Coursework assignment A - 2022-2023

CS4125 Seminar Research Methodology for Data Science

Andrej Tibenský (5882133), Rahul Kochar (4812522), Varun Singh (5441935)

20/06/2023

Contents

1 Part 1 - Design and set-up of true experiment

1.1 The motivation for the planned research

(Max 200 words)

Researchers found that Nicotine caused weight loss in rats and then later in humans. Morean and Wedel noticed that 13.5% of adults in USA vape to lose weight.

1.2 The theory underlying the research

(Max 200 words) Preferable based on theories reported in literature

There are a few theories for how Nicotine reduces weight through Leptin, fight-or-flight receptors but also indications that weight should increase through Ghrelin. However, all of these sources note that there is empirical evidence showing strong correlation between Nicotine consumption and weight loss. We will not go into the different theories in Biology for how this happens. Instead, we noticed that these papers did not always look at how the Nicotine was consumed i.e. through vaping, patches, injection, etc.

1.3 Research questions

The research question that will be examined in an experiment (or alternatively the hypothesis that will be tested in an experiment)

Research Question: Does consuming Nicotine through vape cause more weight loss than patch?

1.4 The related conceptual model

This model should include:

Independent variable(s): Nicotine consumed each day in gram Nicotine will be carefully measured and administered to the participant

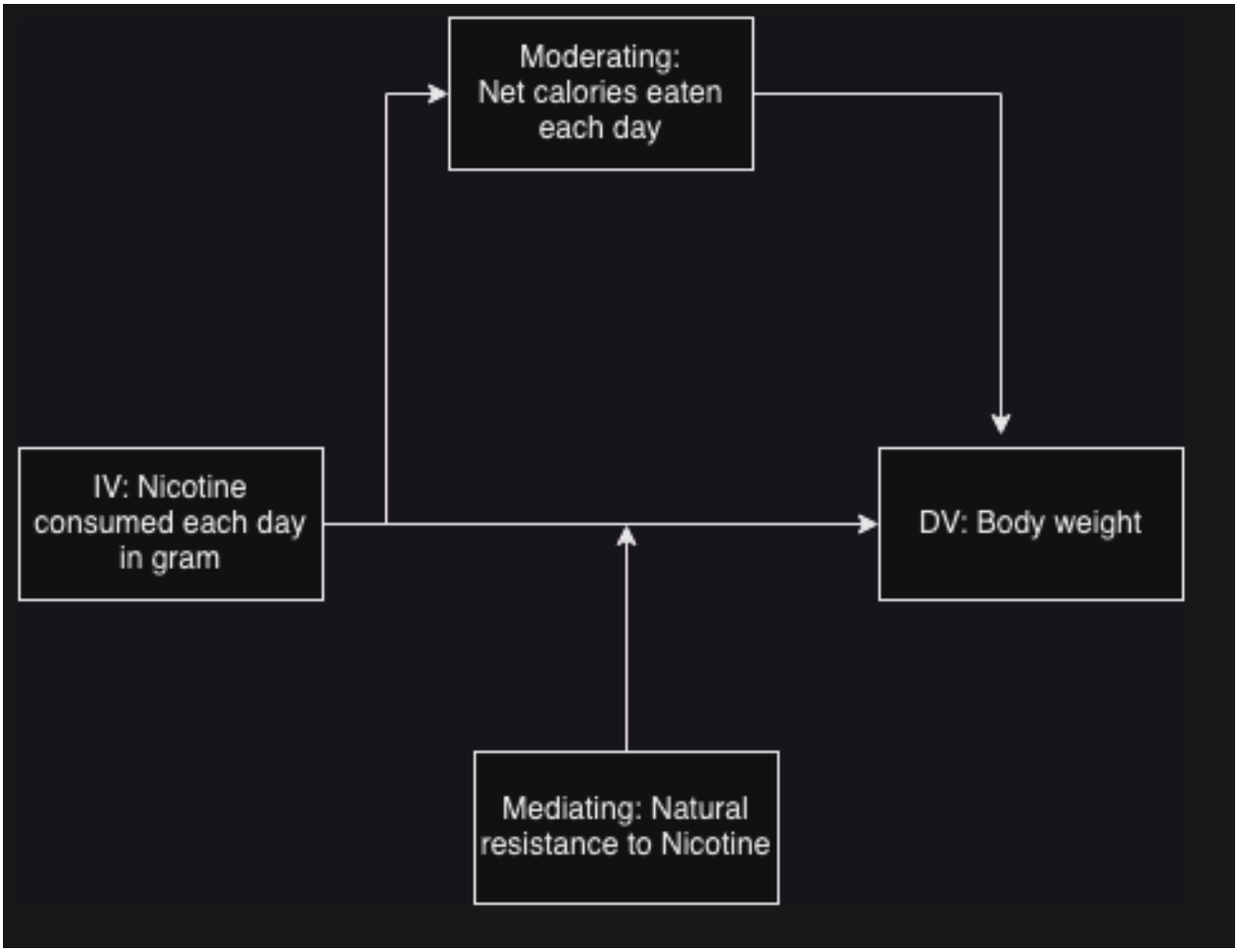
Dependent variable: Body weight The body weight of the patient will change and needs to be measured.

Moderating variable (at least 1): Resistance to Nicotine by body Some people are naturally more resistant to Nicotine than others. To achieve the same effect, different people can require different amounts of Nicotine.

Mediating variable (at least 1): Net calories eaten each day The net intake of calories (calories eaten - calories used from exercise and other activities) will increase, decrease or will not change body weight.

There are some other variables in the real world that complicate such an experiment. We will ignore diabetes, age, eating restrictions and other conditions like paralysis.

```
knitr::include_graphics("conceptualmodelq1.png")
```



1.5 Experimental Design

Experimental Design (the study should have a true experimental design to test a single hypothesis that, for simplicity, includes only independent variable(s) and dependent variable(s). In other words, mediating and moderating variables are not included in the experimental design)

The experiment is to determine if method of consuming Nicotine affects weight loss. All participants will consume nicotine in one of three ways and data collected will be analyzed to check for statistical significance.

1.6 Experimental procedure

Describe how the experiment will be executed step by step

Use "post-test-only two treatment comparison" - Measure weight of all participants (at a specific time of day such as before lunch at 12:00) - Start treatment by assigning users randomly to patch or vaping group with equal probability. All the groups will start consuming Nicotine in linearly increasing amounts. The amount will increase at the same rate for everybody till a pre-determined level through their respective sources of patch or vape. (between subject design). - The nicotine consumed for all participants on any day will always be the same. - After Nicotine consumption has reached pre-determined target level, stop. - Measure weight of all participants once again in the same situation as before in point 1 - Possible values can be that each participant initially consumes 5mg of Nicotine each day. Every week the amount is increased by 2mg till 15mg/day is reached. - To address history threats, all participants should only perform the same amount of exercise (so that the weight loss can only be from Nicotine). We are already monitoring calories eaten so that is accounted for. - To address mortality threat, recruit extra participants to the study.

1.7 Measures

Describe the measure that will be used

The delta in body weight (Weight at end of experiment - weight at start of experiment). The dataset will be two array of floats (one array for delta in weight for vaping and other for delta in weight in patch).

1.8 Participants

Describe which participants will recruit in the study and how they will be recruited

Users that do not consume Nicotine are ideal for this experiment because users that consume Nicotine can have developed resistance to Nicotine. They can be recruited through social media and flyers in different parts of the country and cities to attract a diverse group of participants. StackOverflow says says t-test is designed for small sample sizes and there is no minimum size. Intuitively, a few hundred participants should be sufficient, more are welcome.

1.9 Suggested statistical analyses

Describe the statistical test you suggest to care out on the collected data.

t-test: The standard textbook t-test will be applied to determine if the results are statistically significant. t-test is selected because we want to compare two populations to test a hypothesis. If the number of participants is sufficiently large i.e. it can be turned into a standard normal distribution, z-test can be used.

2 Part 2 - Generalized linear models

2.1 Question 1 Twitter sentiment analysis (Between groups - single factor)

2.1.1 Conceptual model

Make a conceptual model for the following research question: Is there a difference in the sentiment of the tweets related to the different individuals/organisations?

```
knitr::include_graphics("conceptualModelP1.png")
```

2.1.2 Model description

Describe the mathematical model fitted on the most extensive model. (hint, look at the mark down file of the lectures to see example on formulate mathematical models in markdown). Assume a Gaussian distribution for the tweet's sentiments rating. Justify the priors.

We will use the following linear model:

$score \sim Norm(\mu, \sigma)$ [likelihood]

$\mu = a + b * Candidate$ linear model

$a \sim Norm(100, 25)$ [a Intercept]

$b \sim Norm(0, 1)$ [b prior]

2.1.3 Generate Synthetic data

Create a synthetic data set with a clear difference between tweets' sentiments of celebrities for verifying your analysis later on. Report the values of the coefficients of the linear model used to generate synthetic data. (hint, look at class lecture slides of lecture on Generalized linear models for example to create synthetic data)

The mean and SD for each candidate are: Trump = (-2, 1.5), Bernie = (3, 1) and Hillary = (0, 2)

```
#include your code for generating the synthetic data
synthetic_data <- data.frame(Candidate=character(), score=integer(),
                        stringsAsFactors = TRUE)

n_c <- 3
n_p <- 100
B0 <- 0

Trump <- rnorm(n_p, -2, 1.5)
Hillary <- rnorm(n_p, 0, 2)
Bernie <- rnorm(n_p, 3, 1)

labels=c("Donald Trump", "Hillary Clinton", "Bernie Sanders")

for (i in 1:n_p) {
  l_m <- B0 + Trump[i]
  newrow <- data.frame(Candidate=labels[1], score=round(l_m, 0))
  synthetic_data <- rbind(synthetic_data,newrow)
}
for (i in 1:n_p) {
  l_m <- B0 + Hillary[i]
  newrow <- data.frame(Candidate=labels[2], score=round(l_m, 0))
  synthetic_data <- rbind(synthetic_data,newrow)
}
for (i in 1:n_p) {
  l_m <- B0 + Bernie[i]
  newrow <- data.frame(Candidate=labels[3], score=round(l_m, 0))
  synthetic_data <- rbind(synthetic_data,newrow)
}
```

2.1.4 Collecting tweets, and data preparation

Include the annotated R script (excluding your personal Keys and Access Tokens information), but put echo=FALSE, so code is not included in the output pdf file.

```
#during writing you could add "eval = FALSE",  kntr will than not run this code chunk (take some time d

#setwd("C:\Users\Andrej\Documents\School\Q4\SeminarResearchMethod")
# apple , note use / instead of \, which used by windows
#devtools::install_github("rmcelreath/rethinking")
library(rethinking)
```

```
library(BayesianFirstAid)
library(pander)
#install.packages("twitteR", dependencies = TRUE)
library(twitteR)
#install.packages("RCurl", dependencies = T)
library(RCurl)
#install.packages("bitops", dependencies = T)
library(bitops)
#install.packages("plyr", dependencies = T)
library(plyr)
#install.packages('stringr', dependencies = T)
library(stringr)
#install.packages("NLP", dependencies = T)
library(NLP)
#install.packages("tm", dependencies = T)
library(tm)
#install.packages("wordcloud", dependencies=T)
#install.packages("RColorBrewer", dependencies=TRUE)
library(RColorBrewer)
library(wordcloud)
#install.packages("reshape", dependencies=T)
library(reshape)
library(stringi)
library(AICcmodavg) #aictab
```

