Analysis of language games using mixed models

Arjan Bontsema

July 11, 2014

Contents

Abstract

Wordrobe is a set of language games, created in order to crowd source linguistic information. This provides a lot of data from thousands of questions. It is assumed that the true answer is given by the majority. This thesis uses confidence intervals for a binomial proportion to determine the certainty on the true answer. In addition, it can be determined at what moment in time there is enough data. Furthermore, generalized linear mixed models are introduced. The logistic model, including random effects, is applied to the data. For this model, the effects on answering a question correctly are described. Random effects lead to measures for the skill of players and difficulty of questions.

Chapter 1

Introduction

Wordrobe is a collection of games designed to gain linguistic knowledge from the general public. The games all have the same structure, similar to a quiz game. The goal of the game is to answer a series of questions and gain points based on the agreement with other players that have answered the same questions. The creator of Wordrobe wants to use the data to extract linguistic information from the agreement of players for each question. The Wordrobe games can be played freely on the website

`http://wordrobe.org`

There are multiple games. Each game consist of thousands of questions. For each question there are few possible choices that a player can select. The choices differ per question. If a person plays the game, one of the

Note that players do not answer all questions. They play as many as they like. Only a few players are very active and play a lot, where most players are very little active. Also note that different players answer different questions, since questions are presented randomly.

The file `wordrobe20140326.csv` contains all data from the Wordrobe data. It can be downloaded from

`http://gmb.let.rug.nl/data.php`

The data set consists of 47401 records, each record contains the data of a player selecting an answer to a given question. Each record includes the following fields:

- name of the game
- text of the question
- encrypted user name

- text of the selected answer
- bet, in range from 10 to 100
- encoded answer of the answer (BOW)
- the number of choices associated to the question
- expert opinion, having value 1,0 or NA

The problem that the linguistics researchers face is: How much data do we need to be sufficiently sure about the true answer. The true answer is for most questions unknown and therefore it is of our interest to find a method that describes the certainty about the true answer. Then it may be possible to determine how much data is needed in order to achieve enough certainty.

Furthermore we want to know what affects the probability that a player answers a question correctly. One could think of variables like the bet, which are given in the data set. Other effects may be skill and difficulty of players. Is it possible to find a model that describes these effects? In order to do this, generalized linear models will be used and an important extension: generalized linear mixed models.

This thesis consists of two chapters. In the first chapter the mathematical theory behind the thesis is discussed, mostly about the generalized linear (mixed) models and about the binomial proportion confidence interval. In the second chapter the analysis on the data is done. First there is a part on exploratory analysis in order to get a better look at the structure of the data. After that, the theory will be applied to the data and the results will be discussed.

Chapter 2

Theory

2.1 Generalized linear models

Generalized linear models are an extension of the ordinary regression models. GLMs include describing models with non-normal distributions. Therefore GLMs treat a wide range of data, having different types of response variables. First the exponential family is described. It is a necessary property for the data to be a member of the exponential family, in order to apply GLM. Then the model will be described, including procedures to estimate the model.

2.1.1 Exponential family

Generalized linear models are described for response variables that are member of the exponential family. A random variable Y is a member of the exponential family if the corresponding probability distribution, depending on a single parameter θ can be written in the form

$$f(y|\theta) = \exp[a(y)b(\theta) + c(\theta) + d(y)]. \tag{2.1}$$

If $a(y) = y$, then the distribution is in canonical form. The term $b(\theta)$ is called the natural parameter. If the probability distribution depends on other parameters, next to the parameter of interest, then these are treated as known nuisance parameters, being part of the functions a,b,c and d.

A few examples of distributions that are members of the exponential family are Exponential, Gaussian, Binomial, Poisson and Multinomial distribution. Other densities, for example the Weibull and negative binomial distribution are not members of the exponential family, however after some modifications GLM can be fit.

The mean and variance of members of the exponential family can computed in terms of the functions a,b,c and d.

$$\mathrm{E}[a(Y)] = -\frac{c'(\theta)}{b'(\theta)}. \tag{2.2}$$

$$\mathrm{Var}[a(Y)] = \frac{b''(\theta)c'(\theta) - c''(\theta)b'(\theta)}{[b'(\theta)]^3}. \tag{2.3}$$

From 2.1, the likelihood of Y is given by

$$\ell(\theta; y) = a(y)b(\theta) + c(\theta) + d(y). \tag{2.4}$$

The derivative of the likelihood with respect to θ is

$$U(\theta; y) = \frac{d\ell(\theta; y)}{d\theta} = a(y)b'(\theta) + c'(\theta). \tag{2.5}$$

where the function U is called the score statistic. Since it depends on the random variable Y, it can be regarded as a random variable:

$$U = a(Y)b'(\theta) + c'(\theta). \tag{2.6}$$

From 2.2 is follows that $\mathrm{E}(U) = 0$. The variance of the score statistic is called the information, denoted by $\mathfrak{I}$. Using 2.6 and 2.3, the information can be defined as

$$\mathfrak{I} = \mathrm{Var}(U) = [b'(\theta)]^2 \mathrm{Var}[a(Y)] \tag{2.7}$$

Another property that holds is

$$\mathrm{Var}(U) = \mathrm{E}(U^2) = -\mathrm{E}(U'). \tag{2.8}$$

This property is very useful for fitting the GLM.

All details and proofs of the properties of the exponential family are given by Dobson (2001).

2.1.2 Model specification

A generalized linear model can be described by three components. First component identifies the response variable Y and its probability distribution. The response variable should be a member of the exponential family. The second component gives a description of the linear predictor as a linear combination of the explanatory variables. The third component component is the link function, which describes the relation between the mean of the response and the linear predictor; the second component.

The first component is the response variable Y and its probability distribution. Consider the independent observations $(y_1, \ldots, y_N)$ from a distribution that is member of the exponential family, see section 2.1.1. The corresponding probability distributions have the form

$$f(y_i; \theta_i) = \exp[a(\theta_i)b(y_i) + c(\theta_i) + d(y_i)], \quad i = 1, 2, \ldots, N \tag{2.9}$$

Note that the values of θ_i may differ and depend on the values of the explanatory variables.

The second component gives the relation between the vector of linear predictors $(\eta_1, \eta_2, \ldots, \eta_N)$ and the explanatory variables by a linear model. Denote x_{ij} as the value of variable (or predictor) j for subject i, where $j = 1, \ldots, p$. The number of coefficients in the model is given by p. The linear predictors are given by

$$\eta_i = \sum_{j=1}^{p} x_{ij}\beta_j = x_i^T\beta, \quad i = 1, 2, \ldots, N. \tag{2.10}$$

where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_p)^T$.

The third component is the link function. This function gives the link between the first two components. Let $\mu_i = E(Y_i)$ for $i = 1, \ldots, N$. Then the link function g is a monotonic and differentiable function, such that

$$g(\mu_i) = \eta_i = x_i^T\beta, \quad i = 1, 2, \ldots, N.$$

As an example, the link function $g(\mu) = \mu$ is the identity link function. This results in a linear model for the mean and describes ordinary regression, where the response Y is normally distributed. A link function that links the mean of the response to the natural parameter of its probability distribution is called a canonical link function.

2.1.3 Estimation

Now the GLM is specified, the coefficients $\beta_1, \ldots, \beta_p$ are of interest. There are multiple methods to estimate the parameter values for the model. The most commonly used method is maximum likelihood estimation. In some special cases it occurs that parameters can be given by exact mathematical expressions. However, it is usually needed to use numerical methods. For GLMs these methods are based on the Newton-Raphson algorithm.

Let $Y_1, \ldots, Y_N$ be independent random variables. Assume they satisfy the properties of a generalized linear model. We want to estimate the parameters $\beta_1, \ldots, \beta_p$. Recall that the likelihood for each Y_i is

$$\ell(\theta_i; y_i) = \log f(y_i; \theta_i) = y_i b(\theta_i) + c(\theta_i) + d(y_i),$$

since Y_i is a member of the exponential family. It follows that the joint log-likelihood function is given by

$$\ell(y_1, \ldots, y_N) = \sum_{i=1}^{N} \ell(\theta_i; y_i) = \sum_{i=1}^{N} y_i b(\theta_i) + \sum_{i=1}^{N} c(\theta_i) + \sum_{i=1}^{N} d(y_i). \tag{2.11}$$

The parameter values $\beta_1, \ldots, \beta_p$ are then fitted such that the likelihood is maximum. For GLMs this can be done by use of the Newton-Raphson algorithm. Note that the score statistic is the derivative of the log-likelihood, with respect to the parameter θ. This is used to derive the derivative of the log-likelihood with respect to the coefficients β. This equals zero when the likelihood is maximum. The zero is found by use of the Newton-Raphson algorithm. This method is also called the method of scoring. Dobson (2001) describes the procedure in detail. It is furthermore shown that maximum likelihood estimators for generalized linear models are obtained by an iterative weighted least squares procedure.

In **R** the function `glm` uses iterative least squares to estimate the parameters.

2.1.4 Logistic regression model

A specific form of a generalized linear model is the logistic model, a model for binary data. Binary responses usually correspond success and failure, for example presence or absence of a considered object, votes in an election, false or true answering to a question.

Define a binary random variable

$$Y = \begin{cases} 1 & \text{if success} \\ 0 & \text{if failure} \end{cases} \tag{2.12}$$

Assume that the response Y has a binomial distribution, so the probability density function is

$$f_Y(y;\pi) = \pi^y(1-\pi)^{1-y}, \tag{2.13}$$

where the parameter $\pi = \Pr(Y=1)$. The response is a member of the exponential family, since

$$\begin{aligned} f_Y(y;\pi) &= \pi^y(1-\pi)^{1-y} \\ &= \exp\left[y\log(\pi) + (1-y)\log(1-\pi)\right] \\ &= \exp\left[y\log\left(\frac{\pi}{1-\pi}\right) + \log(1-\pi)\right]. \end{aligned}$$

The mean of the response is $\mathrm{E}[Y] = \Pr(Y=1) = \pi$. Recall from the generalized linear model that we want to model a linear combination of the explanatory variables as function of mean. Denote the mean $\mathrm{E}[Y] = \pi(x)$, considering the dependency on the explanatory variables $x = (x_1, \ldots, x_p)$.

Usually, the binary data results in a non-linear relation between x and $\pi(x)$. The most important nonlinear model for $\pi(x)$ is the logistic regression model. It is described by the formula

$$\pi(x) = \frac{\exp x^T\beta}{1+\exp x^T\beta}. \tag{2.14}$$

This formula comes from the following: To ensure that the π lies within the interval $[0,1]$, the probability is often modelled using a cumulative density function

$$\pi = \int_{-\infty}^{x} f(s)ds, \tag{2.15}$$

where f is called the tolerance function, it is non-negative and $\int_{-\infty}^{\infty} f(s)ds = 1$.

Consider for simplicity $x^T\beta = \beta_1 + \beta 2x$, for the logistic (or logit) model, the tolerance function is

$$f(s) = \frac{\beta_2\exp(\beta_1+\beta_2 x)}{[1+\exp(\beta_1+\beta_2 x)]^2}. \tag{2.16}$$

It follows that

$$\pi = \int_{-\infty}^{x} f(s)ds = \frac{\exp(\beta_1 + \beta_2 x)}{1 + \exp(\beta_1 + \beta_2 x)}. \tag{2.17}$$

The link function that links the mean of the response to the linear predictors $x^T\beta$ is then easily computed.

$$x^T\beta = g(\mathrm{E}[Y]) = g(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \mathrm{logit}(\pi). \tag{2.18}$$

The interpretation of the logit function is very natural, since it is the logarithm of the odds. From this, the coefficients *beta* can be easily interpreted.

Consider for example $x^T\beta = \beta_1 + \beta_2 x$, where the explanatory variable represents the presence of something, e.g. the presence of a treatment. In that case the coefficient β_2 represents the effects of the treatment. Filling in x gives

$$\begin{aligned} \beta_1 + \beta_2 &= \mathrm{logit}(\pi(1)) \\ \beta_1 &= \mathrm{logit}(\pi(0)) \\ \beta_2 &= \mathrm{logit}(\pi(1)) - \mathrm{logit}(\pi(0)) = \log(\frac{\pi(1)/(1-\pi(1))}{\pi(0)/(1-\pi(0))}). \end{aligned}$$

Hence β_2 is the log odds ratio. It can be interpreted that for a present treatment, the odds is $\exp(\beta_2)$ times higher than for responses without a treatment x. If x can take any numerical value, that $\exp(\beta_2)$ is interpreted as the change the odds ratio if x increases by 1.

