Reaction times and priming visualised with R

9419009

28 February 2019

Question 1

The date file “data1.csv” contains the reaction time data from 80 participants. The participants’ task was to respond to a word on the computer screen that was either a synonym for “fast” or a synonym for “slow”. We predicted that people will respond more quickly to words that mean “fast” than to words that mean “slow”. We thus have one condition with two levels. Each participant saw 16 items – half in one condition, half in the other.

Firstly, install and load the relevant libraries to analyse dataset.

```
#install.packages("lm4")
library(lme4)

#install.packages("lmerTest")
library("lmerTest")

#install.packages("ggplot2")
library("ggplot2")
#install.packages("stringi")
library("stringi")
library(tidyverse)
```

Secondly, import dataset 1.

```
data1_4_ <-
read_csv("C:/Users/omarf/OneDrive/Desktop/Mixed_Models_Ass/Ass_data/data1(4).csv")
```

```
## Parsed with column specification:
## cols(
##   Participant = col_double(),
##   Item = col_double(),
##   RT = col_double(),
##   Condition = col_character()
## )
```

We then need to create a linear mixed model for our dataset. The model is a repeated measures experiment where participants observed one factor with two levels. The different slopes and intercepts have been added to the code.

```
model_1 <- lmer(RT ~ Condition + (1 + Condition | Participant) + (1 + Condition | Item), data =
data1_4_)
```

We receive an error of “singular fit” as a result, we need to simplyfy the random effect structure. The following are attempts to simplify the R code.

```
model1_2 <-lmer(RT ~ Condition + (1 | Participant) + (1 + Condition | Item), data = data1_4_,
REML = TRUE )
```

```
## control$checkConv, : Model failed to converge with max|grad| = 0.0030936
## (tol = 0.002, component 1)
```

```
model1_2 <-lmer (RT ~ Condition +(1 | Participant) + (1 | Item), data = data1_4_, REML =
TRUE)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0059014
## (tol = 0.002, component 1)
```

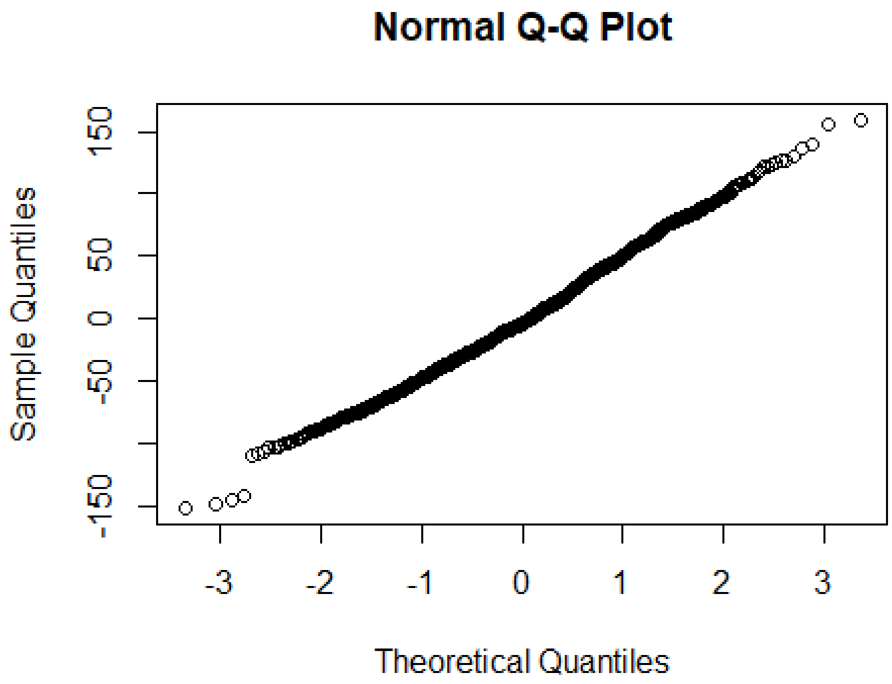
The previous code failed to converge so we need to simplyfy the code futher. The following code allows for the model to be identified.

```
model1_3 <- lmer(RT ~ Condition + (1 | Participant), data = data1_4_, REML = TRUE)
```

When running the model the condition (fast or slow) is a significant predictor of reaction times.

