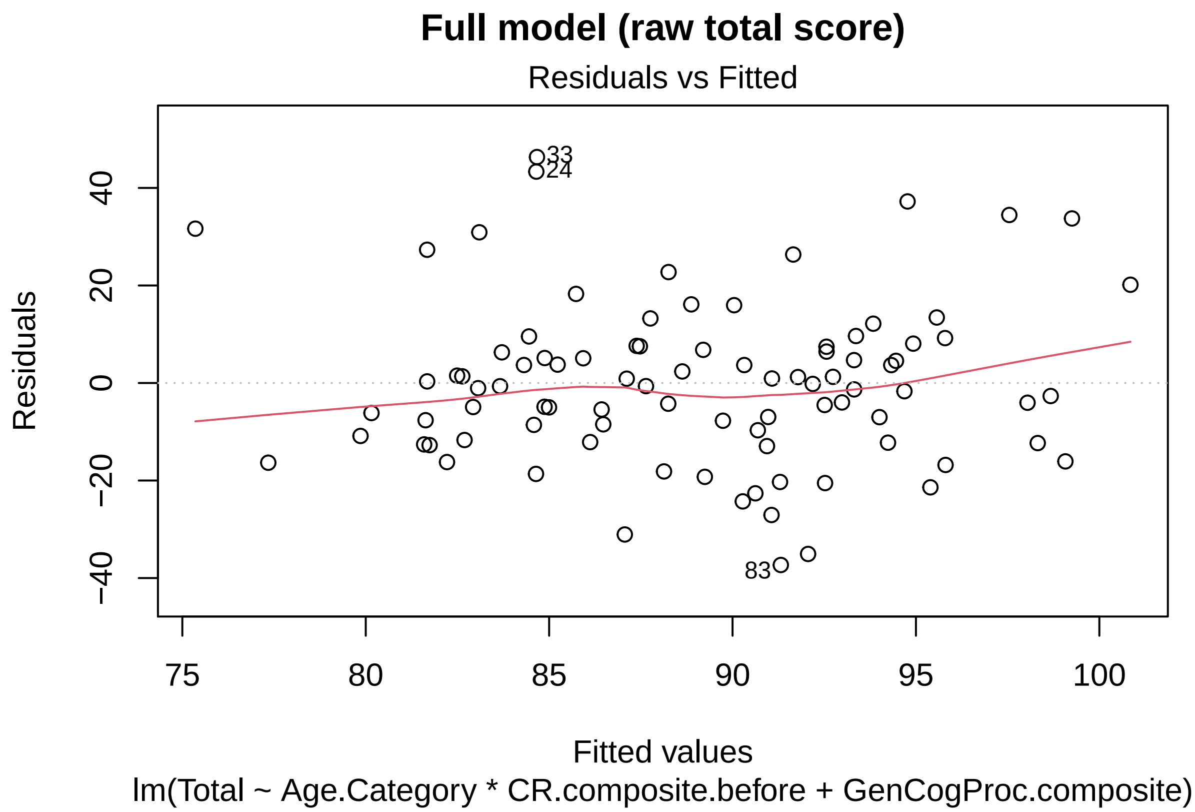
```
## Unconditional model with z composite score
plot(lmUncond.VFcat.Ncorrect, 1, main = "Unconditional model")
```




```r
## Full model with z composite score
plot(lmFull.VFcat.Ncorrect, 1, main = "Full model (z-score composite score)")
```




```r
## Full model with raw total score
plot(lmFull.VFcat.Ncorrect.raw, 1, main = "Full model (raw total score)")
```



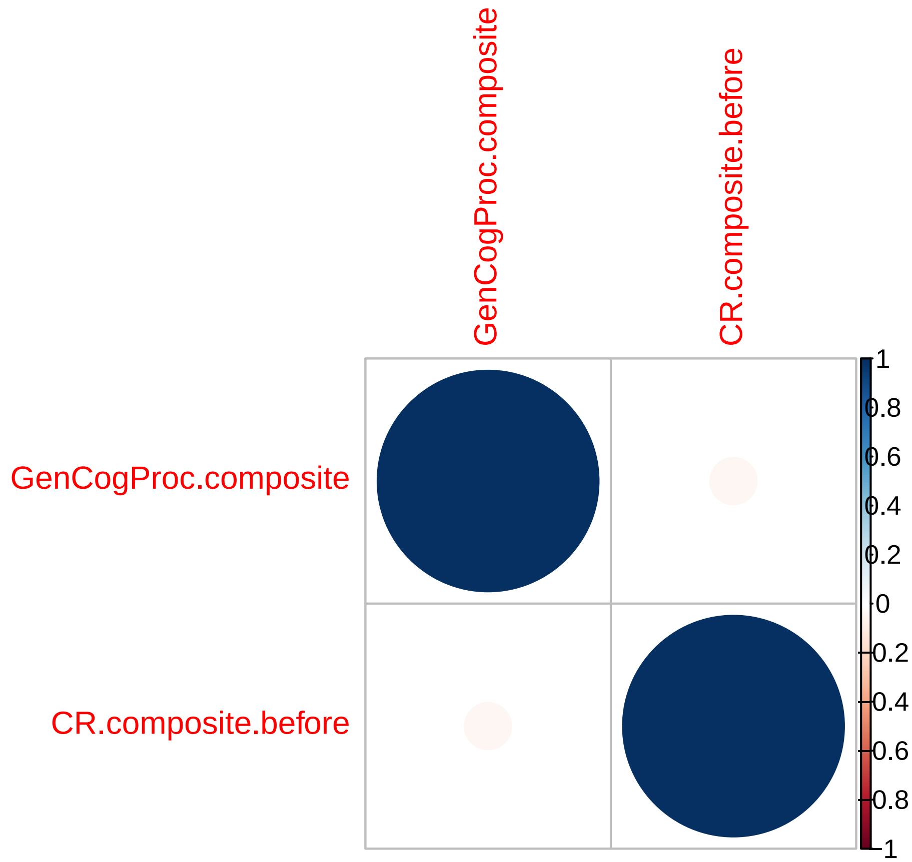
For all models, the red line is mostly horizontal, meaning that there is a linear relationship between the fitted line and the residual value.

Assumption 2 - Independence of Variables

```
# Z composite of VF cat Create table with correlation
# values between predictor variables
tibble::as_tibble(cor(VFcat_Ncorrect_coded[, c(17, 18)]), rownames = "rowname")
```

```
## # A tibble: 2 x 3
##   rowname              GenCogProc.composite CR.composite.before
##   <chr>                               <dbl>               <dbl>
## 1 GenCogProc.composite              1                   -0.0439
## 2 CR.composite.before              -0.0439               1
```

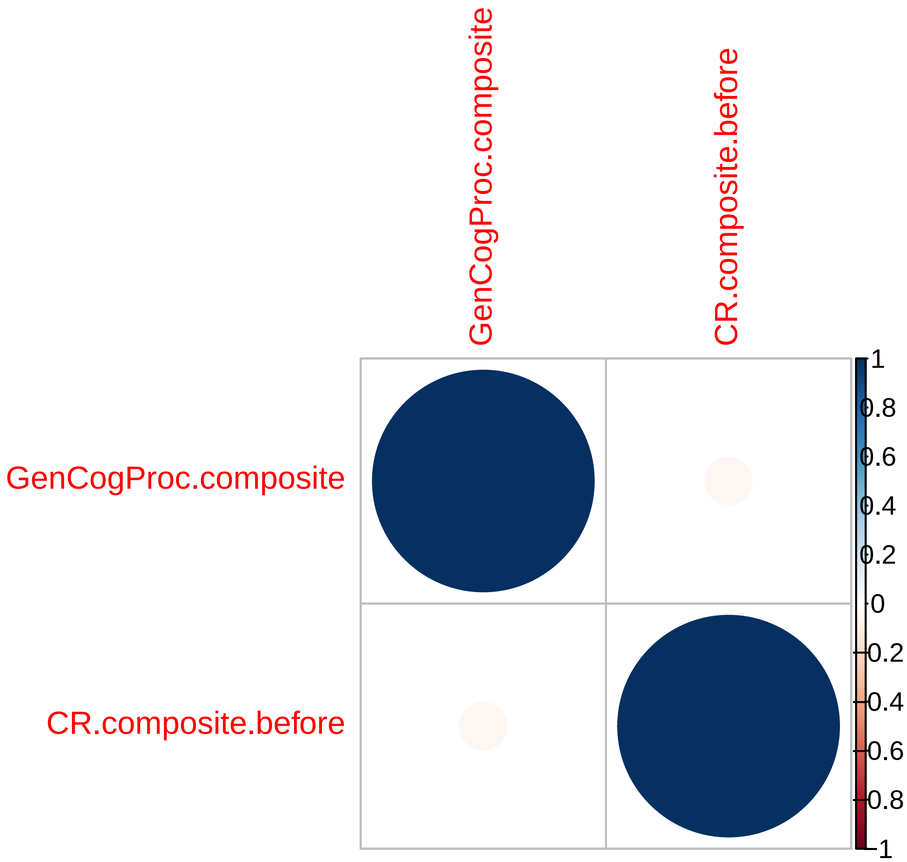
```
## Create correlation plot between predictor variables
corrplot(cor(VFcat_Ncorrect_coded[, c(17, 18)]), method = "circle")
```




```
## Raw scores Create table with correlation values between
## predictor variables
tibble::as_tibble(cor(VFcat_Ncorrect_coded[, c(17, 18)]), rownames = "rowname")
```

```
## # A tibble: 2 x 3
##   rowname              GenCogProc.composite CR.composite.before
##   <chr>                               <dbl>               <dbl>
## 1 GenCogProc.composite               1                  -0.0439
## 2 CR.composite.before               -0.0439               1
```

```
## Create correlation plot between predictor variables
corrplot(cor(VFcat_Ncorrect_coded[, c(17, 18)]))
```

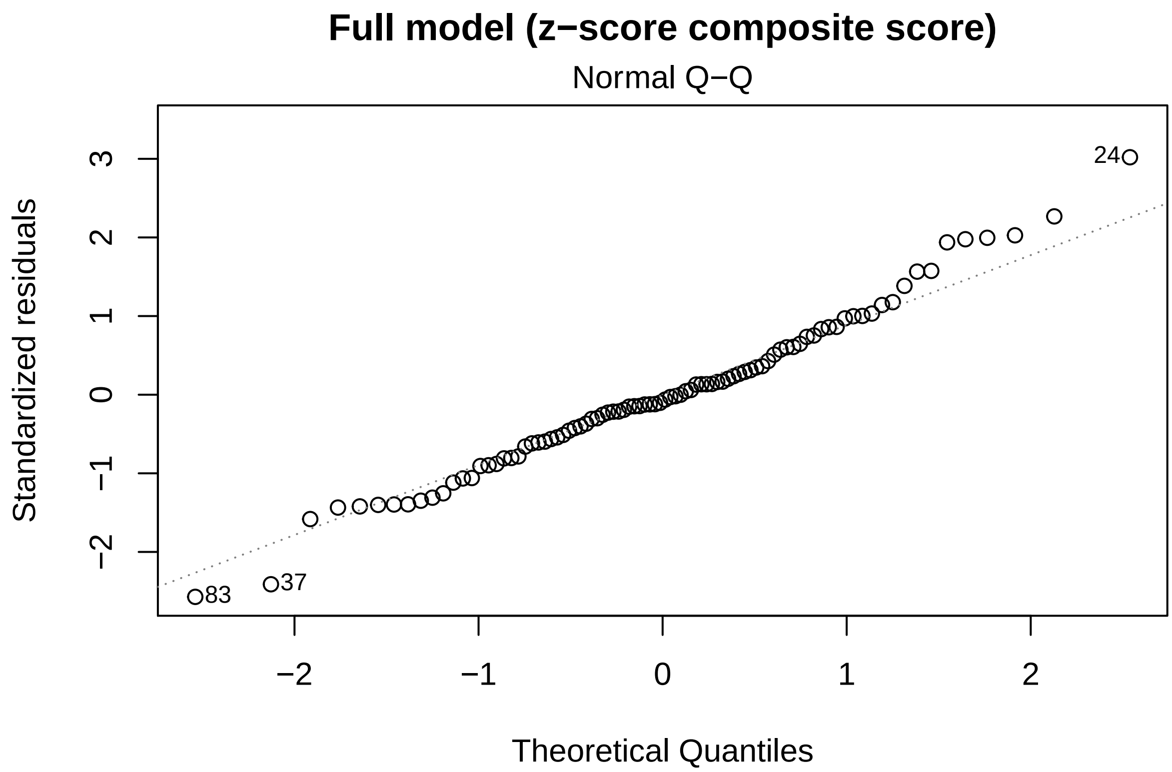


There seems to be no strong relationship between the predictor variables. Hence, we can assume Independence of Variables.

Assumption 3 - Normal Distribution of Residuals

Full model with z composite score

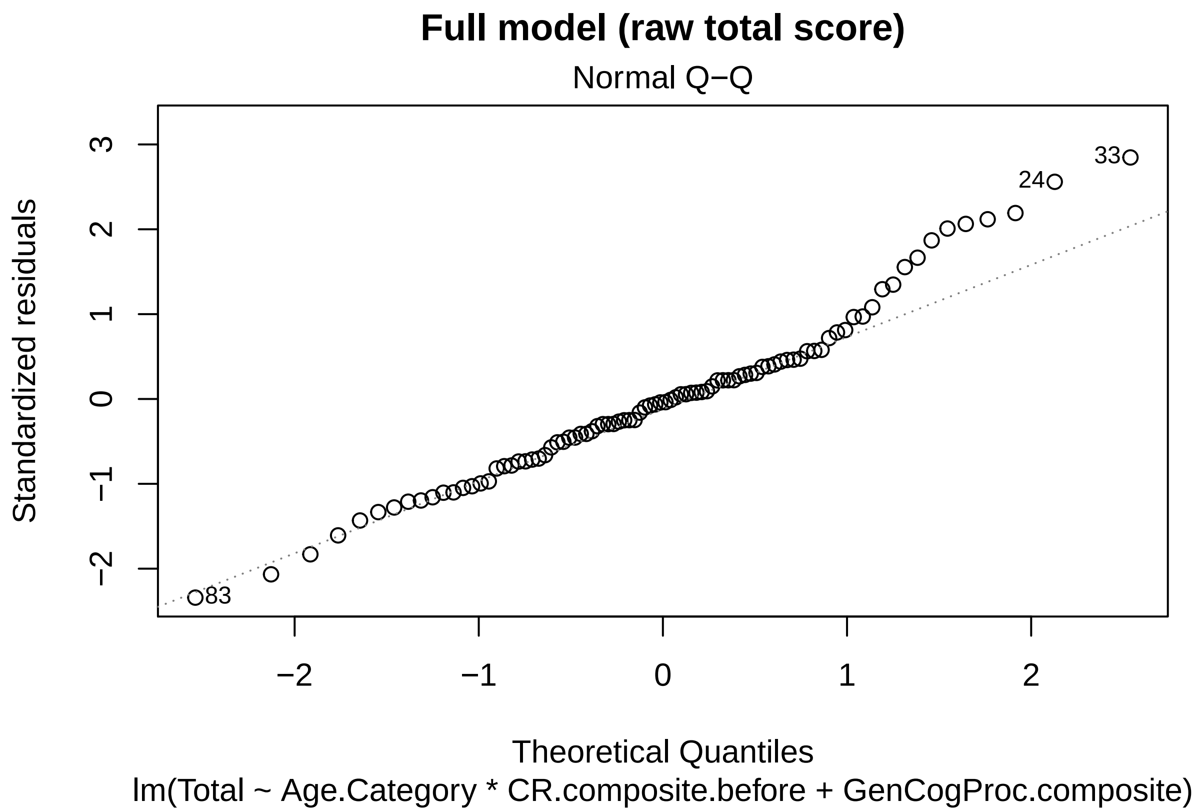
```
plot(lmFull.VFcat.Ncorrect, 2, main = "Full model (z-score composite score)")
```



lm(zComp.Cat ~ Age.Category * CR.composite.before + GenCogProc.composite)

Full model with raw total score

```
plot(lmFull.VFcat.Ncorrect.raw, 2, main = "Full model (raw total score)")
```

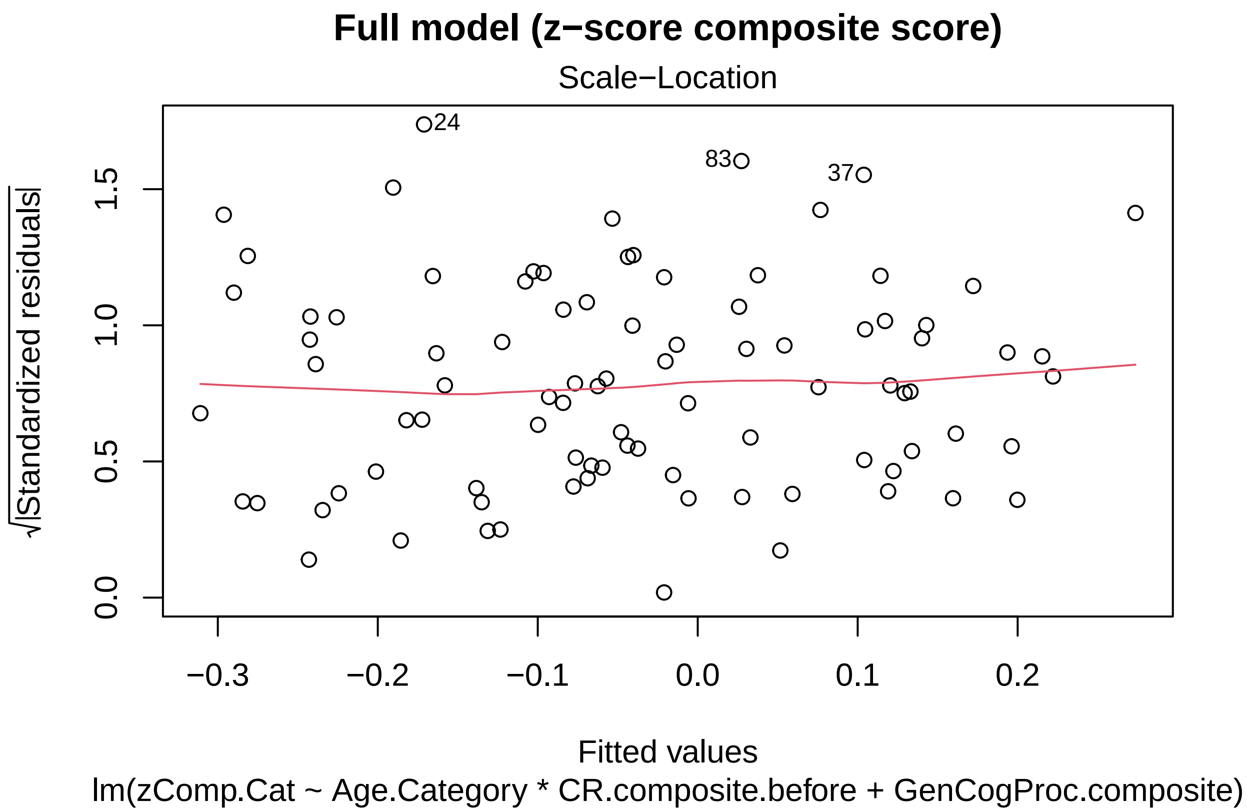


For all models, the points seem to roughly follow a straight line, except for some points on the left and the bulk on the right. Hence, other relationships/predictors that have not been included into the models could explain the variance. One explanation could be the online format of the study. For example, some participants might have scored lower than they would in a lab-based study, due to unclear instructions. Also, it could have caused more distractions than there would have been in a lab based study.

Assumption 4 - Homoscedasticity or Equal Variance of Variables

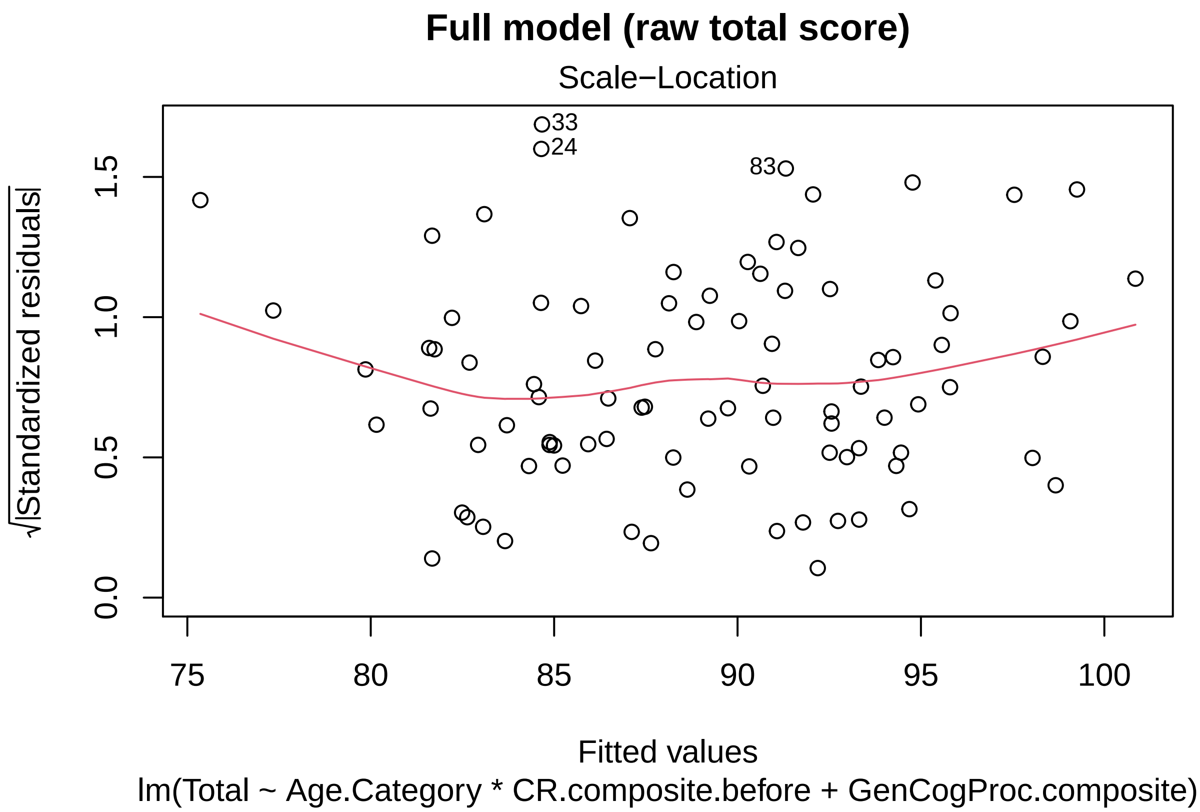
Full model with z composite score

```
plot(lmFull.VFcat.Ncorrect, 3, main = "Full model (z-score composite score)")
```



Full model with raw total score

```
plot(lmFull.VFcat.Ncorrect.raw, 3, main = "Full model (raw total score)")
```



For all models, the variance of residuals seems relatively equal across the predictors. Hence, the error terms are relatively the same across all values of the independent variable for all three models.

