practical_exercise_3, Methods 3, 2021, autumn semester

Aleks

04/10/21

Exercises and objectives

The objectives of the exercises of this assignment are:
1) Download and organise the data and model and plot staircase responses based on fits of logistic functions
2) Fit multilevel models for response times
3) Fit multilevel models for count data

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 assignment will be part of your final portfolio

Exercise 1

Go to https://osf.io/ecxsj/files/ and download the files associated with Experiment 2 (there should be 29). The data is associated with Experiment 2 of the article at the following DOI https://doi.org/10.1016/j.concog.2019.03.007

1) Put the data from all subjects into a single data frame

```
df = read_bulk("experiment_2")
```

```
## Reading 001.csv

## Reading 002.csv

## Reading 003.csv

## Reading 004.csv

## Reading 005.csv

## Reading 006.csv

## Reading 007.csv

## Reading 008.csv
```

```
## Reading 009.csv

## Reading 010.csv

## Reading 011.csv

## Reading 012.csv

## Reading 013.csv

## Reading 014.csv

## Reading 015.csv

## Reading 016.csv

## Reading 017.csv

## Reading 018.csv

## Reading 019.csv

## Reading 020.csv

## Reading 021.csv

## Reading 022.csv

## Reading 023.csv

## Reading 024.csv

## Reading 025.csv

## Reading 026.csv

## Reading 027.csv

## Reading 028.csv

## Reading 029.csv
```

2) Describe the data and construct extra variables from the existing variables

ALEKS:

Before expalining variables, it's important to contextualize them in the light of the study. The primary goal of this study was to assess whether the distinctness of one's expectations towards prospective stimuli influences how clearly they are experienced. The task used to investigate this was based on digit recognition, where participants are (very very briefly) shown a number and have to report whether it was even or odd.

trial.type: Before the actual experiment a staircase procedure was performed to strengthen the statistical power for detecting a positive relationship. This variable indicates whether the data collected for a participant was collected during the experiment, or during the staircase procedure.

pas: The *Perceptual Awareness Scale* is used to assess the "clarity" of the experience. It is ranked from 1 (No Experience) to 4(Clear Experience).

trial: Stimuli number as it was shown to the participant. For *staircase* range from 1 to approx 80 (80 stimuli shown during the staircase procedure) and during the actual *experiment* goes from 1 to approx 420 (420 total stimuli shown to the participant during the experiment stage).

jitter X and Y: Coordinates for a slight jitter that was applied to the position of the target digit, randomly drawn from a uniform distribution that fell within $\pm$ 0.5° of the fixation point.

odd digit:

target.contrast: The contrast of the target stimulus, relative to the background, was adjusted to match the threshold of each individual participant.

target.frames: A number of frames the stimuli(digit) was shown for. 1 frame is 11.8ms

cue: A cue indicating the set of digits from which the target digit would be randomly drawn from on that trial. Could be either 2(2:9), 4(24:57) or 8(2468:3579).

task: Actually I don't understand what this is.

target.type: Whether the digit presented was Even or Odd.

rt.subj: Subjective reaction time as reported by participants?

rt.obj: Objective reaction time(as measured by the script).

even digit:

seed: Seed used for generating random numbers.

obj.resp: The answer of the participant - whether he/she thought an even number was presented or odd number.

```
i. add a variable to the data frame and call it _correct_ (have it be a _logical_ variable). Assign a 1
```

