Coursework assignment A - 2022-2023

CS4125 Seminar Research Methodology for Data Science

Pietro Piccini, Yuan Tian, Rens Oude Elferink

20/06/2023

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

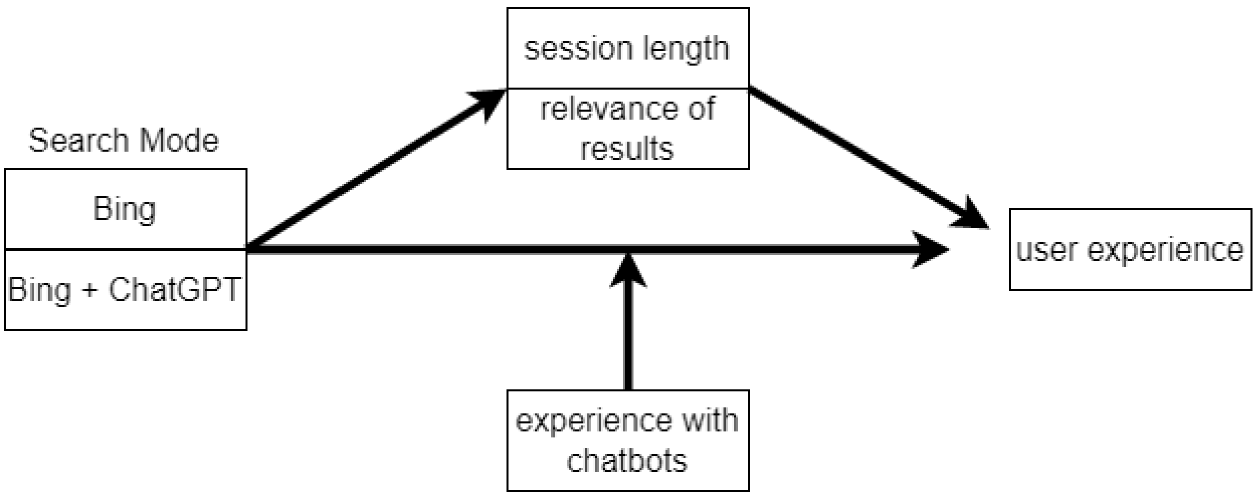


Figure 1: image1

Independent variable: Search Mode (Bing, Bing + ChatGPT) Dependent variable: User experience 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

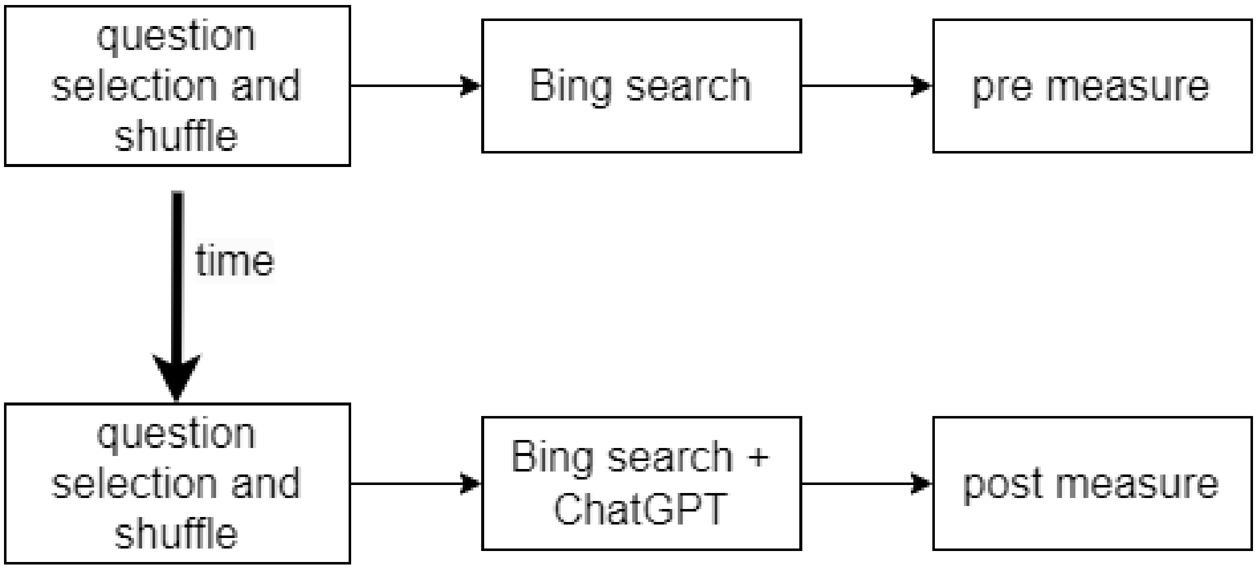

question selection and shuffle
Bing search
pre measure
time
question selection and shuffle
Bing search + ChatGPT
post measure

Figure 2: image

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.

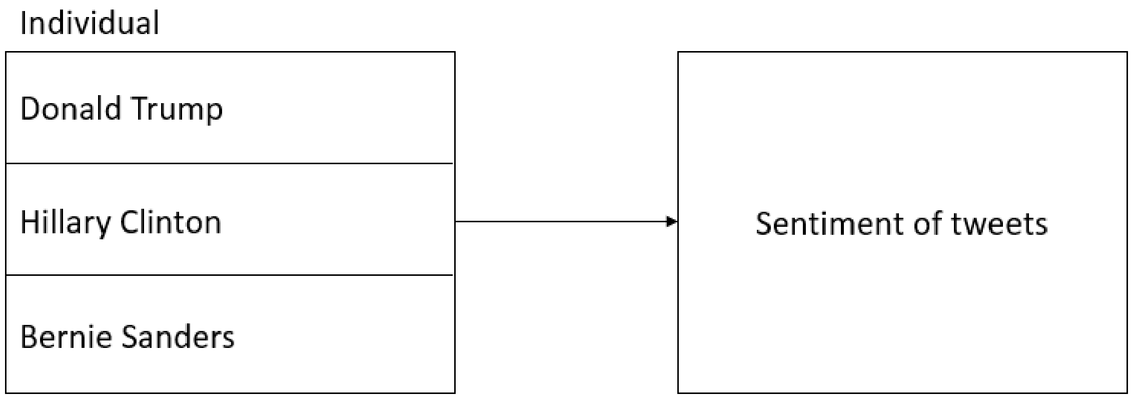
1.9 Suggested statistical analyses

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.

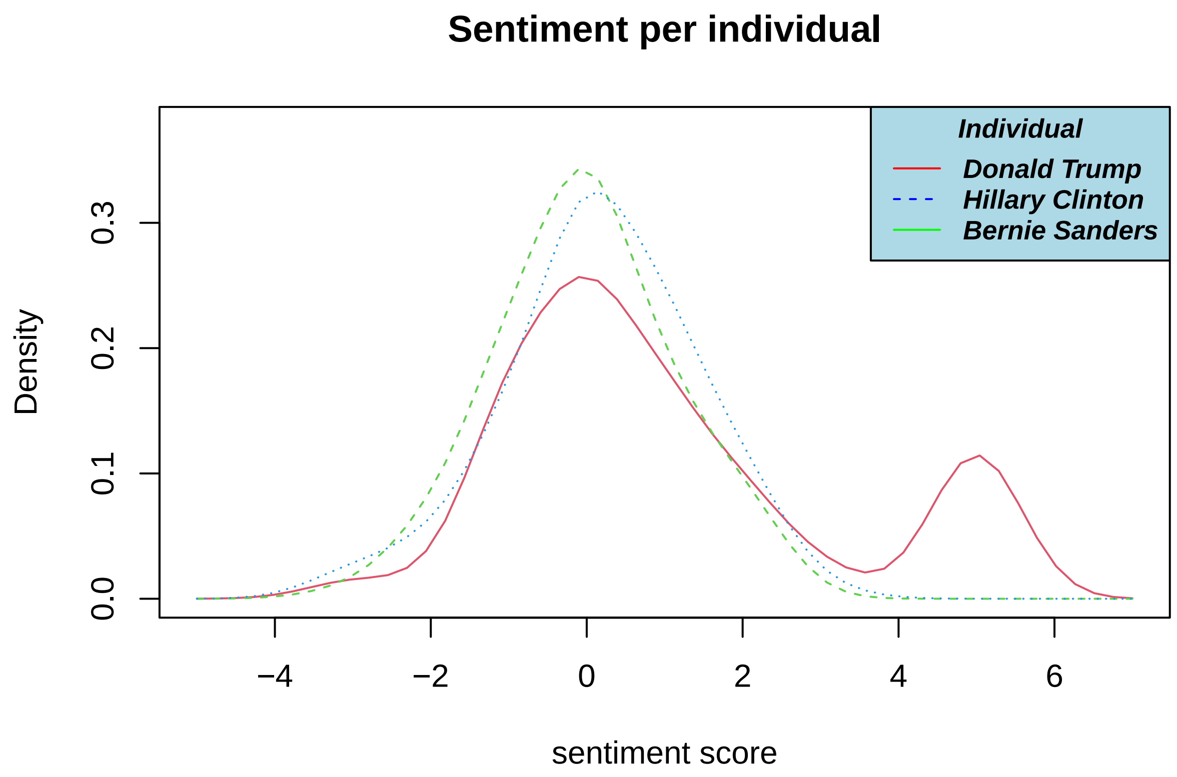
```
# Synthesis of a test data set.
sequence <- seq(-10, 10, by = .1)
test_T = rnorm(sequence, mean=-5, sd=2)
test_C = rnorm(sequence, mean=0, sd=2)
test_B = rnorm(sequence, mean=5, sd=2)

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

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

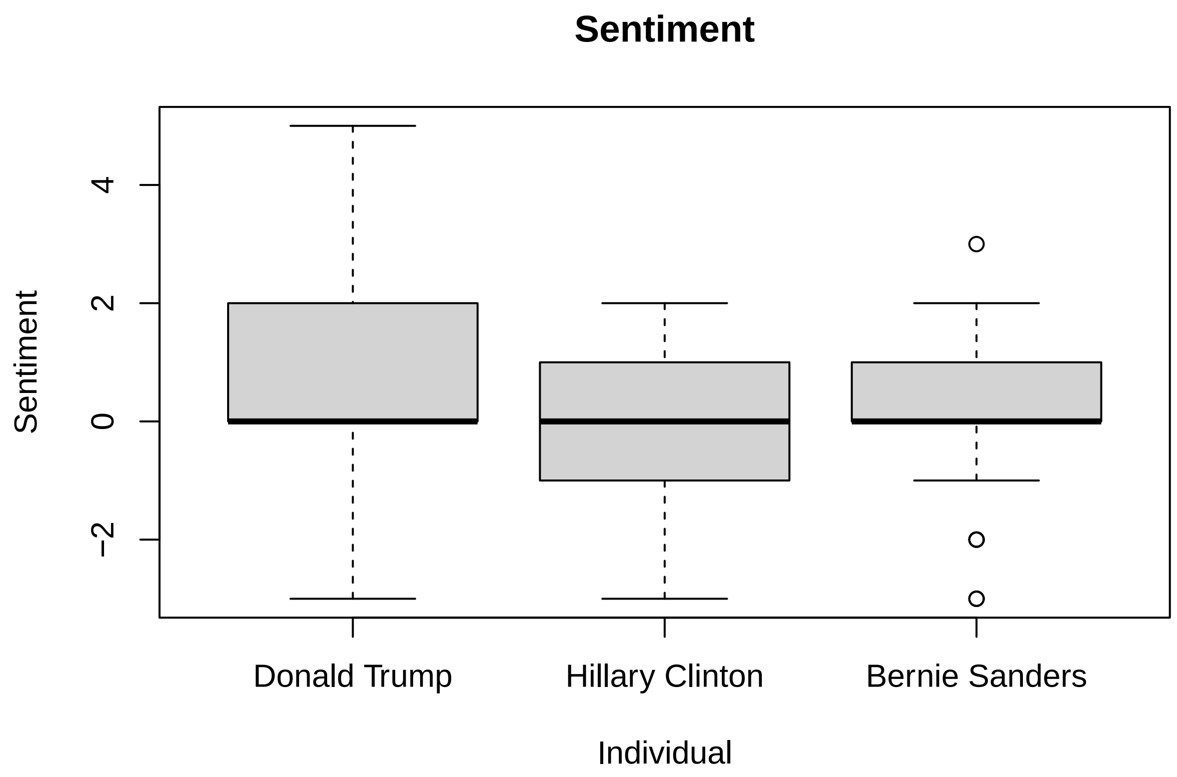
2.1.5 Visual inspection Mean and distribution sentiments

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 maximum. 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)
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.

