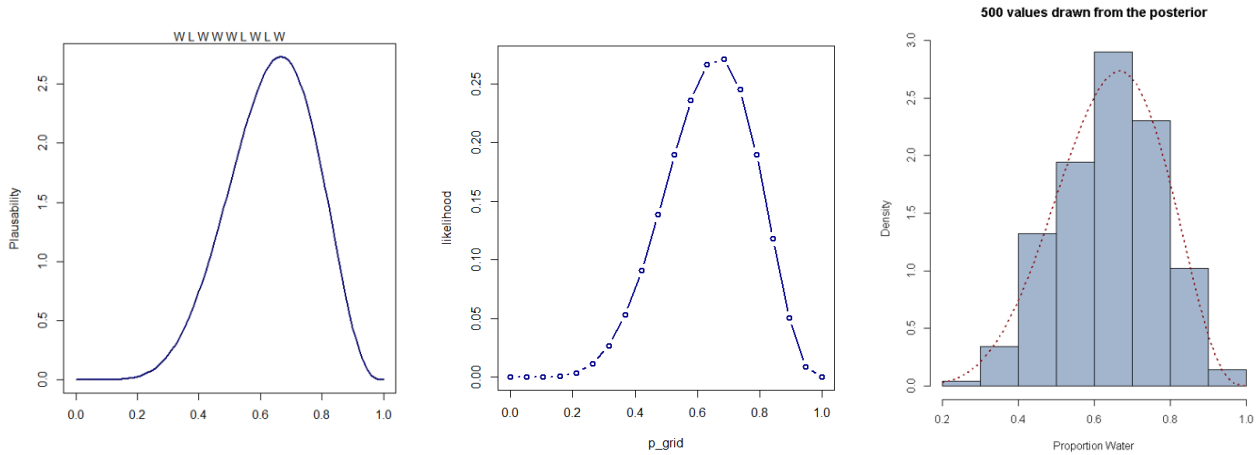
PSY4320 - Introduction to Bayesian statistics

Nikolai Czajkowski

Quick review:

- What characterizes *Bayesian statistics*, and how does it differ from *frequentistic statistics*?
- What is a *posterior distribution*, and how do you use it?
- How do we find a *posterior distribution*?

How do you find the posterior?



Left: Analytic solution (usually impossible).

Middle: Grid approximation (difficult when many parameters).

Right: Monte Carlo Methods; approximate the posterior by simulation (comptationally demanding, but tractable).

Analytic approaches to finding exact expressions for the posterior can be difficult (or impossible) because it requires computing the *evidence*.

$$\underbrace{p(\theta|D)}_{\text{posterior}} = \underbrace{p(D|\theta)}_{\text{likelihood}} \underbrace{p(\theta)}_{\text{prior}} / \underbrace{p(D)}_{\text{evidence}}$$

For a finite set of values of theta functions this requires finding the value for the sum:

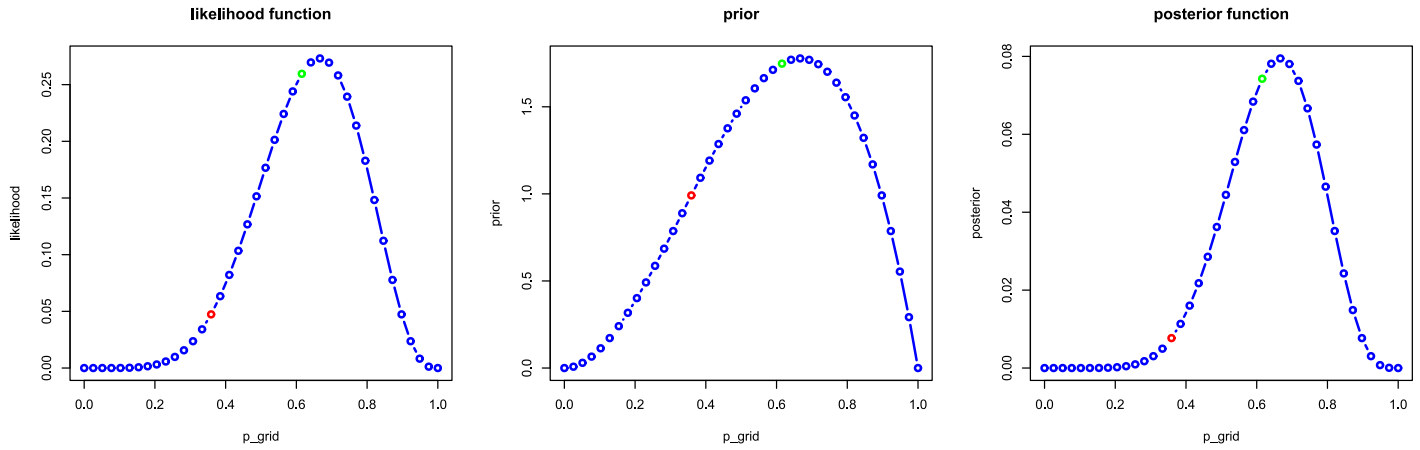
$$p(D) = \sum p(D|\theta)p(\theta)$$

For *continuous* functions this requires finding the value for the integral:

$$p(D) = \int p(D|\theta)p(\theta)d\theta$$

For real problems, finding the integral is usually impossible.

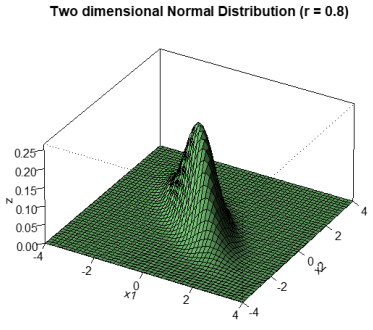
Finding the posterior: Grid approximation



$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

1. Divide parameter range into discrete points.
2. Find the posterior at each grid point by multiplying the likelihood and the prior at that point.
3. Ensure that the posterior sums to 1 by dividing with the sum across all grid points.