Model fit diagnostics *Variation Inflation Factor*

Any VIF value above 4 needs further investigation. There seems to be no concerning signs of colinearity (VIF values higher than 4). The tolerance values indicate the percentage of variance that cannot be explained for by the other predictor variables.

```
ols_vif_tol(lmFull.VFcat.Ncorrect)
```

```
##                            Variables Tolerance      VIF
## 1                      Age.Category1 0.8958833 1.116217
## 2                      Age.Category2 0.7518193 1.330107
## 3                 CR.composite.before 0.9971961 1.002812
## 4                GenCogProc.composite 0.6869407 1.455730
## 5 Age.Category1:CR.composite.before 0.9935202 1.006522
## 6 Age.Category2:CR.composite.before 0.9999277 1.000072
```

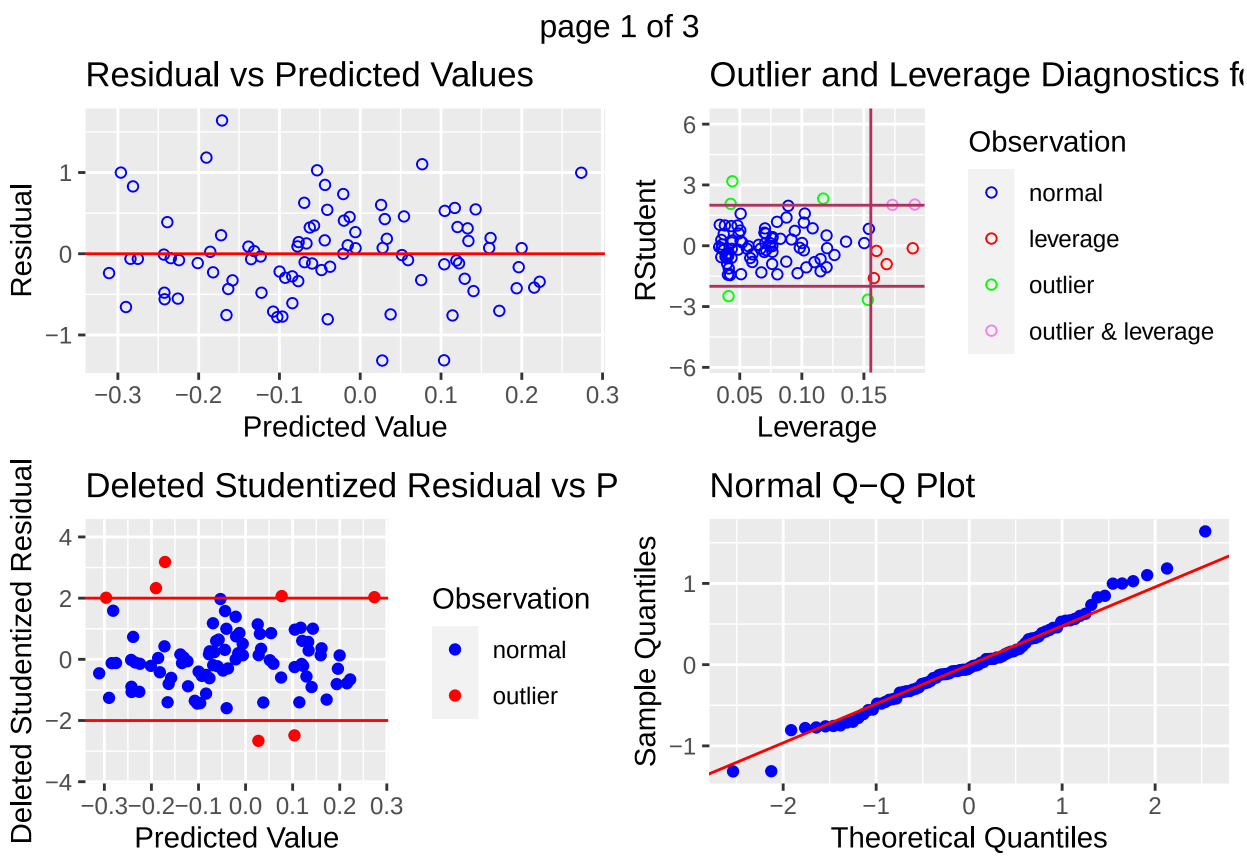
```
ols_vif_tol(lmFull.VFcat.Ncorrect.raw)
```

```
##                            Variables Tolerance      VIF
## 1                      Age.Category1 0.8958833 1.116217
## 2                      Age.Category2 0.7518193 1.330107
## 3                 CR.composite.before 0.9971961 1.002812
```

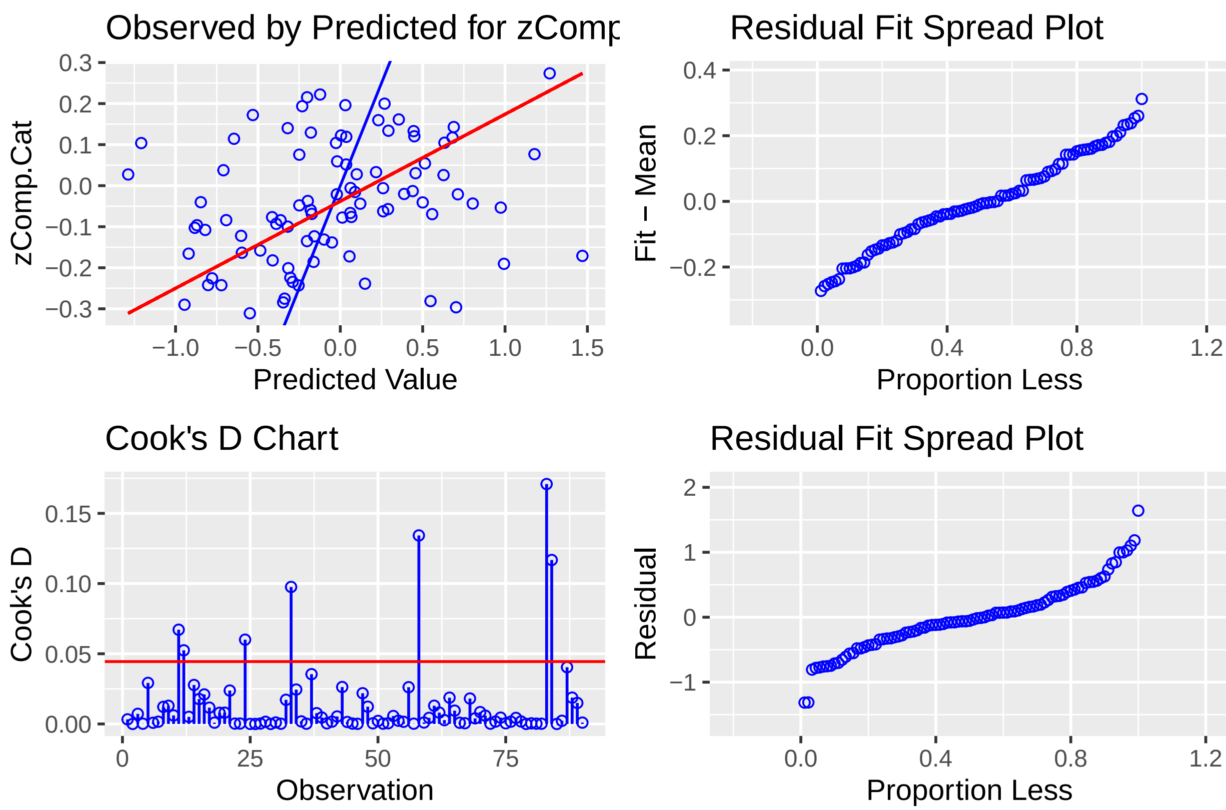
```
## 4               GenCogProc.composite 0.6869407 1.455730
## 5 Age.Category1:CR.composite.before 0.9935202 1.006522
## 6 Age.Category2:CR.composite.before 0.9999277 1.000072
```

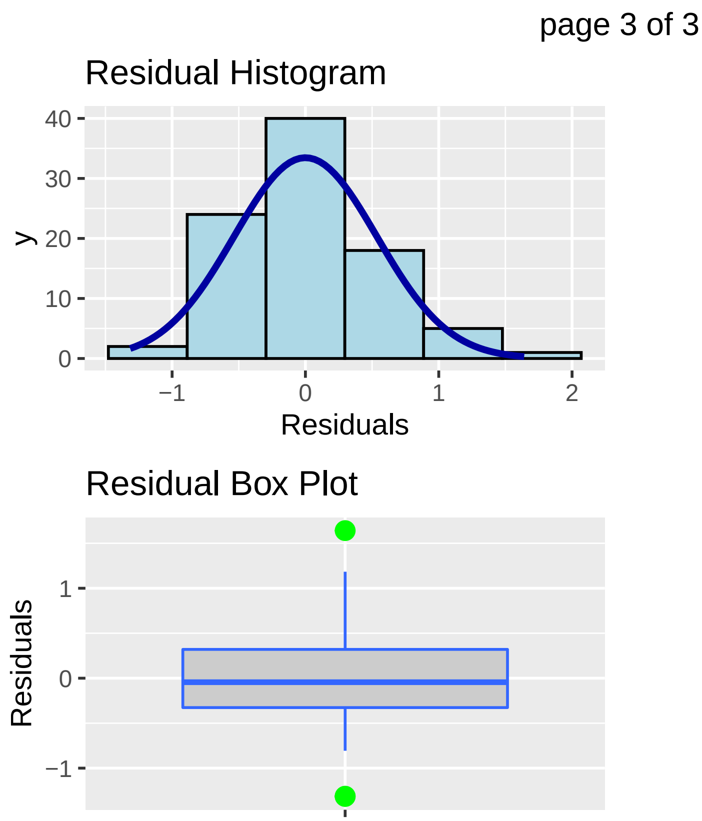
Plot Diagnositcs Full model with z composite score for Semantic Fluency

```
ols_plot_diagnostics(lmFull.VFcat.Ncorrect)
```



page 2 of 3

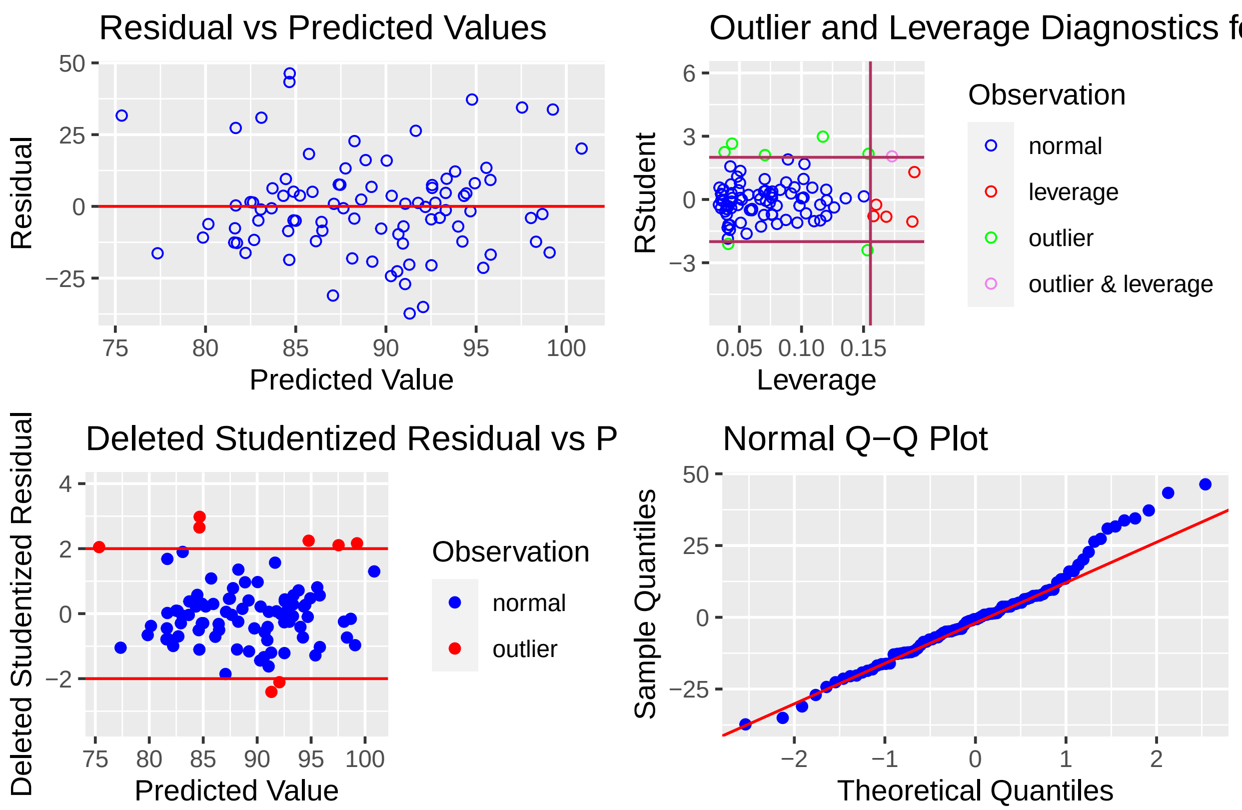

Observed by Predicted for zComp
zComp.Cat
Predicted Value
Residual Fit Spread Plot
Fit − Mean
Proportion Less
Cook's D Chart
Cook's D
Observation
Residual Fit Spread Plot
Residual
Proportion Less

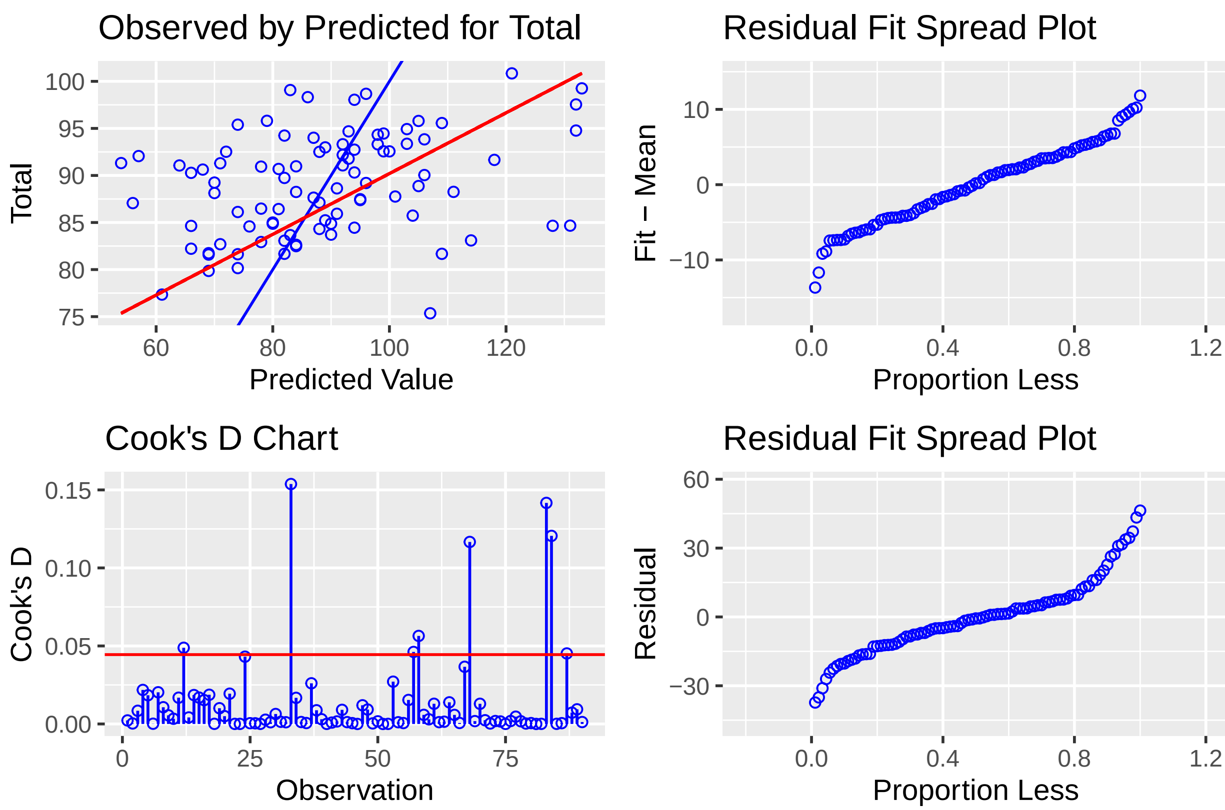


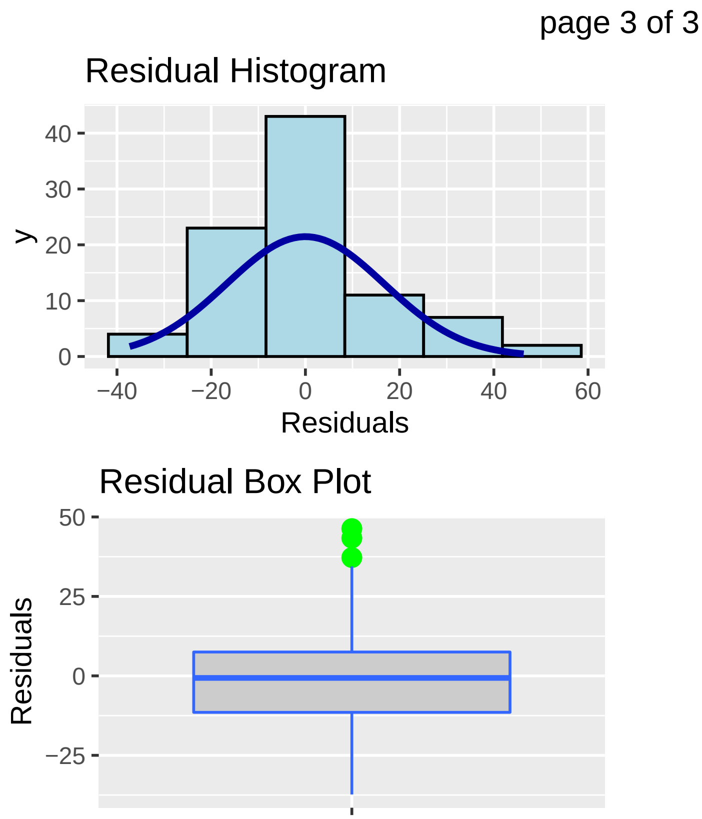
Plot Diagnositcs Full model with raw score for Semantic Fluency

```
ols_plot_diagnostics(lmFull.VFcat.Ncorrect.raw)
```

page 1 of 3


Residual vs Predicted Values
Residual
Predicted Value
Outlier and Leverage Diagnostics fo
RStudent
Leverage
Observation
normal
leverage
outlier
outlier & leverage
Deleted Studentized Residual vs P
Deleted Studentized Residual
Predicted Value
Observation
normal
outlier
Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles


Observed by Predicted for Total
Total
Predicted Value
Residual Fit Spread Plot
Fit − Mean
Proportion Less
Cook's D Chart
Cook's D
Observation
Residual Fit Spread Plot
Residual
Proportion Less



For all models, the Observed vs Predicted Plot shows that the model doesn't fit the data very well (pity) –> due to outliers??

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

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

```
lmFull.VFcat.Ncorrect.during <- lm(zComp.Cat ~ Age.Category *
    CR.composite.during + GenCogProc.composite, data = VFcat_Ncorrect_coded)
summary(lmFull.VFcat.Ncorrect.during)
```

```
##
## Call:
## lm(formula = zComp.Cat ~ Age.Category * CR.composite.during +
##     GenCogProc.composite, data = VFcat_Ncorrect_coded)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2984 -0.3024 -0.0385  0.3256  1.6172
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          -0.03814    0.05871  -0.650   0.5177
## Age.Category1                        -0.13209    0.07610  -1.736   0.0863 .
## Age.Category2                        -0.02254    0.04806  -0.469   0.6404
```

```
## CR.composite.during                0.05198    0.06038   0.861   0.3917
## GenCogProc.composite               0.06542    0.13818   0.473   0.6371
## Age.Category1:CR.composite.during -0.07919    0.07446  -1.064   0.2906
## Age.Category2:CR.composite.during  0.01661    0.04289   0.387   0.6995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.557 on 83 degrees of freedom
## Multiple R-squared:  0.06268,    Adjusted R-squared:  -0.005075
## F-statistic: 0.9251 on 6 and 83 DF,  p-value: 0.4814
```

```
# Model comparisons
anova(lmFull.VFcat.Ncorrect, lmFull.VFcat.Ncorrect.during)
```

```
## Analysis of Variance Table
##
## Model 1: zComp.Cat ~ Age.Category * CR.composite.before + GenCogProc.composite
## Model 2: zComp.Cat ~ Age.Category * CR.composite.during + GenCogProc.composite
##   Res.Df    RSS Df Sum of Sq F Pr(>F)
## 1     83 25.628
## 2     83 25.747  0  -0.11905
```

No differences between the two models

```
# Model comparisons through AIC values
AIC(lmFull.VFcat.Ncorrect)
```

```
## [1] 158.3581
```

```
AIC(lmFull.VFcat.Ncorrect.during)
```

```
## [1] 158.7752
```

The model with the composite score before Covid-19 seems to fit slightly better

Verbal Fluency - Letters

Descriptive Statistics

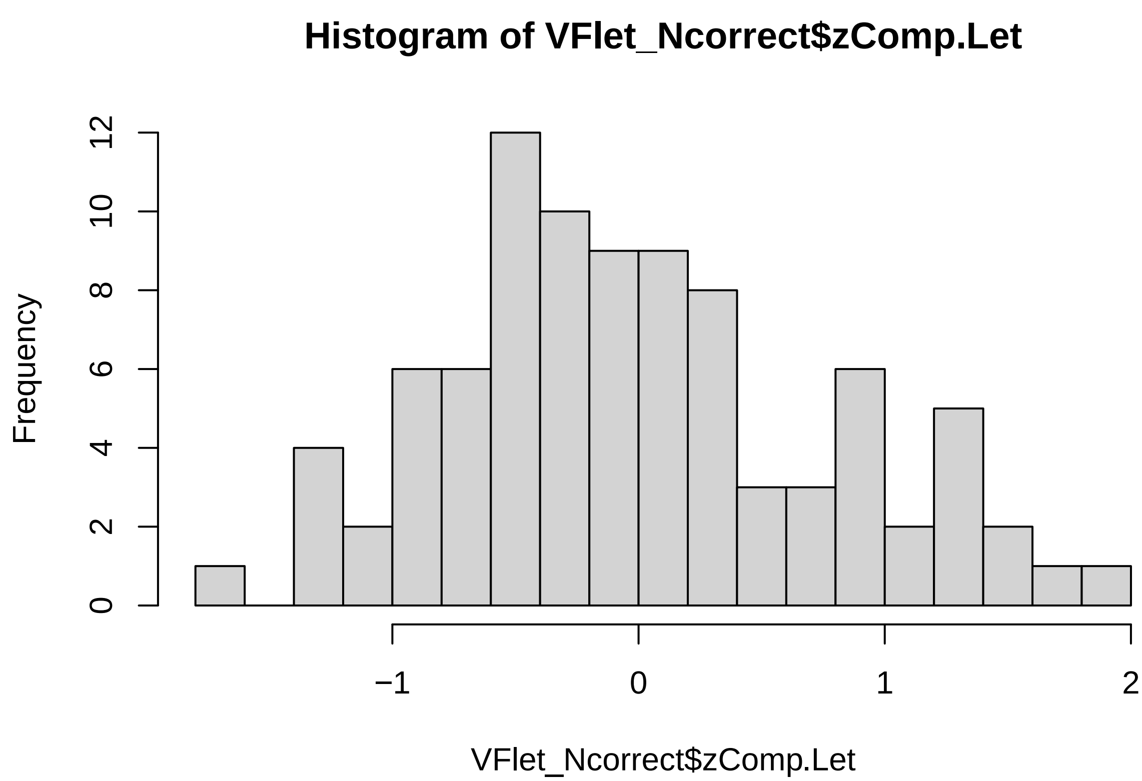
```
## Create dataset that only includes data of the Number of Correct words Produced
VFlet_Ncorrect <- VFlet %>%
  dplyr::filter(Measures=="Ncorrect") %>%
  dplyr::select(-Measures) %>%
      #Recode age groups
  dplyr::mutate(Age.Category=as.factor(dplyr::recode(Age.Category, '18 to 30 years old'="Younger",
                                   '40 to 55 years old'="Middle-Aged",
                                   '65 to 80 years old'="Older"))) %>%
    #Use the z scores to filter out outliers (i.e., exclude values +/-2.5 SD per trial )
  filter(between(zComp.Let, -2.5, +2.5)) %>% #No outliers
  ungroup()

# head(VFlet_Ncorrect_coded, L=6)
```

```r
#Summarise data to present descriptives in a table
(Descr_VFlet <- VFlet_Ncorrect %>%
  #Per Age Group
  group_by(Age.Category) %>%
  summarise(Nppt = length(unique(ID)),
            total = round(mean(Total, na.rm=T),2),
            sdtotal = round(sd(Total,na.rm=T),2),
            # ztotal = mean(zComp.Let, na.rm=T),
            letterM = round(mean(M, na.rm=T),2),
            letterS = round(mean(S, na.rm=T),2),
            letterP = round(mean(P, na.rm=T),2)))
```

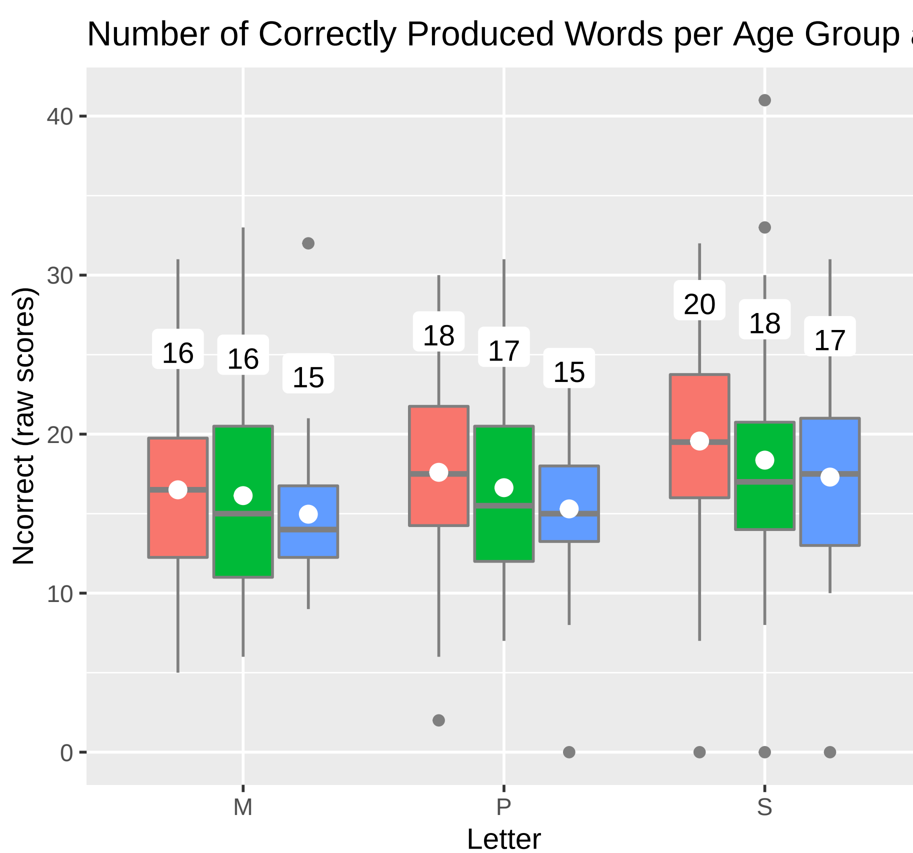
```
## # A tibble: 3 x 7
##   Age.Category  Nppt total sdtotal letterM letterS letterP
##   <fct>        <int> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Middle-Aged     30  53.7    17.6    16.5    19.6    17.6
## 2 Older           30  51.1    19.6    16.1    18.4    16.6
## 3 Younger         30  47.6    14.0    15.0    17.3    15.3
```

```r
#Save table
# write.csv(Descr_VFlet, "./Figures and Tables/Descr_VFlet_Ncorrect.csv", row.names = F)
```

```r
hist(VFlet_Ncorrect$zComp.Let, breaks = 20)  #Composite z-score of Semantic Fluency
```



