Report coursework assignment A - 2021

CS4125 Seminar Research Methodology for Data Science

Nikki Bouman (4597648), Anuj Singh (5305926), Gwennan Smitskamp (4349822)

18/06/2021

Contents

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

1.1 The motivation for the planned research

The recent outbreak of the COVID-19 pandemic has changed our daily lives significantly. People are obligated to stay at home, disrupting their usual social interactions in both work and private life. The situation compels people to meet online. Typically, in such digital interactions, interlocutors can see each other by means of webcam streaming. However, this may not always be the case. Some or all interlocutors may not be visible during online dialogue, which could affect the quality of the conversation and the mutual understanding.

An important effect of the shift from face-to-face to online interaction can be revealed by studying laughter, as it is extremely contagious social behavior (Provine, 1992). Humans are very prone to unintentionally or unconsciously laugh as a social signal in any form; from a minor smile to laughing out loud. Additionally, laughing is one of the most important social signals for lubricating the flow of social interaction (Griffin et al., 2015).

1.2 The theory underlying the research

The effect of visibility on the use of gestures as a communicative function has been studied broadly (Alibali, Heath, & Myers, 2001; J. B. Bavelas, Chovil, Lawrie, & Wade, 1992; Cohen & Harrison, 1973; Cohen, 1977; Emmorey & Casey, 2001; Krauss, Dushay, Chen, & Rauscher, 1995; Rim´e, 1982). , J. Bavelas, Gerwing, Sutton, and Prevost (2008) provide a summary of previous experiments where rate and form of gestures were compared under two conditions: where the addressee could see the speaker and where the addressee could not see the speaker. These experiments show that speakers gestured at higher rate when they communicated with mutual visibility than without. J. Bavelas et al. (2008) extended these experiments by focusing on both visibility and dialogue as a variable, finding similar results. Furthermore, they found that speakers gestured at a significantly higher rate in a telephone dialogue than in a monologue to a tape recorder, confirming that visibility is not the only variable operating in telephone conversations. These experiments showed us that visibility plays a major role in the rate of gesturing, but that people also gesture when they are not visible to each other. As laughter can be seen as a form of gesturing, these findings are relevant for this study.

Laughing together is found to be essentially collaborative (Mehu & Dunbar, 2008; Coates, 2007). Joint laughter therefore serves important means to achieve effective team meetings (Ponton, Osbourne, Greenwood, & Thompson, 2018), considering that people who laugh on video are perceived with a higher likeability than people who do not (Reysen, 2006). This social function of joint laughter emphasises the relevance of studying the occurrence, now that the majority of meetings take place online.

1.3 Research questions

We will aim to answer the following research question:

What is the effect of webcam visibility during online dialogue on the frequency of joint laughter?

When recognizing laughter we do not focus on the reason why someone is laughing. We consider anything from an awkward laugh in a moment of silence to laughing out loud about a joke as a laughter episode regardless of the context.

1.4 The related conceptual model

The conceptual model related to the research question can be viewed in Figure 1. The main question is about the effect of mutual visibility on joint laughter. The mediating variable is chemistry, which we can define as the level of friendliness or intimacy between people who did not know each other before the experiment. This can occur on a high level when people find similarities in their interests and behaviours, or find that they are able to easily talk with another. Familiarity happens when people knew each other already before the experiment. Another moderating variable is the duration of the experiment. The participants will do the experiment for about one hour, which can have a negative effect on the frequency of laughter, thus on the frequency of joint laughter.

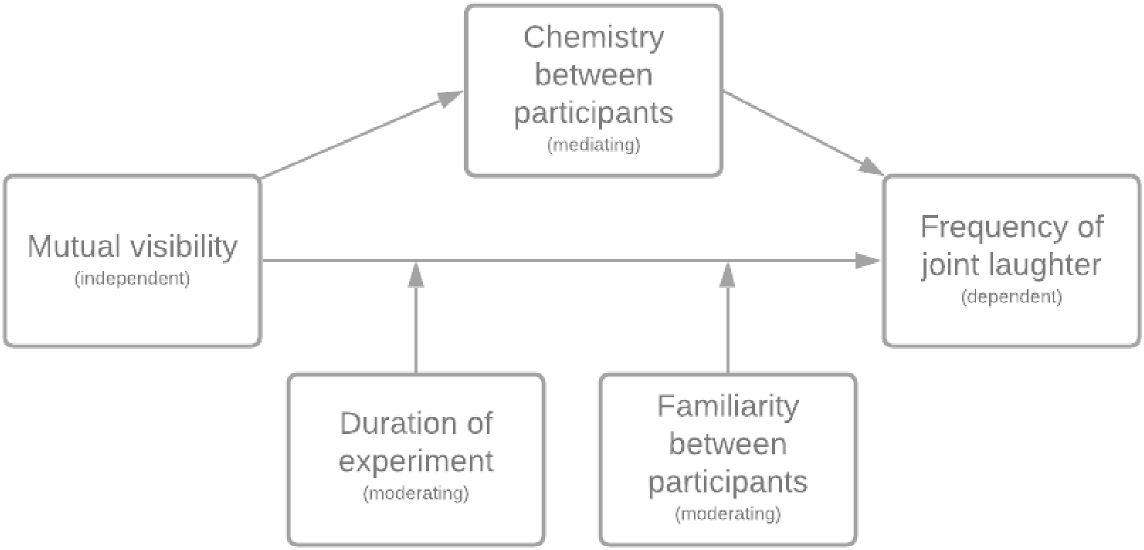


Figure 1: The conceptual model to test the effect of mutual webcam visibility on joint laughter

1.5 Experimental Design

One of the most important requirements for the setup of the experiment was the creation of a comfortable and pleasant ambiance so that people would laugh. Therefore, it was decided that a game would comply, as the participants get the chance to interact with each other in a undemanding setting where the attention of the participants would be drawn to a task. It was reasoned that this would contribute to a reduction of awkwardness and give all the participants the option to speak and laugh. Additionally, the game needed to have a smooth flow that would automatically keep going to keep the interference of the researchers to a minimum.

The game that was chosen is called 30 Seconds. During the game, participants work together in teams (in the case of the experiment: two teams of two people) and gain points by guessing what the team member is describing. These descriptions include concepts such as famous persons, locations, movies and brands. Every team gets 30 seconds to guess as many concepts on the card as possible. Who is describing and who is guessing switches after every card. The participants will be playing on their laptops, seeing the other three participants and a chat screen on which they will receive the words for the game. This within-subject experiment will be held where half of the experiments will start with the webcams on, while the other half will start with the webcams off.

1.6 Experimental procedure

The experiment will be set up in an online setting in a Zoom meeting. The host, one of us (not visible), will be able to send private messages containing the five words, share their screen and sound for a 30 second timer, and turn the participants' webcams on and off.

The following will repeat for every card of five words:

1. The host sends the words to the participant who has the turn to describe.
2. The host starts the 30 second timer.
3. The participant will try to describe as many words as possible, while his/her teammate will try to guess the words.

4. The timer rings, the host puts the score in the chat.

During the experiment, multiple things will happen. Each time all players have guessed and described a card (i.e. after four cards total), their webcams will switch on or off. After each player has guessed and described four cards (i.e. after sixteen cards total), the final score will be displayed and the teams will be rearranged. The previous will then repeat until every participant has been in a team with every other participant. This makes sure every participant has an interaction with all the other participants, so the differences in laughter between people cannot be blamed on the interacting pairs. An example of such an experiment is displayed in Figure 2.

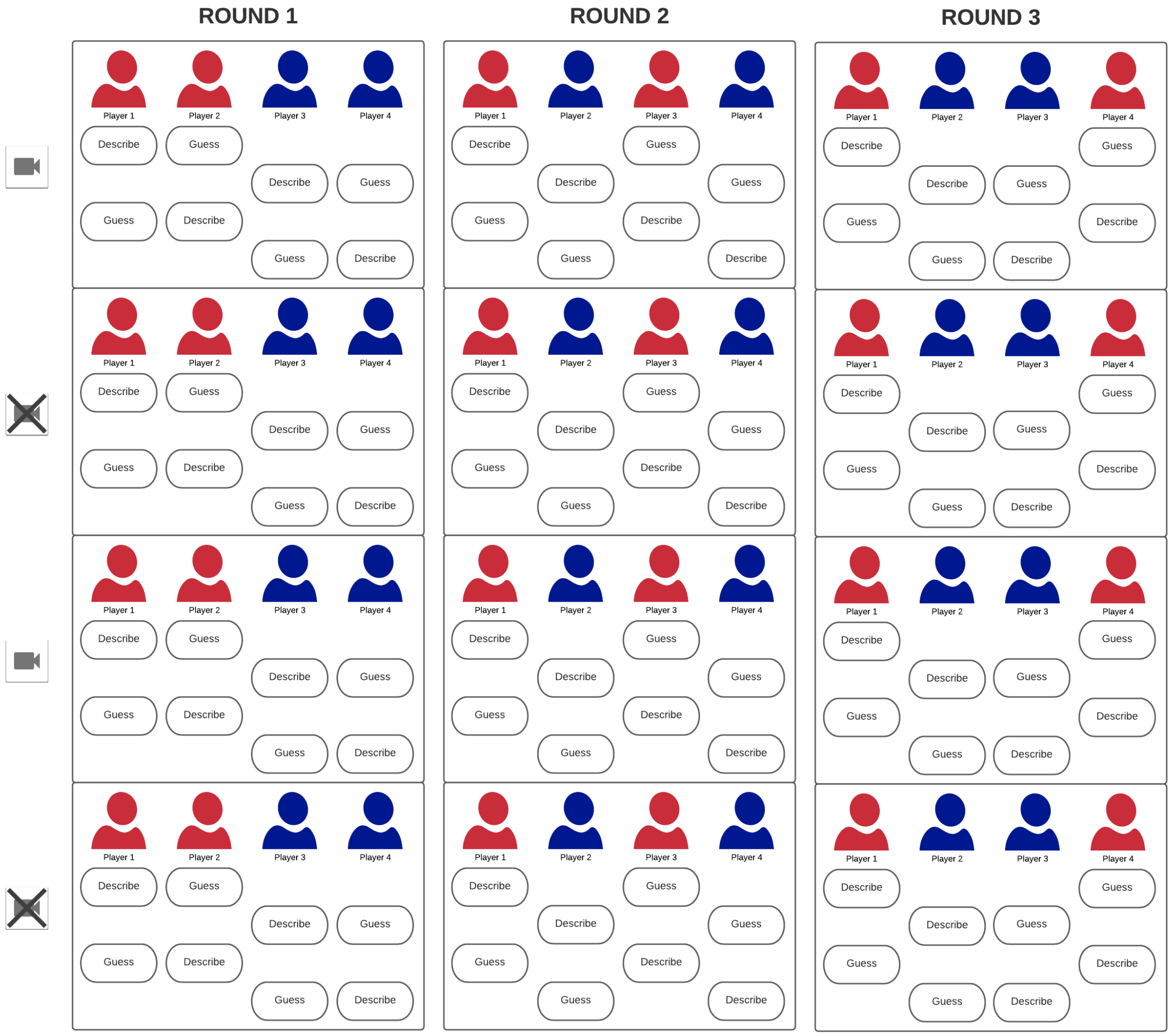


Figure 2: An example of an experimental setup.

