Coursework assignment A - 2022-2023

CS4125 Seminar Research Methodology for Data Science

Student names

16/04/2023

Contents

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

1.1 The motivation for the planned research

During this century we have seen the rise of powerful search engines, which are able to quickly provide you with links to the data you are trying to find. A recent development in this field is the introduction of large language models (LLMs). These LLMs have the capability to present the search results in a more natural and human-friendly way. The hype around these models has led large search engines to quickly incorporate these LLMs in their products. An example is Bing, which has incorporated a version of chatGPT into their search browser. The question that remains is if this integration actually leads to an increase in perceived user experience or if the enthusiasm is purely based on the current hype. By conducting the experiments described below, the goal is to find out if the chatGPT integration in the Bing search engine improves the user experience.

1.2 The theory underlying the research

Chat-bots have been successful as interactive information retrieval tools (Tariverdiyeva, Gunay. "Chatbots' Perceived Usability in Information Retrieval Tasks: An Exploratory Analysis." 2019).

New large language models fall under the chatbot category but it is still unclear whether they offer an advantage when it comes to user experience when viewed as information retrieval systems.

In information retrieval many measures have been proposed as proxy variables to measure user experience (Dalrymple, Prudence Ward, and Douglas L. Zweizig. "Users' Experience of Information Retrieval Systems: An Exploration of the Relationship between Search Experience and Affective Measures." Library and Information Science Research 14.2 (1992): 167-81).

More specifically, session duration has been a very popular measure for evaluating interactive information retrieval systems (Kelly, Diane. "Methods for evaluating interactive information retrieval systems with users." Foundations and Trends® in Information Retrieval 3.1–2 (2009): 1-224.)

1.3 Research questions

Does the integration of chatGPT into the Bing search engine lead to an increase in user experience over the Bing search engine without this integration?

1.4 The related conceptual model

Independent variable: Search Mode - categorial variable - ChatGPT, standard Dependent variable: user experiences Mediating variable: (at least 1): session length, relevance of results Moderating variable: (at least 1): Experience with chatbots

1.5 Experimental Design

Hypothesis: the integration of chatGPT has a positive effect on user experience during search.

Design: we use a within-subject experiment. All the participants both evaluate the standard search engine and the search engine with chatGPT integration. Because we only have one group, random allocation is not an issue. First each participant gets a set of questions to which they should find the answer using the regular Bing search engine. Then, the user experience is evaluated using a questionnaire. Afterwards, the participant is again presented with a new set of questions to which they should find the answers using the search engine with the chatGPT integration. The user experience is again evaluated using a questionnaire. To make sure that the type of questions are equally hard between the two search experiments, we randomly select half of the total number of questions for each of the experiments.

1.6 Experimental procedure

A room is prepared with a laptop with two search engines in it: search engine 1 is a standard version of Bing, search engine 2 is Bing with Chat-GPT integration. Each participant is called into the room and is given a set of questions to answer or find information for. The participant uses the Bing search engine to find information for those questions. Afterwards the user is given a questionnaire to answer. Then, the participant is given a new set of questions and uses the Bing with Chat-GPT engine after which he/she is given the same questionnaire again. Break up order of questions.

1.7 Measures

We use a questionnaire that should give us an indication of the perceived user experience of the participants for both the normal search engine and the chatGPT integrated search engine. The questionnaire consists of a list of statements, where the participants have to fill in whether they agree with a certain statement. The focus of this questionnaire is on the overall user satisfaction, not on the specific results that the search engines return. The answers are based on an interval Likert-scale from 1 to 7. Where a higher number corresponds to a higher agreement with the statement. The answers to the statements are collected and can be compared between the two experiment rounds (with and without chatGPT). The Likert-scale answers allow us to make a clear numerical comparison, which finally is our measure for the experiment.

1.8 Participants

We want to find a group of participants in Delft as diverse as possible, because we want to find the general user experience improvement. If we only take a subsample of the population, like elderly people or students, we do not get a very general idea and loose external validity. We need 119 people, because we use a Likert scale using intervals. This provides us with a 95% confidence interval. We find participants by calling people in Delft and asking if they want to participate in our experiment, after which the experiment is conducted some other moment. This way we hope to get a random sample from the whole population in Delft.

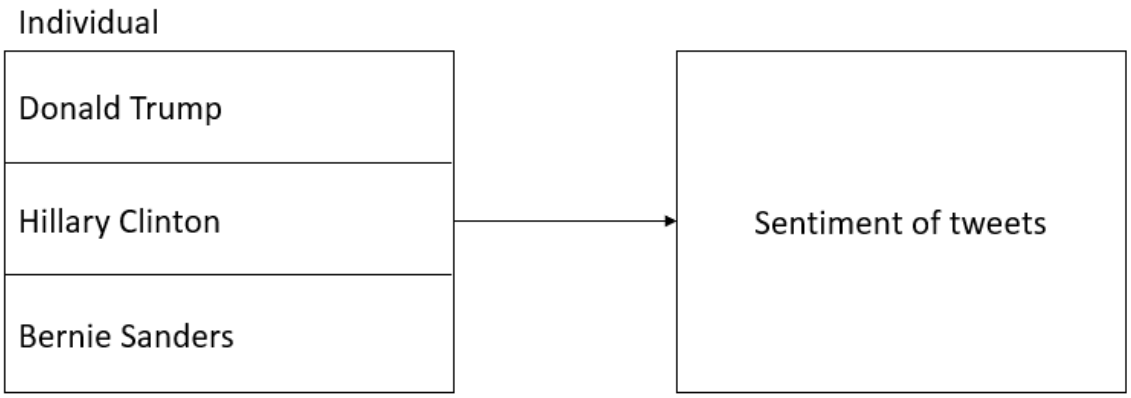
1.9 Suggested statistical analyses

We want to find the difference between the user experience to see if it has increased. This means that the results of the two experiments have to be paired with each other. The results of the two experiments are dependent, because a participant maybe in general already gives higher ratings. So this means that with our interval questions, we could use a paired sample t-test or repeated measure ANOVA.

2 Part 2 - Generalized linear models

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

2.1.1 Conceptual model



2.1.2 Model description

The model that we fit on the data is normal distribution where the mean is dependent on the individual that tweeted the tweet. This way a new mean is created for each individual. So we have the following equations:

$score \sim dnorm(mu, sigma)$

$mu < -a[Candidate]$

$a[Candidate] \sim dnorm(0, 10)$

$sigma \sim dunif(0.001, 20)$

The priors defined here are loosely based on the visual inspection of the data, which is given below. In these graphs we can see that the sentiment of the tweets for each individual are centered around 0. So we take a distribution for the mean of our model that also has a mean of 0, but we add a standard deviation of 10 to be sure that the actual mean is found. For the sigma of our model, we also choose quite an uninformed prior. It seems that the data has a standard deviation at most 5, but to be sure that it is contained within our prior, we extend it to 20.

2.1.3 Generate Synthetic data

Below the code is given for the creation of the synthetic data. We pick three normal distributions, where each distribution has a different mean, but the same standard deviation (sd=2). The three distributions have a mean of -5, 0 and 5 for T, C and B respectively.

```
# Synthesis of a test data set.
sequence <- seq(-10, 10, by = .1)
test_T = rnorm(sequence, mean=-5, sd=2)
test_C = rnorm(sequence, mean=0, sd=2)
test_B = rnorm(sequence, mean=5, sd=2)

sem_test<-data.frame(test_T, test_C, test_B)

semFrameTest <-melt(sem_test, measured=c(test_T, test_C, test_B))
```

```
## Using  as id variables
```

```
names(semFrameTest) <- c("Candidate", "score")
semFrameTest$Candidate <-factor(semFrameTest$Candidate, labels=c("Donald Trump",
                                        "Hillary Clinton", "Bernie Sanders"))
```

2.1.4 Collecting tweets, and data preparation

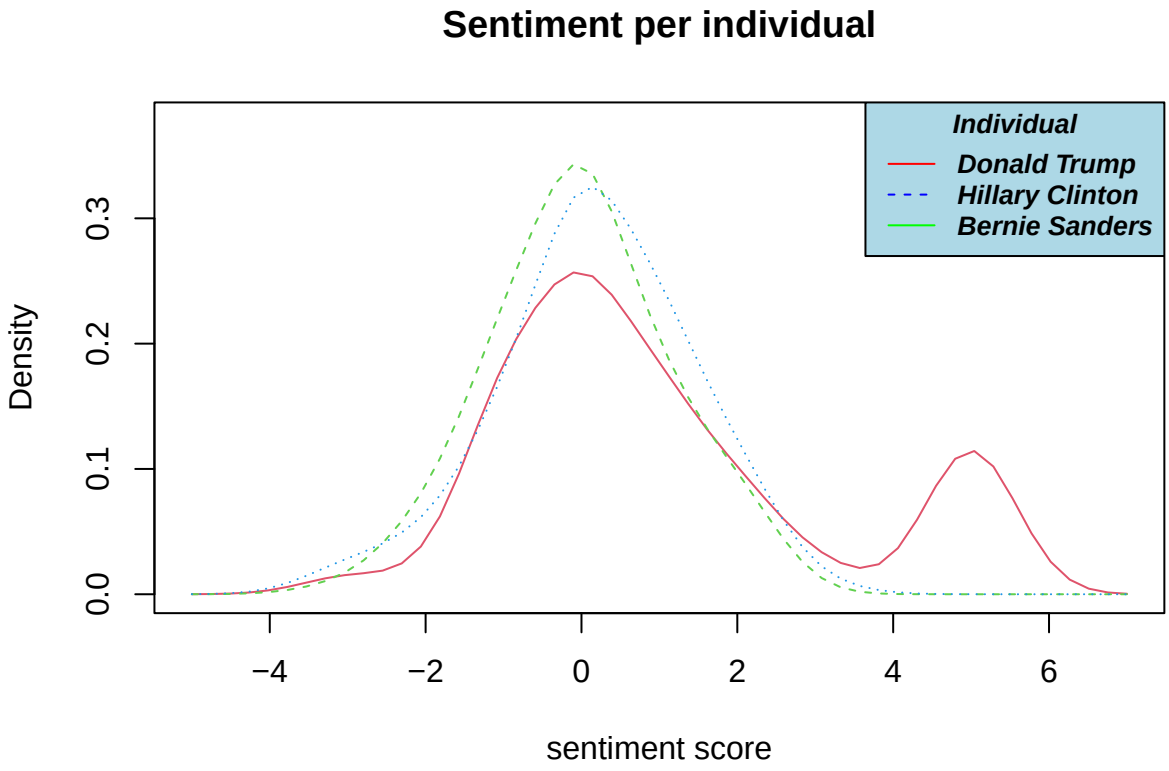
2.1.5 Visual inspection Mean and distribution sentiments

To get an idea of the distributions of the data, we make some graphs. First, we plot the data distributions of the tweet sentiment per individual. This graph shows that the distributions seem similar, all with a mean around 0. However, for the Trump data, there seem to be more more values with a higher sentiment. If we look at the boxplots, we see that Trump seems to have the most positive sentiment, but he also has more extreme maxima. Hillary seems to be a bit less positive than Bernie.