So the logistic model for binary data is now defined. Note that there are multiple types of models for binary data, each having different types of link functions, for example the probit model. However, it is usual to use the logistic model. Furthermore, the interpretation of the logit model is very useful.

2.2 Generalized linear mixed models

Mixed effects models are another extension of the linear model. Consider a GLM. the parameters β may describe the effects of a factor. These parameters are called the fixed effects. They refer to all possible categories. For example when considering gender, age, treatment, etc. The effect is considered to be the same, given the explanatory variables. In this section, the Generalized linear mixed model (GLMM), that considers random effects, is explained.

First the idea of random effects will be explained, then the GLMM will be defined. The effect on the likelihood will be shown. This has an effect on how the model should be estimated.

2.2.1 Adding random effects to the GLM

Consider explanatory variables with a high number of levels. Observations are often given in clusters, for example clusters of observations from a given subject or item. The effects of the explanatory variables may differ per cluster. In other words, given in what cluster a observation is, there is a certain effect on the linear predictors.

As an example, considering the parameter β_j in the GLM, this is a fixed value. This value describes the effect on the linear predictor, given the explanatory variable j. In the contrast, when considering random effects, a cluster of the data is considered. Consider cluster k, corresponding to the data from subject k. Then a coefficient γ_{jk} is considered. This parameter describes the effects of the explanatory variable j in cluster k.

The random effects are usually considered to be normally distributed. The mean effect is assumed to be zero. This is a reasonable assumption, since a random effect with a nonzero mean splits into two parts, where one part is a fixed effect including the mean. The other part describes the random effects, with mean zero.

A GLMM looks like a GLM. Therefore, defining the GLMM is about the same. It is still defined in the same way. First the response variable Y and its probability distribution have to be given. The response variable should be a member of the exponential family. Secondly, the linear predictors as a linear combination of the explanatory variables are given. For a GLMM the linear predictor for observation i, given in cluster k, have the form

$$\eta_{ik} = x_i^T \beta + z_i^T \gamma_k, \tag{2.19}$$

where the vector γ_k is assumed to have a multivariate normal distribution: $\gamma_k \sim \mathcal{N}(0, \Sigma)$ for $k = 1, \ldots, K$, where K represents the number of clusters or subjects. $z_1, \ldots, z_m$ are the explanatory variables of the observations that involve random effects. In many cases, the random effects consider only one variable. The $m \times m$ variance matrix Σ consists of the unknown variances for each random effect. It possibly also includes parameters for co-variances.

The third component is the link function, which describes the relation between the mean of the response and the linear predictor. It is given by

$$g(\mu_{ik}) = \eta_{ik}$$

For GLMM the link function $g(\cdot)$ is the same as for the GLM, so the only change in the definition is the new component in the linear predictor.

Random effects have been used particularly in models for categorical data, where the effects of clusters for a particular category can be considered randomly drawn.

2.2.2 Estimation

In this section, the estimation of the model parameters is explained. Mainly the idea about maximizing the likelihood is explained, as where the parameters for a ordinary GLM are also determined by maximizing the likelihood. Rewriting the likelihood results in a complicated integral, after this is done, methods for estimating the parameters are discussed.

Consider the generalized linear mixed model as it is defined in the previous section. Let $Y_1, \ldots, Y_N$ be independent random variables. We want to maximize the likelihood, with respect to the parameter values β and Σ. The likelihood for these parameters is defined as the probability density function of the responses given β, Σ, but this is not defined. Therefore, it is necessary to condition on γ, since we know that $\gamma \sim \mathcal{N}(0, \Sigma)$. The joint likelihood is given by

$$\begin{aligned} L(\beta, \Sigma; y_1, \ldots, y_n) &= \prod_{i=1}^{N} f(y_i; \beta, \Sigma) && (2.20) \\ &= \prod_{i=1}^{N} \int_{-\infty}^{\infty} f(y_i; \beta, \gamma) f(\gamma; \Sigma) d\gamma, && (2.21) \end{aligned}$$

where

$$f(\gamma; \Sigma) = (2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2} x^T \Sigma^{-1} x}, \tag{2.22}$$

and

$$f(y_i; \beta, \gamma) = \pi_{\beta,\gamma}^{y_i} (1 - \pi_{\beta,\gamma})^{1-y_i},$$

where

$$\pi_{\beta,\gamma} = \frac{\exp\left[x_i^T \beta + z_i^T \gamma\right]}{1 + \exp\left[x_i^T \beta + z_i^T \gamma\right]}.$$

Observe that the likelihood is indeed independent of γ, as there is integrated over all possible values. Note that from this expression, the values of the random vector γ_k, the random effect for cluster k are not derived. It is at last a random variable. It is however possible to compute the expected values of this effects, given the measures: Take for simplicity the example where $z_i = 1$. Then the linear predictors have the form $x_i^T \beta + \gamma_i$, where $\gamma_i \sim \mathcal{N}(0, \sigma_\gamma^2$. This describes a model with only a random intercept for each cluster. Note that $\mathrm{E}\gamma_i = 0$. This is not very interesting, but since $y_1, \ldots, y_n$are known, the expectation $\mathrm{E}(\alpha_i | y_i)$ can be computed. In this simple situation, the expectation can be computed from a Bayesian point of view, the posterior density for γ_i is

$$f(\gamma_i | y_i) \sim f(y | \gamma_i) f(\gamma_i) \tag{2.23}$$

The random effects can be estimated. $\hat{\gamma_i}$ can be computed as the posterior mean

$$E(\gamma|y) = \int \gamma_i f(\gamma_i|y_i) d\gamma_i. \tag{2.24}$$

In the multidimensional case it can also be computed, see Diggle et al. (2002). Most model fitting methods in **R** include the estimated effects.

Finding the maximum of the likelihood, which is a product of integrals, is very difficult. Computing it exactly is in many cases impossible, therefore the maximum likelihood estimators are computed by numerical methods. For GLMs, the Newton-Raphson method is used, but this method is not sufficient for GLMM. When the random effects are normally distributed, the numerical approximation of the integral is usually done by Gauss Hermite quadrature. Then a multivariate version of the Newton-Raphson can be applied. In Tuerlinckx et al. (2006), the Gauss Hermite quadrature method for estimation parameters in GLMMs is explained in full detail. Most important part of this method is that integrals of the form $\int u(t)e^{-t^2}dt$ can be estimated by

$$\int u(t)e^{-t^2}dt \approx \sum_{b=1}^{m} u(t_b)v_b, \tag{2.25}$$

where v_b and t_b are nodes and weights from the Gauss Hermite quadrature.

In **R** there are many packages that can model mixed effects models, each with different types of estimation. In the package `lme4` includes the function `glmer`, which makes use of the GH quadrature. This function also includes the expected values of the random effects.

2.3 Inference

This section considers inference on Generalized linear (mixed) models. Most common tools in statistical inference are confidence intervals and hypothesis testing. Both methods can be used to test goodness of fit of a model. The goodness of fit statistics are based on the maximum value of the log-likelihood, residuals, etc.

2.3.1 Likelihood ratio

A methods for model comparison is be the log likelihood ratio statistic, also denoted by deviance. Suppose we want to test the goodness of a model. The maximum value of the likelihood of this model is denoted by $L(b, y)$. Consider another model that is compared to the model of interest. This model is called the saturated model. The maximum value of the likelihood of this model is denoted by $L(b_{\max}, y)$. Note that the number of parameters in the model of interest is p, where we denote the number of parameters in the saturated model by m. It holds that $p < m \leq N$. Define the likelihood ratio between these two models by

$$\lambda = \frac{L(b_{\max}, y)}{L(b, y)}. \tag{2.26}$$

This ratio provides a method to describe the goodness of fit for the model. The logarithm of the likelihood ratio, which is the difference is log-likelihood, is mostly used in practice.

$$\log \lambda = \ell(b_{\max}, y) - \ell(b, y). \tag{2.27}$$

As $\log \lambda$ has a large value, it means that the saturated model is, compared to the model of interest, a better description of the data. Hence, the model of interest has a relative poor fit. The deviance, also called the log-likelihood ratio statistic, is defined by

$$D = 2log\lambda = 2[\ell(b_{\max}, y) - \ell(b, y)]. \tag{2.28}$$

In order to evaluate the value of the deviance and determine the goodness of fit, a critical region has to be determined. Therefore its sampling distribution is needed.

2.3.2 Deviance

As the deviance is defined, the sampling distribution can be approximated. In this section the sampling distribution is derived. Furthermore it is shown how the deviance can be used in hypothesis testing.

Assume that b is the maximum likelihood estimator of β. The value of the likelihood function at the parameter values β, $\ell(\beta)$, is unknown but estimated by $\ell(b)$. Consider the Taylor approximation that gives an approximation of the log-likelihood near an estimate b.

$$\ell(\beta) \approx \ell(b) + (\beta - b)^T U(b) - \frac{1}{2}(\beta - b)^T \mathfrak{I}(b)(\beta - b),$$

where $U(b) = \frac{d\ell}{d\beta}$ evaluated at b and $\mathfrak{I} = \mathrm{E}[U']$, with $U' = \frac{\partial^2 \ell}{\partial \beta^2}$.

Note that b is the maximum likelihood estimator of *beta*, such that $U(b) = 0$. Therefore it follows that

$$2[\ell(b,y) - \ell(\beta,y)] \approx (\beta - b)^T \mathfrak{I}(b)(\beta - b). \tag{2.29}$$

This has a chi-squared distribution $\chi^2(p)$, where p is the numbers of parameters. This is the Wald-statistic and its distribution is also expressed by

$$b \sim \mathcal{N}(\beta, \mathfrak{I}^{-1}).$$

The sampling distribution of deviance can be approximated using this result.

$$\begin{aligned}
D &= 2[\ell(b_{\max}, y) - \ell(b, y)] \\
&= 2[\ell(b_{\max}, y) - \ell(\beta_{\max}, y)] + 2[\ell(\beta_{\max}, y) - \ell(\beta, y)] - 2[\ell(b, y) - \ell(\beta, y)] \\
&\approx 2[\ell(b_{\max}, y) - \ell(\beta_{\max}, y)] - 2[\ell(b, y) - \ell(\beta, y)]
\end{aligned}$$

Since the term $2[\ell(b_{\max}, y) - \ell(\beta_{\max}, y)]$ is near zero as the model of interest fits the model almost as good as the saturated model. From the previous result it follows that

$$\begin{aligned}
2[\ell(b_{\max}, y) - \ell(\beta_{\max}, y)] &\sim \chi^2(m) \\
2[\ell(b, y) - \ell(\beta, y)] &\sim \chi^2(p) \\
D = 2[\ell(b_{\max}, y) - \ell(\beta_{\max}, y)] - 2[\ell(b, y) - \ell(\beta, y)] &\sim \chi^2(m - p)
\end{aligned}$$

From this the goodness of fit of a model can be tested. If the deviance is consistent with the chis-squared distribution, the smaller model fits the data as good as the larger model. Furthermore, hypothesis testing on values of β can be done. Consider two models, M_0 and M_1. The models need to be nested. That means, the models need to have the same probability distribution, link function, data. Furthermore, M_0 is a special case of M_1. Consider the null hypothesis and alternative hypothesis

$$\begin{aligned}
H_0 &: \beta = \beta_0 \\
H_1 &: \beta = \beta_1,
\end{aligned}$$

where M_0 consists of q parameters and M_1 of p parameters. These are the dimensions of the vectors β_0, β_1. Furthermore, $q < p < N$. The hypothesis can be tested by using the difference in deviance.

$$\Delta D = D_0 - D_1 = 2[\ell(b_1, y) - \ell(b_0, y)].$$

From the earlier results in this section, it follows that $\Delta D \sim \chi^2(p-q)$.

As the value of ΔD is consistent with the chi-squared distribution, model M_0 is generally chosen, since it is simpler. Note that H_1 can be accepted as model M_1 gives a significantly better description of the data, even though the model does not fit the data well.

2.4 Certainty about binomial proportion

In order to determine how much data is needed, a definition must be given when there is enough certainty. It has to be determined when there is enough data to be sufficiently sure about the parameter value. A way to specify this is by confidence intervals for the binomial proportion. Another way to approach this is by sequential hypothesis testing. First some preliminairies about the binomial distribution.

