Coursework assignment A - 2022-2023

CS4125 Seminar Research Methodology for Data Science

Suhaib, Otte, Colin

16/04/2023

Part 1 - Design and set-up of true experiment

The motivation for the planned research

Previous research [1] has shown that while some technologies can improve student achievement, others may actually decrease it. The impact of technology is very dependent on the technology itself but also on who is using it, teachers or students. Given that ChatGPT is a widely-used innovation among students, it is important to investigate its effect on student achievement. By comparing the learning outcomes, intrinsic motivation, and perceived difficulty of students who have access to ChatGPT with those who rely on traditional coding resources, the research aims to determine if ChatGPT can improve student achievement in a challenging coding assignment.

1. New global data reveal education technology's impact on learning. (2020, June 12). McKinsey & Company. https://www.mckinsey.com/industries/education/our-insights/new-global-data-reveal-education-technologys-impact-on-learning#/ (https://www.mckinsey.com/industries/education/our-insights/new-global-data-reveal-education-technologys-impact-on-learning#/)
2. Lancaster, Thomas & wilkinson, richard. (2014). Improving Student Motivation Using Technology Within The STEM Disciplines.

The theory underlying the research

The main theory which encompasses this topic is within the field of educational technologies. According to a paper by Francis et al. discusses the effects of technology on student motivation and engagement in classroom-based learning. The study found that students feel motivated through the specific use of technology in the classroom, whether it be for educational purposes or for accommodation in the classroom. This suggests that as technology continues to advance, there is potential for even greater impact on student motivation and engagement in the classroom. The integration of technology into education can help teachers differentiate instruction, motivate students, and include all skill levels. However, some studies also indicate that the extensive use of internet resources can reduce the difficulty level of college assignments and thus reduce student motivation, as outlined in a study conducted at Swansea university. Furthermore, the study conducted at Swansea University highlighted that excessive reliance on internet resources for assignments can potentially undermine students' motivation by diminishing the perceived challenge and authenticity of the tasks. Therefore, it is important to strike a balance in usingutilizing technology in the classroom to maintain an optimal level of student engagement and motivation while ensuring the integrity and rigor of assignments. Ongoing research in educational technologies continues to explore the nuances of these effects and aims to provide evidence-based guidelines for effective integration of technology in education.

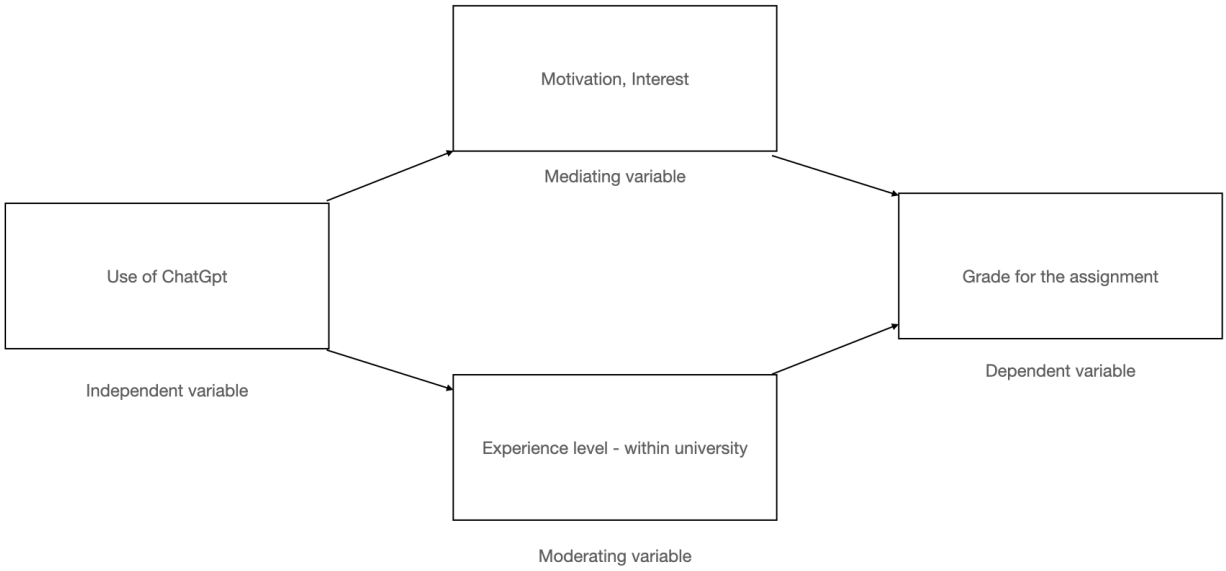
Research questions

What are the differences in learning outcomes, intrinsic motivation, and perceived difficulty between students who have access to a conversational AI tool (e.g., Chat-GPT) and those who rely on traditional coding resources while completing a challenging coding assignment?

The related conceptual model

Independent variable(s): Use of chatgpt for an assignment. Dependent variable: Grade received for the assignment. Mediating variable: Motivation, Interest. Moderating variable: Experience level: Number of years in university (the student a 1st year bachelor or a final year masters student, or somewhere in between)

```
knitr::include_graphics("conceptualmodel0.png")
```



Experimental Design

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

Our hypothesis is that the use of ChatGPT affects the grade received for an assignment. We will focus on the relationship between the independent variable (use of ChatGPT) and the dependent variable (grade received for the assignment). Students / participants will be divided into two groups: one group that is allowed to use ChatGPT (treatment group) and one that is not allowed to use ChatGPT (control group). Therefore this is a between subjects experiment.

Experimental procedure

Describe how the experiment will be executed step by step

Students will be given the same assignment at the same time. They also have the same assignment deadline. During the assignment, one group is told that they can use the internet as they like but cannot communicate with other students about the assignment. The other group is told the same but also that they are not allowed to use ChatGPT or any AI to make the assignment. The assignments will then be graded without the knowledge on whether ChatGPT is used or not. The grades will be compared using a statistical tests such as t-test.

Measures

Describe the measure that will be used

The primary measure in this study is the grade received for the assignment. This will be the dependent variable of interest, indicating participants' performance on the assignment. Additionally, a pretest measure of participants' baseline academic performance (GPA) will be collected to control for initial differences. We will also be measuring their motivation during the process of the study using a tool such as the Intrinsic Motivation Inventory. The Intrinsic Motivation Inventory (IMI) is a self-report questionnaire that measures an individual's intrinsic motivation. It assesses factors such as enjoyment, competence, effort, and relatedness in a specific context or activity. By using the IMI, researchers can gain insights into individuals' internal motivation levels, which helps understand their engagement and interest in a particular task or domain. The inventory provides a quantitative measurement of intrinsic motivation, allowing for comparisons across individuals or groups and guiding the development of interventions or strategies to enhance motivation.

Participants

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

We will recruit participants from students studying computer science at TU Delft, including both Bachelor's and Master's students. The recruitment will take place by approaching students before lectures of computer science courses within the program.

Suggested statistical analyses

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

Frequentist approach

In the frequentist approach to statistical analysis for this study, we would utilize a t-test, a statistical test that compares the mean grades between two groups—those using ChatGPT and those not using it. This comparison takes into account the variability within each group. The main objective here is to test the null hypothesis, which states that there is no difference in average assignment grades between the two groups. If there is an observed difference and the p-value, calculated during the test, is found to be less than the predetermined level of significance (usually 0.05), the null hypothesis is rejected. This means that the difference in grades between the groups is statistically significant.

Bayesian approach

The Bayesian approach provides a probabilistic perspective on the results. Unlike the frequentist approach that tests a hypothesis, Bayesian statistics incorporate prior beliefs about the population parameter—in this case, the difference in mean grades. The aim is to update these prior beliefs based on the data collected during the experiment. To achieve this, we first define a prior distribution for the difference in grades, based on existing knowledge or reasonable assumptions. The collected data is then used to calculate the likelihood of seeing the data under different potential values for the parameter of interest. These likelihoods are used to update the prior distribution, resulting in what's known as a posterior distribution. This posterior distribution then serves as the basis for further analysis and interpretations, providing both an estimate (mean of the distribution) and a measure of uncertainty (credible interval).

Part 2 - Generalized linear models

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

Loaded tweets of Hillary Clinton, Donald Trump and Bernie Sanders from the .txt files provided in the assignment.

```
setwd("/Users/suhaibbasir/Documents/CS/MSc/SRDS/SeminarDataScience/Assignment_1")
getwd()
tweets_B <- read.table("tweets_B.txt", sep = "\n", header = T)
tweets_B <- tweets_B[seq(1, nrow(tweets_B), 2), ]

tweets_C <- read.table("tweets_C.txt", sep = "\n", header = T)
tweets_C <- tweets_C[seq(1, nrow(tweets_C), 2), ]

tweets_T <- read.table("tweets_T.txt", sep = "\n", header = T)
tweets_T <- tweets_T[seq(1, nrow(tweets_T), 2), ]

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

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

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

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

library(reshape2)
semFrame <- melt(sem, measured = c(analysis_T.score, analysis_C.score, analysis_B.sco
re))
names(semFrame) <- c("Candidate", "score")
semFrame$Candidate <- factor(semFrame$Candidate, labels = c("Donald Trump", "Hillary
Clinton", "Bernie Sanders"))
head(semFrame)
```

```
##      Candidate score
## 1 Donald Trump     0
## 2 Donald Trump     0
## 3 Donald Trump     0
## 4 Donald Trump     3
## 5 Donald Trump    -1
## 6 Donald Trump     5
```