It is then imperative to check the normality of the modal using the qqnorm function in R.

```
qqnorm(residuals(model1_3))
```



In order to determine if our model is significant, we need to know if it differs from what we would expect if our condition factor did not influence reaction times. We thus need to create a null model by removing the condition as a predictor.

```
model1_null <- lmer(RT ~ (1 | Participant), data = data1_4_, REML = TRUE)
```

We then compare both models using the anova function.

```
anova(model1_3, model1_null)



## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model shows that values for our fixed effects are significantly different in comparison to the null model.

```
summary(model1_3)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [

## lmerModLmerTest]

## Formula: RT ~ Condition + (1 | Participant)

##    Data: data1_4_

##

## REML criterion at convergence: 13656.3

##

## Scaled residuals:

##     Min      1Q  Median      3Q     Max

## -3.0707 -0.6837 -0.0746  0.6841  3.2209

##

## Random effects:

##  Groups      Name        Variance Std.Dev.

##  Participant (Intercept)  455.4   21.34

##  Residual                2446.8   49.46

## Number of obs: 1274, groups:  Participant, 80

##

## Fixed effects:

##               Estimate Std. Error       df t value Pr(>|t|)

## (Intercept)    713.646      3.091  124.007  230.87  < 2e-16 ***

## ConditionSlow   12.391      2.772 1193.444    4.47 8.57e-06 ***

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr)
## ConditinSlw -0.451
```

In order to analyze the data, the lme4 package (Bates, Maechler, Bolker & Walker, 2015) is used to fit the linear mixed models for the reaction time data measure in R (R development core Team, 2017). Below the reported regression coefficients (b), standard errors, and t-values are reported in table 1. T- values greater than 1.96 indicate an effect is significant at approximately the alpha level of .05. The model demonstrates that participants who responded to words that were synonymous with “fast” were significantly faster to react (m = 713 ms, SE = 3.09) in comparison to when they had viewed words synonymous with “slow” (m = 725 ms, SE = 2.77). This supports the prediction made by the researchers who stated that “fast” synonymous words would be more quickly reacted to compared to “slow” synonymous words.

The following code visualizes our data using ggplot.

```
ggplot(data1_4_, aes(x = Condition, y =
            RT, colour = Condition)) +
  geom_boxplot() + guides(colour =
                    FALSE)

## Warning: Removed 6 rows containing non-finite values (stat_boxplot).
```

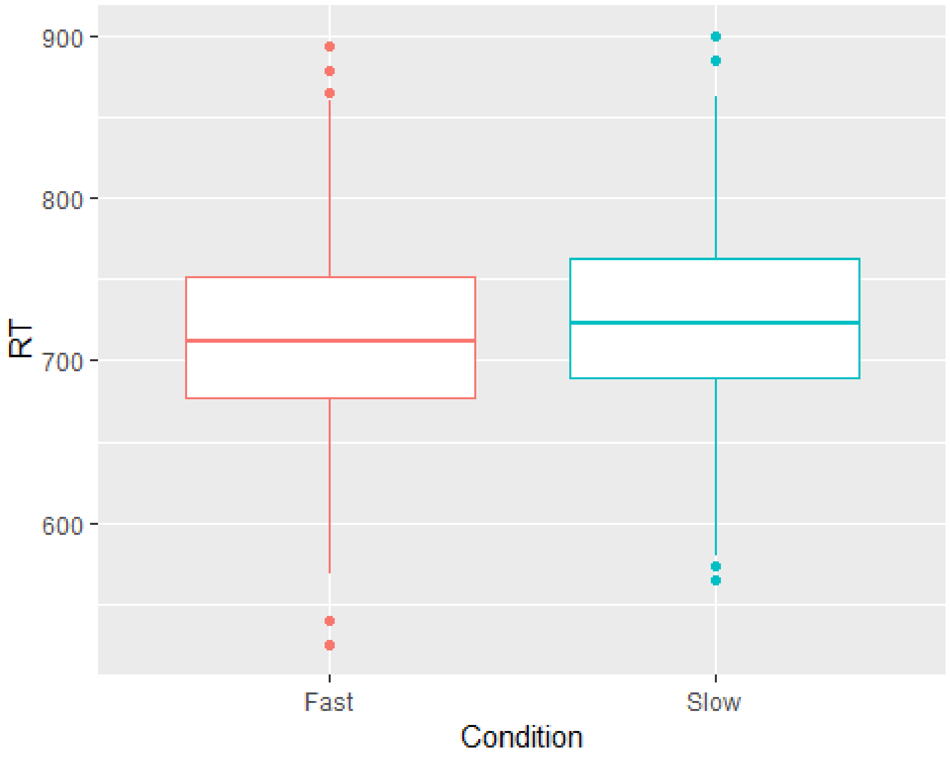


Table 1

	b	SE	t
Intercept	713	3.09	230.87
Condition - Slow	12.39	2.77	4.47

Question 2 (part a)

The data file “data2.csv” contains the reaction time data from 40 participants who took part in a 2 (Prime: Positive vs. Negative) x 2 (Target: Positive vs. Negative) repeated measures experiment to measure people’s reaction times to reading a target sentence following the presentation of an image that acted as a prime. Specifically, the experiment tested our prediction that people would read a sentence describing an emotion more quickly after having just seen an image representing that emotion, relative to after having seen an image representing a different emotion. The Prime images were either Positive or Negative, and the Target sentences described either Positive or Negative emotions. Each participant saw 32 items.

Firstly, import and load the relevant libraries.

