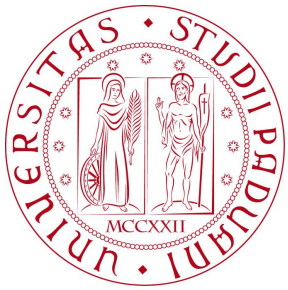
Advanced Statistics 4 Physics Analysis

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Lect. 1



Course Timetable

- 6 CFU: 4 hours/week → 12 full weeks
  - Thursday: Aula P3, 8:30 – 10:15, theory lectures
  - Thursday: Lab P104, 12:30 – 14:15, hands-on R laboratory sessions
- Course MOODLE Web Pages:
  - https://elearning.unipd.it/dfa/course/view.php?id=1400

Week		Mon	Tue	Wed	Thu		Fri
1	Feb 28 - Mar 4				Stat01	R01	
2	Mar 7 - 11				R02	R03	
3	Mar 14 - 18				R04	R05	
4	mar 21 - 25				Stat02	RLab01	
5	Mar 28 - Apr 1				Stat03	RLab02	
6	Apr 4 - 8						
7	Apr 11- 15				Stat04	RLab03	
8	Apr 18 - 22				Stat05	RLab04	
9	Apr 25 - 29				Stat06	RLab05	
10	May 2 - 6				Stat07	RLab06	
11	May 9 - 13				Stat08	RLab07	
12	May 16 - 20				Stat09	R06	
13	May 23 - 27				R07	Stat10	
14	May 30 - Jun 3				buffer	buffer	
15	Jun 6 - 10						

Course Program and Structure

➜ Part I:

- introduction to deductive logic and plausible reasoning
- review of discrete probability distributions
- review of continuous probability distributions
- sampling of random variables and Monte Carlo methods

➜ Part II:

- statistical models and inference: parameter estimation
- linear models
- model selections
- Markov Chain Monte Carlo and Gibbs sampling
- Bayesian Networks

➜ Part III (in parallel to I and II):

- R language structure and R libraries will be presented and used to solve exercises complementing the theory part

Laboratory Assignments and Exams

- during each laboratory session, a set of exercises will be assigned
- you will have to complete the assignments at home and deliver them in due time (within 2 weeks after each laboratory session)

Final Exam

- the final exam is made of two parts:
  - a written test with questions and small R exercises
  - a R computational problem will by assigned to each group of two-three students and the used techniques and the obtained results will be discussed in an oral group presentation
- the final vote will be a combination of three parts:

$$\text{Final Mark} = \frac{1}{3}\text{LaboratoryAssignments} + \frac{1}{3}\text{WrittenTest} + \frac{1}{3}\text{GroupProjectOralPresentation}$$

Statistics and Probability

What is the meaning of Statistics ?

Descriptive Statistics

- it refers to methods for summarizing and organizing the information in a data set
- it uses numbers, graphs and tables to describe data sets, as a first step of data analysis

Probability Theory

- Probability theory is the branch of mathematics concerned with probability
- as a mathematical foundation for statistics, probability theory is essential to many activities involving quantitative analysis of data
- Methods of probability theory also apply to descriptions of complex systems given only partial knowledge of their state, as in statistical mechanics

Inference

- statistical inference consists of methods for estimating and drawing conclusions about population characteristics based on the information contained in a subset (sample) of that population

➜ learning by data

Probability and Inference

- probability theory is the doctrine of chances
- a branch of mathematics that tells us how often different kind of events will happen:
  - what are the chances of a fair coin coming up heads 10 times is a row ?
  - if two six sided dice are rolled, how likely is to get 1-1 ?
- all these probabilistic questions start with a known model of the world and we use the model to perform some calculations
- in probability theory, the model is known but data are not
- statistical questions work the other way around:
  we do not known the truth about the world, and we want to know it from the data:
  - I flip a coint 10 times and get 10 heads, is the coin fair ?
  - while drawing 5 cards from a deck I get all hearts, how likely is that the deck had been shuffled ?

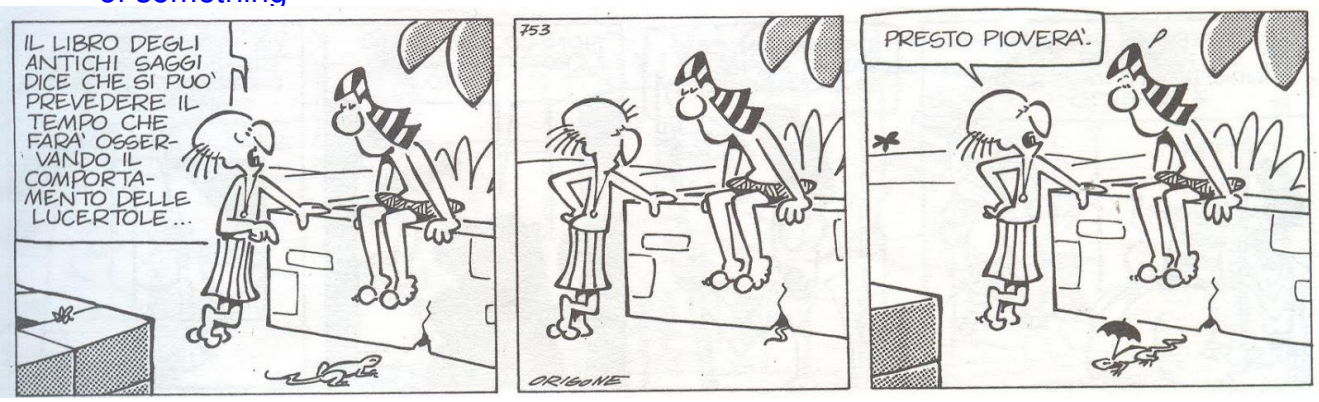
➜ statistical inference questions are not the same as probability questions, but they are deeply connected to one another

Logic, Probability and Uncertainty

- most of the situations we deal with in everyday life are not completely predictable
  - Q: will it rain today, at 14:30, when I will be done with my lecture and go home for lunch ?
  - A: gather information : weather forecast, look at the sky, . . .
  - despite the forecast I could get soaked going home

▷ but . . . there is always uncertainty

- therefore, since we cannot eliminate uncertainty, we need to model it
- when faced with uncertainty, we use plausible reasoning
- we adjust our belief about something, based on the occurrence or non-occurrence of something



Plausible Reasoning

- the policeman, the gentleman, and the jewelry shop broken window
  [E. T. Jaynes, *Probability Theory, The Logic of Science*, Cambridge Univ. Press., 2003]

- after having seen the masked gentleman, with a bag, crawling out of a broken jewelry shop window, the policeman thinks - with no doubts - that the gentleman is a thief

- BUT . . . the policeman's conclusion is NOT logical deduction from evidence
- there may be a perfectly innocent explanations:
  - the gentleman owns the jewelry shop
  - he just came home from a masquerade party and he had left his shop keys at home
  - a truck just passed and threw a stone on the window shop, breaking it

➜ the gentleman was just protecting his own property

- the policeman's reasoning process was not logical deduction, but it had a certain degree of validity

➜ the evidence did not make the gentleman's dishonesty certain, but it did make it extremely plausible

Plausible Reasoning

- the formulation of plausible conclusions is a very subtle process
- we have examples of contrast between deductive reasoning and plausible reasoning

  weak syllogisms:

  `if A is true, then B is true`

  but, if `B is true`, therefore `A becomes more plausible`

- in other words:
- the evidence does not prove that `A is true`
- verification of one of its consequences does `give us more confidence in A`

Example

- `A` ≡ it will start to rain today by 14:30 at the latest
- `B` ≡ the sky will become cloudy before 14:30
- observing clouds at 14:15 does not give us a logical certainty that rain will follow
- nevertheless our commons sense induce us to take an umbrella, or stay inside, if the clouds are sufficiently dark

Plausible reasoning

- in spite of the apparent weakness of its argument, we recognize that the policeman's conclusion has a very strong convincing power

➜ our brain, in doing plausible reasoning, evaluates the degree of plausibility, in some way

- the plausibility of `rain at 14:30` depends very much on the `darkness of the clouds at 14:15`
- the brain makes use of old information as well as specific new data ➜ before deciding what to do, we try to recall our past experience with clouds and rain (and of last night's weather forecast on the news)
- the policeman was also making use of the past experience of policemen in general

> *In our reasoning we depend very much on prior information to help us evaluating the degree of plausibility in a new problem. This reasoning process goes on unconsciously, almost instantaneously, and we conceal how complicated it really is by calling it* ***common sense*** .
>
> [E. T. Jaynes, *Probability Theory, The Logic of Science*, Cambridge Univ. Press., 2003]

➜ we also learn with new situations and data

Decision theory and *uncertainty*

- we are *u*ncertain whenever we do not know what to do
- we may be uncertain because we do not have enough information to know just what we should do
- sometimes we learn new information and decide that if we had this piece of information at the time, we would have acted differently
- some people say ”if the outcome was unsatisfactory, the decision was obviously wrong”. Their point is the following: ”whether a decision was RIGHT or WRONG, is to be decided entirely on the results of the decision and not on the basis of the information available at the time the decision was taken”

Example

- you get the following offer:
  - you pick up a coin from your pocket and toss it ten times
  - if you get at least one `Head` you get 200 € , if not you pay 1 €
- you accept the bet and you lose. Was it a *w*rong decision ?
- given the chance, would you make the same *w*rong decision again ?

Decision theory and Inductive logic

- of course you would bet again!
  - the odds in favor are roughly 1024 to 1
- the conclusion is that you made a good decision, but you had a bad outcome
- if a decision involves a risk, it is always possible that a good decision can lead to a bad outcome. This is the meaning of risk
- if all is known about a situation, our knowledge is deterministic
- on the other hand, if we have some uncertainties, the reasoning is inductive, that is, it requires inductive logic for finding the best solution

➜ the important aspect is that we do not know everything, but we do know something

Statistical Inference

- it is of primary interest to physicists and engineers
- and it allows to learn about theory or models from experimental observations

▷ inference has to be based on probability theory

- and summaries of descriptive statistics can be used in cases in which statistical sufficiency holds, i.e. when sample arithmetic average and standard deviation are used instead of the n data points

How to Measure Plausibility

- suppose we try to use numbers to measure plausibility of propositions
- when we change our plausibility for some propositions, based on the occurrence of some other proposition, we are performing an induction

Properties of Plausibility Measures

- degrees of plausibility are represented by non-negative real numbers
- they qualitatively agree with common sense: larger values mean greater plausibility
- if a proposition can be represented in more than one way, all representations must give the same plausibility
- we must always take all the relevant evidence into account
- equivalent states of knowledge are always given the same plausibility

- a sensible way to revise plausibility is by using the rules of probability
- probability is used as a extension of logic to cases where deduction cannot be made

The adopted Language

a Proposition

- must have an unambiguous meaning
- must be of a simple logical type, i.e. `true` or `false`
- given two propositions, A, and B

$$A \cdot B$$

  - is called `logical product` or conjunction and denotes the proposition: `both A and B are` **`true`**
  - the order does not matter: $A \cdot B$ and $B \cdot A$ say the same thing

$$A + B$$

  - is called `logical sum` or disjunction and stands for `at least one of the propositions A, B are` **`true`**
  - the order does not matter: $A + B$ brings the same information of $B + A$

$$A = B$$

  - means that the propostion on the left side has the same truth value as that on the right side

$$\overline{A}$$

  - indicated the denial of a propostion, i.e. if A `is` **`true`**, $\overline{A}$ `is` **`false`**, and vice versa

Probability and Random Experiments

- we borrow examples from theory of games: in a random experiment (i.e. dice draw, coin tossing, ...) the outcome is uncertain

Definitions

- Random experiment : an experiment with an outcome not completely predictable. When we repeat the experiment we may get a different result (e.g. coin tossing)
- Outcome : the result of a single trial of the experiment
- Sample space : the set of all possible outcomes of one single trial of the experiment. It is usually denoted by Ω. The sample space containing everything we are considering in the analysis of the experiment is called Universe
- Event : any set of possible outcomes of the experiment

Given two events, *E* and *F*, we can construct

- Union of the events : is the set of outcomes in either *E* or *F* (inclusive or)
- Intersection of the events : is the set of the outcomes in both *E* and *F*, simultaneously
- Complement of an event : is the set of all outcomes not in *E*

Axiomatic Definition of Probability

- probabilities are real numbers between 0 and 1
- the higher the probability, the more likely it is to occur
- a probability equals to 1 means the event is certain to occur
- a probability of 0 means that the event cannot possibly occur

You know it all from Probability Theory

The following axioms are satisfied

1) $P(A|I) \geq 0$ for any event E
2) $P(U|I) = 1$, is the probability of the universe (it means that some outcome occurs every time the experiment is performed)
3) $P(AB|I) = P(A|B,I) \cdot P(B|I) = P(B|A,I) \cdot P(A|I)$ **(PRODUCT RULE)**

Other properties, derived from the Axioms

1) $P(\emptyset) = 0$
2) $P(\overline{A}|I) = 1 - P(A|I)$ **(NORMALIZATION)**
3) $P(A+B|I) = P(A|I) + P(B|I) - P(AB|I)$ **(SUM RULE)**
4) $P(A|I) = P(A,B|I) + P(A,\overline{B}|I)$ **(MARGINALIZATION)**

All Probabilities are Conditional

- whenever we talk about probability, we always use the symbol

$$P(A|I)$$

- and never $P(A)$
- I is the background condition, related to information we have
- because it makes no sense to talk about the probability of the truth of the statement A without being explicit about the conditions upon which the assignment of probability is based
- we are engaged in a chain of inductive logic and at each point where an answer is required, we shall report the best inference we can make based upon the evidence available to that point
- as new evidence becomes available, we shall use the same procedure to update our inferences
- it is therefore an iterative process until evidence is so overwhelming that it does not seem worthwhile to pursue the matter any further

A note on the English Language

▷ Ordinary language is not precise

one of the most ambiguous word in the language is OR

- As an example, the statement: *"This is a ceramic or a conductor"* may be represented at least in two ways.
  Let's define:
  - proposition A: "it is a ceramic"
  - proposition B: "it is a conductor"

- the sentence can be represented by $(A + B)$ or by $\left(A\overline{B} + \overline{A}B\right)$
- because it is not clear if the word *or* is used in the inclusive or exclusive sense
  - if we say $A + B$ and we mean $(A + B)$, the word or is used in the inclusive sense
  - instead if $A + B$ is meant by $\left(A\overline{B} + \overline{A}B\right)$, the word or is used in the exclusive sense
- similarly the equal sign does not mean that two statements mean the same thing, but only that they have the same truth table

Independent versus Mutually Exclusive Events

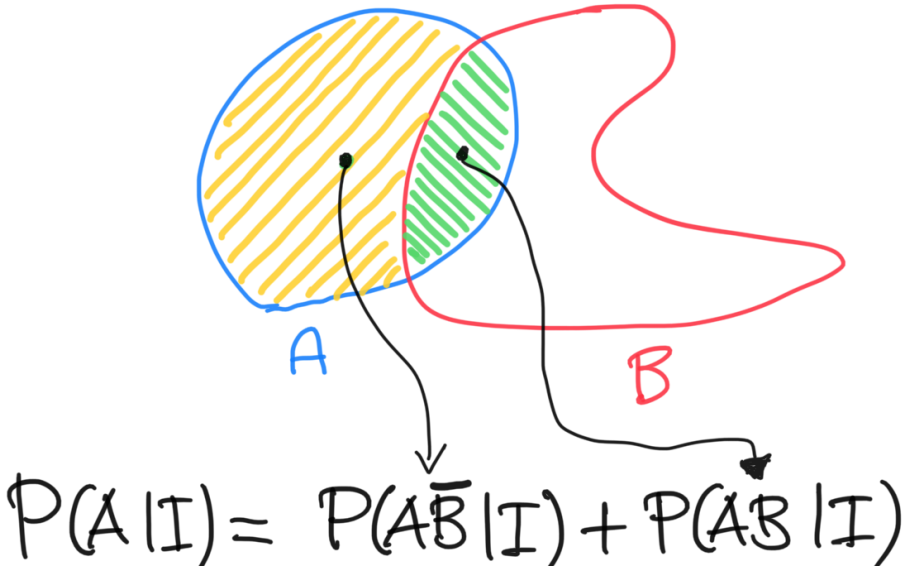
- a semantic confusion may arise because the word independent has several meanings
- primary meaning : two events are independent if the occurrence of the first does not affect the occurrence or non-occurrence of the second
- For independent events, $P(A|B, I) = P(A|I)$, therefore $P(AB|I) = P(A|I)P(B|I)$
- independence of two events is not a property of the events themselves, but it comes from the probabilities of the events and their intersection
  - if A and B are mutually exclusive events, (i.e. they have no outcome in common), $P(A + B|I) = P(A|I) + P(B|I)$
  - mutually exclusive events contain no elements in common, and this is a property of the events

Laplace's "Bayes Theorem"

Marginal Probability

- The marginal probability of one event is found by summing its disjoint parts:

$$P(A|I) = P(AB|I) + P(A\overline{B}|I)$$



Laplace's "Bayes Theorem"

Marginal Probability

- The marginal probability of one event is found by summing its disjoint parts:

$$P(A|I) = P(AB|I) + P(A\overline{B}|I)$$

Bayes' Theorem : 1

- from the definition of the probability of the intersection of two events

$$P(AB|I) = P(A|B,I)P(B|I) = P(B|A,I)P(A|I)$$

- we can re-write one of the two equalities as:

$$P(B|A,I) = \frac{P(AB|I)}{P(A|I)}$$

- and making use of the marginalized expression for $P(A|I)$

$$P(B|A,I) = \frac{P(AB|I)}{P(AB|I) + P(A\overline{B}|I)}$$

Laplace's "Bayes Theorem" : 2

- we now use the product rule to find each of the joint probabilities

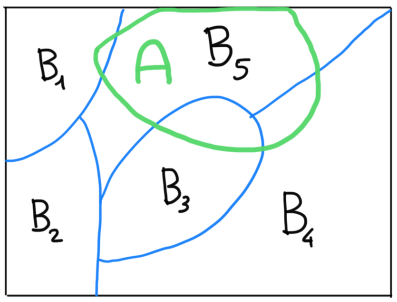
$$P(B|A,I) = \frac{P(A|B,I)P(B|I)}{P(A|B,I)P(B|I) + P(A|\overline{B},I)P(\overline{B}|I)}$$

- Bayes' theorem is a restatement of the original product rule in terms of the $P(B|A)$ probability, where
  1) the probability of A is found as the sum of the probabilities of its disjoints parts AB and $A\overline{B}$
  2) each of the joint probabilities are found using the multiplication rule
- it is important to notice that:
  1) the union of B and $\overline{B}$ is the whole universe
  2) and they are disjoint

Partition and Marginalization

- let's imagine that we have a set of more than two events that partition the universe:

$$U = B_1 \cup B_2 \cup \ldots \cup B_n$$



- and every distinct pair of the events are disjoint

$$B_j \cap B_i = \emptyset \text{ , with } j \neq i$$

- an observable event A will be partitioned into parts

$$A = \bigcup_{j=1}^{n} (A \cap B_j)$$

- and therefore

$$P(A|I) = \sum_{j=1}^{n} P(AB_j|I) = \sum_{j=1}^{n} P(A|B_j,I)P(B_j|I)$$

- where we have used the product rule on each joint probability

Laplace's "Bayes Theorem" : 3

- finally, the conditional probability $P(B_j|A, I)$ is found by dividing each joint probability by the probability of the event A, partitioned over all B_j

$$P(B_j|A, I) = \frac{P(A|B_j, I)P(B_j|I)}{\sum_{j=1}^{n} P(A|B_j, I)P(B_j|I)}$$

  - this is known as Bayes' theorem
  - it was published posthumously in 1763 after the death of its discoverer, Reverend Thomas Bayes
  - Pierre-Simon Laplace reproduced and extended Bayes's results in 1774, apparently unaware of Bayes's work
  - The Bayesian interpretation of probability was developed mainly by Laplace

Note that

- the events A and B_j, $(j = 1, \ldots, n)$, are not treated symmetrically :
  - A is an observable event
  - B_j are considered not observable. We never know which one of them occurred
- The marginal probabilities $P(B_j)$ are assumed known before we start and are called our prior probabilities

Interpretation of the Laplace's "Bayes Theorem"

- we often write Bayes' in its proportional form as

$$P(B_j|A, I) \propto P(A|B_j, I) \cdot P(B_j|I)$$

posterior — likelihood — prior

- this gives the relative weights for each of the events B_j, after we know A has occurred
- dividing by the sum of the relative weights it re-scales the relative weights, they now sum to 1 → making it a probability distribution

$$P(B_j|A, I) = \frac{P(A|B_j, I)P(B_j|I)}{\sum_{j=1}^{n} P(A|B_j, I)P(B_j|I)}$$

Exercise 1

- a new medical diagnostic test for a specific disease comes to market
- the sensitivity of the test, i.e. the probability the test gives a positive response if a person has the disease is 0.95
- the specificity, i.e. the probability that the test gives a negative response if a person does not have the disease, is 0.90

↘ this allows to evaluate the false positive rate to 0.10 (i.e. the probability that the test gives a positive response if a person does not have the disease)

- we know that 1% of the population, in Italy, has the disease

➜ What is the probability that a person has the disease, given the fact that the test gives a positive response ?

- we define the following propositions:
  $D :=$ a person has a disease
  $T :=$ the test gives a positive result

- and the data tell us
  $P(T|D, I) = 0.95$
  $P(\overline{T}|\overline{D}, I) = 0.90$
  $P(D|I) = 0.01$

Contingency Tables

- it's a method of keeping track of the various probabilities, and seeing how to compute one from another
- the following example shows a 2×2 table of probabilities involving two statements A and B :

	B	$\overline{B}$	
A	$P(AB\|I) = \omega_1$	$P(A\overline{B}\|I) = \omega_2$	$P(A\|I) = \omega_1 + \omega_2$
$\overline{A}$	$P(\overline{A}B\|I) = \omega_3$	$P(\overline{A}\overline{B}\|I) = \omega_4$	$P(\overline{A}\|I) = \omega_3 + \omega_4$
	$P(B\|I) = \omega_1 + \omega_3$	$P(\overline{B}\|I) = \omega_2 + \omega_4$	1

Exercise 1, Solution

- the initial data give: $P(T|D,I) = 0.95$, $P(\overline{T}|\overline{D},I) = 0.90$ and $P(D|I) = 0.01$

- we build a contingency table with the propositions:

  $D :=$ a person has a disease

  $T :=$ the test gives a positive result

	T	$\overline{T}$	
D	0.0095	0.0005	0.01
$\overline{D}$	0.099	0.891	0.99
	0.1085	0.8915	1

- we deduce: $P(\overline{D}|I) = 1 - P(D|I) = 0.99$ and $P(T|\overline{D},I) = 1 - P(T|D,I) = 0.05$
- $P(DT|I) = P(T|D,I) \cdot P(D|I) = 0.95 \times 0.01 = 0.0095$
- $P(D\overline{T}|I) = P(D|I) - P(DT|I) = 0.01 - 0.0095 = 0.0005$
- $P(\overline{D}T|I) = P(T|\overline{D},I) \cdot P(\overline{D}|I) = P(T|\overline{D},I)(1 - P(D|I) = 0.1 \times 0.99 = 0.099$
- $P(\overline{D}\,\overline{T}|I) = P(\overline{D}|I) - P(\overline{D}T|I) = 0.99 - 0.099 = 0.891$
- finally,
  $P(T|I) = P(DT|I) + P(\overline{D}T|I) = 0.1085$ and $P(\overline{T}|I) = P(D\overline{T}|I) + P(\overline{D}\,\overline{T}|I) = 0.8915$

Exercise 1, Solution

- the initial data give: $P(T|D,I) = 0.95$, $P(\overline{T}|\overline{D},I) = 0.90$ and $P(D|I) = 0.01$

- we build a contingency table with the propositions:

  $D :=$ a person has a disease

  $T :=$ the test gives a positive result

	T	$\overline{T}$	
D	0.0095	0.0005	0.01
$\overline{D}$	0.099	0.891	0.99
	0.1085	0.8915	1

- let's apply Bayes' theorem:

$$P(D|T,I) = \frac{P(T|D,I)P(D|I)}{P(T|I)}$$

- the normalization factor is

$$\begin{aligned} P(T|I) &= P(T|D,I)P(D|I) + P(T|\overline{D},I)P(\overline{D}|I) \\ &= 0.95 \times 0.01 + 0.10 \times 0.99 = 0.1085 \end{aligned}$$

- and the final answer is:

$$P(D|T,I) = \frac{0.0095}{0.1085} = 0.0876$$

Exercise 2

- a new medical screening procedure for a specific cancer is introduced
- the screening has:
  sensitivity = 0.90, and specificity = 0.95
- suppose the underlying rate of the cancer in the population is 0.001

➜ What is the probability that a person has the disease given the results of the screening is positive ?

➜ Does this show that screening is effective in detecting this cancer ?

- we define the following propositions:
  $B :=$ a person has a specific cancer
  $A :=$ the screening has positive result

- and the data tell us
  $P(A|B, I) = 0.90$
  $P(\overline{A}|\overline{B}, I) = 0.95$
  $P(B|I) = 0.001$

Exercise 2, Solution

- the initial data give: $P(A|B, I) = 0.90$, $P(\overline{A}|\overline{B}, I) = 0.95$ and $P(B|I) = 1 \cdot 10^{-3}$

$B :=$ a person has a specific cancer
$A :=$ the screening has positive result

	A	$\overline{A}$	
B	$9 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$10 \cdot 10^{-4}$
$\overline{B}$	0.050	0.949	0.999
	0.051	0.949	1

Calculations

- given sensitivity and specificity, $P(A|\overline{B}, I) = 1 - 0.95 = 0.05$
- $P(AB|I) = P(A|B, I)P(B|I) = 0.90 \times 0.001 = 9 \cdot 10^{-4}$
- $P(A\overline{B}|I) = P(A|\overline{B}, I)P(\overline{B}|I) = 0.05 \times 0.999 = 0.050$

$$\begin{aligned} P(A|I) &= P(AB|I)P(B|I) + P(A\overline{B}|I)P(\overline{B}|I) \\ &= 9 \cdot 10^{-4} \times 10 \cdot 10^{-4} + 0.04995 \times 0.999 = 0.051 \end{aligned}$$

- Bayes' theorem tells us that

$$P(B|A, I) = \frac{P(A|B, I)P(B|I)}{P(A|I)} = \frac{9 \cdot 10^{-4}}{0.05} = 0.018$$

Odds Ratios

- another way of dealing with uncertain events we are modeling is to form the odd ratio of the event:

$$odds(A|I) = \frac{P(A|I)}{P(\overline{A}|I)}$$

- since $P(\overline{A}|I) = 1 - P(A|I)$,

$$odds(A|I) = \frac{P(A|I)}{1 - P(A|I)}$$

- if we are using prior probabilities, we get the prior odds ratio
- if, instead, if we are using posterior probabilities, we get the posterior odds ratio
- solving the equation for the probability of the event A, we get

$$P(A|I) = \frac{odds(A|I)}{1 + odds(A|I)}$$

Assigning probabilities: 1

- so far we have derived consistent rules that can be used to manipulate probabilities, associated to real numbers
- but nothing tell us what actual numerical value we should assign at the beginning of a problem
- Ssuppose we have n propositions A_j, $j = 1, \ldots, n$, and at least one of them is `true`
- In addition, let's suppose that they are mutually exclusive, i.e.

$$P(A_i A_j|B) = P(A_i|B)\delta_{ij}$$

- therefore

$$P(A_1 + A_2 + \ldots + A_m|B) = \sum_i^m P(A_i|B), \text{ with} 1 \leq m \leq n$$

- we add the hypothesis that the propositions are also exhaustive, the background information B stipulates that one and only one of them must be true. Therefore:

$$\sum_{i=1}^{n} P(A_i|B) = 1$$

Assigning probabilities: 2

Problem I

- we have a set of mutually exclusive and exhaustive propositions: $\{A_1, \ldots, A_n\}$ and we seek to evaluate $P(A_i|B)_I$, with $i = 1, 2, \ldots, n$
- note that the labels are arbitrary: it makes no difference which proposition is called A_1 and which A_2, and so on

Problem II

- we have a set of mutually exclusive and exhaustive propositions: $\{A'_1, \ldots, A'_n\}$, given by
$$A'_1 \equiv A_2, \quad A'_2 \equiv A_1 \text{ and } A'_k = A_k, 3 \leq k \leq n$$
- and we seek to evaluate $P(A_i|B)_{II}$, with $i = 1, 2, \ldots, n$
- since the background information B is the same for both problems, we must have
$$P(A_1|B)_I \equiv P(A'_2|B)_{II} \text{ and } P(A_2|B)_I \equiv P(A'_1|B)_{II}$$
- problem I and II are not only related, but entirely equivalent, therefore
$$P(A_i|B)_I \equiv P(A'_i|B)_{II} \; \forall i = 1, 2, \ldots, n$$

Assigning probabilities: 3

- since equivalent state of knowledge must represent equivalent plausibility assignments
- propositions A_1 and A_2 must be assigned equal probabilities in Problem I and II
- the argument just given is the 'baby' version of the group invariance principle
- more generally, $\{A_1'', \ldots, A_n''\}$ may be any permutation of $\{A_1, \ldots, A_n\}$
- if the background information B is indifferent among all propositions. we are exactly in the same state of knowledge
- we obtain n equation of the form $P(A_i|B)_I = P(A_k|B)_I$ and the relations must hold whatever the particular permutation we used in our problem
- we come to the conclusion that all the $P(A_i|B)$ must be equal
- using the initial assumption that the $\{A_1, \ldots, A_n\}$ are exhaustive, i.e. $\sum_{i=1}^{m} P(A_i|B) = 1$, the only possibility is
$$P(A_j|B) = \frac{1}{n} \text{ with } 1 \leq j \leq n$$

› following Keynes (1921), we shall call this result the principle of indifference

Boolean Algebra

- Given two propositions **A** and **B**, we define two binary operations:

  - **A**· **B** as the logical product or conjunction (also called `AND`). This will be `true`, if 'both A and B are `true`'
  - **A**+ **B** as the logical sum or disjunction; it stands for 'at least one of the propositions A, B is `true`'

- and one unary operation:

  - $\overline{A}$ indicates the denial of a proposition
  - and the relation between A and $\overline{A}$ is reciprocal: $\overline{\overline{A}} = A$

A	B	NOT $\overline{A}$	NOT $\overline{B}$	AND $A \wedge B$	OR $A \vee B$
T	T	F	F	T	T
T	F	F	T	F	T
F	T	T	F	F	T
F	F	T	T	F	F

Boolean Algebra: basic identities

$A \cdot A = A$	$A + A = A$	idempotence
$A \cdot B = B \cdot A$	$A + B = B + A$	commutativity
$A\,(B \cdot C) = (A \cdot B)\,C$	$A + (B + C) = (A + B) + C$	associativity
$A\,(B + C) = A \cdot B + A \cdot C$	$A + B \cdot C = (A + B)\,(A + C)$	distributivity
$C = A \cdot B \rightarrow \overline{C} = \overline{A} + \overline{B}$	$D = A + B \rightarrow \overline{D} = \overline{A} \cdot \overline{B}$	De Morgan's
$A \Longrightarrow B$	to be read as **A implies B**	implication

A implies B does not assert that either A or B is `true`, but it means that $A\overline{B}$ is `false` or that $\left(\overline{A} + B\right)$ is `true`.
That is if A is `true`, then B must be `true`; or if B is `false`, then A must be `false`.

Boolean Algebra: adequate set

- so far we have defined four operations by which, starting from two propositions, A and B, other propositions may be defined
  - the logical product: $A \cdot B$ (also written AB)
  - the logical sum: $A + B$
  - the negation: $\overline{A}$
  - the implication: $A \Longrightarrow B$

NOTE : in ordinary language, 'A implies B' ($A \Longrightarrow B$) means that B is logically deducible from A. In formal logic, 'A implies B' means only that the propositions A and B have the same truth value

- it is possible to demonstrate that the three basic operations of Boolean Algebra, AND, OR and NOT constitute and 'adequate set', i.e. they suffice to generate all possible logic functions

Example

- Boolean algebra rules can be used, for instance in enumerating all possible cases of an event
- let's define the following propositions:
  - E_1 : a coin is tossed once
  - E_2 : a coin is tossed for a second time
  - H_1 : the coin lands 'tail' on the first toss
  - H_2 : the coin lands 'tail' on the second toss
- we can write: $E_1 = H_1 + \overline{H_1}$ and $E_2 = H_2 + \overline{H_2}$ using the multiplication rule:

$$\begin{aligned} E_1 E_2 &= \left(H_1 + \overline{H_1}\right)\left(H_2 + \overline{H_2}\right) \\ &= H_1 H_2 + H_1 \overline{H_2} + \overline{H_1} H_2 + \overline{H_1}\ \overline{H_2} \end{aligned}$$

Boolean Algebra: alternative adequate set

- it is possible to demonstrate that there are operations which, alone, would constitute ad adequate set: these are NAND and NOR

- NAND is the negation of AND:

$$A \uparrow B \equiv \overline{AB} = \overline{A} + \overline{B}$$

- and it can generate the other operations:
  - $\overline{A} = A \uparrow A$
  - $AB = (A \uparrow B) \uparrow (A \uparrow B)$
  - $A + B = (A \uparrow A) \uparrow (B \uparrow B)$

- NOR is the negation of OR:

$$A \downarrow B \equiv \overline{A + B} = \overline{A}\,\overline{B}$$

- and it can generate the other operations:
  - $\overline{A} = A \downarrow A$
  - $A + B = (A \downarrow B) \downarrow (A \downarrow B)$
  - $AB = (A \downarrow A) \downarrow (B \downarrow B)$

NAND formulas demonstration

NAND

- $\overline{A} = A \uparrow A$ □

since $\overline{A} = \overline{A \cdot A}$ using idempotence

- $AB = \overline{\overline{AB}}$ since $\overline{\overline{A}} = A$

$= \overline{A \cdot B \cdot A \cdot B}$ using idempotence

$= (A \uparrow B) \uparrow (A \uparrow B)$ □

- $A + B = \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}}$ using De Morgan's

$= \overline{A \cdot A \cdot B \cdot B}$ using idempotence

$= (A \uparrow A) \uparrow (B \uparrow B)$ □

The Extended Sum Rule

$$
\begin{aligned}
P(A+B|C) &= 1 - P\left(\overline{A+B}|C\right) = 1 - P\left(\overline{A}\cdot\overline{B}|C\right) \\
&= 1 - P\left(\overline{A}|C\right)\cdot P\left(\overline{B}|\overline{A}C\right) = 1 - P\left(\overline{A}|C\right)\left[1 - P\left(B|\overline{A}C\right)\right] \\
&= 1 - P\left(\overline{A}|C\right) + P\left(\overline{A}|C\right)P\left(B|\overline{A}C\right) \\
&= P(A|C) + P\left(\overline{A}B|C\right) = P(A|C) + P(B|C)\,P\left(\overline{A}|BC\right) \\
&= P(A|C) + P(B|C)\,[1 - P(A|BC)] \\
&= P(A|C) + P(B|C) - P(B|C)\,P(A|BC) \\
&= P(A|C) + P(B|C) - P(AB|C)
\end{aligned}
$$

Summary of our mathematical tools

PRODUCT RULE	$P(AB|C) = P(A|BC)\,P(B|C) = P(B|AC)\,P(A|C)$
SUM RULE	$P(A|B) + P(\overline{A}|B) = 1$
EXTENDED SUM RULE	$P(A+B|C) = P(A|C) + P(B|C) - P(AB|C)$
BAYES' THEOREM	$P(B_j|A,I) = \dfrac{P(A|B_j,I)P(B_j|I)}{\sum_{j=1}^{n} P(A|B_j,I)P(B_j|I)}$
PRINCIPLE of INDIFFERENCE	Given $H_1,\ldots,H_n$, mutually exclusive and exhaustive. If B does not favor any of them: $P(H_j|B) = 1/n$

Some useful transformations

- given $P(A|B$, three probability transformations are particularly useful and will be employed later in our calculations:

- Odds :

$$O(A|E) = P(A|E)/P(\overline{A}|E)$$

- Evidence :

$$ev(A|E) = K \ln\left(O(A|E)\right) = K(\ln\left(P(A|E)\right) - \ln\left(P(\overline{A}|E)\right))$$

- Surprisal :

$$u(A|E) = -K \ln\left(P(A|E)\right)$$

Useful formulas

definition	range	denial rule	joint rule
Probability $p(A\|C)$	[0,1]	$P(\overline{A}\|C) = 1 - P(A\|C)$	$P(AB\|C) = P(A\|BC)P(B\|C)$
Odds $O(A\|C) = \frac{P(A\|C)}{P(\overline{A}\|C)}$	$[0,\infty]$	$O(\overline{A}\|C) = \frac{1}{O(\overline{A}\|C)}$	$O(AB\|C) = \frac{O(A\|BC)O(B\|C)}{1+O(A\|BC)+O(B\|C)}$
Surprisal $u(A\|C) = -k \ln P(A\|C)$	$[\infty,0]$	$u(\overline{A}\|C) = -k \ln\left(1 - \exp\frac{-u(A\|C)}{k}\right)$	$u(AB\|C) = u(A\|BC) + u(B\|C)$
Evidence $ev(A\|C) = k \ln \frac{P(A\|C)}{P(\overline{A}\|C)}$	$[-\infty,\infty]$	$ev(\overline{A}\|C) = -ev(A\|C)$	*

* $ev(AB|C) = ev(A|BC) + ev(B|C) - k \ln\left[1 + \exp\frac{ev(A|BC)}{k} + \exp\frac{ev(B|C)}{k}\right]$

References

Books

- Tribus M., *Rational Descriptions, Decisions and Designs*, Pergamon Press, 1969
- Bolstad W. M., *Introduction to Bayesian Statistics*, John Wiley & Sons, 2007, 978-0-470-14115-1
- D'Agostini G., *Bayesian reasoning in data analysis - A critical introduction*, World Scientific Publishing, 2003, ISBN 978-981444795-9
- Jaynes E. T., *Probability Theory, The Logic of Science*, Cambridge University Press, 2003, ISBN 978-0-521-59271-0

Review of Probability Distributions

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Stat Lect. 2

Pairing and Ordering of Objects

Unique pairing of objects

- given n objects, how many possible ways of selecting unique pairs, without caring about ordering ?
- let's consider a matrix $n \times n$
- every element in the matrix, except the leading diagonal, is a paring
- since the two parts on each side of the diagonal are identical (order does not count), we have

$$n_{pairs} = (n^2 - n)/2 = n(n-1)/2$$

Unique ordering of objects

- given n objects, how many possible ways of ordering them ?
- we have n options to select the first element
- $n-1$ for the second, $n-2$ for the third, ...
- therefore it is

$$n(n-1)(n-2)\ldots 2 \cdot 1 = n!$$

Combinations and Permutations

- in the english language the word *"combination"* is used loosely, without specifying if the order of the object is relevant
- examples:
  - when buying an ice cream, we select a *combination* of mint, chocolate and stracciatella. We do not care about the order of the three flavours on the cone
  - the *combination* of my bike locker is 4-3-6-9. In this case, the order of the numbers really matters!
- when we select k elements from a set of n objects
  - if the order of selection is NOT important, we have a combination
  - but if the order matters, we have a permutation

➜ a permutation is and ordered combination

Permutations - ORDER MATTERS

- there are two types of permutations

Repetition IS allowed

- given n objects, how many sequences of r elements ($r \leq n$) can be built ?