2.4.1 Binomial distribution

Let $Y_1, \ldots, Y_N$ be N independent random variables. Assume these random variables are Bernoulli distributed, $Y_i \sim \text{Bern}(\pi)$, $i = 1, \ldots, N$. The probability distribution is defined by

$$\Pr(Y_i = 1) = \pi = 1 - \Pr(Y_i = 0), \quad i = 1, \ldots, N.$$

The probability density function is given by

$$f_{Y_i}(y; \pi) = \pi^y (1-\pi)^{1-y}, \quad y \in \{0, 1\}, \quad i = 1, \ldots, N.$$

The expectation and variance are $\text{E}[Y_i] = \pi$ and $\text{Var}[Y_i] = \pi(1-\pi)$ for $i = 1, \ldots, N$

Now define the random variable Y as a sum of random variables, $Y = \sum_{i=1}^N Y_i$. Then Y has a binomial distribution, $Y \sim \text{Bin}(\pi, N)$. The probability density function is given by

$$f_Y(y; \pi, N) = \pi^y (1-\pi)^{N-y}.$$

Since the random variables $Y_1, \ldots, Y_N$ are independent and identically distributed, the expectation for Y is given by

$$\text{E}[Y] = \text{E}[\sum_{i=1}^N Y_i] = \sum_{i=1}^N \text{E}[Y_i] = N\text{E}[Y_1] = N\pi.$$

The variance can be obtained by the same way:

$$\mathrm{Var}[Y] = \mathrm{Var}[\sum_{i=1}^{N} Y_i] = \sum_{i=1}^{N} \mathrm{Var}[Y_i] = N\mathrm{Var}[Y_1] = N\pi(1-\pi).$$

Another random variable is defined as $\bar{Y} = \frac{1}{N}Y = \frac{1}{N}\sum_{i=1}^{N} Y_i$. This is the mean of the N Bernoulli distributed variables.

The expectation and variance of the mean are

$$\mathrm{E}[\bar{Y}] = \pi, \quad \mathrm{Var}[\bar{Y}] = \frac{\pi(1-\pi)}{N}.$$

2.4.2 Certainty

For observed data that look like drawings from a binomial distributions, the true parameter value is of interest. This parameter is unknown, but it can be estimated, based on the observed data. First a few methods of estimation are described. When estimating parameters, we have to deal with uncertainty. Therefore the confidence intervals are described. For large data samples this can by done by use of the central limit theorem. From the normal distribution, the confidence interval can be computed asymptotically. Also hypothesis testing will be discussed, which is a more direct approach to test certain parameter values. Assuming that the data is time ordered, we arrive at a sequential testing procedure.

Confidence interval

Consider now n observations from a Bernoulli distributed variable. Denote the observations by $(y_1, \ldots, y_n)$. The parameter of interest, π is unknown, but can be estimated. An estimator for p is given by $\hat{\pi} = \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{x}{n}$, where x is the number of successes. However, this estimator does not give a lot of information. It is just a single point, but we have no idea about the accuracy. We want to know how good this estimation is. This will be done by a confidence interval. First the confidence interval for large sample sizes is given.

Recall the central limit theory, as n is large,

$$\frac{\bar{y} - \pi}{\sigma} \xrightarrow{d} \mathcal{N}(0,1), \tag{2.30}$$

where π is the true parameter value and $\sigma^2 = \mathrm{Var}[\bar{y}] = \frac{\pi(1-\pi)}{n}$. Since π is unknown, σ is also unknown, but it can be asymptotically estimated by the sample variance $\tilde{\sigma}$, using $\hat{\pi}$ in stead of p.

$$\hat{\sigma}^2 = \frac{\hat{\pi}(1-\hat{\pi})}{n}.$$

The $(1-\alpha)100\%$ confidence interval can then be described by

$$CI = \hat{\pi} \pm z_{\alpha/2}\frac{\tilde{\sigma}}{\sqrt{n}}. \tag{2.31}$$

In case of the standard confidence interval for the binomial proportion, where $\hat{\pi} = \bar{y}$, the confidence interval is estimated by

$$CI = \bar{y} \pm z_{\alpha/2}\sqrt{\frac{\bar{y}(1-\bar{y})}{n}}.$$

Now the confidence interval is defined for the binomial proportion. This can be used to determine how much data is needed to be sufficiently sure. Note that the confidence interval shrinks as n increases, see figure 2.4.2. This implies that a value of n can be found such that the confidence interval is small enough. This confidence interval, using the exact estimator $\bar{y}$, is known as the Wald interval. Agresti and Coull (1998) described the problem of this type of confidence interval. In case of the 95% CI, they show that the coverage probability of the Wald confidence interval is very poor, especially for extreme values of π. They argue that using an estimation rather than the exact estimation for the binomial proportion gives a better result. Especially in the case where π is close to the boundaries, the Wald interval does not give a good result. They define a approximate interval, for which the coverage is better.

The Agresti-Coull interval has a familiar form as the Wald interval. It can also be written as in equation 2.31, but for the Agresti-Coull interval, another estimation $\hat{\pi}$ is used. Recall that in the standard interval $\hat{\pi} = \frac{x}{n}$ is used. Now define

$$\tilde{x} = x + \frac{1}{2}z^2_{\alpha/2} \text{ and } \tilde{n} = n + z^2_{\alpha/2}.$$

Then the estimation for the binomial proportion is given by $\tilde{\pi} = \frac{\tilde{x}}{\tilde{n}}$, such that the confidence interval is given by

$$CI_{AC} = \tilde{\pi} \pm z_{\alpha/2}\sqrt{\frac{\tilde{\pi}(1-\tilde{\pi})}{\tilde{n}}} \tag{2.32}$$

In Agresti and Coull (2008), this confidence interval is described in full detail. Note that for the 95% CI we have that $z_{\alpha/2} = 1.96$. Consider $z_{\alpha/2} = 2$. Then the Agresti-Coull CI corresponds to the "add 2 successes and 2 failures" confidence interval, this is $\tilde{\pi} = \frac{x+2}{n+4}$. It is shown that the coverage probability of this

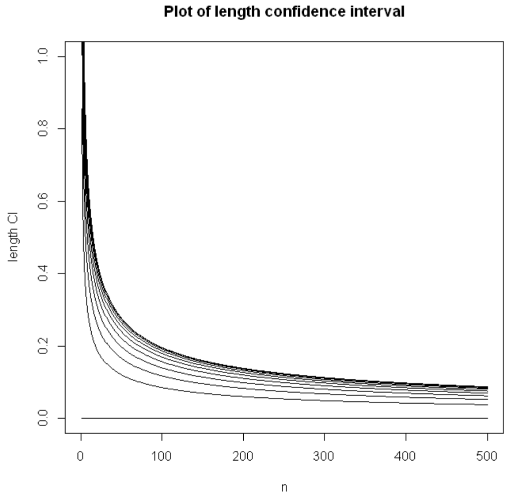

Figure 2.1: Length 95% CI for different values of p, depending of the nr. of observations n.

interval is larger than for the exact confidence interval, even for small n. In figure 2.4.2, this is also illustrated.

Note that for an estimate $\tilde{\pi}$ the $(1-\alpha)100\%$ confidence lower bound is given by

$$\tilde{\pi} - z_\alpha \frac{\tilde{\sigma}}{\sqrt{n}}. \tag{2.33}$$

This means that we are $(1-\alpha)100\%$ certain that the true binomial proportion lies above this bound. Similarly, the confidence upper bound is defined.

$$\tilde{\pi} + z_\alpha \frac{\tilde{\sigma}}{\sqrt{n}}. \tag{2.34}$$

Hypothesis testing

Another way to say something about the certainty of estimated parameter value, is by use of hypothesis testing. Similar to the estimation of the confidence interval, the probability that the binomial proportion lies within a certain region can be computed. Both methods are related, since testing the hypothesis

$$H_0 : \pi = \pi_0$$
$$H_1 : \pi \neq \pi_0$$

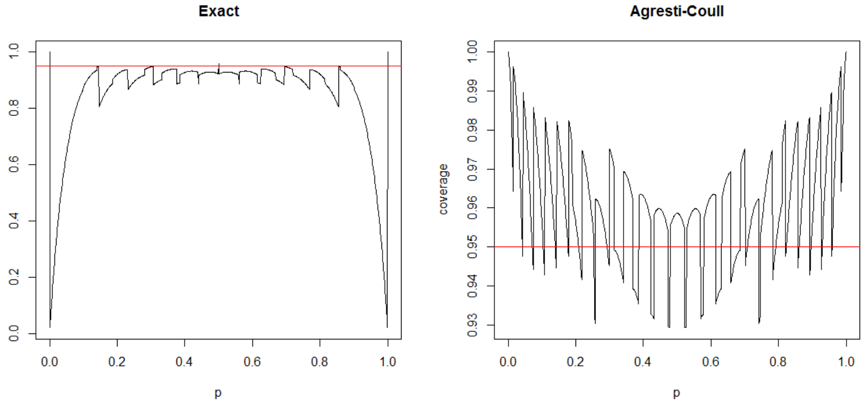

Figure 2.2: Coverage of the 95% CI for n=20

can be performed by determining whether the value π_0 lies within the confidence interval.

Assume again that there are n observations $(y_1, \dots, y_n)$, for which we estimate the binomial proportion $\hat{p} = \frac{1}{n}\sum_{i=1}^{n} y_i$. To test the hypothesis

$$\begin{aligned} H_0 &: \pi \leq \pi_0 \\ H_1 &: \pi > \pi_0 \end{aligned}$$

with a 5% significance level, one can use the 95% confidence interval, 2.33 to test the hypothesis. In the same context, the probability that H_0 is true can also be computed. Assume that $\frac{\tilde{\pi}-\pi_0}{\tilde{sigma}}$ is normally distributed, to obtain

$$\begin{aligned} \Pr(\pi \leq \pi_0) &= \Pr(\frac{\pi - \tilde{\pi}}{\tilde{s}} \leq \frac{\pi_0 - \tilde{\pi}}{\tilde{s}}) \\ &\approx \phi(\frac{\pi_0 - \tilde{\pi}}{\tilde{s}}), \end{aligned}$$

where $\phi(\cdot)$ is the cumulative density function for the standard normal distribution. Observe that having a p-value less than 5% is equivalent to a 95% confidence lower bound that lies above π_0.

2.4.3 How much is enough?

In the previous section we discussed hypothesis testing as there were n observations to estimate the binomial proportion. In some applications, it is interesting to consider how much data is needed to be sufficiently certain about the parameter value. Or in other perspective, how much data is enough to reject a null-hypothesis. One has to assume that as the number of observation is very large, the hypothesis will be rejected or accepted.

Consider as an example the hypothesis

$$H_0 : \pi \leq \pi_0 H_1 : \pi > \pi_0$$

where π is the parameter of interest. π_0 is a chosen value. Recall the confidence lower bound. From the definition of a confidence interval, it can be observed that the amount of needed observations is not always the same, and depends, besides the selected value π_0 also on the value of $\tilde{p}$. And since this value changes as observations add to the data, it is not really possible to pick a integer n^* for which observations $y_1, \ldots, y_{n^*}$ provides enough information. Therefore it necessary to test the null hypothesis each time step, where at each step there is a new observation.

Consider n observed drawings from a binomial distribution, $y_1, \ldots, y_n$. Assume that these observations are time-ordered. Each observation y_k corresponds to a moment in time, k. The amount of observations n^* that provides enough information to reject the null-hypothesis can be defined as

$$n^* = \min\left\{ k \mid \tilde{p} - z_\alpha \sqrt{\frac{\tilde{p}(1-\tilde{p})}{k+4}},\ \text{where } \tilde{p} = \frac{\sum_{i=1}^{k} y_i + 2}{k+4} \right\} \tag{2.35}$$

Since n^* can not be determined on before hand, it is necessary to test the hypothesis after each observation. Therefore, this method results in a sequential hypothesis testing procedure.

The use of this procedure is clear in the context of the linguistic data. Consider the binomial proportion as the proportion of the players that select a specific answer. The null-hypothesis can be defined as $\pi \leq 0.5$. Rejecting this hypothesis implies that the true parameter is larger than 0.5, which means that the majority of the population selects the considered answer. As the hypothesis that the majority selects a considered answer can be accepted, the amount of data is sufficient. The testing procedure will be more clear as it will be applied later on.