```
library("lme4")
library("lmerTest")
library("emmeans")
library("ggplot2")
library("dplyr")
library("magrittr")

## The following object is masked from 'package:tidyr':
t
```

Import the dataset we are going to use.

```
data2_2_ <-
read_csv("C:/Users/omarf/OneDrive/Desktop/Mixed_Models_Ass/Ass_data/data2(2).csv")
```

```
## Parsed with column specification:
## cols(
##   Participant = col_double(),
##   Item = col_double(),
##   RT = col_double(),
##   Prime = col_character(),
##   Target = col_character()
## )
```

We then turn our variables into factors with the follwing code.

```
data2_2_$Prime <- as.factor(data2_2_$Prime)
data2_2_$Target <- as.factor(data2_2_$Target)
```

Contrast weightings need to be implemented to our two factors.

```
contrasts(data2_2_$Prime) <- matrix(c(.5, -.5))
contrasts(data2_2_$Target) <- matrix(c(.5, -.5))
```

Specify the levels for each condition

```
levels(data2_2_$Prime)

## [1] "Negative" "Positive"

levels(data2_2_$Target)

## [1] "Negative" "Positive"
```

We then need to create the model to allow us to conduct a 2 x 2 repeated measures mixed linear model. The different slopes and intercepts are added to the code.

```
model.full <- lmer(RT~Prime*Target + (1+Prime*Target|Participant) + (1+Prime*Target| Item),
data=data2_2_, REML=TRUE)

## singular fit
```

An error of “singular fit” is recieved. We need to simplyfy the random effect structure. The following are attempts to simplyfy the code.

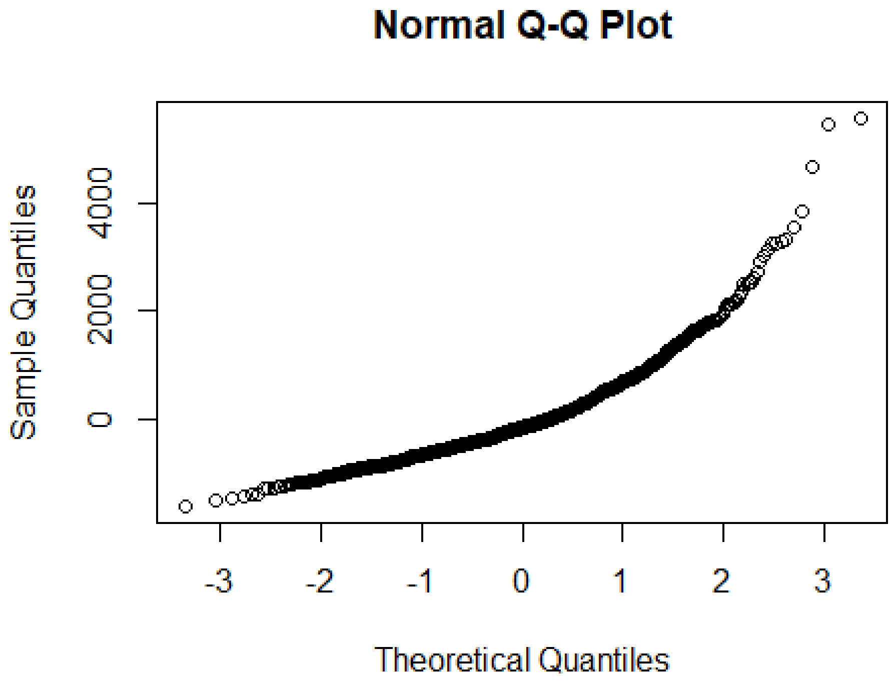
```
model.full <- lmer(RT~Prime*Target + (1+Prime*Target|Participant) + (1+Prime*Target| Item),
data=data2_2_, REML=TRUE)