  Example: given n letters, how many words of r characters can be built with those letters ?
  - each object (character) has n different possibilities, therefore it is

$$n^r$$

Repetition is NOT allowed

- given n objects, we select r elements ($r \leq n$) from the set
- how many unique selections are possible ?
  - there are n ways to select the first, $n-1$ for the second, and $n-r+1$ for the r-th
  - we get:

$$n(n-1)\dots(n-r+1) = \frac{n!}{(n-r)!} = {}^nP_r$$

- this is called permutations, ${}^nP_r$. Note that ${}^nP_n = n!$

Combinations - ORDER IS NOT IMPORTANT

- there are two types of combinations

Repetition is NOT allowed

- we now select r objects, as in the previous case, but we are not concerned about the order
- the number of ways of selecting r object from a set of n without regard to the order of selection is called combinations, ${}^{n}C_{r}$

$$ {}^{n}C_{r} = \frac{{}^{n}P_{r}}{n!} = \binom{n}{r} = \frac{n!}{r!\,(n-r)!} $$

- this is the binomial coefficient, also called *n choose r*

Repetition IS allowed

- finally, the number of ways of choosing r objects from a set of n with replacement and without caring about the order is

$$ \binom{n+r-1}{r} = \frac{(n+r-1)!}{r!(n-1)!} $$

- this is sometimes called *n multichoose r*

Application: the Birthday Paradox

The Problem

- in a large room, full of people, how many of them do you have to ask before there is a 50% chance that any of two, ore more, share a common birthday ?
- assuming $n = 365$ birthday/year and equally probable, we consider r people and we combine them so that they do not share a common birthday

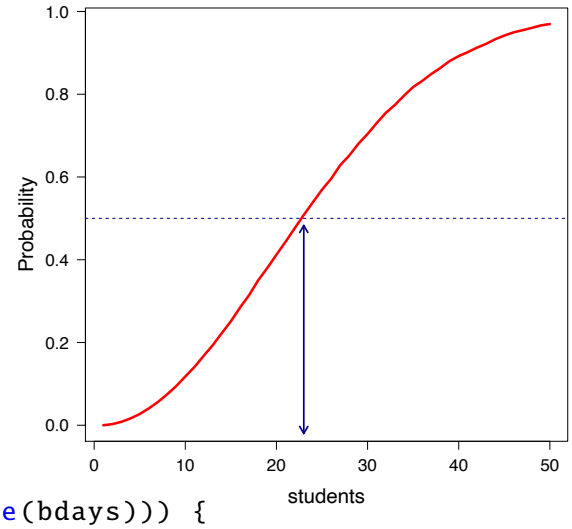
$$ A = n\,(n-1)\,\ldots\,(n-r+1) = \frac{n!}{(n-r)!} $$

- the way of assigning n birthday to r people is $B = n^{r}$
- the probability of no common birthday is A/B
- therefore the probability of at least one birthday is

$$ P(\text{birthday} \geq 1) = 1 - \frac{A}{B} = 1 - \frac{n!}{(n-r)!}\frac{1}{n^{r}} $$

Computation of the birthday problem

```
First element with prob>0.5:  23
```



R code

```
n_people_tot <- 50
pbday <- rep(0, n_people_tot)
for (k in 2:n_people_tot) {
  n_tests = 1E5; cb <- 0
  for (i in 1:n_tests) {
    bdays <- sample(1:365, k ,
                    replace=TRUE)
        if (length(bdays) > length(unique(bdays))) {
            cb = cb + 1
        }
    }
    pbday[k] <- cb/n_tests
    message(paste("k:", k, "pb(",k,"):",pbday[k]))
}
pfunc <- function(f, b) function(a) f(a,b)
p50_index <- Position(pfunc(`>`, 0.5), pbday)

message(paste("First element with prob>0.5:", p50_index))
```


R language note : closures

Anonymous functions

- can be used to create small function, not worth naming
- another important use is to create closures: functions written by functions

```
power <- function(exponent) function(x) x^exponent

square <- power(2)
square(2)
# [1] 4

cube <- power(3)
cube(2)
# [1] 8

pfunc <- function(f, b) {
             function(a) { f(a,b) }
}
p50 <- pfunc(`>`, 0.5)

x <- c(0.3, 0.51, 0.9)
p50(x)
# [1] FALSE  TRUE  TRUE
```

Probability Distributions

- two basic types: discrete distributions and continuous distributions
- discrete distribution : finite or countable set of possible outcomes of the random variable
- continuous distribution : a random variable can have outcomes in an interval of the real line
- probability densities are a way to specify probability distributions
- the cumulative distribution function (CDF) is defined by

$$F(x) = P(X \leq x)$$

- for discrete distributions:

$$F(x_j) = P(X \leq x_j) = \sum_j p_j$$

- while for continuous distributions:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(u)du$$

Probability Distributions

- with distribution functions, we compute the probability for intervals, $(c, d]$ as

$$P(c < X \leq d) = P(X \leq d) - P(X \leq c) = F(d) - F(c)$$

- the expectation, or expected value reflects the location of a distribution

$$E[X] = \sum_i x_i p(x_i) \qquad E[X] = \int_{-\infty}^{+\infty} x\, f(x)\, dx$$

- the variance reflects the dispersion of the distribution:

$$var(X) = E[X - E[X]]^2 = E[X^2] - (E[X])^2$$

- properties:

$$E[a + bX] = a + bE[X] \qquad var(a + bX) = b^2 var(X)$$
$$E[X + Y] = E[X] + E[Y] \qquad var(X + Y) = var(X) + var(Y) + 2cov(X, Y)$$

- with the covariance of the two variables

$$cov(X, Y) = E[(X - E[X])(Y - E[Y]] = E[XY] - E[X]E[Y]$$

Moments of a distribution

- they are analogous to the center-of-mass and to the momentum of inertia

Algebraic Moments

- the moment of order k about the origin is

$$\mu'_k \equiv E[x^k] = \int x^k f(x)dx \quad \text{and} \quad \sum_j x_j^k p_j$$

Central Moments

- the moment of order k about the mean are

$$\mu'_k \equiv E[(x-\mu)^k] = \int (x-\mu)^k f(x)dx \quad \text{and} \quad \sum_j (x_j-\mu)^k p_j$$

$$\begin{array}{ll} \mu'_0 = 1 & \mu_0 = 1 \\ \mu'_1 = \mu & \mu_1 = 0 \\ \mu'_2 = \mu + \sigma^2 & \mu_2 = \sigma^2 \end{array}$$

- the higher order moments become interesting only for studying the behavior of $f(x)$ for large $|x-\mu|$
- for a symmetric distribution, all odd central moments vanish ➜ non zero values are a possible measure of the skewness of a distribution

Probability Distributions in R

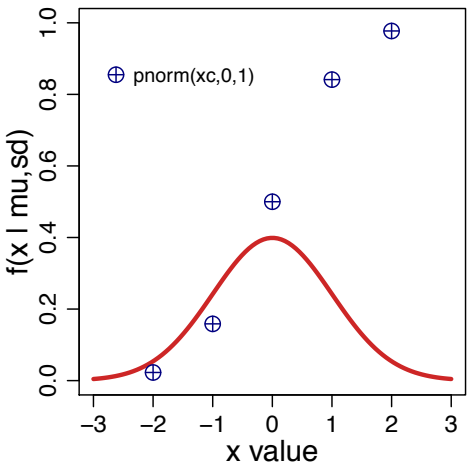
- all standard distributions available
- naming convention: a core name is associated with each distribution, and a prefix is appended to indicate the four basic associated functions:
  - `d` for the probability density function (pdf)
  - `p` for the cumulative density function (cdf)
  - `q` for the quantile function
  - `r` for the sampling from the distribution
- note that `pcore_name()` and `qcore_name()` are one the inverse of one another

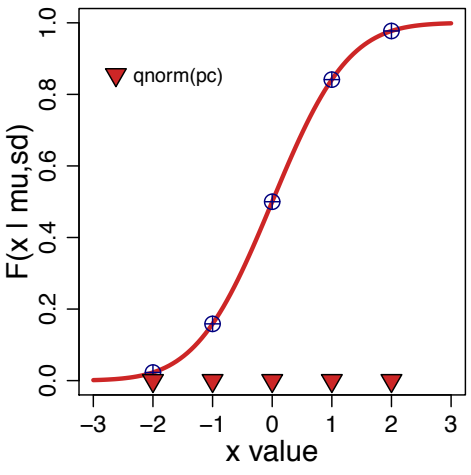
```
xc <- seq(-2,2,1)
pc <- pnorm(xc,0,1)
qc <- qnorm(pc)
```

xc: -2 -1 0 1 2

pc: 0.023 0.159 0.5 0.841 0.978

qc: -2 -1 0 1 2



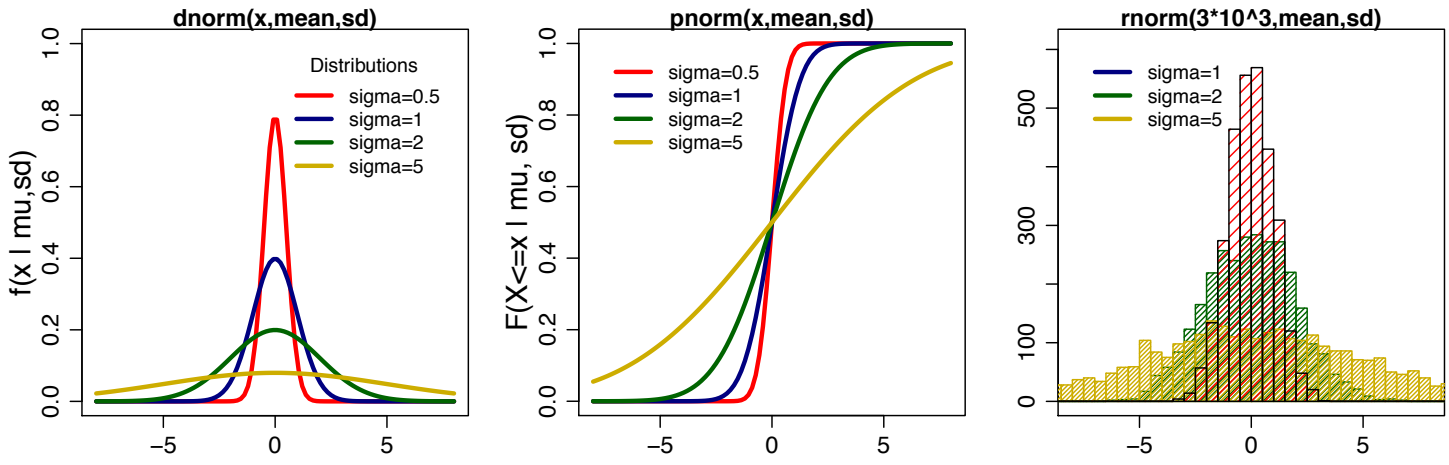


Standard Probability Distributions in R

Distribution	Core name	Parameters	Default values
Beta	`beta`	`shape1, shape2`	
Binomial	`binom`	`size, prob`	
Cauchy	`cauchy`	`location, scale`	0, 1
Chi-square	`chisq`	`df`	
Exponential	`exp`	`1/mean`	1
Fisher	`f`	`df1, df2`	
Gamma	`gamma`	`shape, 1/scale`	`NA`, 1
Geometric	`geom`	`prob`	
Hypergeometric	`hyper`	`m, n, k`	
Log-Normal	`lnorm`	`mean, sd`	0,1
Logistic	`logis`	`location, scale`	0,1
Normal	`norm`	`mean, sd`	0,1
Poisson	`pois`	`lambda`	
Student	`t`	`df`	
Uniform	`unif`	`min, max`	0,1
Weibull	`weibull`	`shape`	

Probability Distributions in R: normal distribution

- `dnorm(x, mean = 0, sd = 1)` gives a density of a normal distribution .i.e. the pdf
- `pnorm(q, mean = 0, sd = 1)` returns the distribution function, i.e. the cdf
- `rnorm(n, mean = 0, sd = 1)` generates random numbers from a normal distribution function
- `qnorm(p, mean = 0, sd = 1` is the quantile function



Standard Discrete Distributions

Bernoulli process

- it is a process with only two possible outcomes: success with probability p and failure with probability $1 - p$ (also called q, since $q = 1 - p$)
- if we call the two outcomes, 0 and 1, we can define $x \in [0, 1]$, and

$$P(X = 1) = p$$

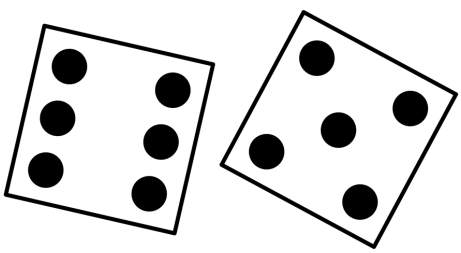
$$P(X = 0) = 1 - p = q$$

- the expected value and variance are

$$E[x] = p \quad \text{and} \quad Var(x) = p(1 - p)$$

Examples

- the toss of a coin
- the draw of a die



Binomial distribution

- the sum of n independent Bernoulli trials, follows a Binomial distribution

$$Bn(x \mid p, n) = \binom{n}{x} p^x (1-p)^{n-x}$$

- it gives the probability of $\mathtt{x}$ successes in $\mathtt{n}$ independent Bernoulli trials
- the expected value and variance are

$$E[x] = np \quad \text{and} \quad Var(x) = np(1-p)$$

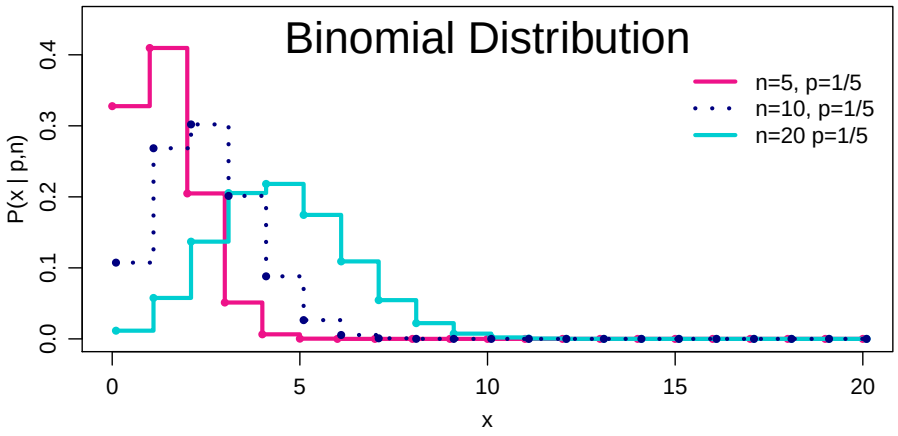
$$\sum_{j=0}^{n} \binom{n}{j} p^j \, (1-p)^{n-j} = (p + 1 - p)^n = 1$$

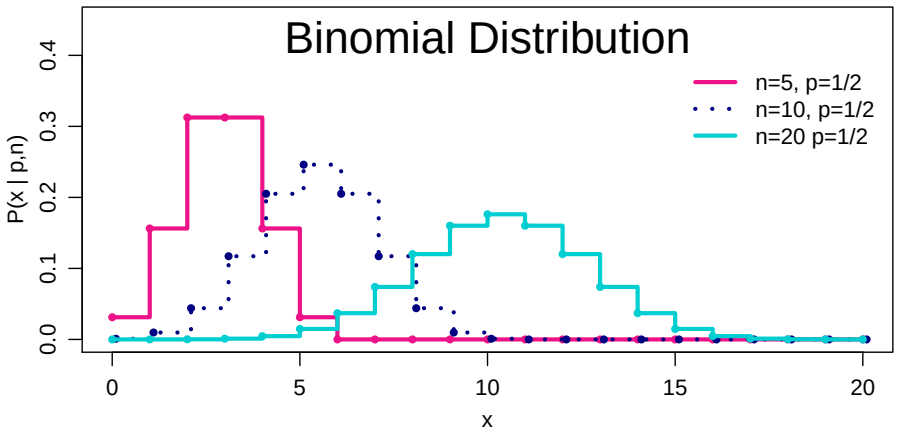
Examples

- multiple toss of a coin, or coins
- draw of dice
- drawing x red balls from an urn with n red and white balls (the fraction of red balls is p). Draws are done with replacement (➜ p remains constant)

Binomial distribution examples

- the distribution is symmetric when $p = 1/2$, and otherwise skew
- the distribution gets increasingly symmetric for higher values of n
- when n becomes large, it takes and approximate Gaussian shape





Binomial distribution - exercise

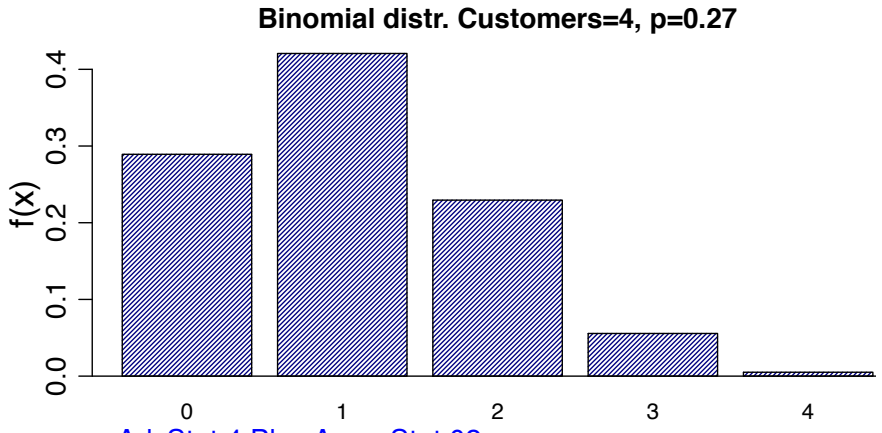
Problem

- in a restaurant 8 entrees of fish, 12 of beef and 10 of poultry are served
- what is the probability that 2 of the 4 next customers order fish entrees ?

Solution

```
cust <- 4
p <- 4/15
x <- 0:4
ap <- dbinom(x,cust,p)
barplot(ap, names=x, col='navy', xlab='x', ylab='f(x)', density=40,
        main = sprintf("Binomial distr. Customers=%d, p=%.2f",cust,p),
  cex.lab=1.5, cex.axis=1.25, cex.main=1.25, cex.sub=1.5)
cat(paste(c("P(2|np) = ", ap[3],'\n')))
```

```
P(2|np) = 0.229451851851852
```



Example: histogramming events

- we are interested in just the events contained in one bin of the histogram
  - A : we get the event of that particular bin (success)
  - $\overline{A}$: correspond to the events in any other bin (failure)
- the probability of having $x_\circ$ out of n events in the bin follows a Binomial distribution:

$$E[x] = np \quad \text{and} \quad Var(x) = np(1-p)$$

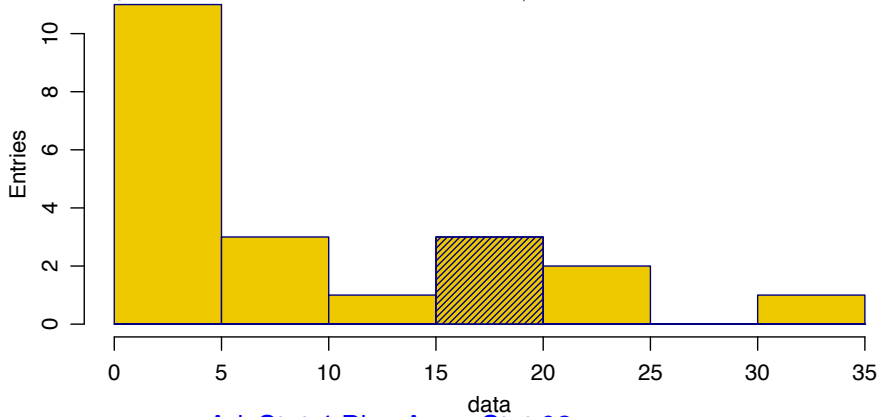
- p can be estimated as the ratio $p = x_\circ/n$:

$$E[x] = np = n\frac{x_\circ}{n} = x_\circ \quad \text{and} \quad Var(x) = x_\circ(1 - \frac{x_\circ}{n})$$

➜ the error on the number of the events is not $\sqrt{x_\circ}$, but a smaller quantity, $\sqrt{x_\circ(1-p)}$. Only in the limit $p \to 0$ (Poisson limit), $\sigma = \sqrt{x_\circ}$

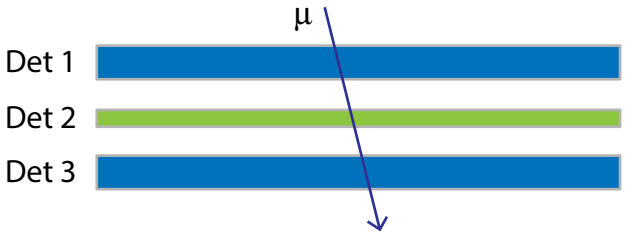
$\sqrt{3} = 1.73$

$\sqrt{3(1-3/21)} = 1.60$



Example: detection efficiency

- we want to compute the efficiency of a detector and evaluate the uncertainty on the measurement
- a muon-like signal has been registered by `Det1` and `Det3`
- what is the detection efficiency of our `Det2` ?



- detection is a Bernoulli process:

$$\epsilon_2 = \frac{N_{det2}}{N_{det1\&det3}} \quad with \quad N_{det2} \subset N_{det1\&det3}$$

- since we are interested in a relative number of success in a trial,

$$E\left[\frac{r}{n}\right] = \frac{1}{n}E[r] = p \quad \text{and} \quad Var\left(\frac{r}{n}\right) = \frac{1}{n^2}V(r) = \frac{p(1-p)}{n} = \frac{pq}{n}$$

- in our case, p is the ratio of events detected with `Det2` with respect to those seen by both `Det1` and `Det3`
- therefore:

$$\sigma(\epsilon_2) = \sqrt{\frac{\epsilon_2(1-\epsilon_2)}{N_{det1\&det3}}}$$

The drunk-man and the home keys problem

The background information

- a man comes back home pretty drunk
- he has 8 keys and tries them randomly to unlock his door apartment
- after each trial he loses memory
- we watch him and bet on the attempt on which he will succeed
- $n_{try} = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \ldots$
- on which number would you bet ?

The problem

- E_j : the door gets unlocked in attempt j, with $j = 1, 2, \ldots$
- we know that: $P(E_j | \overline{\cup_{j<i} E_j}) = 1/8$

$f(1) = P(E_1) = p = 1/8$

$f(2) = P(E_2 \cdot \overline{E}_1) = P(E_2 | \overline{E}_1) \cdot P(\overline{E}_1) = p \cdot (1-p)$

$f(3) = P(E_3 \cdot \overline{E}_2 \cdot \overline{E}_1) = P(E_3 | \cdot \overline{E}_2 \cdot \overline{E}_1) \cdot P(\overline{E}_2 | \cdot \overline{E}_1) \cdot P(\cdot \overline{E}_1) = p \cdot (1-p)^2$

$f(x) = p \cdot (1-p)^{x-1}$

Geometric distribution

- our probabilities follow a geometric distribution with $p = 1/8$

  $f(1) = p = 1/8 = 0.125$ ✓ our best bet!

  $f(2) = p(1-p) = 1/8(7/8) = 0.109$

  $f(3) = p(1-p)^2 = 0.096$

  $f(4) = p(1-p)^3 = 0.084$

  ...

- the geometric distribution gives the number of trials to get the first success

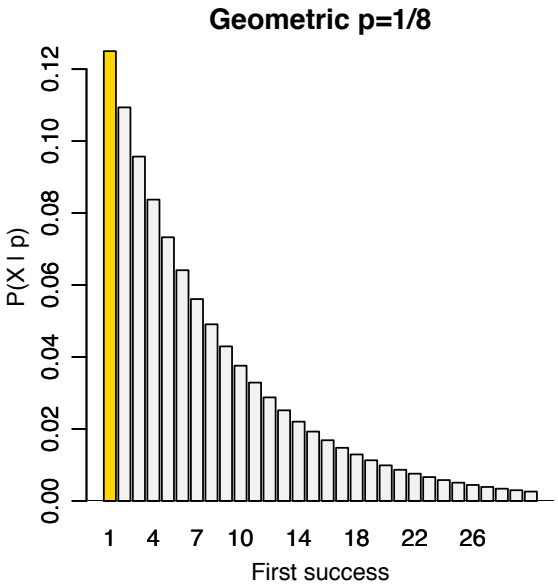
$$Geo(x|p) = p\,(1-p)^x$$

  - the expected value and variance are

$$E[x] = \frac{1}{p} \quad \text{and} \quad Var(x) = \frac{1-p}{p^2}$$

  - useful relations:

$$P(x \le r) = 1 - (1-p)^r = q^r \quad \text{and} \quad P(x > r) = 1 - q^r$$



Geometric distribution examples (1)

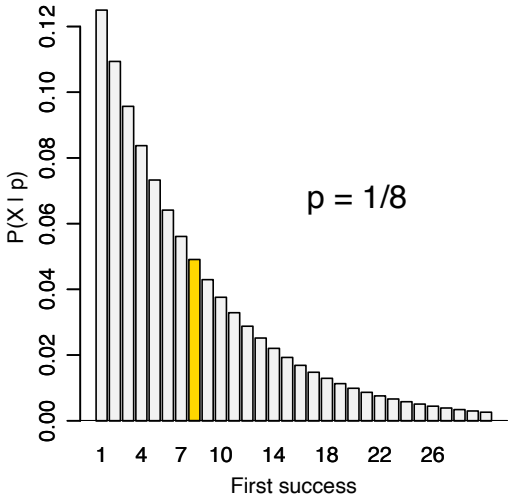
Drunk-man

- the first trial is the most probable
- but

$$E[X] = 1/p = 8$$

  and

$$\sigma = \sqrt{(1-p)/p^2} = 7.5$$

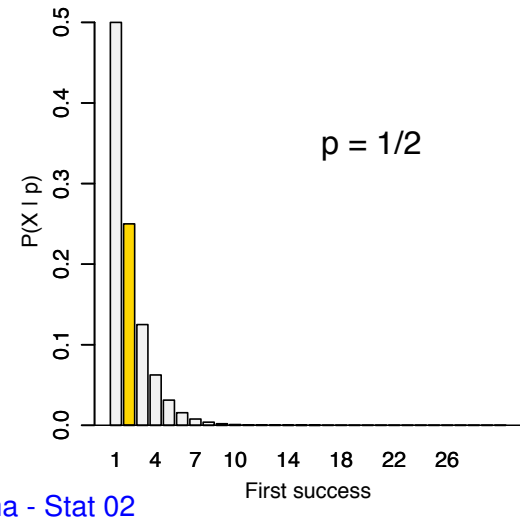


Coin tossing

- if we apply it to the tossing of one coin, we get
- $p_{max} = p = 1/2$
- and $E[X] = 1/p = 2$

  and

  $\sigma = \sqrt{(1-p)/p^2} = 1.4$



Geometric distribution examples (2)

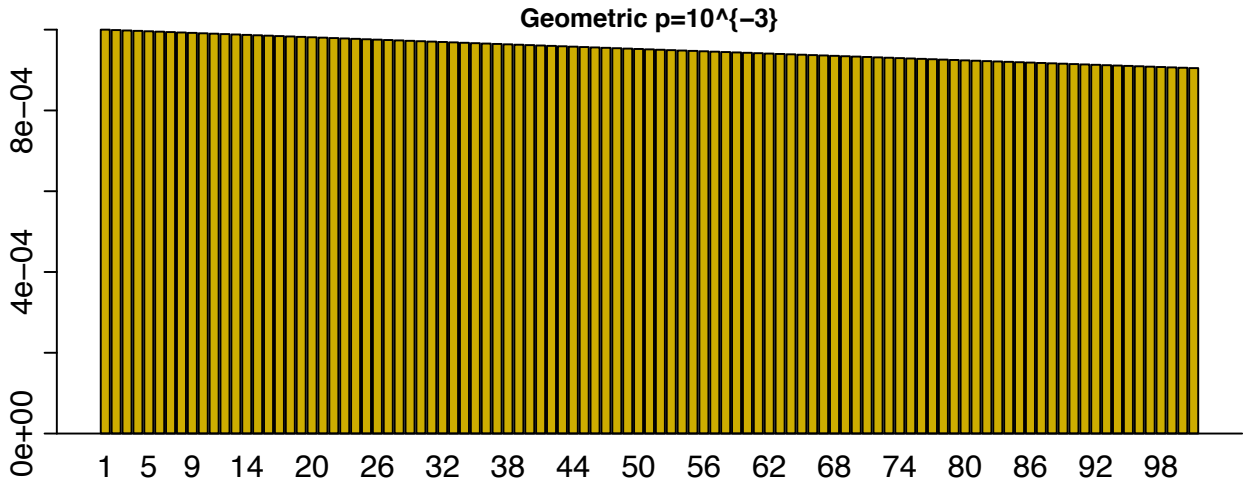
Rare Events

- let's decrease the probability of the event

$$E[X] = 1/p = 10^3$$

$$Var(X) = \frac{\sqrt{1-p}}{p} \xrightarrow[p\to 0]{} \frac{1}{p}$$

- rare moments might happen at any moment
  (even if they have a negligible probability to happen at any moment)



Geometric distribution in R

- given $x = \{1, 2, 3, \ldots\}$ as the number of trials for the first success
  an alternative representation uses
- $y = \{0, 1, 2, \ldots\}$ as the number of failures before the first sucess
- the two representations are equivalent:

```
y = x - 1

f(x) = p(1-p)^x = 1 - q^x
F(x) = 1 - (1-p)^x = 1 - q^x
E[x] = (1-p)/p    Var[x] = (1-p)/p^2

f(y) = p(1-p)^y
F(y) = 1 - (1-p)^(y+1)
E[y] = (1-p)/p    Var[x] = (1-p)/p^2
```

- the geometric distribution in R

```
Geometric                package:stats                R Documentation
The Geometric Distribution
Usage:

     dgeom(x, prob, log = FALSE)
     ...
Arguments:
    x, q: vector of quantiles representing the number of failures in a
          sequence of Bernoulli trials before success occurs.
```

Multinomial distribution

- it is a generalization of the binomial distribution to the case with more than 2 possible outcomes
- labeling the disjoint outcomes $A_1, A_2, \ldots, A_r$, we define $P(A_j) = p_j$, with $1 \leq j \leq r$
- in n independent trials, x_j denotes the number of times that A_j occurs
- assuming, by construction, $n = x_1 + x_2 + \ldots + x_r$, we have

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_r = x_r | p_1, p_2, \ldots p_r, n) = \frac{n!}{x_1! x_2! \ldots x_r!} p_1^{x_1} p_2^{x_2} \ldots p_r^{x_r}$$

Properties

- the expectation for class A_j is $E[x_j] = np_j$
- the variance for class A_j is $Var(x_j) = np_j(1 - p_j)$
- the covariance for classes A_i, A_j is $cov(x_i, x_j) = -n\, p_i\, p_j$
- when n becomes large, the distribution tends to a multinormal distribution

Multinomial distribution - exercise

Problem

- in a certain town, at 20:00, 30% of the TV audience watches the news, 25% a TV show, and the rest other programs
- What is the probability that, selecting 7 random viewers, exactly 3 watch the news and at least 2 watch the TV show ?

Solution

- the probabilities are $p_1 = 3/10$, $p_2 = 1/4$, $p_3 = 9/20$
- the sum of the trials $i + j + k = 7$
  - we write

$$P\left(i, j, k \middle| n = 7\right) = \frac{7!}{i!\, j!\, k!} \left(\frac{3}{10}\right)^i \left(\frac{1}{4}\right)^j \left(\frac{9}{20}\right)^k$$

  - and we compute

$$\begin{aligned} P(i = 3, j \geq 2 | n = 7) &= P(3,2,2|7) + P(3,3,1|7) + P(3,4,0|7) \\ &= \frac{7!}{3!\, 2!\, 2!} \left(\frac{3}{10}\right)^3 \left(\frac{1}{4}\right)^2 \left(\frac{9}{20}\right)^2 + \frac{7!}{3!\, 3!} \left(\frac{3}{10}\right)^3 \left(\frac{1}{4}\right)^3 \left(\frac{9}{20}\right) \\ &+ \frac{7!}{3!\, 4!} \left(\frac{3}{10}\right)^3 \left(\frac{1}{4}\right)^4 \simeq 0.103 \end{aligned}$$

Multinomial distribution marginalization

- let suppose we have a multinomial distribution $P(X_1, X_2, \dots X_r)$ and we want to find the marginal probability $P(X_1)$

$$\begin{aligned} P(X_1) &= \sum_{x_2+x_3+\dots+x_r=n-x_1} \frac{n!}{x_1!x_2!\dots x_r!} p_1^{x_1} p_2^{x_2} \dots p_r^{x_r} \\ &= \frac{n!}{x_1!(n-x_1)!} p_1^{x_1} \sum_{x_2+x_3+\dots+x_r=n-x_1} \frac{(n-x_1)!}{x_2!\dots x_r!} p_2^{x_2} \dots p_r^{x_r} \\ &= \frac{n!}{x_1!(n-x_1)!} p_1^{x_1} (p_2+\dots+p_r)^{n-x_1} \\ &= \frac{n!}{x_1!(n-x_1)!} p_1^{x_1} (1-p_1)^{n-x_1} \end{aligned}$$

- where the multinomial expansion has been used, and also the fact that $p_1 + p_2 + \dots + p_r = 1$
- the obtained distribution coincides with the binomial distribution

Poisson process

- let's consider an event that might happen at a given time, with the following conditions:
  - the probability of 1 count in Δt is proportional to Δt itself, with Δt a 'small' value
  - calling r, the intensity of the process,

$$p = P(\text{'1 count in } \Delta t\text{'}) = r\Delta t$$

- moreover:
  - $P(\geq 2 \text{ counts}) \lll P(1 \text{ count})$
  - what happens in one interval does not depend on other intervals ➜ it has a memory-less property

Examples

- accidents occurring at an intersection
- γ-s emitted from a radioactive substance
- customers entering a post office
- earthquakes in Italy

Poisson distribution

- the Poisson distribution can be derived by the Binomial distribution, in the limit where the rate of success, p, is very small
- we divide a finite time interval, T, in n small intervals:

$$T = n\,\Delta T$$

- and we consider the possible occurrence of an event, an independent Bernoulli trial, in each small interval Δt

$$p = r\,\Delta T = r\frac{T}{n}$$

- if the number of of trials is large, the total number of successes, np, is however considerable: $np = rT = \lambda$
- mathematically, in the limit $p \to 0$, $n \to \infty$ and $np = \lambda$ remaining constant, we get

$$\mathrm{Bn}\left(r\middle|n\,p\right) \to Poi\left(r\middle|\lambda\right)$$

- λ depends only on the intensity of the process, r, and on the finite time of observation

$$\mathrm{Poi}(r|\lambda) = \frac{\lambda^r}{r!}\exp(-\lambda)$$

Poisson distribution

- Given the Poisson distribution function:

$$\mathrm{Poi}\left(r\middle|\lambda\right) = \frac{\lambda^r}{r!}\exp(-\lambda)$$

- the expected value and variance are

$$E[x] = \lambda \quad \text{and} \quad Var(x) = \lambda$$

- Asymptotically, for growing λ values, the Poisson distribution becomes identical to the normal distribution

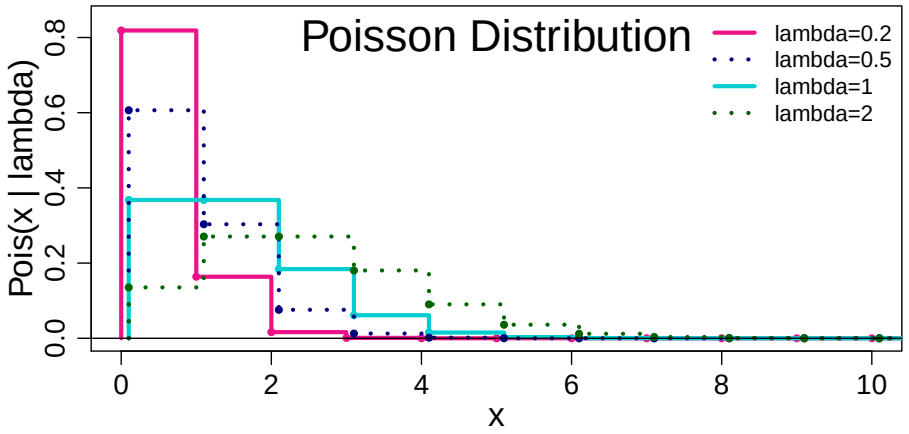
  the similarity is rather close already at $\lambda = 20$
- an interesting property is:

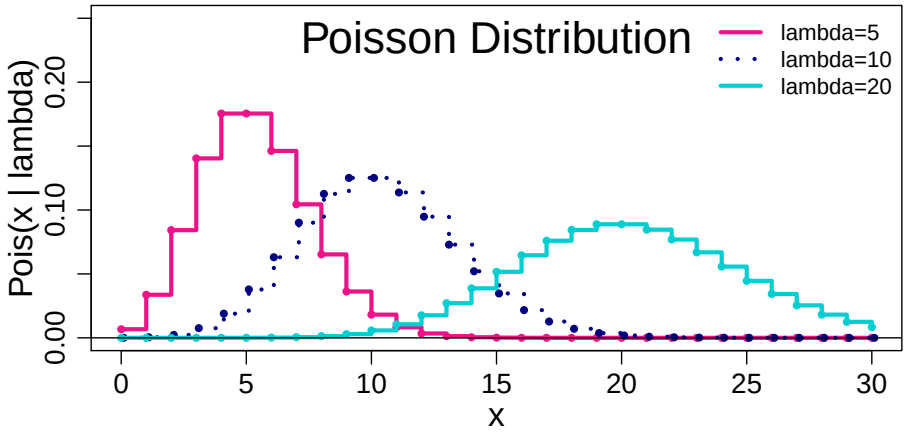
$$\mathrm{Poi}\left(r\middle|\lambda\right) = Poi\left(r-1\middle|\lambda\right)\frac{\lambda}{r}$$

- it is possible to demonstrate that the sum of any independent Poisson variables is itself a Poisson variable with mean value equal to the sum of the individual means

Poisson distribution examples

- the distribution is very asymmetric for small λ and it has a tail to the right of the mean
- the distribution gets increasingly symmetric for higher values of λ
- already for $\lambda = 20$ is very similar to the normal distribution (but it has only integer values)

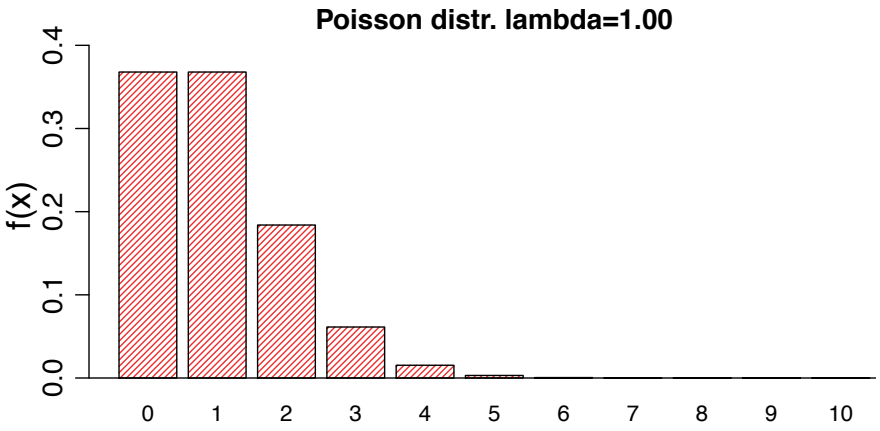




Poisson distribution - exercise 1

Problem

- the average number of received wrong phone calls per week is 7
- what is the probability to get, tomorrow, A) two wrong calls ? B) at least one wrong call ?



Solution

- assuming we get a large number of calls, the number of wrong calls follows, to a good approximation, a Poisson distribution
- we assume $\lambda = 1$

```
P(2|lambda) = 0.184
P(>=1|lambda) = 0.632
```

```
lambda <- 1
x <- 0:10
ap <- dpois(x,lambda)
barplot(ap, names=x, col='firebrick2', xlab='x', ylab='f(x)', density=30,
        main = sprintf("Poisson distr. lambda=%.2f",lambda),
        ylim=c(0,0.415),
        cex.lab=1.5, cex.axis=1.25, cex.main=1.25, cex.sub=1.5)
cat(paste(c("P(2|lambda) = ", ap[3],'\n')))
cat(paste(c("P(>=1|lambda) = ", 1 - ap[1],'\n')))
```

Poisson distribution - exercise 2

Problem

- a radioactive substance emits on average 3.9 α/s per gram
- compute the probability that, in the next second, the number of emitted alpha particles is

  A) at most 6

  B) at least 2

  C) at least 3 and at most 6

Solution

- every gram of element has n atoms
- From the information we have, $E[X] = np = \lambda = 3.9$

$$P(x|\lambda) = \frac{\lambda^x}{x!} \exp(-\lambda)$$

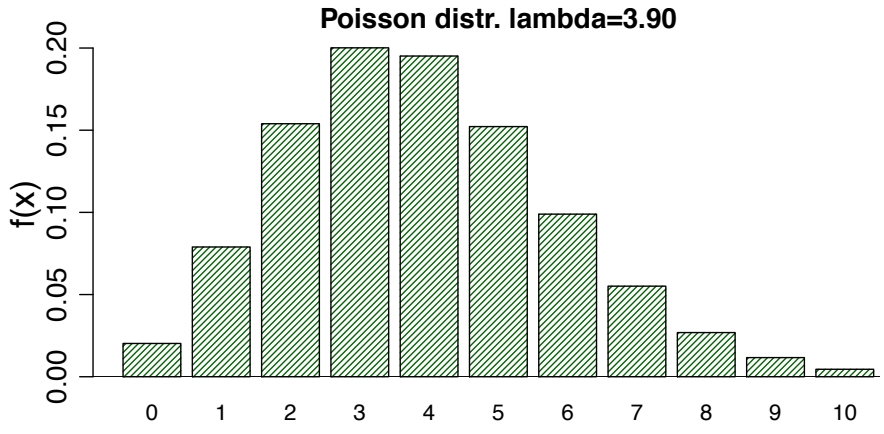
$$\text{A)} \quad P(x \leq 6) = \sum_{x=0}^{6} \frac{3.9^x}{x!} \exp(-3.9)$$

$$\text{B)} \quad P(x \geq 2) = 1 - P(x \leq 1) = 1 - \sum_{x=0}^{1} \frac{3.9^x}{x!} \exp(-3.9)$$

$$\text{C)} \quad P(3 \leq x \leq 6) = \sum_{x=3}^{6} \frac{3.9^x}{x!} \exp(-3.9)$$

Poisson distribution - exercise 2

```
P(<=6) = 0.899483035093612
P(>=2) = 0.900814633915558
P(2<X<=6) = 0.646357932463829
```




```
lambda <- 3.9
x <- 0:10
ap <- dpois(x,lambda)

barplot(ap, names=x, col='darkgreen', xlab='x', ylab='f(x)', density=30,
        main = sprintf("Poisson distr. lambda=%.2f",lambda),
        ylim=c(0,0.21),
        cex.lab=1.5, cex.axis=1.25, cex.main=1.25, cex.sub=1.5)
abline(0,0)

P_6 = sum(ap[x<=6])
P_2 = 1 - sum(ap[x<=1])

cat(paste(c("P(<=6) = ", P_6,'\n')))
cat(paste(c("P(>=2) = ", P_2,'\n')))

pp <- ppois(x, lambda)
P_36 = pp[x==6] - pp[x==2]
cat(paste(c("P(2<X<=6) = ", P_36,'\n')))
```

Pascal or Negative Binomial distribution

- the probability of obtaining the r-th success in n trials, is given by the Negative Binomial, or Pascal, distribution
- since in $n-1$ trials we had $r-1$ successes, the probability is given by the Binomial distribution:

$$\mathrm{Bn}\left(r\middle|n,p\right) = \binom{n-1}{r-1} p^{r-1}(1-p)^{n-1-r+1} = \binom{n-1}{r-1} p^{r-1}(1-p)^{n-r}$$

- but we got the r-th success at the n-th trial, therefore

$$\mathrm{Bneg}\left(r\middle|n,p\right) = \binom{n-1}{r-1} p^{r}(1-p)^{n-r}$$

- the expected value and variance are

$$E[x] = \frac{r}{p} \quad \text{and} \quad Var(x) = \frac{r(1-p)}{p^2}$$

Pascal distribution - exercise

Problem

- Ann and Maggie are playing cards until one of them wins 5 games
- suppose all games are independent and the probability that Ann wins is 58%

  A) what is the probability that they complete in 7 games

  B) if the series ends in 7 games, what is the probability that Ann wins ?

