RWTH AACHEN UNIVERSITY

Designing Interactive Bayesian Statistical Analysis

Master's Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Marty Pye

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Ulrik Schroeder

Registration date: 01/04/2015
Submission date: 04/09/2015

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, September 2015
Marty Pye

Contents

List of Figures

List of Tables

Abstract

Null Hypothesis Significance Testing (NHST) is the established method for data analysis in HCI, yet p-values, the main results of NHST, are widely misinterpreted as posterior probability. This leads to many research papers with issues that undermine the value or validity of the statistical testing. Bayesian analysis is an alternative to NHST that provides posterior probability in the results, which is more intuitive to interpret. However, performing a Bayesian analysis comprises complex subtasks, such as the setup of a prior probability, the creation of hierarchical models and assessing simulation quality. These subtasks can be overwhelming for the majority of scientists who are not extensively trained in statistics. Previous work shows that presenting NHST subtasks with a graphical user interface improves understanding. In this thesis, we developed an interaction design which helps guide the user through the process of a Bayesian analysis, and implemented a modular framework for a web-based application. Finally, we performed a qualitative evaluation of our software using the think-aloud method.

Acknowledgements

First and foremost, I would like to thank Chatchavan Wacharamanotham, M. Sc. This thesis would not have been possible without his valuable guidance and support.

Secondly, I would like to thank my supervisors Prof. Dr. Jan Borchers and Prof. Dr. Ulrik Schroeder.

Additionally, I would like to thank all my colleagues at the Media Computing Group for providing a productive and helpful environment to work in.

Last, but not least, special thanks goes to my family for always supporting me throughout my studies.

Thank you,

- Marty Pye

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Definitions of technical terms or short excursus are set off in coloured boxes.

EXCURSUS:
Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition:
Excursus

Source code and implementation symbols are written in typewriter-style text.

```
myClass
```

The whole thesis is written in British English.

Download links are set off in coloured boxes.

File: myFile[a]

[a] http://hci.rwth-aachen.de/public/folder/file_number.file

Chapter 1

Introduction

NHST has some limitations and pitfalls the experimenter has to take care of.

Statistical analyses are very prominent in scientific research, as they are used to validate hypotheses the authors make. Unfortunately, HCI is plagued with problems when it comes to the analysis and reporting of inferential statistics. Cairns [2007] found that from 41 research papers sampled from HCI journals, all but one had issues that undermined the value or validity of the statistical testing and therefore the research findings. This is often due to the lack of formal education of the experimenters and the vastness of statistical analysis. Attempts at mitigating this have been made by providing software tools which embed a lot of the knowledge usually required from the user. This can help novices avoid some of the common mistakes and pitfalls in inferential statistics.

Bayesian analysis is being promoted, but software is needed.

However, some of these pitfalls are a consequence of the framework we use for statistical analysis, so called **N**ull **H**ypothesis **S**ignificance **T**esting (NHST). NHST has been severely criticised, and some statisticians are pushing for the adoption of Bayesian analysis, as they deem it more suitable for the needs and objectives of scientific research. However, there is a distinct lack of visual tools for conducting Bayesian analyses. Just like for NHST, we need tools which embed as much of the required knowledge as possible, and guide the user through the analysis process. This thesis lays the groundwork for such a tool. We implemented various components of the Bayesian analysis pro-

cess, and propose an interaction design which we believe helps guide the user through the process of a Bayesian analysis.

In summary, the contributions of this thesis to the field of HCI are:

1. We propose a detailed interaction design for the interpretation and report of Bayesian analysis contrasts.
2. We implemented a highly modular, extendible framework built on top of JAGS[1] for a web-based Bayesian analysis application, with both front- and back-end components for each of the vital steps in an analysis.

[1] www.mcmc-jags.sourceforge.net

Chapter 2

Related Work

In this chapter, we give a brief overview of some problems with **N**ull **H**ypothesis **S**ignificance **T**esting (NHST), and how they are alleviated with a Bayesian approach. We then go on to list some reusable UI components from other systems, together with some guidelines on how certain concepts in statistics should be visualised. The following sections describe some of the problems we face with NHST.

2.1 Three problems with NHST

Motivation: NHST has limitations, we list 3 problems.

The established method in HCI for testing hypotheses is NHST. In this procedure, a null hypothesis is compared with an alternative hypothesis, and one of the two is rejected. However, NHST comes with some flaws, and some scientists claim that these flaws have such impact that we should reconsider the use of NHST. Kaptein and Robertson [2012] listed three problems with NHST, which can lead researchers to specifying weak hypotheses of limited scientific use.

2.1.1 Misinterpretation of the p-value

Problem 1: p-value is misinterpreted.

The p-value is defined as the probability of obtaining the

observed value of a sample statistic or a more extreme value if the data were generated from a null-hypothesis population sampled according to the intention of the experimenter (Kruschke [2010]). However, the meaning of the p-value is very often misinterpreted by researchers as the probability of the null hypothesis being true. So a p-value $< .05$ would signify that the probability of the null hypothesis being true is less than $.05$. If this definition of the p-value were true, it would be $P(H_0|D)$, which is the probability that the null hypothesis (H_0) is true, given the data (D) collected. In reality, the p-value is $P(D|H_0)$, the probability of the data, given that the null hypothesis were true.

2.1.2 Lack of Power

Problem 2: NHST often has a lack of power.

With NHST, both Type I and Type II errors can be controlled. Type I errors, or false positives occur when the null hypothesis is rejected when it is actually true. Simply stated, a type I error is detecting an effect which is not present. Type I errors can be controlled by specifying the alpha value, which specifies the value of p below which the null hypothesis will be rejected. Type II errors, or false negatives occur when the null hypothesis is accepted even though it is false. Again more simply stated, a type II error is failing to detect an effect that is present. The proportion of times the null hypothesis is false but accepted is called beta. The power of an experiment is the probability of detecting an effect given that the effect really exists in the population, and is $1 -$ beta.

p > 0.05 is not very informative

While p-values enable researchers to control type I errors, controlling type II errors, i.e. calculating the power of an experiment seems to be performed less often, as pointed out by Cohen [1992]. Experiments which lack power are not very informative when the null is not rejected, because it does not distinguish between the cases where the null hypothesis is actually true, and where the method just failed to reject the null hypothesis. Therefore Cohen [1992] gives some heuristics for the required sample size given the magnitude of effect the researcher wants to achieve.

2.1.3 Confusion between p-values and Estimates of Effects

Experimenters often neglect effect sizes.

A third and fairly severe problem of NHST is that researchers tend to give more value to the question of whether an effect exists, as opposed to how large this effect is, and to whom it matters. A p-value $< .05$ does not necessarily imply that the effect is important. In order to assess this, the researcher needs to take into account the magnitude of the effect. Depending on how large this is, he can decide whether the effect actually has a "significant" practical impact.

Standard effect sizes are hard to interpret and relate to the real world consequences.

In NHST, it is recommended that standard measures of effect size should be reported together with the p-values. Standardized effect sizes are suitable for comparing across different experiments. However, they do not estimate the actual differences in means or parameter estimates, and therefore can not really be used to assess theoretical and practical importance of the findings.

2.2 Alleviation with the Bayesian Approach

This section describes how the three problems listed above can be alleviated by using a Bayesian analysis approach.

2.2.1 Misinterpretation of the p-value

Bayes' Rule actually gives us what we misinterpret the p-value to be.

As stated above, the classical p-value $P(D|H_0)$ is often mistaken for $P(H_0|D)$. However, there is a relationship between a conditional probability and its inverse, and is established in Bayes' Rule (Bayes [1763]).

$$P(H_0|D) = \frac{P(D|H_0)P(H_0)}{P(D)} \tag{2.1}$$

According to (2.1), the probability of the null hypothesis being true given the data depends on the probability

of the data given the null hypothesis, $P(D|H_0)$, the prior probability of the null hypothesis, $P(H_0)$ and the probability of the data, $P(D)$. In practice, $P(D)$ is not really relevant. Usually the experimenter will want to compute $P(H_0|D)$ and $P(H_{alt}|D)$, where $P(D)$ will be the same on both sides of the equation, and can therefore be eliminated. As $P(D|H_0)$ is given by the p-value, the only remaining factor that needs to be specified in order to compute $P(H_0|D)$ is $P(H_0)$, which is the prior probability of the null hypothesis.

This Bayesian approach is gaining popularity in certain scientific fields (Kruschke [2010]). The Bayesian t-test actually computes what the researcher wants to know, the probability of a hypothesis being true, and which is already what researchers are misunderstanding the p-value to be.

2.2.2 Lack of Power

Bayesian analysis quantifies evidence in favour of the null, therefore giving information about the likelihood of a type II error.

As stated above, experiments with a lack of power are not very informative when they do not reject the null hypothesis, because we *cannot distinguish between the cases where the null is true and where the method just did not detect the null*. In contrast, this consequence of lack of power is less severe in a Bayesian analysis, as it is actually quantifying the probability that the null hypothesis is true. Therefore, evidence in favour of the null hypothesis can be only minor because the chances are good that the experiment will fail to detect an effect, or there can be strong evidence for the null hypothesis, implying that the null is actually true.

2.2.3 Confusion between p-values and Estimates of Effects

The value indicating presence of an effect also directly estimates the size of the effect.

As explained above, standardised effect sizes are not very suitable for judging the practical importance of the finding. In contrast, the difference estimates in a Bayesian analysis actually estimate the difference between means, measured in the dependent variable. Therefore, the value which in-

dicates that there is an effect also *directly* implies how large that effect is, in a unit of measurement which is relatable to the user.

Unlike Bayesian analysis, NHST also promotes week hypothesising.

Additionally to these three specific problems associated with NHST, we would like to conclude with a more general criticism of NHST, which Kaptein and Robertson [2012] point out. NHST encourages the proposition of weak hypotheses which make vague claims about the world. With the null hypothesis, the researcher predicts that there is no difference between conditions, and when this is rejected, he accepts the alternative hypothesis. Unfortunately, the alternative hypothesis that matches the null hypothesis, i.e. there is some difference, is fairly vague and not at all specific. It only rules out one case, the one where the means are exactly the same across all conditions. Any other relationship between the variables could be true, and therefore rejecting the null hypothesis is not really particularly informative.

Bayesian data analysis alleviates the problems stated above.

Bayesian analysis techniques can alleviate all the problems described by Kaptein and Robertson [2012], both the specific ones, as described above, and the general criticism about NHST encouraging the postulation of weak hypotheses, because the Bayesian methodology allows for comparing the credibility of multiple possible hypotheses. The limitations of NHST are driving the adoption of Bayesian methods, and in order to help this adoption, we feel an interactive, GUI-based tool for performing Bayesian analyses would be useful.

Existing software is complicated and requires much expertise.

There are some software tools available like SAS[1], SPSS Amos[2], JAGS[3] and BUGS[4] that let you perform Bayesian analyses. Unfortunately, all of these either require the user to create his analysis in code, or if they provide a UI, then they leave all the freedom of choosing the statistical models up to the user. While this is useful for expert users, novices are completely overwhelmed, and the danger of setting up an unsuitable model is pretty high.

[1] www.sas.com
[2] www-03.ibm.com/software/products/en/spss-amos
[3] www.mcmc-jags.sourceforge.net
[4] www.mrc-bsu.cam.ac.uk/software/bugs/

As we want to create a more limited but therefore more fail-safe system, similar to the original Statsplorer, we searched for reusable UI components from systems which let you perform Bayesian analyses.

2.3 Reusable UI components

We searched for reusable components from existing software.

In order to build an intuitive, visual tool for Bayesian analyses, we first looked at some tools which attempt to do this for the classical NHST approach. Visistat by Subramanian [2014], later known as Statsplorer by Wacharamanotham et al. [2015], automatically tests statistical assumptions and visually guides the user through the steps required to perform the analysis, and interpret the results. A formal study showed that Statsplorer helped novices understand statistical assumptions and choosing the appropriate tests. In fact, one of the points in the "Future Work" section listed the extension of the "Statsplorer back-end to support alternative statistical analysis procedures, e.g., Bayesian analysis" . While we were able to transfer some of the front-end solutions from Statsplorer to BayesianStatsplorer, the nature of the Bayesian process required reinventing several parts of the front-end too. In the following section, we list the solutions we could transfer to BayesianStatsplorer.

2.3.1 Visistat/Statsplorer

Reusing a three-column design.

The Statsplorer interface is divided into three separate columns, as can be seen in Figure 2.1. The flow throughout the analysis process is largely from left to right. First, the user selects the variables he wants to include in the analysis in the left panel. He can also choose between some different actions which he wants to perform, like compare the data, correlate the data, etc. Then he interacts with the centre column, checking the assumptions, interpreting statistical tests and understanding why specific tests where chosen. Finally, he can view the summary of the results in the concise report on the right, and can navigate through the history of tests he performed. At any point in time, the user

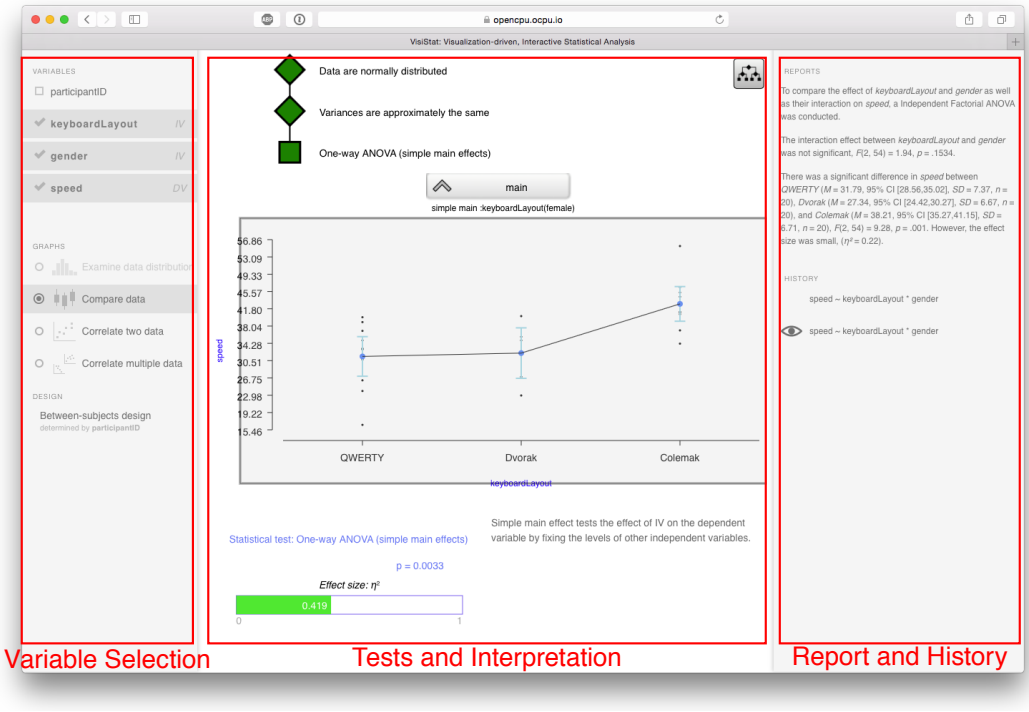

Figure 2.1: The Statsplorer interface (Subramanian [2014]) is divided into 3 columns. The left column handles variable selection, the centre column handles all the statistical procedures and interpretation, and the right column displays the report and history.

can go back to the previous column, to change something, or look at the next column to see what consequences his actions have in the further steps of the analysis process. This makes sense due to the inherently linear process of statistical analyses. However, this layout comes with some limitations, which are elaborated on in chapter 4.1 “Evaluation of Statplorer UI Components”.

The centre column provides the user with a descriptive plot of the data, as shown in Figure 2.2, once he has selected the variables of interest. In the boxplot, he can select the different means he wishes to compare in his test. Once he has selected them, he can choose “compare” to let Statsplorer choose the appropriate test and check the corresponding assumptions. We believe that the boxplot is a good starting point for the user to start his analysis, both for NHST and Bayesian analyses. The user can get a brief overview of the

The boxplot is a useful starting point for a Bayesian analysis too.

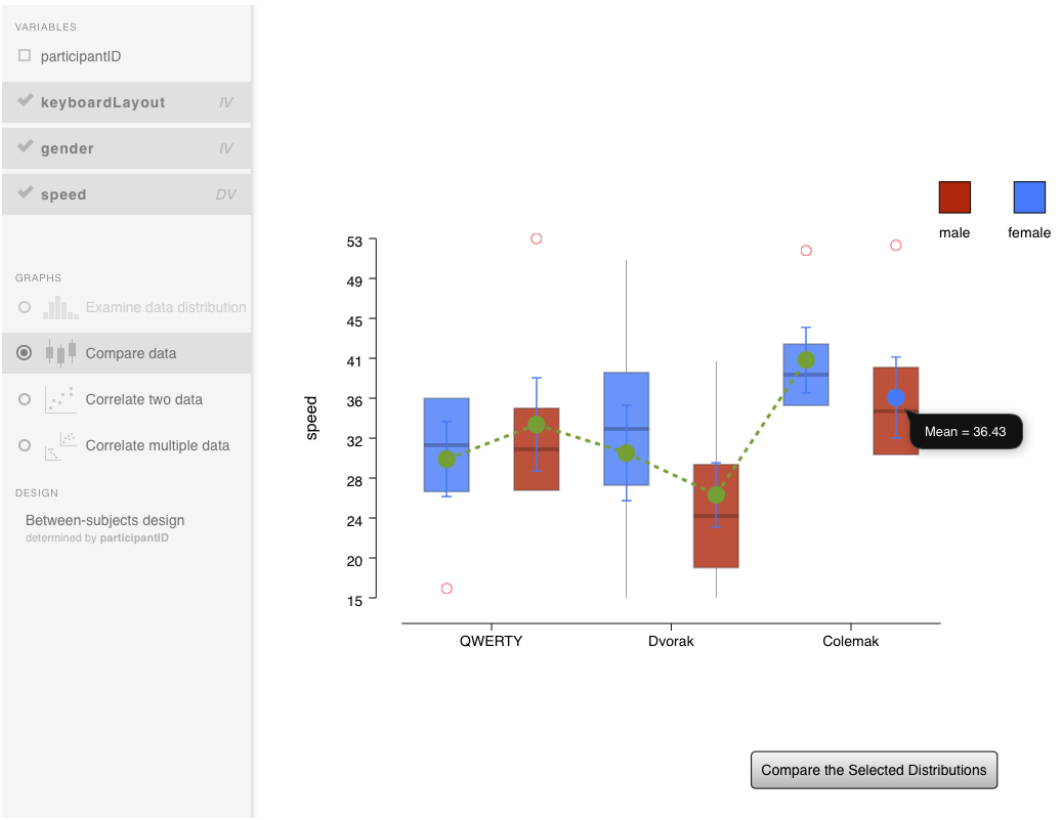

Figure 2.2: A boxplot of the data, as shown in the centre column of Statsplorer (Subramanian [2014]). The user can select specific means he wishes to compare.

trends in his data while selecting the means he wants to include in the comparison, and as this is a descriptive plot, it is not bound to either the classical or Bayesian approach.

The remaining components of Statsplorer needed to be redesigned for BayesianStatsplorer, the reasons for which are stated in chapter 4.1 “Evaluation of Statplorer UI Components”. Additionally to the UI components from Statsplorer, the following section elaborates on some other potential solutions for modules in BayesianStatsplorer that we found in existing literature.

2.3.2 B-Course

The visualisation B-Course uses for probabilistic dependencies could be reused in BayesianStatsplorer.

One task that BayesianStatsplorer will perform for the user is choosing a suitable hierarchical model (see 3.2 "Hierarchical Model". In the original Statsplorer, this is sort of analogous to choosing the correct test, which Statsplorer conveys to the user through an easy to interpret decision tree, shown in Figure 2.3. BayesianStatsplorer could include a similar decision tree, but for the information about the actual hierarchical model that was chosen by the system. For this, we believe a UI similar to the approach taken by Myllymäki et al. [2002] might be suitable. They developed a web-based tool B-Course, which visualises probabilistic dependencies, and lets the user interact with them. The software uses a tutorial style interface which combines the steps of the data analysis with additional support material. The probabilistic dependencies are visualised as in Figure 2.4. Unfortunately, the B-Course tool is not working any more, so we were not able to try out the features the authors describe in the paper. Basically, BayesianStatsplorer could use a similar visualisation of the dependencies of the different parameters of the hierarchical model. Furthermore, BayesianStatsplorer could allow the user to interact with this visualisation, tweaking the hierarchical model to suit his needs and expectations.

2.3.3 WinBUGS

WinBUGS provides helpful UI for model specification, but little UI for result interpretation.