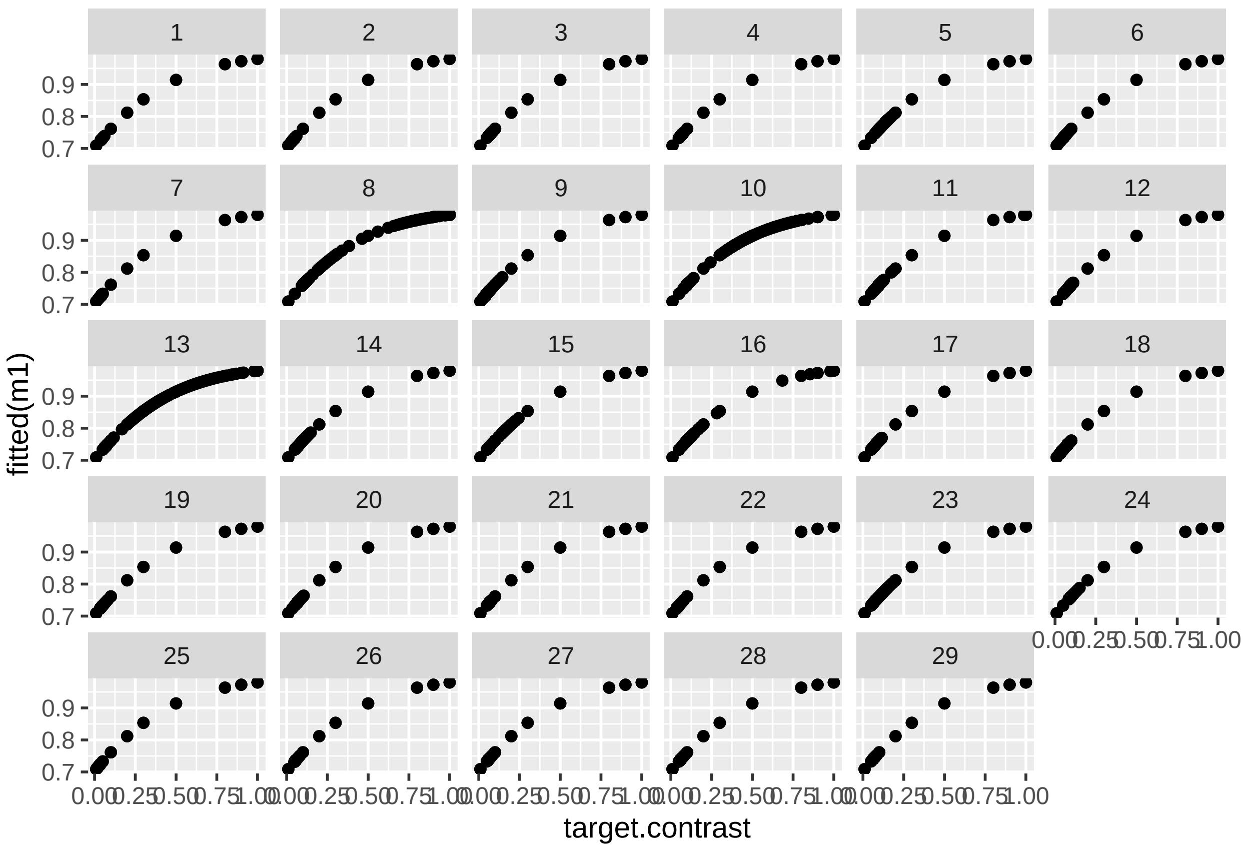
```
df = df %>%
 mutate(correct = ifelse(target.type == "even" & obj.resp == "e" |
                         target.type == "odd" & obj.resp == "o", 1, 0))
```

```
ii. describe what the following variables in the data frame contain, _trial.type_, _pas_, _trial_, _targ
```

```
#Finish this.
```