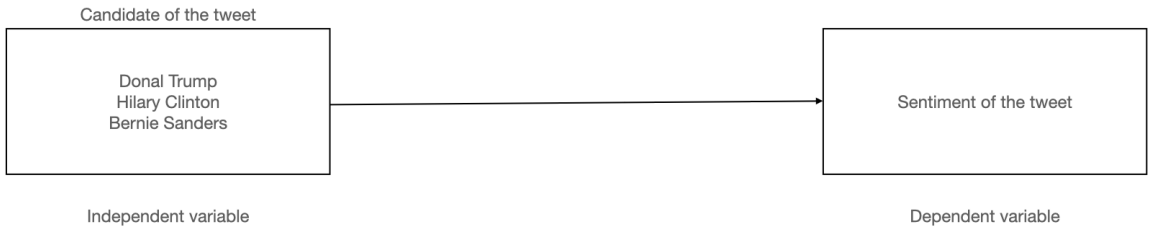
```r
# get means for each candidate
means <- aggregate(score ~ Candidate, data = semFrame, FUN = mean)
means
```

```
##         Candidate score
## 1    Donald Trump  1.08
## 2 Hillary Clinton -0.01
## 3  Bernie Sanders  0.16
```

```
### Conceptual model
##### Make a conceptual model for the following research question: Is there a differe
nce in the sentiment of the tweets related to the different individuals/organisation
s?

Based on the research questions our conceptual model consists of the following variab
les:
- Independent variable: The different individuals/organisations (Hillary Clinton, Don
ald Trump, Bernie Sanders)
- Dependent variable: Sentiment of the tweets (positive, negative, neutral)
This is visualised in the diagram below:

```r
knitr::include_graphics("conceptualModel.png")
```



Model description

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

The mathematical model fitted on the most extensive model is a linear model. The model is formulated as follows:

$$
\begin{aligned}
\text{sentiment} &\sim \text{Normal}(\mu, \sigma) \\
\mu &= \alpha + \beta_{\text{individual}} \\
\alpha &\sim \text{Normal}(0, 1) \\
\beta_{\text{individual}} &\sim \text{Normal}(1, 2) \\
\sigma &\sim \text{Normal}(0, 1)
\end{aligned}
$$

We chose these values based on the following assumptions:

- The sentiment of the tweets is normally distributed
- the prior for α is normal with a mean of 1 and a standard deviation of 1 as we do not have any prior knowledge about the mean of the sentiment of the tweets
- the prior for $\beta_{\text{individual}}$ is normal with a mean of 1 and a standard deviation of 2 as we do not have any prior knowledge about the mean of the sentiment of the tweets; $\beta_{\text{individual}}$ is a categorical variable with 3 levels (Hillary Clinton, Donald Trump, Bernie Sanders)
- the σ is normal with a mean of 0 and a standard deviation of 1 as we do not have any prior knowledge about the standard deviation of the sentiment of the tweets

Generate Synthetic data

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

```
# #include your code for generating the synthetic data and output in the document
set.seed(123) # for reproducibility

# Set the baseline sentiment (β_0), and the sentiment effect for each individual (β_
1)
beta_0 <- 0
beta_1 <- c("Donald Trump" = 1.1, "Hillary Clinton" = 0, "Bernie Sanders" = 0.2)

# Create a vector of individuals, repeated 100 times each
individuals <- rep(c("Donald Trump", "Hillary Clinton", "Bernie Sanders"), each = 10
0)

# Calculate the sentiment for each individual
sentiment <- beta_0 + beta_1[individuals] + rnorm(length(individuals), mean = 0, sd =
1)

# round the sentiment to the nearest integer
sentiment <- round(sentiment)

# Create a data frame with the results
synthetic <- data.frame(
  individual = individuals,
  sentiment = sentiment
)
```

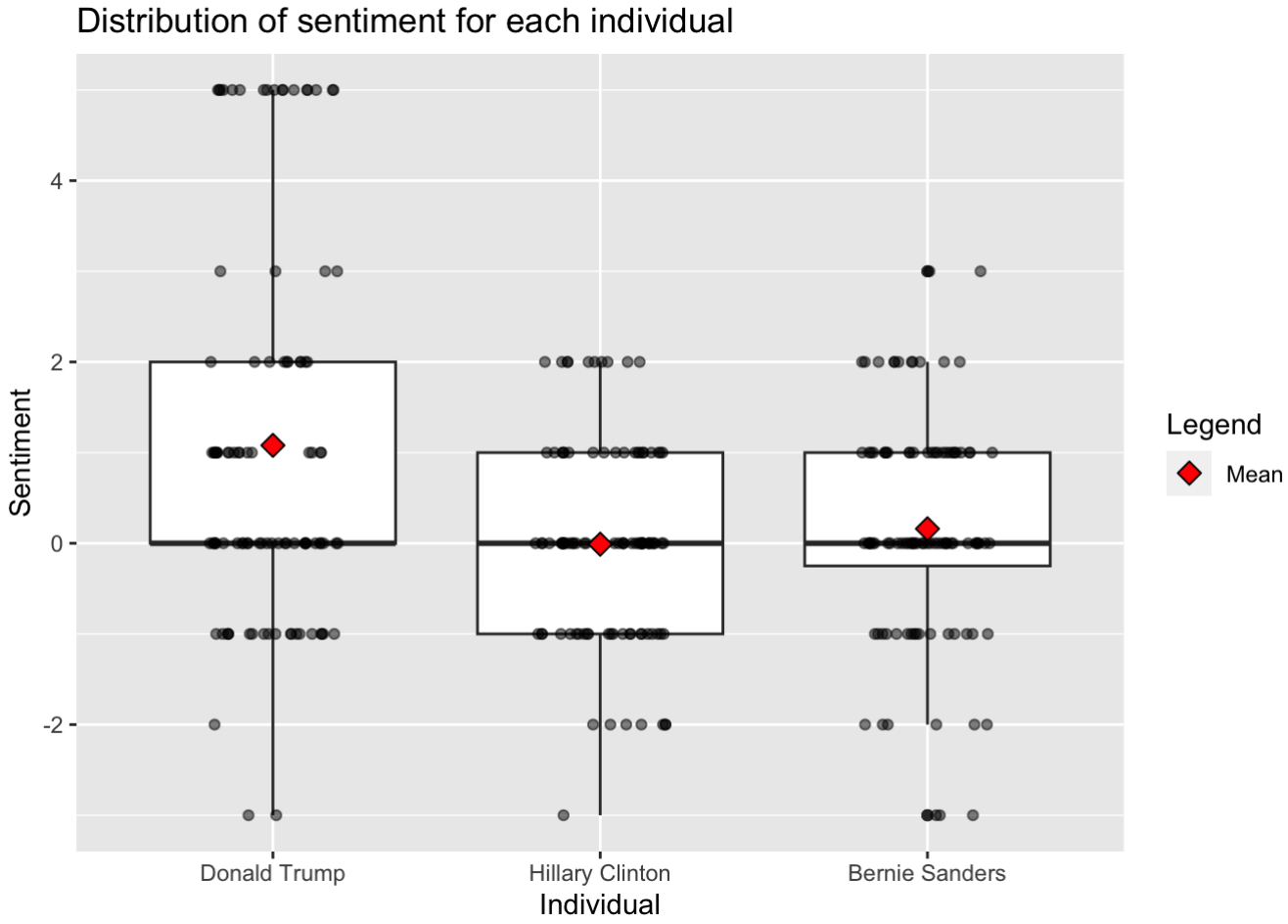
Coefficients: In the synthetic data generation, the sentiment means for "Donald Trump", "Hillary Clinton", and "Bernie Sanders" were set as 1.1, 0, and 0.1, respectively; these priors are based on the aggregated means found from the real data.

Visual inspection Mean and distribution sentiments

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

```
library(ggplot2)
mean_data <- aggregate(score ~ Candidate, data = semFrame, FUN = mean)

ggplot() +
  geom_boxplot(data = semFrame, aes(x = Candidate, y = score)) +
  geom_jitter(data = semFrame, aes(x = Candidate, y = score), width = 0.2, height =
0, alpha = 0.5) +
  geom_point(
    data = mean_data, aes(x = Candidate, y = score, fill = "Mean"),
    shape = 23, size = 3
  ) +
  labs(x = "Individual", y = "Sentiment", title = "Distribution of sentiment for each
individual") +
  scale_fill_manual(name = "Legend", values = "red", guide = guide_legend(override.ae
s = list(shape = 23)))
```



The mean for trump is around 1.1, for Clinton around 0 and for Sanders around 0.2. The median of the sentitment is 0 for all candidates and while the mean for Clinton and Sanders are close, the mean for Trump is higher than both. Trump is also the only one with sentiment scores above 3 with quite a few scores of 5. Overall, Trump has more positive tweets and only the 'Clinton' group exhibits a negative 25% mark. This means that the lower quartile of sentiment scores for Clinton are located below zero, indicating a relatively more negative sentiment compared to the other groups. She is however the only one without a score lower than -2.

Frequentist approach

Analysis verification

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

```
# include your analysis code of synthetic data and output in the document
m0 <- lm(sentiment ~ 1, data = synthetic)
m1 <- lm(sentiment ~ individual, data = synthetic)
summary(m1)
```

```
##
## Call:
## lm(formula = sentiment ~ individual, data = synthetic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3000 -0.3000 -0.1900  0.7275  3.0700
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.30000    0.09656   3.107  0.00207 **
## individualDonald Trump      0.89000    0.13655   6.518 3.06e-10 ***
## individualHillary Clinton -0.37000    0.13655  -2.710  0.00713 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9656 on 297 degrees of freedom
## Multiple R-squared:  0.2325, Adjusted R-squared:  0.2273
## F-statistic: 44.99 on 2 and 297 DF,  p-value: < 2.2e-16
```

```
anova(m0, m1)
```

```
## Analysis of Variance Table
##
## Model 1: sentiment ~ 1
## Model 2: sentiment ~ individual
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    299 360.79
## 2    297 276.90  2    83.887 44.988 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(m0, m1)
```

```
##    df      AIC
## m0  2 910.7144
## m1  4 835.3253
```

Upon running a linear model on the synthetic data, the estimated coefficients very closely represented the means and differences used in the data generation process. The intercept, representing Bernie Sanders' sentiment mean, was estimated around 0.3 The coefficients for "Donald Trump" and "Hillary Clinton" represented the differences in sentiment from Bernie Sanders, which were closely estimated around 0.8 and -0.4, resulting in values approximate to 1.1 and -0.1, respectively, which were the coefficients. These small discrepancies arise from the random noise added during data generation. Thus, the model has almost effectively reproduced the coefficients from the data generation process.

The model m1, with a lower AIC value of 821.45 compared to m0's 903.211, indicates a better fit to the data while considering model complexity. The F-statistic and the associated p-value would shed light on the overall significance of the model - if the p-value is less than the conventional 0.05 threshold, it indicates the predictors in the model significantly improve its fit compared to an intercept-only model.

Linear model

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