Posterior by grid approximation usually impossible



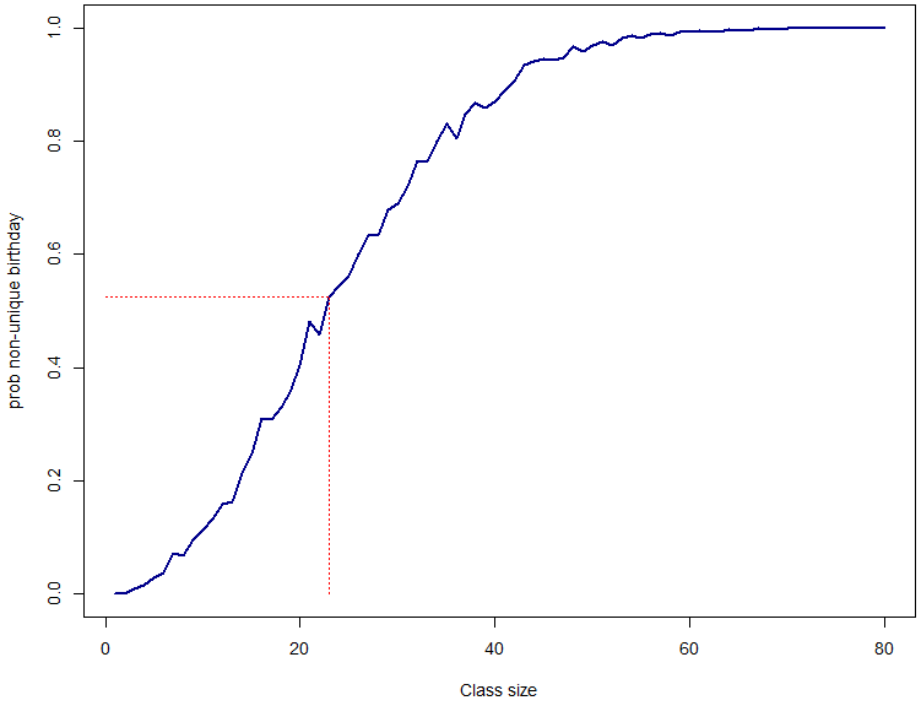
- The prior is specified on a dense grid of points spanning the range of θ values.
- With one parameter with a finite range, approximation by a grid can be a useful procedure. But what if we have several parameters?
  - With six parameters, parameter space is six-dimensional, and involves the joint distribution of all combinations of parameter values.
  - If we set up a grid on each parameter that has 1,000 values, then the six-dimensional parameter space has $1,000^6$ = 1,000,000,000,000,000,000 combinations that must be evaluated.

Markov Chain Monte Carlo Methods

Seminar 3: Terms you should be able to explain to someone else

- MCMC methods
- Random walks
- Trace plots
- Burn in / Thinning
- Autocorrelation

How many kids does there have to be in a school class for it to be more likely that two of them have the same birthday, than that none of them have the same birthday?



Monte Carlo methods are computational algorithms that rely on repeated random sampling to obtain numerical results.

Samples from the unknown

Imagine the exact form of the posterior was unknown, but you were given these values drawn from the posterior distribution:

```
13.86 10.55 7.74 11.60 9.98 13.54 11.57 10.68 6.76 4.05
9.21 9.77 4.32 14.52 7.20 11.23 11.76 12.09 13.54 11.63
```

How/what could you learn about the posterior?

Some background: Computers and random numbers

- Many statistical/mathematical platforms can provide sequences of random numbers drawn from different distributions.
- Computers are deterministic systems, and can't generate numbers that are truly random.
- However, computers can generate sequences of in the correlation between adjecent values are virtually zero, and thus appears to be random.
- Such sequences can then be used to approximate more complex probability distriubution through various algorithms.

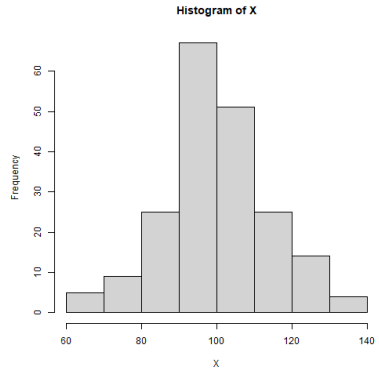
Intro to random numbers in R

```
set.seed(100)
X ← rnorm(200, mean=100, sd=15)

head(X, 10)
```

```
##  [1]  92.46711 101.97297  98.81624 113.30177 101.75457 104.77945  91.27314
##  [8] 110.71799  87.62111  94.60207
```

```
hist(X)
```




```
mean(X)
```

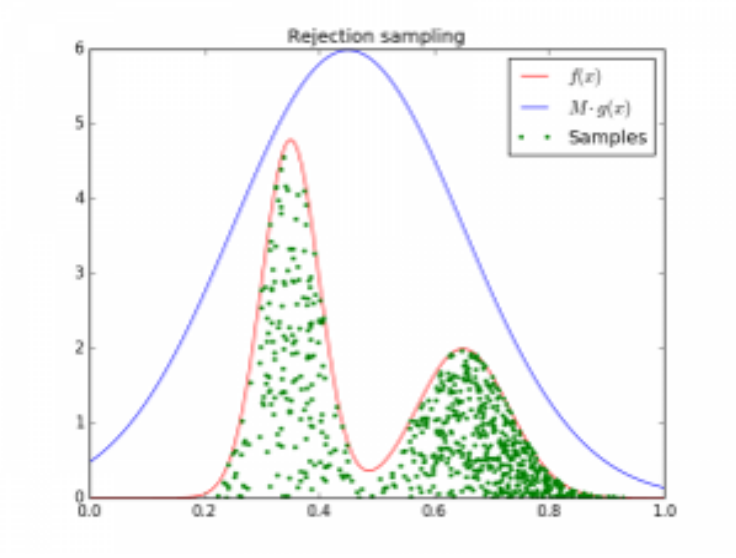
```
## [1] 100.1054
```

Rejection sampling

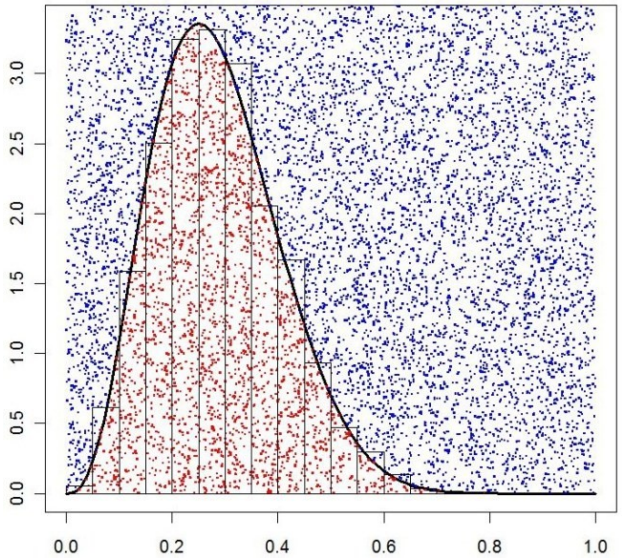
- This is one approach to generate random values from more complex distributions.

Approach:

1. Generate *candidate* values using a *proposal distribution* (blue).
2. Accept candidate with probability equal to the probability of the desired distribution at that parameter value, reject otherwise.



Rejection samploing can be inefficient



Unless the proposal distribution is taylored to fit the desired distribution well, rejection sampling can require you to discarding most values.

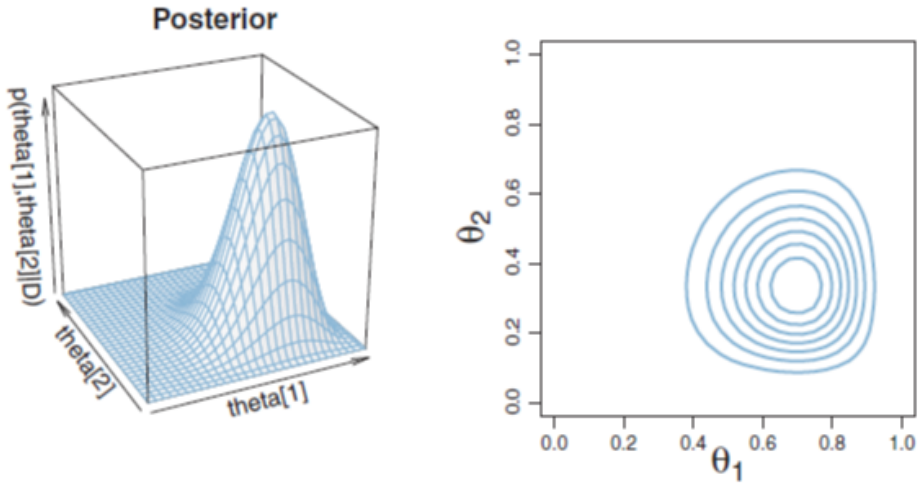
- To get the valdig draws from the target distribution (red), I need to discard all the blue values.
- The rejection rate can be much higher in multivariate distriutions, leading to a lot of

Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo (MCMC): approch that allows us to draw random values from complex distributions (here the posterior).