Chapter 3

Data analysis

3.1 Exploratory data analysis

In this exploratory data summary, the data set of the Wordrobe games will be described. Before applying any of the theory, the data is explored. What is the structure of the data and of what use can it be. After analyzing each variable, the problems of the data set will be discussed. If possible, some suggestions to improve the data are done. Then the part of most interest come in: What can be learned from the data.

3.1.1 Variables

First each variable of the data will be discussed.

The Wordrobe data is given in one data set. It contains 47,401 observations of 11 variables. Each observation is also called a record. Each record corresponds to a answer that is given by one player to one question. The variables describe the question, answer, username, and other variables that correspond to this event.

Each record consists of 11 variables. The variables are:

question The text of the question

game The name of the game: burgers, pointers, twins, senses or names

user Username of the player (anonymized for privacy)

answer The text of the answer

bet The value of the 'bet' slider, ranging from 10 to 100

bow Bit Of Wisdom, the data associated to the answer

n.choices How many possible choices the question had

n.answer How many answers the question received

relmaj The highest number of concordant answers

agreement the measure of agreement between players on the question

expert.opinion Gold standard information, when a match is found with human expert judgement

Game

The variable `game` represents the question that is formulated in the game. The variable contains a string of the question, there are five games: Burgers, Pointers, Twins, Senses and Names. In table 3.1.1, the number of records per game are given.

	unique questions
pointers	10836
senses	9598
twins	20808
names	5185
burgers	974

Table 3.1: Number of records per game

Question

The variable `question` represents the question that is formulated in the game. The variable contains a string of the question, for example

```
 "Ẅhy don’t you <b>kill</b> it at once, like a lady?"̈
```

Note that it is possible that the formulation of a question may occur in multiple games. In the data, it looks like the variable contains the question, but it actually contains only the formulation of the question. This is something different. Consider for example the question

```
"Änd I think they’ll <b>want</b> a one-stop shop in terms of combining security,
immigration, customs, and quarantine togetherâ€¦ just to make sure it’s more
streamlined and provides more certainty."̈
```

In both games Senses and Twins, there are questions with this formulation. However, the question is different since this is not the complete question. It is only the sentence that is considered, but in both games the goal of the game is different and therefore also the possible options. Therefore, the variable `question` should be considered in combination with the variable `game`. It is also possible to consider data for each game apart to avoid this problem.

There are 7805 unique values for the variable `question`, however, in table 3.1.1 we can see how many unique questions there are per game. Considering this, we obtain that we have 8551 unique question.

	unique questions
pointers	2020
senses	3036
twins	1799
names	934
burgers	762

Table 3.2: Number of unique questions per game

In figure 3.1.1 the number of records per question is illustrated. For the game Burgers, Most questions are only answered once. Therefore, this data seems not very useful. For the games pointers and senses, the number of records per question is also low. The game pointers is played often, with respect to the number of unique questions. On average each questions is answered 11.57 times.

Username

The variable `username` represents the player that answered the corresponding question within the record. The variable contains an encrypted string of the username, for example

```
"15b7d8cbe8229f7a06d5911048c6a2cb335bcf0589bd9573cde54c94"
```

There are 883 unique players in all data, so 883 unique values for the variable `username`. There are of course user that play different games. Therefore, we can consider the variable `username` per game:

	unique users
names	136
senses	512
twins	837
pointers	394
burgers	11

Table 3.3: Number of unique players per game

The game Twins seems to be very popular, since 837 out of the 883 unique players played the game at least once. Only 11 players ever played the game Burgers. In figure 3.1.1 of the number of records per user. It represents how much questions each player has answered. Observe that most players do not play a game very often. There are a few players that play the game extremely often, compared to the other players. Note that in this histograms, the high peaks are not shown completely. The histograms are zoomed in towards zero with respect to the frequency. This is done to show the extreme values. Consider for example the game Senses. There is one player that provides 1875 records. This is 19.5 % of all data from this game. We see this pattern

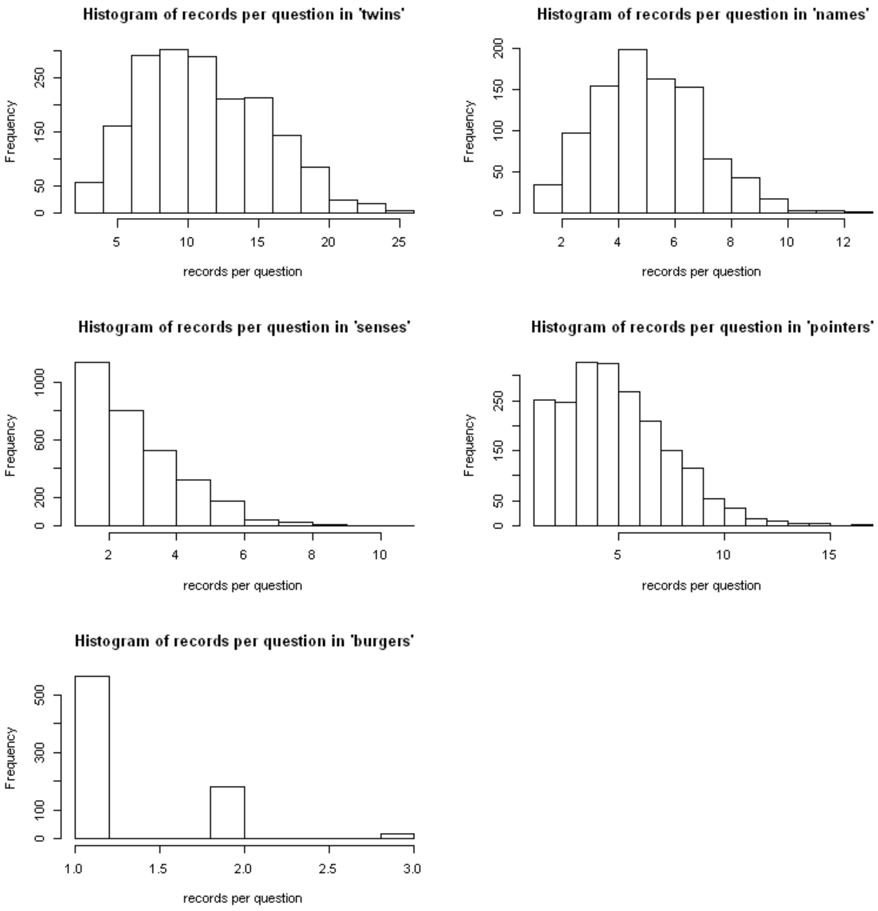

Figure 3.1: Histogram of the number of records per question received

in most of the games, there are a few players that play a lot more than most other players.

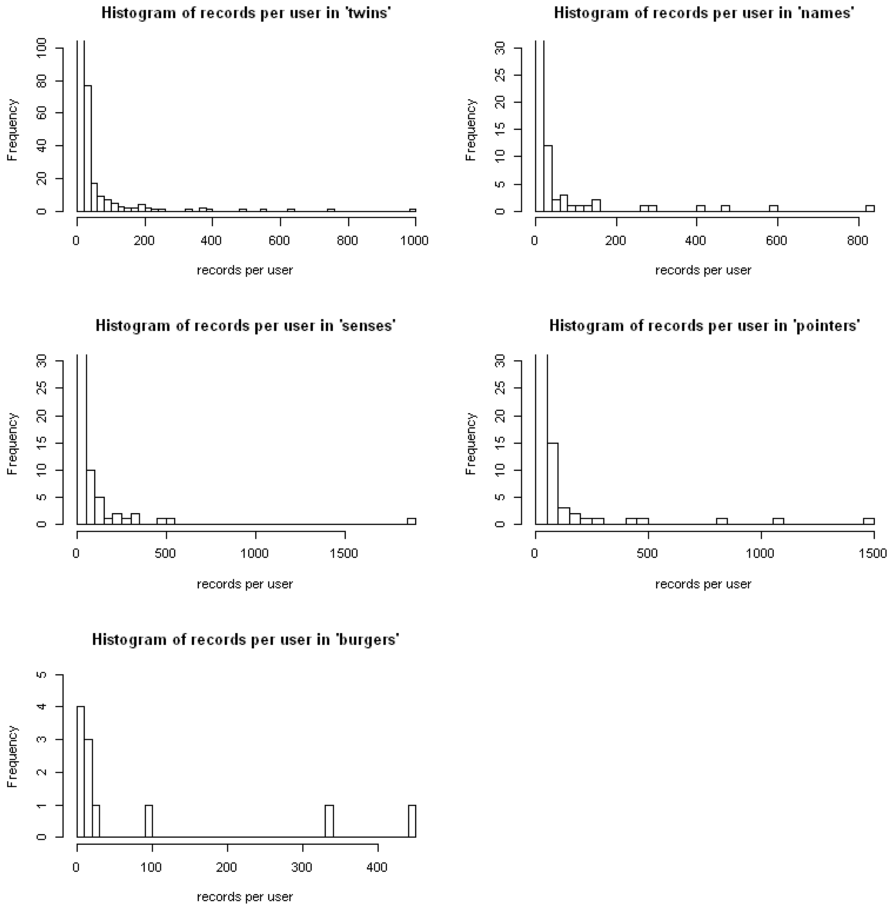

Figure 3.2: Histogram of the number of records per player

Answer

The variable `answer` represents the answer that a player selected to the corresponding question within the record. The variable contains a string with the formulated answer. Consider for example in the game Senses the question

`"Ẅhy don’t you kill it at once, like a <b>lady</b>?"`

The user ”7385a8dda0c552f24a81fbf777496cfca3fe573cec10beba2e751896” selected the answer

`"a woman of refinement (synonyms: dame, madam, ma’am, gentlewoman)".`

The variable answer is not very interesting to analyze. It only gives information that can be used to conclude what the correct answer is. However, since in some games the possible answers differ per question, the values of the variable are not of interest.

For the games Twins and Names, the possible answers are the same for all questions. In the game Twins, a player can choose between two options: 'noun' or 'verb'. In the game Names a player always has the same seven options.

Expert opinion

The variable `expert.opinion` gives the opinion of a group of experts, according to the answers that is given to a question. The variable is a string, and can take 3 values: 'true','false','unknown'. 'true' if the answer corresponds to the answer of the experts, 'false' if not. Since the experts did not evaluate all questions, the variable is 'unknown' otherwise.

The data for questions of which we know the correct answer, hence `expert.opinion` is 'true' or 'false', is called the **gold-standard** data.

	false	true	unknown
burgers	0	1	973
names	100	390	4695
pointers	0	1894	8942
senses	88	75	9435
twins	31	494	20283

Table 3.4: Gold-standard data per game

From this table it can be seen that for most questions the expert opinion is unknown: There is only a small set of data for which the true answer is known. Furthermore, we see that the game Burgers has only one gold-standard question. If we use the variable `expert.opinion` to analyze the data, then the data from the game Burgers seems to be of no use.

Another observation from this table is that the game 'pointers' is (for the gold-standard data) always answered correctly. This is possible, but not very useful. A variable that is always the same is not interesting.

Bet

The variable `bet` is represent the bet that players have to give. In all games, there is a slider that users can set. The idea of this slider is that users can give a bet that should represent the certainty of the given answer. The variable `bet` in our data set gives a number from the set { 10,20,30,40,50,60,70,80,90,100 } and corresponds to the value of the slider. The higher the bet, the more certain a player should be.

However, this method is not completely reliable. How aware are the players of their skills? Another problem might be the fact that it is a game. Asking for a bet might result in gambling behaviour. The effect of the slider is however invisible to the player. Even though it might have an effect on the score, the player can not see it, so maybe he will lose interest in using it. This is a reasonable, since the player is not obligated to use the slider. Once it is set, it stays where it is put and the same bet will be used. It might also be possible that a player forgets to set the slider.

In table 3.1.1, the proportions for the bets for each game are given. In each game, most times the maximum bet of 100 is selected.

	Bet									
	10	20	30	40	50	60	70	80	90	100
senses	0.11	0.01	0.01	0.01	0.03	0.03	0.03	0.03	0.03	0.72
names	0.07	0.00	0.01	0.01	0.01	0.02	0.01	0.02	0.05	0.81
twins	0.14	0.00	0.00	0.00	0.02	0.02	0.01	0.01	0.02	0.78
burgers	0.02	0.01	0.03	0.08	0.04	0.06	0.06	0.15	0.04	0.51
pointers	0.09	0.00	0.00	0.00	0.01	0.01	0.01	0.02	0.01	0.84

Table 3.5: Proportions of bets within each game

The distribution of the bets is in each game about the same. Only in the game Burgers there is a different distribution, however this may have something to do with the fact that only 11 unique players played this game. In the other games much more different players placed bets, such that the distribution, given in the table, is more general.