1.7 Measures

A pre-test will be held to define the level of familiarity. The participants will receive a neutral picture (e.g. from a driver's license) of the other three participants, after which they have to indicate whether they know the person. The remaining test includes a question like "Do you feel related to this person?", after which the participant can indicate why (e.g. because of their skin colour, gender, hair colour, etc.). The same questionnaire will be held after the experiment to check the level of chemistry.

The data that has been collected during the experiment includes audio and video of participants. The first step in data analysis involves annotating the signals. This will be done with a program in which we can manually select timesteps in which the participant is laughing. In the end we will have annotations for every person that contains the total amount of laughter (frequency), the amount of laughter with their webcam off, and the amount of laughter with their webcam on.. In a case where different people will annotate this data, we will first let every one of them annotate the same data sample. Then we can calculate the consistency in the annotations with for example Krippendorf's Alpha or Cohen's Kappa. When this is high enough, they can start annotating separate data.

1.8 Participants

The research is not specifically about a certain group of people, but more in general. However, we do want to aim for people with experience in an online setting and people who speak the same language. To be exact, we will conduct the experiment with people from the ages of 18 to 50 who speak dutch. The number of participants depends on the acceptable margin or error, which we do not know since we would have to go much more in-depth. However, since the population size basically includes more than 10.000 people and the independent variable is categorical, we should aim for around 385 participants (W.P. Brinkman, 2009). This results into around 96 experiments.

The experiments should be easy to conduct since the participants participate online; there is no need to travel. Moreover, the whole experiment should not take long. There will be 48 cards and for each card they have 30 seconds to explain. Taking some talking afterwards into account, the experiment should take 45 to 60 minutes, with taking 10 minutes before and after into account for the pre- and post-test.

To find the participants we can thus search online. Using a medium on which we state the experiment, we can find dutch people from all around the Netherlands of different age groups. They could for example sign up for a specific date, and if four people have signed up, the experiment can be held.

1.9 Suggested statistical analyses

To determine the significance of the results we will subject the data to statistical tests. To test the effect of mutual visibility (categorical, since it is either on or off) on the frequency (numerical) we will use the Wilcoxon signed rank test. The Wilcoxon signed rank test is a suitable statistical test when the measurements for a single variable are taken under two different conditions. It is similar to the paired t-test, however the paired t-test assumes that the data is normally distributed which we cannot assume for frequency of laughter. It tests the null hypothesis that the median difference of a two sets of observations is zero. The frequentist's approach is chosen since the experiment is repeated with the fixed parameters. Moreover, with this approach we can better control the Type I error.

2 Part 2 - Generalized linear models

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

2.1.1 Conceptual model

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

2.1.2 Collecting tweets, and data preparation

We found five celebrities in US politics: Donald Trump, Tucker Carlson, Bernie Sanders, Alexandria Ocasio-Cortez, Joe Biden. As dutch students we are not well-versed in the popular English twitter celebrities, so US politics was the best option for us to find celebrities that had enough recent Tweets for the Twitter API.

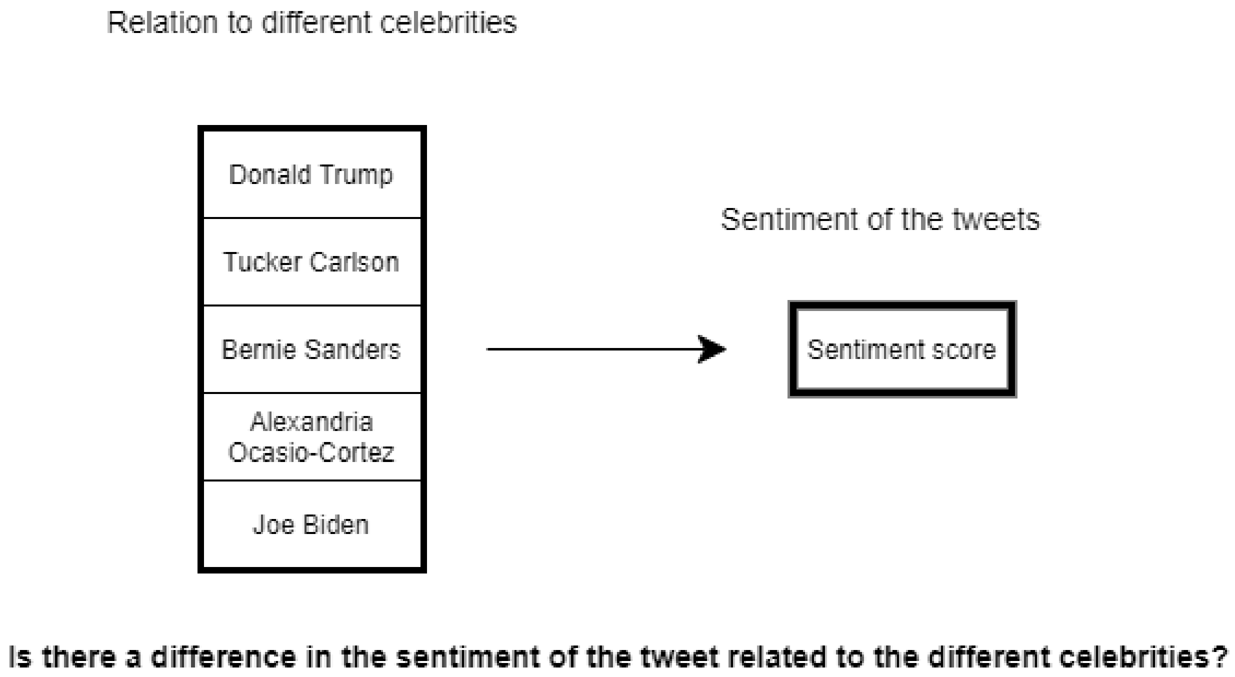

Relation to different celebrities
Donald Trump
Tucker Carlson
Bernie Sanders
Alexandria Ocasio-Cortez
Joe Biden
Sentiment of the tweets
Sentiment score
Is there a difference in the sentiment of the tweet related to the different celebrities?

Figure 3: The conceptual model for the research question: Is there a difference in the sentiment of the tweets related to the different celebrities?

2.1.3 Homogeneity of variance analysis

```
pander(leveneTest(semFrame$score, semFrame$Celeb))
```

Table 1: Levene's Test for Homogeneity of Variance (center = median)

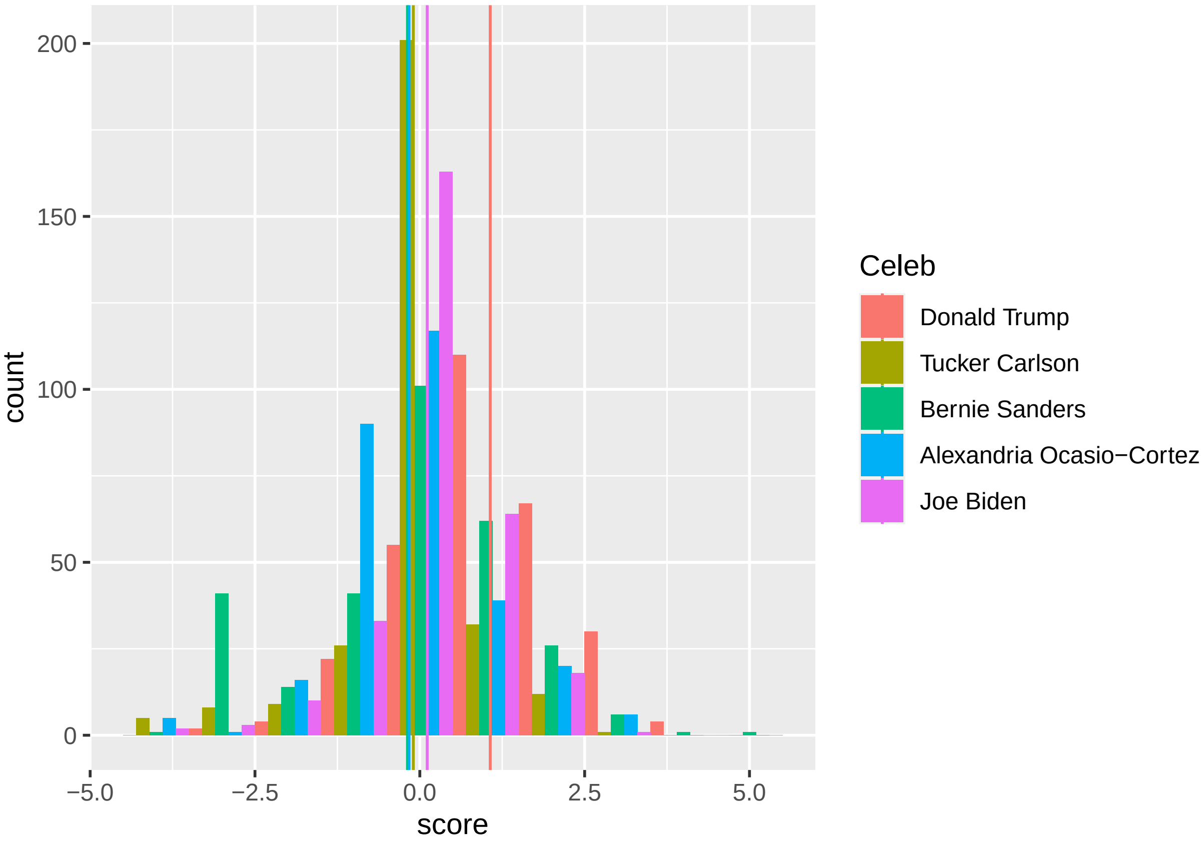
	Df	F value	Pr(>F)
group	4	23.64	5.714e-19
	1465	NA	NA

The Levene test reveals a p-value smaller than 0.001, indicating that there is significant difference between the group variances in sentiment score. We conclude that the variance among the five groups is not equal.