- We can approximate the shape of the posterior, and calculate statistics (mean, median variance), without knowing the exact mathematical expression for the distribution.
- MCMC algorithms and powerful computer hardware now allow us to conduct Bayesian data analysis that would be impossible 30 years ago.
- The cost of MCMC methods is that analyses can take long, (hours, days), and the procedure may fail to generate random numbers.
    - We need to run diagnostic tests to ensure that we have valid draws from the posterior.

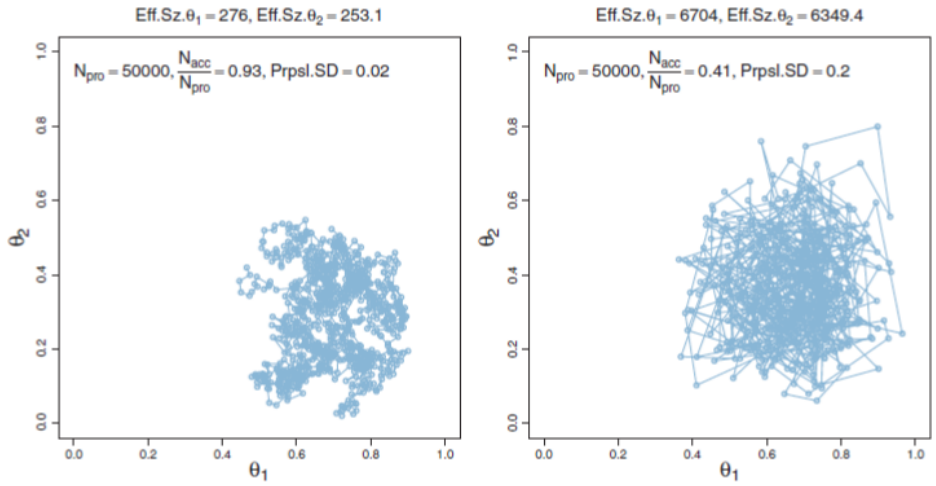
Random walks



- A random walk describes a path that consists of a succession of random steps on some mathematical (here parameter) space.
- We can refer to the movement through parameter space as a *chain*.
- The process will result in a sequence of (apparently) random values drawn from the target distribution
- Important classes are *Metropolis algorithm, Gibbs sampler, Hamiltonian MCMC*

(kruschke, 2015)

https://chi-feng.github.io/mcmc-demo/app.html

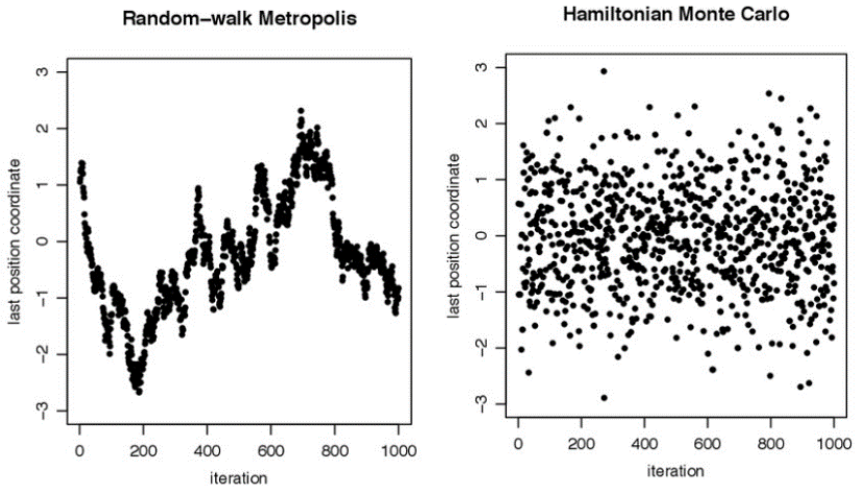


Above: Two different widths for the proposal distribution.

- Left: Too small jumps yield very similar values, and relatively little new information. The effective sample size (ESS) of the chain is very small.

- The left and right panels would eventually converge to an identical and highly accurate approximation to the posterior distribution, but the left one is a lot less efficient, i.e. the one on the right can approximate the distribution well with much fewer draws.

Some MCMC algorithms are more efficient



- Ideally we would like uncorrelated random draws from our target (posterior) distribution.
- However, sucessive steps are *much* more correlated in Metropolis than Hamiltonian MCMC.

In summary; Characteristics of a good MCMC approach

1. The values in the chain must be representative of the posterior distribution.

- Not unduly influenced by the arbitrary initial value of the chain.
- Should fully explore the range of the posterior distribution without getting stuck.

2. The chain should be of sufficient size so that estimates are accurate and stable

- the estimates of the central tendency (such as median or mode), and the limits of the 95% HDI, should not be much different if the MCMC analysis is run again.

3. The chain should be generated efficiently, with as few steps as possible

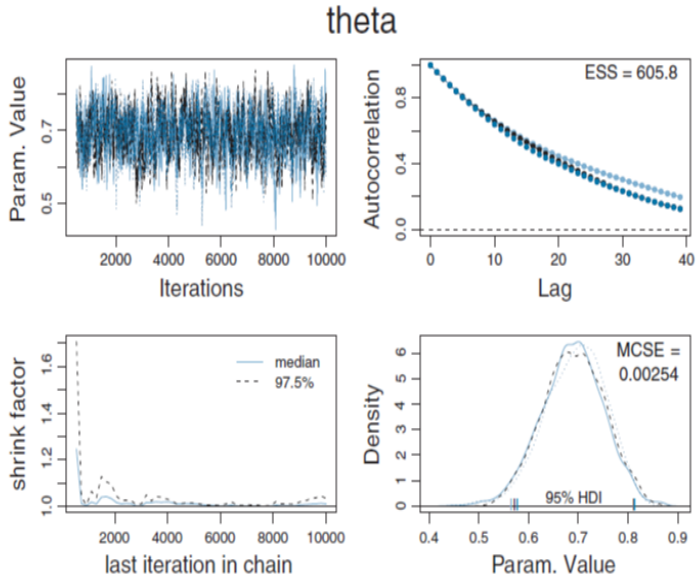
- Should not require vast amounts of time or computing power.

Diagnostic plots

The extent to which a MCMC chain has generated a valid set of random values from the target (posterior) distribution can be evaluetd with diagnostic plots.