```r
#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 12800
## 2    600  2335  2     10465 1344.5 < 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
## -5.665 -1.335 -0.024  1.366  7.554
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              -5.0879     0.1391  -36.56   <2e-16 ***
## semFrameTest$CandidateHillary Clinton    5.1817     0.1968   26.33   <2e-16 ***
## semFrameTest$CandidateBernie Sanders    10.2039     0.1968   51.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.973 on 600 degrees of freedom
## Multiple R-squared:  0.8176, Adjusted R-squared:  0.817
## F-statistic:  1345 on 2 and 600 DF,  p-value: < 2.2e-16
```

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

```
##        df      AIC
## model0  2 3557.573
## model1  4 2535.609
```

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)
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.

```
##              mean         sd      2.5%      97.5%    n_eff     Rhat4
## a[1]  -5.08301034 0.13960152 -5.350767 -4.8151865 2543.746 0.9986685
## a[2]   0.09232482 0.14398991 -0.199747  0.3666127 2127.603 0.9984306
## a[3]   5.11639919 0.14064893  4.846093  5.4017552 2361.868 0.9990799
## sigma  1.97778722 0.05561793  1.871039  2.0920927 1744.755 1.0016410
```

```
##        WAIC       SE    dWAIC      dSE    pWAIC        weight
## m1 2535.794 36.46010    0.000       NA 4.117996  1.000000e+00
## m0 3557.237 25.05794 1021.443 37.06583 1.573209 1.572013e-222
```

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.

```r
#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
)
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
)
```

```r
precis(m1, depth = 2, prob = .95, pars=c('a','sigma'))
```

```
##              mean         sd       2.5%     97.5%    n_eff    Rhat4
## a[1]   1.03718610 0.15196784  0.7457630 1.3263005 2318.427 1.000338
## a[2]  -0.01798962 0.15170197 -0.3164692 0.2823655 2185.967 1.001048
## a[3]   0.17354849 0.15273999 -0.1180837 0.4705474 2122.141 1.000203
## sigma  1.54506255 0.06410017  1.4305783 1.6765248 1946.442 1.000398
```

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

```
##        WAIC       SE   dWAIC      dSE    pWAIC       weight
## m1 1115.545 29.17225  0.00000       NA 4.303811 9.999863e-01
## m0 1137.940 33.92143 22.39503 10.27924 2.710072 1.370801e-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%
## 0.6394394 1.4722286
```

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

```
##         3%        98%
## -0.6149957  0.2213027
```

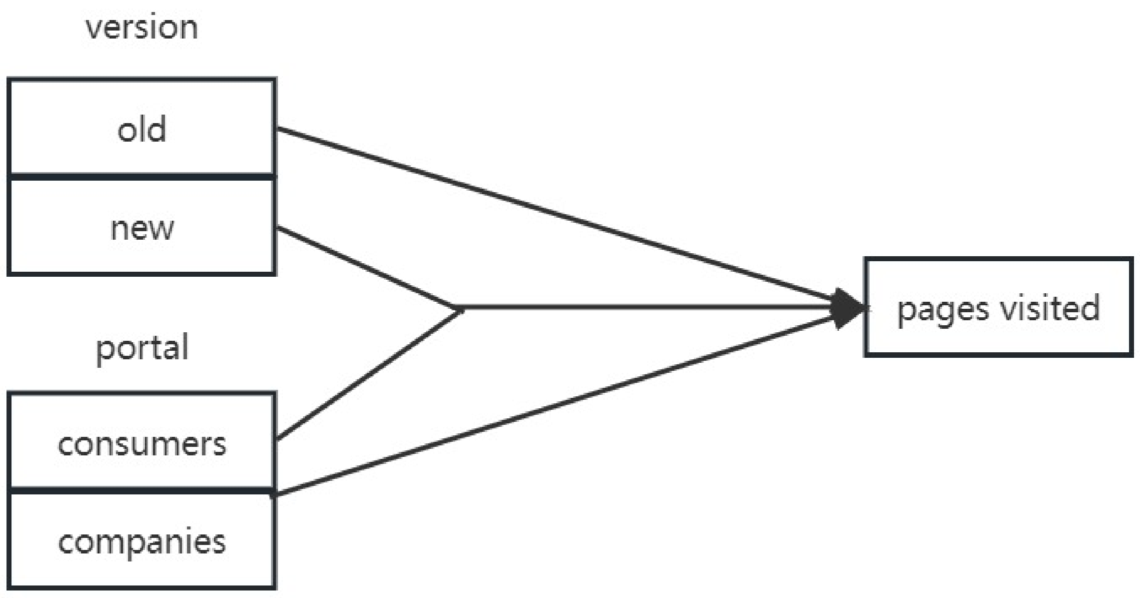
```
PI(difftrump_bernie, prob = 0.95 )
```

```
##       3%      98%
## 0.438850 1.314266
```

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.

The model can expressed as follows:

pages ~ Norm(mu, sigma)

mu = Beta0 + Beta1 * version + Beta2 * portal + Beta3 * version * portal

Beta0 ~ dnorm(20, 10)

c(Beta1, Beta2, Beta3) ~ dnorm(10,10)

sigma ~ dunif(0.001, 40)

The model assumes a Gaussian distribution for the number of page visits, with relatively uninformed priors assigned to the intercept and coefficients (Beta0, Beta1, Beta2, Beta3) as well as the standard deviation (sigma). The interaction term (version * portal) captures the combined effect of the website version and portal. To account for uncertainty, a weakly informative prior is chosen for sigma using a uniform distribution with a range that encompasses a broad yet plausible set of values.

2.2.3 Create Synthetic data

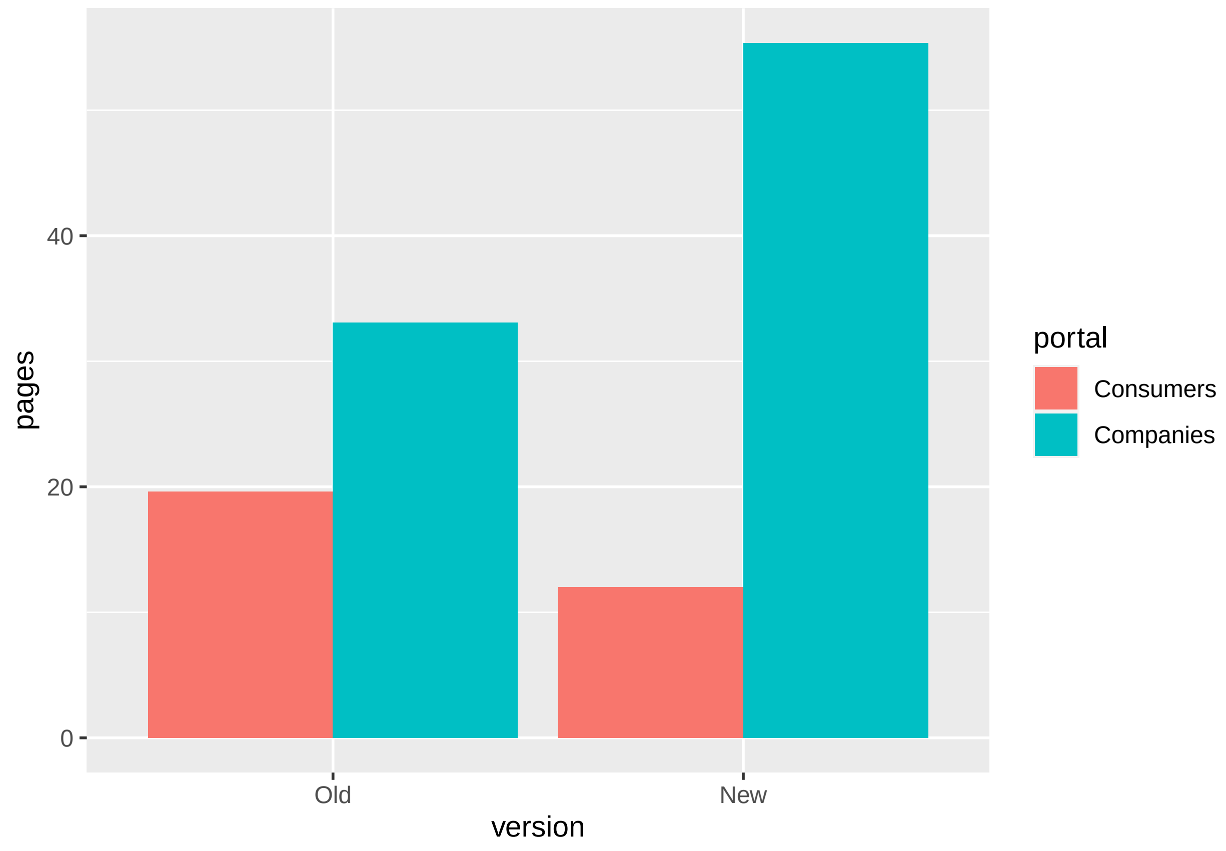
Create a synthetic data set with a clear interaction effect between the two factors for verifying your analysis later on. Report the values of the coefficients of the linear model used to generate synthetic data.