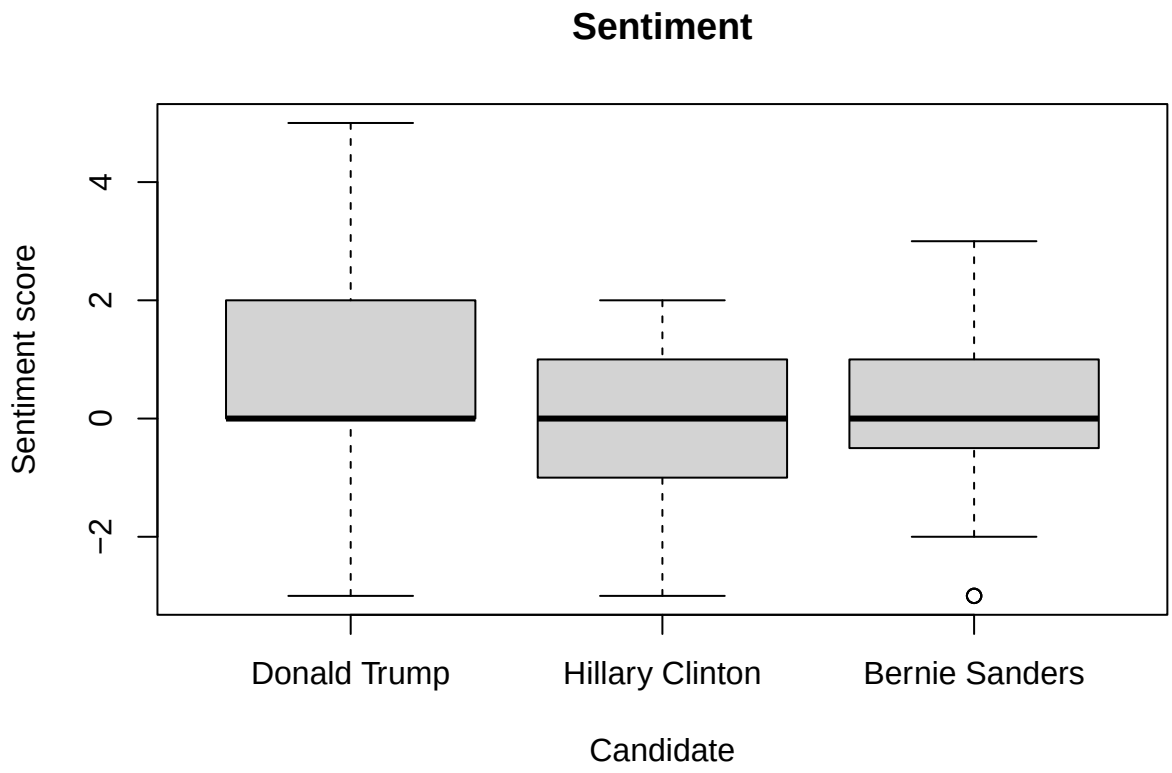
```
#################### functions
#setwd("C:\\Users\\Andrej\\Documents\\School\\Q4/SeminarResearchMethod\\")
#Lec4a <- read.spss("examples_Chi.sav", use.value.labels=TRUE, to.data.frame=TRUE)
#tweets_T <- read.delim("tweets_T.txt", sep = "")

tweet_T_string = readLines("tweets_T.txt")[nchar(readLines("tweets_T.txt")) > 7]
tweets_T_cols = stri_split_fixed(str = tweet_T_string, pattern=":", n=2)
tweets_T_cols = do.call(rbind, tweets_T_cols)
tweets_T = data.frame(tweets_T_cols)

tweet_C_string = readLines("tweets_C.txt")[nchar(readLines("tweets_C.txt")) > 7]
tweets_C_cols = stri_split_fixed(str = tweet_C_string, pattern=":", n=2)
tweets_C_cols = do.call(rbind, tweets_C_cols)
tweets_C = data.frame(tweets_C_cols)

tweet_B_string = readLines("tweets_B.txt")[nchar(readLines("tweets_B.txt")) > 7]
tweets_B_cols = stri_split_fixed(str = tweet_B_string, pattern=":", n=2)
tweets_B_cols = do.call(rbind, tweets_B_cols)
tweets_B = data.frame(tweets_B_cols)


#taken from https://github.com/mjhea0/twitter-sentiment-analysis
pos <- scan('positive-words.txt', what = 'character', comment.char=';') #read the positive words
neg <- scan('negative-words.txt', what = 'character', comment.char=';') #read the negative words

source("sentiment3.R") #load algorithm
# see sentiment3.R form more information about sentiment analysis. It assigns a intereger score
# by subtracting the number of occurrence of negative words from that of positive words
```

```
analysis_T <- score.sentiment(tweets_T$X2, pos, neg)
analysis_C <- score.sentiment(tweets_C$X2, pos, neg)
analysis_B <- score.sentiment(tweets_B$X2, pos, neg)


sem<-data.frame(analysis_T$score, analysis_C$score, analysis_B$score)


semFrame <-melt(sem, measured=c(analysis_T.score,analysis_C.score, analysis_B.score ))
names(semFrame) <- c("Candidate", "score")
semFrame$Candidate <-factor(semFrame$Candidate, labels=c("Donald Trump", "Hillary Clinton", "Bernie San

#The data you need for the analyses can be found in semFrame
```

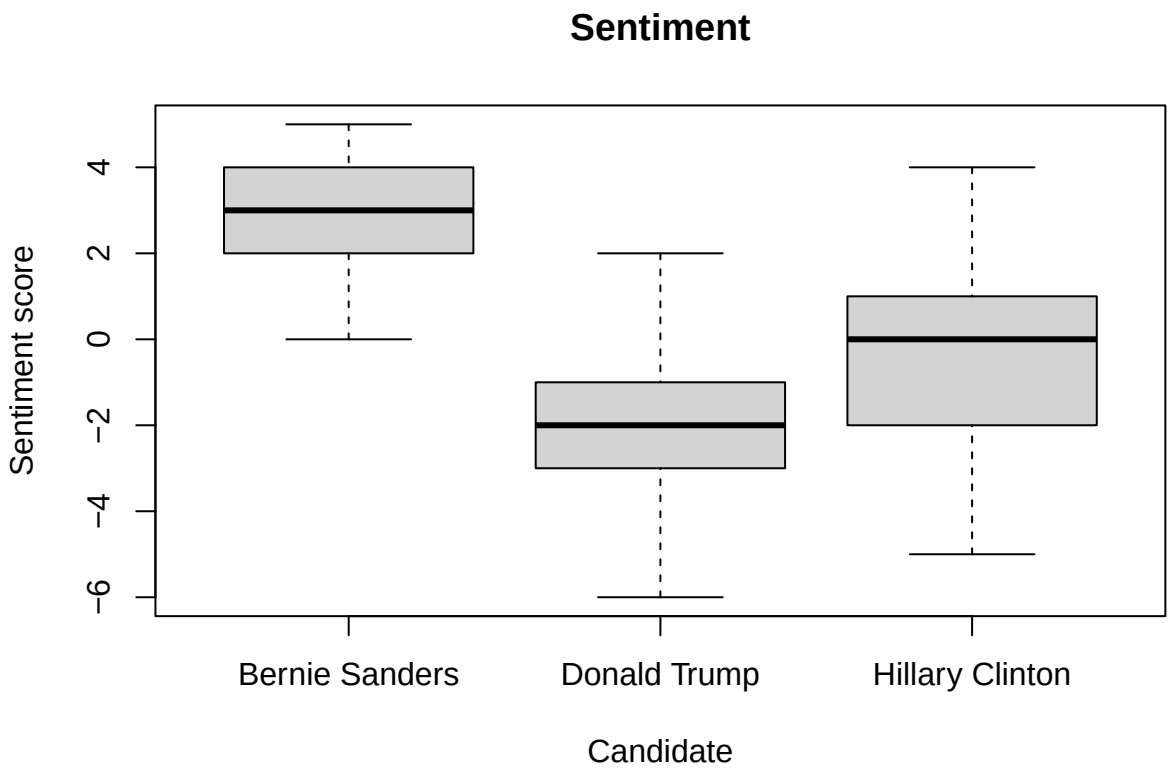
2.1.5 Visual inspection Mean and distribution sentiments

Graphically examine the mean and distribution sentiments of tweets for each individual/organisation, and provide interpretation

```
#include your analysis code and output in the document
boxplot(score ~ Candidate, data=semFrame, main="Sentiment", xlab="Candidate", ylab="Sentiment score")
```




```
boxplot(score ~ Candidate, data=synthetic_data, main="Sentiment", xlab="Candidate", ylab="Sentiment sco
```



We can see that the mean for each candidate is very close to 0, however the spread of the sentiment scores is wider for Donald Trump and more narrow for Hillary and Bernie. The top 25% is also highest for Trump while the low 25% is lowest for Hillary.

2.1.6 Frequentist approach

2.1.6.1 Analysis verification Verify your model analysis with synthetic data and show that it can reproduce the coefficients of the linear model that you used to generate the synthetic data set. Provide a short interpretation of the results, with a reflection of AICc, F-value, p-value etc.

```
#include your analysis code of synthetic data and output in the document synth_data
baseModel <- lm(score ~ 1, data=synthetic_data)
candidateModel <- lm(score ~ Candidate, data=synthetic_data)
aictab(cand.set = list(baseModel, candidateModel), modnames=c("base", "candidate"))
```

```
##
## Model selection based on AICc:
##
##           K    AICc Delta_AICc AICcWt Cum.Wt      LL
## candidate 4 1126.55       0.00      1      1 -559.21
## base      2 1426.00     299.45      0      1 -710.98
```

```
anova(baseModel, candidateModel, test = "F")
```

```
## Analysis of Variance Table
##
## Model 1: score ~ 1
## Model 2: score ~ Candidate
```

```
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    299 2009.75
## 2    297  730.66  2    1279.1 259.96 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
manova <- anova(candidateModel)
manova
```

```
## Analysis of Variance Table
##
## Response: score
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Candidate   2 1279.09  639.54  259.96 < 2.2e-16 ***
## Residuals 297  730.66    2.46
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pander(summary(candidateModel))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.98	0.1568	19	3.125e-53
CandidateDonald Trump	-4.99	0.2218	-22.5	4.101e-66
CandidateHillary Clinton	-3.21	0.2218	-14.47	2.749e-36

Table 2: Fitting linear model: score ~ Candidate

Observations	Residual Std. Error	R^2	Adjusted R^2
300	1.568	0.6364	0.634

We can see that our analysis is able to reproduce the means of our synthetic dataset for each candidate. We conducted a Linear Model analysis to test the sentiment score. The results found a significant effect ($F(2,297) = 259.96$, $5.5621322 \times 10^{-66}$) for the candidates on the sentiment score. The model with the candidates also has the best goodness-of-fit as it has the smaller AICc value.

2.1.6.2 Linear model Redo the analysis now on the real tweet data set. Provide a short interpretation of the results, with an interpretation of AICc, F-value, p-value, etc.

The results found a significant effect($F(2, 297) = 14.637$, $p < 0.001$) for the candidates on the sentiment score. The model with the candidates also has the best goodness-of-fit as it has the smaller AICc value.

```
#include your analysis code and output in the document
baseModel <- lm(score ~ 1, data=semFrame)
candidateModel <- lm(score ~ Candidate, data=semFrame)
aictab(cand.set = list(baseModel, candidateModel), modnames=c("base", "candidate"))
```

```
##
## Model selection based on AICc:
##
##           K    AICc Delta_AICc AICcWt Cum.Wt      LL
## candidate 4 1118.56       0.00      1      1 -555.21
## base      2 1142.67      24.11      0      1 -569.31
```

```
anova(baseModel, candidateModel, test = "F")
```

```
## Analysis of Variance Table
##
## Model 1: score ~ 1
## Model 2: score ~ Candidate
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    299 781.59
## 2    297 711.46  2    70.127 14.637 8.654e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pander(anova(candidateModel))
```

Table 3: Analysis of Variance Table

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Candidate	2	70.13	35.06	14.64	8.654e-07
Residuals	297	711.5	2.395	NA	NA

```
#aictab(cand.set = c(baseModel, candidateModel), modnames=c("baseModel", "candidateModel"))
```

2.1.6.3 Post Hoc analysis If a model that includes the individual better explains the sentiments of tweets than a model without such predictor, conduct a posthoc analysis with, e.g., Bonferroni correction to examine which celebrity tweets differ from the other individual's tweets. Provide a brief interpretation of the results.

We conducted a pairwise t test and a bonferroni adjustment that both showed a significant difference between Donald Trump and the other candidates ($p < 0.001$) but failed to show a significant difference between Hillary and Bernie ($p = 0.718$).