## singular fit

model.full <- lmer(RT~Prime*Target + (1+Prime+Target|Participant) + (1+Prime+Target|
Item), data=data2_2_, REML=TRUE)

## singular fit

model.full <- lmer(RT~Prime*Target + (1+Prime+Target|Participant) + (1+Prime | Item),
data=data2_2_, REML=TRUE)

## singular fit

model.full <- lmer(RT~Prime*Target + (1+Prime|Participant) + (1+Prime | Item),
data=data2_2_, REML=TRUE)

## singular fit
```

The previous code failed to converge so we need to simplyfy the code futher. The following code allows for the model to be identified.

```
model.full2 <- lmer(RT~Prime*Target + (1+Prime|Participant) + (1+Prime), data=data2_2_,
REML=TRUE)
```

The normality of the model will be tested. The model appears to be normally distributed.

```
qqnorm(residuals(model.full2))
```



In order to determine if our model is significant, we need to know if it differs from what we would expect if our condition factor did not influence reaction times. We thus need to create a null model by removing both conditions as predictors.

```
model.null2 <- lmer(RT~ (1+Prime|Participant) + (1+Prime), data=data2_2_, REML=TRUE)
```

We then run an ANOVA to determine if there is a significant difference between the null model and the experimental model.

```
anova(model.full2, model.null2)

## refitting model(s) with ML (instead of REML)

## Data: data2_2_
## Models:
## model.null2: RT ~ (1 + Prime | Participant) + (1 + Prime)
## model.full2: RT ~ Prime * Target + (1 + Prime | Participant) + (1 + Prime)
##             Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## model.null2  6 20759 20790 -10373    20747
## model.full2  8 20750 20791 -10367    20734 12.822      2   0.001643 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our model with the fixed effects appears to be a better fit for our data compared to the model with just random effects. We will use the “summary”" function to check the model parameters.

```
summary(model.full2)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: RT ~ Prime * Target + (1 + Prime | Participant) + (1 + Prime)
##    Data: data2_2_
##
```

```
## REML criterion at convergence: 20693.9

##

## Scaled residuals:

##     Min      1Q  Median      3Q     Max

## -1.9886 -0.6376 -0.2019  0.4141  6.8464

##

## Random effects:

##  Groups      Name        Variance Std.Dev. Corr

##  Participant (Intercept) 106997   327.1

##              Prime1       41934   204.8    0.20

##  Residual                666056   816.1

## Number of obs: 1271, groups:  Participant, 40

##

## Fixed effects:

##                Estimate Std. Error      df t value Pr(>|t|)

## (Intercept)     1596.15      56.56   38.94  28.220  < 2e-16 ***

## Prime1          -125.29      56.08   38.52  -2.234 0.031371 *

## Target1           61.37      45.79 1189.77   1.340 0.180452

## Prime1:Target1  -304.61      91.59 1189.64  -3.326 0.000909 ***

## ---

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##

## Correlation of Fixed Effects:
```

```
##             (Intr) Prime1 Targt1
## Prime1       0.105
## Target1      0.000 -0.001
## Prim1:Trgt1  0.000 -0.001  0.001
```