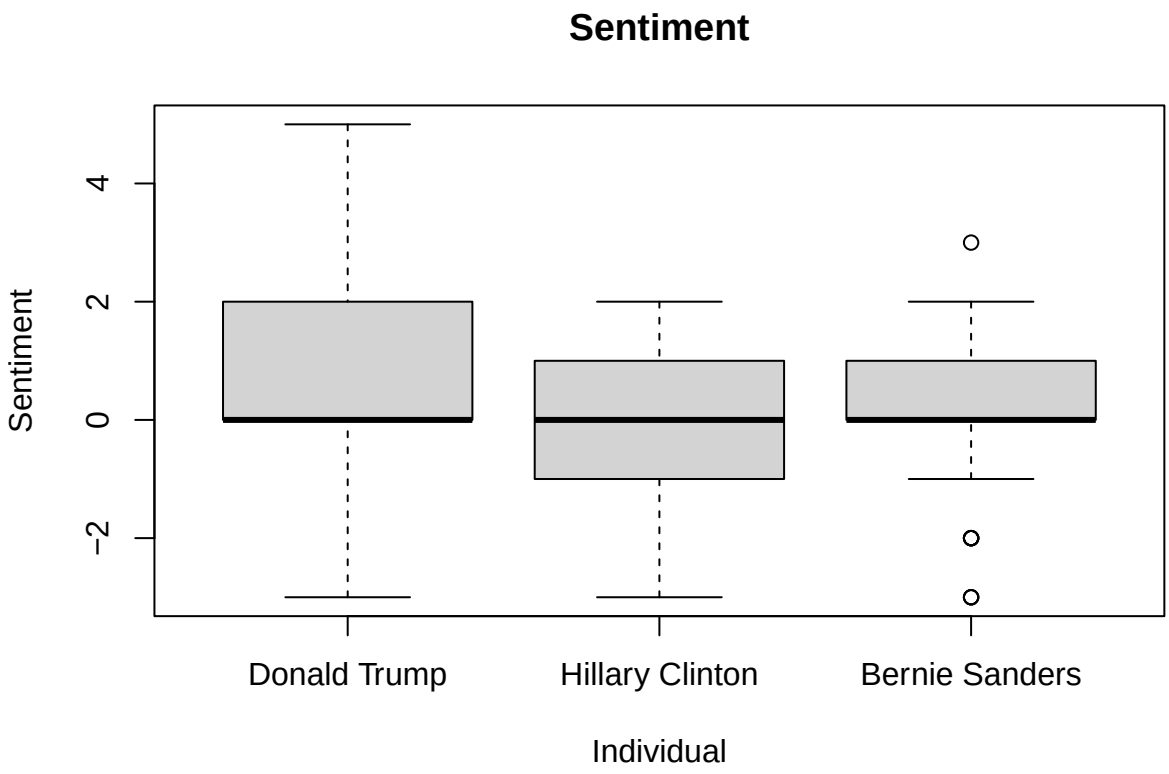
```
#include your analysis code and output in the document
trump_inspect = subset(semFrame, (Candidate == "Donald Trump"),
                       select=c(score))

hillary_inspect = subset(semFrame, (Candidate == "Hillary Clinton"),
                         select=c(score))

bernie_inspect = subset(semFrame, (Candidate == "Bernie Sanders"),
                        select=c(score))

library(sm)
```

```
## Package 'sm', version 2.2-5.7: type help(sm) for summary information
```

```
sm.density.compare(semFrame$score, semFrame$Candidate, xlab = "sentiment score")
title(main="Sentiment per individual")
legend('topright', legend=levels(semFrame$Candidate),
       col=c('red', 'blue', 'green'), lty=1:2, cex=0.8,
       title="Individual", text.font=4, bg='lightblue')
```




```
boxplot(semFrame$score ~ semFrame$Candidate, data=semFrame, main="Sentiment",
xlab="Individual", ylab="Sentiment")
```



2.1.6 Frequentist approach

2.1.6.1 Analysis verification If we input the synthetic data into the model, we see that the individuals are indeed significant for the sentiment we get out of the tweet. This is what we expected, since this is how we created the data. A summary of the model reveals that the means of the groups in the model are indeed the means that we set of the distribution. This can be seen in the coefficient section below. The synthetic Trump data has an estimate around -5, Hillary close to 0 and Bernie close to 5. These are exactly the parameters that we gave the synthetic data. The p-value with a factor of 10**-16 is way lower than the needed 0.05, which shows us that the individuals have a significant effect on the resulting sentiment. The F-value of 1243 shows us that the variation between sample means is way larger than the variance within samples. This again shows that there is a big difference between the individuals. Finally, the AIC value comparison shows that model1 has a better fit of the data than model0, since its AIC value is lower.

```
#include your analysis code of synthetic data and output in the document

# model without predictor
model0 <- lm(semFrameTest$score ~ 1, data = semFrameTest)

# model with predictor
model1 <- lm(semFrameTest$score ~ semFrameTest$Candidate, data = semFrameTest)
anova(model0, model1)
```

```
## Analysis of Variance Table
##
## Model 1: semFrameTest$score ~ 1
```

```
## Model 2: semFrameTest$score ~ semFrameTest$Candidate
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    602 12529
## 2    600  2260  2     10269 1363.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model1)
```

```
##
## Call:
## lm(formula = semFrameTest$score ~ semFrameTest$Candidate, data = semFrameTest)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9238 -1.2354  0.0906  1.3900  4.9297
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           -5.1822     0.1369  -37.86   <2e-16 ***
## semFrameTest$CandidateHillary Clinton  5.3004     0.1936   27.38   <2e-16 ***
## semFrameTest$CandidateBernie Sanders  10.1045     0.1936   52.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.941 on 600 degrees of freedom
## Multiple R-squared:  0.8196, Adjusted R-squared:  0.819
## F-statistic:  1363 on 2 and 600 DF,  p-value: < 2.2e-16
```

```
AIC(model0, model1)
```

```
##        df      AIC
## model0  2 3544.692
## model1  4 2515.924
```

2.1.6.2 Linear model If we input the actual data into the model, we see that the individuals are indeed significant for the sentiment we get out of the tweet. This can be concluded by looking at the p and F-value. The p-value is way lower than 0.05, which shows us that the individuals have a significant effect on the resulting sentiment. The F-value of 13.478 shows us that the variation between sample means is way larger than the variance within samples. This again shows that there is a difference between the individuals, regarding tweet sentiment. The AIC values show us that model1 has a better fit, since its its AIC value is lower than model0.

```
#include your analysis code and output in the document

# model without predictor
model0 <- lm(semFrame$score ~ 1, data = semFrame)

# model with predictor
model1 <- lm(semFrame$score ~ semFrame$Candidate, data = semFrame)

anova(model0, model1)
```

```
## Analysis of Variance Table
##
## Model 1: semFrame$score ~ 1
## Model 2: semFrame$score ~ semFrame$Candidate
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    299 767.80
## 2    297 703.91  2    63.887 13.478 2.496e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model1)
```

```
##
## Call:
## lm(formula = semFrame$score ~ semFrame$Candidate, data = semFrame)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4.04  -1.04  -0.04   0.83   3.96
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          1.0400     0.1540   6.755 7.51e-11 ***
## semFrame$CandidateHillary Clinton   -1.0600     0.2177  -4.869 1.83e-06 ***
## semFrame$CandidateBernie Sanders    -0.8700     0.2177  -3.996 8.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.54 on 297 degrees of freedom
## Multiple R-squared:  0.08321,    Adjusted R-squared:  0.07703
## F-statistic: 13.48 on 2 and 297 DF,  p-value: 2.496e-06
```

```
AIC(model0, model1)
```

```
##        df      AIC
## model0  2 1137.286
## model1  4 1115.224
```

2.1.6.3 Post Hoc analysis If we conduct a bonferroni post hoc analysis, we see that Donald Trump has a significant difference with both Hillary and Bernie, regarding tweet sentiment. However, we also see that the tweet sentiment difference of Bernie and Hillary is not significant.

The results of the normality tests show us that the spread of tweet sentiments per individual can indeed be seen as a normal distribution. This confirms the normality assumption.

The results of the levene test show that we can indeed assume the individual normal distributions to have the same variance.

So the post-hoc analysis shows that our assumptions of the data could indeed be made.

```
#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 5.5e-06      -
## Bernie Sanders  0.00024      1.00000
##
## P value adjustment method: bonferroni
```

```
library(car)
```

```
## Loading required package: carData
```

```
tapply(semFrame$score, semFrame$Candidate, shapiro.test)
```

```
## $`Donald Trump`
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.85938, p-value = 2.676e-08
##
##
## $`Hillary Clinton`
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.91847, p-value = 1.168e-05
##
##
## $`Bernie Sanders`
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.92669, p-value = 3.247e-05
```

```
leveneTest(semFrame$score, semFrame$Candidate)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   2  14.217 1.269e-06 ***
##       297
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.1.6.4 Report section for a scientific publication The analysis focuses on the effect of an individual on the sentiment of their tweets. In the experiment, two models are created. The first model only has an

intercept and no additional information about the individual that tweeted the tweet. The second model does have this information. The experiment has shown that the addition of data about the individual significantly improves the model (F=13.478, Pr(>F)<0.001).

Moreover, a post-hoc analysis of the experiment was conducted. This showed that the sentiment differences between Donald Trump and Hillary Clinton (p<0.001) and Bernie Sanders (p<0.001) were significant. However, the sentiment difference between Hillary and Bernie (p=1) was not significant. The post-hoc analysis also showed that the model assumptions regarding normality and equal variances could be made. The Shapiro-Wilk normality tests showed that for both Trump (W = 0.85938, p-value < 0.001), Hillary (W = 0.91847, p-value < 0.001) and Bernie (W = 0.92669, p-value < 0.001) the distributions can be regarded as normal distributions. Finally, Levene's test for homogeneity of variances showed that all distributions indeed have equal variances (F=14.217, Pr(>F)<0.001).

2.1.7 Bayesian Approach

2.1.7.1 Analysis verification We fit two models, a model with only an intercept and a model with the relation between sentiment and individual added. The second model has a better fit, since its WAIC value is lower than the model without the additional relation.

We can see that the means of the synthetic data are correctly reproduced by the second model. a[1], a[2] and a[3] correspond to the synthetic Trump, Hillary and Bernie data, respectively. Also the sigma of 1.98 is almost a precise reproduction of the actual standard deviation of 2. For a[1], a[2] and a[3] the 95% credible intervals are [-5.18,-4.63], [-0.51,0.05] and [4.66,5.21], respectively.

The WAIC values show that m1 has a better fit, since its WAIC value is lower than model0.

```
## Loading required package: rstan

## Loading required package: StanHeaders

##
## rstan version 2.26.22 (Stan version 2.26.1)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
## For within-chain threading using ‘reduce_sum()‘ or ‘map_rect()‘ Stan functions,
## change ‘threads_per_chain‘ option:
## rstan_options(threads_per_chain = 1)

## Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file

## Loading required package: parallel

## Loading required package: dagitty

## rethinking (Version 2.01)

##
## Attaching package: 'rethinking'
```

```
## The following object is masked from 'package:car':
##
##     logit

## The following object is masked from 'package:stats':
##
##     rstudent
```

```
#include your analysis code of synthetic data and output in the document
da <- subset(semFrameTest, select = c(score, Candidate))
m0 <-map2stan(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a,
    a ~ dnorm(0, 10),
    sigma ~ dunif(0.001, 20)
  ), data = da, iter = 1000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
m1 <-map2stan(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a[Candidate] ,
    a[Candidate] ~ dnorm(0, 10),
    sigma ~ dunif(0.001, 20)
  ), data = da, iter = 1000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
precis(m1, depth = 2, prob = .95)
```

```
##             mean         sd       2.5%      97.5%    n_eff     Rhat4
## a[1]  -5.1838473 0.14014698 -5.4583330 -4.8991079 2506.318 0.9987316
## a[2]   0.1163955 0.13515442 -0.1450215  0.3735193 2340.095 1.0002944
## a[3]   4.9212921 0.13594002  4.6539247  5.1775289 2661.317 0.9997366
## sigma  1.9446829 0.05591143  1.8377023  2.0517958 2513.010 0.9992918
```

```
compare(m0, m1)
```

```
##        WAIC       SE    dWAIC      dSE    pWAIC        weight
## m1 2515.851 33.32604    0.000       NA 3.899444  1.000000e+00
## m0 3544.252 25.20653 1028.401 35.42962 1.542206 4.846833e-224
```

2.1.7.2 Model comparison Again we make two models, one with only intercept and one with an additional relation to the individual. The WAIC shows that the model with the additional relation is better than the model with only intercept. The 95% credible intervals are [0.74,1.34] for Trump, [-0.32,0.29] for Hillary and [-0.14,0.48] for Bernie. It shows that there is no overlap between Trump and any other in the credible intervals, but there is some overlap between Hillary and Bernie.

The WAIC values show that m1 has a better fit, since its WAIC value is lower than model0.