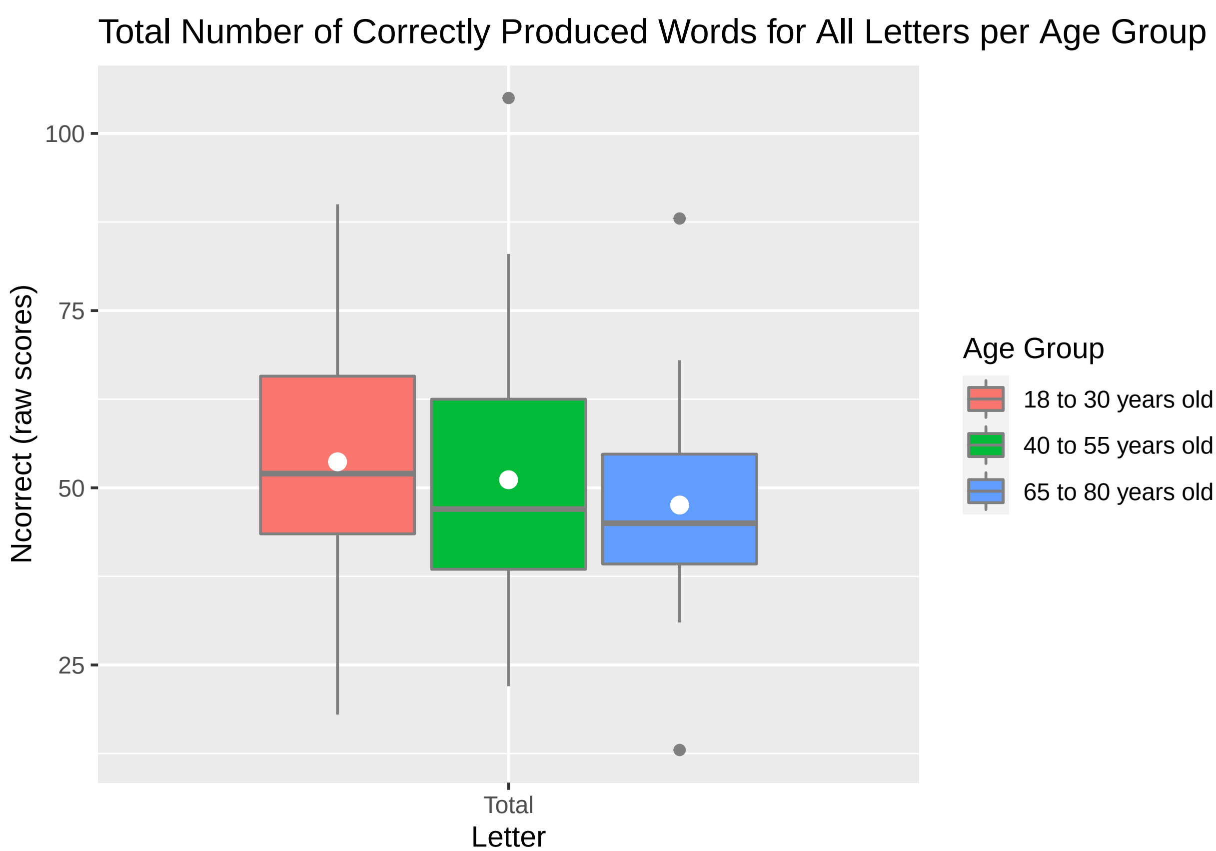
```
VFlet_Ncorrect.long <- VFlet_Ncorrect %>%
  pivot_longer(cols=Total:S, names_to = "letter", values_to = "Ncorrect")

# Boxplot VFlet Ncorrect
# png(file="./Figures and Tables/Boxplot_VFletNcorrect.png",
# width=600, height=350)

(Boxplot_VF <- VFlet_Ncorrect.long %>%
    dplyr::filter(letter!="Total") %>%
    ggplot(aes(x=letter, y=Ncorrect,
                                              fill = as.factor(Age.Category))) +
    geom_boxplot(colour="grey50")+
    stat_summary(aes(label=round(..y..), group=as.factor(Age.Category)),
                 fun=mean, geom = "label", size=4,
                 fill="white", show.legend=NA, label.size=NA,
                 position = position_dodge(.75), vjust=-3) +
      stat_summary(fun = "mean", position = position_dodge(.75),
                 show.legend=F, colour="white")+ #Mean as white dot
    labs(x = "Letter",
         y = "Ncorrect (raw scores)",
         title = "Number of Correctly Produced Words per Age Group and Letter")+
        scale_fill_discrete(guide=guide_legend(title = "Age Group"), labels=c("18 to 30 years old","40 
```



Visualisation Ncorrect Letter Fluency

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

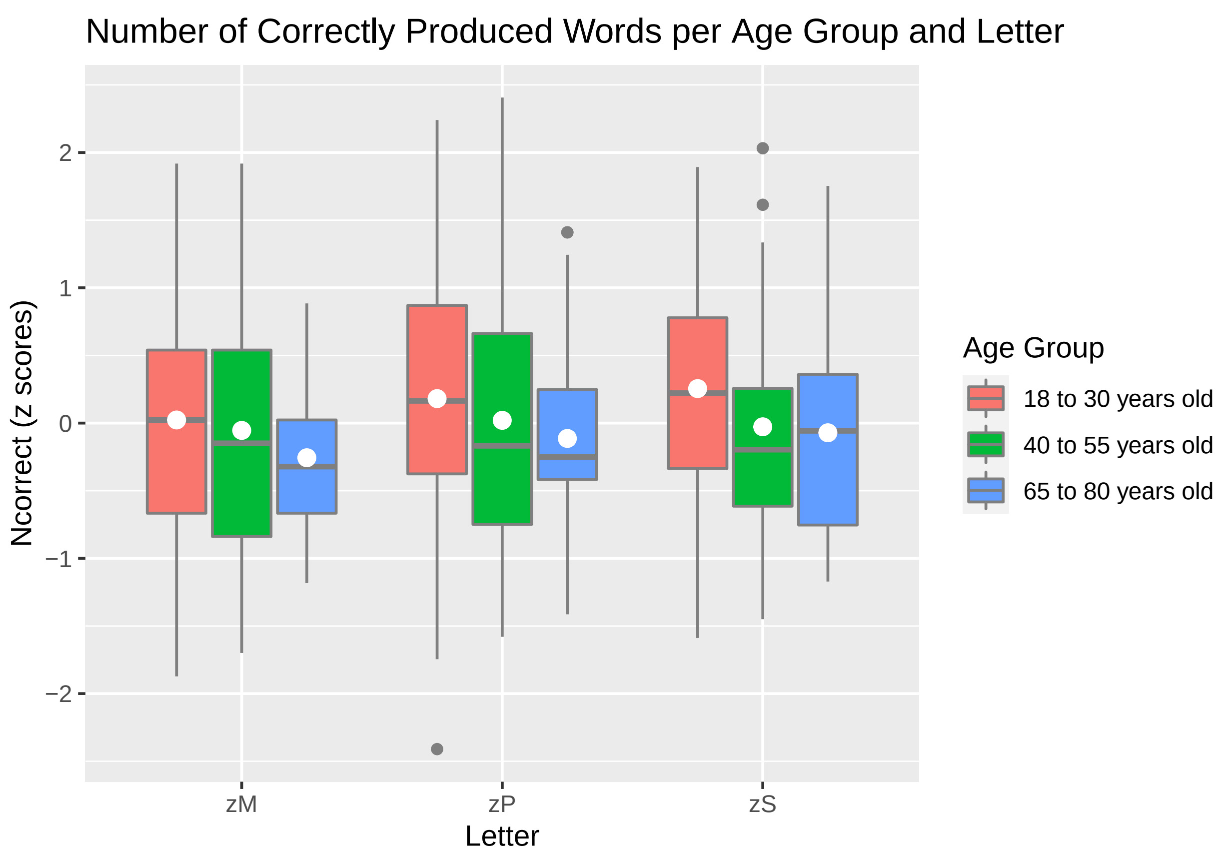
```r
# Boxplot VFlet Ncorrect Raw Total
# png(file="./Figures and Tables/Boxplot_VFletNcorrect_RawTotal.png",
# width=600, height=350)

(Boxplot_VF <- VFlet_Ncorrect.long %>%
    dplyr::filter(letter=="Total") %>%
    ggplot(aes(x=letter, y=Ncorrect,fill = as.factor(Age.Category))) +
    geom_boxplot(colour="grey50")+
    # stat_summary(aes(label=round(..y..), group=as.factor(Age.Category)),
    #              fun=mean, geom = "label", size=4,
    #              fill="white", show.legend=NA, label.size=NA,
    #              position = position_dodge(.75), vjust=-3) +
      stat_summary(fun = "mean", position = position_dodge(.75),
                   show.legend=F, colour="white")+ #Mean as white dot
    labs(x = "Letter",
         y = "Ncorrect (raw scores)",
         title = "Total Number of Correctly Produced Words for All Letters per Age Group")+
        scale_fill_discrete(guide=guide_legend(title = "Age Group"), labels=c("18 to 30 years old","40 t
```




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

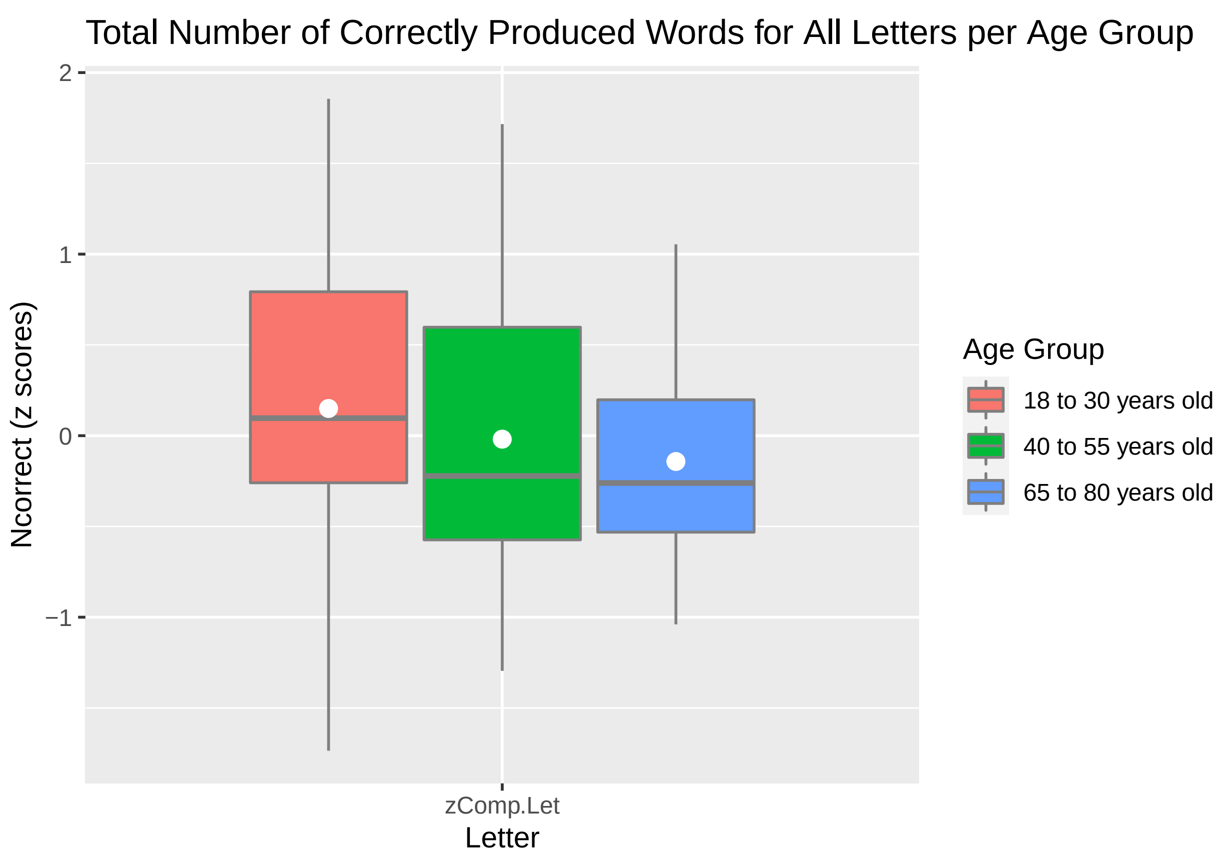
Figures Ncorrect Letter Fluency z-scores

```r
VFlet_Ncorrect.long.zScores <- VFlet_Ncorrect %>%
  pivot_longer(cols=zComp.Let:zP, names_to = "zletter", values_to = "zNcorrect")

# Boxplot VFlet Ncorrect
# png(file="./Figures and Tables/Boxplot_VFletNcorrect_zscores.png",
# width=600, height=350)

(Boxplot_VF <- VFlet_Ncorrect.long.zScores %>%
    dplyr::filter(zletter!="zComp.Let") %>%
    ggplot(aes(x=zletter, y=zNcorrect,
                                                fill = as.factor(Age.Category))) +
    geom_boxplot(colour="grey50")+
    # stat_summary(aes(label=round(..y..), group=as.factor(Age.Category)),
    #             fun=mean, geom = "label", size=4,
    #             fill="white", show.legend=NA, label.size=NA,
    #             position = position_dodge(.75), vjust=-3) +
      stat_summary(fun = "mean", position = position_dodge(.75),
               show.legend=F, colour="white")+ #Mean as white dot
    labs(x = "Letter",
         y = "Ncorrect (z scores)",
         title = "Number of Correctly Produced Words per Age Group and Letter")+
        scale_fill_discrete(guide=guide_legend(title = "Age Group"), labels=c("18 to 30 years old","40
```




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

```r
# Boxplot VFlet Ncorrect Raw Total

# png(file="./Figures and Tables/Boxplot_VFletNcorrect_Total_zscores.png",
# width=600, height=350)


(Boxplot_VF <- VFlet_Ncorrect.long.zScores %>%
    dplyr::filter(zletter=="zComp.Let") %>%
    ggplot(aes(x=zletter, y=zNcorrect,fill = as.factor(Age.Category))) +
    geom_boxplot(colour="grey50")+
    # stat_summary(aes(label=round(..y..), group=as.factor(Age.Category)),
    #             fun=mean, geom = "label", size=4,
    #             fill="white", show.legend=NA, label.size=NA,
    #             position = position_dodge(.75), vjust=-3) +
      stat_summary(fun = "mean", position = position_dodge(.75),
               show.legend=F, colour="white")+ #Mean as white dot
    labs(x = "Letter",
         y = "Ncorrect (z scores)",
         title = "Total Number of Correctly Produced Words for All Letters per Age Group")+
        scale_fill_discrete(guide=guide_legend(title = "Age Group"), labels=c("18 to 30 years old","40 t
```



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

Multiple Linear Regression - Letter Fluency

Create user-defined contrasts for the Age Category variable

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

VFlet_Ncorrect_coded <- VFlet_Ncorrect
contrasts(VFlet_Ncorrect_coded$Age.Category) <- contr.helmert(3)
contrasts(VFlet_Ncorrect_coded$Age.Category)
```

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

Unconditional model, i.e. without covariates

```
lmUncond.VFlet.Ncorrect <- lm(zComp.Let ~ Age.Category * CR.composite.before,
    data = VFlet_Ncorrect_coded)
# broom::tidy(lmUncond.VFlet.Ncorrect, conf.int=T)
```

Full model including the covariates; outcome variable in z-distribution

```
lmFull.VFlet.Ncorrect <- lm(zComp.Let ~ Age.Category * CR.composite.before +
    GenCogProc.composite, data = VFlet_Ncorrect_coded)
summary(lmFull.VFlet.Ncorrect)
```

```
##
## Call:
## lm(formula = zComp.Let ~ Age.Category * CR.composite.before +
##     GenCogProc.composite, data = VFlet_Ncorrect_coded)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.94094 -0.47542 -0.06421  0.50226  1.67115
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -0.003046   0.081901  -0.037   0.9704
## Age.Category1                     -0.186888   0.105972  -1.764   0.0815 .
## Age.Category2                      0.032087   0.066788   0.480   0.6322
## CR.composite.before                0.106729   0.083415   1.280   0.2043
## GenCogProc.composite               0.227047   0.189985   1.195   0.2355
## Age.Category1:CR.composite.before -0.049753   0.102351  -0.486   0.6282
## Age.Category2:CR.composite.before  0.031134   0.058902   0.529   0.5985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7769 on 83 degrees of freedom
## Multiple R-squared:  0.06206,    Adjusted R-squared:  -0.005743
## F-statistic: 0.9153 on 6 and 83 DF,  p-value: 0.4882
```

```r
# Tidy table output
(tidylmFull.VFlet.Ncorrect <- broom::tidy(lmFull.VFlet.Ncorrect,
    conf.int = T) %>%
    mutate_if(is.numeric, round, 3))
```

```
## # A tibble: 7 x 7
##   term                     estimate std.error statistic p.value conf.low conf.high
##   <chr>                       <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)                -0.003     0.082    -0.037   0.97    -0.166     0.16
## 2 Age.Category1              -0.187     0.106    -1.76    0.081   -0.398     0.024
## 3 Age.Category2               0.032     0.067     0.48    0.632   -0.101     0.165
## 4 CR.composite.before         0.107     0.083     1.28    0.204   -0.059     0.273
## 5 GenCogProc.composite        0.227     0.19      1.20    0.235   -0.151     0.605
## 6 Age.Category1:CR.comp~     -0.05      0.102    -0.486   0.628   -0.253     0.154
## 7 Age.Category2:CR.comp~      0.031     0.059     0.529   0.599   -0.086     0.148
```

```r
# write.csv(tidylmFull.VFlet.Ncorrect, './Figures and
# Tables/VFlet_zNcorrect_lmFull.csv')
```

Full model including the covariates; outcome variable as raw score

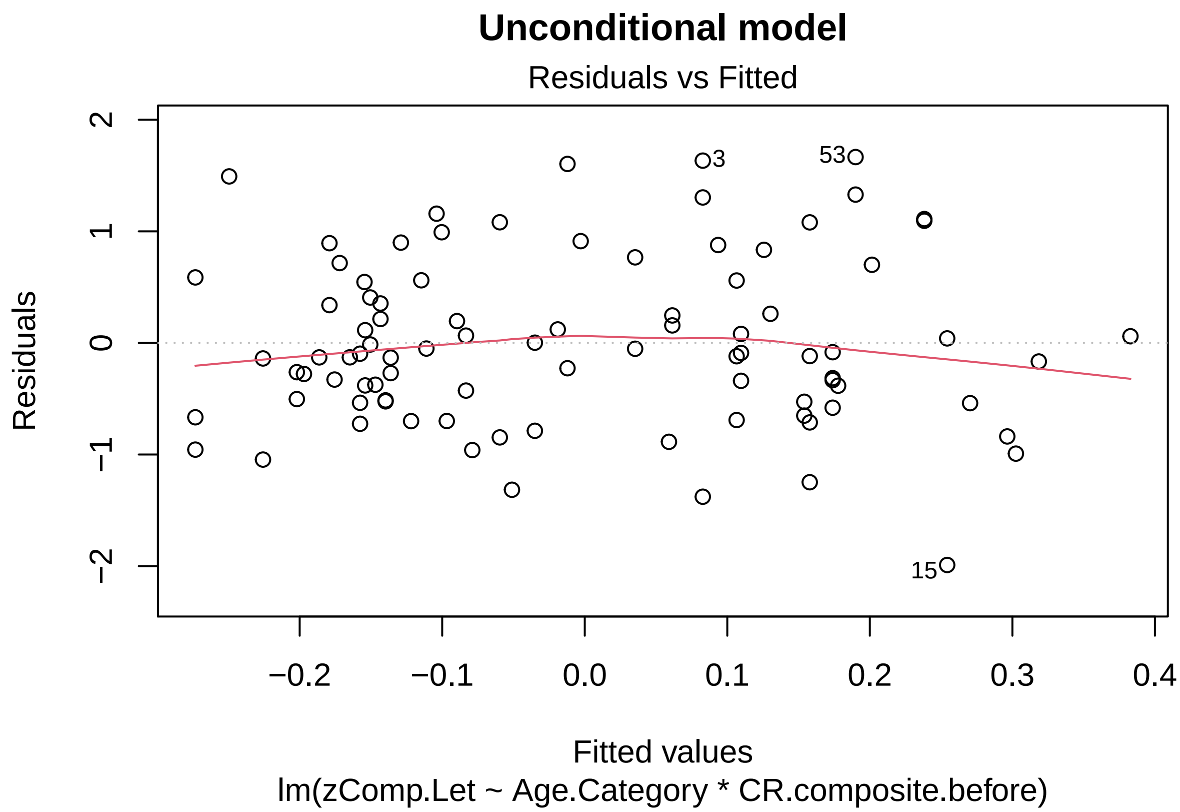
```r
lmFull.VFlet.Ncorrect.raw <- lm(Total ~ Age.Category * CR.composite.before +
    GenCogProc.composite, data = VFlet_Ncorrect_coded)

# Tidy table output
broom::tidy(lmFull.VFlet.Ncorrect.raw, conf.int = T) %>%
    mutate_if(is.numeric, round, 3)
```

```
## # A tibble: 7 x 7
##   term                     estimate std.error statistic p.value conf.low conf.high
##   <chr>                       <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)                50.8        1.82    27.9     0         47.2     54.4
## 2 Age.Category1              -4.08       2.36    -1.73    0.087     -8.77     0.601
## 3 Age.Category2               1.18       1.48     0.793   0.43      -1.78     4.13
## 4 CR.composite.before         2.40       1.85     1.29    0.2       -1.29     6.08
## 5 GenCogProc.composite        5.74       4.22     1.36    0.178     -2.66    14.1
## 6 Age.Category1:CR.comp~     -0.41       2.27    -0.18    0.857     -4.93     4.11
## 7 Age.Category2:CR.comp~      0.402      1.31     0.307   0.759     -2.20     3.00
```

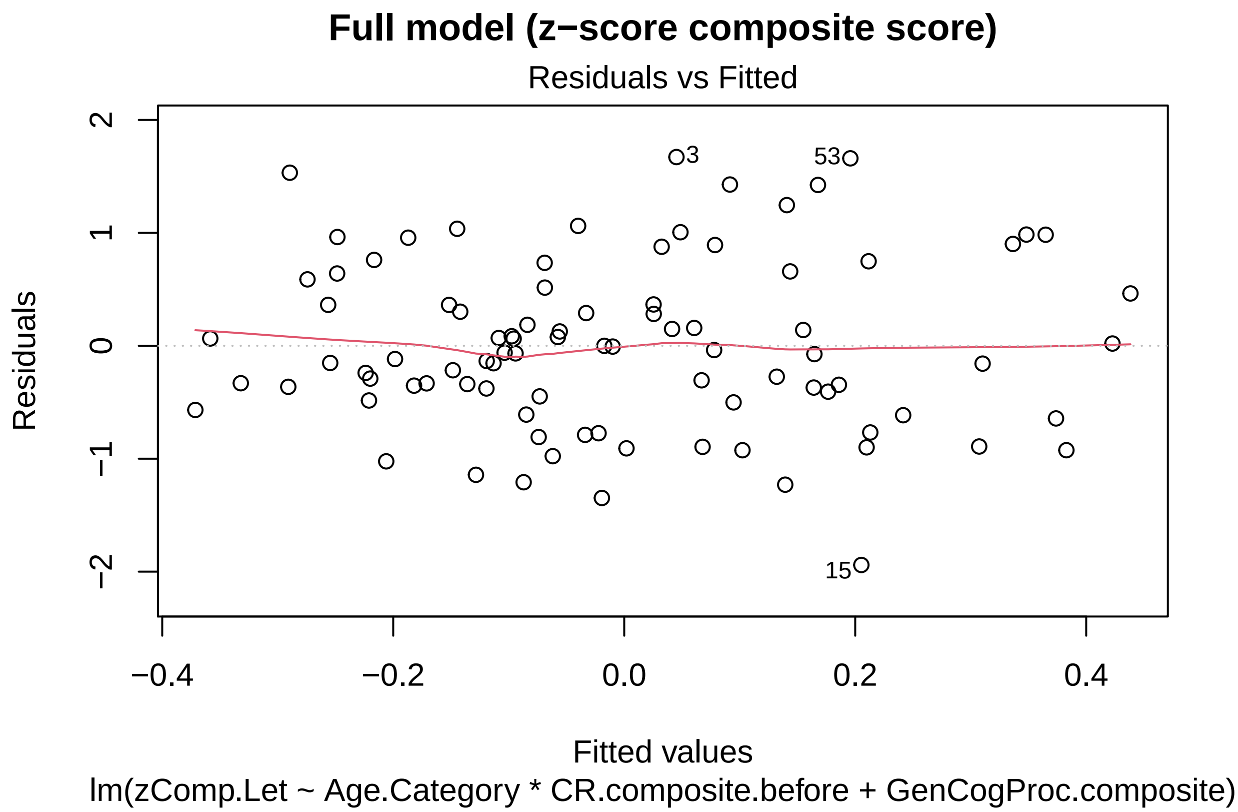
The model doesn't seem to predict the composite score (z-distribution) or raw Total score for Verbal Fluency Letter. Let's check the model assumptions + fit.

Assumption 1 - Linearity

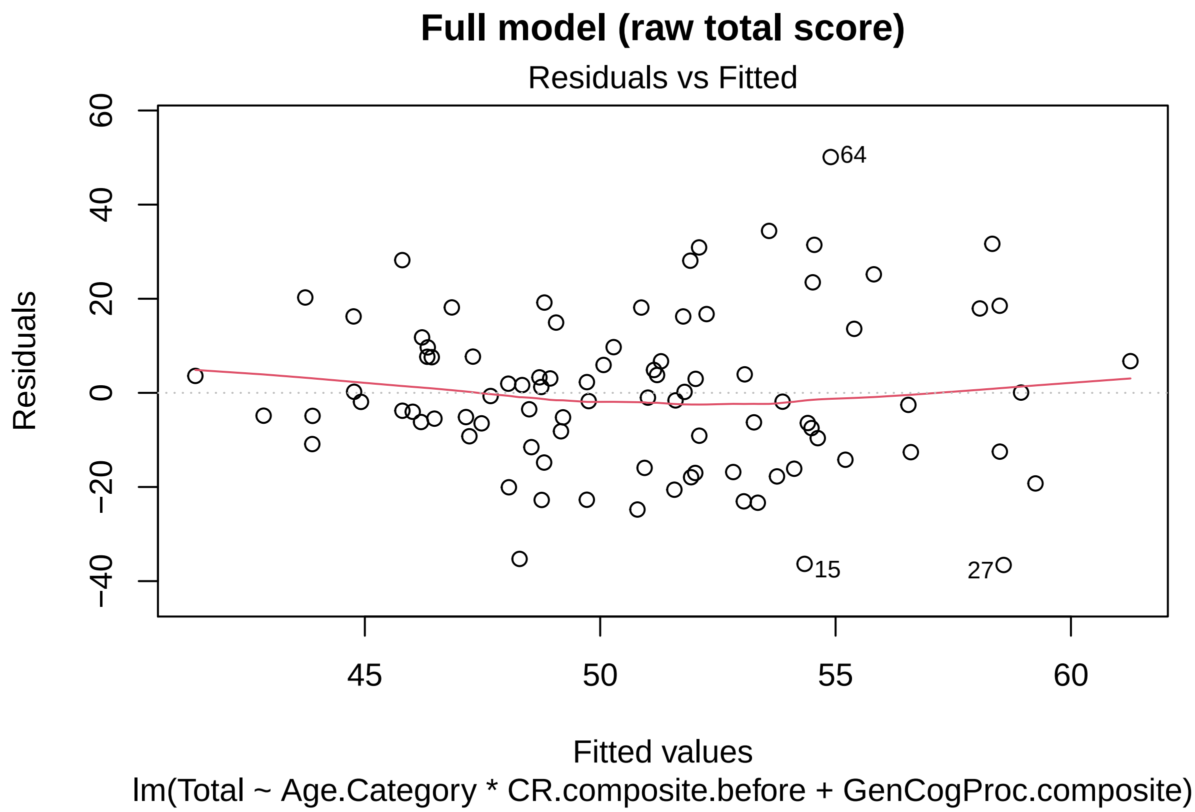
```r
## Unconditional model with z composite score
plot(lmUncond.VFlet.Ncorrect, 1, main = "Unconditional model")
```