```r
#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)
pages <- beta0 + beta1 * version + beta2 * portal + beta3 * interaction + rnorm(n,0,1)

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

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

```
##   version portal interaction    pages
## 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

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

# Load the required library
library(ggplot2)

web_data <- read.csv("data/webvisit0.csv")
web_data$version <- factor(web_data$version, levels = c(0:1), labels = c("Old","New"))
web_data$portal <- factor(web_data$portal, levels = c(0:1), labels = c("Consumers","Companies"))

bar <- ggplot(web_data, aes(version , pages, fill = portal))
bar + stat_summary(fun.y = mean, geom = "bar", position="dodge")
```



The figure shows the mean page visits obtained in four conditions. 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. Last, large difference between the portals for consumers and companies exist in both the new and old version conditions.

2.2.5 Frequentist Approach

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

library(pander) #for rendering output
library(AICcmodavg) #aictab

# Fit the linear regression model
model0 <- lm(pages ~ 1, data = fake_data, na.action = na.exclude)
model1 <- lm(pages ~ version, data = fake_data, na.action = na.exclude)
model2 <- lm(pages ~ portal, data = fake_data, na.action = na.exclude)
model3 <- lm(pages ~ version + portal, data = fake_data, na.action = na.exclude)
model4 <- lm(pages ~ version + portal + version:portal, data = fake_data, na.action = na.exclude)

# see the values of coefficients
summary(model0)
```

2.2.5.1 Model verification

```
##
## Call:
## lm(formula = pages ~ 1, data = fake_data, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8486 -0.9714 -0.0576  0.9652  3.7387
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.9339     0.1398   28.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.398 on 99 degrees of freedom
```

```r
summary(model1)
```

```
##
## Call:
## lm(formula = pages ~ version, data = fake_data, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6723 -0.5478  0.0528  0.6679  2.8053
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0004     0.1473  20.368  < 2e-16 ***
## version       1.8669     0.2083   8.961 2.17e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.042 on 98 degrees of freedom
## Multiple R-squared:  0.4504, Adjusted R-squared:  0.4448
## F-statistic:  80.3 on 1 and 98 DF,  p-value: 2.173e-14
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = pages ~ portal, data = fake_data, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3369 -0.9444 -0.0880  0.9452  3.2504
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.4455     0.1861  18.519  < 2e-16 ***
## portal        0.9767     0.2631   3.712 0.000341 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.316 on 98 degrees of freedom
## Multiple R-squared:  0.1233, Adjusted R-squared:  0.1143
## F-statistic: 13.78 on 1 and 98 DF,  p-value: 0.0003415
```

```
summary(model3)
```

```
##
## Call:
## lm(formula = pages ~ version + portal, data = fake_data, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4035 -0.5121  0.0142  0.5757  2.3169
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.5121     0.1597  15.728  < 2e-16 ***
## version       1.8669     0.1844  10.122  < 2e-16 ***
## portal        0.9767     0.1844   5.296 7.39e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9222 on 97 degrees of freedom
## Multiple R-squared:  0.5736, Adjusted R-squared:  0.5648
## F-statistic: 65.25 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
summary(model4)
```

```
##
## Call:
## lm(formula = pages ~ version + portal + version:portal, data = fake_data,
##     na.action = na.exclude)
##
## 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 *
## version:portal   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
```

```
# compare the models with each other.
pander(anova(model0,model1), caption = "version as main effect on page visits")
```

Table 1: version as main effect on page visits

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
99	193.5	NA	NA	NA	NA
98	106.3	1	87.13	80.3	2.173e-14

```
pander(anova(model0,model2), caption = "portal as main effect on page visits")
```

Table 2: portal as main effect on page visits

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
99	193.5	NA	NA	NA	NA
98	169.6	1	23.85	13.78	0.0003415

```
pander(anova(model3,model4), caption = "Interation effect on top of two main effects")
```

Table 3: Interation effect on top of two main effects

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
97	82.49	NA	NA	NA	NA
96	79.1	1	3.385	4.108	0.04546

```
pander(anova(model4), caption = "Effect of version, portal and interaction effect on page visits")
```

Table 4: Effect of version, portal and interaction effect on page visits

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
version	1	87.13	87.13	105.7	3.671e-17
portal	1	23.85	23.85	28.94	5.252e-07
version:portal	1	3.385	3.385	4.108	0.04546

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Residuals	96	79.1	0.824	NA	NA

```
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 5 270.98       0.00   0.73   0.73 -130.17
## model3 4 272.96       1.97   0.27   1.00 -132.27
## model1 3 296.18      25.20   0.00   1.00 -144.97
## model2 3 342.88      71.89   0.00   1.00 -168.31
## model0 2 353.91      82.92   0.00   1.00 -174.89
```

The results indicate that the website version, portal, and their interaction effect have a significant impact on the number of page visits ($p < 0.01$). Model 4, which includes the interaction effect, demonstrates the best goodness-of-fit with the smallest AICc value. The reproduced coefficients in Model 4 closely align with the original ones, further supporting its superiority over the other models.

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

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

web_data$version <-as.numeric(web_data$version)
web_data$portal <-as.numeric(web_data$portal)

# Fit the linear regression model
model0 <- lm(pages ~ 1, data = web_data, na.action = na.exclude)
model1 <- lm(pages ~ version, data = web_data, na.action = na.exclude)
model2 <- lm(pages ~ portal, data = web_data, na.action = na.exclude)
model3 <- lm(pages ~ version + portal, data = web_data, na.action = na.exclude)
model4 <- lm(pages ~ version + portal + version:portal, data = web_data, na.action = na.exclude)

# see the values of coefficients
summary(model4)
```

2.2.5.2 Model analysis with Gaussian distribution assumed

```
##
## Call:
## lm(formula = pages ~ version + portal + version:portal, data = web_data,
##     na.action = na.exclude)
##
## 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)     43.6664     1.7265   25.29   <2e-16 ***
## version        -37.5047     1.0928  -34.32   <2e-16 ***
## portal         -16.4145     1.0928  -15.02   <2e-16 ***
## version:portal  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
```

```r
# compare the models with each other.
pander(anova(model0,model1), caption = "version as main effect on page visits")
```

Table 5: version as main effect on page visits

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
999	305972	NA	NA	NA	NA
998	289291	1	16682	57.55	7.536e-14

```r
pander(anova(model0,model2), caption = "portal as main effect on page visits")
```

Table 6: portal as main effect on page visits

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
999	305972	NA	NA	NA	NA
998	99272	1	206700	2078	3.535e-246

```r
pander(anova(model3,model4), caption = "Interation effect on top of two main effects")
```

Table 7: Interation effect on top of two main effects

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
997	85267	NA	NA	NA	NA
996	29510	1	55757	1882	1.025e-231

```r
pander(anova(model4), caption = "Effect of version, portal and interaction effect on page visits")
```

Table 8: Effect of version, portal and interaction effect on page visits

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
version	1	16682	16682	563	5.178e-99
portal	1	204024	204024	6886	0
version:portal	1	55757	55757	1882	1.025e-231
Residuals	996	29510	29.63	NA	NA

```r
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 6232.66       0.00      1      1 -3111.30
## model3 4 7291.70    1059.04      0      1 -3641.83
## model2 3 7441.76    1209.10      0      1 -3717.87
## model1 3 8511.33    2278.67      0      1 -4252.65
## model0 2 8565.38    2332.72      0      1 -4280.69
```

The results indicate that the website version, portal, and their interaction effect have a significant impact on the number of page visits ($p < 0.01$). Model 4, which includes the interaction effect, demonstrates the best goodness-of-fit with the smallest AICc value.