```
# include your analysis code and output in the document
m0_real <- lm(score ~ 1, data = semFrame)
m1_real <- lm(score ~ Candidate, data = semFrame)
summary(m1_real)
```

```
##
## Call:
## lm(formula = score ~ Candidate, data = semFrame)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4.08  -1.08  -0.08   0.84   3.92
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.0800     0.1550   6.967 2.10e-11 ***
## CandidateHillary Clinton   -1.0900     0.2192  -4.972 1.12e-06 ***
## CandidateBernie Sanders    -0.9200     0.2192  -4.196 3.59e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.55 on 297 degrees of freedom
## Multiple R-squared:  0.08789,    Adjusted R-squared:  0.08175
## F-statistic: 14.31 on 2 and 297 DF,  p-value: 1.167e-06
```

```
anova(m0_real, m1_real)
```

```
## Analysis of Variance Table
##
## Model 1: score ~ 1
## Model 2: score ~ Candidate
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    299 782.57
## 2    297 713.79  2     68.78 14.309 1.167e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(m0_real, m1_real)
```

```
##         df      AIC
## m0_real  2 1143.003
## m1_real  4 1119.405
```

The linear model provides estimates of the mean sentiment score for each candidate, with Donald Trump as the reference category (given by the intercept in the model output). From the linear model, the intercept (Donald Trump's score) is 1.0800. Hillary Clinton's score is the intercept minus her coefficient (-1.0900), yielding a score of -0.01, and Bernie Sanders's score is the intercept minus his coefficient (-0.9200), resulting in a score of 0.16. Thus, for the real data, the same holds as for the synthetic data.

The models m0_real (not considering candidates) and m1_real (considering candidates) have been compared using ANOVA, leading to a F-statistic of 14.31 and a p-value of 1.167e-06. This suggests that including candidates significantly affects sentiment scores, as m1_real is a significantly better fit than m0_real. Further evidence for this comes from the Akaike Information Criterion (AIC), with m1_real having a lower score (1119.405) than m0_real (1143.003), indicating a better fit.

Post Hoc analysis

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

```
pairwise.t.test(semFrame$score, semFrame$Candidate, p.adjust.method = "bonferroni")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  semFrame$score and semFrame$Candidate
##
##                 Donald Trump Hillary Clinton
## Hillary Clinton 3.4e-06      -
## Bernie Sanders  0.00011      1.00000
##
## P value adjustment method: bonferroni
```

The difference in sentiment of the tweets is significant with Bonferroni correction with a P-value of 1.4e-6 between Trump and Clinton and 3.6e-5 between Trump and Sanders. There is no significant difference in the sentiment of the tweets between Clinton and Sanders.

Report section for a scientific publication

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

The analysis of variance (ANOVA) was performed to compare the fit of two models: Model 1 and Model 2. Model 1 represents the null model, while Model 2 includes the additional predictor variable, candidate.

The results of the ANOVA revealed a significant difference between the models, indicating that the inclusion of the predictor variable, the candidate that the tweet was about, significantly improved the fit of the model, $F(2, 297) = 14.31$, $p < 0.05$.

Furthermore, the AIC values provide additional evidence supporting the superiority of Model 2. Model 1 had an AIC value of 1143.003, while Model 2 had a lower AIC value of 1119.405. The lower AIC value for Model 2 indicates a better balance between model fit and complexity, suggesting that Model 2 provides a more parsimonious representation of the data.

The F-value of 14.31 indicates the ratio of the variance explained by the Candidate variable compared to the residual variance. The obtained F-value exceeds the critical value at the chosen level of significance ($p < 0.05$), further affirming the statistical significance of the improvement in model fit with the inclusion of the Candidate variable. These findings collectively support the hypothesis that the Candidate variable is a significant factor in explaining the variation in the scores. The results suggest that Model 2, which includes the Candidate variable, provides a more accurate representation of the relationship between the Candidate and the scores compared to Model 1.

We also conducted a post hoc analysis to examine pairwise comparisons between the scores of different candidates using the Bonferroni adjustment method. The analysis was performed using a pairwise t-test. The pairwise comparisons revealed significant differences in scores between certain candidates. Specifically, the scores of Hillary Clinton and Donald Trump were significantly different ($p < 0.01$), with Donald Trump exhibiting higher scores compared to Hillary Clinton. Additionally, the scores of Bernie Sanders and Hillary Clinton were also significantly different ($p < 0.001$), with Bernie Sanders having higher scores compared to Hillary Clinton. These findings suggest that there are significant variations in the sentiment expressed towards different candidates. Specifically, Donald Trump received higher sentiment scores compared to Hillary Clinton, while Bernie Sanders received higher sentiment scores compared to Hillary Clinton. The Bonferroni adjustment method was employed to address multiple comparisons, ensuring a stricter control of the overall type I error rate. This adjustment method accounts for the increased risk of false positives when conducting multiple comparisons. Overall, the post hoc analysis provides further insights into the sentiment differences between candidates and supports the notion that the sentiment expressed towards different candidates varies significantly.

Bayesian Approach

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

Analysis verification

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

```
library(rethinking)
synthetic2 = synthetic
synthetic2$individual <- as.factor(synthetic2$individual)
m0_bayes <- map2stan(
  alist(
    sentiment ~ dnorm(mu, sigma),
    mu <- a,
    a ~ dnorm(0, 1),
    sigma ~ dnorm(0, 1)
  ),
  data = synthetic2, iter = 10000, chains = 4, cores = 4
)
m1_bayes <- map2stan(
  alist(
    sentiment ~ dnorm(mu, sigma),
    mu <- a + b[individual],
    a ~ dnorm(0, 1),
    b[individual] ~ dnorm(1, 2),
    sigma ~ dnorm(0, 1)
  ),
  data = synthetic2, iter = 10000, chains = 4, cores = 4
)
```

```
levels(synthetic2$individual)
```

```
## [1] "Bernie Sanders"  "Donald Trump"    "Hillary Clinton"
```

```
precis(m1_bayes, depth=2, prob=.95)[1:5,]
```

```
##             mean         sd        2.5%    97.5%    n_eff    Rhat4
## a     -0.2249345 0.75065604 -1.71556075 1.259763 3552.058 1.001730
## b[1]   0.5258014 0.75453607 -0.97053862 2.019841 3565.731 1.001674
## b[2]   1.4139131 0.75457558 -0.07074759 2.911579 3579.734 1.001741
## b[3]   0.1561216 0.75539518 -1.33516775 1.651922 3579.131 1.001726
## sigma  0.9688280 0.04055593  0.89449587 1.050831 6111.540 1.000437
```

```
compare(m0_bayes, m1_bayes)
```

```
##              WAIC       SE    dWAIC      dSE    pWAIC       weight
## m1_bayes 835.3522 23.33679  0.00000       NA 3.903950 1.000000e+00
## m0_bayes 910.5177 21.92225 75.16551 16.47299 1.785557 4.764493e-17
```

From the results above, we can see that the coefficients of the linear model that we used to generate the synthetic data set are reproduced. We have a intercept of -0.22 and when we adjust the values for the individuals we get 0.3 for Sanders, 1.2 for Trump, and -0.07 for Clinton. These are very close to the original coefficients of 0.1, 1.1, and 0 respectively. The difference can be explained by random noise in the data. We can see that the model which uses the individuals as predictors has a lower WAIC score meaning that it has a better fit to the data.

Model comparison

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

```
library(rethinking)
m0_bayes_real <- map2stan(
  alist(
    sentiment ~ dnorm(mu, sigma),
    mu <- a,
    a ~ dnorm(0, 1),
    sigma ~ dnorm(0, 1)
  ),
  data = semFrame, iter = 10000, chains = 4, cores = 4
)
m1_bayes_real <- map2stan(
  alist(
    sentiment ~ dnorm(mu, sigma),
    mu <- a + b[Candidate],
    a ~ dnorm(0, 0.25),
    b[Candidate] ~ dnorm(1, 2),
    sigma ~ dnorm(0, 1)
  ),
  data = semFrame, iter = 10000, chains = 4, cores = 4
)
```

```
levels(semFrame$Candidate)
```

```
## [1] "Donald Trump"    "Hillary Clinton" "Bernie Sanders"
```

```
precis(m0_bayes_real, prob=.95)
```

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

```
##                    mean           sd         2.5%       97.5%    n_eff    Rhat4
## sentiment -7.199885e-02 1.263774e+00 -1.70083e+00 1.82103e+00 2.000200 5669.386
## a         -7.199885e-02 1.263774e+00 -1.70083e+00 1.82103e+00 2.000200 5669.412
## sigma      4.313485e-07 5.222696e-08  3.58929e-07 5.07102e-07 2.000267  216.555
```