The distribution of the bets leads to the question: Do all players behave the same? There is a small peak at the value 10 and a large peak at 100. Do most players have a same distribution for their bets? In that case, the variable bet would give a measure for the certainty of the player. However, one could feel that this is not completely true. Each player may have its own tactic or interpretation of the use of the slider. In figure 3.3, the mean bet for each player is illustrated. This shown clearly that each player uses the slider differently and that the distribution of the bets, see table 3.1.1 is mainly caused by the difference in use by the players. It can for example be seen that a relative large amount of players always use the value 10 as bet.

Agreement

The variable `agreement` gives a measure of agreement between the players on the question. Even though it will not be used in modelling, it gives a nice preview of the data.

The variable `agreement` is a variable that gives an idea about how the questions are answered. This may be interesting to get a better idea about the data. In figure 3.4, a histogram of the agreement per question is given. The agreement is a value between 0 and 1. A value of 1 corresponds to the situation where all players selected the same answer to the considered question, where a value of 0 corresponds to questions that have no , e.g. two distinct answers. From the histogram it follows that most questions have a high agreement (equal to 1), i.e. for most questions, the players are unanimous about the correct answer.

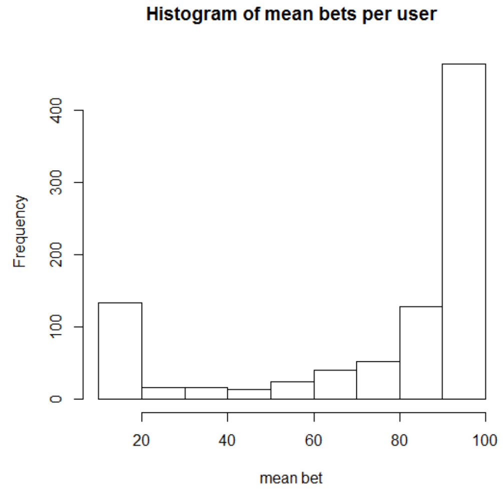

Figure 3.3: Histogram of the average bet per player

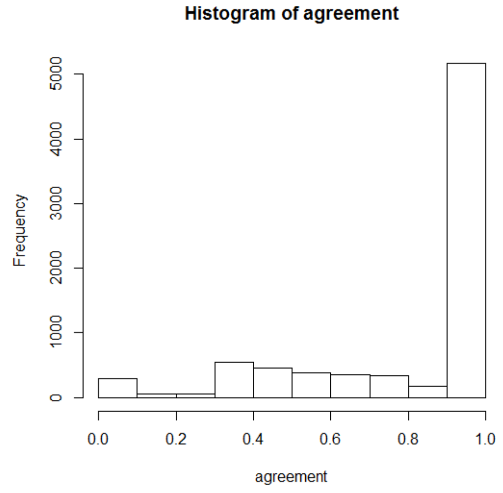

Figure 3.4: Histogram of the agreement between players per question

Other variables

There are a few other variables given in the data set. They will not be extensively observed, but some may be interesting for modeling the data. Other variables are in the data to give extra information.

bow Bit of wisdom. This variable gives a link to the data that is associated to the answer.

n.choices This variable contains a value, an integer between 2 and 7. It represents the number of choices the question had. It may be interesting to investigate the influence of this variable on the observations.

n.answers This variable represents the number of answers that the question received. It it based on the other data. It may be useful and important for computations, but it does not give new information.

relmaj The variable represents the highest number of concordant answers. A variable that is based on the other data. For now the variable will not be considered.

3.1.2 Problems with the data

Games

The data consists of data from several games. An adjustment that can be done is considering the data for each game apart. This has multiple advantages. Every game is different and also relations between variables may differ per game. Furthermore it avoids the problem considering the variable 'question'. Within the data sets for each game this variable represents a question, but in the complete data set it only represents a sentence that is considered.

Furthermore, it was earlier found that the data for the games Pointers and Burgers seems (for now) not useful. The data set for the game Pointers gives the strange result that (for the 1894 gold-standard records) each question is answered correctly, possibly since there is only one possible choice to the question. It may however be that the variable `n.choices` is not computed correctly. For the game Burgers, the number of answers per questions is very little. Questions are answer at most three times, this is not enough. There is also just one record for which the expert opinion is known.

Number of answers

The variable `n.answers` is not computed correctly. Consider for example the question

```
" "And I think they'll <b>want</b> a one-stop shop in terms of combining security,
immigration, customs, and quarantine togetherâ€¦ just to make sure it's more
streamlined and provides more certainty." "
```

The variable `n.answers` is computed with respect to the variable `question`. Therefore we find that for each record according to the question above, the variable `n.answer` equals 19. But looking into the details of the data, this value is not correct. The question is answered 13 times in the game 'twins' and 6 times in the game 'senses'.

Since a question can occur in multiple games, `n.answer` should be computed with respect to `question` and `game`. This can be done easily. Since an earlier suggestion was to separate the data for each game apart, the variable `n.answers` can be computed for each data set apart. Then the value will be correct.

Multiple correct answers

During the analysis of the data, a surprising observation was done.

Consider the game Names, and within the data for this game the question

```
A media rights group says <strong>Burma</strong>'s military-led government
has released two Burmese journalists working for a Japanese television station.<br/><br/><em>What
is <strong>Burma</strong> in this text?</em>
```

Within the subset of data for this question we find something disturbing. For different answers, the variable `expert.opinion` has value 1. This means that both answers would be correct.

The same kind of observation is done for several questions. Since it can only be determined for the gold-standard data, which is a relative small subset of the complete data set. It is assumed that multiple correct answers may occur more often.

It questions the reliability of the data. How is expert opinion computed? If it is really possible that there are multiple correct answers, then this might give problems. If the expert opinion is unknown, it is impossible to determine the correct answer. An answer that is selected most can correspond to the correct answer, but what about the second most selected answer? Since this phenomenon causes problems, the data from the game Names is omitted.

Time order

It is unknown whether the data is time ordered. It can be seen that the data is ordered per question. However, it is useful to know whether the records for each question are time ordered. It is assumed that the records are time ordered within each question. There is no other ordering observed. Furthermore this assumption can be done without any loss of generality. Note that only problems occur when the data is ordered by any variable that might affect answering a question correctly.

3.1.3 What can we learn from the data

The main goal of the linguistics researchers was to learn something from the data. To learn something about language. There are multiple subject that can be studied using the data set. Each game has of course a different interpretation, so most important question is whether the correct answer can be determined with use of the data. Of most interest is the number of data that is sufficient to determine this.

Some assumptions have to be done in order to determine the correct answer:

- The correct answer should be unique.
- The correct answer is defined as the most used answer (not in the sample, but total population)

The first assumption is done to prevent situations like in the data set of the game Names, where within one question, multiple questions are accepted as a correct answer. Based on the second assumption it seems possible to find the amount of data that is enough to be sufficiently certain about the correct answer. This assumption corresponds to the statement that *the most frequently used language is the correct language*. It seems impossible to find the correct answer while most people select an incorrect answer. Since for most questions, the players agree upon the correct answer, it should be possible to find a most selected answer.

Other points of interest may be the relation between the variables. Especially how variables like the bet that a player gives is related to the probability of answering the question correctly. The opposite can also be questioned: What is the expected bet, given that a player answers a question correctly or not.

When considering the effects on answering the correct answer, it may also be interesting to consider the effect of individual players and questions, described as skill and difficulty.

3.2 Data selection

In this section, some new variables will be computed. The theory about the certainty of binomial proportions is applied. By using this, a subset of data can be selected for which the correct answer is sufficiently certain. The resulting data set can be used for the Generalized Linear Mixed Models.

3.2.1 Majority decides

In the previous section the assumption that the most frequent language is the correct language was done. This implies that an extra variable can be computed. Consider a question

What is the correct answer?
A, B or C?

with a set of answers $\{x_1, x_2, \ldots, x_n\}$, so $x_i \in \{A, B, C\}$. Denote x as the most selected answer. Then the majority opinion can be defined as

$$y_i = \begin{cases} 1 & \text{if } x_i = x \\ 0 & \text{if } x_i \neq x \end{cases} \tag{3.1}$$

An advantage of this variable is that it can be computed for each question, where the expert opinion variable is only known for a small subset of the data. Note that the majority opinion y_i is sometimes unknown. Consider for example the situation where two answers are selected equally often. In that case one of these answered will be randomly assumed to be correct. Even though this is not very elegant, it does not really matter, since the effect will later on be described in terms of uncertainty. Also the situation where the question is only answered once. Then this single answer determines the majority opinion. This implies that the majority opinion is not directly related to the true correct answer. Also the uncertainty has to be considered. In these extreme examples, there is a high uncertainty.

In the **R** code, the majority opinion y_i is denoted by `majority.opinion`.

3.2.2 Certainty

As the majority opinion is not always a reliable variable, we want to describe the certainty of this variable that describes what the correct answer is. Recall the binomial proportion hypothesis testing from section 2.4. Consider as an example the question

What is the correct answer?
A, B or C?

Suppose that the data consists of a set of answers, $\{x_1, \ldots, x_n\}$, where $x_i \in A, B, C$. Then x is defined as the most frequently selected answer, such that $y_1, \ldots, y_n$ can be defined as the majority opinion. Take π as the true proportion of the population that selects answer x. Then this proportion can be estimated by

$$\hat{\pi} = \frac{\sum_{i=1}^{n} y_i + 2}{n+4} \tag{3.2}$$

As we want to know what the true answer is, this has to be tested. Testing whether x is the true answer, corresponds to testing the hypothesis $H_0 : \pi \leq \frac{1}{2}$. If this hypothesis can be rejected with a significance level α, it can be assumed with $(1-\alpha)\%$ confidence that x is the true answer (most used language is the true language). One way to do this is by the confidence interval, consider the one-sided confidence interval with upper bound $\pi = 1$ and lower bound

$$\hat{p} - z_\alpha \sqrt{\frac{\hat{p}(1-\hat{p})}{n+4}}$$

If the lower bound lies above $\frac{1}{2}$, the hypothesis is rejected with at least a $(1-\alpha)100\%$ confidence. So now the hypothesis can be tested for each question, it is however of interest how much data is needed. It can be tested at which point of time n^*, it was already possible to reject the null-hypothesis. In order to compute this, the assumption that $x_1, \ldots, x_n$ is time ordered needs to be done. This is not a problem, as we assume that the answer to a question does not depend on time. Observe that we can compute the lower bound of the confidence interval after each observation. Then n^* is defined as the minimum value in $i, \ldots, n$ for which the lower bound satisfies our demands. To illustrate this, we specify the example:

Suppose $n = 15$, and that the observations are $\{x_1, \ldots, x_n\} = \{A, B, B, B, B, A, B, B, B, B, B, B, B, B, B\}$. To test what the true answer is, with a significance level of 0.05, the following procedure arises:

For $k = 1$:

$$x_1 = A \quad \Rightarrow \quad x = A \quad \Rightarrow \quad y_1 = 1$$

This results in a lower bound $\pi \geq 0.24$.

For $k = 2$:

$$(x_1, x_2) = (A, B) \quad \Rightarrow \quad x = A \text{ or } x = B \quad \Rightarrow \quad (y_1, y_2) = (1, 0) \text{ or } (y_1, y_2) = (0, 1)$$

This results in both cases in a lower bound $\pi \geq 0.16$. Note that the hypothesis is always rejected, since only one answer can be the most selected.

For $k = 3$:

$$(x_1, x_2, x_3) = (A, B, B) \quad \Rightarrow \quad x = B \quad \Rightarrow \quad (y_1, y_2, y_3) = (0, 1, 1)$$

This results in a lower bound $\pi \geq 0.43$.

Et cetera.

At $k = 10$, the lower bound is $\pi \geq 0.52$. This is the first moment in time for which the null hypothesis can be rejected. Therefore, at this question $n^* = 10$. At $k = 15$, The lower bound is $\pi \geq 0.63$.

The one-sided confidence interval can be plotted too, to illustrate what is going on. See figure 3.5.

