Analyses for Wright & Celic

March 16, 2024

Contents

This is the technical report for Wright & Celic (under review). The pre-registration is Wright and Celic (2024), and the aim is to explore the differences in findings between Skagerberg and Wright (2008) and Carol, Carlucci, Eaton, and Wright (2013). The `rnw` of the submitted final submitted manuscript will also be placed in this folder. The data file is also on this page (though in the code here it is called from a folder on the hard drive so that would need to be changed for replicating). The second and third lines from the Qualtrics file were removed, as are the variables for IP address, their Prolific ID, and the location variables. The code checking for duplicate IP addresses is not run. If you wish access to the Qualtrics survey (i.e., the code, email us).

1 Loading some packages (not using all)

```
library(lme4)
library(lavaan)
library(effectsize)
library(lattice)
library(EFA.dimensions)
library(xtable)
library(psych)
library(plot.matrix)
library(grDevices)
library(car)
library(Hmisc)
library(e1071)
library(tm)
library(textstem)
library(SentimentAnalysis)
library(sentimentr)
```

2 Loading Data

Also creating the

```
fnamea <- "C:\\Users\\wrighd12\\Documents\\Vuk\\memconfpow1aTR.csv"
nn <- read.csv(fnamea)
nn$ControlWrite[nn$ControlWrite == ""] <- NA
nn$LMWrite[nn$LMWrite == ""] <- NA
nn$LEWrite[nn$LEWrite == ""] <- NA
nn$HMWrite[nn$HMWrite == ""] <- NA
nn$HEWrite[nn$HEWrite == ""] <- NA
nn$cond <- rep(NA,nrow(nn))
nn$cond[!is.na(nn$ControlWrite)] <- "C"
nn$cond[!is.na(nn$LEWrite)] <- "LE"
nn$cond[!is.na(nn$LMWrite)] <- "LM"
nn$cond[!is.na(nn$HEWrite)] <- "HE"
nn$cond[!is.na(nn$HMWrite)] <- "HM"
table(nn$cond,useNA="always")

##
##    C   HE   HM   LE   LM <NA>
##   82   79   80   76   81   40

table(nn$studydescription) # all consent
```

```
##
## I consent.
##        438

nn$cond <- as.factor(nn$cond)
table(nn$Progress) # most of the non-100 were at the end, I

##
##   1   2   3   4   5   6  12  17  18  25  27  28  32  65  76  93 100
##   7   1   1   1   1   1   1   1   1   4  21   1   1   1   1   1 393

         # pressed pause at some point thinking it
         # would not affect those doing it, but it might have
                   # and
nn <- nn[nn$Progress == 100,]
#names(nn)  #too long to print
dim(nn)

## [1] 393 816
```

Checking time to see if payment to participants was about right. Note that on MTurk a lot of people do several tasks before going back to get paid; it may be similar on Prolific. Our payment was about as expected (slightly more on average).

```
quantile(nn$Duration..in.seconds.)/60

##       0%      25%      50%      75%     100%
## 10.15000 15.96667 18.78333 22.55000 92.31667
```

3 More exclusions

3.1 IP Address (not run)

```
dupIPs <- duplicated(nn$IPAddress)
table(dupIPs,useNA="always")
# no missing after the Progress < 100 removed
dim(nn) # same
```

3.2 For being too quick on the MDS

Memory Distrust Scale (Nash, Saraiva, & Hope, 2023).

```
#speed
MDStimes <- with(nn,cbind(
MDS01time_Last.Click,MDS02time_Last.Click,
MDS03time_Last.Click,MDS04time_Last.Click,
MDS05time_Last.Click,MDS06time_Last.Click,
MDS07time_Last.Click,MDS08time_Last.Click,
MDS09time_Last.Click,MDS10time_Last.Click,
MDS11time_Last.Click,MDS12time_Last.Click,
MDS13time_Last.Click,MDS14time_Last.Click,
MDS15time_Last.Click,MDS16time_Last.Click,
```

```
MDS17time_Last.Click,MDS18time_Last.Click,
MDS19time_Last.Click,MDS20time_Last.Click))
totMDStimes <- rowMeans(MDStimes)
quantile(totMDStimes,probs=seq(0,1,.25))

##       0%      25%      50%      75%     100%
##  1.13265  3.80355  4.93340  6.65385 73.18015

table(totMDStimes > 2,useNA="always")

##
## FALSE  TRUE  <NA>
##    11   382     0

#11 removed
nn <- nn[totMDStimes > 2,]
dim(nn)

## [1] 382 816

table(nn$cond,useNA="always")

##
##    C   HE   HM   LE   LM <NA>
##   80   75   76   73   78    0
```

Qualtrics produces much information that we do not use. This includes multiple variables corresponding to the presentation of the items. We had these set so the other person's response was shown for four seconds and then the next refresh it moved on. This means that participants were likely ready to respond when prompted. All timing questions produce multiple measures. We focus on the last click measures (as above). For many the page automatically submits at the next refresh. The first click, particularly for the confidence scale, is just getting the curser to the right height.

4 Demographics and a couple of more exclusions

```
# Person variables
demos <- data.frame(PID=nn$Prolific_PID,cond=nn$cond,
  gender=nn$gender,ethnicity=nn$ethnicity,age=nn$age,
  Fluent=nn$Fluent)
table(demos$age,useNA="always")

##
##  18 - 24  25 - 34  35 - 44  45 - 54  55 - 64  65 - 74  75 - 84 Under 18
##       26      115       90       76       51       19        4        1
##     <NA>
##        0

table(demos$Fluent,useNA="always")

##
## I speak and understand English well, but I do not consider myself fluent.
##                                                                          3
##                                                          Yes, I am fluent.
##                                                                        379
```

```
##                                                                   <NA>
##                                                                      0
```

There was one who self-reported being under 18 and four saying not fluent. These also excluded. These were screening filters set up on Prolific.

```
dim(nn)

## [1] 382 816

nn <- nn[nn$age != "Under 18",]
nn <- nn[nn$Fluent == "Yes, I am fluent.",]
dim(nn)

## [1] 378 816

#remake demos
demos <- data.frame(PID=nn$Prolific_PID,cond=nn$cond,
  gender=nn$gender,ethnicity=nn$ethnicity,age=nn$age,
  Fluent=nn$Fluent)
table(demos$cond,useNA="always")

##
##    C   HE   HM   LE   LM <NA>
##   80   73   76   73   76    0

table(demos$gender,useNA="always")

##
##                    Female                      Male Non-binary / third gender
##                       187                       188                         3
##                      <NA>
##                         0

table(demos$ethnicity,useNA="always")

##
##                                  Asian             Black             Other
##                   1                 30                12                 2
## Two or more races                White              <NA>
##                  15                318                 0

table(demos$age,useNA="always")

##
## 18 - 24 25 - 34 35 - 44 45 - 54 55 - 64 65 - 74 75 - 84    <NA>
##      25     114      90      76      50      19       4       0

table(demos$Fluent,useNA="always")

##
## Yes, I am fluent.              <NA>
##               378                 0
```

5 Adding in whether the other person was right and creating matrices

There are no missing on the key 48 trials as Qualtrics was set to require answers. As such we can create the long format in lots of ways, including just repeating things 48 times (instead of using one of the `reshape` functions). This makes adding things like whether the other person was right a little easier, but requires some care

```
person <- rep(1:nrow(nn),48)
person[44:55]

##  [1] 44 45 46 47 48 49 50 51 52 53 54 55

otherright <-
  c(0,1,1,1,1,0,1,0,0,1,
    0,0,1,0,0,0,1,1,0,1,
    1,1,1,1,1,0,1,0,1,0,
    1,0,0,0,0,1,0,1,0,0,
    0,1,1,0,1,1,0,0)
qnums <-
  c(14, 8,18,22,25,33,21, 1,38,39,
    16,40,44,36,32,10,12, 4,42,35,
    48, 3,27,20,17,23,15,11,28,30,
     2,24,45, 5,26,31,46, 7,37, 6,
    19,29,41,13,47,43, 9,34)
qnumsch <- sub("0.","",sprintf("%2.2f",qnums/100))
mc1wide <- nn  #more descriptive name
mc1wide$right14 <- mc1wide$rec14 == "Psolged"
mc1wide$right08 <- mc1wide$rec08 == "Grymmed"
mc1wide$right18 <- mc1wide$rec18 == "Hunged"
mc1wide$right22 <- mc1wide$rec22 == "Sourche"
mc1wide$right25 <- mc1wide$rec25 == "Trewl"
mc1wide$right33 <- mc1wide$rec33 == "Guarcs"
mc1wide$right21 <- mc1wide$rec21 == "Smeighck"
mc1wide$right01 <- mc1wide$rec01 == "Shrurched"
mc1wide$right38 <- mc1wide$rec38 == "Snourged"
mc1wide$right39 <- mc1wide$rec39 == "Thyf"
mc1wide$right16 <- mc1wide$rec16 == "Ghweiled"
mc1wide$right40 <- mc1wide$rec40 == "Cews"
mc1wide$right44 <- mc1wide$rec44 == "Sprult"
mc1wide$right36 <- mc1wide$rec36 == "Scrourth"
mc1wide$right32 <- mc1wide$rec32 == "Strafth"
mc1wide$right10 <- mc1wide$rec10 == "Froaced"
mc1wide$right12 <- mc1wide$rec12 == "Blezz"
mc1wide$right04 <- mc1wide$rec04 == "Whatts"
mc1wide$right42 <- mc1wide$rec42 == "Shrypth"
mc1wide$right35 <- mc1wide$rec35 == "Spleuks"
mc1wide$right48 <- mc1wide$rec48 == "Treeld"
mc1wide$right03 <- mc1wide$rec03 == "Splobed"
mc1wide$right27 <- mc1wide$rec27 == "Phuke"
mc1wide$right20 <- mc1wide$rec20 == "Groothe"
mc1wide$right17 <- mc1wide$rec17 == "Snoarnths"
mc1wide$right23 <- mc1wide$rec23 == "Thrighsts"
mc1wide$right15 <- mc1wide$rec15 == "Greabed"
```

```
mc1wide$right11 <- mc1wide$rec11 == "Splonc"
mc1wide$right28 <- mc1wide$rec28 == "Starnd"
mc1wide$right30 <- mc1wide$rec30 == "Spreenes"
mc1wide$right02 <- mc1wide$rec02 == "Psoarm"
mc1wide$right24 <- mc1wide$rec24 == "Proled"
mc1wide$right45 <- mc1wide$rec45 == "Fowche"
mc1wide$right05 <- mc1wide$rec05 == "Jyfed"
mc1wide$right26 <- mc1wide$rec26 == "Jelmed"
mc1wide$right31 <- mc1wide$rec31 == "Seuggs"
mc1wide$right46 <- mc1wide$rec46 == "Kalls"
mc1wide$right07 <- mc1wide$rec07 == "Yoise"
mc1wide$right37 <- mc1wide$rec37 == "Krymmth"
mc1wide$right06 <- mc1wide$rec06 == "Yolthed"
mc1wide$right19 <- mc1wide$rec19 == "Cloughged"
mc1wide$right29 <- mc1wide$rec29 == "Flulgn"
mc1wide$right41 <- mc1wide$rec41 == "Tuiv"
mc1wide$right13 <- mc1wide$rec13 == "Thrurphed"
mc1wide$right47 <- mc1wide$rec47 == "Blirds"
mc1wide$right43 <- mc1wide$rec43 == "Chylth"
mc1wide$right09 <- mc1wide$rec09 == "Skorts"
mc1wide$right34 <- mc1wide$rec34 == "Lawlds"

recs <- with(mc1wide,cbind(
    rec14,rec08,rec18,rec22,rec25,rec33,rec21,rec01,rec38,rec39,
    rec16,rec40,rec44,rec36,rec32,rec10,rec12,rec04,rec42,rec35,
    rec48,rec03,rec27,rec20,rec17,rec23,rec15,rec11,rec28,rec30,
    rec02,rec24,rec45,rec05,rec26,rec31,rec46,rec07,rec37,rec06,
    rec19,rec29,rec41,rec13,rec47,rec43,rec09,rec34))
dim(recs)

## [1] 378  48

right <- with(mc1wide,cbind(
    right14,right08,right18,right22,right25,
    right33,right21,right01,right38,right39,
    right16,right40,right44,right36,right32,
    right10,right12,right04,right42,right35,
    right48,right03,right27,right20,right17,
    right23,right15,right11,right28,right30,
    right02,right24,right45,right05,right26,
    right31,right46,right07,right37,right06,
    right19,right29,right41,right13,right47,
    right43,right09,right34))
sort(apply(right,2,sd))

##   right21   right35   right48   right08   right27   right15   right04   right47
## 0.2344336 0.3146862 0.3179363 0.3420916 0.3420916 0.3557176 0.3608880 0.3608880
##   right32   right17   right02   right29   right18   right25   right03   right31
## 0.3659086 0.3731719 0.3755245 0.3801305 0.3889626 0.3889626 0.3889626 0.3931972
##   right43   right41   right39   right01   right22   right30   right11   right10
## 0.3952708 0.3973160 0.4090125 0.4108697 0.4127017 0.4197823 0.4264822 0.4280998
##   right09   right19   right14   right44   right12   right07   right28   right16
## 0.4280998 0.4312681 0.4343486 0.4343486 0.4402538 0.4430819 0.4535998 0.4548280
##   right23   right40   right05   right24   right36   right06   right42   right26
```

```
## 0.4618097 0.4650576 0.4691439 0.4773511 0.4789947 0.4797925 0.4805743 0.4820905
##   right13   right20   right33   right46   right45   right38   right37   right34
## 0.4828250 0.4881441 0.4955276 0.4955276 0.4986333 0.4999614 0.4999614 0.5005506
```

```
sort(apply(right,2,mean))
```

```
##   right34   right38   right37   right45   right33   right46   right20   right13
## 0.4894180 0.5264550 0.5264550 0.5449735 0.5714286 0.5714286 0.6111111 0.6322751
##   right26   right42   right06   right36   right24   right05   right40   right23
## 0.6349206 0.6402116 0.6428571 0.6455026 0.6507937 0.6746032 0.6851852 0.6931217
##   right16   right28   right07   right12   right14   right44   right19   right10
## 0.7089947 0.7116402 0.7328042 0.7380952 0.7486772 0.7486772 0.7539683 0.7592593
##   right09   right11   right30   right22   right01   right39   right41   right43
## 0.7592593 0.7619048 0.7724868 0.7830688 0.7857143 0.7883598 0.8042328 0.8068783
##   right31   right18   right25   right03   right29   right02   right17   right32
## 0.8095238 0.8148148 0.8148148 0.8148148 0.8253968 0.8306878 0.8333333 0.8412698
##   right04   right47   right15   right08   right27   right48   right35   right21
## 0.8465608 0.8465608 0.8518519 0.8650794 0.8650794 0.8862434 0.8888889 0.9417989
```

```
# from around 50% to over 90%
quantile(apply(right,1,mean),probs=seq(0,1,.1))
```

```
##        0%       10%       20%       30%       40%       50%       60%       70%
## 0.4166667 0.5625000 0.6041667 0.6458333 0.6875000 0.7395833 0.7916667 0.8333333
##       80%       90%      100%
## 0.8750000 0.9166667 1.0000000
```

```
# some at 100%, most above chance
table(apply(right,1,mean)<.5)
```

```
##
## FALSE  TRUE
##   367    11
```

```
mean(right)
```

```
## [1] 0.7391975
```