```
#include your code and output in the document
pairwise.t.test(semFrame$score, semFrame$Candidate,
                paired = FALSE, p.adjust.method = "bonferroni")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  semFrame$score and semFrame$Candidate
##
##                 Donald Trump Hillary Clinton
## Hillary Clinton 2.6e-06      -
## Bernie Sanders  8.6e-05      1
##
## P value adjustment method: bonferroni
```

```
aov_model <- aov(score ~ Candidate, data = semFrame)
TukeyHSD(aov_model)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = score ~ Candidate, data = semFrame)
##
## $Candidate
```

```
##                                 diff        lwr        upr     p adj
## Hillary Clinton-Donald Trump   -1.10 -1.6155843 -0.5844157 0.0000026
## Bernie Sanders-Donald Trump    -0.93 -1.4455843 -0.4144157 0.0000852
## Bernie Sanders-Hillary Clinton  0.17 -0.3455843  0.6855843 0.7176469
```

2.1.6.4 Report section for a scientific publication Write a small section for a scientific publication (journal or a conference), in which you report the results of the analyses, and explain the conclusions that can be drawn in a format commonly used by the scientific community Look at Brightspace for examples papers and guidelines on how to do this. (Hint, there are strict guidelines for reporting statistical results in paper, I expect you to follow these here)

A linear model was fitted on the sentiment score for tweets about Donald Trump, Hillary Clinton and Bernie Sanders, taking the candidate as an independent variable. The analysis found a significant effect($F(2, 297) = 14.637$, $p < 0.001$) for the candidates on the sentiment score. The model with the candidates also has the best goodness-of-fit as it has the smaller AICc value ($1118.56 < 1142.67$). We also conducted a pairwise t test and a bonferroni adjustment that both showed a significant difference between the sentiment score for Donald Trump and the other candidates ($p < 0.001$) but failed to show a significant difference between the sentiment scores of Hillary Clinton and Bernie Sanders ($p = 0.718$).

2.1.7 Bayesian Approach

For the Bayesian analyses, use the rethinking and/or BayesianFirstAid library #### Analysis verification Verify your model analysis with synthetic data and show that it can reproduce the coefficients of the linear model that you used to generate the synthetic data set. Provide a short interpretation of the results, with a reflection of WAIC, and 95% credibility interval of coefficients for individual celebrities.

```
synthetic_data$Candidate_Nominal <-factor(synthetic_data$Candidate, labels=c("Donald Trump", "Hillary C
```

```
m0F <-ulam(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a,
    a ~ dnorm(100, 25),
    sigma ~ dunif(0.1, 2)
  ), data = synthetic_data ,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m1F <-ulam(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a + b*Candidate_Nominal,
    a ~ dnorm(100, 25),
    b ~ dnorm(0, 1),
    sigma ~ dunif(0.1, 2)
  ), data = synthetic_data ,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)
```

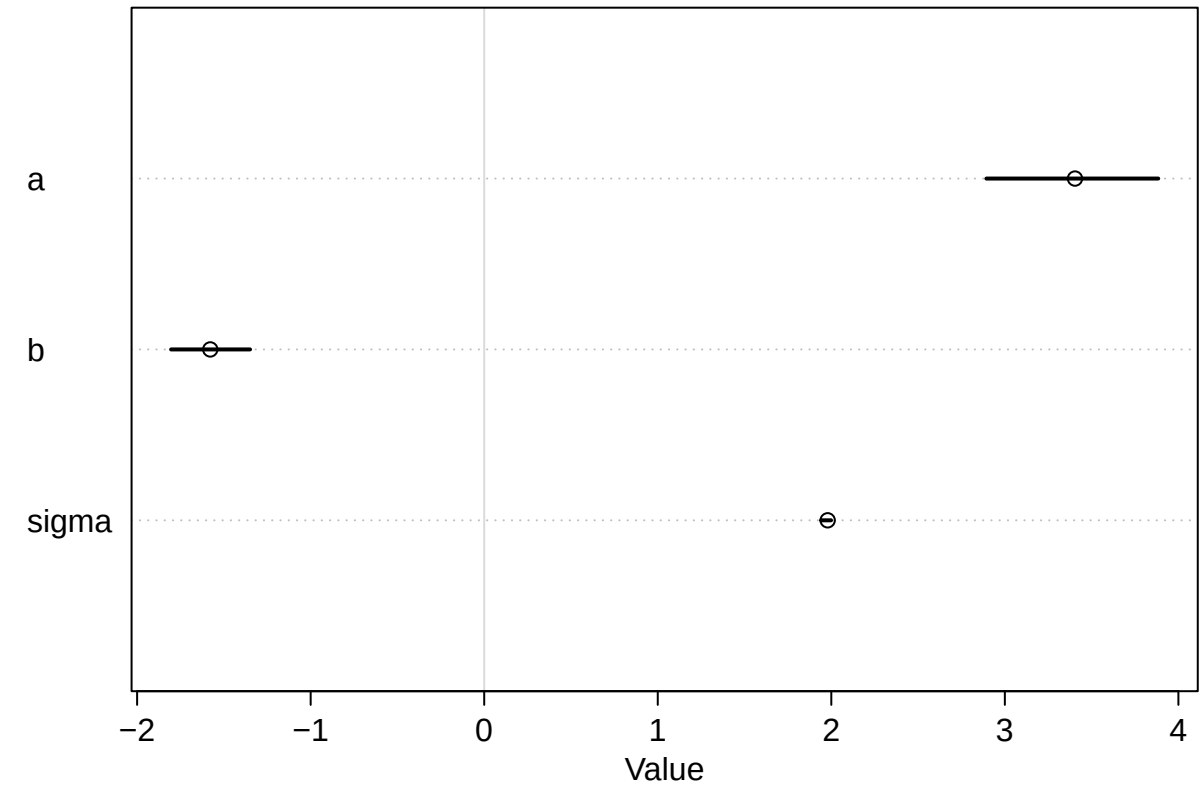
```
compare(m0F,m1F) #WAIC - m1F is best out-of-sample fit because of the smallest WAIC value
```

```
##         WAIC       SE    dWAIC    dSE    pWAIC       weight
## m1F 1347.103 27.33299   0.0000     NA 2.363112 1.000000e+00
## m0F 1474.132 28.26513 127.0288 22.604 1.521208 2.606459e-28
```

```
precis(m1F, prob=0.95) #95%  credibility interval of coefficients
```

```
##             mean         sd      2.5%     97.5%    n_eff    Rhat4
## a      3.404201 0.31701150  2.780352  3.984202 510.1630 1.009148
## b     -1.578508 0.14429069 -1.851579 -1.299368 497.9587 1.010785
## sigma  1.978996 0.01932915  1.927650  1.999510 943.4713 1.005460
```

```
plot(m1F) #
```



The WAIC score for model m1F is lower than m0F indicating a better fit which shows that having the candidate variable positively affects the linear model. Coefficients a and b cannot be null.

2.1.7.1 Model comparison Redo the analysis on the actual tweet data set. Provide a short interpretation of the results, with a reflection of WAIC, and 95% credibility interval of coefficients for individual celebrities.

```
m0 <-ulam(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a,
    a ~ dnorm(100, 25),
    sigma ~ dunif(0.1, 2)
  ), data = semFrame ,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m1 <-ulam(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a + b*Candidate,
    a ~ dnorm(100, 25),
```

```
    b ~ dnorm(0, 1),
    sigma ~ dunif(0.1, 2)
  ), data = semFrame ,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)
```

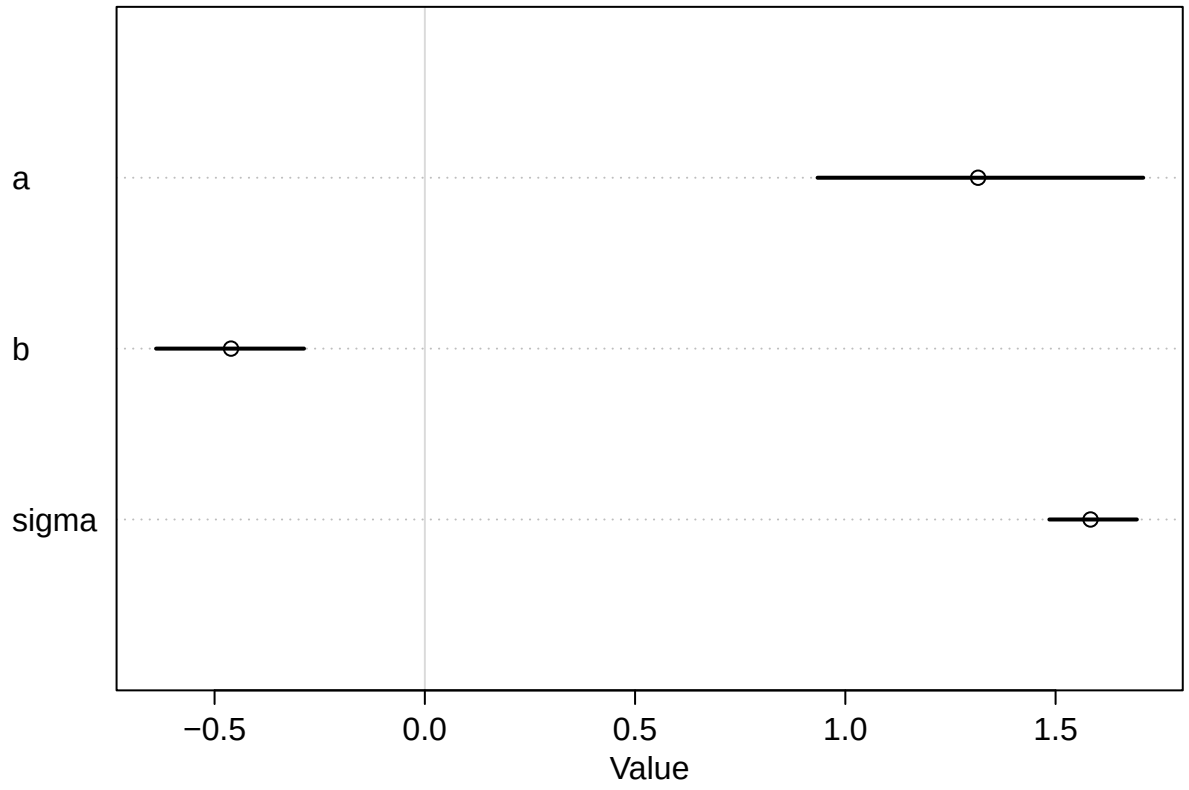
```
compare(m0,m1) #WAIC - m1 is best out-of-sample fit because of the smallest WAIC value
```

```
##        WAIC       SE    dWAIC      dSE    pWAIC       weight
## m1 1128.165 29.48792  0.00000       NA 3.603225 0.9995146437
## m0 1143.425 33.52355 15.26028 7.950135 2.761813 0.0004853563
```

```
precis(m1, prob=0.95) #95%  credibility interval of coefficients
```

```
##             mean         sd       2.5%      97.5%    n_eff    Rhat4
## a      1.3156884 0.24150043  0.8491853  1.8147407 821.1363 1.002053
## b     -0.4608234 0.11085621 -0.6815166 -0.2467183 787.9380 1.001558
## sigma  1.5830883 0.06473288  1.4667423  1.7217318 939.3164 1.005253
```

```
plot(m1) #
```



The WAIC score for model m1 is lower than m0 indicating a better fit which shows that having the candidate variable positively affects the linear model. Coefficients a and b cannot be null.

2.1.7.2 Comparison individual/organisation pair

Compare sentiments of individual pairs and provide a brief interpretation (e.g. CIs)