Solution to A

- `X`: number of games played until Ann wins 5 games
- `Y`: number of games played until Maggie wins 5 games
- both `X` and `Y` follow a Pascal distribution

$$P(X=7, r=5) = \binom{6}{4} 0.58^5 0.42^2 = 0.174$$

$$P(Y=7, r=5) = \binom{6}{4} 0.42^5 0.58^2 = 0.066$$

- we get $P(X=7, r=5) + P(Y=7, r=5) = 0.24$

Pascal distribution - exercise

Solution to B

- A: Ann wins
- B: the series ends in 7 games

$$P(A|B) = \frac{P(AB)}{P(B)} = \frac{P(X=7)}{P(X=7)+P(Y=7)} = \frac{0.17}{0.24} = 0.71$$

Solution with R

```
dnbinom(x, size, prob, mu)

The negative binomial distribution with 'size' = n and 'prob' = p
...
for x = 0, 1, 2, ..., n > 0 and 0 < p <= 1.

This represents the number of failures which occur in a sequence
of Bernoulli trials before a target number of successes is
reached.  The mean is mu = n(1-p)/p and variance n(1-p)/p^2.

P_Ann    <- dnbinom(2,5,0.58) # 0.173672
P_Maggie <- dnbinom(2,5,0.42) # 0.0659468
```


Exercise : Binomianl/Poisson

Defective screws

- a company produces screws
- the probability of a screw to be defective is $p = 0.015$
- a box with $n = 100$ screws is packaged.

Compute:

A) the probability that all screws are non defective

B) the defective screws distribution comparing the Binomial and Poisson distributions

C) how many extra screws should the box contain in order to have $n = 100$ non defective screws with probability greater than 80%

Review of Probability Distributions - II

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Stat Lect. 3

Poisson Process

- Let $\{N(t), t \geq 0\}$ be a Poisson process
- $N(t)$ represents the number of events occurred at or prior to time t
- a Poisson process has the properties:
  - stationarity: given two equal time intervals, Δt_1 and Δt_2, the probability of n events in Δt_1 is equal to that in Δt_2
  - independent increments: the probability of n events in $[t, t+s]$ is independent of how many events have accurred earlier
  - orderliness: the occurrence of two or more events in a small time is practically impossible.
- given $t > 0$, $N(t)$ is the number of events occurred between 0 and t.
  $N(t)$ is a Poisson random variable with parameter λt:

$$P\left(N(t) = n\right) = \frac{(\lambda t)^n e^{-\lambda t}}{n!}$$

- a Poisson process is stationary and possesses independent increments: at any time t the process probabilistically starts all over again

Poisson Process: example

Problem

- children are born at a Poisson rate of 5/day in a countryside hospital. What is the probability that:

a) at least two babies are born in the next six hours ?

b) no babis are born in the next two days ?

Solution

- $N(t)$ is the number of babes born from 0 to time t
- we assume that $\{N(t), t \geq 0\}$ is a Poisson process:
  it is stationary, it has independent increments and the probability of simultaneous births is zero
- one day is a time unit $\rightarrow \lambda = E[N(1)] = 5$

$$P(N(t) = n) = \frac{(5t)^n e^{-5t}}{n!}$$

a) the probability of two babies in the next six hours, i.e. $t = 6/24 = 1/4$ is

$$P(N(1/4) \geq 2) = 1 - P(N(1/4) = 0) - P(N(1/4) = 1) = 0.36$$

b) the probability of no baby born during the next two days is

$$P(N(2) = 0) = \frac{(10)^0 e^{-10}}{0!} = 4.5 \cdot 10^{-5}$$

The Exponential distribution

- Let $\{N(t), t \geq 0\}$ be a Poisson process
- $N(t)$ represents the number of events occurred at or prior to time t
- If T_1 is the time arrival of the 1^{st} event
- T_j represents the elapsed time between the events T_j and T_{j-1}
- the ordered set $\{T_1, T_2, \ldots, T_n\}$ is a sequence of inter-arrival times of the Poisson process
- setting

$$P\left(N(t) = n\right) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

- we can evaluate the probability distribution function of the random variables T_j:

$$P\left(T_1 > t\right) = P\left(N(t) = 0\right) = e^{-\lambda t}$$

$$P\left(T_1 \leq t\right) = 1 - P\left(T_1 > t\right) = 1 - e^{-\lambda t}$$

- a Poisson process is stationary and possesses independent increments: at any time t the process probabilistically starts all over again

➜ the inter-arrival time of any two consecutive events has the same distribution as T_1

The Exponential distribution

- the cumulative distribution is therefore

$$F(t) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

- and the probability density function is

$$f(t) = \frac{dF(t)}{dt} = \begin{cases} \lambda e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

- the expected value and variance are

$$E[x] = \frac{1}{\lambda} \quad \text{and} \quad Var(x) = \frac{1}{\lambda^2}$$

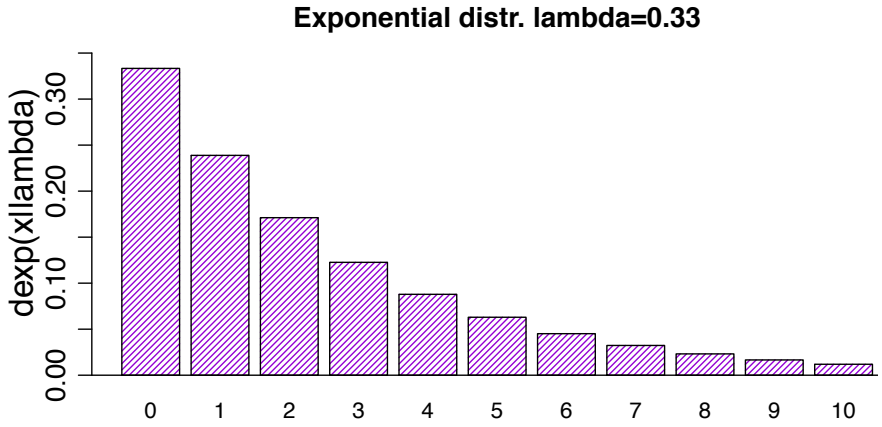
Examples

- the inter-arrival time between two customers in a shop
- the duration of my next telephone call
- the time between two accidents at an intersection
- time until the next baby is born in a hospital
- the time to failure of the next chip in a large group of such devices when all of them are initially fault free

Exponential distribution - exercise

Problem

- suppose that every three months, an earthquake of some entity happens in Italy
- what is the probability that the next earthquake happens after three but before seven months ?



Solution

- X: the time, in months, until the next earthquake
- X is exponential with $\lambda = 1/3$

```
P(3<X<7) = 0.270907473307037
```

$$P(3 < X < 7) = F(7) - F(3) = (1 - e^{7/3}) - (1 - e^{3/3})$$

```
lambda <- 1/3; x <- 0:10; ap <- dexp(x,lambda)

barplot(ap, names=x, col='darkviolet', xlab='x', ylab='dexp(x|lambda)',
        density=30,
        main = sprintf("Exponential distr. lambda=%.2f",lambda),
        ylim=c(0,0.375),
        cex.lab=1.5, cex.axis=1.25, cex.main=1.25, cex.sub=1.5)

 cat(paste(c("P(3<X<7) = ", pexp(7,lambda) - pexp(3,lambda),'\n')))
```

The memory-less feature of the Exp distr

- a non-negative random variable X is memory-less if

$$P(X > s + t | X > t) = P(X > s) \qquad \forall s, t \geq 0$$

- since

$$P(X > s + t, X > t) = P(X > s + t | X > t) P(X > t)$$

$$\frac{P(X > s + t, X > t)}{P(X > t)} = P(X > s)$$

- and

$$P(X > s + t) = P(X > s) \cdot P(X > t)$$

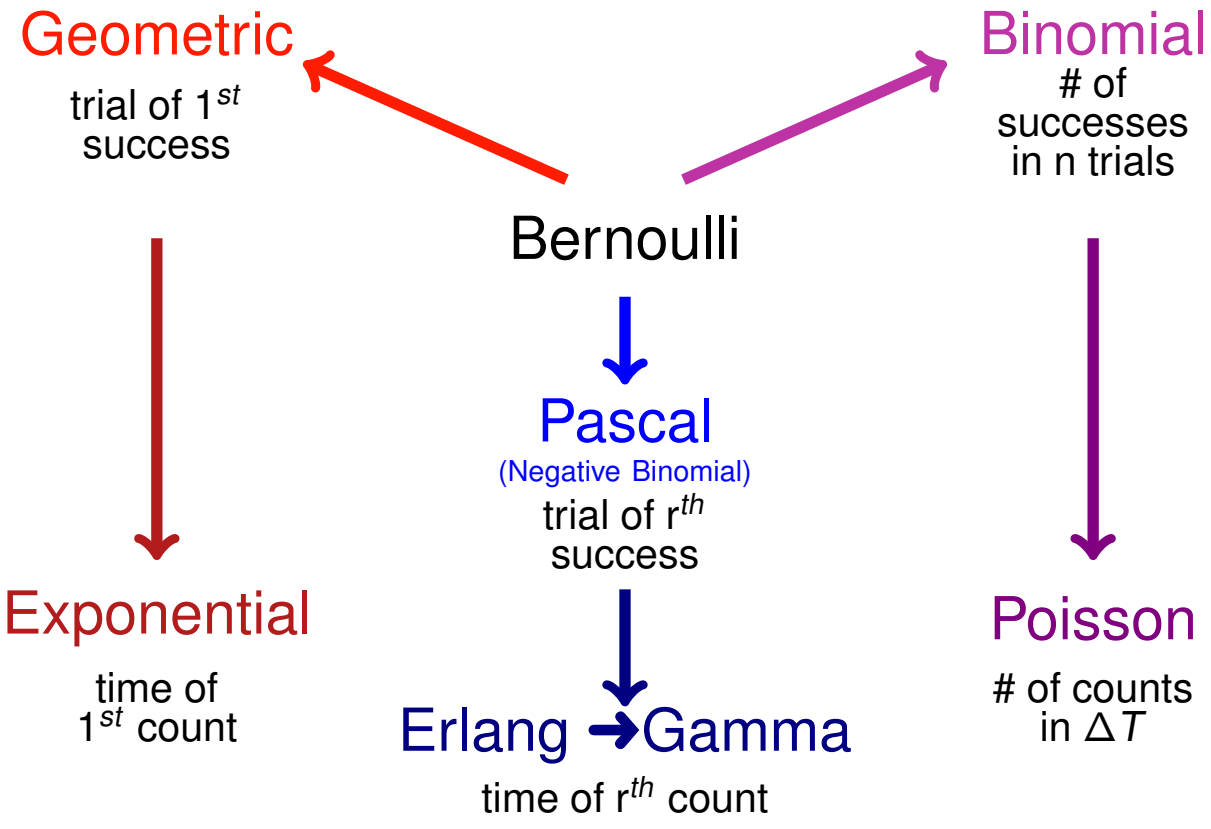
- since

$$P(X > s + t) = 1 - (1 - \exp(-\lambda(s + t))) = \exp(-\lambda(s + t)) = \exp(-\lambda s) \exp(-\lambda t)$$

- and

$$P(X > s) = 1 - (1 - \exp(-\lambda s)) = \exp(-\lambda s))$$
$$P(X > t) = 1 - (1 - \exp(-\lambda t)) = \exp(-\lambda t)$$

Summary of discrete probability distributions



Hypergeometric distribution

- suppose we have a box containing B black stones and $N - B$ white stones and we draw them, randomly, without replacement
- if the number of drawn items, n, does not exceed the number of black or white balls, i.e. $n \leq min(B, N - B)$
- and if X identifies the number of black stones extracted, its probability distribution follows the Hypergeometric distribution

$$P(x \mid N, B, n) = \frac{\binom{B}{x}\binom{N-B}{n-x}}{\binom{N}{n}} \quad \text{with } x = \{0, 1, 2, \ldots, n\}$$

- the expected value and variance are

$$E[x] = \frac{nB}{N} \quad \text{and} \quad Var(x) = \frac{nB(N-B)}{N^2}\left(1 - \frac{n-1}{N-1}\right)$$

- note that if sampling is done with replacement, X follows a binomial distribution with parameters n and B/N

$$E[x] = n\frac{B}{N} \quad \text{and} \quad Var(x) = n\frac{B}{N}\left(1 - \frac{B}{N}\right)$$

Standard Continuous Distributions

The Uniform Distribution

- a random variable $X \sim \mathcal{U}(a, b)$ follows a uniform distribution if the pdf is given by the following:

$$f(X) = \begin{cases} \frac{1}{b-a} & a < x < b \\ 0 & otherwise \end{cases}$$

- the cumulative density function is

$$F(X) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & x > b \end{cases}$$

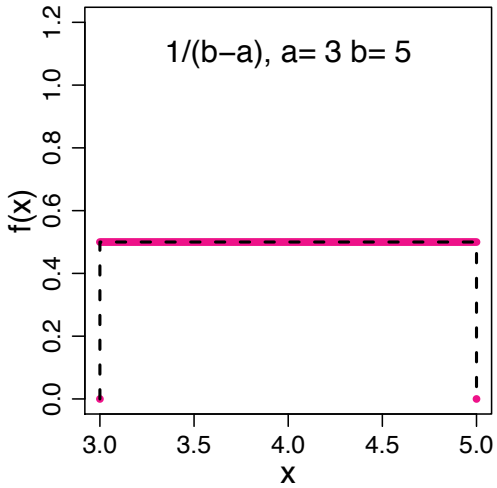
- and the expected value and variance are

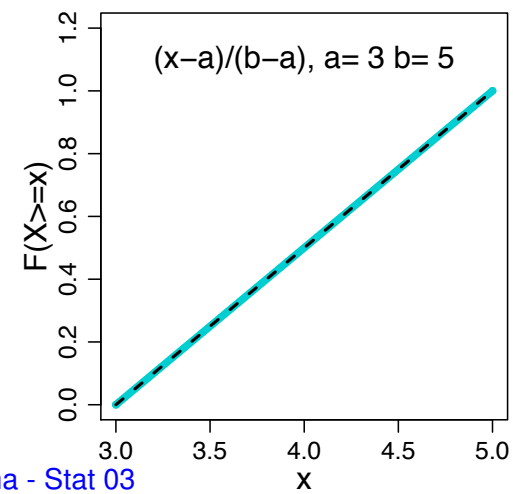
$$E[X] = \frac{a+b}{2} \quad \text{and} \quad Var(x) = \frac{(b-a)^2}{12}$$

The Uniform distribution in R

- we have four pre-defined functions:
  - `dunif(x, min=0, max=1)` returns the probability density function
  - `punif(q, min=0, max=1)` gives the cumulative distribution function
  - `qunif(p, min=0, max=1)` is the quantile returning function
  - `runif(n, min=0, max=1)` generate a vector with random values from a uniform distribution
- if not specified, the default interval is $(0, 1)$

```
x <- seq(3, 5, 0.01)
a <- min(x); b <- max(x)
xp <- c(a, x, b)

yp1 <- c(0, dunif(x, a, b), 0)
plot(xp, yp1)

yp2 <- c(0, punif(x, a, b), 1)
plot(xp, yp2)
```





Example: sum of two Uniform distributions

- let's suppose two random variables, x_1 and x_2 follow a uniform distribution, $x_j \sim \mathcal{U}(0,1)$
- let's compute the $y = x_1 + x_2$ distribution function

$$f(y) = \begin{cases} y & 0 \le y \le 1 \\ 2-y & 1 \le y \le 2 \\ 0 & \text{otherwise} \end{cases}$$

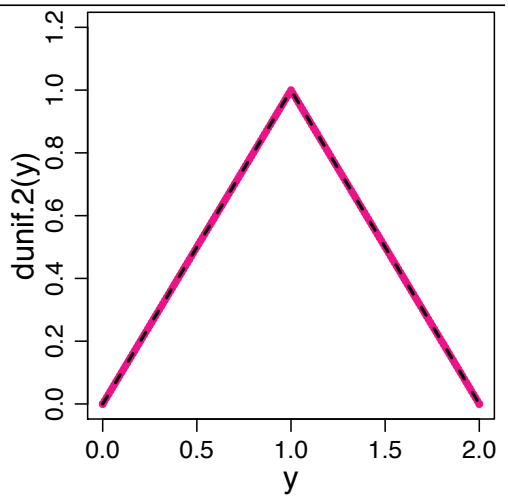
- integrating $f(y)$ in the domain we get

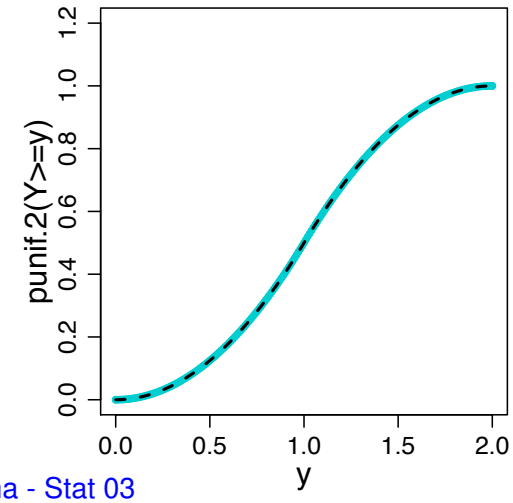
$$F(y) = \begin{cases} 0 & y < 0 \\ y^2/2 & 0 \le y \le 1 \\ -y^2/2 + 2y - 1 & 1 \le y \le 2 \\ 1 & y > 2 \end{cases}$$

- the expected value and variance are

$$E[X] = \int_0^1 y f(y) dy = 1\ , \quad E[X^2] = \frac{7}{6}$$

$$Var(x) = \int_0^1 (y-1)^2 f(y) dy = \frac{1}{6}$$





Example: sampling from a user's pdf

- all cumulative distributions are monotone increasing functions in the interval $[0,1]$
- if the analytical form of $F(X)$ is known, it is also invertible:

$$F^{-1}(y) = \inf\{x : F(x) \ge y\} \quad u \in [0,1]$$

- there is a 1:1 correspondence between CDFs, since they have the same image

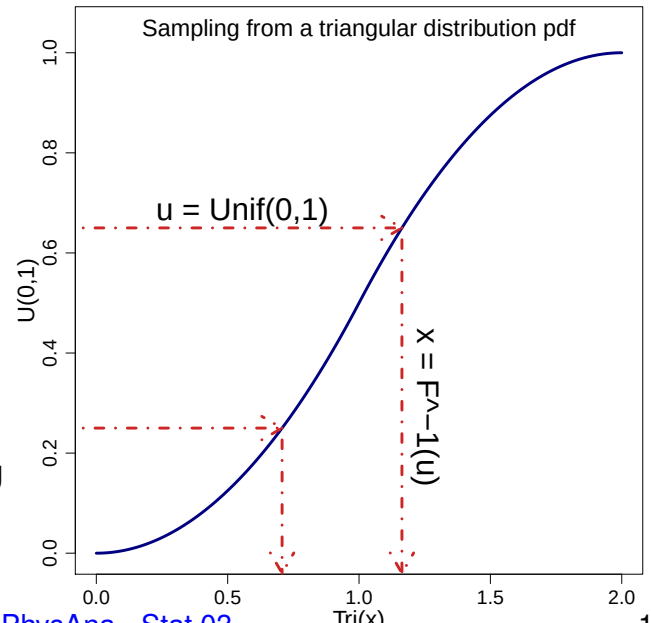
- given X and Y with CDFs $F(X)$ and $G(Y)$
- we ask for the same probability, and search for x_i and y_i such that

$$F(x_i) \equiv P(X \le x_i) = G(y_i) \equiv P(Y \le y_i)$$

- assuming

$$\begin{aligned} G(y) &= \mathcal{U}(0,1) = u \\ &\rightarrow F(x_i) = u \\ &\rightarrow x_i = F^{-1}(u) \end{aligned}$$

- this is called the inverse transform sampling method

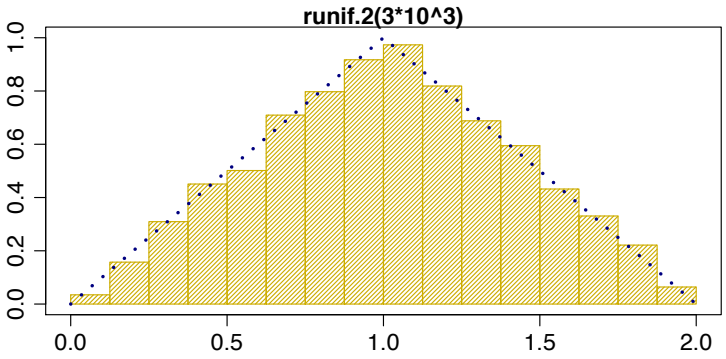


The Inverse Transform on the Triangular distribution

Algorithm

1) extract a sample from $\mathcal{U}(0,1)$
2) compute $F^{-1}(u) = x$
3) release x as sampled from our $F(x)$

- we define the new `rinf.2()` and `qinf.2()` functions




```
dunif.2 <- function(x) {
  duni2 <- ifelse(x <= 1,
                  x,
                  2 - x)
  return (duni2)
}

runif.2 <- function(n) {
  us <- runif(n)
  runi2 <- ifelse(us <= 0.5,
                  sqrt(2 * us),
              2 - sqrt(2*(1 - us)))
  return (runi2)
}
```

```
punif.2 <- function(x) {
  puni2 <- ifelse(x <= 1,
                  0.5 * x^2,
            (4*x - x^2 - 2)/2)
  return (puni2)
}

qunif.2 <- function(p) {
  quni2 <- ifelse(p<=0.5,
                  sqrt(2*p),
              2 - sqrt(2*(1 - p)))
  return (quni2)
}
```


Integrating a pdf with R

- we have computed the mean and variance values, analytically

$$E[X] = \int_0^1 y\, f(y)\, dy \quad \text{and} \quad E[X^2] = \int_0^1 y^2\, f(y)\, dy$$

- and now we ask R to do it for us

```
Evaluate the mean value and variance, by integration
The mean value of the distribution is: 1
and the variance: 0.166666666666667
```

```
# Evaluate integral of pdf
# using a anonymous function
E.X.integral <- integrate(function(x) {x * dunif.2(x)},
                          lower=0, upper=2)

E.X <- E.X.integral$value

E.X2.integral <- integrate(function(x) {x^2 * dunif.2(x)},
                           lower=0, upper=2)

E.X2 <- E.X2.integral$value

Var.X <- E.X2 - E.X^2

cat(paste("The mean value of the distribution is: ", E.X, '\n'))
cat(paste("and the variance: ", Var.X, '\n'))
```

The `integrate()` R function

- an adaptive quadrature of functions of one variable over a finite or infinite interval

```
integrate(f, lower, upper, ...)
```

- `f()` is an R function taking a numeric first argument and returning a numeric vector of the same length.

```
x.integral <- integrate(function(x) {x*dunif.2(x)}, lower=0, upper=2)
```

```
> class(x.integral)
[1] "integrate"
> summary(x.integral)
             Length Class  Mode
value        1      -none- numeric
abs.error    1      -none- numeric
subdivisions 1      -none- numeric
message      1      -none- character
call         4      -none- call
> names(x.integral)
[1] "value"        "abs.error"    "subdivisions" "message"      "call"
```

```
> x.integral$value
[1] 1
> x.integral$abs.error
[1] 1.110223e-14
> x.integral$subdivisions
[1] 2
```

```
> x.integral$message
[1] "OK"
> x.integral$call
integrate(f = function(x) {
    x * dunif.2(x)
}, lower = 0, upper = 2)
```


Inequalities

- we will discuss three important inequalities:
  - Markov's inequality
  - Jensen's inequality
  - Chebyshev's inequality

- they are very useful when we do not have enough information about the distribution of random variables
- but we can calculate their expected values and/or variances
- using the, bounds on probabilities can be derived

Markov's Inequality

- X is a non-negative random variable with $E[x] = \mu$
- for any $k > 0$

$$P(X \geq k) \leq \frac{\mu}{k}$$

Proof

- let's do it for a discrete random variable X, with pdf $p(x)$ over a set A
- let $B \subset A$, defined as $B = \{x \in A : x \geq k\}$

$$E[X] = \sum_{x \in A} x\ p(x) \geq \sum_{x \in B} x\ p(x)$$

$$\geq k \sum_{x \in B} p(x) = k\ P(X \geq k)$$

- in a similar way it can be demonstrated for continuous variables

Markov's Inequality application

Exercise

- a post office handles, on average, 10^4 letters per day

(a) what is the probability that, tomorrow, it will handle at least $1.5\ 10^4$ letters ?

(b) and less than $1.5\ 10^4$ letters ?

Solution

- the average value of handled letters is $E[X] = 10^4$
- from Markov's inequality

$$P(X \geq 1.5\ 10^4) \leq \frac{E[X]}{1.5\ 10^4} = \frac{2}{3}$$

- the second question is answered using the normalization of the probability

$$P(X < 1.5\ 10^4) = 1 - P(X \geq 1.5\ 10^4) = \frac{1}{3}$$

Jensen's Inequality

- the variance of a random variable is always a positive value

$$Var(X) = E[X^2] - (E[X])^2 \geq 0$$

- therefore the most basic moment inequality is

$$E[X^2] \geq (E[X])^2$$

Jensen's inequality

- let X be a random variable with finite mean $\mu = E[X]$
- let $g(x) : \mathbb{R} \longmapsto \mathbb{R}$, a convex function (i.e. $d^2g/dx^2 > 0$)

$$g\left(E[X]\right) \leq E\left[g\left(X\right)\right]$$

Example

- X, positive random variable with $E[X] = \mu$, finite
- we consider $g(x) = x^{-1}$, $x > 0$
- g is convex, since $g^{''} = 2 \cdot x^{-3} > 0$, $\forall x > 0$
- from Jensen's inequality:

$$E\left[\frac{1}{x}\right] \geq \frac{1}{E[X]} \Longleftrightarrow E\left[\frac{1}{x}\right] \cdot \frac{1}{E[X]} \geq 0$$

Chebyshev's Inequality

- X is a non-negative random variable with $E[x] = \mu$ and $Var(x) = \sigma^2$
- for any $k > 0$

$$P\left(\left|X - \mu\right| \geq k\right) \leq \frac{\sigma^2}{k^2}$$

Proof

- we know that

$$(X - \mu)^2 \geq 0$$

- therefore, applying Markov's inequality

$$P\left(\ (X - \mu)^2 \geq k^2\right) \leq \frac{E\left[(X - \mu)^2\right]}{k^2} = \frac{\sigma^2}{k^2}$$

but since $(X - \mu)^2 \geq k^2 \ \rightarrow \left|X - \mu\right| \geq k$, Chebyshev's inequality follows

Chebyshev's Inequality implications

- if $k = r\sigma$

$$P\left(\left|X-\mu\right| \geq r\sigma\right) \leq \frac{\sigma^2}{r^2\sigma^2} = \frac{1}{r^2}$$

- meaning that the probability that X deviates from its expected value at least r standard deviations is less than $1/r^2$
- as an example

  $P(\left|X-\mu\right| \geq 2\sigma) \leq 1/4 = 25\%$

  $P(\left|X-\mu\right| \geq 4\sigma) \leq 1/16 = 6.25\%$

  $P(\left|X-\mu\right| \geq 10\sigma) \leq 1/100 = 1\%$

- since

$$1 - P\left(\left|X-\mu\right| < r\sigma\right) = P\left(\left|X-\mu\right| \geq r\sigma\right) \leq \frac{1}{r^2}$$

- it follows that

$$P\left(\left|X-\mu\right| < r\sigma\right) \geq 1 - \frac{1}{r^2}$$

Chebyshev's Inequality application

Exercise

- the same a post office handles, on average, 10^4 letters per day, with a variance of 2000 letters

(a) what is the probability it will handle between 8000 and 12000 letters, tomorrow ?

Solution

- we know that $E[X] = 10^4$ and $\sigma^2 = Var(X) = 2 \cdot 10^3$
- we need to evaluate

$$\begin{aligned} P(8\cdot 10^3 < X < 12\cdot 10^3) &= P(\left|X-10^4\right| < 2\cdot 10^3) \\ &= 1 - P(\left|X-10^4\right| \geq 2\cdot 10^3) \end{aligned}$$

- since $k\sigma = 2000 \rightarrow k = 2000/\sigma = 2000/\sqrt{2000}$
- therefore

$$P(\left|X-10^4\right| \geq 2\cdot 10^3) = 1 - P(8\cdot 10^3 < X < 12\cdot 10^3) \geq \frac{2\cdot 10^3}{(2\cdot 10^3)^2} = 5\cdot 10^{-4}$$

- and

$$P(8\cdot 10^3 < X < 12\cdot 10^3) \geq 1 - 5\cdot 10^{-4} = 0.9995$$

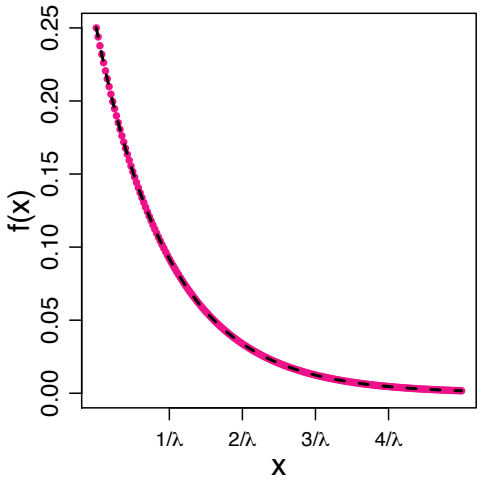
Exponential Random Variables

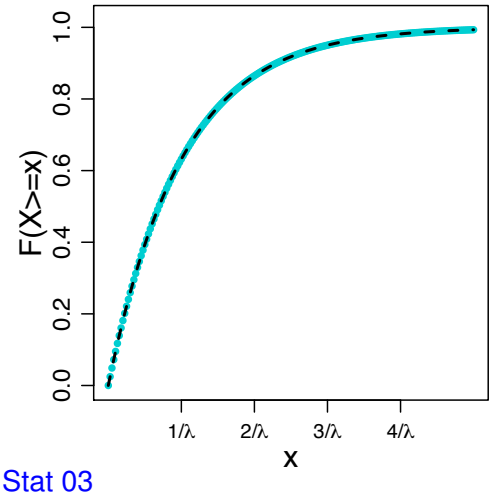
- consider a Poisson process $\{N(t) : t \geq 0\}$ where $N(t)$ represents the number of events that happened at or before time t:
  - T_1 is the time of the first event
  - T_2 is the time between the first and the second event
  - T_j is the time between events $j-1$ and j
- this sequence, also called inter-arrival times follows an exponential distribution, $X \sim \text{Exp}(\lambda)$

$$f(x) = \lambda\, e^{-\lambda x} \quad \text{with } \lambda > 0$$

- the expected value and variance are

$$E[X] = \frac{1}{\lambda} \qquad Var(x) = \frac{1}{\lambda^2}$$





$\text{Exp}(\lambda)$ property: lack of memory

- important feature of the exponential distribution is the memory-less property
- a positive random variable X is called memory-less if $\forall s,t \geq 0$,

$$P\left(X > s+t \,\middle|\, X > s\right) = P\left(X > t\right)$$

--→ suppose you are in front of an elevator
--→ and you have already waited for three minutes $(s = 3)$
--→ the probability to wait for another two minutes $(t = 2)$ is the same as you just arrived in front of the same elevator
⇉ but this is only true for an exponential distribution

Proof

- our requirements is

$$\begin{aligned} P\left(X > s+t \,\middle|\, X > s\right) &= \frac{P\left(X > s+t\right)}{P\left(X > s\right)} \\ &= \frac{e^{-(s+t)/\lambda}}{e^{-s/\lambda}} \\ &= e^{-t/\lambda} = P\left(X > t\right) \end{aligned}$$

- therefore our hypothesis follows: X is memory-less

Analogy between $\mathrm{Exp}(\lambda)$ and $\mathrm{Geo}(p)$

- a Bernoulli trial is performed successively and independently
  - the number of trials until the first success occurs follows $\mathrm{Geo}(p)$
  - but also the number of trials between two consecutive successes follows $\mathrm{Geo}(p)$
- let's now consider a Poisson process
  - the time it will take until the first event occurs is $\mathrm{Exp}(\lambda)$
  - the time between two consecutive events is also $\mathrm{Exp}(\lambda)$

➜ moreover $\mathrm{Exp}(\lambda)$ is the only memory-less continuous distribution, and $\mathrm{Geo}(p)$ is the only memory-less discrete distribution

The Erlang distribution

- let's consider again a Poisson process $\{N(t) : t \geq 0\}$ where $N(t)$ represents the number of events that happened at or before time t:
  - being T_j the time between events $j-1$ and j, the sequence $\{T_1, T_2, \ldots\}$ distributes as $\mathrm{Exp}(\lambda)$
- let now X be the time of the n-th event
- X follows a so-called Gamma distribution with parameters n and λ

$$f(x) = \frac{x^{n-1}\ \lambda^n\ \mathrm{e}^{-\lambda x}}{(n-1)!}$$

➜ Exponential is the time to wait for the first event to occur

➜ Gamma is the time to wait for the n-th event to occur

- an Erlang distribution with parameters $(1, \lambda)$ is an exponential distribution:

$$\mathrm{Gamma}(1, \lambda) \sim \mathrm{Exp}(\lambda)$$

From Erlang to Gamma distributions

- we want to extend the parameters of the Erlang distribution from (n, λ) to (r, λ), where r is a real and positive number
- the factorial $(n-1)!$ can be extended using the Gamma function, $\Gamma : (0, \infty) \longmapsto \mathbb{R}$:

$$\Gamma(r) = \int_0^\infty x^{r-1} e^{-x} dx$$

- the function has the same property of the factorials:

  $n! = n \cdot (n-1)!$

  $\Gamma(\alpha) = (\alpha - 1) \cdot \Gamma(\alpha - 1)$ with $\alpha > 1$

- if r is integer, we get back the factorials:

$$\Gamma(1) = \int_0^\infty e^{-x}\, dx = 1$$

- and

$$\begin{array}{lcl} \Gamma(2) & = & (2-1)\,\Gamma(2-1) = 1 = 1! \\ \Gamma(3) & = & (3-1)\,\Gamma(3-1) = 2 \cdot 1 = 2! \\ & \cdots & \\ \Gamma(n+1) & = & n! \end{array}$$

The Gamma distribution

- a random variable X follows a gamma distribution, $X \sim \text{Gamma}(\alpha, \lambda)$, if the pdf has the form

$$f(x) = \frac{1}{\Gamma(\alpha)}\, x^{\alpha-1} \lambda^\alpha\, e^{-\lambda x} \quad \text{with } x \geq 0$$

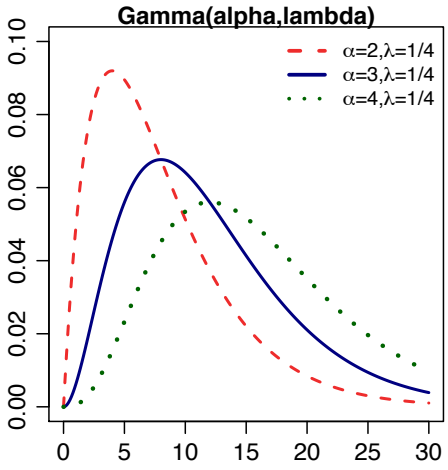
- the parameters $\alpha > 0$ and $\lambda > 0$ are called shape and scale parameters, respectively

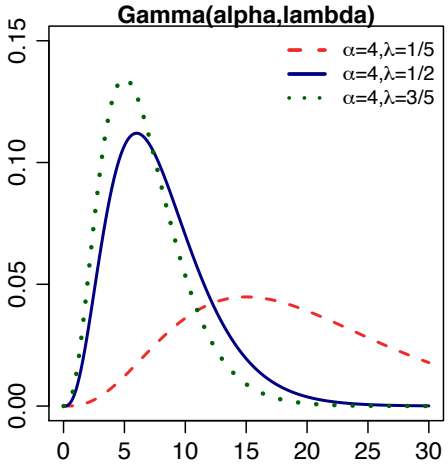
$\rightsquigarrow$ therefore, if $X \sim \text{Gamma}(\alpha, 1)$, then $X/\lambda \sim \text{Gamma}(\alpha, \lambda)$

- the Gamma distribution is a generalization of the exponential density with a mode at some strictly positive m value
- it includes the exponential as a special case and can be very skewed, to being almost a bell-shaped density
- we will show that it arises, naturally, as the density of the sum of a number of independent exponential random variables
- the CDF of the Gamma distribution does not exist in explicit form, therefore the inverse method cannot be used for variate generation
- in Bayesian analysis is a natural conjugate prior for the standard deviation of a normal distribution

Gamma distribution in R

- keeping λ fixed, the maximum of the peak moves to the right with increasing values of α
- a similar behavior can be seen by keeping α fixed, and increasing λ to higher values





Sum of variables with an exponential distribution

- let's suppose we have n independent variables $T_j \sim \mathrm{Exp}(\lambda)$, with $j = 1, \ldots, n$
- we build $Y_n = \sum_{j=1}^{n} T_j$
- it can be proved that $Y_n \sim \mathrm{Gamma}(n, \lambda)$

Proof

- T_j are all independent for $t < 1/\lambda$

$$
\begin{aligned}
E[\exp(Y_n\, t)] &= E[\exp((T_1 + T_2 + \ldots T_n)t)] \\
&= E[\exp(T_1\, t)\exp(T_2\, t)\ldots\exp(T_n\, t)] \\
&= E[\exp(T_1\, t)]E[\exp(T_2\, t)]\ldots E[\exp(T_n\, t)] \\
&= \prod_{j=1}^{n}(1-\lambda t)^{-1} \\
&= (1-\lambda t)^{-n}
\end{aligned}
$$

- therefore $Y_n \sim \mathrm{Gamma}(n, \lambda)$

The Beta distribution

- a random variable X follows a beta distribution, $X \sim \text{Beta}(\alpha, \beta)$, if the pdf is

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\,\Gamma(\beta)}\, x^{\alpha-1}\,(1-x)^{\beta-1} \quad \text{with } 0 \le x \le 1 \text{ and } \alpha, \beta > 0$$

--→ beta densities appear in the study of the median of a sample of random points $\sim \mathcal{U}(0,1)$

--→ let's generate n points $X_j \sim \text{Beta}(\alpha, \beta)$ and assume they are ordered $X_1, X_2, \ldots, X_n$ with $X_{j+1} > X_j$

--→ if $n = 2k+1$ (n is odd), the median is X_{k+1}

--→ if $n = 2k$ (n is even), the median is $(X_k + X_{k+1})/2$

⇉ the median of $2n+1$ random numbers $\sim \mathcal{U}(0,1)$ is $\sim \text{Beta}(n+1, n+1)$

- the expected value and variance are

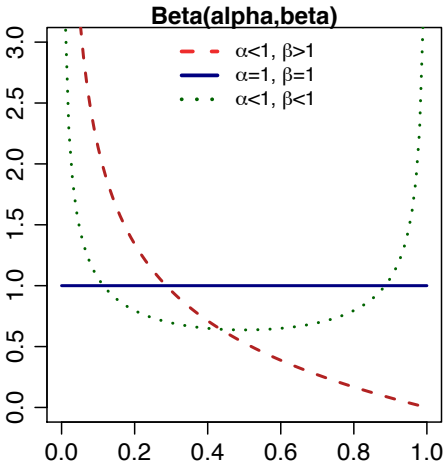
$$E[X] = \frac{\alpha}{\alpha+\beta} \quad \text{and} \quad Var(x) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

- the central moments are

$$E[X^n] = \frac{\Gamma(\alpha+n)\Gamma(\alpha+\beta)}{\Gamma(\alpha+\beta+n)\Gamma(\alpha)}$$