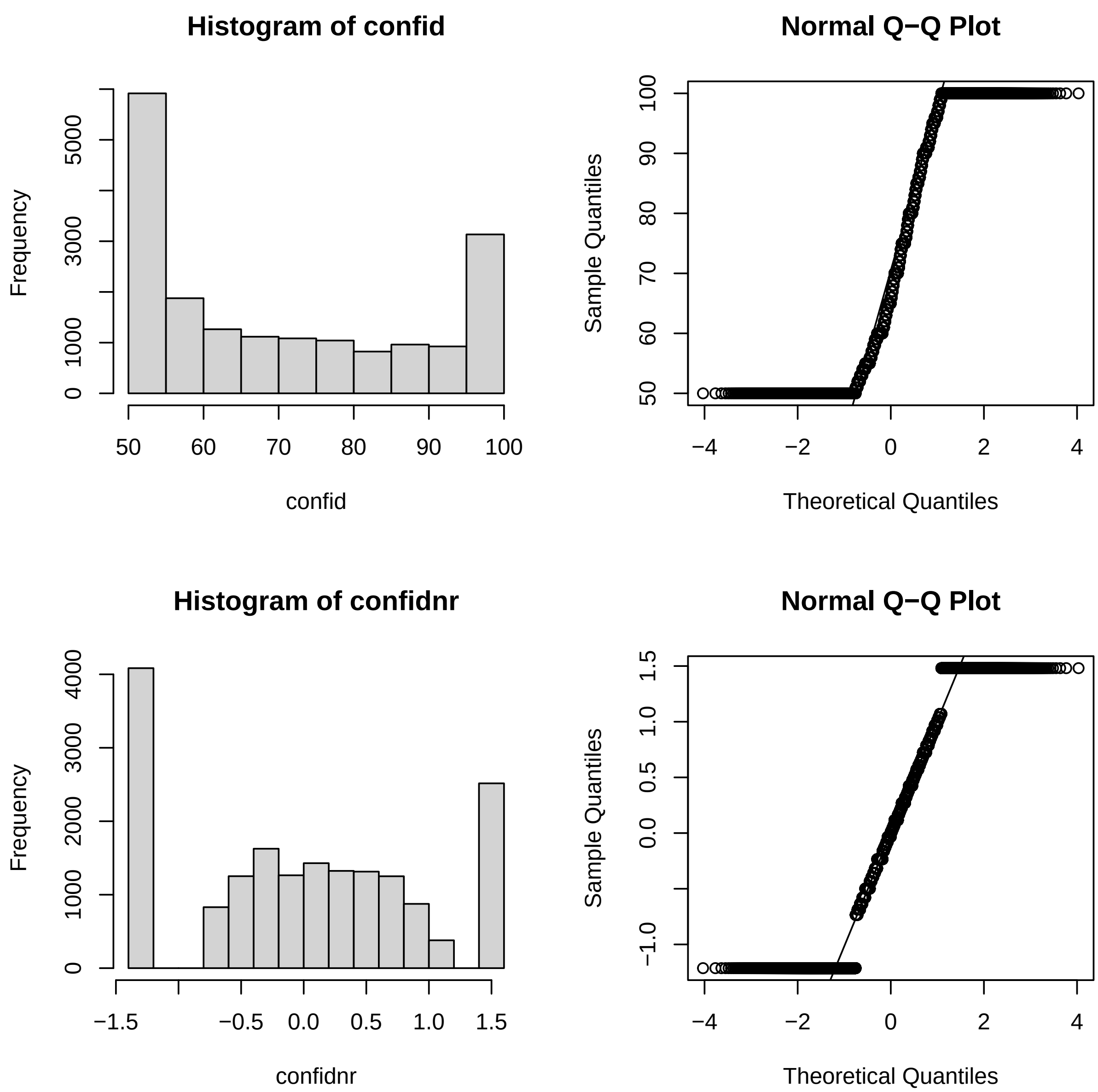
```
confid <- with(mc1wide,cbind(
    confid14_1,confid08_1,confid18_1,confid22_1,confid25_1,
    confid33_1,confid21_1,confid01_1,confid38_1,confid39_1,
    confid16_1,confid40_1,confid44_1,confid36_1,confid32_1,
    confid10_1,confid12_1,confid04_1,confid42_1,confid35_1,
    confid48_1,confid03_1,confid27_1,confid20_1,confid17_1,
    confid23_1,confid15_1,confid11_1,confid28_1,confid30_1,
    confid02_1,confid24_1,confid45_1,confid05_1,confid26_1,
    confid31_1,confid46_1,confid07_1,confid37_1,confid06_1,
    confid19_1,confid29_1,confid41_1,confid13_1,confid47_1,
    confid43_1,confid09_1,confid34_1))
dim(mc1wide); dim(confid); length(c(confid))
```

```
## [1] 378 864
## [1] 378  48
## [1] 18144
```

```
timelast <- with(mc1wide,cbind(
    timer14b_Last.Click,timer08b_Last.Click,timer18b_Last.Click,
    timer22b_Last.Click,timer25b_Last.Click,
    timer33b_Last.Click,timer21b_Last.Click,timer01b_Last.Click,
    timer38b_Last.Click,timer39b_Last.Click,
    timer16b_Last.Click,timer40b_Last.Click,timer44b_Last.Click,
    timer36b_Last.Click,timer32b_Last.Click,
    timer10b_Last.Click,timer12b_Last.Click,timer04b_Last.Click,
    timer42b_Last.Click,timer35b_Last.Click,
    timer48b_Last.Click,timer03b_Last.Click,timer27b_Last.Click,
    timer20b_Last.Click,timer17b_Last.Click,
    timer23b_Last.Click,timer15b_Last.Click,timer11b_Last.Click,
    timer28b_Last.Click,timer30b_Last.Click,
    timer02b_Last.Click,timer24b_Last.Click,timer45b_Last.Click,
    timer05b_Last.Click,timer26b_Last.Click,
    timer31b_Last.Click,timer46b_Last.Click,timer07b_Last.Click,
    timer37b_Last.Click,timer06b_Last.Click,
    timer19b_Last.Click,timer29b_Last.Click,timer41b_Last.Click,
    timer13b_Last.Click,timer47b_Last.Click,
    timer43b_Last.Click,timer09b_Last.Click,timer34b_Last.Click))
colnames(timelast) <- paste0("t",qnumsch)
```

The confidence measure was bi-modal, and it was decided not to transform the variable as we believe the underlying psychological construct likely is bi-modal. However, we had mentioned the following transformation so here is the result.

```
normrank <- function(x,delta)
  qnorm((rank(x) - delta)/(length(x) - 2 * delta + 1))

confidnr <- normrank(confid,delta=.7)
par(mfrow=c(2,2))
hist(confid);qqnorm(confid);qqline(confid)
hist(confidnr);qqnorm(confidnr);qqline(confidnr)
```

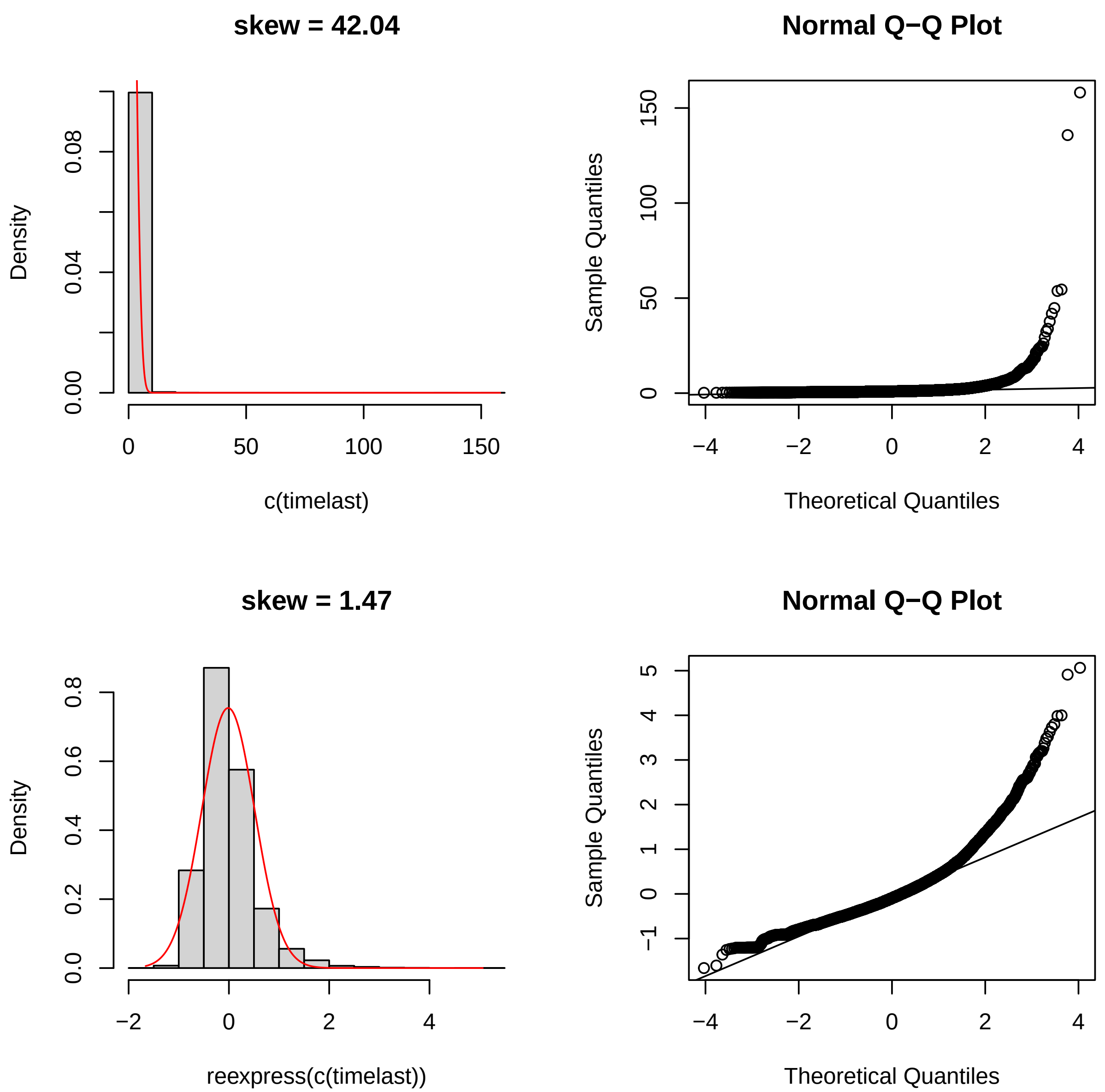


6 Looking at the time to answer the vocabulary questions

These are not really measuring time to answer BECAUSE the items were shown for four seconds prior just with the confederate's response, so the participant is ready to respond.

```
#time EDA
par(mfrow=c(2,2))
hist(c(timelast),freq=FALSE,
  main=paste("skew =", sprintf("%.2f",skewness(c(timelast)))))
tvals <- seq(min(c(timelast)),max(c(timelast)),length=1000)
yvals <- dnorm(tvals,mean=mean(c(timelast)),sd=sd(c(timelast)))
lines(tvals,yvals,col="red")
```

```
qqnorm(c(timelast)); qqline(c(timelast))
#likely with real data log will work better
reexpress <- function(x) log(x)
hist(reexpress(c(timelast)),freq=FALSE,
  main=paste("skew =", sprintf("%.2f",
                  skewness(reexpress(c(timelast))))))
tvals <- seq(min(reexpress(c(timelast))),
              max(reexpress(c(timelast))),length=1000)
yvals <- dnorm(tvals,mean=mean(reexpress(c(timelast))),
                sd=sd(reexpress(c(timelast))))
lines(tvals,yvals,col="red")
qqnorm(reexpress(c(timelast))); qqline(reexpress(c(timelast)))
```