```
## Full model with z composite score
plot(lmFull.VFlet.Ncorrect, 1, main = "Full model (z-score composite score)")
```




```r
## Full model with raw total score
plot(lmFull.VFlet.Ncorrect.raw, 1, main = "Full model (raw total score)")
```



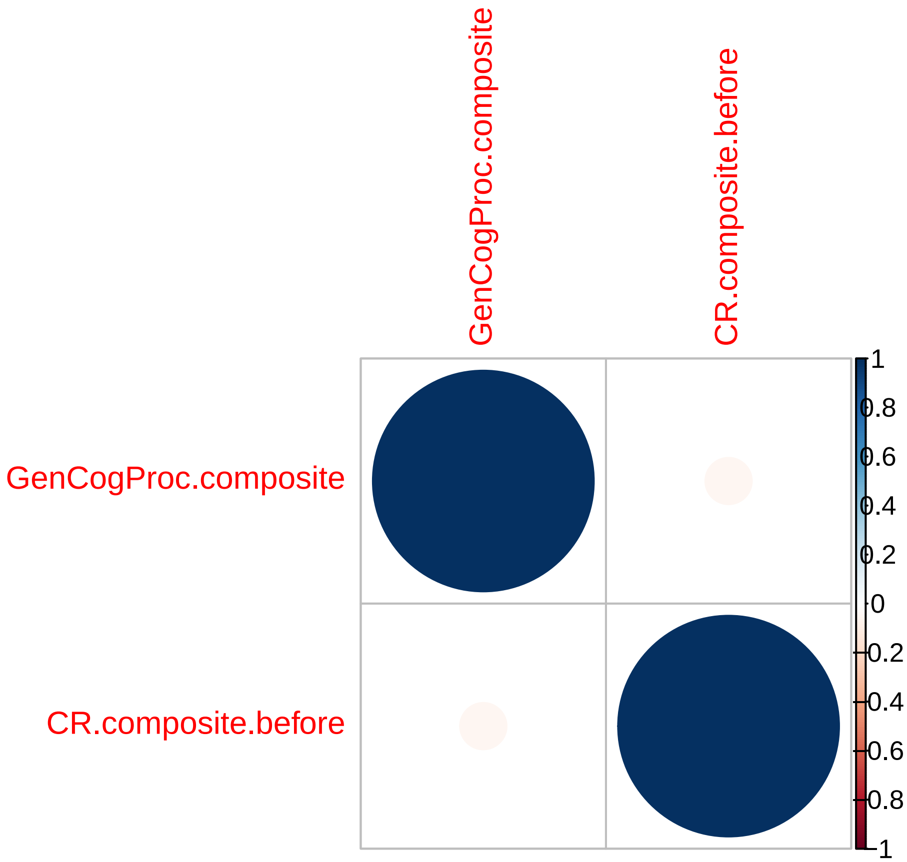
For all models, the red line is mostly horizontal, meaning that there is a linear relationship between the fitted line and the residual value.

Assumption 2 - Independence of Variables

```
# Z composite of VF cat Create table with correlation
# values between predictor variables
tibble::as_tibble(cor(VFlet_Ncorrect_coded[, c(13, 14)]), rownames = "rowname")
```

```
## # A tibble: 2 x 3
##   rowname              GenCogProc.composite CR.composite.before
##   <chr>                               <dbl>               <dbl>
## 1 GenCogProc.composite               1                  -0.0439
## 2 CR.composite.before              -0.0439               1
```

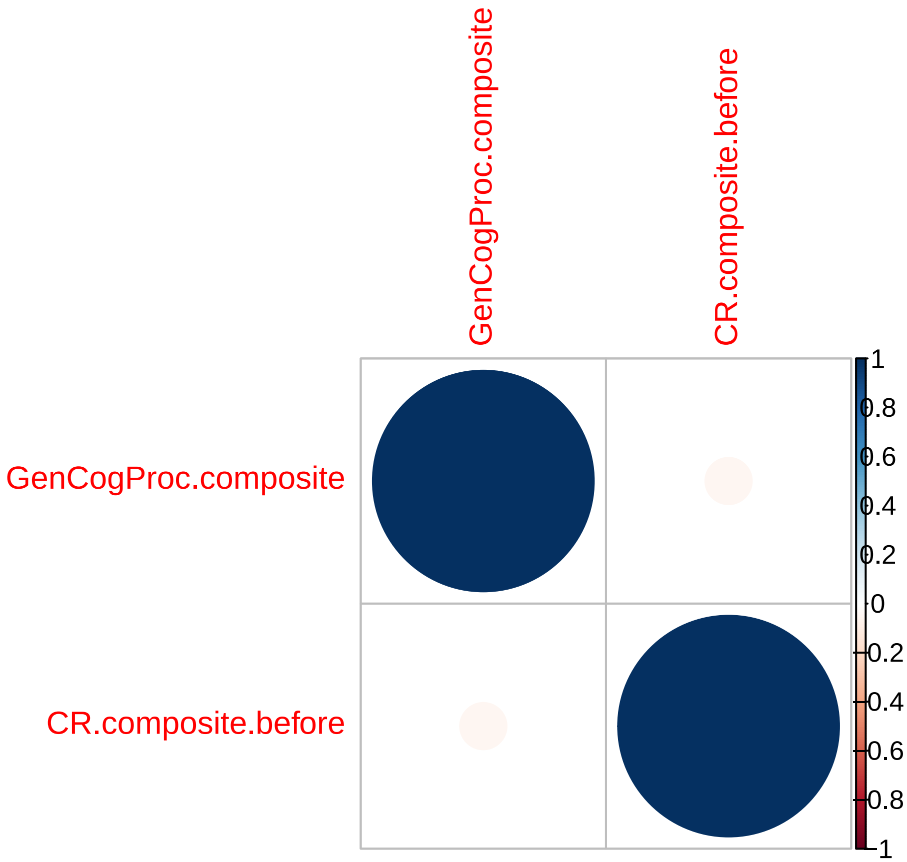
```
## Create correlation plot between predictor variables
corrplot(cor(VFlet_Ncorrect_coded[, c(13, 14)]), method = "circle")
```




```r
## Raw scores Create table with correlation values between
## predictor variables
tibble::as_tibble(cor(VFlet_Ncorrect_coded[, c(13, 14)]), rownames = "rowname")
```

```
## # A tibble: 2 x 3
##   rowname              GenCogProc.composite CR.composite.before
##   <chr>                               <dbl>               <dbl>
## 1 GenCogProc.composite               1                  -0.0439
## 2 CR.composite.before               -0.0439               1
```

```r
## Create correlation plot between predictor variables
corrplot(cor(VFlet_Ncorrect_coded[, c(13, 14)]))
```

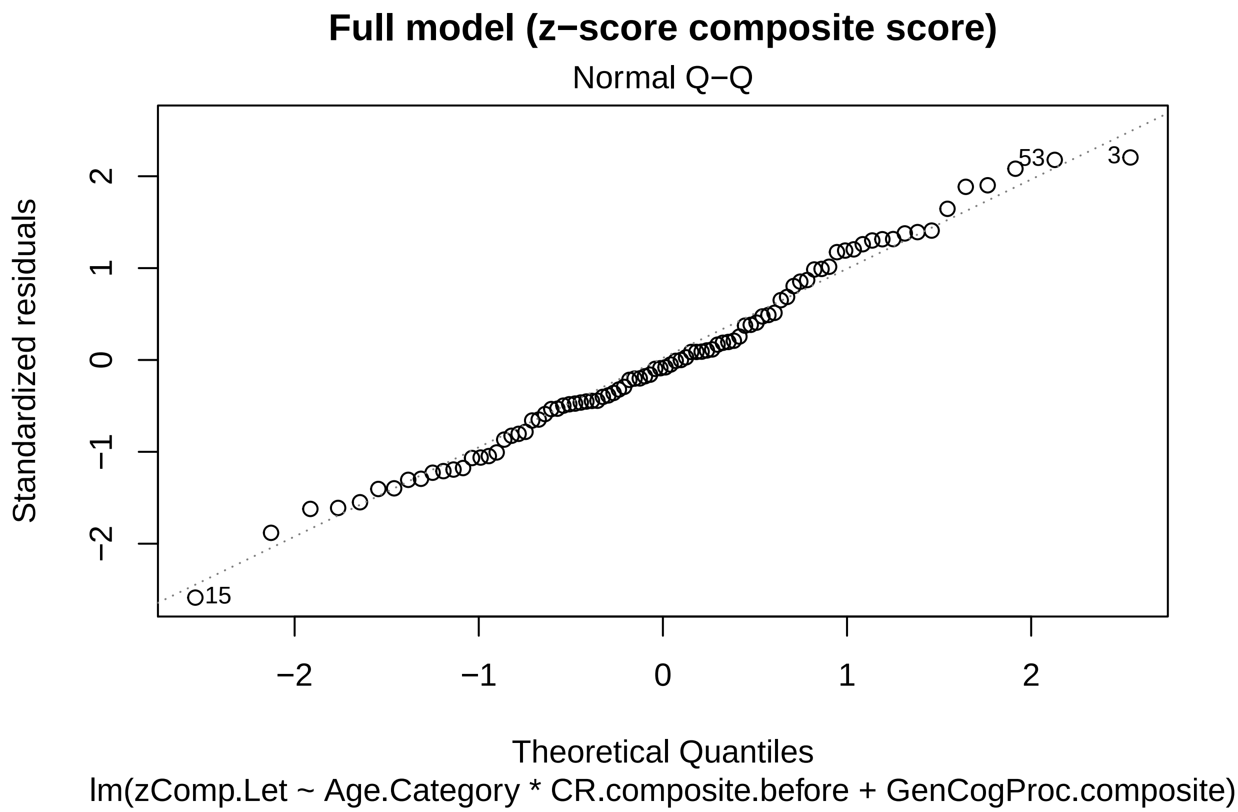


There seems to be no strong relationship between the predictor variables. Hence, we can assume Independence of Variables.

Assumption 3 - Normal Distribution of Residuals

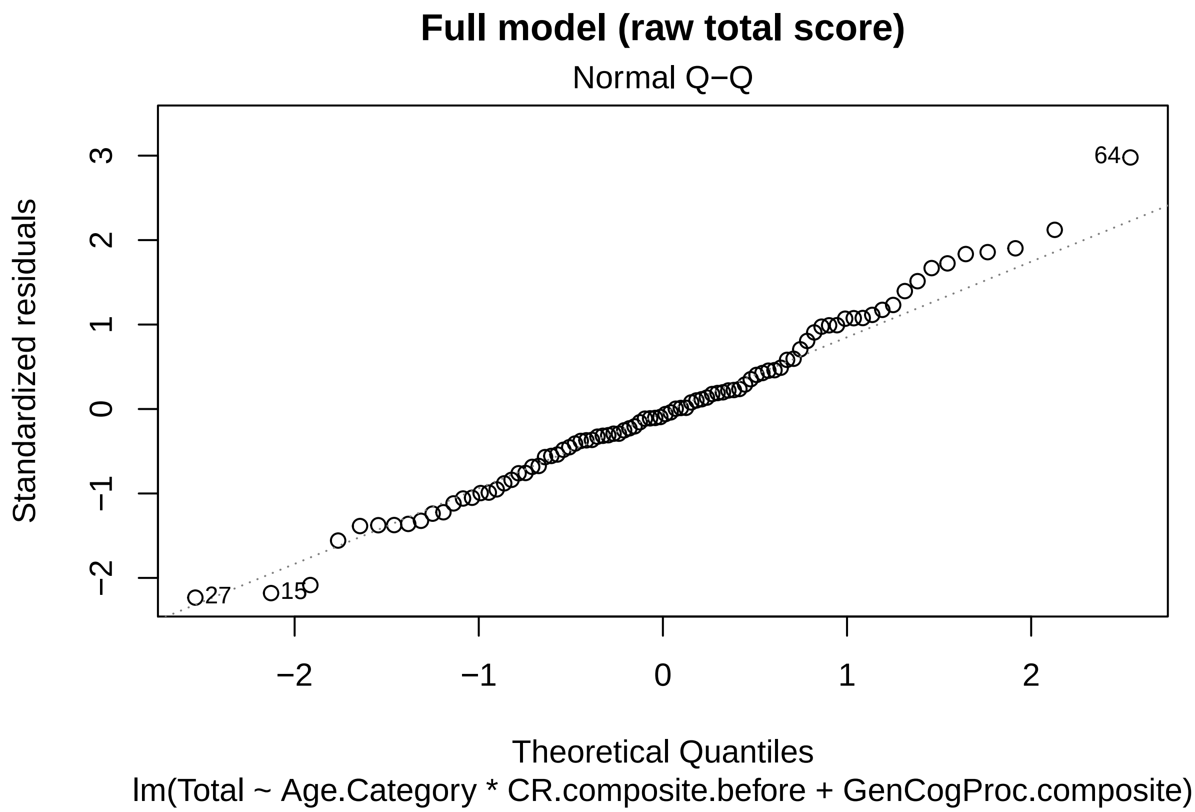
Full model with z composite score

```
plot(lmFull.VFlet.Ncorrect, 2, main = "Full model (z-score composite score)")
```



Full model with raw total score

```
plot(lmFull.VFlet.Ncorrect.raw, 2, main = "Full model (raw total score)")
```

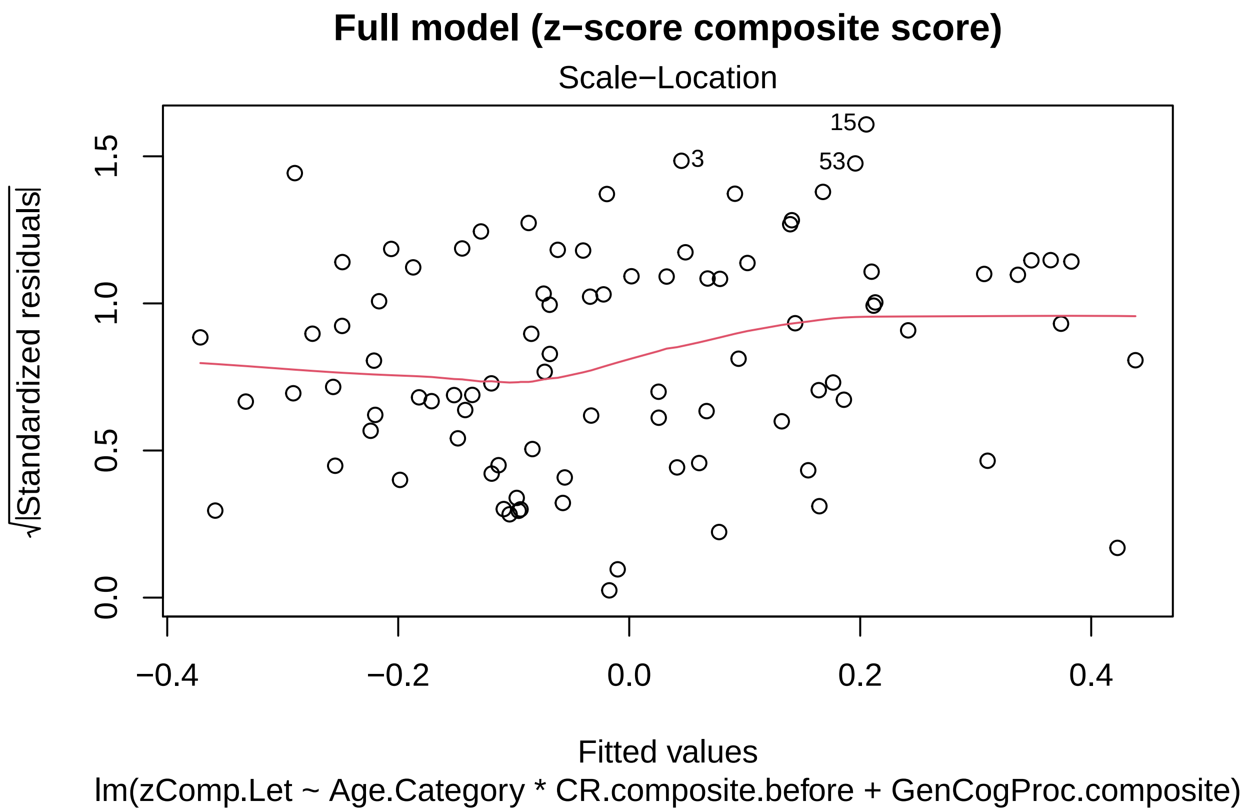


For all models, the points seem to roughly follow a straight line, except for a few points on the left and right. Hence, other relationships/predictors that have not been included into the models could explain the variance. This could be caused by outliers in the data.

Assumption 4 - Homoscedasticity or Equal Variance of Variables

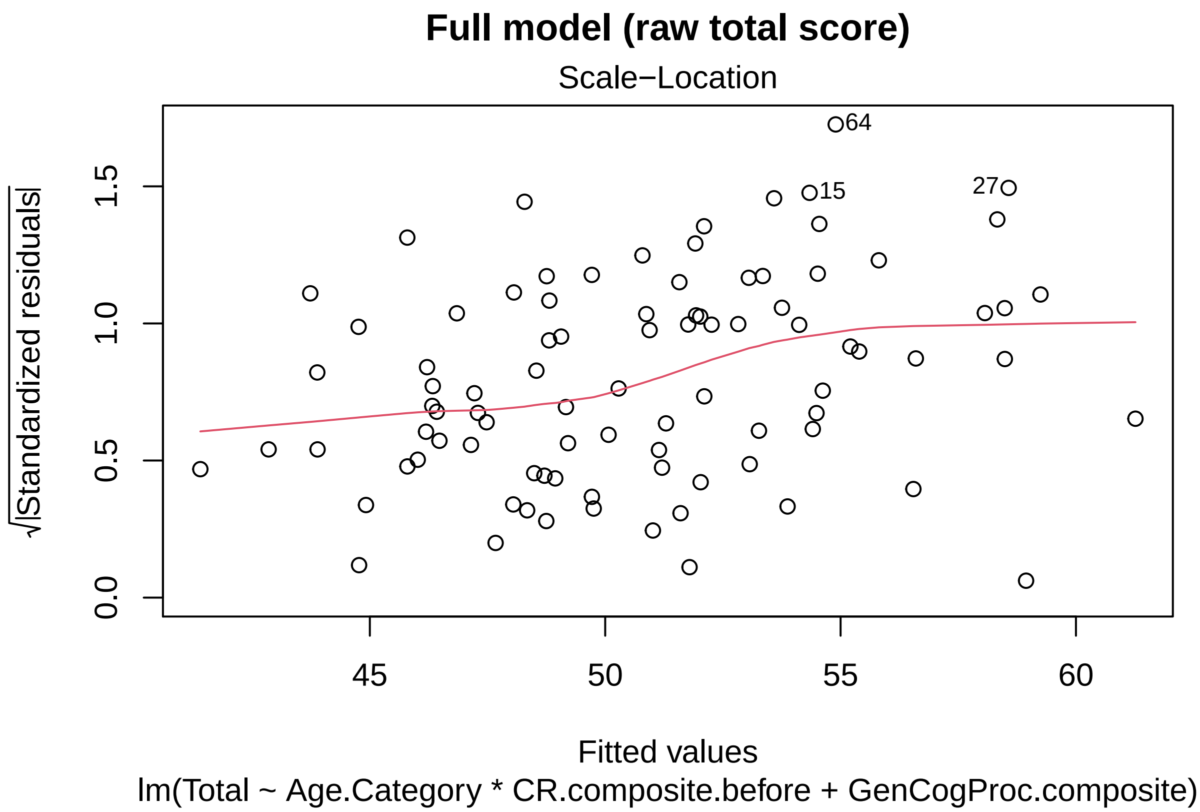
Full model with z composite score

```
plot(lmFull.VFlet.Ncorrect, 3, main = "Full model (z-score composite score)")
```



Full model with raw total score

```
plot(lmFull.VFlet.Ncorrect.raw, 3, main = "Full model (raw total score)")
```



For the unconditional model, the variance of residuals seems relatively equal across the predictors. Hence, the error terms are relatively the same across all values of the independent variable for this model. However, the variance of residuals doesn't seem quite equally distributed across the predictors. So, for the full models (z-score and raw score), the spread is not entirely constant, hence, the error terms does not appear to be the same across all values of the outcome variable.

Model fit diagnostics *Variation Inflation Factor*

Any VIF value above 4 needs further investigation. There seems to be no concerning signs of collinearity (VIF values higher than 4). The tolerance values indicate the percentage of variance that cannot be explained for by the other predictor variables.