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

gaussian_model <- lm(pages ~ version + portal + version:portal, data = web_data)
poisson_model <- glm(pages ~ version + portal + version:portal, data = web_data, family = poisson)
summary(gaussian_model)
```

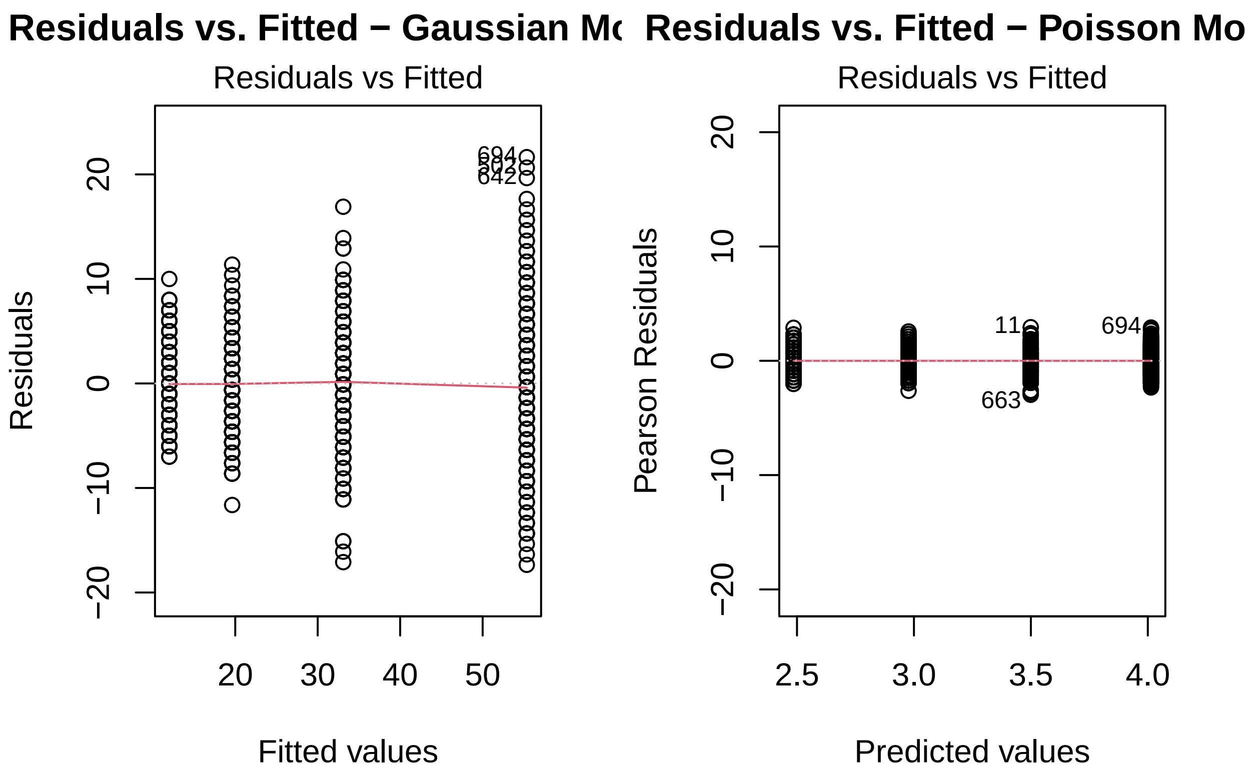
2.2.5.3 Assumption analysis

```
##
## Call:
## lm(formula = pages ~ version + portal + version:portal, 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)     43.6664     1.7265   25.29   <2e-16 ***
## version        -37.5047     1.0928  -34.32   <2e-16 ***
## portal         -16.4145     1.0928  -15.02   <2e-16 ***
## version:portal  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
```

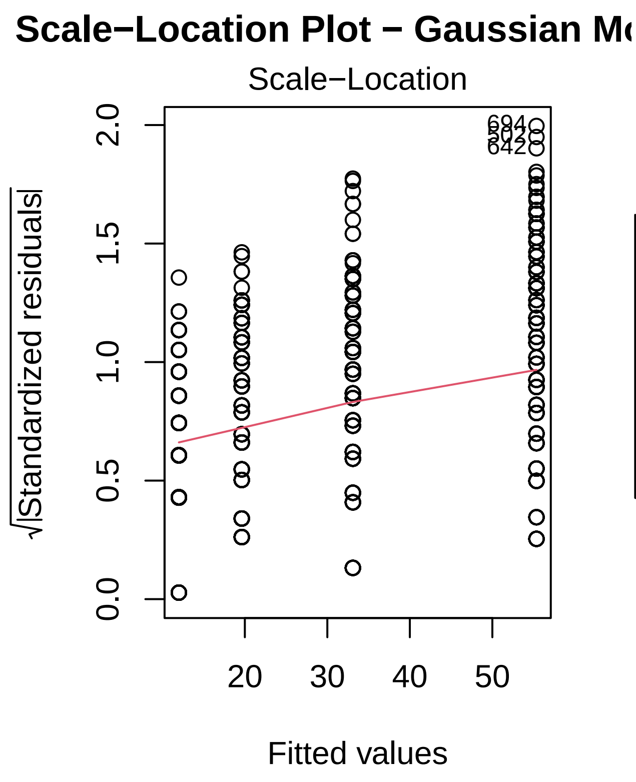
```
summary(poisson_model)
```

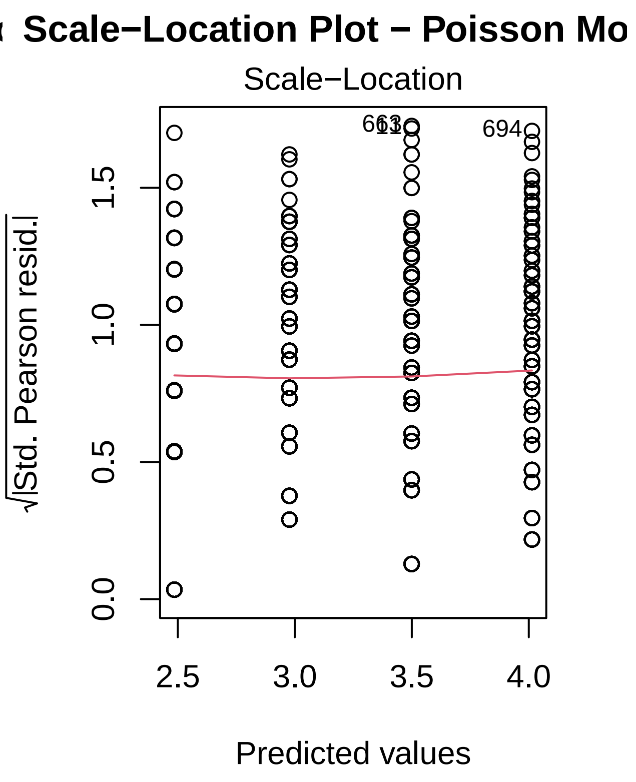
```
##
## Call:
## glm(formula = pages ~ version + portal + version:portal, family = poisson,
##     data = web_data)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.95231    0.07205   54.86   <2e-16 ***
## version        -1.49776    0.04872  -30.74   <2e-16 ***
## portal         -0.48364    0.04148  -11.66   <2e-16 ***
## version:portal  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
```

