practical_exercise_5, Methods 3, 2021, autumn semester

Anja, Astrid, Jessica, Juli & Magnus

27 okt 2021

Exercises and objectives

The objectives of the exercises of this assignment are based on: https://doi.org/10.1016/j.concog.2019.03.007

4) Download and organise the data from experiment 1

5) Use log-likelihood ratio tests to evaluate logistic regression models

6) Test linear hypotheses

7) Estimate psychometric functions for the Perceptual Awareness Scale and evaluate them

REMEMBER: In your report, make sure to include code that can reproduce the answers requested in the exercises below (**MAKE A KNITTED VERSION**)
REMEMBER: This is part 2 of Assignment 2 and will be part of your final portfolio

EXERCISE 4

Download and organise the data from experiment 1

Go to https://osf.io/ecxsj/files/ and download the files associated with Experiment 1 (there should be 29). The data is associated with Experiment 1 of the article at the following DOI https://doi.org/10.1016/j.concog.2019.03.007

4.1)

Put the data from all subjects into a single data frame - note that some of the subjects do not have the *seed* variable. For these subjects, add this variable and make in *NA* for all observations. (The *seed* variable will not be part of the analysis and is not an experimental variable)

```
#@ output
df <- read_bulk('experiment_1/')
```

4.1.i.

Factorise the variables that need factorising

```
glimpse(df)
```

```
## Rows: 25,602
## Columns: 18
## $ trial.type      <chr> "practice", "practice", "practice", "practice", "pr...
## $ pas             <int> 4, 4, 3, 2, 3, 2, 1, 1, 1, 4, 3, 4, 3, 3, 3, 1, 1, ...
## $ trial           <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...
```

```
## $ jitter.x        <dbl> -0.43065222, -0.29154764, 0.30826178, 0.23776194, -...
## $ jitter.y        <dbl> -0.33548153, -0.18243930, 0.40576314, 0.17474446, -...
## $ odd.digit       <int> 9, 5, 5, 9, 3, 9, 3, 5, 7, 9, 7, 9, 5, 3, 7, 7, 3, ...
## $ target.contrast <dbl> 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0...
## $ target.frames   <int> 9, 8, 7, 6, 5, 4, 3, 2, 1, 9, 8, 7, 6, 5, 4, 3, 2, ...
## $ cue             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ task            <chr> "quadruplet", "quadruplet", "quadruplet", "quadrupl...
## $ target.type     <chr> "even", "odd", "even", "odd", "even", "odd", "even"...
## $ rt.subj         <dbl> 4.9348001, 8.6709449, 8.4068649, 2.1268780, 1.82294...
## $ rt.obj          <dbl> 1.4180470, 0.8598070, 0.7461591, 1.6758869, 0.84708...
## $ even.digit      <int> 8, 4, 2, 8, 6, 6, 6, 4, 4, 8, 2, 2, 4, 2, 4, 6, 2, ...
## $ seed            <int> 93764, 93764, 93764, 93764, 93764, 93764, 93764, 93...
## $ obj.resp        <chr> "e", "o", "e", "o", "e", "o", "o", "e", "e", "o", "...
## $ subject         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ File            <chr> "001.csv", "001.csv", "001.csv", "001.csv", "001.cs...
```

```r
df$trial.type <- as.factor(df$trial.type)
  # The staircase trial types were only introduced in experiment 2 and is an
  # adaptive procedure that allows to collect more data at the threshold
  # visibility. It is not explained what a staircase procedure actually is, but it
  # is used at the beginning of the study before collecting the actual trials.The
  # other trials are experiment trials.

df$pas <- as.factor(df$pas)
  # The Perceptual Awareness Scale, ranging from 1-4.
  # No Experience (NE) -> 1
  # Weak Glimpse (WG) -> 2
  # Almost Clear Experience (ACE) -> 3
  # Clear Experience (CE) -> 4

# trial
  # The text said that each participant performed 864 experiment trials and 18
  # practice trials beforehand. In the data, it seems that 431   trials are recorded
  # for each participant and a varying number of staircase trials

# jitter.x
# jitter.y
# odd.digit

# target.contrast
  # The opacity of the target relative to the background, it was adjusted to
  # match the threshold of each participant

# target.frames
  # the length of time (in frames) the target was shown to participants

df$cue <- as.factor(df$cue)
  # A cue of the possible number of digits presented for each trial, i.e. a kind
  # of framing. Repeated 12 times before a new cue was used

df$task <- as.factor(df$task)
  # The 3 types of tasks:
  # Singles
  # Pairs
```

```
  # Quadruplets

df$target.type <- as.factor(df$target.type)
  # Whether the target was odd or even

# rt.subj
  # Reaction time of the subjective response

# even.digit
# seed

df$rt.obj <- as.numeric(df$rt.obj)
  # Reaction time of the objective response

df$obj.resp <- as.factor(df$obj.resp)
  # The subjects answer to whether the number was odd or even

df$subject <- as.factor(df$subject)
  # ParticipantID

# File
```

4.1.ii.