7 Turning to long data

I am doing this just by piecing the data together rather than the `reshape` function (and related ones).

```
long1 <- data.frame(
    trial=1:(48*nrow(recs)),
    person=rep(1:nrow(recs),48),condit=rep(mc1wide$cond,48),
    wordno=rep(qnums,each=nrow(recs)),
    partRight=as.numeric(c(right)),oright=rep(otherright,each=nrow(recs)),
    conf=c(confid),rt1=c(timelast))
dim(long1)

## [1] 18144     8
```

```
long1[1:10,]

##    trial person condit wordno partRight oright conf   rt1
## 1      1      1     LE     14         1      0   50 1.379
## 2      2      2     LM     14         1      0   85 2.447
## 3      3      3     HM     14         1      0   60 2.644
## 4      4      4     HE     14         1      0   77 2.173
## 5      5      5     HE     14         1      0   65 1.424
## 6      6      6     HE     14         1      0   90 1.216
## 7      7      7     LM     14         1      0   90 1.328
## 8      8      8     LE     14         0      0   70 4.814
## 9      9      9      C     14         1      0   60 4.029
## 10    10     10     LM     14         1      0   54 3.559

cbind(1:10,1:10,mc1wide$cond[1:10],rep(qnums[1],10),
      recs[1:10,1],right[1:10,1],otherright[1],confid[1:10,1],
      timelast[1:10,1])

##       [,1] [,2] [,3] [,4] [,5]      [,6]    [,7] [,8] [,9]
##  [1,] "1"  "1"  "4"  "14" "Psolged" "TRUE"  "0"  "50" "1.379"
##  [2,] "2"  "2"  "5"  "14" "Psolged" "TRUE"  "0"  "85" "2.447"
##  [3,] "3"  "3"  "3"  "14" "Psolged" "TRUE"  "0"  "60" "2.644"
##  [4,] "4"  "4"  "2"  "14" "Psolged" "TRUE"  "0"  "77" "2.173"
##  [5,] "5"  "5"  "2"  "14" "Psolged" "TRUE"  "0"  "65" "1.424"
##  [6,] "6"  "6"  "2"  "14" "Psolged" "TRUE"  "0"  "90" "1.216"
##  [7,] "7"  "7"  "5"  "14" "Psolged" "TRUE"  "0"  "90" "1.328"
##  [8,] "8"  "8"  "4"  "14" "Flyfed"  "FALSE" "0"  "70" "4.814"
##  [9,] "9"  "9"  "1"  "14" "Psolged" "TRUE"  "0"  "60" "4.029"
## [10,] "10" "10" "5"  "14" "Psolged" "TRUE"  "0"  "54" "3.559"
```

8 Multilevel 1: The CIs for the conditions depending on whether the other person was right or wrong

For consistency using the bobyqa optimizer.

```
m1 <- glmer(partRight~oright*condit + (1|wordno) + (1|person),
            family="binomial",data=long1, control=glmerControl(optimizer="bobyqa",
                           optCtrl=list(maxfun=2e5)))
```

```r
summary(m1)

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: partRight ~ oright * condit + (1 | wordno) + (1 | person)
##    Data: long1
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
##      AIC      BIC   logLik deviance df.resid
##  18940.2  19033.9  -9458.1  18916.2    18132
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0070 -0.7213  0.3982  0.5876  2.0119
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  person (Intercept) 0.5971   0.7727
##  wordno (Intercept) 0.2131   0.4616
## Number of obs: 18144, groups:  person, 378; wordno, 48
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.90177    0.13853   6.510 7.54e-11 ***
## oright           0.67654    0.15520   4.359 1.31e-05 ***
## conditHE         0.04607    0.14713   0.313  0.75418
## conditHM         0.12028    0.14605   0.824  0.41017
## conditLE        -0.32324    0.14596  -2.215  0.02679 *
## conditLM        -0.21364    0.14470  -1.476  0.13982
## oright:conditHE  0.02529    0.11544   0.219  0.82657
## oright:conditHM  0.03712    0.11536   0.322  0.74764
## oright:conditLE  0.51917    0.11587   4.481 7.44e-06 ***
## oright:conditLM  0.31786    0.11381   2.793  0.00522 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) oright cndtHE cndtHM cndtLE cndtLM org:HE org:HM org:LE
## oright      -0.538
## conditHE    -0.504  0.118
## conditHM    -0.508  0.119  0.479
## conditLE    -0.509  0.119  0.479  0.482
## conditLM    -0.513  0.120  0.483  0.487  0.487
## orght:cndHE  0.169 -0.350 -0.335 -0.160 -0.160 -0.161
## orght:cndHM  0.169 -0.351 -0.159 -0.336 -0.160 -0.162  0.472
## orght:cndLE  0.169 -0.349 -0.158 -0.159 -0.321 -0.161  0.470  0.470
## orght:cndLM  0.171 -0.355 -0.161 -0.162 -0.162 -0.326  0.478  0.478  0.476

mnoint <- glmer(partRight~ 0 + condit + oright:condit +
              (1|wordno) + (1|person),
            family="binomial",data=long1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  Model failed
to converge with max|grad| = 0.00779361 (tol = 0.002, component 1)
```

```r
# warning
mnoint <- update(mnoint, .~., control=glmerControl(optimizer="bobyqa",
                            optCtrl=list(maxfun=2e5)))
confint(mnoint,method="Wald") #profile") # slower
```

```
##                     2.5 %    97.5 %
## .sig01                 NA        NA
## .sig02                 NA        NA
## conditC         0.6303417 1.1731908
## conditHE        0.6688047 1.2268676
## conditHM        0.7451934 1.2989099
## conditLE        0.3020744 0.8549904
## conditLM        0.4142580 0.9620035
## conditC:oright  0.3725516 0.9805408
## conditHE:oright 0.3930544 1.0106237
## conditHM:oright 0.4049476 1.0223743
## conditLE:oright 0.8862825 1.5051443
## conditLM:oright 0.6879811 1.3008337
```

```r
summary(m1)$coef
```

```
##                    Estimate Std. Error    z value     Pr(>|z|)
## (Intercept)      0.90176767  0.1385295  6.5095737 7.536437e-11
## oright           0.67653853  0.1552029  4.3590592 1.306228e-05
## conditHE         0.04607003  0.1471285  0.3131277 7.541836e-01
## conditHM         0.12028469  0.1460470  0.8236028 4.101653e-01
## conditLE        -0.32323944  0.1459628 -2.2145331 2.679213e-02
## conditLM        -0.21363660  0.1446959 -1.4764523 1.398225e-01
## oright:conditHE  0.02529347  0.1154430  0.2190992 8.265728e-01
## oright:conditHM  0.03711530  0.1153558  0.3217463 7.476449e-01
## oright:conditLE  0.51916923  0.1158678  4.4807051 7.439684e-06
## oright:conditLM  0.31786372  0.1138078  2.7929880 5.222362e-03
```

```r
summary(m1)$coef[7:10,4]
```

```
## oright:conditHE oright:conditHM oright:conditLE oright:conditLM
##    8.265728e-01    7.476449e-01    7.439684e-06    5.222362e-03
```

```r
p.adjust(summary(m1)$coef[7:10,4])
```

```
## oright:conditHE oright:conditHM oright:conditLE oright:conditLM
##    1.000000e+00    1.000000e+00    2.975874e-05    1.566709e-02
```

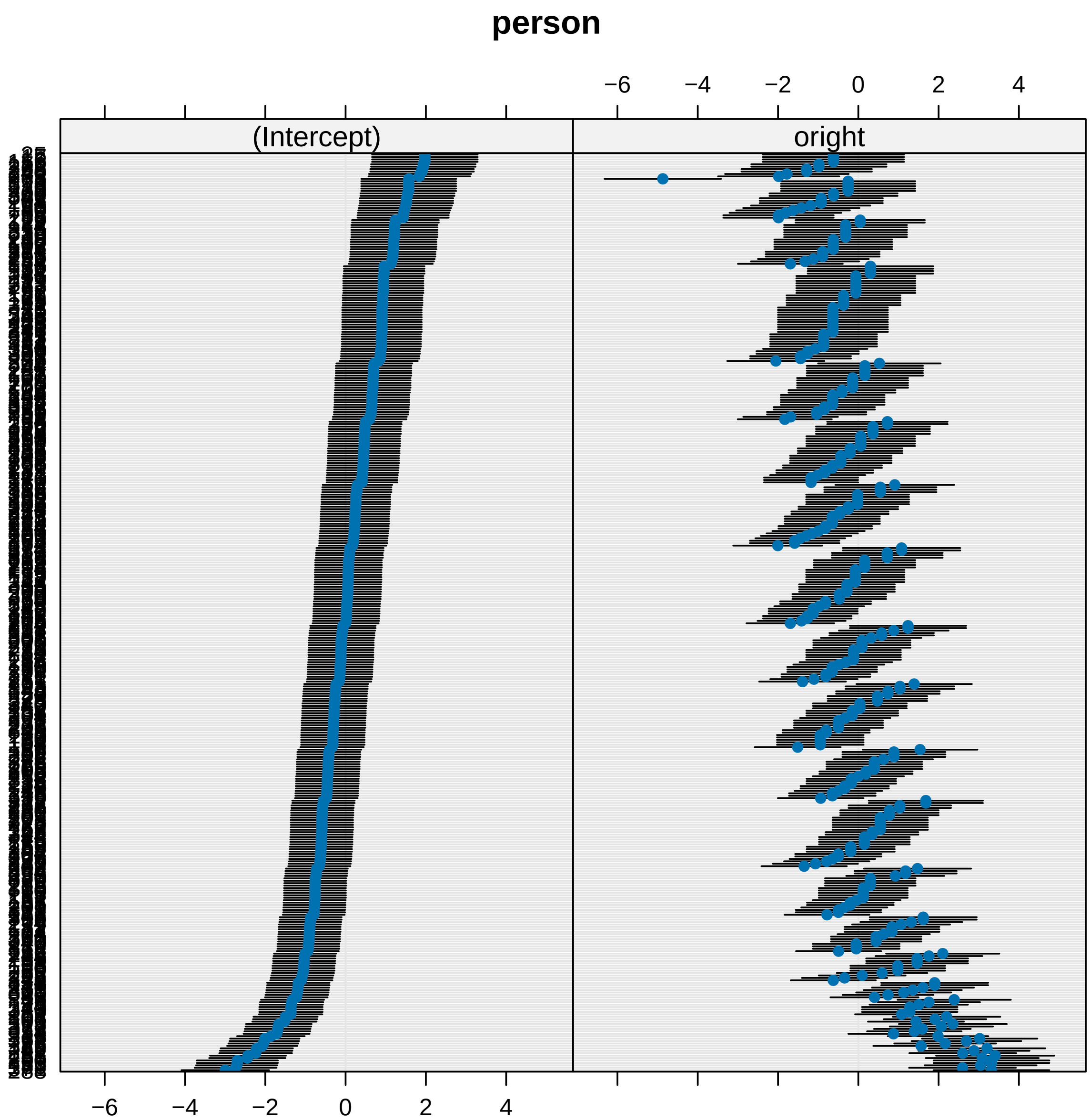
```r
p.adjust(summary(m1)$coef[7:10,4],method="none")
```

```
## oright:conditHE oright:conditHM oright:conditLE oright:conditLM
##    8.265728e-01    7.476449e-01    7.439684e-06    5.222362e-03
```

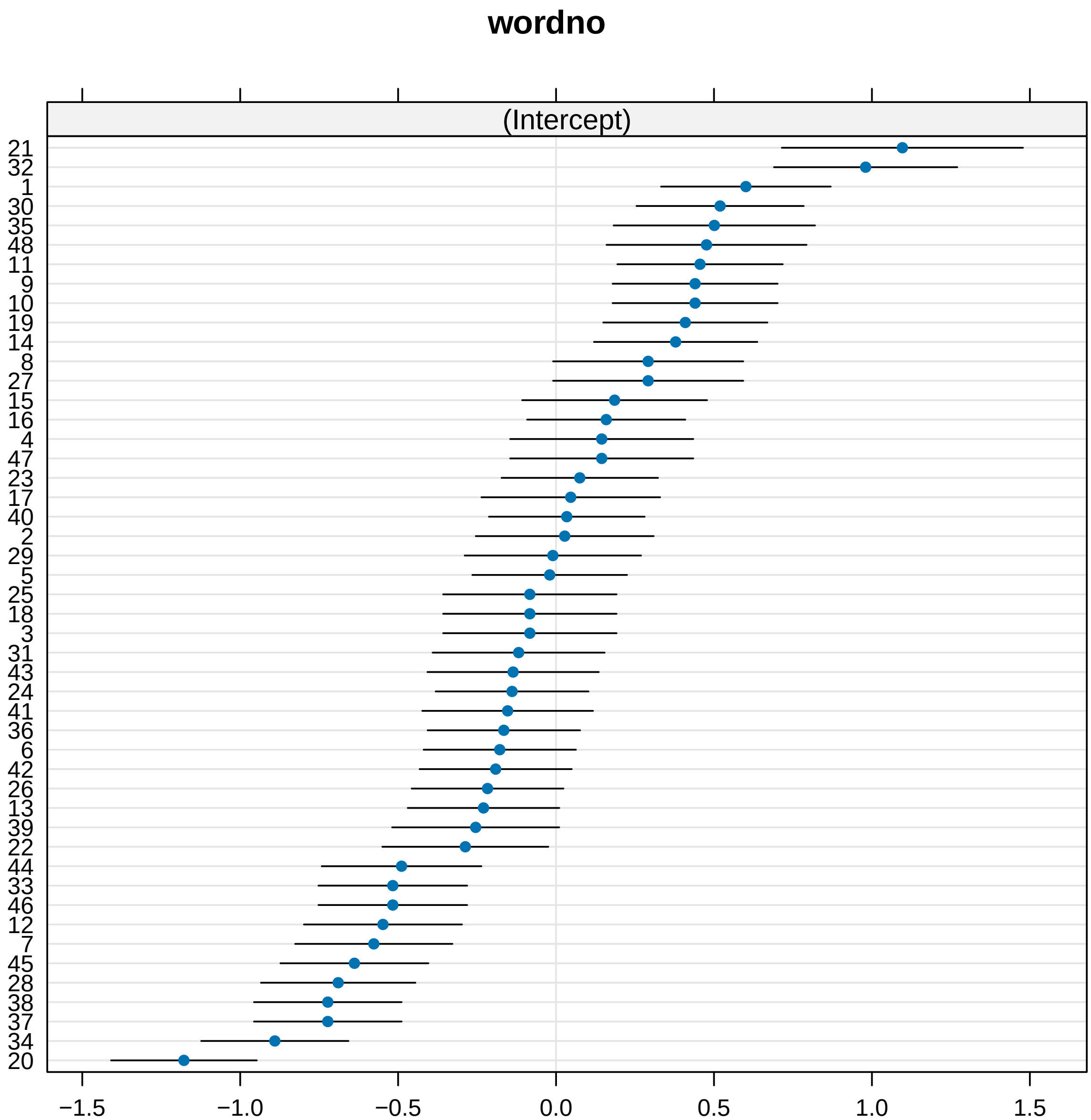
9 Creating conditional modes

```r
mcm <- glmer(partRight ~ oright + (1|wordno) + (oright|person),
     family="binomial",data=long1)
cms <- ranef(mcm)
```

```r
dotplot(cms) # not letting me just

## $person
```




```
##
## $wordno
```




```
#get the CMs for the random slopes
```