There appears to be a significant interaction between both conditions of Prime and Target. A further pairwise t-test will be conducted to investigate this further.

```
emmeans(model.full2, pairwise~Prime*Target, adjust="none")

## $emmeans
##  Prime    Target   emmean   SE   df lower.CL upper.CL
##  Negative Negative   1488 73.2 60.0     1342     1635
##  Positive Negative   1766 68.6 64.4     1629     1903
##  Negative Positive   1579 73.3 60.2     1432     1726
##  Positive Positive   1552 68.6 64.4     1415     1689
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $contrasts
##  contrast                              estimate   SE   df t.ratio p.value
##  Negative,Negative - Positive,Negative   -277.6 72.4  107 -3.835  0.0002
##  Negative,Negative - Negative,Positive    -90.9 64.8 1190 -1.403  0.1608
##  Negative,Negative - Positive,Positive    -63.9 72.4  107 -0.883  0.3792
##  Positive,Negative - Negative,Positive    186.7 72.4  107  2.577  0.0113
##  Positive,Negative - Positive,Positive   213.7 64.7 1189  3.301  0.0010
##  Negative,Positive - Positive,Positive     27.0 72.4  107  0.373  0.7099
```

The analysis for the second dataset was carried out using the lme4 package (Bates et al. 2015) to fix the linear mixed models for the reaction time measure of reading a sentence using R (R Development Core Team, 2017). Pairwise comparisons conducted with the emmeans package (Lenth, 2018) were used to investigate a significant interaction for the reaction time measure. Table 2 presents regression coefficients (*b*), standard errors, and *t*-values. Deviation coding was used for each of the two experimental factors (Barr et al. 2013). T-values equal to or greater than 1.96 indicated an effect that is significant relative to the .05 alpha level. As a significant interaction was found between the two conditions of Prime and Target a pairwise t-test was conducted to investigate this effect further.

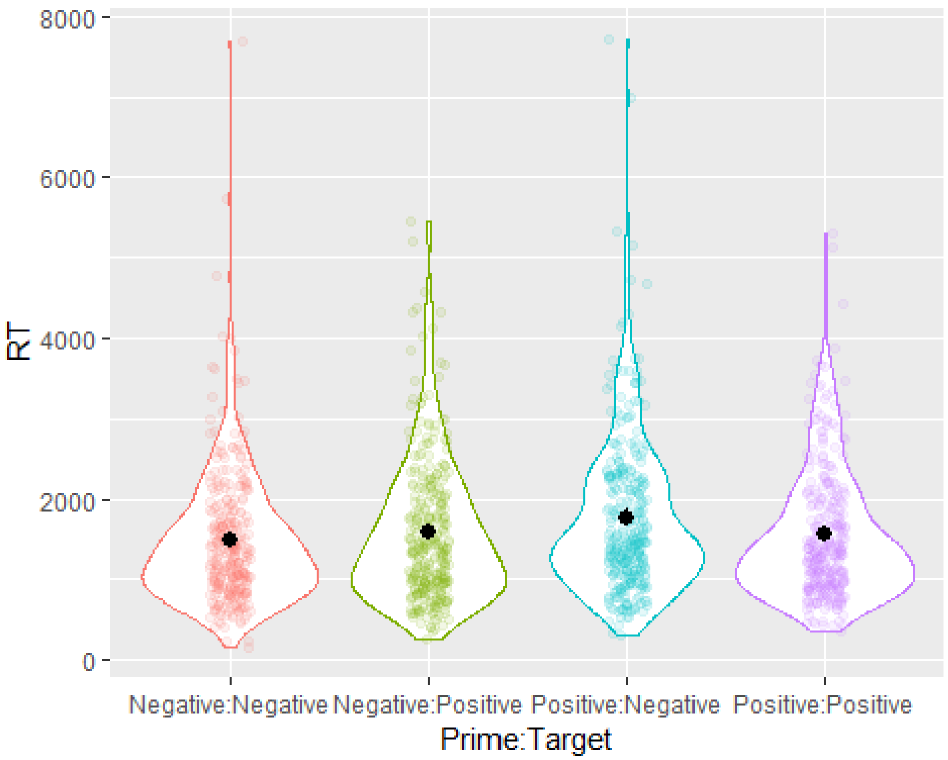
The t-test reveals two key comparisons. Firstly, that when the primed image was negative followed by a negative sentence (the target) there was no significant difference than if they had read a positive sentence (M = 1488 ms, SD = 73.2 vs. M = 1579 ms, SD = 73.3), t(1190) = -1.40, p =0.161. In comparison when the primed image was positive participant's read positive sentences significantly faster compared to negative sentences (M = 1552 ms, SD = 68.6 vs M = 1766 ms, SD = 68.6), t(1189) = 3.30, p = 0.001. The researcher's predictions were that congruent factors would produce faster reading times in comparison to factors with different prime and target emotions. However, this was only partially supported as participants do read a sentence describing an emotion more quickly after being presented with a priming image, but only if both factors are positive.