2.1.4 Visual inspection Mean and distribution sentiments

We plot both a line density and a distribution histogram which includes a mean line.

```
cdat <- ddply(semFrame, "Celeb", summarise, score.mean=mean(score))
ggplot(semFrame, aes(x=score, fill=Celeb)) +
  geom_histogram(binwidth=1, position="dodge") +
  geom_vline(data=cdat, aes(xintercept=score.mean,  colour=Celeb),
             linetype="solid", size=0.5)
```



```
cdat_grouped <- group_by(semFrame, Celeb)
summarize(cdat_grouped,
mean_score = mean(score),
sd_score = sd(score))
```

```
## # A tibble: 5 x 3
##   Celeb                           mean_score sd_score
##   <fct>                                <dbl>    <dbl>
## 1 "Donald Trump"                      1.07      1.22
## 2 "Tucker Carlson"                   -0.0986    1.01
## 3 "Bernie Sanders"                   -0.187     1.58
## 4 "Alexandria Ocasio-Cortez "        -0.163     1.18
## 5 "Joe Biden"                         0.112     0.966
```

We see all US politic Celebs have a mean around 0. #trump has the highest sentiment mean and the largest difference with the rest.

### Frequentist approach

2.1.4.1 Linear model

A one-way between subjects ANOVA was conducted to compare the effect of relation to celebrities on sentiment score in five conditions.

```r
modelTwitter0 <- lm(formula = score ~ 1 , data = semFrame)
modelTwitter1 <- lm(formula = score ~ Celeb , data = semFrame)
pander(anova(modelTwitter0, modelTwitter1, test = "F"))
```

Table 2: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1469	2476	NA	NA	NA	NA
1465	2147	4	328.6	56.06	4.859e-44

There was a significant effect of relation to celebrities on sentiment score at the p<.001 level for the five conditions [F(4, 1465) = 56.06, p < 0.001].

```r
#AIC
modelsTwitter <- list(modelTwitter0, modelTwitter1)
modelTwitter.names <- c("modelTwitter0", "modelTwitter1")
pander(aictab(cand.set = modelsTwitter, modnames=modelTwitter.names),
       caption="Model selection based on AICc.")
```

Table 3: Model selection based on AICc. (continued below)

	Modnames	K	AICc	Delta_AICc	ModelLik	AICcWt	LL
2	modelTwitter1	6	4741	0	1	1	-2364
1	modelTwitter0	2	4942	201.3	1.939e-44	1.939e-44	-2469

	Cum.Wt
2	1
1	1

A lower AIC indicates a better fit, which is the model with the predictor.

```r
pander(pairwise.t.test(semFrame$score, semFrame$Celeb, paired = FALSE, p.adjust.method = "bonferroni"))
```

2.1.4.2 Post Hoc analysis

```
## Warning in pander.default(pairwise.t.test(semFrame$score, semFrame$Celeb, : No
## pander.method for "pairwise.htest", reverting to default.
```

- **method**: t tests with pooled SD
- **data.name**: semFrame*scoreandsemFrame*Celeb
- **p.value**:

Table 5: Table continues below

	Donald Trump	Tucker Carlson	Bernie Sanders	Alexandria Ocasio-Cortez
Tucker Carlson	3.197e-29	NA	NA	NA
Bernie Sanders	1.727e-33	1	NA	NA
Alexandria Ocasio-Cortez	2.575e-32	1	1	NA
Joe Biden	4.257e-20	0.3485	0.02765	0.05864

- **p.adjust.method**: bonferroni

Post hoc comparisons using the Bonferroni correction indicated that the corrected p-value for the trump condition was significantly different than the other conditions (p<0.001). However, between the others condition it does not show a significantly difference.

2.1.4.3 Report section for a scientific publication

A one-way between subjects ANOVA was conducted to compare the effect of relation to celebrities on sentiment score in five conditions. There was a significant effect of relation to celebrities on sentiment score at the p<.001 level for the five conditions [F(4, 1465) = 56.06, p < 0.001]. However, post hoc comparisons using the Bonferroni correction indicated that only the corrected p-value for the trump condition (M = 1.07, SD = 1.22) was significantly different than the other conditions (p<0.001), between the others condition it does not show a significantly difference. Taken together, these results suggest that some celebrities really do have an effect on the sentiment in Tweets.

2.1.5 Bayesian Approach

2.1.5.1 Model description

The sentiment scores seem to center around 0, and all seem to be single digits.

$$score \sim Norm(\mu, \sigma)$$

$$\mu = \alpha_{Celeb}$$

$$\alpha_{Celeb} \sim Norm(0, 10)$$

$$\sigma \sim Uniform(0.001, 10)$$

```
mTwitter0 <-map2stan(alist(
  score ~ dnorm(mu, sigma),
  mu <-a,
  a ~ dnorm(0, 10),
  sigma ~ dunif(0.001, 10)),
  data =  semFrame ,  iter= 10000, chains = 4,   cores = 4 )
```

2.1.5.2 Model comparison

```
## Computing WAIC
```

```
mTwitter1 <-map2stan(alist(
  score ~ dnorm(mu, sigma),
  mu <-a[Celeb] ,
  a[Celeb] ~ dnorm(0, 10),
  sigma ~ dunif(0.001, 10)),
  data =  semFrame ,iter= 10000, chains = 4, cores = 4 )
```

```
## Computing WAIC
```

```
pander(compare(mTwitter0, mTwitter1, func=WAIC))
```

	WAIC	SE	dWAIC	dSE	pWAIC	weight
mTwitter1	4741	70.19	0	NA	6.656	1
mTwitter0	4942	67.74	201.3	28.94	2.572	1.988e-44

Lower WAIC indicates a better performing model, so with predictors (mTwitter1) is the better approach to model what is happening.

```

pander(precis(mTwitter1, depth=2, prob = .95))
```

2.1.5.3 Comparison celebrity pair

	mean	sd	2.5%	97.5%	n_eff	Rhat4
a[1]	1.069	0.07011	0.933	1.205	29601	1
a[2]	-0.09794	0.07027	-0.2337	0.03982	31418	1
a[3]	-0.1873	0.07084	-0.3275	-0.04712	30524	0.9999
a[4]	-0.1625	0.07028	-0.3	-0.0261	29746	0.9998
a[5]	0.1121	0.07066	-0.02603	0.2508	29969	0.9999
sigma	1.212	0.02256	1.168	1.257	30924	1

```
pander(select(pairwise_comparisons(data = semFrame, x = Celeb, y = score, type = "bayes",
                                    paired = FALSE, p.adjust.method = "bonferroni" ), group1,
              group2, estimate, conf.level, pd, bf10))
```

Table 9: Table continues below

group1	group2	estimate	conf.level
Alexandria Ocasio-Cortez	Joe Biden	0.2714	0.89
Bernie Sanders	Alexandria Ocasio-Cortez	0.0241	0.89
Bernie Sanders	Joe Biden	0.2959	0.89
Donald Trump	Alexandria Ocasio-Cortez	-1.22	0.89
Donald Trump	Bernie Sanders	-1.242	0.89
Donald Trump	Joe Biden	-0.9467	0.89
Donald Trump	Tucker Carlson	-1.16	0.89
Tucker Carlson	Alexandria Ocasio-Cortez	-0.06552	0.89
Tucker Carlson	Bernie Sanders	-0.08669	0.89
Tucker Carlson	Joe Biden	0.206	0.89

pd	bf10
0.9982	9.613
0.5867	0.0938
0.9962	3.804
1	2.81e+28
1	4.596e+21
1	5.869e+20
1	1.746e+29
0.7618	0.1177
0.7945	0.1263
0.9938	2.357

Looking at the credibility intervals of the celebrities effects, We see the conditions where the mean of a condition does not fall within a credibility interval of an other condition. This holds for the a[1] (Trump) condition and a couple other combinations. We can again conclude that some celebrities really do have an effect on the sentiment in Tweets.

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

2.2.1 Conceptual model

2.2.2 Visual inspection

```r
dataWeb <- read.csv("webvisit0.csv")


# changing dtype of the factors
dataWeb$portal = as.factor(dataWeb$portal)
dataWeb$version = as.factor(dataWeb$version)

# Function to calculate the mean and the standard deviation for each factor group


data_summary <- function(data, varname, groupnames){
  summary_func <- function(x, col){
    c(mean = mean(x[[col]], na.rm=TRUE),
      sd = sd(x[[col]], na.rm=TRUE))
  }
  data_sum<-ddply(data, groupnames, .fun=summary_func,
                  varname)
  data_sum <- rename(data_sum, c("mean" = varname))
 return(data_sum)
}
#
# df3 <- data_summary(dataWeb, varname="pages",
#                      groupnames=c("version", "portal"))

# p <- ggplot(df3, aes(x=version, y=pages, fill=portal)) +
#     geom_bar(stat="identity", position=position_dodge()) +
#    geom_errorbar(aes(ymin=pages-sd, ymax=pages+sd), width=.2,
#                  position=position_dodge(.9))
#
# p + scale_fill_brewer(palette="Paired") + theme_minimal()
```