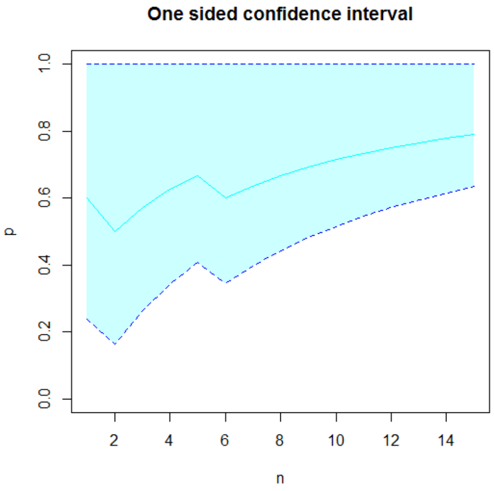

Figure 3.5: Transformation of the 95 % one-sided confidence interval as the number of observations increase

In the **R** code, included in appendix A, this procedure is done for each question.

3.2.3 New data set

The procedure from the previous section can be done for each question. For each question it can be determined how confident we are about the true answer. In order to model the probability that a question is answered correctly, it is necessary to know what the true answer is. The data provides only a small set of data for which this is known. But, considering the method of the previous section, it is possible to determine the true answer as there is enough data.

The one sided confidence interval is computed for each question. It is possible to select a subset of data. Take a 5 % significance level and test the hypothesis $H_0 : \pi \leq \frac{1}{2}$. Note that these values can be determined for a higher certainty, or a different hypothesis value, depending on what one finds acceptable. In the games Twins and Senses, it is shown for how much of the questions the true answer can be determined, see tables 3.2.3 and 3.2.3. It is also shown how much observations were needed to determine the true answer.

Twins

n^*	5	8	10	13	NA
# questions	1510	139	20	2	128

Table 3.6: Minimal number of observations needed per question in the game Twins

Twins

n^*	5	8	10	NA
# questions	283	3	1	2749

Table 3.7: Minimal number of observations needed per question in the game Senses

For the game Twins, the true answer can be determined for 1671 out of the 1799 unique questions. For the game Senses, there is still a lot of uncertainty. For this game, the true answer can be determined with a 95 % certainty for only 287 out of the 3036 unique questions. The fact that for the game Twins there is much more certainty corresponds to the fact that questions in this game are on average answered much more. This corresponds to the observations in 3.1.1, about the number of records per question.

Note that it is possible that, however there is at least 95% certainty at n^* observations, at this moment there is less than 95% certainty. This occurs at 10 questions for the game Twins, these are not included in the new data set, as the data set consists of the questions where there is at least 95% certainty at this moment.

The goodness of this method can also be illustrated. As a new set of data consider the questions for which the true answer is for at least 95 % certain. For the subset of data from Twins, the majority opinion can be compared to the expert opinion. In this subset, there are 45 questions for which the true answer is - according to the experts - is known. It turns out that for all these questions the majority opinion and the expert opinion are the same. It could be tested, as there is more data for which the answer can be estimated asymptotically, how good this method is. An important remark is that the wanted certainty of 95% does not result in an exact certainty of 95%. The sequential testing is a discrete process, such that the certainty has to be at least 95%, but in practice it results in a higher certainty, since the lower bound makes 'jumps'.

Note that the current method to test the hypothesis uses an asymptotic approach of the confidence interval. The fact that we use the Agresti-Coull interval ensures however that for small sample sizes, this method is also appropriate. It is also possible to use a Bayesian approach. The function `binom.test` in **R** gives an method to test the hypothesis from a Bayesian point of view. In that case we would consider a uniform prior. The results of this method are however about the same. For the 1799 questions from the game Twins, 66 questions have a different result. These differences do not really matter as the asymptotic approach is a little more skeptical; there are slightly more observations needed to find the true answer. In the **R** code in appendix A, it is possible to use this method, to compare the results.

3.3 GLMM for the Wordrobe data

In this section, the generalized linear mixed models from 2.2 will be applied. From the exploratory analysis and the data selection, it followed that only the game Twins is appropriate to model. For the game Senses, there is much data for which the true answer is uncertain. Therefore only the data set of Twins will be used to apply the generalized linear mixed models. From this data set, the new data set is extracted, for which the answer for each question is at least 95% certain.

3.3.1 Logistic model

Recall the logistic model. This can be applied to model the probability that a question is answered correctly. Define

$$y_{ij} = \begin{cases} 1 & \text{if player i answers question j correct} \\ 0 & \text{if player i answers question j false} \end{cases},$$

for $i = 1, \ldots, 835$ and $j = 1, \ldots, 1661$. So there are 835 unique players in the data set and 1661 unique questions. Note that y_{ij} does not exist for all i, j. An individual player did not answer all questions. Assume that these responses are drawings from a Bernoulli distribution, $y_{ij} \sim \text{Bern}(\pi_{ij})$. The parameter is described as following

$$\pi_{ij} = \Pr(y_{ij} = 1) = \Pr(\text{ player i answers question j correct })$$

These probabilities are fitted by a logistic model. Hence $\eta_{ij} = \text{logit}\pi_{ij}$. We want to find the best model that describes the linear predictors as a linear combination of the explanatory variables. Note that there are not a lot of variables. Bet, username and question are known, another variable that might have an influence is the number of choices. This is however the same for each question in this game Twins.

A model that gives a initial design for our model comes from the item response theory, De Boeck and Wilson (2004). Considering no explanatory variables, except for the question and username, the probability that a player i answers a question j is given by

$$\text{logit}\pi_{ij} = \alpha_0 + \alpha_i + \beta_j,$$

where α_0 corresponds to an intercept. The variable α_i corresponds to the effect of player i for $i = 1, \ldots, 835$. It is a numeric value that represents the skill of a player. It can be interpreted as the log-odds that player i answers an average question (for which the effect is 0) correctly. In a similar way the difficulty of question j is defined for $j = 1, \ldots, 1661$. Observe that a positive value of α_i corresponds to a high skill, i.e. a higher probability of answering correctly, where a negative value of β_j corresponds to a high difficulty, i.e. a lower probability of answering a question correctly.

It is assumed that skill and difficulty are independent random variables.

$$\alpha_i \sim \mathcal{N}(0, \sigma_\alpha^2) \ \forall i = 1, \ldots, 835,$$

and

$$\beta_j \sim \mathcal{N}(0, \sigma_\beta^2) \ \forall j = 1, \ldots, 1661.$$

Both random variables are independent and correspond to intercepts, given the username or question. These type of random effects are also called crossed random effects. This mean that the factors are not nested. When there occurs at least some crossing, nested effects cannot be used. A first approach for the player effects or question effect might be as a fixed effect, considering the question and username as factor levels. This model can however not be computed. Question and username are high-level factors. Random effects provide only one coefficient, that describes the effects by a distribution. The effects are relative to the average factor level. Fixed effects describe the effects within each factor level apart, with respect to one reference level.

As an extension of the item response model, the variable bet is also involved. Consider the model M_0, for which the linear predictors are defined as

$$\text{logit}\pi_{ij} = \alpha_0 + \alpha_i + \beta_j + \gamma b_{ij},$$

where b_{ij} is the bet that player i gave to question j. It is defined for each observation y_{ij}.

As the model is specified, it is of interest what the values of $\alpha_0, \gamma, \sigma_\alpha, \sigma_\beta$ are. These can be estimated by maximizing the likelihood. Note that the values α_i, β_j are dependent on $\sigma_\alpha, \sigma_\beta$ and that the likelihood is also independent on these values. Therefore, these values are not estimated from fitting the model. The expected values of the skill/difficulty can however be computed, given the observations.

There are 19973 observations in the data. The likelihood for the model M_0 is given by

$$\begin{aligned}\ell(\alpha_0,\gamma,\sigma_\alpha,\sigma_\beta|y) &= \log\prod_{i=1}^{19973} f(y_i|\sigma_\alpha,\sigma_\beta,\alpha_0,\gamma)\\ &= \sum_{i=1}^{19973}\log\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(y_i|\alpha_0,\gamma,\alpha,\beta)f(\alpha|\sigma_\alpha)f(\beta|\sigma_\beta)d\alpha d\beta,\end{aligned}$$

where

$$f(y_i|\alpha_0,\gamma,\alpha,\beta) = \pi_{b,\alpha_0,\gamma,\alpha,\beta}^{y_i}(1-\pi_{b,\alpha_0,\gamma,\alpha,\beta})^{1-y_i},$$

with

$$\pi_{b,\alpha_0,\gamma,\alpha,\beta} = \frac{\exp(\alpha_0+\gamma b+\alpha+\beta)}{1+\exp(\alpha_0+\gamma b+\alpha+\beta)}.$$

This log-likelihood can be computed by numerical methods. To determine the parameters $\alpha_0,\gamma,\sigma_\alpha,\sigma_\beta$, such that this likelihood is maximized, numerical methods are used, as mentioned in section 2.2.

In **R** the package `lme4` is used to fit the model:

```
> library(lme4)
> mt0<-glmer(majority.opinion ~ bet + (1|question) +
(1|username),family=binomial(link="logit"),data=datt)
> summary(mt0)
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial ( logit )
Formula: majority.opinion ~ bet + (1 | question) + (1 | username)
   Data: d

     AIC      BIC   logLik deviance df.resid
  3690.7   3722.3  -1841.4   3682.7    19969

Scaled residuals:
     Min       1Q   Median       3Q      Max
-10.3985   0.0423   0.0670   0.1110   1.4779

Random effects:
 Groups   Name        Variance Std.Dev.
 question (Intercept) 1.390    1.179
 username (Intercept) 3.853    1.963
Number of obs: 19973, groups: question, 1661; username, 835

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   4.4119     0.2427  18.179   <2e-16 ***
bet           1.5834     0.1851   8.555   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Correlation of Fixed Effects:
    (Intr)
bet -0.492
```

So we have $\hat{\alpha_0} = 4.41$, $\hat{\gamma} = 1.58$, $\hat{\sigma_\alpha} = 1.96$, $\hat{\sigma_\beta} = 1.179$.

For now we consider that this model, M_0, is the full model. For now, there are no other terms that can be involved. From this full model we are interested in the best model. From the summary that is given above, it is given that the the fixed effects are all significant. In the next section, the best model will be selected using deviance to test hypothesis, but also the Akaike Information criterion (AIC) will be discussed.

3.3.2 Model selection

Independently of the goodness of fit, it can be tested what the best model is. This can be done by hypothesis testing and comparison of the AIC values. The model M_0 as described in the previous section is defined as the full model, hypothesis testing is done to find nested models that fit the data better.

Consider the null-hypothesis $H_0 : \gamma = \gamma_1 = 0$. We want to test whether there is an effect of the bet. As alternative hypothesis we have $H_1 : \gamma = \gamma_0 \neq 0$. Recall the theory on model comparison by the deviance, 2.3. We want to compare two nested models, M_0 as defined earlier and M_1, for which

$$\text{logit}\pi_{ij} = \alpha_0 + \alpha_i + \beta_j.$$

```
> mt0<-glmer(majority.opinion ~ bet + (1|question) +
(1|username),family=binomial(link="logit"),data=datt)
> mt1<-glmer(majority.opinion ~ (1|question) + (1|username),family=binomial(link="logit"),data=datt)
> deviance(mt0)
[1] 3682.715
> deviance(mt1)
[1] 3750.16
```

$$\Delta D = D_1 - D_0 = 2[\ell(\alpha_0, \sigma_\alpha, \sigma_\beta, \gamma_1, y) - \ell(\alpha_0, \sigma_\alpha, \sigma_\beta, \gamma_0, y)] = 67.44 > 3.841 = \chi^2_{1,0.95}$$

```
> pchisq(deviance(mt1)-deviance(mt0),df=1,lower.tail=F)
[1] 2.167176e-16
```

Therefore, we can reject the null-hypothesis that the bets have no effect.

We can do the same for the other variables. We can neglect one of the random effects and compare this model with the full model. Define the following models:

$$M_2 : \text{logit}\pi_{ij} = \alpha_0 + \beta_j,$$

and

$$M_3 : \text{logit}\pi_{ij} = \alpha_0 + \alpha_i.$$

These models can be used to test the null-hypothesis $H_0 : \sigma_\alpha = 0$ and similarly $H_0 : \sigma_\beta = 0$.