10 Psychometrics for the MDS

```
# could have taken numeric values from Qualtrics
mdsnum <- function(x) {
  newvar <- rep(NA,length(x))
  newvar[x == "Strongly disagree"] <- -3
  newvar[x == "Slightly disagree"] <- -1
  newvar[x == "Disagree"] <- -2
  newvar[x == "Agree"] <- 2
```

```
  newvar[x == "Neither agree nor disagree"] <- 0
  newvar[x == "Slightly agree"] <- 1
  newvar[x == "Strongly agree"] <- 3
  return(as.numeric(newvar))}
table(mc1wide$MDS01,mdsnum(mc1wide$MDS01),useNA="always")

##
##                              -3  -2  -1   0   1   2   3 <NA>
##    Agree                      0   0   0   0   0  75   0    0
##    Disagree                   0  58   0   0   0   0   0    0
##    Neither agree nor disagree 0   0   0  40   0   0   0    0
##    Slightly agree             0   0   0   0 100   0   0    0
##    Slightly disagree          0   0  62   0   0   0   0    0
##    Strongly agree             0   0   0   0   0   0  20    0
##    Strongly disagree         23   0   0   0   0   0   0    0
##    <NA>                       0   0   0   0   0   0   0    0

MDS <- cbind(mdsnum(mc1wide$MDS01),mdsnum(mc1wide$MDS02),
             mdsnum(mc1wide$MDS03),mdsnum(mc1wide$MDS04),
             mdsnum(mc1wide$MDS05),mdsnum(mc1wide$MDS06),
             mdsnum(mc1wide$MDS07),mdsnum(mc1wide$MDS08),
             mdsnum(mc1wide$MDS09),mdsnum(mc1wide$MDS10),
             mdsnum(mc1wide$MDS11),mdsnum(mc1wide$MDS12),
             mdsnum(mc1wide$MDS13),mdsnum(mc1wide$MDS14),
             mdsnum(mc1wide$MDS15),mdsnum(mc1wide$MDS16),
             mdsnum(mc1wide$MDS17),mdsnum(mc1wide$MDS18),
             mdsnum(mc1wide$MDS19),mdsnum(mc1wide$MDS20))
colnames(MDS) <- c(paste0("MDS0",1:9),paste0("MDS",10:20))
```

```
tabcorMDS <- cor(MDS[,c(11,13,14,15,1:10,12,16:20)])
colnames(tabcorMDS) <- rownames(tabcorMDS) <-
  paste0("i",c(11,13,14,15,1:10,12,16:20))
tabcorMDS[upper.tri(tabcorMDS,diag=TRUE)] <- NA
print(xtable(tabcorMDS[2:20,1:19]),size="tiny")
```

	i11	i13	i14	i15	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i12	i16	i17	i18	i19
i13	0.61																		
i14	0.60	0.65																	
i15	0.60	0.76	0.68																
i1	0.24	0.28	0.29	0.26															
i2	0.37	0.41	0.38	0.38	0.52														
i3	0.42	0.41	0.37	0.41	0.34	0.51													
i4	0.55	0.53	0.44	0.52	0.46	0.52	0.65												
i5	0.50	0.50	0.41	0.49	0.37	0.40	0.49	0.60											
i6	0.50	0.50	0.42	0.49	0.29	0.41	0.50	0.60	0.78										
i7	0.38	0.43	0.40	0.36	0.20	0.30	0.35	0.42	0.63	0.70									
i8	0.45	0.43	0.37	0.39	0.28	0.33	0.36	0.44	0.57	0.67	0.67								
i9	0.52	0.53	0.50	0.55	0.32	0.45	0.50	0.57	0.53	0.59	0.53	0.55							
i10	0.59	0.57	0.54	0.50	0.26	0.36	0.41	0.52	0.56	0.58	0.53	0.56	0.56						
i12	0.51	0.57	0.48	0.49	0.20	0.32	0.36	0.45	0.42	0.43	0.43	0.45	0.45	0.54					
i16	0.48	0.55	0.46	0.53	0.16	0.25	0.34	0.48	0.46	0.48	0.42	0.46	0.47	0.49	0.42				
i17	0.50	0.55	0.49	0.57	0.21	0.31	0.39	0.51	0.50	0.53	0.44	0.48	0.46	0.50	0.45	0.65			
i18	0.49	0.52	0.51	0.58	0.18	0.36	0.40	0.49	0.46	0.49	0.44	0.41	0.55	0.45	0.46	0.48	0.62		
i19	0.49	0.57	0.51	0.56	0.32	0.39	0.44	0.55	0.58	0.64	0.55	0.59	0.54	0.65	0.47	0.61	0.66	0.55	
i20	0.43	0.55	0.48	0.56	0.25	0.37	0.41	0.50	0.60	0.65	0.59	0.56	0.53	0.57	0.47	0.56	0.64	0.57	0.79

Used rainbow colors from **grDevices** (R Core Team, 2022). The blues and purples have the largest differences. The results are in Figure 1.

```
par(mar=c(4,4,4,7))
plot(tabcorMDS[2:20,1:19],col=rainbow(6),border="grey",ylab="",xlab="",
     main="Correlations MDS",axis.col=NULL, axis.row=NULL,font.main=1)
```

```r
axis(1,c(2.5,12),c("Other","Self"))
axis(2,c(8.5,18),c("Self","Other"))
rect(0.5,0.5,4.5,16.5,lwd=2)
```

One, two, or three dimensions?

```r
DIMTESTS(MDS,display=1)

##
##
## DIMTESTS results:

##        # of Factors:
## EMPKC              1
## HULL               1
## RAWPAR             2

fa.parallel(MDS,fa='pc')
```

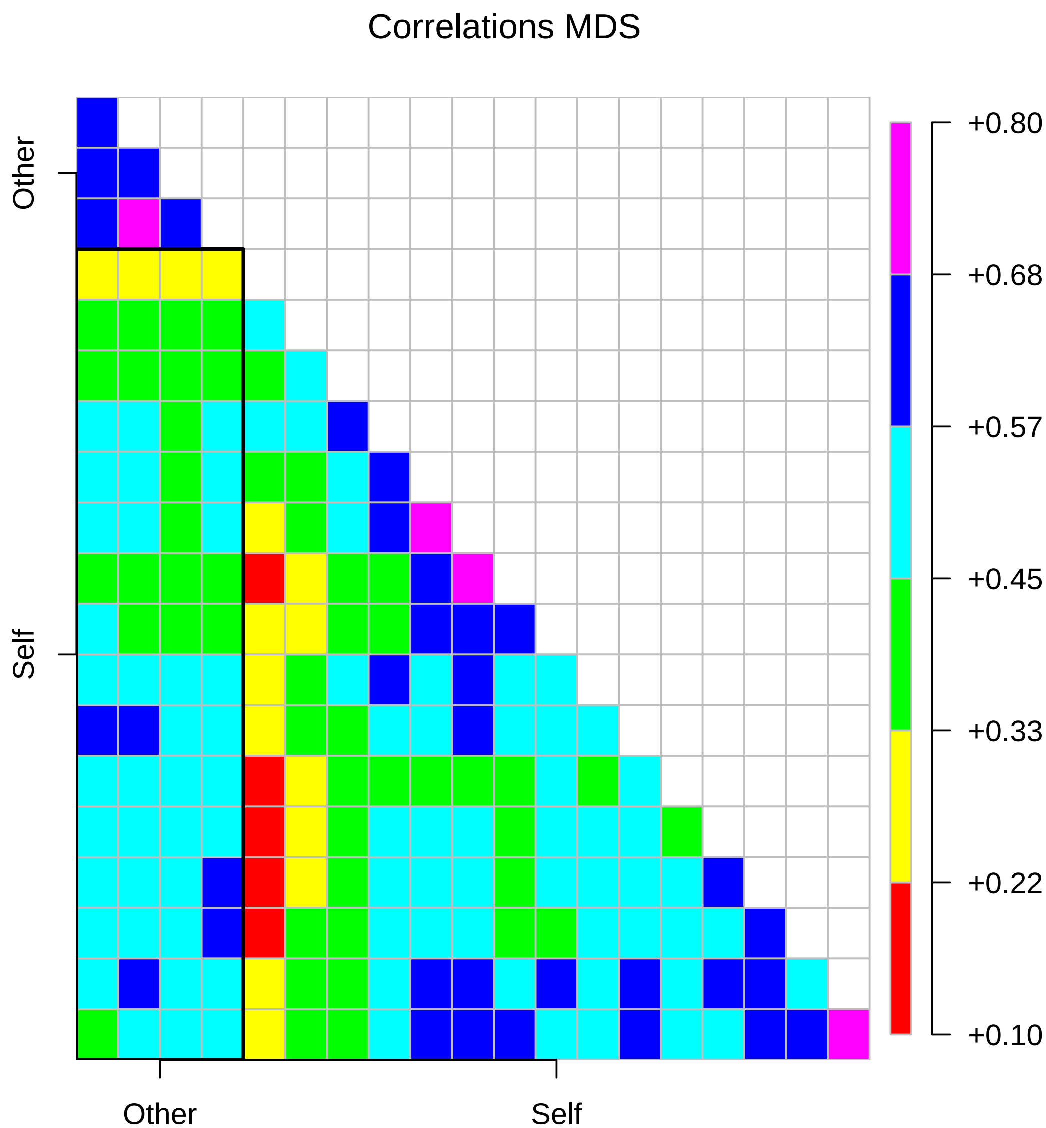
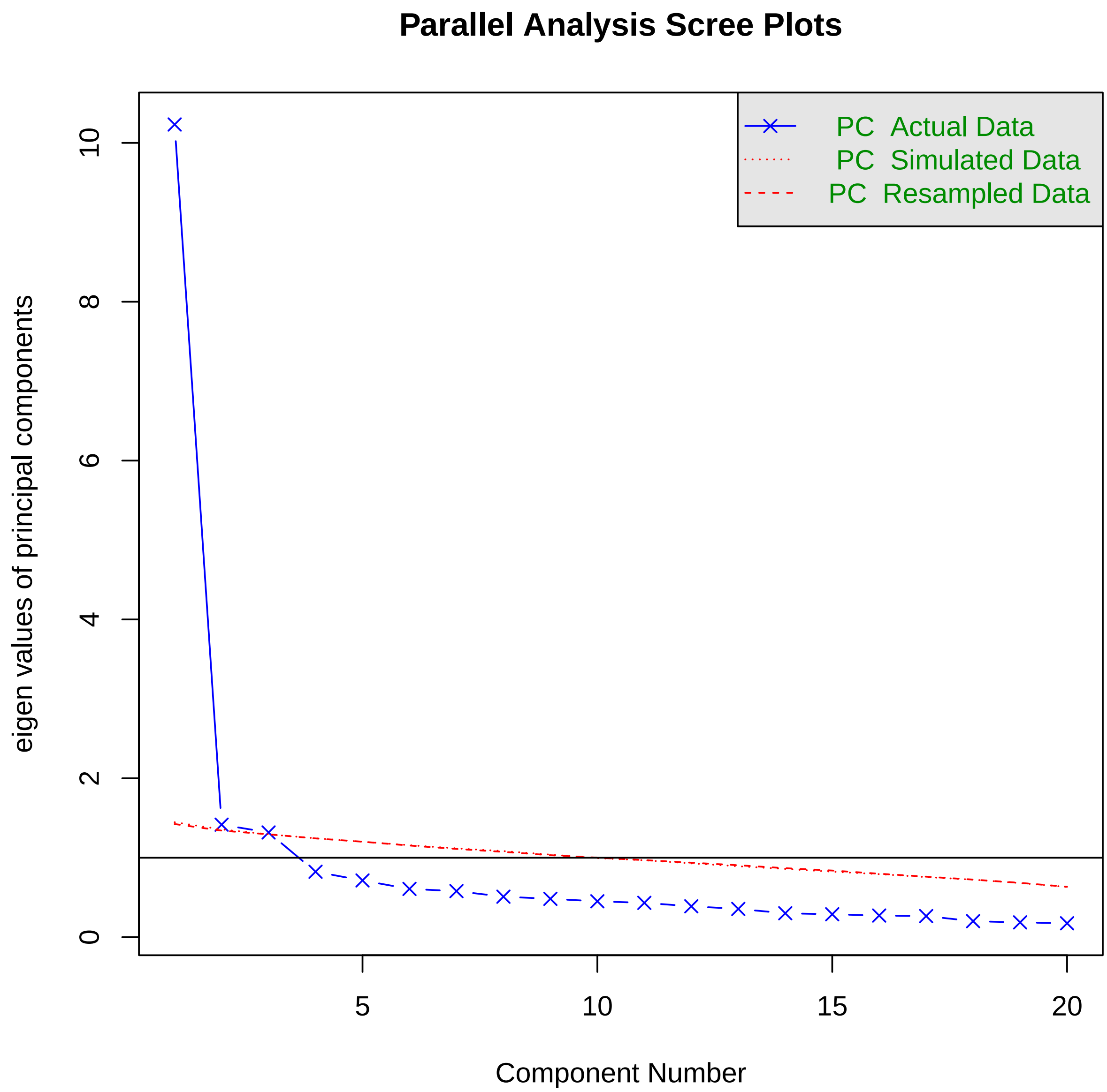


Figure 1: Corrs.