```
#include your code and output in the document
dat <- subset(semFrame, select = c(score, Candidate))
m0 <-map2stan(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a ,
    a ~ dnorm(0, 10),
    sigma ~ dunif(0.001, 20)
  ), data = dat, iter = 1000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
m1 <-map2stan(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a[Candidate] ,
    a[Candidate] ~ dnorm(0, 10),
    sigma ~ dunif(0.001, 20)
  ), data = dat, iter = 1000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
precis(m1, depth = 2, prob = .95)
```

```
##              mean         sd       2.5%     97.5%    n_eff     Rhat4
## a[1]   1.03753918 0.15651419  0.7417658 1.3488103 2506.092 1.0013436
## a[2]  -0.01719681 0.15659192 -0.3272905 0.2838630 2766.627 1.0007434
## a[3]   0.16249491 0.15498751 -0.1474189 0.4694793 2775.875 0.9993421
## sigma  1.54687412 0.06393133  1.4308835 1.6795764 2674.765 0.9983989
```

```
compare(m0, m1)
```

```
##        WAIC       SE    dWAIC      dSE    pWAIC       weight
## m1 1115.899 29.12571  0.00000       NA 4.505856 9.999872e-01
## m0 1138.432 34.20888 22.53305 10.47565 3.072309 1.279394e-05
```

2.1.7.3 Comparison individual/organisation pair We compare the three possible pairs, and we see basically the same results as in the frequentist approach. The 95% credible intervals with Trump both do not include 0, which makes it likely that the tweet sentiments of the others are not equal to those of Trump. We can also see that the 95% credible interval of Hillary and Bernie does contain 0, which still maintains the possibility that there is not a difference in tweet sentiments between these two individuals.

```
#include your code and output in the document
post <- extract.samples(m1, n=1e5)
diffhill_trump <- post$a[,1] - post$a[,2]
diffhill_bernie <- post$a[,2] - post$a[,3]
difftrump_bernie <- post$a[,1] - post$a[,3]
PI(diffhill_trump, prob = 0.95 )
```

```
##        3%       98%
## 0.6389805 1.4977688
```

```
PI(diffhill_bernie, prob = 0.95 )
```

```
##         3%        98%
## -0.6138551  0.2505533
```

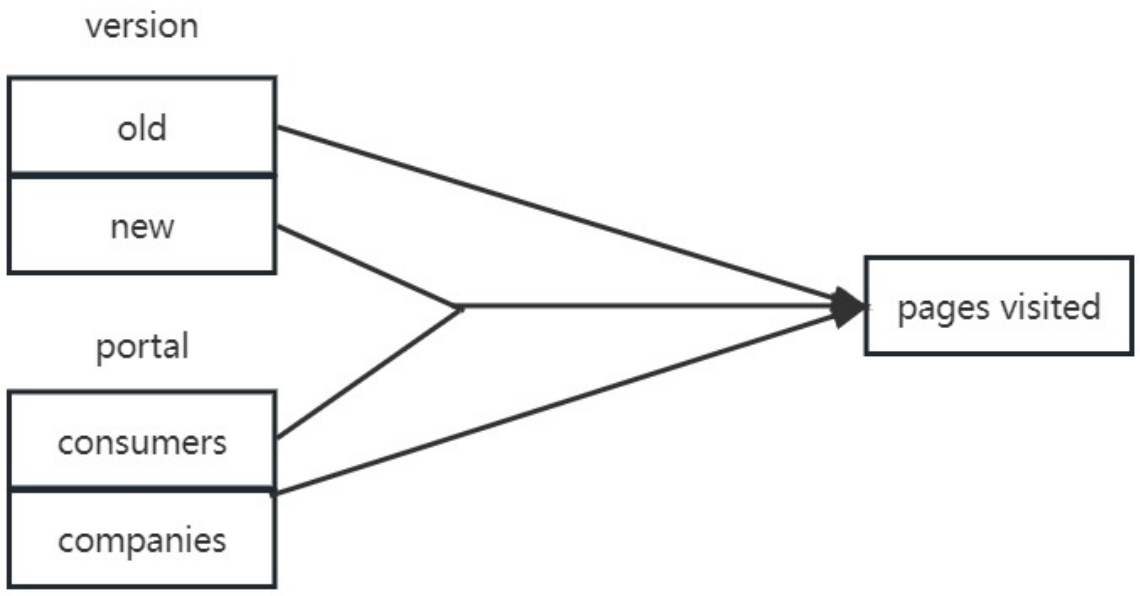
```
PI(difftrump_bernie, prob = 0.95 )
```

```
##        3%       98%
## 0.4524721 1.3237344
```

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

2.2.1 Conceptual model

Make a conceptual model underlying this research question



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.

I adopt linear regression model o fit the data, which can be expressed as follows:

y = β_0 + β_1 * version + β_2 * portal + β_3 * version * portal + ϵ

- y represents the number of page visits, which is assumed to follow a Gaussian distribution.
- version is a binary variable (0 for the old version, 1 for the new version) indicating the website version.
- portal is a binary variable (0 for consumers, 1 for companies) indicating the web portal entry.
- version * portal represents the interaction term between the website version and portal.
- β_0 is intercept and β_1, β_2, and β_3 are the coefficients for the three terms. I adopt Cauchy distribution for them assuming weakly informative prior, which has a large scale value has a gentle slope, letting data in the more extreme region still be of influence if the likelihood is strong here
- ϵ represents the error term, assumed to follow a Gaussian distribution, ϵ $Normal(0, \sigma)$.

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.

```
#include your code for generating the synthetic data

# Set the seed for reproducibility
set.seed(1)

# Specify the sample size
n <- 100

# Create the independent variables
version <- rep(c(0, 1), each = n/2)
portal <- rep(c(0, 1), times = n/2)

# Generate the interaction effect
interaction <- version * portal

# the values of the coefficients of the linear model
beta0 <- 2.5     # Intercept
beta1 <- 1.5     # Coefficient for version
beta2 <- 0.8     # Coefficient for portal
beta3 <- 0.7     # Coefficient for interaction

# Generate the dependent variable (number of page visits)
page_visits <- beta0+beta1*version+beta2*portal+beta3*interaction+rnorm(n)

# Combine the variables into a data frame
data <- data.frame(version, portal, interaction, page_visits)

# View the first few rows of the synthetic data set
head(data)
```

```
##   version portal interaction page_visits
## 1       0      0           0    1.873546
## 2       0      1           0    3.483643
## 3       0      0           0    1.664371
## 4       0      1           0    4.895281
## 5       0      0           0    2.829508
## 6       0      1           0    2.479532
```

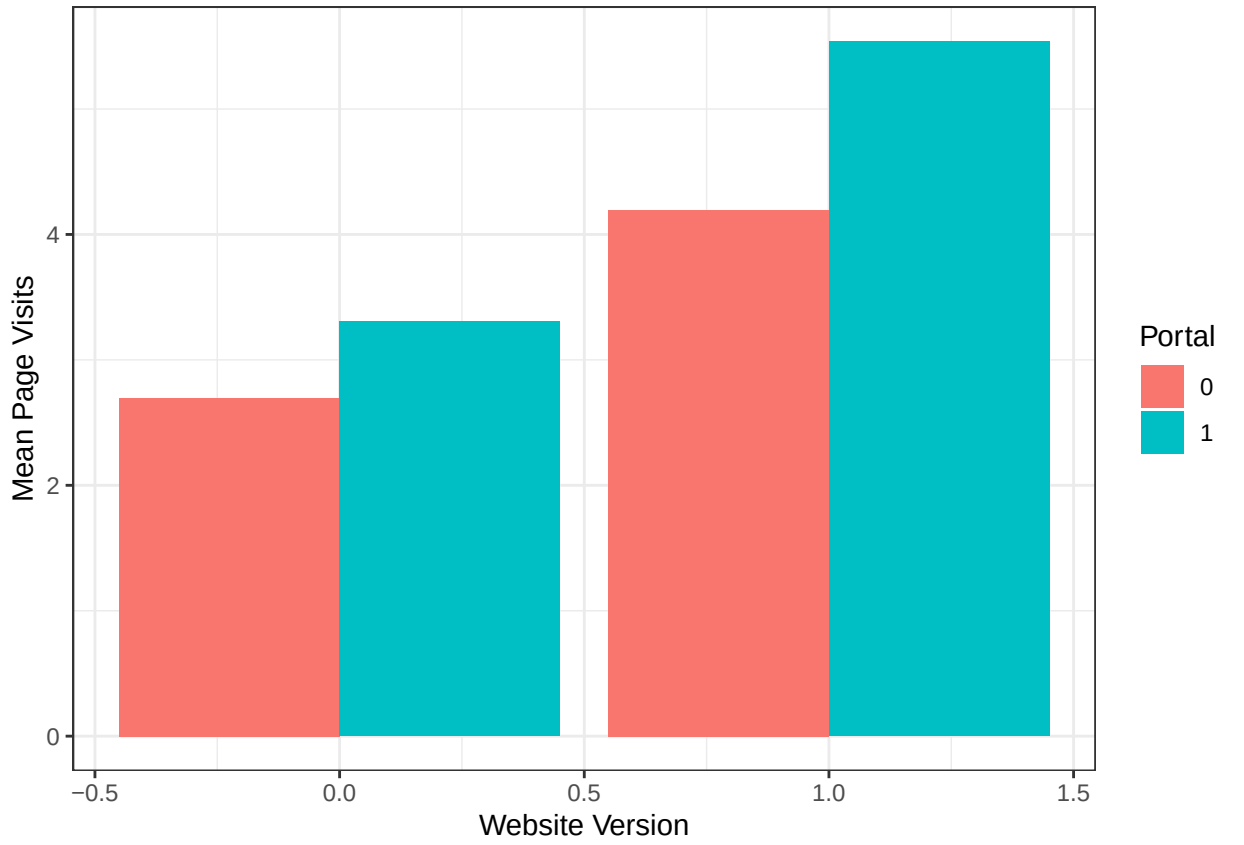
2.2.4 Visual inspection

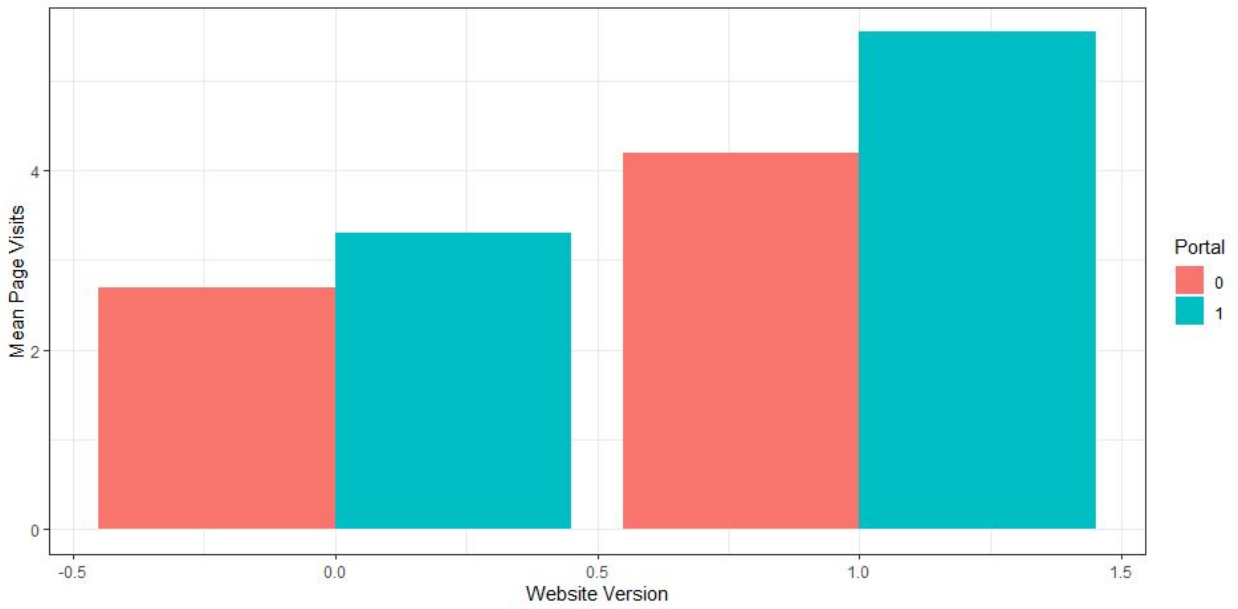
Graphically examine the mean page visits for the four different conditions. Give a short explanation of the figure.