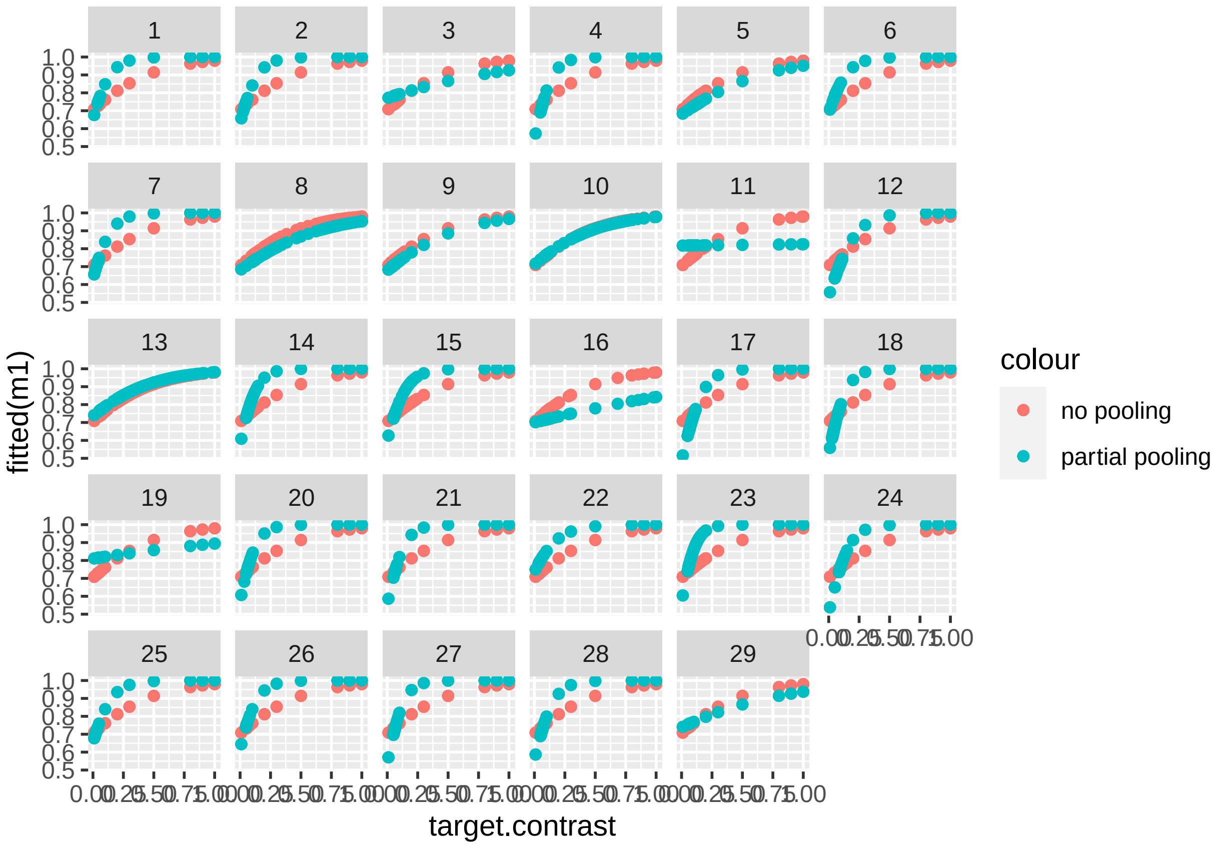
```
iii. for the staircasing part __only__, create a plot for each subject where you plot the estimated func
```

```r
df_staircase = df %>%
  filter(trial.type == "staircase")

m1 = glm(correct ~ target.contrast, df_staircase, family = "binomial")

ggplot(data = df_staircase, aes(x = target.contrast, y = fitted(m1))) +
  geom_point() +
  facet_wrap( ~ subject)
```



iv. on top of those plots, add the estimated functions (on the _target.contrast_ range from 0-1) for ea

```r
m2 = glmer(correct ~ target.contrast + (1+target.contrast|subject), df_staircase, family = "binomial")

ggplot(data = df_staircase) +
  geom_point(aes(x = target.contrast, y = fitted(m1), color = "no pooling")) +
  geom_point(aes(x = target.contrast, y = fitted(m2), color = "partial pooling")) +
  facet_wrap( ~ subject)
```




```
v. in your own words, describe how the partial pooling model allows for a better fit for each subject
```

Exercise 2

Now we **only** look at the *experiment* trials (*trial.type*)

```
df_experiment = df %>%
  filter(trial.type == "experiment")
```

1) Pick four subjects and plot their Quantile-Quantile (Q-Q) plots for the residuals of their objective response times (*rt.obj*) based on a model where only intercept is modelled

```
df_subject1 = df_experiment[which(df$subject == "1"),]
df_subject2 = df_experiment[which(df$subject == "6"),]
df_subject3 = df_experiment[which(df$subject == "11"),]
df_subject4 = df_experiment[which(df$subject == "19"),]

m3 = lm(rt.obj ~ 1, df_subject1)
m4 = lm(rt.obj ~ 1, df_subject2)
m5 = lm(rt.obj ~ 1, df_subject3)
m6 = lm(rt.obj ~ 1, df_subject4)

par(mfrow=c(2,2))
qqPlot(resid(m3))
```

```
## [1] 199  97
```