```
ols_vif_tol(lmFull.VFlet.Ncorrect)
```

```
##                          Variables Tolerance      VIF
## 1                     Age.Category1 0.8958833 1.116217
## 2                     Age.Category2 0.7518193 1.330107
## 3               CR.composite.before 0.9971961 1.002812
## 4              GenCogProc.composite 0.6869407 1.455730
## 5 Age.Category1:CR.composite.before 0.9935202 1.006522
## 6 Age.Category2:CR.composite.before 0.9999277 1.000072
```

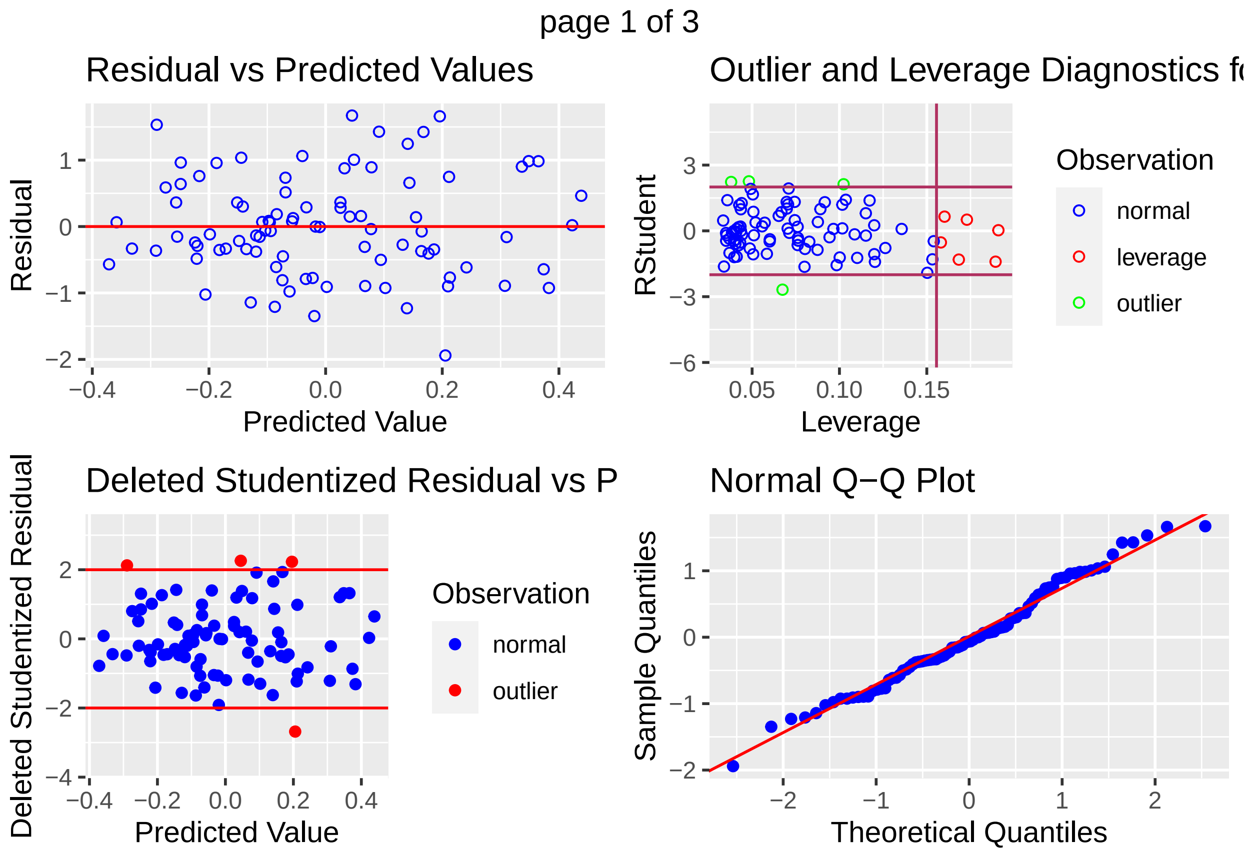
```
ols_vif_tol(lmFull.VFlet.Ncorrect.raw)
```

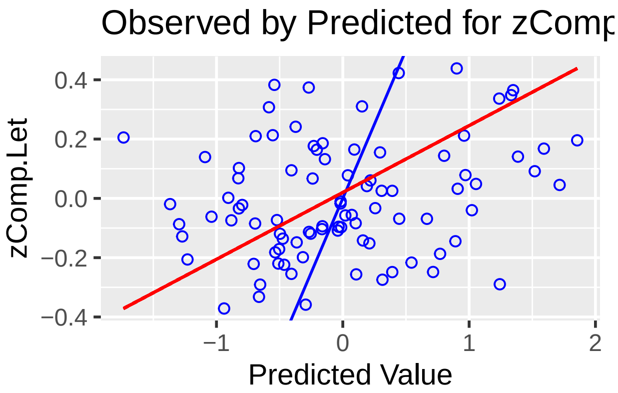
```
##                          Variables Tolerance      VIF
```

```
## 1                      Age.Category1 0.8958833 1.116217
## 2                      Age.Category2 0.7518193 1.330107
## 3                CR.composite.before 0.9971961 1.002812
## 4               GenCogProc.composite 0.6869407 1.455730
## 5 Age.Category1:CR.composite.before 0.9935202 1.006522
## 6 Age.Category2:CR.composite.before 0.9999277 1.000072
```

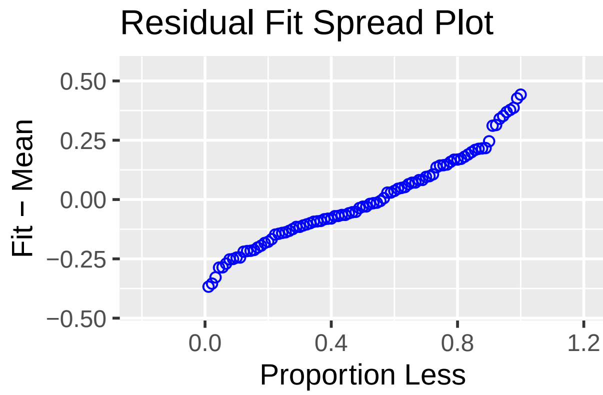
Plot Diagnositcs Full model with z composite score for Semantic Fluency

```
ols_plot_diagnostics(lmFull.VFlet.Ncorrect)
```

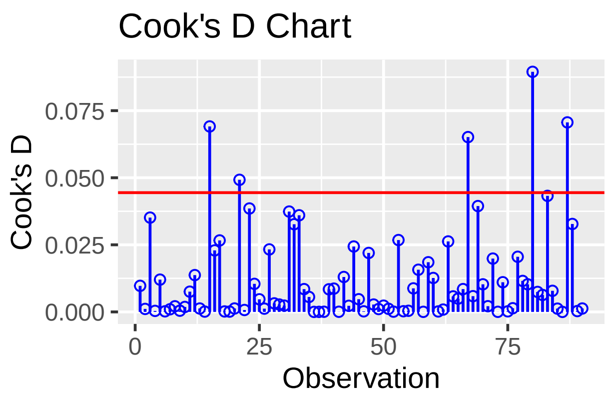




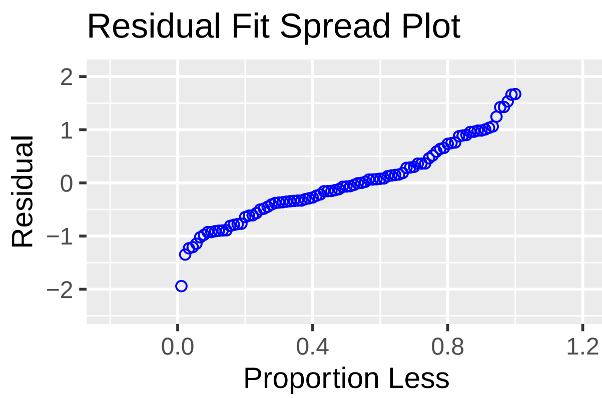
Observed by Predicted for zComp
zComp.Let
Predicted Value



Residual Fit Spread Plot
Fit – Mean
Proportion Less

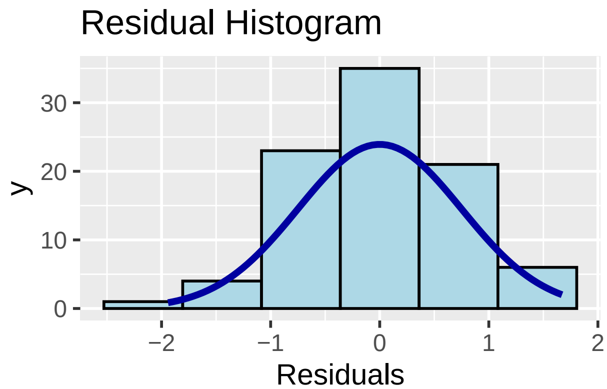


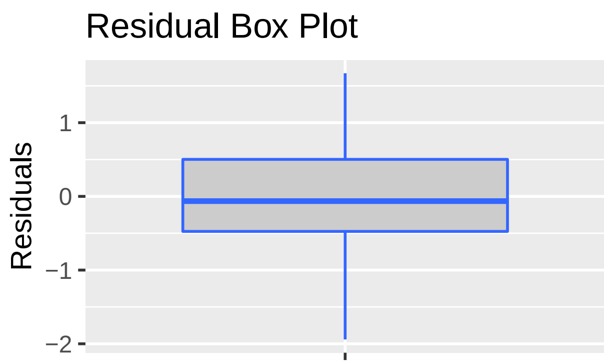
Cook's D Chart
Cook's D
Observation



Residual Fit Spread Plot
Residual
Proportion Less

page 3 of 3

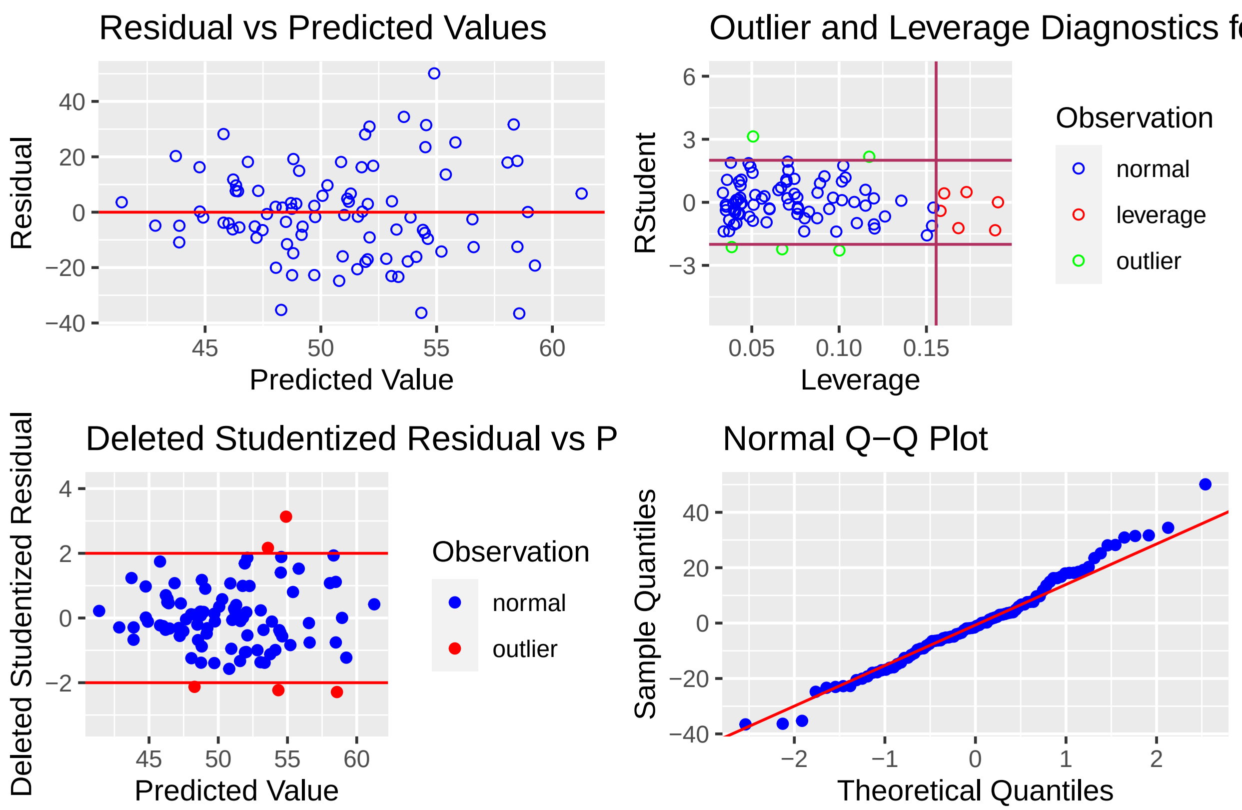


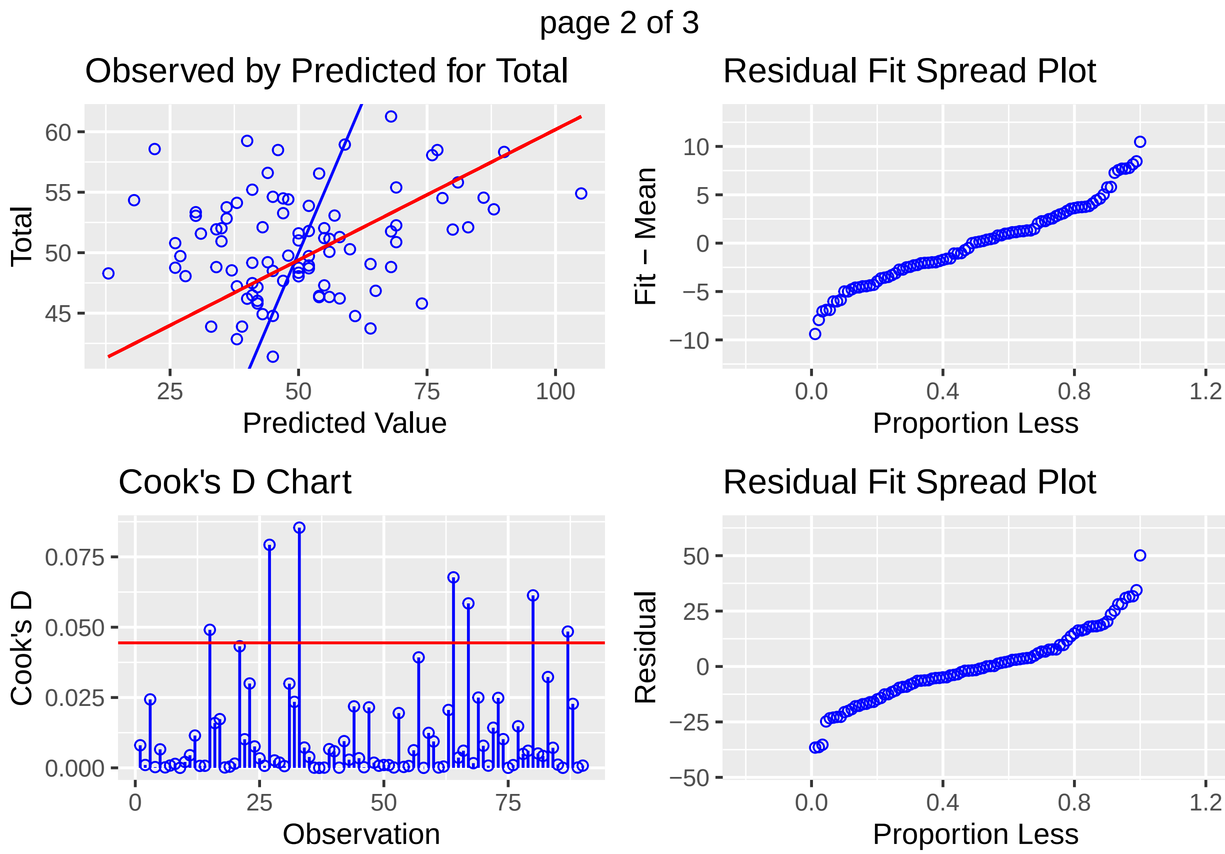


Plot Diagnositcs Full model with raw score for Semantic Fluency

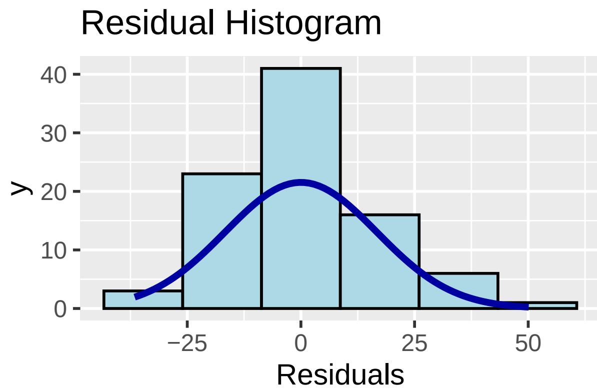
```
ols_plot_diagnostics(lmFull.VFlet.Ncorrect.raw)
```

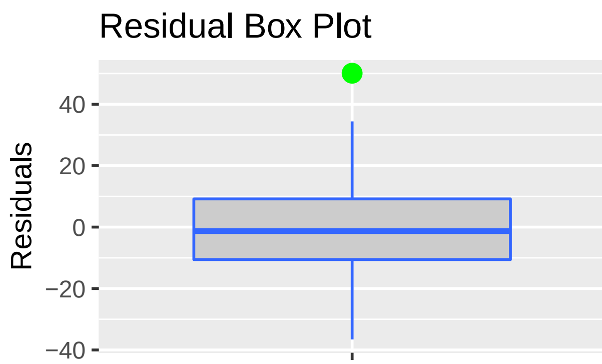
page 1 of 3


Residual vs Predicted Values
Residual
Predicted Value
Outlier and Leverage Diagnostics fo
RStudent
Leverage
Observation
normal
leverage
outlier
Deleted Studentized Residual vs P
Deleted Studentized Residual
Predicted Value
Observation
normal
outlier
Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles


page 2 of 3
Observed by Predicted for Total
Total
Predicted Value
Residual Fit Spread Plot
Fit − Mean
Proportion Less
Cook's D Chart
Cook's D
Observation
Residual Fit Spread Plot
Residual
Proportion Less

page 3 of 3





For all models, the Observed vs Predicted Plot shows that the model doesn't fit the data very well (pity) –> due to outliers?? What to do.... –> perhaps identify outliers using the Cook chart and residual plot and rerun the models?

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

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

```
lmFull.VFlet.Ncorrect.during <- lm(zComp.Let ~ Age.Category *
    CR.composite.during + GenCogProc.composite, data = VFlet_Ncorrect_coded)
summary(lmFull.VFlet.Ncorrect.during)
```

```
##
## Call:
## lm(formula = zComp.Let ~ Age.Category * CR.composite.during +
##     GenCogProc.composite, data = VFlet_Ncorrect_coded)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.87069 -0.45660 -0.04052  0.47506  1.71393
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         -0.002566   0.081696  -0.031   0.9750
## Age.Category1                       -0.186063   0.105889  -1.757   0.0826 .
```

```
## Age.Category2                        0.031746   0.066873   0.475   0.6362
## CR.composite.during                  0.130513   0.084014   1.553   0.1241
## GenCogProc.composite                 0.226481   0.192272   1.178   0.2422
## Age.Category1:CR.composite.during    0.040434   0.103611   0.390   0.6974
## Age.Category2:CR.composite.during    0.009143   0.059678   0.153   0.8786
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.775 on 83 degrees of freedom
## Multiple R-squared:  0.06677,    Adjusted R-squared:  -0.0006978
## F-statistic: 0.9897 on 6 and 83 DF,  p-value: 0.4377
```

```
# Model comparisons
anova(lmFull.VFlet.Ncorrect, lmFull.VFlet.Ncorrect.during)
```

```
## Analysis of Variance Table
##
## Model 1: zComp.Let ~ Age.Category * CR.composite.before + GenCogProc.composite
## Model 2: zComp.Let ~ Age.Category * CR.composite.during + GenCogProc.composite
##   Res.Df    RSS Df Sum of Sq F Pr(>F)
## 1     83 50.103
## 2     83 49.852  0   0.25134
```

No differences between the two models

```
# Model comparisons through AIC values
AIC(lmFull.VFlet.Ncorrect)
```

```
## [1] 218.6932
```

```
AIC(lmFull.VFlet.Ncorrect.during)
```

```
## [1] 218.2406
```

The model with the composite score during Covid-19 seems to fit slightly better

Verbal Fluency - Actions

Descriptive Statistics

```
## Create dataset that only includes data of the Number of Correct words Produced
VFact_Ncorrect <- VFact %>%
  dplyr::filter(Measures=="Ncorrect") %>%
  dplyr::select(-Measures) %>%
      #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"))) %>%
    #Use the z scores to filter out outliers (i.e., exclude values +/-2.5 SD per trial )
```

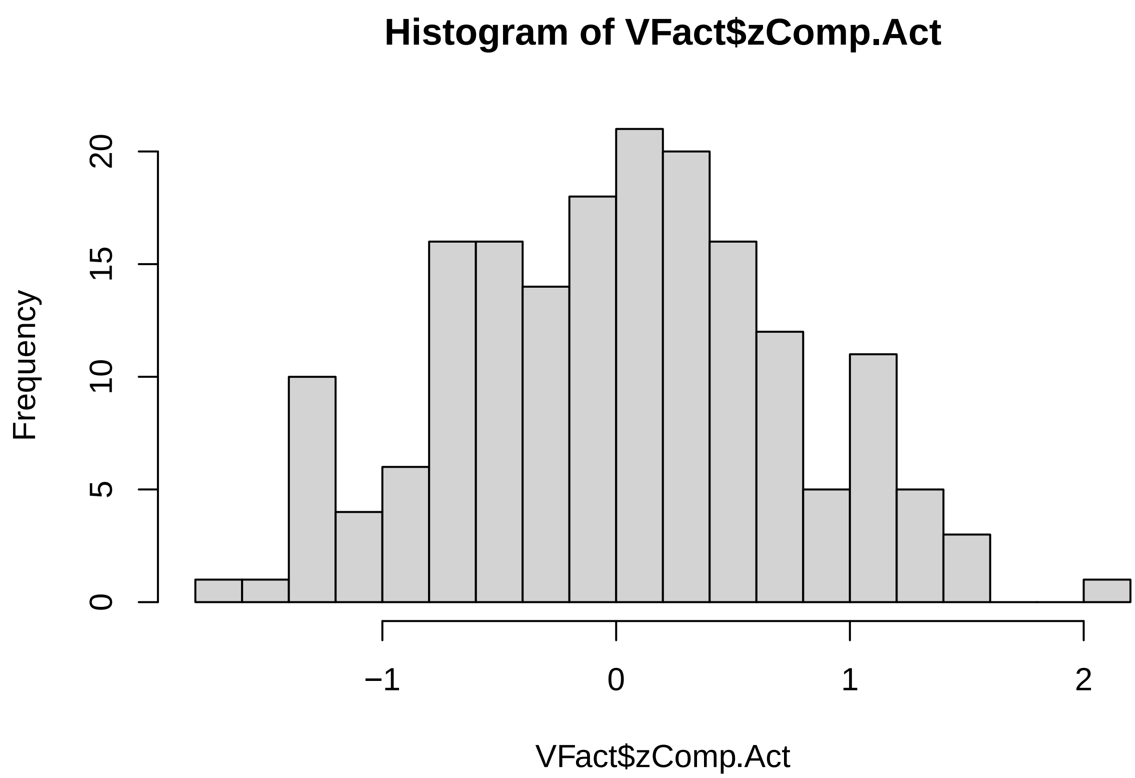
```
  filter(between(zComp.Act, -2.5, +2.5)) %>% #No outliers
  ungroup()

# head(VFact_Ncorrect_coded, L=6)
```

```
#Summarise data to present descriptives in a table
(Descr_VFact <- VFact_Ncorrect %>%
#Per Age Group
  group_by(Age.Category) %>%
  summarise(Nppt = length(unique(ID)),
            total = round(mean(Total, na.rm=T),2),
            sdtotal = round(sd(Total,na.rm=T),2),
            # ztotal = mean(zComp.Act, na.rm=T),
            people = round(mean(Things.people.do, na.rm=T),2),
            eggs = round(mean(Egg, na.rm=T),2)))
```

```
## # A tibble: 3 x 6
##   Age.Category  Nppt total sdtotal people  eggs
##   <fct>        <int> <dbl>   <dbl>  <dbl> <dbl>
## 1 Middle-Aged     30  35.9     9.3   22.4  13.4
## 2 Older           30  32.9     6.4   19.9  13.0
## 3 Younger         30  35.7     7.3   22.4  13.2
```

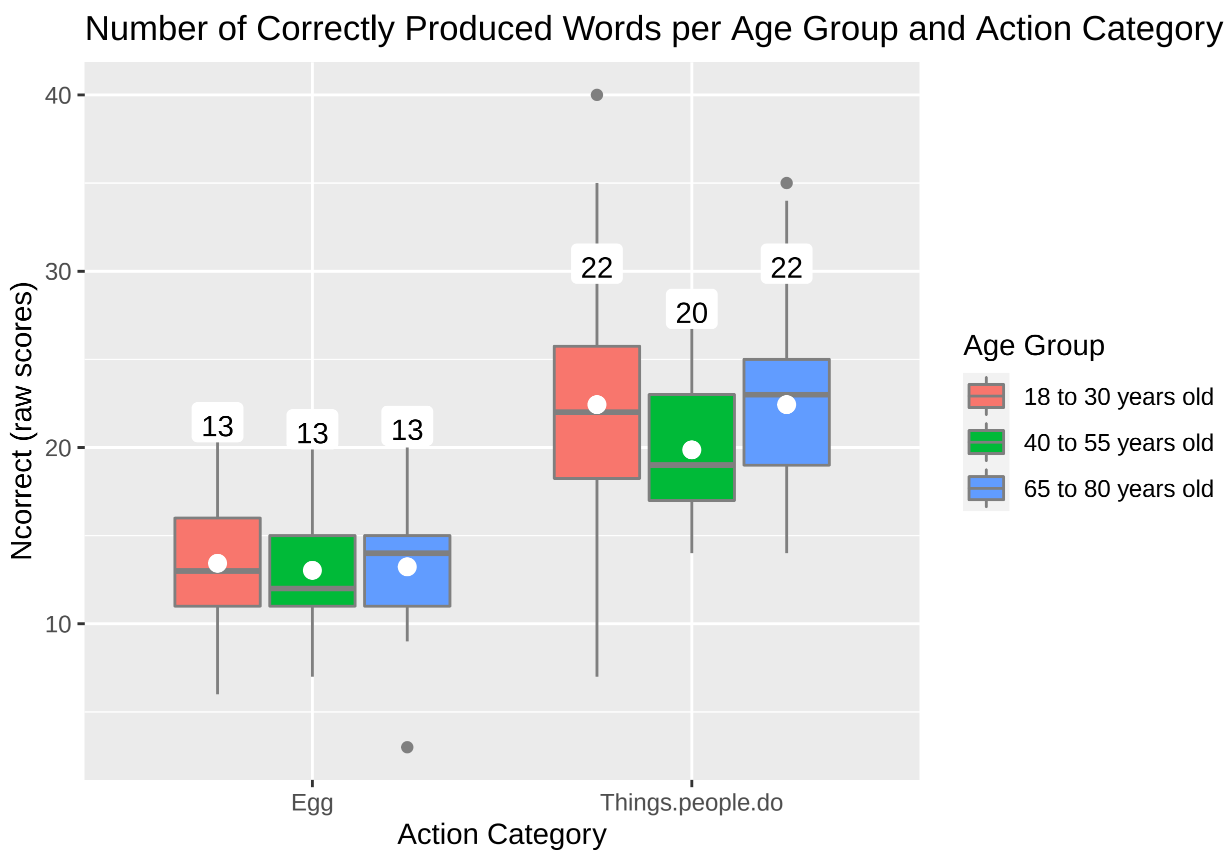
```
# write.csv(Descr_VFact, "./Figures and Tables/Descr_VFact_Ncorrect.csv", row.names = F)
```

