Cognitive Modeling (INFOMCM)
Lab Assignment 3: Probabilistic models (esp. Bayes)

Developer and coordinator: Chris Janssen, with input from Leendert van Maanen
Teachers: see course manual

Background and goals

Different modeling techniques have different uses. *Probabilistic models* are useful to account for uncertainty. *Bayesian modeling* techniques in particular are useful for handling situations with uncertainty. Therefore, they are useful for model comparison purposes. In this lab, you will use probabilistic techniques – mostly based upon Bayes' theorem – to relate models to *psychological data and theories*.
At the end of this lab session, you can:

1. Implement and apply Bayes' theorem for experiments
2. Implement probabilistic models to formalize theory
3. Implement competing (probabilistic) models to compare their predictions
4. Use BIC to compare models
5. Systematically evaluate how (free) parameters impact model fit
6. Evaluate model outcomes to select the best fitting model that considers model fit and model complexity

Structure and time-management

There are 4 sections, each with 1 Blackboard question (so: 4 points total). You learn:

1. Section 1: how Bayes' theorem can be used for experimental tests (as an alternative for classic hypothesis testing)
2. Section 2: how to make a cognitive model using probabilistic techniques, as instantiated in a Relative Frequency model
3. Section 3: how to make a cognitive model using probabilistic techniques (Bayesian Sampling) and how to compare different models using Bayesian Information Criterion
4. Section 4: how to systematically evaluate how (free) parameters impact model fit

The suggested time-line for the average student is:

- Session 1: Finish sections 1 and 2.
- Session 2: Finish sections 3 and 4. Optional: bonus exercises.

Literature

Two papers can be used as background information (see later for more info):

- Section 1: Etz, A., & Vandekerckhove, J. (2018). Introduction to Bayesian inference for psychology. *Psychonomic bulletin & review*, *25*(1), 5-34. https://link.springer.com/article/10.3758/s13423-017-1262-3
- Section 2-4: Zhu, J. Q., Sanborn, A. N., & Chater, N. (2020). The Bayesian sampler: Generic Bayesian inference causes incoherence in human probability judgments. *Psychological review*, *127*(5), 719. https://doi.org/10.1037/rev0000190

Section 1: Using Bayes' theorem for statistical tests

1.a. Bayes basics

The basics of Bayesian statistics have been explained in class, and are assumed. If you want even more background, I recommend reading this article:

Etz, A., & Vandekerckhove, J. (2018). Introduction to Bayesian inference for psychology. *Psychonomic bulletin & review*, *25*(1), 5-34. https://link.springer.com/article/10.3758/s13423-017-1262-3

In statistics, Bayes' rule (or Bayes' theorem) can be used to test how likely some hypothesis H is[1], given the data D:

$$P(H|D) = \frac{P(H)P(D|H)}{P(D)} \qquad \text{(equation 1.1: Bayes rule)}$$

Self-test 1.1:

You should be able to answer these questions before you continue:

- What is the name of each component in equation 1.1? What are the prior, posterior, likelihood?
- How do you rewrite equation 1 to show P(D | H)?
- P(D) can be rewritten as: P(D) = P(D & H) + P(D & !H) (where "&" stands for "conjunction" and "!" stands for "not" – where !H is seen as any other situation than H). This function itself can be rewritten using only the prior odds of H and !H and conditional probabilities of D given H and of D given !H. What is this formula?

If you struggled with these questions, you need to catch up on the basics of Bayesian statistics. For example, with the Etz & Vandekerckhove (2018) article.

Engineering 1.1: Bayes with two competing hypotheses

- Implement Bayes' rule (equation 1.1) as a function in the programming language that you work with (R or Python).
- Call the function "bayesFunction"
- Assume that there are only 2 hypotheses H or !H.
- Write the function without making use of external packages.
- The function gets as input P(H), P(D|H), and P(D|!H).
- The output of the function should be P(H | D).

[1] Sometimes instead of H, textbooks also refer to Model M – to make explicit that you can have different "models of the world" or scientific theories. Etz and Vandekerckhove also use Model. We use Hypothesis here, to make explicit that we are testing hypotheses in an experiment, and to avoid confusion with computational cognitive models that you develop later. Sometimes an index is used to refer to specific hypotheses, such as H_1, H_2, H_3.

Self-test 1.2 (function check):

In the table below are some examples of inputs and outputs that your function should return. Do you get the same values?

Example label	**Input**			**Output**
	P(H)	P(D | H)	P(D | !H)	P(H | D)
A	0.1	0.9	0.3	0.25
B	0.9	0.9	0.3	0.96
C	0.9	0.3	0.9	0.75
D	0.001	0.99	0.02	0.047
E	0.3	0.5	0.5	0.3

Self-check 1.3: conceptual

In the above examples, the posterior is sometimes higher, lower, or equal to the prior. Do you understand conceptually why this is?

For Bayesian statistics, it matters how many hypotheses you specify. Bayes rule only calculates the posterior for hypotheses that are specified. In the previous example, this was some specific hypothesis H (e.g., "a person is a student of the AI program") and !H (e.g., "a person is NOT a student of the AI program").

There are many situations where we might have multiple hypotheses. For example:

- "a person is a student of AI"
- "a person is a student of HCI"
- "a person is a student of computer science"

When you read these sentences, you (probably) look at the content. Bayesian statistics in itself only looks at the numbers. However, you can specify Bayes' equation to consider multiple hypotheses. In essence, having multiple hypotheses changes the way that the denominator of equation 1.1 is implemented. It should now consider all possible alternative hypotheses (e.g., the person being an AI, HCI, computer science student) under the data at hand.