Table 2

	b	SE	t
Intercept	1596	56.56	28.22
Prime	-125.29	56.08	-2.23
Target	61.37	45.79	1.34
Prime * Target	-304.61	91.59	-3.33

The data is visualized with the following code using a violin plot.

```
ggplot(data2_2_, aes(x = Prime:Target, y = RT, colour = Prime:Target)) +
  geom_violin() +
  geom_jitter(width = .1, alpha = .1) +
  stat_summary(fun.data = "mean_cl_boot", colour="black") +
  guides(colour = FALSE)
```



Question 2 (part b)

In addition, we also measured whether people moved their eyes to re-look at the image when they had finished reading the sentence. We predicted people would do this more when the sentence and image emotions mismatched. The data file “data3.csv” contains these data, with a ‘1’ in the ‘Regress’ column corresponding to trials where people re-fixated on the image after reading the sentence, and a ‘0’ corresponding to trials where people did no re-fixate

Firstly, to analyse this dataset we must import the dataset.

```
data3_1_ <-
read_csv("C:/Users/omarf/OneDrive/Desktop/Mixed_Models_Ass/Ass_data/data3(1).csv")
```

We then turn our variables into factors with the follwing code.

```
data3_1_$Prime <- as.factor(data3_1_$Prime)

data3_1_$Target <- as.factor(data3_1_$Target)
```

Contrast weightings need to be implemented to our two factors.

```
contrasts(data3_1_$Prime) <- matrix(c(.5, -.5))

contrasts(data3_1_$Target) <- matrix(c(.5, -.5))
```

We then need to create a 2 x 2 repeated measures mixed linear model. We will be using the “glmer” function in R to analyse bimodial responses given by the participant. The different slopes and intercepts are added to the code below.

```
model_full <- glmer(Regress ~ Prime * Target + (1 + Prime * Target | Participant) + (1 + Prime * Target | Item), data = data3_1_, family = binomial)

## singular fit
```

An error of “singular fit” is recieved. We need to simplyfy the random effect structure. The following are attempts to simplyfy the code.

```
model_full <- glmer(Regress ~ Prime * Target + (1 + Prime + Target | Participant) + (1 + Prime + Target | Item), data = data3_1_, family = binomial)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0265976
## (tol = 0.001, component 1)

model_full <- glmer(Regress ~ Prime * Target + (1 + Prime + Target | Participant) + (1 + Prime | Item), data = data3_1_, family = binomial)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00553026
## (tol = 0.001, component 1)

model_full <- glmer(Regress ~ Prime * Target + (1 + Prime + Target | Participant) + (1 + Prime
), data = data3_1_, family = binomial)

## singular fit
```

The previous code failed to converge so we need to simplyfy the code futher. The following code allows for the model to be identified.

```
model_full_done <- glmer(Regress ~ Prime * Target + (1 + Prime  |Participant) + (1 + Prime ),
data = data3_1_, family = binomial)
```

The normality of the model will be tested using the Shapiro Wilko test of normality. The model appears to be normally distributed.

```
shapiro.test(residuals(model_full_done))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_full_done)
## W = 0.84533, p-value < 2.2e-16
```

In order to determine if our model is significant, we need to know if it differs from what we would expect if our condition factor did not influence reaction times. We thus need to create a null model by removing both conditions as predictors.

```
model_null_1 <- glmer(Regress ~  (1 + Prime | Participant) + (1 + Prime), data = data3_1_,
family = binomial)
```

