Portfolio Assignment 2 part 1, Methods 3, 2021, autumn semester

Study Group 8, Luke Ring (LR), Emma Rose Hahn (EH), Viara Krasteva (VK), Kristian Nøhr Villebro (KV)

Student numbers

Emma Rose Hahn: 202004249 Viara Krasteva: 202005673 Kristian Nøhr Villebro: 202005289 Luke Ring: 202009983

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/ (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 (https://doi.org/10.1016/j.concog.2019.03.007)

1. Put the data from all subjects into a single data frame
2. Describe the data and construct extra variables from the existing variables
    i. add a variable to the data frame and call it *correct* (have it be a *logical* variable). Assign a 1 to each row where the subject indicated the correct answer and a 0 to each row where the subject indicated the incorrect answer (**Hint:** the variable *obj.resp* indicates whether the subject answered "even", *e* or "odd", *o*, and the variable *target_type* indicates what was actually presented.
    ii. describe what the following variables in the data frame contain, *trial.type*, *pas*, *trial*, *target.contrast*, *cue*, *task*, *target_type*, *rt.subj*, *rt.obj*, *obj.resp*,

subject and *correct*. (That means you can ignore the rest of the variables in your description). For each of them, indicate and argue for what `class` they should be classified into, e.g. *factor*, *numeric* etc.

iii. for the staircasing part **only**, create a plot for each subject where you plot the estimated function (on the *target.contrast* range from 0-1) based on the fitted values of a model (use `glm`) that models *correct* as dependent on *target.contrast*. These plots will be our *no-pooling* model. Comment on the fits - do we have enough data to plot the logistic functions?

iv. on top of those plots, add the estimated functions (on the *target.contrast* range from 0-1) for each subject based on partial pooling model (use `glmer` from the package `lme4`) where unique intercepts and slopes for *target.contrast* are modelled for each *subject*

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

Answers

Exercise 1, part 1 - EH

```r
# get list of all CSV files
temp <- list.files(path = "./experiment_2/",
    pattern = "*.csv", full.names = TRUE)

# load into single data frame
samples <- map_df(temp, read_csv, trim_ws = TRUE, na = c("", "NA"),
    col_types = cols(
        trial.type = col_factor(),
        pas = col_integer(),
        trial = col_factor(),
        jitter.x = col_double(),
        jitter.y = col_double(),
        odd.digit = col_integer(),
        target.contrast = col_double(),
        target.frames = col_double(),
        cue = col_factor(),
        task = col_factor(),
        target.type = col_factor(),
        rt.subj = col_double(),
        rt.obj = col_double(),
        even.digit = col_integer(),
        seed = col_double(),
        obj.resp = col_factor(),
        subject = col_factor()
    ))

rm(temp)
# peek data
head(samples)
```

```
## # A tibble: 6 x 17
##   trial.type   pas trial jitter.x jitter.y odd.digit target.contrast
##   <fct>      <int> <fct>    <dbl>    <dbl>     <int>           <dbl>
## 1 staircase      4 0      -0.343    0.449         9               1
## 2 staircase      4 1       0.0623   0.0291        9               1
## 3 staircase      4 2      -0.406    0.500         7             0.9
## 4 staircase      4 3      -0.362   -0.222         7             0.9
## 5 staircase      4 4       0.289    0.413         7             0.8
## 6 staircase      4 5       0.0824  -0.0934        7             0.8
## # ... with 10 more variables: target.frames <dbl>, cue <fct>, task <fct>,
## #   target.type <fct>, rt.subj <dbl>, rt.obj <dbl>, even.digit <int>,
## #   seed <dbl>, obj.resp <fct>, subject <fct>
```

```
# total number of samples
nrow(samples)
```

```
## [1] 18131
```

Exercise 1, part 2 - KV

```
# Add column to indicate if participant response was correct
samples <- mutate(samples,
    correct = as.logical(
        ifelse(substr(target.type, 1, 1) == obj.resp, 1, 0)
    )
)
```

The dataset contains the following variables:

Variable	Description	Class
trial.type	either staircase(practice) or experiment	factor: categorical, reused
pas	any number from 1-4, indicating the reported experience on the Perceptual Awareness Scale	factor: categorical, not continuous
trial	trial number zero-indexed for the practice and experiment blocks	factor: categorical, not continuous
target.contrast	the grey-scale proportion of the target digit	double: numeric, continous variable from 0-1

| Variable	Description	Class
cue	number code for pre-stimulus cue	factor: categorical, not continuous
task	how many numbers could be shown; has the levels quadruplet: (all numbers); pairs: (2 even and 2 odd numbers); singles: (1 even and 1 odd number)	factor: categorical, no continuous relationship
target_type	whether the target shown was the chosen even.digit or the chosen odd.digit	factor: categorical, no continuous relationship
rt.subj	reaction time (seconds) on the PAS response	double: continous variable with decimal places
rt.obj	reaction time (seconds) on the target digit	double: continous variable with decimal places
obj.resp	the key actually pressed e for even and o for odd	factor: categorical, no continuous relationship
subject	subject number	factor: categorical
correct	Whether answer was correct(1) or incorret (0)	logical, boolean either true or false

```
# only use staircase trial types.
samples_staircase <- samples %>% filter(trial.type == "staircase")
#making a no-pooling model
m1 <- glm(correct ~ target.contrast + subject,
    data = samples_staircase,
    family = binomial(link = "logit"))
# We should see an intercept per subject
summary(m1)
```

```
##
## Call:
## glm(formula = correct ~ target.contrast + subject, family = binomial(link = "logi
t"),
##     data = samples_staircase)
##
## Deviance Residuals:
##     Min      1Q   Median      3Q      Max
## -3.2225   0.1836   0.7262   0.7907   0.9772
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.237950   0.310228   3.990 6.59e-05 ***
## target.contrast 3.446129   0.439005   7.850 4.17e-15 ***
## subject002     -0.200038   0.396544  -0.504   0.6139
## subject003     -0.126797   0.367239  -0.345   0.7299
## subject004     -0.447583   0.341680  -1.310   0.1902
## subject005     -0.729034   0.342889  -2.126   0.0335 *
## subject006      0.204637   0.412265   0.496   0.6196
## subject007     -0.257607   0.397907  -0.647   0.5174
## subject008     -0.728765   0.345702  -2.108   0.0350 *
## subject009     -0.600476   0.339050  -1.771   0.0766 .
## subject010     -0.450820   0.353083  -1.277   0.2017
## subject011     -0.003327   0.353925  -0.009   0.9925
## subject012     -0.751795   0.339166  -2.217   0.0267 *
## subject013     -0.251208   0.354408  -0.709   0.4784
## subject014     -0.279840   0.345979  -0.809   0.4186
## subject015     -0.330359   0.349908  -0.944   0.3451
## subject016     -0.781364   0.340545  -2.294   0.0218 *
## subject017     -0.691345   0.339823  -2.034   0.0419 *
## subject018     -0.546979   0.340267  -1.607   0.1079
## subject019      0.502524   0.488717   1.028   0.3038
## subject020     -0.271168   0.356758  -0.760   0.4472
## subject021     -0.399562   0.346301  -1.154   0.2486
## subject022      0.523698   0.488385   1.072   0.2836
## subject023     -0.065056   0.359321  -0.181   0.8563
## subject024     -0.531046   0.341990  -1.553   0.1205
## subject025     -0.160067   0.426526  -0.375   0.7075
## subject026     -0.222237   0.374445  -0.594   0.5528
## subject027     -0.487602   0.342189  -1.425   0.1542
## subject028     -0.547571   0.339972  -1.611   0.1073
## subject029     -0.335918   0.355318  -0.945   0.3445
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6127.2  on 5602  degrees of freedom
## Residual deviance: 5936.1  on 5573  degrees of freedom
## AIC: 5996.1
##
## Number of Fisher Scoring iterations: 6
```

```
# plot the fitted values per subject
samples_staircase %>%
    ggplot(aes(target.contrast, as.integer(correct), color = correct)) +
    geom_point() +
    geom_line(aes(target.contrast, fitted(m1),
        linetype = "fitted values"), inherit.aes = FALSE) +
    scale_linetype_manual(name = "single-level model", values = c("dashed")) +
    facet_wrap(~subject) +
    labs(title = "Correct answers by target contrast per participant",
        subtitle = "With single-level model fitted values (no-pooling)",
        color = "subject correct") +
    ylab("Correct (probability)") +
    xlab("target contrast")
```