Then the nested models can be compared by looking at the deviance. The differences of deviance are

$$D_2 - D_0 = 2[\ell(\alpha_0, \sigma_\alpha, \sigma_\beta, \gamma, y) - \ell(\alpha_0, 0, \sigma_\beta, \gamma, y)] = 423.33 > 3.841 = \chi^2_{1,0.95},$$

and

$$D_3 - D_0 = 2[\ell(\alpha_0, \sigma_\alpha, \sigma_\beta, \gamma, y) - \ell(\alpha_0, \sigma_\alpha, 0, \gamma, y)] = 79.2219 > 3.841 = \chi^2_{1,0.95}.$$

From this we can reject both null-hypothesis and conclude that both random effects are nonzero.

```
> mt2<-glmer(majority.opinion ~ bet + (1|question) ,family=binomial(link="logit"),data=datt)
> mt3<-glmer(majority.opinion ~ bet + (1|username) ,family=binomial(link="logit"),data=datt)
> pchisq(deviance(mt2)-deviance(mt0),df=1,lower.tail=F)
[1] 4.595747e-94
> pchisq(deviance(mt3)-deviance(mt0),df=1,lower.tail=F)
[1] 5.551125e-19
```

The model selection can also be done by comparing the AIC values. For a model, the AIC is a measure of quality of a model. It is defined as

$$\text{AIC} = 2k - 2\ell, \tag{3.3}$$

where k is the number of parameters in the model, ℓ is the maximized value of the log-likelihood, corresponding to the model. The model for which the AIC is minimum, is selected as the best model.

```
> AIC(mt0)
[1] 3690.715
> AIC(mt1)
[1] 3756.16
> AIC(mt2)
[1] 4112.047
> AIC(mt3)
[1] 3767.937
```

It turns out from this method that the full model is the best representation of the data, this corresponds to the results from comparison of the deviance.

3.3.3 Goodness of fit

Apart from selecting the best fitting model, it is of course of interest to test whether the model is a good representation of the data. Since the best model is selected in the previous section, the goodness of fit is tested in this section.

A way to test the goodness of fit is by use of the model's deviance. For the model M_0, the deviance is given by $D = 3683$. This goodness of fit statistic. The model is good as the deviance follow a chi-squared distribution. The degree of freedom is 19969.

$$D = 3683 < \chi^2_{19969,0.95} = 20303 \Rightarrow \text{The model fits the data well} \tag{3.4}$$

```
> deviance(mt0)
[1] 3682.715
> df.residual(mt0) # degrees of freedom
[1] 19969
> qchisq(0.95,df=df.residual(mt0))
[1] 20298.85
>
> pchisq(deviance(mt0),df=df.residual(mt0),lower.tail=F)
[1] 1
```

3.3.4 Interpretation

In the logistic model, the estimated coefficients can be interpreted very well. Since the model fits the log-odds, the coefficients are related to the (log-)odds. Recall that the model that was found is estimated as:

$$\text{logit}(\mu_{ij}) = 4.41 + 1.58 \cdot b_{ij} + \alpha_i + \beta_j$$

where skill α_i and difficulty β_j are normally distributed random variables, $\alpha_i \sim \mathcal{N}(0, 3.85)$ and $\beta_j \sim \mathcal{N}(0, 1.39)$.

Some interpretations follow from this model:

- If a player gives a 0.1 higher bet, it results in a $\exp(1.58 \cdot 0.1) = 1.17 \Rightarrow 17\%$ higher odds of having success. Note that this does not mean that giving a higher bet leads to success, but that the expected success is 17% higher as it is given that a player gave a 0.1 higher bet. (Assuming that all other variables are the same.)
- Similarly, the odds-ratio between a player that gives a maximum bet 1 and a player that gives a minimum bet 0.1 is $\exp(1.58 \cdot 0.9) = 4.16$. (Assuming that all other variables are the same.)
- If a player has an estimated skill of $\hat{\alpha_i}$, the odds ratio between this player and an average player is $\exp(\hat{\alpha_i})$.
- Define the best player as the player which has the highest value $\hat{\alpha_i}$. $\max\{\hat{\alpha_i}\} = 1.65)$. It means that the odds ratio between the best player and an average player is 5.19.
- If a question has an estimated difficulty of $\hat{\beta}_j$, the odds ratio between this question and an average question is $\exp(\hat{\beta}_j)$
- Define the most difficult question as the question which has the lowest value of $\hat{\beta}_j$. $\min\{\beta_j\} = -3.08$. it means that for an average question, the odds of having success is $\exp(3.08) = 21.86$ times higher than for the most difficult question.

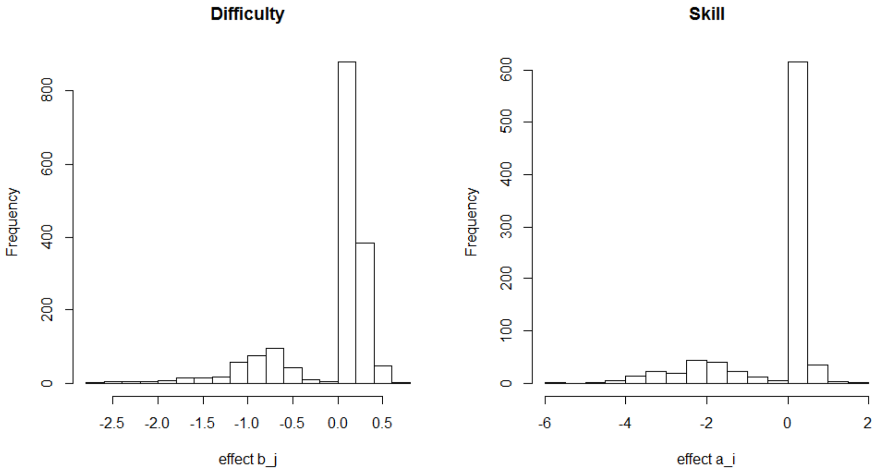

Figure 3.6: Histograms of the estimated values of effects within questions and players

Note that as the number of players increases, the maximum value may be estimated larger, as it is assumed that skill follows a normal distribution. Interpretation of this values can however become extreme. The minimum estimated value of skill is $\min\{\hat{\alpha}_i\} = -7.21$. It leads to the interpretation: The odds-ration between an average players and the worst player is $\exp(5.70) = 1352$. In figure 3.6 it is illustrated how the expected values of skill and difficulty are distributed. From this illustration it can be observed that the distribution does not seem to follow a normal distribution. It looks like there is a gap in the middle. This appearance might come from the fact that there was a data selection. A result of this selection might be that it mainly selected data for which effects were extreme. The gap may correspond to the removed data.

Another 'problem' that occurs has to do with the fact that each player uses the bet-slider differently. This was already observed in the exploratory analysis of the data. A result might be that players who always give a low bet results in a higher value of their 'skill'. Therefore the estimated value $\hat{\alpha_i}$ is not only related to the skill of a player, but may also depend on the use of the slider. It does however not mean that the effects skill and bet give a bad representation of the data. In fact, the variable skill takes into account that each player is different and therefore the effect of the bets is more accurate. Only point is that it does not necessarily mean that players with the highest values of $\hat{\alpha_i}$ are the best players.

Chapter 4

Conclusions

An exploratory analysis of the data showed that there is not a lot of data available. For the game Twins, the most data was available. For most games it is required to obtain more data in order to analyze it. The binomial proportion that selects the true answer was considered. For this approach the assumption is done that the correct answer is determined by the majority. Another interpretation of this assumption is that the majority selects the true answer. A way to select the true answer with enough certainty was by considering an asymptotic approach of the confidence interval. By this approach it turned out that the number of required data per question differs, as the agreement between the players also differs per question. For the game Twins the true answer could be selected for most questions. For other games the true answer was mostly uncertain. This corresponds to the findings in the exploratory analysis.

The game Twins is therefore the most appropriate to use for modeling generalized linear mixed models. As These models are interested in the first chapter of this thesis, they can be applied to the new data set: The data for which the true answer is at least 95% certain.

The results from applying GLMM on the selected data were discussed in the final chapter. A logistic model was defined in order to fit the probability that player i answers question j correctly. It is possible to describe the skill of individual players and difficulty of single questions by crossed random effects. Skill and difficulty are assumed to be values that are normally distributed. The skill and difficulty can be estimated for each player or question. The numerical values describe the effects on the log-odds. Another result is that the bet seems to have a statistically significant effect, even though the variable may be unreliable.

For further work on this subject, the goodness of the asymptotic approach for selecting the true answer may be tested. For this thesis the data set `wordrobe20140326.csv` was used. Just before the end of my project, a new data set was released. This data set, and other data set that will come in the future can lead to new results.

For fitting the generalized linear mixed models, it may be studied how the difference in use of the sliders can be modeled. From the exploratory analysis it followed that each player uses the slider differently. Currently it seems that the effect of skill compensates this differences, but it may be interesting to model the data such that the betting behavior of each player is described apart.

Another subject that may be considered is using the generalized linear mixed models to describe the certainty on knowing the true answer. Using the fitted values can lead to more accurate results. Consider for example that we received five equal answers on for one question. Then the current method concludes that there is at least 95% certainty. From the generalized linear mixed model there is however more information. What is the certainty is we know that the five answers are given by players with a lower skill, as they often give an incorrect answer. Or suppose that the player give a low bet, so they already describe that their certainty is low. This may be implemented in determining the certainty about the true answer. This may also avoid problems in situations where the majority is wrong about the true answer.

Appendix A

Results

A.1 Top ten players and questions

From the generalized linear mixed model, the following results are obtained:

```
> x<-ranef(mt0)$username
> rownames(x)[order(x[,1],decreasing=T)[1:10]]
 [1] "bf6031a8bd6fa5042f885205195a0273a55a570edb6f3cca71deb1ef"
 [2] "ef9c660f801270321e66aa402c21b7303d352fe2eda62e65747c6dd6"
 [3] "56957146195ed4164330c0e7f1914349c4f667991291073598080acd"
 [4] "bfbcd82fb345327d184e0c59f6c217cf79cdc838925c8610d0ac7954"
 [5] "c3782668f5d617d321bb04822cceca5e891145baae79283ef635f968"
 [6] "ac1681325807e4c96e5058e0609bc5f4c6cab07b3349e04ae2c75a31"
 [7] "6505f843ea25fef1f8efdd4297c597896709c960465edd3367bc16ae"
 [8] "29bc3aa82c8c54167d64dfb6da17bd1cc7d530d35ec1733fe7bda6e4"
 [9] "f077be668a381c0d824a1c266aa9160e1ee04e6366bbe71ef6ce9d30"
[10] "55aacde77dc32ab1051c81b75fe66ed8b8e3a075c2cf6920ad37e7d5"
```

These are the best 10 players. It can be computed what the proportion is for these player to answer a question correctly:

```
> X<-rownames(x)[order(x[,1],decreasing=T)[1:10]]
> for (i in X){
+ print(mean(subset(datt,username==i)$majority))
+ }
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
```

```
[1] 1
[1] 1
[1] 1
```

So the 'best' players answer all questions correctly. We can do a the same for the 'worst' players: players with least skill:

```
> rownames(x)[order(x[,1],decreasing=F)[1:10]]
 [1] "72c63fc685881f1ade0882383bb0795ff7decd55d853554b01d4e526"
 [2] "301d07b9d8d053c83ddf85659dc9c9d8624ae8e69e259fc5ac547908"
 [3] "42364c9556d2a57bae2682d1a62f810f8f18587a2fd6799fe3dded70"
 [4] "e999e4a3dd817fd6112c3ee518e480e376726e89b9517b1fa8bb0269"
 [5] "25468e5f8427736a187d3fa0b9f624b25fa17c7f145210b416f80ab2"
 [6] "9d0ab6d1ccefb1c46f4a1495848599534c1dfea883f9a84317a01035"
 [7] "ee2df30309463bdacd7ff0cae816ee6578494cedf903aab148401075"
 [8] "8e55559a6a6d61816dbc7d67d69db2413bf49dda0a607305ccf42976"
 [9] "c5fc55a7cf97cb02239051d479d05ae0331734049cf3ac7e555253da"
[10] "055669dfc49b7353088df1f7377f347a9856988164671ce161a98987"

> X<-rownames(x)[order(x[,1],decreasing=F)[1:10]]
> for (i in X){
+ print(mean(subset(datt,username==i)$majority))
+ }
[1] 0
[1] 0.2
[1] 0.7586207
[1] 0.5
[1] 0.375
[1] 0.8040541
[1] 0.8
[1] 0.7
[1] 0.7
[1] 0.8
```