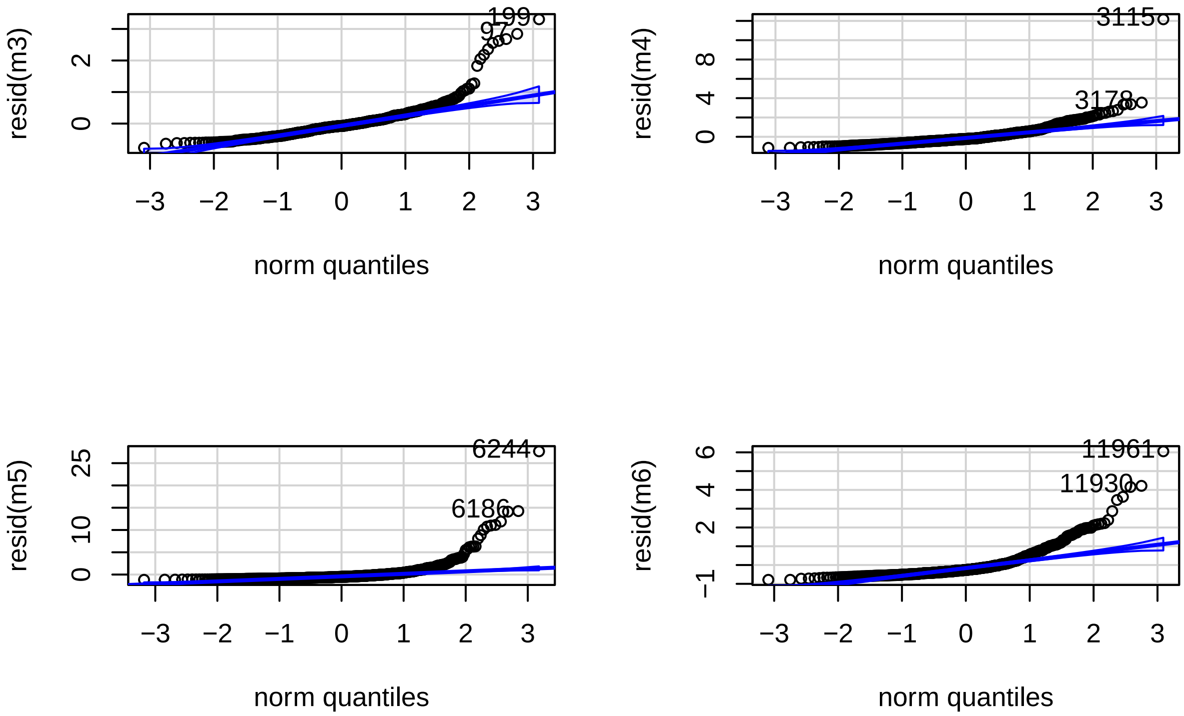
```
qqPlot(resid(m4))
```

```
## 3115 3178
##  119  182
```

```
qqPlot(resid(m5))
```

```
## 6244 6186
##  137   79
```

```
qqPlot(resid(m6))
```




```
## 11961 11930
##   458   427
```

```
i. comment on these
```

ALEKS: All of them look pretty right skewed, which is not optimal.

```
ii. does a log-transformation of the response time data improve the Q-Q-plots?
```

```
m3_log = lm(log(rt.obj) ~ 1, data = df_subject1)
m4_log = lm(log(rt.obj) ~ 1, data = df_subject2)
m5_log = lm(log(rt.obj) ~ 1, data = df_subject3)
m6_log = lm(log(rt.obj) ~ 1, data = df_subject4)
```

```r
par(mfrow=c(2,2))

qqPlot(resid(m3_log))
```

```
## [1] 347 199
```