Figures on the right illustrate healthy process:

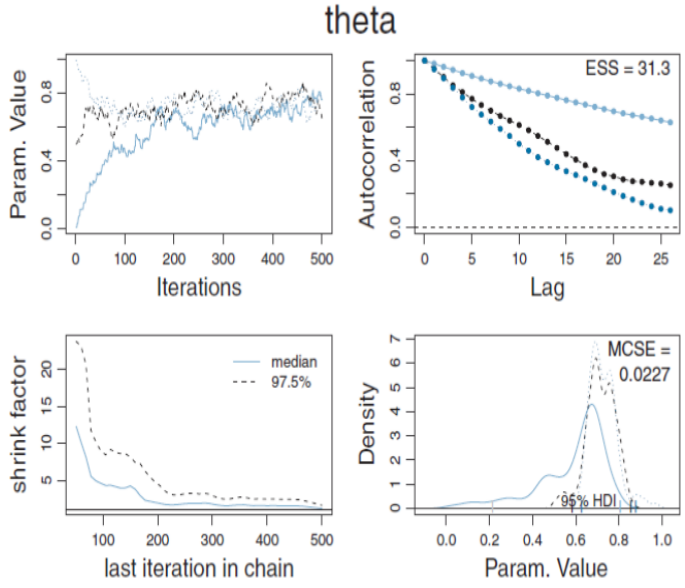
- Traceplot (upper left) looks like a hairy cattepillar.
- Autocorrelation drops similarly for all chains.
- Posterior plot does not look too weird, and similar for all chains.



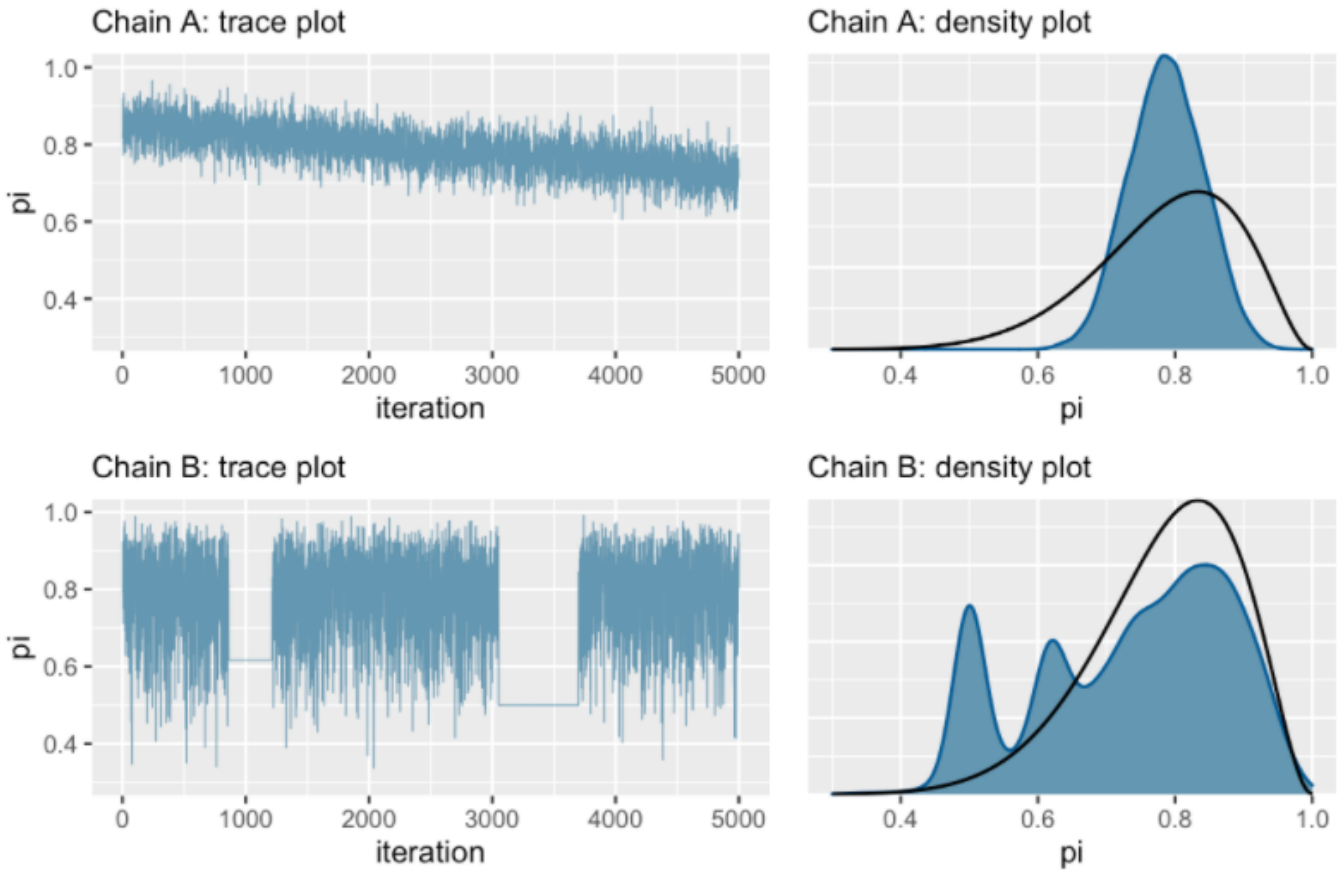
Problematic diagnostic plots

The figures on the right illustrate an unhealthy process:

- Traceplot reveals that it takes a few hundred steps for the three chains to converge to the same region of the parameter.
    - Should be excluded from the sample because they are not representative.
- The preliminary steps, during which the chain moves from its unrepresentative initial value to the modal region of the posterior, is called the burn-in period.



More example of problematic plots



Comparing paralell chains (1)

```
# Density plots of individual chains
mcmc_dens_overlay(bb_sim, pars = "pi") +
  ylab("density")
```