Correct answers by target contrast per participant
With single-level model fitted values (no-pooling)
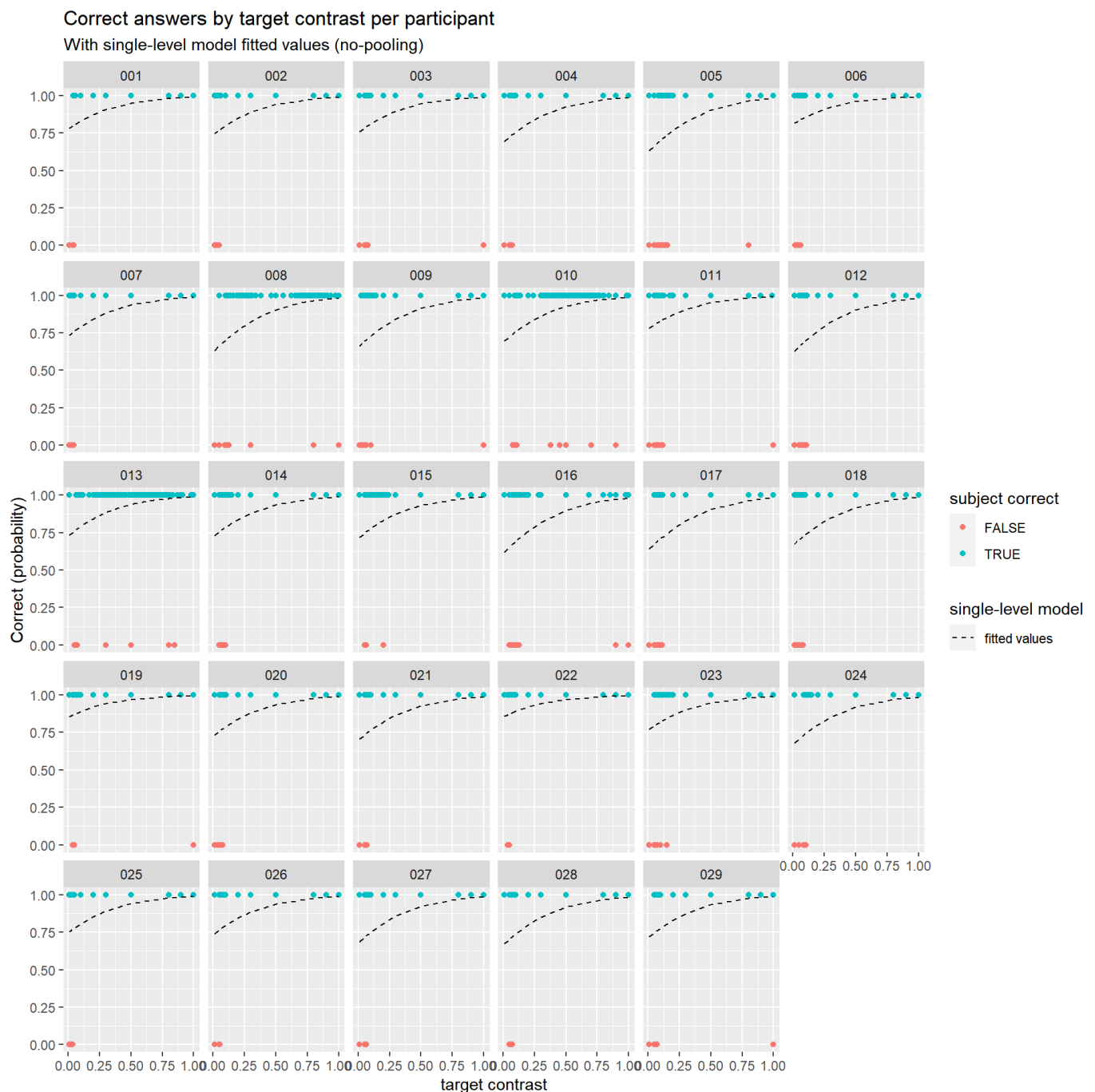
Ex1.p2.iii Comment on the fits - do we have enough data to plot the logistic functions?

The fits are not great. We can see that the lower contrast seems to result in more incorrect answers but even so, subjects still get correct answer at low contrast. As this is a logical outcome variable, the probabilities for most subjects start around 75%. If the contrast is 0.00, we would expect that the subject would not be able to correctly identify the target stimulus, and should start at 50% (chance level). It does seem as though there is sufficient data, the minimum number of samples for a subject is 70, although there are not the same number of samples for each contrast value.

```
m2 <- glmer(correct ~ target.contrast + (1 + target.contrast | subject),
    data = samples_staircase,
    family = binomial(link = "logit"))
summary(m2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: correct ~ target.contrast + (1 + target.contrast | subject)
##    Data: samples_staircase
##
##      AIC      BIC   logLik deviance df.resid
##   5988.5   6021.6  -2989.2   5978.5     5598
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.7671  0.0068  0.5532  0.5915  0.9264
##
## Random effects:
##  Groups  Name             Variance Std.Dev. Corr
##  subject (Intercept)       0.2717  0.5213
##          target.contrast 42.7577  6.5389   -0.84
## Number of obs: 5603, groups:  subject, 29
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.5619     0.1580   3.557 0.000376 ***
## target.contrast   8.7132     2.3604   3.691 0.000223 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## trgt.cntrst -0.907
```

```
samples_staircase %>%
    ggplot(aes(target.contrast, as.integer(correct), color = correct)) +
    geom_point() +
    geom_line(aes(target.contrast, fitted(m2),
        linetype = "fitted values"), inherit.aes = FALSE) +
    scale_linetype_manual(name = "multi-level model", values = c("dashed")) +
    facet_wrap(~subject) +
    labs(title = "Correct answers by target contrast per participant",
        subtitle = "With multi-level model fitted values (partial-pooling)",
        color = "subject correct") +
    ylab("Correct (probability)") +
    xlab("target contrast")
```