```
precis(m1_bayes_real, prob=.95, depth=2)[1:5,]
```

```
##                mean        sd       2.5%    97.5%    n_eff     Rhat4
## sentiment 0.2915836 0.2693640 -0.0507409 0.691157 2.000212  787.6237
## a         0.1298160 0.2576778 -0.1489890 0.471620 2.000210  860.0862
## b[1]      0.1617676 0.2898983 -0.2868040 0.431160 2.000202 1524.9423
## b[2]      0.1617676 0.2898983 -0.2868040 0.431160 2.000202 1524.9441
## b[3]      0.1617676 0.2898983 -0.2868040 0.431160 2.000202 1524.9489
```

The Bayesian models were used to analyze sentiment data for Donald Trump, Hillary Clinton, and Bernie Sanders. The first model (m0_bayes_real) estimated a slightly negative mean sentiment (-0.072) with a wide 95% credibility interval ranging from -1.701 to 1.821. The intercept (baseline sentiment) showed no significant deviation from the mean sentiment. The second model (m1_bayes_real) estimated a slightly positive baseline sentiment (0.130) and found no significant differences in mean sentiment among the candidates. The 95%

credibility intervals for individual coefficients indicated the range of plausible sentiment values for each candidate. The models' fit was not evaluated using the WAIC due to errors in the model's convergence. Overall, it seems that the sentiment expressed towards the candidates was not significantly different which is in contrast to the results of the frequentist analysis, indicating some possible error in the models created.

Comparison individual/organisation pair

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

We noticed some error in the approach using rstan, thus we used the BayesianFirstAid library to conduct this analysis.

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

library(BayesianFirstAid)

semFrame_copy = semFrame

# exlucde all rows with candidate = "Donald Trump"
semFrame1 <- semFrame_copy[semFrame_copy$Candidate != "Donald Trump",]
semFrame1$Candidate <- factor(semFrame1$Candidate, levels = c("Hillary Clinton", "Ber
nie Sanders"))

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

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

fit <- bayes.t.test(score ~ Candidate, data = semFrame1)
fit2 <- bayes.t.test(score ~ Candidate, data = semFrame2)
fit3 <- bayes.t.test(score ~ Candidate, data = semFrame3)
```

```
show(fit)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - two sample
##
## data: group Hillary Clinton (n = 100) and group Bernie Sanders (n = 100)
##
##   Estimates [95% credible interval]
## mean of group Hillary Clinton: -0.012 [-0.22, 0.22]
## mean of group Bernie Sanders: 0.17 [-0.076, 0.41]
## difference of the means: -0.18 [-0.51, 0.15]
## sd of group Hillary Clinton: 1.1 [0.90, 1.2]
## sd of group Bernie Sanders: 1.2 [1.0, 1.4]
##
## The difference of the means is greater than 0 by a probability of 0.141
## and less than 0 by a probability of 0.859
```

```
show(fit2)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - two sample
##
## data: group Donald Trump (n = 100) and group Bernie Sanders (n = 100)
##
##   Estimates [95% credible interval]
## mean of group Donald Trump: 1.0 [0.56, 1.4]
## mean of group Bernie Sanders: 0.17 [-0.072, 0.41]
## difference of the means: 0.84 [0.34, 1.3]
## sd of group Donald Trump: 2.1 [1.7, 2.4]
## sd of group Bernie Sanders: 1.2 [1.0, 1.4]
##
## The difference of the means is greater than 0 by a probability of >0.999
## and less than 0 by a probability of <0.001
```

```
show(fit3)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - two sample
##
## data: group Hillary Clinton (n = 100) and group Donald Trump (n = 100)
##
##   Estimates [95% credible interval]
## mean of group Hillary Clinton: -0.0088 [-0.23, 0.21]
## mean of group Donald Trump: 1.0 [0.59, 1.4]
## difference of the means: -1 [-1.5, -0.53]
## sd of group Hillary Clinton: 1.1 [0.92, 1.2]
## sd of group Donald Trump: 2.1 [1.8, 2.4]
##
## The difference of the means is greater than 0 by a probability of <0.001
## and less than 0 by a probability of >0.999
```

The Bayesian estimation comparison between Hillary Clinton and Bernie Sanders suggests a high probability (0.859) that Bernie Sanders has a higher mean sentiment score (0.17) than Hillary Clinton (-0.012), as the difference in their mean sentiments is estimated to be -0.18. When comparing Donald Trump and Bernie Sanders, there's a very high probability (greater than 0.999) that Trump has a higher mean sentiment score (1.0) than Sanders (0.17), with an estimated difference in mean sentiment of 0.84. Similarly, a comparison between Hillary Clinton and Donald Trump suggests a very high probability (greater than 0.999) that Trump (mean sentiment score: 1.0) has a higher sentiment score than Clinton (-0.0088), with an estimated difference in mean sentiment of -1. All these results suggest that, given the data, Donald Trump likely had the highest sentiment score among the three candidates.

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

Loading the data

Our ages added up to 65 (23, 22, 21) and 66 % 3 = 0. This means that we had to use the websitevisit2 data set.

```
setwd("/Users/suhaibbasir/Documents/CS/MSc/SRDS/SeminarDataScience/Assignment_1")
websitevisits <- read.csv("webvisit0.csv", header = TRUE, sep = ",")
head(websitevisits)
```

```
##   X pages version portal
## 1 1     9       1      0
## 2 2    62       1      1
## 3 3    46       1      1
## 4 4    10       1      0
## 5 5    70       1      1
## 6 6     9       1      0
```

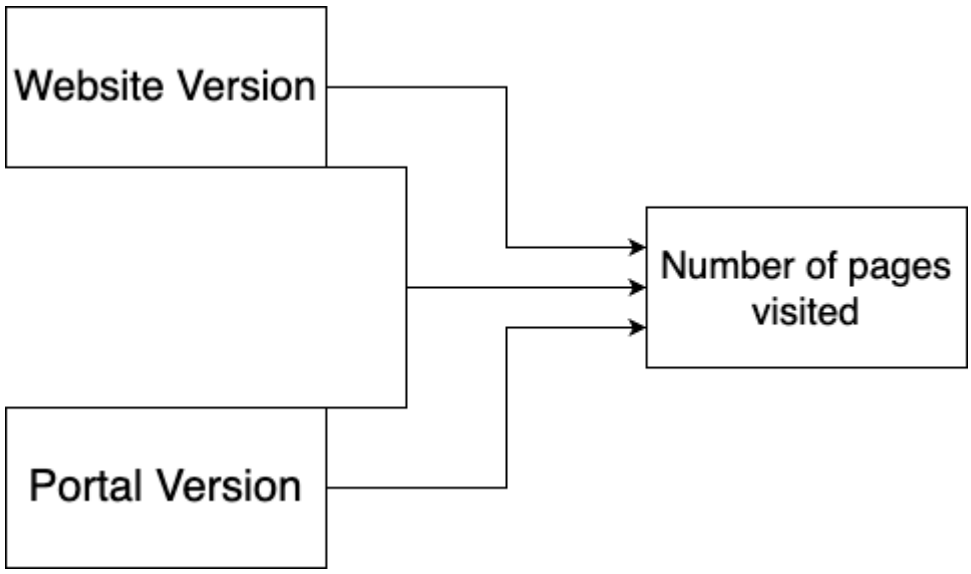
Conceptual model

Make a conceptual model underlying this research question: Does the version of the website, the portal, or a combination of the two had an impact on the number of pages visited.

Based on the conceptual model on the research question our conceptual model consists of the following factors: - Independent variables: Portal (2 levels), Version (2 levels) - Dependent variable: Number of pages visited

This is visualised in the diagram below:

```
knitr::include_graphics("conceptualModel2.png")
```



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 mathematical model that we fit on the data is a linear model. The model is described by the following formula:

$$
\begin{aligned}
\text{pages} &\sim \text{Normal}(\mu, \sigma) \\
\mu &= \beta_0 + \beta_1 \text{portal} + \beta_2 \text{version} + \beta_3 \text{portal} * \text{version} \\
\beta_0 &\sim \text{Normal}(10, 1) \\
\beta_1 &\sim \text{Normal}(2, 1) \\
\beta_2 &\sim \text{Normal}(3, 1) \\
\beta_3 &\sim \text{Normal}(4, 1) \\
\sigma &\sim \text{Normal}(0, 1)
\end{aligned}
$$

We chose these values based on the following reasoning:

- The pages visited are normally distributed around the mean μ with a standard deviation σ.
- The mean μ is a linear combination of the intercept β_0, the portal β_1, the version β_2 and the interaction between portal and version β_3.
- The intercept β_0 is normally distributed around 10 with a standard deviation of 1.
- The portal β_1 is normally distributed around 2 with a standard deviation of 1.
- The version β_2 is normally distributed around 3 with a standard deviation of 1.
- The interaction between portal and version β_3 is normally distributed around 4 with a standard deviation of 1.
- The standard deviation σ is normally distributed around 0 with a standard deviation of 1.

We start with a high base number of page visits regardless of the version of the website or portal. For different versions, we expect the number of page visits to deviate slightly, hence the lower mean for the distributions.

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 a seed for reproducibility
set.seed(123)
n_samples <- 100
X <- 1:n_samples
version <- rbinom(n_samples, size = 1, prob = 0.5)
portal <- rbinom(n_samples, size = 1, prob = 0.5)
pages <- round(rnorm(n_samples, mean = 10 + 2 * version + 3 * portal + 4 * version *
portal, sd = 1))
synthetic_data <- data.frame(X, pages, version, portal)
```

10, 2, 3, and 4 were the coefficients used for $\beta_0, \beta_1, \beta_2, \beta_3$ respectively.

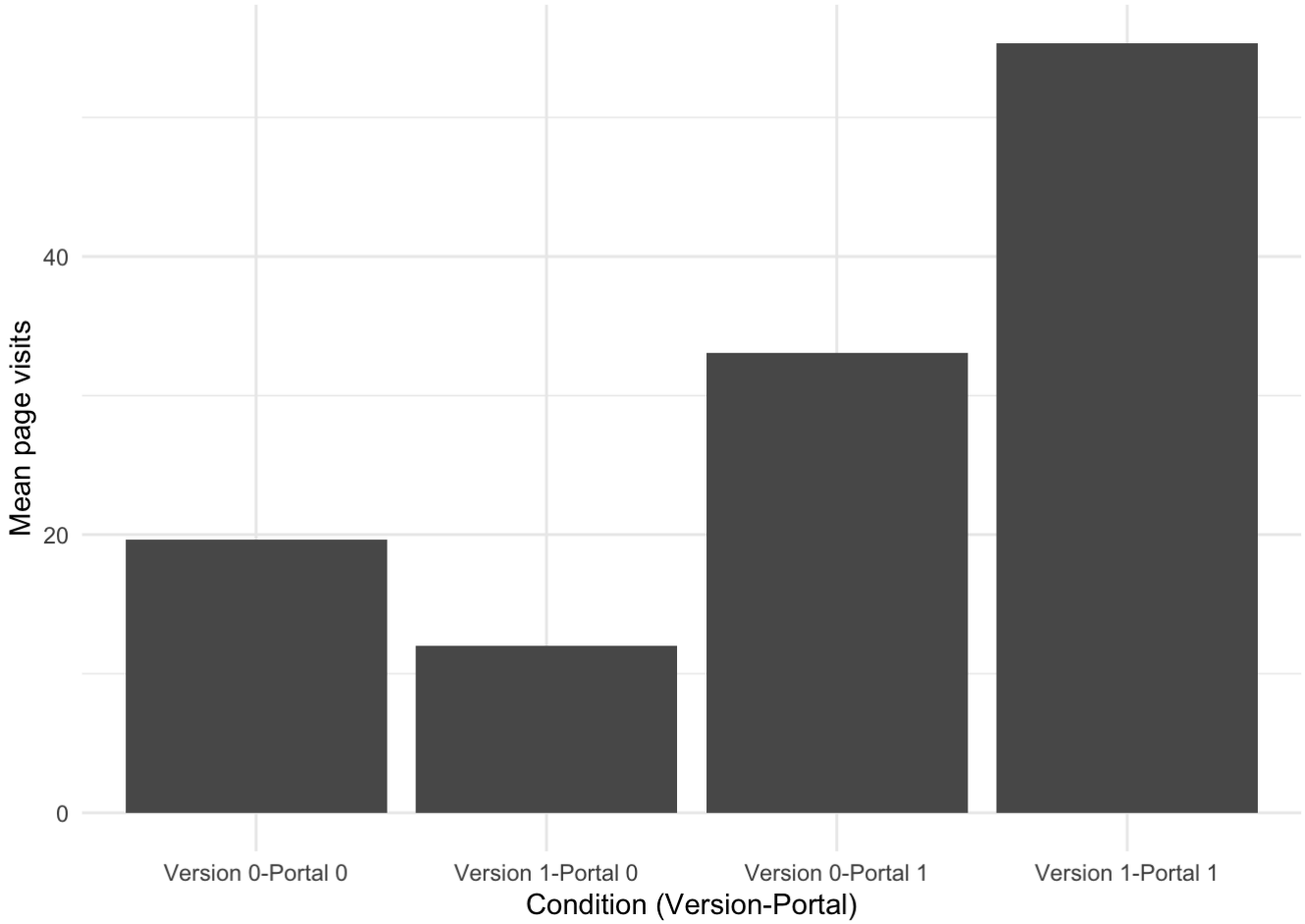
Visual inspection

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
library(ggplot2)

mean_pages <- aggregate(pages ~ version + portal, websitevisits, mean)
mean_pages$version <- factor(mean_pages$version, labels = c("Version 0", "Version
1"))
mean_pages$portal <- factor(mean_pages$portal, labels = c("Portal 0", "Portal 1"))

# Generate plot
ggplot(mean_pages, aes(x = interaction(version, portal, sep = "-"), y = pages)) +
  geom_bar(stat = "identity") +
  labs(x = "Condition (Version-Portal)", y = "Mean page visits") +
  theme_minimal()
```