Engineering 1.2: Bayes function with more than 2 hypotheses

- Implement a second version of Bayes' rule (equation 1.1) that can consider multiple hypotheses.
- Call the function "bayesFunctionMultipleHypotheses"
- To do this, give two arguments as input:
    - A vector of prior probabilities of all possible hypotheses.
    - A vector of all likelihood functions of the data given these hypotheses.
- Important: Make sure that:
    - The order in which you give priors matches the order in which you give conditionals
    - The first item in each vector relates to the item of interest (e.g., P("a person is an AI student") and P(Data | "a person is an AI student"))

Self-check 1.4:

1. To test if your function works, you can run the same examples as you used in self-check 1.2 (depending on how you code, you might want to give input in a different way). Do you get the same output as indicated in the table for hypothesis 1? (you should!)
2. Below are some more examples of inputs and output. Do you get the same output? (note: $P(H_1 \mid D)$ gives the output you expect when checking the first hypothesis)
3. For the new examples: Do you understand why the output is higher/lower under some conditions compared to others? Do you have an intuition?
(maybe for these cases it helps to think of the priors as concrete hypotheses such as "this person is a student AI" or "this student is a Computer Science Student" and the conditionals as datapoints such as "likelihood that student passes a test on programming, given that they are an AI student" or "likelihood … given they are a Computer Science student")

Example label	**Input**		**Output**
	Priors	**Conditional probabilities**	**$P(H_1 \mid D)$**
F	(0.4, 0.3, 0.3)	(0.99, 0.9, 0.2)	0.545
G	(0.4, 0.3, 0.3)	(0.9, 0.9, 0.2)	0.522
H	(0.3, 0.3, 0.4)	(0.9, 0.9, 0.2)	0.435
I	(0.3, 0.3, 0.4)	(0.9, 0.2, 0.2)	0.659
J	(0.4, 0.2, 0.2, 0.2)	(0.9, 0.3, 0.3, 0.3)	0.667
K	(0.4, 0.2, 0.2, 0.2)	(0.9 ,0.6, 0.3, 0.3)	0.600
L	(0.01, 0.2, 0.2, 0.2)	(0.99, 0.01, 0.01, 0.01)	0.623

1.b Bayes factor

In the examples so far, you studied the posterior value: the likelihood of a hypothesis given the data. In the case of a binary decision (e.g., H or !H) you can express this as posterior odds:

$$\frac{P(H|D)}{P(!H\,|D)} \qquad \text{(equation 1.2: Posterior odds)}$$

This can help to make claims such as "The hypothesis is 3 times more likely than the alternative hypothesis under this dataset" (e.g., for example "A" that you tested on your BayesFunction).

The posterior odds can be related to how likely two hypotheses were before the experiment, and how likely the data is under an experiment. As follows (see Etz & Vandekerckhove, 2018, for more details):

$$\frac{P(H|\,D)}{P(!\,H|\,D)} = \frac{P(H)}{P(!H)} \times \frac{P(D\mid H)}{P(D\mid !\,H)} \qquad \text{(equation 1.3)}$$

The first term is the posterior odds. What you might have already guessed based on the previous self-tests is that the posterior odds depends on:

The prior odds: P(H) / P(!H) How much more likely is one hypothesis (e.g. “elephants can fly”) compared to another?
Note that this might depend on who you ask – my son will have a different likelihood of “elephants can fly” after watching Disney’s Dumbo then I have. Similarly, different scientists might assign different prior odds to different scientific hypotheses.

The Bayes Factor: P(D | H) / P (D | !H). This factor tells you how informative the data is to distinguish hypotheses. Let’s say again that your hypothesis is that “elephants can fly”, and the data is a counting of the amount of elephant droppings. If you count the elephant droppings in a circus ring (i.e. D = “ at least some elephant droppings are found in a circus ring”) then we might expect that this is the same for when elephants can fly (P(D | H)) compared to when they cannot fly (P(D | !H)).

However, if you count the number of elephant droppings on the top of a roof, then the likelihood of observing elephant droppings will differ substantially between the two hypotheses (i.e., P(D | H) >>> P(D | ! H)).

Another way to think about this in the context of experiments is: how useful is your experiment? Does collecting more data add anything to the scientific outcome and insights? Does the posterior odds depend mostly on the prior odds, or did the experiment add some meaningful insight (Bayes factor).

Bayesian researchers often use the Bayes factor to interpret about how much evidence there is in support of one hypothesis over another. The following heuristic by Jeffreys (1961) is often used to describe the relative confidence in H_1 (e.g, “H”) over the alternative hypothesis H_2 (e.g., “!H”). Note that $BF_{1,2}$ refers to the fraction where P(D | H1) is in the numerator and P(D | H2) is in the denominator.
If these are turned around, you have the $BF_{2,1}$

$BF_{1,2}$	**Interpretation**
>100	Extreme evidence for H_1
30-100	Very strong evidence for H_1
10-30	Strong evidence for H_1
3-10	Moderate evidence for H_1
1-3	Anecdotal evidence for H_1
1	No evidence (i.e., equal size)
1-1/3	Anecdotal evidence for H_2
1/3-1/10	Moderate evidence for H_2
1/10-1/30	Strong evidence for H_2
1/30-1/100	Very strong evidence for H_2
<1/100	Extreme evidence for H_2

<u>Self-check 1.5:</u>
For an experiment the posterior odds of $P(H_1) / P(H_2)$ are 3, the prior odds are 2.
What is the $BF_{1,2}$ and what is the $BF_{2,1}$?
Going by Jeffreys’ heuristic, what type of evidence does the data deliver for the hypotheses?

Engineering 1.3: Bayes Factor

Implement a function called "bayesFactor". The function takes as input 2 vectors:

- A vector of posteriors (e.g., ($P(H_1 \mid D)$, $P(H_2 \mid D)$, $P(H_3 \mid D)$) if there are 3 – but in principle it could be *N* posteriors).
- A vector of priors (e.g., ($P(H_1)$, $P(H_2)$, $P(H_3)$) if there are 3)

It gives as output different Bayes Factors (for different comparisons), see below for examples.

Important: Make sure that:

- The order in which you give posteriors matches the order in which you give priors
- The first item in each vector relates to the item of interest P("a person is an AI student" | Data) and (e.g., P("a person is an AI student")
- When you give input for the priors, that the sum of all priors equals to 1
- The posteriors sum to 1

Self-check 1.6

Here is some example output in R (I use the "print" command to print output. Note that here the first argument is the posterior, the second the prior.):

```
> bayesFactor(c(0.9,0.05,0.05), c(0.2,0.6,0.2))
[1] "BF 1 vs not 1:  36"
[1] "BF 1 vs  2 : 54"
[1] "BF 1 vs  3 : 18"

> bayesFactor(c(0.85,0.05,0.1),c(0.2,0.6,0.2))
[1] "BF 1 vs not 1:  22.6666666666667"
[1] "BF 1 vs  2 : 51"
[1] "BF 1 vs  3 : 8.5"

> bayesFactor(c(0.15,0.35,0.5),c(0.4,0.3,0.3))
[1] "BF 1 vs not 1:  0.264705882352941"
[1] "BF 1 vs  2 : 0.321428571428571"
[1] "BF 1 vs  3 : 0.225"

> bayesFactor(c(0.35,0.15,0.5),c(0.3,0.4,0.3))
[1] "BF 1 vs not 1:  1.25641025641026"
[1] "BF 1 vs  2 : 3.11111111111111"
[1] "BF 1 vs  3 : 0.7"
```

As you can see: what type of evidence my experiment delivers (e.g., anecdotal, moderate, strong, etc) depends on what I compare my hypothesis / model to.