Correct answers by target contrast per participant

With multi-level model fitted values (partial-pooling)
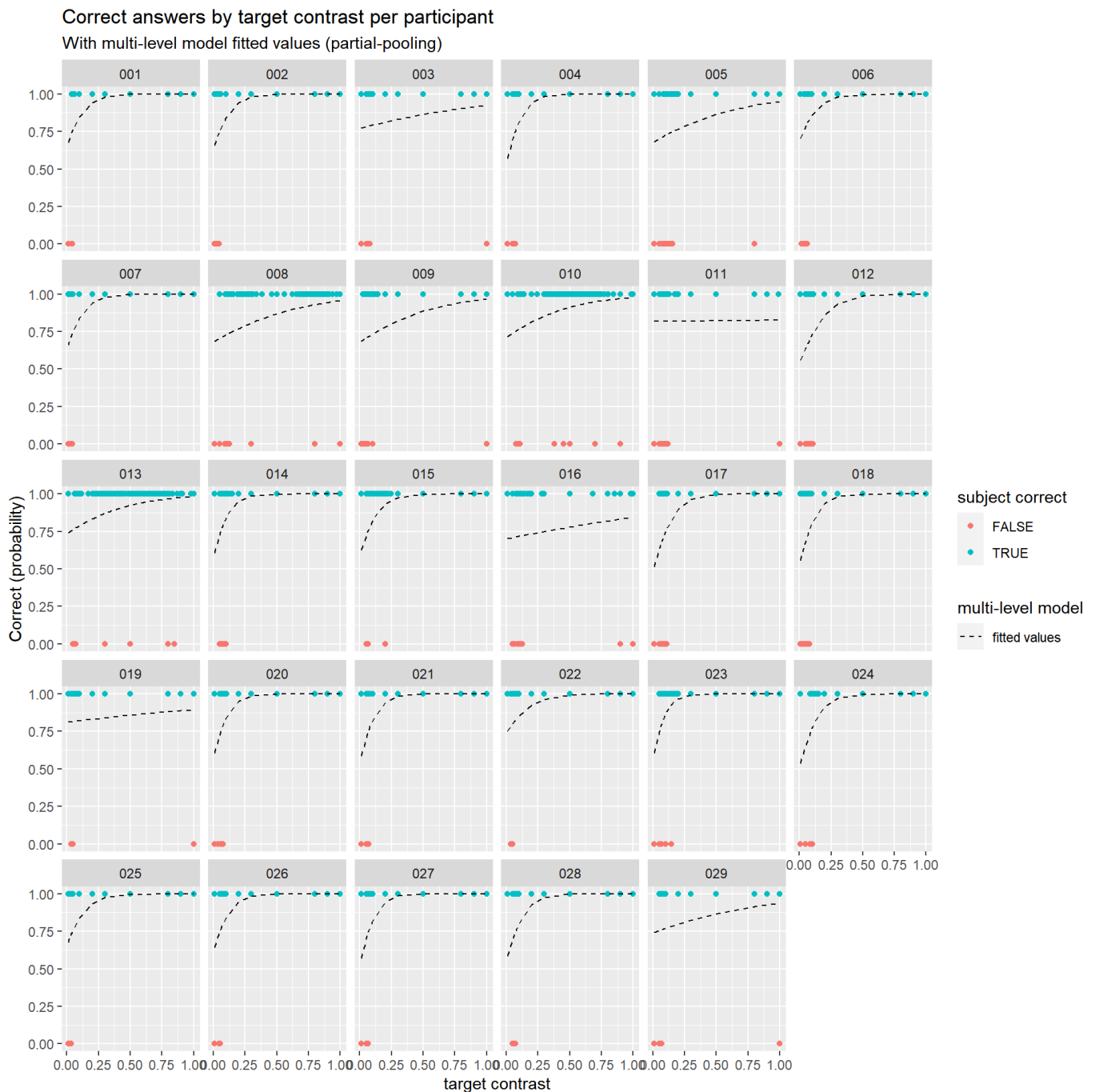
Ex1.p2. v. in your own words, describe how the partial pooling model allows for a better fit for each subject

It allows for per-subject differences in their response to the target contrast. If some subjects have more difficulty seeing differences in low contrast images, we would expect their accuracy to decrease more than other subjects as the contrast values becomes lower. We would conceptually expect no-pooling models to create a better fit for each subject(?), as the subjective models are modelled solely on a given subject. However we would of course always choose the partial pooling models as these would give us great subject-fits but also a model generalizabel for the population.

Exercise 2

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

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
   i. comment on these
   ii. does a log-transformation of the response time data improve the Q-Q-plots?
2. Now do a partial pooling model modelling objective response times as dependent on *task*? (set `REML=FALSE` in your `lmer`-specification)
   i. which would you include among your random effects and why? (support your choices with relevant measures, taking into account variance explained and number of parameters going into the modelling)
   ii. explain in your own words what your chosen models says about response times between the different tasks
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?
   ii. create a model by adding random intercepts (without modelling slopes) that results in a singular fit - then use `print(VarCorr(<your.model>), comp='Variance')` to inspect the variance vector - explain why the fit is singular (Hint: read the first paragraph under details in the help for `isSingular`)
   iii. in your own words - how could you explain why your model would result in a singular fit?

Answers

Exercise 2, part 1 - KV

We will use subjects 002, 009, 014 and 015.

```
samples_experiment <- samples %>% filter(
    trial.type == "experiment" &
    (
        subject == "002" |
        subject == "009" |
        subject == "014" |
        subject == "015"
    ))

m3 <- lmer(rt.obj ~ (1 | subject),
    data = samples_experiment)
summary(m3)
```
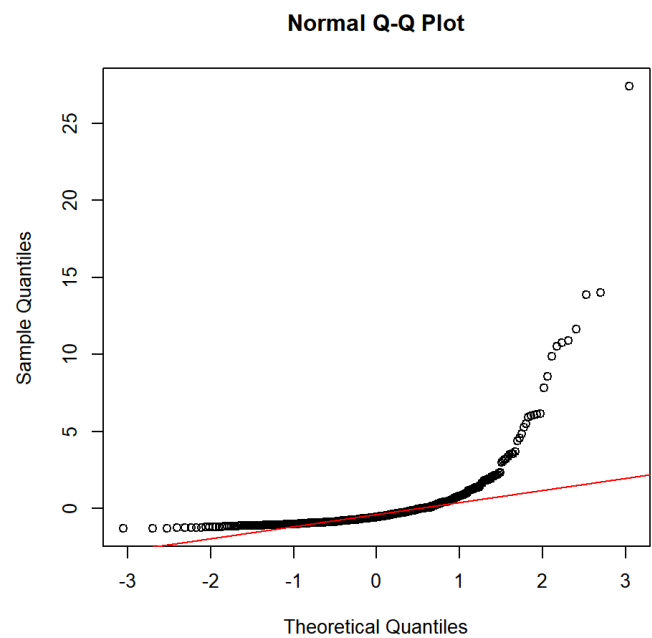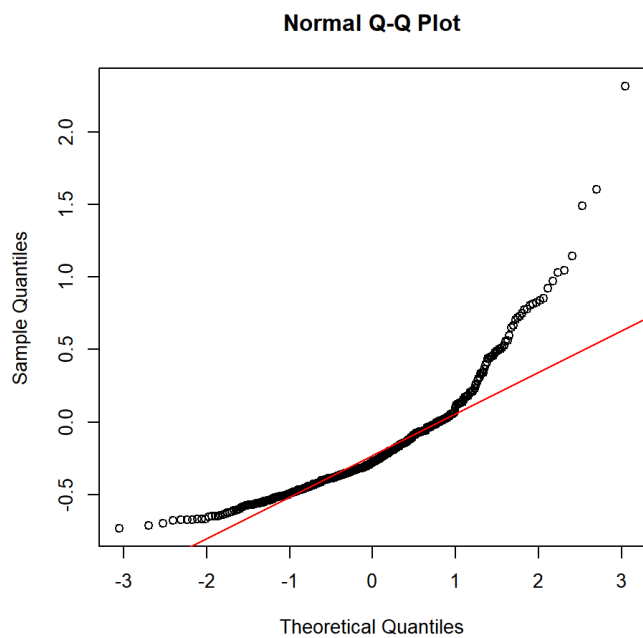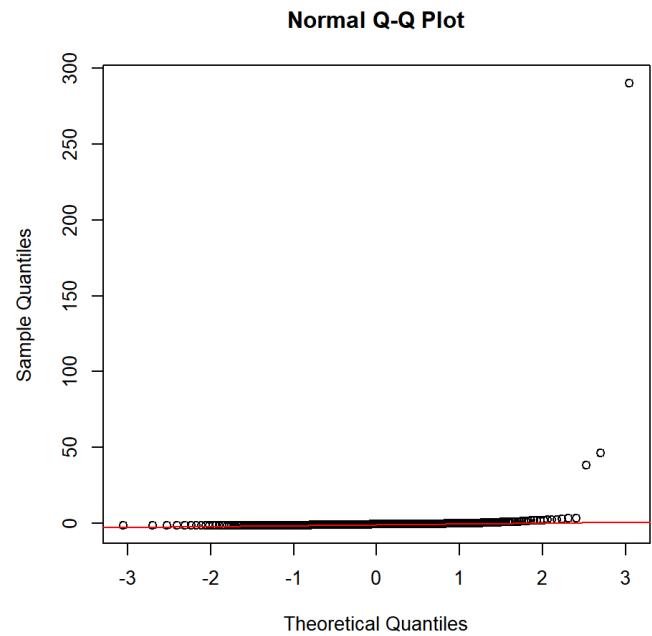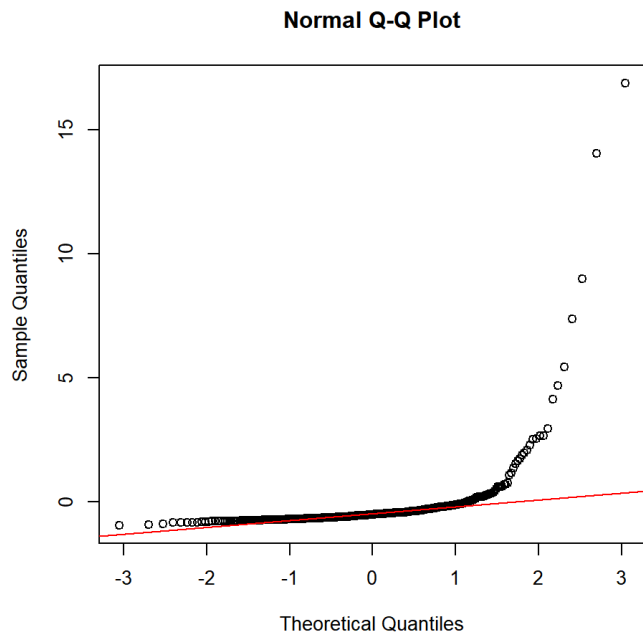
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: rt.obj ~ (1 | subject)
##    Data: samples_experiment
##
## REML criterion at convergence: 11774.5
##
## Scaled residuals:
##    Min     1Q Median    3Q    Max
## -0.214 -0.098 -0.063 -0.019 39.793
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept)  0.1621  0.4026
##  Residual             53.2091  7.2945
## Number of obs: 1728, groups:  subject, 4
##
## Fixed effects:
##             Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)    1.239     0.267 3.000    4.64   0.0189 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
samples_experiment$m3_resid <- residuals(m3)