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

```
mds1 <- '
  md   =~ MDS11 + MDS13 + MDS14 + MDS15 +
          MDS01 + MDS02 + MDS03 + MDS04 + MDS05 + MDS06 +
          MDS07 + MDS08 + MDS09 + MDS10 + MDS12 + MDS16 +
          MDS17 + MDS18 + MDS19 + MDS20'
mds2 <- '
  other =~ MDS11 + MDS13 + MDS14 + MDS15
  self =~ MDS01 + MDS02 + MDS03 + MDS04 + MDS05 + MDS06 +
          MDS07 + MDS08 + MDS09 + MDS10 + MDS12 + MDS16 +
          MDS17 + MDS18 + MDS19 + MDS20'
cfa1 <- cfa(mds1,data=MDS)
```

```r
cfa2 <- cfa(mds2,data=MDS)
anova(cfa1,cfa2)

##
## Chi-Squared Difference Test
##
##       Df   AIC   BIC   Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## cfa2 169 23159 23321  925.87
## cfa1 170 23368 23526 1136.98     211.11 0.74555       1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(cfa2,fit.measures=TRUE)

## lavaan 0.6.15 ended normally after 43 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        41
##
##   Number of observations                           378
##
## Model Test User Model:
##
##   Test statistic                               925.870
##   Degrees of freedom                               169
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              5194.953
##   Degrees of freedom                               190
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.849
##   Tucker-Lewis Index (TLI)                       0.830
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -11538.596
##   Loglikelihood unrestricted model (H1)     -11075.660
##
##   Akaike (AIC)                               23159.191
##   Bayesian (BIC)                             23320.522
##   Sample-size adjusted Bayesian (SABIC)      23190.438
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.109
##   90 Percent confidence interval - lower         0.102
##   90 Percent confidence interval - upper         0.116
##   P-value H_0: RMSEA <= 0.050                    0.000
```

```
##   P-value H_0: RMSEA >= 0.080                    1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.065
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   other =~
##     MDS11             1.000
##     MDS13             1.311    0.080   16.493    0.000
##     MDS14             1.141    0.077   14.807    0.000
##     MDS15             1.348    0.082   16.477    0.000
##   self =~
##     MDS01             1.000
##     MDS02             1.303    0.196    6.635    0.000
##     MDS03             1.431    0.205    6.995    0.000
##     MDS04             1.728    0.232    7.460    0.000
##     MDS05             1.850    0.244    7.590    0.000
##     MDS06             2.011    0.261    7.696    0.000
##     MDS07             1.753    0.237    7.405    0.000
##     MDS08             1.656    0.223    7.435    0.000
##     MDS09             1.722    0.231    7.465    0.000
##     MDS10             1.507    0.201    7.515    0.000
##     MDS12             1.403    0.199    7.065    0.000
##     MDS16             1.391    0.191    7.281    0.000
##     MDS17             1.531    0.205    7.463    0.000
##     MDS18             1.646    0.225    7.306    0.000
##     MDS19             1.534    0.199    7.720    0.000
##     MDS20             1.644    0.214    7.670    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   other ~~
##     self              0.500    0.079    6.352    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .MDS11             0.731    0.060   12.102    0.000
##    .MDS13             0.523    0.054    9.710    0.000
##    .MDS14             0.752    0.065   11.652    0.000
##    .MDS15             0.557    0.057    9.740    0.000
##    .MDS01             2.399    0.176   13.616    0.000
##    .MDS02             1.919    0.143   13.468    0.000
##    .MDS03             1.588    0.119   13.339    0.000
##    .MDS04             1.169    0.090   12.937    0.000
##    .MDS05             1.016    0.080   12.679    0.000
```

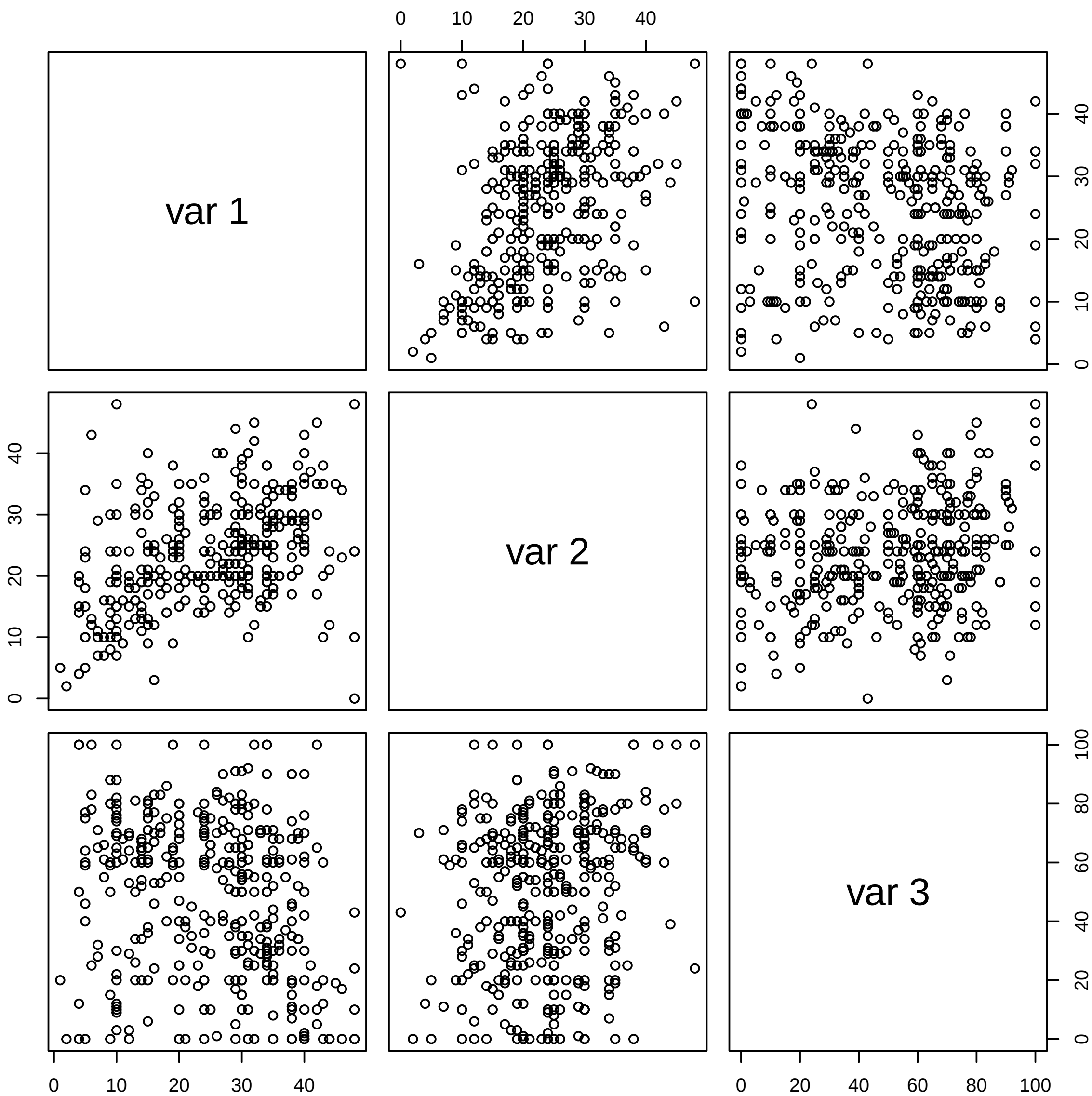
```
##    .MDS06             0.901    0.073   12.321    0.000
##    .MDS07             1.331    0.102   13.016    0.000
##    .MDS08             1.127    0.087   12.976    0.000
##    .MDS09             1.149    0.089   12.929    0.000
##    .MDS10             0.798    0.062   12.844    0.000
##    .MDS12             1.404    0.106   13.303    0.000
##    .MDS16             1.028    0.078   13.151    0.000
##    .MDS17             0.913    0.071   12.934    0.000
##    .MDS18             1.384    0.105   13.127    0.000
##    .MDS19             0.484    0.040   12.205    0.000
##    .MDS20             0.649    0.052   12.425    0.000
##     other             0.861    0.107    8.064    0.000
##     self              0.433    0.112    3.863    0.000
```

This two factor model fits better than the one factor one, and is well-suited for our analyses. However, the RMSEA is about 1, so there is room for improvement, and we will be discussing this with Nash.

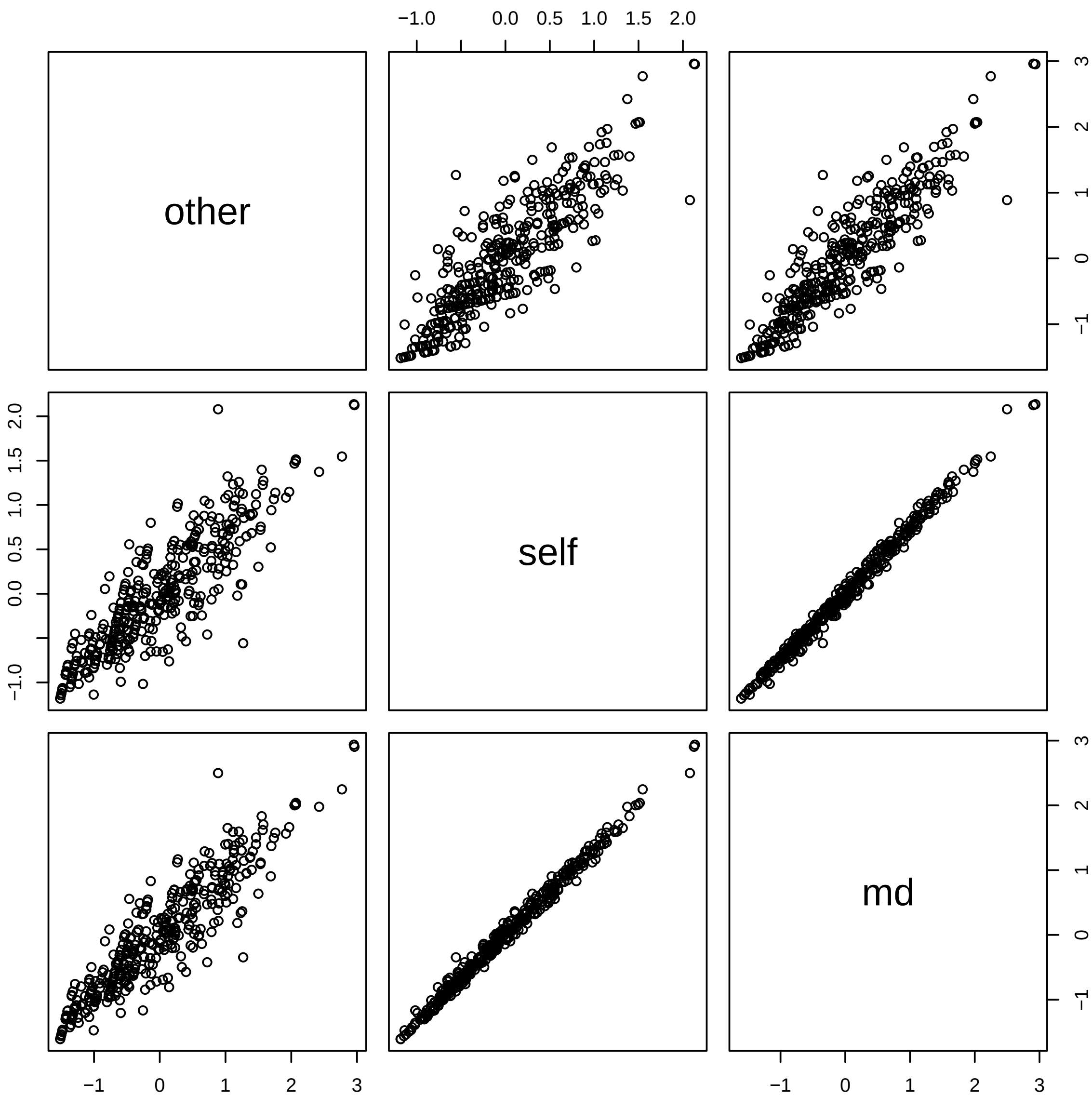
```
mdsfac <- cbind(predict(cfa2),predict(cfa1))
selfratings <- cbind(mc1wide$SubjPerformance_1, mc1wide$SubjPerformance_2,mc1wide$Influence_1)
```

11 Looking at correlations at the person level

```
pairs(selfratings)
```




```
pairs(mdsfac)
```




```
dim(mdsfac);dim(selfratings);dim(cms$person)

## [1] 378   3
## [1] 378   3
## [1] 378   2

# now 8, but ...
vars7 <- cbind(mdsfac,selfratings,cms$person)
ctab <- matrix(rep("a",64),nrow=8,ncol=8) # so strings
apac <- function(x) sub("0.",".",sprintf("%0.3f",x),fixed=TRUE)
for (i in 1:8)
  for (j in 1:8){
    if (i == j) ctab[i,i] <- sprintf("%0.3f",sd(vars7[,i]))
    if (i > j) ctab[i,j] <- apac(cor(vars7[,c(i,j)])[1,2])
```

```
    if (j > i) {
        cis <- cor.test(vars7[,i],vars7[,j])$conf.int
        ciss <- paste0("(",apac(cis[1]),", ",apac(cis[2]),")")
        ctab[i,j] <- ciss}}
rownames(ctab) <-c("MDSOther","MDSSelf","MDS1dim","MeRight",
  "OtherRight","Influence","Accuracy","MemConf")
colnames(ctab) <-   c("MDSO","MDSS","MDS1","Me","Other","Infl","Acc","MC")
```

The MDS scores correlate less than $|r| = .1$ with other measures.

```
print(xtable(ctab),size="footnotesize")
```

	MDSO	MDSS	MDS1	Me	Other	Infl	Acc	MC
MDSOther	0.886	(.843, .892)	(.894, .928)	(-.182, .018)	(-.084, .118)	(.136, .326)	(-.175, .026)	(-.077, .125)
MDSSelf	.869	0.642	(.994, .996)	(-.152, .050)	(-.065, .137)	(.139, .330)	(-.182, .019)	(-.093, .109)
MDS1dim	.913	.995	0.860	(-.159, .042)	(-.068, .133)	(.145, .335)	(-.183, .018)	(-.089, .113)
MeRight	-.083	-.052	-.059	11.094	(.351, .514)	(-.324, -.133)	(.276, .451)	(-.165, .036)
OtherRight	.017	.037	.033	.436	8.462	(.089, .284)	(-.146, .056)	(.097, .291)
Influence	.233	.237	.242	-.230	.188	26.853	(-.556, -.401)	(.303, .474)
Accuracy	-.075	-.083	-.083	.367	-.046	-.483	0.946	(-.760, -.661)
MemConf	.024	.008	.012	-.065	.196	.392	-.714	1.056

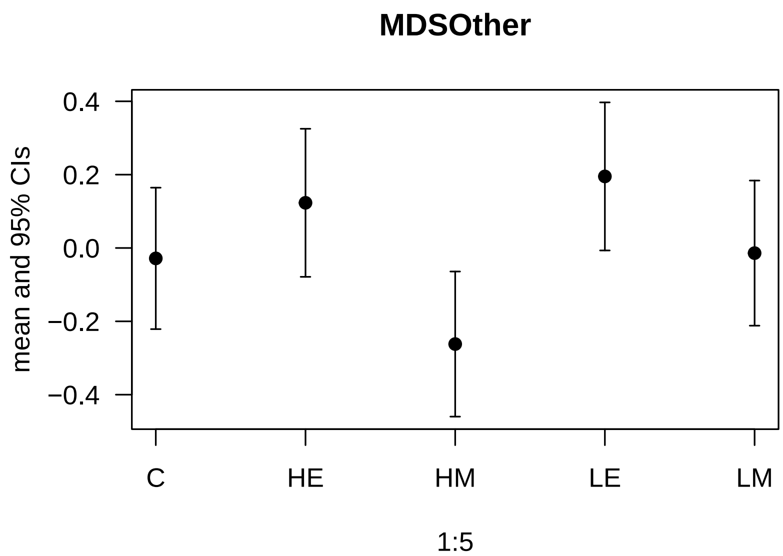
12 Comparing the conditional modes (and other things) with condition

```
newvars <- vars7
newvars$cond <- mc1wide$cond
newvars$exp <- newvars$cond != "C"
newvars$low <- newvars$cond == "LM" | newvars$cond =="LE"
newvars$manage <- newvars$cond == "LM" | newvars$cond =="HM"
#checked
```

These ANOVAs will be done in a loop for all seven of the variables. Three models will be run for each. First the model with no intercept will be run to get the confidence intervals for the individual conditions. Then the ANOVA comparing each to the control, and finally the 2×2 ANOVA without the control group. **car** (Fox & Weisberg, 2011) is loaded for Type II sum of squares, as discussed in their book. The key one is the one for memory conformity.