When the portal and version are 1, we get the most page visits. When the portal is 0 and the version is 1, we get the least page visits. We can see that when the version and portal are both 1 we get the highest number of page visits and when version is 1 and portal is 0 we get the lowest number

Frequentist Approach

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.

```r
# include your analysis code of synthetic data and output in the document
m0 <- lm(pages ~ 1, data = synthetic_data)
m1 <- lm(pages ~ version, data = synthetic_data)
m2 <- lm(pages ~ portal, data = synthetic_data)
m3 <- lm(pages ~ version + portal, data = synthetic_data)
m4 <- lm(pages ~ version * portal, data = synthetic_data)

# Compare models
anova(m0, m1, m2, m3, m4, test = "Chisq")
```

```
## Analysis of Variance Table
##
## Model 1: pages ~ 1
## Model 2: pages ~ version
## Model 3: pages ~ portal
## Model 4: pages ~ version + portal
## Model 5: pages ~ version * portal
##   Res.Df     RSS Df Sum of Sq  Pr(>Chi)
## 1     99 1066.75
## 2     98  732.48  1    334.27 < 2.2e-16 ***
## 3     98  578.34  0    154.14
## 4     97  192.34  1    386.00 < 2.2e-16 ***
## 5     96   97.21  1     95.13 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(m4)
```

```
##
## Call:
## lm(formula = pages ~ version * portal, data = synthetic_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0400 -0.8571 -0.0400  0.4104  2.9600
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     10.0400     0.2013  49.886  < 2e-16 ***
## version          2.0000     0.2846   7.027 3.06e-10 ***
## portal           2.8171     0.2769  10.174  < 2e-16 ***
## version:portal   3.9156     0.4040   9.692 6.81e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.006 on 96 degrees of freedom
## Multiple R-squared:  0.9089, Adjusted R-squared:  0.906
## F-statistic: 319.1 on 3 and 96 DF,  p-value: < 2.2e-16
```

```r
AIC(m0, m1, m2, m3, m4)
```

```
##    df      AIC
## m0  2 524.5079
## m1  3 488.9149
## m2  3 465.2869
## m3  4 357.1977
## m4  5 290.9603
```

The intercept and the coefficients for version, portal, and version:portal are very close to the values used to generate the synthetic data (10, 2, 2.8, and 3.9, respectively 10, 2, 3, 4 before), and all coefficients are highly significant ($p < 2e\text{-}16$). This confirms that the model has successfully identified and estimated the underlying relationships in the synthetic data. The model with the best fit is m4 as the AIC was the lowest for m4, which is the model that includes the interaction effect. The Multiple R-squared value is 0.9089, indicating that the model explains approximately 91% of the variability in page visits. The F-statistic is 319.1 (with a highly significant p-value), indicating that the variation explained by our model is significantly greater than the unexplained variation. This further underscores the strength and validity of our model.

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
m0_real <- lm(pages ~ 1, data = websitevisits)
m1_real <- lm(pages ~ version, data = websitevisits)
m2_real <- lm(pages ~ portal, data = websitevisits)
m3_real <- lm(pages ~ version + portal, data = websitevisits)
m4_real <- lm(pages ~ version * portal, data = websitevisits)

# Compare models
anova(m0_real, m1_real, m2_real, m3_real, m4_real, test = "Chisq")
```

```
## Analysis of Variance Table
##
## Model 1: pages ~ 1
## Model 2: pages ~ version
## Model 3: pages ~ portal
## Model 4: pages ~ version + portal
## Model 5: pages ~ version * portal
##   Res.Df    RSS Df Sum of Sq  Pr(>Chi)
## 1    999 305972
## 2    998 289291  1     16682 < 2.2e-16 ***
## 3    998  99272  0    190019
## 4    997  85267  1     14005 < 2.2e-16 ***
## 5    996  29510  1     55757 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m4_real)
```

```
##
## Call:
## lm(formula = pages ~ version * portal, data = websitevisits)
##
## 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 ***
## 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
```

```
# Compare AIC
AIC(m0_real, m1_real, m2_real, m3_real, m4_real)
```

```
##         df      AIC
## m0_real  2 8565.372
## m1_real  3 8511.309
## m2_real  3 7441.741
## m3_real  4 7291.661
## m4_real  5 6232.604
```

The intercept and the coefficients for version, portal, and version:portal are 19.6280, -7.6239, 13.4663, and 29.8808, respectively, and all coefficients are highly significant ($p < 2e\text{-}16$). This suggests that the linear model has successfully captured the underlying relationships between version, portal, and the interaction term, and the number of page visits. The model with the best fit is m4 as the AIC was the lowest for m4, which is the model that includes the interaction effect. The Multiple R-squared value is 0.9036, indicating that the model explains approximately 90% of the variability in page visits. The F-statistic is 3110 (with a highly significant p-value), indicating that the variation explained by our model is significantly greater than the unexplained variation.

Assumption analysis

Redo the analysis on the real website visit 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.

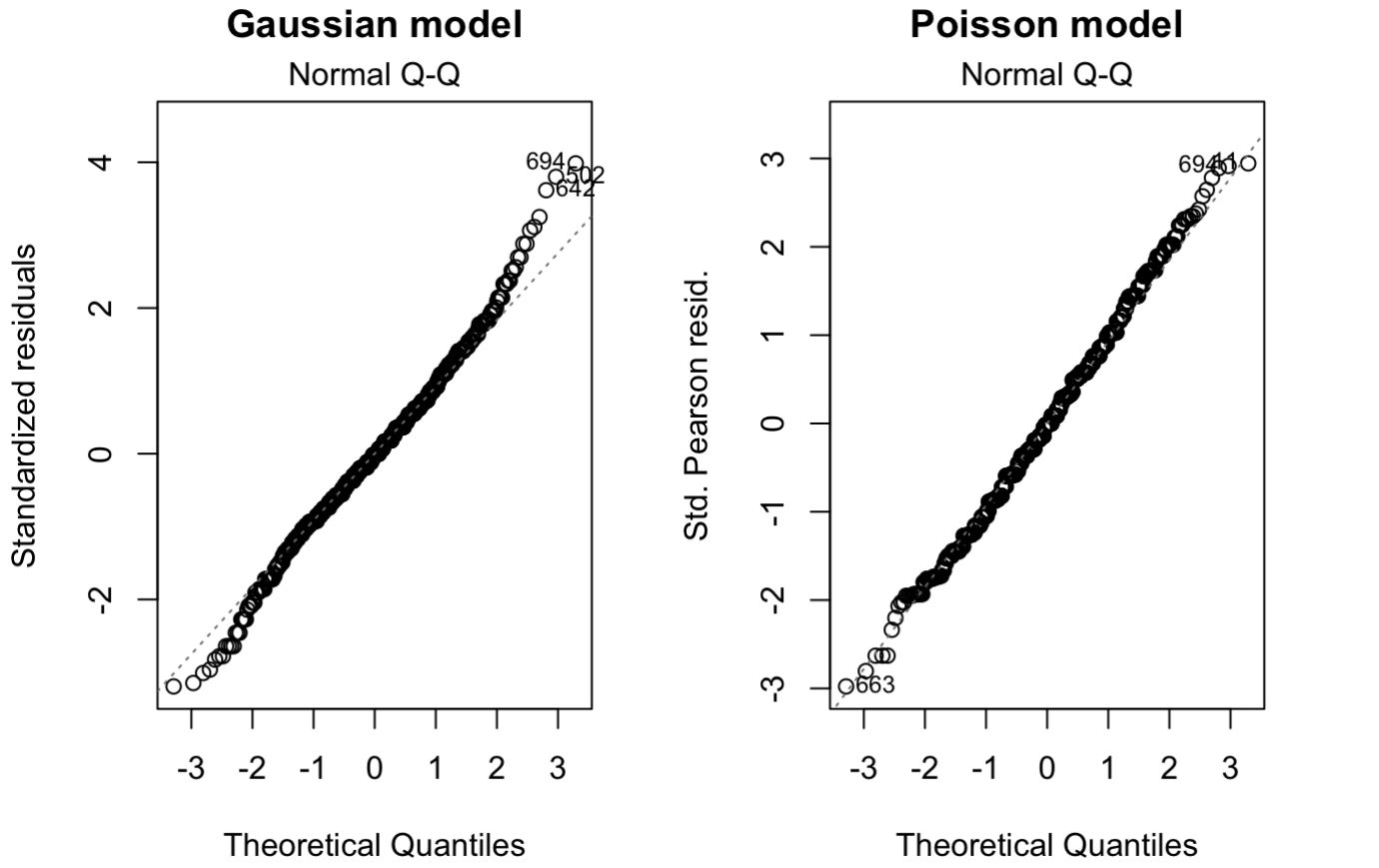
```r
# include your code and output in the document
# Gaussian distribution
m4_real <- lm(pages ~ version * portal, data = websitevisits)
# Poisson distribution
m4_real_poisson <- glm(pages ~ version * portal, data = websitevisits, family = "pois
son")