```
# # Residual analysis

par(mfrow = c(1, 2))
plot(gaussian_model, which = 1, main = "Residuals vs. Fitted - Gaussian Model")
plot(poisson_model, which = 1, extend.ylim.f = c(3.,3.), main = "Residuals vs. Fitted - Poisson Model")
```

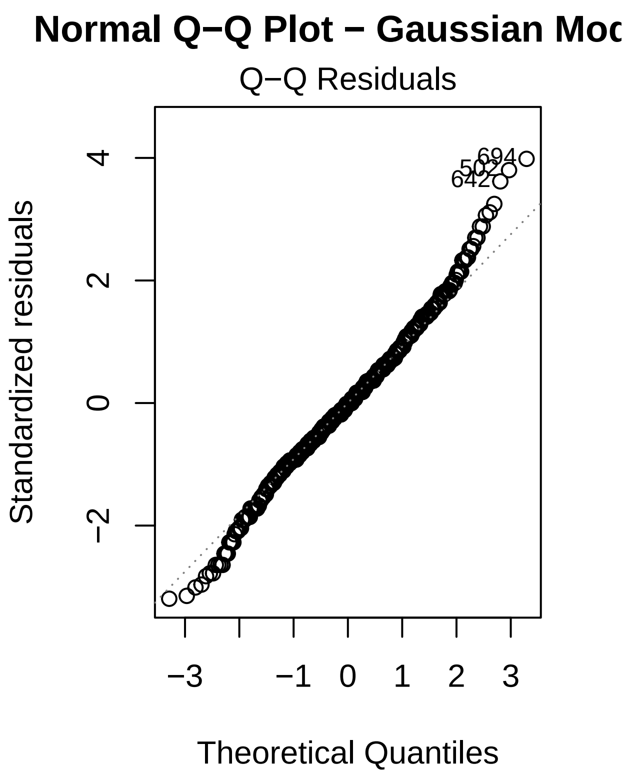


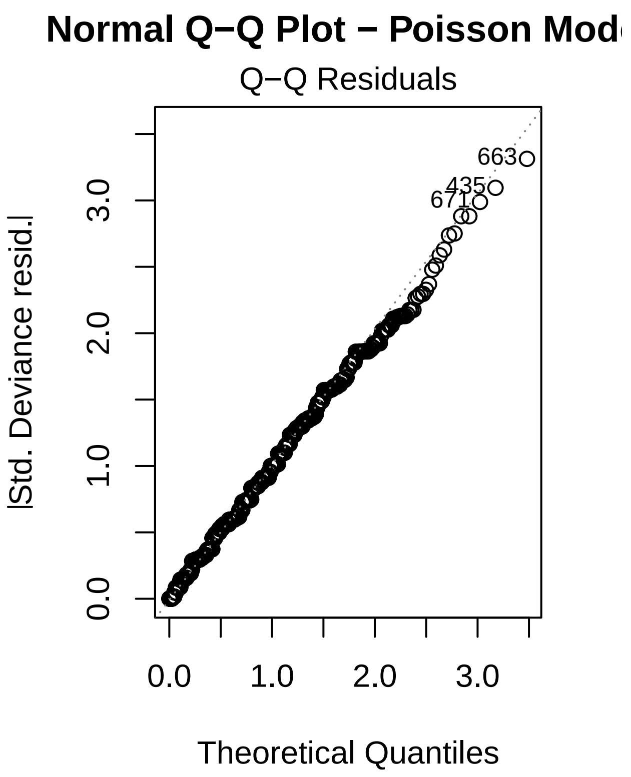
```
par(mfrow = c(1, 2))
plot(gaussian_model, which = 3, main = "Scale-Location Plot - Gaussian Model")
plot(poisson_model, which = 3, extend.ylim.f = c(3.,4.5), main = "Scale-Location Plot - Poisson Model")
```



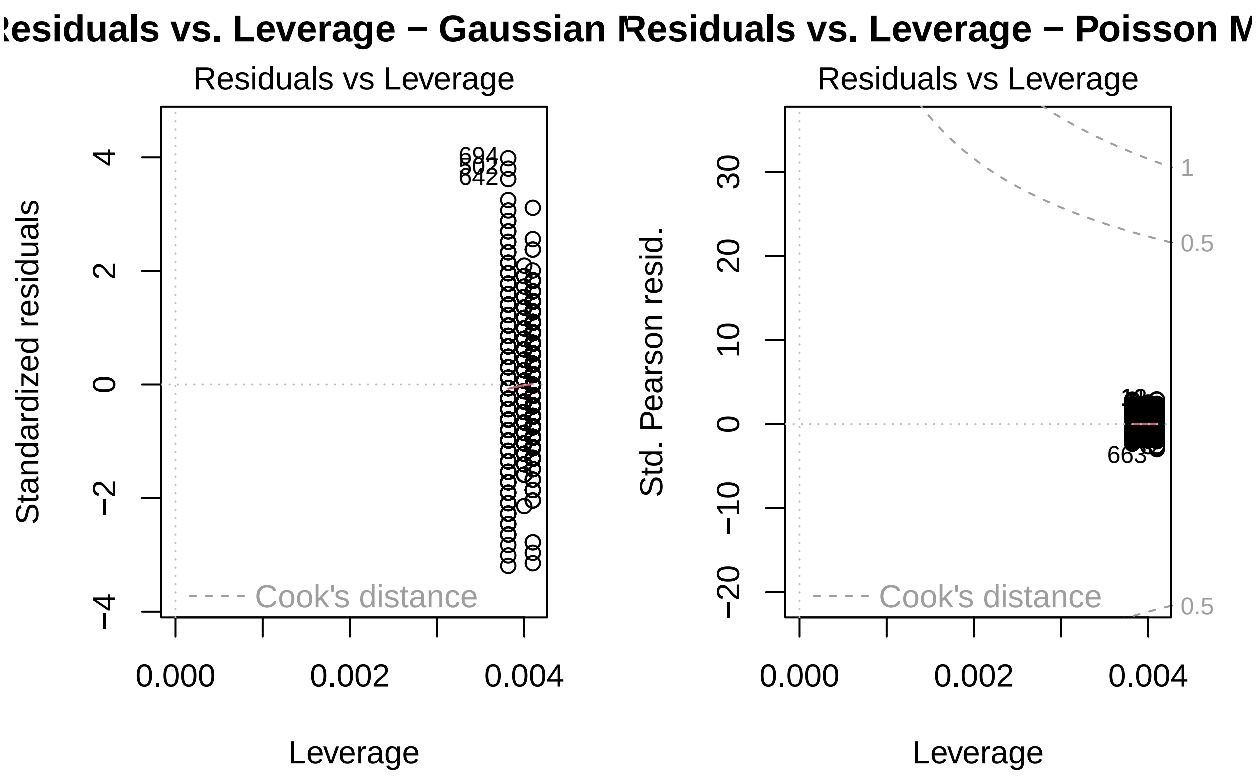


```
par(mfrow = c(1, 2))
plot(gaussian_model, which = 2, main = "Normal Q-Q Plot - Gaussian Model")
plot(poisson_model, which = 2, extend.ylim.f = c(-0.4,4.5), main = "Normal Q-Q Plot - Poisson Model")
```





```
plot(gaussian_model, which = 5, main = "Residuals vs. Leverage - Gaussian Model")
plot(poisson_model, which = 5, extend.ylim.f = c(3.,5.5), main = "Residuals vs. Leverage - Poisson Mode
```



The "Residuals vs. Fitted" and "Scale-Location" plots indicate that the spread of residuals in the Gaussian model increases as the fitted values change, suggesting a violation of the Gaussian assumption. However, for the Poisson model, the spread of residuals remains relatively constant, indicating that the Poisson assumption holds true and it is more suitable to assume a poisson distribution for page visits.

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 + version:portal, family = poisson,
##     data = web_data)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.95231    0.07205   54.86   <2e-16 ***
## version        -1.49776    0.04872  -30.74   <2e-16 ***
## portal         -0.48364    0.04148  -11.66   <2e-16 ***
## version:portal  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
```

```
library(pander)

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

```
## [1] "1.1" "2.1" "1.2" "2.2"
```

```
contrastConsumers <-c(-1,1,0,0)
contrastCompanies <-c(0,0,-1,1)

SimpleEff <- cbind(contrastConsumers,contrastCompanies)
contrasts(web_data$simple_portal) <- 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_portal , 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
simple_portalcontrastConsumers	-3.812	0.2449	-15.56	4.691e-49
simple_portalcontrastCompanies	11.13	0.2421	45.96	2.195e-248
simple_portal	28.41	0.3444	82.48	0

Table 10: Fitting linear model: pages ~ simple_portal

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

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

```
## [1] "1.1" "2.1" "1.2" "2.2"
```

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

SimpleEff <- cbind(contrastOld,contrastNew)
contrasts(web_data$simple_version) <- 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.
```

```r
simpleEffectModel <-lm(pages ~ simple_version , 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
simple_versioncontrastOld	6.733	0.2449	27.49	2.928e-124
simple_versioncontrastNew	21.67	0.2421	89.51	0
simple_version	7.316	0.3444	21.24	6.746e-83

Table 12: Fitting linear model: pages ~ simple_version

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

The analysis investigated the interaction effect between two variables, version and portal, by dividing the data into two groups. The results indicate significant differences ($p < 0.01$) in the number of page visits based on the version type for both company and consumer portals. Additionally, the portal variable also has a significant impact ($p < 0.01$) on page visits for both the old and new versions of the website. Furthermore, the contrasts for both version and portal variables are also statistically significant ($p < 0.01$).

2.2.5.5 Report section for a scientific publication A linear model was fitted to analyze the number of page visits on a website, considering the website version and portal as independent variables, with a two-way interaction. The results indicated significant main effects for both the version (F(1, 996) = 563, $p < 0.01$) and the portal (F(1, 996) = 6886, $p < 0.01$). Furthermore, a significant two-way interaction effect was observed (F(1, 996) = 1882, $p < 0.01$). Further analysis of the interaction revealed a significant difference ($p < 0.01$) in page visits based on the website portal for old/new website version. Additionally, the website version also has a significant impact ($p < 0.01$) on page visits for both the companies and consumers portals.

2.2.6 Bayesian Approach

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

options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)


model_bay_fake0 <- map2stan(
  alist(
    pages ~ dnorm(mu, sigma),
    mu <- Beta0,
    Beta0 ~ dnorm(2,2),
    sigma ~ dunif(0.1,2)
    ),
  data = fake_data,
  chains = 4,
  cores = 4,
  iter = 1000
```

```
)

model_bay_fake1 <- map2stan(
  alist(
    pages ~ dnorm(mu, sigma),
    mu <- Beta0 + Beta1 * version,
    Beta0 ~ dnorm(2,2),
    Beta1 ~ dnorm(2,2),
    sigma ~ dunif(0.1,2)
    ),
  data = fake_data,
  chains = 4,
  cores = 4,
  iter = 1000
)

model_bay_fake2 <- map2stan(
  alist(
    pages ~ dnorm(mu, sigma),
    mu <- Beta0 + Beta2 * portal,
    Beta0 ~ dnorm(2,2),
    Beta2 ~ dnorm(1,2),
    sigma ~ dunif(0.1,2)
    ),
  data = fake_data,
  chains = 4,
  cores = 4,
  iter = 1000
)

model_bay_fake3 <- map2stan(
  alist(
    pages ~ dnorm(mu, sigma),
    mu <- Beta0 + Beta1 * version + Beta2 * portal,
    Beta0 ~ dnorm(2,2),
    Beta1 ~ dnorm(2,2),
    Beta2 ~ dnorm(1,2),
    sigma ~ dunif(0.1,2)
    ),
  data = fake_data,
  chains = 4,
  cores = 4,
  iter = 1000
)

model_bay_fake4 <- map2stan(
  alist(
    pages ~ dnorm(mu, sigma),
    mu <- Beta0 + Beta1 * version + Beta2 * portal + Beta3 * version * portal,
    Beta0 ~ dnorm(2,2),
    Beta1 ~ dnorm(2,2),
    Beta2 ~ dnorm(1,2),
    Beta3 ~ dnorm(1,2),
    sigma ~ dunif(0.1,2)
    ),
  data = fake_data,
  chains = 4,
  cores = 4,
  iter = 1000
)
```

```
compare(model_bay_fake0,model_bay_fake1,model_bay_fake2,model_bay_fake3,model_bay_fake4)
```

2.2.6.1 Verification Analysis

```
##                     WAIC       SE     dWAIC       dSE    pWAIC       weight
## model_bay_fake4 270.5863 14.07475  0.000000        NA 4.810996 7.372313e-01
## model_bay_fake3 272.6496 15.26187  2.063272  3.865576 3.934553 2.627664e-01
## model_bay_fake1 295.9647 14.98267 25.378379  9.694721 2.961867 2.273834e-06
## model_bay_fake2 342.5917 14.03763 72.005394 10.576166 2.865024 1.705419e-16
## model_bay_fake0 353.7257 13.41754 83.139328 11.955678 1.838124 6.518206e-19
```