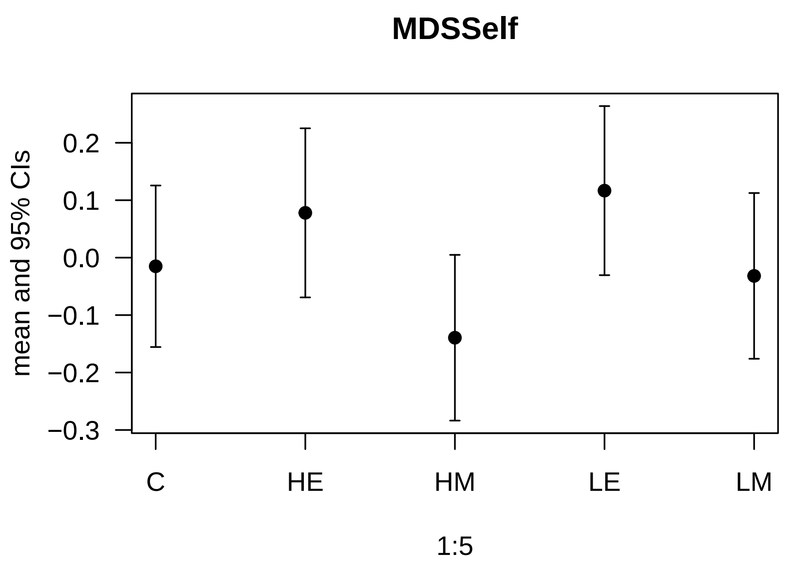
```
cis <- {}
# eventually saving the anova stuff too for printing
for (i in 1:7){
  print(rownames(ctab)[i])
  cisi <- confint(lm(newvars[,i] ~ 0 + newvars$cond))
  cis <- rbind(cis,cisi)
  plot(1:5,mv <- tapply(newvars[,i],newvars$cond,mean),las=1,
       main=rownames(ctab)[i],xaxt='n',ylim=range(cisi),
       ylab="mean and 95% CIs")
  axis(1,1:5,levels(newvars$cond))
  errbar(1:5,mv,cisi[,1],cisi[,2],add=TRUE)
  m2 <- lm(newvars[,i] ~ newvars$cond) #control is baseline
  summary(m2)
```

```
  m3 <- lm(newvars[newvars$cond != "C",i] ~
    newvars$low[newvars$cond != "C"]*newvars$manage[newvars$cond != "C"])
  a3 <- Anova(m3,type="II")
  #mod <- aov(overc~as.factor(group))
  print(eta_squared(a3, ci=.95))
  print(omega_squared(a3, ci=.95))

  rownames(a3) <- c("low","manage","interaction","residuals")
  print(a3)
}

## [1] "MDSOther"
```



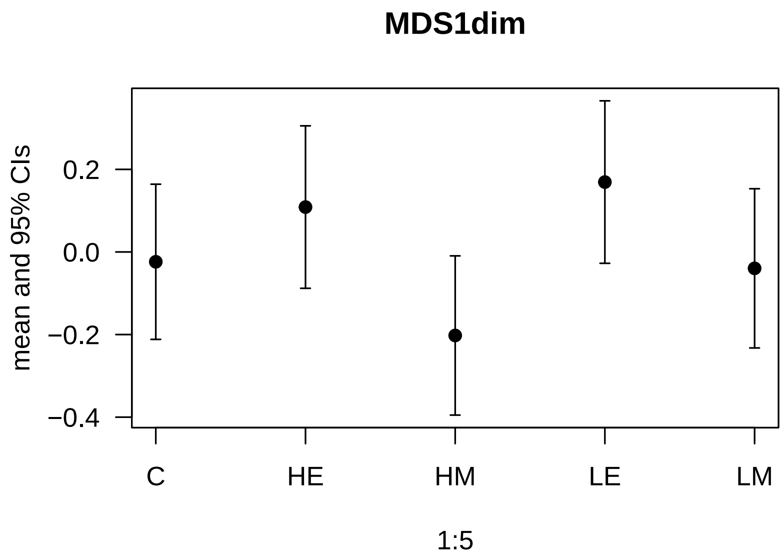

```
## # Effect Size for ANOVA (Type II)
##
## Parameter                                                          | Eta2 (partial) |       95% CI
## ---------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                   |       8.56e-03 | [0.00, 1.00]
## newvars$manage[newvars$cond != "C"]                                |           0.03 | [0.01, 1.00]
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |       2.55e-03 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].# Effect Size for ANOVA (Type II)
##
## Parameter                                                          | Omega2 (partial) |       95% C
## ---------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                   |         5.13e-03 | [0.00, 1.0
## newvars$manage[newvars$cond != "C"]                                |             0.02 | [0.00, 1.0
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |             0.00 | [0.00, 1.0
##
## - One-sided CIs: upper bound fixed at [1.00].Anova Table (Type II tests)
##
## Response: newvars[newvars$cond != "C", i]
##              Sum Sq  Df F value   Pr(>F)
```

```
## low            1.949   1  2.5375 0.112242
## manage         6.574   1  8.5618 0.003701 **
## interaction    0.577   1  0.7515 0.386724
## residuals    225.759 294
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "MDSSelf"
```

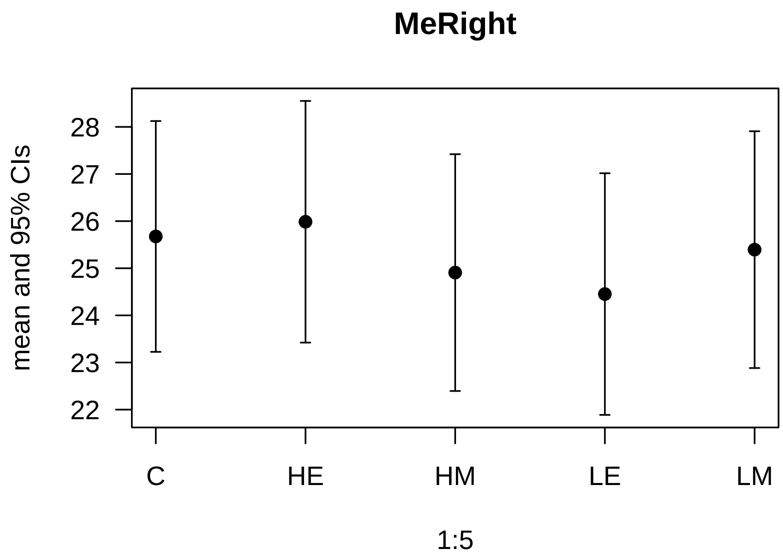



```
## # Effect Size for ANOVA (Type II)
##
## Parameter                                                              | Eta2 (partial) |       95% CI
## ----------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                       |       3.26e-03 | [0.00, 1.00]
## newvars$manage[newvars$cond != "C"]                                    |           0.02 | [0.00, 1.00]
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |       7.12e-04 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].# Effect Size for ANOVA (Type II)
##
## Parameter                                                              | Omega2 (partial) |       95% C
## ------------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                       |             0.00 | [0.00, 1.0
## newvars$manage[newvars$cond != "C"]                                    |             0.02 | [0.00, 1.0
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |             0.00 | [0.00, 1.0
##
## - One-sided CIs: upper bound fixed at [1.00].Anova Table (Type II tests)
##
## Response: newvars[newvars$cond != "C", i]
##              Sum Sq  Df F value  Pr(>F)
## low           0.406   1  0.9620 0.32750
## manage        2.491   1  5.8983 0.01575 *
## interaction   0.088   1  0.2095 0.64751
## residuals   124.182 294
## ---
```

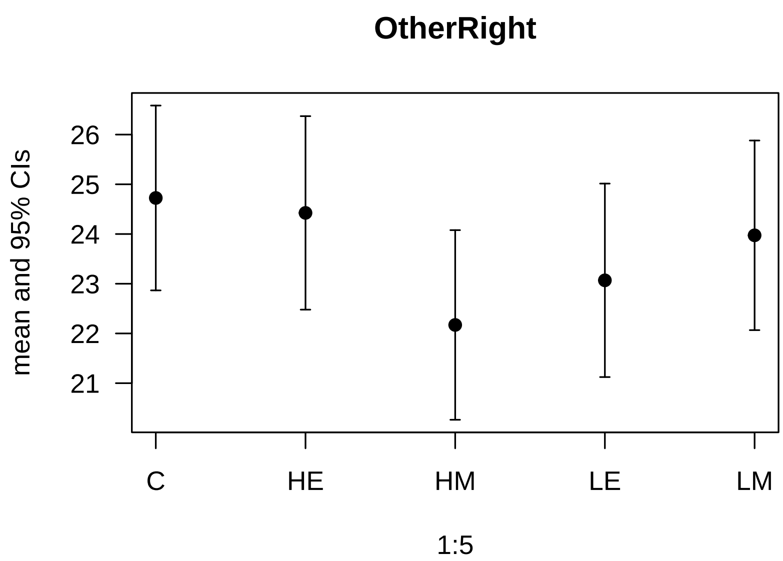
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "MDS1dim"
```



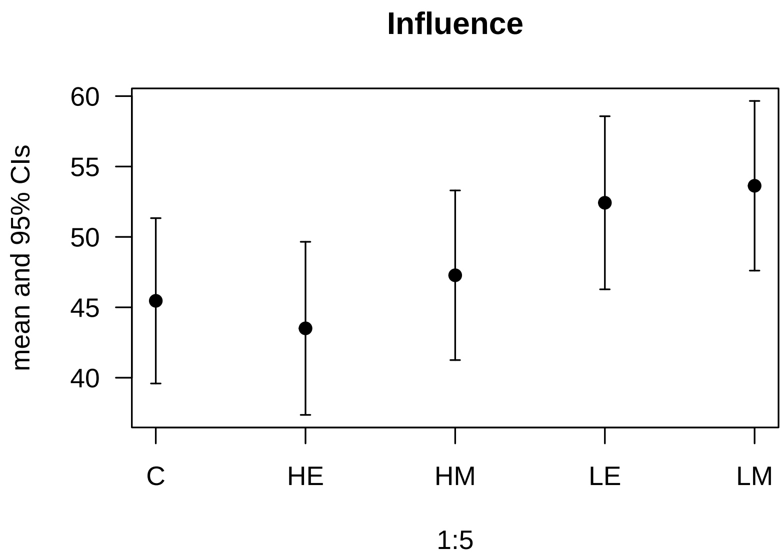

```
## # Effect Size for ANOVA (Type II)
##
## Parameter                                                            | Eta2 (partial) |       95% CI
## -------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                     |       4.27e-03 | [0.00, 1.00]
## newvars$manage[newvars$cond != "C"]                                  |           0.02 | [0.00, 1.00]
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |       8.80e-04 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].# Effect Size for ANOVA (Type II)
##
## Parameter                                                            | Omega2 (partial) |       95% C
## -------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                     |         8.74e-04 | [0.00, 1.0
## newvars$manage[newvars$cond != "C"]                                  |             0.02 | [0.00, 1.0
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |             0.00 | [0.00, 1.0
##
## - One-sided CIs: upper bound fixed at [1.00].Anova Table (Type II tests)
##
## Response: newvars[newvars$cond != "C", i]
##              Sum Sq  Df F value  Pr(>F)
## low           0.944   1  1.2608 0.26242
## manage        5.031   1  6.7191 0.01002 *
## interaction   0.194   1  0.2589 0.61123
## residuals   220.131 294
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "MeRight"
```



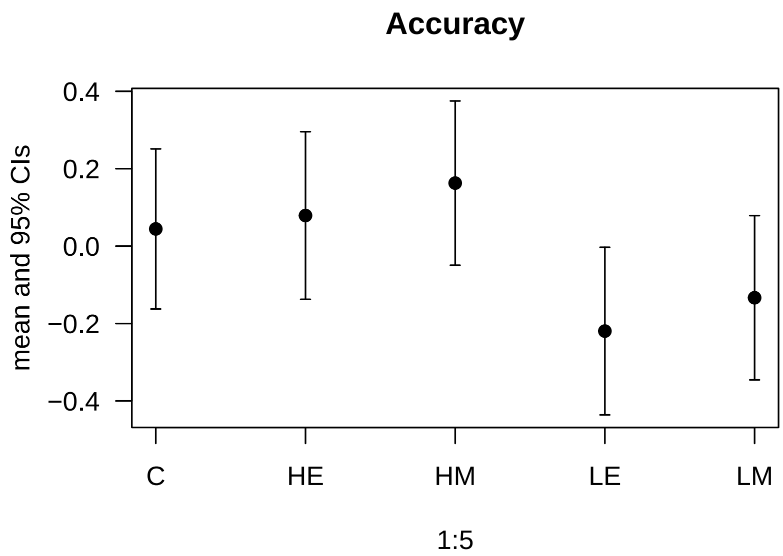