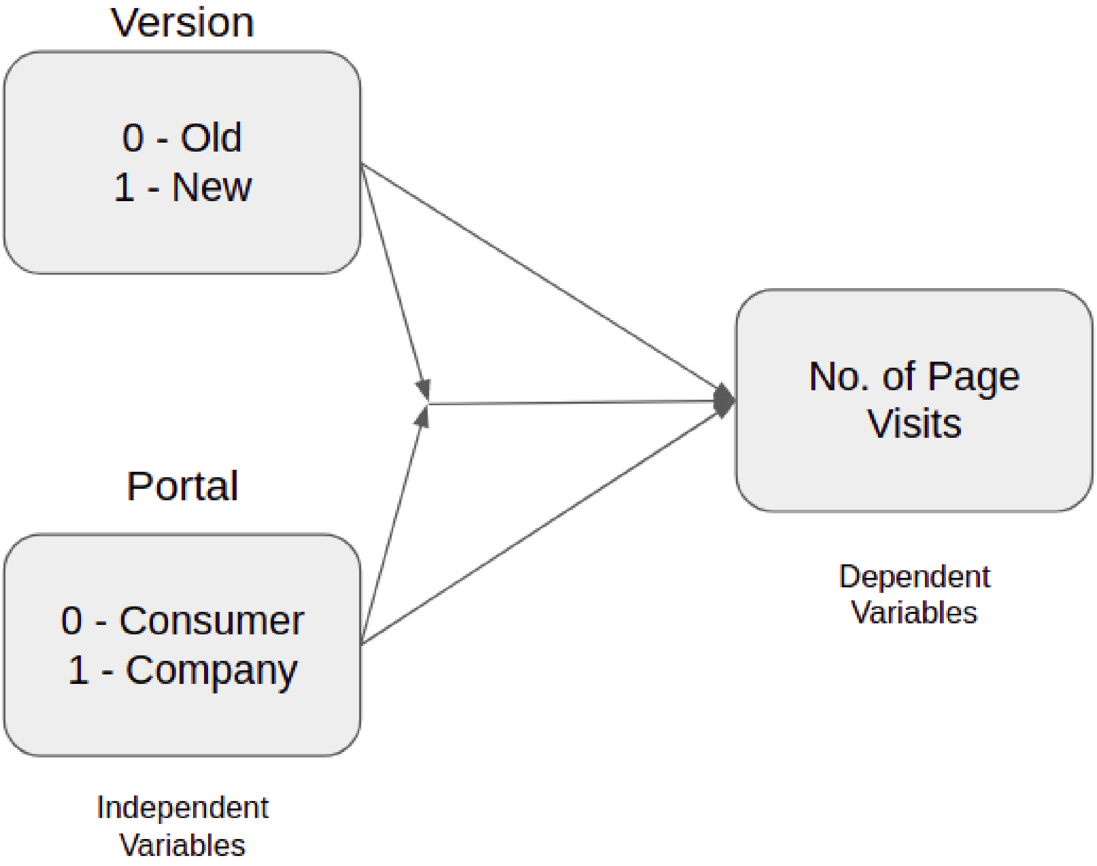
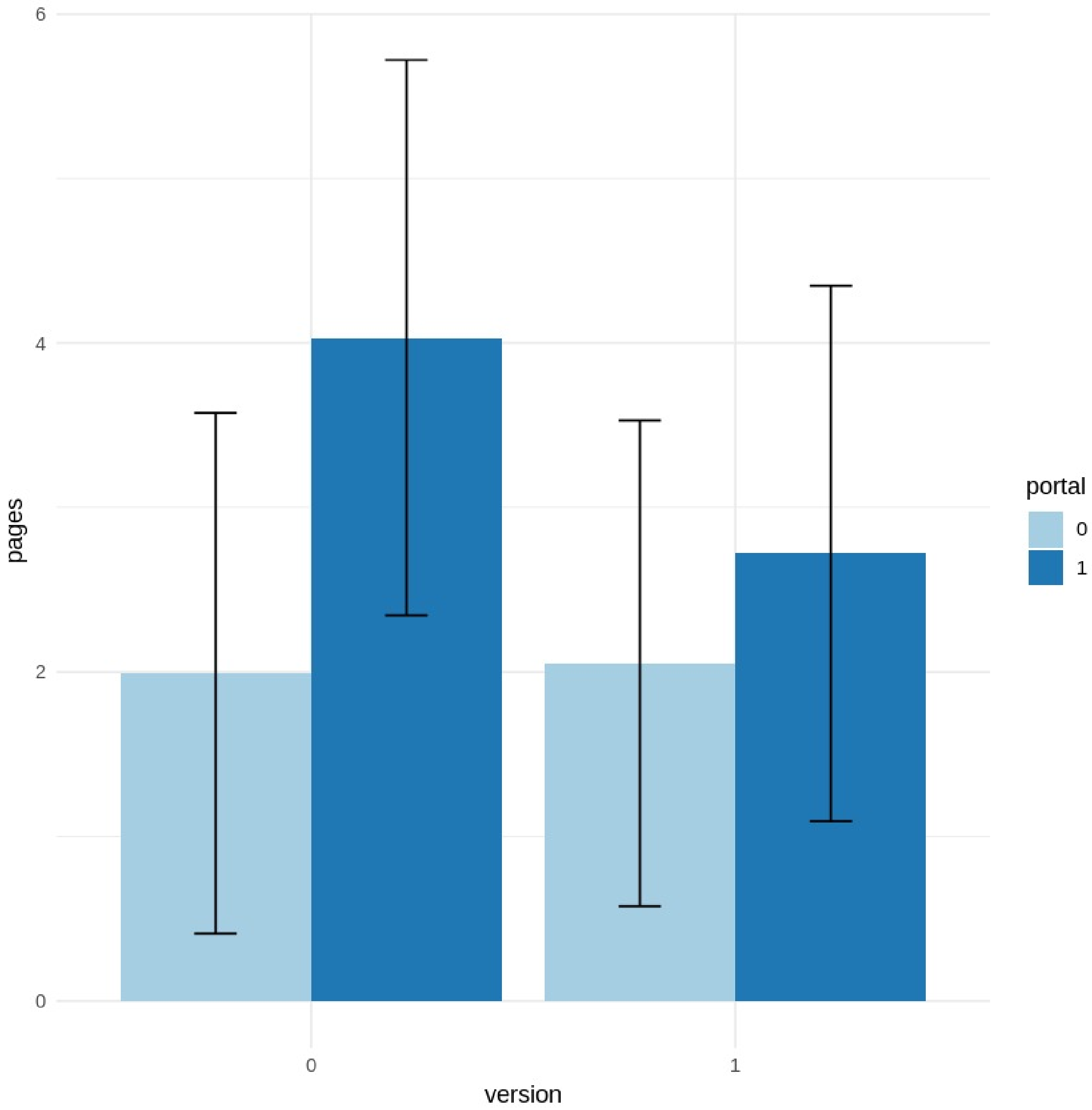


Figure 4: The conceptual model underlying the research question.

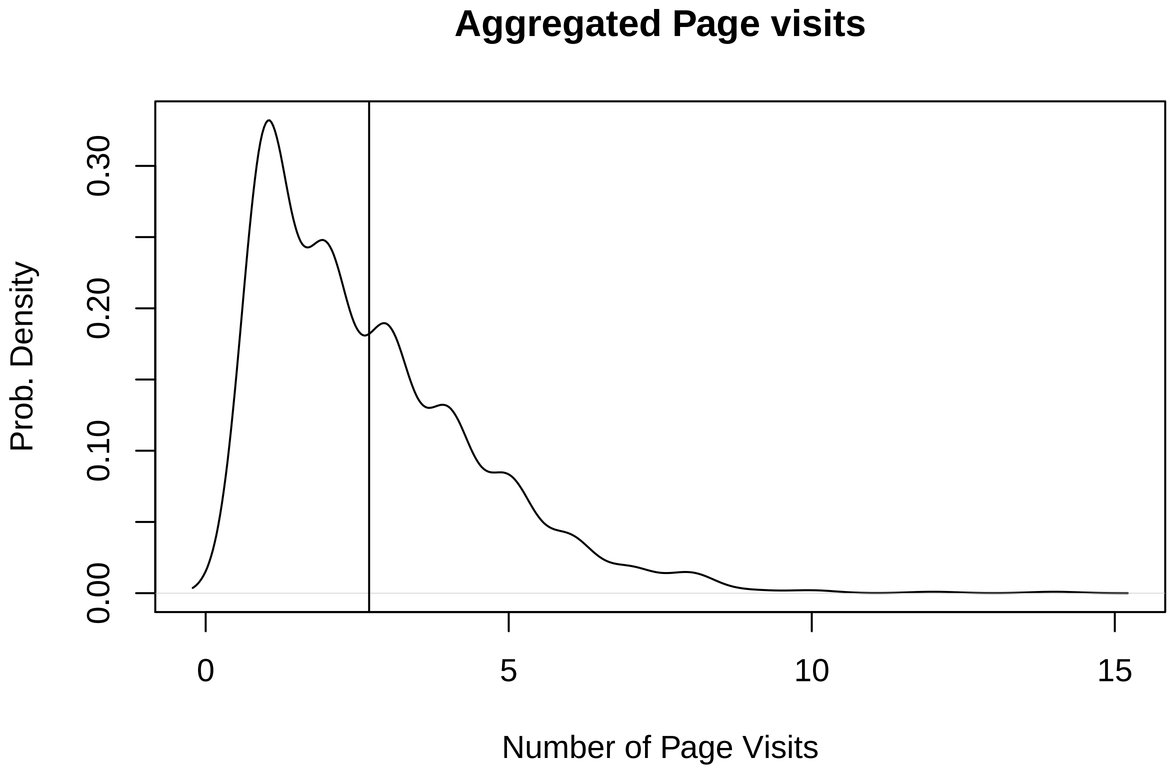


Notable observations from the bar-plot demonstrating the mean and standard deviation of the page visits are that the mean page visits across both the versions for the 0 - portal entries (Consumer) are almost the same but vary significantly for the 1 - portal entries (Company).