1.C. Power of collecting more data: yesterday's posterior is today's prior

A fundamental idea in science is that we gain more confidence in a particular pattern if the pattern / experiment is replicated. This can be expressed in Bayesian terms. Namely: whatever posterior you get from one particular experiment can be used as a prior in subsequent experiments.

Blackboard question 1: (1 point)

Consider the following case study:

Daryl does an experiment whether people can look into the future[2]. Participants see two curtains, and need to indicate behind which curtain there might be an erotic stimulus. 100 participants take part; on 53.1% (or, proportionally: 0.531) of the trials, the participants correctly guess the curtain with erotic stimuli. A classical t-test confirms that this is higher than chance (50% or 0.50). Daryl concludes that some people can sometimes look into the future.

Answer the following questions in the dedicated answer sheet (see upload link):

A. Calculate the posterior value that people can see in the future under the data. Assume:
   a. The prior probability that people can see in the future is equal to not seeing in the future: 50%
   b. The probability to observe this data under the hypothesis that people can see in the future is 0.531
   c. The probability to observe this data under the hypothesis that people cannot see in the future is 0.52 (this captures the intuition that values around 0.5 are likely when fully guessing)

B. What is the Bayes factor? ($BF_{\text{people can see in the future, not see in the future}}$)

C. Eric-Jan, a more skeptic researcher, thinks the prior probability that people can look into the future is 0.001. What are the posterior odds for such a skeptic researcher?

D. The experiment is replicated three times by different researchers, independently. So, the conditional probabilities are estimated independently. The priors are updated based on the outcomes of earlier experiments, starting with the outcome of the initial experiment. The relevant other data is in the table below. For each experiment give the new posteriors of $P(H_{\text{people see in the future}} \mid \text{Data})$.

E. Using the Bayes Factor, argue briefly whether the data convinces that people can look into the future.

Experiment	**% trials where people correctly guessed future**	**$P(\text{Data} \mid H_{\text{people see in the future}})$**	**$P(\text{Data} \mid !\, H_{\text{people see in the future}})$**
2	47.1%	.471	0.520
3	49.1%	.491	0.65
4	50.5%	.505	0.70

[2] This is based on an article by Daryl Bem that has been highly debated in the scientific community. E.g.:

- Bem, D. J. (2011). Feeling the future: experimental evidence for anomalous retroactive influences on cognition and affect. *Journal of personality and social psychology*, *100*(3), 407.
- Rouder, J. N., & Morey, R. D. (2011). A Bayes factor meta-analysis of Bem's ESP claim. *Psychonomic Bulletin & Review*, *18*(4), 682-689.
- Wagenmakers, E. J., Wetzels, R., Borsboom, D., & Van Der Maas, H. L. (2011). Why psychologists must change the way they analyze their data: the case of psi: comment on Bem (2011).

1.d. Summary of section 1

You have implemented basic versions of Bayesian tests.

1. In Bayesian analyses, the **posterior** probability that a hypothesis (or model) is true, given the data depends on the **prior** probability, and the **likelihood function** that the data could be observed under the hypothesis at hand (versus the alternative hypothesis).
2. Yesterday's posterior is today's prior (i.e., we learn from each observation)
3. As the posterior (also) depends on one's prior assumptions, in Bayesian statistics we often look at the **Bayes Factor** that expresses how much evidence **the data** provided for a specific hypothesis.
4. When comparing models using Bayesian statistics, it makes a difference which models are specified (e.g., only H_1 and H_0; or H_1, H_2, H_3, H_4, H_5)
5. Bayesian analysis can be used to calculate how likely a hypothesis (or model) is, given the observed data.

Section 2: Building probabilistic models (Relative Frequency)

You will now work with two cognitive models that have a probabilistic mechanism at its core, but that differ in the number of parameters (and assumptions) that they have. The model captures phenomena of this paper:

Zhu, J. Q., Sanborn, A. N., & Chater, N. (2020). The Bayesian sampler: Generic Bayesian inference causes incoherence in human probability judgments. *Psychological review*, *127*(5), 719.

2.a. Zhu et al's experiment: why does human judgment deviate from normative expectations?

The premise of the experiment by Zhu et al (2020) is the following apparent contradiction:

1. Many facets of cognition seem to be well described by the rules of Bayesian inference (see the paper for examples). Such reasoning can be considered efficient.
2. Human reasoning is not always "perfect", as humans sometimes seem to follow heuristics and simplifications during reasoning, which do not always align with the predictions of probability theory or Bayesian inference.

In other words: can mistakes and errors arise if the reasoning mechanism is Bayesian? And if so: how? This can be perfectly explored by crafting competing models, and then comparing their predictions with human performance and with each other's predictions. Which model gives the best description of the human data?

To test this idea, an experiment is run (based on Costello and Watts, 2014, Psychological Review) in which participants are asked to give assessments of statements such as the following:

"What is the probability that the weather will be ***cold*** *on a random day in England?"*

The rating was given as integers on a 100-point scale [0,100], which can be transformed into a probability [0.00, 1.00].
In the example above, the word "cold" can be abstractly represented as "A".
Similarly, one can ask about:

- "rainy" ("B" in Zhu et al.'s Figure 6A1 and 6B1 – see also below)
- "windy" ("A" in Figure 6A2 and 6B2)
- "cloudy" ("B" in Figure 6A2 and 6B2)

Apart from simple events (such as A or not A), the experiment also asked about combination statements of "cold" and "rainy", and of "windy" and "cloudy"

Below is a repeat of Figure 6 from Zhu et al. If human participants assigned probability estimate $\hat{P}$ according to the rules of probability theory, then the ratings of statements such as "A or not A" should have value 1, and statements such as "A and not A" should have value 0. If you look at Figure 6A1, you can see that the sum of ascribed probabilities for A1 + not A1 (i.e., sum of values of the first bar and third bar; the authors use "¬" as symbol for "not") is probably not 1. Note that different participants might vary in their estimates, hence there are error bars.