```
## # Effect Size for ANOVA (Type II)
##
## Parameter                                                          | Eta2 (partial) |       95% CI
## -------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                   |       5.08e-04 | [0.00, 1.00]
## newvars$manage[newvars$cond != "C"]                                |       9.23e-06 | [0.00, 1.00]
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |       2.04e-03 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].# Effect Size for ANOVA (Type II)
##
## Parameter                                                          | Omega2 (partial) |       95% C
## ---------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                   |             0.00 | [0.00, 1.0
## newvars$manage[newvars$cond != "C"]                                |             0.00 | [0.00, 1.0
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |             0.00 | [0.00, 1.0
##
## - One-sided CIs: upper bound fixed at [1.00].Anova Table (Type II tests)
##
## Response: newvars[newvars$cond != "C", i]
##             Sum Sq  Df F value Pr(>F)
## low             19   1  0.1494 0.6994
## manage           0   1  0.0027 0.9585
## interaction     76   1  0.6019 0.4385
## residuals    37146 294
## [1] "OtherRight"
```




```
## # Effect Size for ANOVA (Type II)
##
## Parameter                                                              | Eta2 (partial) |       95% CI
## -----------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                       |       2.34e-04 | [0.00, 1.00]
## newvars$manage[newvars$cond != "C"]                                    |       1.63e-03 | [0.00, 1.00]
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |       8.90e-03 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].# Effect Size for ANOVA (Type II)
##
## Parameter                                                              | Omega2 (partial) |       95% C
## ------------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                       |             0.00 | [0.00, 1.0
## newvars$manage[newvars$cond != "C"]                                    |             0.00 | [0.00, 1.0
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |         5.47e-03 | [0.00, 1.0
##
## - One-sided CIs: upper bound fixed at [1.00].Anova Table (Type II tests)
##
## Response: newvars[newvars$cond != "C", i]
##              Sum Sq  Df F value Pr(>F)
## low             4.8   1  0.0689 0.7932
## manage         33.9   1  0.4811 0.4885
## interaction   185.8   1  2.6403 0.1053
## residuals   20685.2 294
## [1] "Influence"
```




```
## # Effect Size for ANOVA (Type II)
##
## Parameter                                                              | Eta2 (partial) |       95% CI
## ---------------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                       |           0.02 | [0.00, 1.00]
## newvars$manage[newvars$cond != "C"]                                    |       2.16e-03 | [0.00, 1.00]
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |       5.74e-04 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].# Effect Size for ANOVA (Type II)
##
## Parameter                                                              | Omega2 (partial) |       95% C
## -----------------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                       |             0.02 | [0.00, 1.0
## newvars$manage[newvars$cond != "C"]                                    |             0.00 | [0.00, 1.0
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |             0.00 | [0.00, 1.0
##
## - One-sided CIs: upper bound fixed at [1.00].Anova Table (Type II tests)
##
## Response: newvars[newvars$cond != "C", i]
##             Sum Sq  Df F value  Pr(>F)
## low           4315   1  5.9590 0.01523 *
## manage         461   1  0.6367 0.42557
## interaction    122   1  0.1688 0.68146
## residuals   212905 294
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Accuracy"
```




```
## # Effect Size for ANOVA (Type II)
##
## Parameter                                                            | Eta2 (partial) |       95% CI
## --------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                     |           0.02 | [0.00, 1.00]
## newvars$manage[newvars$cond != "C"]                                  |       1.98e-03 | [0.00, 1.00]
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |       3.74e-07 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].# Effect Size for ANOVA (Type II)
##
## Parameter                                                            | Omega2 (partial) |       95% C
## ----------------------------------------------------------------------------------------------------
## newvars$low[newvars$cond != "C"]                                     |             0.02 | [0.00, 1.0
## newvars$manage[newvars$cond != "C"]                                  |             0.00 | [0.00, 1.0
## newvars$low[newvars$cond != "C"]:newvars$manage[newvars$cond != "C"] |             0.00 | [0.00, 1.0
##
## - One-sided CIs: upper bound fixed at [1.00].Anova Table (Type II tests)
##
## Response: newvars[newvars$cond != "C", i]
##              Sum Sq  Df F value   Pr(>F)
## low           6.586   1  7.1449 0.007938 **
## manage        0.537   1  0.5829 0.445772
## interaction   0.000   1  0.0001 0.991636
## residuals   271.018 294
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#cis  add cohen's d to the confint
```

So this is a good fairly clear finding. Low power people are more influenced by the other person whether it is evaluative or managerial power.

13 Are response times longer with the other making an error?

... and other things with response times.

As noted above, they already will have been looking at this for 4s, so this is NOT a good measure.

The response times matrix as `timelast`. Here I will make them long and add them to the long data object. Some are very slow, suggesting the doorbell rung, or something like that. However, the longest was only 158.14 seconds, so just over two minutes, which is not huge.

```
dim(long1)

## [1] 18144     8

long1$rt <- c(timelast)
dim(long1)

## [1] 18144     9

skewness(long1$rt)

## [1] 42.04122

skewness(log(long1$rt))

## [1] 1.468619

geomean <- function(x) exp(mean(log(x)))
tapply(long1$rt,long1$condit,geomean)

##         C        HE        HM        LE        LM
## 0.9946217 1.0093365 1.0143069 0.9187191 0.9940261

tapply(long1$rt,long1$condit,mean)

##        C       HE       HM       LE       LM
## 1.211836 1.261740 1.246973 1.126865 1.204872

tapply(long1$rt,long1$partRight,geomean)

##         0         1
## 1.0239142 0.9730029

tapply(long1$rt,long1$oright,geomean)

##         0         1
## 0.9763932 0.9957648

tapply(long1$rt,
  list(long1$oright,long1$partRight),geomean)

##          0         1
## 0 1.016271 0.9571212
## 1 1.037488 0.9862755
```

The differences by condition are person variables, so best to examine these in models.

By condition, here just all 5. It is non-significant, as I imagined it would be.

```r
m0 <- lmer(log(rt)~1 + (1|wordno) + (1|person),data=long1,REML=FALSE)
m1 <- update(m0, .~. + condit)
anova(m0,m1)

## Data: long1
## Models:
## m0: log(rt) ~ 1 + (1 | wordno) + (1 | person)
## m1: log(rt) ~ (1 | wordno) + (1 | person) + condit
##    npar   AIC   BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m0    4 19102 19133 -9546.9    19094
## m1    8 19105 19168 -9544.6    19089 4.5822  4     0.3329
```

The question is whether confronted with errant information whether people are slower.

```r
dim(long1)

## [1] 18144     9

m0 <- lmer(log(rt)~1 + (1|wordno) + (1|person),data=long1,REML=FALSE)
m1 <- update(m0, .~. + oright)
m2 <- update(m0, .~. + partRight)
m3 <- update(m2, .~. + oright)
m4 <- update(m3, .~. + oright:partRight)
rightsn <- long1$oright + 2*long1$partRight
rights <- vector(length=length(rightsn))
rights[rightsn == 0] <- "OwSw"
rights[rightsn == 1] <- "OrSw"
rights[rightsn == 2] <- "OwSr"
rights[rightsn == 3] <- "OrSr"
dim(long1)

## [1] 18144     9

long1$rights <- rights
dim(long1)

## [1] 18144    10

table(long1$rights,long1$oright,useNA="always")

##
##           0    1 <NA>
##   OrSr    0 7356    0
##   OrSw    0 1716    0
##   OwSr 6056    0    0
##   OwSw 3016    0    0
##   <NA>    0    0    0

table(long1$rights,long1$partRight,useNA="always")

##
##           0    1 <NA>
##   OrSr    0 7356    0
##   OrSw 1716    0    0
##   OwSr    0 6056    0
##   OwSw 3016    0    0
##   <NA>    0    0    0
```

```
table(long1$rights,long1$condit,useNA="always")

##
##            C   HE   HM   LE   LM <NA>
##   OrSr 1532 1411 1484 1446 1483    0
##   OrSw  388  341  340  306  341    0
##   OwSr 1313 1208 1277 1087 1171    0
##   OwSw  607  544  547  665  653    0
##   <NA>    0    0    0    0    0    0
```

14 Now confidence

Confidence can be dealt with a few way, largely in how being right is used with it. This has a bimodal distribution. I am going to keep it like that.

```
dim(long1);length(c(confid))

## [1] 18144    10
## [1] 18144

long1$conf <- c(confid)

m0conf <- lmer(conf ~ 1 + (1|wordno) + (1|person),data=long1,REML=FALSE)
m1conf <- update(m0conf, .~. + partRight)
m2conf <- update(m0conf, .~. + oright)
m3conf <- update(m2conf, .~. + partRight)
m4conf <- update(m3conf, .~. + partRight:oright)

m0conf <- lmer(normrank(conf,delta=.7) ~ 1 + (1|wordno) + (1|person),data=long1,REML=FALSE)
m1conf <- update(m0conf, .~. + partRight)
m2conf <- update(m0conf, .~. + oright)
m3conf <- update(m2conf, .~. + partRight)
m4conf <- update(m3conf, .~. + partRight:oright)

with(long1,tapply(conf,list(partRight,oright),mean))

##          0        1
## 0 60.19330 60.79604
## 1 74.27246 74.23545

with(long1,tapply(normrank(conf,delta=.7),list(partRight,oright),mean))

##            0          1
## 0 -0.5048927 -0.4422235
## 1  0.1956323  0.1765643
```

```
anova(m0conf,m1conf,m2conf,m3conf,m4conf)

## Data: long1
## Models:
## m0conf: normrank(conf, delta = 0.7) ~ 1 + (1 | wordno) + (1 | person)
## m1conf: normrank(conf, delta = 0.7) ~ (1 | wordno) + (1 | person) + partRight
## m2conf: normrank(conf, delta = 0.7) ~ (1 | wordno) + (1 | person) + oright
```

```
## m3conf: normrank(conf, delta = 0.7) ~ (1 | wordno) + (1 | person) + oright + partRight
## m4conf: normrank(conf, delta = 0.7) ~ (1 | wordno) + (1 | person) + oright + partRight + oright:partR
##        npar   AIC   BIC logLik deviance   Chisq Df Pr(>Chisq)
## m0conf    4 41348 41380 -20670    41340
## m1conf    5 39784 39823 -19887    39774 1566.82  1     <2e-16 ***
## m2conf    5 41347 41386 -20669    41337    0.00  0
## m3conf    6 39785 39832 -19887    39773 1564.03  1     <2e-16 ***
## m4conf    7 39786 39841 -19886    39772    1.01  1     0.3149
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m4conf)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: normrank(conf, delta = 0.7) ~ (1 | wordno) + (1 | person) + oright +
##     partRight + oright:partRight
##    Data: long1
##
##      AIC      BIC   logLik deviance df.resid
##  39786.3  39840.9 -19886.1  39772.3    18137
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.6332 -0.6724  0.0035  0.6478  3.5611
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  person   (Intercept) 0.2078   0.4559
##  wordno   (Intercept) 0.0280   0.1673
##  Residual             0.4878   0.6984
## Number of obs: 18144, groups:  person, 378; wordno, 48
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)      -0.375676   0.043528  -8.631
## oright            0.003175   0.053074   0.060
## partRight         0.506956   0.016717  30.326
## oright:partRight  0.025844   0.025712   1.005
##
## Correlation of Fixed Effects:
##             (Intr) oright prtRgh
## oright      -0.581
## partRight   -0.256  0.207
## orght:prtRg  0.165 -0.364 -0.643
```

15 The Text Analysis

The first task is checking for gibberish. I think that is a human task. They all look like words. I am not printing these out here, but do look at them. Technically, this could go at the start with the other exclusions, but since there were none for this I have kept this here.