Similarly the ten easiest questions:

```
> z<-ranef(mt0)$question
> rownames(z)[order(z[,1],decreasing=T)[1:10]]
[1] "The space probe began transmitting data to the Cassini spacecraft while landing on Saturn's large
 [2] "The secretary refused to <b>offer</b> a timetable, saying any withdrawal depends on Iraqi forces
 [3] "While other paramilitary groups in the nation have grown rich from the drug trade, the ELN <b>fu
 [4] "The <b>attacks</b> occurred after the government said it will go ahead with a reconciliation con
 [5] "Iran's Press TV says Tehran's Air Quality Control Company has found the amount of potentially ha
 [6] "In addition, beginning in 1973, he engaged in military operations in northern Chad's Aozou Strip
 [7] "Before now, strong overnight winds and dry conditions had been making it difficult to <b>keep</b
 [8] "Government authorities have declared Tuesday and Wednesday a public holiday, and Iranian media s
 [9] "Militant <b>groups</b> frequently attack oil operations in the Niger Delta to demand social serv
[10] "He also said inaction on the <b>issue</b> is not an option. "
```

For these questions the proportion that selected the true answer can be computed:

```
> Z <- rownames(z)[order(z[,1],decreasing=T)[1:10]]
> for (i in Z){
+ print(mean(subset(datt,question==i)$majority))
+ }
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
```

Which corresponds to the fact that it are easy questions. Difficult questions are:

```
> rownames(z)[order(z[,1],decreasing=F)[1:10]]
[1] "The United Nations says December's Indian Ocean tsunami caused around $520 million in damage to <
 [2] "The United Nations says December's Indian Ocean tsunami caused around $520 million in damage to
 [3] "The talks could come before six-party talks on North Korea's nuclear ambitions <b>resume</b>. "
 [4] "It said the loss was significant in a region where <b>fishing</b> provides a vital source of foo
 [5] "Israeli police say the bomber was spotted and prevented from <b>entering</b> the club. "
 [6] "They say police opened <b>fire</b> Monday on a car carrying the suspects outside the northwester
 [7] "Reporters Without Borders and the Burma Media Association welcomed the <b>release</b> of the rep
 [8] "The AP also reports Khayam has been on parole from prison since last year after serving half his
 [9] "However, a lack of foreign investment in the key sectors of <b>mining</b> and hydrocarbons and h
[10] "He says the <b>bombings</b> continue Friday. "
```

It can be checked that the proportion that selects the true answer is lower for these questions.

```
> Z<-rownames(z)[order(z[,1],decreasing=F)[1:10]]
> for (i in Z){
+ print(mean(subset(datt,question==i)$majority))
+ }
[1] 0.6956522
[1] 0.7692308
[1] 0.826087
[1] 0.8125
[1] 0.8095238
[1] 0.7857143
[1] 0.7692308
[1] 0.8235294
[1] 0.8333333
[1] 0.7692308
>
```

Note that the proportion of success is not the only variable that determines the value of skill/difficulty.

Appendix B

R code

B.1 Initial data set

```
dat<-read.csv("wordrobe20140326.csv",header=T,stringsAsFactors=FALSE)

senses<-subset(dat,game=="senses") # split data for each game
twins<-subset(dat,game=="twins")
names<-subset(dat,game=="names")
pointers<-subset(dat,game=="pointers")
burgers<-subset(dat,game=="burgers")


senses<-senses[,c(1,3,4,5,6,7,11)] # substract relevant variables
names<-names[,c(1,3,4,5,6,7,11)]
twins<-twins[,c(1,3,4,5,6,7,11)]
pointers<-pointers[,c(1,3,4,5,6,7,11)]
burgers<-burgers[,c(1,3,4,5,6,7,11)]

## change expert.opinion to numeric values: 0,1,NA

a<-(names$expert.opinion=="true")*1 - (names$expert.opinion=="unknown")*1
names$expert.opinion<-replace(a,a==-1,NA)

a<-(senses$expert.opinion=="true")*1 - (senses$expert.opinion=="unknown")*1
senses$expert.opinion<-replace(a,a==-1,NA)

a<-(twins$expert.opinion=="true")*1 - (twins$expert.opinion=="unknown")*1
twins$expert.opinion<-replace(a,a==-1,NA)

a<-(pointers$expert.opinion=="true")*1 - (pointers$expert.opinion=="unknown")*1
pointers$expert.opinion<-replace(a,a==-1,NA)
```

```
a<-(burgers$expert.opinion=="true")*1 - (burgers$expert.opinion=="unknown")*1
burgers$expert.opinion<-replace(a,a==-1,NA)

## n.answers

k=1
n.answers<-rep(NA,nrow(names))
for(i in names$question){
n.answers[k]<-nrow(subset(names,question==i))
k<-k+1
}
names<-cbind(names,n.answers)

k=1
n.answers<-rep(NA,nrow(twins))
for(i in twins$question){
n.answers[k]<-nrow(subset(twins,question==i))
k<-k+1
}
twins<-cbind(twins,n.answers)

k=1
n.answers<-rep(NA,nrow(senses))
for(i in senses$question){
n.answers[k]<-nrow(subset(senses,question==i))
k<-k+1
}
senses<-cbind(senses,n.answers)

k=1
n.answers<-rep(NA,nrow(pointers))
for(i in pointers$question){
n.answers[k]<-nrow(subset(pointers,question==i))
k<-k+1
}
pointers<-cbind(pointers,n.answers)

k=1
n.answers<-rep(NA,nrow(burgers))
for(i in burgers$question){
n.answers[k]<-nrow(subset(burgers,question==i))
k<-k+1
}
burgers<-cbind(burgers,n.answers)

## Majority opinion

mca<-NA
k=1
for (i in senses$question){
```

```
table<-with(subset(senses,question==i),table(answer))
mca[k]<-names(which.max(table))
k<-k+1
}
majority.opinion<-as.numeric(mca==senses$answer)
senses<-cbind(senses,majority.opinion)

mca<-NA
k=1
for (i in pointers$question){
table<-with(subset(pointers,question==i),table(answer))
mca[k]<-names(which.max(table))
k<-k+1
}
majority.opinion<-as.numeric(mca==pointers$answer)
pointers<-cbind(pointers,majority.opinion)

mca<-NA
k=1
for (i in burgers$question){
table<-with(subset(burgers,question==i),table(answer))
mca[k]<-names(which.max(table))
k<-k+1
}
majority.opinion<-as.numeric(mca==burgers$answer)
burgers<-cbind(burgers,majority.opinion)


mca<-NA
k=1
for (i in names$question){
table<-with(subset(names,question==i),table(answer))
mca[k]<-names(which.max(table))
k<-k+1
}
majority.opinion<-as.numeric(mca==names$answer)
names<-cbind(names,majority.opinion)

mca<-NA
k=1
for (i in twins$question){
table<-with(subset(twins,question==i),table(answer))
mca[k]<-names(which.max(table))
k<-k+1
}
majority.opinion<-as.numeric(mca==twins$answer)
twins<-cbind(twins,majority.opinion)

## Change bet (seems just better for convergence)
senses$bet<-as.numeric(0.01*senses$bet)
```

```
twins$bet<-as.numeric(0.01*twins$bet)
names$bet<-as.numeric(0.01*names$bet)
pointers$bet<-as.numeric(0.01*pointers$bet)
burgers$bet<-as.numeric(0.01*burgers$bet)

write.csv(file="names.csv", x=names)
write.csv(file="senses.csv", x=senses)
write.csv(file="twins.csv", x=twins)
write.csv(file="pointers.csv", x=pointers)
write.csv(file="burgers.csv", x=burgers)


names<-read.csv(file="names.csv", stringsAsFactors=FALSE)
senses<-read.csv(file="senses.csv", stringsAsFactors=FALSE)
twins<-read.csv(file="twins.csv", stringsAsFactors=FALSE)
pointers<-read.csv(file="pointers.csv", stringsAsFactors=FALSE)
burgers<-read.csv(file="burgers.csv", stringsAsFactors=FALSE)
```

B.2 Data selection

```
senses<-read.csv(file="senses.csv", stringsAsFactors=FALSE)
pointers<-read.csv(file="pointers.csv", stringsAsFactors=FALSE)
burgers<-read.csv(file="burgers.csv", stringsAsFactors=FALSE)
names<-read.csv(file="names.csv", stringsAsFactors=FALSE)
twins<-read.csv(file="twins.csv", stringsAsFactors=FALSE)


dat<-twins # equivalent names,senses,...

dat$true.ans<-dat$q.lower<-dat$ncrit<-NA

ncrit<-true.ans<-q.lower<-lower<-NA

for (i in 1:length(unique(dat$question))){

idat<-subset(dat,question==unique(dat$question)[i])
x<-p<-s<-ikdat<-ans<-NA

for (k in 1:nrow(idat)){

ikdat<-idat[1:k,]
ans[k]<-names(which.max(table(ikdat$answer)))
p[k]<-(sum(ikdat$ans==ans[k])+2)/(k+4)
s[k]<-sqrt(p[k]*(1-p[k])/(k+4-1)) # = se/sqrt(n)
}

lower<-p-1.644854*s     # 1.644854 = qnorm(0.95)

x<-(1:length(lower))*(lower>0.5)
ncrit[i]<-min(x[x>0])

true.ans[i]<-ans[length(ans)]

q.lower[i]<-lower[length(lower)]

dat[which(dat$question==unique(dat$question)[i]),]$ncrit<-ncrit[i]
dat[which(dat$question==unique(dat$question)[i]),]$true.ans<-true.ans[i]
dat[which(dat$question==unique(dat$question)[i]),]$q.lower<-q.lower[i]

}

##########################

table(ncrit)
solution.data<-cbind(unique(dat$question),true.ans)


write.csv(file="twins2.csv", x=subset(dat,q.lower>0.5)) # equivalent senses,names,...
```

B.3 Models

```
#Final models
setwd("C:/Users/Arjan/Desktop/BACHELOR PROJECT/")

datt<- read.csv("twins2.csv",header=T,stringsAsFactors=FALSE)
twins<- read.csv("twins.csv",header=T,stringsAsFactors=FALSE)

library(lme4)

table(ds$maj) # senses not enough 0's, no modelling

summary()

mt0<-glmer(majority.opinion ~ bet + (1|question) + (1|username),
family=binomial(link="logit"),data=datt)
mt1<-glmer(majority.opinion ~ (1|question) + (1|username),
family=binomial(link="logit"),data=datt)
mt2<-glmer(majority.opinion ~ bet + (1|question) ,
family=binomial(link="logit"),data=datt)
mt3<-glmer(majority.opinion ~ bet + (1|username) ,
family=binomial(link="logit"),data=datt)

summary(mt0)


# Random effects

a<-ranef(mt0)$question
b<-ranef(mt0)$user

par(mfrow=c(1,2))
hist(a[,1],main="Difficulty",xlab="b_j",breaks=20)
hist(b[,1],main="Skill",xlab="a_i",breaks=20)

#

plot(fitted.values(mt1),resid(mt1,type="pearson")) # plot(mt1)
plot(fitted.values(mt1),resid(mt1))
```

Bibliography

[1] Alan Agresti. *Categorical data analysis.* John Wiley & Sons, 2014.

[2] Alan Agresti and Brent A Coull. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126, 1998.

[3] Douglas Bates, C+ Eigen, and LinkingTo Rcpp. Package 'lme4', 2014.

[4] Lawrence D Brown, T Tony Cai, and Anirban DasGupta. Interval estimation for a binomial proportion. *Statistical Science*, pages 101–117, 2001.

[5] Paul De Boeck and Mark Wilson. *A framework for item response models.* Springer, 2004.

[6] Peter Diggle, Patrick Heagerty, Kung-Yee Liang, and Scott Zeger. *Analysis of longitudinal data.* Oxford University Press, 2002.

[7] Annette J Dobson. *An introduction to generalized linear models.* CRC press, 2001.

[8] Julian J Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models.* CRC press, 2005.

[9] R Michael Furr and Verne R Bacharach. *Psychometrics: an introduction.* Sage, 2013.

[10] Irwin Miller. *John E. Freund's Mathematical Statistics: With Applications.* Pearson Education India, 2004.

[11] Francis Tuerlinckx, Frank Rijmen, Geert Verbeke, and Paul Boeck. Statistical inference in generalized linear mixed models: A review. *British Journal of Mathematical and Statistical Psychology*, 59(2):225–255, 2006.

[12] Myron A Waclawiw and Kung-Yee Liang. Prediction of random effects in the generalized linear model. *Journal of the American Statistical Association*, 88(421):171–178, 1993.