Another Bayesian modelling framework is WinBUGS (Lunn et al. [2000]), which allows full probability models to be specified either textually with the BUGS[5] language or with a graphical interface called DoodleBUGS. Figure 2.5 and 2.6 show some screenshots of WinBUGS and DoodleBUGS. WinBUGS processes the specified model and performs an analysis with the aid of Markov chain Monte Carlo. The UI components of WinBUGS are suitable for the Bayesian modeling process, but WinBUGS lacks good UI for presenting the results of the analysis. It merely generates numbers and graphs. While the usability and learn-

[5] www.mrc-bsu.cam.ac.uk/software/bugs/

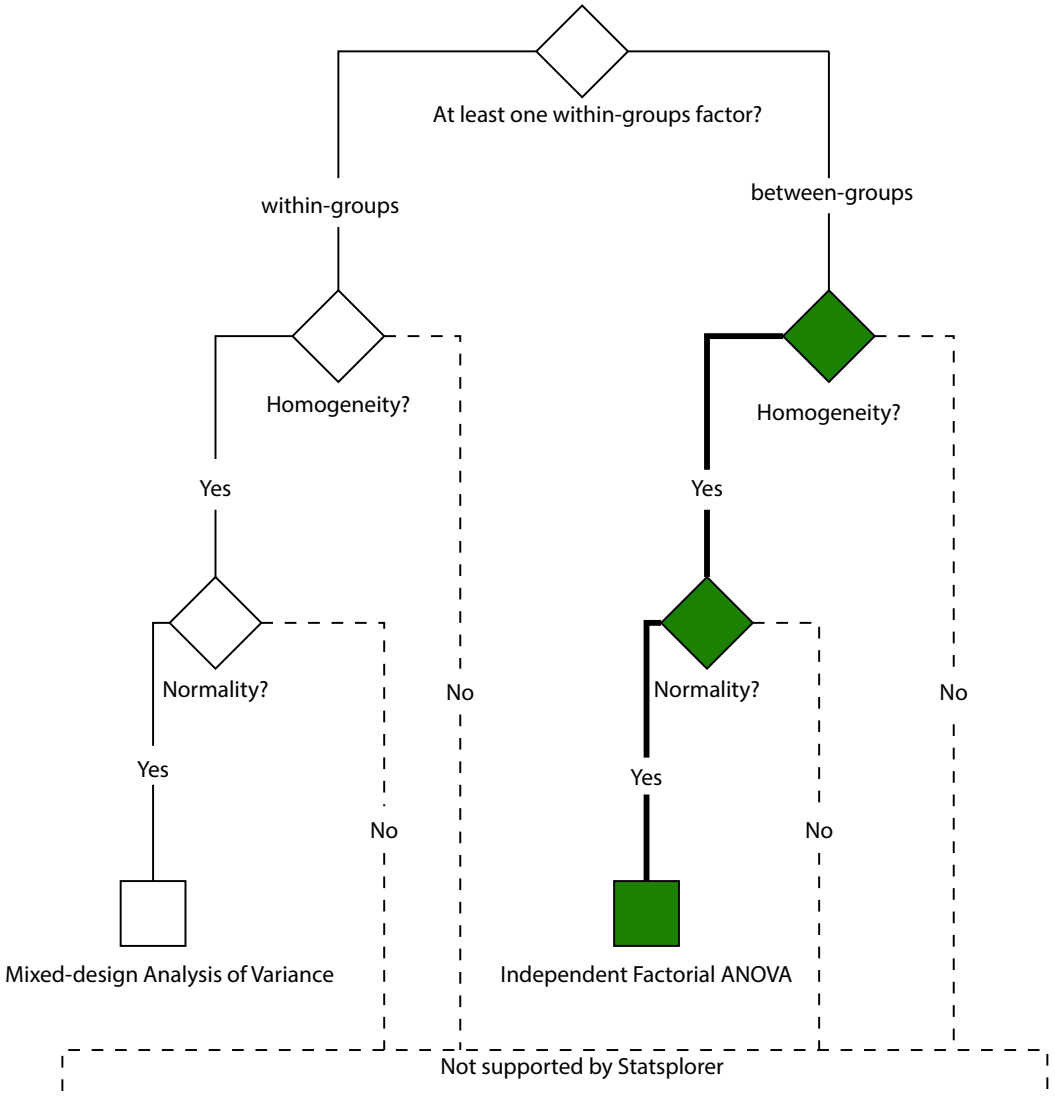

Figure 2.3: The decision tree shown to the user in Statsplorer (Subramanian [2014]). It allows the user to understand why a certain test was chosen. BayesianStatsplorer could include a similar tree structure for displaying why a certain hierarchical model is deemed suitable by the system.

ing curve of the model specification process can and needs to be improved, this is not the focus of this thesis. We focus less on enhancing the process of model specification, and more on a suitable UI for presenting the Bayesian results.

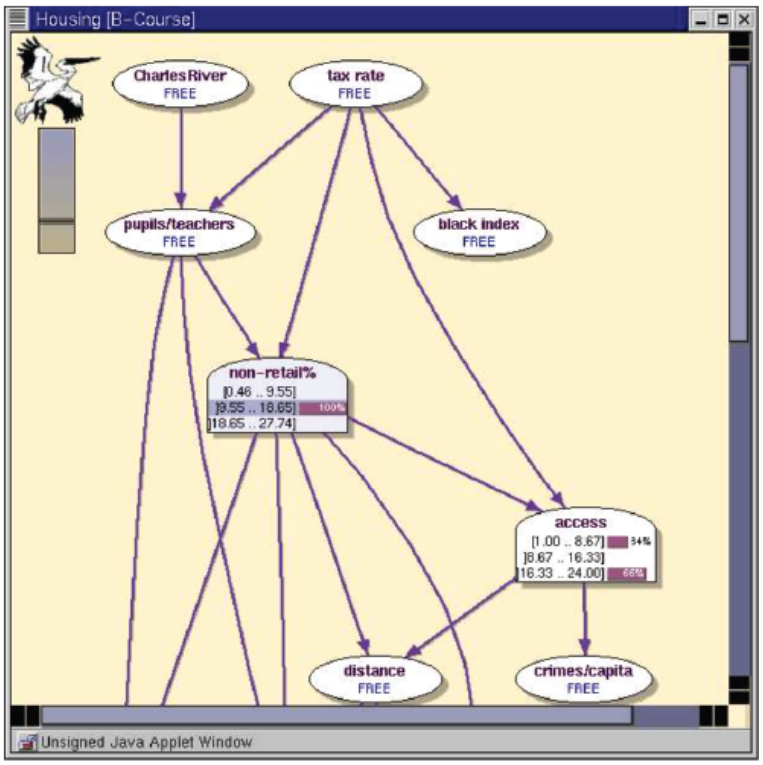

Figure 2.4: Visualisation of probabilistic dependencies as implemented in B-Course by Myllymäki et al. [2002].

2.4 Uncertainty Visualisation

Bayesian Analysis, just like NHST, needs to visualize uncertainty. We can apply guidelines about visualizing uncertainty to the Bayesian world too.

Most of the visualisations in BayesianStatsplorer have to show some kind of uncertainty, which is the highest density interval (HDI). Chapter 3.4 "Interpreting the Posterior" explains what exactly the HDI is. Correll and Gleicher [2014] worked on some alternative representations of uncertainty for classical statistics in order to alleviate some of the problems with error bars. Most of these problems do not apply to Bayesian statistics, however the authors composed a set of general guidelines about the visualisation of uncertainty. Most of these guidelines can also be applied to Bayesian statistics, and we assured the compliance of our design with them. The guidelines applicable to Bayesian statistics are listed below.

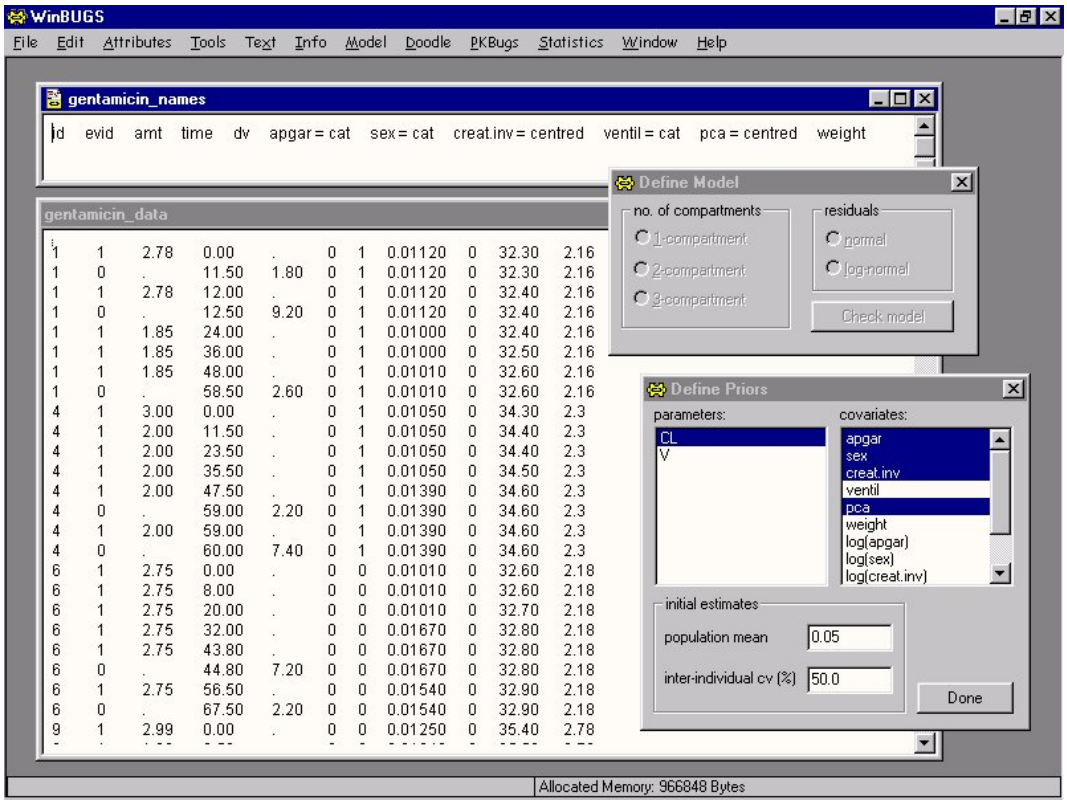

Figure 2.5: A screenshot of WinBUGS (Lunn et al. [2000]). The user can import some data, define a hierarchical model and create priors for different parameters through a simple GUI.

- The visual encoding should clearly communicate the effect size, i.e. the visualisation of the uncertainty should not come at the expense of the visualisation of the mean.
- The encoding should encourage the "correct behaviour" of the user, for example refraining from judgement if the means are dissimilar, but the uncertainty is very high.
- The encoding should promote the comparison or estimation of inferences that have not been explicitly supplied.
- The encoding should avoid binary encodings, which likely requires encodings which display certainty continuously.

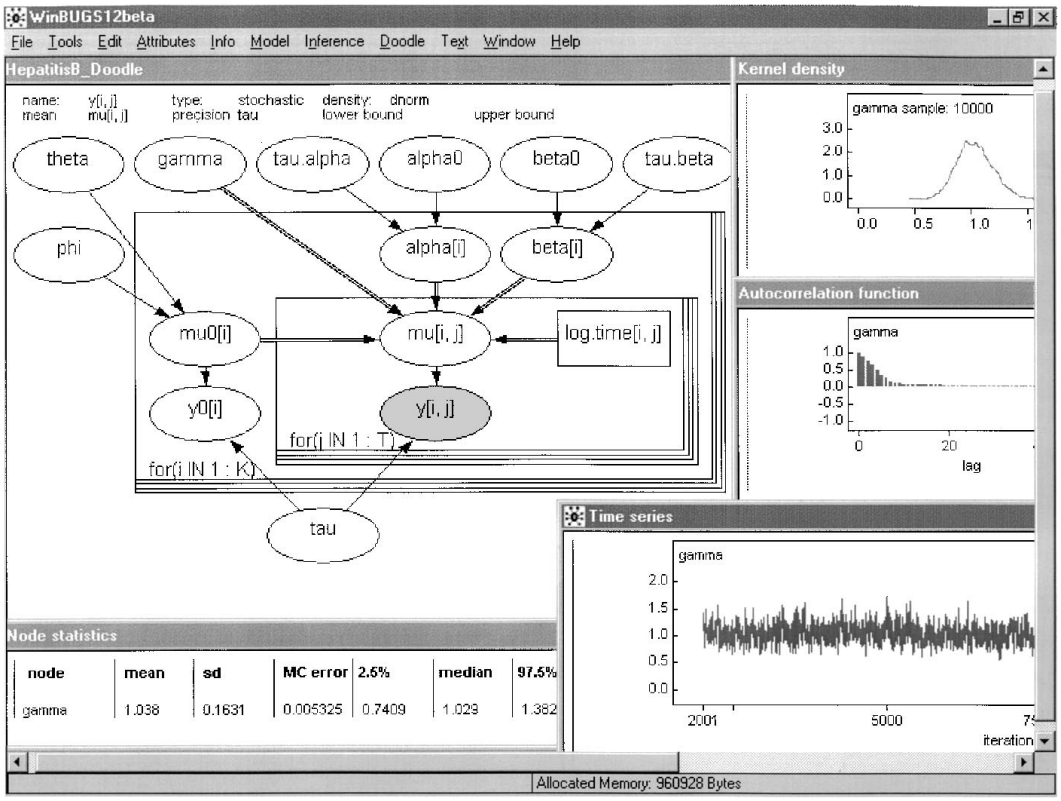

Figure 2.6: A screenshot of DoodleBUGS (Lunn et al. [2000]). The hierarchical model can be specified graphically in a similar manner to B-Course (Myllymäki et al. [2002]). Kruschke [2015] uses a similar visualisation (see Figure 3.2) to explain his hierarchical models.

In chapter 4.4 “Contrasts”, we discuss the design decisions of our contrast visualisations, the results of which are visible in Figure 4.8 and 4.11. These visualisations comply with the guidelines proposed by Correll and Gleicher [2014].

2.5 Reporting Bayesian Data Analyses

Guidelines for reporting Bayesian analyses exist.

Unfortunately, a standard for reporting Bayesian analyses does not exist yet. Therefore, we tried to pull together some guidelines from existing literature on what to report from a Bayesian analysis. Kruschke [2015] lists some essential points which should be in a Bayesian report.

- The report should *motivate the use of Bayesian, non-NHST analysis*. As many scientific audiences and reviewers are familiar with NHST, they appreciate an explanation as to why the experimenter used Bayesian analysis instead of NHST. For example, the experimenter can argue that Bayesian models are designed to be appropriate to the data structure, without having to make assumptions typical in NHST. The inferences from Bayesian analyses are more informative than NHST, as the posterior distribution reveals probabilities of combinations of parameter values.
- The report should *describe the data structure, the model and the model's parameters*. In his interpretation, the experimenter wants to interpret the meaningful parameter values, but for this, he needs to explain the model. And he can only explain the chosen model by explaining the data being modelled. Therefore, it makes sense for the experimenter to recapitulate the data structure, with the predicted and predictor variables. Then he can describe the model together with the meaningful parameters.
- The report should *clearly describe and justify the prior*. It is very important for the experimenter to convince the audience that a suitable prior was used, which did not predetermine the outcome of the experiment. A sceptical audience should be able to accept the used prior. The prior should be mildly informed by the scale of the gathered data, and if there is applicable existing research, it should not be ignored. Optionally, the experimenter can report the robustness of the posterior distribution with different priors.
- The report should *include the MCMC details*, especially evidence that the chains were converged and of sufficient length. It should indicate that the chains were checked for convergence, and indicate the ESS of the relevant parameters.
- The report should *interpret the posterior distribution*. As many models can have dozens or even hundreds of different parameters, it is important for the experimenter to summarise the important ones. Which

parameters to report is domain-specific and is influenced by the actual results themselves. The posterior central tendency of a parameter and its HDI can be reported in text alone, histograms of posteriors may be unnecessary in a concise report. If the model includes interactions of predictors, the lower order effects need to be interpreted carefully. If the experimenter uses a ROPE, he should justify its limits.

These essential points for reporting Bayesian analyses inform the questions in the report section of BayesianStatsplorer, as described in chapter 4.6 “Report”.

Chapter 3

Bayesian Analysis Theory and Workflow

"How often have I said to you that when you have eliminated the impossible, whatever remains, however improbable, must be the truth?"

—Sherlock Holmes

In this section we will introduce Bayesian data analysis, which relies on two fundamental concepts. The first concept is that Bayesian inference reallocates credibility across different possibilities. Certain possibilities can have different prior credibility distributions. Based on the information which is then gathered, the credibilities are reallocated to a new distribution of credibility, which is called the posterior credibility distribution. It is called the posterior distribution because it is what we believe *after* having gathered the new evidence. This posterior distribution can then be used as a prior for subsequent inferences.

Bayesian analysis relies on two fundamental concepts.

Although this first concept of Bayesian inference may sound quite complex, the mathematics are actually very similar to the logical reasoning humans perform on a day to day basis. Kruschke [2015] gives the following example.

Bayesian inference reallocates credibility across possibilities.

Definition: *Bayesian reasoning in everyday life*

BAYESIAN REASONING IN EVERYDAY LIFE:
Suppose you step outside one morning and notice that the pavement is wet, and wonder why. You consider all the possible causes of wetness, including possibilities such as recent rain, recent garden irrigation, a newly erupted underground spring, a broken pipe, a passerby spilled a drink, etc. If all you know until that point is that the pavement is wet, then all those possibilities will have some prior credibility based on previous knowledge. For example, recent rain may have greater prior probability than a spilled drink. You now continue to make more observations, you gather more data. If you observe that the pavement is wet for as far as you can see, and so are the trees, then you reallocate credibility to the hypothesis of recent rain. Inversely, if you observe that the wetness is localised to a small area, and there is an empty cup lying on the ground, then you would reallocate credibility to the hypothesis of a spilled drink, even though it had a lower prior probability.

This reallocation of credibility across possibilities is the essence of Bayesian inference. Another example is the quote at the beginning of this chapter. Sherlock Holmes conceived a set of possible causes for a crime. Each cause had a certain prior probability. When Holmes then gathered evidence which ruled out causes one-by-one, Bayesian reasoning forced him to conclude that the remaining possible cause was fully credible, even if it had a very low prior probability.

The possibilities are parameter values of mathematical formulas describing the data.

The second fundamental concept of Bayesian analysis is related to the nature of typical data in scientific research. Measurements are full of random variation and influences of external factors, therefore creating the need to use mathematical formulas which describe the trends and spreads of the data. These formulas have parameters like e.g. mean and standard deviation, and these parameters are critical to the second concept of Bayesian analysis. As mentioned above, the first concept of Bayesian data analysis is that it reallocates credibility across different possibilities. The second concept is that these possibilities are actually potential parameter values for the mathematical formulas describing

the trends of the data. It is important that the parameters in the mathematical model are meaningful. While theoretically any mathematical model which describes the data could be used to perform Bayesian analysis, it is not really productive to use mathematical models that we do not understand, with parameters we cannot interpret. The model also needs to be a suitable description of the data. If the model does not fit the data well enough, then any trends might not actually reflect the reality. Bayesian analysis can also be used to assess relative credibility of different candidate descriptions of the data, as described in Kruschke [2015], chapter 10.

The parameters must be meaningful and interpretable.

According to Kruschke [2015], Bayesian data analysis typically involves the following steps:

A Bayesian analysis typically involves 5 steps.

1. The data relevant to the research questions needs to be identified. What are the independent and dependent variables? What is the measurement scale of the variables?
2. A descriptive model needs to be defined. The parameter values of this model will be estimated in the Bayesian analysis process.
3. A prior distribution needs to be defined for the parameter values. This prior needs to be justifiable, and should be acceptable to a sceptical audience.
4. Bayesian inference is used to reallocate credibility across parameter values. The resulting posterior distribution needs to be interpreted based on the meaningful parameters. This assumes that the model was a reasonable description of the data, which is verified in the last step.
5. The posterior distribution needs to resemble the real data with reasonable accuracy. This is a posterior predictive check. Often this involves plotting a summary of predicted data from the descriptive model against the actual data.

In the following section, we'll expand the points above and describe each step in more detail, together with a descrip-

```
dependentVariable = list(name="DV", type="metric")

keyboardLayout = factor(list( name="IV1",
                              type="nominal",
                              ordered=FALSE,
                              level=c("Qwerty", "Colemak", "Dvorak",
                              labels=c("Qwerty", "Colemak", "Dvorak")))


gender = factor(list( name="IV1",
                      type="nominal",
                      ordered=FALSE,
                      levels=c("Male", "Female"),
                      labels=c("Male", "Female")))
```

Figure 3.1: Specification of data types in R. Both *keyboardLayout* and *gender* are nominal, with three and two levels respectively.

Most Bayesian analysis tools rely on R.

tion on how this is currently achieved. Current tools and frameworks to perform Bayesian statistical analyses rely heavily on R[1] scripts. Therefore in this chapter, we'll illustrate how one could perform an analysis only relying on R. As an example we use a simulated dataset from a hypothetical experiment which measured typing speed across different keyboard layouts, Qwerty, Colemak and Dvorak, and across different genders, male and female.

3.1 Data Identification

The experimenter needs to identify the type of the variables, which can be continuous or categorical.