```
# z-distribution of composite score for Action fluency
hist(VFact$zComp.Act, breaks = 20)
```



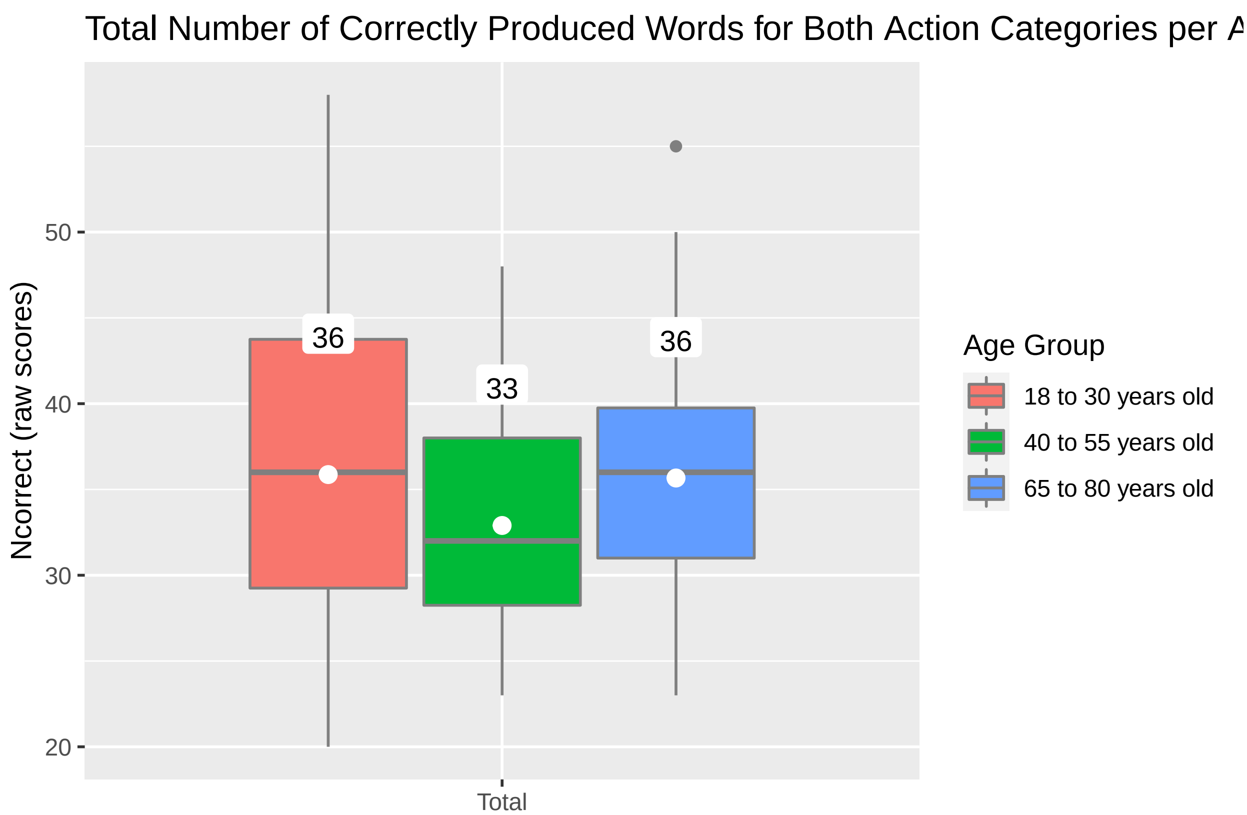
Visualisation Ncorrect Semantic Fluency

```
## Convert wide to long format for visualisation of data
VFact_Ncorrect.long <- VFact_Ncorrect %>%
  pivot_longer(cols=Total:Egg, names_to = "action", values_to = "Ncorrect")

#Boxplot VFact Ncorrect
# png(file="./Figures and Tables/Boxplot_VFactNcorrect.png",
# width=600, height=350)

(Boxplot_VF <- VFact_Ncorrect.long %>%
    dplyr::filter(action!="Total") %>%
    ggplot(aes(x=factor(action), y=Ncorrect,
                                                fill = as.factor(Age.Category))) +
    geom_boxplot(colour="grey50")+
    stat_summary(aes(label=round(..y..), group=as.factor(Age.Category)),
               fun=mean, geom = "label", size=4,
               fill="white", show.legend=NA, label.size=NA,
               position = position_dodge(.75), vjust=-3) +
      stat_summary(fun = "mean", position = position_dodge(.75),
               show.legend=F, colour="white")+ #Mean as white dot
    labs(x = "Action Category",
         y = "Ncorrect (raw scores)",
         title = "Number of Correctly Produced Words per Age Group and Action Category")+
        scale_fill_discrete(guide=guide_legend(title = "Age Group"), labels=c("18 to 30 years old","40 
```




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

```r
# Boxplot VFact Ncorrect Raw Total
# png(file="./Figures and Tables/Boxplot_VFactNcorrect_RawTotal.png",
# width=600, height=350)

(Boxplot_VF <- VFact_Ncorrect.long %>%
    dplyr::filter(action=="Total") %>%
    ggplot(aes(x=action, y=Ncorrect,fill = as.factor(Age.Category))) +
    geom_boxplot(colour="grey50")+
    stat_summary(aes(label=round(..y..), group=as.factor(Age.Category)),
                 fun=mean, geom = "label", size=4,
                 fill="white", show.legend=NA, label.size=NA,
                 position = position_dodge(.75), vjust=-3) +
      stat_summary(fun = "mean", position = position_dodge(.75),
                 show.legend=F, colour="white")+ #Mean as white dot
    labs(x = "",
         y = "Ncorrect (raw scores)",
         title = "Total Number of Correctly Produced Words for Both Action Categories per Age Group")+
        scale_fill_discrete(guide=guide_legend(title = "Age Group"), labels=c("18 to 30 years old","40 t
```

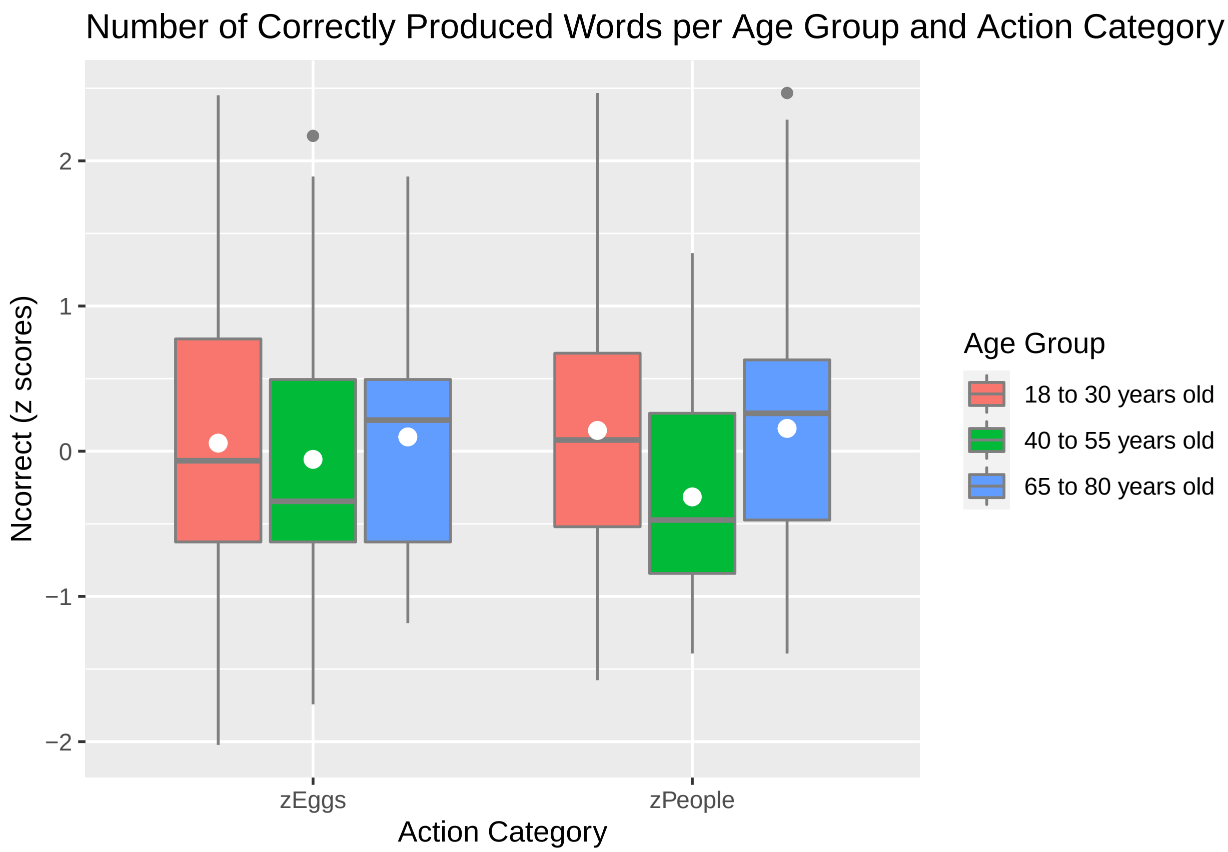



```
# dev.off()
```

Figures Ncorrect Action Fluency z-scores

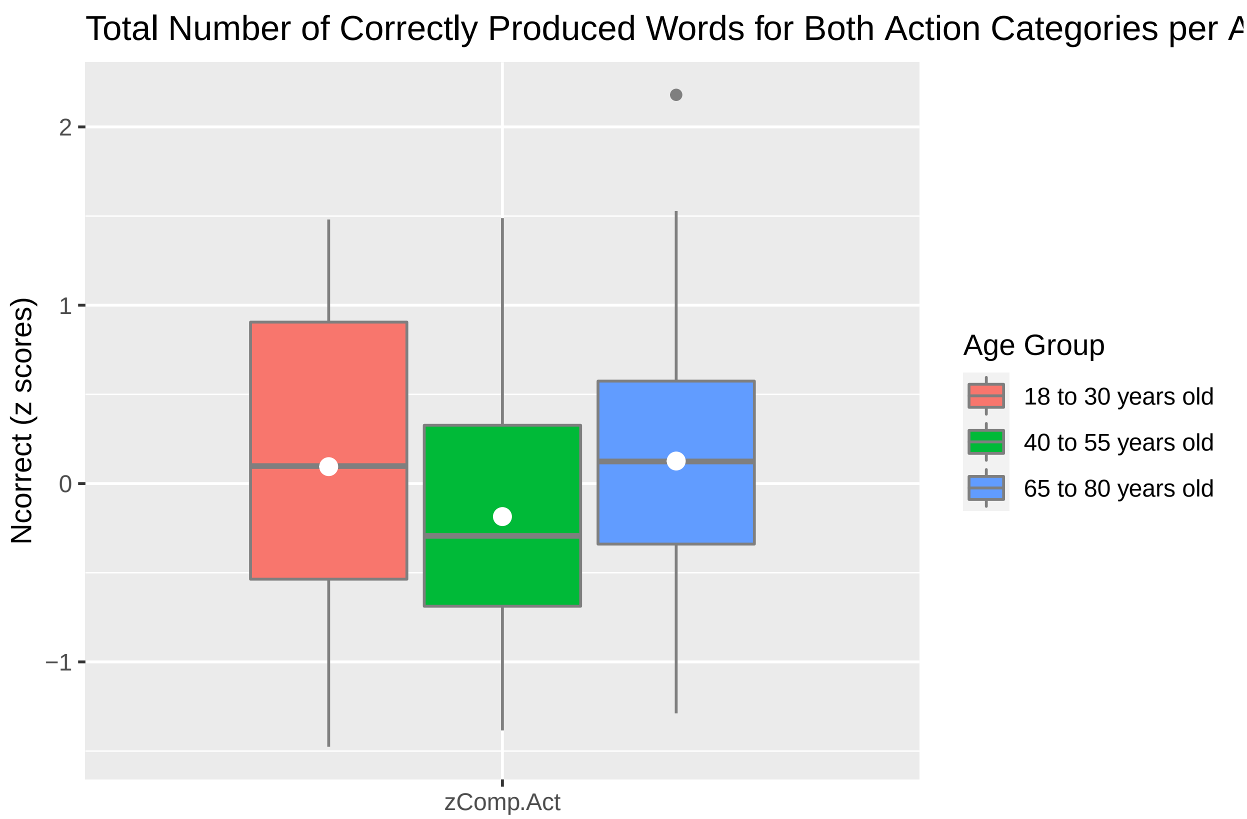
```
## Convert wide to long format for visualisation of data
VFact_Ncorrect.long.zScores <- VFact_Ncorrect %>%
  pivot_longer(cols=zComp.Act:zEggs, names_to = "zaction", values_to = "zNcorrect")

#Boxplot VFact Ncorrect
# png(file="./Figures and Tables/Boxplot_VFactNcorrect_zscores.png",
# width=600, height=350)

(Boxplot_VF <- VFact_Ncorrect.long.zScores %>%
    dplyr::filter(zaction!="zComp.Act") %>%
    ggplot(aes(x=factor(zaction), y=zNcorrect,
                                                fill = as.factor(Age.Category))) +
    geom_boxplot(colour="grey50")+
    # stat_summary(aes(label=round(..y..), group=as.factor(Age.Category)),
    #             fun=mean, geom = "label", size=4,
    #             fill="white", show.legend=NA, label.size=NA,
    #             position = position_dodge(.75), vjust=-3) +
      stat_summary(fun = "mean", position = position_dodge(.75),
               show.legend=F, colour="white")+ #Mean as white dot
    labs(x = "Action Category",
         y = "Ncorrect (z scores)",
```

```
        title = "Number of Correctly Produced Words per Age Group and Action Category")+
       scale_fill_discrete(guide=guide_legend(title = "Age Group"), labels=c("18 to 30 years old","40 
```




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

```
# Boxplot VFact Ncorrect Raw Total
# png(file="./Figures and Tables/Boxplot_VFactNcorrect_Total_zscores.png",
# width=600, height=350)

(Boxplot_VF <- VFact_Ncorrect.long.zScores %>%
    dplyr::filter(zaction=="zComp.Act") %>%
    ggplot(aes(x=zaction, y=zNcorrect,fill = as.factor(Age.Category))) +
    geom_boxplot(colour="grey50")+
    # stat_summary(aes(label=round(..y..), group=as.factor(Age.Category)),
    #             fun=mean, geom = "label", size=4,
    #             fill="white", show.legend=NA, label.size=NA,
    #             position = position_dodge(.75), vjust=-3) +
      stat_summary(fun = "mean", position = position_dodge(.75),
               show.legend=F, colour="white")+ #Mean as white dot
    labs(x = "",
        y = "Ncorrect (z scores)",
        title = "Total Number of Correctly Produced Words for Both Action Categories per Age Group")+
       scale_fill_discrete(guide=guide_legend(title = "Age Group"), labels=c("18 to 30 years old","40 
```




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

Multiple Linear Regression - Action Fluency

Create user-defined contrasts for the Age Category variable

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

VFact_Ncorrect_coded <- VFact_Ncorrect
contrasts(VFact_Ncorrect_coded$Age.Category) <- contr.helmert(3)
contrasts(VFact_Ncorrect_coded$Age.Category)
```

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

Unconditional model, i.e. without covariates

```
lmUncond.VFact.Ncorrect <- lm(zComp.Act ~ Age.Category * CR.composite.before,
    data = VFact_Ncorrect_coded)
# broom::tidy(lmUncond.VFact.Ncorrect, conf.int=T)
```

Full model including the covariates; outcome variable in z-distribution

```
lmFull.VFact.Ncorrect <- lm(zComp.Act ~ Age.Category * CR.composite.before +
    GenCogProc.composite, data = VFact_Ncorrect_coded)
summary(lmFull.VFact.Ncorrect)
```

```
##
## Call:
## lm(formula = zComp.Act ~ Age.Category * CR.composite.before +
##     GenCogProc.composite, data = VFact_Ncorrect_coded)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56751 -0.49290 -0.02566  0.42847  1.87434
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       0.01355    0.08121   0.167   0.8679
## Age.Category1                    -0.06054    0.10507  -0.576   0.5660
## Age.Category2                    -0.02436    0.06622  -0.368   0.7139
## CR.composite.before               0.10692    0.08271   1.293   0.1997
## GenCogProc.composite              0.42392    0.18837   2.250   0.0271 *
## Age.Category1:CR.composite.before -0.13722    0.10148  -1.352   0.1800
## Age.Category2:CR.composite.before  0.06060    0.05840   1.038   0.3024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7704 on 83 degrees of freedom
## Multiple R-squared:  0.1255, Adjusted R-squared:  0.06233
## F-statistic: 1.986 on 6 and 83 DF,  p-value: 0.07682
```

```
# Tidy table output
(tidylmFull.VFact.Ncorrect <- broom::tidy(lmFull.VFact.Ncorrect,
    conf.int = T) %>%
    mutate_if(is.numeric, round, 3))
```

```
## # A tibble: 7 x 7
##   term                    estimate std.error statistic p.value conf.low conf.high
##   <chr>                      <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)                0.014     0.081     0.167   0.868   -0.148     0.175
## 2 Age.Category1             -0.061     0.105    -0.576   0.566   -0.27      0.148
## 3 Age.Category2             -0.024     0.066    -0.368   0.714   -0.156     0.107
## 4 CR.composite.before        0.107     0.083     1.29    0.2     -0.058     0.271
## 5 GenCogProc.composite       0.424     0.188     2.25    0.027    0.049     0.799
## 6 Age.Category1:CR.comp~    -0.137     0.101    -1.35    0.18    -0.339     0.065
## 7 Age.Category2:CR.comp~     0.061     0.058     1.04    0.302   -0.056     0.177
```

```
# write.csv(tidylmFull.VFact.Ncorrect, './Figures and
# Tables/VFact_Ncorrect_lmFull.csv')
```

```
# Look at pairwise comparisons between contrasts
lmFull.VFact.Ncorrect.emmeans <- emtrends(lmFull.VFact.Ncorrect,
```

```r
    ~Age.Category, var = "CR.composite.before")
pairs(lmFull.VFact.Ncorrect.emmeans)
```

```
##  contrast                 estimate    SE df t.ratio p.value
##  (Middle-Aged) - Younger   0.2744 0.203 83   1.352  0.3707
##  (Middle-Aged) - Older    -0.0446 0.203 83  -0.220  0.9736
##  Younger - Older          -0.3190 0.202 83  -1.576  0.2617
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

Full model including the covariates; outcome variable as raw score

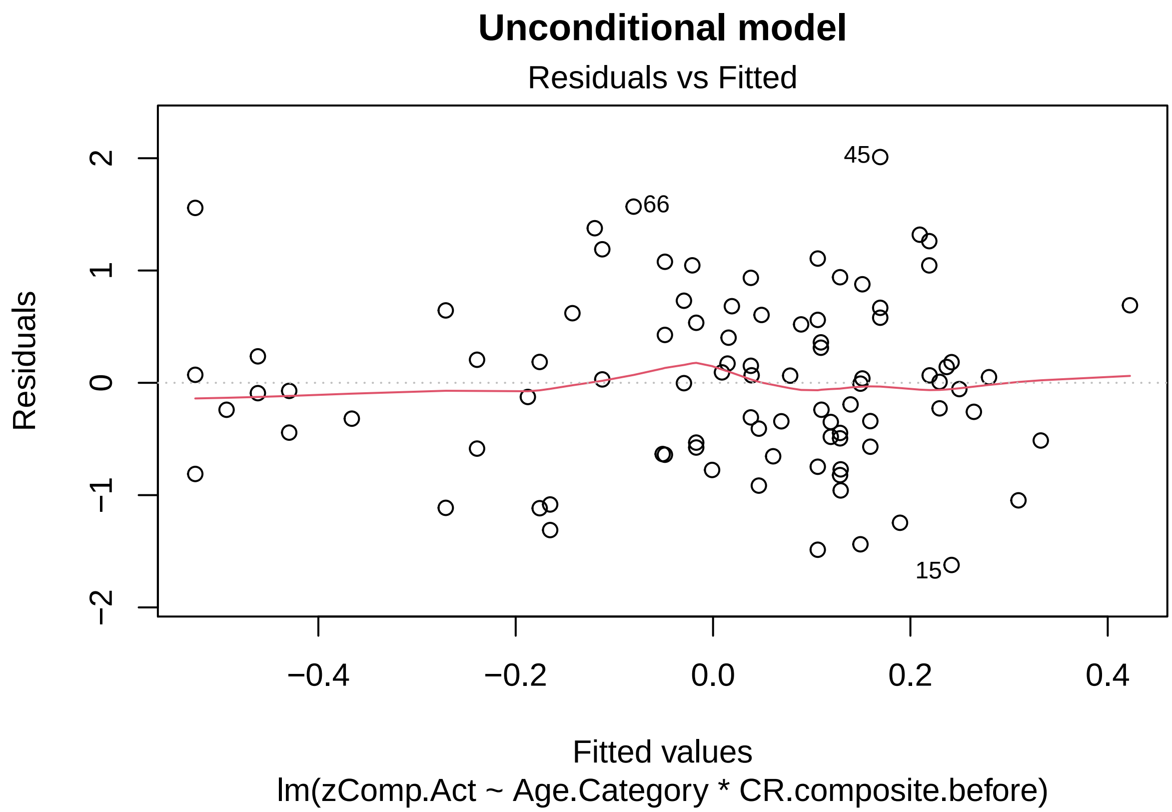
```r
lmFull.VFact.Ncorrect.raw <- lm(Total ~ Age.Category * CR.composite.before +
    GenCogProc.composite, data = VFact_Ncorrect_coded)
# Tidy table output
broom::tidy(lmFull.VFact.Ncorrect.raw, conf.int = T) %>%
    mutate_if(is.numeric, round, 3)
```