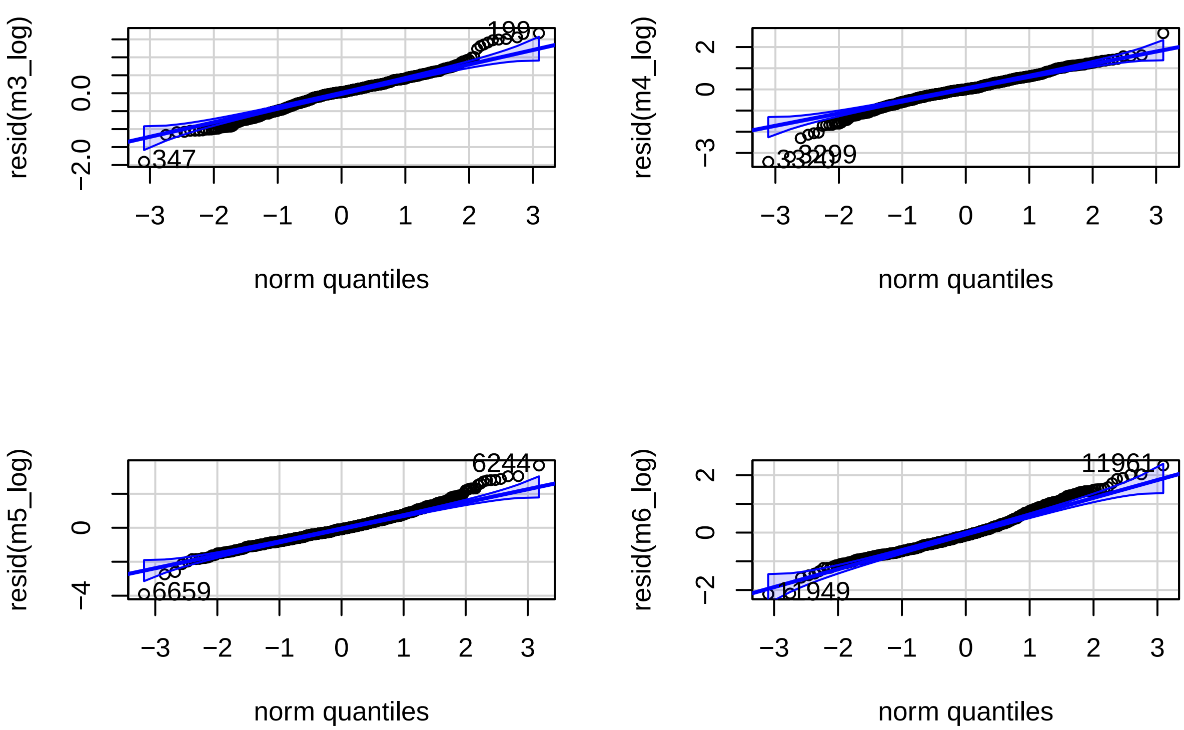
```r
qqPlot(resid(m4_log))
```

```
## 3320 3299
##  324  303
```

```r
qqPlot(resid(m5_log))
```

```
## 6659 6244
##  552  137
```

```r
qqPlot(resid(m6_log))
```




```
## 11961 11949
##   458   446
```

ALEKS: Yes. Whilst some light skeweness remains, log transformation does helo.

2) Now do a partial pooling model modelling objective response times as dependent on *task*? (set `REML=FALSE` in your `lmer`-specification)

```r
m7 = lmer(rt.obj ~ task + (1|subject), df_experiment, REML = FALSE)
summary(m7)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: rt.obj ~ task + (1 | subject)
##    Data: df_experiment
## 
##      AIC      BIC   logLik deviance df.resid 
##  61940.2  61977.4 -30965.1  61930.2    12523 
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
##  -0.630  -0.155  -0.072   0.051 101.443 
## 
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 0.1147   0.3386  
##  Residual             8.1739   2.8590  
## Number of obs: 12528, groups:  subject, 29
## 
## Fixed effects:
##                   Estimate Std. Error         df t value Pr(>|t|)    
## (Intercept)      1.120e+00  7.689e-02  4.775e+01  14.568  < 2e-16 ***
## taskquadruplet -1.532e-01  6.257e-02  1.250e+04  -2.449  0.01433 *  
## tasksingles    -1.915e-01  6.257e-02  1.250e+04  -3.061  0.00221 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##             (Intr) tskqdr
## taskqudrplt -0.407       
## tasksingles -0.407  0.500
```

```
i. which would you include among your random effects and why? (support your choices with relevant measu
```

```r
#Finish this. Play around with different random effects and look at variance.
```

```
ii. explain in your own words what your chosen models says about response times between the different ta
```