Remove the practice trials from the dataset (see the *trial.type* variable)

```
df <- df %>%
  filter(trial.type != "practice")
```

4.1.iii.

Create a *correct* variable

```
df$correct <- ifelse(df$obj.resp == "o" & df$target.type == "odd" |
                       df$obj.resp == "e" & df$target.type == "even", 1, 0)

class(df$correct)
```

```
## [1] "numeric"
```

```
df$correct <- as.logical(df$correct)
```

4.1.iv.

Describe how the *target.contrast* and *target.frames* variables differ compared to the data from part 1 of this assignment

> Target contrast is the same value for all - 0.1. In experiment 2 it varied between 1 and 0.01000000. Explains how opaque the target number was. Target frames varies between 1 and 6, in experiment 2 they were all 3. Explains for how many frames (11.8 ms per frame) the target was shown

EXERCISE 5

Use log-likelihood ratio tests to evaluate logistic regression models

5.1)

Do logistic regression - *correct* as the dependent variable and *target.frames* as the independent variable. (Make sure that you understand what *target.frames* encode). Create two models - a pooled model and a partial-pooling model. The partial-pooling model should include a subject-specific intercept.

```
m.5.1.pool <- glm(correct ~ target.frames, df, family = "binomial")
m.5.1.part <- glmer(correct ~ target.frames + (1 | subject), df, family = "binomial") #@ do we need to
```

5.1.i.

The likelihood-function for logistic regression is: $L(p) = \prod_{i=1}^{N} p^{y_i}(1-p)^{(1-y_i)}$ (Remember the probability mass function for the Bernoulli Distribution). Create a function that calculates the likelihood.

```
lik.fun <- function(model, y){
  p <- fitted(model) # estimated likelihood
  y <- y #@ what is this?

  return(prod(p^(y)*(1-p)^(1-y)))
}
```

5.1.ii.

The log-likelihood-function for logistic regression is: $l(p) = \sum_{i=1}^{N} [y_i \ln p + (1-y_i) \ln(1-p)$. Create a function that calculates the log-likelihood

```
loglik.fun <- function(model, y){
  p <- fitted(model) # estimated likelihood
  y <- y #@ what is this?

  return(sum(y*log(p)+(1-y)*log(1-p)))
}
```

5.1.iii.

Apply both functions to the pooling model you just created. Make sure that the log-likelihood matches what is returned from the *logLik* function for the pooled model. Does the likelihood-function return a value that is surprising? Why is the log-likelihood preferable when working with computers with limited precision?

```
lik.fun(m.5.1.pool, df$correct)
```

```
## [1] 0
```

```
loglik.fun(m.5.1.pool, df$correct)
```

```
## [1] -10865.25
```

```
logLik(m.5.1.pool)
```

```
## 'log Lik.' -10865.25 (df=2)
```

When checking our function with the logLik, the same numbers are returned. The difference is that the R-log-likelihood-function returns degrees of freedom which our model does not. The likelihood-function returns a value of 0 which is surprising. There are no decimals — so the number is probably just extremely small and not entirely 0. **The log-likelihood is better because**

some probability distributions are only logarithmically concave. And furthermore, this computer is not precise enough to show the exact value.

5.1.iv.

Now show that the log-likelihood is a little off when applied to the partial pooling model - (the likelihood function is different for the multilevel function - see section 2.1 of https://www.researchgate.net/profile/Douglas-Bates/publication/2753537_Computational_Methods_for_Multilevel_Modelling/links/00b4953b4108d73427000000/Computational-Methods-for-Multilevel-Modelling.pdf if you are interested)

5.2)