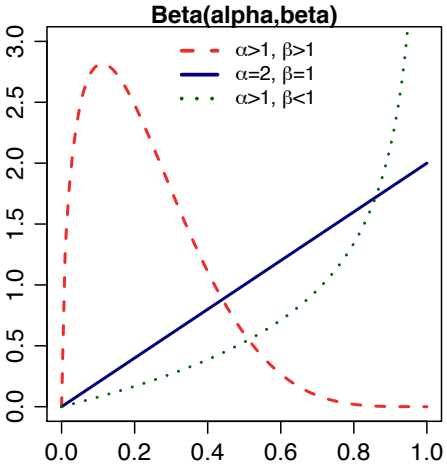
Beta distribution in R

- a beta distribution can be
  - ↱ increasing $\alpha > 1$ and $\beta < 1$
  - ↱ decreasing $\alpha < 1$ and $\beta > 1$
  - ↱ symmetric unimodal $\alpha = \beta$
  - ↱ asymmetric unimodal $\alpha \neq \beta$
  - ↱ U-shaped $\alpha < 1$ and $\beta < 1$



- it cannot be bimodal: it cannot have two local maxima in the interval $[0, 1]$

- note that $\text{Beta}(1,1)$ is simply $\mathcal{U}(0,1)$

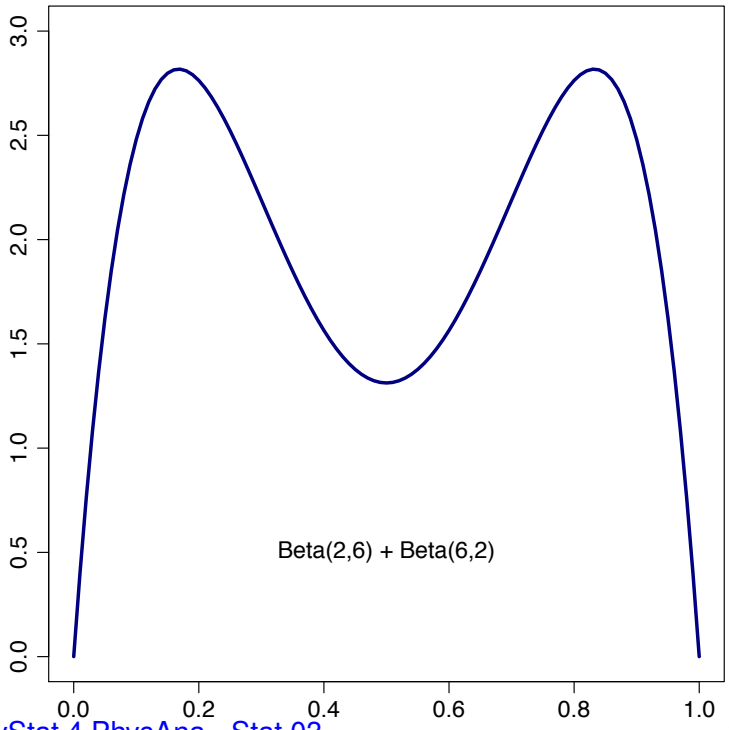


Example: mixture of Beta distributions

- a beta distribution can have only one mode in $[0, 1]$
- in some cases we have to model a random variable that exhibit two modes, for some physical reason

- this can be done by mixing two beta distributions

$$
\begin{aligned}
f(x) &= \frac{1}{2}\mathrm{Beta}(2,6) + \frac{1}{2}\mathrm{Beta}(6,2) \\
&= \frac{1}{2}\left[42x^5(1-x)\right] \\
&+ \frac{1}{2}\left[42x(1-x)^5\right] \\
&= 21\ x\ (1-x)\ \left[x^4 + (1-x^4\right]
\end{aligned}
$$



The Normal distribution

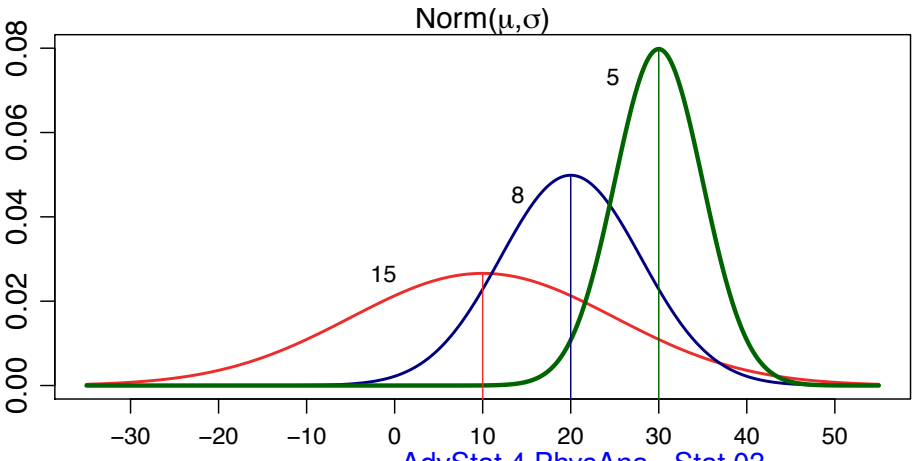
- a random variable X follows a normal distribution, $X \sim \mathrm{N}(\mu, \sigma^2)$, if the pdf is

$$f(x) = \frac{1}{\sigma\sqrt{2\,\pi}}\ \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

--> where μ can be any real number and $\sigma > 0$ and teh distribution is called standard normal

- if $\mu = 0$ and $\sigma = 1$, it is called a standard normal distribution $X \sim \mathrm{N}(0, 1)$
- the expected value and variance are

$$E[X] = \mu \quad \text{and} \quad Var(x) = \sigma^2$$



The sum of independent normal variables

- the standard normal distribution is symmetric and unimodal about the mean, μ
- keeping σ^2 fixed, and changeing μ the normal distribution only gets shifted to a new center
- mantaining μ fixed and increasing σ^2, the distribution becomes more spread out about the same mean value

Theorem

- let $X_1, X_2, \ldots, X_n$, independent random variables with $X_j \sim \mathrm{N}(\mu_j, \sigma_j^2)$
- we build $Y_n = \sum_{j=1}^{n} X_j$
- it can be proved that

$$Y_n \sim \mathrm{Norm}\left(\sum_{j=1}^{n} \mu_j, \sum_{j=1}^{n} \sigma_j^2\right)$$

Still on the sum of independent normal variables

Corollary

- given n random variables, all following the same $\mathrm{N}(\mu, \sigma^2)$ distribution
- then

$$\overline{X} = \frac{\sum X_j}{n} \sim \mathrm{Norm}\left(\mu, \frac{\sigma^2}{n}\right)$$

$\mapsto$ the distribution of X gets more concentrated around the mean value μ as n increases, because the variance, σ^2, decreases with n

Theorem

- any linear combination of independent normal variables is also normal

$$\sum_{j=1}^{n} a_j X_j \sim \mathrm{Norm}\left(\sum_{j=1}^{n} a_j \mu_j, \sum_{j=1}^{n} a_j^2 \sigma_j^2\right)$$

Limit Theorems

- let $X_1, X_2, \ldots, X_n$ independent random variables from the same distribution with mean μ and variance σ^2
- we define $S_n = \sum_{j=1}^{n} X_j$
- since X_j are independent and identically distributed,

$$E[S_n] = nE[X_j] = n\mu \text{ and } \mathrm{Var}(S_n) = n\mathrm{Var}(X_j) = n\sigma^2$$

- the following theorems apply

Strong Law of Large Numbers

$$P\left(\lim_{n\to\infty} \frac{S_n}{n} = \mu\right) = 1$$

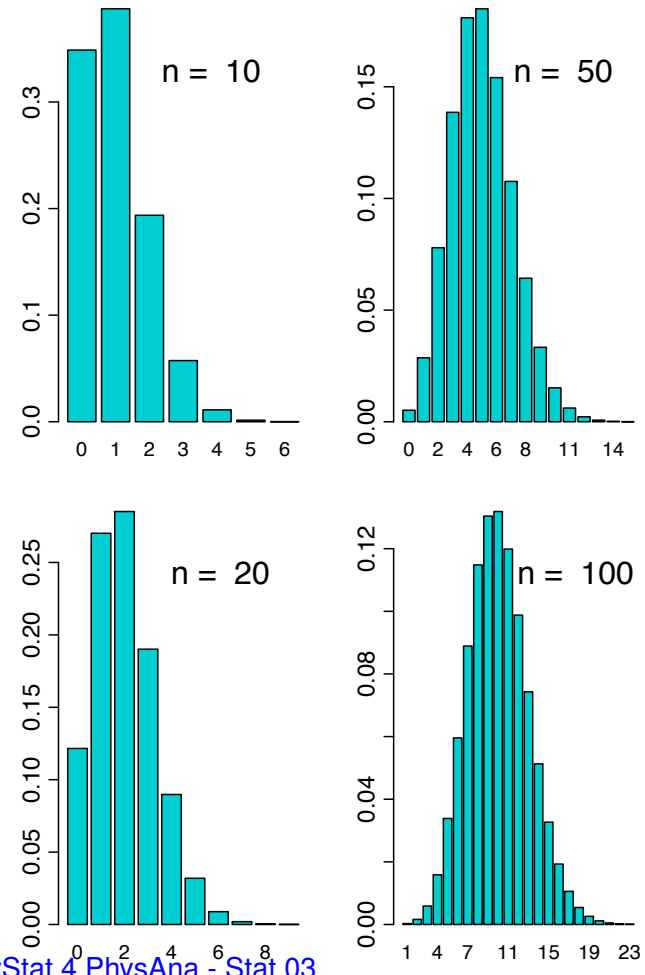
Central Limit Theorem

$$\lim_{n\to\infty} P\left(\frac{S_n - n\mu}{\sigma\sqrt{n}} \leq x\right) = \Phi(x)$$

Where $\Phi(x)$ is the CDF of the standard normal distribution

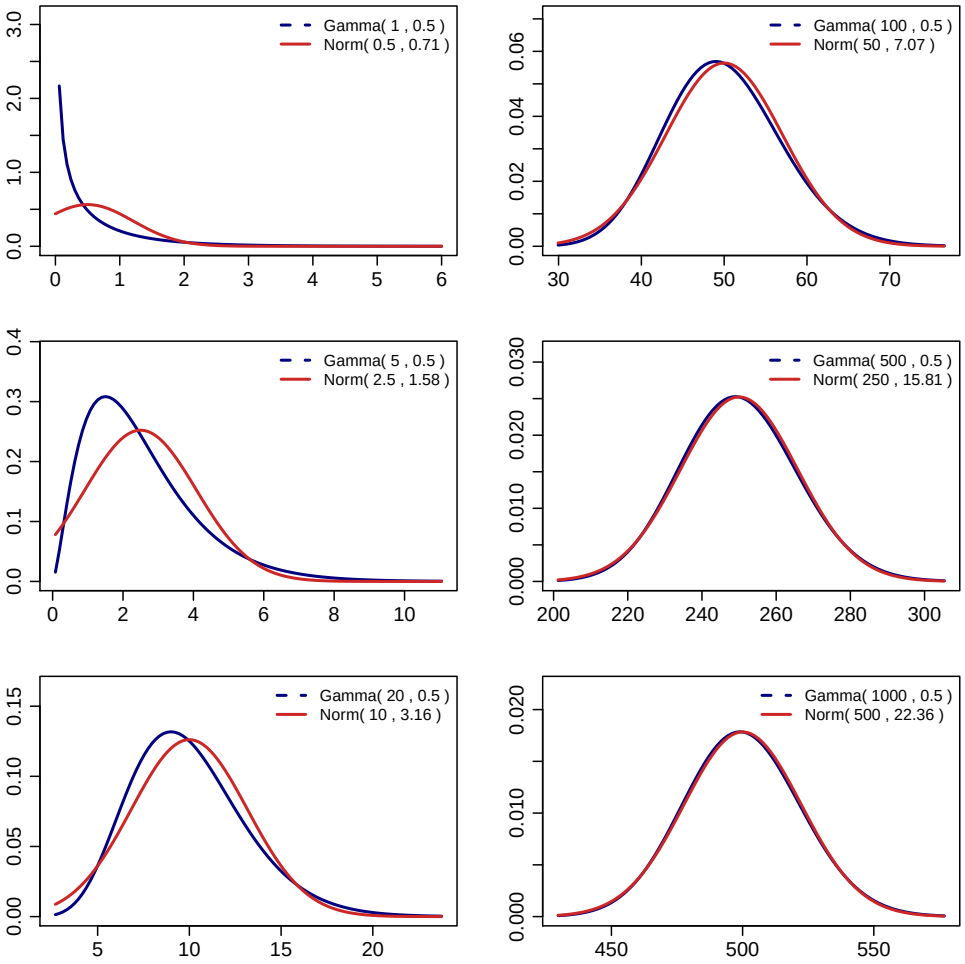
CLT for Binomial distribution

- $X \sim \mathrm{Binom}(n, p)$ with $p = 0.1$
- the histogram is rather skewed for the small n values
- as n increases, it gets less skewed, and for the largest value, $n = 100$, the histogram looks bell-shaped, centered between 10 and 11, resembling a normal density curve
- indeed, the binomial distribution, $\mathrm{Binom}(n, p)$ can be well approximated by $\mathrm{Norm}\,(np, np(1-p))$, for any fixed p, when n is large



CLT for Gamma distribution

- the sum of variables distributed according to $\mathrm{Gamma}(\alpha, \lambda)$ is again a gamma distribution
- the CLT tell us that when the number of terms in the sum is large, the resulting gamma distribution should be approximately normal
- the larger alpha, the less skewed the distribution of the individual terms is
  ➜ the smaller *n* has to be to get a good normal approximation



The 6 boxes toy model

Alberto Garfagnini

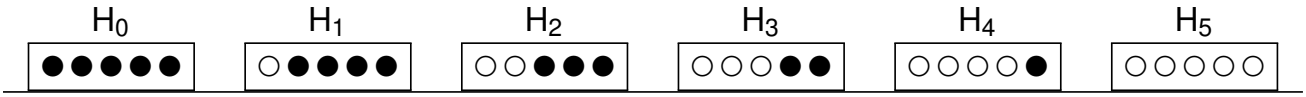
Università di Padova

AA 2021/2022 - Stat Lect. 4

The 6 Boxes Sampling Experiment

The Game

- 6 indistinguishable boxes are prepared with 5 black & white stone
- the composition differs for each box
- boxes are labeled H_j, according to the numbers of white stones in the box, with $j = 0, 1, \ldots, 5$



The Rules of the Game

- we choose one box, randomly
- we try to infer the box content (i.e. the box id) by extracting at random one stone from the box
- the extracted stone is reinserted in the box (sampling with replacement)

The 6 Boxes Sampling Experiment

Our Background Information, I

- the following propositions are defined :

  H_j : box j is selected ($j = 0, 1, \ldots, 5$)

  E_w : a white stone is extracted

  E_b : a black stone is extracted

Our Quests

1) what is the probability of selecting one box ?
2) with the extraction of one stone, what is the probability of observing white, $P(E_w|I)$, or black, $P(E_b|I)$ on the next draw ?
3) how does the probability of the next extraction changes after the stone is extracted, and its color known ?

The space Ω of the events

- the following relations apply:

$$\bigcup_{j=0}^{5} H_j = \Omega, \quad \text{and} \quad \bigcup_{k=b}^{w} E_k = \Omega$$

- in general, we are uncertain about all the combinations of E_k and H_j: the 12 constituents, $E_k \cap H_j$ do not share the same probability
- as an example:

$$P(E_w \cdot H_0 \big| I) = 0, \quad P(E_w \cdot H_5 \big| I) = 1$$

- E_k and H_j form a complete class of hypotheses, each event can be written as a logical sum of the constituents:

$$E_k = \bigcup_j \left(E_k \bigcap H_j\right), \quad \text{and} \quad H_j = \bigcup_k \left(E_k \bigcap H_j\right)$$

- since the events $E_k \cap H_j$ are mutually exclusive, by construction, we have:

$$P(E_k) = \sum_j P(E_k \cdot H_j \big| I) = \sum_j P(E_k \big| H_j I)\ P(H_j \big| I)$$

- and

$$P(H_j) = \sum_k P(H_j \cdot E_k \big| I) = \sum_k P(H_j \big| E_k I)\ P(E_k \big| I)$$

The Process of Knowledge

- E_k is an observable effect: we can experience it with our senses
- H_j is a physical hypothesis: it is not directly observable

  Another rule of the game: we are not allowed to look inside the box !

  ➜ H_j are the possible causes of the effect
- Inference : guessing the causes from the effects

Our experiment consists in

1. extracting stones, randomly and with replacement, from an unknown box
2. evaluating the probability that the box is one of the six boxes

➜ aim of each measurement: update our beliefs about each cause, given all available information

and our calculations

- after the first extraction, $E^{(1)}$, we will compute:

$$P\left(H_j \mid E^{(1)} I\right)$$

- and, after the second extraction $E^{(2)}$:

$$P\left(H_j \mid E^{(1)} E^{(2)} I\right)$$

- and so forth
- what can be easily calculated is the probability of observing the different effects, giving each cause, $P(E_k \mid H_j I)$:

$$P\left(E_w \mid H_j I\right) = \frac{j}{5}, \quad \text{and} \quad P\left(E_b \middle| H_j I\right) = 1 - P\left(E_w \mid H_j I\right) = \frac{5-j}{5}$$

and our calculations ...

- the product rule

$$\begin{aligned} P\left(E_k H_j \middle| I\right) &= P\left(E_k \middle| H_j I\right) P\left(H_j \middle| I\right) \\ &= P\left(H_j \middle| E_k I\right) P\left(E_k \middle| I\right) \end{aligned}$$

- can be rewritten as

$$\frac{P\left(E_k \middle| H_j I\right)}{P\left(E_k \middle| I\right)} = \frac{P\left(H_j \middle| E_k I\right)}{P\left(H_j \middle| I\right)}$$

- we know $P(E_k|H_j I)$ and $P(E_k|I)$ can be evaluated as:

$$P\left(E_k \middle| I\right) = \sum_j P\left(E_k \middle| H_j I\right) P\left(H_j \middle| I\right) = \frac{0+1+2+3+4+5}{5} \cdot \frac{1}{6} = \frac{1}{2}$$

- as we would expect

and our calculations ... ...

- we can rewrite the product rule as

$$\frac{P\left(H_j \middle| E_k I\right)}{P\left(H_j \middle| I\right)} = \frac{P\left(E_k \middle| H_j I\right)}{P\left(E_k \middle| I\right)} = 2 \cdot P\left(E_k \middle| H_j I\right)$$

- in case of a white stone, $P(E_w|I) = 1$,

$$\frac{P\left(H_j \middle| E_w I\right)}{P\left(H_j \middle| I\right)} = 2 \cdot \frac{j}{5}$$

- while, for a black stone, $P(E_b|I) = 1$,

$$\frac{P\left(H_j \middle| E_b I\right)}{P\left(H_j \middle| I\right)} = 2 \cdot \frac{5-j}{5}$$

and our calculations ... ...

- putting all the ingredients together, we get Bayes' theorem

$$P\left(H_j \mid E_k I \right) = \frac{P\left(E_k \mid H_j I \right) P\left(H_j \mid I \right)}{\sum_j P\left(E_k \mid H_j I \right) P\left(H_j | I \right)}$$

- the denominator is just a normalization factor, and we can simply write:

$$P\left(H_j | E_k I \right) \propto P\left(E_k | H_j I \right) P\left(H_j | I \right)$$

- or, in clear text

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

- Bayes' theorem is simply a compact representation of what has been done in the previous steps.
- it is a formal tool for updating beliefs using logic instead of only intuition

Running the experiment

- we randomly select a box, and start to sample stones from the box
- after each extraction, we update the probabilities of each hypothesis, using Bayes' theorem:

$$P\left(H_j | I_n \right) = \frac{P\left(E^{(n)} | H_j \; I_{n-1} \right) P\left(H_j | I_{n-1} \right)}{\sum_l P\left(E^{(n)} | H_l \; I_{n-1} \right) P\left(H_l | I_{n-1} \right)}$$

- where $E^{(n)}$ refers to the n-th extraction,
- $P(E^{(n)}|H_j)$ have been computed before:

$$P(E_w^{(n)}|H_j) = \frac{j}{5}, \quad P(E_b^{(n)}|H_j) = \frac{5-j}{5}$$

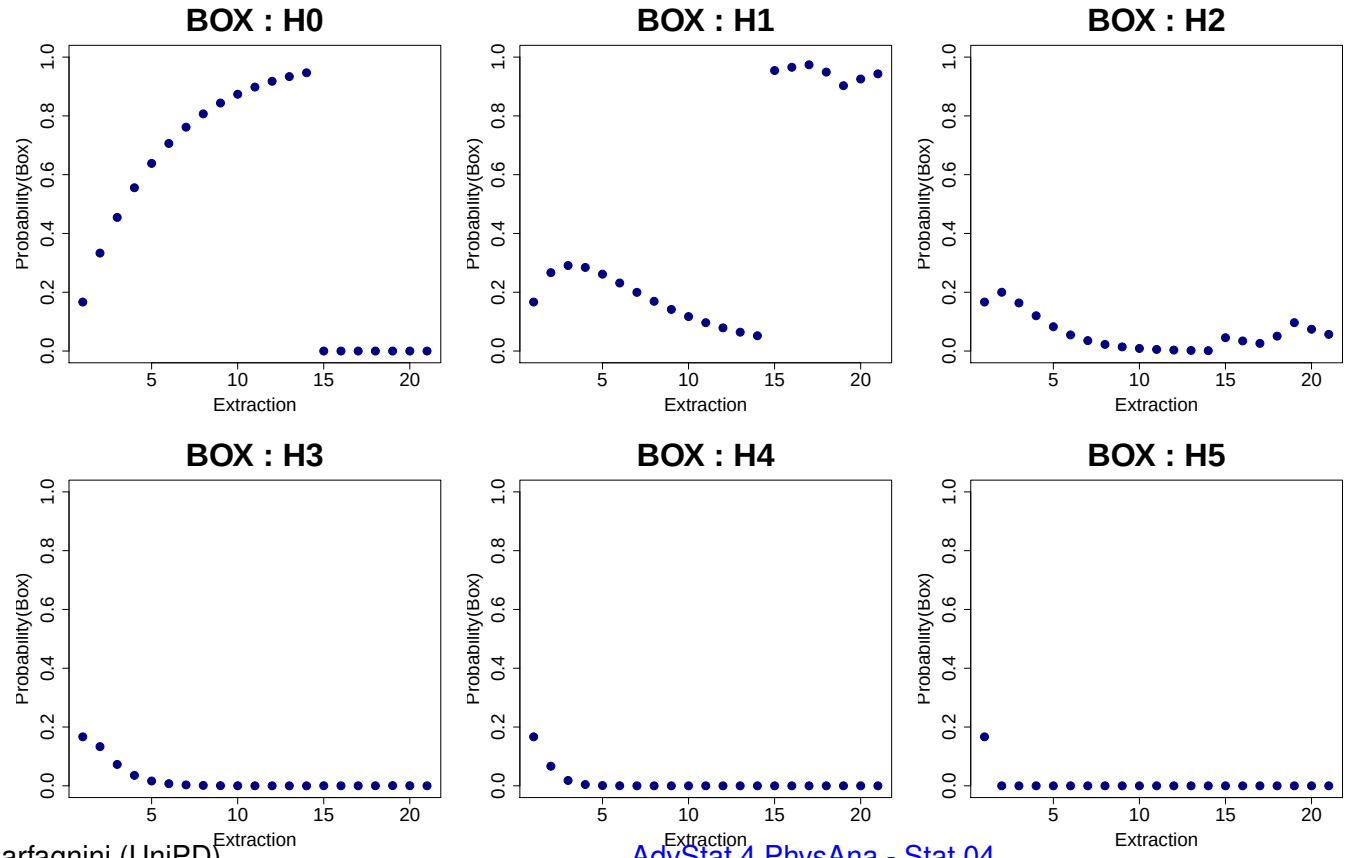
- and $P(H_j|I_{n-1})$ have been given by the calculations at extraction $(n-1)$-th

Running the experiment

Trial	E	H_0	H_1	H_2	H_3	H_4	H_5	$P(E_w \vert I_n)$
0	-	0.167	0.167	0.167	0.167	0.167	0.167	0.5
1	B	0.33	0.27	0.2	0.13	0.06	0	0.27
2	B	0.45	0.29	0.163	0.073	0.0182	0	0.18
3	B	0.55	0.28	0.12	0.036	0.004	0	0.13
4	B	0.64	0.26	0.08	0.016	0.001	0	0.096
5	B	0.71	0.23	0.05	0.007	2.2E-4	0	0.072
6	B	0.76	0.20	0.04	0.003	4.9e-5	0	0.056
7	B	0.81	0.17	0.02	0.001	1.0e-5	0	0.044
8	B	0.84	0.14	0.01	5.5e-4	2.2e-6	0	0.034
9	B	0.87	0.12	0.009	2.3e-4	4.5e-7	0	0.027
10	B	0.90	0.10	0.005	9.4e-5	9.2e-8	0	0.022
11	B	0.92	0.08	0.003	3.8e-5	1.9e-8	0	0.017
12	B	0.93	0.06	0.002	1.6e-5	3.8e-9	0	0.014
13	B	0.95	0.05	0.001	6.3e-6	7.8e-10	0	0.011
14	W	0	0.95	0.045	3.5e-4	5.7e-8	0	0.21
20	B	0	0.93	7.4e-2	3.8e-4	1.4e-8	0	0.21
40	W	0	0.998	1.4e-3	7.1e-9	8.7e-19	0	0.20

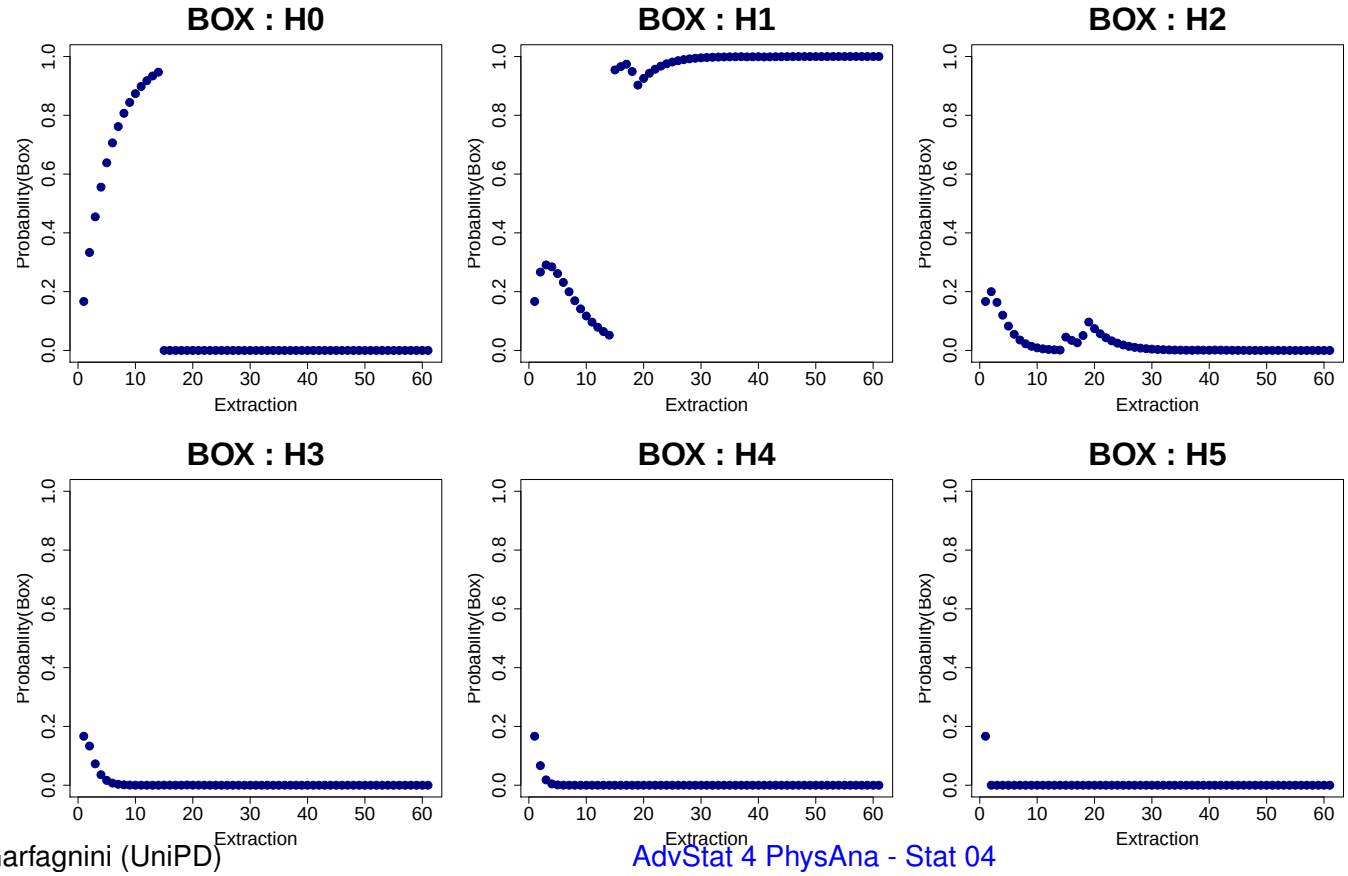
Run results : 20 samplings

- Run performed with `set.seed(89540)`
- important extraction at round 14



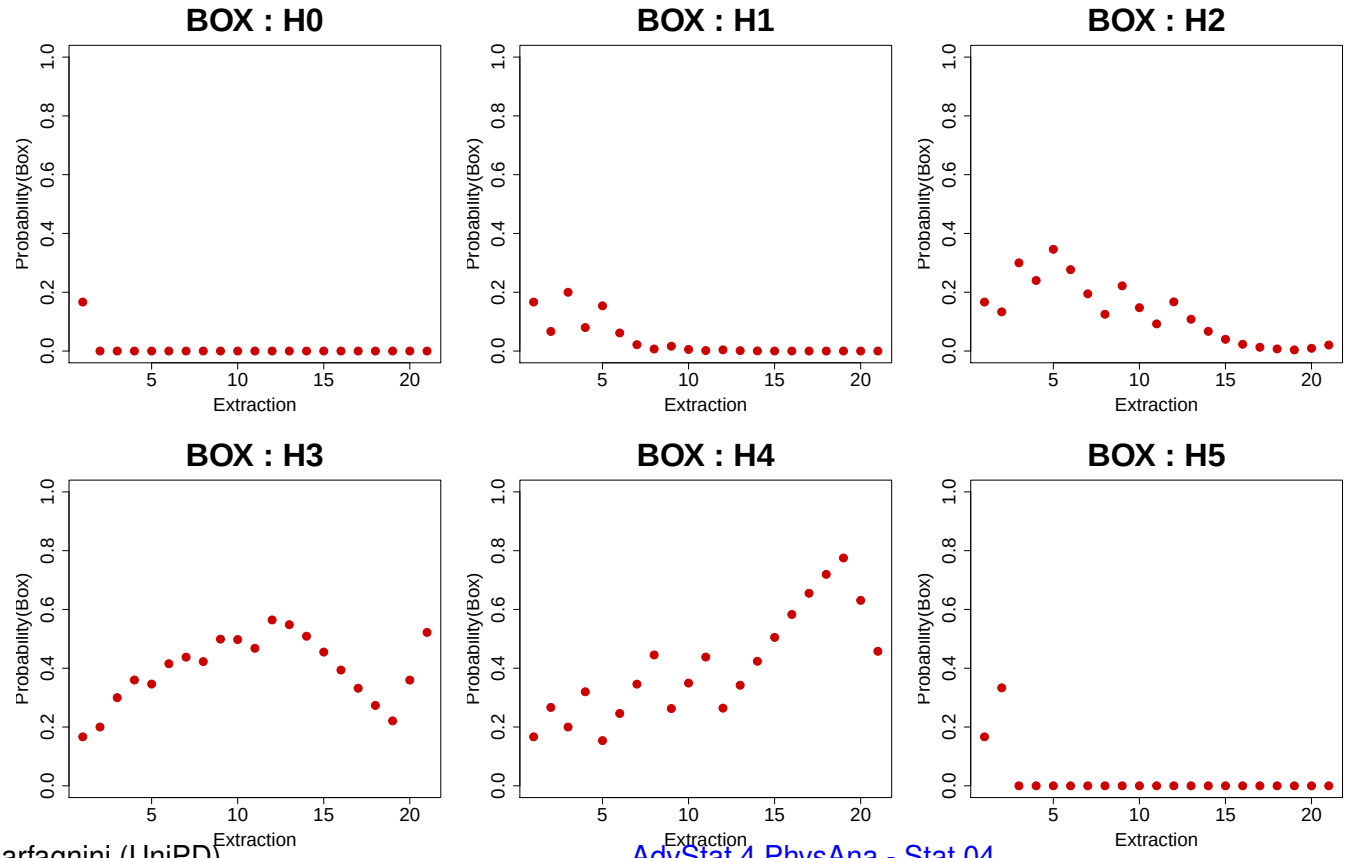
Run results : 60 samplings

- Box H_1 is the most probable : $\boxed{\circ\bullet\bullet\bullet\bullet}$ $P(E_w|I_n) = 0.2$, as expected



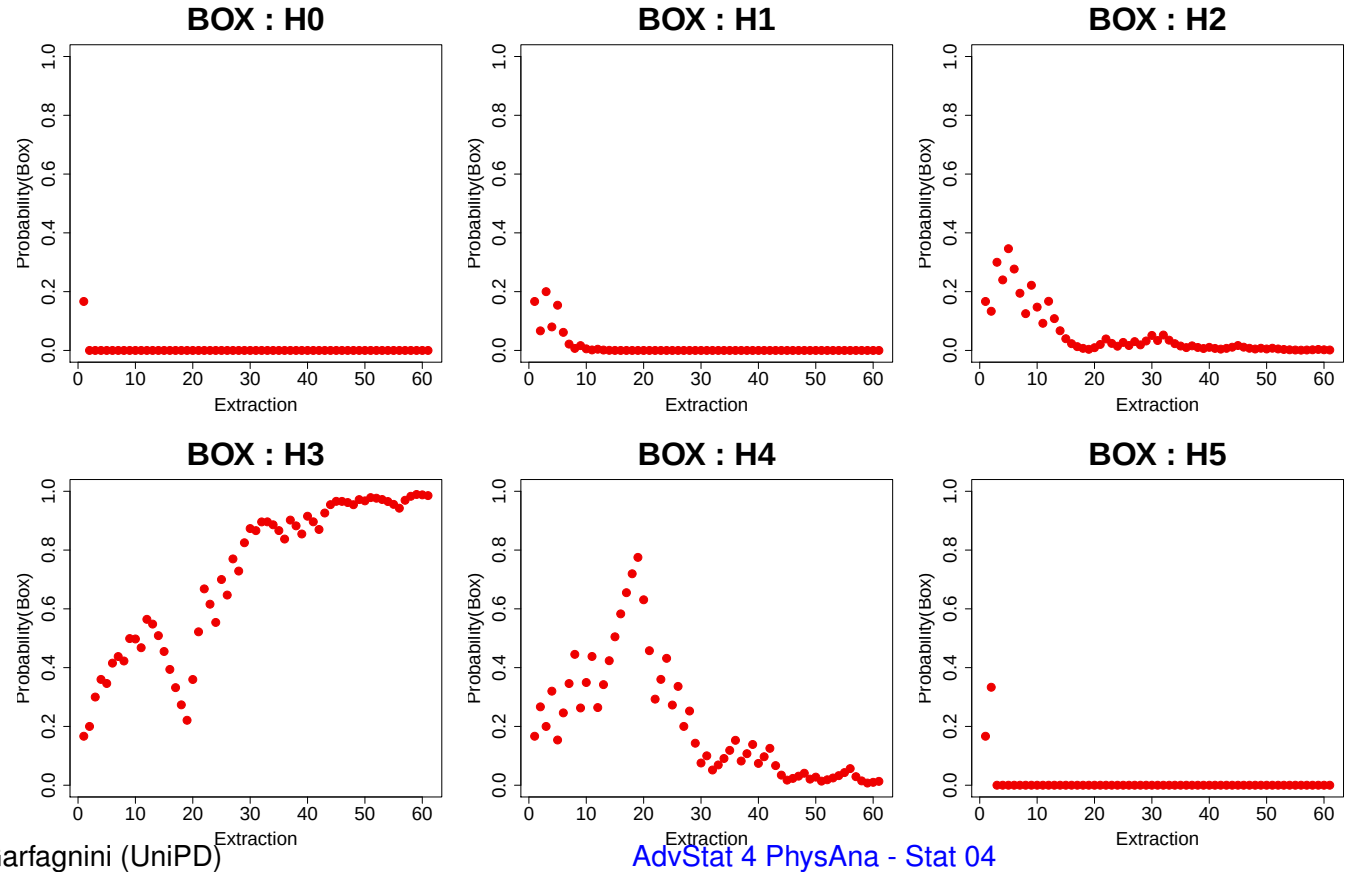
New run results: 20 samplings

- Run performed with `set.seed(89540)`
- most flavored oscillates between H_3 and H_4



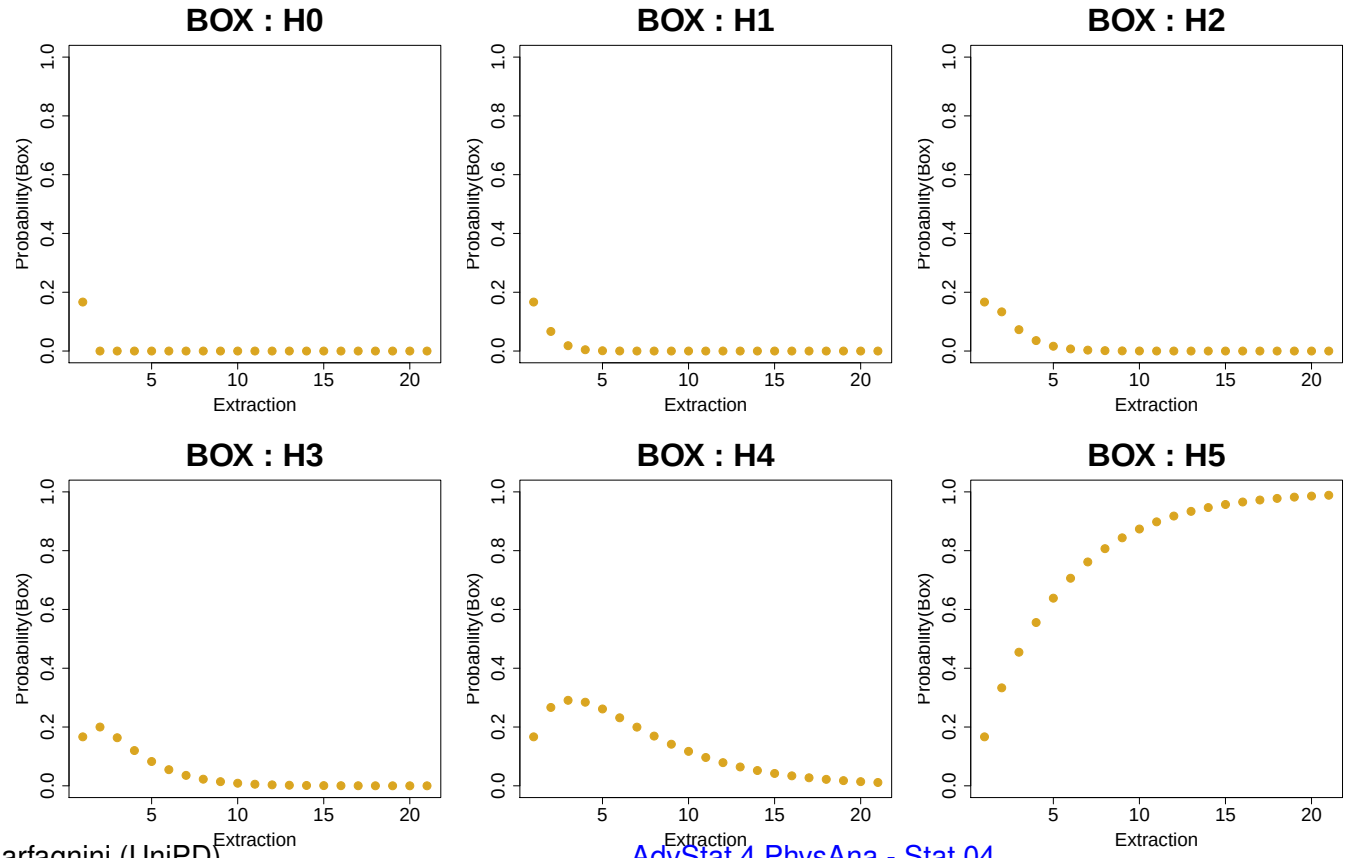
New run results : 60 samplings

- Box H_3 is the most probable : ○○○●● $P(E_w|I_n) = 0.6$, as expected



Run with an extreme box

- Run performed with `set.seed(89540)` and box ○○○○○



References for the 6 Boxes Toy Model

Articles

- G. D'Agostini, *Teaching statistics in the physics curriculum: Unifying and clarifying role of subjective probability*, Am. Jour. Phys. 67, 1260 (1999), arXiv:physics/9908014
- G. D'Agostini, *More lessons from the six box toy experiment*, arXiv:1701.01143
- G. D'Agostini, *Probability, propensity and probabilities of propensities (and of probabilities)*, arXiv:1612.05292

Additional Material

- G. D'Agostini Web Page at University of Rome, La Sapienza, http://www.roma1.infn.it/~dagos/teaching.html

Statistical Models and Inference - Part I

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Stat Lect. 6

Data Modeling

- we perform experiments and make observations to learn about a phenomenon
- to interpret `data`, we have to model them

Inference

- make general statements about a phenomenon through a model, using noisy and incomplete data
- must describe both the Phenomenon (i.e. Model) and the Measurement Process

▷ Key to Data Modeling: use data together with generative model (theory) and measurement model (experimental practice) to derive consistent probabilistic inferences

Data Modeling

- given some `data`, `D`, we want to perform three actions:

▷ parameter estimation:
for a specific Model `M`, with parameters θ, infere the values of model parameters, i.e. $P\left(\theta \middle| D\ M\right)$, the parameter posterior pdf

▷ model comparison:
given a set of models $\{M_j\}$, find out which one is best supported by data. This means finding $P\left(M_j \middle| D\right)$, the model posterior probability

▷ prediction:
given a model `M`, inferred from the data, predict new data at some new location (in the parameter space or time)

Bayesian Model Comparison

- we start by looking at model comparison for the simple case of models with no parameters

▷ using our data `D`, we look for $P\left(M \middle| D\right)$

- since $\mathrm{M} \cdot \overline{\mathrm{M}} = 0$ and $\mathrm{M} + \overline{\mathrm{M}} = \Omega$, we can write

$$\begin{aligned} P(D) &= P(DM) + P(D\overline{M}) \\ &= P(D \mid M)\ P(M) + P(D \mid \overline{M})\ P(\overline{M}) \end{aligned}$$

- our quantity of interest, $P\left(M \middle| D\right)$, is related to Bayes' theorem by

$$P(M \mid D) = \frac{P(D \mid M)\ P(M)}{P(D)} = \frac{P(D \mid M)\ P(M)}{P(D \mid M)\ P(M) + P(D \mid \overline{M})\ P(\overline{M})}$$

$$= \frac{1}{1 + \dfrac{P(D \mid \overline{M})\ P(\overline{M})}{P(D \mid M)\ P(M)}} = \frac{1}{1 + \dfrac{1}{R}}$$

- with $R = \dfrac{P(D \mid M)\ P(M)}{P(D \mid \overline{M})\ P(\overline{M})}$ the posterior odd ratio of the models

Bayesian Model Comparison

- it is easy to demonstrate that

$$\frac{P(M \mid D)}{P(\overline{M} \mid D)} = R = \frac{P(D \mid M)\, P(M)}{P(D \mid \overline{M})\, P(\overline{M})}$$

- in order to determine $P(M \mid D)$, we need three quantities:

▷ $P(D \mid M)$: the probability of measuring `D` when `M` is true

▷ $P(D \mid \overline{M})$: the probability of measuring `D` when `M` is not true (i.e. false)

▷ $P(M)$: the probability that `M` is true, independently of the data (and, of course, $P(\overline{M}) = 1 - P(M)$) $\Rightarrow$ $P(M)$ tells us how probable the model is

- but, shouldn't we have information to tell us that `M` is more likely than $\overline{\texttt{M}}$, we could set

$$P(M) = P(\overline{M})$$

- and R becomes the Bayes factor

$$BF = \frac{P(D \mid M)}{P(D \mid \overline{M})}$$

- i.e. the ratio of the probability of the data under each model

Bayesian Model Comparison

- should we have more models, $\{M_j\}$, with $\sum P(M_j) = 1$, the probability of data becomes

$$P(D) = \sum_j P\left(D \mid M_j\right)\, P\left(M_j\right)$$

- and the posterior probability of model # 1, M_1, becomes

$$P\left(M_1 \mid D\right) = \frac{P\left(D \mid M_1\right)\, P(M_1)}{P(D)}$$

- if we do not have a complete set of models, we cannot compute the posterior probabilities, but we can still compute the odds ratio or Bayes factor between any two models

$$BF = \frac{P(D \mid M_1)}{P(D \mid M_2)} \quad \text{and} \quad R = \frac{P(D \mid M_1)\, P(M_1)}{P(D \mid M_2)\, P(M_2)}$$

Example

Problem

- a test for a disease is 90% reliable
- the probability of testing positive, in absence of the disease, is 0.07
- we know that among people aged 40 to 50 with no symptoms 8 in 1000 have the disease

Q: if a person in his/her 40 tests positive, what is the probability that he/she has the disease ?