To make this even more explicit, Zhu et al calculated the probabilities of more complex sentences, listed in the Figure below as $\hat{Z}_1$ - $\hat{Z}_{18}$. The definitions of these sentences are given in table 1 in Zhu et al., copied below. If calculated according to the rules of propositional logic and normative probability theory, then all these sentences should have the outcome 0. However, they do not in quite a few cases, while they do in other cases. Interesting! ☺

Self-check 2.1:
Test for yourself:
1. What the value of specific statements should be if the rules of probability theory were followed (e.g.: A or notA = 1; A and not A = 0) by working through a couple of sentences (for example, $\hat{Z}_1$), and *without* using the human data (i.e., work through the logic).
2. That you understand the way that Zhu et al calculated the $\hat{Z}$-values, by adding up the scores by the participants (you can do this roughly, no need to get a precise value) for a couple of sentences (for example, $\hat{Z}_1$).

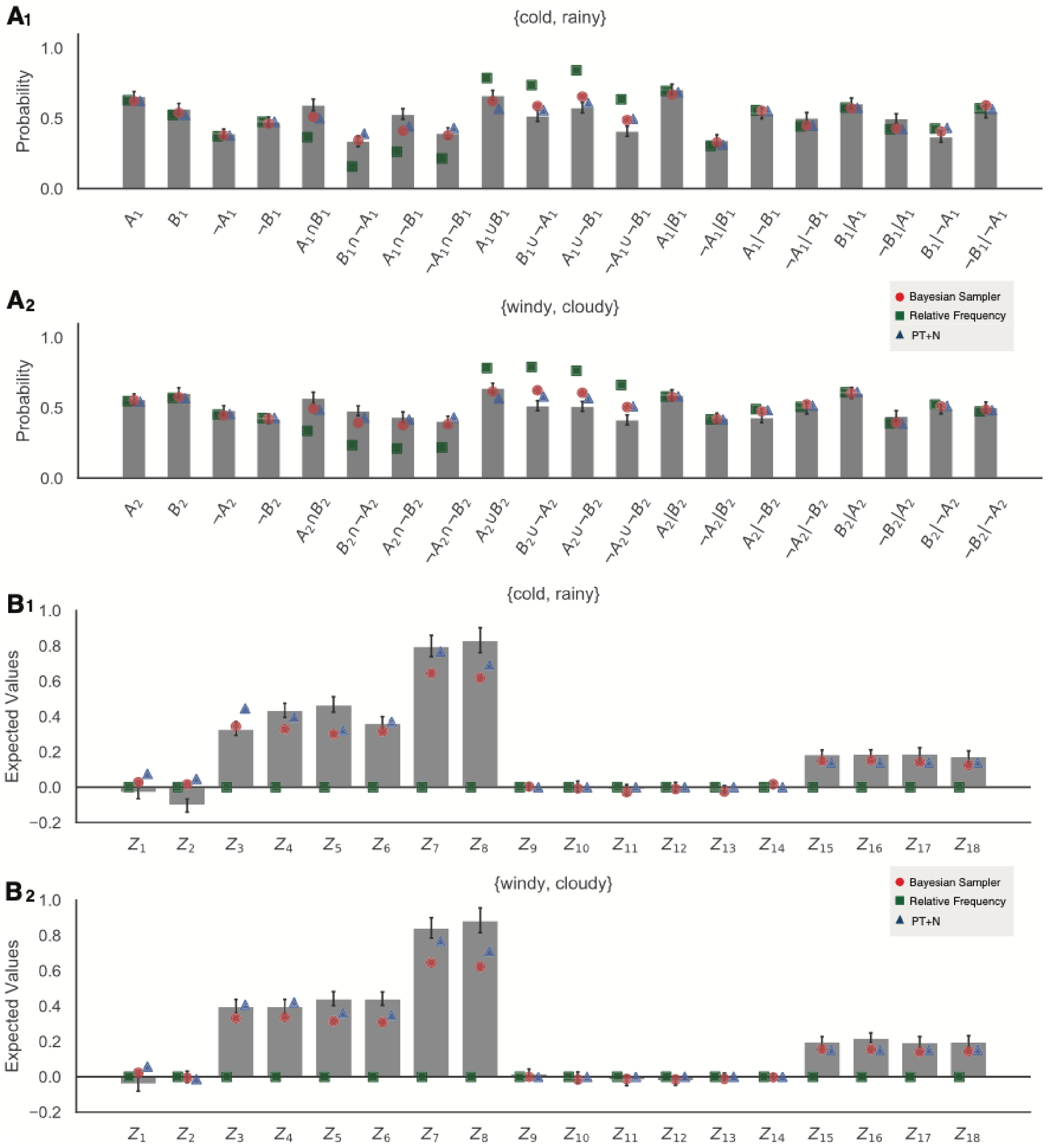


Figure 6. Human probability estimates and model predictions. (A) Mean probability estimates and 95% confidence intervals across participants. The overlaid dots are best-fitting model predictions generated by the most general form of each model (*red dot*: the Bayesian sampler, *green square*: the relative frequency model, and *blue triangle*: the probability theory plus noise model). (B) The mean of the probabilistic identities from $\widehat{Z}_1$ to $\widehat{Z}_{18}$ with 95% confidence intervals across participants. The overlaid dots are best-fitting model predictions for models fit to the mean estimates in (A). See the online article for the color version of this figure.