Use log-likelihood ratio tests to argue for the addition of predictor variables, start from the null model, `glm(correct ~ 1, 'binomial', data)`, then add subject-level intercepts, then add a group-level effect of *target.frames* and finally add subject-level slopes for *target.frames*. Also assess whether or not a correlation between the subject-level slopes and the subject-level intercepts should be included.

```r
# start from the null model
nullmodel <- glm(correct ~ 1, family = "binomial", df)

# add subject-level intercepts
m.subj.int <- glmer(correct ~ 1 + (1 | subject), family = "binomial", df)

# add a group-level effect of _target.frames_
m.gr.ef <- glmer(correct ~ target.frames + (1 | subject), family = "binomial", df)

# add subject-level slopes for _target.frames_
# m.subj.sl <- glmer(correct ~ target.frames + (subject | target.frames), family = "binomial", df)

 #  We could exchange the 1 with target.frames, but he writes it as "add" to the
 #  nullmodel, so we are unsure.
 #  What does "group-level effect" mean? Is it an intercept for of target.frames?
 # "Also assess whether or not a correlation between the subject-level slopes and
 #  the subject-level intercepts should be included." Are you asking us to consider
 #  whether an interaction effect between two predictors would be approporiate for
 #  this model? Or are you asking us to check whether the values for the intercept
 #  and slope for the subject are correlated?

# interactions::interact_plot(model = m.subj.int, pred = "target.frames", modx = "pas") # visualizing t
```

5.2.i.

Write a short methods section and a results section where you indicate which model you chose and the statistics relevant for that choice. Include a plot of the estimated group-level function with `xlim=c(0, 8)` that includes the estimated subject-specific functions.

```r
# We are waiting to write and plot this until we're sure about the models
```

5.2.ii.

Also include in the results section whether the fit didn't look good for any of the subjects. If so, identify those subjects in the report, and judge (no statistical test) whether their performance (accuracy) differed from that of the other subjects. Was their performance better than chance? (Use a statistical test this time) (50 %)

5.3)

Now add *pas* to the group-level effects - if a log-likelihood ratio test justifies this, also add the interaction between *pas* and *target.frames* and check whether a log-likelihood ratio test justifies this

5.3.i.

If your model doesn't converge, try a different optimizer

5.3.ii.

Plot the estimated group-level functions over `xlim=c(0, 8)` for each of the four PAS-ratings - add this plot to your report (see: 5.2.i) and add a description of your chosen model. Describe how *pas* affects accuracy together with target duration if at all. Also comment on the estimated functions' behaviour at target.frame=0 - is that behaviour reasonable?

EXERCISE 6

Test linear hypotheses

In this section we are going to test different hypotheses. We assume that we have already proved that more objective evidence (longer duration of stimuli) is sufficient to increase accuracy in and of itself and that more subjective evidence (higher PAS ratings) is also sufficient to increase accuracy in and of itself.
We want to test a hypothesis for each of the three neighbouring differences in PAS, i.e. the difference between 2 and 1, the difference between 3 and 2 and the difference between 4 and 3. More specifically, we want to test the hypothesis that accuracy increases faster with objective evidence if subjective evidence is higher at the same time, i.e. we want to test for an interaction.

6.1)

Fit a model based on the following formula: `correct ~ pas * target.frames + (target.frames | subject))`

```
df$target.frames <- as.integer(df$target.frames)

m.6.1 <- glmer(correct ~ pas * target.frames + (target.frames | subject),
               family = "binomial", df)


o <- fitted(m.6.1)
view(o)
```

6.1.i.

First, use `summary` (yes, you are allowed to!) to argue that accuracy increases faster with objective evidence for PAS 2 than for PAS 1.

```
summary(m.6.1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: correct ~ pas * target.frames + (target.frames | subject)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
```

```
##  19506.1  19595.5  -9742.0  19484.1    25033
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -19.0112   0.0537   0.1606   0.4849   1.4465
##
## Random effects:
##  Groups  Name          Variance Std.Dev. Corr
##  subject (Intercept)   0.03697  0.1923
##          target.frames 0.02057  0.1434   -0.76
## Number of obs: 25044, groups:  subject, 29
##
## Fixed effects:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.12163    0.06414  -1.896 0.057929 .
## pas2                -0.57140    0.08944  -6.388 1.68e-10 ***
## pas3                -0.53845    0.13947  -3.861 0.000113 ***
## pas4                 0.20152    0.25107   0.803 0.422197
## target.frames        0.11480    0.03708   3.096 0.001961 **
## pas2:target.frames  0.44719    0.03475  12.869  < 2e-16 ***
## pas3:target.frames  0.74868    0.04593  16.300  < 2e-16 ***
## pas4:target.frames  0.75929    0.06861  11.066  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) pas2   pas3   pas4   trgt.f ps2:t. ps3:t.
## pas2         -0.462
## pas3         -0.307  0.248
## pas4         -0.175  0.122  0.091
## target.frms -0.811  0.306  0.207  0.124
## ps2:trgt.fr  0.482 -0.874 -0.245 -0.125 -0.428
## ps3:trgt.fr  0.393 -0.279 -0.891 -0.111 -0.358  0.370
## ps4:trgt.fr  0.276 -0.163 -0.121 -0.919 -0.260  0.226  0.200
```

```
# For pas 1
```