Background information

- we build the following propositions:
  - `D`: a person is tested positive
  - `M`: a person has the disease
- and probabilities
  - $P(D \mid M) = 0.9$
  - $P(D \mid \overline{M}) = 0.07$
  - $P(M) = 0.008$

Example - analytical solution

- we build

$$R = \frac{P(D \mid M)\ P(M)}{P(D \mid \overline{M})\ P(\overline{M})} = \frac{9 \cdot 10^{-1} \times 8 \cdot 10^{-3}}{7 \cdot 10^{-2} \times (1 - 8 \cdot 10^{-3})} = 0.1035$$

- therefore

$$P(M \mid D) = \frac{1}{1 + 1/R} = 0.094$$

- even though a positive test result is quite probable (assuming the person has the disease), it is very unlikely that he/she has the disease
- what is decisive in the computation of $P(M \mid D)$ is the ratio between

$$P(D\ M) = P(D \mid M)\ P(M) = 7.2 \cdot 10^{-3}$$

(positive result, assuming the disease is present)

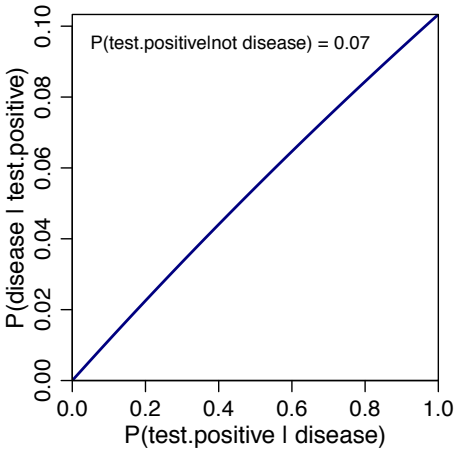
- and

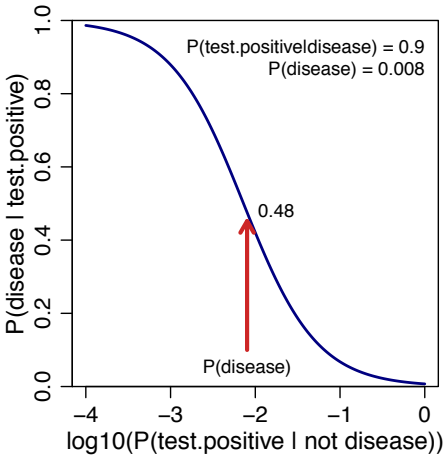
$$P(D\ \overline{M}) = P\left(D \mid \overline{M}\right)\ P\left(\overline{M}\right) = 7 \cdot 10^{-2}$$

(positive result, assuming the disease is absent)

Example - R solution

```
post <- function(p.d.m, p.d.notm, p.m) {
   p.notm <- 1 - p.m
   odds.ratio <- (p.d.m * p.m) /
                 (p.d.notm * p.notm)
   p.m.d <- 1/(1 + 1/odds.ratio)
}

p.d.m <- seq(0, 1, 0.01) # True positive
p.d.notm <- 0.07         # False positive
p.m <- 0.008             # Disease Prior

p.m.d <- post(p.d.m, p.d.notm, p.m)
plot(p.d.m, p.m.d, type='l', lwd=2, col='navy')

p.d.m <- 0.9                    # True positive
p.d.notm <- 10^seq(-4,0, 0.02) # False positive
p.m <- 0.008                    # Disease Prior

p.m.d <- post(p.d.m, p.d.notm, p.m)
plot(log10(p.d.notm), p.m.d, type='l', col='navy')
```





- only once the false positive rate drops below the base rate ($P(M)$) does the test starts to be useful

Data Modeling with Parametric Models

- generative model : theory predicting observable data from model parameters
  - the model just studied did not have any parameter: it was either `true` or `false`
- the simplest generative model is a straight line

$$f(x; a, b) = a + b \cdot x$$

- but our measurements will differ from the model due to noise

$$y = f(x; a, b) + \epsilon$$

- and the noise model - we call it the measurement model - has also parameters
  - given our set of data $\mathrm{D} = \{y_j\}$ at specified values $\{x_j\}$, we want to infer the values of the parameters for the generative model
  - in some cases we want to find the best set of parameters that predicts the data
  - but data are noisy → there is no unique solution
- we look for the probability distributions of the parameters, $P\left(\theta \mid D\ M\right)$, also called parameter posterior pdf. Thanks to Bayes' theorem

$$P\left(\theta \mid D\ M\right) = \frac{P\left(D \mid \theta\ M\right)\ P\left(\theta \mid M\right)}{P\left(D \mid M\right)}$$

The Likelihood

- $P\left(D \mid \theta\ M\right)$ is the Likelihood probability
  - it is a key function since it describes both the phenomenon and the data
  - it tells us the probability of getting the data we measured, given some value of the parameters
- `M` specifies:
  - a generative model ← the equation for the straight line $f(x; a, b)$
  - a measurement model ← how the measurement of y at a given x differs from $f(x; a, b)$ due to noise
- the measurement model describes ϵ in $y = f(x; a, b) + \epsilon$
  - example: Gaussian distribution with variance σ^2. The Likelihood for any measurement is

$$P\left(y \mid \theta\ M\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y - f(x; a, b))^2}{2\sigma^2}\right)$$

- telling us that the measurement has a Gaussian distribution about the true value
- $\theta = \theta(a, b; \sigma)$ is the union of the generative and measurements models

The Prior

- $P\left(\theta \mid M\right)$ is the Prior probability
  - it encapsulates all the information we have, independent of the data
- it is called Prior because is the background information we have before obtaining the Data
- different people may have different information, or different opinion on what prior information is important
- this is not a weakness of inference
- it just reflects reality: we do not only use our immediate measurements to reach scientific conclusions

The Posterior

- $P\left(\theta \mid D\ M\right)$ is the Posterior probability
  - it is the pdf over the model parameters, given `data` and background information
- from Bayes' theorem

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

- the proportionality is through $P\left(D \mid M\right)$, a normalization factor which is independent of θ. Therefore:

$$P\left(\theta \mid D\ M\right) = \frac{1}{Z} P\left(D \mid \theta\ M\right) P\left(\theta \mid M\right)$$

- with $Z = P\left(D \mid M\right)$
- from a conceptual point of view, inference is really that strightforward
- Bayesian inference is the process of improving our knowledge of the model paramaters by using the data

▷ we update the Prior using the Likelihood to obtain the Posterior

The Evidence

- $P\left(D \mid M\right)$ is the Evidence
  - is the denominator of Bayes's equation and it gives the probability of observing the Data `D`, assuming the model `M` to be true, for any values of θ

$$P\left(D \mid M\right) = \int P\left(D \mid \theta M\right) P\left(\theta \mid M\right) d\theta$$

- evidence plays a key role in model comparison
- as a normalization constant, it is very important if we want to compute certain quantities from the posterior
- sometimes the integral can be calculated analytically, but for many real-world problems, we have to resort to numerical integration → Markov Chain Monte Carlo

Bayesian Inference of repeated Bernoulli trials

Bayesian analysis of coin tossing

Problem

- we have a coin and we toss it *n* times
- the coin lands heads in r of them

Q is the coin fair ? (i.e. $\pi = \frac{1}{2}$)

Comment

- no definitive answer exists
- only a probabilistic answer can be provided
- we are looking for

$$P\left(\pi \mid n, r, M\right)$$

- from Bayes' theorem

$$P\left(\pi \mid n, r, M\right) = \frac{P\left(r \mid \pi, n, M\right)\ P\left(\pi \mid M\right)}{P\left(r \mid n, M\right)}$$

Comment: *n* is not part of the Prior since it is independent of the number of coin tosses

Coin tossing model and probabilities

Our Measurement Model

- π : probability of getting heads in one toss
- π is constant in all the tosses
- all tosses are independent

The Likelihood

- the appropriate Likelihood is the binomial distribution

$$P\left(r \mid \pi, n, M\right) = \binom{n}{r} \pi^r (1-\pi)^{n-r} \quad \text{with} \quad r \leq n$$

Comment: n is part of the data, but it is on the right side since it is fixed before starting to collect data

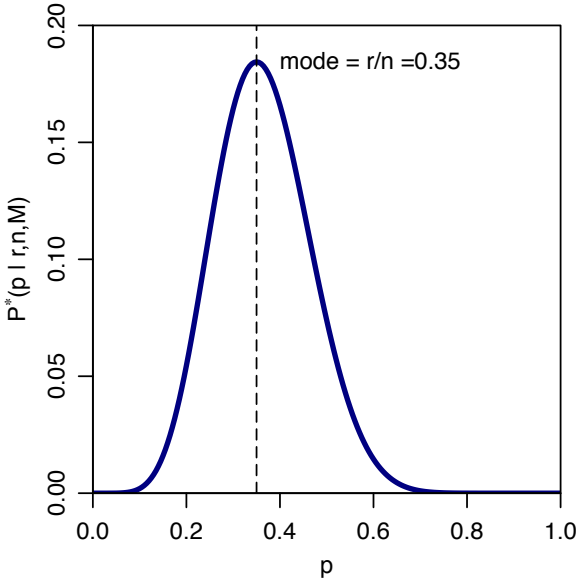
Coin tossing : a uniform Prior

- let's adopt a uniform prior, $P\left(\pi \mid M\right) \sim \mathcal{U}(0,1)$
- the Posterior pdf is simply proportional to the Likelihood

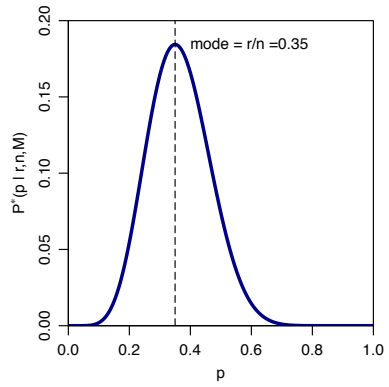
$$P\left(\pi \mid r, n, M\right) = \frac{1}{Z}\, \pi^r (1-\pi)^{n-r} = \frac{1}{Z} P^*\left(\pi \mid r, n, M\right)$$

- the normalization factor Z (i.e. the evidence $P\left(r \mid n, M\right)$ does not depend on π
- the mode is at r/n

```
n <- 20
r <- 7
p <- seq(0, 1, length.out = 201)
p.post <- dbinom(x=r, size=n, prob=p)

plot(p, p.post,
     xaxs='i', yaxs='i', col='navy',
     type='l', lty=1, lwd = 3,
     ylim=c(0,0.2),
     xlab="p",
     ylab=expression(paste(P^symbol("*"),
                     "(p␣|␣r,n,M)")))
```



Uniform Prior



Comments

- the curve is not binomial in π, but it is binomial in r
- the posterior is not-normalized: the integral over π is not unity
- we need the normalization factor only if we want to calculate expected values: i.e. mean and variance

- given the un-normalized posterior pdf, $P^*(\pi \mid r, n, M)$,

$$E[\pi] = \int_0^1 \pi \cdot P\left(\pi \mid r, n, M\right)\, d\pi = \frac{1}{Z} \int_0^1 \pi \cdot \pi^r (1-\pi)^{n-r}\, d\pi$$

- with

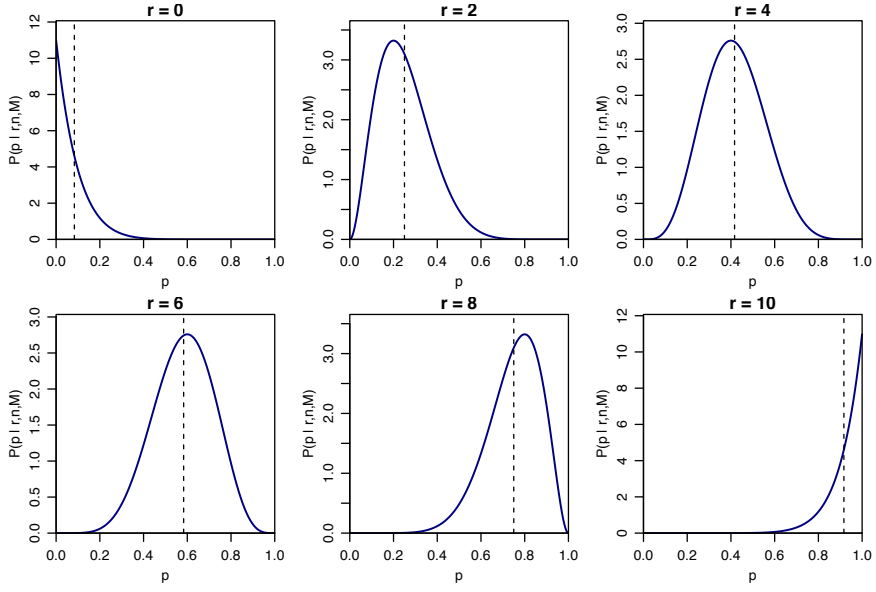
$$Z = \int_0^1 P^*\left(\pi \mid r, n, M\right)\, d\pi \approx \sum_j P^*\left(\pi_j \mid r, n, M\right) \Delta\pi_j$$

- estimated using numerical integration

Uniform Prior

```
n <- 10; n.sample <- 2000; delta.p <- 1/n.sample
p <- seq(from=1/(2*n.sample), by=1/n.sample, length.out=n.sample)

for(r in seq(from=0, to=10, by=2)) {
  p.star <- dbinom(x=r, size=n, prob=p)
  p.norm <- p.star/(delta.p*sum(p.star))
  plot(p, p.norm, type="l", lwd=1.5, col='navy',
       xlim=c(0,1), ylim=c(0,1.1*max(p.norm)),
       xaxs="i", yaxs="i", xlab="p", ylab="P(p␣|␣r,n,M)")
  title(main=paste("r␣=",r), line=0.3, cex.main=1.2)
  p.mean <- delta.p*sum(p*p.norm)
  abline(v=p.mean, lty=2)
}
```

- interval $[0, 1]$ is divided into `n.sample` intervals
- un-normalized pdf is evaluated at the center of each point
- a grid of probability is created
- with the normalized posterior, the expected value is computed



Coin tossing : a Beta Prior

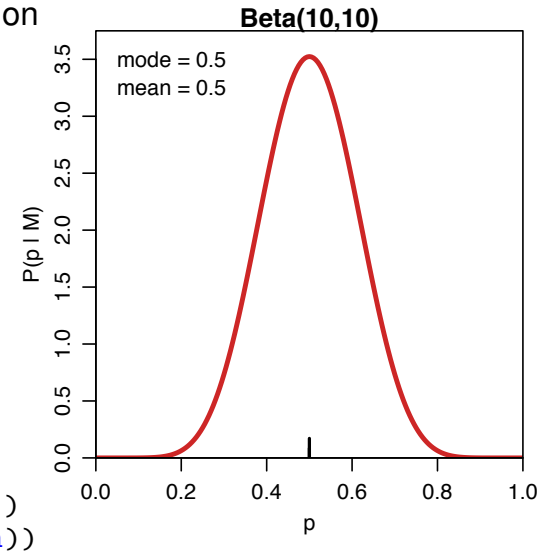
- given a random coin, we may believe the coin is fair, or close to fair
- an appropriate probability density function is the Beta distribution

$$P\left(\pi \mid r, n, M\right) = \frac{1}{\mathrm{B}(\alpha, \beta)}\, \pi^{\alpha-1}(1-\pi)^{\beta-1} \quad \text{with } \alpha > 0\ ,\beta > 0$$

Note: for $\alpha = \beta = 1$ we get a uniform distribution

- if $\alpha = \beta$ the function is symmetric, and the mean and mode are 0.5
- the larger α (when $\alpha \geq 1$), the narrower the distribution

```
alpha <- 10; beta  <- 10
p <- seq(0, 1, length.out = 201)
p.prior <- dbeta(p, alpha, beta)
plot(p, p.prior, xaxs='i', yaxs='i',
     col='navy', type='l', lty=1, lwd = 3,
     ylim=c(0,3.75),
     xlab="p", ylab=paste("P(p␣|␣M)"),
     main=paste("Beta(",alpha,",",beta,")"))
mode <- (alpha - 1)/(alpha + beta - 2)
lines(c(mode, mode), c(0, 0.2), lty=5, lwd=2)
mean <- alpha/(alpha + beta)
lines(c(mean, mean), c(0, 0.2), lty=2, lwd=2)
text(0.05, 3.5, adj=0, paste("mode␣=␣", mode))
text(0.05, 3.25, adj=0, paste("mean␣=␣", mean))
```



Beta Prior

- multiplying the Prior by the likelihood, and absorbing the terms not depending on π in the constant term Z, we get

$$\begin{aligned} P\left(\pi \mid r, n, M\right) &= \frac{1}{Z}\, \pi^{r}(1-\pi)^{n-r} \times \pi^{\alpha-1}(1-\pi)^{\beta-1} \\ &= \frac{1}{Z}\, \pi^{r+\alpha-1}(1-\pi)^{n-r+\beta-1} \end{aligned}$$

- multiplying the Posterior with this Likelihood, we get the same form for the Posterior (another Beta distribution)
- the normalization constant is

$$Z = \mathrm{B}(r+\alpha, n-r+\beta)$$

- we say the Prior and Posterior are conjugate distributions

▷ the Prior is the *conjugate Prior* for this Likelihood function

Beta Prior

- if we start with a Beta Prior with parameters α_p and β_p, and then measure r heads in n tosses, the Posterior is a Beta functions with parameters

$$\alpha = \alpha_p + r \quad \text{and} \quad \beta = \beta_p + n - r$$

- mean and mode for the Posterior are

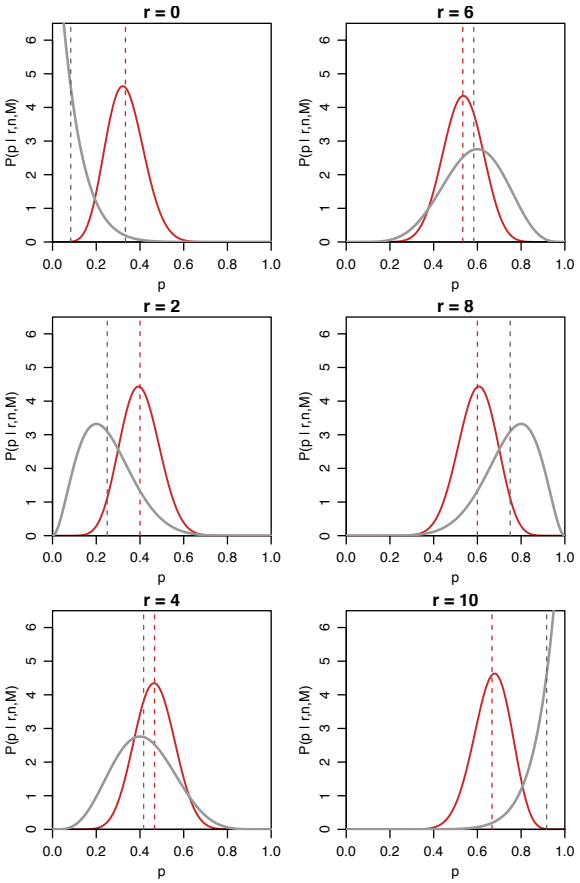
$$\text{mean} = \frac{\alpha_p + r}{\alpha_p + \beta_p + n} \quad \text{and} \quad \text{mode} = \frac{\alpha_p + r - 1}{\alpha_p + \beta_p + n - 2}$$

- if we compare the result with that obtained with a Uniform Prior ($\mathcal{U}(0,1) \sim \text{Beta}(\alpha = 1, \beta = 1)$), we get

$$\text{mean} = \frac{1 + r}{2 + n} \quad \text{and} \quad \text{mode} = \frac{r}{n}$$

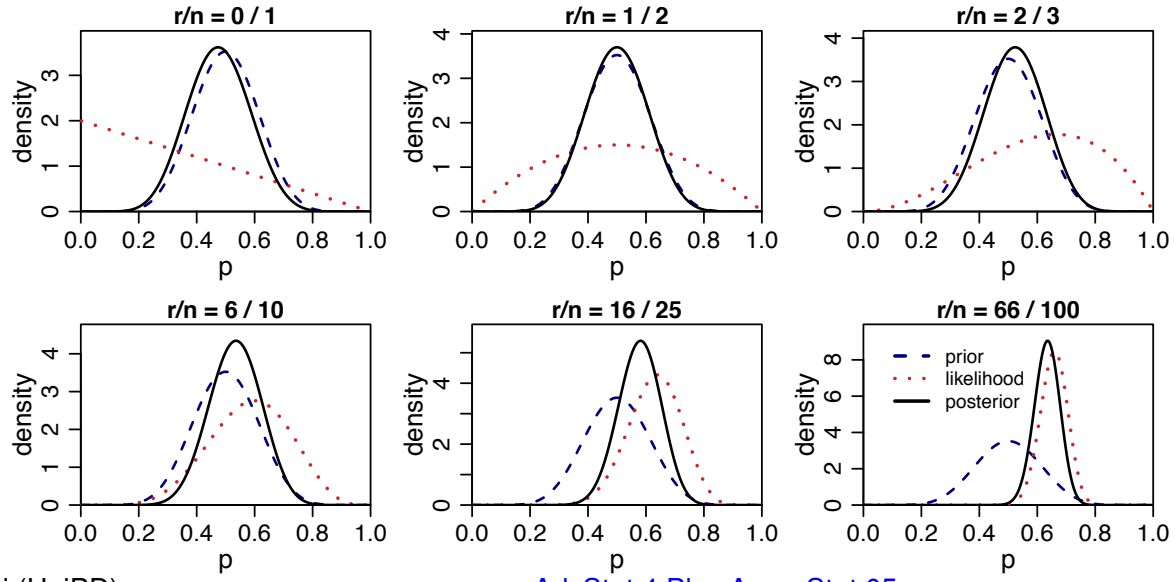
Beta Prior vs Uniform Prior

```
n <- 10;
alpha.prior <- 10;  beta.prior  <- 10
n.sample <- 2000;  delta.p <- 1/n.sample

p <- seq(from=1/(2*n.sample),
         by=1/n.sample, length.out=n.sample)

par(mfrow=c(3,3))

for(r in seq(from=0, to=10, by=2)) {
  post.beta <- dbeta(x=p,
                     alpha.prior+r,
                     beta.prior+n-r)
  plot(p, post.beta, type="l", lwd=1.5,
       col='firebrick3', ...)
  p.mean.b <- delta.p*sum(p*post.beta)
  abline(v=p.mean.b,
         col='firebrick3',lty=2)

  # overplot posterior with Unif Prior
  post.unif <- dbinom(x=r, size=n, prob=p)
  lines(p,
        post.unif/(delta.p*sum(post.unif)))
  p.norm.u <- post.unif/
              (delta.p*sum(post.unif))
  p.mean.u <- delta.p*sum(p*p.norm.u)
  abline(v=p.mean.u, col="grey60", lty=2)
}
```



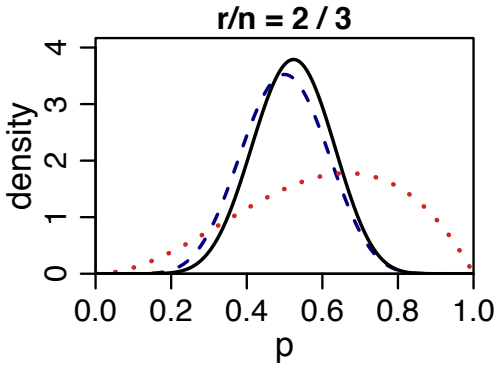
Posterior evolution with data size

- the outcome of only few coin flips tells us little about the fairness of a coin. Our state of knowledge after the analysis of the data is strongly dependent on what we knew or assumed a priori
- as the evidence grows, we are eventually bought to the same conclusions irrespective of our initial beliefs
- the posterior pdf is then dominated by the likelihood function
- the choice of the prior becomes largely irrelevant



Posterior Evolution, R code

```
alpha.prior <- 10;  beta.prior  <- 10
Nsamp <- 200

delta.p <- 1/Nsamp
p <- seq(from=1/(2*Nsamp),
         by=1/Nsamp,
         length.out=Nsamp)
p.prior <- dbeta(x=p,
                 alpha.prior,
                 beta.prior)

n.str <- readline("Enter␣n␣extractions:␣")
n.seq <- as.numeric(unlist(strsplit(n.str, ",")))

# Loop over the vector
for (n in n.seq) {
  r <- as.integer((2/3) * n)

  p.like  <- dbinom(x=r, size=n, prob=p)
  p.like  <- p.like/(delta.p*sum(p.like))
  p.post  <- dbeta(x=p, shape1=alpha.prior+r, shape2=beta.prior+n-r)

  plot(p, p.prior, type="l", xlim=c(0,1), ...)

  lines(p, p.like, col='firebrick3',lwd=2, lty=3)
  lines(p, p.post, lwd=1.5)
  title(main=paste("r/n␣=",r,"/",n), line=0.3, cex.main=1.2)
...
}
```



Parameters best estimates and reliability

- once the posterior is determined, we wish to summarize our inference on a parameter with two numbers:
  - the best estimates
  - and a measure of its reliability
- probability distribution associated with the parameter ⇒ a measure of how much we believe the result lies in the neighborhood of that point
- Best estimate ➜ maximum of the posterior pdf

$$\theta_\circ = \mathrm{MAX}\left\{P\left(\theta \mid D, M\right)\right\}$$

- which means

$$\left.\frac{dP}{d\theta}\right|_{\theta_\circ} = 0 \quad \text{and} \quad \left.\frac{d^2P}{d\theta^2}\right|_{\theta_\circ} < 0$$

- to get a measurement of the reliability of our 'best estimate', we need to look at the spread of the posterior pdf around $\theta_\circ$

Parameters best estimates and reliability

- let's consider a Taylor expansion of the posterior pdf around $\theta_\circ$
- rather than working with the pdf, the calculations will be done with the natural logarithm

$$\begin{aligned} L &= \ln P\left(\theta \mid D, M\right) \\ &= L(\theta_\circ) + \frac{1}{2}\left.\frac{d^2P}{d\theta^2}\right|_{\theta_\circ}(\theta - \theta_\circ)^2 + \ldots \end{aligned}$$

Comments

- $L(\theta_\circ)$ is a constant and tells us nothing about the slope of the posterior pdf
- the linear term in $(\theta - \theta_\circ)$ is missing since we are expanding about a maximum
- the quadratic term is the dominant factor and it determines the width of the pdf

- ignoring higher order contributions and taking the exponential of the Taylor expansion

$$P\left(\theta \mid D, M\right) \sim A \ \exp\left[\frac{1}{2}\left.\frac{d^2P}{d\theta^2}\right|_{\theta_\circ}(\theta - \theta_\circ)^2\right]$$

with A, a normalization constant

Parameters best estimates and reliability

- we have approximated our posterior pdf by a Gaussian distribution

$$P\left(\theta \mid \theta_\circ, \sigma\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\frac{(\theta-\theta_\circ)^2}{\sigma^2}\right]$$

- comparing the two functions, we get

$$\left.\frac{d^2L}{d\theta^2}\right|_{\theta_\circ} = -\frac{1}{\sigma^2} \quad \Rightarrow \quad \sigma = \left(-\left.\frac{d^2L}{d\theta^2}\right|_{\theta_\circ}\right)^{-1/2}$$

- our inference about the quantity of interest is

$$\theta = \theta_\circ \pm \sigma$$

- with:
  - $\theta_\circ$ our best estimate for θ
  - σ a measurement of its reliability
- for a Gaussian distribution

$$P\left(\left|\theta-\theta_\circ\right| \le \sigma \mid DM\right) \sim 0.67$$

$$P\left(\left|\theta-\theta_\circ\right| \le 2\sigma \mid DM\right) \sim 0.95$$

Parameters estimates, coin example, Uniform Prior

- the Posterior is

$$P\left(\pi \mid r, n, M\right) \propto \pi^r(1-\pi)^{n-r}$$

- taking the natural logarithm

$$L = \text{const} + r\ln\pi + (n-r)\ln(1-\pi)$$

$$\frac{dL}{d\pi} = \frac{r}{\pi} - \frac{n-r}{1-\pi} \quad \text{and} \quad \frac{d^2L}{d\pi^2} = -\frac{r}{\pi^2} - \frac{n-r}{(1-\pi)^2}$$

- from the request of a maximum

$$\frac{dL}{d\pi} = 0 \quad \Rightarrow \quad \pi_\circ = \frac{r}{n}$$

- the reliability is given by the second derivative

$$\left.\frac{d^2L}{d\pi^2}\right|_{\pi_\circ} = -\frac{r}{\pi_\circ^2} - \frac{n-r}{(1-\pi_\circ)^2} = -\frac{n}{\pi_\circ(1-\pi_\circ)}$$

- therefore

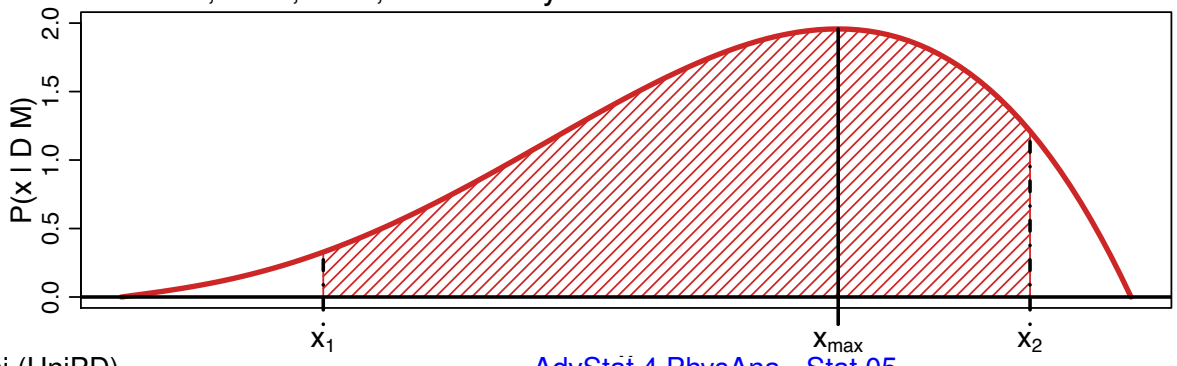
$$\sigma = \left(-\left.\frac{d^2L}{d\theta^2}\right|_{\theta_\circ}\right)^{-1/2} = \sqrt{\frac{\pi_\circ(1-\pi_\circ)}{n}} = \frac{1}{n}\sqrt{\frac{r(n-r)}{n}}$$

Parameters estimates, coin example, Beta Prior

- the Posterior is
$$P\left(\pi \mid r, n, M\right) \propto \pi^{r+\alpha-1}(1-\pi)^{n-r+\beta-1}$$
- taking the natural logarithm
$$L = \text{const} + (r+\alpha-1)\ln\pi + (n-r+\beta-1)\ln(1-\pi)$$
$$\frac{dL}{d\pi} = \frac{r+\alpha-1}{\pi} - \frac{n-r+\beta-1}{1-\pi} \quad \text{and} \quad \frac{d^2L}{d\pi^2} = -\frac{r+\alpha-1}{\pi^2} - \frac{n-r+\beta-1}{(1-\pi)^2}$$
- from the request of a maximum
$$\frac{dL}{d\pi} = 0 \quad \Rightarrow \quad \pi_\circ = \frac{r+\alpha-1}{n+\alpha+\beta-2}$$
- the reliability is given by the second derivative
$$\left.\frac{d^2L}{d\pi^2}\right|_{p\circ} = -\frac{r+\alpha-1}{\pi_\circ^2} - \frac{n-r+\beta-1}{(1-\pi_\circ)^2} = -(\alpha+\beta+n-2)\,\frac{\alpha+r}{\alpha+r-1}$$
- therefore
$$\sigma = \left(-\left.\frac{d^2L}{d\theta^2}\right|_{\theta_\circ}\right)^{-1/2} = \frac{1}{\alpha+\beta+n-2}\sqrt{\frac{\alpha+r-1}{\alpha+r}}$$

Asymmetric Posterior pdfs

- our derivation of the reliability of the parameter estimate (i.e. the error) relies on the validity of the quadratic expansion
- this is usually a reasonable approximation
- however there are times when the posterior pdf is markedly asymmetric
- while the maximum of the posterior can still be regarded as giving the best estimate, the concept of symmetric error bars does not seem appropriate
- a good way to express the reliability is through a confidence interval
$$P\left(x_1 \le x < x_2 \mid D, M\right) = \int_{x_1}^{x_2} P\left(x \mid D, M\right)\, dx \sim 0.95$$
- Why 95% confidence level ?
- it is traditionally seen as a reasonable value, but nothing stops us from quoting other values, 50%, 70%, 99% or any other value



Assigning Priors

- probabilistic inference provides answers to well-posed problems

  but
- it does not define our models
- it does not define the priors
- or tell us which data to collect and how

- with the coin example we learned how the posterior pdf depends on both the prior and the likelihood

  ➜ when data are poor, the prior plays a more dominant role

How do we assign a Prior ?