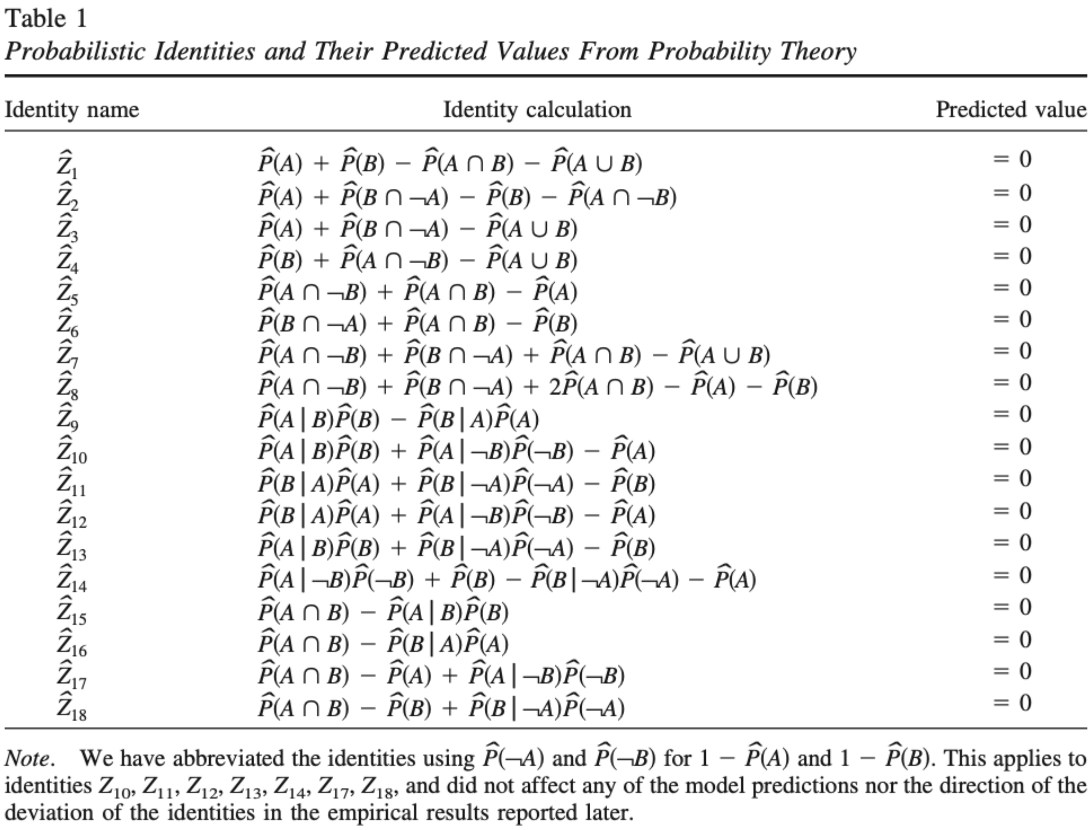


Table 1

Probabilistic Identities and Their Predicted Values From Probability Theory

Identity name	Identity calculation	Predicted value
$\hat{Z}_1$	$\hat{P}(A) + \hat{P}(B) - \hat{P}(A \cap B) - \hat{P}(A \cup B)$	$= 0$
$\hat{Z}_2$	$\hat{P}(A) + \hat{P}(B \cap \neg A) - \hat{P}(B) - \hat{P}(A \cap \neg B)$	$= 0$
$\hat{Z}_3$	$\hat{P}(A) + \hat{P}(B \cap \neg A) - \hat{P}(A \cup B)$	$= 0$
$\hat{Z}_4$	$\hat{P}(B) + \hat{P}(A \cap \neg B) - \hat{P}(A \cup B)$	$= 0$
$\hat{Z}_5$	$\hat{P}(A \cap \neg B) + \hat{P}(A \cap B) - \hat{P}(A)$	$= 0$
$\hat{Z}_6$	$\hat{P}(B \cap \neg A) + \hat{P}(A \cap B) - \hat{P}(B)$	$= 0$
$\hat{Z}_7$	$\hat{P}(A \cap \neg B) + \hat{P}(B \cap \neg A) + \hat{P}(A \cap B) - \hat{P}(A \cup B)$	$= 0$
$\hat{Z}_8$	$\hat{P}(A \cap \neg B) + \hat{P}(B \cap \neg A) + 2\hat{P}(A \cap B) - \hat{P}(A) - \hat{P}(B)$	$= 0$
$\hat{Z}_9$	$\hat{P}(A \mid B)\hat{P}(B) - \hat{P}(B \mid A)\hat{P}(A)$	$= 0$
$\hat{Z}_{10}$	$\hat{P}(A \mid B)\hat{P}(B) + \hat{P}(A \mid \neg B)\hat{P}(\neg B) - \hat{P}(A)$	$= 0$
$\hat{Z}_{11}$	$\hat{P}(B \mid A)\hat{P}(A) + \hat{P}(B \mid \neg A)\hat{P}(\neg A) - \hat{P}(B)$	$= 0$
$\hat{Z}_{12}$	$\hat{P}(B \mid A)\hat{P}(A) + \hat{P}(A \mid \neg B)\hat{P}(\neg B) - \hat{P}(A)$	$= 0$
$\hat{Z}_{13}$	$\hat{P}(A \mid B)\hat{P}(B) + \hat{P}(B \mid \neg A)\hat{P}(\neg A) - \hat{P}(B)$	$= 0$
$\hat{Z}_{14}$	$\hat{P}(A \mid \neg B)\hat{P}(\neg B) + \hat{P}(B) - \hat{P}(B \mid \neg A)\hat{P}(\neg A) - \hat{P}(A)$	$= 0$
$\hat{Z}_{15}$	$\hat{P}(A \cap B) - \hat{P}(A \mid B)\hat{P}(B)$	$= 0$
$\hat{Z}_{16}$	$\hat{P}(A \cap B) - \hat{P}(B \mid A)\hat{P}(A)$	$= 0$
$\hat{Z}_{17}$	$\hat{P}(A \cap B) - \hat{P}(A) + \hat{P}(A \mid \neg B)\hat{P}(\neg B)$	$= 0$
$\hat{Z}_{18}$	$\hat{P}(A \cap B) - \hat{P}(B) + \hat{P}(B \mid \neg A)\hat{P}(\neg A)$	$= 0$

Note. We have abbreviated the identities using $\hat{P}(\neg A)$ and $\hat{P}(\neg B)$ for $1 - \hat{P}(A)$ and $1 - \hat{P}(B)$. This applies to identities Z_{10}, Z_{11}, Z_{12}, Z_{13}, Z_{14}, Z_{17}, Z_{18}, and did not affect any of the model predictions nor the direction of the deviation of the identities in the empirical results reported later.

2.b. Model 1: Relative Frequency

Zhu et al develop and compare 3 cognitive models. In this lab, you will work with 2 of those. The first model is the relative frequency model (see Zhu et al page 721). At its core, this model observes what the situation is and assigns odds based on its direct observation. Zhu et al use the example of throwing a ball at a fair to hit something (coconuts, or cans as you would find on a Dutch "Kermis"). Based on what you observe, you can then assign probabilities. For example, on the first throw does someone hit the coconuts/cans, and do they fall down? And does this happen on a second throw?

The estimated probability of observing a hit in the relative frequency model is then calculated based on how often you've observed the hit/fact (frequency), relative to the total number of observations/throws:

$$\hat{P}_{RF}(hit) = \frac{N_{hit}}{N_{thrown}} \qquad \text{(equation 2.1)}$$

The explanation for why the relative frequency model does not observe "perfect" probability estimates (e.g., in which $\hat{P}$(A) + $\hat{P}$(not A) = 1) is twofold:

1. When drawing examples from memory to base your decision on (i.e., how many hits and how many throws can I remember?) the number of

samples/memories a person/model retrieves might differ between different judgements (i.e., when recalling samples of A versus samples of notA).