The approach of identifying the data relevant to the research questions is similar to the approach in null hypothesis testing. Once the experimenter has created an experimental protocol (O'Brien and Wright [2002]), he should be able to identify the measures of the independent and dependent variables. A variable can be of continuous (metric) or categorical (nominal, ordinal or dichotomous) type. Having the types of all the variables is a prerequisite to continuing with step 2, and would look somewhat similar to Figure 3.1 in R.

[1] www.r-project.org

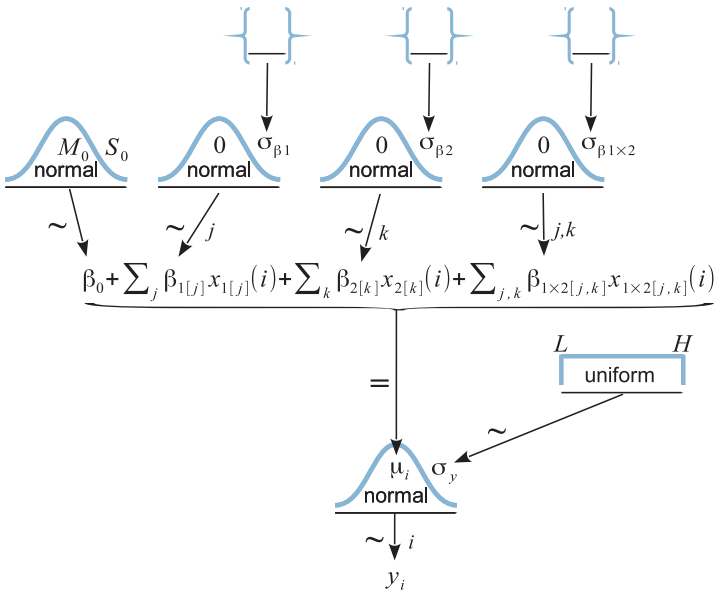

Figure 3.2: Hierarchical diagram that describes the data from two nominal predictors (Kruschke [2015]), in our example *keyboardLayout* and *gender*.

3.2 Hierarchical Model

A hierarchical statistical model needs to be specified.

The next step of the Bayesian analysis involves specifying a descriptive hierarchical model which represents the data. A hierarchical model for our typing speed example is shown in Figure 3.2. At the bottom, it is shown that the data is assumed to be normally distributed around the predicted value μ_i. Above we see that the predicted value is equal to the baseline together with the deflections of each predictor, *keyboardLayout* and *gender*, and their interaction. More information about this equation, given by the generalized linear model, can be found in Kruschke [2015], p. 429. Each parameter in the equation is given its own prior distribution, which is explained in 3.3 "Prior".

A hierarchical model can be specified with the help of the JAGS modelling language.

An experienced Bayesian analyst could specify the hierarchical model which suits his data using JAGS[2]. However, Kruschke [2015] has composed a set of R scripts suitable for

[2] www.mcmc-jags.sourceforge.net

common experimental designs. Depending on the amount and types of independent and dependent variables, the experimenter can choose from this set of scripts, which contain sample analyses. He can then modify the example to fit his dataset.

3.3 Prior

A prior distribution for the parameters needs to be provided.

Next, a prior distribution for the parameters needs to be defined. The prior might be able to be informed by previously conducted and accepted research. If no such prior knowledge exists or is applicable to the given scenario, a vague and non-committal prior can be placed on the parameters, which give equal prior credibility across a range of possible values. In Figure 3.2, you can see such a non-committal prior for the standard deviation parameter σ_y. The model assumes only one within-group standard deviation across all groups, i.e. the model assumes homogeniety of variance. The priors for the parameters can also be specified by the analyst in JAGS. Most of the R scripts available use non-informative priors as the default setting. Once the priors have been defined, the MCMC process can be run. When that is finished the experimenter has to check that the chains are of sufficient length and have converged. The R script generates a set of plots that help diagnosing the generated chains. More about the MCMC process is described in chapter 3.6 “MCMC”.

3.4 Interpreting the Posterior

The posterior distribution needs to be interpreted.

The next step is the actual interpretation of the posterior distribution. According to which contrasts the experimenter specified in the R script, several probability distributions are generated. A collection of all the generated contrasts for the typing speed example are shown in Figure 3.3. These contrasts are all so-called difference contrasts. The parameter being estimated is the difference in typing speed between different groups. Figure 3.4 shows the dif-

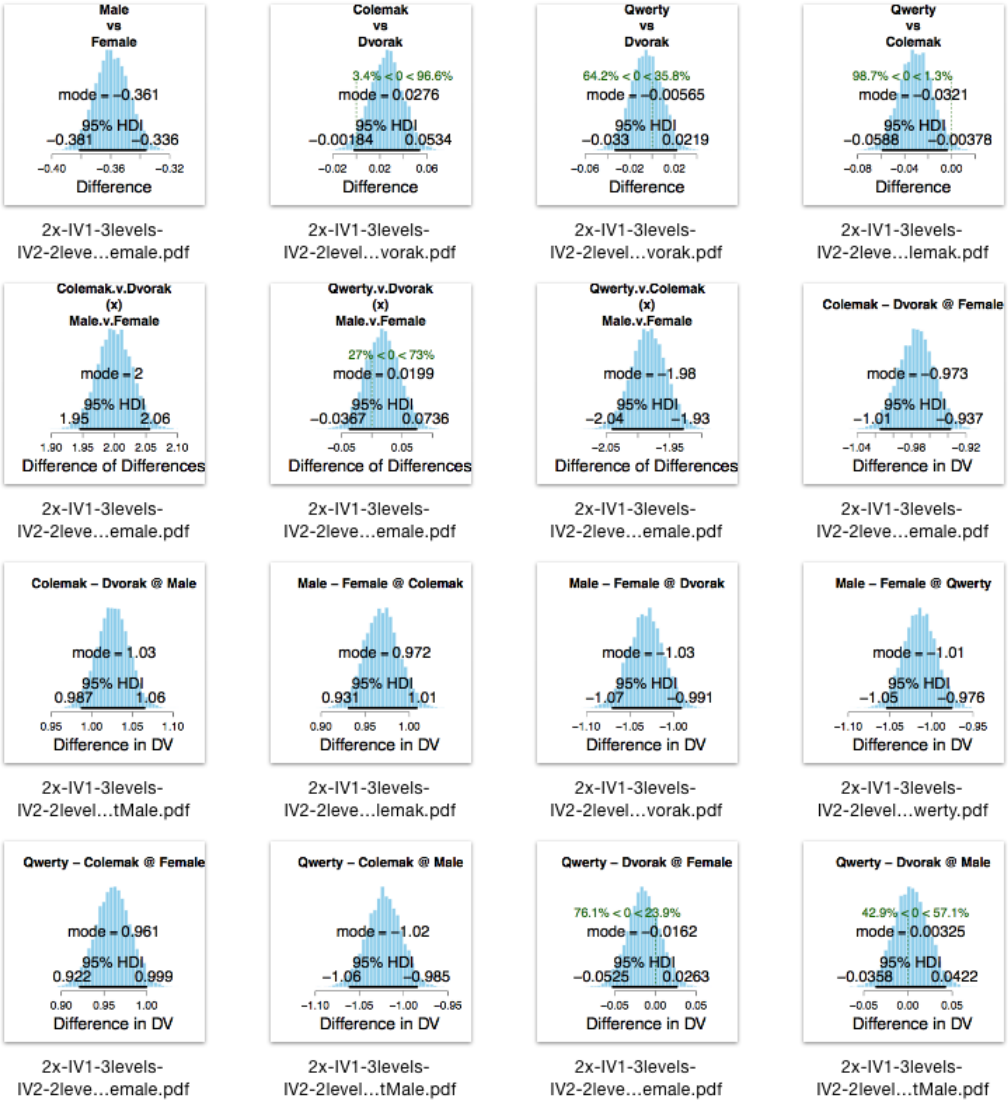

Figure 3.3: The Bayesian analysis generates many contrast plots, which the user needs to interpret. These were generated using the R script provided by Kruschke [2015].

ference between Colemak and Dvorak. We can see that the most credible value for the difference is 0.028. It also shows the level of certainty in that estimation, by showing the **H**ighest **D**ensity **I**nterval (HDI). The 95% HDI spans the values which cover 95% of the distribution. Here, the HDI ranges from -0.002 to 0.053. The HDI can also be used to

The 95% HDI spans the most credible parameter values.

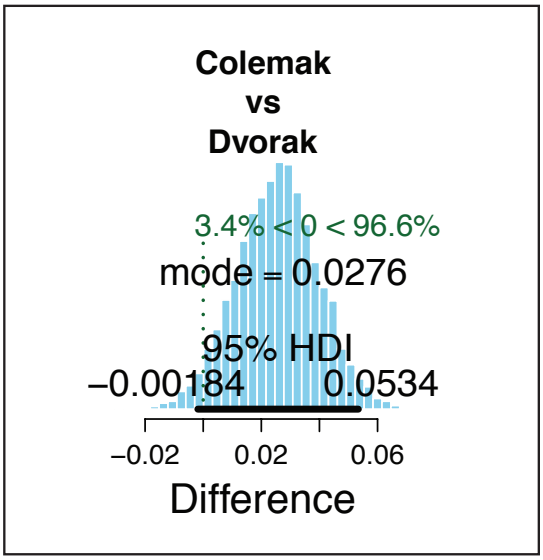

Figure 3.4: Difference in typing speed between the keyboard layouts Colemak and Dvorak. As zero is included in the 95% HDI, we can credibly say that there is no difference between the two.

make discrete decisions. In Figure 3.4, we can for example see that 0 is included in the HDI. Therefore we can credibly say that there is no difference between the two keyboard layouts. In summary, these are generally the questions the experimenter asks for each contrast:

The experimenter is usually interested in several questions.

- What is the mean estimate of the difference between the two groups? In Figure 3.4, this is shown as the mode.
- How precise is the estimate of the mean difference? This can be assessed by looking at the limits of the HDI and calculating its width. The narrower it is, the more certain the experimenter can be about the difference estimate.
- Is the difference between the two groups zero or close to zero? In order to assess this, the plot in Figure 3.4 shows zero, together with the probability that the true

difference is larger than zero, or smaller than zero.

The experimenter has to interpret several contrasts for interaction effects and main effects.

In practice the experimenter has to examine all the distributions in Figure 3.3 in order to understand the effects present in the dataset. Usually, he would start by interpreting the interaction effect plots. For our typing speed example, these are the first three plots in the second row of Figure 3.3. One can identify that there is no interaction effect between (Qwerty, Dvorak) and (male, female), but there is an interaction effect between the others, because 0 is not in the HDI. Therefore the experimenter should not just interpret the main effect of Qwerty vs. Colemak and Colemak vs. Dvorak, but also interpret the simple effects, i.e. The difference between Qwerty and Colemak for males, and the difference between Qwerty and Colemak for females. The same thing applies for Colemak vs. Dvorak. However, for Qwerty vs. Dvorak, the experimenter can directly interpret the main effect contrast, because there is no interaction effect between those levels of the independent variable. Chapter 4.4 "Contrasts" describes how our design facilitates the correct interpretation behaviour by the experimenter.

3.5 Posterior Predictive Check

In the posterior predictive check, the predicted data is compared with the actual data.

The final step involves checking whether the chosen model with the estimated parameter values fits the data reasonably well. This is the so called "Posterior Predictive Check". In this step, the predicted data from the model is plotted against the actual data. This plot can then be visually inspected to determine whether the model describes the data well. If it does not, the experimenter can consider alternative descriptive models. For example, if the data appears to have outliers, the experimenter could choose a heavy-tailed distribution instead of a normal distribution.

Now that the five steps of a Bayesian analysis have been discussed, the following section describes the sampling process which is used to generate the posterior distribution.

3.6 MCMC

MCMC is a procedure for producing an approximation of the posterior distribution.

Details of the **M**arkov **C**hain **M**onte **C**arlo (MCMC) method are described in Chapter 7 of Kruschke's book. MCMC is the procedure used for producing accurate approximations of posterior distributions. For every parameter value in the defined model, such as the mean, the MCMC algorithm generates a random walk in the space of possible parameters. The MCMC procedure guarantees that, in appropriate randomization conditions, the shape of the frequency distribution approximates the shape of the unknown probability distribution. The MCMC method can be used to approximate the shape of posterior distributions only from the likelihood and the prior, based on ratios of relative probability. The generated points from the random walk, altogether, constitute a frequency distribution. This set of points can be used to calculate the central tendency of the posterior, together with the HDI. The mathematics of the MCMC procedure is out of the scope of this thesis. The experimenter performing a Bayesian analysis should not have to know the details of the MCMC procedure itself. In theory, the mathematics of MCMC guarantee that infinitely long random walks, also called chains, will represent the posterior distribution perfectly. In practice, the experimenter must check the quality of the generated chains. There a two main criteria which should be fulfilled:

Two criteria need to be fulfilled.

3.6.1 Representativeness

The generated values should be representative. This can be checked visually or numerically.

The values in the generated chain should be representative of the posterior distribution. The arbitrarily chosen initial value for the random walk should not skew the values in the chain, and all the values in the posterior distribution should be sufficiently explored. Checking for representativeness can be performed by visually checking the chains trajectory, or by some numerical metric of convergence. A sample plot of a random walk trajectory is shown in Figure 3.5. On the left side, the chains have not converged for the first few hundred steps. In real-world analyses, the first few steps are usually excluded. This is the so called burn-in

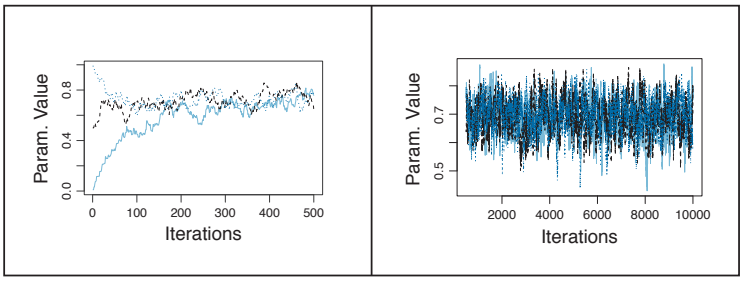

Figure 3.5: Random walk trajectories of a parameter. On the left, the trajectories have not converged, especially in the first 100 steps. On the right, the chains are nicely converged. (Adapted from Kruschke [2015])

period. On the right, the chains are nicely converged. As a rule of thumb, the trajectory plot should look like a "hairy caterpillar". Convergence can also be checked with a numerical metric, the so called Gelman-Rubin statistic (Gelman and Rubin [1992]) or "shrink factor". If the value is 1.0, the chains are fully converged, if the value is larger, then the chains are orphaned or stuck in an unrepresentative parameter space. If the shrink factor is larger than 1.1, the experimenter should investigate whether the chains have converged sufficiently. If they haven't, it can help to perform more iterations in the MCMC procedure.

A numerical metric to check for representativeness is called the "shrink factor".

3.6.2 Accuracy

The generated chain should be large enough so that the estimates of the parameter values are accurate and stable. The central tendencies and limits of the HDI should not change significantly if the MCMC process is run again, even a different initial value is chosen for the random walk. However, just using chain length as a metric does not suffice. If the chains are clumpy, the random walk got stuck in a small part of the parameter space, resulting in overrepresentation of those values. These clumpy regions of the chain do not provide independent information, and therefore need to be longer. Clumpiness is difficult to detect visually, but it can be measured with autocorrelation. In

The generated estimations should be accurate and stable. Inaccuracy can be detected by highly autocorrelated chains, and compensated by using more steps in the simulation.

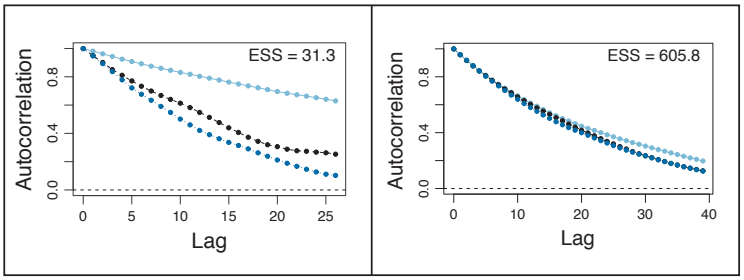

Figure 3.6: Diagnostic plot for autocorrelation. On the left, the chains are highly autocorrelated for large lags. This indicates that the chains may need to be longer. On the right, autocorrelation is low for large lags, indicating that the chains contain independent information. (Adapted from Kruschke [2015])

The numerical metric for accuracy is ESS.

Figure 3.6, you can see a diagnostic plot of autocorrelation. If the values are well above 0 for large lags, then this indicates that the chains are highly autocorrelated, therefore may need to be longer in order to provide sufficient information about the full posterior distribution. Inversely, if the autocorrelation for large lags is close to 0, then the chains are not autocorrelated. The numerical metric for checking sufficient sample size is called the "**E**ffective **S**ample **S**ize" (ESS), which is just the actual sample size divided by the amount of autocorrelation. How large the ESS has to be depends on which details of the posterior distribution the experimenter is interested in. If it is mainly the central tendencies, then the ESS does not have to be as large as when he's interested in the limits of the 95% HDI, which are visited less by the random walk. Kruschke recommends an ESS of $10\,000$ or larger, however he states that this number is only based on experience with practical applications. Depending on the required accuracy of the HDI limits, the necessary ESS may be less.

Chapter 4

Interaction Design

Chapter 3 "Bayesian Analysis Theory and Workflow" described the necessary steps to perform a Bayesian analysis. Next we would like to illustrate how we envision BayesianStatsplorer to aid in the individual stages of the Bayesian analysis process. In this thesis, we specifically focus on how the interaction design can help facilitate the interpretation and reporting of the Bayesian analysis. Whenever an example is required, we use our exemplary dataset containing typing speed measured depending on gender (male/female) and keyboard layout (Qwerty/Colemak/Dvorak).

Before elaborating on the design of each component in our BayesianStatsplorer, we'll briefly evaluate the reusable UI components of the original VisiStat/Statsplorer.

4.1 Evaluation of Statplorer UI Components

As mentioned in 2.3.1 "Visistat/Statsplorer", Statsplorer uses a three-column layout. Below is a list of the pros and cons of such a layout.

Advantages and Disadvantages of a three-column layout

✓ The left-to-right layout allows for the user to intu-

itively start with the left most actions, and then transition into the next operations in a natural order.

✓ At any point in time, the user can check any of the three columns for information he needs, without having to open or close any tabs, or having to navigate any menus.

✗ The layout does not scale well to smaller screens. The centre column becomes too small to fit all the required information and visualisations.

✗ There is no way for the user to hide the left and right columns if he is currently not interested in them.

Each column has a different purpose. The left one is basically used for variable selection and is therefore trivially applicable to BayesianStatsplorer.

Advantages and Disadvantages of the results layout

The centre column displays interaction, main and simple main effects, all of which need to be visualised in BayesianStatsplorer as well. The pros and cons of how these effects are laid out in Statsplorer are listed below:

✓ The entry view shows the main effects in the centre, with options to investigate interaction and simple main effects. This lets the user jump right in, interpreting the main effects first, which usually reflect at least some of his research questions.

✗ Showing the main effects first could falsely lead the user to interpret all the main effects before taking the interaction effects into account. There is a note at the bottom indicating that the user should interpret the interaction effects first, but it is not immediately apparent.

✗ The user is told to check the interaction effects, even if there are none.

✗ When going down to investigate the simple main effects, the user has to select which variable to fix, and at what level, and which one to vary. This becomes kind of tedious, and hard to remember which combinations have already been visited.

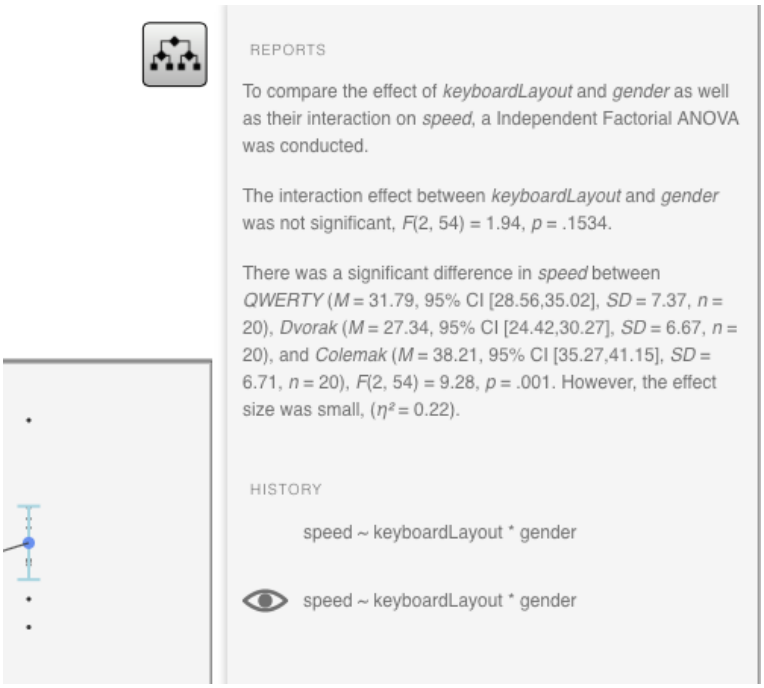

Figure 4.1: The right column contains the statistical report and the history of tests the user performed. The report provides a guideline for which information should be included in scientific publications. The history lets the user revisit the different tests that were performed.