We then run an ANOVA to determine if there is a significant difference between the null model and the experimental model.

```
anova(model_full_done, model_null_1)

## Data: data3_1_
## Models:
## model_null_1: Regress ~ (1 + Prime | Participant) + (1 + Prime)
## model_full_done: Regress ~ Prime * Target + (1 + Prime | Participant) + (1 + Prime)
##                 Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## model_null_1     5 1533.2 1559.0 -761.61   1523.2
## model_full_done  7 1526.3 1562.4 -756.17   1512.3 10.887      2   0.004324
##
## model_null_1
## model_full_done **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our model with the fixed effects appears to be a better fit for our data compared to the model with just random effects. We will use the “summary”" function to check the model parameters.

```
summary(model_full_done)
```

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

##   Approximation) [glmerMod]

##  Family: binomial  ( logit )

## Formula:

## Regress ~ Prime * Target + (1 + Prime | Participant) + (1 + Prime)

##    Data: data3_1_

##

##      AIC      BIC   logLik deviance df.resid

##   1526.3   1562.4   -756.2   1512.3     1272

##

## Scaled residuals:

##     Min      1Q  Median      3Q     Max

## -1.9406 -0.7078 -0.4017  0.9328  3.4158

##

## Random effects:

##  Groups      Name        Variance Std.Dev. Corr

##  Participant (Intercept) 0.96468  0.9822

##              Prime1      0.01489  0.1220   0.48

## Number of obs: 1279, groups:  Participant, 40

##

## Fixed effects:

##                Estimate Std. Error z value Pr(>|z|)

## (Intercept)     -0.7808     0.1705  -4.580 4.65e-06 ***
## Prime1          -0.4670     0.1381  -3.382  0.00072 ***
## Target1          0.3159     0.1279   2.470  0.01351 *
## Prime1:Target1  -0.5285     0.2558  -2.067  0.03877 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Prime1 Targt1
## Prime1       0.104
## Target1     -0.014  0.025
## Prim1:Trgt1  0.010 -0.036  0.041
```

There appears to be a significant interaction between both conditions of Prime and Target. A further pairwise t-test will be conducted to investigate this further.

```
emmeans(model_full_done, pairwise ~ Prime * Target, adjust = "none", type = "response")
```

```
## $emmeans
##  Prime    Target    prob     SE  df asymp.LCL asymp.UCL
##  Negative Negative 0.271 0.0417 Inf     0.197     0.360
##  Positive Negative 0.436 0.0480 Inf     0.345     0.531
##  Negative Positive 0.261 0.0409 Inf     0.189     0.349
##  Positive Positive 0.302 0.0423 Inf     0.226     0.391
##
## Confidence level used: 0.95
## Intervals are back-transformed from the logit scale
##
## $contrasts
##  contrast                              odds.ratio    SE  df z.ratio
##  Negative,Negative / Positive,Negative      0.481 0.089 Inf -3.957
##  Negative,Negative / Negative,Positive      1.053 0.194 Inf  0.280
##  Negative,Negative / Positive,Positive      0.860 0.164 Inf -0.793
##  Positive,Negative / Negative,Positive      2.188 0.407 Inf  4.212
##  Positive,Negative / Positive,Positive      1.786 0.316 Inf  3.276
##  Negative,Positive / Positive,Positive      0.816 0.156 Inf -1.059
##  p.value
##  0.0001
##  0.7798
##  0.4277
##  <.0001
##  0.0011
##  0.2898
##
## Tests are performed on the log odds ratio scale
```

The researcher’s hypothesis that participants would regress more when the prime and target conditions mismatched was only partially supported. It was found that in mismatched conditions where the prime condition was positive, but the target sentence was negative led participants to regress on the image significantly more in comparison to conditions where both conditions were positive (M = 0.44, SE = 0.05   vs. M = 0.302, SD = 0.04, p = 0.001). This was true for when both conditions were negative (M = 0.271, SE = 0.04, p < 0.001). However, when the prime was negative, and the target was positive (M = 0.26, SE = 0.04) there was no significant difference when comparing to conditions with matching positive (p = 0.290) or matching negative factors (p = 0.780). There was also a significant difference between when participants partook in the positive prime and negative target condition in comparison to the negative prime and positive target condition (p < 0.001). The pairwise analysis suggests that participants regress more compared to congruent factors, but only if the mismatch is a positive prime image with a negative sentence. The same effect is not found with a negative prime and positive sentence. It can also be shown that participants are more likely to regress on the sentence if presented with a positive prime and a negative sentence to read in comparison to any other condition.