```
semFrame_copy = semFrame
```

```r
# exlucde all rows with candidate = "Donald Trump"
semFrame1 <- semFrame[semFrame$Candidate != "Donald Trump",]
semFrame1$Candidate <- factor(semFrame1$Candidate, levels = c("Hillary Clinton", "Bernie Sanders"))

semFrame2 <- semFrame[semFrame$Candidate != "Hillary Clinton",]
semFrame2$Candidate <- factor(semFrame2$Candidate, levels = c("Donald Trump", "Bernie Sanders"))

semFrame3 <- semFrame[semFrame$Candidate != "Bernie Sanders",]
semFrame3$Candidate <- factor(semFrame3$Candidate, levels = c("Hillary Clinton", "Donald Trump"))

model1 <- bayes.t.test(score ~ Candidate, data = semFrame1)
model2 <- bayes.t.test(score ~ Candidate, data = semFrame2)
model3 <- bayes.t.test(score ~ Candidate, data = semFrame3)
```

```r
show(model1)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - two sample
##
## data: group Hillary Clinton (n = 100) and group Bernie Sanders (n = 100)
##
##   Estimates [95% credible interval]
## mean of group Hillary Clinton: -0.031 [-0.26, 0.19]
## mean of group Bernie Sanders: 0.16 [-0.080, 0.40]
## difference of the means: -0.19 [-0.51, 0.14]
## sd of group Hillary Clinton: 1.1 [0.92, 1.3]
## sd of group Bernie Sanders: 1.2 [0.99, 1.4]
##
## The difference of the means is greater than 0 by a probability of 0.129
## and less than 0 by a probability of 0.871
```

```r
show(model2)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - two sample
##
## data: group Donald Trump (n = 100) and group Bernie Sanders (n = 100)
##
##   Estimates [95% credible interval]
## mean of group Donald Trump: 1.0 [0.57, 1.4]
## mean of group Bernie Sanders: 0.15 [-0.083, 0.40]
## difference of the means: 0.85 [0.36, 1.4]
## sd of group Donald Trump: 2.1 [1.8, 2.4]
## sd of group Bernie Sanders: 1.2 [0.98, 1.4]
##
## The difference of the means is greater than 0 by a probability of >0.999
## and less than 0 by a probability of <0.001
```

```r
show(model3)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - two sample
##
## data: group Hillary Clinton (n = 100) and group Donald Trump (n = 100)
##
##   Estimates [95% credible interval]
```

```
## mean of group Hillary Clinton: -0.032 [-0.25, 0.19]
## mean of group Donald Trump: 1.0 [0.58, 1.5]
## difference of the means: -1 [-1.5, -0.56]
## sd of group Hillary Clinton: 1.1 [0.93, 1.3]
## sd of group Donald Trump: 2.1 [1.8, 2.4]
##
## The difference of the means is greater than 0 by a probability of <0.001
## and less than 0 by a probability of >0.999
```

The Bayesian estimation comparison suggests that Bernie Sanders has a higher mean sentiment score (0.16) than Hillary Clinton (-0.032) with a high probability of 0.87. The estimated difference in their mean sentiments is -0.19.

When comparing Donald Trump and Bernie Sanders, there is a very high probability (greater than 0.999) that Trump has a higher mean sentiment score (1.0) than Sanders (0.15). The estimated difference in their mean sentiments is 0.85.

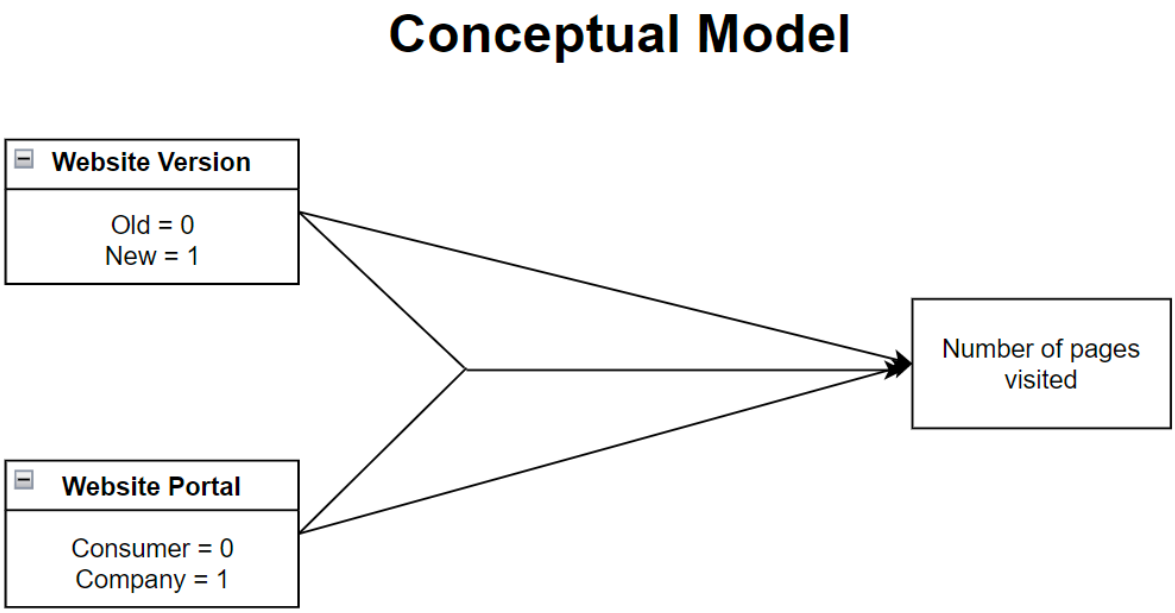
Similarly, when comparing Hillary Clinton and Donald Trump, there is a very high probability (greater than 0.999) that Trump (mean sentiment score: 1.0) has a higher sentiment score than Clinton (-0.031). The estimated difference in their mean sentiments is -1.

In summary, based on the available data, it is likely that Donald Trump had the highest sentiment score among the three candidates.

2.2 Question 2 - Website visits (between groups - Two factors)

2.2.1 Conceptual model

```
knitr::include_graphics("conceptualmodelQ2.png")
```



According to description of analysis we have to examine whether the version of the website, the portal, or a combination of the two had an impact on the number of pages visited. We have 2 Independent variables (IVs) namely Website Version (WV) and Website Portal (WP). Each of these variables can take a binary value of 0 or 1. For WV, 0 is the old website version whereas 1 is the new version. For WP, 0 represents the market which in this case is consumers and 1 are the companies. The dependent variable (DV) is the Number of

Pages visited. The IVs influence the DV separately but they also have an interaction effect between each other.

2.2.2 Specific Mathematical model

Describe the mathematical model that you fit on the data. Take for this the complete model that you fit on the data. Also, explain your selection for the priors. Assume Gaussian distribution for the number of page visits.

We will use the following model:

$pages \sim Norm(\mu, \sigma)$ [likelihood]

$\mu = a + b * version + c * portal + d * version * portal$ linear model

$a \sim Norm(100, 20)$ [a Intercept]

$b \sim Norm(0, 1)$ [b prior]

$c \sim Norm(0, 1)$ [c prior]

$d \sim Norm(0, 1)$ [d prior]

$\sigma \sim Uniform(0.1, 2)$ [σ prior]

2.2.3 Create Synthetic data

Create a synthetic data set with a clear interaction effect between the two factors for verifying your analysis later on. Report the values of the coefficients of the linear model used to generate synthetic data.

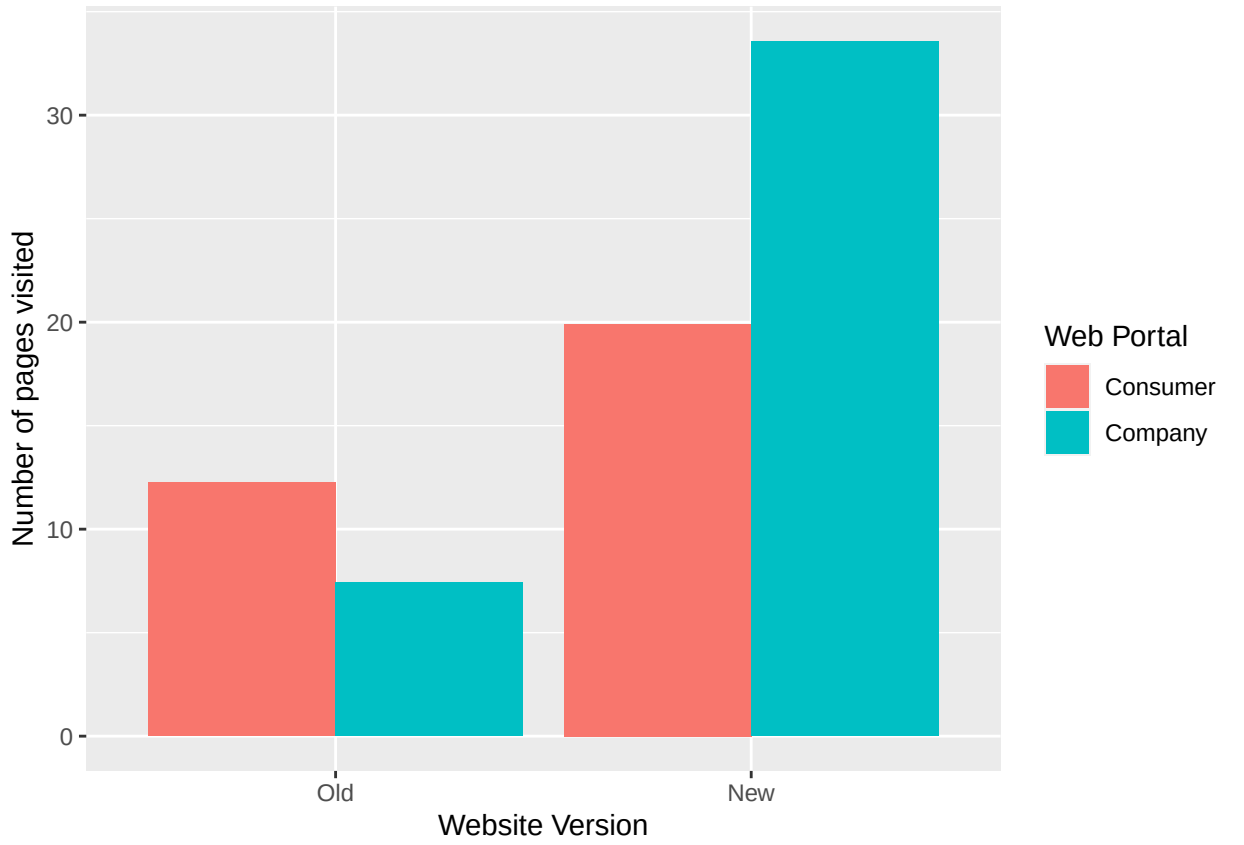
```
#Code block to load the csv file and convert the version and portal variables to
#Nominal labels for the real dataset

df<-read.csv("webvisit2.csv")

# Convert version and portal values to nominal labels
df$versionNominal <- factor(df$version, levels = c(0:1), labels = c("Old","New"))
df$portalNominal <- factor(df$portal, levels = c(0:1), labels = c("Consumer","Company"))
```

```
#include your code for generating the synthetic data
versionF<-rbinom(1000000,1,0.5)
portalF<-rbinom(1000000,1,0.5)
versionFNominal <- factor(versionF, levels = c(0:1), labels = c("Old","New"))
portalFNominal <- factor(portalF, levels = c(0:1), labels = c("Consumer","Company"))

linear_model_equ = 2+3*versionF+5*portalF+1*versionF*portalF

pagesF<-round(rnorm(1000000,mean=linear_model_equ+runif(1000000, 0, 1) * 49 + 1))
dFake<-data.frame(pagesF,versionFNominal,portalFNominal)
```

2.2.4 Visual inspection

Graphically examine the mean page visits for the four different conditions. Give a short explanation of the figure.