```r
mc1wide$ControlWrite[!is.na(mc1wide$ControlWrite)]
mc1wide$LEWrite[!is.na(mc1wide$LEWrite)]
mc1wide$LMWrite[!is.na(mc1wide$LMWrite)]
mc1wide$HEWrite[!is.na(mc1wide$HEWrite)]
mc1wide$HMWrite[!is.na(mc1wide$HMWrite)]
```

Creating a single text variable.

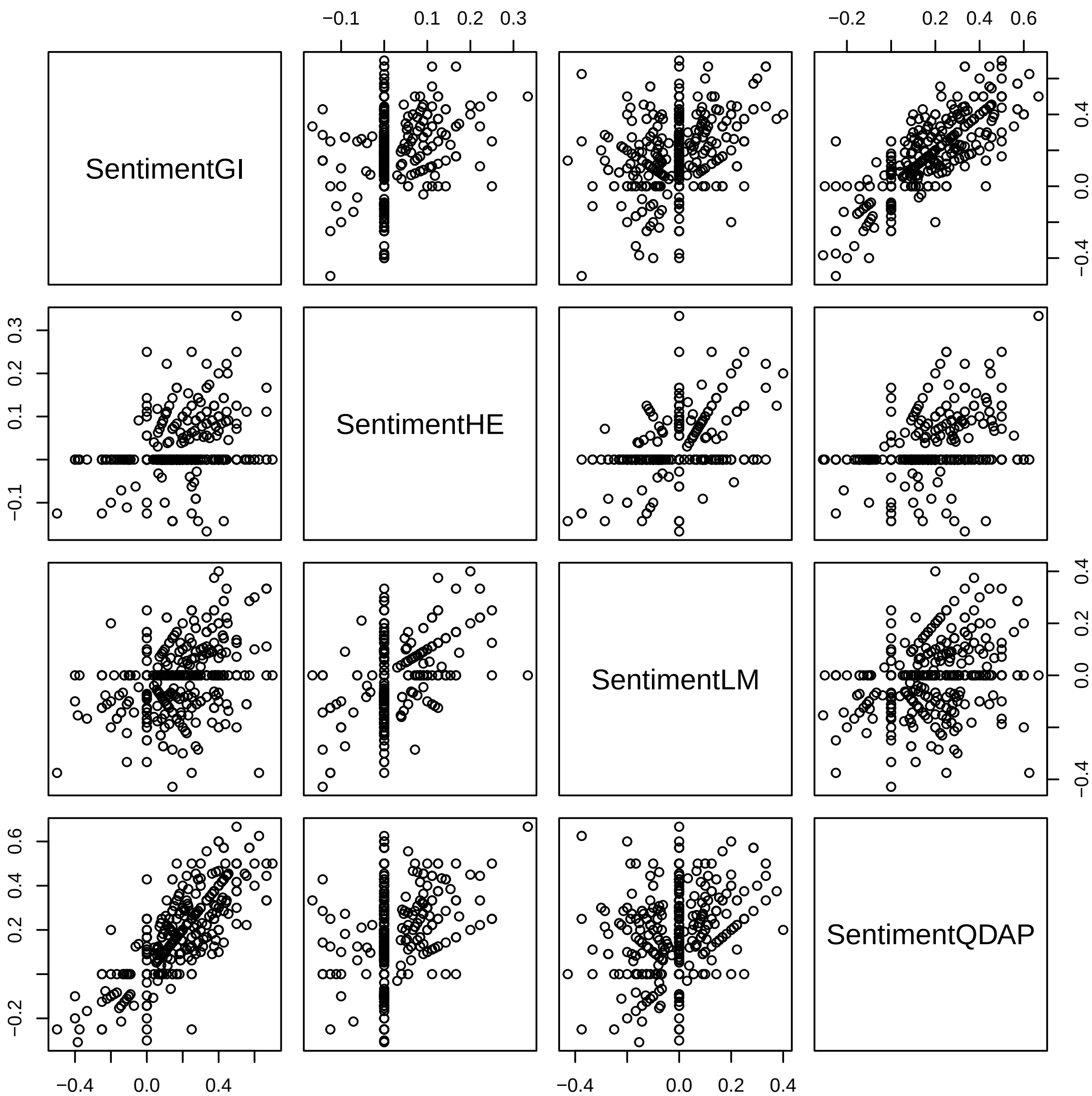
```r
mc1wide$textans <- mc1wide$ControlWrite
mc1wide$textans[!is.na(mc1wide$LEWrite)] <- mc1wide$LEWrite[!is.na(mc1wide$LEWrite)]
mc1wide$textans[!is.na(mc1wide$LMWrite)] <- mc1wide$LMWrite[!is.na(mc1wide$LMWrite)]
mc1wide$textans[!is.na(mc1wide$HEWrite)] <- mc1wide$HEWrite[!is.na(mc1wide$HEWrite)]
mc1wide$textans[!is.na(mc1wide$HMWrite)] <- mc1wide$HMWrite[!is.na(mc1wide$HMWrite)]
sum(is.na(mc1wide$textans))

## [1] 0
```

Doing the fairly standard things of removing stuff.

```r
tt <- mc1wide$textans
tt <- removePunctuation(tt)
tt <- removeNumbers(tt)
tt <- removeWords(tt,stopwords("english"))
tt <- lemmatize_strings(tt)
```

There are lots of functions in lots of packages for sentiment analysis. **sentimentr** (Rinker, 2019), **SentimentAnalysis** (Proellochs & Feuerriegel, 2021), **tidytext** (Silge & Robinson, 2016), **syuzhet** (Jockers, 2015), *etc.* The goal here is to get a single measure for people's views about the task. The `analyzeSentiment` from **SentimentAnalysis** does this with multiple sentiment files. It uses four dictionaries, and these are compared, and their average taken for each individual.

```r
sent <- analyzeSentiment(tt)
pairs(sent[,c(2,5,8,12)])
```




```r
cor(cbind(sent[,c(2,5,8,12)],rowMeans(sent[,c(2,5,8,12)])))
```

```
##                                SentimentGI SentimentHE SentimentLM
## SentimentGI                      1.0000000   0.2765649   0.3605177
## SentimentHE                      0.2765649   1.0000000   0.4188457
## SentimentLM                      0.3605177   0.4188457   1.0000000
## SentimentQDAP                    0.7777403   0.3217056   0.2883371
## rowMeans(sent[, c(2, 5, 8, 12)])  0.8969228   0.5155913   0.6228726
##                                SentimentQDAP rowMeans(sent[, c(2, 5, 8, 12)])
## SentimentGI                        0.7777403                        0.8969228
## SentimentHE                        0.3217056                        0.5155913
## SentimentLM                        0.2883371                        0.6228726
## SentimentQDAP                      1.0000000                        0.8739146
## rowMeans(sent[, c(2, 5, 8, 12)])    0.8739146                        1.0000000
```

```r
names(sent)[c(2,5,8,12)]

## [1] "SentimentGI"   "SentimentHE"   "SentimentLM"   "SentimentQDAP"

eigen(cor(sent[,c(2,5,8,12)]))$values

## [1] 2.2474462 0.9501480 0.5897339 0.2126719

factanal(sent[,c(2,5,8,12)],1)

##
## Call:
## factanal(x = sent[, c(2, 5, 8, 12)], factors = 1)
##
## Uniquenesses:
##   SentimentGI   SentimentHE   SentimentLM SentimentQDAP
##         0.195         0.875         0.848         0.255
##
## Loadings:
##               Factor1
## SentimentGI   0.897
## SentimentHE   0.353
## SentimentLM   0.390
## SentimentQDAP 0.863
##
##                Factor1
## SS loadings      1.827
## Proportion Var   0.457
##
## Test of the hypothesis that 1 factor is sufficient.
## The chi square statistic is 53.5 on 2 degrees of freedom.
## The p-value is 2.42e-12
```

These differ (on the oneway ANOVA) by condition, with the evaluation conditions having the highest sentiment, and the control being the lowest.

```r
sentvals <- rowMeans(sent[,c(2,5,8,12)])
tapply(sentvals,mc1wide$cond,mean)

##          C         HE         HM         LE         LM
## 0.08190388 0.12897275 0.08596830 0.10742123 0.09066301

oneway.test(sentvals~mc1wide$cond)

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  sentvals and mc1wide$cond
## F = 2.3438, num df = 4.00, denom df = 185.58, p-value = 0.05641

pairwise.t.test(sentvals,mc1wide$cond)

##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  sentvals and mc1wide$cond
```

```
##
##    C     HE    HM    LE
## HE 0.098 -     -     -
## HM 1.000 0.176 -     -
## LE 1.000 1.000 1.000 -
## LM 1.000 0.300 1.000 1.000
##
## P value adjustment method: holm
```

```
pairwise.t.test(sentvals,mc1wide$cond,method="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  sentvals and mc1wide$cond
##
##    C     HE    HM    LE
## HE 0.098 -     -     -
## HM 1.000 0.176 -     -
## LE 1.000 1.000 1.000 -
## LM 1.000 0.300 1.000 1.000
##
## P value adjustment method: holm
```

If you treat the two effects together that would be less than $p < .05$, but these are exploratory anyway, so . . . .

```
long1$senti <- rep(sentvals,48)
vars7$sentvals <- sentvals
cor(vars7)
```

```
##                   other         self          md            1           2
## other        1.00000000  0.869482188  0.91292805 -0.083124803  0.01690759
## self         0.86948219  1.000000000  0.99503960 -0.051501416  0.03651611
## md           0.91292805  0.995039600  1.00000000 -0.059458318  0.03307401
## 1           -0.08312480 -0.051501416 -0.05945832  1.000000000  0.43611294
## 2            0.01690759  0.036516111  0.03307401  0.436112937  1.00000000
## 3            0.23325456  0.236742608  0.24198258 -0.230336000  0.18817318
## (Intercept) -0.07521948 -0.082516920 -0.08322000  0.366779145 -0.04554083
## oright       0.02396440  0.008399771  0.01224482 -0.064950797  0.19608626
## sentvals     0.04252122  0.015504394  0.01917844 -0.003891647  0.03383895
##                       3 (Intercept)       oright     sentvals
## other        0.23325456 -0.07521948  0.023964396  0.042521223
## self         0.23674261 -0.08251692  0.008399771  0.015504394
## md           0.24198258 -0.08322000  0.012244822  0.019178439
## 1           -0.23033600  0.36677914 -0.064950797 -0.003891647
## 2            0.18817318 -0.04554083  0.196086262  0.033838950
## 3            1.00000000 -0.48259814  0.392132436  0.061992719
## (Intercept) -0.48259814  1.00000000 -0.714063199 -0.102168259
## oright       0.39213244 -0.71406320  1.000000000  0.075072497
## sentvals     0.06199272 -0.10216826  0.075072497  1.000000000
```

```
m0 <- glmer(partRight ~ 1 + (1|wordno) + (1|person),family=binomial,data=long1)
m1 <- update(m0, .~. + oright)
```

```
m2 <- update(m1, .~. + senti)
m3 <- update(m2, .~. + senti:oright)
anova(m0,m1,m2,m3)

## Data: long1
## Models:
## m0: partRight ~ 1 + (1 | wordno) + (1 | person)
## m1: partRight ~ (1 | wordno) + (1 | person) + oright
## m2: partRight ~ (1 | wordno) + (1 | person) + oright + senti
## m3: partRight ~ (1 | wordno) + (1 | person) + oright + senti + oright:senti
##    npar   AIC   BIC  logLik deviance   Chisq Df Pr(>Chisq)
## m0    3 18985 19008 -9489.5    18979
## m1    4 18959 18990 -9475.3    18951 28.2591  1  1.061e-07 ***
## m2    5 18957 18996 -9473.6    18947  3.5380  1    0.05998 .
## m3    6 18953 19000 -9470.5    18941  6.2297  1    0.01256 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m2)

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: partRight ~ (1 | wordno) + (1 | person) + oright + senti
##    Data: long1
##
##      AIC      BIC   logLik deviance df.resid
##  18957.1  18996.2  -9473.6  18947.1    18139
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.1349 -0.7351  0.3998  0.5897  1.8490
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  person (Intercept) 0.5959   0.7719
##  wordno (Intercept) 0.2122   0.4607
## Number of obs: 18144, groups:  person, 378; wordno, 48
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.8991     0.1121   8.023 1.03e-15 ***
## oright        0.8563     0.1382   6.196 5.80e-10 ***
## senti        -0.7395     0.3919  -1.887   0.0592 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr) oright
## oright -0.607
## senti  -0.348 -0.001

summary(m3)

## Generalized linear mixed model fit by maximum likelihood (Laplace
```

```
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## partRight ~ (1 | wordno) + (1 | person) + oright + senti + oright:senti
##    Data: long1
##
##      AIC      BIC   logLik deviance df.resid
##  18952.9  18999.7  -9470.5  18940.9    18138
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.1005 -0.7341  0.3990  0.5886  1.9063
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  person (Intercept) 0.5957   0.7718
##  wordno (Intercept) 0.2124   0.4609
## Number of obs: 18144, groups:  person, 378; wordno, 48
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.9337     0.1130   8.266  < 2e-16 ***
## oright         0.7718     0.1422   5.428 5.71e-08 ***
## senti         -1.0782     0.4144  -2.602  0.00928 **
## oright:senti   0.8292     0.3294   2.518  0.01182 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) oright senti
## oright      -0.615
## senti       -0.367  0.078
## oright:sent  0.123 -0.234 -0.328
```

```
with(long1,tapply(senti,list(partRight,oright),mean))
```

```
##            0          1
## 0 0.10595573 0.09984933
## 1 0.09478711 0.09818539
```

First we examine which variables relate with the size of the memory conformity effect (the final line and final row of the table). It is uncorrelated (all $|r| < .03$) with the factors from the MDS. It also has a low correlation with estimates of how many items the participant thought that they answered correctly. The correlation between this and how many the participant thought the other person accurately answered was $r = .196$. The correlation between how muc

These ANOVAs will be done in a loop for all seven of the variables. Three models will be run for each. First the model with no intercept will be run to get the confidence intervals for the individual conditions. Then the ANOVA comparing each to the control, and finally the 2×2 ANOVA without the control group. **car** (Fox & Weisberg, 2011) is loaded for Type II sum of squares, as discussed in their book. The key one is the one for memory conformity.

h they thought they were influenced is $r = .392$. These correlations do not show the causal patterns among these.

The conditional modes for memory accuracy show a similar pattern. The negative correlations between it and MDS factors were between -.109 and -.086. It was correlated with people's estimates for how many they got right ($r = .461$), which shows some metacognitive accuracy. It showed a negative correlation with how much people felt they were influenced.

References

Carol, R. N., Carlucci, M. E., Eaton, A. A., & Wright, D. B. (2013). The power of a co-witness: When more power leads to more conformity. *Applied Cognitive Psychology*, *27*, 344-351. doi: 10.1002/acp.2912

Fox, J., & Weisberg, S. (2011). *An R companion to applied regression* (2nd ed.). Thousand Oaks, CA: Sage.

Jockers, M. L. (2015). **syuzhet**: Extract sentiment and plot arcs from text [Computer software manual]. Retrieved from `https://github.com/mjockers/syuzhet`

Nash, R. A., Saraiva, R. B., & Hope, L. (2023). Who doesn't believe their memories? Development and validation of a new Memory Distrust Scale. *Journal of Applied Research in Memory & Cognition*, *12*, 401-411. doi: 10.1037/mac0000061

Proellochs, N., & Feuerriegel, S. (2021). **SentimentAnalysis**: Dictionary-based sentiment analysis [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=SentimentAnalysis` (R package version 1.3-4)

R Core Team. (2022). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Rinker, T. W. (2019). **sentimentr**: Calculate text polarity sentiment [Computer software manual]. Buffalo, New York. Retrieved from `http://github.com/trinker/sentimentr` (version 2.7.1)

Silge, J., & Robinson, D. (2016). **tidytext**: Text mining and analysis using tidy data principles in R. *Journal of Open Source Software*, *1*(3). Retrieved from `http://dx.doi.org/10.21105/joss.00037` doi: 10.21105/joss.00037

Skagerberg, E. M., & Wright, D. B. (2008). Manipulating power can affect memory conformity. *Applied Cognitive Psychology*, *22*, 207–216. doi: 10.1002/acp.1353

Wright, D. B., & Celic, V. (2024). How is the memory conformity effect influenced by the relative power of the individuals involved? *Applied Cognitive Psychology*, *38*, e4164. doi: 10.1002/acp.4164