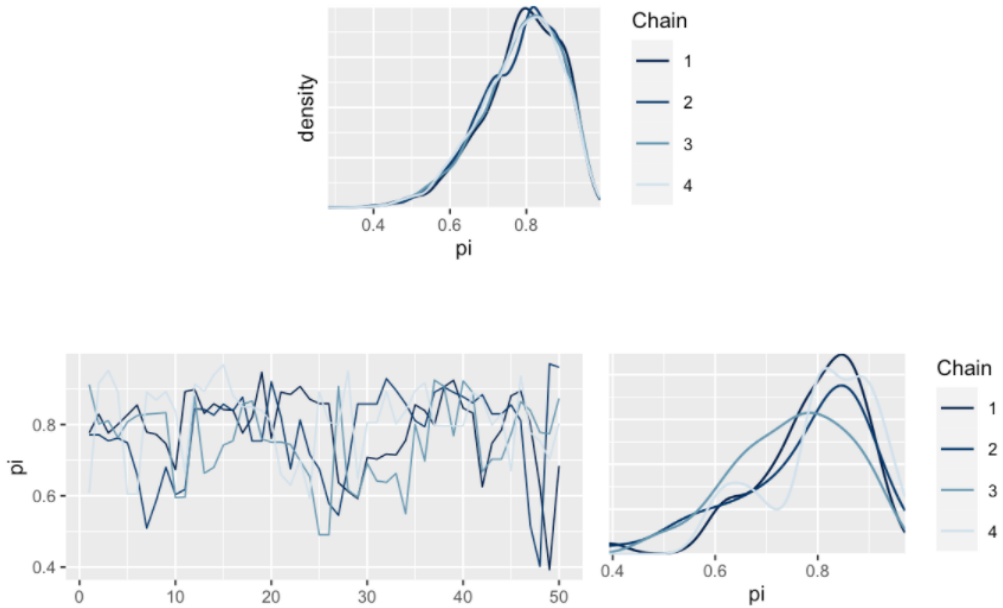
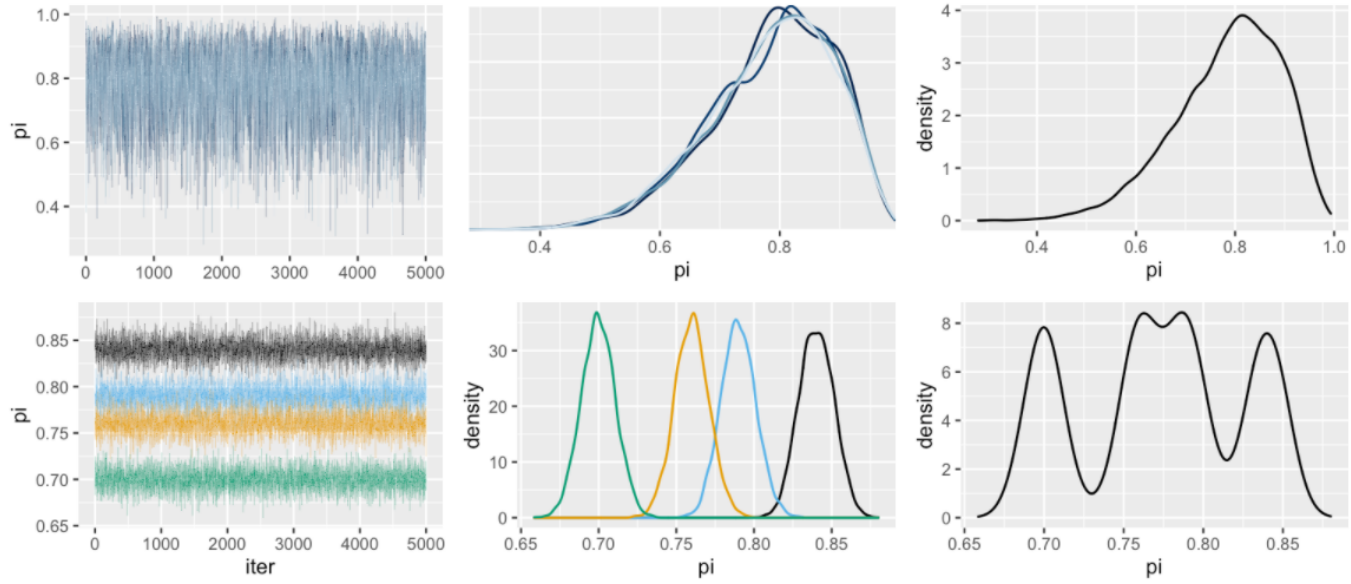


FIGURE 6.14: Trace plots and density plots of the four short parallel Markov chains for π, each of length 50.

Comparing paralell chains (2)



R-HAT (2)

R-hat

Consider a Markov chain simulation of parameter θ which utilizes four parallel chains. Let $\mathrm{Var}_{\mathrm{combined}}$ denote the variability in θ across all four chains combined and $\mathrm{Var}_{\mathrm{within}}$ denote the typical variability within any individual chain. The R-hat metric calculates the ratio between these two sources of variability:

$$\text{R-hat} \approx \sqrt{\frac{\mathrm{Var}_{\mathrm{combined}}}{\mathrm{Var}_{\mathrm{within}}}}.$$

Ideally, $\text{R-hat} \approx 1$, reflecting stability across the parallel chains. In contrast, $\text{R-hat} > 1$ indicates instability, with the variability in the combined chains exceeding that within the chains. Though no golden rule exists, an R-hat ratio greater than 1.05 raises some red flags about the stability of the simulation.

How many samples do you need?

The larger the sample, the more stable and accurate (on average) will be the estimates of the central tendency and HDI limits.

How many samples are necessary?

- For aspects of the distribution that are strongly influenced by dense regions, such as the median in unimodal distributions, the ESS can be modest.
- For aspects of the distribution that are strongly influenced by sparse regions, such as the limits of the 95% HDI, the ESS needs to be large (10.000 recommended).

Thinning: In thinning, only every k'th step in the chain is stored. This reduces autocorrelation. However, it is only really necessary if storing the full original chain would take too much computer memory, or if subsequent processing of the full original chain would take too much time.

Stan

- Stan is program that can provide you with random values drawn from a complex (posterior) distribution.
  - Created by Andrew Gelman, and a 34 person development team.
  - Named after Stanislaw Ulam, pioneer of Monte Carlo methods

- Stan uses hamiltonian Monte Carlo (HMC) for generating Monte Carlo steps.
  - For large data sets or complex models, Stan can provide solutions when other software packages fail (bugs/Jags that use Gibbs sampling).
  - HMC uses a proposal distribution that changes depending on the current position. HMC figures out the direction in which the posterior distribution increases, called its gradient, and warps the proposal distribution toward the gradient.
  - Gibbs samplers use a symmetric proposal distribution, and can end up proposing draws that will nearly all be rejected.

Setting up Stan on your system

- On windows, Stan requires you to install *Rtools*
  - Windows: https://cran.r-project.org/bin/windows/Rtools/
  - The necessary software (c++ comiler) should be installed as default on Mac/Linux.

- In Rstudio, you must install and load the rstan package

```
install.packages("rstan")
library (rstan)
```

- For a detailed description in how to set up Stan on your system: https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started

Stan model structure

```
data{ ... declarations ...
  }

parameters{ ... declarations ...
  }

model{ ... declarations ... statements ...
  }
```

Structure of a Stan program

- **Data block:** where you declare the data types, their dimensions, any restrictions (i.e. upper = or lower = , which act as checks for Stan), and their names. Any names you give to your Stan program will also be the names used in other blocks.
- **Parameters" block:** This is where you indicate the parameters you want to model, their dimensions, restrictions, and name. For a linear regression, we will want to model the intercept, any slopes, and the standard deviation of the errors around the regression line.
- **"Model" block:** This is where you include any sampling statements, including the "likelihood" (model) you are using. The model block is where you indicate any prior distributions you want to include for your parameters. If no prior is defined, Stan uses default priors with the specifications uniform(-infinity, +infinity). You can restrict priors using upper or lower when declaring the parameters (i.e. lower = 0> to make sure a parameter is positive). You can find more information about prior specification here.

https://ourcodingclub.github.io/tutorials/stan-intro/

Binomial example

```
# STEP 1: DEFINE the model
bb_model <- "
  data {
    int<lower = 0, upper = 10> Y;
  }
  parameters {
    real<lower = 0, upper = 1> pi;
  }
  model {
    Y ~ binomial(10, pi);
    pi ~ beta(2, 2);
  }
"
```

```
# STEP 2: SIMULATE the posterior
bb_sim <- stan(model_code = bb_model, data = list(Y = 9),
               chains = 4, iter = 5000*2, seed = 84735)
```