```
#include your code and output in the document

# Load the required library
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':
##
##     annotate
```

```
# Compute the mean page visits for each condition
mean_data <- aggregate(page_visits ~ version + portal, data, mean)

# Create a grouped bar plot
ggplot(mean_data, aes(x = version, y = page_visits, fill = factor(portal))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Website Version", y = "Mean Page Visits") +
  scale_fill_discrete(name = "Portal") +
  theme_bw()
```





I conducted a simple effect analysis to examine the influence of one independent variable on different levels of another independent variable. The analysis revealed that the combination of the new version and web portal for companies resulted in a highest number of page visits. Furthermore, regardless of the website version, the portal for companies showed a higher number of page visits compared to the portal for consumers. Additionally, irrespective of the portal, the old version exhibited fewer page visits compared to the new version.

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.

```
#include your analysis code of synthetic data and output in the document

# Fit the linear regression model
model <- lm(page_visits ~ version + portal + interaction, data = data)

# Print the model summary
summary(model)
```

```
##
## Call:
## lm(formula = page_visits ~ version + portal + interaction, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.21954 -0.63146 -0.02511  0.56114  2.20663
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.6961     0.1815  14.850  < 2e-16 ***
## version       1.4989     0.2567   5.838 7.16e-08 ***
```

```
## portal        0.6088     0.2567   2.371   0.0197 *
## interaction   0.7359     0.3631   2.027   0.0455 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9077 on 96 degrees of freedom
## Multiple R-squared:  0.5911, Adjusted R-squared:  0.5784
## F-statistic: 46.26 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
# Calculate AIC
AIC <- AIC(model)

# Calculate the number of parameters
k <- length(coef(model))

# Calculate the AICc
n <- nrow(data)
AICc <- AIC + (2 * k * (k + 1)) / (n - k - 1)


# Print the results
cat("Coefficients:\n")
```

```
## Coefficients:
```

```
print(coef(model))
```

```
## (Intercept)     version      portal interaction
##   2.6960603   1.4989306   0.6087760   0.7358952
```

```
cat("\nAIC:", AIC)
```

```
##
## AIC: 270.3466
```

```
cat("\nAICc:", AICc)
```

```
##
## AICc: 270.7676
```

```
#result:

#Coefficients:
#            Estimate Std. Error t value Pr(>|t|)
#(Intercept)   2.6961     0.1815  14.850  < 2e-16 ***
#version       1.4989     0.2567   5.838 7.16e-08 ***
#portal        0.6088     0.2567   2.371   0.0197 *
#interaction   0.7359     0.3631   2.027   0.0455 *

#F-statistic: 46.26 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
#AIC: 270.3466

#AICc: 270.7676

#interpretation
#1. All coefficients are comparable to original values and statistically
#significant.
#2. The p-value is reported as "< 2.2e-16", which is essentially zero. This
#extremely low p-value indicates strong evidence against the null hypothesis,
#suggesting that the predictors collectively have a significant effect on the
#outcome variable (page visits).
#3. The AICc value of 270.3466 suggests that the fitted linear regression model
#has a relatively good fit to the data compared to alternative 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.

```
#include your code and output in the document

#include your analysis code of synthetic data and output in the document

#set work path

#read data and add interaction
web_data <- read.csv("data/webvisit0.csv")
web_data$interaction <- web_data$version * web_data$portal

# Fit the linear regression model
model0 <- lm(pages ~ version + portal + interaction, data = web_data)

# Print the model summary
summary(model0)
```

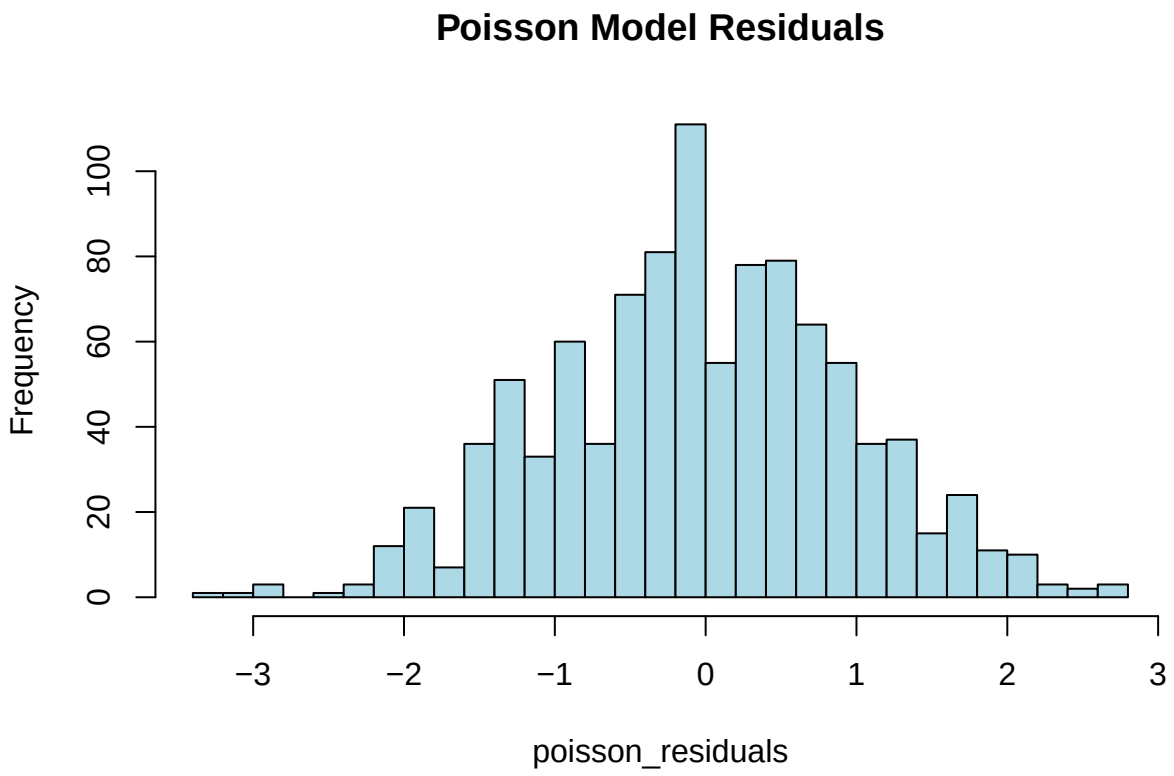
```
##
## Call:
## lm(formula = pages ~ version + portal + interaction, data = web_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.3511  -3.3511  -0.0943   3.3720  21.6489
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.6280     0.3443   57.02   <2e-16 ***
## version      -7.6239     0.4898  -15.56   <2e-16 ***
## portal       13.4663     0.4898   27.49   <2e-16 ***
## interaction  29.8808     0.6888   43.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.443 on 996 degrees of freedom
```

```
## Multiple R-squared:  0.9036, Adjusted R-squared:  0.9033
## F-statistic:  3110 on 3 and 996 DF,  p-value: < 2.2e-16
```

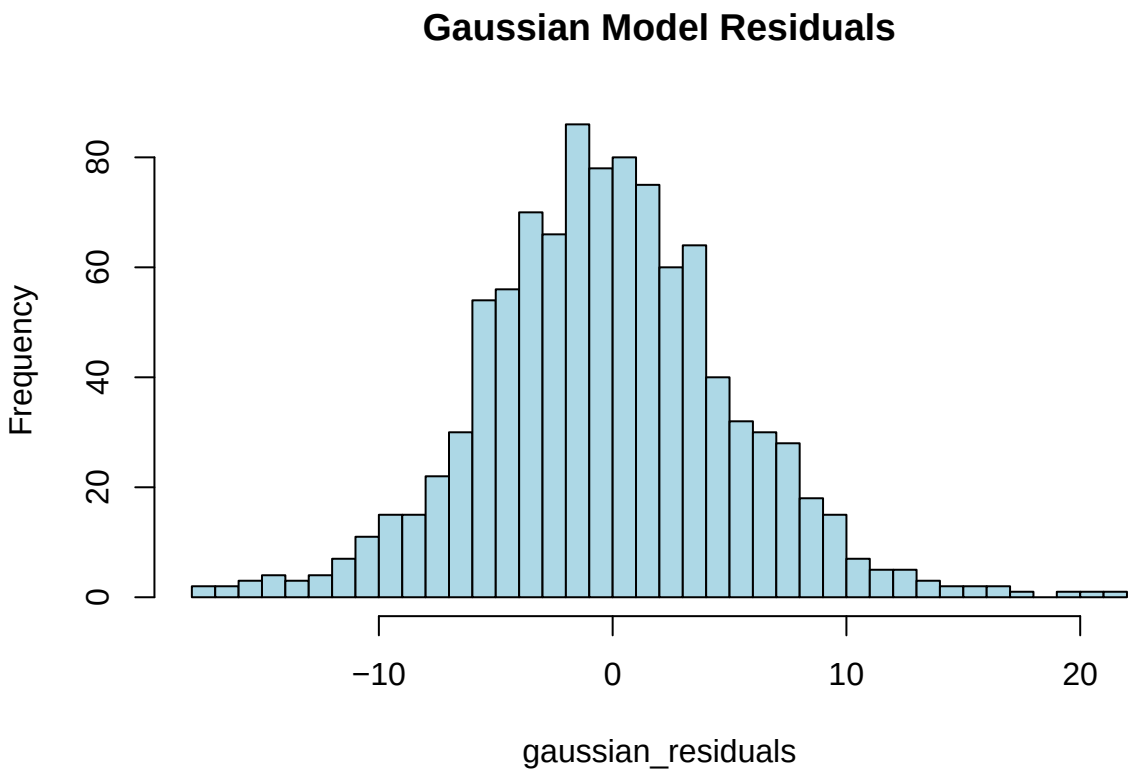
```
#result:

#Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
#(Intercept)  19.6280     0.3443   57.02   <2e-16 ***
#version      -7.6239     0.4898  -15.56   <2e-16 ***
#portal       13.4663     0.4898   27.49   <2e-16 ***
#interaction  29.8808     0.6888   43.38   <2e-16 ***

#F-statistic:  3110 on 3 and 996 DF,  p-value: < 2.2e-16

#interpretation
#1. All coefficients are statistically significant, which show that version,
#portal and interaction have a strong influence on the page visit.
#2. The p-value is reported as "< 2.2e-16", which is essentially zero. This
#extremely low p-value indicates strong evidence against the null hypothesis,
#suggesting that the predictors collectively have a significant effect on the
#outcome variable (page visits).
```