```
precis(model_bay_fake4, prob= .95)
```

```
##            mean         sd       2.5%    97.5%     n_eff    Rhat4
## Beta0 2.6911089 0.17867622 2.34110600 3.036688  870.8097 1.003502
## Beta1 1.5096800 0.25007843 1.02528975 1.997862  928.9897 1.001316
## Beta2 0.6117098 0.25194152 0.10277456 1.091688  821.9522 1.002986
## Beta3 0.7297707 0.35741466 0.03873513 1.441595  918.3563 1.002340
## sigma 0.9212389 0.06804313 0.80036368 1.063305 1183.5961 1.002526
```

Model_bay_fake4, which includes the combination of version, portal, and the interaction effect, exhibits the lowest WAIC value, indicating the best out-of-sample fit among the models compared. Additionally, the credible intervals for all coefficients encompass their original values, suggesting that the estimated coefficients are statistically consistent with the data.

2.2.6.2 Model description

Model:

Pages ~ Poisson(lambda)

lambda = exp(Beta0 + Beta1 * version + Beta2 * portal + Beta3 * version * portal)

Beta0 ~ Normal(10, 20) Beta1 ~ Normal(10, 20) Beta2 ~ Normal(10, 20) Beta3 ~ Normal(10, 20)

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.

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


# Model formulation


model_bay_web0 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- Beta0,
```

```r
    Beta0 ~ dnorm(10, 20)
    ),
  data = web_data,
  chains = 4,
  cores = 4,
  iter = 1000
)

model_bay_web1 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- Beta0 + Beta1 * version ,
    Beta0 ~ dnorm(10, 20),
    Beta1 ~ dnorm(10, 20)
    ),
  data = web_data,
  chains = 4,
  cores = 4,
  iter = 1000
)

model_bay_web2 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- Beta0  + Beta2 * portal,
    Beta0 ~ dnorm(10, 20),
    Beta2 ~ dnorm(10, 20)
    ),
  data = web_data,
  chains = 4,
  cores = 4,
  iter = 1000
)

model_bay_web3 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- Beta0 + Beta1 * version + Beta2 * portal,
    Beta0 ~ dnorm(10, 20),
    Beta1 ~ dnorm(10, 20),
    Beta2 ~ dnorm(10, 20)
    ),
  data = web_data,
  chains = 4,
  cores = 4,
  iter = 1000
)

model_bay_web4 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- Beta0 + Beta1 * version + Beta2 * portal + Beta3 * version * portal,
    Beta0 ~ dnorm(10, 20),
    Beta1 ~ dnorm(10, 20),
    Beta2 ~ dnorm(10, 20),
    Beta3 ~ dnorm(10, 20)
    ),
  data = web_data,
  chains = 4,
  cores = 4,
  iter = 1000
)
```

```
precis(model_bay_web4, prob = .95)
```

2.2.6.3 Model comparison

```
##              mean          sd       2.5%      97.5%    n_eff    Rhat4
## Beta0  3.9613340 0.07665883  3.8157615  4.1045542 214.2480 1.041405
## Beta1 -1.5044095 0.05190904 -1.6045547 -1.4043978 227.7689 1.038275
## Beta2 -0.4883318 0.04452106 -0.5726301 -0.4007736 232.1841 1.042024
## Beta3  1.0094556 0.02921298  0.9520513  1.0660932 202.8012 1.039424
```

```
compare(model_bay_web0,model_bay_web1,model_bay_web2,model_bay_web3,model_bay_web4)
```

```
##                     WAIC        SE    dWAIC       dSE     pWAIC        weight
## model_bay_web4  6057.784  45.60067    0.000        NA  4.039210  1.000000e+00
## model_bay_web3  7481.180  85.14666 1423.396  76.70048  7.137224 8.195393e-310
## model_bay_web2  7940.521 108.21941 1882.737  90.76414  5.087877  0.000000e+00
## model_bay_web1 14506.690 284.58164 8448.906 287.87128 16.514876  0.000000e+00
## model_bay_web0 15050.530 286.87418 8992.746 283.07265 10.461784  0.000000e+00
```

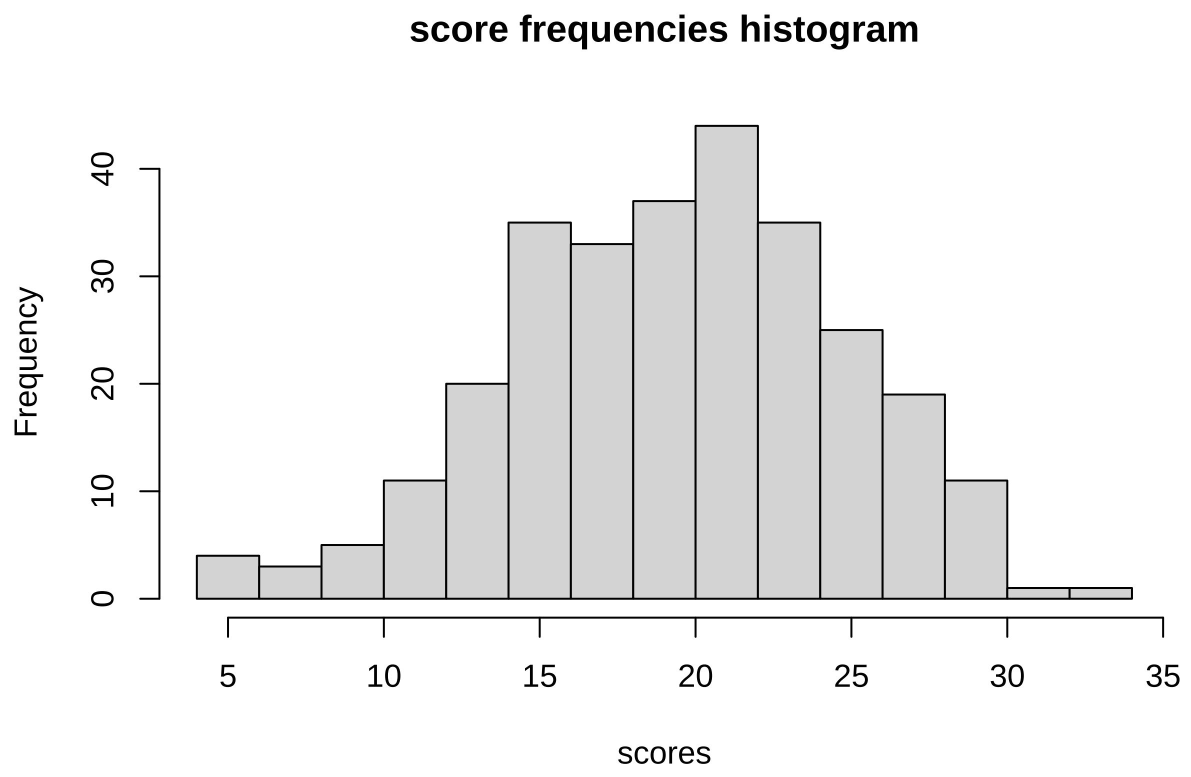
The model_bay_web4, which incorporates the combination of version, portal, and the interaction effect, demonstrates the lowest WAIC value, indicating superior out-of-sample fit compared to the other models evaluated. The inclusion of both web portal and the interaction effect results in a substantial decrease in the WAIC, suggesting the significance of these two factors in explaining the observed data. Furthermore, the credible intervals for all coefficients encompass their values when using a linear function, indicating that the estimated coefficients are statistically consistent.