```
bar <- ggplot(df, aes(versionNominal , pages, fill = portalNominal))
bar + stat_summary(fun = mean, geom = "bar", position="dodge")+
  labs(x = "Website Version", y = "Number of pages visited")+
  guides(fill=guide_legend(title="Web Portal"))
```



The figure shows the mean number of pages visited in each of the 4 condition. Looking at just the Website version, there is an increase in the number of pages visited by both consumers and companies in the new version compared to the old version. Overall it seems consumers don't visit a lot of pages since the increase between the versions is not a lot. However the new version of the website had more drastic increase in the number of pages visited for companies.

Overall the new version of the website appears to be much better in terms of increasing the number of pages visited for both markets of consumers and companies.

2.2.5 Frequentist Approach

2.2.5.1 Model verification Verify your model analysis with synthetic data and show that it can reproduce the coefficients of the linear model that you used to generate the synthetic data set. Provide a short interpretation of the results, with a reflection of AICc, F-value, p-value etc.

```
# Linear model

modelF0 <- lm(pagesF ~ 1, data = dFake, na.action = na.exclude)
modelF1 <- lm(pagesF ~ versionFNominal, data = dFake, na.action = na.exclude)
modelF2 <- lm(pagesF ~ portalFNominal, data = dFake, na.action = na.exclude)
modelF3 <- lm(pagesF ~ versionFNominal+portalFNominal, data = dFake, na.action = na.exclude)
modelF4 <- lm(pagesF ~ versionFNominal + portalFNominal + versionFNominal:portalFNominal, data = dFake,
```

```
# F-value
anova(modelF4)
```

```
## Analysis of Variance Table
##
```

```
## Response: pagesF
##                                   Df    Sum Sq Mean Sq F value    Pr(>F)
## versionFNominal                1e+00   3026824 3026824 15073.9 < 2.2e-16 ***
## portalFNominal                 1e+00   7587175 7587175 37785.0 < 2.2e-16 ***
## versionFNominal:portalFNominal 1e+00     69196   69196   344.6 < 2.2e-16 ***
## Residuals                      1e+06 200797752     201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The anova function summarises the analysis of variance (or deviance) tables for modelF4 on synthetic data. The F-value for modelF4 is the lowest in comparison to the other models indicating that the group means are clustered together closely and have low variance. A lower F-value also indicates that the interaction effect of the IVs is not as significant if we compare it with respect to the other models but in this case still all of them are significant.

```
# p-value
summary(modelF4)
```

```
##
## Call:
## lm(formula = pagesF ~ versionFNominal + portalFNominal + versionFNominal:portalFNominal,
##     data = dFake, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.5143 -12.4718  -0.4718  12.4857  28.4931
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              27.51431    0.02835  970.39   <2e-16
## versionFNominalNew                        2.95745    0.04008   73.78   <2e-16
## portalFNominalCompany                     4.98289    0.04008  124.33   <2e-16
## versionFNominalNew:portalFNominalCompany  1.05220    0.05668   18.56   <2e-16
##
## (Intercept)                              ***
## versionFNominalNew                       ***
## portalFNominalCompany                    ***
## versionFNominalNew:portalFNominalCompany ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.17 on 999996 degrees of freedom
## Multiple R-squared:  0.05052,    Adjusted R-squared:  0.05051
## F-statistic: 1.773e+04 on 3 and 999996 DF,  p-value: < 2.2e-16
```

We can observe that all the factors (versionFNominal, portalFNominal, and their interaction) are highly significant. This suggests that both the version and portal factors, as well as their interaction, have a significant impact on the variable "pagesF". The coefficients of the linear model are being reproduced approximately through which we are able to verify that our model has fit well with the variables and the data.

```
# AICc for synthetic data
modelsF <-list(modelF0, modelF1, modelF2, modelF3, modelF4)
modelF.names <-c("modelF0","modelF1","modelF2","modelF3","modelF4")
aictab(cand.set = modelsF, modnames=modelF.names)
```

```
##
## Model selection based on AICc:
```

```
##
##         K    AICc Delta_AICc AICcWt Cum.Wt       LL
## modelF4 5 8140185       0.00      1      1 -4070088
## modelF3 4 8140528     342.55      0      1 -4070260
## modelF2 3 8155517   15332.11      0      1 -4077756
## modelF1 3 8177602   37416.79      0      1 -4088798
## modelF0 2 8192016   51830.71      0      1 -4096006
```

The AICc value for modelF4 is the lowest which indicates it has the best fit out of all the models.

2.2.5.2 Model analysis with Gaussian distribution assumed Redo the analysis now on the real data set. Assume Gaussian distribution for the number of page visits. Provide a short interpretation of the results, with an interpretation of AICc, F-value, p-value, etc.

```
# Linear model
model0 <- lm(pages ~ 1 , data = df, na.action = na.exclude)
model1 <- lm(pages ~ versionNominal , data = df, na.action = na.exclude)
model2 <- lm(pages ~ portalNominal , data = df, na.action = na.exclude)
model3 <- lm(pages ~ versionNominal + portalNominal , data = df, na.action = na.exclude)
model4 <- lm(pages ~ versionNominal + portalNominal + versionNominal:portalNominal , data = df, na.acti
```

```
anova(model4)
```

```
## Analysis of Variance Table
##
## Response: pages
##                              Df Sum Sq Mean Sq F value    Pr(>F)
## versionNominal                1  72190   72190 4437.31 < 2.2e-16 ***
## portalNominal                 1   6896    6896  423.86 < 2.2e-16 ***
## versionNominal:portalNominal  1  21174   21174 1301.51 < 2.2e-16 ***
## Residuals                   996  16204      16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model4)
```

```
##
## Call:
## lm(formula = pages ~ versionNominal + portalNominal + versionNominal:portalNominal,
##     data = df, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5669  -2.5669  -0.2658   2.5806  14.4331
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           12.2658     0.2620   46.82   <2e-16 ***
## versionNominalNew                      7.6464     0.3616   21.15   <2e-16 ***
## portalNominalCompany                  -4.8465     0.3790  -12.79   <2e-16 ***
## versionNominalNew:portalNominalCompany 18.5012     0.5128   36.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.033 on 996 degrees of freedom
## Multiple R-squared:  0.8609, Adjusted R-squared:  0.8604
```

```
## F-statistic:  2054 on 3 and 996 DF,  p-value: < 2.2e-16
```

The anova function summarises the analysis of variance (or deviance) tables for model4 on synthetic data. The F-value for model4 is 1301.51 which is higher than model3 but lower than model2. This suggests that the interaction effect of the IVs on the real data is potentially significant. This is useful to know since the p-values of all the models are highly significant but the higher F-value of the models gives an indication of which model interaction is highly significant. From the anova table it appears that versionNominal is the most significant.

```
# AICc for real data
models <-list(model0, model1, model2, model3, model4)
model.names <-c("model0","model1","model2","model3","model4")
aictab(cand.set = models, modnames=model.names)
```

```
##
## Model selection based on AICc:
##
##        K    AICc Delta_AICc AICcWt Cum.Wt       LL
## model4 5 5633.18       0.00      1      1 -2811.56
## model3 4 6467.00     833.81      0      1 -3229.48
## model1 3 6634.29    1001.11      0      1 -3314.13
## model2 3 7522.14    1888.96      0      1 -3758.06
## model0 2 7599.47    1966.29      0      1 -3797.73
```

The AICc value for model4 is the lowest which indicates it has the best fit out of all the models.

2.2.5.3 Assumption analysis Redo the analysis on the real tweet data set. This time assume a Poisson distribution for the number of page visits. For the best fitting models (Gaussian and Poisson), examine graphically the distribution of the residuals for the model that assumes Gaussian distribution and the model that assumes Poisson distribution. Give a brief interpretation of Poisson and Gaussian distribution assumptions.

```
# general linear model with poisson distribution
modelP0 <- glm(pages ~ 1 , data = df, na.action = na.exclude, family = poisson)
modelP1 <- glm(pages ~ versionNominal , data = df, na.action = na.exclude, family = poisson)
modelP2 <- glm(pages ~ portalNominal , data = df, na.action = na.exclude, family = poisson)
modelP3 <- glm(pages ~ versionNominal + portalNominal , data = df, na.action = na.exclude, family = pois
modelP4 <- glm(pages ~ versionNominal + portalNominal + versionNominal:portalNominal, data = df, na.acti
```

```
# AICc for real data but with poisson distribution
modelsP <-list(modelP0, modelP1, modelP2, modelP3, modelP4)
modelP.names <-c("modelP0","modelP1","modelP2","modelP3","modelP4")
aictab(cand.set = modelsP, modnames=modelP.names)
```

```
##
## Model selection based on AICc:
##
##         K     AICc Delta_AICc AICcWt Cum.Wt       LL
## modelP4 4  5566.07       0.00      1      1 -2779.01
## modelP3 3  6434.06     867.99      0      1 -3214.02
## modelP1 2  6791.72    1225.65      0      1 -3393.86
## modelP2 2 10328.83    4762.76      0      1 -5162.41
## modelP0 1 10789.83    5223.76      0      1 -5393.91
```

The AICc value for the interaction effect model i.e. modelP4 is the lowest out of all the models indicating the best fit even when we assume a poisson distribution.

```r
# Residual values interpretation
summary(model4) # MAX RESIDUAL 14.4331
```

```
##
## Call:
## lm(formula = pages ~ versionNominal + portalNominal + versionNominal:portalNominal,
##     data = df, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5669  -2.5669  -0.2658   2.5806  14.4331
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          12.2658     0.2620   46.82   <2e-16 ***
## versionNominalNew                     7.6464     0.3616   21.15   <2e-16 ***
## portalNominalCompany                 -4.8465     0.3790  -12.79   <2e-16 ***
## versionNominalNew:portalNominalCompany  18.5012     0.5128   36.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.033 on 996 degrees of freedom
## Multiple R-squared:  0.8609, Adjusted R-squared:  0.8604
## F-statistic:  2054 on 3 and 996 DF,  p-value: < 2.2e-16
```

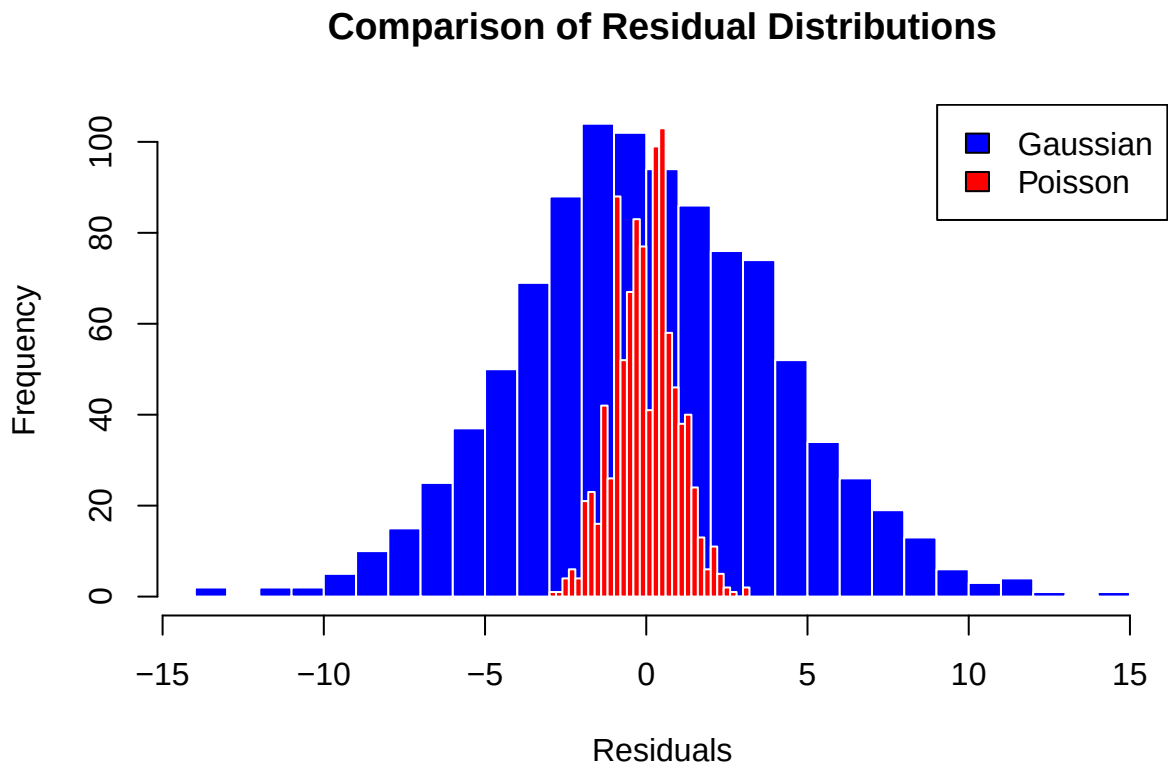
```r
summary(modelP4) #MAX RESIDULA 3.1835
```

```
##
## Call:
## glm(formula = pages ~ versionNominal + portalNominal + versionNominal:portalNominal,
##     family = poisson, data = df, na.action = na.exclude)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9716  -0.6696  -0.0762   0.5829   3.1835
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          2.50682    0.01855  135.16   <2e-16 ***
## versionNominalNew                    0.48452    0.02314   20.93   <2e-16 ***
## portalNominalCompany                -0.50272    0.03107  -16.18   <2e-16 ***
## versionNominalNew:portalNominalCompany  1.02493    0.03552   28.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 6164.72  on 999  degrees of freedom
## Residual deviance:  934.92  on 996  degrees of freedom
## AIC: 5566
##
## Number of Fisher Scoring iterations: 4
```