# make 2x2 plot grid
par(mfrow = c(2, 2))
# plot qq plots for objective response time of subjects:
# 002
qqnorm(samples_experiment[samples_experiment$subject == "002", ]$m3_resid)
qqline(samples_experiment[samples_experiment$subject == "002", ]$m3_resid,
    col = "red")
# 009
qqnorm(samples_experiment[samples_experiment$subject == "009", ]$m3_resid)
qqline(samples_experiment[samples_experiment$subject == "009", ]$m3_resid,
    col = "red")
# 014
qqnorm(samples_experiment[samples_experiment$subject == "014", ]$m3_resid)
qqline(samples_experiment[samples_experiment$subject == "014", ]$m3_resid,
    col = "red")
# 015
qqnorm(samples_experiment[samples_experiment$subject == "015", ]$m3_resid)
qqline(samples_experiment[samples_experiment$subject == "015", ]$m3_resid,
    col = "red")
```

Ex2.p1.i

These data show a significant tail with some reaction times going far beyond the mean (e.g. 15 seconds +).

```
m3l <- lmer(log(rt.obj) ~ (1 | subject),
    data = samples_experiment)
summary(m3l)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: log(rt.obj) ~ (1 | subject)
##    Data: samples_experiment
##
## REML criterion at convergence: 4775.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -8.4491 -0.4344  0.0168  0.4933  6.4941
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 0.08853  0.2975
##  Residual             0.92008  0.9592
## Number of obs: 1728, groups:  subject, 4
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  -0.3916     0.1505  3.0000  -2.601   0.0803 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
samples_experiment$m3l_resid <- residuals(m3l)

# plot qq plots for log transformed objective response time of subjects:
# 002
qqnorm(log(samples_experiment[samples_experiment$subject == "002", ]$m3l_resid))
qqline(log(samples_experiment[samples_experiment$subject == "002", ]$m3l_resid),
    col = "red")
```

Normal Q-Q Plot
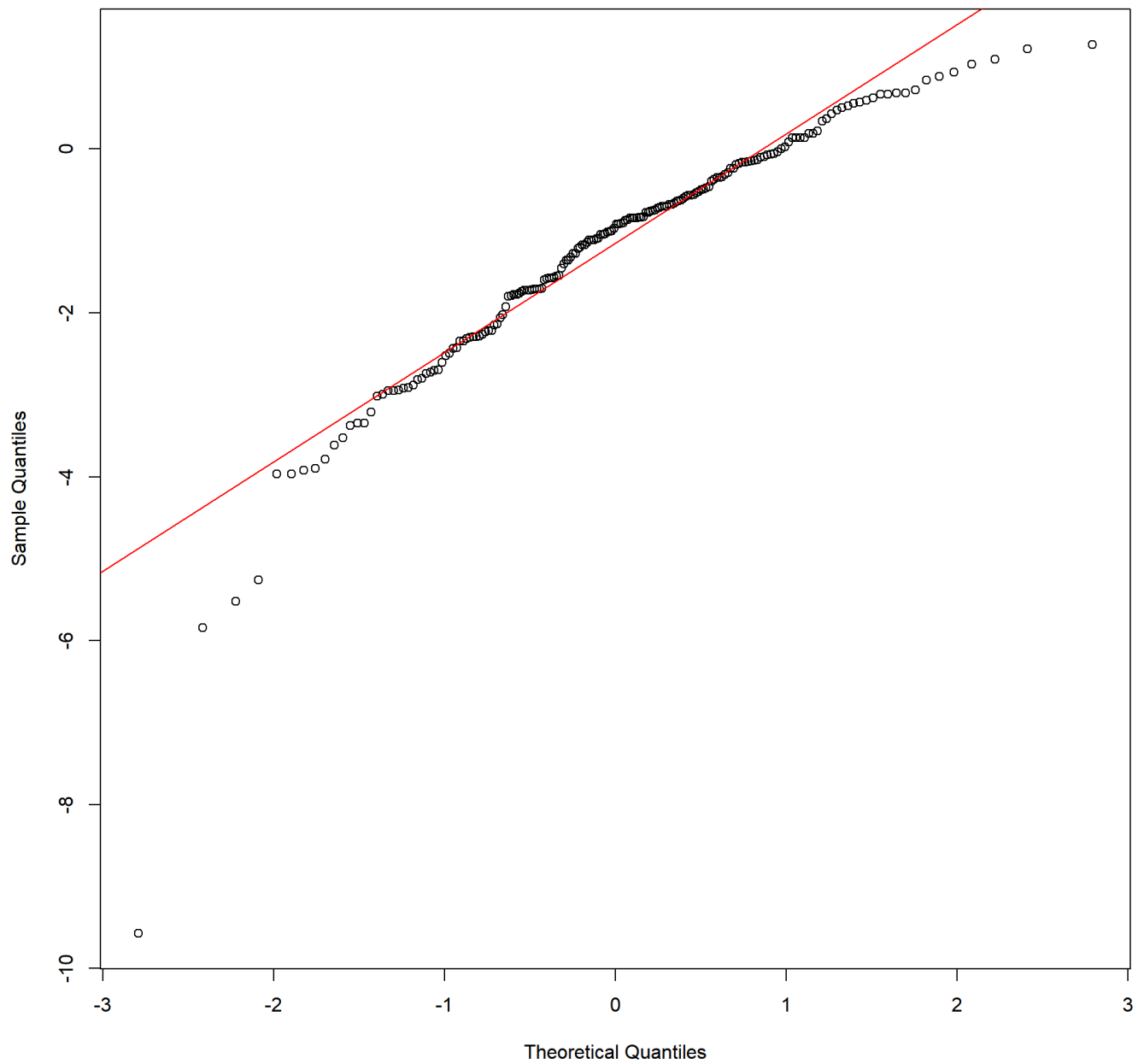