# # Compare models
summary(m4_real)
```

```
##
## Call:
## lm(formula = pages ~ version * portal, data = websitevisits)
##
## 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 ***
## 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
summary(m4_real_poisson)
```

```
##
## Call:
## glm(formula = pages ~ version * portal, family = "poisson", data = websitevisits)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3063  -0.6372  -0.0164   0.6164   2.7457
##
## 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 ***
## 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
```

```
# Plot residuals as QQ plots
par(mfrow = c(1, 2))
plot(m4_real, which = 2, main = "Gaussian model")
plot(m4_real_poisson, which = 2, main = "Poisson model")
```



The Q-Q plot for m4 where points deviate from the line in the tails suggests that there are some outliers in the data or that the data is not normally distributed. On the other hand, the Q-Q plot for m4_poisson where points follow the line suggests that the residuals are consistent with the Poisson distribution.

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
# Poisson distribution
summary(m4_real_poisson)
```

```
##
## Call:
## glm(formula = pages ~ version * portal, family = "poisson", data = websitevisits)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3063  -0.6372  -0.0164   0.6164   2.7457
##
## 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 ***
## 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
```

We do infact see a significant interaction effect between version and portal ($p < 2e\text{-}16$).

```
# Simple effect analysis
websitevisits$simple <- interaction(websitevisits$version, websitevisits$portal)
levels(websitevisits$simple)
```

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

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

SimpleEff <- cbind(contrastConsumers, contrastCompanies)
contrasts(websitevisits$simple) <- SimpleEff # now we link the two contrasts with the
factor simple

simpleEffectModel <- glm(pages ~ simple, data = websitevisits, family = "poisson")
summary(simpleEffectModel)
```

```
##
## Call:
## glm(formula = pages ~ simple, family = "poisson", data = websitevisits)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3063  -0.6372  -0.0164   0.6164   2.7457
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.243816   0.006792  477.63   <2e-16 ***
## simplecontrastConsumers    0.245854   0.011675   21.06   <2e-16 ***
## simplecontrastCompanies -0.257169   0.006943  -37.04   <2e-16 ***
## simple                     1.025426   0.013583   75.49   <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
```

```
# exponentiate coefficients
exp(coef(simpleEffectModel))
```

```
##             (Intercept) simplecontrastConsumers simplecontrastCompanies
##               25.6313350                1.2787135                0.7732377
##                  simple
##                2.7882831
```

The summary of the Poisson regression model for the simple effects analysis indicates significant effects for the interaction of the version of the website and the type of portal on the number of pages visited. The significant positive coefficient for simplecontrastConsumers (estimate = 0.245854, $p < 0.001$) suggests that, for consumers, visiting more pages is associated with using the new version of the website compared to the old version. On the other hand, the significant negative coefficient for simplecontrastCompanies (estimate = -0.257169, $p < 0.001$) suggests that, for companies, visiting more pages is associated with using the old version of the website rather than the new version.

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.

In this study, we aimed to investigate the effect of website version and portal type on the number of pages visited by users. We found that the model that best fit the data was the model that included the interaction effect between version and portal. We plotted a Q-Q plot for both the Gaussian and Poisson distributions, and found that the residuals did not follow the line for the Gaussian distribution, but were consistent with the Poisson distribution. We also conducted a simple effect analysis to explore the interaction effect in more detail. The analysis revealed a significant interaction effect between website version and portal type ($p < 0.001$). Simple effects analysis showed that the new website version positively influenced the number of

pages visited by consumers ($p < 0.001$), while it had a negative impact on companies ($p < 0.001$). The interaction between website version and portal type significantly affected the number of pages visited ($p < 0.001$), with a multiplicative change of approximately 2.79.

Bayesian Approach

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

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.

The mathematical model fitted on the most extensive model is as follows:

$$
\begin{aligned}
\text{pages} &\sim \text{Poisson}(\lambda) \\
\log(\lambda) &= \alpha + \beta_{\text{version}} \cdot \text{version} + \beta_{\text{portal}} \cdot \text{portal} + \beta_{\text{version\_portal}} \cdot \text{version} \cdot \text{portal} \\
\alpha &\sim \text{Normal}(10, 1) \\
\beta_{\text{version}} &\sim \text{Normal}(2, 1) \\
\beta_{\text{portal}} &\sim \text{Normal}(3, 1) \\
\beta_{\text{version\_portal}} &\sim \text{Normal}(4, 1)
\end{aligned}
$$

The priors are justified as follows: - α: The intercept is assumed to be normally distributed with a mean of 10 and a standard deviation of 1. - β_{version}: The version coefficient is assumed to be normally distributed with a mean of 2 and a standard deviation of 1. - β_{portal}: The portal coefficient is assumed to be normally distributed with a mean of 3 and a standard deviation of 1. - $\beta_{\text{version\_portal}}$: The interaction coefficient is assumed to be normally distributed with a mean of 4 and a standard deviation of 1.

We start with a high base number of page visits regardless of the version of the website or portal. For different versions, we expect the number of page visits to deviate slightly, hence the lower mean for the distributions.

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

library(rethinking)
synthetic_data$portalF <- as.factor(synthetic_data$portal)
synthetic_data$versionF <- as.factor(synthetic_data$version)

m_synthetic0 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- intercept,
    intercept ~ dnorm(10, 10)
  ),
  data = synthetic_data, iter = 10000, chains = 4, cores = 4
)

m_synthetic1 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- intercept + beta_version * version,
    intercept ~ dnorm(10, 10), # Prior for intercept
    beta_version ~ dnorm(2, 10) # Prior for version coefficient
  ),
  data = synthetic_data, iter = 10000, chains = 4, cores = 4
)

m_synthetic2 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- intercept + beta_portal * portal + beta_interaction * version * po
rtal,
    intercept ~ dnorm(10, 10), # Prior for intercept
    beta_portal ~ dnorm(3, 10), # Prior for portal coefficient
    beta_interaction ~ dnorm(4, 10) # Prior for interaction coefficient
  ),
  data = synthetic_data, iter = 10000, chains = 4, cores = 4
)


m_synthetic3 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- intercept + beta_version * version + beta_portal * portal + beta_i
nteraction * version * portal,
    intercept ~ dnorm(10, 10), # Prior for intercept
    beta_version ~ dnorm(2, 10), # Prior for version coefficient
    beta_portal ~ dnorm(3, 10), # Prior for portal coefficient
    beta_interaction ~ dnorm(4, 10) # Prior for interaction coefficient
    ),
  data = synthetic_data, iter = 10000, chains = 4, cores = 4
)
```

```
compare(m_synthetic0, m_synthetic1, m_synthetic2, m_synthetic3)
```

```
##                   WAIC        SE     dWAIC      dSE     pWAIC       weight
## m_synthetic3 453.2114  2.435803  0.000000       NA 0.4067440 8.451836e-01
## m_synthetic2 456.6061  2.319292  3.394628 1.425868 0.3508545 1.548164e-01
## m_synthetic1 495.8551  5.413787 42.643711 4.426016 1.0850861 4.645012e-10
## m_synthetic0 519.7711 10.181613 66.559621 9.117376 0.8196019 2.976548e-15
```

```
# Get the posterior summary for the model parameters
precis(m_synthetic3, depth = 1, prob = 0.95)
```

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

```
##                       mean         sd        2.5%     97.5%    n_eff    Rhat4
## intercept        2.3064125 0.06389783  2.18045925 2.4306018 4925.527 1.000335
## beta_version     0.1797505 0.08598933  0.01238038 0.3498189 4779.201 1.000677
## beta_portal      0.2454224 0.08335195  0.08100647 0.4103534 4991.001 1.000223
## beta_interaction 0.1994451 0.11299583 -0.02412582 0.4193657 4944.658 1.000467
```

```
# Get the posterior summary for the model parameters
posterior_summary <- precis(m_synthetic3, depth = 1, prob = 0.95)
```

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

```
# Extract the estimated coefficients
coefficients <- posterior_summary[, "mean"]
# Exponentiate the coefficients
exp_coefficients <- exp(coefficients)
# Print the exponentiated coefficients
print(exp_coefficients)
```

```
## [1] 10.038348  1.196919  1.278161  1.220725
```

From the results we can see that the model with the best fit, which is the one with the lowest WAIC score, is m_synthetic3 with a WAIC of 453.2114. The coefficients of this model indicate the log change in the expected count for a one-unit change in the predictor, holding all other predictors constant. However, in practice, it is often more intuitive to interpret the exponentiated coefficients, which indicate the multiplicative change in the expected count for a one-unit change in the predictor. We can see that the exponentiated coefficients are very close to the coefficients used to generate the synthetic data. There is a slight difference in the intercept, but this could be because of the randomness in the data generation process. The exponentiated coefficients suggest that, on average, using the new version of the website leads to an approximately 20% increase in page visits, visiting the website as a company (vs. consumer) leads to an approximately 28% increase in page visits, and the combination of the new website version and company portal leads to an additional 22% increase in page visits. These estimates are all with 95% credibility, implying a high degree of confidence in these effects.