6.2)

`summary` won't allow you to test whether accuracy increases faster with objective evidence for PAS 3 than for PAS 2 (unless you use `relevel`, which you are not allowed to in this exercise). Instead, we'll be using the function `glht` from the `multcomp` package ### 6.2.i. To redo the test in 6.1.i, you can create a *contrast* vector. This vector will have the length of the number of estimated group-level effects and any specific contrast you can think of can be specified using this. For redoing the test from 6.1.i, the code snippet below will do ### 6.2.ii. Now test the hypothesis that accuracy increases faster with objective evidence for PAS 3 than for PAS 2. ### 6.2.iii. Also test the hypothesis that accuracy increases faster with objective evidence for PAS 4 than for PAS 3 ## 6.3) Finally, test that whether the difference between PAS 2 and 1 (tested in 6.1.i) is greater than the difference between PAS 4 and 3 (tested in 6.2.iii)

Snippet for 6.2.i

```
## testing whether PAS 2 is different from PAS 1
contrast.vector <- matrix(c(0, 0, 0, 0, 0, 1, 0, 0), nrow=1)
gh <- glht(pas.intact.tf.ranslopeint.with.corr, contrast.vector)
```

```
print(summary(gh))
## as another example, we could also test whether there is a difference in
## intercepts between PAS 2 and PAS 3
contrast.vector <- matrix(c(0, -1, 1, 0, 0, 0, 0, 0), nrow=1)
gh <- glht(pas.intact.tf.ranslopeint.with.corr, contrast.vector)
print(summary(gh))
```

EXERCISE 7

Estimate psychometric functions for the Perceptual Awareness Scale and evaluate them

We saw in 5.3 that the estimated functions went below chance at a target duration of 0 frames (0 ms). This does not seem reasonable, so we will be trying a different approach for fitting here.
We will fit the following function that results in a sigmoid, $f(x) = a + \frac{b-a}{1+e^{\frac{c-x}{d}}}$
It has four parameters: a, which can be interpreted as the minimum accuracy level, b, which can be interpreted as the maximum accuracy level, c, which can be interpreted as the so-called inflexion point, i.e. where the derivative of the sigmoid reaches its maximum and d, which can be interpreted as the steepness at the inflexion point. (When d goes towards infinity, the slope goes towards a straight line, and when it goes towards 0, the slope goes towards a step function).

We can define a function of a residual sum of squares as below

```
RSS <- function(dataset, par)
{
    ## "dataset" should be a data.frame containing the variables x (target.frames)
    ## and y (correct)

    ## "par" are our four parameters (a numeric vector)
    ## par[1]=a, par[2]=b, par[3]=c, par[4]=d
    x <- dataset$x
    y <- dataset$y
    y.hat <- ## you fill in the estimate of y.hat
    RSS <- sum((y - y.hat)^2)
    return(RSS)
}
```

7.1)

Now, we will fit the sigmoid for the four PAS ratings for Subject 7 ### 7.1.i. Use the function `optim`. It returns a list that among other things contains the four estimated parameters. You should set the following arguments:
`par`: you can set c and d as 1. Find good choices for a and b yourself (and argue why they are appropriate)
`fn`: which function to minimise?
`data`: the data frame with x, *target.frames*, and y, *correct* in it
`method`: 'L-BFGS-B'
`lower`: lower bounds for the four parameters, (the lowest value they can take), you can set c and d as `-Inf`. Find good choices for a and b yourself (and argue why they are appropriate)
`upper`: upper bounds for the four parameters, (the highest value they can take) can set c and d as `Inf`. Find good choices for a and b yourself (and argue why they are appropriate)
7.1.ii. Plot the fits for the PAS ratings on a single plot (for subject 7) `xlim=c(0, 8)` ### 7.1.iii. Create a similar plot for the PAS ratings on a single plot (for subject 7), but this time based on the model from 6.1 `xlim=c(0, 8)`
7.1.iv. Comment on the differences between the fits - mention some advantages and disadvantages of each way

7.2) Finally, estimate the parameters for all subjects and each of their four PAS ratings. Then plot the estimated function at the group-level by taking the mean for each of the four parameters, a, b, c and d across subjects. A function should be estimated for each PAS-rating (it should look somewhat similar to Fig. 3 from the article: https://doi.org/10.1016/j.concog.2019.03.007) ### 7.2.i. Compare with the figure you made in 5.3.ii and comment on the differences between the fits - mention some advantages and disadvantages of both.