Regression example

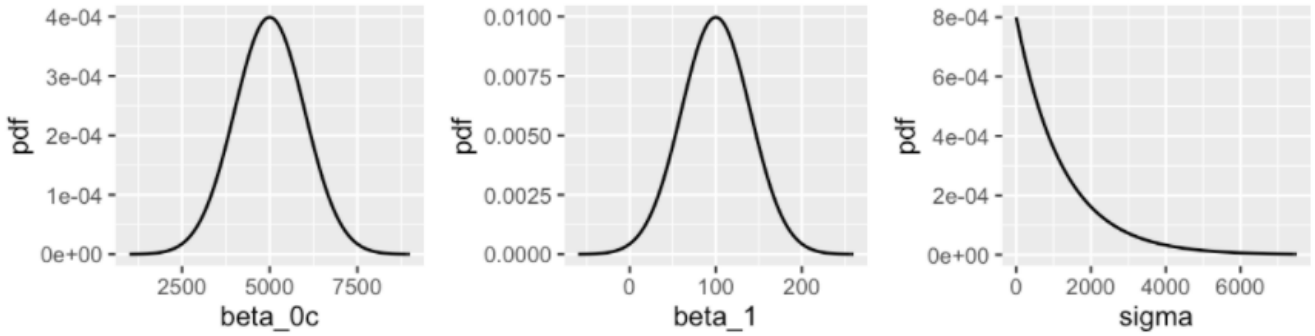
What is the likelihood in a linear regression model?

What is the likelihood in a linear regression model?

$$Y_i|\beta_0, \beta_1, \sigma \overset{ind}{\sim} N\left(\mu_i, \sigma^2\right) \quad \text{with} \quad \mu_i = \beta_0 + \beta_1 X_i. \tag{9.3}$$

What is the likelihood in a linear regression model?

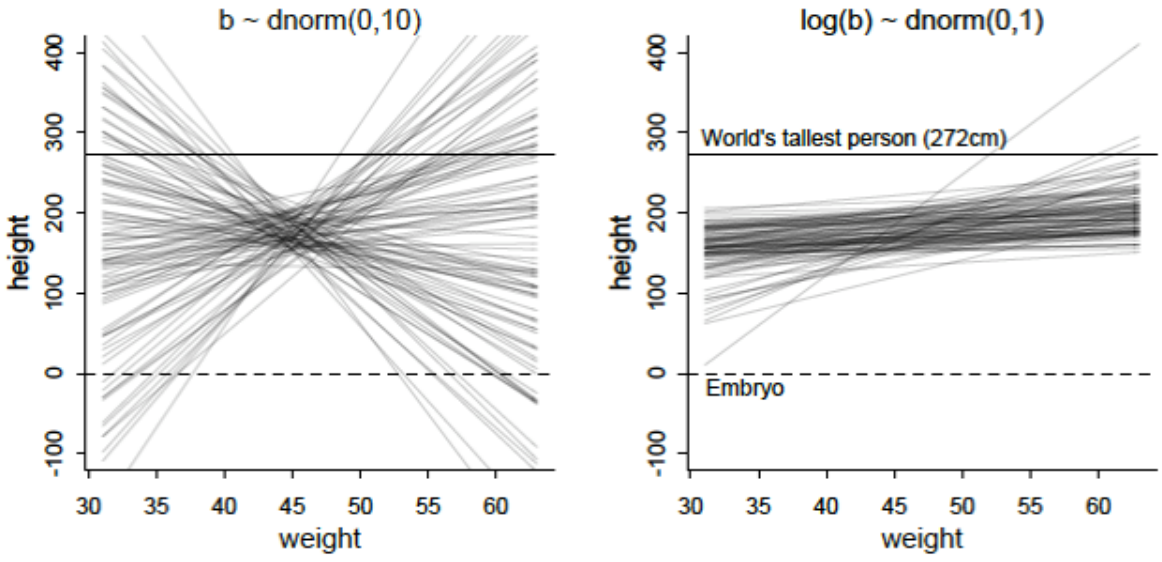
$$
\begin{aligned}
Y_i|\beta_0, \beta_1, \sigma &\overset{ind}{\sim} N\left(\mu_i, \sigma^2\right) \quad \text{with} \quad \mu_i = \beta_0 + \beta_1 X_i \\
\beta_{0c} &\sim N\left(5000, 1000^2\right) \\
\beta_1 &\sim N\left(100, 40^2\right) \\
\sigma &\sim \text{Exp}(0.0008).
\end{aligned} \tag{9.7}
$$



Bivariate regression in Stan?

```
# STEP 1: DEFINE the model
stan_bike_model <- "
  data {
    int<lower = 0> n;
    vector[n] Y;
    vector[n] X;
  }
  parameters {
    real beta0;
    real beta1;
    real<lower = 0> sigma;
  }
  model {
    Y ~ normal(beta0 + beta1 * X, sigma);
    beta0 ~ normal(-2000, 1000);
    beta1 ~ normal(100, 40);
    sigma ~ exponential(0.0008);
  }
"
```

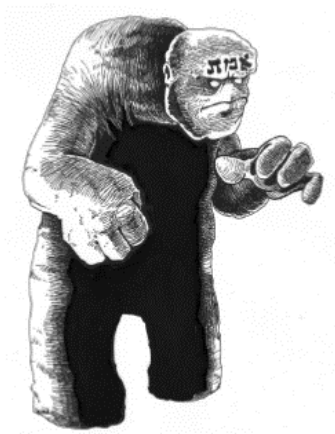
Prior predictive checks

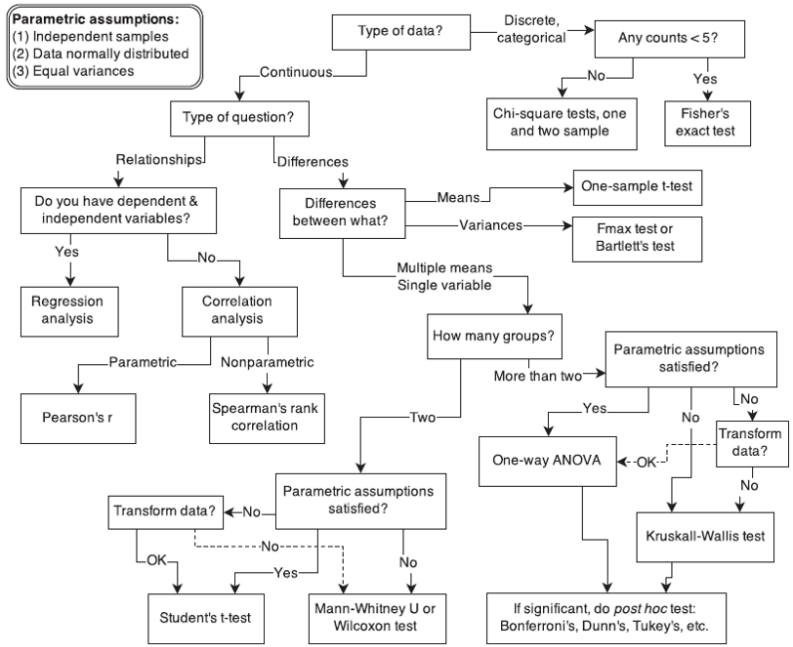


- When you have several parameters in your model, it may be difficult to foresee what the expected values based on their collective impact will be.
- A prior predictive simulation means simulating predictions from a model, using only the *prior distribution*.
    - This is very useful for understanding the implications of a prior.