```
## # A tibble: 7 x 7
##   term                   estimate std.error statistic p.value conf.low conf.high
##   <chr>                     <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)              34.8       0.791    44.0     0        33.3      36.4
## 2 Age.Category1            -0.875     1.02     -0.854   0.395    -2.91      1.16
## 3 Age.Category2            -0.202     0.645    -0.313   0.755    -1.48      1.08
## 4 CR.composite.before       1.12      0.806     1.39    0.169    -0.484     2.72
## 5 GenCogProc.composite      4.30      1.84      2.35    0.021     0.653     7.95
## 6 Age.Category1:CR.comp~   -1.67      0.989    -1.69    0.095    -3.64      0.296
## 7 Age.Category2:CR.comp~    0.484     0.569     0.851   0.397    -0.647     1.62
```

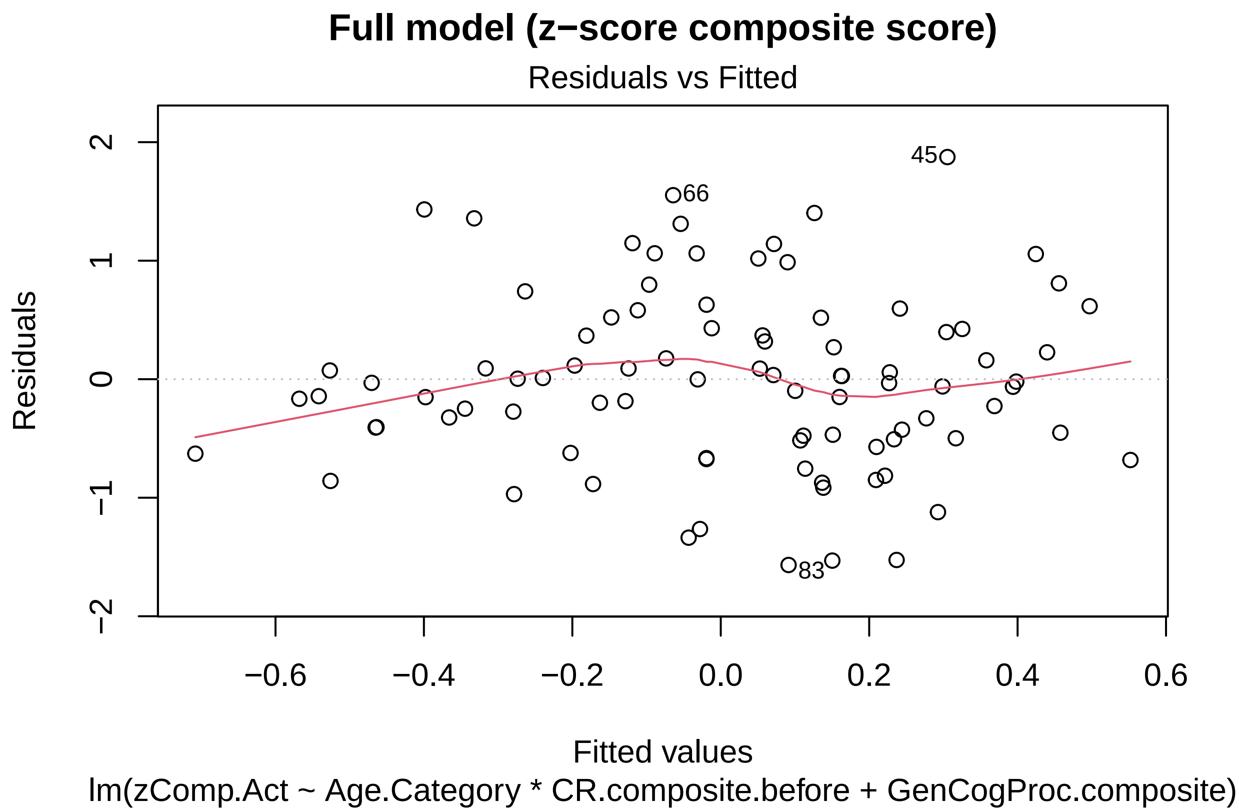
The model doesn't seem to predict either the composite score (z-distribution) or the raw Total score for Verbal Fluency Action Categories. Let's check the model assumptions + fit.

Assumption 1 - Linearity

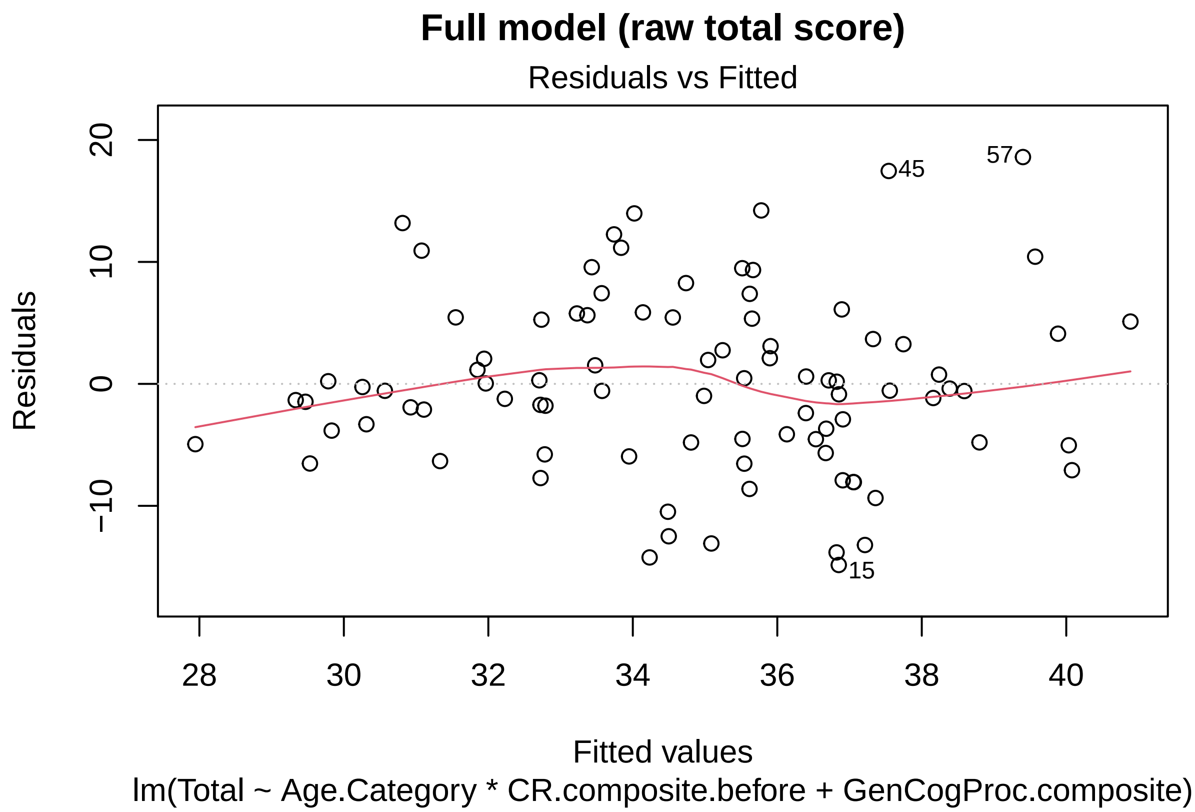
```r
## Unconditional model with z composite score
plot(lmUncond.VFact.Ncorrect, 1, main = "Unconditional model")
```




```
## Full model with z composite score
plot(lmFull.VFact.Ncorrect, 1, main = "Full model (z-score composite score)")
```




```r
## Full model with raw total score
plot(lmFull.VFact.Ncorrect.raw, 1, main = "Full model (raw total score)")
```



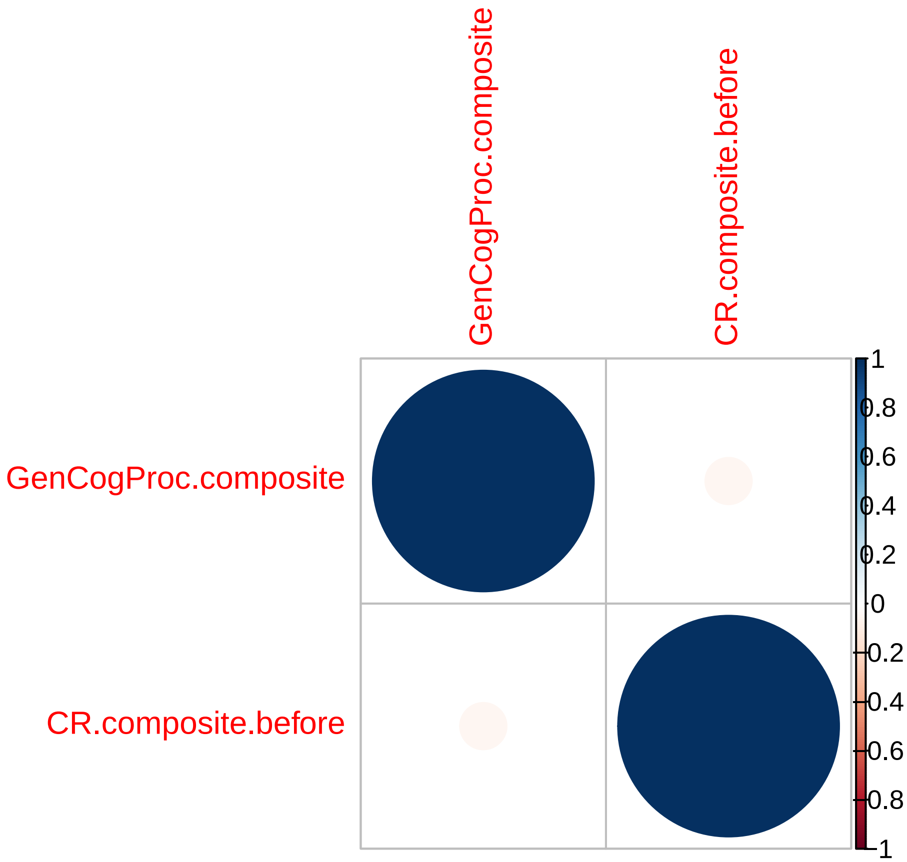
For all models, the red line is mostly horizontal, meaning that there is a linear relationship between the fitted line and the residual value.

Assumption 2 - Independence of Variables

```
# Z composite of VF cat Create table with correlation
# values between predictor variables
tibble::as_tibble(cor(VFact_Ncorrect_coded[, c(11, 13)]), rownames = "rowname")
```

```
## # A tibble: 2 x 3
##   rowname              GenCogProc.composite CR.composite.before
##   <chr>                               <dbl>               <dbl>
## 1 GenCogProc.composite               1                  -0.0439
## 2 CR.composite.before               -0.0439              1
```

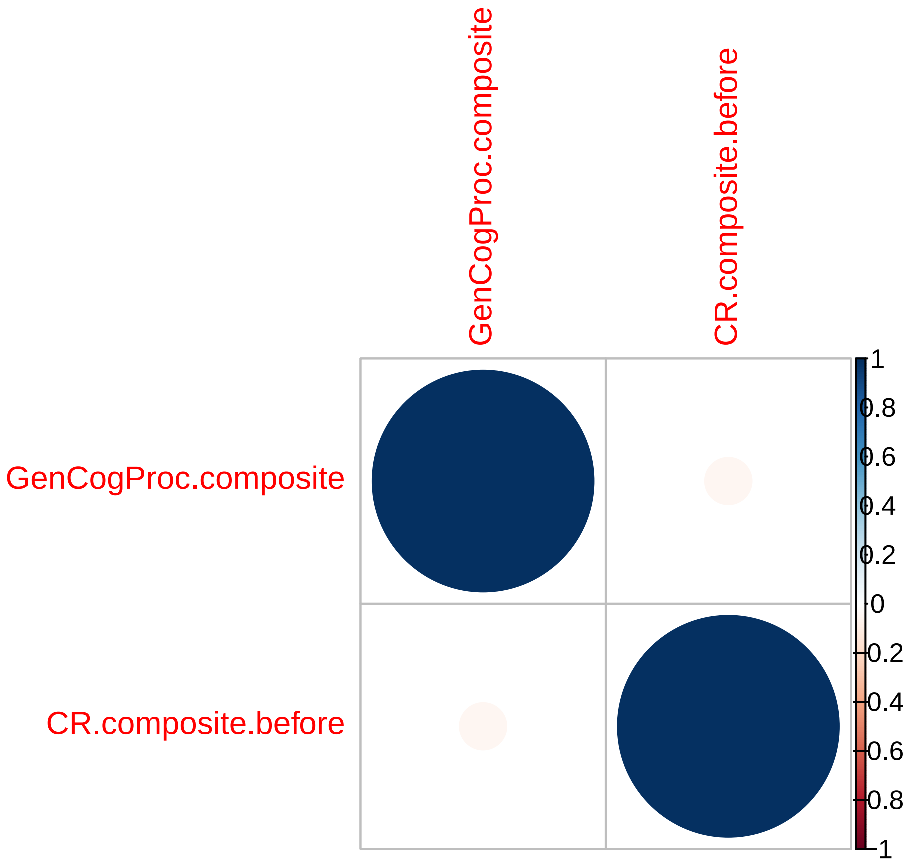
```
## Create correlation plot between predictor variables
corrplot(cor(VFact_Ncorrect_coded[, c(11, 13)]), method = "circle")
```




```r
## Raw scores Create table with correlation values between
## predictor variables
tibble::as_tibble(cor(VFact_Ncorrect_coded[, c(11, 13)]), rownames = "rowname")
```

```
## # A tibble: 2 x 3
##   rowname              GenCogProc.composite CR.composite.before
##   <chr>                               <dbl>               <dbl>
## 1 GenCogProc.composite               1                  -0.0439
## 2 CR.composite.before               -0.0439              1
```

```r
## Create correlation plot between predictor variables
corrplot(cor(VFact_Ncorrect_coded[, c(11, 13)]))
```

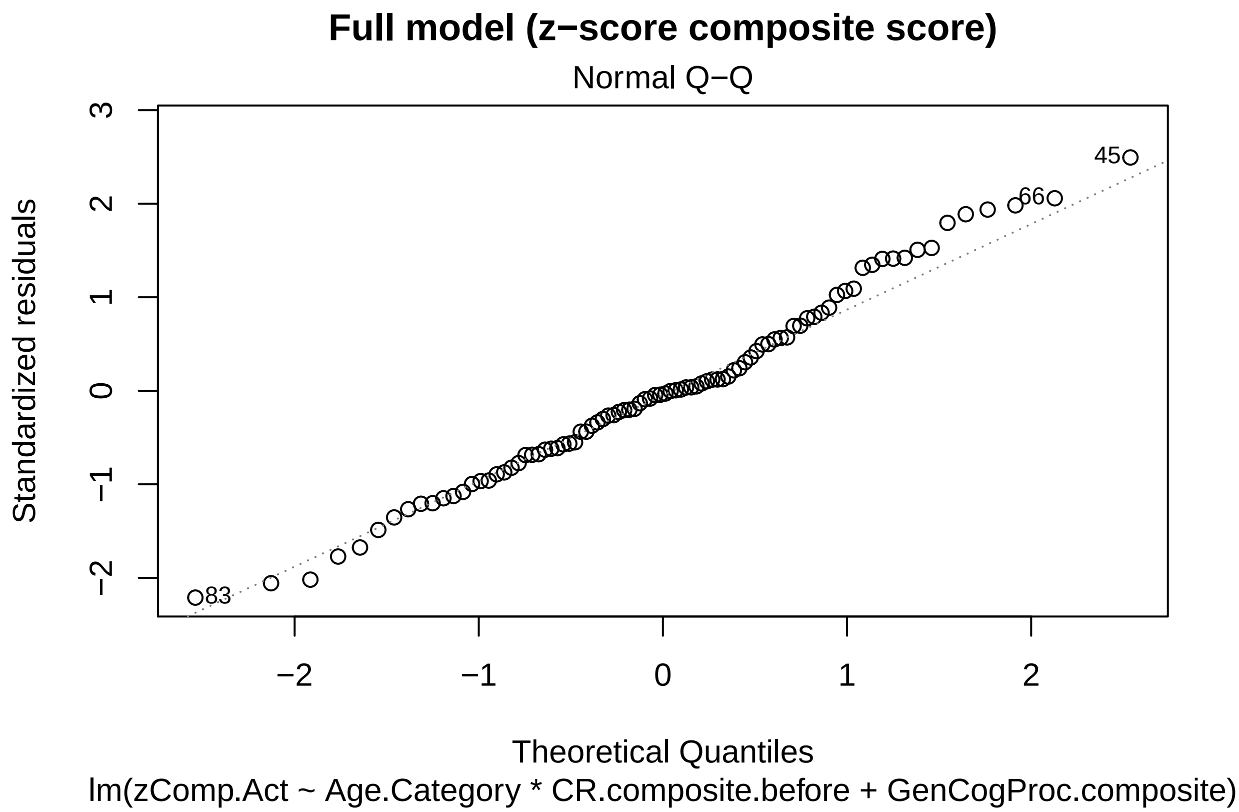


There seems to be no strong relationship between the predictor variables. Hence, we can assume Independence of Variables.

Assumption 3 - Normal Distribution of Residuals

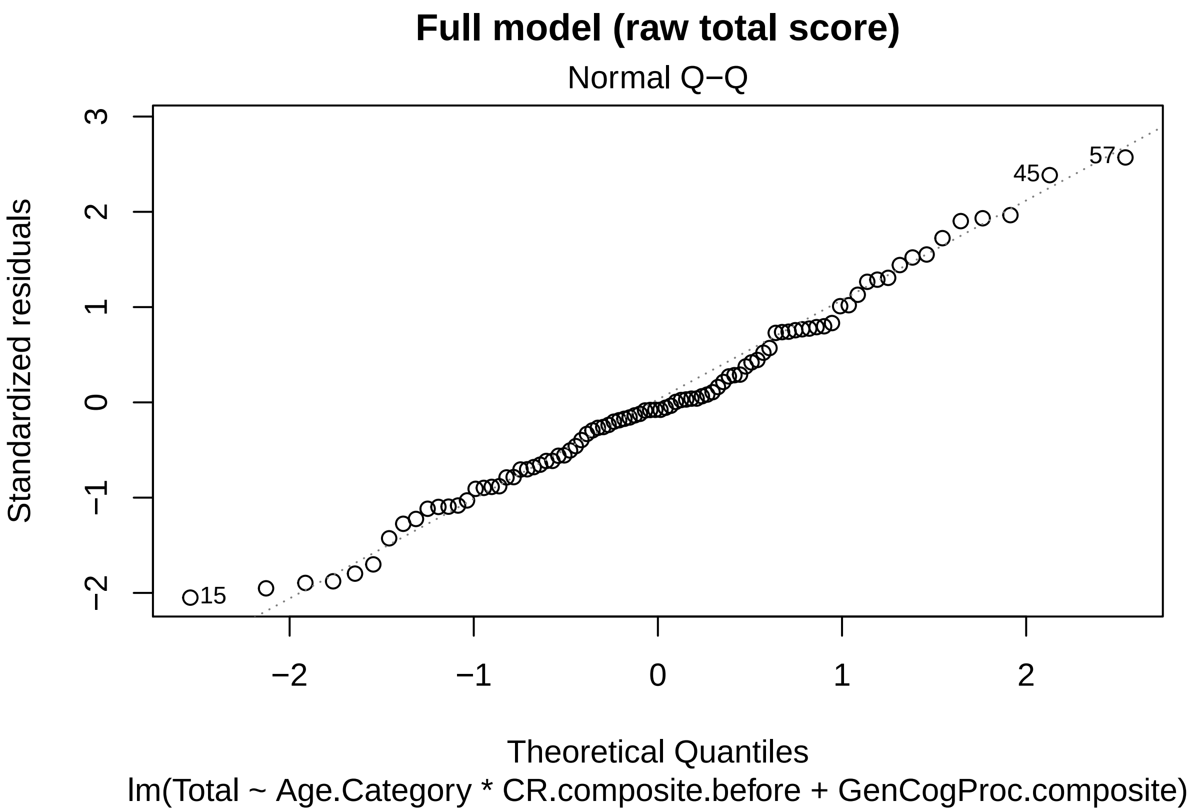
Full model with z composite score

```
plot(lmFull.VFact.Ncorrect, 2, main = "Full model (z-score composite score)")
```



Full model with raw total score

```
## Full model with raw total score
plot(lmFull.VFact.Ncorrect.raw, 2, main = "Full model (raw total score)")
```

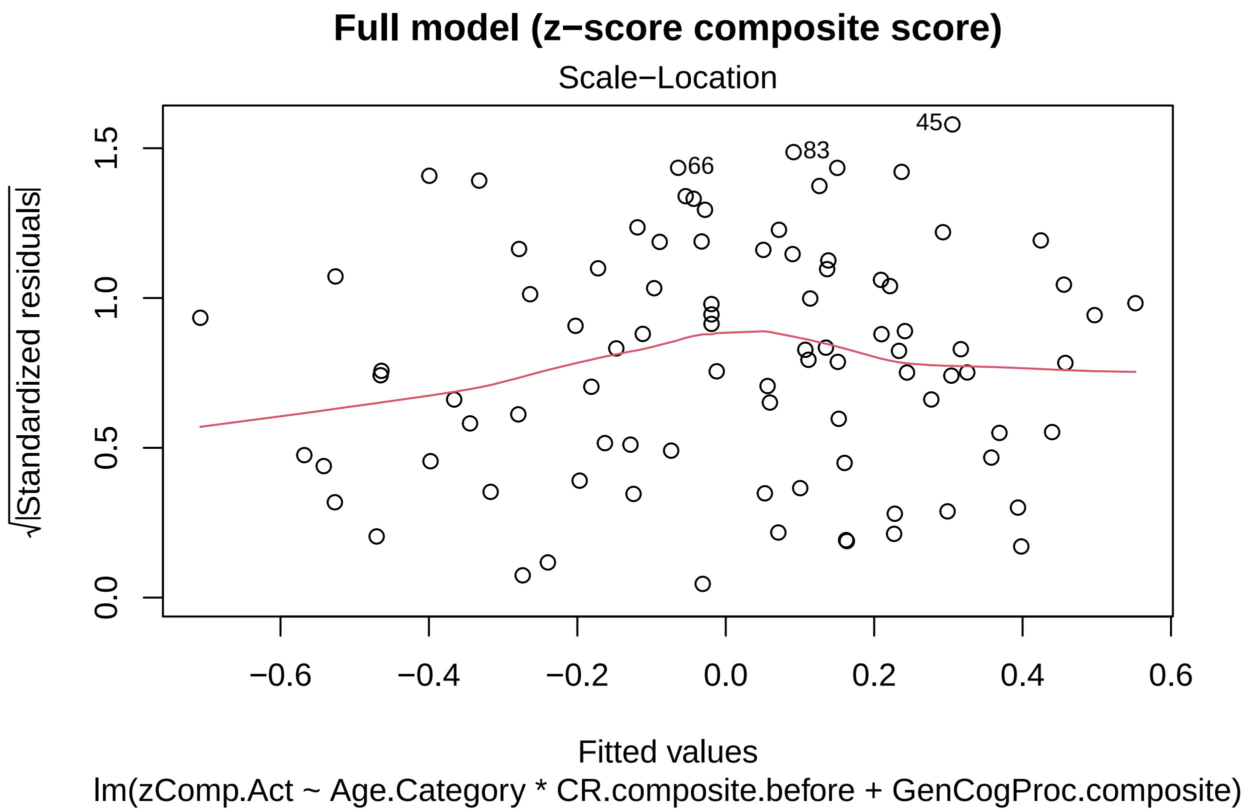


For the unconditional model and the full model with raw scores, the points seem to roughly follow a straight line. For the full model with the z-composite score, there is a small bulk on the left and right. Hence, other relationships/predictors that have not been included into the models could explain the variance for this model. One explanation could be the online format of the study. For example, some participants might have scored lower than they would in a lab-based study, due to unclear instructions. Also, it could have caused more distractions than there would have been in a lab based study.

Assumption 4 - Homoscedasticity or Equal Variance of Variables

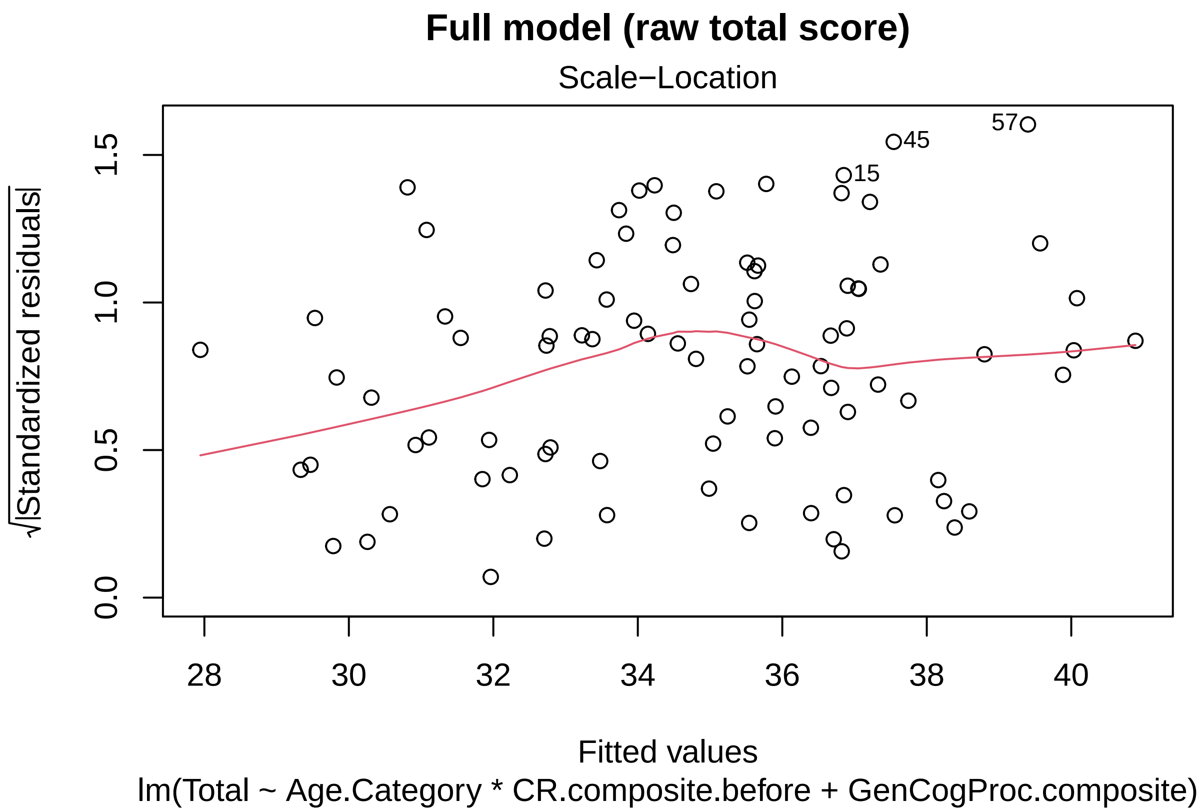
Full model with z composite score

```
plot(lmFull.VFact.Ncorrect, 3, main = "Full model (z-score composite score)")
```



Full model with raw total score

```
plot(lmFull.VFact.Ncorrect.raw, 3, main = "Full model (raw total score)")
```



For all models, the variance of residuals seems relatively equal across the predictors. Hence, the error terms are relatively the same across all values of the independent variable for all three models.

Model fit diagnostics *Variation Inflation Factor*

Any VIF value above 4 needs further investigation. There seems to be no concerning signs of collinearity (VIF values higher than 4). The tolerance values indicate the percentage of variance that cannot be explained for by the other predictor variables.