ALEKS: The current chosen model(m7) suggests that the people generally react a bit faster on single(0.19ms) and quadruplet(0.15ms) tasks than on the paired task.

```r
#Not sure about the above, though, double check again when a bit less sleep deprived.
```

3) Now add *pas* and its interaction with *task* to the fixed effects
   i. how many types of group intercepts (random effects) can you add without ending up with convergence issues or singular fits?

```r
m8 = lmer(rt.obj ~ task * pas + (1|subject), df_experiment, REML = FALSE)
#all ok

m9 = lmer(rt.obj ~ task * pas + (1|subject) + (1|task) + (1|pas), df_experiment, REML = FALSE)
```

```
## boundary (singular) fit: see ?isSingular
```

```r
#singular fit

m10 = lmer(rt.obj ~ task * pas + (1|subject) + (1|task), df_experiment, REML = FALSE)
```

```
## boundary (singular) fit: see ?isSingular
```

```r
#singular fit

m11 = lmer(rt.obj ~ task * pas + (1|subject) + (1|pas), df_experiment, REML = FALSE)
#all ok

#Models that converged.
summary(m8)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: rt.obj ~ task * pas + (1 | subject)
##    Data: df_experiment
##
##      AIC      BIC   logLik deviance df.resid
##  61911.5  61970.9 -30947.7  61895.5    12520
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
##  -0.668  -0.152  -0.064   0.048 101.530
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 0.09744  0.3122
##  Residual             8.15379  2.8555
## Number of obs: 12528, groups:  subject, 29
##
## Fixed effects:
##                       Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)          1.541e+00  1.197e-01  3.160e+02  12.872  < 2e-16 ***
## taskquadruplet      -3.639e-01  1.397e-01  1.251e+04  -2.605  0.00919 **
## tasksingles         -1.790e-01  1.448e-01  1.252e+04  -1.236  0.21648
## pas                 -1.876e-01  4.234e-02  8.855e+03  -4.431 9.51e-06 ***
## taskquadruplet:pas   9.140e-02  5.646e-02  1.251e+04   1.619  0.10547
## tasksingles:pas      1.312e-02  5.529e-02  1.252e+04   0.237  0.81248
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tskqdr tsksng pas    tskqd:
```

```
## taskqudrplt -0.592
## tasksingles -0.563  0.489
## pas         -0.793  0.599  0.567
## tskqdrplt:p  0.522 -0.894 -0.433 -0.658
## tsksngls:ps  0.534 -0.459 -0.901 -0.673  0.506
```

```
summary(m11)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: rt.obj ~ task * pas + (1 | subject) + (1 | pas)
##    Data: df_experiment
##
##      AIC      BIC   logLik deviance df.resid
##  61912.9  61979.9 -30947.5  61894.9    12519
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
##  -0.664  -0.151  -0.064   0.046 101.543
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 0.097953 0.31297
##  pas      (Intercept) 0.001805 0.04248
##  Residual             8.152408 2.85524
## Number of obs: 12528, groups:  subject, 29; pas, 4
##
## Fixed effects:
##                       Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)          1.550e+00  1.308e-01  1.981e+01  11.854  1.9e-10 ***
## taskquadruplet      -3.620e-01  1.397e-01  1.251e+04  -2.592 0.009552 **
## tasksingles         -1.866e-01  1.449e-01  1.232e+04  -1.288 0.197857
## pas                 -1.915e-01  4.650e-02  1.875e+01  -4.119 0.000598 ***
## taskquadruplet:pas  9.058e-02  5.645e-02  1.251e+04   1.604 0.108639
## tasksingles:pas     1.669e-02  5.535e-02  1.195e+04   0.302 0.762978
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tskqdr tsksng pas    tskqd:
## taskqudrplt -0.541
## tasksingles -0.517  0.488
## pas         -0.813  0.544  0.519
## tskqdrplt:p  0.477 -0.894 -0.433 -0.599
## tsksngls:ps  0.490 -0.458 -0.901 -0.615  0.505
```

```
ii. create a model by adding random intercepts (without modelling slopes) that results in a singular fi
```

```
#Do this.
```

```
iii. in your own words - how could you explain why your model would result in a singular fit?
```

```r
#Do this.
```