1) a prior should incorporate any relevant information we have about the problem

   (➜ we implicitly use priors all the time in every day life)
2) some principles can help us to adopt an appropriate prior

Principle of insufficient reason

- also called the principle of indifference
- if we have a set of mutually exclusive outcomes, and we do not expect any one of them more likely, we should assign equal probabilities

Assigning Priors

Maximum Entropy

- it is based on the idea of finding the least informative (most entropic) distribution, given certain information
- example:

  if only mean and variance are known, it shows that the Gaussian is the least informative distribution

Empirical Bayes

- priors are estimated from some general properties of the data
- we can take the posterior from one analysis to be the prior of the next analysis, if they involve independent data
- the final posterior will be identical to having combined the two data sets together with the original prior
- let D_1 and D_2 be two independent data sets

$$P\left(\theta \mid D_1 D_2\right) \propto P\left(D_1 D_2 \mid \theta\right) P\left(\theta\right)$$

$$\propto P\left(D_2 \mid \theta\right) P\left(D_1 \mid \theta\right) \times P\left(\theta\right)$$

likelihood for D_2 — posterior from D_1

Exercise : a survey for the next Uni elections

The Problem

- In proximity of the elections for student's representatives in some University board, Anna, Chris and Maggie decide to perform a survey among their classmates to check how strong is their candidate friend
- the aim is to infere the probability that she gets elected

Step 1: choosing the Priors

- Before starting the interviews, they have differnt opinions about the results of the elections:

- Anna thinks that there will be a 20% chanches that their friend will be elected, and moreover, the probability has a standard deviation of 0.08.
  She therefore assumes a Beta prior such that:

$$E[x] = \frac{a}{a+b} = 0.2 \quad 1 - E[x] = \frac{b}{a+b} = 0.8 \qquad \frac{0.2 \times 0.8}{a+b+1} = 0.08^2 \,,$$

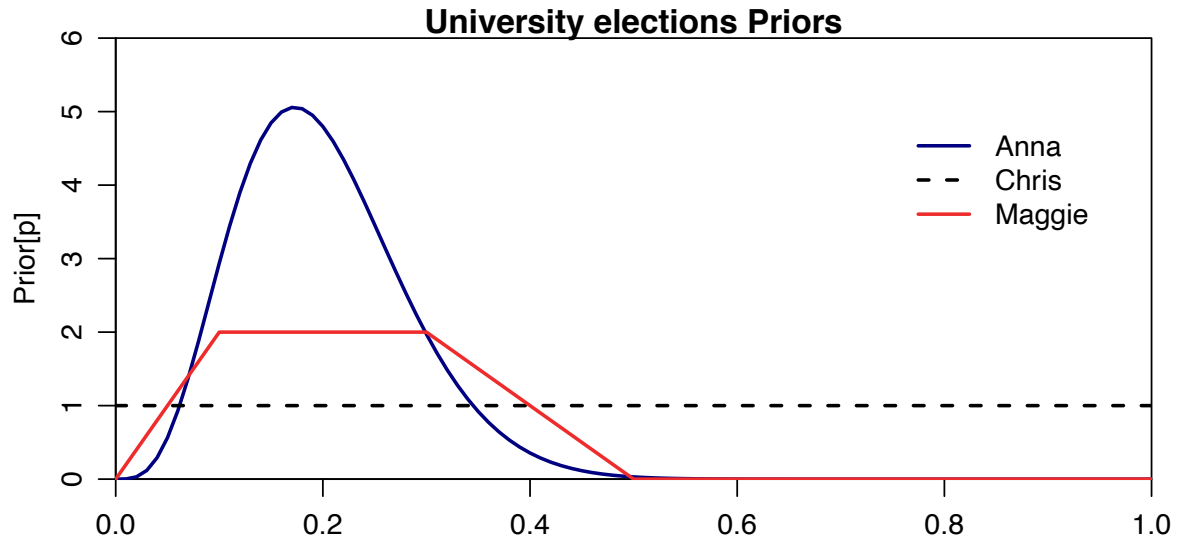
  which means $a = 4.8$ and $b = 19.2$
- Chris is a new student and he does not have any feeling how popular their candidate is, therefore he assumes a Uniform prior distribution. For him $a = b = 1$

Exercise : a survey for the next Uni elections (2)

Step 1: choosing the Priors (cont'd)

Before starting the interviews, they have different opinions about the results of the elections:

- Maggie thinks that the probability distribution is flat, but not over the whole domain. Therefore she assumes a trapezoidal distribution which is flat between 0.1 and 0.3, and goes to zero outside that domain



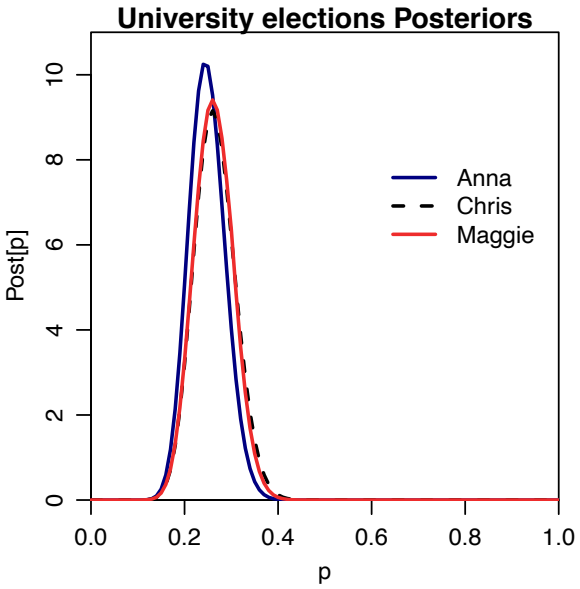
Exercise : a survey for the next Uni elections (3)

Step 2: getting the data

- now they start the survey and decide to interview $n = 100$ students regularly coming to the University canteen but they do not personally know
- out of the interviewed students, $x = 26$ claim they will support and vote the candidate

Step 3: computing the Posterior

- Anna and Chris use a Beta prior ➜ they get a conjugate prior $\mathrm{Beta}(\alpha = a + x, \beta = b + n - x)$
  - Anna has $\mathrm{Beta}(\alpha = 4.8 + 26, \beta = 19.2 + 74)$
  - Chris gets $\mathrm{Beta}(\alpha = 1 + 26, \beta = 1 + 74)$
  - Maggie has to perform a numerical computation of the posterior, given her user-defined Prior

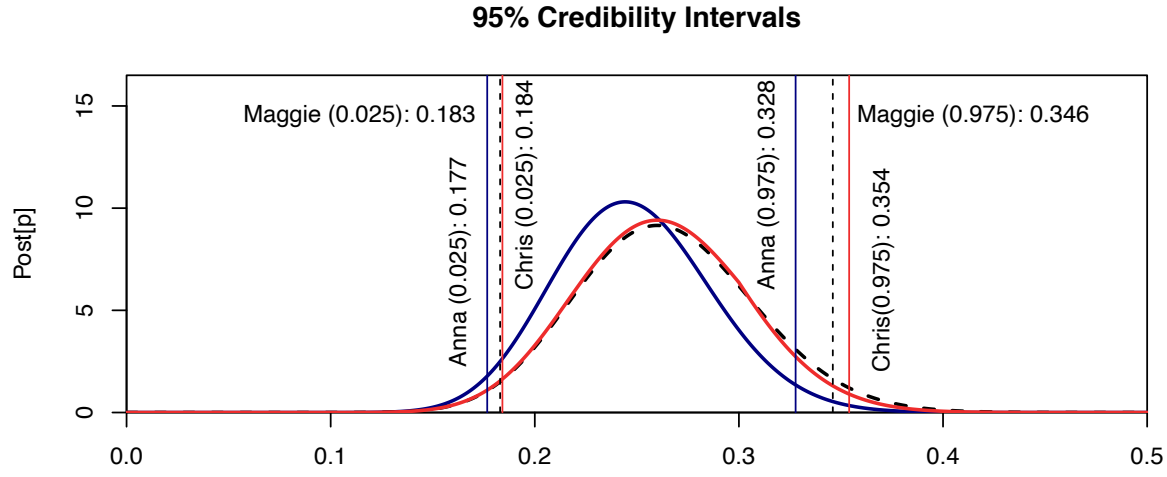


Exercise : a survey for the next Uni elections (4)

Step 4: computing Credibility Intervals

- given the Posterior distributions, we can compute the mean value and the variance
- by integrating the Posterior distribution, it is possible to compute the Credibility Interval, 95%, as the area between the 2.5% and 97.5%
- Maggie's estimate must be done by numerical integration

	$\mathrm{Post}(\alpha, \beta)$	mean	sigma	95% Cr. Int.
Anna	$\mathrm{Beta}(\alpha = 30.8, \beta = 93.2)$	0.248	0.039	0.177 - 0.328
Chris	$\mathrm{Beta}(\alpha = 27, \beta = 75)$	0.265	0.043	0.184 - 0.354
Maggie	numerical	0.262	0.042	0.183 - 0.346



Statistical Models and Inference - Part II

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Stat Lect. 6

Bayesian inference for a Bernoulli process

- we used Bayes' theorem

$$P\left(p \mid \{y_j\}\ M\right) \propto f\left(\{y_j\} \mid p\ M\right) \times g\left(p \mid M\right)$$

- the likelihood of our data follows a binomial distribution

$$f(y \mid p) = \binom{n}{y} p^y (1-p)^{n-y}$$

- multiplying a binomial likelihood times a beta prior ➜ new beta posterior distribution
- the beta distribution is the conjugate family for the binomial probability inference
  - $\texttt{Beta}(1,1)$ gives a uniform distribution
  - $\texttt{Beta}(\frac{1}{2},\frac{1}{2})$ gives a Jeffrey's prior: a distribution that is invariant under any continous transformation of the parameter
- the posterior distribution summarizes our belief about the parameter after having seen the data
- it takes into account our prior belief (the prior distribution) and the data (likelihood).
- we may want to determine an interval that has a high probability of containing the parameter. These are known as Bayesian credible intervals and are somewhat analogous to confidence intervals. But, they have the direct probability interpretation that confidence intervals lack

Bayesian inference for a Poisson process

- useful for counting the occurrences of rare events that happen at a constant rate both in time or in space
- example: number of accidents at a street crossing over a month
- the form of our Bayes' theorem is

$$P\left(\mu \,\middle|\, \{y_j\}\ M\right) \propto f\left(\{y_j\} \,\middle|\, \mu\ M\right) \times g\left(\mu \,\middle|\, M\right)$$

- $\{y_j\}$ indicates our measurement data set
- the parameter μ can assume any positive value ➜ use a continuous prior defined for positive values only
- the scale factor is given by the evidence

  The normalized posterior is

$$P\left(\mu \,\middle|\, \{y_j\}\ M\right) = \frac{f\left(\{y_j\} \,\middle|\, \mu\ M\right) \times g\left(\mu \,\middle|\, M\right)}{\int f\left(\{y_j\} \,\middle|\, \mu\ M\right) \times g\left(\mu \,\middle|\, M\right) d\mu}$$

Likelihood for a Poisson process

- the likelihood for a single measurement of a Poisson process is

$$f(y \,|\, \mu) = \frac{\mu^y e^{-\mu}}{y!}$$

- with $\mu > 0$ and $y = 0, 1, \ldots$
- in case of multiple independent measurements, the likelihood becomes

$$f(\{y_j\} \,|\, \mu) = \prod_{j=1}^{n} f(y_j \,|\, \mu) \quad \propto \quad \mu^{\sum y_j} \times e^{(-n\mu)}$$

- the function looks similar to the `Gamma` distribution function:

$$\texttt{Gamma}(y \,|\, \alpha, \lambda) = k\ y^{\alpha-1}\ e^{-\lambda y}$$

  with

$$k = \frac{\lambda^\alpha}{\Gamma(\alpha)}$$

- comparing with our case: $\alpha = \sum y_j + 1$ and $\lambda = n$

Posterior for a Poisson process (1)

- according to the background knowledge of the researcher, we can have different prior distribution functions:
  - a positive uniform prior
  - a Jeffrey's prior, which is invariant under any continuous transformation of the parameter
  - a Gamma prior, which is the conjugate family for the Poisson inference

Uniform prior

- it is used when there is no idea on what the μ parameter value could be

$$g(\mu) = 1 \quad \text{for} \quad \mu > 0$$

- it is also called an improper prior, since the integral on all the possible value of the parameter diverges
- the posterior becomes:

$$\begin{aligned} P(\mu \,|\, \{y_j\}) &\propto f(\{y_j\} \,|\, \mu) \times g(\mu) \\ &\propto \mu^{\sum y_j} \; e^{-n\mu} \end{aligned}$$

- it's a $\texttt{Gamma}(\alpha, \lambda)$ function with $\alpha = \sum y_j + 1$ and $\lambda = n$

Posterior for a Poisson process (2)

Jeffrey's prior

- it is a pior whichis invariant under any continuous transformation of the parameter

$$g(\mu) \propto \frac{1}{\sqrt{\mu}} \quad \text{for} \quad \mu > 0$$

- it is an improper prior ➜ its integral over the whole parameter range in infinite
- combining with the likelihood, we get

$$\begin{aligned} P(\mu \,|\, \{y_j\}) &\propto f(\{y_j\} \,|\, \mu) \times g(\mu) \\ &\propto \mu^{\sum y_j} \; e^{-n\mu} \times \frac{1}{\sqrt{\mu}} \\ &\propto \mu^{\sum y_j - 1/2} \; e^{-n\mu} \end{aligned}$$

- it's again a $\texttt{Gamma}(\alpha, \lambda)$ function with $\alpha = \sum y_j + \frac{1}{2}$ and $\lambda = n$

Posterior for a Poisson process (3)

Conjugate family prior

- the conjugate family of functions for the Poisson process with parameter μ will have the same form of the likelihood

$$\begin{aligned} g(\mu) &\propto e^{-k\,\mu}\ e^{\log\mu\times r} \\ &\propto e^{-k\,\mu}\ \mu^{r} \end{aligned}$$

- a distribution having this shape is the $\texttt{Gamma}(\alpha,\lambda)$ function with $\alpha - 1 = r$ and $\lambda = k$
- the normalization scale factor is $\dfrac{\lambda^{\alpha}}{\Gamma(\alpha)}$

Posterior for a single observation

- using a $\texttt{Gamma}(\alpha,\lambda)$ prior

$$\begin{aligned} P(\mu \mid y) &\propto f(y \mid \mu)\times g(\mu) \\ &\propto \frac{\mu^{y}e^{-\mu}}{y!}\times\frac{\lambda^{\alpha}\ \mu^{\alpha-1}\ e^{-\lambda\mu}}{\Gamma(\alpha)} \\ &\propto \mu^{\alpha-1+y}\ e^{-(\lambda+1)\mu} \end{aligned}$$

- it's a $\texttt{Gamma}(\alpha',\lambda')$ with $\alpha' = \alpha + y$ and $\lambda' = \lambda + 1$

Posterior for a Poisson process (4)

Posterior for multiple observations

- we have a set of n observations $\{y_j\}$
  - assume a $\texttt{Gamma}(\alpha,\lambda)$ prior
    - ➜ a uniform prior, $g(\mu) = 1$, has the form of $\texttt{Gamma}(1,0)$
    - ➜ the Jeffrey's prior for Poisson, $g(\mu^{-1/2})$ is equivalent to $\texttt{Gamma}(\frac{1}{2},0)$ they are both considered as a limiting case of $\texttt{Gamma}(\alpha,\lambda)$, with $\lambda \to 0$
  - start with the first observation and evaluate the posterior distribution
  - repeat the updating after each observation, using the posterior from the j-th observation as the prior for the observation j+1
- we end up with a $\texttt{Gamma}(\alpha',\lambda')$ posterior where

$$\alpha' = \alpha + \sum y \quad \text{and} \quad \lambda' = \lambda + n$$

- the expected value and variance of the posterior are

$$E[\mu \mid y] = \frac{\alpha'}{\lambda'} \quad \text{and} \quad \mathrm{Var}[\mu \mid y] = \frac{\alpha'}{\lambda'^{2}}$$

How to choose the conjugate prior

- the $\texttt{Gamma}(\alpha, \lambda)$ family of distributions is the conjugate family for the inference of μ parameter from a Poisson distribution
- the advantage of using a conjugate prior is that the posterior will be from the same family and can be found with simple updating rules
- to determine the parameters of the $\texttt{Gamma}(\alpha, \lambda)$ prior that matches our belief, try to summarize your belief into a prior mean m and a prior standard deviation s
- since

$$m = \frac{\alpha}{\lambda} \quad \text{and} \quad s^2 = \frac{\alpha}{\lambda^2}$$

- we get, by inversion:

$$\lambda = \frac{m}{s^2} \quad \text{and} \quad \alpha = \left(\frac{m}{s}\right)^2$$

Exercise

- the weekly number of traffic accidents on a highway between two towns follows a Poisson distribution
- four students are going to count the number of traffic accidents for the next eight weeks

1 Student 1 has no prior information. Therefore she will assume all possible values for μ are equally likely

➜ positive uniform prior, $g(\mu) = 1$

2 Student 2 has no prior information, either. But she wants her prior to be invariant if the parameter is multiplied by a constant

➜ Jeffrey's prior, $g(\mu) = \mu^{-1/2}$

3 Student 3 believes the prior mean should be 2.5 with a standard deviation of 1

➜ Gamma prior, $\texttt{Gamma}(\alpha = 6.25, \lambda = 2.5)$

4 Student 4 thinks his prior has a trapezoidal shape. He draws the prior function by interpolating the values with the following weights

Accidents	0	2	4	8	10
Weight	0	2	2	0	0

Exercise : solution

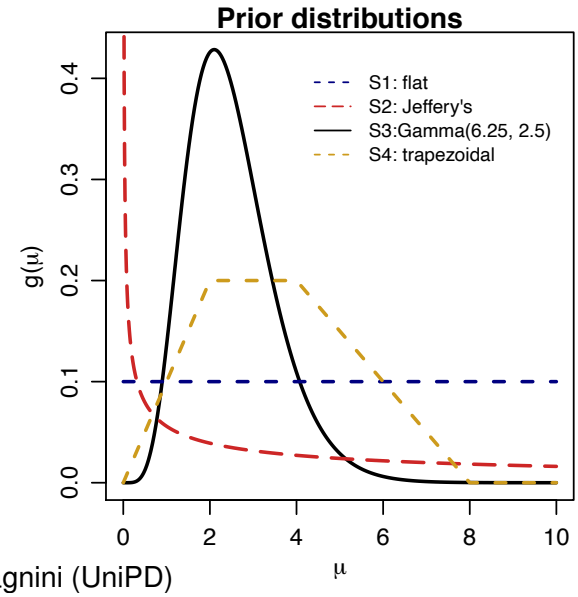
- the 8 weeks measurements bring the following number of accidents per week:
  `3, 2, 0, 8, 2, 4, 6, 1`
- let's evaluate the posteriors:

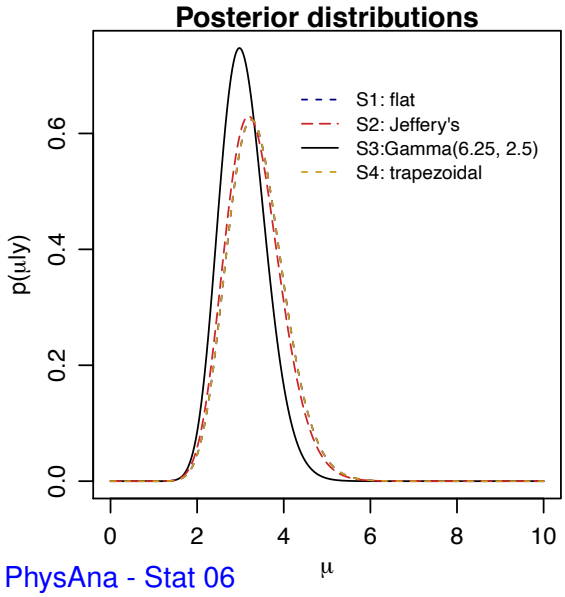
⊏⟶ Student 1 : $\texttt{Gamma}(\alpha', \lambda')$ where $\alpha' = \sum y_j + 1 = 27$ and $\lambda' = n = 8$

◇⟶ Student 2 : $\texttt{Gamma}(\alpha, \lambda')$ where $\alpha' = \sum y_j + \frac{1}{2} = 26.5$ and $\lambda' = n = 8$

○⟶ Student 3 : $\texttt{Gamma}(\alpha', \lambda')$ where $\alpha' = \sum y_j + \alpha = 32.25$ and $\lambda' = \lambda + n = 10.5$

⊡⟹ Student 4 : numerical integration





Example : quantitative results

- the posterior distribution is the complete inference in Bayesian modeling and it allows to extract all the possible parameter values
- three possible measure of the location of a distribution are:
  - the mean
  - the mode (i.e. the most probable value)
  - the median (i.e. the value for which $P(x \leq \text{med}) = P(x > \text{med}) = 0.5$)
- we summarize by extracting the numerical estimates for the four students

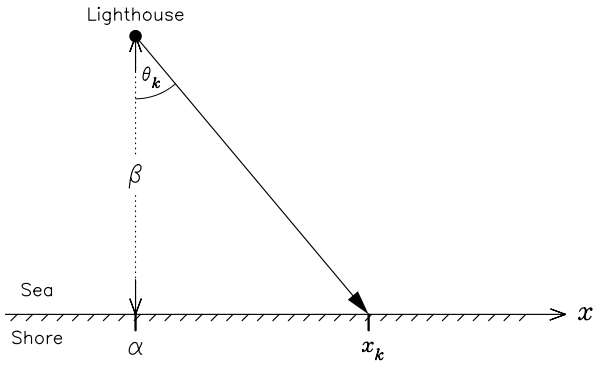
Student	S1	S2	S3	S4
Post	$\texttt{Gamma}(27, 8)$	$\texttt{Gamma}(26.5, 8)$	$\texttt{Gamma}(32.25, 10.5)$	numerical
Mean	3.37	3.31	3.07	3.35
Median	3.33	3.27	3.04	3.32
Mode	3.25	3.19	2.98	-
Std. Dev.	0.65	0.64	0.54	0.63

Credibility Interval 95%

low	2.22	2.17	2.10	2.22
high	4.76	4.69	4.22	4.67

The Lighthouse problem

- A lighthouse is located at a position α along the shore and at a distance β out at sea
- It emits a series of short highly collimated flashes at random intervals and at random angles
- we detect the pulses on the coast using photo-detectors; they record only the position x_k of the flash arrival on the coast, but not the angle of emission



- N flashes have been recorded at positions $\{x_k\}$ ➜ We want to estimate the position of the lighthouse
- it looks reasonable to assign a uniform Likelihood pdf on the azimuth angle θ_k

$$P\left(\theta_k \mid \alpha, \beta\right) = \frac{1}{\pi}$$

- where θ_k is connected to α and β by the relation

$$x_k - \alpha = \beta \tan \theta_k$$

- we operate a change of variable on the pdf

$$P(X|M) = P(Y|M)\left|\frac{dY}{dX}\right|$$

The Lighthouse problem

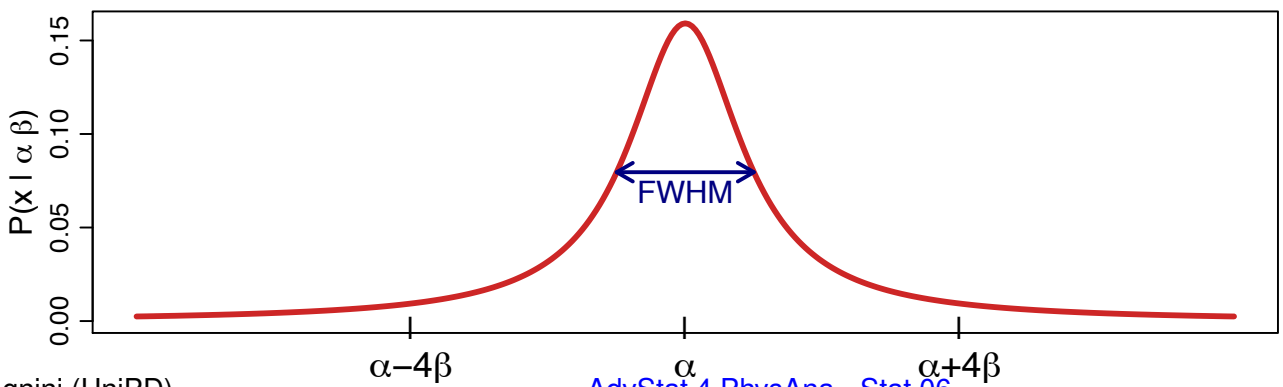
- applying the transformation

$$\begin{aligned} x &= \beta \tan\theta + \alpha \\ dx &= \beta \frac{1}{\cos^2\theta} d\theta = \beta(1+\tan^2\theta)d\theta \\ \left|\frac{dx}{d\theta}\right| &= \beta(1+\tan^2\theta) = \beta\left[1+\frac{(x-\alpha)^2}{\beta^2}\right] = \frac{\beta^2+(x-\alpha)^2}{\beta} \end{aligned}$$

- we get

$$P(x \mid \alpha, \beta) = P(\theta|\, \alpha, \beta)\left|\frac{d\theta}{dx}\right| = \frac{1}{\pi}\frac{\beta}{\beta^2+(x-\alpha)^2}$$

- we have obtained a Cauchy distribution, which is symmetric about the maximum α and with a FWHM of 2β



The Lighthouse problem

- we assume the distance (i.e. β) is known and the only missing parameter is α
- from Bayes' theorem

$$P\left(\alpha \mid D,\beta\right) \propto P\left(D \mid \alpha,\beta\right) \times P\left(\alpha \mid \beta\right)$$

- since β tells us nothing about α, we assume a uniform prior

$$P(\alpha \mid \beta) = P(\alpha) = \begin{cases} \dfrac{1}{\alpha_{max} - \alpha_{min}} & \text{for } x \in [\alpha_{min}, \alpha_{max}] \\ 0 & \text{otherwise} \end{cases}$$

- the recording of a signal at one photo-detector does not influence what we can infer about the position of another measurement (given the same location of the lighthouse)

➜ the Likelihood function is just the product of the probabilities for N individual detections

$$P\left(D \mid \alpha,\beta\right) = \prod P\left(x_j \mid \alpha,\beta\right)$$

- taking the natural logarithm of the Posterior probability function

$$L = \ln P\left(\alpha \mid D,\beta\right) = \text{const} - \sum \ln\left[\beta^2 + (x_j - \alpha)^2\right]$$

- where `const` includes all the terms not depending on α

The Lighthouse problem

- the best estimate $\alpha_\circ$ is given by the maximum of the posterior pdf

$$\left.\frac{dL}{d\alpha}\right|_{\alpha_\circ} = 2\sum_j \frac{x_j - \alpha_\circ}{\beta^2 + (x_j - \alpha_\circ)^2} = 0$$

- an analytical solution is difficult to be done, but nothing stops us from evaluating it numerically

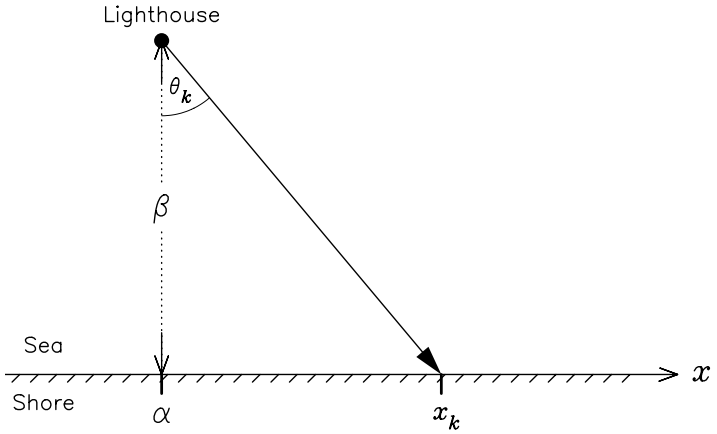
Homework

- write a small R program to evaluate the posterior distribution as a function of the collected data
- assume $\beta = 1$ km, $\alpha_{TRUE} = 1$ km and sample data in the range $x \in [-2\text{ km}, +2\text{ km}]$
- plot the posterior as a function of the number of collected data (assume $n = \{1, 2, 5, 10, 20, 50, 100\}$)

The Lighthouse problem

Homework

- write a small R program to evaluate the posterior distribution as a function of the collected data
- assume $\beta = 1$ km, $\alpha_{TRUE} = 1$ km and sample data in the range $x \in [-2 \text{ km}, +2 \text{ km}]$
- plot the posterior as a function of the number of collected data (assume $n = \{1, 2, 5, 10, 20, 50, 100\}$)



- our Swiss Army knife is always Bayes' theorem

$$P\left(\alpha \,\middle|\, D, \beta\right) \propto P\left(D \,\middle|\, \alpha, \beta\right) \times P\left(\alpha \,\middle|\, \beta\right)$$

A solution to the Lighthouse problem

- instead of calculating the Posterior, it is better to evaluate the logarithm of the posterior

$$L = \log P\left(\alpha \,\middle|\, \{x_k\}, \beta\right) = \text{const} - \log\left(1 + \left(\frac{x - \alpha}{\beta}\right)^2\right)$$

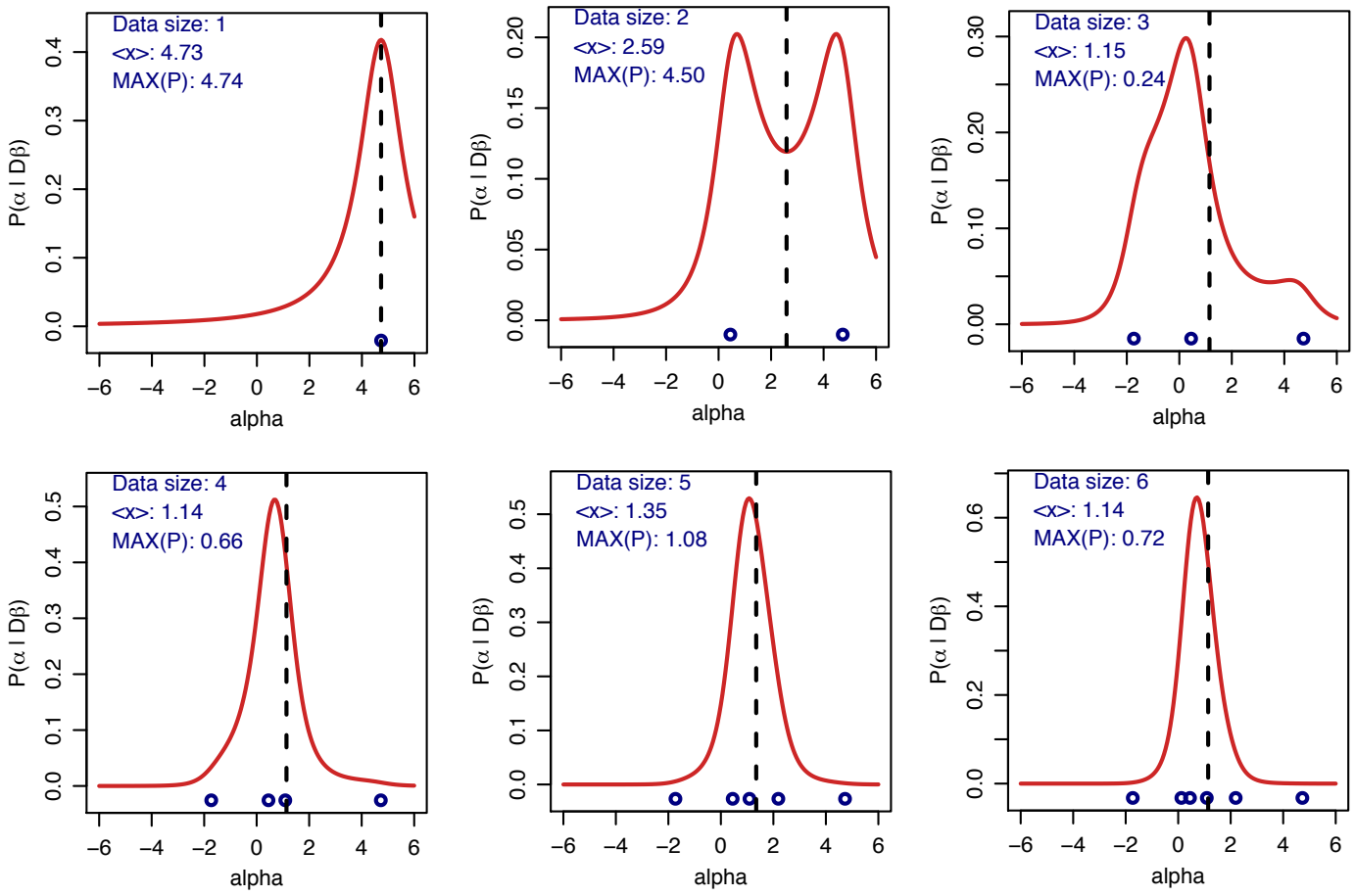
- and afterwards take the exponential

```
p.log.like <- function(a, data) {
  b <- 1
  logL <- 0.0
  for (x in data) {
    logL <- logL - log(1 + ((x-a)/b)^2)
  }
  return(logL)
}

n.sample <- 200
x.min <- -6; x.max <- +6
h <- (x.max - x.min)/n.sample
alpha <- seq(from=x.min, by=h, length.out=n.sample+1)
```

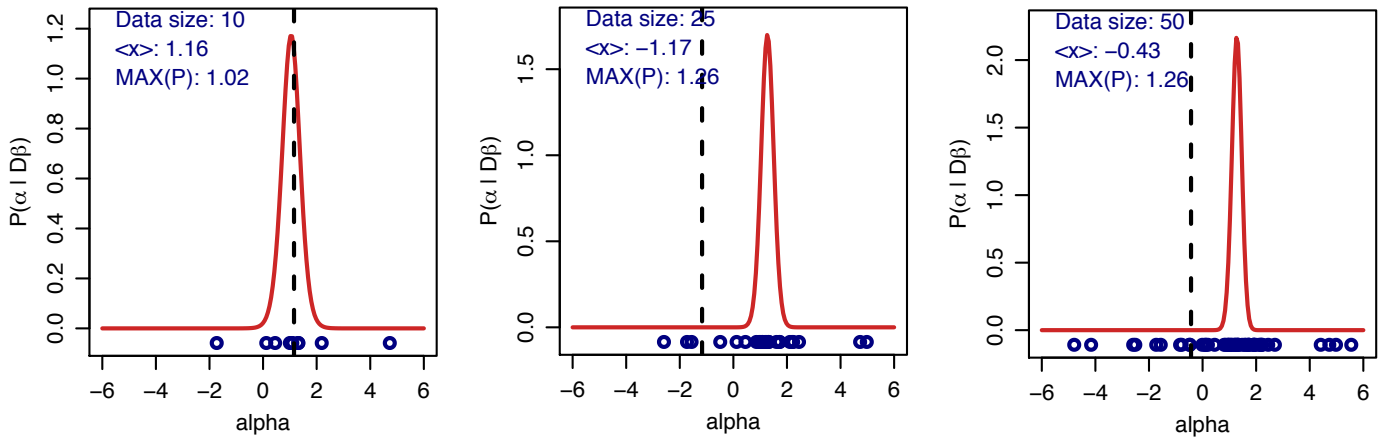
A solution to the Lighthouse problem

```
n.str <- readline("Enter␣data␣set␣dimension:␣")
n.plot <- as.numeric(unlist(strsplit(n.str, ",")))
dt <- data[1:n.plot]

# Get the LogLikelihood
y.log.star <- p.log.like(alpha, dt)
# - Find the maximum
index.max <- which.max(y.log.star)
alpha.max <- alpha[index.max]
cat(paste("Alpha␣max:␣", alpha.max, '\n'))

# - get the exponential and normalize the posterior
y.post.star <- exp(y.log.star)
y.post <- y.post.star/(h*sum(y.post.star))

plot(alpha, y.post, type='l', lwd=2, col='firebrick3')

dt.mean <- mean(dt)
abline(v=dt.mean, lty=2, lwd=2)

y.band <- (max(y.post) - min(y.post))*0.05
text(-6, max(y.post)+y.band, col='navy', lwd = 2, pos=4,
     paste("Data␣size:␣", n.plot, sep=''))
text(-6, max(y.post)-y.band, col='navy', lwd = 2, pos=4,
     sprintf("<x>:␣%.2f", dt.mean))
text(-6, max(y.post)-3*y.band, col='navy', lwd = 2, pos=4,
     sprintf("MAX(P):␣%.2f", alpha.max))
```


The Posterior of the Lighthouse problem



The Posterior of the Lighthouse problem



- the positions of the flashes are marked by the open circles
- the posterior is very broad for small data sets and can also be multimodal if the flashes locations are well separated
- already with ~ 10 measurements the posterior becomes a well shaped-like Gaussian
- it becomes narrower as the data size increases $(\mathrm{FWHM} \propto 1/\sqrt{N})$

➜ a simple average of the measurement data gives us a wrong result

Signal Amplitude in presence of Background

- given a set of counts $\{N_k\}$, masured at values $\{x_k\}$ we want the best estimate of the amplitude of the signal peak and of the background staying below
- for instance, in case of a photon spectrum, we measure the number of photons in bins of wavelength or energy
- this number is proportional to the exposure (time of measurement) and to both signal and background amplitudes through the expression

$$S_k = \Delta t \left[A \exp\left(-\frac{(x_k - x_\circ)^2}{2w^2} \right) + B \right]$$

where Δt is the exposure time and $x_\circ$ and w are the centre and widh of the signal peak

- the number of expected photons is S_k, not generally an integer
- the number of observed photons, N, is an integer number and follows the Poisson distribution

$$P(N|S) = \frac{S^N \mathrm{e}^{-S}}{N!}$$

- and this gives us the Likelihood of the data $(D = \{N_j\})$

$$P(D \mid A, B, M) = \prod_j \frac{S_k^{N_k} \mathrm{e}^{-S_k}}{N_k!}$$

Signal Amplitude in presence of Background

- the model has 5 parameters, but we assume that $x_\circ$, w and Δt are known
- we want to infer $P(A, B \mid D, M)$ from the data, where M identifies the model (the shape of the line and the values of the fixed parameters)
- we adopt a minimalistic prior: we only assume that A and B cannot be negative. Therefore the Prior $P(A, B \mid M)$ is constant when both A and B are positive, and zero otherwise
- the Posterior is

$$P(A, B \mid D, M) = \frac{1}{Z} \prod_j \frac{S_k^{N_k} e^{-S_k}}{N_k!}$$

- and the log Posterior

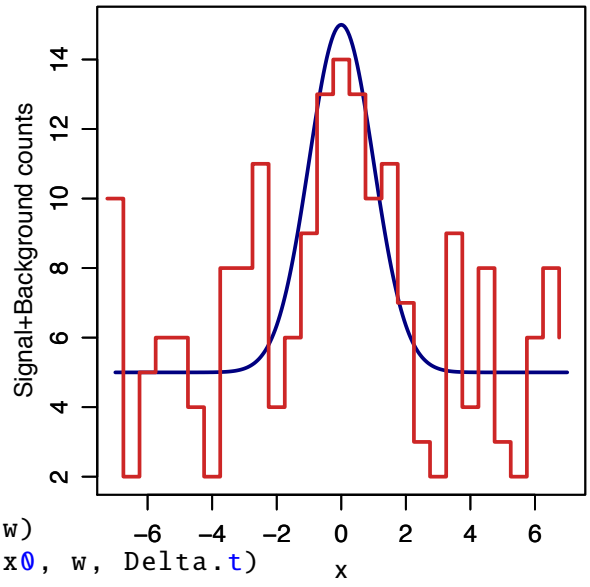
$$L = \log P(A, B \mid D, M) = \text{const} + \sum [N_k \log S_k - S_k]$$

- where the constant term absorbs terms that do not depend on A or B
- the best estimates are given by values of A and B that maximize L

Signal+Background - data generation

- given the positions $\{x_k\}$, the number of expected photons is evaluated using the generative model

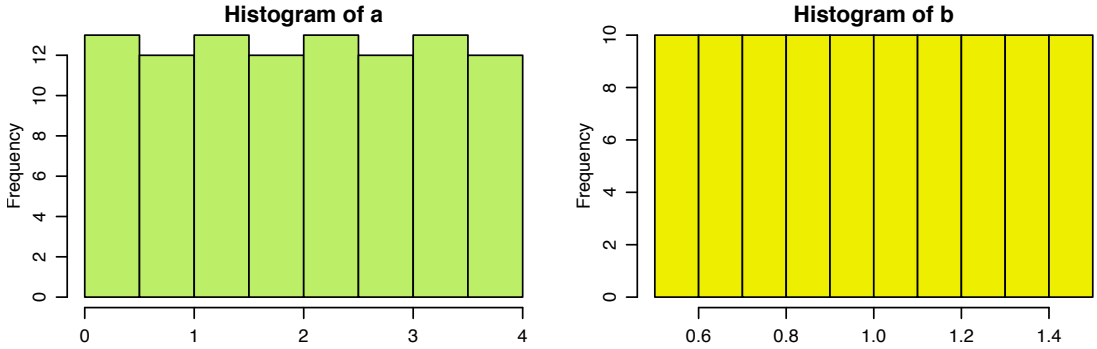
```
# - Generative model
signal <- function(x, a, b, x0, w, t) {
    t * (a*exp(-(x-x0)^2/(2*w^2)) + b)
}

# Define model parameters
x0   <- 0      # Signal peak
w    <-1       # Signal width
A.true  <- 2 # Signal amplitude
B.true  <- 1 # Background amplitude
Delta.t <- 5 # Exposure time

# - Generate the observed data
set.seed(205)
xdat  <- seq(from=-7*w, to=7*w, by=0.5*w)
s.true <- signal(xdat, A.true, B.true, x0, w, Delta.t)
ddat  <- rpois(length(s.true), s.true)

xplot <- seq(from=min(xdat), to=max(xdat), by=0.05*w)
splot <- signal(xplot, A.true, B.true, x0, w, Delta.t)
plot(xplot, splot,
     xlab="x", ylab="Signal+Background␣counts")
par(new=TRUE)
xdat.off <- xdat-0.25
plot(xdat.off, ddat, type='s',col='firebrick3',
     lwd=2,xlim=range(xplot), ylim=range(c(splot, ddat)))
```



Signal+Background - posterior calculation 1

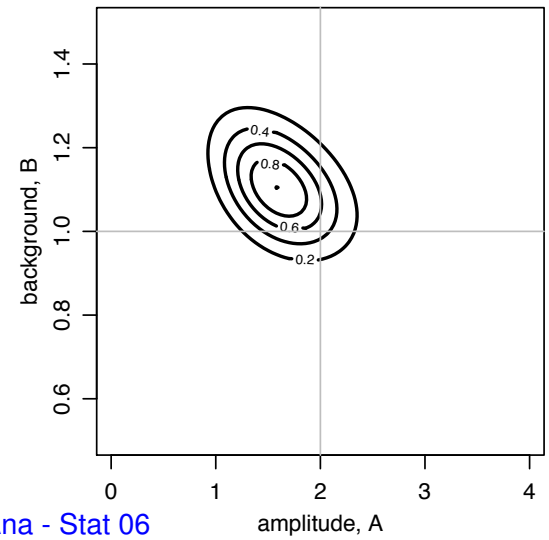
- the posterior has a nonlinear dependence on the parameters
- we calculate it on a grid of values of $\{a_k, b_k\}$
- a regular grid of size $K \times K$, with $K = 100$ is used and the `contour()` function is used to plot lines of constant probability density

```
# - Sampling grid for computing posterior
alim  <- c(0.0, 4.0)
blim  <- c(0.5, 1.5)
Nsamp <- 100
uniGrid <- seq(from=1/(2*Nsamp),
to=1-1/(2*Nsamp), by=1/Nsamp)
delta_a <- diff(alim)/Nsamp
delta_b <- diff(blim)/Nsamp
a <- alim[1] + diff(alim)*uniGrid
b <- blim[1] + diff(blim)*uniGrid
```



Signal+Background - posterior calculation 2

```
# Log posterior
log.post <- function(d, x, a, b, x0, w, t) {
    if(a<0 || b <0) {return(-Inf)} # the effect of the prior
    sum(dpois(d, lambda=signal(x, a, b, x0, w, t), log=TRUE))
}

# Compute log unnormalized posterior, z = ln P^*(a,b|D), on a regular grid
z <- matrix(data=NA, nrow=length(a), ncol=length(b))
for(j in 1:length(a)) {
    for(k in 1:length(b)) {
        z[j,k] <- log.post(ddat, xdat, a[j], b[k], x0, w, Delta.t)
    }
}
z <- z - max(z) # set maximum to zero

# Plot unnormalized 2D posterior as contours.
contour(a, b, exp(z),
        nlevels = 5,
        labcex = 0.5,
        lwd = 2,
        xlab="amplitude, A",
        ylab="background, B")
abline(v=2,h=1,col="grey")
```



Marginal posterior pdf

- given our two parameters posterior pdf

$$P\left(A,B \mid D\right) = \frac{1}{Z_{ab}} P\left(D \mid A,B\right) P\left(A,B\right)$$

- we are interested in the posterior only for one parameter, for instance A➜ we must *marginalize* (integrate) over B

$$P\left(A \mid D\right) = \int P\left(A,B \mid D\right) dB = \frac{1}{Z_{ab}} \int P\left(D \mid A,B\right) P\left(A,B\right) dB$$

- if the Priors are independent, $P(AB) = P(A)P(B)$, marginalizing is like projecting the distribution along an axis

➜ marginalization is a powerful feature of probability analysis because it allows us to include parameters which are an essential part of the model, but which we may not actually be interested in

- We marginalize over them to get the posterior pdf for the parameters of interest

Signal+Background - marginalization

- the marginalization is performed on the grid, simply by summing over one of the two parameters