Table 3

	b	SE	z
Intercept	-0.78	0.17	-4.58
Prime	-0.47	0.14	-3.38
Target	0.32	0.13	2.47
Prime * Target	-0.53	0.26	-2.07

To plot the bar chart to represent this data firstly you must remove the missing value from the dataset. The bar chart will recieve an error if the missing data is not removed when performing ggplot.

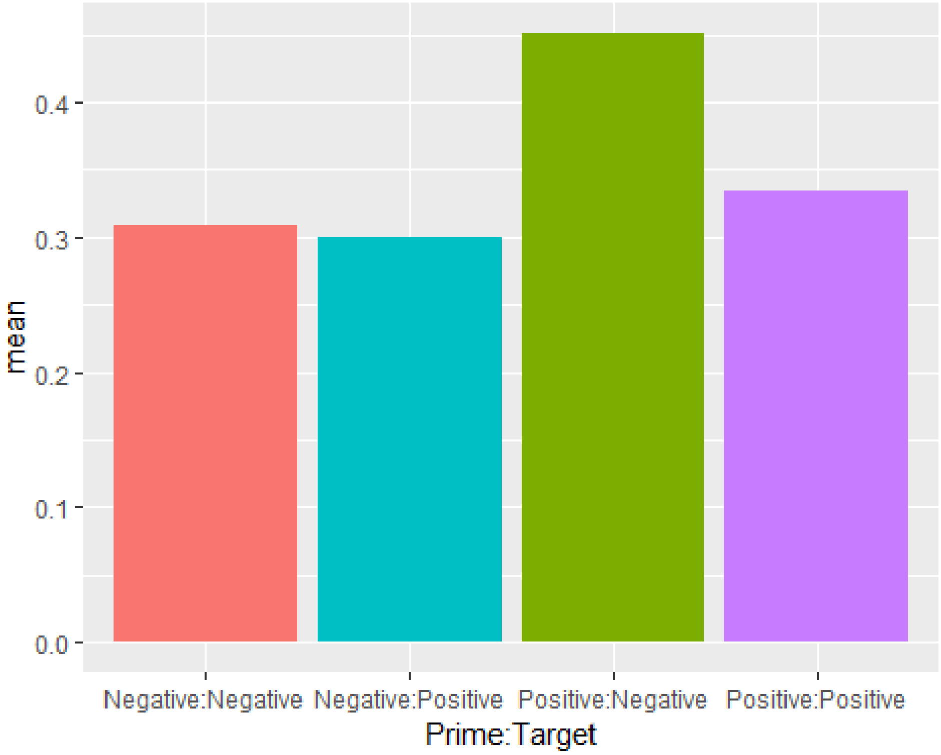
```
test <- data3_1_[-c(1265),]
```

We then need to calculate the aggregate of the bomodial data.

```
data_agg <- test %>% group_by(Prime, Target) %>% summarise(mean=mean(Regress),
sd=sd(Regress))
```

We then use ggplot to plot the dataset.

```
ggplot(data_agg, aes(x = Prime:Target, y = mean, fill = Target:Prime)) + geom_col() +
guides(fill = FALSE)
```



References

Douglas Bates, Martin Maechler, Ben Bolker, Steve Walker (2015). Fitting Linear Mixed-Effects Models Using lme4.

Journal of Statistical Software, 67(1), 1-48. doi:10.18637/jss.v067.i01.

Russell Lenth (2019). emmeans: Estimated Marginal Means, aka Least-Squares Means. R package version 1.3.2.

https://CRAN.R-project.org/package=emmeans

Barr, D.J., Levy, R., Scheepers, C., & Tily, H.J. (2013, April). Random effects structure for confirmatory hypothesis testing: Keep it maximal. Retrieved from *https://www.ncbi.nlm.nih.gov/pubmed/24403724*