3 Part 3 - Multilevel model

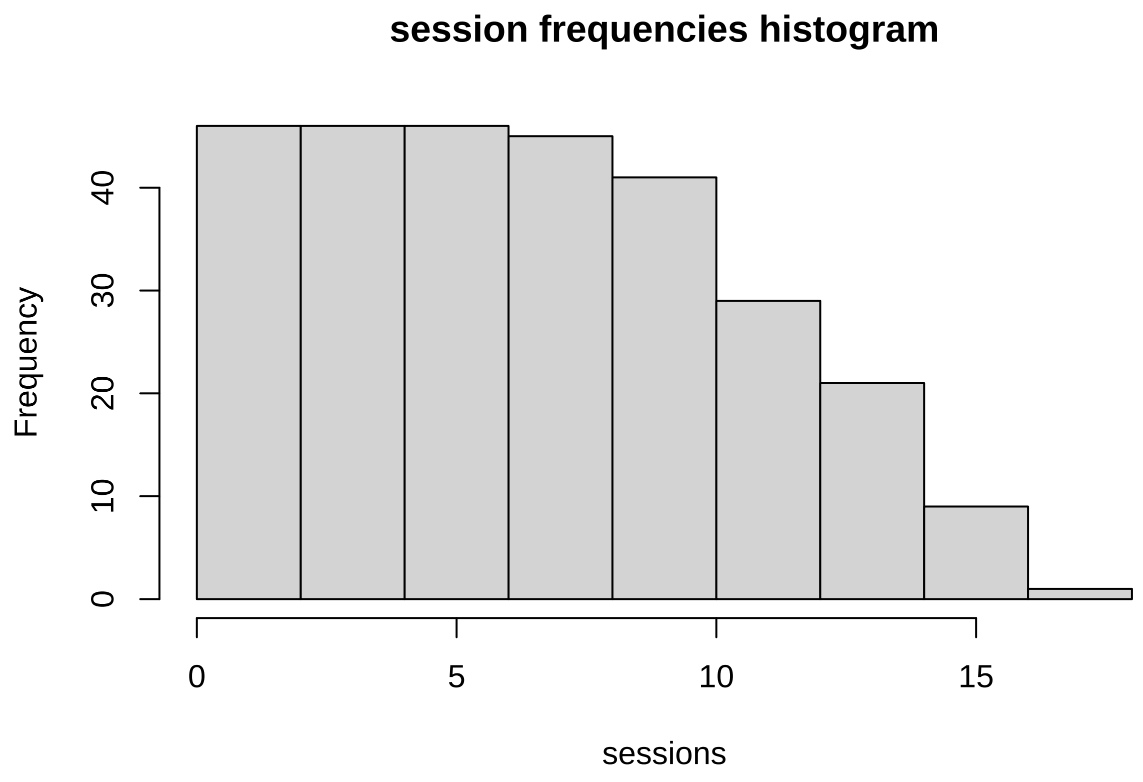
3.1 Visual inspection

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

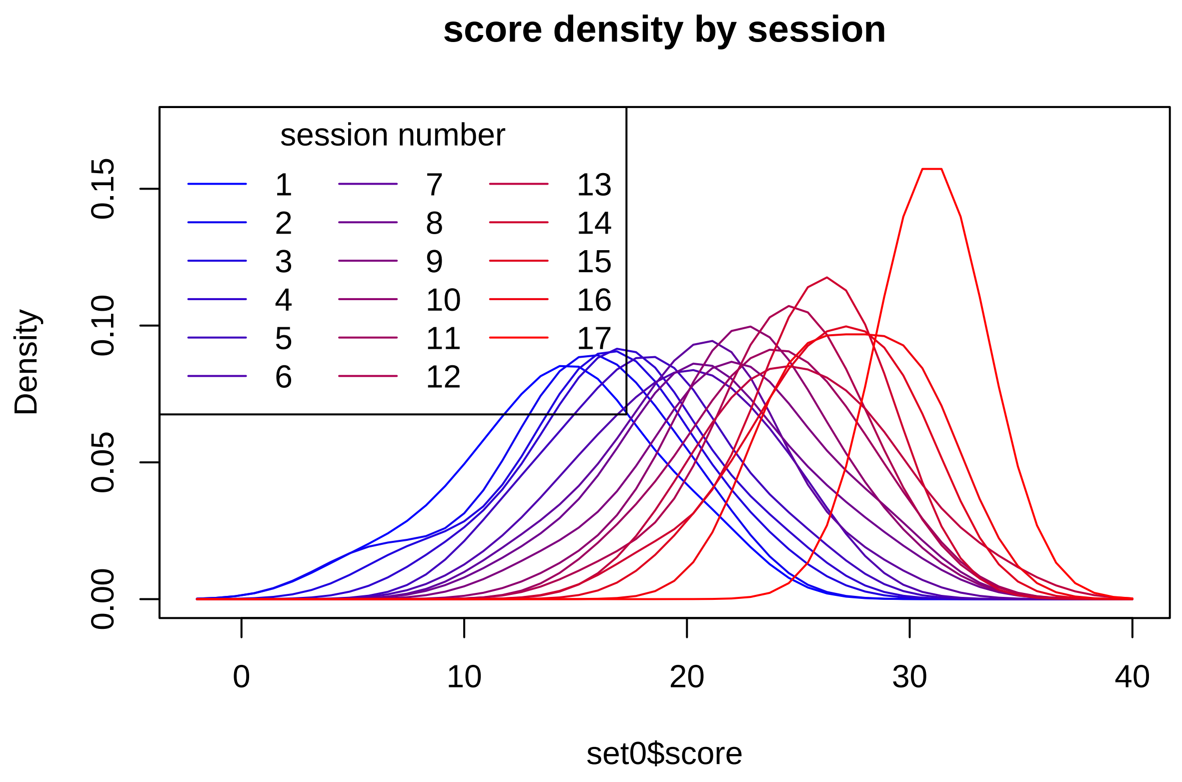
```
#include your code and output in the document
library(sm)
library(car)

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")
```



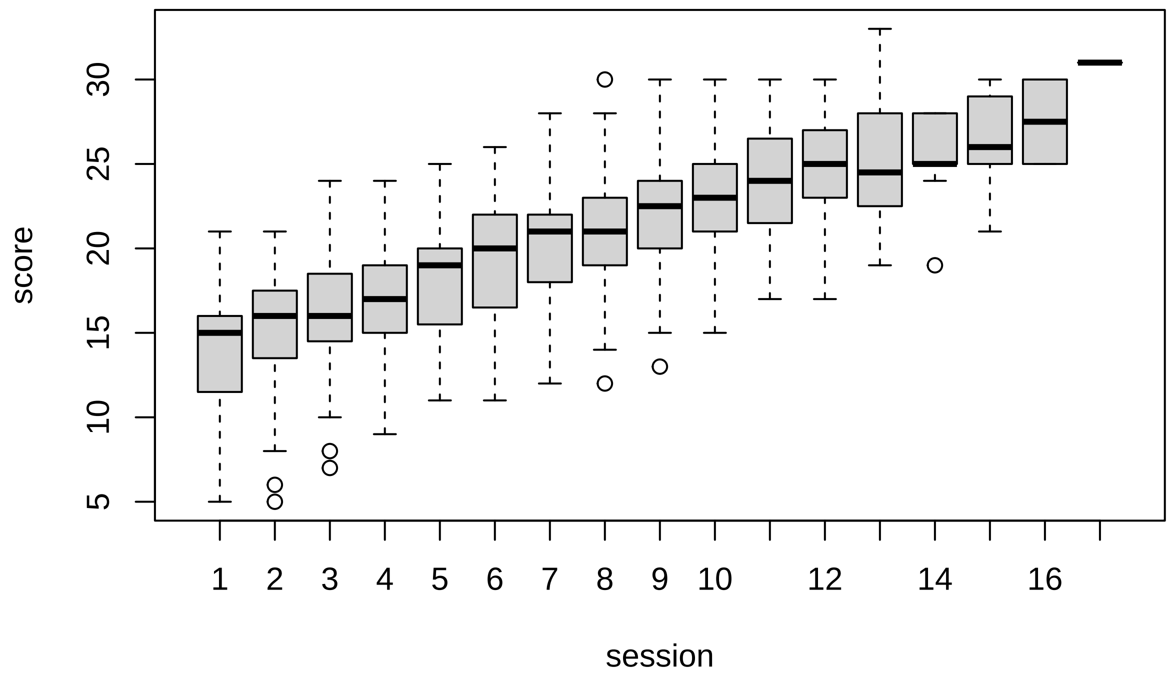
```
hist(set0$session, xlab="sessions", main="session frequencies histogram")
```



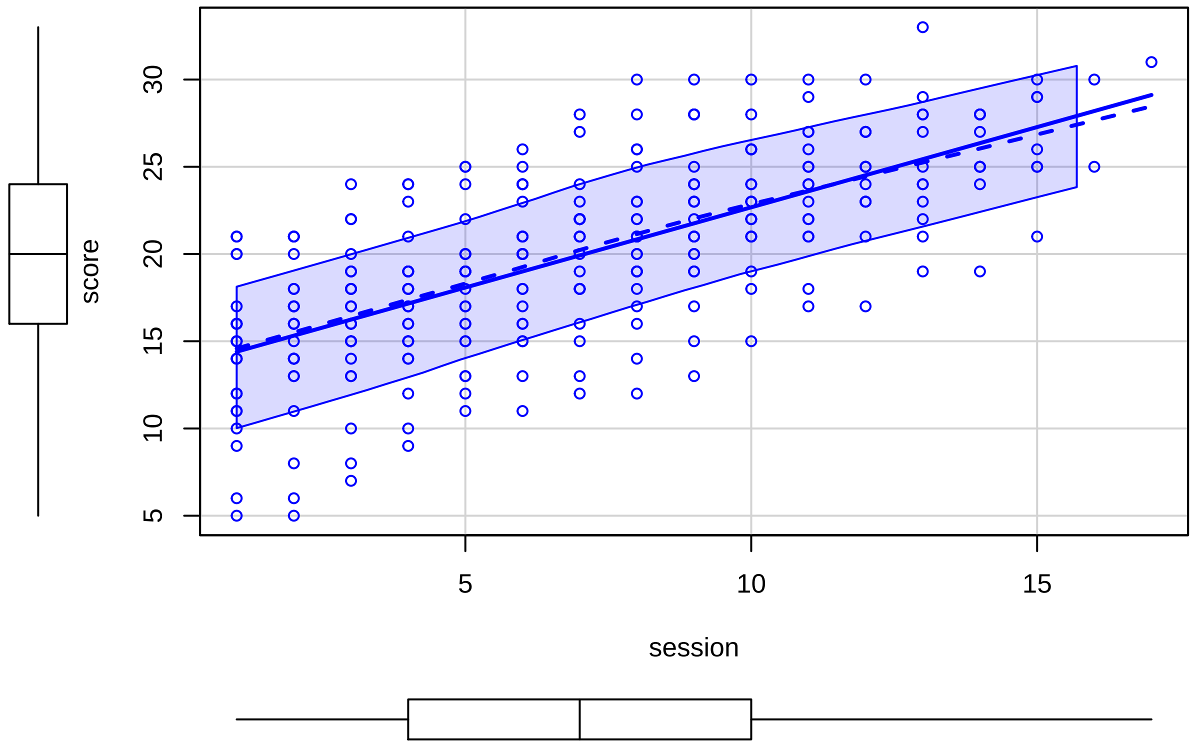
```
# Define color gradient
color_range <- colorRampPalette(c("blue", "red"))

# 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")
```



```r
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 13: 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 14: 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

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

Table 15: 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.

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

3.3.2 Model comparison

Compare models with with increasing complexity.