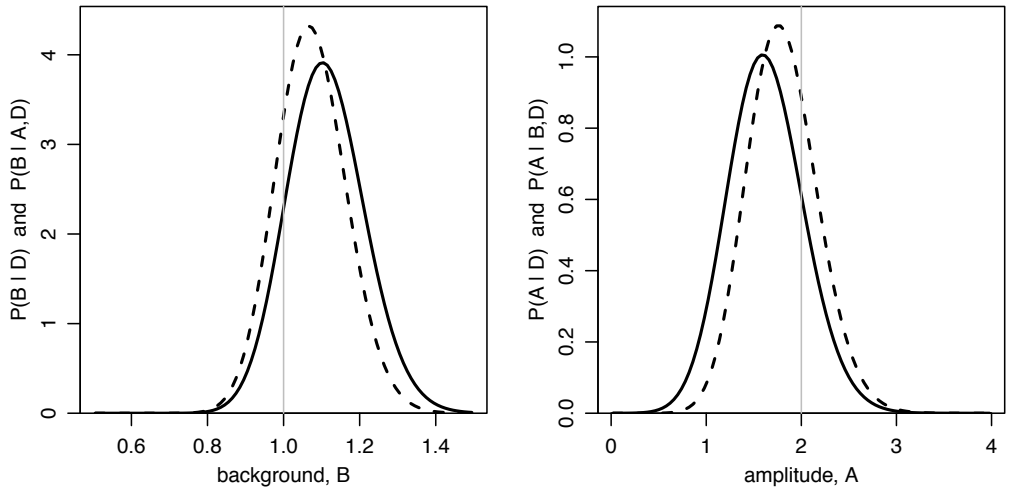
$$P\left(A_J \mid D\right) \sim \Delta B \sum_{k=1}^{K} P\left(A_j, B_k \mid D\right)$$

- where K is the grid size
- finally, the posterior is normalized in via the rectangle rule using our grid

```
# Compute normalized marginalized posteriors, P(a|D) and P(b|D)
# by summing over other parameter. Normalize by gridding.
p_a_D <- apply(exp(z), 1, sum)
p_a_D <- p_a_D/(delta_a*sum(p_a_D))
p_b_D <- apply(exp(z), 2, sum)
p_b_D <- p_b_D/(delta_b*sum(p_b_D))

# Compute normalized conditional posteriors, P(a|b,D) and P(b|a,D)
# using true values of conditioned parameters. Vectorize(func, par)
# makes a vectorized function out of func in the parameter par.
p_a_bD <- exp(Vectorize(log.post, "a")(ddat, xdat, a, B.true,
                                       x0, w, Delta.t))
p_a_bD <- p_a_bD/(delta_a*sum(p_a_bD))
p_b_aD <- exp(Vectorize(log.post, "b")(ddat, xdat, A.true, b,
                                       x0, w, Delta.t))
p_b_aD <- p_b_aD/(delta_b*sum(p_b_aD))
```

Signal+Background - marginalization

```
par(mfrow=c(2,2), mgp=c(2,0.8,0), mar=c(3.5,3.5,1,1), oma=0.1*c(1,1,1,1))

# Plot the 1D marginalized posteriors
plot(b, p_b_D, xlab="background␣B", yaxs="i",
     ylim=1.05*c(0,max(p_b_D, p_b_aD)), ylab="P(B␣|␣D)␣␣and␣␣P(B␣|␣A,D)",
     type="l", lwd=2)
lines(b, p_b_aD, lwd=2, lty=2)
abline(v=B.true, col="grey")

plot(a, p_a_D, xlab="amplitude␣A", yaxs="i",
     ylim=1.05*c(0,max(p_a_D, p_a_bD)), ylab="P(A␣|␣D)␣␣and␣␣P(A␣|␣B,D)",
     type="l", lwd=2)
lines(a, p_a_bD, lwd=2, lty=2)
abline(v=A.true, col="grey")
```



Best estimates and reliability

- finally we want to get the best estimate of our parameters from the inferred posterior

$$\mu_A = \int A \cdot P\left(A \mid D\right) dA \sim \Delta A \sum_{k=1}^{K} A \cdot P\left(A_j \mid D\right)$$

$$\sigma_A^2 = \int (A - \mu_A)^2 P\left(A \mid D\right) dA \sim \Delta A \sum_{k=1}^{K} (A_k - \mu_A)^2 P\left(A_j \mid D\right)$$

- and

$$\begin{aligned} \mathrm{Cov}(A,B) &= \iint (A - \mu_A)(B - \mu_B) P\left(A, B \mid D\right) dA dB \\ &\sim \sum_{j=1}^{K} \sum_{k=1}^{K} \left(A_j - \mu_A\right)(B_k - \mu_B) P\left(A_j, B_k \mid D\right) \\ \rho &= \frac{\mathrm{Cov}(A,B)}{\sigma_A \sigma_B} \end{aligned}$$

- **Note**: there should be a factor $K/(K-1)$ in the variance and covariance calculations, but we can neglect it due to the large number of points in the grid (K)

Signal+Background - marginalization

```
# Compute normalized marginalized posteriors, P(a|D) and P(b|D)
# by summing over other parameter. Normalize by gridding.
p_a_D <- apply(exp(z), 1, sum)
p_a_D <- p_a_D/(delta_a*sum(p_a_D))
p_b_D <- apply(exp(z), 2, sum)
p_b_D <- p_b_D/(delta_b*sum(p_b_D))

# Compute mean, standard deviation, covariance, correlation, of A and B
mean_a <- delta_a * sum(a * p_a_D)
mean_b <- delta_b * sum(b * p_b_D)
sd_a   <- sqrt( delta_a * sum((a-mean_a)^2 * p_a_D) )
sd_b   <- sqrt( delta_b * sum((b-mean_b)^2 * p_b_D) )

# Covariance nurmalization is performed with 'brute force'
# The normalization constant is Z = delta_a*delta_b*sum(exp(z)).
# This is independent of (a,b) so can be calculated outside of the loops.
cov_ab <- 0
for(j in 1:length(a)) {
    for(k in 1:length(b)) {
        cov_ab <- cov_ab + (a[j]-mean_a)*(b[k]-mean_b)*exp(z[j,k])
    }
}
cov_ab <- cov_ab / sum(exp(z))
rho_ab <- cov_ab / (sd_a * sd_b)

cat("␣␣a␣=␣", mean_a, "+/-", sd_a, "\n")
cat("␣␣b␣=␣", mean_b, "+/-", sd_b, "\n")
cat("rho␣=␣", rho_ab, "\n")
```

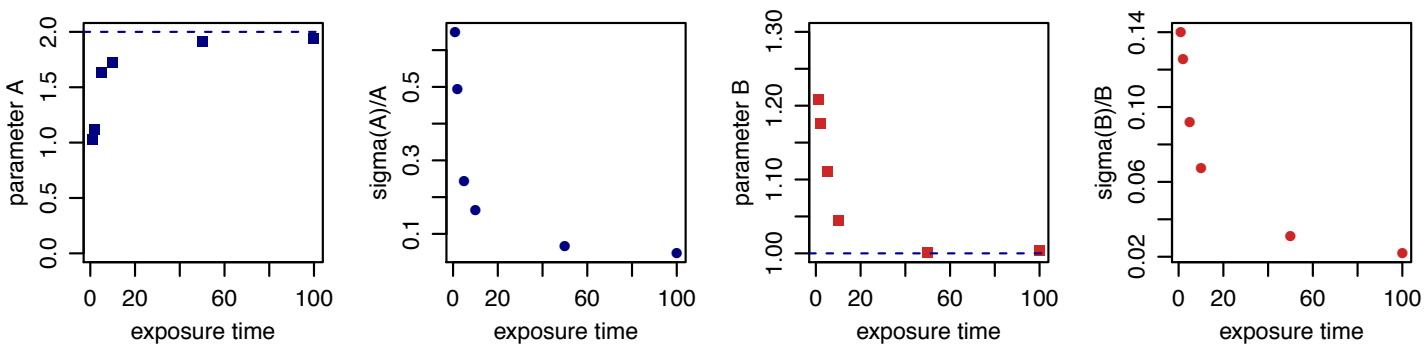
```
a = 1.630189 +/- 0.3983222
b = 1.111212 +/- 0.1020915
rho = -0.3968818
```


Signal+Background - further studies

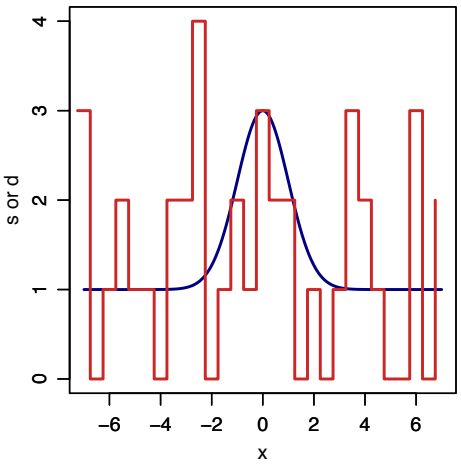
Dependence on exposure

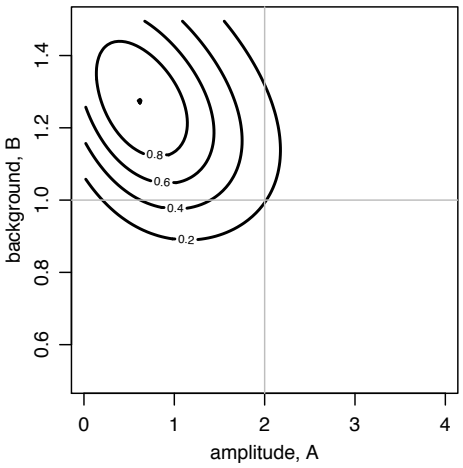
- the exposure time (Δt parameter) is directly connect to the number of collected photons
- larger exposures ➜ more collected photons
- smaller exposures ➜ decrease in accuracy and precision

Q: change the exposure to $\Delta t = \{1, 2, 10, 50, 100\}$ and check the effect on the results



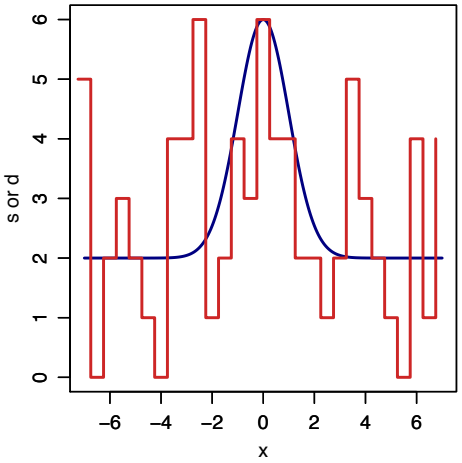
Signal+Background vs Exposure Time

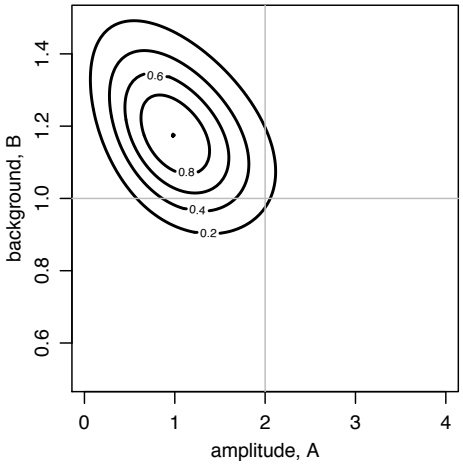




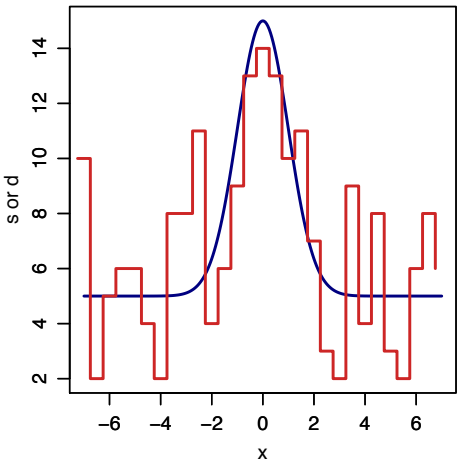
$\Delta t = 1$ a.u.

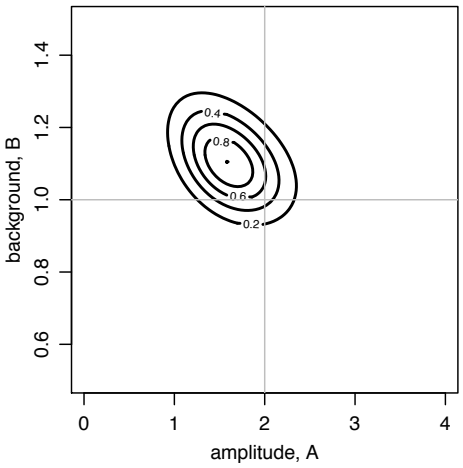
$\Delta t = 2$ a.u.





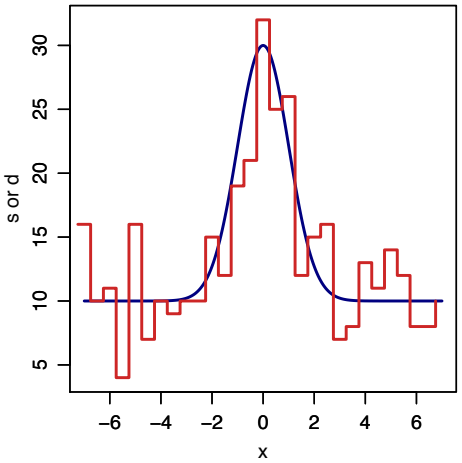
Signal+Background vs Exposure Time

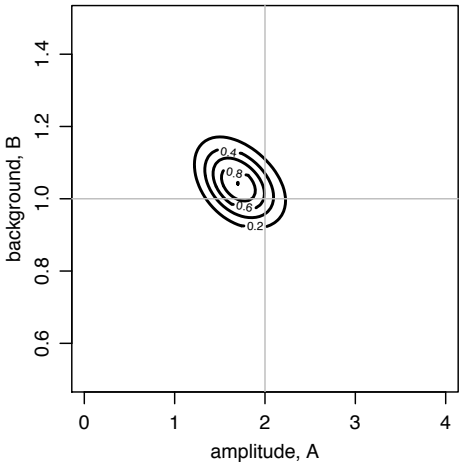




$\Delta t = 5$ a.u.

$\Delta t = 10$ a.u.





Signal+Background : other possible studies

Dependence on resolution

- vary the sampling resolution of used to generate the data, keeping the same sampling range

  ```
  xdat <- seq(from=-7*w, to=7*w, by=0.5*w)
  ```

- change the resolution $w = \{0.1, 0.25, 1, 2, 3\}$

Q: Check the effect on the results

Dependence on resolution

- change the ratio A/B used to simulate the data (keeping both positive in accordance with the prior)

Q: Check the effect on the results

Comparing Frequentist and Bayesian inference for a Bernoulli process

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Stat Lect. 7

Two different approaches

Frequentist paradigm

- it allows to perform inference about the parameter using probabilities calculated from the sampling distribution of the data

➜ the parameter is unknown, but fixed ➜ we cannot associate a probability to it

➜ the only probability is that of the random sample

➜ probabilities are not conditional on the actual data sample that has been measured and are interpreted as a long run relative frequency

- different types of inferences are possible:

1 - point estimation

2 - interval estimation

3 - hypothesis testing

Bayesian paradigm

- the posterior distribution is the key point

➜ it summarizes our belief about the parameter, after we have analyzed the data

➜ it allows to extract all the estimates on the parameter

1-Point Estimation

- a single statistic is calculated from the sample data and used to estimate the unknown parameter
- several theoretical approaches are possible: an example is the Maximum Likelihood Estimation (MLE)
- since the true value of the parameter is unknown, we can judge an estimator only on the sampling distribution of the estimator, i.e. the distribution of the estimator over all the possible random samples
- the expected value of an estimator measures the center of its distribution
- the *Bias* of an estimator is the difference from its expected value and the true value of the parameter

$$\text{Bias}\left[\hat{\theta}, \theta\right] = \text{E}\left[\hat{\theta}\right] - \theta$$

- an estimator is *unbiased* if the mean of its sampling distribution is the true parameter value
- the *Mean Squared Error* of an estimator is

$$\begin{aligned} \text{MSE}\left[\hat{\theta}\right] &= \text{E}\left[\hat{\theta} - \theta\right]^2 \\ &= \int \left(\hat{\theta} - \theta\right)^2 f\left(\hat{\theta} \,|\, \theta\right) \, d\hat{\theta} \end{aligned}$$

- it can be demonstrated that

$$\text{MSE}\left[\hat{\theta}\right] = \text{Bias}\left[\hat{\theta}, \theta\right]^2 + \text{Var}\left[\hat{\theta}\right]$$

Frequentist estimator

- in the Frequentist approach, an unbiased estimator for the Binomial distribution is

$$\hat{p}_F = \frac{y}{n}$$

- where y is the number of successes in n trials
- the properties of the estimator are:

$$\begin{aligned} \text{E}\left[\hat{p}_F\right] &= p \\ \text{Var}\left[\hat{p}_F\right] &= \frac{p(1-p)}{n} = \frac{pq}{n} \\ \text{MSE}\left[\hat{p}_F\right] &= \text{Bias}\left[p_F, p\right]^2 + \text{Var}\left[p_F\right] \\ &= 0^2 + \frac{p(1-p)}{n} \end{aligned}$$

Bayesian estimator

- with the Bayesian approach, we use the posterior mean as an estimate for p
- let's assume we imposed a uniform prior, `Beta(1,1)`
- the posterior mean is

$$\hat{p}_B = m' = \frac{a'}{a'+b'}$$

- with $a' = 1 + y$ and $b' = 1 + n - y$
- therefore

$$\begin{aligned} \hat{p}_B &= \frac{1+y}{1+y+1+n-y} = \frac{y+1}{n+2} \\ &= \frac{y}{n+2} + \frac{1}{n+2} = \frac{np}{n+2} + \frac{1}{n+2} \end{aligned}$$

- the variance of the distribution is

$$\mathrm{Var}\left[\hat{p}_B\right] = \left(\frac{1}{n+2}\right)^2 n\,p\,(1-p)$$

- and the *Mean Square Error* becomes

$$\begin{aligned} \mathrm{MSE}\left[\hat{p}_B\right] &= \left[\frac{np}{n+2} + \frac{1}{n+2} - p\right]^2 + \left(\frac{1}{n+2}\right)^2 n\,p\,(1-p) \\ &= \left(\frac{1-2p}{n+2}\right)^2 + \left(\frac{1}{n+2}\right)^2 n\,p\,(1-p) \end{aligned}$$

Example: point estimation

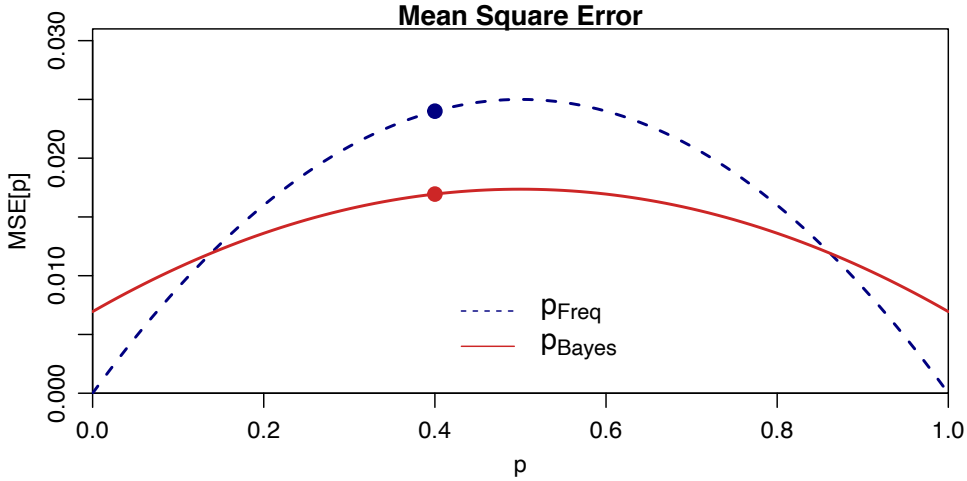
- let's suppose we have a Bernoulli process with $p = \frac{2}{5}$. We perform multiple samples from the distribution and the sample size is $n = 10$
- let's evaluate and compare the *Mean Square Error* for both Frequentist and Bayesian estimators
- we get

$$\mathrm{MSE}\left[\hat{p}_F\right] = \frac{0.4 \times 0.6}{10} = 0.024$$

$$\mathrm{MSE}\left[\hat{p}_B\right] = \left(\frac{1-0.8}{12}\right)^2 + \left(\frac{1}{12}\right)^2 \times 10 \times 0.4 \times 0.6 = 0.0169$$

- we can scan scan the estimator for different values of the true value domain

the Bayesian estimator is closer to the true value over most of the true value range



2-Interval Estimation

- we wish to find an interval (*low, high*) that has a predetermined probability of containing the parameter

Frequentist approach

- the parameter is fixed but unknown
- before the sample is taken, the interval endpoints are random
- once the data is known and the endpoints computed, there is nothing random anymore
- the interval is called a confidence interval for the parameter

- $(1-\alpha)\times 100\%$ confidence interval for a parameter θ is the interval (*low, high*) such that

$$P\left(\text{low} \le \theta \le \text{high}\right) = 1-\alpha$$

- the most common criteria used to select the interval endpoints are

1 equal ordinates on the sampling distribution, $f(\text{low}) = f(\text{high})$

2 equal tail area on the sampling distribution

Frequentist Interval Estimation

once the interval is calculated, there is nothing left that is random

➜ the interval either contains the unknown fixed parameter or it does not

➜ the interval can no longer be regarded as a probability interval

The correct Frequentist paradigm is:

- $(1-\alpha)\times 100\%$ of the random intervals calculated in this way will contain the true value ➜ we have a $(1-\alpha)\times 100\%$ confidence that our interval does cointain it
- it is a misinterpretation to make a probability statement about the parameter θ from the calculated confidence interval

- very often the sampling distribution of the estimator can be approximated with a normal distribution, with the mean equal to the true value of the parameter

➜ the confidence interval gets the form

$$\text{estimator} \pm \text{critical value} \times \text{estimator standard deviation}$$

- if *n* is large:

$$\hat{p}_f = y/n \text{ is normal with mean } p \text{ and } \quad \sigma = \sqrt{p(1-p)/n}$$

- the approximate $(1-\alpha)\times 100\%$ equal area confidence interval for p is

$$\hat{p}_f \pm z_{\alpha/2}\sqrt{\frac{\hat{p}_f(1-\hat{p}_f)}{n}}$$

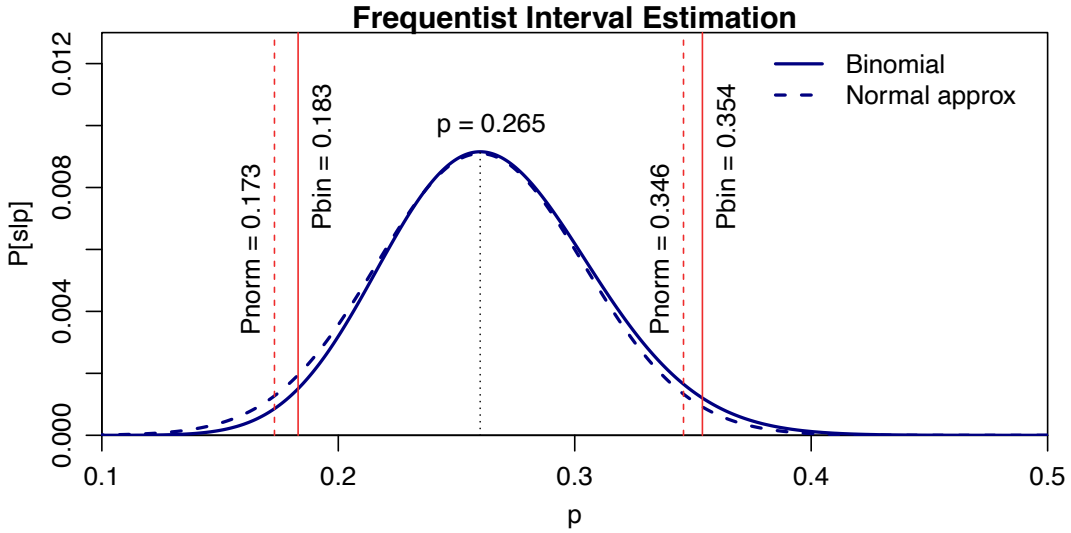
Example: interval estimation (F)

The problem

- a small town residents sample ($n = 100$) are interview about the construction of a new concert hall
- $y = 26$ express a positive opinion about it

Frequentist approach solution

- an unbiased estimator is $\hat{p}_F = y/n = 0.26$
- with standard deviation $\sigma = \sqrt{0.26 * (1 - 0.26)/100} = 0.0439$



Example: interval estimation (B)

Bayesian approach solution

1 - let's select a uniform prior, i.e. $\texttt{Beta}(1,1)$, for our unknown parameter

- our posterior distribution is given by a $\texttt{Beta}$ distribution. since a $\texttt{Beta}$ prior is a conjugate function for the $\texttt{Binomial}$ distribution
- the posterior distribution is

$$\texttt{Beta}(a' = a + y, b' = b + n - y) = \texttt{Beta}(1 + 26, 1 + 74)$$

2 - as a second example, let's choose a $\texttt{Beta}$ prior with a mean value $m = 0.2$ and a standard deviation $\sigma = 0.08$. Since

$$m = \frac{a}{a+b} = p_\circ \quad \text{and} \quad \sigma_\circ^2 = \frac{ab}{(a+b)^2(a+b+1)} = np_\circ(1-p_\circ)$$

- it can be rewritten giving:

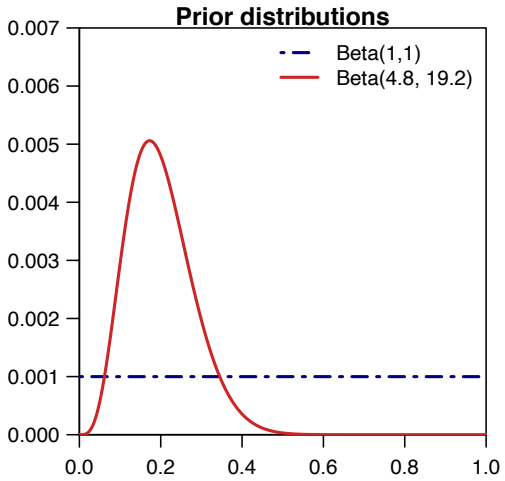
$$a + b + 1 = \frac{p_\circ(1-p_\circ)}{\sigma_\circ^2} \quad \text{and} \quad a + b = \frac{a}{p_\circ}$$

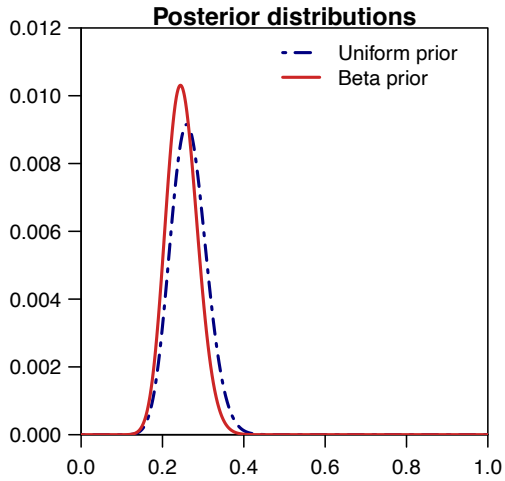
- a $\texttt{Beta}(4.8, 19.2)$ prior gives a posterior distribution

$$\texttt{Beta}(a' = a + y, b' = b + n - y) = \texttt{Beta}(4.8 + 26, 19.2 + 74)$$

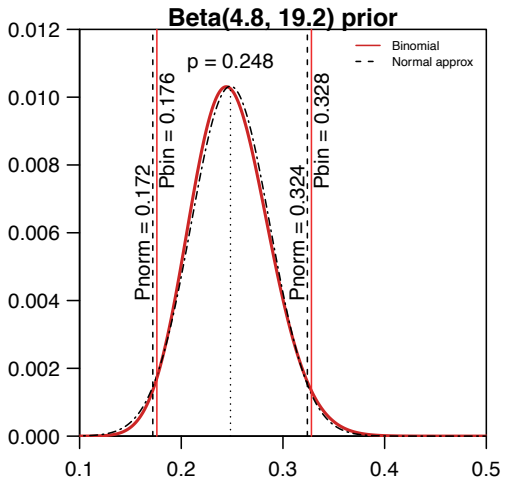
Example: interval estimation (B)

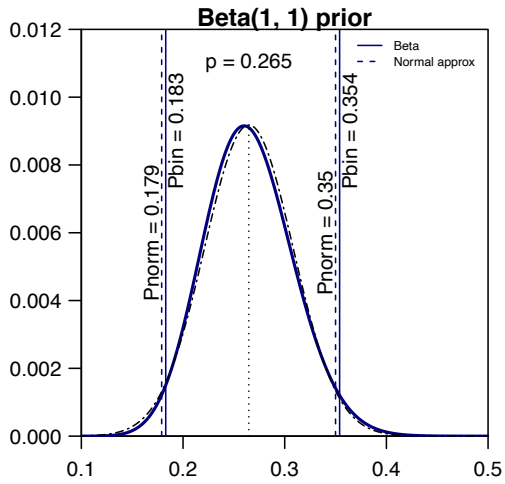
starting with different prior distribution, we get similar posteriors





with the posterior distribution we can calculate the credibility interval





3-Hypothesis Testing

Idea Behind

- researchers have some theory and want to know wheather or not the data actually support that theory
- scientists should not claim the discovery of a new effect if the discrepancy observed in the data could be due to chance alone
- Hypothesis Testing, also called Significance Testing, is the Frequentist statistical method used to check against claims unjustified in the data
- the nonexistence of the effect is set up as the null hypothesis
- when we accept the null hypothesis as true, it does not mean that we believe is is 'literally true'. Rather it means that chance alone remains a reasonable explanation for the observed discrepancy. Therefore we cannot discard chance as the sole explanation
- we distinguish

1 testing a one-side hypothesis when we are interested in detecting the effect in one direction

2 two-sided hypothesis when a test hypothesis is tested against two sided alternatives

3-Hypothesis Testing (HT)

ESP: Extrasensory perception experiment

- θ: probability of correctly choosing the colours
- if participants have paranormal abilities: $\theta > 0.5$

- the researchers has to formulate two distinct and alternative hypotheses:
  - the NULL Hypothesis, H_0: $\theta = 0.5$
  - and the alternative Hypothesis, H_1: $\theta > 0.5$

- the goal of HT is not to show that the alternative hypothesis is TRUE, but to show that the null hypothesis is FALSE

The TRIAL of NULL Hypothesis

- the NULL Hypothesis is the defendant
- the researcher is the persecutor
- the statistical test is the judge

presumption of innocence: the NULL Hypothesis is deemed to be TRUE unless you, the researcher, can prove beyound reasonable doubts that it is FALSE

Errors in HT

- the goal is not to eliminate errors, but to minimize them

	accept H_0	reject H_0
H_0 is TRUE	ok	error, type I
H_0 is FALSE	error, type II	ok

➜ important design principle: control the probability of type error I and keep it below some fixed probability α

➜ α is called the significance level of the test

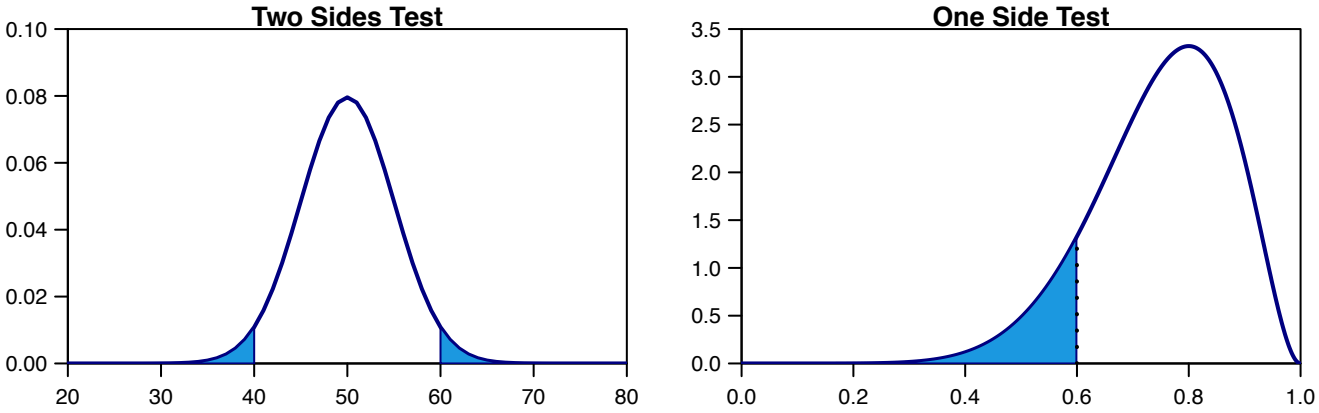
the power of the test is the probability with which we reject the NULL Hypothesis when it is really FALSE

	accept H_0	reject H_0
H_0 is TRUE	$(1-\alpha)$ probability of correct retention	α type I error rate
H_0 is FALSE	β, type II error rate	$(1-\beta)$, power of the test

- a powerful HT has small values of β while keeping α fixed at some small desired level
- α values used by convention among scientists: 0.05, 0.1 and 0.01

HT prescriptions

1) setup the NULL and alternative hypotheses
2) determine what the sampling distribution of the test statistic would be if the NULL hypothesis were TRUE
3) choose the level of significance, α and associate the critical regions to the distribution



4) calculate the value of the test statistic for the real data and compare to the critical value to make our decision : critical region ➜ values for which we would reject the NULL hypothesis
5) if we reject the NULL hypothesis, we say that the test has produced a significant result

Example: One-Side Hypothesis Test

The problem

- we wish to test the effect of a new treatment, to verify if it is better than the standard treatment as a parameter in the model
- $p =$ fraction of patients who benefit from the new treatment
- $p_\circ =$ fraction of patients who benefit from the standard treatment
- we know that $p_\circ = 0.6$
- 10 patients are given the new treatment and be observe that $y = 8$ patients benefit from the new treatment
- do we conclude that $p > 0.6$ at the 5% level of significance ?

Frequentist approach

1 - setup a null hypothesis

$$H_\circ : p \leq 0.6$$

2 - the alternative hypothesis (the new treatment is better) is

$$H_1 : p > 0.6$$

3 - the NULL distribution of the test statistic is the sampling distribution of the test statistic, given that the NULL hypothesis is true

$$\texttt{Binom}(y \,|\, n, p = 0.6)$$

Example: One-Side Hypothesis Test (F)

4 - choose a level of significance

$$\alpha = 5\%$$

Note that since y has a discrete distribution, only some values of α are possible

5 - the rejection region is chosen so that it has a probability of α under the NULL distribution (Neyman and Pearson approach)

$y = 8$ lies in the acceptance region ➜ we do not reject $H_\circ$

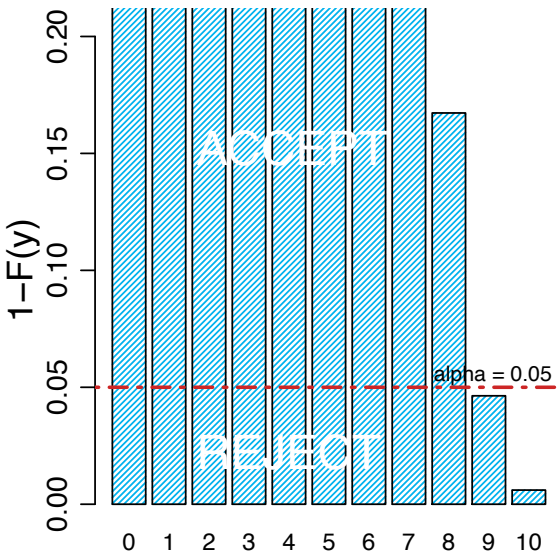
6 - the p-value is the probability of getting what we observed:

$$\texttt{p-value} = \sum_{y_{obs}}^{n} f(y \mid p_\circ) = 0.1672$$

if $\texttt{p-value} < \alpha$ ➜ the test statistic lies in the rejection region

α represents the long-run rate of rejecting a true null hypothesis

7 - an alternative way, due to Fisher, is to reject $H_\circ$ if $\texttt{p-value} < \alpha$



Example: One-Side Hypothesis Test (B)

Bayesian approach

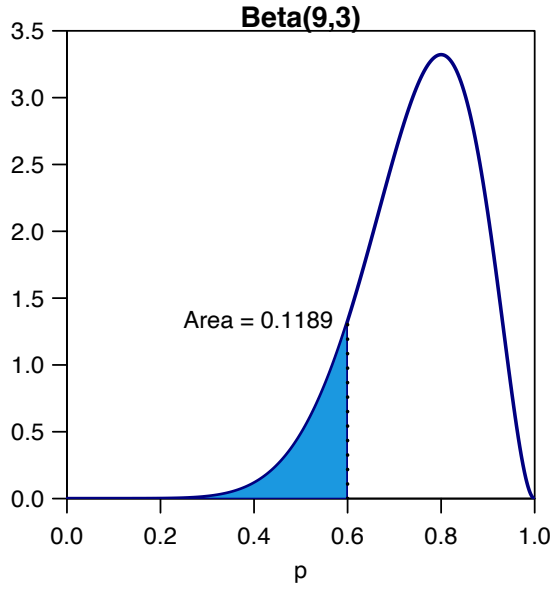
- we wish to test $H_\circ : p \leq p_\circ$ versus $H_1 : p > p_\circ$ at a level of significance α
- we evaluate the posterior probability of the null hypothesis, and integrate over the required region:

$$P\left(H_\circ : p \leq p_\circ \mid y\right) = \int_0^{p_\circ} g\left(p \mid y\right)\, dp$$

- we reject the null hypothesis if the posterior probability is less than α, the level of significance
- we use a uniform prior, $\texttt{Beta}(1,1)$, for the parameter p
- given $y = 8$, the posterior density is $\texttt{Beta}(9,3)$

$$P\left(p \leq 0.6 \middle| y = 8\right) = \int_0^{0.6} \frac{\Gamma(12)}{\Gamma(3)\Gamma(9)} p^8 (1-p)^2 dp$$

$$= 0.1189$$



- the result, 11.89%, is higher than $\alpha = 5\%$, therefore we cannot reject the null hypothesis at the 5% level of significance

Example: Two-Sides Hypothesis Test

- we want to detect any changes from the value $p_\circ$
- we setup the null hypothesis $H_\circ : p = p_\circ$ against the alternative hypothesis $H_1 : p \neq p_\circ$

The problem

- a coin is tossed $n = 15$ times
- we observe $y = 10$ heads

Q: Is the coin fair ?