```
# 009
qqnorm(log(samples_experiment[samples_experiment$subject == "009", ]$m3l_resid))
qqline(log(samples_experiment[samples_experiment$subject == "009", ]$m3l_resid),
    col = "red")
```

Normal Q-Q Plot
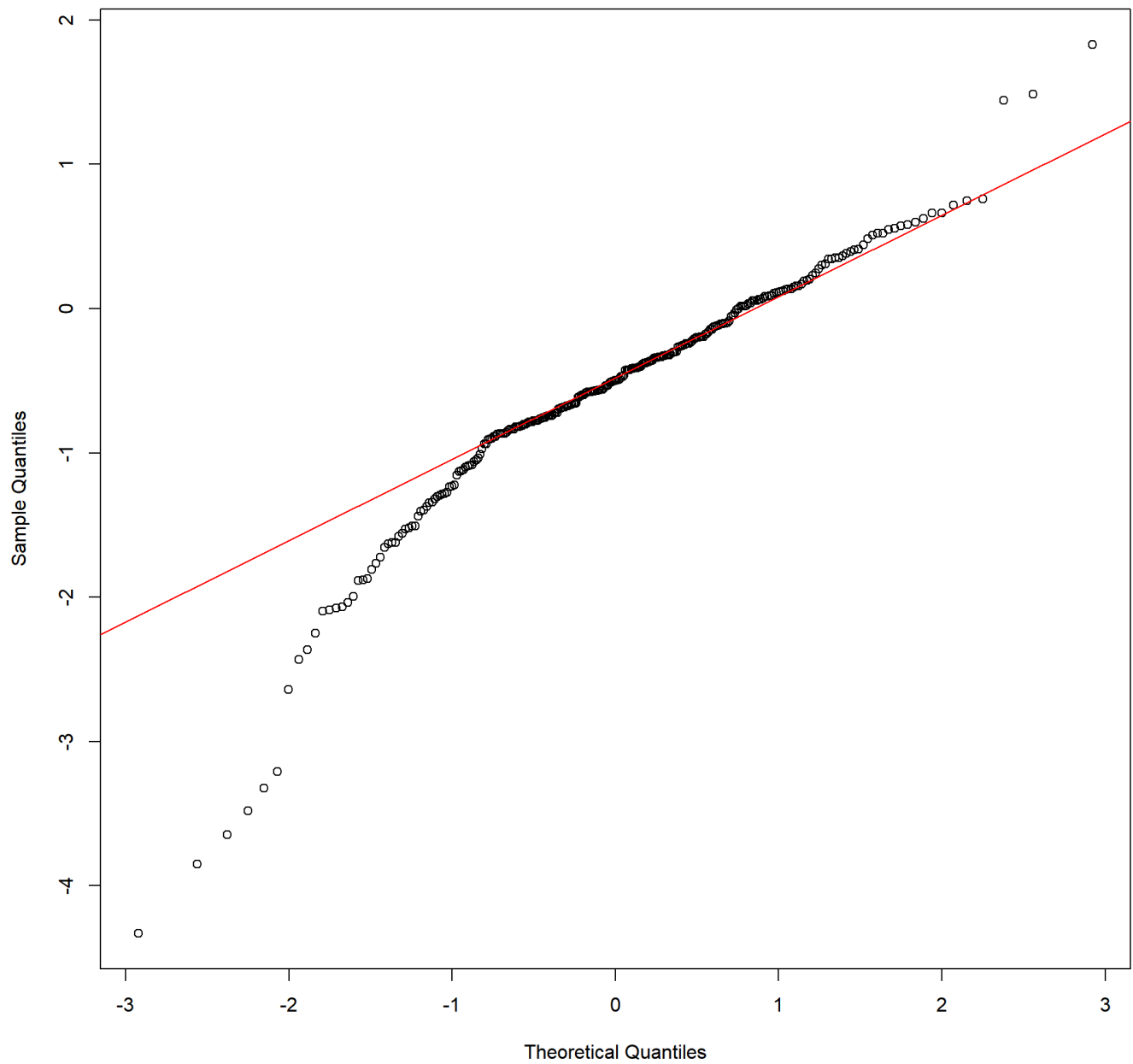
```
# 014
qqnorm(log(samples_experiment[samples_experiment$subject == "014", ]$m3l_resid))
qqline(log(samples_experiment[samples_experiment$subject == "014", ]$m3l_resid),
    col = "red")
```
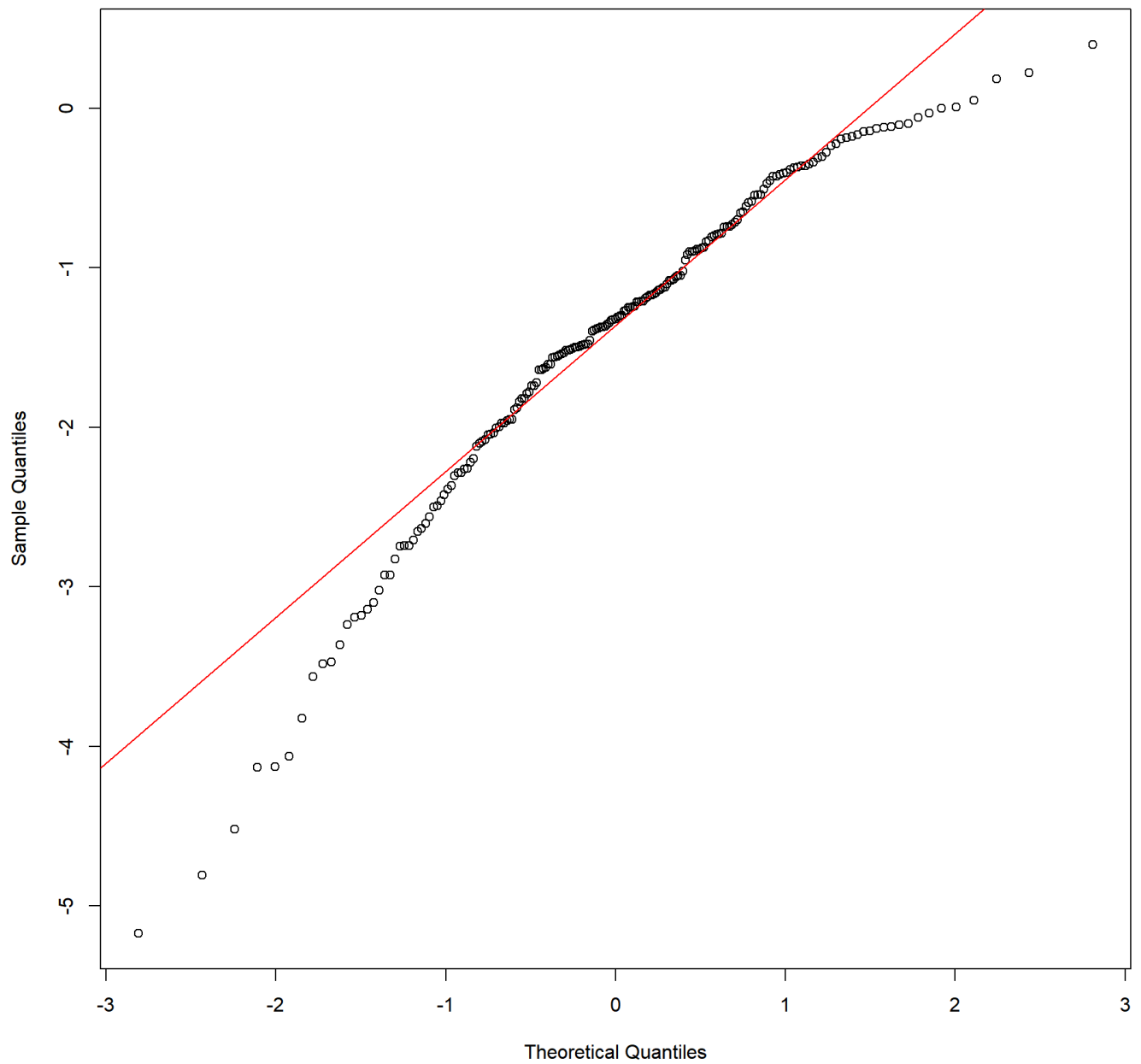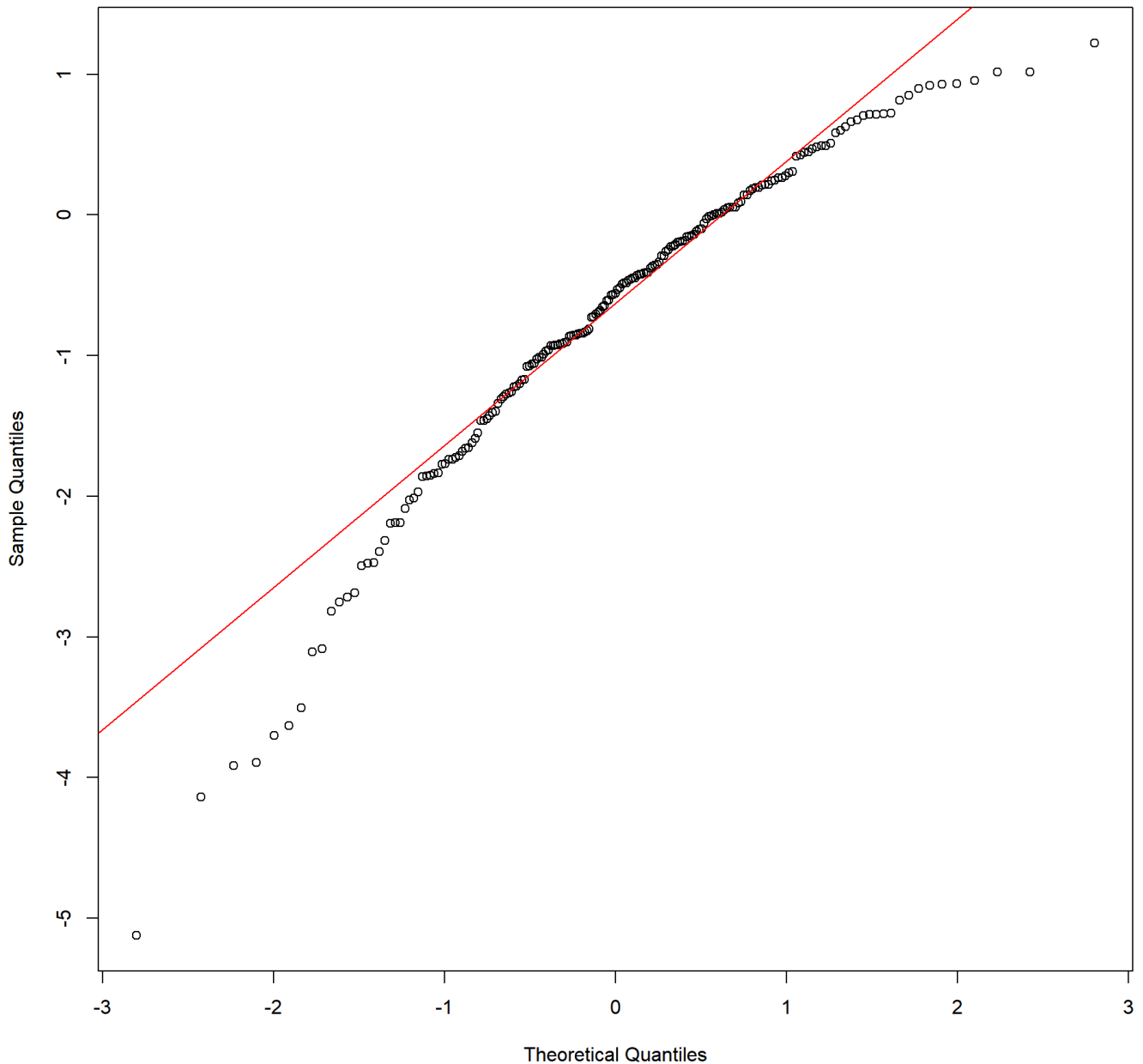
Normal Q-Q Plot