✗ This layout does not scale well for more variables with more levels. The main effect view is then too small to contain all the necessary plots.

Later in section 4.4 "Contrasts", we demonstrate how we solved the above issues with BayesianStatsplorer. The design choices could also largely be applied to NHST, so future versions of Statsplorer could adopt some of the more suitable design solutions.

Advantages and Disadvantages of putting the statistical report in the right column.

Finally, we analyse the right column, where the user can see his statistical report, and the history of the tests he performed, as shown in Figure 4.1. The report shows him which information he needs to include in potential scientific publications, while the history lets him revisit different tests he performed. The pros and cons of this design are

listed below.

- ✓ The concise reporting style makes it very easy for the user to copy and paste that information into his own document.
- ✓ The report is driven by statistical reporting standards [2012], helping the user's confidence in the format of his report.
- ✗ While useful for copy-paste, the report text block does not contain information about which question each paragraph answers.

Chapter 2.5 "Reporting Bayesian Data Analyses" discussed what content should go into a Bayesian analysis report, and later in section 4.6 "Report", we present our design for the report module, which not only allows easy exportation, but also clarifies which parts of the report answer which questions.

The limitations of the original Statsplorer drove the design of BayesianStatsplorer.

Bearing in mind the limitations of the reusable UI components of Statsplorer, we designed the individual components of BayesianStatsplorer. The modules went through several design iterations, the final one is provided in appendix A "BayesianStatsplorer Mockup". The following sections describe some of the rationale behind the design decisions made for each component.

4.2 Dataset Selection

The dataset selection UI could be adopted from Visistat.

As described in 3.1 "Data Identification", the first step of the Bayesian analysis is to specify the roles and types of the variables in the given dataset. This is analogous to the process in the original VisiStat by Subramanian [2014], so we basically adopted the UI from VisiStat. The dataset selection view is depicted in Figure 4.2. Once the user has uploaded the dataset, he can select the variables he wants to include in his analysis.

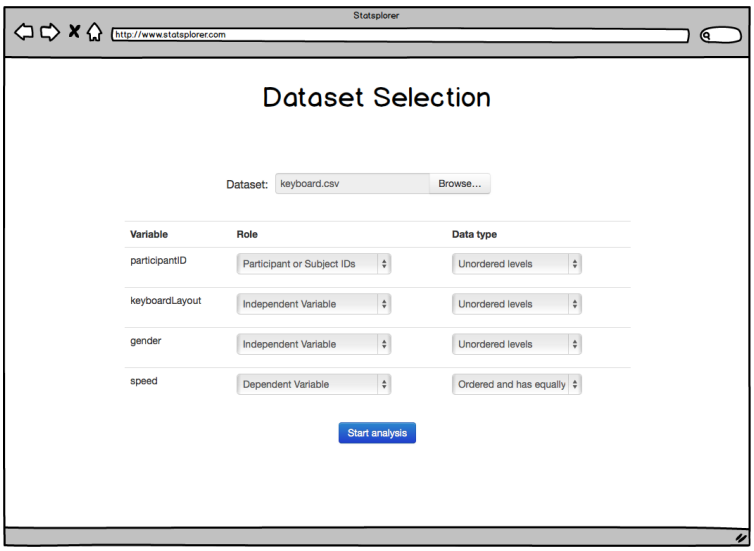

Figure 4.2: The user can upload a dataset and provide information about the roles and data types of each variable.

4.3 Hierarchical Model Picker

A suitable hierarchical model is derived from the experimental design behind the scenes.

Once the variable roles and data types have been specified by the user, the next step would be choosing the corresponding hierarchical model. Visistat by Subramanian [2014] determines the type of statistical test according to the variables the user selects. A similar mechanism can be used to select the appropriate hierarchical model. However, we do not want the user to have to understand the details of the Bayesian analysis process. Therefore we automatically choose the hierarchical model suitable for the selected variables. As Kruschke [2015] composed a set of R scripts which cover most common experimental designs, BayesianStatsplorer just loads the corresponding hierarchical model on the R side, and does not rely on any user input at this point.

4.4 Contrasts

Expert and novice analysts tackle the interpretation of contrasts differently. The following chapter explains some of the rationale behind the design of the contrast interpretation module in BayesianStatsplorer.

4.4.1 Design Rationale

Novices and Experts tackle contrast interpretation differently.

Whenever the experimental design contains more than one independent variable, the experimenter needs to potentially analyse interaction effects, main effects and simple effects. The expert might want to evaluate the interaction effects before he evaluates the main effects, because he knows that higher order interaction effects should be interpreted before the lower order effects, as they influence them. However, the novice might not know about the interaction effect, and just want to answer his research questions. Often, the formulated hypotheses are just main effect contrasts (e.g. I expect Qwerty to be faster than Colemak), hence the novice is looking for answers to these questions first. He might not even be aware of what an interaction effect is, which does not matter as long as there aren't any interaction effects. In order for the system to cater to both novices and experts, we want to achieve three goals:

We have three goals for ensuring that BayesianStatsplorer caters to both novices and experts.

1. The system should not initially overwhelm the user, but still allow the expert to start analysing right away.
2. The system should help transition the novice user to a more experienced and finally expert user.
3. The design should scale for experimental designs with potentially many contrasts.

Additionally to the three goals above, we would like the UI to expand vertically instead of horizontally, as recommended by Vora [2009], in order to keep the possibility of adding a sidebar for filtering on the left and a sidebar on

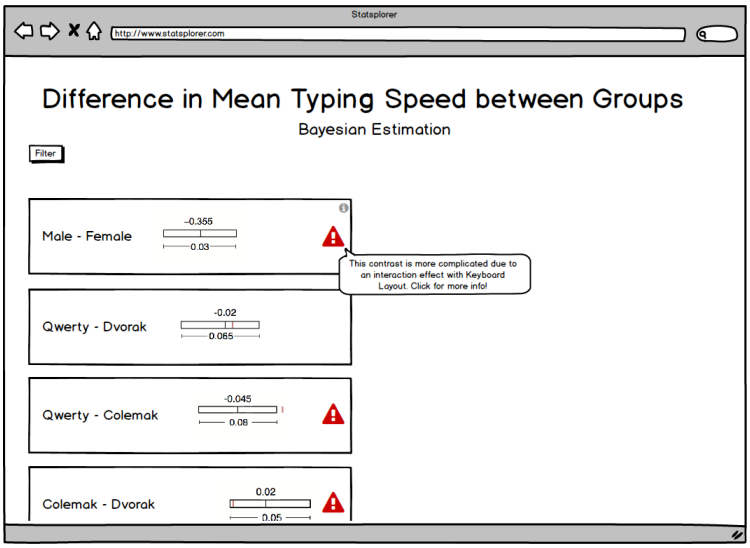

Figure 4.3: The main effect contrasts are the entry point for the contrast analysis. Contrasts influenced by interaction effects have a warning next to them, which guide the user towards the underlying additional information.

the right for reporting, as this has proven to be a useful layout for users (Subramanian [2014]). These goals drive the information hierarchy and design of every component in the contrast interpretation module of BayesianStatsplorer.

4.4.2 Layout and Navigation

Main effect contrasts are shown first, but if interaction effects are involved, a warning is provided.

As we wanted to avoid overwhelming the user with the interpretation of interaction effects at the beginning, the entry point of the contrasts view shows the main effects. If there aren't any interaction effects, the experimenter can just analyse these main effects, and does not even need to check for the existence of potential interaction effects. However, if there are interaction effects influencing the main effects, that main effect tile is highlighted with a warning, as visible in Figure 4.3. This warning acts as an information scent to the user that there is more underlying information to investigate. A user familiar with interaction effects will learn to parse the contrasts view for warning symbols, and

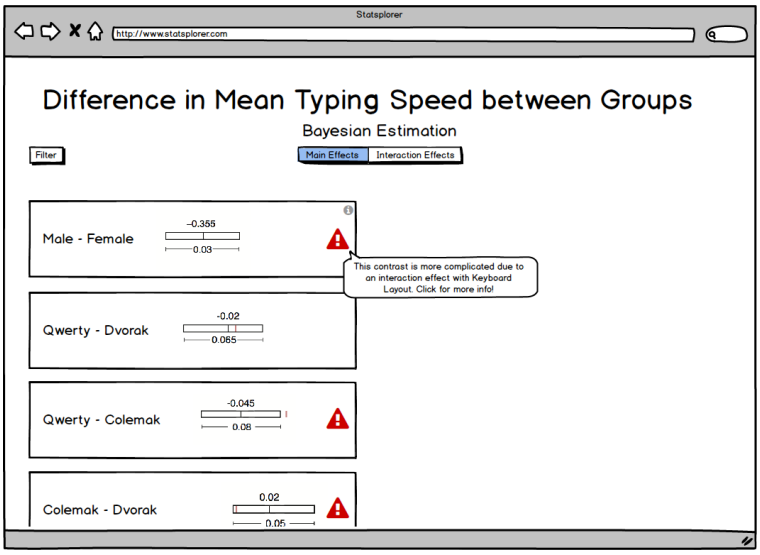

Figure 4.4: Main and interaction effects are separated in different tabs.

investigate the details of the simple effects.

A tab-view separating main effects from interaction effects suffices for experts, but could be confusing for novices.

Now that we've established, at least per default, that the main effects should be displayed first, we need to decide on how to provide the link and navigation from main effects to interaction effects and vice versa. One possibility would be showing the different contrasts in tab views, as mocked up in Figure 4.4 This layout would work reasonably well for the expert user. He could first evaluate the interaction effects tab, making a note of which contrasts levels of the independent variables have an interaction effect. He could then switch to the main effect view, and evaluate the main effects while bearing in mind the higher order interaction effects. This layout would presumably be understood by the expert, because he is aware of the fact that he has to interpret both interaction and main effects. However, this layout may not be intuitively understood by the novice user. He might not know what an interaction effect is, therefore be reluctant to investigate the interaction effects view, and not necessarily understand the relationship between the interaction and main effects. In order to alleviate this problem, we crafted the information hierarchy in a drill-down, accordion style view, as shown in Figure 4.5.

Instead, we chose a drill-down, hierarchical layout.

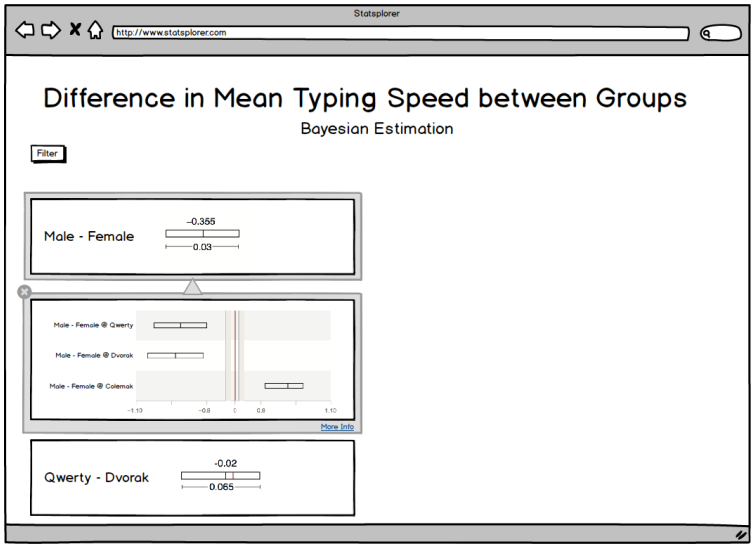

Figure 4.5: The user can expand the main effects influenced by an interaction effect, in order to see the simple effects.

The drill-down layout allows natural exploration, links the meaning of different contrasts and reduces cognitive load.

This hierarchical layout allows for the user to first orient his mindset by the main effects, potentially evaluating these and then explore the simple effects for the contrasts with an interaction effect. It also provides a natural way of linking the simple effects with the corresponding main effects, and reduces cognitive load. By further expanding the details view, the user gets a textual description of the effects, together with plots of the individual means and descriptive plots of the underlying data, as shown in Figure 4.6. The textual description can be fine tuned to contain both a general description on how to interpret the effects and a description dynamically adjusted to the context of the current analysis. This text together with the descriptive data plots may aid the user in further confirming the interpretations he made in the plots above. The text also teaches him how to interpret the plots directly, so as he becomes more experienced, he won't need the text any more, and will be able to interpret the contrast plots directly. This drill-down layout allows for incrementally providing the novice with more detail, while the expert will be able to quickly extract the desired information from the contrast plots in a streamlined work flow. It also has the benefit of providing a clean and uncluttered user interface which is pleasing to the eye.

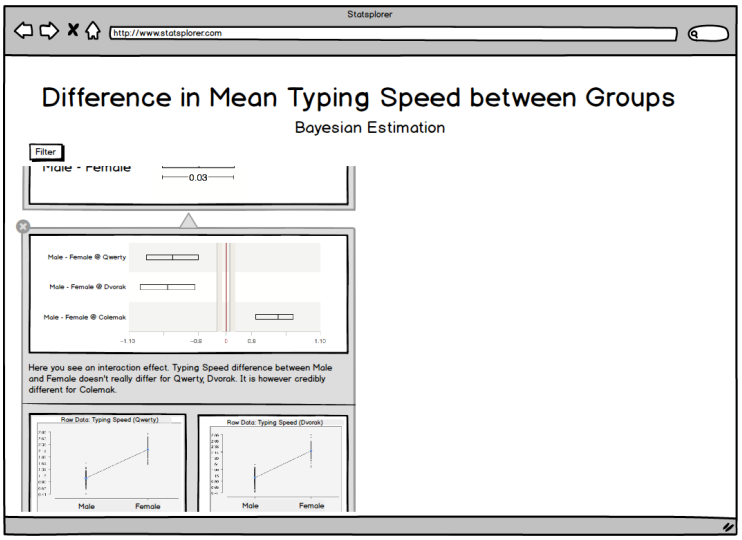

Figure 4.6: The simple effects can be further expanded in order to see a textual description on how to interpret the effects, together with descriptive plots of the underlying data.

Now that a spacial layout for the different contrast types has been established, the following section discusses the design decisions related to the ordering of the contrasts.

4.4.3 Contrast Order

Contrasts are grouped by predictor variable, and sorted by effect size.

In experimental designs with multiple independent variables with multiple levels, the Bayesian estimation can easily generate dozens of contrasts. These contrasts need to be sorted in some way when they are displayed to the user. The goal of the user once he reaches the contrasts is to assess how each condition of the independent variables affects the dependent variable. Therefore, we first group the contrasts by independent variable. This is apparent in Figure 4.3, where you first see the gender contrast, and then the keyboard layout contrasts. Additionally, independent variables that have a strong effect to the dependent variable will have a stronger causal evidence. Therefore, the contrasts are sorted within independent variables according to

the mean difference estimate, from smallest to largest. A visual cue for this is given by the location of the HDI plot within the contrast tile. Across different tiles, they form a diagonal line, subtly communicating the relation between the point estimates of different contrasts. While this grouping and sorting makes sense, the user might prefer to sort the contrasts by something other than the effect size, for example by the certainty of the estimated effects. He might want to show the most/least certain effects first. Therefore, BayesianStatsplorer could also include a "Sort by" dropdown selector, as seen in various commercial software.

The user could also select how he wants to sort the contrasts.

4.4.4 Filtering

While sorting can help the user find the results he is looking for, another important feature is filtering. Ideally, the user should be able to quickly filter the contrasts so that he only sees the ones he is currently interested in. Therefore we created a filter sidebar, where the user can filter the contrasts by independent variable or levels involved. This can be achieved by either selecting the independent variables or levels from a list, or by typing keywords into a search field, as shown in Figure 4.7.

The contrast filter reduces the amount of visible contrasts, and eases the process of finding the contrasts of interest.

Additionally to sorting and filtering, we envisioned a mechanism by which the contrasts could be highlighted according to the users expectations. Basically, the experimenter could formulate his hypotheses in the system, and then the contrasts which support his hypotheses could be highlighted in one colour, while contrasts which contradict his hypotheses could be highlighted in a different colour. This mechanism could help streamline the process of interpretation, and expedite the discovery of unexpected effects, and is explained in detail in chapter 4.5 "Semantic Contrast Layout".

4.4.5 A Contrast Tile

An individual contrast tile contains the title of the contrast and a summarised version of the probability density plot

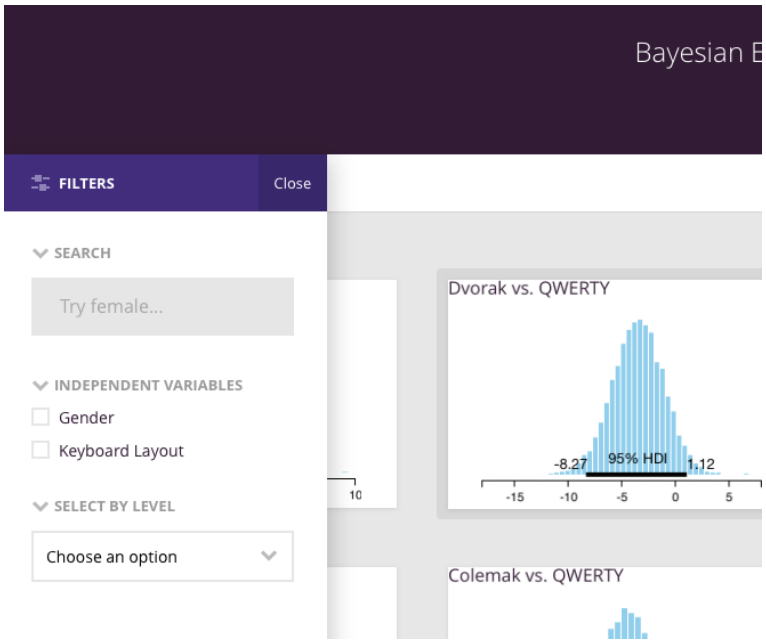

Figure 4.7: In the sidebar, the user can filter the contrasts by different variables and levels involved.

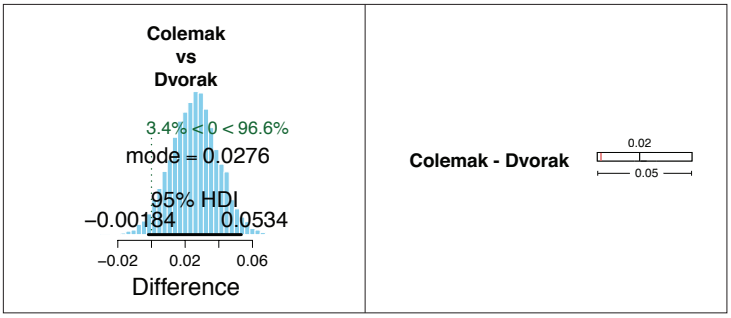

Figure 4.8: Left: the probability distribution generated by Kruschke's R scripts [2015]. Right: simplified contrast plot as used in BayesianStatsplorer.

of the mean difference estimate. Figure 4.8 shows the probability density plot as generated from the R scripts by Kruschke [2015] on the left, and the abstracted meaningful information on the right. We decided to reduce the probabil-

ity density plot as much as possible, until it contained the minimal amount of information necessary to interpret that contrast, while still conforming with the guidelines listed in chapter 2.4 "Uncertainty Visualisation". First of all we removed the histogram. While it could be useful to assess the relative credibility of different mean estimates, it is usually not required in order to draw conclusions from the plot. Second, we removed the axis, as the range is already given by the limits of the HDI. The central tendency and HDI is key to interpreting a contrast. Often, when interpreting a difference contrast, the user is attempting to find out several things.

We severely reduced the complexity of the probability density plot.

- What is the mean estimate of the difference between the two groups? All he has to do to answer this question is look at the point estimate of the mean, shown above the HDI.
- How precise is the estimate of the mean difference? To answer this question, the user can look at the span of the 95% HDI. We decided to show the size of the HDI instead of the two limits, because otherwise the user would have to perform the mental operation of estimating the HDI's size. As this is the main interpretation you can make from the HDI, we do not show the HDI limits. However, if the user hovers over the plot, the limits of the HDI are also shown.
- Is the difference between the two groups zero or close to zero? In order emphasize the credibility of zero difference, a red bar on the HDI hints towards the fact that zero is included in the HDI. This helps the user check for inclusion of zero at a glance.

Our plot makes it easy for the user to answer these questions.

Every element in the plot answers a specific question the user may be asking. However, the layout of the plots on the tiles would allow for the histogram and additional information to be rendered, as a user specified option for example.

Every contrast tile has a little menu in the top right corner, as shown in Figure 4.9. This contains operations specific to the given contrast. The menu contains four entries:

Every contrast has a small menu.