The summary of all the models (not shown above due to space constraints but can be easily verified by replacing the model4 with the other models) from both distributions reveal the lowest Max Deviance Residuals

for model4 and modelP4 i.e. the models with the interaction effect. These models are significant, have the lowest AICc value indicating the best fit, hence we choose them for further graphical examination.

```
#Graphical examination of residuals

residual_gaussian <- residuals(model4)
residual_poisson <- residuals(modelP4)

#Combined histogram plot of residuals
hist(residual_gaussian, breaks = "FD", col = "blue", border = "white",
     xlab = "Residuals", main = "Comparison of Residual Distributions")
hist(residual_poisson, breaks = "FD", col = "red", border = "white", add = TRUE)
legend("topright", legend = c("Gaussian", "Poisson"),
       col = c("blue", "red"), fill = c("blue", "red"))
```



Looking at the combined histogram plot of the residuals it appears to be more oriented towards a gaussian distribution than poisson. The variance in gaussian is also quite high compared to the poisson distribution. Overall based on the plot, it appears that the residuals follow the gaussian distribution more than the poisson distribution.

2.2.5.4 Simple effect analysis Continue with the model that assumes a Poisson distribution. If the analysis shows a significant two-way interaction effect, conduct a Simple Effect analysis to explore this interaction effect in more detail. Provide a brief interpretation of the results.

Our initial observation based on the plot of the mean page visits shows that there appears to be an increase between the number of pages visited by both consumers and companies with the new version. To further investigate this we conduct a simple effect analysis on the different markets i.e. consumers and companies.

```
#df$website_version_effect <- interaction(df$portalNominal, df$versionNominal) #merge two factors
df$portal_effect <- interaction(df$versionNominal, df$portalNominal) #merge two factors

levels(df$portal_effect) #to see the level in the new factor
```

```
## [1] "Old.Consumer" "New.Consumer" "Old.Company"  "New.Company"
```

```
contrastConsumer <-c(1,-1,0,0) #Only the consumer portal version
contrastCompany <-c(0,0,1,-1) #Only the company portal version

portalEff <- cbind(contrastConsumer,contrastCompany)
contrasts(df$portal_effect) <- portalEff #now we link the two contrasts with the factor simple

portalEffectModel <-glm(pages ~ portal_effect , data = df, na.action = na.exclude, family = poisson)

summary.lm(portalEffectModel)
```

```
##
## Call:
## glm(formula = pages ~ portal_effect, family = poisson, data = df,
##     na.action = na.exclude)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3601 -0.6526 -0.0759  0.5926  3.6360
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    2.753946   0.008563 321.600   <2e-16 ***
## portal_effectcontrastConsumer -0.242258   0.011159 -21.709   <2e-16 ***
## portal_effectcontrastCompany  -0.754724   0.012992 -58.093   <2e-16 ***
## portal_effect                  0.009741   0.017127   0.569     0.57
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9643 on 996 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 6.446e+05 on 3 and 996 DF,  p-value: < 2.2e-16
```

The two-way signifcant test is significant and we can observe that the effect of the both of the IVs are significant and its not just one Independent variable influencing the Dependent variable. This is confirmed by fitting a linear model with the interaction effects which shows that it has the best fit (lowest AICc and significant p-value).

```
anova(modelP4,test = "LRT")
```

2.2.5.5 Report section for a scientific publication

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: pages
##
## Terms added sequentially (first to last)
```

```
##
##
##                              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                            999     6164.7
## versionNominal                1   4000.1       998     2164.6 < 2.2e-16 ***
## portalNominal                 1    359.7       997     1804.9 < 2.2e-16 ***
## versionNominal:portalNominal  1    870.0       996      934.9 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.lm(portalEffectModel)
```

```
##
## Call:
## glm(formula = pages ~ portal_effect, family = poisson, data = df,
##     na.action = na.exclude)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3601 -0.6526 -0.0759  0.5926  3.6360
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    2.753946   0.008563 321.600   <2e-16 ***
## portal_effectcontrastConsumer -0.242258   0.011159 -21.709   <2e-16 ***
## portal_effectcontrastCompany  -0.754724   0.012992 -58.093   <2e-16 ***
## portal_effect                  0.009741   0.017127   0.569     0.57
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9643 on 996 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 6.446e+05 on 3 and 996 DF,  p-value: < 2.2e-16
```

A linear model was fitted on the number of pages visited on a website, taking the website version and website portal as independent variables, and including a two-way interaction between these variables. The analysis of deviance table revealed significant effects for all predictor variables: versionNominal ($X^2 = 4000.1$, df = 1, $p < 0.001$), portalNominal ($X^2 = 359.7$, df = 1, $p < 0.001$), and the interaction term versionNominal:portalNominal ($X^2 = 870.0$, df = 1, $p < 0.001$)." A simple two-way interaction analysis revealed significant effects for the predictor variables for Consumer market (t = -21.709, $p < 0.001$), and Companies market (t = -58.093, $p < 0.001$).

2.2.6 Bayesian Approach

For the Bayesian analyses, use the rethinking and/or BayesianFirstAid library

2.2.6.1 Verification Analysis Verify your model analysis with synthetic data and show that it can reproduce the coefficients of the linear model that you used to generate the synthetic data set. Provide a short interpretation of the results, with a reflection of WAIC, and 95% credibility interval of coefficients for individual celebrities.

```
versionFBayes<-rbinom(10000,1,0.5)
portalFBayes<-rbinom(10000,1,0.5)

linear_model_equ = 2+3*versionFBayes+5*portalFBayes+1*versionFBayes*portalFBayes
```

```
pagesFBayes<-round(rnorm(10000,mean=linear_model_equ+runif(10000, 0, 1) * 49 + 1))
pagesFBayes_Poisson<-round(rpois(10000,lambda=linear_model_equ+runif(10000, 0, 1) * 49 + 1))
pagesFBayes <- as.integer(pagesFBayes)
pagesFBayes_Poisson <- as.integer(pagesFBayes_Poisson)

df_bayesF<-data.frame(pagesFBayes,pagesFBayes_Poisson,versionFBayes,portalFBayes)
```

```
m0 <-ulam(
  alist(
    pagesFBayes ~ dnorm(mu, sigma),
    mu <- a,
    a ~ dnorm(100, 20),
    sigma ~ dunif(0.1, 2)
  ), data = df_bayesF ,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m1 <-ulam(
  alist(
    pagesFBayes ~ dnorm(mu, sigma),
    mu <- a + b*versionFBayes,
    a ~ dnorm(100, 20),
    b ~ dnorm(0, 1),
    sigma ~ dunif(0.1, 2)
  ), data = df_bayesF ,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)
m2 <-ulam(
  alist(
    pagesFBayes ~ dnorm(mu, sigma),
    mu <- a + c*portalFBayes,
    a ~ dnorm(100, 20),
    c ~ dnorm(0, 1),
    sigma ~ dunif(0.1, 2)
  ), data = df_bayesF,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m3 <-ulam(
  alist(
    pagesFBayes ~ dnorm(mu, sigma),
    mu <- a + b*versionFBayes + c*portalFBayes,
    a ~ dnorm(100, 20),
    b ~ dnorm(0, 1),
    c ~ dnorm(0, 1),
    sigma ~ dunif(0.1, 2)
  ), data = df_bayesF,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m4 <-ulam(
  alist(
    pagesFBayes ~ dnorm(mu, sigma),
    mu <- a + b*versionFBayes + c*portalFBayes + d*versionFBayes*portalFBayes,
    a ~ dnorm(100, 20),
    c(b,c,d) ~ dnorm(0, 1),
    sigma ~ dunif(0.1, 2)
  ), data = df_bayesF,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
```

```
)
```

```
compare(m0,m1,m2,m3,m4) #WAIC - m4 is best out-of-sample fit because of the smallest WAIC value
```

```
##        WAIC       SE       dWAIC       dSE     pWAIC       weight
## m3 539843.0 4568.145     0.00000        NA 149.61601 1.000000e+00
## m4 539880.8 4567.632    37.73188   32.1308 190.03890 6.406587e-09
## m2 547290.0 4729.768  7447.01402 1231.6710 103.26871 0.000000e+00
## m1 557588.5 4968.685 17745.43017 1897.2739 107.22115 0.000000e+00
## m0 565106.1 5120.623 25263.05328 2271.4601  53.02846 0.000000e+00
```

```
precis(m4, prob=0.95) #95%  credibility interval of coefficients
```

```
##             mean           sd        2.5%      97.5%     n_eff     Rhat4
## a     27.3515246 3.846569e-02 27.27259000 27.4254075  820.1811 1.0012467
## d      0.1803615 7.552212e-02  0.02919642  0.3318224  716.3472 1.0014525
## c      5.2361551 5.506244e-02  5.13109000  5.3446272  763.0739 1.0006588
## b      3.3669825 5.313841e-02  3.26100900  3.4717485  690.7068 1.0015634
## sigma  1.9999968 5.239819e-06  1.99998975  2.0000000 1851.0388 0.9991671
```

Model m4 has the best fit because of the smallest WAIC value. The credibility interval shows that the coefficients b, c and d are non-null and positive which indicates that the type of version, portal and the interaction effect contribution was positive towards the number of pages.

2.2.6.2 Model description We take model m4 since its WAIC is the lowest which means its the best fit and hence we will use it as our mathemathical model.

$pages \sim Poisson(\lambda)$ [likelihood]

$\log(\lambda) = a + b * version + c * portal + d * version * portal$ linear model

$a \sim Norm(100, 20)$ [a Intercept]

$b \sim Norm(0, 1)$ [b prior]

$c \sim Norm(0, 1)$ [c prior]

$d \sim Norm(0, 1)$ [d prior]

```
#Assuming poisson distribution

m0PFake <-ulam(
  alist(
    pagesFBayes_Poisson ~ dpois(p),
    log(p)<-a,
    a ~ dnorm(100, 25)
  ), data = df_bayesF ,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m1PFake <-ulam(
  alist(
    pagesFBayes_Poisson ~ dpois(p),
    log(p) <- a + b*versionFBayes,
    a ~ dnorm(100, 25),
    b ~ dnorm(0, 1)
  ), data = df_bayesF ,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m2PFake <-ulam(
```