Exercise 3

1) Initialise a new data frame, `data.count`. *count* should indicate the number of times they categorized their experience as *pas* 1-4 for each *task*. I.e. the data frame would have for subject 1: for task:singles, pas1 was used # times, pas2 was used # times, pas3 was used # times and pas4 was used # times. You would then do the same for task:pairs and task:quadruplet

```r
data.count = df %>%
  group_by(subject, task, pas) %>%
  summarise("count" = n())
```

```
## 'summarise()' has grouped output by 'subject', 'task'. You can override using the '.groups' argument
```

2) Now fit a multilevel model that models a unique "slope" for *pas* for each *subject* with the interaction between *pas* and *task* and their main effects being modelled

```r
m12 = glmer(count ~ pas * task + (pas|subject), data.count, family = poisson)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00689826 (tol = 0.002, component 1)
```

```r
summary(m12)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: count ~ pas * task + (pas | subject)
##    Data: data.count
##
##      AIC      BIC   logLik deviance df.resid
##   4685.1   4719.6  -2333.6   4667.1      331
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.7718 -1.9208 -0.1275  1.6133 11.6478
##
## Random effects:
##  Groups  Name        Variance Std.Dev. Corr
##  subject (Intercept) 1.2016   1.0962
##          pas         0.2203   0.4694   -0.99
## Number of obs: 340, groups:  subject, 29
##
## Fixed effects:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          4.29023    0.20576  20.851  < 2e-16 ***
## pas                 -0.19516    0.08798  -2.218   0.0265 *
## taskquadruplet       0.16667    0.04007   4.160 3.19e-05 ***
## tasksingles         -0.39664    0.04192  -9.462  < 2e-16 ***
```

```
## pas:taskquadruplet -0.07193    0.01606  -4.479 7.50e-06 ***
## pas:tasksingles     0.16857    0.01587  10.623  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) pas    tskqdr tsksng ps:tskq
## pas         -0.989
## taskqudrplt -0.100  0.083
## tasksingles -0.096  0.078  0.490
## ps:tskqdrpl  0.088 -0.091 -0.891 -0.430
## ps:tsksngls  0.089 -0.091 -0.456 -0.900  0.501
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00689826 (tol = 0.002, component 1)
```

i. which family should be used?

```
#I think Poisson. Get back to this.
```

ii. why is a slope for _pas_ not really being modelled?
iii. if you get a convergence error, try another algorithm (the default is the _Nelder_Mead_) - try (_bo

```
m13 = glmer(count ~ pas * task + (pas|subject), data.count, family = poisson, control = glmerControl(opt
#alright thats cool
```

iv. when you have a converging fit - fit a model with only the main effects of _pas_ and _task_. Compare

```
m14 = glmer(count ~ pas + task + (pas|subject), data.count, family = poisson, control = glmerControl(opt
tibble(sum(residuals(m13)**2), sum(residuals(m14)**2))
```

```
## # A tibble: 1 x 2
##   `sum(residuals(m13)^2)` `sum(residuals(m14)^2)`
##                     <dbl>                   <dbl>
## 1                   2508.                   2749.
```

```
AIC(m13, m14)
```

```
##     df      AIC
## m13  9 4685.119
## m14  7 4923.190
```

v. indicate which of the two models, you would choose and why

```
#Finish this.
```

vi. based on your chosen model - write a short report on what this says about the distribution of rating

```
#And this.
```

vii. include a plot that shows the estimated amount of ratings for four subjects of your choosing

```r
#And this. I guess I can just facet-wrap all of subjects as done previously?
```