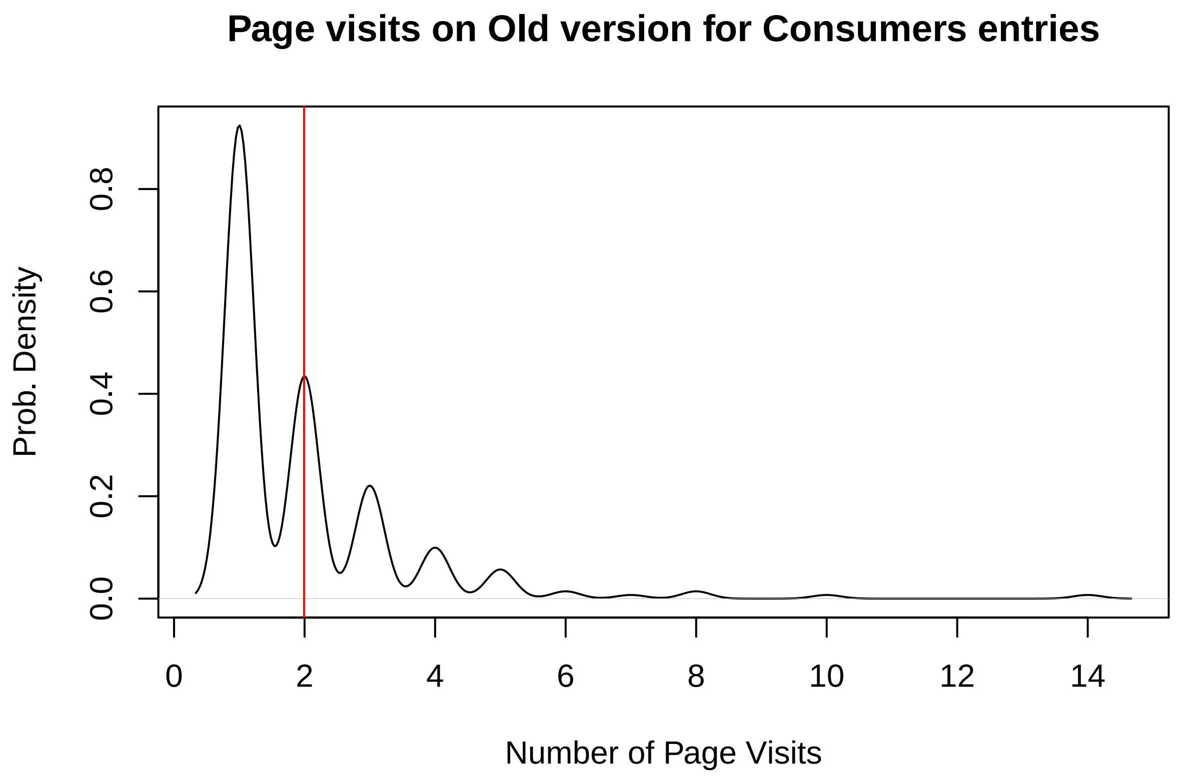
2.2.3 Normality check

```
# Creating subsets of data for each combination of factors

subset00 <- subset(dataWeb, version == '0' & portal == '0')
subset01 <- subset(dataWeb, version == '0' & portal == '1')
subset10 <- subset(dataWeb, version == '1' & portal == '0')
subset11 <- subset(dataWeb, version == '1' & portal == '1')

# Generating density plots

d <- density(dataWeb$pages)
```

```
plot(d, xlab='Number of Page Visits', ylab='Prob. Density', main='Aggregated Page visits')
abline(v = mean(dataWeb$pages), col = "black")
```

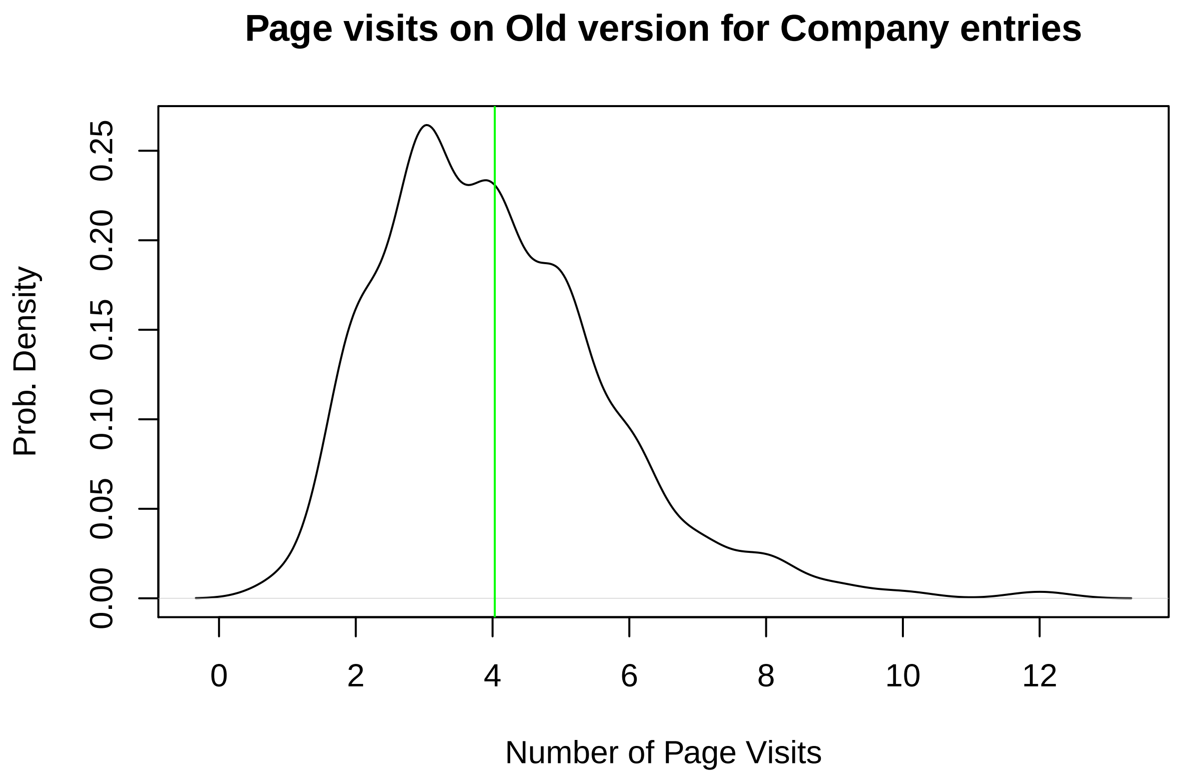


```
d <- density(subset00$pages)
plot(d, xlab='Number of Page Visits', ylab='Prob. Density', main='Page visits on Old version for Consume
abline(v = mean(subset00$pages), col = "red")
```



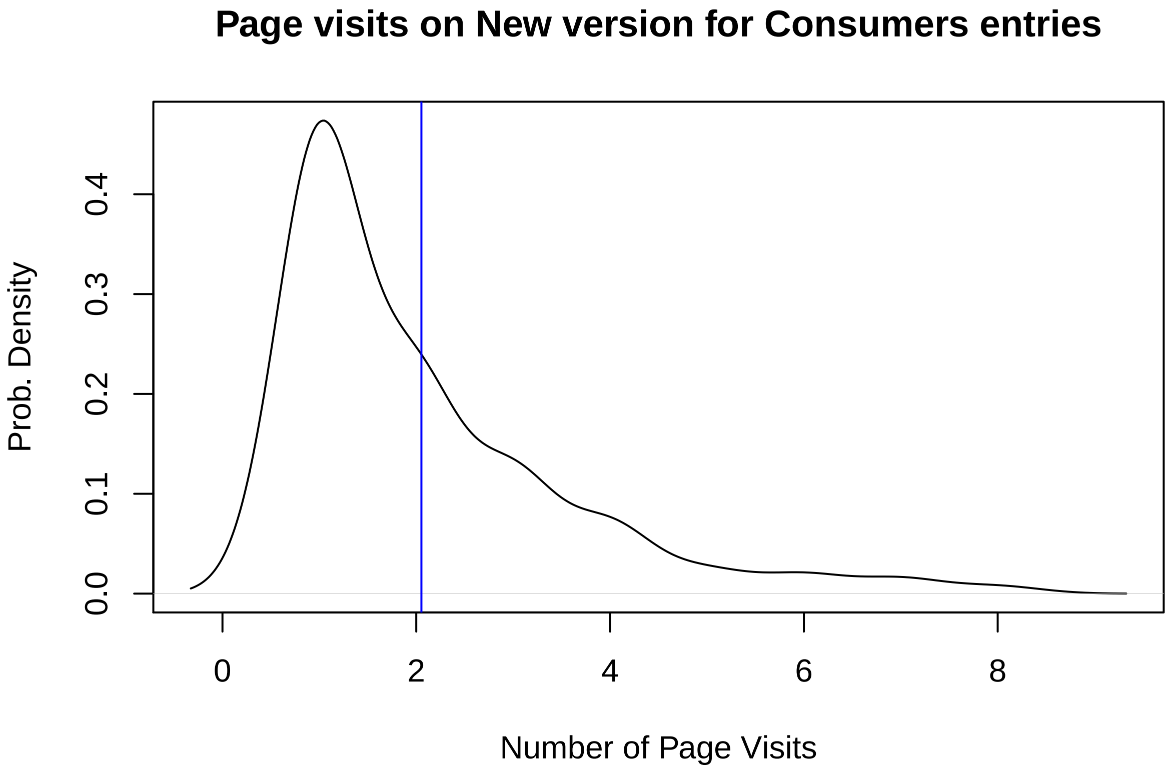
Page visits on Old version for Consumers entries

```
d <- density(subset01$pages)
plot(d, xlab='Number of Page Visits', ylab='Prob. Density', main='Page visits on Old version for Compan
abline(v = mean(subset01$pages), col = "green")
```

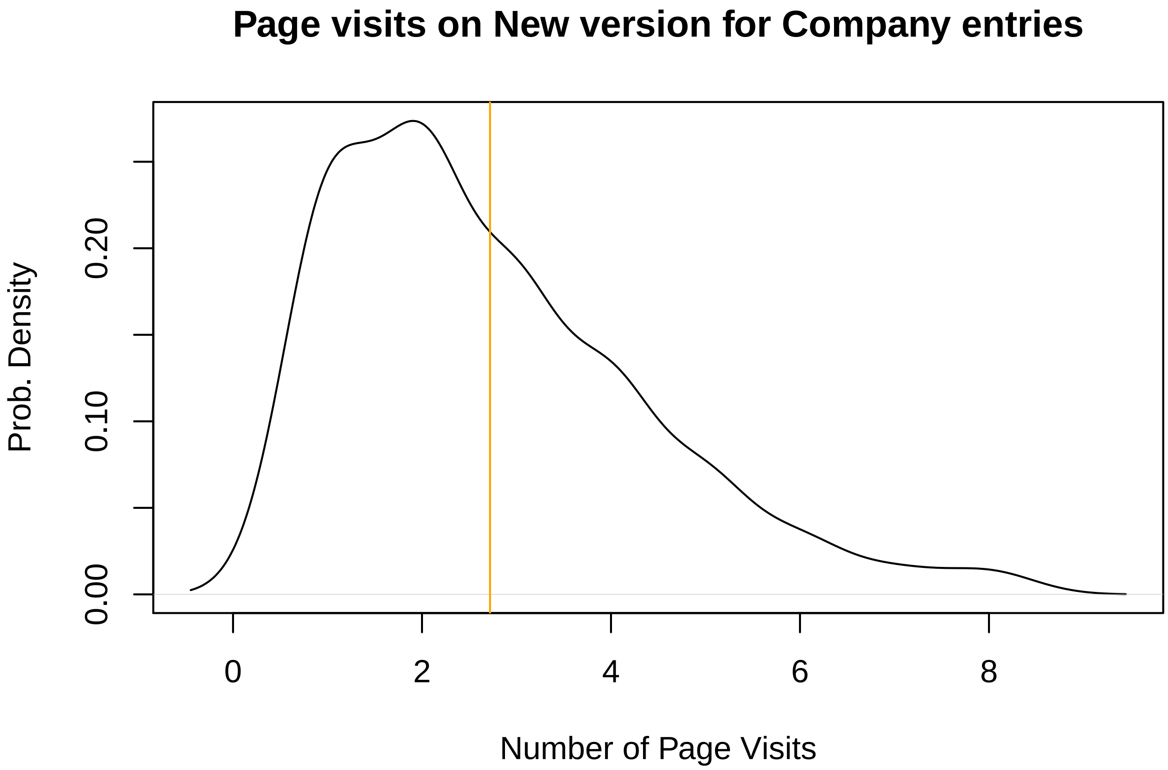


Page visits on Old version for Company entries

```
d <- density(subset10$pages)
plot(d, xlab='Number of Page Visits', ylab='Prob. Density', main='Page visits on New version for Consume
abline(v = mean(subset10$pages), col = "blue")
```



```
d <- density(subset11$pages)
plot(d, xlab='Number of Page Visits', ylab='Prob. Density', main='Page visits on New version for Company
abline(v = mean(subset11$pages), col = "orange")
```



```
# Levene test to check difference of variance
pander(leveneTest(dataWeb$pages, interaction(dataWeb$version, dataWeb$portal)))
```