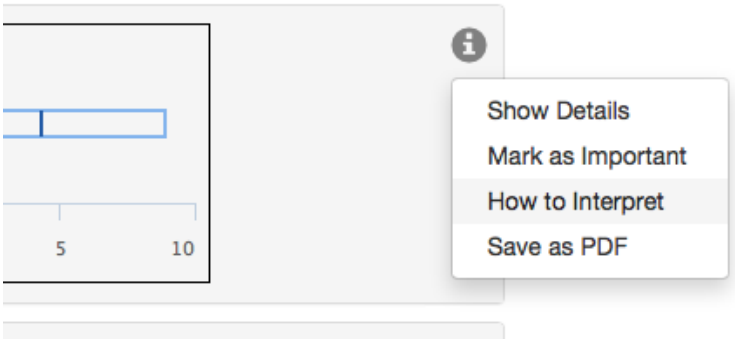

Figure 4.9: Every contrast has a menu in the top right corner, populated with some contrast specific actions.

The user can reveal more details.

- *Show Details:* This option is supposed to help novices discover that they can expand the individual contrasts. Once they have expanded a contrast, the animation of the panel opening should make them realise that they can open the panel directly by clicking on it. When the panel is open, this option changes to "Hide Details", and the users can see the expanded contrast. The underlying simple effects plot shows multiple HDIs for the given main contrast, separated by the various levels of the independent variable causing the interaction effect. Details on this plot are provided in chapter 4.4.6 "Simple Effect Contrasts".

The user can tag a contrast as important.

- *Mark as Important:* This option allows the user to mark a contrast as important. When there are many contrasts, only some of them may be meaningful to the experimenter. By marking them as important, the results of these contrasts could be explicitly listed in the report. Additionally, the user could later filter the contrasts to only show the ones he has tagged as meaningful. When the user clicks "Mark as Important", a little star appears in the top left corner of the contrast tile, as shown on the right in Figure 4.10. In order to streamline the process of tagging contrasts as meaningful, the user can also directly click on the star to toggle it on and off. When he hovers over the left corner, an outline of the star appears, to signify to the user that he can click there directly. Therefore,

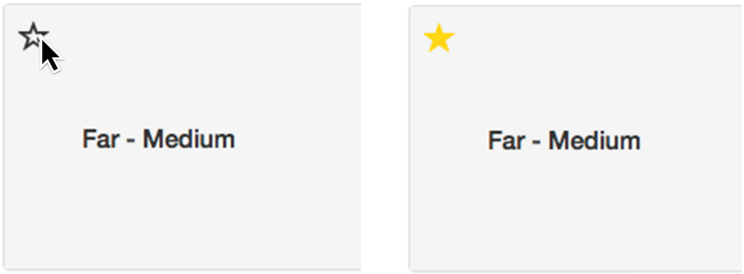

Figure 4.10: Left: The user sees an outline of the star when he hovers over the left corner of a contrast tile. Right: When he clicks, the contrast tile is tagged as meaningful.

a novice will perhaps first use the menu, but then he should discover that he can toggle the star directly. From then on he can scroll through the list of contrasts fluidly while toggling important contrasts.

The user can enter a tutorial mode.

- *How to Interpret:* When the user clicks on how to interpret, he gets taken to a modal view, which teaches him in a step by step process what a contrast is, how to interpret the HDI and what to look for. The step by step guide could either contain fixed examples which illustrate the meaning of the HDI, or the information could be given in the context of the current analysis. The latter would require somewhat smarter text generation, so that the explanations still make sense regardless of the dataset currently being analysed.
- *Save as PDF:* This option allows the user to save the contrast plot as a PDF, so that he can include a plot of the HDI in his report.

This contrast menu gives the novice user a possibility to pick operation from a textual, descriptive UI. Once he becomes more accustomed to the actions, he can use the less obvious, but therefore quicker to execute short cuts. In future iterations of BayesianStatsplorer, this contrast specific menu could be populated with other operations which should be applied to a given contrast.

The last component that a contrast tile can contain is a warning. This warning acts as an information scent to the

The warning acts as an information scent for the user to explore the interaction effect.

user that this contrast has to be interpreted with caution, due to the fact that there is an interaction effect at work. By expanding the warning, the user opens the main effect contrast panel, which reveals the simple effect contrast caused by the underlying interaction effect. As the novice becomes more experienced, he will learn to quickly parse the main effect contrasts for warnings, in order to assess which main effects he can interpret directly, and which effects he has to expand in order to get a realistic picture of the effects.

4.4.6 Simple Effect Contrasts

Interpreting simple effect contrasts requires a comparison of multiple HDIs.

Once the user expands a main effect contrast, that contrast is grouped by the levels of the independent variable causing the interaction effect. The relevant HDIs are merged into one plot, so that the user can easily perform the comparisons between the HDIs. When interpreting such a simple effects contrast plot, there are several things a user should be looking for. The tasks of looking for these things are what drive the design of the simple effects plot.

- The user should check whether the individual HDIs overlap. If there is significant overlap between HDIs, then there is no interaction effect between these groups. If the HDIs do not overlap, then the differences are credibly different. Next the user should find out how they differ.
- In order to assess the difference between the individual HDIs, it can be useful to check were zero is. If zero is between the HDIs, then the difference is positive for one group, but negative for the other group. If one of the HDIs includes zero, then the difference is credibly zero for one group, but not for the other.

The HDI comparison tasks drove the design of our simple effects plot.

Rooted in the tasks listed above, we derived the design shown in Figure 4.11. The HDIs are stacked vertically. This allows for verifying overlap at a glance, as can be seen in the left HDIs in Figure 4.11. Zero is explicitly highlighted to help orient the user, and the axis is interrupted. Using a

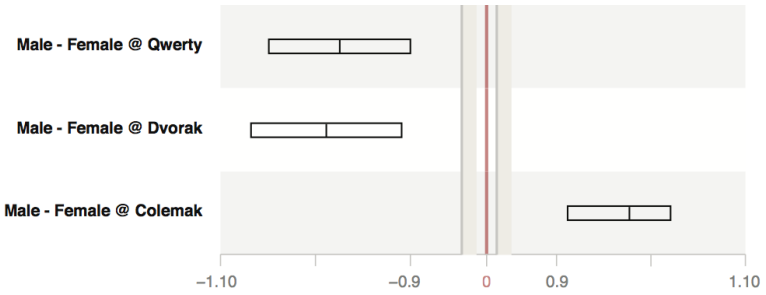

Figure 4.11: In this plot of simple effect contrasts, the HDIs are stacked vertically, in order to ease the checking for overlap. Zero is explicitly highlighted with a red line. Additionally, the axis is interrupted in order to provide a high resolution of the HDIs without having to show a lot of white space between them.

broken axis allows the plot to show the HDIs in high resolution, while removing unnecessary space between the HDIs. This is particularly useful for very precise HDIs, for example $[-10.9, -10.8]$ and $[23.4, 23.5]$. If these were plotted on one continuous axis, they would end up being tiny dots in the plot. With additional HDIs of that precision, it would become hard to check for overlap. Our technique allows for easy comparison and interpretation of both wide and narrow HDIs. Technical details on how the plot is created are provided in 5.2.5 "Contrasts"

The broken axis allows for displaying high resolution HDIs, even if they are far apart.

4.5 Semantic Contrast Layout

The design of the contrast tiles makes the interpretation of contrasts as streamlined and as easy as possible. In order to optimise this view further, we envisioned adding semantic context into the system, so that the results of the Bayesian analysis would actually be displayed in the context of the research questions the experimenter has in mind. To achieve this, two components need to be created:

Further optimisation would require the system knowing about the users' expectations of the outcome.

1. First, the experimenter needs to be given a way to formalise his expectations of the experiment's outcome

in the system.

2. Second, the system would have to compare the results of the actual Bayesian analysis and sort them according to whether they correspond or conflict with the experimenters expectations.

As a potential solution to the first necessary component, we present our hypothesis builder.

4.5.1 Hypothesis Builder

The experimenter's hypotheses need to be conveyed to the system.

To help the experimenter formalise his hypotheses, we leverage the fact that most experimenters will have concise textual hypotheses that they have already formulated as part of the experimental setup. Ideally, the user could just give the hypotheses to the system in natural language, and the system would be able to understand and create the mathematical formalisms out of the provided hypotheses. However, this requires advanced AI, so we propose an alternative design which is still relatively easy for the user to use, but also computationally feasible.

We envision a drag-and-drop hypothesis builder.

A hypothesis statement consists of relationships between different variables in the experiment. Therefore, we envision a system were the user can drag individual levels of the independent variables into different boxes, and select the relationship he expects to see between them. A prototypical UI approach for this is shown in Figure 4.12. If the users' hypotheses involve relationships between more than two levels, they can expand the "formula" with more boxes. Figure 4.13 shows some examples of possible hypotheses for our exemplary typing speed experiment together with how one would formulate them in the hypothesis builder UI.

Ideally, we would want the UI for the hypothesis builder to be complete and correct, i.e. every valid hypothesis can be specified, and no invalid hypothesis can be specified. Therefore, once the user has dragged a level into one box, he would not be able to drag a level into a box which would

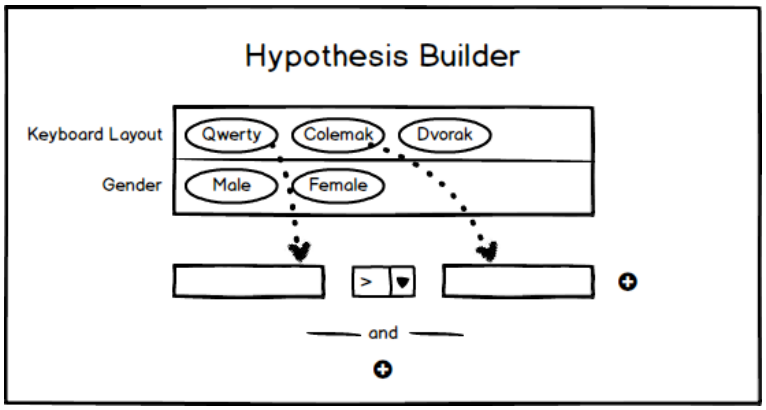

Figure 4.12: The hypothesis builder lets the user drag different levels into different boxes, and then specify the relationship between those levels. In this example, the user hypothesizes that typing speed is faster with Qwerty than with Colemak.

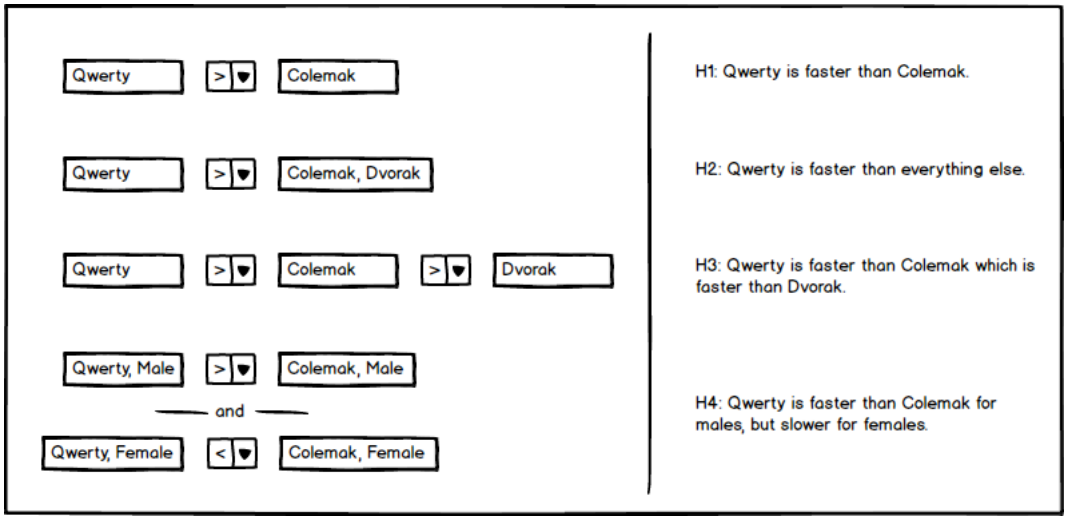

Figure 4.13: Some examples of how different hypotheses can be formulated in the hypothesis builder.

create an invalid hypothesis. This would help avoid the user specifying invalid hypotheses by mistake.

Now that the hypotheses have been specified, the results of the Bayesian analysis need to be compared with the hypotheses the experimenter set up. Basically, we envi-

Contrasts could be highlighted depending on whether they support or conflict with the specified hypotheses.

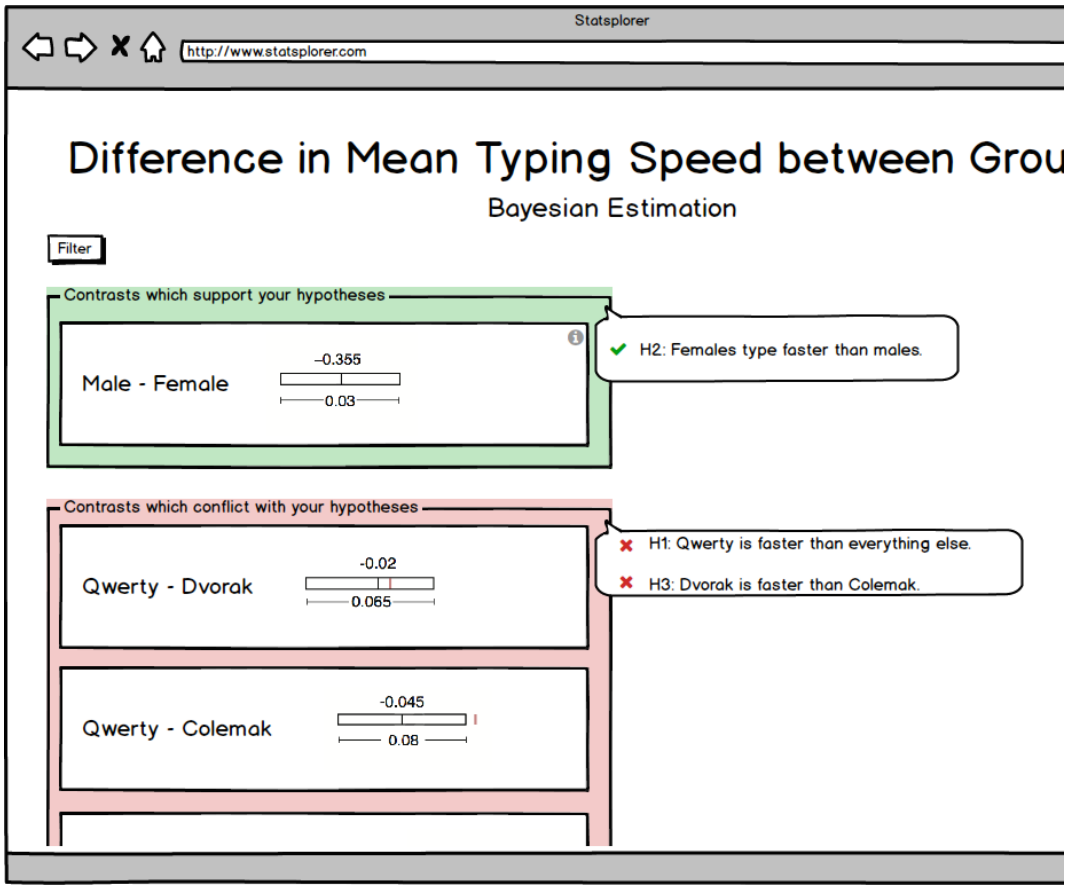

Figure 4.14: The system can group contrasts according to whether they support or conflict with the hypotheses specified by the user, and show the relevant hypotheses.

sion that the contrasts which directly affect the specified hypotheses would be at the top of the contrasts list, and they would be highlighted in either green or red depending on whether they support or conflict with the specified hypotheses, as shown in Figure 4.14.

Unfortunately, in the scope of this thesis, we were not able to find sufficient literature on what a valid hypothesis is, and therefore were not able to prove the correctness and completeness of our approach. As the idea for the hypothesis builder developed fairly late in the time line of this thesis, we were not able to create a specific implementation. The verification and implementation of the hypothesis

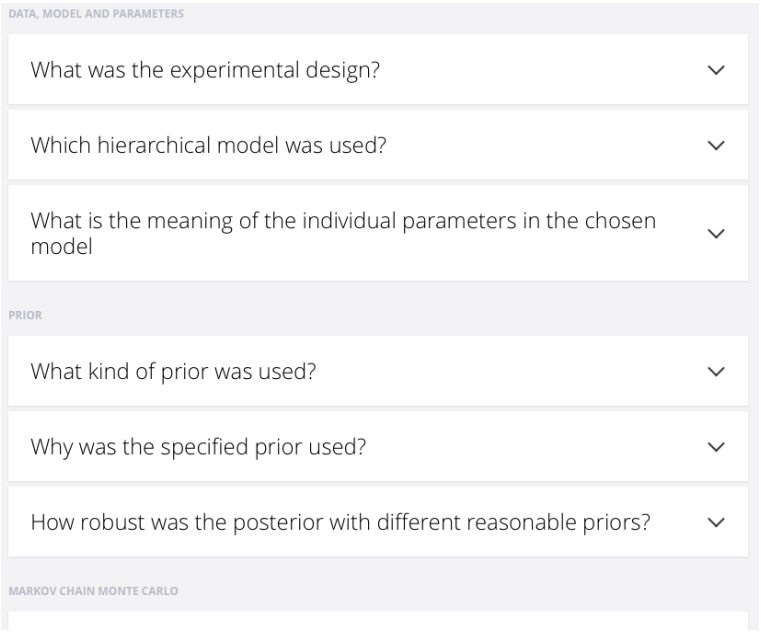

Figure 4.15: The report section is composed of different categories, each of which contains questions relevant to the data analysis. The answers to the different questions can be revealed by expanding those questions.

builder remains as one of the goals for future work.

4.6 Report

A conventional format for the report of Bayesian analyses does not exist yet.

As Bayesian analyses are not yet standard practice in many fields of research, a conventional format for reporting Bayesian analysis results does not really exist yet. Therefore, our report module is informed by the guidelines proposed by Kruschke [2015], which we summarised in chapter 2.5 "Reporting Bayesian Data Analyses". The report section consists of several categories, each of which contains questions with corresponding answers (see Figure 4.15. We chose this design for several reasons:

Answer both the "what" and the "why".

1. According to Pullenayegum et al. [2012], when experimenters report their analyses, they should avoid fol-

lowing a set of rules unthinkingly. Instead, they need to focus on thinking critically about what is needed. That is, they should not just know the "what", but also the "why". The Question-Answer format helps the experimenter understand both the "what" and the "why".

2. The Question-Answer format helps assuring the user about his interpretation. For example, he may have concluded from the contrast interpretation that there were no interaction effects. When he then expands the "Where there any interaction effects between independent variables?", and it corresponds with his conclusion, he can be reassured that he has interpreted the results correctly.

3. This format makes it easy for the user to adjust his report according to the background knowledge of his targeted audience, as Kruschke [2015] recommends. He can glance at the questions, and decide whether the answers to those questions need to be provided to the user.

A verbosity slider could make the explanations in the report more or less detailed.

In the following, we expand on the last point. For an experienced audience, certain questions from the report may be able to be dropped completely. Other questions may still have to be answered, however they could be answered more concisely, possibly with less detail. Therefore, we envision some sort of verbosity slider, which lets the user adjust how much descriptive text is in the answers of the questions. High verbosity would lead to a very elaborate explanations, whereas low verbosity would make the answer as short and concise as possible. The user would be able to specify both the overall verbosity of the whole report, and then potentially the verbosity of individual sections, which then override the overall verbosity. The report could then be exported into different formats once it has been adjusted to the appropriate verbosity level.

Chapter 5

Implementation

This chapter provides an overview of the modules in BayesianStatsplorer together with some details about how they were implemented. The following sections briefly describes the technologies involved.

5.1 Technologies and Frameworks

BayesianStatsplorer uses R for the statistical computation, CoffeeScript and AngularJS for the application logic, and HTML5/CSS for the UI.

BayesianStatsplorer was built from the ground up with a modular approach in mind, using AngularJS[1], CoffeeScript[2] and R[3]. AngularJS is a framework for building large-scale, interactive web applications, and lends itself well to thorough testing. CoffeeScript is a language that compiles into JavaScript, without any interpretation at runtime. The language makes certain "everyday" programming tasks a little more convenient than their JavaScript counterparts. R is a programming environment for statistical computing. It has an extensive range of packages for statistics of all kinds. We use the OpenCPU[4] API for integrating CoffeeScript and R. The statistical computations are performed on the R side by OpenCPU, which returns

[1] www.angularjs.org
[2] www.coffeescript.org
[3] www.r-project.org
[4] www.opencpu.org

the results to our Angular application. This allows us to separate the statistical computations from the behaviour, logic and UI of BayesianStatsplorer. For the graphs and visualisation, we use both Vega[5] and Highcharts[6].

The following section describes the modular design of BayesianStatsplorer, and briefly elaborates the responsibilities of each module.

5.2 BayesianStatsplorer Modules

BayesianStatsplorer is composed of several main modules, which are shown in a system context diagram in Figure 5.1. Each module is as self-contained as possible, and interface with the other modules through defined APIs. This allows for easy modification and testing of individual components.

5.2.1 Dataset

As a first step the user chooses a .csv file and uploads it to the system. The Dataset module handles the identification of variables and levels. It contains basic heuristics to identify the type of the variable (Metric, Nominal, Ordinal, Dichotomous) and the role (Independent, Dependent). Additionally, it could perform sanity checks of the data to compensate for missing data points, etc. The dataset module also exposes UI with which the user can edit the roles and types of the independent variables, and select/deselect the variables that should be involved in the analysis.