- Prior predictive checks have become increasingly popular in bayesian analyses.

Reviewing the benefits of a Bayesian approach



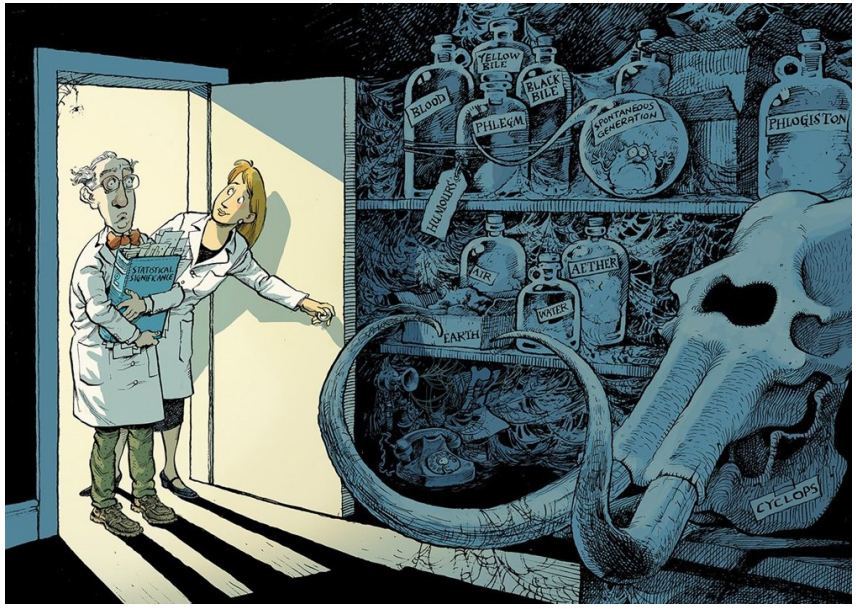


Traditional statistical methods

- makes it difficult to grasp the unified nature of statistical methods.
- lacks a single unified method of building, refining and critiquing statistical models.

(McElreath, 2020)

2. Avoid null-hypothesis significance testing



- Bayesian approach allows us to avoid «*folk popperism*» (falsify straw-man hypotheses), and *stargazing* (searching for *'s in the output)

- Instead of falsifying null models, compare meaningful models.
    - Through the *Bayes factor*, Bayesian inference has a formal framework for comparing non-null models.

3. Naturally incorporate known information into the model

- Bayesian methods provides a principled way of combining new evidence with prior beliefs, through the application of Bayes' rule.
    - Contrast this with frequentist inference, which relies only on the evidence as a whole, with no reference to prior beliefs.
- As a result, Bayesian methods will typically produce stronger inferences from the same data.

- Frequentist formulation of p-values and confidence intervals is rarely fully understood correctly.
  - When told that H is rejected at the 5% level, this is almost universally interpreted as saying that there is only a 5% chance that H is true.
  - Similarly, a frequentist confidence interval is nearly always interpreted as a Bayesian credible interval.

- Bayesian interval estimates have a clearer and more direct interpretation than classical confidence intervals.
  - That is, we can directly conclude that a parameter falls in some interval with some probability.

5. Richer results than p-values

- Bayesian approach allows more detailed summaries concerning parameters.
  - Not simply obtain maximum likelihood estimate and standard error.
- In the posterior we have an entire distribution that can be summarized using various measures (e.g., mean, median, mode, and interquartile range).

6. Flexibility

- Classical inferences may be valid under certain assumptions, but what if assumptions don't hold?
    - Mention the problem in the limitation section?
- Typically, you have no way of dealing with a violation of an assumption, as you don't know how this will impact the sampling distribution.
- Bayesian methods make it much easier to adjust your model to handle such violations, or in other ways.

7. Results are valid for any sample size

- No reliance on test statistics whose sampling distributions are only asympotically known.
    - E.g. in classical statistics, the sampling distribution in t-test is approximately t-distributed if not, as sample grows.

- In Bayesian statistics, results hold for any sample size (even n=2).

8. Other benefits

- Validity of the results does not depend on sampling approach.
    - Sample sizes and stopping rules do not need to be defined in advance.

- Bayesian approaches handles multiple testing better, and do not require correction for multiple testing.

Are Bayesian methods subjective?

Some claim that the use of a prior injects too much subjectivity.

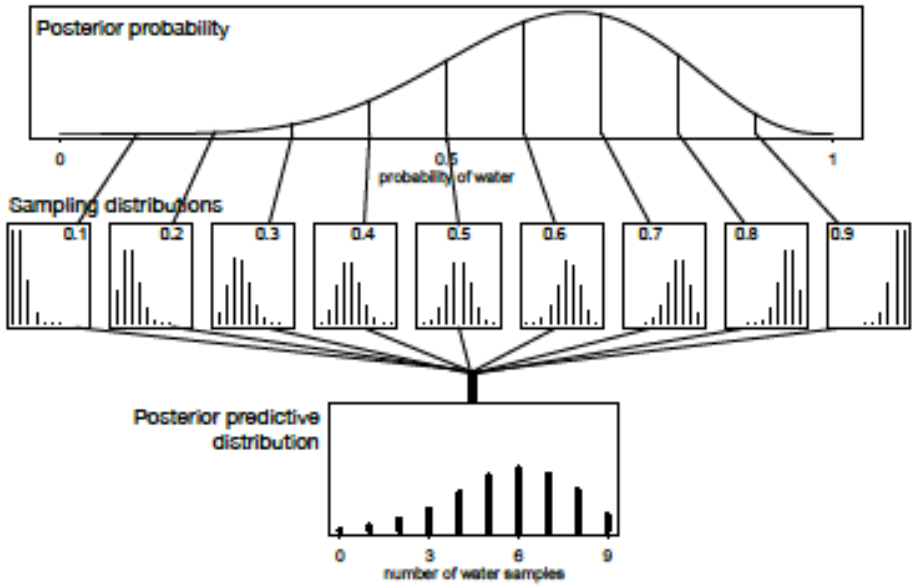
- Strange priors are easily identified, and non-informative / weakly informative can be used.
- Priors are quickly overwhelmed by likelihood.
- Other kinds of p-hacking are a much greater threat.
- Posterior distributions are asymptotically normal.
- Again...it works.

Challenges in using Bayesian methods

- Many (most) reviewers are not familiar with statistics
- Requires more independent thought.
- Can be extremely computationally intensive for some complex models.