Table 11: Levene's Test for Homogeneity of Variance (center = median)

	Df	F value	Pr(>F)
group	3	2.562	0.05358
	995	NA	NA

The Density plots indicate that none of the Page visit densities resemble a Gaussian distribtuion, apart from the page visits for New version and Old version company entries. The rest have skewed distributions and the Page visits for Old version consumer entries resembles a mixture of densities. The levene test also indicates that the Homogeniety of Variance assumption has been violated since the p-value is 0.053. Linear Model analysis assumes the fact that the target continuous variable has Gaussian-error distribution and thus uses appropriate log-likelihoods for the best MLE regression fit. Since the densities resemble a Poisson distribution, we use a General Linear Model with the family of distributions set to Poisson in order to conduct model fitting.

2.2.4 Frequentist Approach

```
# Model fitting for each factor and a combination of them

modelWeb0 <- glm(pages ~ 1, data=dataWeb, na.action=na.exclude, family=poisson())
modelWeb1 <- glm(pages ~ version, data=dataWeb, na.action=na.exclude, family=poisson())
modelWeb2 <- glm(pages ~ portal, data=dataWeb, na.action=na.exclude, family=poisson())
modelWeb3 <- glm(pages ~ version + portal, data=dataWeb, na.action=na.exclude, family=poisson())
```

```
modelWeb4 <- glm(pages ~ version + portal + version:portal, data=dataWeb, na.action=na.exclude, family=

# ANOVA results of the effect of adding the factors

pander(anova(modelWeb0, modelWeb1, test="Chisq"), caption='Version as main effect on Page visits')
```

2.2.4.1 Model analysis

Table 12: Version as main effect on Page visits

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
998	1067	NA	NA	NA
997	1033	1	34.25	4.85e-09

```
pander(anova(modelWeb0, modelWeb2, test="Chisq"), caption='Portal as main effect on Page visits')
```

Table 13: Portal as main effect on Page visits

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
998	1067	NA	NA	NA
997	898.8	1	168.2	1.871e-38

```
pander(anova(modelWeb3, modelWeb4, test="Chisq"), caption='Interaction effect vs 2 main effects')
```

Table 14: Interaction effect vs 2 main effects

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
996	862	NA	NA	NA
995	834	1	28.01	1.205e-07

```
pander(anova(modelWeb4, test="Chisq"), caption='Version, Portal and interaction effect on Page visits')
```

Table 15: Version, Portal and interaction effect on Page visits

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL	NA	NA	998	1067	NA
version	1	34.25	997	1033	4.85e-09
portal	1	170.8	996	862	5.015e-39
version:portal	1	28.01	995	834	1.205e-07

```
# AICc scores of the models

modelWebs <-list(modelWeb0, modelWeb1, modelWeb2, modelWeb3, modelWeb4)
model.names <-c("modelWeb0","modelWeb1","modelWeb2","modelWeb3","modelWeb4")
aictab(cand.set = modelWebs, modnames=model.names)
```

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

```
##
##            K    AICc Delta_AICc AICcWt Cum.Wt       LL
## modelWeb4 4 3553.53       0.00      1      1 -1772.75
## modelWeb3 3 3579.53      26.00      0      1 -1786.75
## modelWeb2 2 3614.38      60.85      0      1 -1805.19
## modelWeb1 2 3748.29     194.76      0      1 -1872.14
## modelWeb0 1 3780.53     227.00      0      1 -1889.26
```

The ANOVA results for the comparison of each model type indicate that the added values by including the factors individually, together and their interaction effect is statistically significant since all their p-values are <0.001. The AICc results show that model4 has the best goodness of fit since its corrected-AIC value is the least with the best log-likelihood score too.

```
dataWeb$simple <- interaction(dataWeb$version, dataWeb$portal)
contrast0 <-c(1,-1,0,0) #Only the 0-portal data
contrast1 <-c(0,0,1,-1) #Only the 1-portal data

SimpleEff <- cbind(contrast0,contrast1)
contrasts(dataWeb$simple) <- SimpleEff

simpleEffectModel <-glm(pages ~ simple , data = dataWeb, na.action = na.exclude, family=poisson())
pander(summary.lm(simpleEffectModel))
```

2.2.4.2 Simple effect analysis

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.9509	0.02005	47.43	1.325e-257
simplecontrast0	-0.01509	0.03159	-0.4777	0.633
simplecontrast1	0.1969	0.0247	7.972	4.272e-15
simple	0.4932	0.0401	12.3	1.799e-32

Table 17: Fitting linear model: pages ~ simple

Observations	Residual Std. Error	R^2	Adjusted R^2
999	1	0.6744	0.6735

After fitting a linear model on the data, it can be observed that the company portal entries (1) have a statistically significant difference and not the consumer portal entries (0). This observation also agrees with the first plot indicating the variation in page visits for the 2 factors. The 1-portal page visits have a larger difference than the 0-portal page visits for the 0 and 1 - versions.

2.2.4.3 Report section for a scientific publication

A general linear model with poisson distribution was fit on the number of page visits by users, taking website version and web portal entires as independent variables, and including a two-way interaction between these variables. The analysis found a significant main effect (Chisq (1, 997) = 1033, p. < 0.01) for the version factor and (Chisq (1, 996) = 862, p. < 0.01) for portal factor. The analysis also found a significant two-way interaction effect (Chisq (1, 995) = 834, p. < 0.01) between these two variables. A Simple Effect analysis further examined the two-way interaction. It revealed a significant (z = 7.975, p. < 0.01) difference for the web portal entries by companies (1), but no significant effect (t = -0.4779, p. = 0.6328) was found for the web portal entries by consumers (0).

2.2.5 Bayesian Approach

2.2.5.1 Model description A Poisson distribution model is fitted to each of the models. Model m0 is the base model with only an intercept. Model m1 is an extension of model m0 where the version in introduced as a predictor. Model m2 is again an extension of model m0 with portal as a predictor. In model m3, both predictors are added as main effects, and model m4 extends model m3 by adding a two-way interaction effect between version and portal in the model.

The most complete model is the one which uses both the factors (Version and Portal) to determine the lambda (mean and variance) of the Poisson distribution to model the dependent variable of Page Visits. The prior for the first variable 'a' is chosen to be a normal distribution with the mean as the mean of the page visits from the data and the uncertainty in this estimate is reflected by the standard deviation of the mean page visits as 1. The priors for the coefficients of Version and Portal are chosen to be normal distributions of mean 2 and deviation 1, mean 0 and deviation 1 since these will be anyway adjusted by the counts of the factors. The choice of these priors are based on intuitve understanding of the page visits based on the 2 factors of version and portal type that influence the number of page visits.

$$pages \sim Poisson(\lambda)$$

$$\lambda = a + b * versionN + c * portalN + d * versionN * portalN$$

$$a = Norm(2, 1)$$

$$b = Norm(2, 1)$$

$$c = Norm(0, 1)$$

$$d = Norm(0, 1)$$

```r
datasub <- subset(dataWeb, select = c(pages, version, portal))
datasub$versionN <- as.numeric(datasub$version)
datasub$portalN <- as.numeric(datasub$portal)

#Fitting each variant of the model

mWeb0 <-map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- a ,
    a ~ dnorm(2, 1)
  ), data = datasub, iter = 10000, chains = 4, cores = 4
)
```

2.2.5.2 Model comparison

```
## Computing WAIC
```

```
mWeb1 <-map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- a + b*versionN ,
    a ~ dnorm(2, 1),
    b ~ dnorm(2, 1)
  ), data = datasub, iter = 10000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
mWeb2 <-map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- a + b*portalN ,
    a ~ dnorm(2, 1),
    b ~ dnorm(0, 1)
  ), data = datasub, iter = 10000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
mWeb3 <-map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- a + b*versionN + c*portalN ,
    a ~ dnorm(2, 1),
    b ~ dnorm(2, 1),
    c ~ dnorm(0, 1)
  ), data = datasub, iter = 10000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
mWeb4 <-map2stan(
  alist(
    pages ~ dpois(lambda),
    log(lambda) <- a + b*versionN + c*portalN + d*versionN*portalN ,
    a ~ dnorm(2, 1),
    b ~ dnorm(2, 1),
    c ~ dnorm(0, 1),
    d ~ dnorm(0, 1)
  ), data = datasub, iter = 10000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
pander(compare(mWeb0,mWeb1,mWeb2,mWeb3,mWeb4))
```

	WAIC	SE	dWAIC	dSE	pWAIC	weight
mWeb4	3554	59.31	0	NA	3.893	1
mWeb3	3580	58.61	25.98	9.982	2.999	2.284e-06
mWeb2	3615	60.14	60.98	14.49	2.106	5.741e-14
mWeb1	3749	55.01	195.2	27.47	2.35	4.204e-43
mWeb0	3781	56.82	227.2	29.37	1.194	4.638e-50

```
pander(precis(mWeb4, prob= .95))
```

	mean	sd	2.5%	97.5%	n_eff	Rhat4
a	-0.3735	0.2031	-0.7671	0.0336	2497	1.002
b	0.3974	0.1306	0.1318	0.648	2437	1.002
c	1.073	0.119	0.8333	1.302	2501	1.002
d	-0.391	0.07758	-0.5402	-0.2332	2460	1.003