2. A high number of samples is needed to get very accurate estimates for probabilities. Therefore, having limited memories can at a minimum lead to rounding errors.

In the code that you use, an evolutionary algorithm (a form of statistical or machine learning) is used to find the best fit for a relative frequency model. The assumption is that human probability estimates are based on observations – but the experimenter (and model) has not observed these. Therefore, these are estimated using free parameters. Per pair or words {A, B} (e.g., {cold, rainy} one needs to estimate the odds of:

- a: “cold, and rainy”
- b: “not cold, and rainy”
- c: “cold, and not rainy”
- d: “not cold, and not rainy”

The authors explain how once you know the first three {a,b,c}, you can calculate the fourth one {d} without having to estimate it. That is, even though the algorithm estimates its value, this can be considered not completely a free parameter.

As there are two scenarios ({cold, rainy} and {windy, cloudy}), the model needs to estimate three parameters per scenario, so has effectively 6 free parameters in total (even though in practice it estimates 4 x 2 = 8 parameters).

The crucial function that is used is “differential_evolution” in Python and “DEoptim” in R. This uses an evolutionary algorithm:

- To **minimize** the value of a **function.** Here: the function calls another function (MSE) that calculates the mean squared error between human observed values and model estimated values. This is a distance measure between model predictions and human data. The smaller MSE (the mean squared error) in the prediction, the better prediction is. So, the function tries to minimize MSE.
- While knowing that parameters have **bounds** (for a, b, c, and d, that they are somewhere between 0 and 100)
- Using an evolutionary process in which each generation has a certain number of **samples**
- While having some **tolerance** in when to call it a good enough fit.

Self-check 2.2

If you want to know more about the evolutionary algorithm, look up the help-files of the function “differential_evolution” in Python and “DEoptim”

If you want, you can also test whether the estimated parameters for d converge to the values that the calculation d = 1 – (a + b+ c) gives

Engineering 2.1: Download and execute code of RF model
Download the participant files and the (R or Python) code from Blackboard. Open the file for the RF model. Make sure you understand what the code does. Once you do, you can execute the code. As it iterates through all the participants, this might take a while (note: you need to place the data in a specific subfolder "all_data" and also need to make subfolder called "fit_results".
Hint: To answer Blackboard question 2, you could consider editing the code such that it only runs on a subset of the data. To answer the question, you also need to know what the function returns exactly (that is: understand the code)

Blackboard question 2 (1 point)
For participant file PrEstExp_811_111418_122039.csv calculate the RF estimate. In your blackboard file, report these three values:
(A) The MSE (round to 5 decimals)
(B) The estimate of the model for the first index/item of the vector 'b' (round to 5 decimals) (note: in Python this would be element '0', in R it would be element '1')
(C) The estimate of the model for the sentence 'A1 and B1' (i.e., "cold and rainy"). (round to 5 decimals). Hint: This is one of the 40 outcomes of statements, not the value of parameter of the model.

2.C. Summary of Section 2

You have implemented a basic probabilistic cognitive model. While doing that:

1. You have observed that **human judgements of probability** sometimes **deviate from normative probability theory**.
2. Therefore, a more detailed **cognitive model is needed** to understand how these deviations might arise.
3. You have applied a combination of **machine learning** (evolutionary algorithm) **and probabilistic modeling**
4. You have calculated **Mean Square Error** as one measure of **model fit**.

Section 3: Building probabilistic models (Bayesian Sampling) and model comparison / selection

3.A. Model 2: Relative Frequency + Bayesian sampling

The second model is the Bayesian sampling model (explained in Zhu et al on various pages including 722 and 727). It works the same as the Relative Frequency model, but has 1 added component for Bayesian sampling. The assumption of the Bayesian sampler is that people don't just try to remember past events (i.e., they do not just estimate "a": "cold and rainy") but also have *prior information.* They integrate this prior information with the estimates they get from the relative frequency estimates.

In the model of Zhu et al., it is assumed that the prior information can be approximated by a symmetric beta distribution. Beta distribution is a probability distribution over probabilities (that is, it is a probability distribution that expresses for any probability value how likely or unlikely that probability value is). A symmetric beta distribution has one parameter, which Zhu et al. call (somewhat confusingly) ß. When the parameter ß equals 1, the beta distribution assigns equal probability to any probability value between 0 and 1 (uniform distribution: outcome of 0 and 1 and everything in between are equally likely). As ß approaches 0, it is more and more likely that either the 0 probability value is true or that the 1 probability value is true, and nothing in between.

The advantage of using beta distribution as a prior is that it is very easy to update it with novel information about hits/misses to arrive at the posterior distribution. We will not provide this formula here, instead, we will only provide the way to calculate posterior mean (the expected value after the prior information was updated with novel information). This is calculated as follows:

$Expectedvalue\left[\hat{P}(hit)\right] = \frac{N_{hit}+ß}{N_{attempts}+2ß}$ (equation 3.1)

You can connect Bayesian sampling with Relative Frequency by including the relative frequency of a statement. This results in the following formula:

$$ExpectedValue\left[\hat{P}(statement)\right] = \frac{N}{N+2ß}\hat{P}(statement) + \frac{ß}{N+2ß}$$

(equation 3.2)

In this formula:

- N represents how many memories / experiences the subject might have recalled (we don't know, but the model tries to estimate/fit this);
  - ß represents the parameter discussed above;
  - $\hat{P}$(statement) is the RF calculation (e.g., a, b, c, or d).

The intuition behind equation 3.2 is that N and ß have a relationship:

- If you are more certain that the value is at the extremes (e.g., as ß approaches 0, and you either have a high probability value or low probability value), then it matters less how many observations (of hits or misses) you have. The value will approximate the estimate of RF ($\hat{P}$(statement)). The real-world example here is that when you know that someone is a good darts player (or player at the fair), you don't have to see them play darts a lot (i.e., N can stay low) to estimate that they can hit bulls-eye. Similarly, if you know that they are a complete novice, you might also be quite confident that they will miss without seeing many observations (i.e., low N).

- If you are less sure about what the value of observation is (i.e., ß approaches 1, you do not know how good the darts player is), then the number of observations start to matter more as it is needed to resolve the uncertainty.