```
ols_vif_tol(lmUncond.VFact.Ncorrect)
```

```
##                           Variables Tolerance VIF
## 1                     Age.Category1         1   1
## 2                     Age.Category2         1   1
## 3               CR.composite.before         1   1
## 4 Age.Category1:CR.composite.before         1   1
## 5 Age.Category2:CR.composite.before         1   1
```

```
ols_vif_tol(lmFull.VFact.Ncorrect)
```

```
##                           Variables Tolerance      VIF
## 1                     Age.Category1 0.8958833 1.116217
## 2                     Age.Category2 0.7518193 1.330107
## 3               CR.composite.before 0.9971961 1.002812
## 4              GenCogProc.composite 0.6869407 1.455730
```

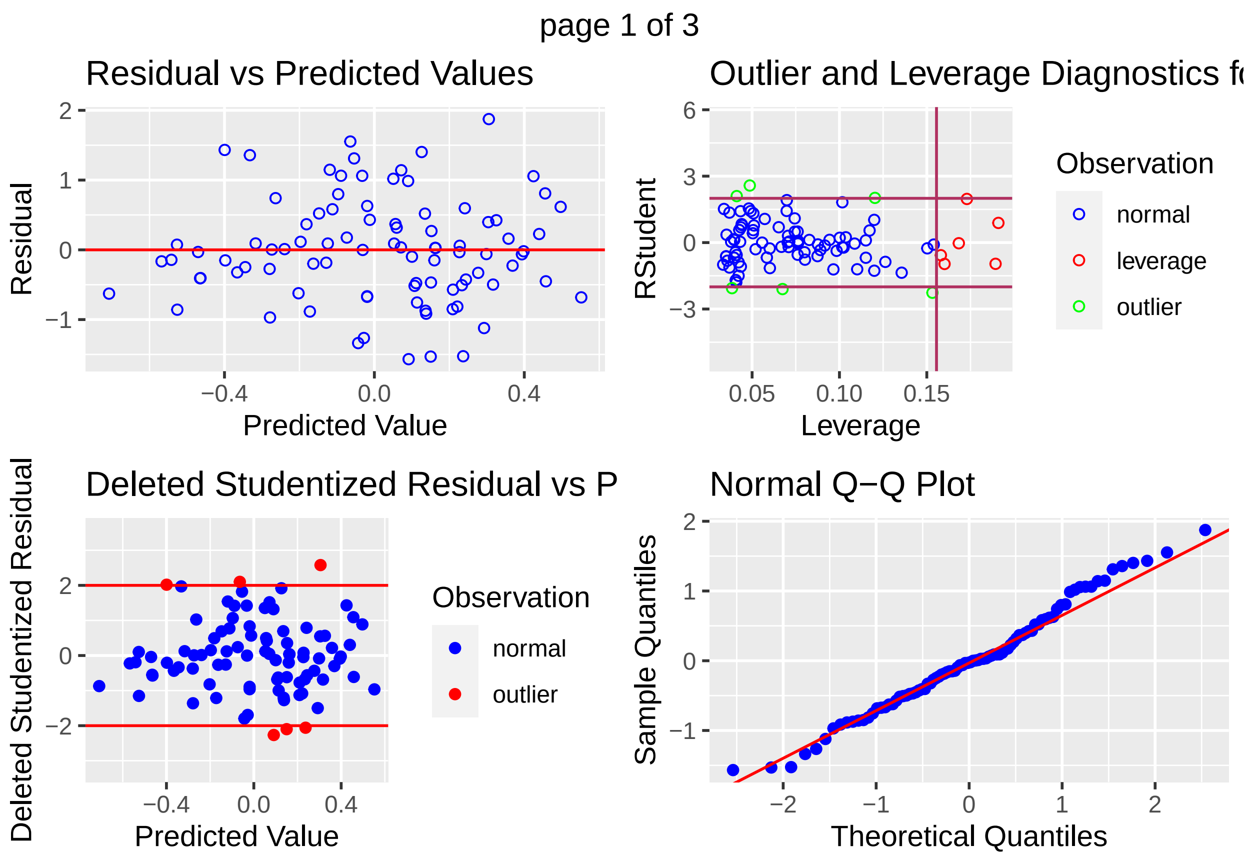
```
## 5 Age.Category1:CR.composite.before 0.9935202 1.006522
## 6 Age.Category2:CR.composite.before 0.9999277 1.000072
```

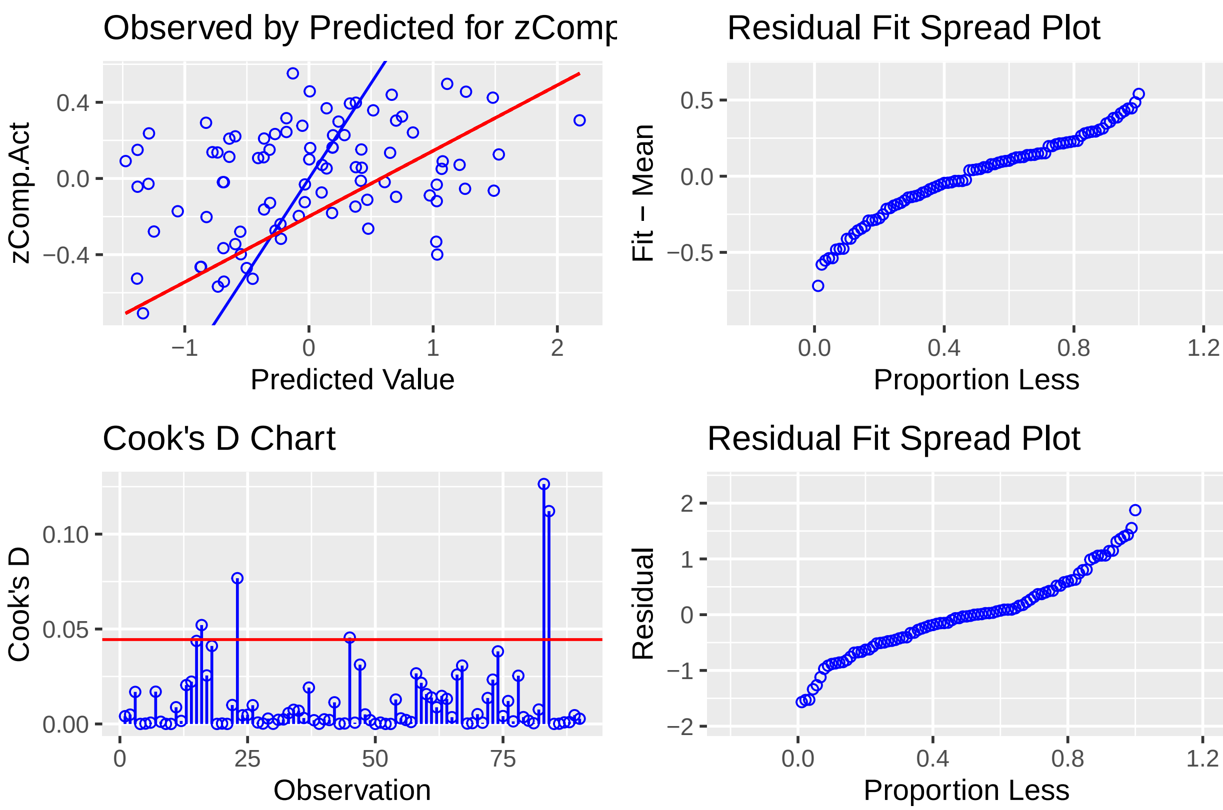
```
ols_vif_tol(lmFull.VFact.Ncorrect.raw)
```

```
##                             Variables Tolerance      VIF
## 1                       Age.Category1 0.8958833 1.116217
## 2                       Age.Category2 0.7518193 1.330107
## 3                  CR.composite.before 0.9971961 1.002812
## 4                 GenCogProc.composite 0.6869407 1.455730
## 5 Age.Category1:CR.composite.before 0.9935202 1.006522
## 6 Age.Category2:CR.composite.before 0.9999277 1.000072
```

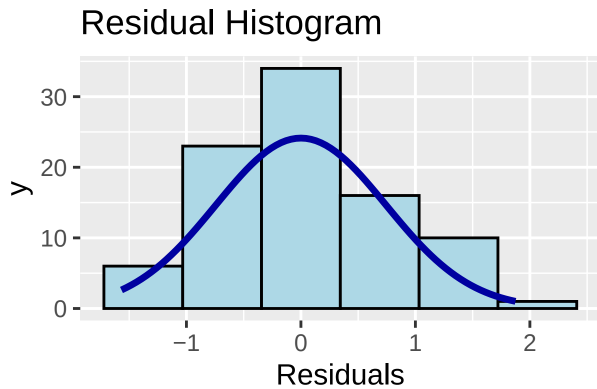
Plot Diagnositcs Full model with z composite score for Semantic Fluency

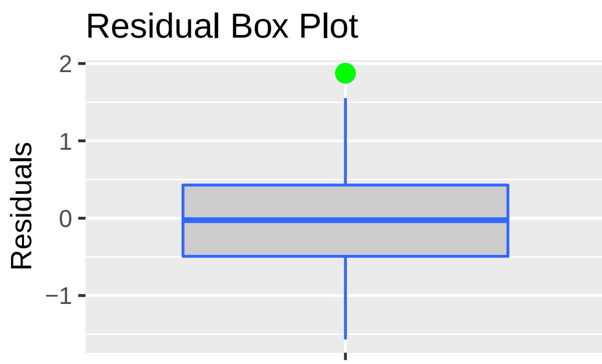
```
ols_plot_diagnostics(lmFull.VFact.Ncorrect)
```




Observed by Predicted for zComp
zComp.Act
Predicted Value
0.4
0.0
−0.4
−1
0
1
2
Residual Fit Spread Plot
Fit − Mean
Proportion Less
0.5
0.0
−0.5
0.0
0.4
0.8
1.2
Cook's D Chart
Cook's D
Observation
0.10
0.05
0.00
0
25
50
75
Residual Fit Spread Plot
Residual
Proportion Less
2
1
0
−1
−2
0.0
0.4
0.8
1.2

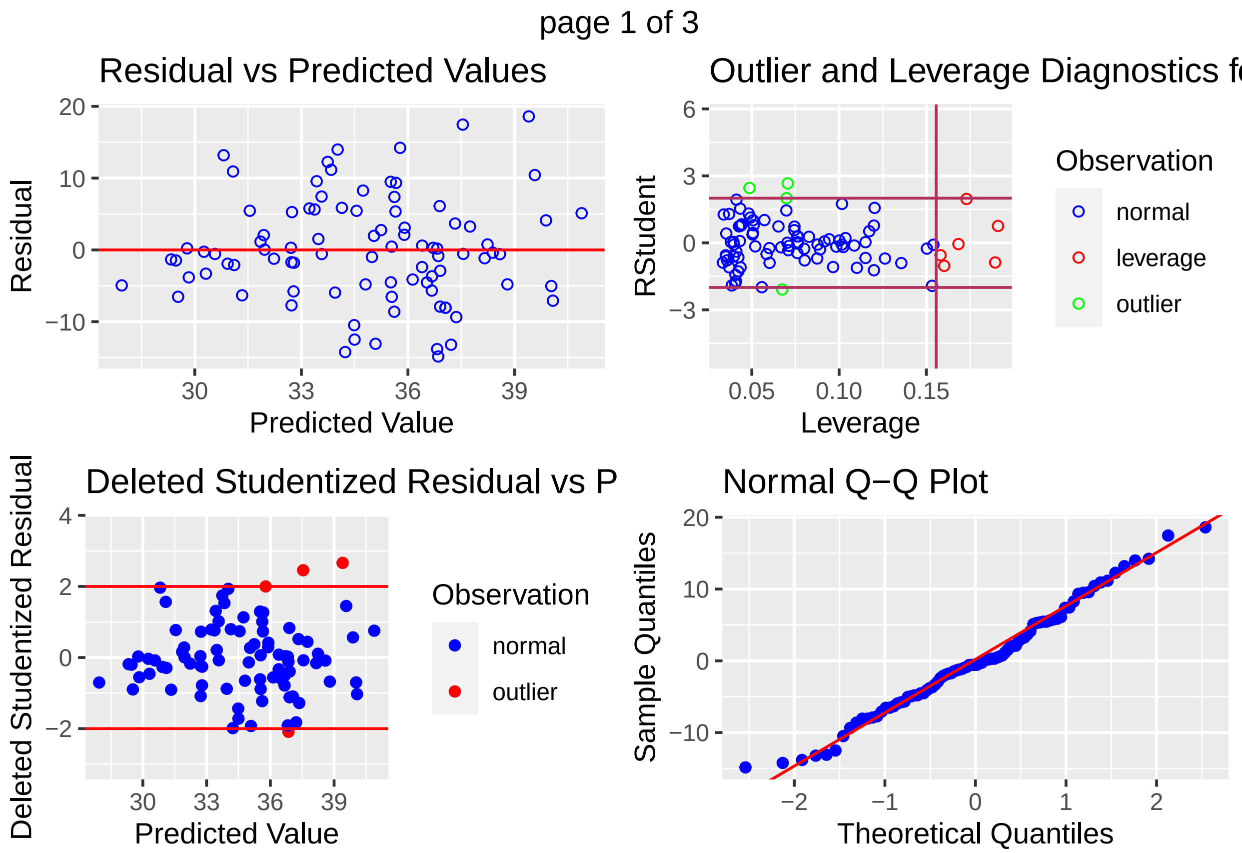
page 3 of 3



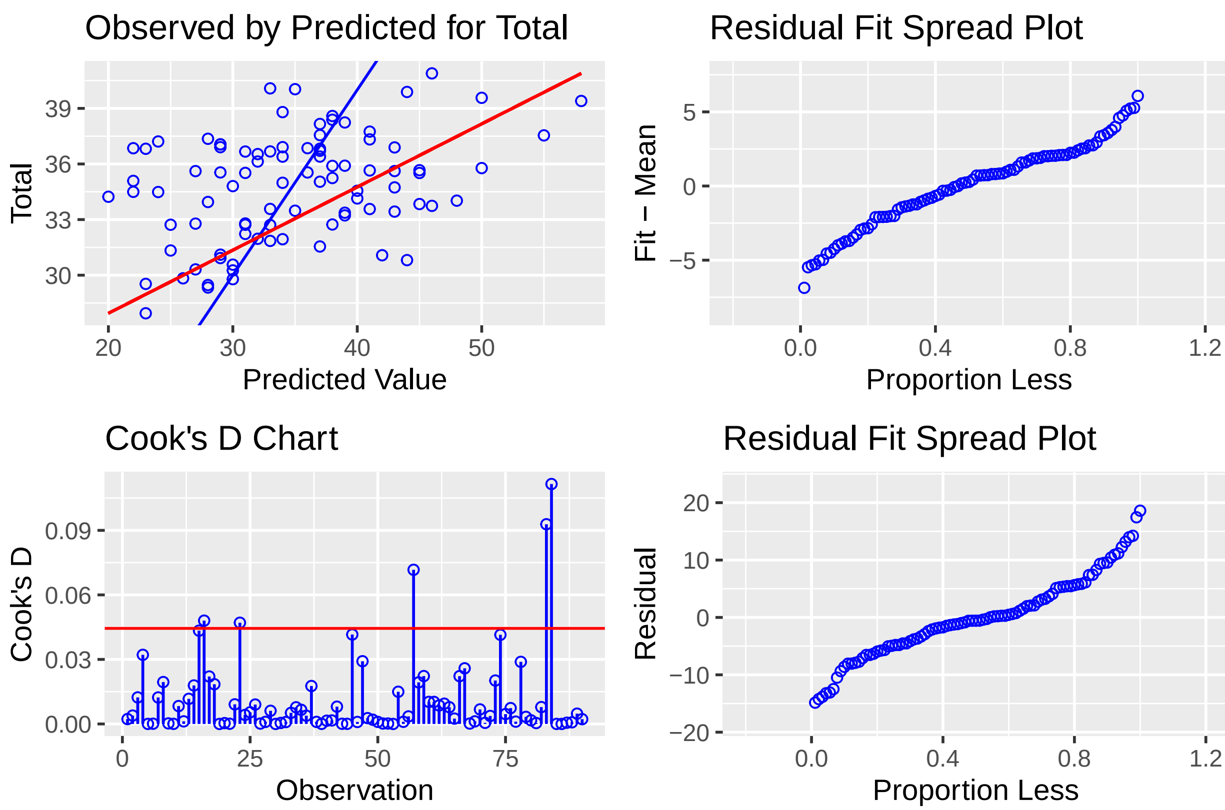


Plot Diagnositcs Full model with raw score for Semantic Fluency

```
ols_plot_diagnostics(lmFull.VFact.Ncorrect.raw)
```

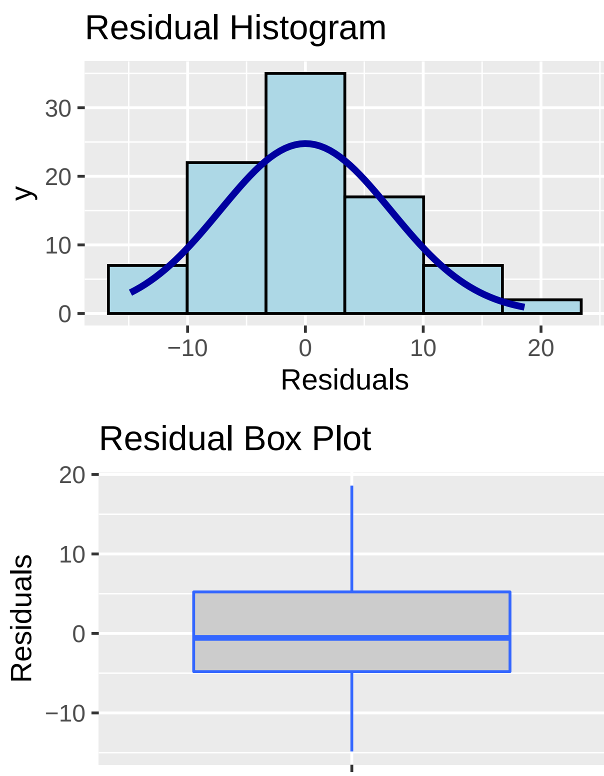

page 1 of 3
Residual vs Predicted Values
Residual
Predicted Value
Outlier and Leverage Diagnostics fo
RStudent
Leverage
Observation
normal
leverage
outlier
Deleted Studentized Residual vs P
Deleted Studentized Residual
Predicted Value
Observation
normal
outlier
Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles

page 2 of 3



Observed by Predicted for Total
Total
Predicted Value
Residual Fit Spread Plot
Fit − Mean
Proportion Less
Cook's D Chart
Cook's D
Observation
Residual Fit Spread Plot
Residual
Proportion Less

page 3 of 3



For all models, the Observed vs Predicted Plot shows that the model doesn't fit the data very well (pity) –> due to outliers?? What to do. . . .

It can explain the fact that the models do not explain the outcome variable in any case.

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

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

```
lmFull.VFact.Ncorrect.during <- lm(zComp.Act ~ Age.Category *
    CR.composite.during + GenCogProc.composite, data = VFact_Ncorrect_coded)
summary(lmFull.VFact.Ncorrect.during)
```

```
##
## Call:
## lm(formula = zComp.Act ~ Age.Category * CR.composite.during +
##     GenCogProc.composite, data = VFact_Ncorrect_coded)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56972 -0.46039 -0.07224  0.51015  1.92485
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           0.01339    0.08039   0.167   0.8681
## Age.Category1                        -0.06879    0.10420  -0.660   0.5110
```

```
## Age.Category2                      -0.01665    0.06580  -0.253   0.8009
## CR.composite.during                 0.12514    0.08267   1.514   0.1339
## GenCogProc.composite                0.46705    0.18920   2.469   0.0156 *
## Age.Category1:CR.composite.during -0.19480    0.10195  -1.911   0.0595 .
## Age.Category2:CR.composite.during  0.03737    0.05872   0.636   0.5263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7626 on 83 degrees of freedom
## Multiple R-squared:  0.1431, Adjusted R-squared:  0.08112
## F-statistic:  2.31 on 6 and 83 DF,  p-value: 0.04121
```

```
# Model comparisons
anova(lmFull.VFact.Ncorrect, lmFull.VFact.Ncorrect.during)
```

```
## Analysis of Variance Table
##
## Model 1: zComp.Act ~ Age.Category * CR.composite.before + GenCogProc.composite
## Model 2: zComp.Act ~ Age.Category * CR.composite.during + GenCogProc.composite
##   Res.Df    RSS Df Sum of Sq F Pr(>F)
## 1     83 49.256
## 2     83 48.269  0   0.98697
```

No differences between the two models

```
# Model comparisons through AIC values
AIC(lmFull.VFact.Ncorrect)
```

```
## [1] 217.1589
```

```
AIC(lmFull.VFact.Ncorrect.during)
```

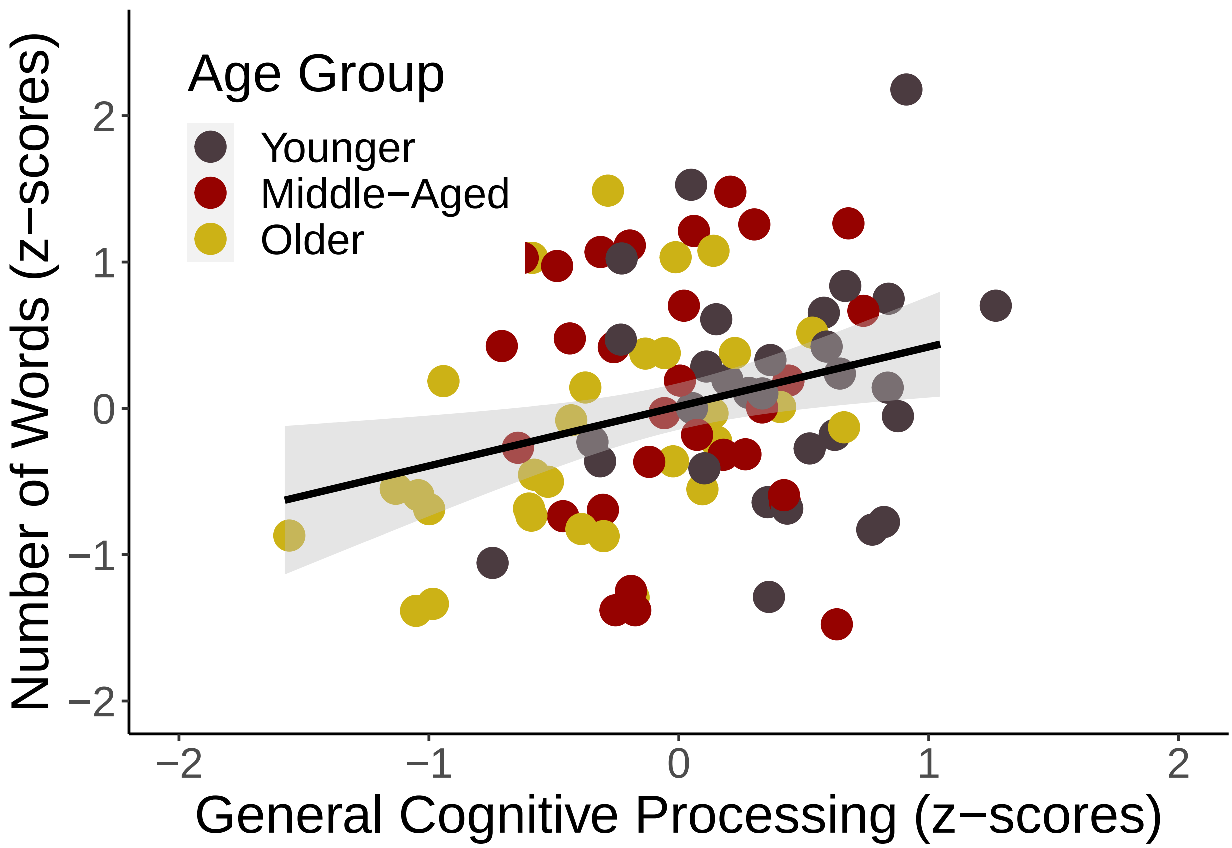
```
## [1] 215.3372
```

The model with the composite score during Covid-19 seems to fit slightly better

Relationship between General Cognitive Processing and Action Fluency

```
# tiff(file="../Figures and Tables/RelationVFact_Ncor_GenCogProc.tiff",
# res = 500, family = "sans", width = 12, height=9, units="in")

#Relationship Action Fluency and General cognitive processing
(plot.VFact_GenCogProc <- VFact_Ncorrect_coded %>%
       mutate(Age.Category_ordered = factor(Age.Category, levels=c("Younger", "Middle-Aged", "Older"))) %>%
    ggplot(aes(x=GenCogProc.composite, y=zComp.Act, colour = as.factor(Age.Category_ordered))) +
    geom_jitter(width = 0.25, size=5)+
   geom_smooth(method = "glm", formula = y~x, fill="grey", colour="black", show.legend = F, size=1.3) +
   labs(x = "General Cognitive Processing (z-scores)",
        y= "Number of Words (z-scores)") +
      coord_cartesian(ylim = c(-2,2.5), xlim = c(-2,2)) +
   scale_x_continuous(breaks = seq(-2, 2, 1)) +
    scale_y_continuous(breaks = seq(-2,2,1)) +
```

```
  # 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"),
        legend.position = c(0.2, 0.8)) +
        scale_colour_manual(values = confPalette, guide=guide_legend(title = "Age Group")))
```




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

References

Chan, Chung-hong, and Thomas J. Leeper. 2021. *Rio: A Swiss-Army Knife for Data i/o*. https://github.com/leeper/rio.

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.

Hebbali, Aravind. 2020. *Olsrr: Tools for Building OLS Regression Models*. https://CRAN.R-project.org/package=olsrr.

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.

Lenth, Russell V. 2021. *Emmeans: Estimated Marginal Means, Aka Least-Squares Means*. https://github.com/rvlenth/emmeans.

Lüdecke, 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.

Lüdecke, 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/.

Wei, Taiyun, and Viliam Simko. 2021a. *Corrplot: Visualization of a Correlation Matrix*. https://github.com/taiyun/corrplot.

———. 2021b. *R Package 'Corrplot': Visualization of a Correlation Matrix*. https://github.com/taiyun/corrplot.

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 Grolemund, 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. 2022. *formatR: Format r Code Automatically*. https://github.com/yihui/formatR.