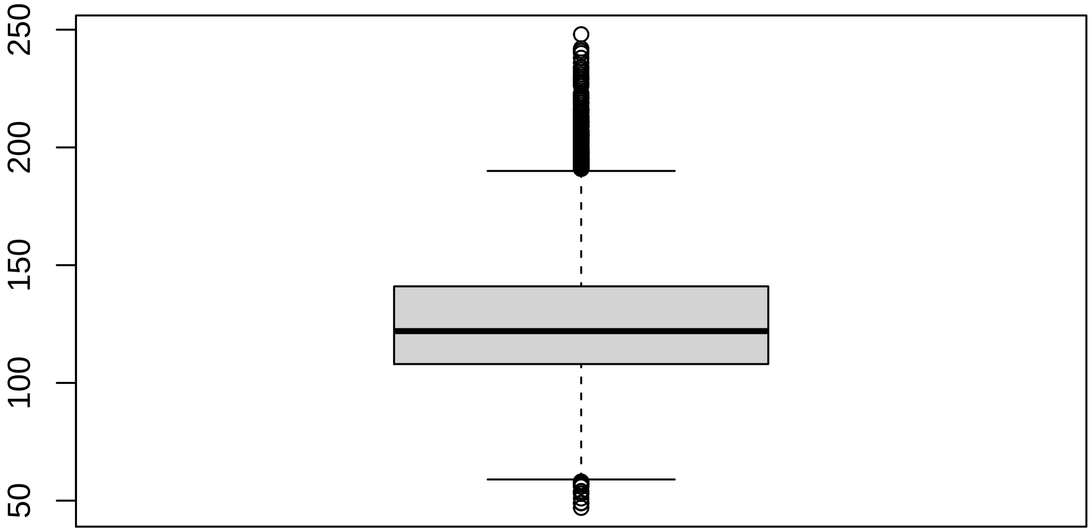
The comparison between all the models indicates that the model m4, which includes the main effects and interaction effects of version and portal type, has the best out-of-sample fit since it has the lowest WAIC score. Looking at the 95% credible intervals of the parameters of model m4, it is to be noted that none of the coefficients include 0 in their 95% confidence interval.

3 Part 3 - Multilevel model

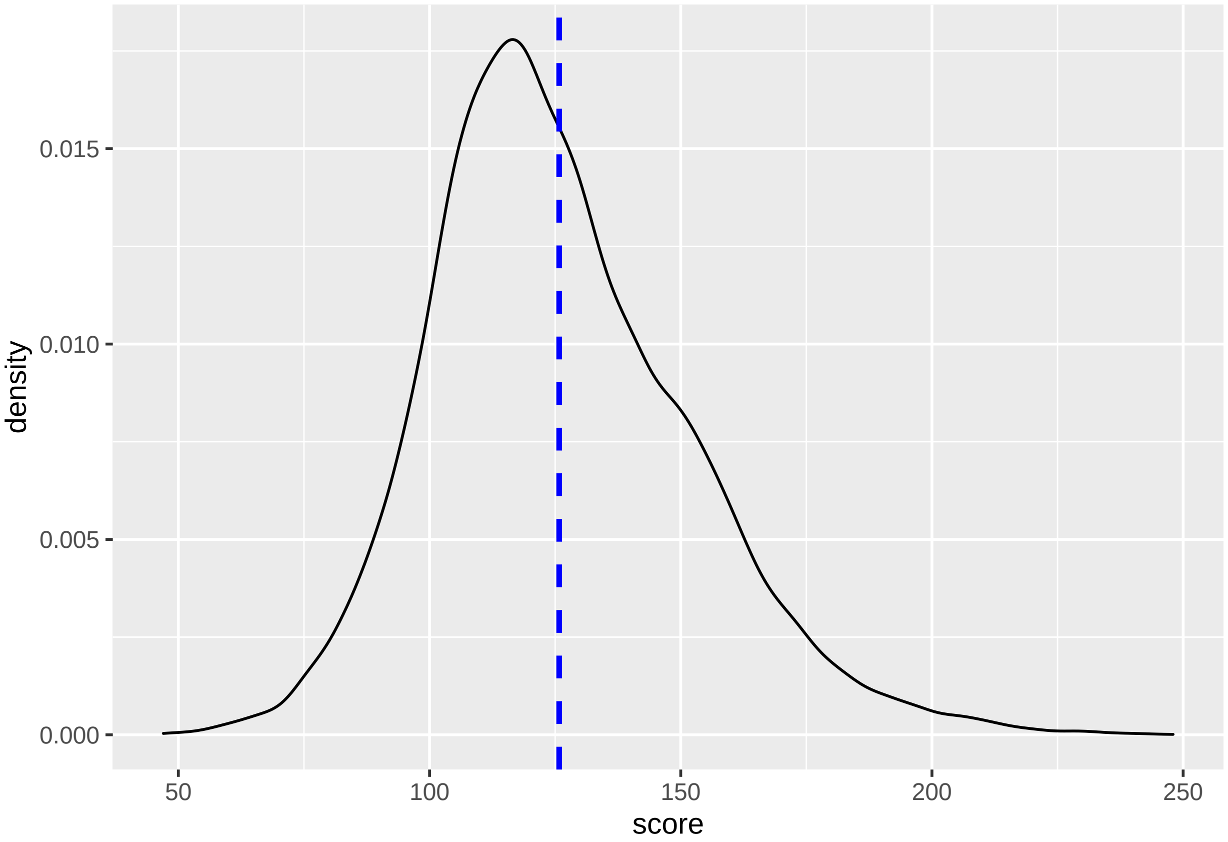
3.1 Visual inspection

The boxplot and density plot show the distribution of the score. We can see that the mean score is 122 points. The minimum is set at 59, with outliers until 46, while the maximum is set at 190, with outliers until 248.

```
# Get data
filepath <- ("set0.csv")
ds <- read.csv(file=filepath, header=TRUE)
ds <- data.frame(ds)

# boxplot score overall distribution (session independent)
boxplot(ds$score)
```

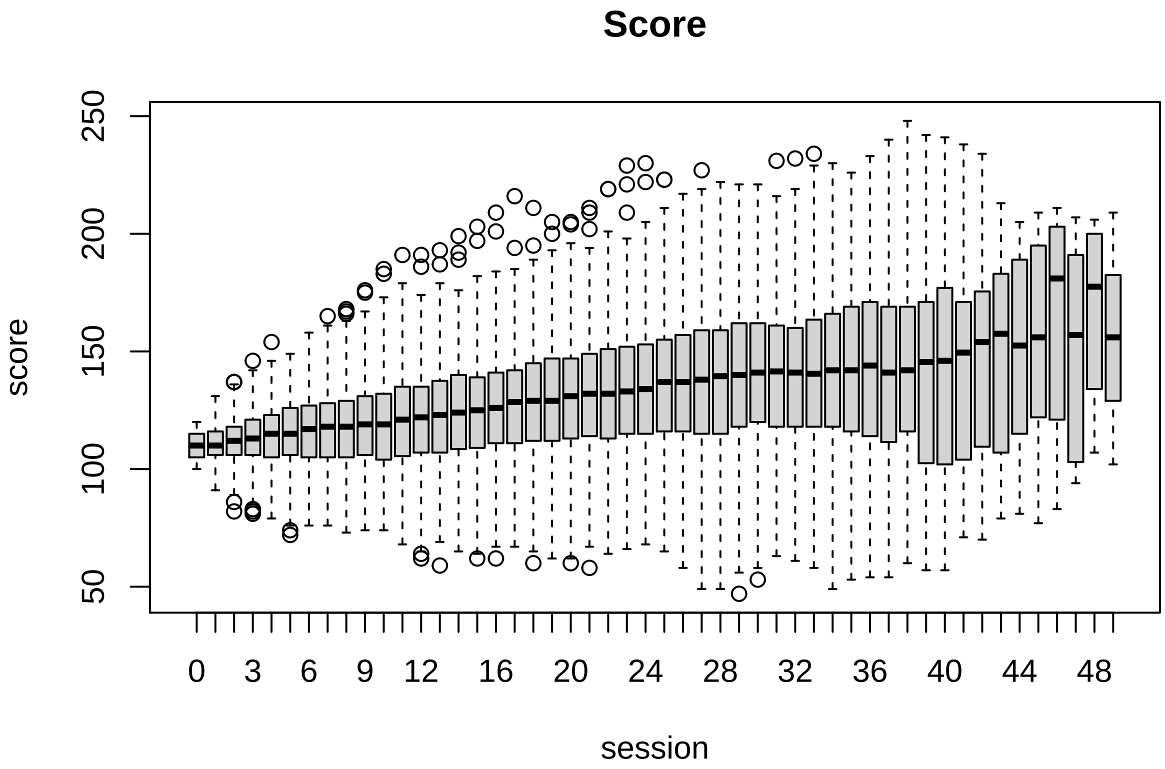


```
# density score overall distribution (with mean line)
p <- ggplot(ds, aes(x=score)) + geom_density()
p + geom_vline(aes(xintercept=mean(score)), color="blue", linetype="dashed",size=1)
```