Self-check 3.1: Check the mathematics of the above intuition by filling in values of 0, 0.5, and 1 for ß in the equation 3.2 and seeing what the impact is of small N (e.g., 1 or 2) or bigger N (e.g., 25, 100).

The model estimates again the values for a, b, c, and d. It now *also* estimates ß and N. As the model makes a mean prediction (and not a prediction on every trial), the authors make an argument (see page 730) that the parameters N and ß are not independent, and can be considered as effectively estimating 1 parameter. Therefore the effective number of free parameters becomes 7 (a,b,c for two scenarios as before, plus 1 parameter for N and ß combined).

Engineering 3.1: download and check the code
Download the code for the BS (Bayesian Sampling) model. It is very similar to the RF (Relative Frequency) code. Go through the code so you understand what is going on. N and beta again have bounds. For N this is [1,250] and for ß [0,1].
My (Chris') suggestion is to not yet run the code for al participants, as I want you to add a function for calculating the Bayesian Information Criterion (BIC) first.

3.B. Model comparison and selection: Using BIC to understand the balance between model fit and model complexity

The two models (RF and BS) use an evolutionary algorithm to minimize the error between model prediction and human observation (in other words: to maximize model fit). But how does one select the "best model"? Purely going by the best fit seems unfair, as one model (BS) has more free parameters to find the best fit. Theoretically, having more parameters can make it easier to find a better model fit. Therefore, one would want to weigh in the model's complexity (number of free parameters): when both models have an equal fit score, you might prefer the one with fewer parameters.

Compared to process models (lab 1) – that contained many process steps – the models that are used by Zhu et al have a clear, finite set of parameters. Therefore, the model fit can elegantly be estimated using a Bayesian Information Criterion (BIC) score. Zhu et al use a complete version of the BIC, which can be expressed as:

BIC = n * log(MSE) + log(n) * (parameters + 1) + n * log(2*pi) + n (equation 3.3)

Where:

- *n* is the number of observations per participant. In this experiment each participant rates 40 unique statements (see Figure 6A1 and 6A2) three times. So, in total each participant has *n* = 120 (40 x 3));
- *parameters* is a count of the effective number of free parameters (6 or 7 depending on the model);
- *MSE* is the outcome of the fitting procedure: the minimal MSE that the evolutionary algorithm found.

(note: various textbooks leave out the last two terms as they are the same for each model. However, like Zhu et al, we will include them)
What you can see is that as the MSE value gets lower (i.e., fit gets better), the value of n * log(MSE) becomes more negative and the BIC becomes more negative. However, as the number of parameters increases, the BIC increases again.

Self-check 3.2:

- Check that you understand how the BIC formula makes a trade-off between number of model parameters and fit score.
- When comparing models, would a better model have a higher or lower BIC?
- Zhu et al. make elaborate arguments that the number of effective parameters is small. Now that you know that number of parameters is part of the BIC, do you understand why they want to keep this number low?

Engineering 3.2: add function to calculate BIC
Write a function that calculates the BIC score using the equation 3.3. Your function gets three values as input: n, number of effective free parameters, MSE (see text under equation 3.3 for descriptions of the three parameters).
Use the base-10 log for the logarithmic element in the calculation.

Engineering 3.3: add code to calculate total BIC across all runs
Within the function init_fit, you loop through all participants. For each participant you need to calculate a BIC score individually. Add the scores of all participants to get the sum of the BIC score across all participants.
Implement this step for the Relative Frequency model and the Bayesian sampling model. You can now calculate the total BIC scores for both models, and are now ready to answer Blackboard questions 3A and 3B.

Blackboard question 3 (1 point)

A. Report the total BIC score (score over all participants) for:
   (i) the RF model
   (ii) the BS model
B. Based on your answer to question A, explain briefly which of the two models (Relative Frequency or Bayesian Sampling) you prefer.

3.C. Summary of section 3

You have implemented a **Bayesian Sampling** model as a competing model of human probability estimation.

1. The model combines a frequency estimate of likelihood of event (like the Relative Frequency model) with **prior** information. This makes it Bayesian
2. **Likelihood** information is captured using parameter N (how many samples do I need?) and beta (how confident am I that the outcome will be in the extremes of 0 or 1?)
3. The **shape / values of the** prior affects the model outcome.
4. To compare two models with a different number of parameters, other measures are needed. You applied the **BIC to trade-off model fit with model complexity**.

Section 4: Exploration of the interaction between N and ß

In section 3 it was described that N and ß have an intricate relationship. However, the procedure by which final values for ß and N were derived was bottom-up, data driven through model fitting. Alternatively, to make more explicit what the relationship is, one can systematically set the parameters to specific combinations (i.e., not optimize them through model fitting) and see how good the value is of any specific combination of parameters. You will do that here.
You will also explore what the effect is of values of ß that fall outside of the range explored in Zhu et al.'s work (where the bounds are [0,1])

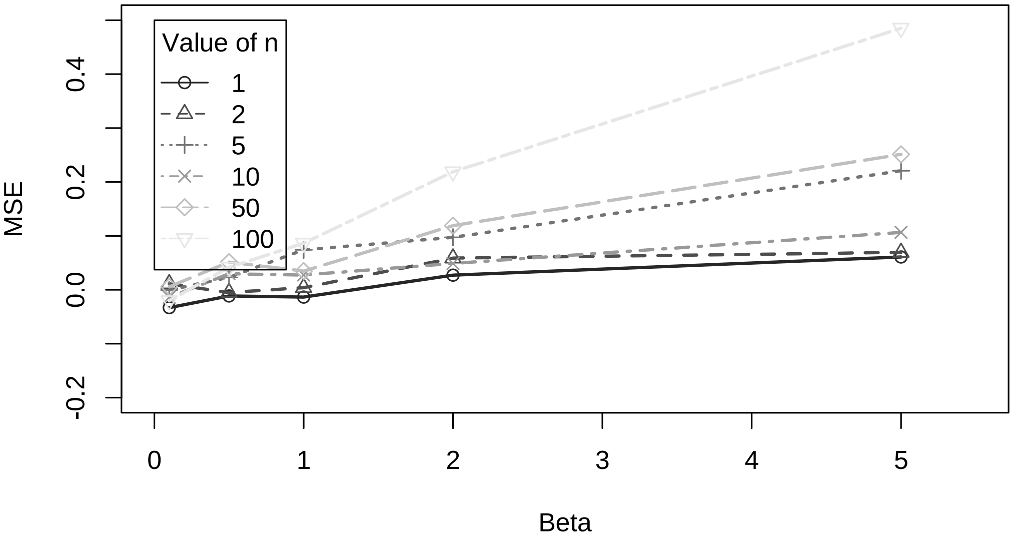
4.A. Systematic analysis of relation between N and ß