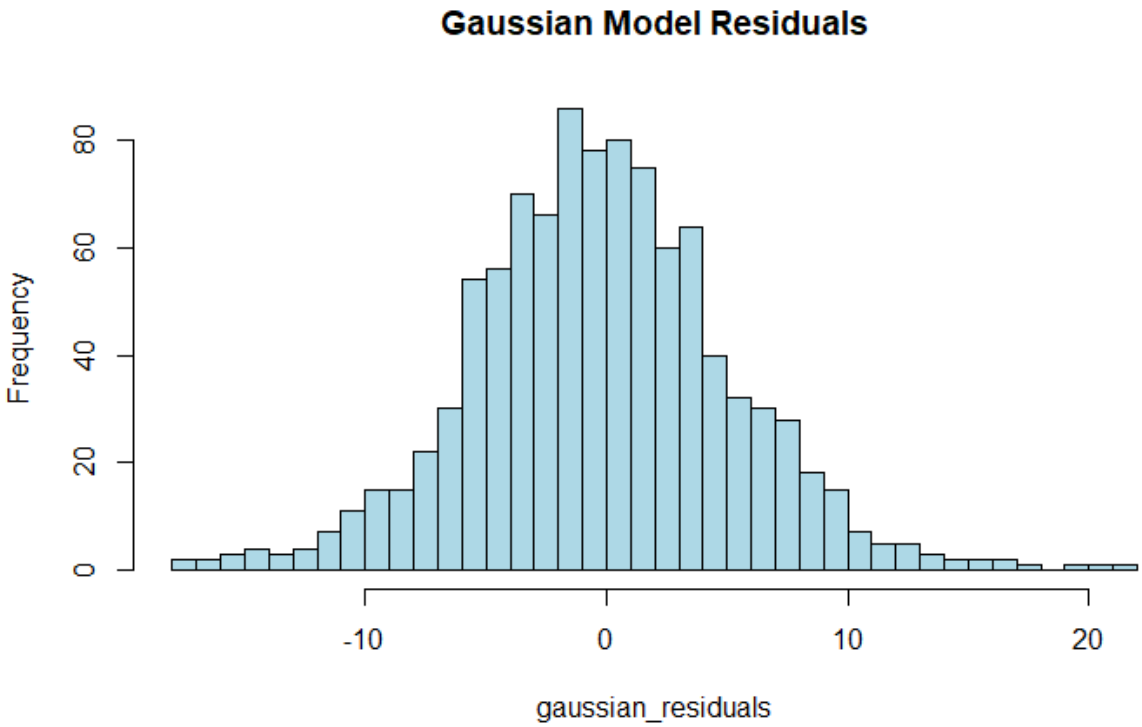
2.2.5.3 Assumption analysis Redo the analysis on the real tweet data set. This time assume a Poisson distribution for the number of page visits. For the best fitting models (Gaussian and Poisson), examine graphically the distribution of the residuals for the model that assumes Gaussian distribution and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

```
#include your code and output in the document

# Poisson regression
poisson_model <- glm(pages ~ version + portal + interaction, data = web_data,
                     family = poisson)

# Gaussian regression
gaussian_model <- model0

# Residual analysis
poisson_residuals <- resid(poisson_model)
gaussian_residuals <- resid(gaussian_model)

# Histogram of Poisson model residuals
hist(poisson_residuals, breaks = 30, col = "lightblue",
     main = "Poisson Model Residuals")
```




```
# Density plot of Gaussian model residuals
hist(gaussian_residuals, breaks = 30, col = "lightblue",
     main = "Gaussian Model Residuals")
```


Gaussian Model Residuals
Frequency
0
20
40
60
80
−10
0
10
20
gaussian_residuals


Gaussian Model Residuals
Frequency
0
20
40
60
80
-10
0
10
20
gaussian_residuals

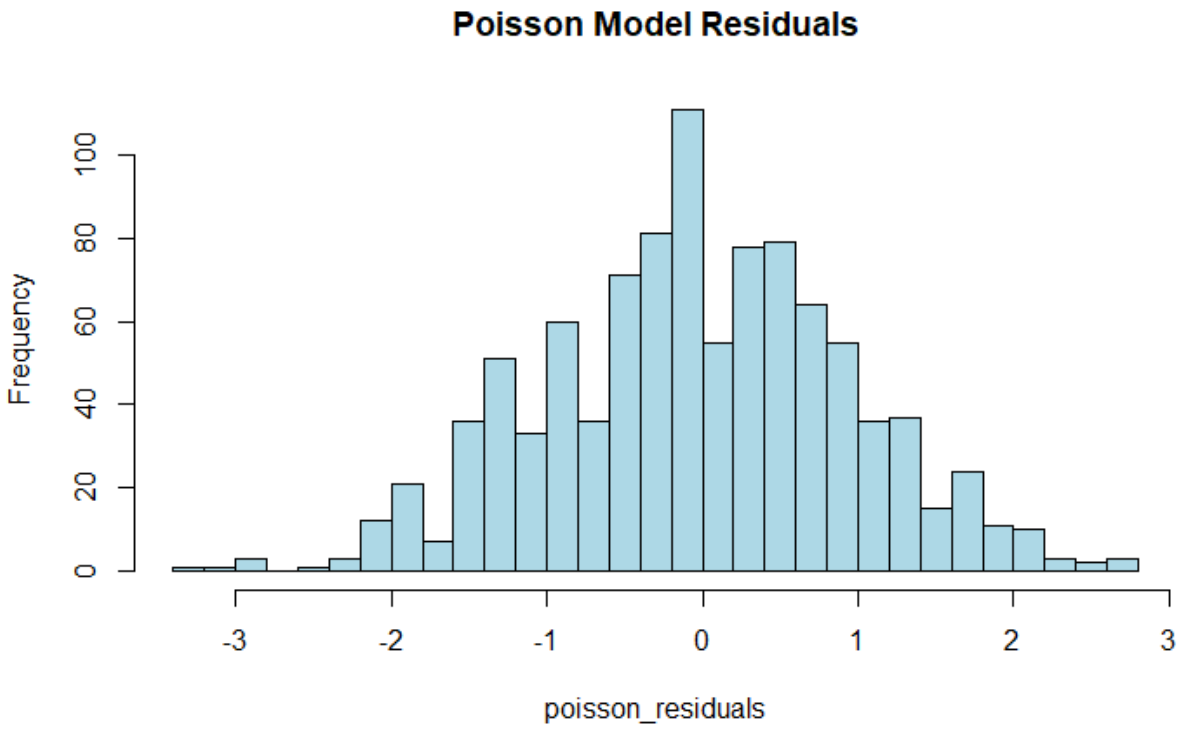


Figure 1: poisson

Both the Gaussian and Poisson residuals appear to follow a normal distribution, which suggests that the Gaussian model is a better fit for the data.

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.

```
#include your code and output in the document

library(rethinking)
summary(poisson_model)
```

```
##
## Call:
## glm(formula = pages ~ version + portal + interaction, family = poisson,
##     data = web_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.97696    0.01428  208.54   <2e-16 ***
## version     -0.49171    0.02335  -21.06   <2e-16 ***
## portal       0.52240    0.01810   28.86   <2e-16 ***
## interaction  1.00605    0.02717   37.03   <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: 9959.66  on 999  degrees of freedom
## Residual deviance:  970.17  on 996  degrees of freedom
## AIC: 6057.5
##
## Number of Fisher Scoring iterations: 4
```

```
#interaction  1.00605    0.02717   37.03   <2e-16 ***
#The p-value is reported as "< 2.2e-16", which is essentially zero.
#This extremely low p-value indicates strong evidence against the null
#hypothesis, suggesting that the interaction have a significant effect
#on the page visits.

library(pander)

# create two contrasts and combine them and associate the contrast to a variable
#merge two factors
web_data$simple <- interaction(web_data$version, web_data$portal)
levels(web_data$simple) #to see the level in the new factor
```

```
## [1] "0.0" "1.0" "0.1" "1.1"
```

```
contrastOld <-c(1,-1,0,0)
contrastNew <-c(0,0,1,-1)
```

```
SimpleEff <- cbind(contrastOld,contrastNew)
contrasts(web_data$simple) <- SimpleEff #now we link the two contrasts with
#the version

# we fit a linear model on the data, using this two-level variable
#as an independent factor.
simpleEffectModel <-lm(pages ~ simple , data = web_data, na.action = na.exclude)
pander(summary.lm(simpleEffectModel))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	30.02	0.1722	174.3	0
simplecontrastOld	3.812	0.2449	15.56	4.691e-49
simplecontrastNew	-11.13	0.2421	-45.96	2.195e-248
simple	28.41	0.3444	82.48	0

Table 2: Fitting linear model: pages ~ simple

Observations	Residual Std. Error	R^2	Adjusted R^2
1000	5.443	0.9036	0.9033

```
# result:
#------------------------------------------------------------------------
#                    Estimate   Std. Error   t value    Pr(>|t|)
#----------------------- ---------- ------------ --------- ------------
#    **(Intercept)**       30.02       0.1722      174.3        0
#
# **simplecontrastOld**    3.812       0.2449      15.56     4.691e-49

# **simplecontrastNew**    -11.13      0.2421     -45.96     2.195e-248

#      **simple**           28.41       0.3444      82.48        0
#------------------------------------------------------------------------

# It revealed a significant (t = 15.56, p. < 0.01) difference for old version
#in page visits, and also a significant effect (t = -45.96,p. < 0.01 )
#was found for the onew version in page visits.
```

2.2.5.5 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.

Paper: Effects of different real-time feedback types on human performance in high-demanding work conditions

Result:

Table 10
Effects of feedback types on SUS scores; Model 4 including the main effects, 2-way interactions and 3-way interaction.

Effects	*df*	Sum of squares	*F*	*p*
HR	1	1017.89	7.75	0.006**
Performance	1	4.72	0.04	0.850
Error	1	307.62	2.34	0.128
HR × performance	1	26.81	0.20	0.652
HR × error	1	1265.88	9.64	0.002**
Performance × error	1	307.62	2.34	0.128
HR × performance × error	1	132.84	1.01	0.316

** $p < 0.01$.

The table shows the simple effect of HR, performance and error as well as the 2-way/3-way interaction between/among them. We can see from the table that there is a significant two-way interaction effect between HR and Error.(p<0.002) Also, the HR itself is a significant effect(p<0,006).

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.

```
#include your analysis code of synthetic data and output in the document
library(rethinking)

model_bay_fake <- map(
  alist(
    page_visits ~ dnorm(mu, sigma),
    mu <- beta0 + beta1*version + beta2*portal + beta3*interaction,
    beta0 ~ dnorm(0, 10),
    beta1 ~ dnorm(0, 10),
    beta2 ~ dnorm(0, 10),
    beta3 ~ dnorm(0, 10),
    sigma ~ dcauchy(0, 2.5)
  ),
  data = data,
  start = list(beta0 = 2.5, beta1 = 1.5, beta2 = 0.8, beta3 = 0.7, sigma = 1)
)


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

```
##            mean        sd       2.5%    97.5%
## beta0 2.6956423 0.1775695 2.34761243 3.043672
## beta1 1.4991075 0.2510816 1.00699650 1.991218
## beta2 0.6092341 0.2510817 0.11712314 1.101345
## beta3 0.7354461 0.3550131 0.03963326 1.431259
## sigma 0.8884077 0.0626871 0.76554325 1.011272
```

```
waic <- WAIC(model_bay_fake)
waic
```

```
##       WAIC      lppd  penalty  std_err
## 1 271.6464 -130.2559 5.567342 15.19836
```

```
#Result
#      mean   sd 2.5% 97.5%
#beta0 2.70 0.18 2.35  3.04
#beta1 1.50 0.25 1.01  1.99
#beta2 0.61 0.25 0.12  1.10
#beta3 0.74 0.36 0.04  1.43
#sigma 0.89 0.06 0.77  1.01

#      WAIC      lppd  penalty  std_err
#1 271.2522 -130.2396 5.386505 15.16342

# The estimated coefficients of the synthetic data closely match the original
#coefficients used to generate the data, with portal, version and interaction
#all positively affect the page visit.
```

2.2.6.2 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 Poisson distribution for the number of page visits. Justify the priors.

Model:

Pages ~ Poisson(lambda)

lambda = exp(beta0 + beta1 * Version + beta2 * Portal + beta3 * Interaction)

Priors:

Because there is limited prior information or no strong prior beliefs, I use weakly informative priors that allow the data to have a larger influence on the posterior results.