```
# 015
qqnorm(log(samples_experiment[samples_experiment$subject == "015", ]$m3l_resid))
qqline(log(samples_experiment[samples_experiment$subject == "015", ]$m3l_resid),
    col = "red")
```

Normal Q-Q Plot


```
# reset plot grid
par(mfrow = c(1, 1))
```

Ex2.p1.ii

Log-transforming the data definitely creates a better fit, as can be seen in the qq-plots.

Exercise 2, part 2 - VK

```
m4 <- lmer(rt.obj ~ task + (1 | subject),
    REML = FALSE, data = samples_experiment)
summary(m4)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##    method [lmerModLmerTest]
## Formula: rt.obj ~ task + (1 | subject)
##    Data: samples_experiment
##
##       AIC      BIC   logLik deviance df.resid
##   11780.8  11808.0  -5885.4  11770.8     1723
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -0.258 -0.111 -0.058 -0.010 39.780
##
## Random effects:
##  Groups   Name         Variance Std.Dev.
##  subject  (Intercept)   0.09096 0.3016
##  Residual              53.12390 7.2886
## Number of obs: 1728, groups:  subject, 4
##
## Fixed effects:
##                  Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)        1.6487     0.3391   18.4270   4.863 0.000117 ***
## taskquadruplet    -0.6554     0.4295 1724.0000  -1.526 0.127187
## tasksingles       -0.5736     0.4295 1724.0000  -1.336 0.181883
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tskqdr
## taskqudrplt -0.633
## tasksingles -0.633  0.500
```

```
piecewiseSEM::rsquared(m4)
```

```
##   Response   family    link method   Marginal Conditional
## 1   rt.obj gaussian identity   none 0.001596197  0.00330281
```

2.2.i. which would you include among your random effects and why?

The above model uses random intercepts for subject. We considered that task and trial might be good additional random effects, but these result in a singular fit. We considered task to be a relevant random effect, as it was expected that the different tasks would influence the reaction time. We considered trial to be a relevant random effect as participants might perform better throughout trials.

Overall this model does not explain much of the variance (0.33% about 0.16% fixed effects and 0.17% random effects). The model shows no statistically significant difference between response times per task.

Exercise 2, part 3 - LR

```
# model with PAS-task interaction.
m5 <- lmer(rt.obj ~ task + pas:task + (1 | subject),
    REML = FALSE, data = samples_experiment)
summary(m5)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: rt.obj ~ task + pas:task + (1 | subject)
##    Data: samples_experiment
##
##      AIC      BIC   logLik deviance df.resid
##  11785.3  11828.9  -5884.6  11769.3     1720
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -0.298 -0.107 -0.053 -0.006 39.756
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept)  0.08422 0.2902
##  Residual             53.08183 7.2857
## Number of obs: 1728, groups:  subject, 4
##
## Fixed effects:
##                     Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)          2.38869    0.73777 284.85013   3.238  0.00135 **
## taskquadruplet      -1.54762    1.03036 1727.71994  -1.502  0.13328
## tasksingles         -1.03349    1.01117 1727.59916  -1.022  0.30689
## taskpairs:pas       -0.42621    0.37819 1539.29789  -1.127  0.25993
## taskquadruplet:pas   0.09117    0.40230 1716.78531   0.227  0.82075
## tasksingles:pas     -0.15525    0.35936 1174.87743  -0.432  0.66581
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tskqdr tsksng tskpr: tskqd:
## taskqudrplt -0.685
## tasksingles -0.693  0.497
## taskpars:ps -0.890  0.633  0.639
## tskqdrplt:p -0.005 -0.648 -0.002  0.006
## tsksngls:ps -0.014  0.004 -0.631  0.015  0.010
```

Ex2.p3.i

Any additional random effects we added besides subject resulted in singular fits.

```
# here already it becomes a singular fit
m6 <- lmer(rt.obj ~ task + pas:task + (1 | subject) + (1 | task),
    REML = FALSE, data = samples_experiment)
```

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

```
summary(m6)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: rt.obj ~ task + pas:task + (1 | subject) + (1 | task)
##    Data: samples_experiment
##
##      AIC      BIC   logLik deviance df.resid
##  11787.3  11836.4  -5884.6  11769.3     1719
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -0.298 -0.107 -0.053 -0.006 39.756
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept)  0.08422 0.2902
##  task     (Intercept)  0.00000 0.0000
##  Residual             53.08183 7.2857
## Number of obs: 1728, groups:  subject, 4; task, 3
##
## Fixed effects:
##                    Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)         2.38869    0.73777  284.85013   3.238  0.00135 **
## taskquadruplet     -1.54762    1.03036 1727.71994  -1.502  0.13328
## tasksingles        -1.03349    1.01117 1727.59916  -1.022  0.30689
## taskpairs:pas      -0.42621    0.37819 1539.29788  -1.127  0.25993
## taskquadruplet:pas  0.09117    0.40230 1716.78531   0.227  0.82075
## tasksingles:pas    -0.15525    0.35936 1174.87743  -0.432  0.66581
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) tskqdr tsksng tskpr: tskqd:
## taskqudrplt -0.685
## tasksingles -0.693  0.497
## taskpars:ps -0.890  0.633  0.639
## tskqdrplt:p -0.005 -0.648 -0.002  0.006
## tsksngls:ps -0.014  0.004 -0.631  0.015  0.010
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
print(VarCorr(m6), comp = "Variance")
```