3) Finally, fit a multilevel model that models *correct* as dependent on *task* with a unique intercept for each *subject*

```r
m15 <- glmer(correct ~ task + (1|subject), df, family = binomial)
summary(m15)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: correct ~ task + (1 | subject)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
##  19927.2  19958.4  -9959.6  19919.2    18127
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7426 -1.0976  0.5098  0.6101  0.9111
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  subject (Intercept) 0.1775   0.4214
## Number of obs: 18131, groups:  subject, 29
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.10071    0.08386  13.125  < 2e-16 ***
## taskquadruplet -0.09825    0.04189  -2.345    0.019 *
## tasksingles      0.18542    0.04336   4.276  1.9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tskqdr
## taskqudrplt -0.256
## tasksingles -0.247  0.495
```

```
i. does _task_ explain performance?
```

ALEKS: Yes. Singles are a bit easier than pairs, and quadruplets are a bit harder than pairs.

```
ii. add _pas_ as a main effect on top of _task_ - what are the consequences of that?
```

```r
m16 <- glmer(correct ~ task + pas + (1|subject), df, family = binomial)
summary(m16)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
```

```
## Formula: correct ~ task + pas + (1 | subject)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
##  17425.0  17464.0  -8707.5  17415.0    18126
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -8.1096 -0.6101  0.3181  0.5653  1.6476
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  subject (Intercept) 0.2004   0.4477
## Number of obs: 18131, groups:  subject, 29
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.950104   0.098389  -9.657   <2e-16 ***
## taskquadruplet -0.029418   0.045017  -0.653    0.513
## tasksingles    -0.008914   0.046887  -0.190    0.849
## pas             1.014031   0.022900  44.282   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tskqdr tsksng
## taskqudrplt -0.247
## tasksingles -0.190  0.489
## pas         -0.421  0.030 -0.083
```

ALEKS: Pretty interesting. I'll revisit this at a later point but it seems that the inclusion of PAS scores makes the influence of task on performance insignificant.

```
iii. now fit a multilevel model that models _correct_ as dependent on _pas_ with a unique intercept for
```

```
m17 <- glmer(correct ~ pas + (1|subject), df, family = binomial)
summary(m17)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: correct ~ pas + (1 | subject)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
##  17421.5  17444.9  -8707.7  17415.5    18128
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -8.1864 -0.6117  0.3187  0.5664  1.6348
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
```

```
##  subject (Intercept) 0.2005   0.4478
## Number of obs: 18131, groups:  subject, 29
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.96488    0.09503  -10.15   <2e-16 ***
## pas          1.01488    0.02275   44.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr)
## pas -0.440
```

iv. finally, fit a model that models the interaction between _task_ and _pas_ and their main effects

```
m18 <- glmer(correct ~ task * pas + (1|subject), df, family = binomial)
summary(m18)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: correct ~ task * pas + (1 | subject)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
##  17422.8  17477.5  -8704.4  17408.8    18124
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -7.6024 -0.6183  0.3163  0.5755  1.6810
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  subject (Intercept) 0.2007   0.448
## Number of obs: 18131, groups:  subject, 29
##
## Fixed effects:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.93319    0.11175  -8.350   <2e-16 ***
## taskquadruplet      0.05257    0.09954   0.528    0.597
## tasksingles        -0.15465    0.10523  -1.470    0.142
## pas                 1.00447    0.03752  26.769   <2e-16 ***
## taskquadruplet:pas -0.04783    0.05057  -0.946    0.344
## tasksingles:pas     0.07731    0.05142   1.504    0.133
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tskqdr tsksng pas    tskqd:
## taskqudrplt -0.468
## tasksingles -0.436  0.494
## pas         -0.601  0.628  0.584
```

```
## tskqdrplt:p  0.414 -0.893 -0.438 -0.694
## tsksngls:ps  0.406 -0.456 -0.894 -0.681  0.505
```

v. describe in your words which model is the best in explaining the variance in accuracy