5.2.2 Boxplot

The boxplot module plots the selected dependent variable grouped by the selected independent variables. In the box-

[5] www.github.com/trifacta/vega
[6] www.highcharts.com

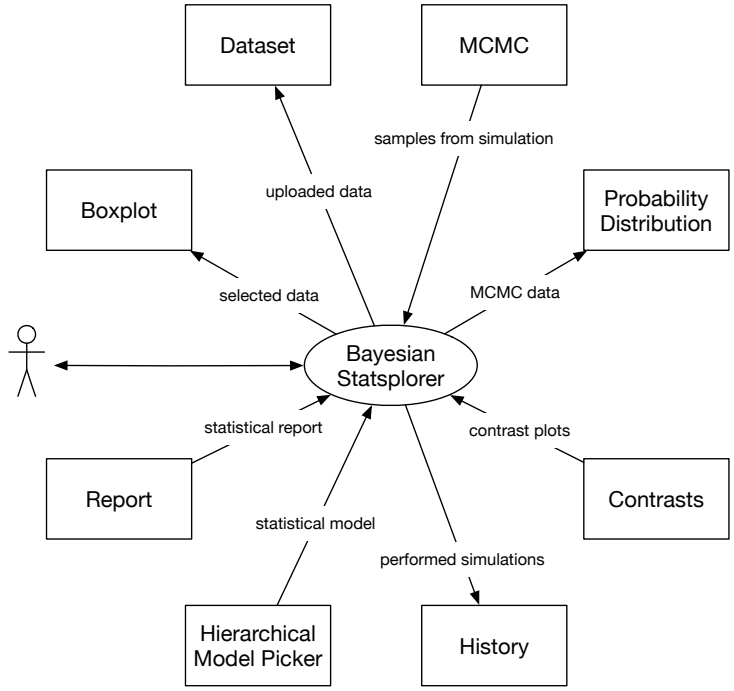

Figure 5.1: This system context diagram shows the main modules of BayesianStatsplorer.

plot, the user can choose to select multiple means that he's interested in comparing. The boxplot is the first visualisation of the data the user sees. It gives him some descriptive information about the tendencies in his data.

5.2.3 Hierarchical Model Picker

The hierarchical model picker decides which model to load based on the experimental design.

Based on the experimental design, a different hierarchical model is required for Bayesian analysis. The hierarchical model picker decides which model to load and use in the analysis process. It receives the selected variables as input, and chooses the appropriate hierarchical model, with default priors. This module could also contain UI for the user to specify a custom prior, and potentially modify the model to fit his needs.

5.2.4 MCMC

The MCMC module runs the sampling algorithm, and passes the posterior distribution on to the Contrasts module.

The MCMC module handles the actual details of the Bayesian analysis. It runs the MCMC sampling process in R, and calculates the posterior distribution of the parameters. When the MCMC process has finished, it evaluates the chain based on criteria specified in chapter 3.6 "MCMC" If the criteria are not fulfilled, the MCMC can be run again with a longer chain length. The MCMC module also provides UI with the diagnostics of the MCMC chains, so that the user can evaluate the representativeness and accuracy of the generated chains. If the MCMC passes the necessary criteria, the posterior distribution is passed on to the contrasts module.

5.2.5 Contrasts

A contrasts service holds all the data, which the contrasts controller can access and display.

The contrast module is responsible for generating the necessary comparisons within the specified dataset. It generates all the main and interaction effect contrasts, renders the HDI plots, sorts them and presents them to the user. The `ContrastsController` instantiates a `ContrastsService` with the posterior distribution, who then handles the generation of the required contrasts, and fetches the HDI information for each contrast from the R side. When it has finished fetching all the contrast data from R, it passes an array of contrast information back to the `ContrastsController`, who then uses the `SimpleHDIFactory` and `MultipleHDIFactory` to create the HDI plots as discussed in chapter 4.4.5 "A Contrast Tile" and 4.4.6 "Simple Effect Contrasts". This is illustrated in a sequence diagram in Figure 5.2.

Different factories exist for our various plots.

The `MultipleHDIFactory` instantiates a `HDIPlotController`, who handles the logic behind creating the HDI plots as shown in Figure 4.11. The approach to achieve the visualisation in Figure 4.11 is illustrated in Figure 5.3. First, the HDIs are sorted in ascending order by their left limit. Then, the sorted HDIs are divided into groups. All HDIs which share some common values are placed into one group. After that, the algorithm

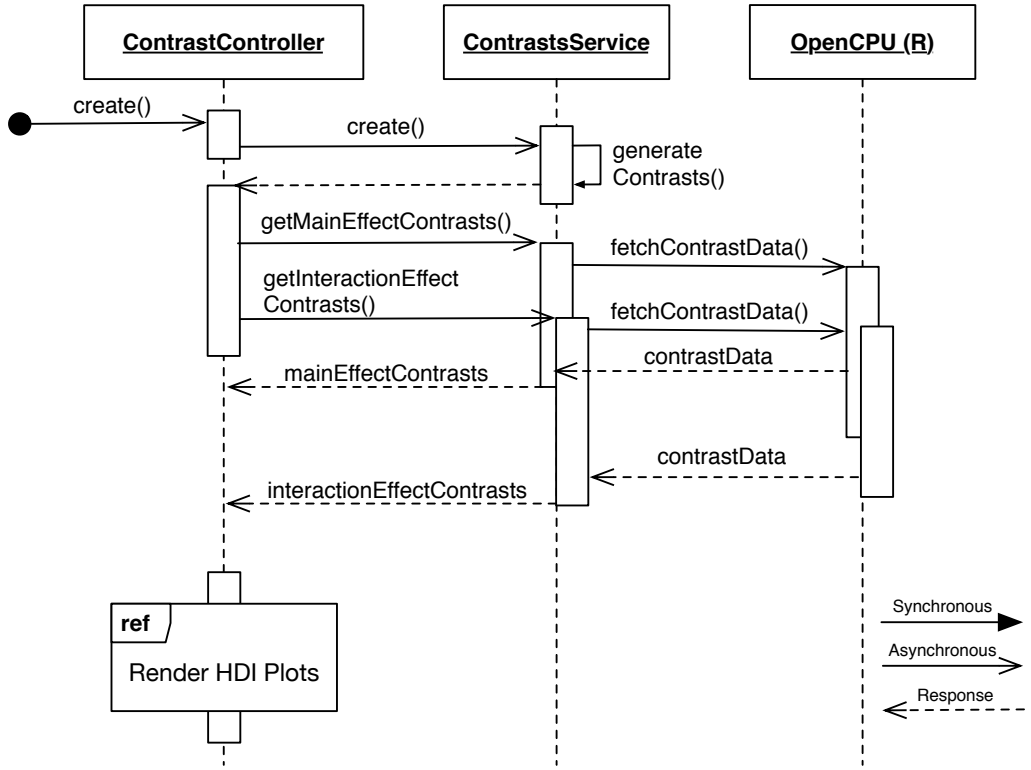

Figure 5.2: This sequence diagram shows how the contrasts are generated and populated with the real data.

computes the largest scale of all groups, as individual HDIs can have different ranges. The largest scale is then used to scale the axis. At this point, there may be large areas of white space between different HDI groups, which is now removed by breaking the axis up in between groups, and replacing it with the interrupted canvas visualisation shown in 4.11. Finally, a vertical line is plotted to show the user where zero is. Once all these properties have been calculated, the `HDIPlotController` loads the Highchart configuration with the calculated properties. The plot is then rendered in the `ContrastsView`.

The plot with multiple HDIs is generated step by step.

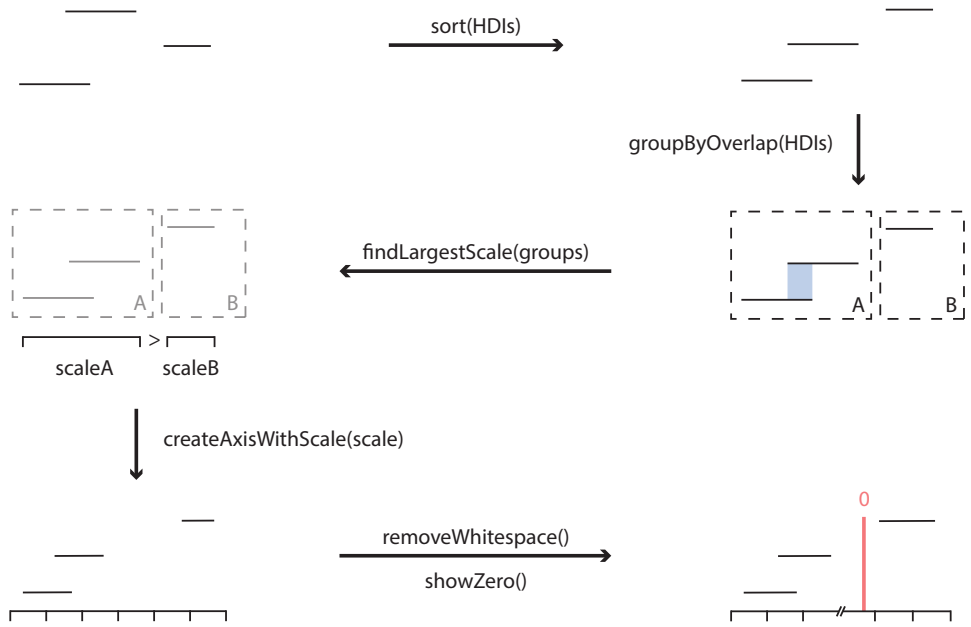

Figure 5.3: Illustration of how the multiple HDI plot with a broken axis is generated.

5.2.6 Report

The report module was implemented specifically for easy modification in the future.

The report module is responsible for generating the textual report of the Bayesian analysis. As explained in previous chapters, there is no established set of conventions for reporting Bayesian analyses. Therefore, we wanted to create a solution which will allow us to update and extend the contents of the report easily, without having to write too much code. We came up with the following approach.

The report template is in a human readable format, which is loaded by the report service.

The `ReportFactory` instantiates a `ReportController`, who in turn instantiates the `ReportService` and acts as the controller for the `ReportView`. The `ReportService` loads the appropriate `ReportTemplate` for the given hierarchical model. The `ReportService` can parse the template and replace all the placeholders with the real values from the analysis. Whenever all the placeholders in a question/answer have been replaced, i.e. the user has completed the steps necessary to answer that question,

the controller is notified and he can update the questions shown in the view. The `ReportController` and the `ReportService` are designed such that they do not need any modification, even if the questions and answers are changed entirely. The only thing which has to be consistent between the `ReportTemplate` and the `ReportService` the names of the placeholders. The `ReportTemplate` is a JSON file which contains all the questions and answers.

The placeholders in the template get replaced with real data at runtime.

```
{
  "title":"Is the MCMC chain representative of the posterior
      distribution?",
  "answers":[
    {
      "condition":"true",
      "value": "The values in the MCMC chain..."
    },
    {
      "condition": "shrinkFactor <= 1.1",
      "value": "Visual inspection of the trace plot..."
    },
    {
      "condition": "shrinkFactor > 1.1",
      "value": "The trace plot shows that..."
    }
  ],
  "category": "mcmc"
}
```

Listing 5.1: A JSON question object. It contains the question's title, category and an array of answer sentences, each of which can have a condition under which they should be displayed in the report.

It's basically just an array of questions, where each question has a category, a title and an array of answers, as shown in listing 5.1. Each answer can have a condition under which it should be shown. For example in listing 5.1, we can specify an answer for the case that the shrink factor is larger than 1.1, and an answer for the case that the shrink factor is smaller than 1.1. We can also provide sentences which should always be shown, regardless of the analysis outcome, by specifying the condition as just "true". These sentences are usually descriptions which are independent

All sentences in the report can have a condition which must be fulfilled in order for it to be shown.

of the actual outcome of the experiment.

The report template could easily be extended to support the verbosity slider.

The design of the report module allows for more questions and answers to be added easily, without having to rewrite much code. It also allows for easy implementation of the "verbosity slider" as suggested in chapter 4.6 "Report". Each answer sentence could be given a verbosity level, and the `ReportService` would only have to add the sentences to the displayed answer which correspond to the user's selected verbosity level. Additionally to the easy customisability of the report template, the visual report can also be generated bit by bit. The `ReportService` updates the report whenever new data is available, which allows the report to grow gradually as the user progresses through his analysis, starting with the variable selection, the MCMC diagnosis and finally the interpretation of the posterior distribution. This means the user is not forced to complete his full analysis before he looks into the report, but can perform it step by step if he desires, and consult the report in parallel, potentially helping him in his interpretations.

Chapter 6

Evaluation

6.1 Without Users

We simulated experimental outcomes to evaluate the interaction design.

Once we had developed the interaction design for the contrast interpretation, we wanted to assure that it was suitable for interpreting any possible relationship between variables. We therefore simulated datasets for all possible outcomes of a two-by-two factorial design as shown in Figure 6.1. For each dataset, we verified that the layout of the contrast results still made it easy for the user to find the intended effects. This allowed us to be relatively certain that there were no blatant flaws in the interaction design regarding the hierarchy of results.

6.2 With Users

We performed two brief qualitative evaluations.

During the course of the development cycles of BayesianStatsplorer, we performed think-aloud evaluations in order to gather qualitative feedback. The first one evaluated the information hierarchy and overall design of BayesianStatsplorer using a click-through prototype created in Balsamiq[1]. The prototype is shown in chapter A "BayesianStatsplorer Mockup". We asked the main

[1] www.balsamiq.com

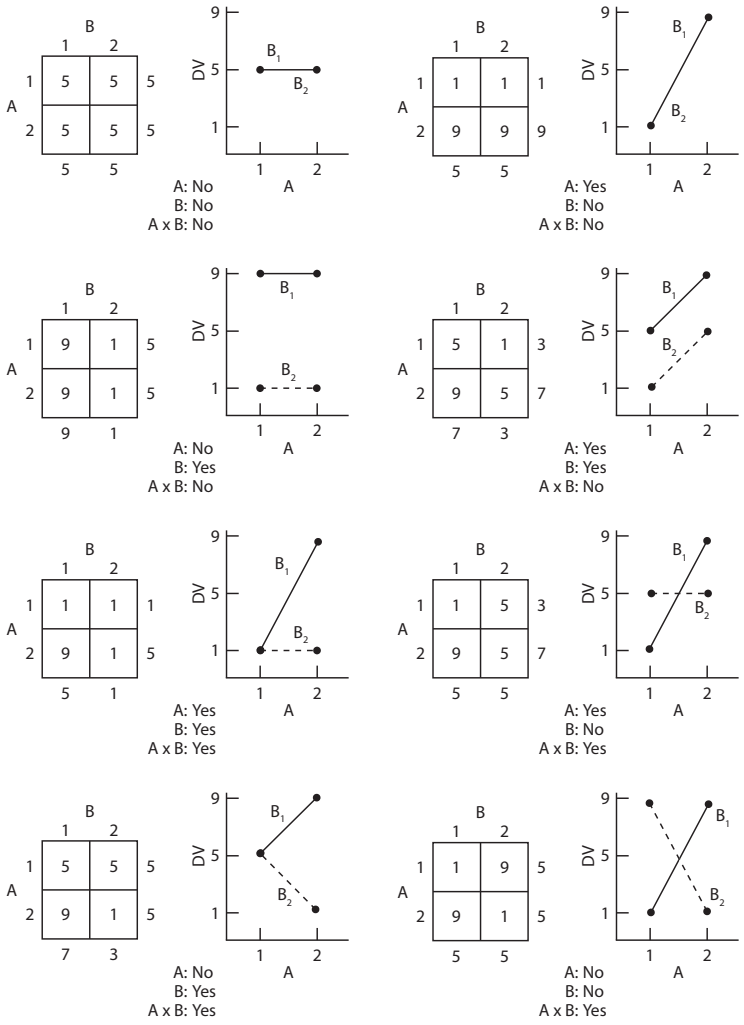

Figure 6.1: All possible outcomes of a two-by-two factorial design (Cozby and Bates [2012]).

developer of Visistat (Subramanian [2014]) and another HCI researcher with basic knowledge in experimental procedures to give us some feedback. In the second evaluation, we gathered further qualitative feedback on the individual components of BayesianStatsplorer with the implemented prototype. We walked three users through two datasets and evaluated whether their thought processes were matched with the UI of the system, and whether they could make the correct interpretations. None of the

User	Background	Experience
E	HCI	Researcher, expert in NHST, developer of Visistat.
U1, U2, U3	HCI	Students, solid understanding of NHST.
N	CS	Student, no experience in statistical analysis.

Table 6.1: Summary of the users' background and experience.

users had ever performed a Bayesian analysis, but their expertise in NHST varied, as shown in Table 6.1. Below is a summary of the feedback we received in both evaluations.

- Once a contrast with interaction effects was fully expanded to the simple effect contrasts, U1 stated that he was not particularly interested in the descriptive text being in the context of the current data. It was more important to him that the text included instructions on what to check the plot for, so the position of zero in relation to the HDIs, the overlap of the HDIs and if any of them include zero. Therefore, we tried to make the text include both a description of what to look for, and what these visual properties of the plot actually mean, as shown in Figure 6.2.

  U1 suggested that the detail view should contain some bullet points on what to check the plots for.

- User E suggested inverting the contrasts were the difference of means was negative, so that all the contrasts had a positive difference. For example, if a difference was "Male - Female: -0.3", then the contrast should be displayed as "Female - Male: 0.3". Showing all the differences as a positive number reduces the cognitive load of interpreting whether a certain difference between Group A and Group B actually means Group A or Group B is "better". In the simple effects plot however, the user will still be required to sometimes interpret negative differences.

  All difference contrasts could be made positive by inverting the comparisons when necessary.

- When tagging a main effect contrast as meaningful, U1 was not certain whether this would mean that all the lower level plots are tagged as meaningful too. We decided it was less destructive to include too much information and have the user filter it out, than to include too little information and have the user for-

  The users were not sure whether tagging a contrast as meaningful would tag all underlying contrasts.

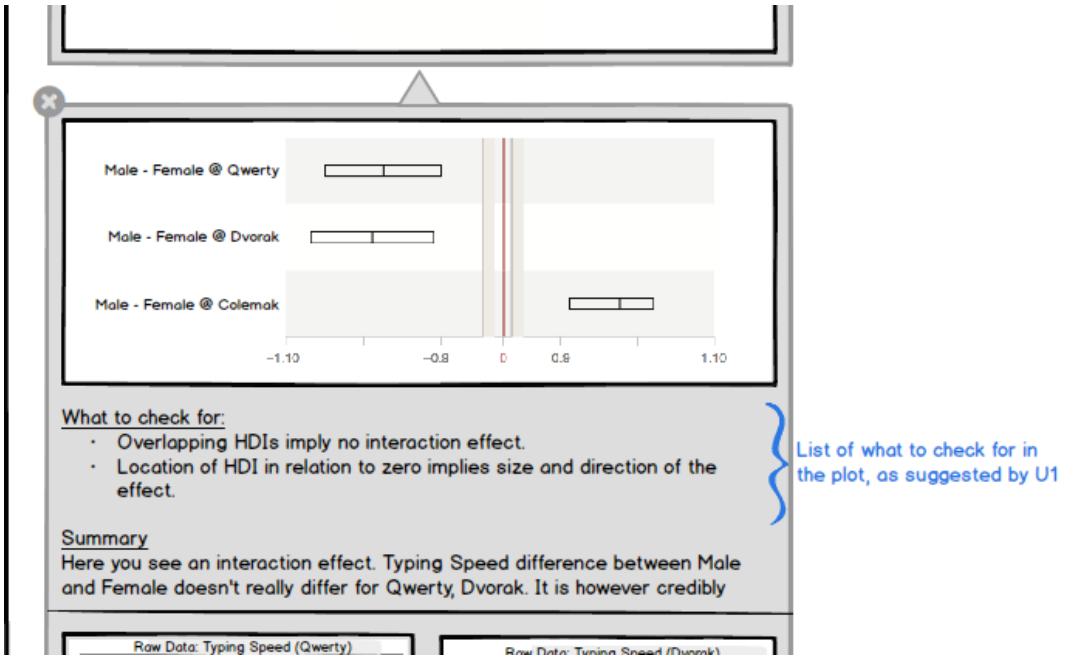

Figure 6.2: User U1 suggested adding a list of what to check for in the plot. We integrated this list into the detail view.

get about it. Therefore, when a main effect contrast is tagged as meaningful, the underlying simple effect contrasts are tagged as meaningful as well.

- All terms which the user might not understand should show a question mark on hover, so that they have a direct link to the meaning (User N).

Users wanted a description of Bayesian data analysis at the beginning.

- At the beginning of the Bayesian analysis procedure, there should be a description of what a Bayesian analysis is. We did have that description in the report, but some users (N, U2) mentioned that they would like to have that information before even starting to use the system. This text could easily be added in the dataset selection step, which is the first screen the user would see.