```
##  Groups   Name        Variance
##  subject  (Intercept)  0.084224
##  task     (Intercept)  0.000000
##  Residual             53.081833
```

Ex2.p3.iii. in your own words - how could you explain why your model would result in a singular fit?

In the example above, model m6, it results in a singular fit. Singular fits occur when the variances of some of the linear combinations of the effects are either zero or close to zero, and it is indicating that the model may be overfitting for the data. In model m6, we get the variance for task as 0, which results in a singular fit - this indicates to us that this model is overfitted to the data.

We expect that the reason that the model results in a singular fit is because we have too many parameters, and is therefore too too complex. As mentioned above, adding any additional random effects besides subject resulted in getting singular fits, so in order for the example above, m6, to not result in a singular fit, we would have to remove (1|Task).

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
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
   i. which family should be used?
   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 (*bobyqa*) for which the `dfoptim` package is needed. In `glmer`, you can add the following for the `control` argument: `glmerControl(optimizer="bobyqa")` (if you are interested, also have a look at the function `allFit`)
   iv. when you have a converging fit - fit a model with only the main effects of *pas* and *task*. Compare this with the model that also includes the interaction
   v. indicate which of the two models, you would choose and why
   vi. based on your chosen model - write a short report on what this says about the distribution of ratings as dependent on *pas* and *task*
   vii. include a plot that shows the estimated amount of ratings for four subjects of your choosing
3. Finally, fit a multilevel model that models *correct* as dependent on *task* with a unique intercept for each *subject*
   i. does *task* explain performance?

ii. add *pas* as a main effect on top of *task* - what are the consequences of that?

iii. now fit a multilevel model that models *correct* as dependent on *pas* with a unique intercept for each *subject*

iv. finally, fit a model that models the interaction between *task* and *pas* and their main effects

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

Answers

Exercise 3, part 1 - KV

```
data.count <- samples %>% select(subject, pas, task, correct) %>%
    group_by(subject, task, pas) %>%
    summarize(
        subject = subject[1],
        task = task[1],
        pas = pas[1],
        count = n(),
        accuracy = sum(correct) / n(),
        .groups = "drop")
head(data.count)
```

```
## # A tibble: 6 x 5
##    subject task          pas count accuracy
##    <fct>   <fct>       <int> <int>    <dbl>
## 1 001     pairs           1   109    0.532
## 2 001     pairs           2    45    0.756
## 3 001     pairs           3     4    1
## 4 001     pairs           4    12    1
## 5 001     quadruplet      1   141    0.596
## 6 001     quadruplet      2    23    0.652
```

Exercise 3, part 2 - EH

```
# multi-level with slope and interaction
m7 <- glmer(count ~
    pas + task + pas:task +
    (1 + pas | subject),
    data = data.count,
    family = poisson,
    control = glmerControl(optimizer = "bobyqa"))
summary(m7)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson ( log )
## Formula: count ~ pas + task + pas:task + (1 + pas | subject)
##    Data: data.count
## Control: glmerControl(optimizer = "bobyqa")
##
##      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 11.6477
##
## Random effects:
##  Groups  Name        Variance Std.Dev. Corr
##  subject (Intercept) 1.2017   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)         4.28937    0.20577  20.846  < 2e-16 ***
## pas                -0.19479    0.08798  -2.214   0.0268 *
## taskquadruplet      0.16669    0.04007   4.160 3.18e-05 ***
## tasksingles        -0.39660    0.04192  -9.461  < 2e-16 ***
## pas:taskquadruplet -0.07195    0.01606  -4.480 7.47e-06 ***
## pas:tasksingles     0.16855    0.01587  10.622  < 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
```

Ex3.p2.i

For this model (m7) we used the poisson family because the poisson distribution is specifically used to describe probabilities of event frequencies.

Ex3.p2.ii

There is not really a slope being modelled for PAS as PAS is not continuous, it is a factor, so we only get esimates for each PAS level. As a factor, PAS is treated categorically (PAS1, PAS2, PAS3, PAS4), there is no such thing as PAS 1.3 etc and as such, it can not really be modelled as a slope.

```
m7.1 <- glmer(count ~
    pas + task + (1 + pas | subject),
    data = data.count,
    family = poisson,
    control = glmerControl(optimizer = "bobyqa"))
summary(m7.1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##    Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: count ~ pas + task + (1 + pas | subject)
##     Data: data.count
## Control: glmerControl(optimizer = "bobyqa")
##
##       AIC       BIC    logLik deviance df.resid
##    4923.2    4950.0   -2454.6   4909.2      333
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.9891 -2.1490 -0.1911  1.8233 11.0534
##
## Random effects:
##  Groups  Name        Variance Std.Dev. Corr
##  subject (Intercept) 1.2147   1.1021
##          pas         0.2232   0.4725   -0.99
## Number of obs: 340, groups:  subject, 29
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     4.213654   0.205804  20.474   <2e-16 ***
## pas            -0.162486   0.088064  -1.845    0.065 .
## taskquadruplet  0.006593   0.018184   0.363    0.717
## tasksingles     0.004307   0.018189   0.237    0.813
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) pas    tskqdr
## pas         -0.989
## taskqudrplt -0.044  0.000
## tasksingles -0.044  0.000  0.501
```

```
AIC(m7, m7.1)
```

```
##      df      AIC
## m7    9 4685.119
## m7.1  7 4923.190
```

Ex3.p2.v Indicate which of the two models, you would choose and why?

Between m7 and m7.1, we believe that the m7 is the best model. Out of the two, m7 is the only one thats predictors have any significance on the outcome, in particular, the interactions in m7 have significant effect on the count variable. When comparing the AIC output, it is also shown that model m7 has a better fit, with an AIC of 3148.441, compared to m7.1 of 3398.549. Taking all this into consideration, we are inclined to use this model m7 out of the two as it is the better model.

Ex3.p2.vi based on your chosen model - write a short report on what this says about the distribution of ratings as dependent on pas and task ?

m7 tells us that there is an important interaction between pas and task when it comes to the distribution of ratings, as all interactions reach significance, $p < 0.05$, except for pas3:taskquadruplet, $p = 0.92020$ and pas2:taskquadruplet, $p = 0.06010$

We can see from m7 that there is a decrease of all the interactions between pas and quadruplet tasks (pas:quadruplet), est. -0.0719. We see the opposite in the case all counts of interactions between pass and single tasks, where there is an increase of est. ~0.169.

Looking at pas in m7, there is an increase of count in all pas when going from pas4 to pas1 (0.772932), 2 (est. 0.749155) and 3 (est. 0.259281). Only the predictions for pas1 and pas2 are significant, $p < 0.05$.

Regarding task in our model, count decreases going from task pairs to task quadruplets, est. -0.100, and count increases going from task pairs to task singles, ~ est. 0.333. Both of these predications are significant, $p < 0.05$.

```
# get only specified subjects
data.count_subset <- data.count %>%
    filter(
        subject == "002" |
        subject == "009" |
        subject == "014" |
        subject == "015")