```
  alist(
    pagesFBayes_Poisson ~ dpois(p),
    log(p) <- a + c*portalFBayes,
    a ~ dnorm(100, 25),
    c ~ dnorm(0, 1)
  ), data = df_bayesF,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m3PFake <-ulam(
  alist(
    pagesFBayes_Poisson ~ dpois(p),
    log(p) <- a + b*versionFBayes + c*portalFBayes,
    a ~ dnorm(100, 25),
    b ~ dnorm(0, 1),
    c ~ dnorm(0, 1)
  ), data = df_bayesF,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m4PFake <-ulam(
  alist(
    pagesFBayes_Poisson ~ dpois(p),
    log(p) <- a + b*versionFBayes + c*portalFBayes + d*versionFBayes*portalFBayes,
    a ~ dnorm(100, 25),
    c(b,c,d) ~ dnorm(0, 1)
  ), data = df_bayesF,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)
```

```
compare(m0PFake,m1PFake,m2PFake,m3PFake,m4PFake) #WAIC
```

```
##             WAIC       SE      dWAIC       dSE     pWAIC        weight
## m4PFake 131381.2 842.4616    0.00000        NA 30.263826  9.999994e-01
## m3PFake 131409.8 842.3524   28.59434  33.80869 21.847921  6.177557e-07
## m2PFake 132397.7 866.0513 1016.44059 171.67162 15.613425 1.917477e-221
## m1PFake 134060.9 898.8733 2679.64580 281.39032 14.776323  0.000000e+00
## m0PFake 135058.5 922.8476 3677.30469 331.71153  7.397198  0.000000e+00
```

```
precis(m4PFake, prob=0.95) #95%  credibility interval of coefficients
```

```
##         mean          sd       2.5%      97.5%    n_eff    Rhat4
## a 3.31604673 0.003937826 3.30831975 3.32394025 520.3138 1.002394
## d 0.04396793 0.007474694 0.02921151 0.05873129 516.3585 1.004731
## c 0.16011282 0.005329123 0.14980285 0.17043622 526.2156 1.002569
## b 0.08813063 0.005474995 0.07807329 0.09922360 493.0408 1.005752
```

Because of the randomness in data generation either model m3PFake or m4PFake has the best fit because of the smallest WAIC value. The credibility interval for both of them in at least our test showed that shows that the coefficients b, c and d are non-null but they can lie close to 0, so it is possible that their range is null.

2.2.6.3 Model comparison Redo the analysis on actual data. Assume Poisson distribution for the number of page visits. Provide brief interpretation of the analysis results (e.g. WAIC, and 95% credibility interval of coefficients).

```
m0P <-ulam(
  alist(
    pages ~ dpois(p),
    log(p) <- a ,
```

```
    a ~ dnorm(100, 25)
  ), data = df ,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m1P <-ulam(
  alist(
    pages ~ dpois(p),
    log(p) <- a + b*version,
    a ~ dnorm(100, 25),
    b ~ dnorm(0, 1)
  ), data = df ,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m2P <-ulam(
  alist(
    pages ~ dpois(p),
    log(p) <- a + c*portal,
    a ~ dnorm(100, 25),
    c ~ dnorm(0, 1)
  ), data = df,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m3P <-ulam(
  alist(
    pages ~ dpois(p),
    log(p) <- a + b*version + c*portal,
    a ~ dnorm(100, 25),
    b ~ dnorm(0, 1),
    c ~ dnorm(0, 1)
  ), data = df,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)

m4P <-ulam(
  alist(
    pages ~ dpois(p),
    log(p) <- a + b*version + c*portal + d*version*portal,
    a ~ dnorm(100, 25),
    c(b,c,d) ~ dnorm(0, 1)
  ), data = df,iter = 1000, chains = 4, cores = 4, log_lik = TRUE
)
```

```
compare(m0P,m1P,m2P,m3P,m4P) #WAIC
```

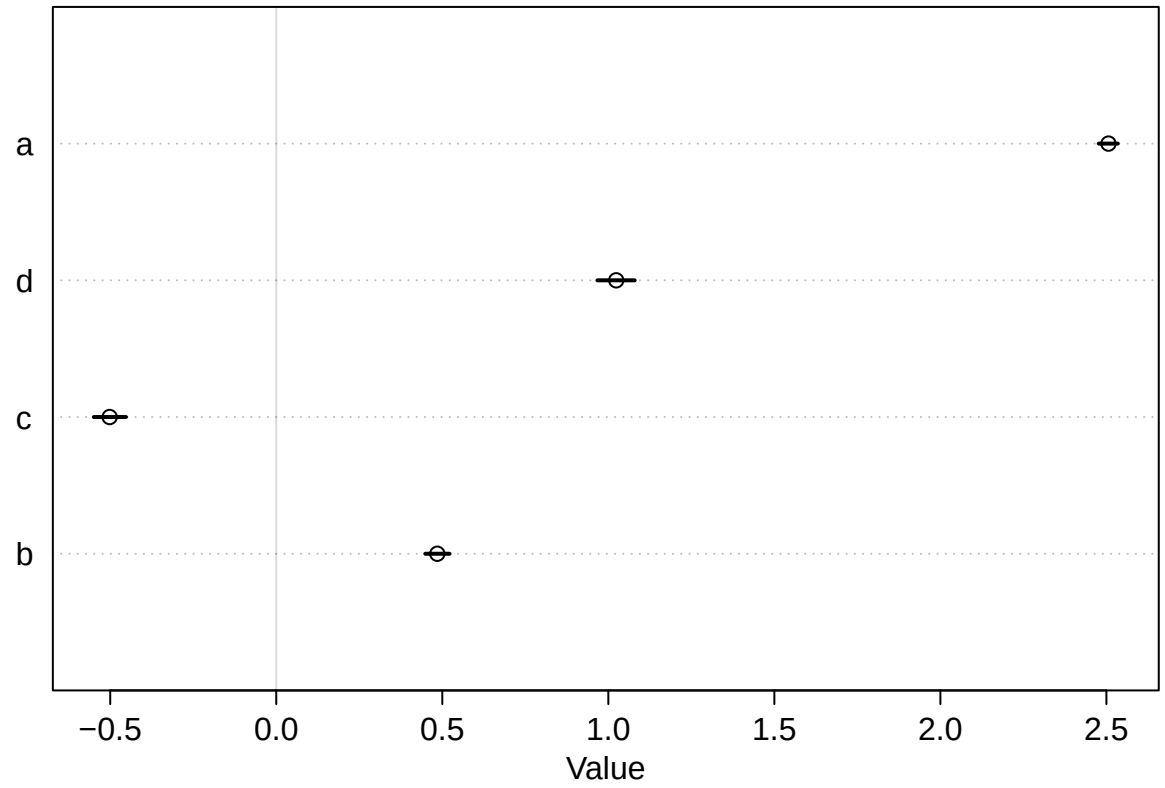
```
##          WAIC        SE    dWAIC       dSE    pWAIC        weight
## m4P  5565.715  42.19794    0.000        NA 3.716071  1.000000e+00
## m3P  6435.749  71.53397  870.034  61.99695 4.827704 1.187209e-189
## m1P  6794.000  79.89120 1228.286  66.92445 4.228597 1.910458e-267
## m2P 10336.660 191.55365 4770.946 197.57720 9.913321  0.000000e+00
## m0P 10795.123 182.49958 5229.408 184.00586 6.259782  0.000000e+00
```

```
precis(m4P, prob=0.95) #95%  credibility interval of coefficients
```

```
##         mean         sd       2.5%      97.5%    n_eff    Rhat4
## a  2.5062276 0.01805649  2.4694835  2.5408958 505.1253 1.002683
## d  1.0238290 0.03534506  0.9528506  1.0917838 505.9599 1.004004
```

```
## c -0.5017501 0.03067557 -0.5602661 -0.4408118 504.9598 1.004815
## b  0.4849631 0.02295646  0.4398218  0.5314009 542.5986 1.002392
```

```
plot(m4P)
```



Model m4 has the best fit because of the smallest WAIC value of 5565.7, this is different from the previous result we got on synthetic data where model 3 or 4 was better but on the real data, the interaction effect between the IVs appears to fit better. The credibility interval shows that all coefficients b,c and d are non null. The contribution of d and b is positive but coefficient c's contribution is negative towards the number of pages.

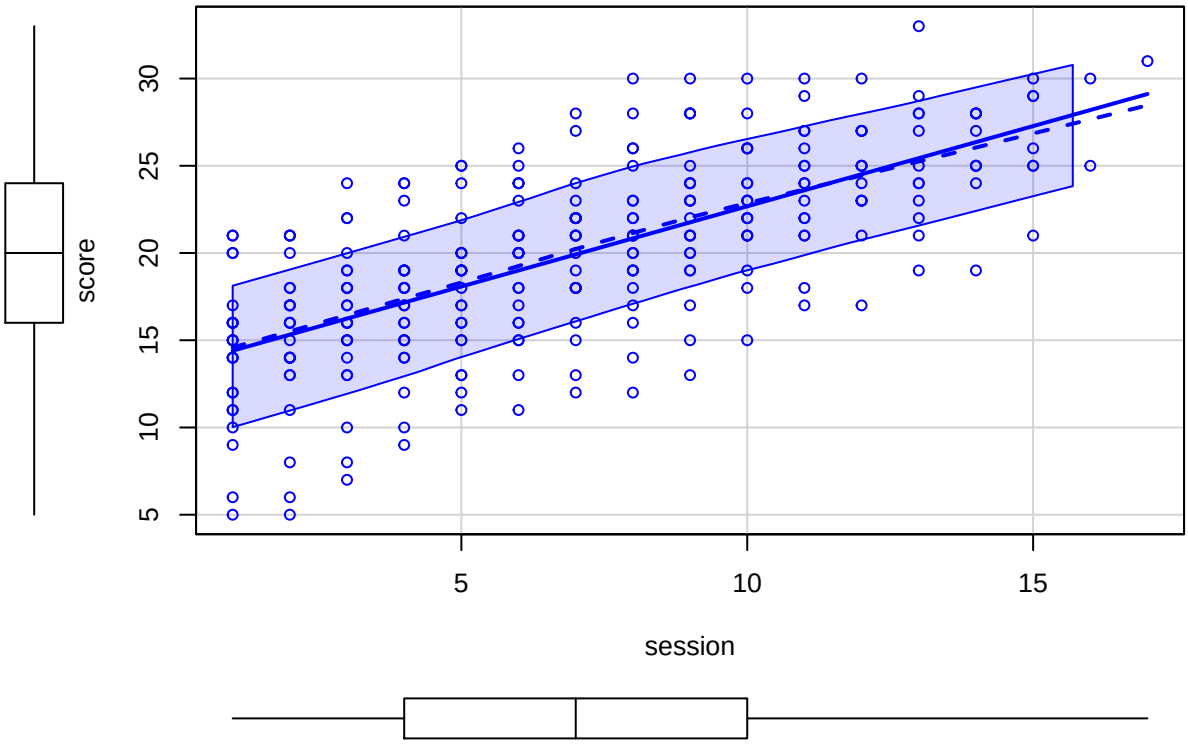
3 Part 3 - Multilevel model

3.1 Visual inspection

Use graphics to inspect the distribution of the score, and relationship between session and score. Give a short description of the figure.

We can see from the scatterplot of the test score and session that the test score and session are linearly correlated.