- The axis in the contrasts should be annotated with the dependent variable name, so that the measure is clear (U3).

- Several users (U1, U3) mentioned a desire for the system to highlight and sort contrasts based on whether

the contrasted groups were credibly non-zero or not. This desire is rooted in the binary interpretation style of NHST (significant or not significant). As Bayesian analysis allows for richer inference, we wanted to avoid tempting the user to only make such conclusions.

- User U2 initially thought the warning meant the data was in some way not complete. Therefore rephrasing it to "interaction effect found" might be more suitable.

Feedback from the evaluation indicates that the design goals were fulfilled.

All users were able to make the right interpretation from the contrasts. They were also able to recognize the interaction effects instantly, without any aid. This indicates that the chosen visualisation is suitable for communicating the effects in the data, while not overwhelming the user with information, even when there are many contrasts. These were goals we had in mind when we designed the system in chapter 4.4 "Contrasts", and the evaluation with users confirms our confidence in the chosen design. The feedback received from the evaluation with users also helped refine our software, and can be integrated into our existing design with ease.

Chapter 7

Summary and Future Work

In the previous chapter, we discussed the evaluation at various development stages of BayesianStatsplorer. In this chapter, we summarise the contributions of this thesis, and provide an outlook into potential future work and features that could be added to BayesianStatsplorer.

7.1 Summary

In the first chapter, we discussed some limitations and problems of NHST, and how they can be alleviated with a Bayesian analysis approach. We reviewed some related work which enables Bayesian analysis, and gathered some reusable UI components from software which tries to simplify statistical analyses, together with some guidelines on good visualisations and reporting.

The second chapter introduces the theory and workflow of a Bayesian analysis. It explains each of the fundamental steps in more detail, and shows how such an analysis could currently be performed, using R scripts for the statistical computation and visualisation. The basic tasks the user needs to be perform become clear, and those tasks are sim-

plified by the interaction design we proposed in the next chapter.

The third chapter discusses our interaction design for BayesianStatsplorer. For each step which requires UI, we propose a UI solution, with particular focus on the interpretation of the contrasts and reporting. We explain the design rationales and benefits of our proposed design.

The fourth chapter reveals some implementation details of our modular framework for a web-based Bayesian analysis application, and demonstrates some of the approaches we used to separate the statistical computation, the application logic, and the UI.

Finally, in the fifth chapter, we discussed the feedback received from two qualitative, small scale evaluations with users.

7.2 Future Work

More hierarchical models need to be added.

In the scope of this thesis, we did not integrate all hierarchical models into the system. We used a hierarchical model for multiple nominal predictors and a metric predicted variable. The design of the backend however allows for further hierarchical models to be added relatively easily. In order to deploy BayesianStatsplorer, the most common hierarchical models should be included.

Other modules can still be improved.

Additionally, there are still several modules in BayesianStatsplorer which could be severely improved. An appropriate interaction design for the creation of custom hierarchical models, prior specification and MCMC diagnosis could really enhance the ease with which inexperienced experimenters could perform Bayesian analyses. An actual version of our hypothesis builder, as presented in chapter 4.5.1 “Hypothesis Builder”, could be implemented and tested. We think making a system that links expectations with results could be crucial to making a really intuitive statistical analysis tool. These components could then all be integrated into one web application.

Once all these components are integrated, various user studies could be performed. Are HCI researchers with very little to no experience in Bayesian analysis able to interpret their data correctly? Do they gather more information from BayesianStatsplorer than with other commercial tools? Does the interaction design aid in the understanding of the underlying Bayesian concepts? These are questions which could be addressed once all components of BayesianStatsplorer are integrated.

Ultimately, we envision an integration of the original Statsplorer and BayesianStatsplorer, where the experimenter can analyse his data either with the classical NHST approach, or with the Bayesian approach, compare the results and gather rich inference from his datasets.

Appendix A

BayesianStatsplorer Mockup

The following figures show some screenshots from our final mockup of BayesianStatsplorer which was used to perform the first evaluation.

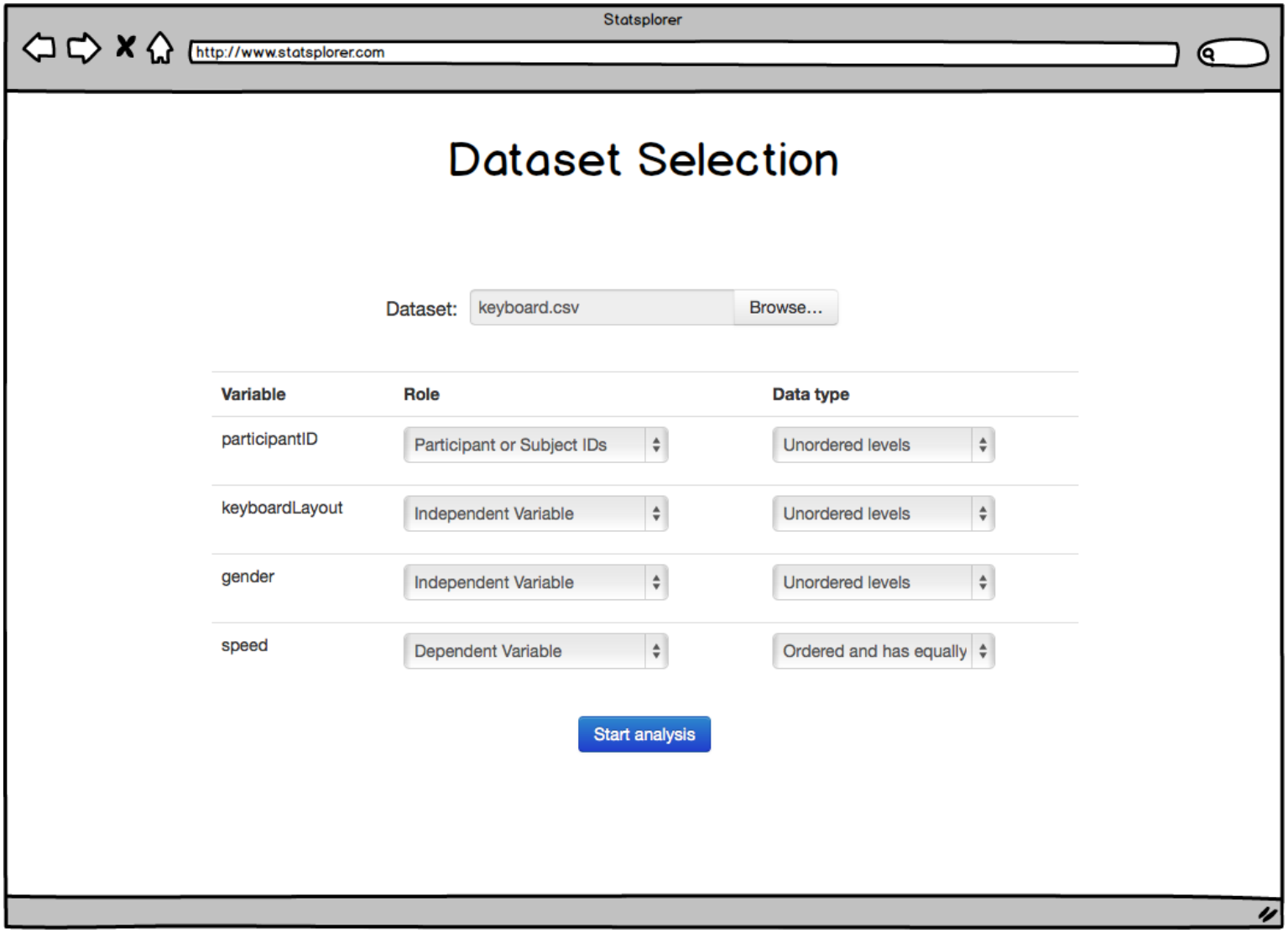

Figure A.1: Dataset selection and variable specification.

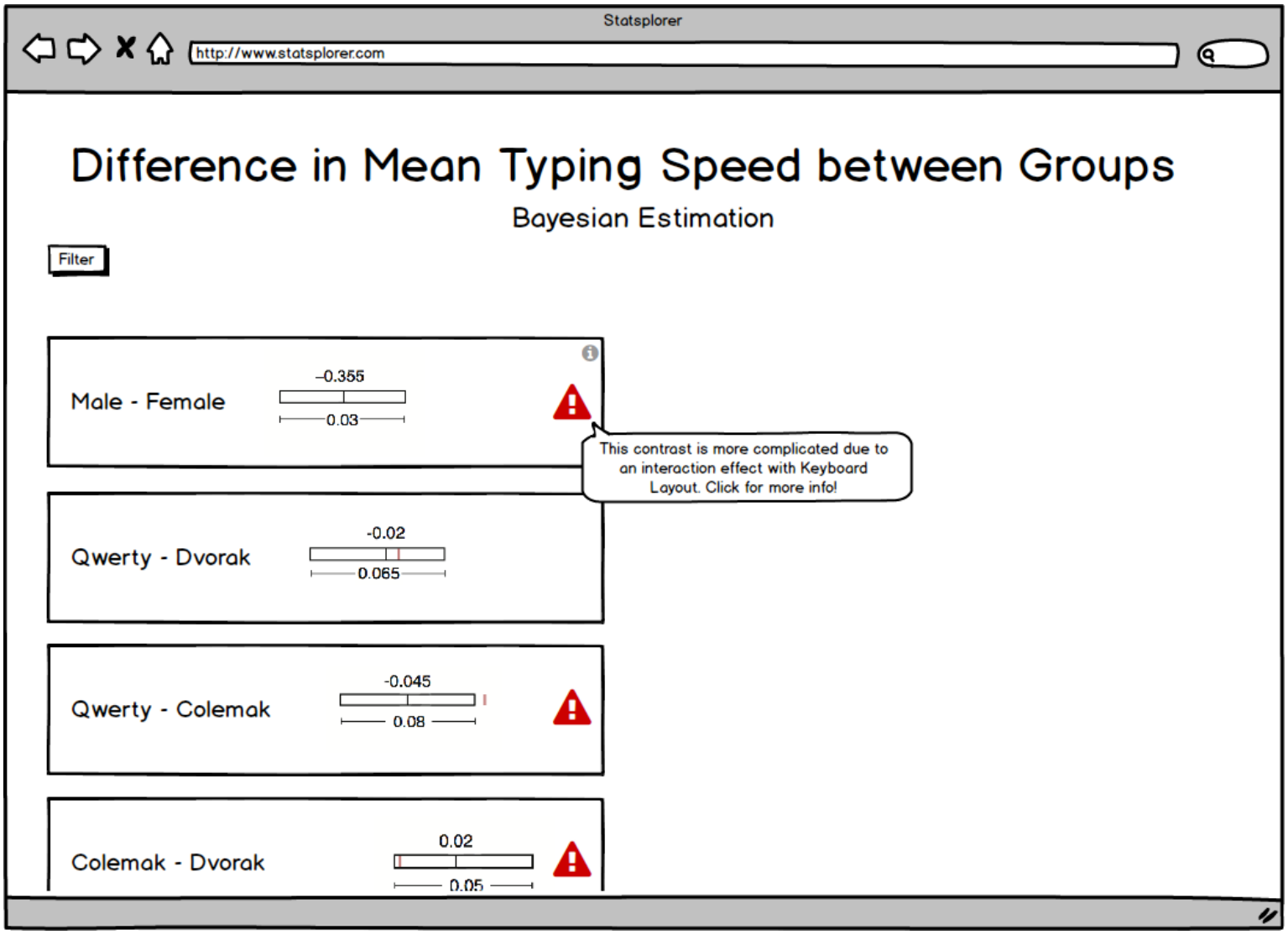

Figure A.2: Main effect contrasts have warnings if there is an interaction effect interfering. This acts as an information scent to dig deeper and interpret those effects.

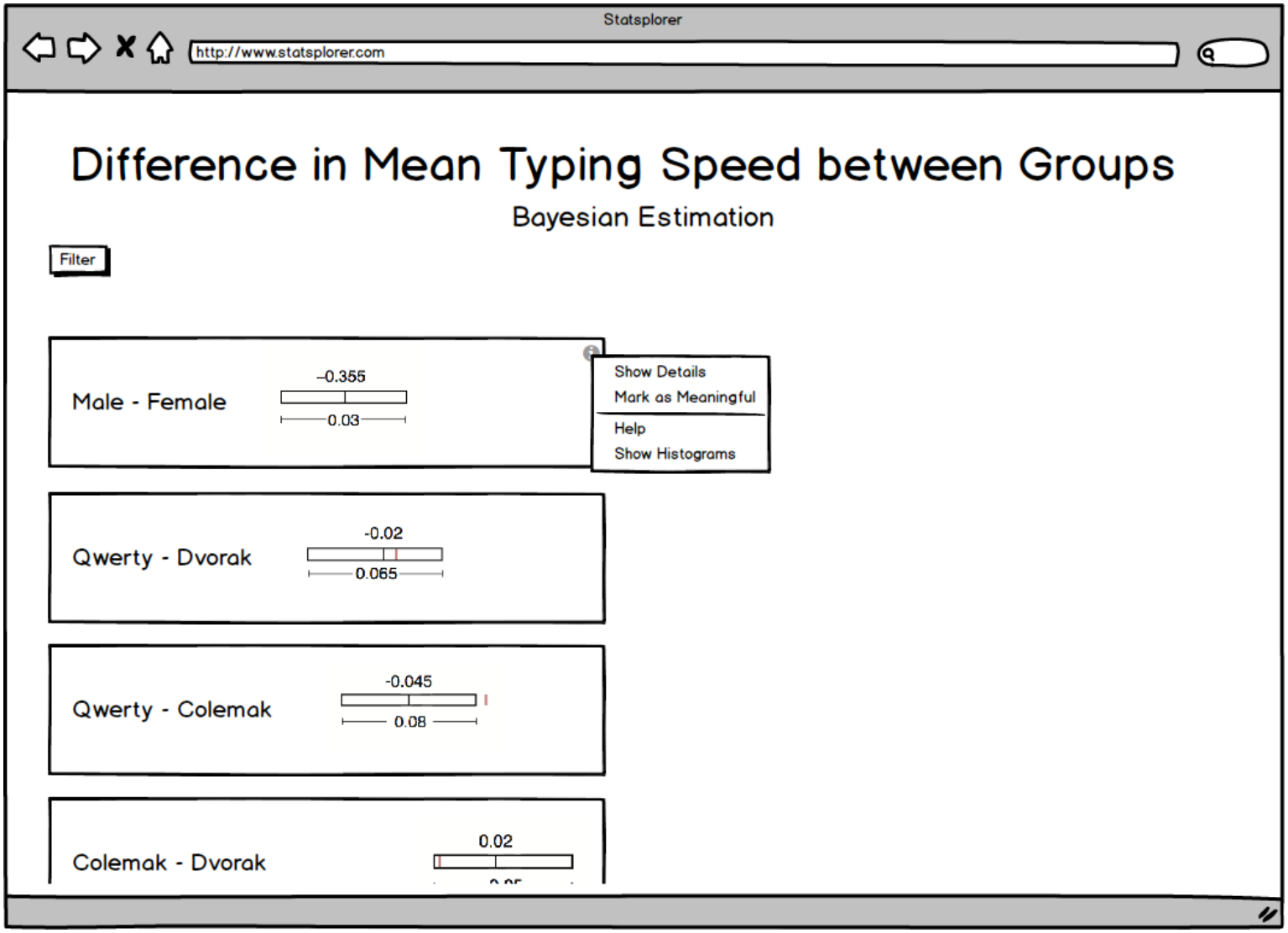

Figure A.3: Each contrast has a menu with several possible operations.

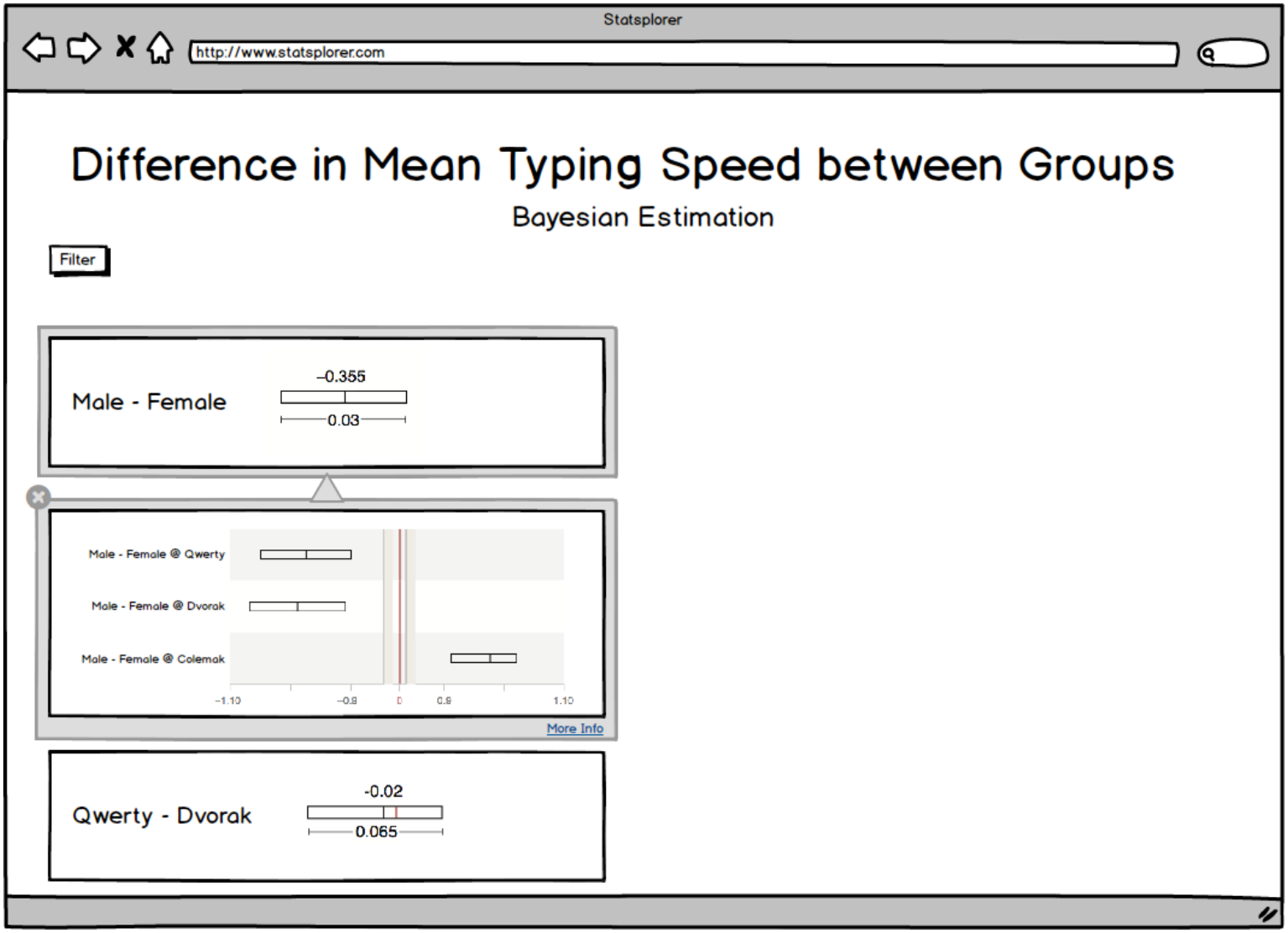

Figure A.4: Multiple related simple effects are summarised in one plot for easy interpretation.