# predict using model for only selected subjects
m7_fitted <- predict(m7, newdata = data.count_subset)

# plot. is this right? the predictions seem way off...
data.count_subset %>% ggplot(aes(task, count, color = task)) +
    geom_point() +
    geom_point(aes(x = task, y = m7_fitted), shape = 18, size = 3) +
    facet_wrap(subject~pas)
```
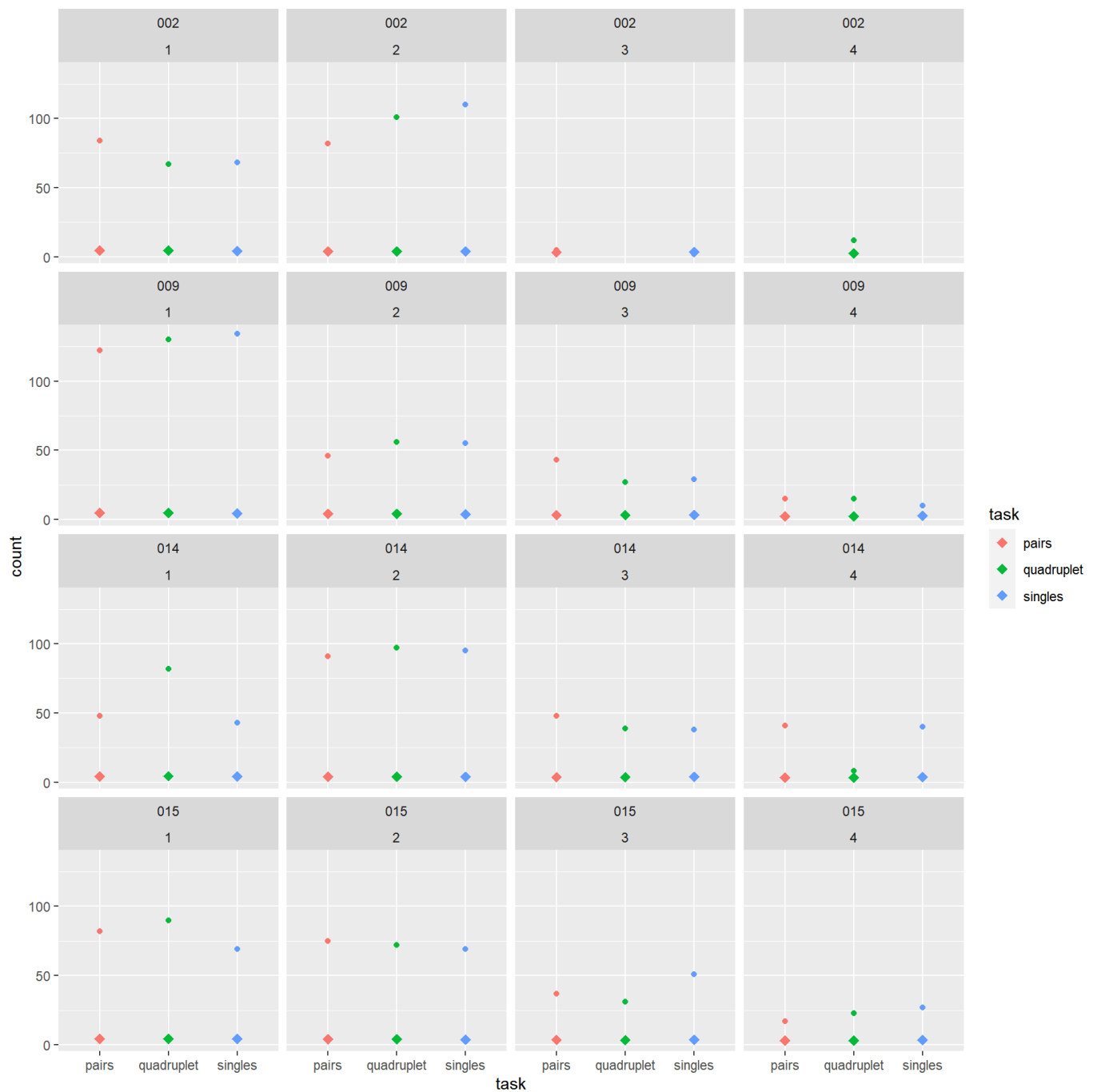
Exercise 3, part 3 - LR

```
m8 <- glmer(correct ~ task +
    (1 | subject),
    data = samples,
    family = binomial(link = "logit"))
summary(m8)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: correct ~ task + (1 | subject)
##    Data: samples
##
##      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    0.04190  -2.345    0.019 *
## tasksingles     0.18542    0.04337   4.276 1.91e-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
```

```
# seems task only explains a tiny amount of performance
piecewiseSEM::rsquared(m8)
```

```
##   Response   family  link method    Marginal Conditional
## 1  correct binomial logit  delta 0.002526141  0.03489787
```

Ex3.p3.i

Differences in the task seem to explain performance (number of correct answers)
to a statistically significant level.

```
m9 <- glmer(correct ~ task + pas +
    (1 | subject),
    data = samples,
    family = binomial(link = "logit"))
summary(m9)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial ( logit )
## Formula: correct ~ task + pas + (1 | subject)
##    Data: samples
##
##      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   0.098399  -9.656   <2e-16 ***
## taskquadruplet -0.029418   0.045016  -0.653    0.513
## tasksingles    -0.008914   0.046889  -0.190    0.849
## pas             1.014031   0.022900  44.281   <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 -0.189  0.489
## pas         -0.421  0.030 -0.083
```

```
# better
piecewiseSEM::rsquared(m9)
```

```
##   Response   family  link method  Marginal Conditional
## 1  correct binomial logit  delta 0.1925254   0.2219821
```

Ex3.p3.ii

When we add PAS as a main effect on top of task, we can see that PAS actually
has a significant effect on performance, whereas now task no longer displays a
significant effect. This means that by including PAS as a main effect in the model, it
altered the significance of task on performance.

```
m10 <- glmer(correct ~ pas +
    (1 | subject),
    data = samples,
    family = binomial(link = "logit"))
summary(m10)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: correct ~ pas + (1 | subject)
##    Data: samples
##
##      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    0.09504  -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
```

```
# task makes negligible diff
piecewiseSEM::rsquared(m10)
```

```
##   Response   family  link method  Marginal Conditional
## 1  correct binomial logit  delta 0.1925749   0.2220484
```

```
m11 <- glm(correct ~ pas + task +
    pas:task,
    data = samples,
    family = binomial(link = "logit"))
summary(m11)
```

```
## 
## Call:
## glm(formula = correct ~ pas + task + pas:task, family = binomial(link = "logit"),
##     data = samples)
## 
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4709  -1.2372   0.4923   0.7704   1.1188
## 
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.70591    0.07006 -10.077   <2e-16 ***
## pas                   0.88428    0.03526  25.081   <2e-16 ***
## taskquadruplet        0.05152    0.09696   0.531    0.595
## tasksingles          -0.10956    0.10248  -1.069    0.285
## pas:taskquadruplet   -0.04835    0.04925  -0.982    0.326
## pas:tasksingles       0.07069    0.05017   1.409    0.159
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 20418  on 18130  degrees of freedom
## Residual deviance: 17864  on 18125  degrees of freedom
## AIC: 17876
## 
## Number of Fisher Scoring iterations: 5
```

```
piecewiseSEM::rsquared(m11)
```

```
##   Response   family  link     method R.squared
## 1  correct binomial logit nagelkerke       NaN
```

```
m12 <- glmer(correct ~ pas + task +
    pas:task + (1 | subject),
    data = samples,
    family = binomial(link = "logit"))
summary(m12)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: correct ~ pas + task + pas:task + (1 | subject)
##    Data: samples
##
##      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    0.11163  -8.360   <2e-16 ***
## pas                  1.00447    0.03749  26.794   <2e-16 ***
## taskquadruplet       0.05256    0.09943   0.529    0.597
## tasksingles         -0.15466    0.10509  -1.472    0.141
## pas:taskquadruplet  -0.04782    0.05053  -0.946    0.344
## pas:tasksingles      0.07731    0.05136   1.505    0.132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) pas    tskqdr tsksng ps:tskq
## pas         -0.600
## taskqudrplt -0.467  0.628
## tasksingles -0.435  0.583  0.493
## ps:tskqdrpl  0.413 -0.694 -0.893 -0.437
## ps:tsksngls  0.405 -0.680 -0.455 -0.894  0.505
```

```
piecewiseSEM::rsquared(m12)
```

```
##   Response   family  link method  Marginal Conditional
## 1  correct binomial logit  delta 0.1944348    0.2238636
```

```
AIC(m8, m9, m10, m12)
```

```
##      df      AIC
## m8    4 19927.21
## m9    5 17425.01
## m10   3 17421.46
## m12   7 17422.82
```

Ex3.p3.v Describe in your words which model is the best in explaining the variance in accuracy?

Of the above models, m12 is the best at explaining the variance in accuracy. But we think that model 10 is a bettter model to use because the difference in explained variance is negligible and it is a simpler model which means it is less prone to overfitting. Adding the random intercepts per subject allows for differences in subjects ability to correctly classify the stimuli. These difference in subjects may come from situational stress. Model m10 also has the lowest AIC score (17421.39), even though it has slightly more residual deviance than model 12, but this makes sense as more complex models are punished in this area.