```r
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 = 1000,
  chains=4,log_lik = TRUE,
  cores=4,control = list(adapt_delta=.99)
```

```
)

#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 = 1000,
  chains=4,log_lik = TRUE,
  cores=4, control = list(adapt_delta=.99)
)


#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 = 1000,
  chains=4,log_lik = TRUE,
  cores=4, control = list(adapt_delta=.99)
```

```
)
```

```
precis(bays_m1, depth=2, prob=.95, pars=c('a_subject', 'a',  'sigma'))
```

```
##                      mean        sd       2.5%     97.5%      n_eff     Rhat4
## a_subject[1]   2.157995168 2.3633688  -2.7694650  6.636949   85.24436 1.0189492
## a_subject[2]  -5.544250184 2.3673741 -10.2131975 -1.086049   87.06226 1.0171153
## a_subject[3]   2.989581332 2.3557532  -1.6370110  7.347579   82.02946 1.0176558
## a_subject[4]  -1.384001852 2.3454859  -6.2662010  3.040478   80.70662 1.0195740
## a_subject[5]  -1.661271629 2.3341235  -6.3277507  2.615665   79.76998 1.0197518
## a_subject[6]  -0.696115631 2.5671605  -5.8786630  4.249111   97.31642 1.0163862
## a_subject[7]  -9.281141200 2.4978731 -14.4577950 -4.712164   94.80998 1.0157266
## a_subject[8]   2.065318841 2.3376638  -2.6351617  6.291306   81.02534 1.0203683
## a_subject[9]   3.990742058 2.3368765  -0.6548960  8.240709   79.14983 1.0194785
## a_subject[10]  5.026370192 2.3896692   0.1867185  9.558477   82.81247 1.0184090
## a_subject[11] -0.756125903 2.3171055  -5.5408400  3.561442   84.15336 1.0159261
## a_subject[12]  6.930414701 2.3896082   2.1989995 11.421410   85.25453 1.0176888
## a_subject[13]  2.313263402 2.4060344  -2.6034677  6.685853   89.62456 1.0167124
## a_subject[14] -2.071919558 2.4570629  -7.2277550  2.513711   94.13762 1.0130660
## a_subject[15]  3.046212931 2.4760270  -1.9190660  7.553671   94.70385 1.0157080
## a_subject[16] -0.580772174 2.3122210  -5.3004135  3.790844   82.08421 1.0179152
## a_subject[17] -0.726808235 2.3325087  -5.6583860  3.548294   81.40472 1.0166720
## a_subject[18] -0.009754712 2.5090793  -5.1725485  4.823123   88.75268 1.0207526
## a_subject[19] -6.070100173 2.3393414 -10.7181700 -1.729843   86.04650 1.0177396
## a_subject[20]  0.409901186 2.4503392  -4.5592672  4.986829   88.28205 1.0179164
## a_subject[21]  1.148147411 2.3685927  -3.9691882  5.558483   82.21363 1.0193626
## a_subject[22]  0.289308288 2.3576276  -4.5073542  4.718821   87.46779 1.0211985
## a_subject[23]  5.801473721 2.4386125   0.9253778 10.333630   87.08009 1.0180194
## a             19.467230800 2.1078379  15.6075075 23.709012   67.47850 1.0236686
## sigma          4.071754385 0.1820272   3.7501472  4.455853 1396.10480 0.9996831
```

```
precis(bays_m2, depth=2, prob=.95, pars=c('a_subject', 'sigma_subject', 'a', 'sigma'))
```

```
##                     mean        sd        2.5%      97.5%    n_eff     Rhat4
## a_subject[1]   1.77141764 1.2719219  -0.7748411  4.2167032 380.0656 1.0046463
## a_subject[2]  -5.28915747 1.3430908  -7.9363175 -2.6889783 344.3711 1.0052756
## a_subject[3]   2.64405268 1.2944882   0.1282816  5.1086738 401.7081 1.0059602
## a_subject[4]  -1.46372287 1.2903479  -3.9811960  1.1126100 354.2971 1.0092093
## a_subject[5]  -1.76333485 1.2691801  -4.2774487  0.7475804 318.9620 1.0058032
## a_subject[6]  -0.84960407 1.6181747  -3.9647932  2.3389755 557.5656 1.0046040
## a_subject[7]  -8.56446284 1.5077322 -11.5328925 -5.5484695 542.0287 1.0040359
## a_subject[8]   1.68620795 1.2870736  -0.9011811  4.2366890 350.9761 1.0080071
## a_subject[9]   3.50974836 1.2481252   0.9972439  5.8597813 315.3091 1.0068063
## a_subject[10]  4.37343559 1.3765960   1.7903553  7.1767167 448.9729 1.0040931
## a_subject[11] -0.95204215 1.2718833  -3.3951422  1.4688512 312.5482 1.0074586
## a_subject[12]  6.19356748 1.3620011   3.4902003  8.7802110 414.7010 1.0068694
## a_subject[13]  1.91244811 1.3935042  -0.8038960  4.5214415 445.8314 1.0023734
## a_subject[14] -2.03205031 1.4069392  -4.8923275  0.6853157 468.4624 1.0059622
## a_subject[15]  2.52391257 1.4885808  -0.5147945  5.3500242 517.6081 1.0030735
## a_subject[16] -0.77484049 1.2484503  -3.1946620  1.7250207 348.6491 1.0057633
## a_subject[17] -0.91591018 1.2415209  -3.3891705  1.5007040 343.7568 1.0070129
## a_subject[18] -0.24768351 1.4777514  -3.1446422  2.7920477 492.7184 1.0031627
## a_subject[19] -5.88588506 1.2745192  -8.3973403 -3.4629570 363.4788 1.0058194
## a_subject[20]  0.18598423 1.4312405  -2.5414662  3.0081367 437.2986 1.0054651
## a_subject[21]  0.81704194 1.3059505  -1.7343540  3.4074652 398.1011 1.0068992
## a_subject[22]  0.06321918 1.3606053  -2.5479237  2.7679667  367.5006 1.0077164
## a_subject[23]  5.05525217 1.4296528   2.4096885  7.9861857  612.9667 1.0045739
## sigma_subject  3.82722453 0.6875955   2.6959373  5.4089720 1545.7474 1.0040390
## a             19.71010580 0.8319639  18.0670800 21.3426150  163.9089 1.0163202
## sigma          4.06786616 0.1819103   3.7395380  4.4299382 1867.7617 0.9993985
```

```
precis(bays_m3, depth=2, prob=.95, pars=c('a_subject', 'a', 'b', 'sigma', 'a_sigma_subject'))
```

```
##                       mean         sd         2.5%      97.5%     n_eff
## a_subject[1]     0.6453768 1.03490629  -1.39547675  2.6928025 166.07578
## a_subject[2]    -5.7635190 1.05161098  -7.81915400 -3.7170218 174.31507
## a_subject[3]     1.9536812 1.00069616  -0.09470868  3.9007840 160.24532
## a_subject[4]    -1.9013275 1.02025905  -3.89658625  0.0448468 172.50890
## a_subject[5]    -3.0124194 1.00354151  -5.01258475 -1.0741925 178.71584
## a_subject[6]     0.9581842 1.17809543  -1.31452525  3.3812005 238.66915
## a_subject[7]    -9.0305920 1.11350907 -11.28004250 -6.8042963 208.66692
## a_subject[8]     0.6709294 1.01185035  -1.36498425  2.6386892 160.88632
## a_subject[9]     0.7355102 0.99169187  -1.22745500  2.6609567 159.95283
## a_subject[10]    6.7036761 1.07218897   4.55921400  8.7897862 180.28612
## a_subject[11]   -3.2886727 0.99187253  -5.25040575 -1.2734460 164.28901
## a_subject[12]    6.2311494 1.04310824   4.16847850  8.2362905 185.82983
## a_subject[13]    1.8877036 1.05048631  -0.14210815  3.9857585 164.13641
## a_subject[14]   -1.7811363 1.08540167  -3.99389150  0.2637756 193.04241
## a_subject[15]    4.3768496 1.07929984   2.23074025  6.5210767 201.94506
## a_subject[16]   -2.2551454 1.00941460  -4.31029125 -0.3301699 167.65308
## a_subject[17]   -2.1822482 1.00187563  -4.20074300 -0.1401680 165.34014
## a_subject[18]    1.6575120 1.11680286  -0.52338613  3.8616367 197.43505
## a_subject[19]   -8.2452565 0.99896278 -10.24757250 -6.2691355 160.74976
## a_subject[20]    2.2525434 1.10773276   0.13047815  4.4060473 189.06226
## a_subject[21]    0.1317577 1.04108277  -1.97081900  2.1424037 174.27511
## a_subject[22]    0.3460335 1.04612469  -1.77580750  2.3259990 189.21280
## a_subject[23]    5.9421189 1.06901496   3.88663250  8.0154922 189.87273
## a               13.2897568 0.86445281  11.59700000 14.9974525 121.52020
## b                1.3689825 1.82860376  -2.24103525  4.6105427  40.84992
## sigma            1.0083361 0.04807948   0.92099633  1.1057915 970.37282
## a_sigma_subject  4.3779011 0.71177986   3.25254125  6.0368055 849.02175
##                    Rhat4
## a_subject[1]    1.030249
## a_subject[2]    1.027051
## a_subject[3]    1.029969
## a_subject[4]    1.024822
## a_subject[5]    1.023184
## a_subject[6]    1.020706
## a_subject[7]    1.019723
## a_subject[8]    1.028298
## a_subject[9]    1.025410
## a_subject[10]   1.023524
## a_subject[11]   1.029080
## a_subject[12]   1.024512
## a_subject[13]   1.029739
## a_subject[14]   1.021655
## a_subject[15]   1.023776
## a_subject[16]   1.025707
## a_subject[17]   1.027135
## a_subject[18]   1.022474
## a_subject[19]   1.028623
## a_subject[20]   1.025160
## a_subject[21]   1.026234
## a_subject[22]   1.024399
## a_subject[23]   1.023698
## a               1.037600
## b               1.084510
## sigma           1.004197
## a_sigma_subject 1.000558
```

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

```
##              WAIC       SE    dWAIC dSE    pWAIC        weight
## bays_m3  861.1266 23.04022   0.0000  NA 43.55642  1.000000e+00
## bays_m2 1621.5650 17.71362 760.4384  NA 19.01618 7.462743e-166
## bays_m1 1624.0324 17.51936 762.9058  NA 20.91240 2.173271e-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.

3.3.3 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.