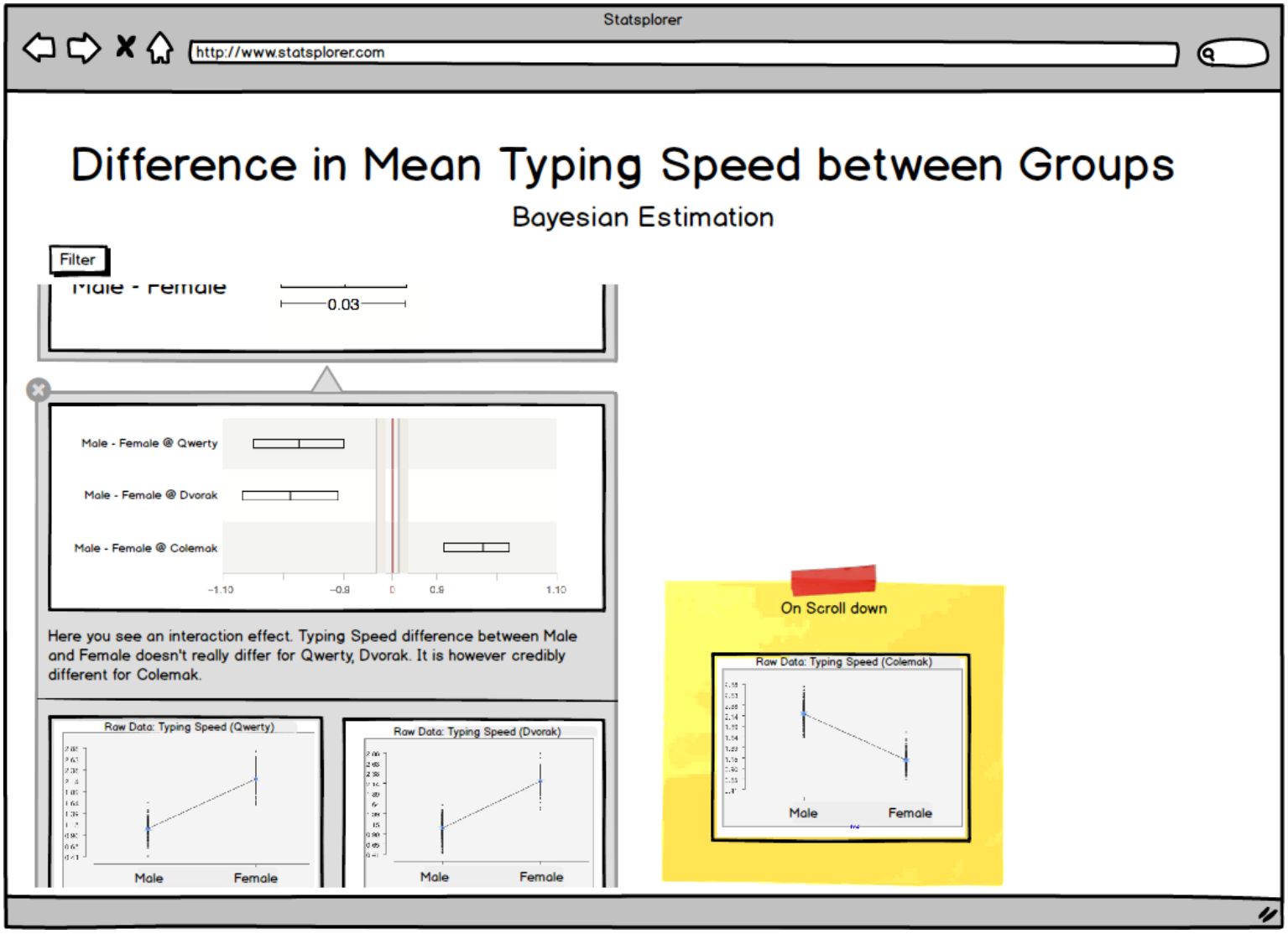

Figure A.5: Detailed view of multiple simple effects.

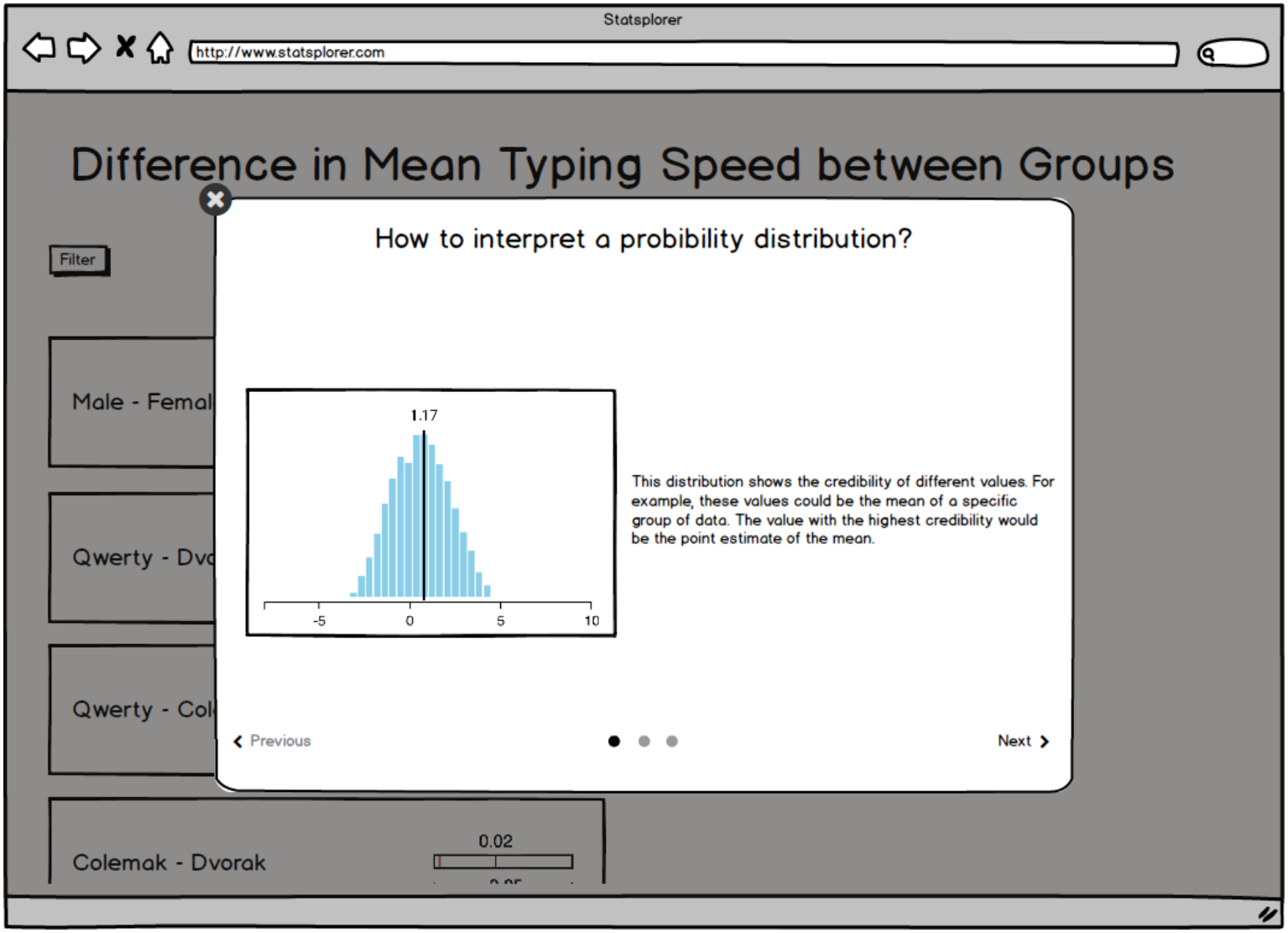

Figure A.6: Tutorial as a modal view, where the user can click through and learn how to interpret Bayesian results.

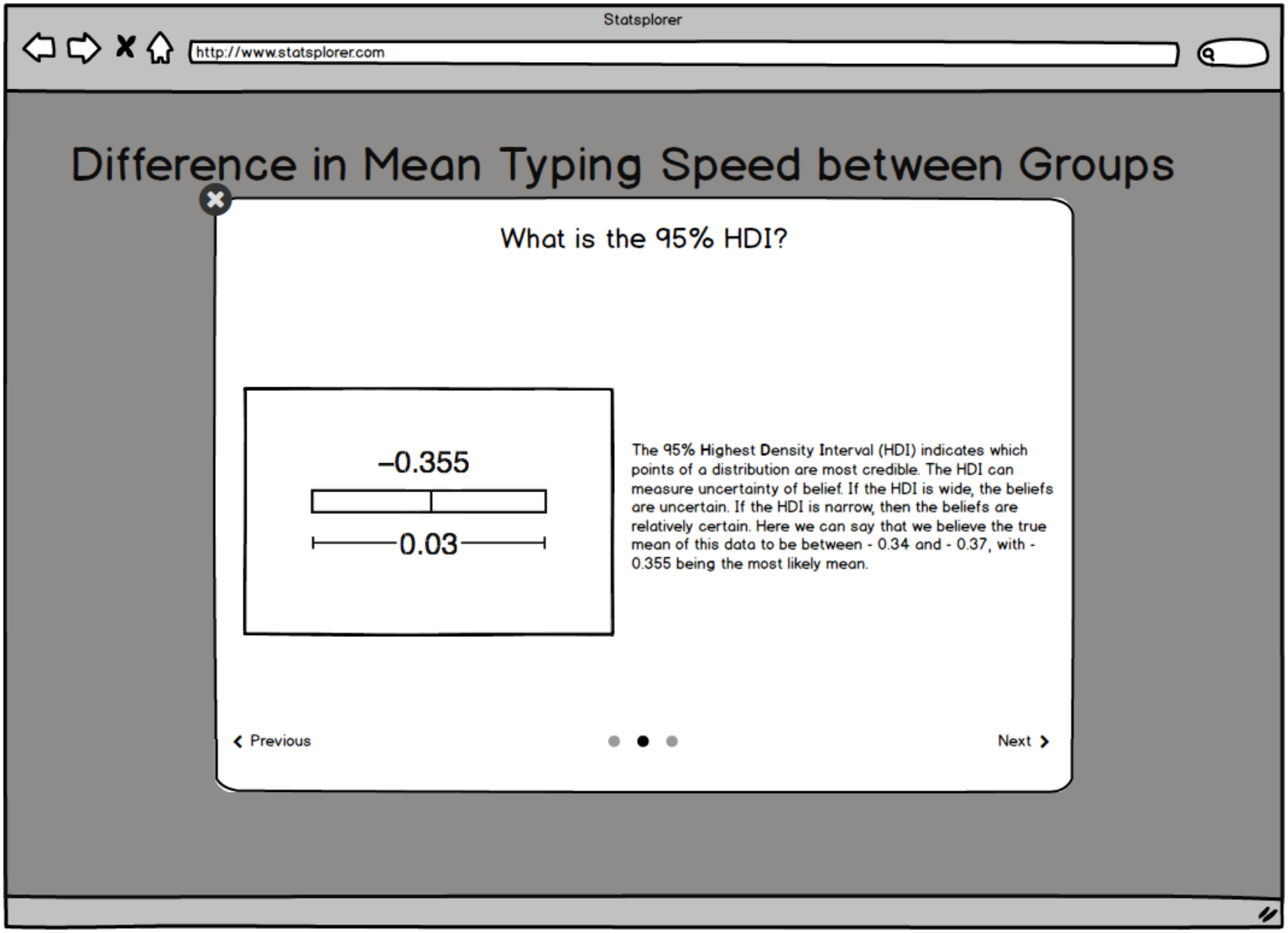

Figure A.7: Tutorial as a modal view, where the user can click through and learn how to interpret Bayesian results.

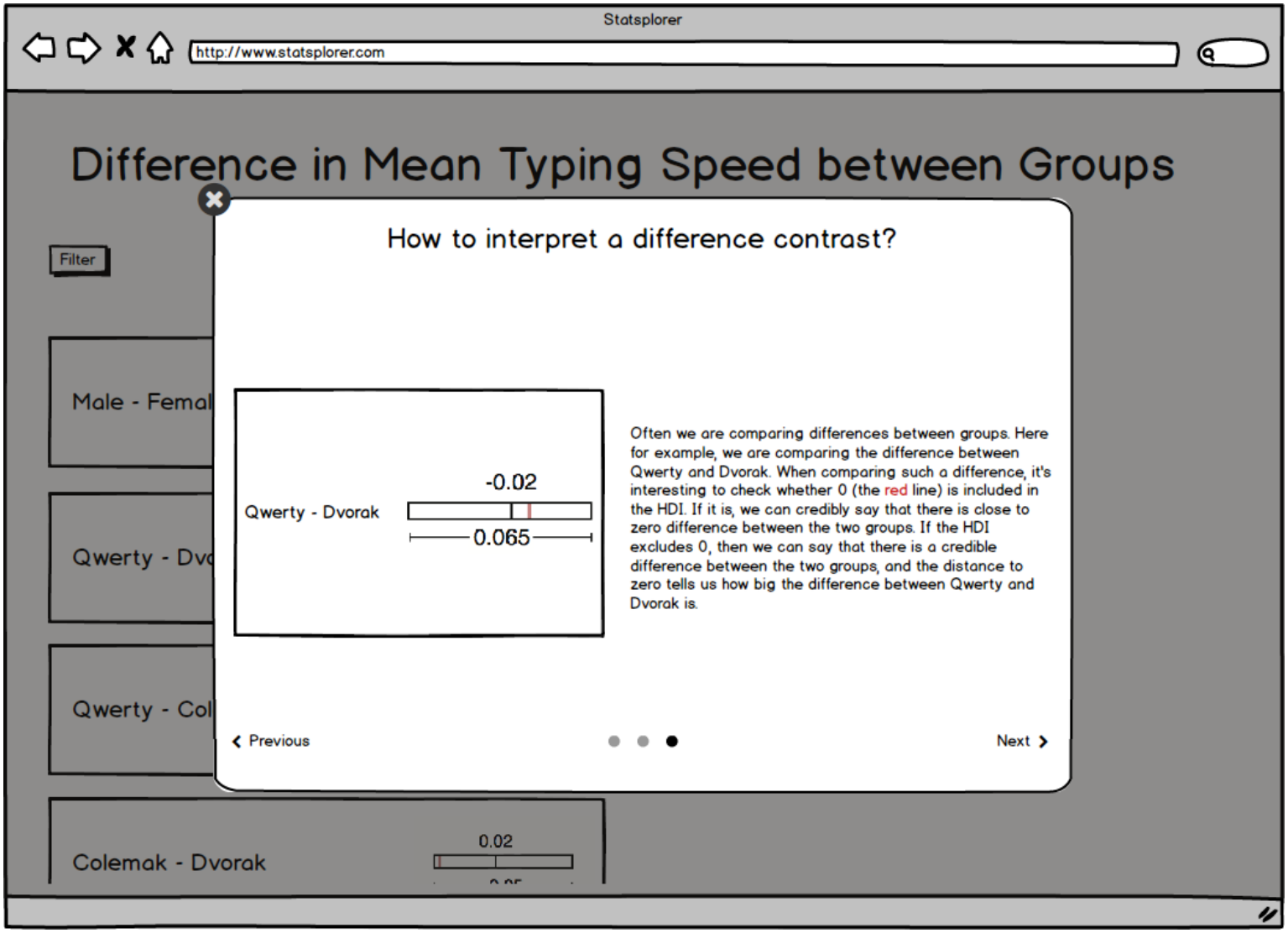

Figure A.8: Tutorial as a modal view, where the user can click through and learn how to interpret Bayesian results.

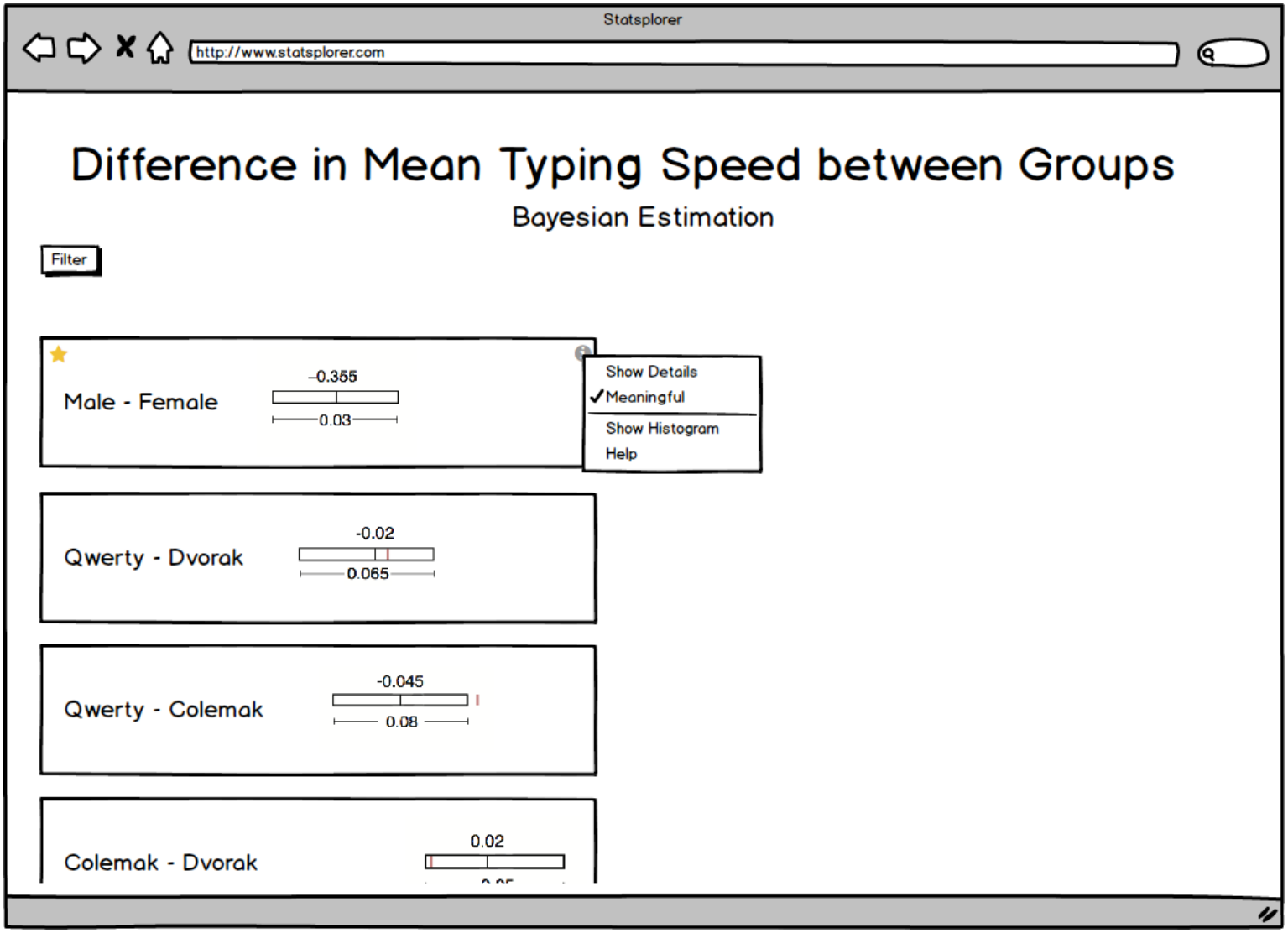

Figure A.9: Contrasts can be tagged as meaningful, either in the contrast menu, or by directly clicking on the star.

Bibliography

American Psychological Association et al. APA style guide to electronic references. 2012.

Thomas Bayes. An Essay towards Solving a Problem in the Doctrine of Chances. by the Late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S. *Philosophical Transactions (1683-1775)*, 53: pp. 370–418, 1763. ISSN 02607085. URL http://www.jstor.org/stable/105741.

Paul Cairns. Hci... Not As It Should Be: Inferential Statistics in HCI Research. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...But Not As We Know It - Volume 1*, BCS-HCI '07, pages 195–201, Swinton, UK, UK, 2007. British Computer Society. ISBN 978-1-902505-94-7. URL http://dl.acm.org/citation.cfm?id=1531294.1531321.

Jacob Cohen. A power primer. *Psychological bulletin*, 112(1): 155, 1992.

M. Correll and M. Gleicher. Error Bars Considered Harmful: Exploring Alternate Encodings for Mean and Error. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2142–2151, Dec 2014. ISSN 1077-2626. doi: 10.1109/TVCG.2014.2346298.

Paul C. Cozby and Scott C. Bates. *Methods in behavioral research*. McGraw-Hill New York, 2012.

Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):pp. 457–472, 1992. ISSN 08834237. URL http://www.jstor.org/stable/2246093.

Maurits Kaptein and Judy Robertson. Rethinking Statistical Analysis Methods for CHI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1105–1114, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208557. URL http://doi.acm.org/10.1145/2207676.2208557.

John Kruschke. What to believe: Bayesian methods for data analysis. *Trends in Cognitive Sciences*, 14(7):293 – 300, 2010. ISSN 1364-6613. doi: http://dx.doi.org/10.1016/j.tics.2010.05.001. URL http://www.sciencedirect.com/science/article/pii/S1364661310000926.

John Kruschke. *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*. Academic Press, 2015.

David J. Lunn, Andrew Thomas, Nicky Best, and David Spiegelhalter. WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, 10(4):325–337, 2000. ISSN 0960-3174. doi: 10.1023/A:1008929526011. URL http://dx.doi.org/10.1023/A%3A1008929526011.

Petri Myllymäki, Tomi Silander, Henry Tirri, and Pekka Uronen. B-Course: A Web-Based Tool for Bayesian and Causal Data Analysis. *International Journal on Artificial Intelligence Tools*, 11(03):369–387, 2002. doi: 10.1142/S0218213002000940. URL http://www.worldscientific.com/doi/abs/10.1142/S0218213002000940.

Kevin O'Brien and Jean Wright. How to Write a Protocol. *Journal of orthodontics*, 29(1):58–61, 2002.

Eleanor M. Pullenayegum, Qing Guo, and Robert B. Hopkins. Developing critical thinking about reporting of Bayesian analyses. *Journal of Statistics Education*, 20(1):n1, 2012.

Krishna Subramanian. Visistat: Visualization-driven, interactive statistical analysis. Master's thesis, RWTH Aachen University, Aachen, March 2014.

Pawan Vora. *Web application design patterns*. Morgan Kaufmann, 2009.

Chat Wacharamanotham, Krishna Subramanian, Sarah Theres Völkel, and Jan Borchers. Statsplorer: Guiding Novices in Statistical Analysis. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2693–2702, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702347. URL `http://doi.acm.org/10.1145/2702123.2702347`.

Index

Typeset September 2, 2015