- beta0 ~ Normal(0, 10)
- beta1 ~ Normal(0, 10)
- beta2 ~ Normal(0, 10)
- beta3 ~ Normal(0, 10)

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).

```r
#include your code and output in the document

#set work path

#read data and add interaction
web_data <- read.csv("data/webvisit0.csv")
web_data$interaction <- web_data$version * web_data$portal

library(rethinking)


# Model formulation
model_bay_web <- map(
  alist(
    pages ~ dpois(lambda),
    log(lambda)<- beta0 +beta1 * version + beta2 * portal + beta3 * interaction,
    beta0 ~ dnorm(0, 10),
    beta1 ~ dnorm(0, 10),
    beta2 ~ dnorm(0, 10),
    beta3 ~ dnorm(0, 10)
  ),
  data = web_data,
)

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

```
##             mean         sd       2.5%      97.5%
## beta0  2.9771152 0.01427433  2.9491380  3.0050924
## beta1 -0.4915666 0.02334657 -0.5373250 -0.4458081
## beta2  0.5222265 0.01809963  0.4867518  0.5577011
## beta3  1.0058816 0.02716356  0.9526420  1.0591212
```

```r
waic <- WAIC(model_bay_web)
waic
```

```
##       WAIC     lppd  penalty  std_err
## 1 6057.409 -3024.84 3.864437 45.57437
```

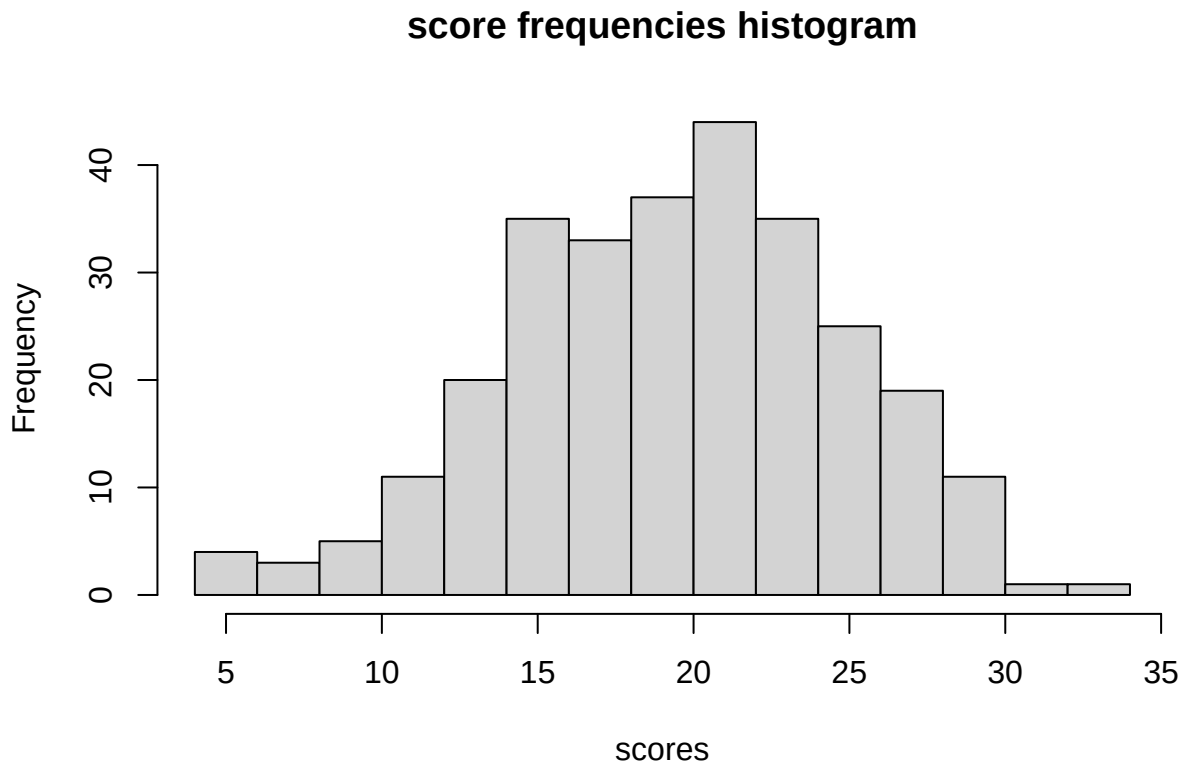
```r
# Results:
#       mean   sd  2.5% 97.5%
#beta0  2.98 0.01  2.95  3.00
#beta1 -0.49 0.02 -0.54 -0.45
#beta2  0.52 0.02  0.49  0.56
#beta3  1.01 0.03  0.95  1.06

#     WAIC      lppd  penalty  std_err
#1 6057.62 -3024.845 3.965474 45.57388

#the analysis suggests that the version, portal, and interaction variables have
#significant effects on the number of page visits.
#The version and portal variables have opposite effects.
#The interaction variable shows a stronger positive effect on page visits.
#The model's WAIC and lppd indicate reasonable fit to the data,
```

3 Part 3 - Multilevel model

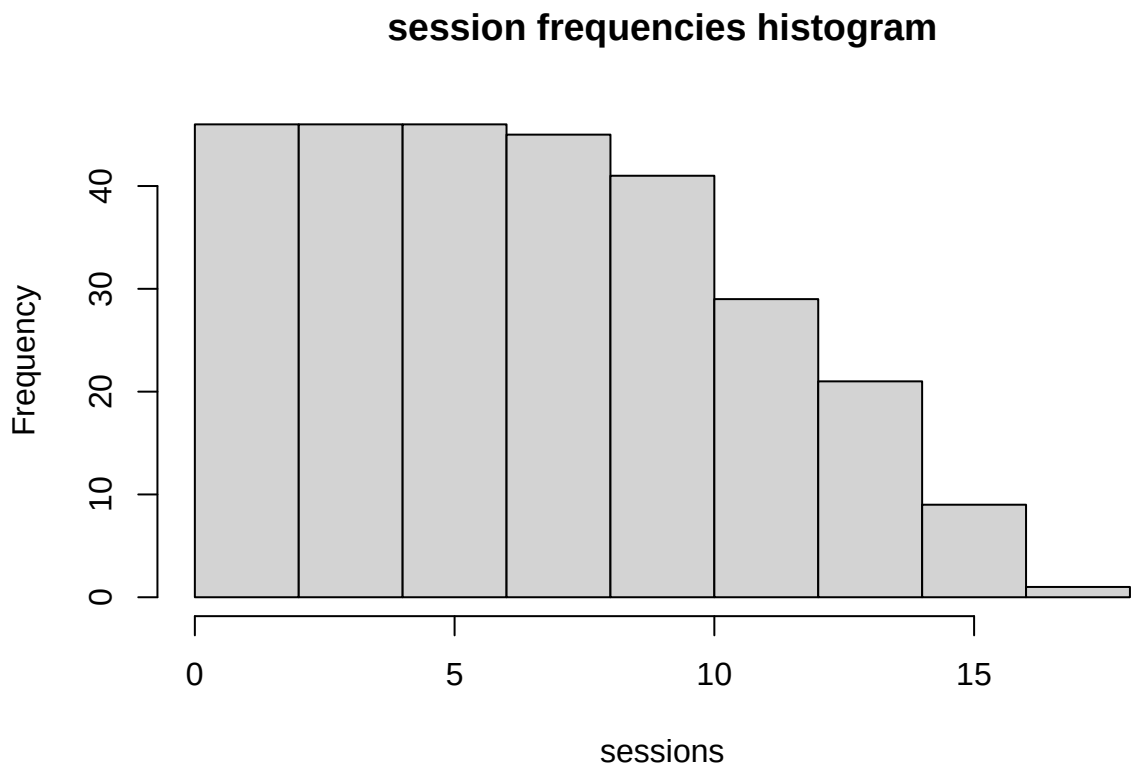
3.1 Visual inspection

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

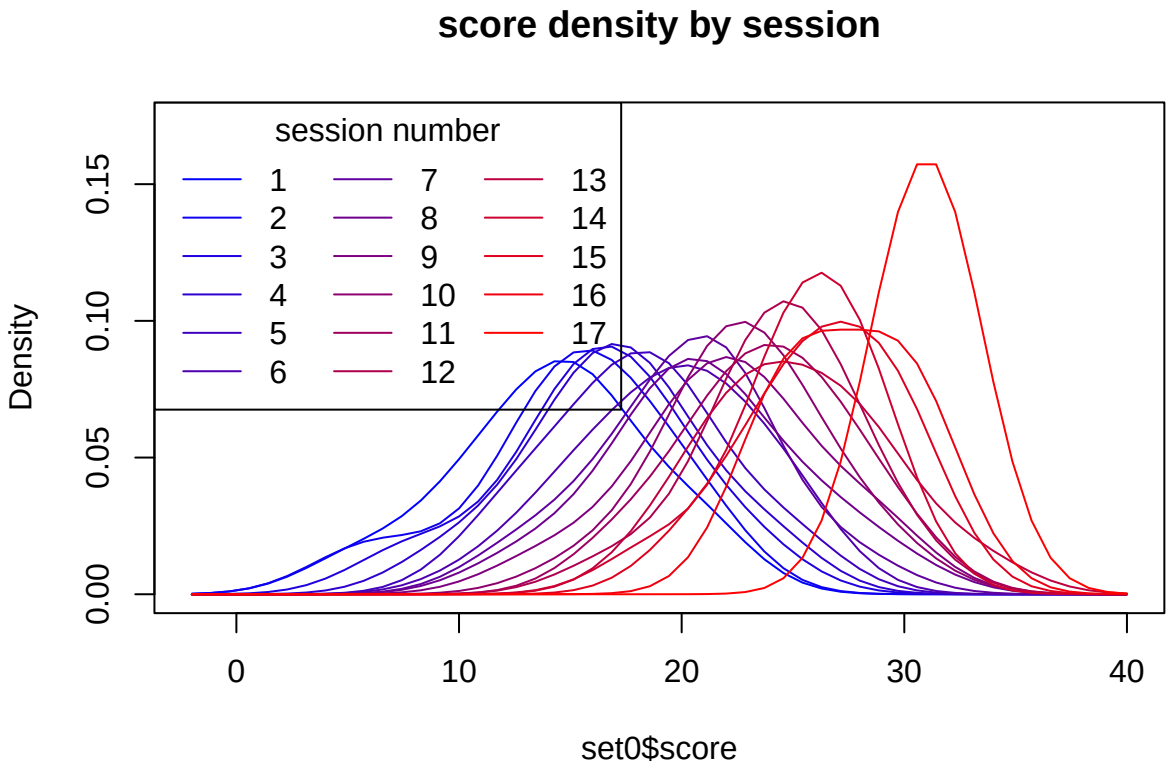
```
#include your code and output in the document
library(sm)
library(car)

set0 <- read.csv("data/set1.csv")
#stem leaf plot is useless in this case
#stem(set1$score, atom = 1e-04)

hist(set0$score, xlab="scores", main="score frequencies histogram")
```



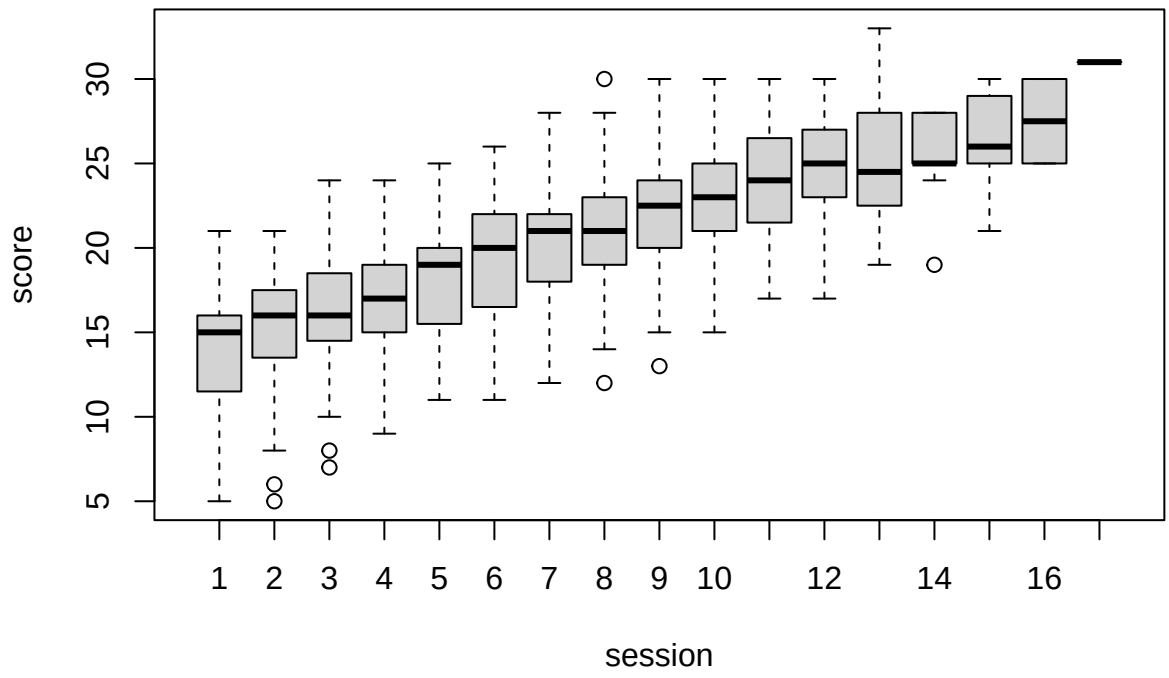