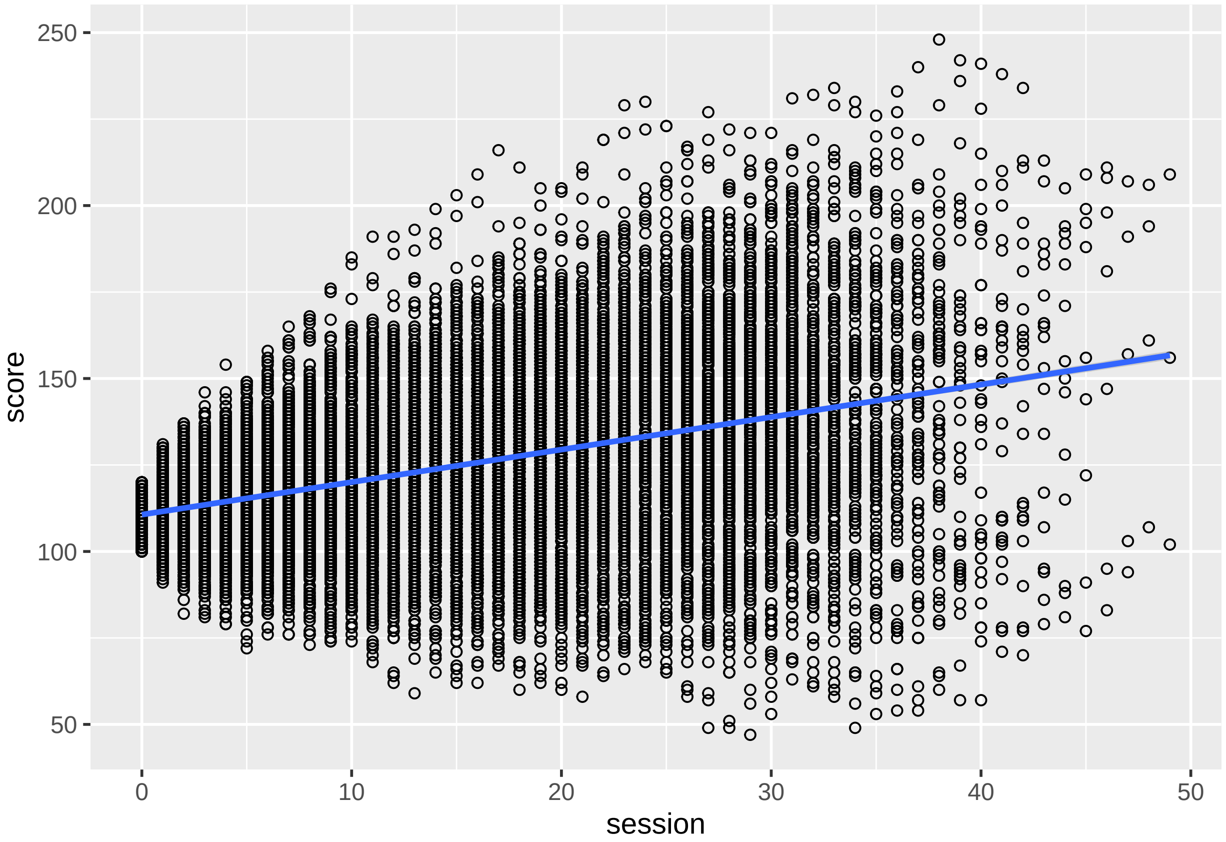


The relationship between the score and the session can be observed with the next two figures. The regression line (blue) in the scatterplot clearly shows how the score rises with the amount of sessions. This can also be observed in the box plot when looking at the mean (black line) for every box. Because of this, we expect that there is a fixed effect for the score over the sessions.

```
# boxplot score per session
boxplot(score~session, data=ds, main="Score", xlab="session", ylab="score")
```



```r
# ggplot score per session
hp <- ggplot(ds, aes(x=session, y=score)) + geom_point(shape=1) +
  geom_smooth(formula = y ~ x,method=lm)
hp
```



3.2 Frequentist approach

3.2.1 Multilevel analysis

We have conducted a multilevel analysis. We have a model with a fixed intercept (model0, with *1*) and a random intercept, *1|Subject*. This model is being compared to a second model (model1), which includes the variable session, i.e. the session numb into *score~1+session*. By comparing the fit of the two models, we can see that the model extension results in a significantly better fit, $\chi^2(1)$=6306.00, p<0.001.

```
model0 <- lme(score~1, random = ~1|Subject, data=ds, na.action=na.exclude, method="ML")

model1 <- lme(score~session, random = ~1|Subject, data=ds, na.action=na.exclude, method="ML")

anova(model0,model1)
```

```
##        Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## model0     1  3 138742.5 138765.6 -69368.25
## model1     2  4 132438.5 132469.2 -66215.25 1 vs 2 6306.005  <.0001
```

The output of the summary of model0 shows the standard deviation between the random-effect terms, being 19.07 for the intercept per subject, and 16.8 for the residuals. The fixed intercept value is 125.5. The summary of model1 shows an estimated fixed effect for session on the score of 0.98. With a p-value of 0.000, making this fixed effect significant.

```
pander(summary(model0))
```

Table 20: Fixed effects: score ~ 1

	Value	Std.Error	DF	t-value	p-value
(Intercept)	125.5	0.8626	15627	145.5	0

Table 21: Standardized Within-Group Residuals

Min	Q1	Med	Q3	Max
-4.616	-0.6201	-0.01503	0.5934	5.382

Table 22: Linear mixed-effects model fit by maximum likelihood : score ~ 1

	Observations	Groups	Log-restricted-likelihood
Subject	16128	501	-69368

```
pander(summary(model1))
```

Table 23: Fixed effects: score ~ session

	Value	Std.Error	DF	t-value	p-value
(Intercept)	110.2	0.879	15626	125.4	0
session	0.9814	0.01114	15626	88.13	0

Table 24: Standardized Within-Group Residuals

Min	Q1	Med	Q3	Max
-4.471	-0.6351	-0.005518	0.6229	5.295

Table 25: Linear mixed-effects model fit by maximum likelihood : score ~ session

	Observations	Groups	Log-restricted-likelihood
Subject	16128	501	-66215

The summary compares the first session (intercept) with the other sessions. Looking at the estimates, we can see that compared to the first session, the scores are higher every later session. Next, we will take a look at the Akaike Information Criterion (AIC) to compare the models on the goodness-of-fit concering the out-of-sample deviance. Here we can see that model1 has the best goodness-of-fit as it has the smallest AICc value.

```
models <- list(model0, model1)
model.names <- c("model0", "model1")
pander(aictab(cand.set = models, modnames=model.names),
       caption="Model selection based on AICc.")
```

Table 26: Model selection based on AICc.

	Modnames	K	AICc	Delta_AICc	ModelLik	AICcWt	LL	Cum.Wt
2	model1	4	132438	0	1	1	-66215	1
1	model0	3	138742	6304	0	0	-69368	1

Lastly, we will obtain a 95% confidence interval, giving an insight into whether the terms are significant.

```
intervals(model0,0.95)
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                lower     est.    upper
## (Intercept) 123.8475 125.5381 127.2288
## attr(,"label")
## [1] "Fixed effects:"
##
##  Random Effects:
##   Level: Subject
##                    lower     est.   upper
## sd((Intercept)) 17.89637 19.06765 20.3156
##
##  Within-group standard error:
##    lower     est.    upper
## 16.66315 16.84891 17.03674
```

```
intervals(model1,0.95)
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                   lower        est.      upper
## (Intercept) 108.4981564 110.2209275 111.943699
## session       0.9595362   0.9813616   1.003187
## attr(,"label")
## [1] "Fixed effects:"
##
##  Random Effects:
##   Level: Subject
##                    lower     est.    upper
## sd((Intercept)) 17.95951 19.12538 20.36693
##
##  Within-group standard error:
##    lower     est.    upper
## 13.61904 13.77087 13.92439
```

3.2.2 Report section for a scientific publication

To confirm whether the sessions had an impact on the subject's scores, a multilevel analysis was performed. This showed that there was a significant variance between the subjects and their score, SD = 19.07 (95% CI 17.90, 20.31). The relationship between the score and the session showed significant variance in intercepts across subjects as well, SD = 19.13 (95% CI 17.96, 20.37), $\chi^2(1)$=6306.00, p<0.001. The score had an estimated fixed effect of 0.981 over the sessions.

3.3 Bayesian approach

3.3.1 Model description

For model2, the model with session as a factor, we have a normal distribution containing a linear model and a fixed prior. The linear model consists of a fixed prior with normal distribution of N(125,27). This comes from the mean of the score, 125, and the standard deviation, which is around 27. It further consists of an adaptive prior for subject, being a normal distribution with mean coming from the mean of the sessions, and a fixed prior for session, also being a normal distribution. The fixed prior, sigma, is a cauchy distribution Cauchy(20,0.5), estimated from the expected variance in scores.

```
ds <- ds[!(ds$Subject>99),] # select first 100 subjects
ds$Subject <- ds$Subject +1 # increase subject number with 1 to overcome Stan zero index problem

mean(ds$score) # check mean of score 125
```

```
## [1] 125.5142
```

```
sd(ds$score) # check standard deviation of score 27.4 (met hoger aantal 25.77 -> 1.7)
```

```
## [1] 27.402
```

$$score \sim Norm(\mu, \sigma)$$

$$\mu = a + a_{subject_i} + b * session$$

$$a_{subject_i} \sim Norm(0, a_\sigma)$$

$$a_\sigma \sim Cauchy(0, 10)$$

$$a \sim Norm(125, 27)$$

$$b \sim Norm(0, 10)$$

$$\sigma = Cauchy(20, 0.5)$$

3.3.2 Model comparison

We have created and compared the three described models. From the results we can see that model2 has the best fit since it has the smallest WAIC value and largest Akaike weight.

```
# create model with fixed intercept (i)
m0 <- map2stan(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a_subject[Subject],

    # fixed prior
    a_subject[Subject] ~ dnorm(125,27),
    sigma ~dcauchy(20,0.5)
  ), data = ds, iter = 10000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
# create model extended with an adaptive prior for subject id (ii)
m1 <- map2stan(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a + a_subject[Subject],

    # adaptive prior
    a_subject[Subject] ~ dnorm(0,sigma_subject),

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

    # fixed prior
    a ~ dnorm(125,27),
    sigma ~dcauchy(20,0.5)
  ), data = ds, iter = 10000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
# create model with session as a factor (iii)
m2 <- map2stan(
  alist(
    score ~ dnorm(mu, sigma),
    mu <- a + a_subject[Subject] + b*session,

    # adaptive prior
    a_subject[Subject] ~ dnorm(0,sigma_subject),

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

    # fixed priors
    a ~ dnorm(125,27),
    b ~ dnorm(0,10),
    sigma ~dcauchy(20,0.5)
  ), data = ds, iter = 10000, chains = 4, cores = 4
)
```

```
## Computing WAIC
```

```
pander(compare(m0,m1,m2,func=WAIC))
```

	WAIC	SE	dWAIC	dSE	pWAIC	weight
m2	26948	91.46	0	NA	95.25	1
m1	28322	95.9	1373	75.81	92.51	5.759e-299
m0	28322	96.16	1374	75.87	93.5	4.814e-299

3.3.3 Estimates examination

From the previous comparison we could see that model2 is the best fit model. We take a look at the estimated values of the parameters of this model. We can see that not all values were correctly estimated by us.

```
pander(precis(m2, depth=1, prob=.95))
```

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

	mean	sd	2.5%	97.5%	n_eff	Rhat4
sigma_subject	20.28	1.454	17.67	23.37	12554	0.9999
a	108.5	2.012	104.6	112.5	445.5	1.003
b	1.043	0.02576	0.9927	1.093	12938	1
sigma	14.5	0.1835	14.14	14.86	14377	0.9999