```
#include your code and output in the document
library(car)
set = read.csv("set1.csv")
scatterplot(score ~ session, data = set)
```



3.2 Frequentist approach

3.2.1 Multilevel analysis

Conduct multilevel analysis and calculate 95% confidence intervals thereby assuming a Gaussian distribution for the scores, determine:

- If session has an impact on people score
- If there is significant variance between the participants in their score

```
#include your code and output in the document
library(nlme)
```

```
## Warning: package 'nlme' was built under R version 4.2.3
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
randomIntercept <- lme(score ~ 1, data = set, random = ~1|subject, method = "ML", control = list(opt="o
randomSession <- update(randomIntercept, .~. + session)
anova.lme(randomIntercept,randomSession)
```

```
##                 Model df       AIC       BIC    logLik   Test  L.Ratio p-value
## randomIntercept     1  3 1660.3009 1671.2478 -827.1504
## randomSession       2  4  939.2478  953.8437 -465.6239 1 vs 2 723.0531  <.0001
```

```
summary(randomSession)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: set
##        AIC      BIC    logLik
##   939.2478 953.8437 -465.6239
##
## Random effects:
##  Formula: ~1 | subject
##         (Intercept) Residual
## StdDev:     3.84441 1.011498
##
## Fixed effects:  score ~ session
##                 Value Std.Error  DF  t-value p-value
## (Intercept) 13.167788 0.8137662 260 16.18129       0
## session      0.991579 0.0158911 260 62.39823       0
##  Correlation:
##         (Intr)
## session -0.13
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.53305203 -0.52709350  0.01234201  0.58981846  2.43523007
##
## Number of Observations: 284
## Number of Groups: 23
```

```
Anova(randomSession)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: score
##          Chisq Df Pr(>Chisq)
## session 3921.2  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
intervals(randomSession, 0.95)
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                  lower       est.     upper
## (Intercept) 11.5710291 13.1677882 14.764547
## session      0.9603981  0.9915794  1.022761
##
##  Random Effects:
##   Level: subject
##                    lower    est.    upper
## sd((Intercept)) 2.874468 3.84441 5.141645
##
##  Within-group standard error:
##     lower      est.     upper
## 0.9286347 1.0114984 1.1017561
```

```r
3.844^2 / (3.844^2 + 1.012^2)
```

```
## [1] 0.9351827
```

3.2.2 Report section for a scientific publication

Write a small section for a scientific publication, in which you report the results of the analyses, and explain the conclusions that can be drawn.

A multilevel analysis was performed on the learning dataset taking the session as a fixed intercept and the subject id as a random intercept. The results showed a significant effects of the session on the test score ($t(260) = 62.4$, $p < 0.001$), ($X^2(1) = 3921.2$, $df = 1$, $p < 0.001$) including when compared to the random intercept ($L(4) = 723.05$, $p < 0.0001$) which was also shown by the lower AICc scores. The relationship between the session and test score showed significant variance in intercepts across sessions, $SD = 3.84$ (95% Cl 2.87, 5.14). In addition, We found that the interclass correlation between participants is very high (0.94).

3.3 Bayesian approach

For the Bayesian analyses, use the rethinking and/or BayesianFirstAid library

3.3.1 Model description

Describe the mathematical model fitted on the most extensive model. (hint, look at the mark down file of the lectures to see example on formulate mathematical models in markdown). Assume a Gaussian distribution for the scores. Justify the priors.

$score \sim Norm(\mu, \sigma)$ [likelihood]

$\mu = a + a\_subject[subject] + b * session$ linear model

$a \sim Norm(20, 5)$ [a prior]

$b \sim Norm(0, 1)$ [b prior]

$a\_subject[subject] \sim Norm(0, sigma\_subject)$ [*adaptive* prior]

$sigma\_subject \sim dcauchy(0, 1)$ [*hyper* prior]

$\sigma \sim dcauchy(0, 1)$ [σ prior]

We chose the to distribute the a prior at (20, 5) based on visual analysis of the data.

3.3.2 Model comparison

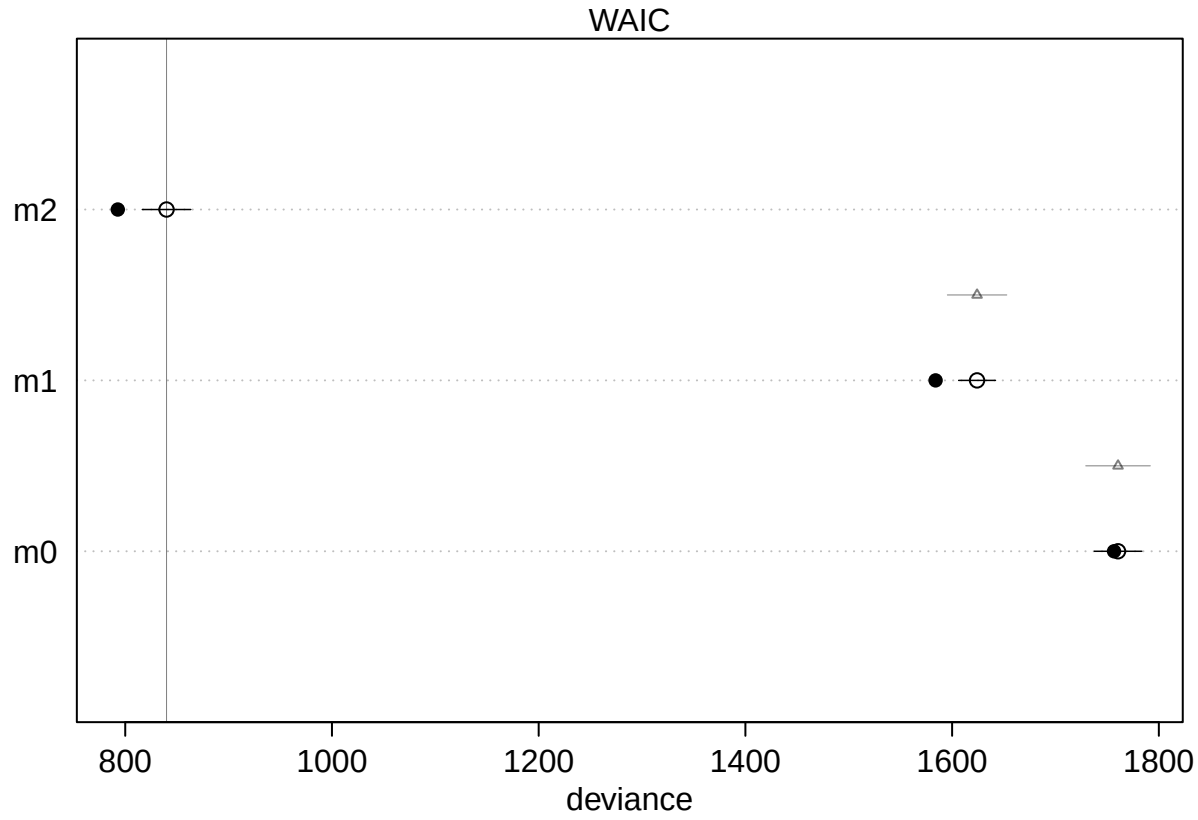
Compare models with increasing complexity.

While adding the adaptive prior for Subject id slightly improves the WAIC of the second model, the improvement provided by extending the model with session as a fixed factor was much greater.

```r
#include your code and output in the document
m0 <- ulam(
  alist(
    #Likelihood
    score ~ dnorm(mu, sigma),
    #linear model
    mu <- a ,
    #fixed priors
    a ~ dnorm(20, 5),
    sigma ~ dcauchy(0, 1)
  ), data = set, iter = 1000, chains = 4, cores = 4, log_lik = TRUE
```

```
)

m1 <- ulam(
  alist(
    #Likelihood
    score ~ dnorm(mu, sigma),
    #linear model
    mu <- a + a_subject[subject],
    #adaptive prior
    a_subject[subject] ~ dnorm(0, sigma_subject),
    #hyper prior
    sigma_subject ~ dcauchy(0, 1),
    #fixed priors
    a ~ dnorm(20, 5),
    sigma ~ dcauchy(0, 1)
  ), data = set, iter = 1000, chains = 4, cores = 4, log_lik = TRUE, control=list(adapt_delta=.99)
)

m2 <- ulam(
  alist(
    #Likelihood
    score ~ dnorm(mu, sigma),
    #linear model
    mu <- a + a_subject[subject] + b*session,
    #adaptive prior
    a_subject[subject] ~ dnorm(0, sigma_subject),
    #hyper prior
    sigma_subject ~ dcauchy(0, 1),
    #fixed priors
    a ~ dnorm(20, 5),
    b ~ dnorm(0, 1),
    sigma ~ dcauchy(0, 1)
  ), data = set, iter = 1000, chains = 4, cores = 4, log_lik = TRUE, control=list(adapt_delta=.99)
)
```

```
plot(compare(m0,m1,m2))
```



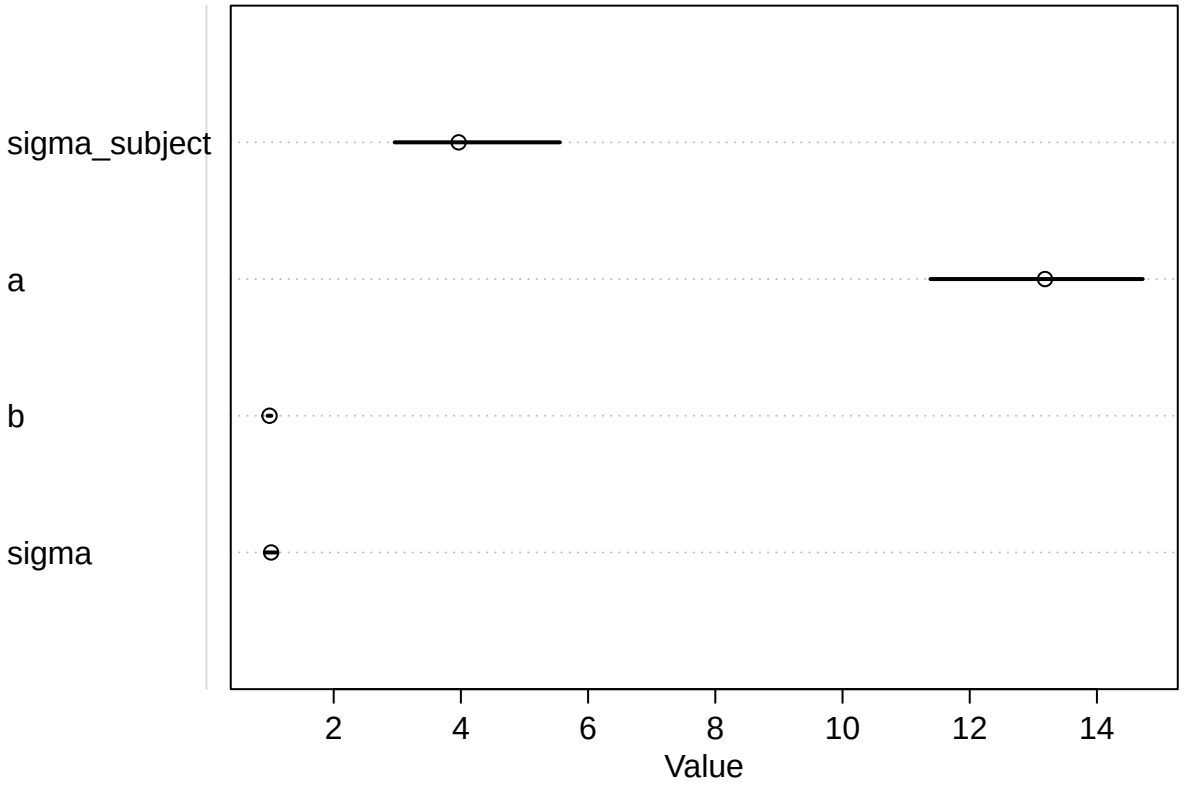
3.3.3 Estimates examination

Examine the estimate of parameters of the model with best fit, and provide a brief interpretation.

We can see that the means of all coefficients are above 0 which means they are all relevant for the performance of the model, with a and sigma_subject having a higher mean and standard deviations of 0.88 and 0.61 while b and sigma have lower means around 1 but also smaller standard deviations of 0.02 and 0.04 with their confidence intervals not falling bellow 0.

```
#include your code and output in the document
plot(precis(m2, prob=.95))
```

```
## 23 vector or matrix parameters hidden. Use depth=2 to show them.
```




```
precis(m2, prob=.95)
```

```
## 23 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
##                      mean         sd       2.5%     97.5%     n_eff    Rhat4
## sigma_subject  3.9650513 0.67878164  2.9578675  5.556015 469.82292 1.001681
## a             13.1836550 0.81202860 11.3823225 14.719210  83.93741 1.068163
## b              0.9911494 0.01619595  0.9602860  1.020751 511.40368 1.003369
## sigma          1.0162809 0.04264381  0.9332039  1.100174 611.10665 1.006192
```