```
hist(set0$session, xlab="sessions", main="session frequencies histogram")
```



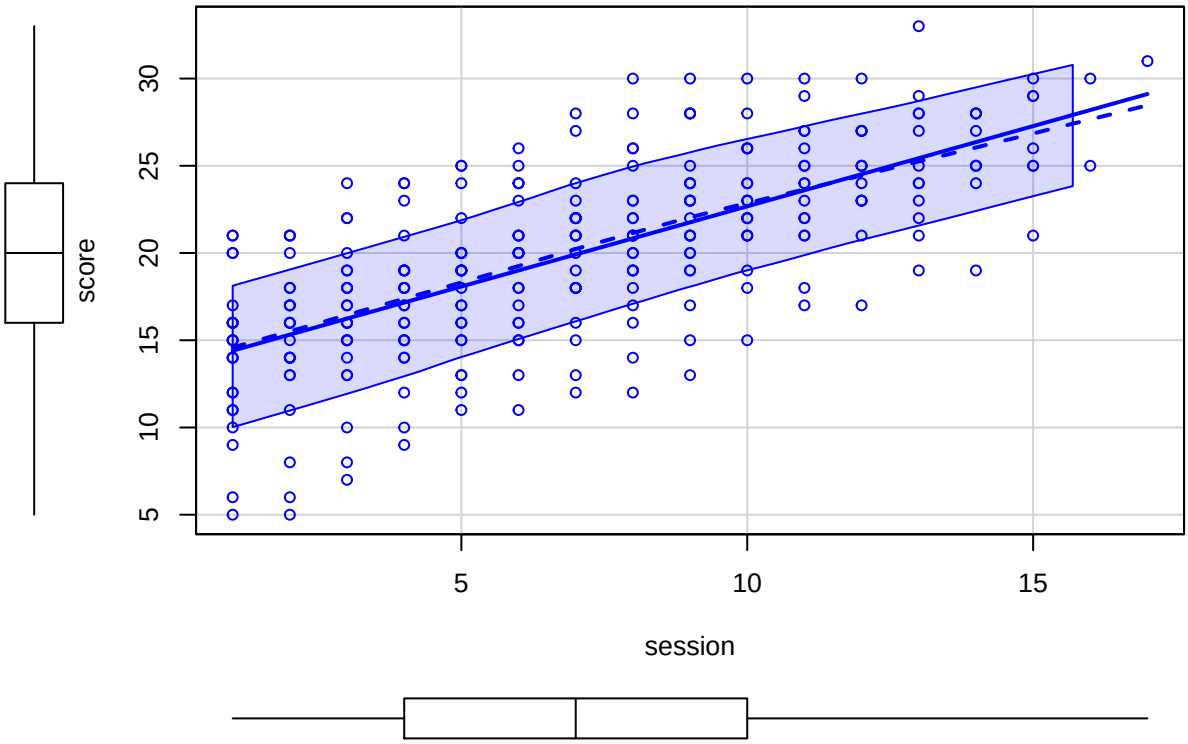

```
# Define color gradient
color_range <- colorRampPalette(c("blue", "red"))

# Assign colors based on session number
color_vector_temp <- color_range(length(unique(set0$session)))
color_vector <- color_vector_temp[as.numeric(unique(set0$session))]
lty_vector <- rep(1, each=length(unique(set0$session)))

note_text <- "17"



den <- sm.density.compare(set0$score, group = set0$session, h=2.5,
                          col= color_vector, lty=lty_vector)
title(main="score density by session")
#text(x=0, labels ="17", adj=c(-16,-18))

legend("topleft", den$levels, lty=den$lty, lwd=den$lwd, y.intersp=1, ncol=3,
       col=den$col, title="session number")
```




```
boxplot(score ~ session, data=set0)
```




```
#boxplot(session ~ score, data=set0)
scatterplot(score ~ session, data=set0)
```



the first two histograms are used to understand the distribution of the dependent and independent variables. the score approximately seemed to follow a normal distribution whereas the session index seem to contain less participants as it increases, by the 17th session there's only one participant.

The third graph offer a visual inspection of the distribution of scores by session number which follows a color range from blue to red. This makes it easy to see that as the score increases the color shifts from blue to red which indicates that later sessions get higher scores.

Then I plot a box plot and a scatter plot to visualize the relationship between score and session index and indeed there seem to be a pretty clear positive effect that the session index as on the score.

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

```
library(nlme)
library(pander)
ctrl <- lmeControl(opt='optim');

freq_m1 <- lme(score ~ 1,
```

```
          data = set0,
          random = ~ 1 | subject,
          method="ML",
          control=ctrl
          )

freq_m2 <- lme(score ~ session,
          data = set0,
          random = ~ 1 | subject,
          method="ML",
          control=ctrl
          )

sm1 <- summary(freq_m1)
sm2 <- summary(freq_m2)
pander(sm1$tTable, caption="model 1")
```

Table 3: model 1

	Value	Std.Error	DF	t-value	p-value
(Intercept)	19.8	0.7851	261	25.22	6.416e-72

```
pander(sm2$tTable, caption="model 2")
```

Table 4: model 2

	Value	Std.Error	DF	t-value	p-value
(Intercept)	13.17	0.8138	260	16.18	3.232e-41
session	0.9916	0.01589	260	62.4	1.82e-158

```
#print(sm2)
pander(anova(freq_m1, freq_m2), caption="models comparison")
```

Table 5: models comparison (continued below)

	call	Model	df	AIC	BIC
freq_m1	lme.formula(fixed = score ~ 1, data = set0, random = ~1 | subject, method = "ML", control = ctrl)	1	3	1660	1671
freq_m2	lme.formula(fixed = score ~ session, data = set0, random = ~1 | subject, method = "ML", control = ctrl)	2	4	939.2	953.8

	logLik	Test	L.Ratio	p-value
freq_m1	-827.2		NA	NA
freq_m2	-465.6	1 vs 2	723.1	2.902e-159

```
#include your code and output in the document
```

The first model only finds a fixed intercept with value 19.8 but no slope. The second model, which is a bit more complicated, takes into account the effect that session has on the score and found a slope of 0.991 and the intercept now became 13.17. the second model is able to perform much better than the first one as indicated by the AIC scores: 939.2, 1660 respectively.

3.2.2 Report section for a scientific publication

we constructed a multi-level regression model and the experiment showed a significant association between the session number and the score obtained with t(260, N=284)=15.48, p<0.0001 for the intercept of value 13.19 and t(260, N=284)=15.48, p<0.0001 for the slope of value 0.99.

the relationship between sessions and scores show significant variance across subjects in the intercept SD=4.03 (95% CI: 2.99, 5.41) and in the slope SD=0.04 (95% CI: 0.01, 0.12) and the slopes and intercepts were negatively significantly correlated, cor=-.81.

3.3 Bayesian approach

3.3.1 Model description

The most complete model to predict subjects' scores assumes that the scores are Gaussian distributed. The mean, or expected value, of the score is then modeled through a linear function which depends on: a fixed intercept a modeled with a normal prior. A varying intercept $a_{subject}$ with an adaptive prior, used to explain the variation of the intercept between subjects. a fixed coefficient b which is the slope that explains the session effect on the score. a varying coefficient $b_{subject}$ with a fixed prior, used to explain the variation of the slope for different subjects.

The prior values, Are chosen based on the previous visual inspection of the mean and previous intercepts and coefficient values from the frequentist models.

$score \sim Norm(\mu, \sigma)$ $[likelihood]$
$\mu = a + a_{subject} + (b + b_{subject}) * session$ $[linear\ model]$
$a_{subject} = Norm(0, a_\sigma)$ $[adaptive\ prior]$
$a_\sigma = HalfCouch(0, 10)$ $[hyper\ prior]$
$b_{subject} = Norm(0, 10)$ $[fixed\ prior]$
$a = Norm(15, 25)$ $[fixed\ prior]$
$b = Norm(0, 50)$ $[fixed\ prior]$
$\sigma = Norm(0, 10)$ $[fixed\ prior]$

3.3.2 Model comparison

Compare models with with increasing complexity.

```
library(rethinking)
#fixed intercept
#this model learns fixed mu for each subject and a fixed intercept
bays_m1 <- map2stan(
  alist(
    #likelihood
    score ~ dnorm(mu, sigma),
```

```
    #linear model
    mu <- a + a_subject[subject],

    #fixed priors
    a_subject[subject] ~ dnorm(0, 10),
    a ~ dnorm(15, 25),
    sigma ~ dcauchy(0,10)

  ),
  data=set0,iter = 10000,
  chains=4,log_lik = TRUE,
  cores=4,control = list(adapt_delta=.99)
)
```

```
## Computing WAIC
```

```
#fixed intercept with subject adaptive prior
bays_m2<- map2stan(
  alist(
    #likelyhood
    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, 7),

    #fixed prior
    a ~ dnorm(15, 25),
    sigma ~ dcauchy(0,10)
  ),
  data=set0,iter = 10000,
  chains=4,log_lik = TRUE,
  cores=4, control = list(adapt_delta=.99)
)
```

```
## Computing WAIC
```

```
#adding random slope by subject
bays_m3<- map2stan(
  alist(
    #likelihood
    score ~ dnorm(mu, sigma),

    #linear model
    mu <- a + a_subject[subject] + (b_subject[subject]+b)*session,
```

```
    #adaptive prior
    a_subject[subject] ~ dnorm(0, a_sigma_subject),
    b_subject[subject] ~ dnorm(0, 10),

    #hyper prior
    a_sigma_subject ~ dcauchy(0, 10),

    #fixed prior
    a ~ dnorm(15, 25),
    b ~ dnorm(0, 50),
    sigma ~ dcauchy(0,10)
  ),
  data=set0,iter = 10000,
  chains=4,log_lik = TRUE,
  cores=4, control = list(adapt_delta=.99)
)
```

```
## Warning: There were 19999 transitions after warmup that exceeded the maximum treedepth. Increase max
## https://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quanti
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess

## Computing WAIC
```

```
precis(bays_m1, depth=2, prob=.95)
```

```
##                     mean        sd         2.5%     97.5%     n_eff    Rhat4
## a_subject[1]   1.8901792 2.3737153  -2.76010946  6.504720  611.4867 1.006648
## a_subject[2]  -5.8269941 2.4523784 -10.65330275 -1.015012  663.7699 1.005384
## a_subject[3]   2.7731884 2.3606498  -1.85986416  7.348979  614.1465 1.007148
## a_subject[4]  -1.6260609 2.4093963  -6.36680835  3.107863  641.5396 1.006740
## a_subject[5]  -1.9081954 2.3613717  -6.52724733  2.676325  612.3506 1.006457
## a_subject[6]  -1.0004938 2.6119580  -6.12514094  4.083053  743.7549 1.005659
## a_subject[7]  -9.5383942 2.5255212 -14.51054684 -4.582517  684.7786 1.006121
## a_subject[8]   1.7847579 2.3594325  -2.87023475  6.376309  609.3388 1.007177
## a_subject[9]   3.6959290 2.3453516  -0.89917929  8.278687  593.8027 1.006908
## a_subject[10]  4.7435654 2.4390807  -0.03772508  9.537307  664.3305 1.006407
## a_subject[11] -1.0399571 2.3573007  -5.63823969  3.591487  606.5329 1.007194
## a_subject[12]  6.6429041 2.4061632   1.90934837 11.405600  647.5288 1.006588
## a_subject[13]  2.0572272 2.4381686  -2.76894224  6.843112  653.0722 1.006001
## a_subject[14] -2.3130198 2.4729294  -7.21150771  2.530985  661.9405 1.006811
## a_subject[15]  2.7695710 2.5103292  -2.13214450  7.657905  693.9288 1.006593
## a_subject[16] -0.8610652 2.3803197  -5.53185354  3.802298  631.9773 1.006557
```