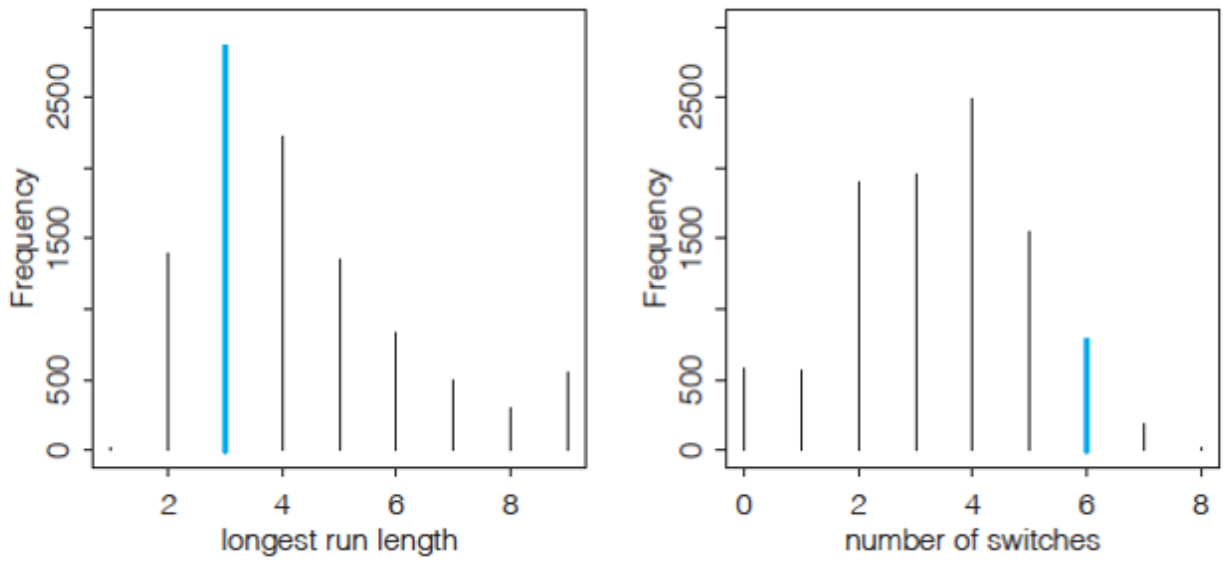
Some important bayesian concepts

Posterior predictive checks (1)



- Posterior predictive checks are a common way of evaluating the fit of a bayesian model.
- They involve simulating replicated data under the fitted model and then comparing these to the observed data.
- You then use posterior predictive results to look for systematic discrepancies between real and simulated data.

Posterior predictive checks (2)

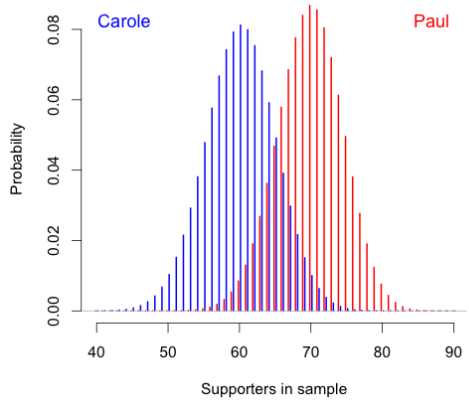


Our goal of seeking out aspects of prediction in which the model might fail.

We are looking at the expectations under our model in different ways:

- Left: Longest run length of water is quite in line with what we would expect.
- Right: Number of switches is less lin line with what we would expect (but not extremely off).

Bayes factor example (1)

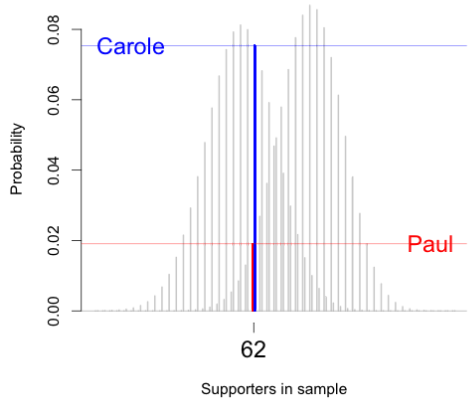


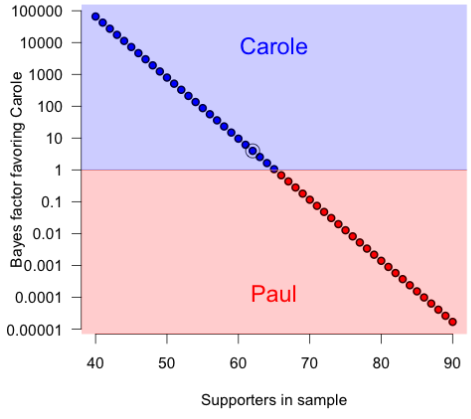
Two researchers are interested in public opinion about public smoking bans. Paul believes that 70% of the public support such bans; Carole believes that the support is less, at 60%. Paul and Carole decide ask 100 randomly selected people whether they support public smoking bans.

- If 62 people in a sample of 100 say they support the ban. How does this support the different models?

https://www.r-bloggers.com/2014/02/what-is-a-bayes-factor/

Bayes factor example (2)



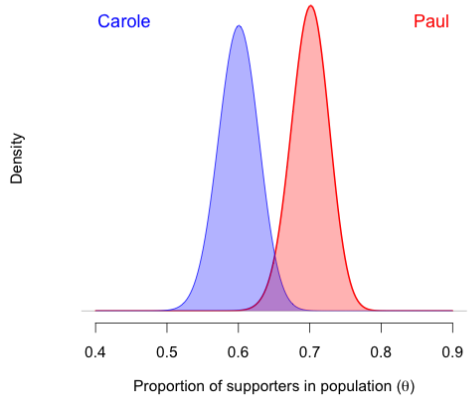


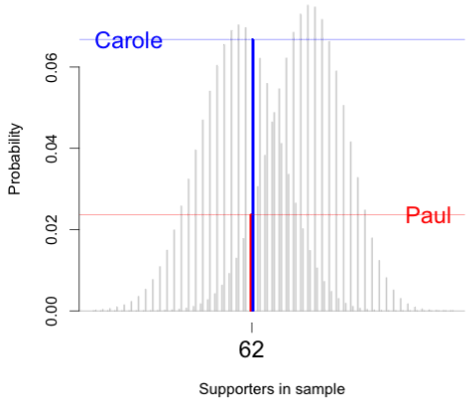
In this example the bayes factor is 3.95.

Bayes factor	Interpretation
1 - 3	Negligible evidence
3 - 20	Positive evidence
20 - 150	Strong evidence
>150	Very strong evidence

https://www.r-bloggers.com/2014/02/what-is-a-bayes-factor/

Bayes factor example (3)





Often our hypotheses are more diffuse.

- Above left: Our hypotheses are here expressed as distributions.
- Anove left: The odds of 62 must now be seen relative to all possible values under our hypotheses, and the bayes factor drops from 3.95 to 2.83.

https://www.r-bloggers.com/2014/02/what-is-a-bayes-factor/

$$BF_{ab} = \frac{P(D|H_a)}{P(D|H_b)}$$

- Bayes factors is a Bayesian alternative to classical hypothesis testing. The aim of the Bayes factor is to quantify the support for a model over another, regardless of whether these models are correct.
- Bayes factors are the degree to which the data shift the relative odds between two hypotheses.
- They have been proposed as more principled replacements for common classical statistical procedures such as p-values.
- One popular use for bayes factors is the tesing of *null models*.