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).

```
# include your code and output in the document
library(rethinking)
m_actual0 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- intercept,
    intercept ~ dnorm(10, 10)
  ),
  data = websitevisits, iter = 10000, chains = 4, cores = 4
)
m_actual1 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- intercept + beta_version * version,
    intercept ~ dnorm(10, 10), # Prior for intercept
    beta_version ~ dnorm(2, 10) # Prior for version coefficient
  ),
  data = websitevisits, iter = 10000, chains = 4, cores = 4
)
m_actual2 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- intercept + beta_portal * portal + beta_interaction * version * po
rtal,
    intercept ~ dnorm(10, 10), # Prior for intercept
    beta_portal ~ dnorm(3, 10), # Prior for portal coefficient
    beta_interaction ~ dnorm(4, 10) # Prior for interaction coefficient
  ),
  data = websitevisits, iter = 10000, chains = 4, cores = 4
)
m_actual3 <- map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- intercept + beta_version * version + beta_portal * portal + beta_i
nteraction * version * portal,
    intercept ~ dnorm(10, 10), # Prior for intercept
    beta_version ~ dnorm(2, 10), # Prior for version coefficient
    beta_portal ~ dnorm(3, 10), # Prior for portal coefficient
    beta_interaction ~ dnorm(4, 10) # Prior for interaction coefficient
  ),
  data = websitevisits, iter = 10000, chains = 4, cores = 4
)
```

```
compare(m_actual0, m_actual1, m_actual2, m_actual3)
```

```
##                WAIC        SE     dWAIC       dSE     pWAIC        weight
## m_actual3  6057.359  45.58333    0.0000        NA  3.839453  1.000000e+00
## m_actual2  6514.281  57.33676  456.9223  43.67656  3.873033 6.033814e-100
## m_actual1 14507.912 284.63345 8450.5534 287.86709 17.594218  0.000000e+00
## m_actual0 15049.814 286.93500 8992.4553 283.07025  9.824250  0.000000e+00
```

```
precis(m_actual3, depth = 1, prob = 0.95)
```

```
## 1000 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
##                       mean         sd       2.5%      97.5%    n_eff    Rhat4
## intercept          2.9765390 0.01432516  2.9480995  3.0041005 5673.379 1.000683
## beta_version      -0.4913328 0.02346621 -0.5367762 -0.4444957 5353.576 1.000884
## beta_portal        0.5228913 0.01821531  0.4875159  0.5586488 5628.846 1.001035
## beta_interaction   1.0054813 0.02722948  0.9522749  1.0584300 5287.690 1.001205
```

```
# get the posterior summary for the model parameters
posterior_summary <- precis(m_actual3, depth = 1, prob = 0.95)
```

```
## 1000 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
# extract the estimated coefficients
coefficients <- posterior_summary[, "mean"]
# exponentiate the coefficients
exp_coefficients <- exp(coefficients)
# print the exponentiated coefficients
print(exp_coefficients)
```

```
## [1] 19.6197947  0.6118104  1.6868979  2.7332224
```

Comparing the four models, we see that m_actual3 has the lowest WAIC score of 6057.3 indicating that it is the best fit model. The coefficients of this model are also the exact same as the coefficients generated from the glm() function used earlier with the actual data on the poisson distribution. The exponentiated coefficients suggest that, on average, the new version of the website leads to about a 39% decrease in page visits (exp(-0.49) ≈ 0.61), being a company (vs. consumer) leads to a 68% increase in page visits (exp(0.52) ≈ 1.68), and the interaction between the new website version and company portal leads to about a 174% increase in page visits (exp(1.01) ≈ 2.74). These estimates are with 95% credibility, implying a high degree of confidence in these effects.

Part 3 - Multilevel model

Loading the data

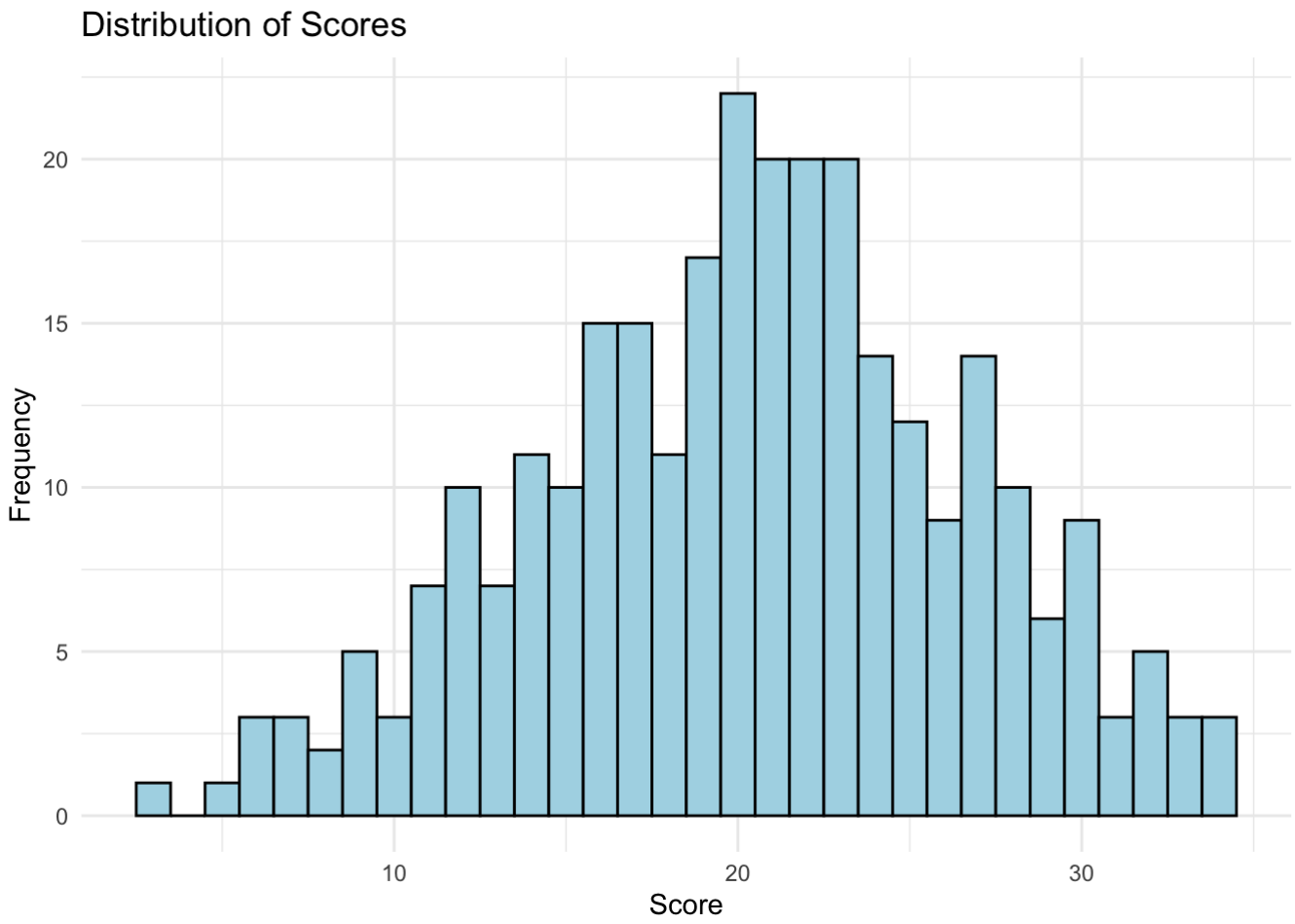
Our combined student numbers are: 5059151+5096790+4658272 = 14824213 % 3 = 0. Therefore we will use the file set0.csv

```
# load set0.csv
set0 <- read.csv("set0.csv")
```

Visual inspection

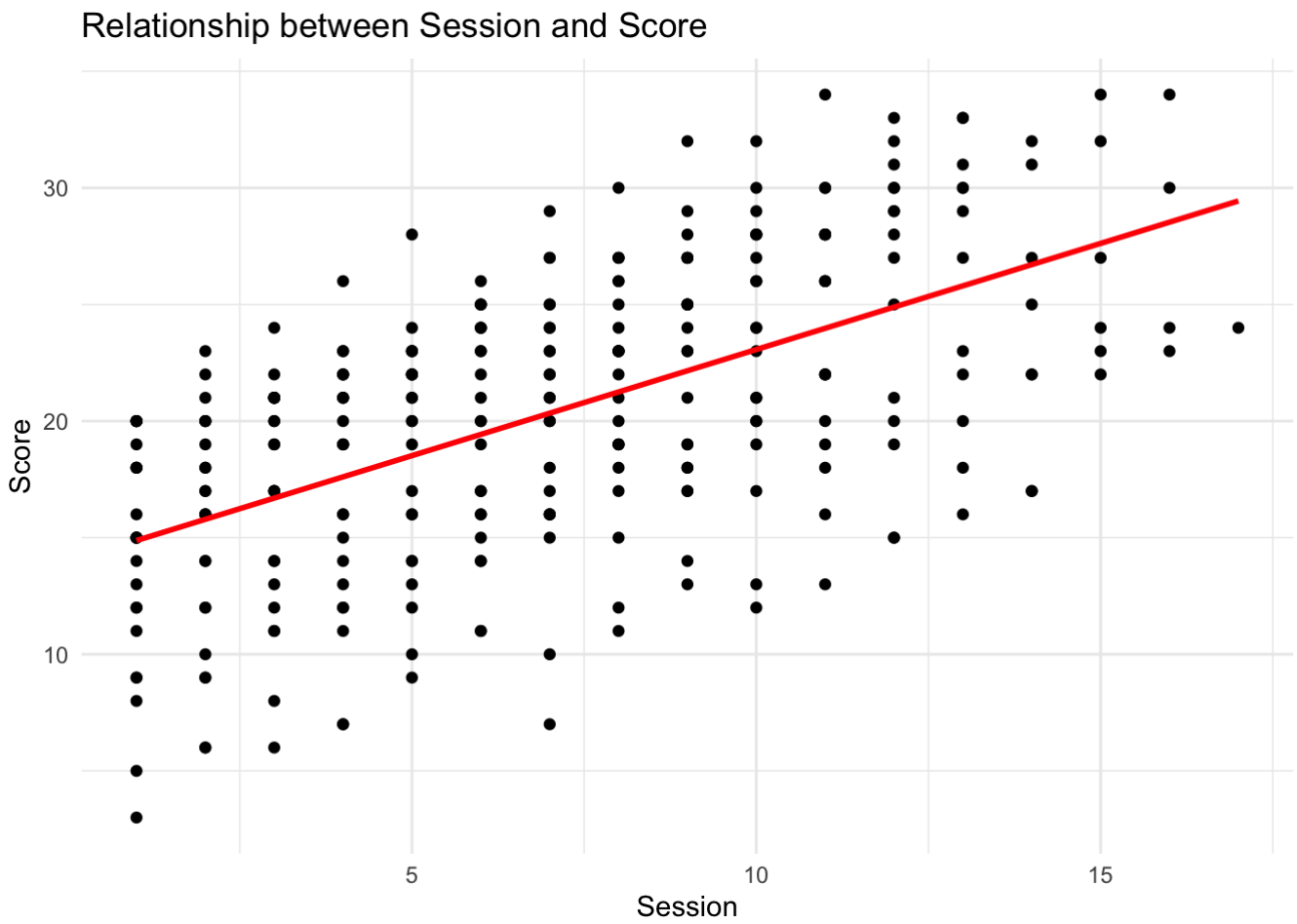
Use graphics to inspect the distribution of the score, and relationship between session and score. Give a short description of the figure.