```
## a_subject[17] -0.9886351 2.3765584  -5.64391082  3.699173  626.3979 1.006875
## a_subject[18] -0.2907721 2.4919639  -5.17097670  4.556745  676.6260 1.006344
## a_subject[19] -6.3260491 2.3673730 -10.93364963 -1.695226  628.6393 1.006582
## a_subject[20]  0.1592489 2.4767871  -4.66443454  5.008593  671.3287 1.006611
## a_subject[21]  0.8716818 2.4011360  -3.82094221  5.547138  643.2288 1.006608
## a_subject[22]  0.0137823 2.4297889  -4.78930633  4.800411  653.1548 1.006542
## a_subject[23]  5.5672617 2.4703870   0.67320639 10.391635  653.1326 1.006583
## a             19.7354463 2.1526666  15.53297790 23.911318  507.8462 1.007988
## sigma          4.0666403 0.1788541   3.73475456  4.439282 7127.0635 1.000644
```

```
precis(bays_m2, depth=2, prob=.95)
```

```
##                      mean        sd         2.5%      97.5%     n_eff     Rhat4
## a_subject[1]   1.68521000 1.3156055  -0.92703331  4.2870367  5351.700 1.0001144
## a_subject[2]  -5.39859662 1.4133836  -8.23223493 -2.6839152  5862.846 1.0001431
## a_subject[3]   2.51934162 1.2896347  -0.01576803  5.0500642  5386.866 1.0000621
## a_subject[4]  -1.57930068 1.3279752  -4.20652001  0.9845964  5508.057 1.0001700
## a_subject[5]  -1.86413781 1.2868412  -4.42013853  0.6631375  4978.108 1.0002078
## a_subject[6]  -0.94373668 1.6208460  -4.15110653  2.1857812  7837.430 1.0000397
## a_subject[7]  -8.65377028 1.5343897 -11.71420099 -5.6906730  6922.590 0.9999880
## a_subject[8]   1.58400271 1.2889538  -0.98763143  4.1328255  5382.426 1.0000837
## a_subject[9]   3.41206667 1.2433220   0.94839094  5.8820171  4965.560 1.0001502
## a_subject[10]  4.26633820 1.4039981   1.56015532  7.0512210  5756.065 1.0001299
## a_subject[11] -1.04782434 1.2641826  -3.57497909  1.4255906  5055.159 1.0001496
## a_subject[12]  6.08322795 1.3385036   3.48469958  8.7574381  5954.162 1.0001112
## a_subject[13]  1.80187870 1.3927243  -0.90762167  4.5397087  5963.334 1.0000836
## a_subject[14] -2.15773650 1.4401140  -5.02737019  0.6671864  6292.414 1.0002428
## a_subject[15]  2.41651558 1.4867076  -0.48008807  5.3380121  6772.184 1.0001163
## a_subject[16] -0.87483341 1.3039929  -3.45697881  1.6506045  5176.914 1.0001013
## a_subject[17] -0.99186849 1.2919715  -3.55541010  1.5215453  5242.269 1.0002875
## a_subject[18] -0.31884310 1.4826084  -3.25566291  2.5647253  6788.186 1.0000592
## a_subject[19] -5.99856391 1.3120751  -8.62014968 -3.4246741  4996.064 1.0001029
## a_subject[20]  0.08007795 1.4316155  -2.78704267  2.8193016  5956.102 1.0000637
## a_subject[21]  0.72953516 1.3341431  -1.93745011  3.3096737  5684.915 0.9999714
## a_subject[22] -0.04826359 1.3593083  -2.73378876  2.5931875  5858.379 0.9999937
## a_subject[23]  4.97037113 1.4456744   2.18053531  7.8498742  6859.754 1.0000168
## sigma_subject  3.83389580 0.6701758   2.74442601  5.3574088 14108.774 1.0001484
## a             19.81330989 0.8531891  18.13784408 21.5599386  2508.427 1.0003680
## sigma          4.06525506 0.1781568   3.73123735  4.4351374 17880.123 0.9999892
```

```
precis(bays_m3, depth=2, prob=.95)
```

```
##                       mean        sd         2.5%       97.5%     n_eff
## a_subject[1]     0.697006843 1.0977845  -1.42819636  2.85253165  309.2850
## a_subject[2]    -5.721849909 1.1344630  -7.98493254 -3.50337072  330.8693
## a_subject[3]     1.997275327 1.0859153  -0.10743257  4.12153949  300.7085
## a_subject[4]    -1.850619257 1.1052416  -4.05786170  0.30435067  312.1510
## a_subject[5]    -2.967795084 1.0853593  -5.12571183 -0.84164202  304.5394
## a_subject[6]     0.995091670 1.2396895  -1.43873918  3.41514321  428.5866
## a_subject[7]    -8.963034004 1.1851552 -11.30879932 -6.67593666  384.4329
## a_subject[8]     0.739322884 1.0873624  -1.40541405  2.87009796  299.2691
## a_subject[9]     0.776460394 1.0697410  -1.31227589  2.85866822  300.7829
## a_subject[10]    6.725953246 1.1386926   4.49868505  8.95033369  335.6440
```

```
## a_subject[11]   -3.232269986 1.0788547  -5.38448286 -1.14573198  305.5726
## a_subject[12]    6.280632133 1.1012595   4.10941483  8.42795980  316.7974
## a_subject[13]    1.924282535 1.1360539  -0.29719221  4.14357910  336.6862
## a_subject[14]   -1.726818378 1.1587193  -4.03652992  0.48817249  347.5641
## a_subject[15]    4.427368723 1.1836255   2.13925445  6.68976287  353.1475
## a_subject[16]   -2.191914895 1.0926627  -4.35184296 -0.10443265  311.1473
## a_subject[17]   -2.129460197 1.0897465  -4.29225734 -0.02537038  303.7402
## a_subject[18]    1.756071879 1.1731972  -0.55430218  4.02665732  375.0820
## a_subject[19]   -8.190038857 1.0920679 -10.36649830 -6.07662052  303.9051
## a_subject[20]    2.298390108 1.1532847   0.03135382  4.57241915  342.7794
## a_subject[21]    0.178958996 1.1020450  -1.99566162  2.33510991  316.6792
## a_subject[22]    0.380911616 1.1151386  -1.79887309  2.53835896  319.6672
## a_subject[23]    6.001511387 1.1548409   3.77423348  8.27905714  336.3674
## b_subject[1]    -0.046765461 2.1812898  -4.08043319  4.37932854  161.9052
## b_subject[2]    -0.018630590 2.1805691  -4.05297259  4.40215799  162.1084
## b_subject[3]    -0.162626203 2.1796614  -4.19791382  4.25134275  161.8685
## b_subject[4]    -0.116859356 2.1798734  -4.15231663  4.30644625  161.8144
## b_subject[5]    -0.131257089 2.1794966  -4.16657570  4.29354607  161.8427
## b_subject[6]     0.043983199 2.1871698  -4.02732353  4.48101701  163.0731
## b_subject[7]     0.083430630 2.1822764  -3.97148361  4.50704572  162.0280
## b_subject[8]    -0.129218812 2.1796315  -4.17076847  4.29842045  161.6035
## b_subject[9]    -0.023593230 2.1794447  -4.04964129  4.40286881  161.7270
## b_subject[10]   -0.315754423 2.1805584  -4.36343051  4.09364541  162.4263
## b_subject[11]   -0.051119943 2.1791321  -4.07934367  4.37285947  161.6690
## b_subject[12]   -0.085810602 2.1795343  -4.12649548  4.33303520  161.6696
## b_subject[13]    0.033596288 2.1810475  -4.00127921  4.44594988  161.9961
## b_subject[14]   -0.001858753 2.1815871  -4.03084704  4.40216720  161.6878
## b_subject[15]   -0.101559601 2.1824207  -4.13650031  4.31916127  162.2442
## b_subject[16]   -0.031342847 2.1788282  -4.06372649  4.40815813  161.6908
## b_subject[17]   -0.119615749 2.1790621  -4.14822839  4.30244265  161.8974
## b_subject[18]   -0.184877330 2.1821777  -4.21136826  4.22355295  162.2644
## b_subject[19]   -0.033995498 2.1796420  -4.06248959  4.40198718  161.7577
## b_subject[20]   -0.282895177 2.1814390  -4.33431258  4.14474109  162.1570
## b_subject[21]   -0.045290138 2.1793872  -4.07019823  4.39542064  161.6863
## b_subject[22]   -0.131153012 2.1802874  -4.17454776  4.28266155  161.6946
## b_subject[23]    0.040538983 2.1798026  -3.98340294  4.44567631  161.9367
## a_sigma_subject  4.403070738 0.7280871   3.22714835  6.10091842 3177.5772
## a               13.237501257 0.9540539  11.37792516 15.13637347  237.3950
## b                1.073801724 2.1788487  -3.35150080  5.10018197  161.6672
## sigma            1.011507053 0.0465322   0.92499314  1.10763288 3860.4448
##                    Rhat4
## a_subject[1]    1.004765
## a_subject[2]    1.005161
## a_subject[3]    1.005787
## a_subject[4]    1.005229
## a_subject[5]    1.005079
## a_subject[6]    1.003170
## a_subject[7]    1.003312
## a_subject[8]    1.004141
## a_subject[9]    1.004839
## a_subject[10]   1.003760
## a_subject[11]   1.004388
## a_subject[12]   1.004032
## a_subject[13]   1.004773
```

```
## a_subject[14]   1.004832
## a_subject[15]   1.004920
## a_subject[16]   1.004974
## a_subject[17]   1.004601
## a_subject[18]   1.005470
## a_subject[19]   1.004858
## a_subject[20]   1.004672
## a_subject[21]   1.003899
## a_subject[22]   1.005728
## a_subject[23]   1.004870
## b_subject[1]    1.016509
## b_subject[2]    1.016643
## b_subject[3]    1.016520
## b_subject[4]    1.016526
## b_subject[5]    1.016552
## b_subject[6]    1.016709
## b_subject[7]    1.016489
## b_subject[8]    1.016548
## b_subject[9]    1.016519
## b_subject[10]   1.016592
## b_subject[11]   1.016612
## b_subject[12]   1.016567
## b_subject[13]   1.016513
## b_subject[14]   1.016716
## b_subject[15]   1.016698
## b_subject[16]   1.016552
## b_subject[17]   1.016505
## b_subject[18]   1.016420
## b_subject[19]   1.016433
## b_subject[20]   1.016540
## b_subject[21]   1.016574
## b_subject[22]   1.016515
## b_subject[23]   1.016569
## a_sigma_subject 1.002888
## a               1.006584
## b               1.016561
## sigma           1.000044
```

```
compare(bays_m1, bays_m2, bays_m3)
```

```
##              WAIC       SE    dWAIC dSE    pWAIC        weight
## bays_m3  860.5927 22.91883   0.0000  NA 43.15478  1.000000e+00
## bays_m2 1622.4645 17.76982 761.8718  NA 19.48845 3.644563e-166
## bays_m1 1623.5540 17.54556 762.9613  NA 20.76841 2.113810e-166
```

The first two models performed very similarly (WAIC: 1623.7, 1622.6) but the third model showed a significant improvement (WAICC: 861.2) this is due to the addition of the session intercept and slope which contains most of the score variance.

3.3.2.1 scientific report we fitted a muli-level Bayesian regression model on our data and found

3.3.3 Estimates examination

the fixed intercept a is 13.04 this means that in absence of subject information during session 0 the score will be 13.04.

The fixed slope is 1.27 this means that in absence of subject information the score will increase by 1.27 for every session.

a_sigma_subject is 4.41 which means that the subject specific intercept has a standard deviation of 4.41

sigma is 1 which is the standard deviation of the score.

subject 23 is the subject with the highest average score which is on average 6.20 points above the intercept whereas subject 7 has the lowest with an intercept correction of -8.79.

subject 10 has a slope correction of -0.51 which means that he/she is the subject that improved less as session went on in relative terms.

subject 7 has a slope correction of -0.11 which means that he/she is the subject that improved the most as session went on in relative terms.