Engineering 4.1: systematically explore the parameter space of N and ß
Make a copy of your file that runs the code for the Bayesian Sample model and rename that to "model_fitting_BS_MSE_forFixedValues". Within that new file, change the code such that the optimization procedure does not search what values N and ß should have (i.e., does not use them as part of the fitting procedure). Rather, let the code iterate through fixed values for N, namely: 1, 2, 5, 10, 50, 100. Then also let the model iterate through fixed values for ß, namely: 0.1, 0.5, 1, 2, 5.
This gives a total of 6 x 5 = 30 models to explore.
With these models, run the code such that it fits the parameters a, b, c, and d (as before). For each model run, once it is over, store the MSE values.
As this code might take a while, you are allowed to run it for a subset of participants (instead of all 84 participants), for example for the first 10 or first 20 participants.
With the outcomes, make a plot that shows on the horizontal axis the value for ß, and on the vertical axis the MSE. Use different lines for different values of N.
Below is a graph that uses fake data, but illustrates what stylistically your graph should look like (the shape and data might be very different though!). You are now ready to answer Blackboard question 4 A and B.

Hint 4.1:

- You calculate the MSE per participant, but you report a value across multiple participants. Please use the MEAN over all participants (i.e., if you run the code for 10 participants, then you calculate 10 MSEs, and then calculate the mean and report that in the Figure). As there are multiple conditions (i.e., multiple combinations of N and beta), you calculate multiple such means.

- You loop over N and beta. My recommendation is to do this looping within the "init_fit_BS" function. Make sure to assign the local values of N and Beta to GLOBAL variables, such that you can access these values in other functions. (the original refer to "params" in the function "generativeModel_BS" is then changed for beta and N. You do not include these in "params" anymore, as you have specific values (and not a range of values). Instead, you use the value that you had set as a global variable)



<u>Blackboard question 4 (1 point)</u>

A. Write down for how many participants you ran the analysis (e.g., 10, or 20). You do not get points for this subquestion – but I need to know your answer.
B. Copy the plot that you produced and paste it in the answer sheet. The plot should match in style to the plot above (note: the values in my plot are fake and made-up, so you might have to choose different values on the y-axis).
C. Explain the figure. Specifically, describe the pattern of how N and ß affect MSE.
D. There is something systematic in the relationship between N and ß, and how this influences model fit. Explain how this relationship might arise (e.g., why is the fit better for specific combinations of N and ß?)

When you are done with your questions, make a PDF file from your answer sheet and have 1 group member upload it on behalf of the entire group. If you are up for a challenge, consider completing the bonus questions.

4.B. Summary of Section 4

You have used a **systematic approach to investigate the impact of parameter choice** on model fit. Such systematic (perhaps: theory-driven, top-down) approaches are complementary to (data-driven, bottom-up) approaches that fit the model. The downside of bottom-up approaches is that your algorithm might get stuck in particular patterns (such as local maxima), and you might not discover that there is a systematic relationship. By purposefully exploring a wider space, you can explicitly test whether such relationships exist.

Bonus question (at most 0.3 points)

If you are up for a challenge, you can work on this Bonus assignment that gives at most 0.3 points, but will be marked strictly.

Zhu et al (2020) argue that the number of effective free parameters is 6 for the Relative Frequency model and 7 for the Bayesian sampling model. The number of parameters does impact the BIC score. What would this be if other choices were made?

Calculate the BIC scores in six cases (note: for 2 you have already calculated it):

- The Relative Frequency model that assumes number of parameters = 6
- The Relative Frequency model that assumes number of parameters = 8 (6+2 for a, b, c, d x 2)
- The Bayesian Sampling model that assumes number of parameters = 7
- The Bayesian Sampling model that assumes number of parameters = 8 (6 + N + ß instead of 1 component for BS)
- The Bayesian Sampling model that assumes number of parameters = 9 (a, b, c, d x 2 + 1 BS component)
- The Bayesian Sampling model that assumes number of parameters = 10 (a,b,c,d x 2 + component for N + component for ß)

In your report hand in the following:

1. A screenshot (or screenshots) of the most critical part(s) of the code that is needed for calculating these values (i.e., not all code, but the part(s) where the most critical step is taken).
2. A short explanation what this code does, so I can understand what you did.
3. A table with values, formatted as below. Please round your BIC scores to a meaningful number of decimals.
4. Explain under what (parameter) conditions you would now prefer the RF model or the BS sampling model

In your calculations, please use the log-10 as logarithm.

Model class	Free parameters	Total nr parameters	BIC
RF	a,b,c x 2	6	
RF	a,b,c,d x 2	8	
BS	a,b,c x 2 + 1 BS	7	
BS	a,b,c x 2 + beta + N	8	
BS	a,b,c,d x 2 + 1 BS	9	
BS	a,b,c,d x 2 + beta + N	10	

References

Bem, D. J. (2011). Feeling the future: experimental evidence for anomalous retroactive influences on cognition and affect. *Journal of personality and social psychology*, *100*(3), 407.

Costello, F., & Watts, P. (2014). Surprisingly rational: probability theory plus noise explains biases in judgment. *Psychological review*, *121*(3), 463.

Etz, A., & Vandekerckhove, J. (2018). Introduction to Bayesian inference for psychology. *Psychonomic bulletin & review*, *25*(1), 5-34. https://link.springer.com/article/10.3758/s13423-017-1262-3

Jeffreys, H. (1961). *Theory of probability* (3 ed.). Oxford, UK: Oxford University Press.

Rouder, J. N., & Morey, R. D. (2011). A Bayes factor meta-analysis of Bem's ESP claim. *Psychonomic Bulletin & Review*, *18*(4), 682-689.

Wagenmakers, E. J., Wetzels, R., Borsboom, D., & Van Der Maas, H. L. (2011). Why psychologists must change the way they analyze their data: the case of psi: comment on Bem (2011). *Journal of Personality and Social Psychology, 100*(3), 426–432

Zhu, J. Q., Sanborn, A. N., & Chater, N. (2020). The Bayesian sampler: Generic Bayesian inference causes incoherence in human probability judgments. *Psychological review*, *127*(5), 719. https://doi.org/10.1037/rev0000190