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

library(ggplot2)
ggplot(set0, aes(x = score)) +
  geom_histogram(binwidth = 1, color = "black", fill = "lightblue") +
  theme_minimal() +
  labs(title = "Distribution of Scores",
       x = "Score",
       y = "Frequency")
```




```
ggplot(set0, aes(x = session, y = score)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  theme_minimal() +
  labs(title = "Relationship between Session and Score",
       x = "Session",
       y = "Score")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



The visual presents a bell-shaped curve, characteristic of a normal distribution. The scores range from 3 to 34, with a central tendency represented by the median value of 19. The distribution is symmetrically centered around the median, indicating a balanced dataset.

The relationship between session and scores shows a linear function between the session number and the score. The higher the session number, the higher the score.

Frequentist approach

Multilevel analysis

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

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

```
# include your code and output in the document
library(nlme)
m0 <- lme(score ~ 1, random = ~ 1 | subject, data = set0, method = "ML")
m1 <- lme(score ~ session, random = ~ 1 | subject, data = set0, method = "ML")
summary(m0)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: set0
##        AIC      BIC    logLik
##   1723.387 1734.407 -858.6936
##
## Random effects:
##  Formula: ~1 | subject
##         (Intercept) Residual
## StdDev:    4.564947 4.145961
##
## Fixed effects:  score ~ 1
##                Value Std.Error  DF  t-value p-value
## (Intercept) 20.38525 0.9851451 268 20.69264       0
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -1.99996298 -0.73308345  0.03269887  0.81058602  1.98164739
##
## Number of Observations: 291
## Number of Groups: 23
```

```
summary(m1)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: set0
##        AIC      BIC    logLik
##   978.3459 993.0392 -485.1729
##
## Random effects:
##  Formula: ~1 | subject
##         (Intercept) Residual
## StdDev:    4.836719 1.026429
##
## Fixed effects:  score ~ session
##                Value Std.Error  DF  t-value p-value
## (Intercept) 13.51858 1.0195619 267 13.25920       0
## session      1.00562 0.0157501 267 63.84847       0
##  Correlation:
##         (Intr)
## session -0.105
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -3.0917418 -0.6250027 -0.0700297  0.5312065  2.9559338
##
## Number of Observations: 291
## Number of Groups: 23
```

```
conf_int <- intervals(m1, level = 0.95)
conf_int
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                  lower     est.     upper
## (Intercept) 11.5180863 13.51858 15.519074
## session      0.9747168  1.00562  1.036524
##
##  Random Effects:
##   Level: subject
##                    lower     est.    upper
## sd((Intercept)) 3.619257 4.836719 6.463716
##
##  Within-group standard error:
##     lower      est.     upper
## 0.9431835 1.0264286 1.1170207
```

```
anova(m0, m1)
```

```
##    Model df       AIC       BIC    logLik   Test  L.Ratio p-value
## m0     1  3 1723.3872 1734.4071 -858.6936
## m1     2  4  978.3459  993.0392 -485.1729 1 vs 2 747.0413  <.0001
```

```
# calculate the intra-class correlation coefficient
v <- VarCorr(m1)
tau<-as.numeric(v["(Intercept)","StdDev"])
sigma<-as.numeric(v["Residual","StdDev"])
icc <- tau^2 / (tau^2 + sigma^2)
icc
```

```
## [1] 0.9569052
```

Report section for a scientific publication

In this study, we examined the impact of session on people's scores using a multilevel analysis. The dataset consisted of 291 observations from 23 participants. We compared two linear mixed-effects models: the first model (m0) included only the intercept, while the second model (m1) included the effect of session. The analyses were conducted assuming a Gaussian distribution for the scores. The results showed that including the session variable significantly improved the model fit ($p < 0.0001$), indicating that session has a significant impact on people's scores. The fixed effect estimate for session was 1.006 ($p < 0.0001$), suggesting that, on average, scores increased by approximately 1 unit for each session. Furthermore, we assessed the variance between participants by calculating the intra-class correlation coefficient (ICC), which measures the proportion of total variance attributed to the differences between participants. The ICC for the random intercept was estimated to be 0.957, indicating substantial variability between participants in their scores. Overall, these findings highlight the importance of considering the session variable when analyzing people's scores and suggest that there is significant variance between participants in their scores. This information can guide interventions and tailored approaches to optimize scores based on session and individual characteristics, ultimately leading to improved outcomes and user experiences.

Bayesian approach

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

Model description

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

The mathematical model that is fitted on the most extensive model is as follows:

$$
\begin{aligned}
\text{score}_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \alpha_{\text{subject[i]}} + \beta_{\text{session}} \cdot \text{session}_i \\
\alpha_{\text{subject[i]}} &\sim \text{Normal}(0, \sigma_{\text{subject}}) \\
\alpha &\sim \text{Normal}(0, 10) \\
\beta_{\text{session}} &\sim \text{Normal}(0, 10) \\
\sigma &\sim \text{Exponential}(1) \\
\sigma_{\text{subject}} &\sim \text{Normal}(0, 10)
\end{aligned}
$$

The priors are justified as follows:

- α: The intercept is assumed to be normally distributed with a mean of 0 and a standard deviation of 10. This is because we do not have any prior knowledge about the intercept and we assume that the intercept is normally distributed.
- β_{session}: The slope is assumed to be normally distributed with a mean of 0 and a standard deviation of 10. This is because we do not have any prior knowledge about the slope and we assume that the slope is normally distributed.
- β_{subject}: The slope is assumed to be normally distributed with a mean of 0 and a standard deviation of 10. This is because we do not have any prior knowledge about the slope and we assume that the slope is normally distributed.
- σ: The standard deviation is assumed to be exponentially distributed with a rate of 1. This is because we do not have any prior knowledge about the standard deviation and we assume that the standard deviation is exponentially distributed.

Model comparison

Compare models with with increasing complexity.

```
# include your code and output in the document
library(rethinking)
m0_set0 <- map2stan(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a,
    # fixed priors
    a ~ dnorm(0, 10),
    sigma ~ dexp(1)
  ),
  data = set0, iter = 10000, chains = 4, cores = 4,
  log_lik = TRUE, control = list(adapt_delta = 0.99)
)
m1_set0 <- map2stan(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a + a[subject],
    # adaptive priors
    a[subject] ~ dnorm(0, sigma_subject),
    # hyperpriors
    sigma_subject ~ dnorm(0, 10),
    # fixed priors
    a ~ dnorm(0, 10),
    sigma ~ dexp(1)
  ),
  data = set0, iter = 10000, chains = 4, cores = 4,
  log_lik = TRUE, control = list(adapt_delta = 0.99)
)
m2_set0 <- map2stan(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a + a[subject] + b_session * session ,
    # adaptive priors
    a[subject] ~ dnorm(0, sigma_subject),
    # hyperpriors
    sigma_subject ~ dnorm(0, 10),
    # fixed priors
    a ~ dnorm(0, 10),
    b_session ~ dnorm(0, 10),
    sigma ~ dexp(1)
  ),
  data = set0, iter = 10000, chains = 4, cores = 4,
  log_lik = TRUE, control = list(adapt_delta = 0.99)
)
```

```
compare(m0_set0, m1_set0, m2_set0)
```

```
##              WAIC       SE     dWAIC dSE     pWAIC        weight
## m2_set0  869.4752 26.01391    0.0000  NA 24.195178  1.000000e+00
## m1_set0 1677.5977 17.36812  808.1225  NA 21.476558 3.299320e-176
## m0_set0 1897.2454 22.50451 1027.7702  NA  1.814029 6.645598e-224
```

Based on the Watanabe-Akaike Information Criterion (WAIC), which is used to compare model fit, model m2_set0, with a WAIC score of 869.2 and a weight of 1, provides the best fit to the data. This model incorporates both subject-specific variability and the fixed effect of sessions. Conversely, the models m1_set0 and m0_set0, excluding the session effect and both subject and session effects respectively, perform significantly worse. These findings show the importance of subject and session when interpreting the scores data.

Estimates examination

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

```
# include your code and output in the document
precis(m2_set0, depth = 1, prob = 0.95)
```

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

```
##                   mean         sd      2.5%     97.5%     n_eff     Rhat4
## sigma_subject 7.475807 1.15235087 5.6083055 10.087812 26952.208 1.0000121
## b_session     1.010100 0.01589652 0.9788608  1.041100  6671.303 0.9999550
## sigma         1.031513 0.04479896 0.9477387  1.122713 27360.396 0.9998566
```

These results suggest that both the subject and session have a significant impact on the score. There is considerable variation in score across different subjects, and there is also a significant increase in score with each increase in session. The model fits the data reasonably well, but there is still some unexplained variation in score, as indicated by the value of sigma.