Frequentist approach

1 - setup a null hypothesis

$$H_\circ : p = 0.5$$

2 - we want to test it against the alternative hypothesis

$$H_1 : p \neq 0.5$$

3 - the null distribution is the sampling distribution of y: $\texttt{Bin}(y \mid n = 15, p = 0.5)$

Example: Two-Sides Hypothesis Test (F)

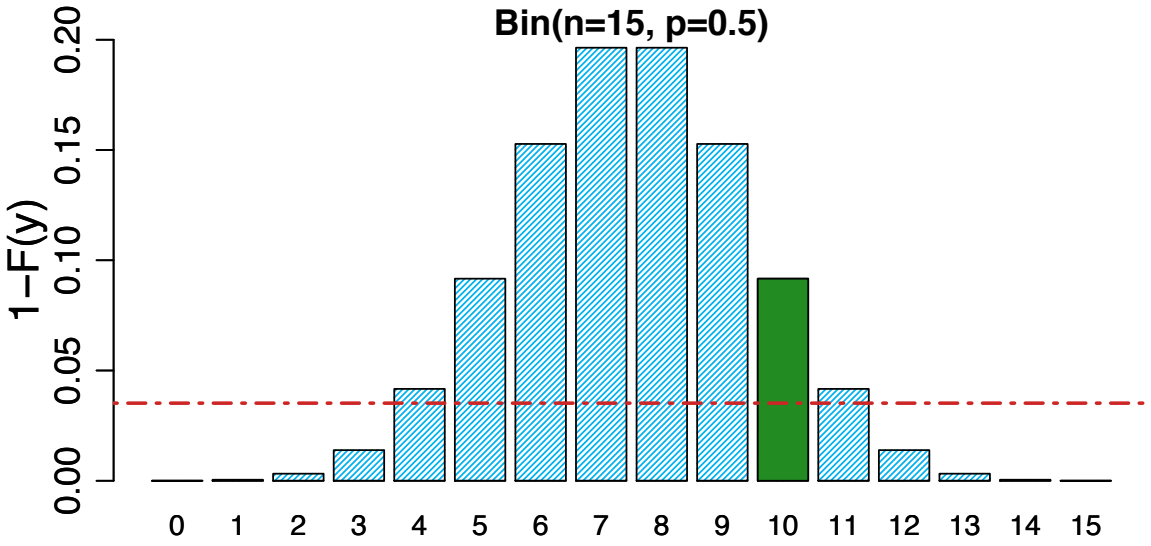
4 - in defining the rejection region, we take into account that y has a discrete distribution, and choose the level of significance as close to 5% as possible

$$\{y \leq 3\} \cup \{y \geq 12\} \text{ with } \alpha = 0.0352$$

5 - we observe $y = 10$, which lies inside the acceptance region

6 - we would have not rejected the null hypothesis also evaluating the `p-value`

$$P(y \geq 10) + P(y \leq 5) = 0.3018$$



Example: Two-Sides Hypothesis Test (B)

Bayesian approach

- the posterior distribution of the parameter, given the data, constraints our entire belief after getting the data
- but since the probability of an exact value represented by the point null hypothesis is zero

➜ need a correspondence similar to that of confidence intervals, using credible intervals

- we compute a $(1-\alpha)\times 100\%$ credible interval for p
- if $p_\circ$ lies inside the interval, we do not reject the null hypothesis, $H_\circ$; if it is outside, we reject $H_\circ$

The problem

- $n = 15$ coin tosses. We observe $y = 10$ heads

1 - set up a uniform prior $\texttt{Beta}(1,1)$

2 - the posterior is $\texttt{Beta}(10+1, 5+1)$

3 - we calculate a 95% Bayesian credible interval

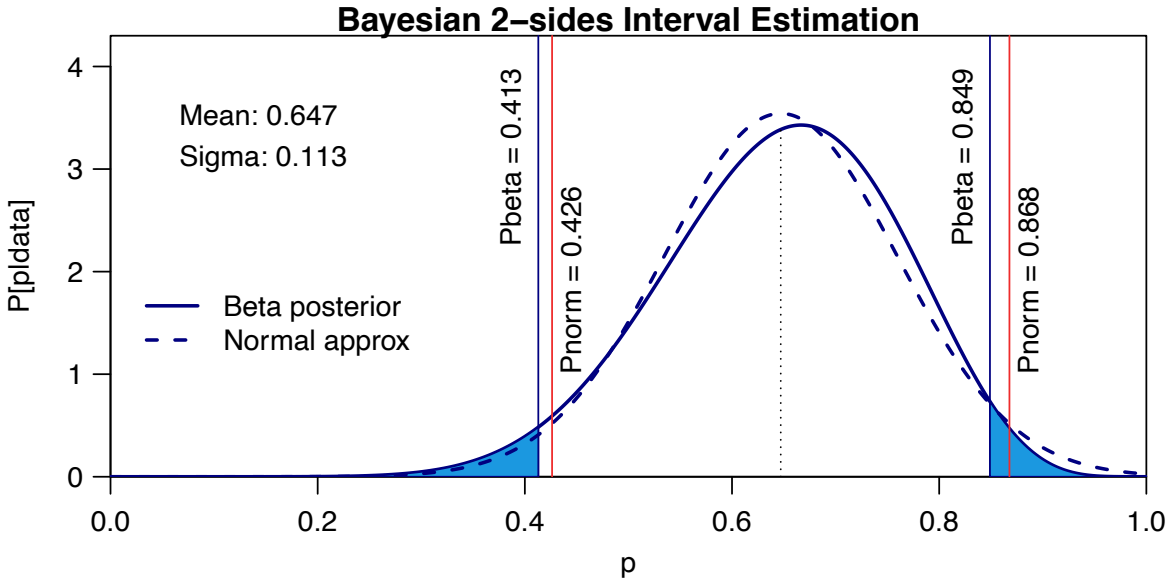
Example: Two-Sides Hypothesis Test (B)

4 - using a normal approximation we would get

$$\frac{11}{17} \pm 1.96 \times \sqrt{\frac{11\times 6}{(11+6)^2(11+6+1)}} = 0.647 \pm 0.221$$

5 - our credibility interval is:

- $(0.413, 0.849)$, using a $\texttt{Beta}$ posterior
- $(0.426, 0.868)$, using a $\texttt{Normal}$ approximation



Some considerations on the *p*-value of the test

Neyman view

- the HT described does not make a distinction at all between a result that is barely significant and those highly significant
- let's run several HT on the same data:

Value of α	0.05	0.04	0.03	0.02	0.01
Reject H_0 ?	Y	Y	Y	N	N

- between 0.02 and 0.03 there is a value of α that would allow us to reject the NULL hypothesis

  the *p*-value is defined to be the smallest Type I error rate (α) that we are willing to tollerate if we want to reject the NULL hypothesis

- *p* summarizes all the possible hypothesis tests that we could have run:
  if $p \leq \alpha$ we would reject the NULL hypothesis

Some considerations on the *p*-value of the test

but

- the *p* value is not the probability that the NULL hypothesis is TRUE
- this statement is absolutely and completely wrong:

1) NULL Hypothesis tesing is a frequentist tool: we are not allowed to assign probability to a NULL hypothesis
   according to this view of probability, the NULL hypothesis is either TRUE or FALSE

Running HT in R

- R contains a whole lot of functions correspionding to different kinds of hypothesis test

```
binom.test(x=62, n=100, p=0.5)

        Exact binomial test

data:  62 and 100
number of successes = 62, number of trials = 100, p-value = 0.02098
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.5174607 0.7152325
sample estimates:
probability of success
                  0.62
```


Summary - global considerations

Frequentist paradigm

- it handles, separately, point estimation, confidence intervals and hypothesis tests
- the Frequentist statistics considers the parameter a fixed but unknown constant
- the sampling distribution of a statistic is its distribution over all the possible random samples, given the fixed parameter value
- the only probability allowed is a long-run relative frequency

Bayesian paradigm

- it bases all the estimates on the posterior distribution of the parameter

Summary - confidence/credibility intervals

Frequentist paradigm

- a $(1-\alpha)\times 100\%$ Frequentist interval for a parameter θ is an interval (θ_l, θ_h) such that
$$P(\theta_l \leq \theta \leq \theta_h) = 1-\alpha$$
- $(1-\alpha)\times 100\%$ of the random intervals calculated this way do contain the true value ➜ we say we are $(1-\alpha)\times 100\%$ confident that the calculated interval contains the true parameter
- the `p-value` allows to reject the null hypothesis, at level α, if `p-value`$< \alpha$
- the `p-value` is not the probability the null hypothesis is true. It is the probability of observing what we observed given that the null hypothesis is true

Bayesian paradigm

- a $(1-\alpha)\times 100\%$ Bayesian credible interval for a parameter θ is a range of parameter values that has a posterior probability $(1-\alpha)$

Statistical Models and Inference - Part III

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Stat Lect. 8

Bayesian inference for Normal distribution

- many random variables seem to follow a normal distribution, at least approximately
- any random variable that is the sum of a large number of similar size random variables from independent causes ➜ approximately normal
- let's analyze a single observation from a conditional density $f(y \mid \mu)$ that is known to be normal with known variance σ^2
- we have a discrete set of possible k values for the mean, $\mu_1, \mu_2 \ldots \mu_k$
- thanks to Bayes' theorem

$$P\left(\mu \mid D, \sigma\right) = \frac{f\left(D \mid \mu, \sigma\right) g\left(\mu \mid \sigma\right)}{\int f\left(D \mid \mu, \sigma\right) g\left(\mu \mid \sigma\right) d\mu}$$

Single observation Likelihood

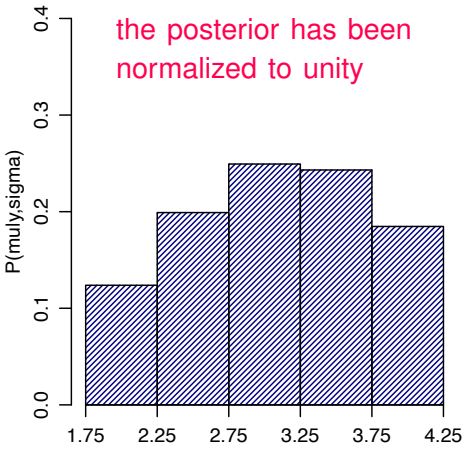
- the probability of the measurement of having a value y is

$$P\left(y \mid \mu, \sigma\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\frac{(y-\mu)^2}{\sigma^2}\right]$$

Example: single normal observation

The problem

- let's assume our variance is $\sigma^2 = 1$
- we know μ can have 5 possible values 2.0, 2.5, 3.0, 3.5 and 4.0
- a single observation is taken with $y = 3.2$



A Bayesian solution

- for the prior, we assume all values are equally possible
- we introduce a standardized variable $z = (y - \mu)/\sigma$
- let's report evaluated data in a table:

μ	$g(\mu\|\sigma)$ Prior	z	$f(y\|\mu,\sigma)$ Likelihood	$f \times g$	$P(\mu\|y,\sigma)$ Posterior
2.0	0.2	1.2	0.1942	0.03884	0.1238
2.5	0.2	0.7	0.3123	0.06245	0.1991
3.0	0.2	0.2	0.3910	0.07821	0.2493
3.5	0.2	-0.3	0.3814	0.07628	0.2431
4.0	0.2	-0.8	0.2897	0.05794	0.1847
				0.31372	1.0000

Estimating the mean of a Normal distribution (1)

- given a set of N measurements, $D = \{y_j\}$, what is the best estimate of the parameter μ and how confident are we with the prediction ?
- let's assume that σ is known and is the same for all the measurements
- from Bayes' theorem

$$P\left(\mu \,\middle|\, D,\sigma\right) \propto P\left(D \,\middle|\, \mu,\sigma\right) \times P\left(\mu \,\middle|\, \sigma\right)$$

- we assume that data are independent, i.e. a measurement of one datum does not interfere on the outcome of another (given μ and σ)
- the Likelihood of the data is

$$P\left(D \,\middle|\, \mu,\sigma\right) = \prod_j P\left(y_j \,\middle|\, \mu,\sigma\right) = \prod_j \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\frac{(y_j-\mu)^2}{\sigma^2}\right]$$

- since the knowledge of the width of a Gaussian distribution does not tell us anything about the position of its centre, let us assume a uniform Prior pdf

$$P(\mu \,|\, \sigma) = P(\mu) = \begin{cases} \dfrac{1}{\mu_{max} - \mu_{min}} & \text{for } x \in [\mu_{min}, \mu_{max}] \\ 0 & \text{otherwise} \end{cases}$$

Estimating the mean of a Normal distribution (2)

- let's combine Likelihood and Prior and write the natural logarithm of the posterior

$$L = \ln P\left(\mu \mid D, \sigma\right) = \text{const} - \sum_j \frac{(y_j - \mu)^2}{2\sigma^2}$$

- differentiating L and setting it to zero

$$\frac{dL}{d\mu} = \sum_j \frac{y_j - \mu}{\sigma^2} = 0 \Rightarrow \mu_\circ = \frac{1}{N}\sum_j y_j$$

- the reliability of the estimate is given by the second derivative

$$\left.\frac{d^2 L}{d\mu^2}\right|_{\mu_\circ} = -\sum_j \frac{1}{\sigma^2} = \frac{N}{\sigma^2}$$

- therefore

$$\mu = \mu_\circ \pm \frac{\sigma}{\sqrt{N}}$$

Estimating the mean of a Normal distribution (3)

- our estimate relies on the validity of a quadratic expansion of the natural logarithm of the Posterior around the maximum
- for the Gaussian distribution, this is an exact identity, because all higher derivatives of L are zero

- what happens when data have individual errors σ_j ?
- our single measurement Likelihood becomes

$$P\left(y_j \mid \mu, \sigma_j\right) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left[-\frac{1}{2}\frac{(y_j - \mu)^2}{\sigma_j^2}\right]$$

- and the logarithm of the Posterior

$$L = \ln P\left(\mu \mid \{y_j\}, \{\sigma_j\}\right) = \text{const} - \sum_j \frac{(y_j - \mu)^2}{2\sigma_j^2}$$

- taking the derivative of L and setting it to zero

$$\frac{dL}{d\mu} = \sum_j \frac{y_j - \mu}{\sigma_j^2} = 0 \Rightarrow \mu_\circ = \sum_j \frac{y_j}{\sigma_j^2} \Big/ \sum_j \frac{1}{\sigma_j^2}$$

- the reliability of the estimate is given by the second derivative

$$\left.\frac{d^2 L}{d\mu^2}\right|_{\mu_\circ} = -\sum_j \frac{1}{\sigma_j^2} \quad \text{and, therefore} \quad \mu = \mu_\circ \pm \left(\sum_j 1/\sigma_j^2\right)^{-1/2}$$

Single observation with a Normal Prior

- let's assume our Prior has a Normal shape with mean m and variance s^2, $\texttt{Norm}(m, s^2)$

$$g(\mu \mid m, s) \propto \exp\left[-\frac{1}{2}\frac{(\mu - m)^2}{s^2}\right]$$

- the shape of the Likelihood is

$$f(y \mid \mu, \sigma) \propto \exp\left[-\frac{1}{2}\frac{(y - \mu)^2}{\sigma^2}\right]$$

- the product Likelihood $\times$ prior becomes

$$f(y \mid \mu, \sigma) \times g(\mu \mid m, s) \propto \exp -\frac{1}{2}\left[\frac{(y - \mu)^2}{\sigma^2} + \frac{(\mu - m)^2}{s^2}\right]$$

- with little algebra, it can be seen that the Posterior is a Normal distribution itself with mean and variance given by

$$m' = \frac{\sigma^2 m + s^2 y}{\sigma^2 + s^2} \quad \text{and} \quad (s')^2 = \frac{\sigma^2 s^2}{\sigma^2 + s^2}$$

➜ the $\texttt{Norm}(m, s^2)$ distribution is the conjugate family for the normal observation distribution (i.e. likelihood) with known variance

Updating rules for Normal inference with fixed variance

Single observation y

- the precision is the reciprocal of the variance, and we know from basics probability that precisions are additive. Therefore:

$$\frac{1}{(s')^2} = \frac{1}{s^2} + \frac{1}{\sigma^2} = \frac{\sigma^2 + s^2}{\sigma^2 s^2}$$

- the Posterior mean is given by

$$m' = \frac{\sigma^2 m + s^2 y}{\sigma^2 + s^2} = \frac{\sigma^2}{\sigma^2 + s^2} m + \frac{s^2}{\sigma^2 + s^2} y$$

- which can also be written as

$$m' = \frac{1/s^2}{1/\sigma^2 + 1/s^2} m + \frac{1/\sigma^2}{1/\sigma^2 + 1/s^2} y$$

Multiple observations $y_1, y_2 \ldots y_n$

- with the definition $\overline{y} = \frac{1}{N}\sum_j y_j$, it is possible to demonstrate that

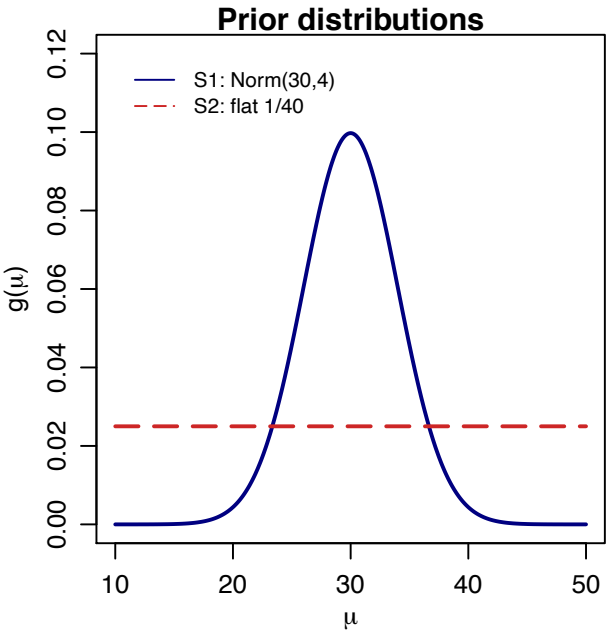
$$\frac{1}{(s')^2} = \frac{1}{s^2} + \frac{n}{\sigma^2} = \frac{\sigma^2 + n s^2}{\sigma^2 s^2} \quad \text{and} \quad m' = \frac{1/s^2}{n/\sigma^2 + 1/s^2} m + \frac{n/\sigma^2}{n/\sigma^2 + 1/s^2} \overline{y}$$

Example : aquaculture

- two students have been asked to estimate the average length of a special fish in its first year of age leaving in a mountain lake
- previous studies in other lakes have shown that the length of the fish has a Normal distribution with known standard deviation $\sigma = 2$ cm

Assigning the Priors

1. Student 1 decides that her prior mean is $m = 30$ cm. Moreover she thinks that for such kind of fish in its first year it is not possible to have length below 18 cm or above 42 cm. Therefore her standard deviation is $s = 4$ cm
   ➜ her Prior is $\texttt{Norm}(30, 4^2)$
2. Student 2 does not know anything about this kind of fish, therefore he decides to assume a Uniform Prior



Example : aquaculture

- they take a random sample of 12 fish in their first year and they find that the sample mean is

$$\overline{y} = 32 \text{ cm}$$

Evaluating the Posteriors

1. Student 1, using the simple rule for the conjugate prior, gets:

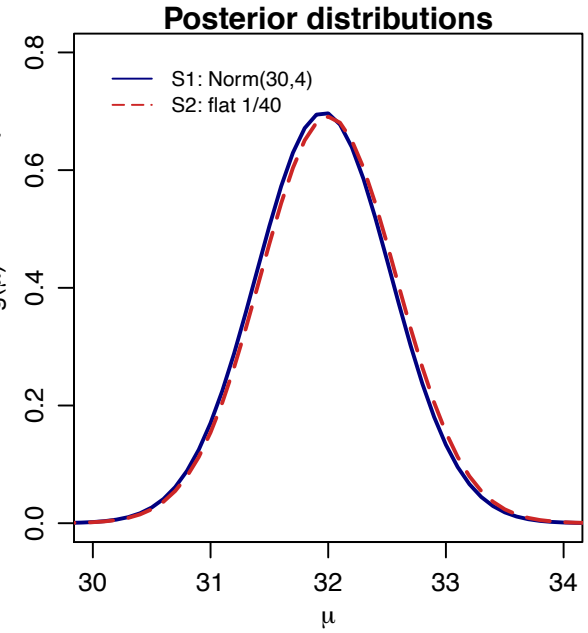
$$\frac{1}{(s')^2} = \frac{1}{4^2} + \frac{12}{2^2} \quad \text{which gives:} \quad s' = 0.5714$$

The Posterior mean is

$$m' = \frac{1/4^2}{1/0.3265} \times 30 + \frac{12/2^2}{1/0.3265} \times 32 = 31.96$$

2. Student 2 uses a flat prior and gets

$$(s')^2 = \frac{2^2}{12} = 0.3333 \quad \text{and} \quad m' = 32$$



Example: K/Ar rock dating methods

The problem

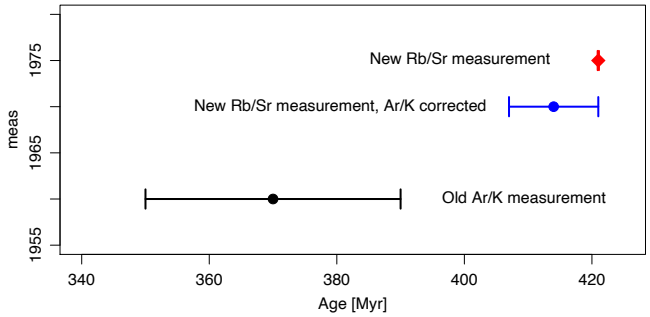
- K/Ar dating methods have been developed in the 60s for geochronology and archaeology research
- a measurement in the 60s of some rock samples gave an age $T_1 = 370 \pm 20$ Myr
- in the 70s, new methods based on the Rb/Sr method allowed to reach more precise measurements with a precision of $\sigma_2 = 8$ Myr and with a measurement result $T_2 = 421$ Myr

How to combine the measurements

- we assume that the measurements of the rocks with the K/Ar method gave $t_1 \sim \text{Norm}(\mu = 370, \sigma^2 = 20^2)$
- investigations with Rb/Sr will produce results of the type $t \sim \text{Norm}(\mu, 8^2)$ with well established precision
- the new prior is $\text{Norm}(m = 370, s = 8^2)$:

$$(s')^2 = \frac{\sigma^2 s^2}{\sigma^2 + s^2} = \frac{8^2 \cdot 20^2}{8^2 + 20^2} \sim 55 \sim 7^2$$

$$m' = \frac{\sigma^2 m + s^2 t_2}{\sigma^2 + s^2} = \frac{8^2 \cdot 370 + 20^2 \cdot 421}{8^2 + 20^2} = 414$$



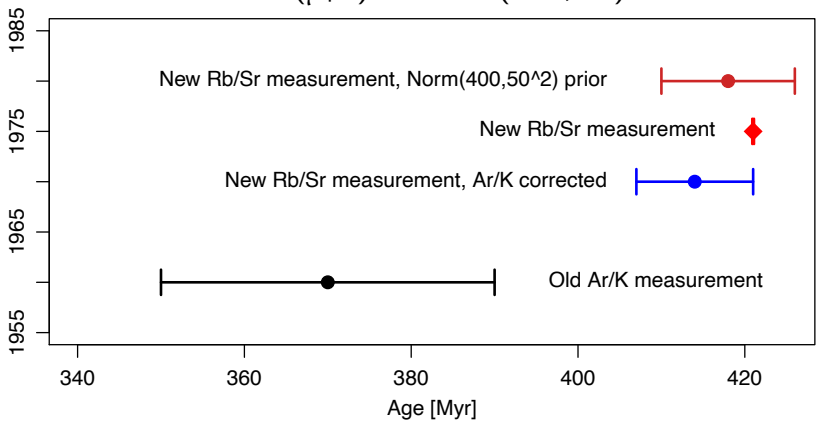
- the posterior for the age of the rock is $\text{Norm}(414, 7^2)$

Example: K/Ar rock dating methods (2)

Performing new measurements

- another scientist performs the same measurements but he is not aware of previous K/Ar dating results
- he considers a Normal prior with the assumption that the rock age is 400 ± 50 Myr
- the posterior variance is $(s')^2 = \frac{\sigma^2 s^2}{\sigma^2 + s^2} = \frac{1}{50^{-2} + 8^{-2}} \sim 62 \sim 8^2$
- and the posterior mean is $m' = \frac{\sigma^2 m + s^2 t_2}{\sigma^2 + s^2} = 62 \cdot \left(\frac{400}{50^2} + \frac{421}{8^2}\right) = 418$
- therefore the posterior is

$$P(\mu|T) = \text{Norm}(418, 8^2)$$



Bayesian Credible Interval for Normal μ

The variance is known

- $\{y_1 \dots y_n\}$ follows Norm(μ, σ^2)
- $\overline{y} = \frac{1}{n} \sum_{j=1}^{n} y_j$
- using either a flat prior or Norm(m, s^2) prior, the $(1-\alpha) \times 100\%$ credible interval for μ is:

$$m' \pm z_{\alpha/2} \times s'$$

with $z_{\alpha/2}$ the quantiles for a standardized normal distribution

The variance is unknown

- we evaluate the sample variance $\hat{\sigma}^2 = \frac{1}{n-1} \sum_{j=1}^{n} (y_j - \mu)^2$
- the $(1-\alpha) \times 100\%$ credible interval for μ is:

$$m' \pm t_{\alpha/2} \times s'$$

with $t_{\alpha/2}$ the quantiles for a Student's t distribution with $n-1$ degrees of freedom

Predictive density for the next observation

- $y_1, \dots, y_n, y_{n+1}$ is a random sample from a Normal distribution with mean μ and known variance σ^2
- with bayesian statistics it is possible to write a conditional probability for the next random observation, given the actual random sample:

$$f\left(y_{n+1} | y_1 \dots y_n\right)$$

- the question is how to combine the uncertainty from the measured sample with that in the observation distribution

➜ by writing Bayes theorem and using the likelihood and prior distribution

- a Theorem of probability theory says that if

$$X \sim \text{Norm}(\mu_X, \sigma_X^2) \quad \text{and} \quad Y \sim \text{Norm}(\mu_Y, \sigma_Y^2)$$

➜

$$Z = X + Y \sim \text{Norm}(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$$

- therefore, writing
  $y_{n+1} = y_{n+1} - \mu + \mu$
  $y_{n+1} - \mu \sim \text{Norm}(0, \sigma^2)$
  $\mu \sim \text{Norm}(m, s^2)$
- we get:

$$y_{n+1} \sim \text{Norm}(m,\ \sigma^2 + s^2)$$

Confidence Interval versus Credibility Interval

- we perform inference about the population mean when we have a random sample from a normally distributed population

Frequentist Confidence Interval

$$P\left(\mu - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} < \overline{y} < \mu + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right) = 1-\alpha$$

$\overline{y}$ is the sample mean and follows a Norm$(\mu, \sigma^2/n)$ distribution

we can re-write it as

$$P\left(\overline{y} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} < \mu < \overline{y} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right) = 1-\alpha$$

- μ is a fixed but unknown parameter
- $(1-\alpha)\times 100\%$ of the intervals so computed will contain the true value
- by taking the random sample, and computing $\overline{y}$ there is nothing random left to attach a probability
- the computed interval either contains the true value or it does not

Confidence Interval versus Credibility Interval

Bayesian Credibility Interval

- using a flat prior for μ, the posterior mean is $m' = \overline{y}$
- the posterior variance is $(s')^2 = \sigma^2/n$

the interval is

$$P\left(\overline{y} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} < \mu < \overline{y} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right) = 1-\alpha$$

the same form of the frequentist C.I., but with different interpretation:

- μ is a random variable ➜ probability statements are allowed
- the Credibility Interval is computed from the posterior distribution, given the observed sample
- the Credibility Interval contains a conditional probability of containing μ, given the data
- we are not concerned with a repetition of the experiment giving all possible data sets ➜ the only data set that matters is the one that occurred

Frequentist one-side HT for Normal μ

1 - setup the null hypothesis and alternative hypotheses

$$H_\circ : \mu \leq \mu_\circ \quad \text{versus} \quad H_1 : \mu > \mu_\circ$$

2 - the null distribution for $\overline{y}$ is Norm$(\mu_\circ, \sigma^2/n)$
The null distribution of the standardized variable $z \sim$ Norm$(0,1)$:

$$z = \frac{\overline{y} - \mu_\circ}{\sigma/\sqrt{n}}$$

3 - choose a level of significance α

4 - determine the rejection region. This is the region that has probability α when the NULL hypothesis is true. For $\alpha = 0.05$ ➜ the rejection region is $z > 1.645$

5 - take the sample and compute $\overline{y}$. If the value falls in the rejection region, we reject the hypothesis at level of significance α, otherwise we do not reject the NULL hypothesis

6 - or we can compute the P-value, which is the probability of observing what we observed, or something more extreme, given the NULL hypothesis:

$$P_{value} = P\left(z \geq \frac{\overline{y} - \mu_\circ}{\sigma/\sqrt{n}}\right)$$

if $P_{value} \leq \alpha$, we reject the NULL hypothesis, otherwise we cannot reject it

Bayesian one-side HT for Normal μ

1 - the posterior distribution

$$g(\mu | y_1 \ldots y_n)$$

summarizes our entire belief about the parameter, after having seen the data

2 - we setup the two hypotheses:

$$H_\circ : \mu \leq \mu_\circ \quad \text{versus} \quad H_1 : \mu > \mu_\circ$$

3 - we choose a level of significance α

4 - testing a one-side hypothesis is done by computing the following:

$$P(H_\circ : \mu \leq \mu_\circ | y_1 \ldots y_n) = \int_{-\infty}^{\mu_\circ} g\left(\mu \,\middle|\, y_1 \ldots y_n\right)\, d\mu$$

when the Posterior is Norm$(m', (s')^2)$ the computation is straightforward:

$$P(H_\circ : \mu \leq \mu_\circ | y_1 \ldots y_n) = P\left(Z \leq \frac{\mu_\circ - m'}{s'}\right)$$

5 - if the probability is less than α, we reject the NULL hypothesis and we can conclude that $\mu > \mu_\circ$

Example: One-Side Hypothesis Test (F)

The problem

- we wish to estimate the length of one-year old mountain trouts in a mountain river
- from measurements performed in the previous years, we know that the average length is $\mu_\circ = 31$ cm
- we want to test if the mean length is greater than that value, i.e.

$$H_\circ : \mu \leq 31 \text{ cm} \quad \text{versus} \quad H_1 : \mu > 31 \text{ cm}$$

with $\alpha = 0.05$

Frequentist approach

- the researchers measure $n = 12$ fish samples and measure $\overline{y} = 32$ cm
  - we build the normalized variable

$$z = \frac{\overline{y} - 31}{\sigma/\sqrt{n}} = \frac{32 - 31}{2/\sqrt{12}} = 1.732$$

  - we compute the P_{value}:

$$P_{value} = P\left(z > \frac{32 - 31}{2/\sqrt{12}}\right) = P(z > 1/732) = 0.04163678$$

  - the value is less that the level of significance, so the NULL hypothesis is rejected

Example: One-Side Hypothesis Test (B)

Bayesian approach

- the researchers measure $n = 12$ fish samples and compute $\overline{y} = 32$ cm
  - building the normalized variable

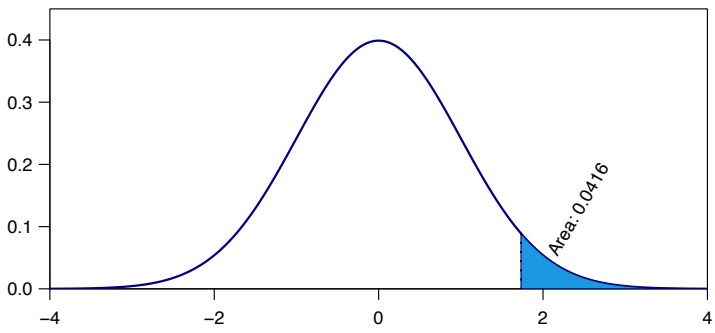
$$z = \frac{\overline{y} - 31}{\sigma/\sqrt{n}} = \frac{32 - 31}{2/\sqrt{12}} = 1.732$$

  - we compute the P_{value}:

$$P_{value} = P\left(z > \frac{32 - 31}{2/\sqrt{12}}\right) = P(z > 1/732) = 0.0416$$

  - the value is less that the level of significance, so the NULL hypothesis is rejected

```
y_ave <- 32
sigma <- 2
n <- 12
z <- (y_ave - 31)/(sigma/sqrt(n))
pnorm(z, lower.tail=FALSE)
[1] 0.04163226
```



Comparing μ of two Normal distributions

equal known variance

- samples and priors are independent ➜ the posteriors are independent

$$P\left(\mu_1 \,\middle|\, y_1 \dots y_{n1}\right) = \text{Norm}(m_1', (s_1')^2)$$

$$P\left(\mu_2 \,\middle|\, z_1 \dots z_{n2}\right) = \text{Norm}(m_2', (s_2')^2)$$

  with m_1', s_1' and m_2', s_2' determined by Normal μ inferences

- since both samples are independent, we can easily build the posterior distribution for $\mu_d = \mu_1 - \mu_2$:

$$P\left(\mu_d \,\middle|\, y_1 \dots y_{n1},\ z_1 \dots z_{n2}\right) = \text{Norm}(m_d', (s_d')^2)$$

- where $m_d' = m_1' - m_2'$ and $(s_d')^2 = (s_1')^2 + (s_2')^2$
- the $(1-\alpha)\times 100\%$ bayesian credible interval for $\mu_d = \mu_1 - \mu_2$ is:

$$m_d' \pm z_{\alpha/2} \times s_d'$$

  and can be written as:

$$m_1' - m_2' \pm z_{\alpha/2} \times \sqrt{(s_1')^2 + (s_2')^2}$$

Example: measuring the speed of light

The problem

- Michelson made two series of measurements in 1879 and 1882, respectively

Michelson (1879)				Michelson (1882)			
850	740	900	1070	883	816	778	796
930	850	950	980	682	711	611	599
980	880	1000	980	1051	781	578	796
930	650	760	810	774	820	772	696
1000	1000	960	960	573	748	748	797
				851	809	723	

- let's suppose the measurements are normally distributed with a known standard deviation, $\sigma = 100$
- we use independent priors, Norm$(m = 3 \cdot 10^5, s^2 = 500^2)$
- we compute the posteriors for μ_{1879} and μ_{1882} using the conjugate prior formulas
- we get:

$$\text{for} \quad \mu_{1879} : m_{1879}' = 299909 \text{ and } \quad (s1879')^2 = 499$$

$$\text{for} \quad \mu_{1882} : m_{1882}' = 299757 \text{ and } \quad (s1882')^2 = 434$$

Example: measuring the speed of light (2)

- the posterior distribution for $\mu_d = \mu_{1879} - \mu_{1882}$ will be Normal$(m'_d, (s'_d)^2)$, with

$$m'_d = 2999909 - 299757 = 152$$

and

$$(s'_d)^2 = 499 + 434 = 933 \sim 30.5^2$$

- the 95% Bayesian credible interval for $\mu_d = \mu_{1879} - \mu_{1882}$ is:

$$152 \pm 1.96 \times 30.5 = (92.1, 211.9)$$

- we perform an hypothesis test on the difference:

$$H_\circ : \mu_d \leq 0 \quad \text{versus} \quad H_1 : \mu_d > 0$$

- we compute the posterior probability of the null hypothesis $P(\mu_d < 0 \mid data)$:

$$\begin{aligned} P\left(\mu_d < 0 \mid data\right) &= P\left(\frac{\mu_d - m'_d}{s'_d} \leq \frac{0 - m'_d}{s'_d}\right) \\ &= P\left(z \leq \frac{0 - m'_d}{s'_d}\right) \end{aligned}$$

- since 0 lies outside the Bayesian credible interval, we reject the null hypothesis
  ➜ we conclude that the two sets of measurements are different

Comparing μ of two Normal distributions

variance unknown and flat priors are used

- we use independent flat priors for μ_1 and μ_2
- we get $m'_1 = \overline{y}$, $s'_1 = \sigma/\sqrt{n_1}$ and $m'_2 = \overline{z}$, $s'_2 = \sigma/\sqrt{n_2}$ and

- since we do not know the variance, we have to estimate it from the data:

$$\hat{\sigma}^2 = \frac{\sum_{j=1}^{n_1}(y_j - \overline{y})^2 + \sum_{k=1}^{n_2}(z_k - \overline{z})^2}{n_1 + n_2 - 2}$$

- since we used an estimate of the unknown true variance, the credible interval should be widened to allow for the additional uncertainty
- the $(1-\alpha) \times 100\%$ Bayesian credible interval for $\mu_1 - \mu_2$ is:

$$\overline{y} - \overline{z} \pm t_{\alpha/2} \times \hat{\sigma} \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

where $t_{\alpha/2}$ comes from a Sudent's t distribution with $n_1 + n_2 - 1$ degrees of freedom

Infering the difference of two proportions with Normal approximation

- want to compare the proportion of a certain attribute of two populations
- let's assume the true proportions in population 1 and 2 are π_1 and π_2
- y_1 and y_2 are the observation, from each population, having sampled n_1 and n_2 objects
- the distributions $P(y_1 \mid \pi_1)$ and $P(y_2 \mid \pi_2)$ are both binomial and independent
- letting the priors of π_1 and π_2 be beta distributed:

$$\mathrm{Beta}(a_1, b_1) \quad \text{and} \quad \mathrm{Beta}(a_2, b_2)$$

- the posteriors are beta distributed:

$$\mathrm{Beta}(a_1', b_1') \quad \text{with} \quad a_1' = a_1 + y_1 \quad \text{and} \quad b_1' = b_1 + n_1 - y_1$$

$$\mathrm{Beta}(a_2', b_2') \quad \text{with} \quad a_2' = a_2 + y_2 \quad \text{and} \quad b_2' = b_2 + n_2 - y_2$$

approximating the posterior with a Normal distribution, the posterior of the difference $\pi_d = \pi_1 - \pi_2$ is approximately normal with mean

$$m_d' = \frac{a_1'}{a_1' + b_1'} - \frac{a_2'}{a_2' + b_2'}$$

and variance

$$(s_d')^2 = \frac{a_1' b_1'}{(a_1' + b_1')^2 (a_1' + b_1' + 1)} + \frac{a_2' b_2'}{(a_2' + b_2')^2 (a_2' + b_2' + 1)}$$

Example: smoking habits in students (1)

Problem

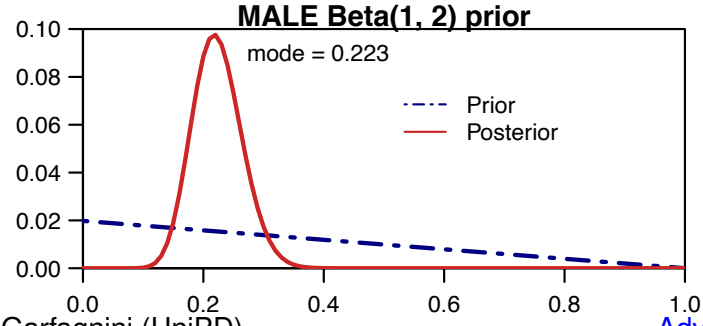
- we want to investigate the smoking habits between students
- 200 students are interviewed (100 male and 100 female)
- 22 male students claim they are regular smokers
- 31 female students declare that they are regular smokers
- is there a difference between the two populations ?

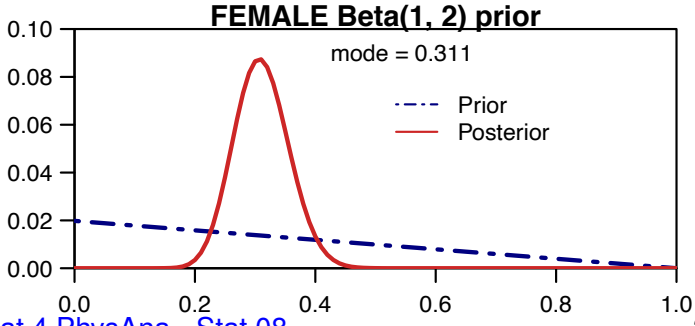
Solution

- Miki analyzes the data and she assumes a $\mathrm{Beta}(1,2)$ prior for both π_{male} and π_{female}. The posteriors are:

$$\mathrm{Beta}_{\mathrm{male}}(a', b') \quad \text{with} \quad a' = a + y_{male} = 23 \quad \text{and} \quad b' = b + n_{male} - y_{male} = 80$$

$$\mathrm{Beta}_{\mathrm{female}}(a', b') \quad \text{with} \quad a' = a + y_{female} = 32 \quad \text{and} \quad b' = b + n_{female} - y_{female} = 71$$





Example: smoking habits in students (2)

- we assume the difference between the proportions

$$\pi_d = \pi_{male} - \pi_{female}$$

is approximately normal

$$\mathrm{Norm}(m_d, s_d^2)$$

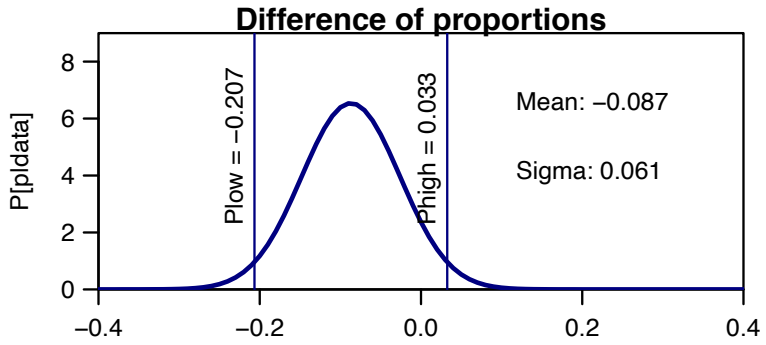
with

$$m_d = \frac{a_1'}{a_1' + b_1'} - \frac{a_2'}{a_2' + b_2'} = \frac{23}{23 + 80} - \frac{32}{32 + 71} = -0.087$$

and

$$(s_d)^2 = \frac{a_1' b_1'}{(a_1' + b_1')^2 (a_1' + b_1' + 1)} + \frac{a_2' b_2'}{(a_2' + b_2')^2 (a_2' + b_2' + 1)} = 0.061^2$$

- The 95% Credibility interval is $(-0.207, 0.033)$ and it contains 0. Therefore we cannot reject the null hypotheis $H_\circ : \pi_{male} - \pi_{female} = 0$



Markov Chain Monte Carlo - part I

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Stat Lect. 9

Integration in Bayesian inference

- there are four cases, in Bayesian inference, that require integration:

Marginalization

- given a two dimensional (or higher) posterior pdf over the parameters (θ_1, θ_2), we can determine the posterior over just one parameter by integration

$$P\left(\theta_1 \mid D, M\right) = \int P\left(\theta_1, \theta_2 \mid D, M\right) d\theta_2$$

Expectation values

- this is defined as

$$E\left[\theta\right] = \int \theta P\left(\theta \mid D, M\right) d\theta$$

- if the posterior pdf, $P\left(\theta \mid D, M\right)$ is normalized, or by

$$E\left[\theta\right] = \frac{1}{Z^*} \int \theta P^*\left(\theta \mid D, M\right) d\theta$$

- with $Z^* = \int P^*\left(\theta \mid D, M\right) d\theta$ when $P^*\left(\theta \mid D, M\right)$ is the unnormalized posterior pdf distribution

Integration in Bayesian inference

Model Comparison

- for comparing models, we need to evaluate the *evidence* of Bayesian theorem

$$P\left(D \mid M\right) = \int P\left(D \mid \theta, M\right) P\left(\theta \mid M\right) d\theta$$

Data prediction

- given a data set $D = \{y_j\}$ obtained at fixed $\{x_j\}$, we have determined the posterior pdf over the model parameters. Now, we are looking for the prediction y_p over a new point x_p
- the Bayesian approach is to find the posterior pdf over y_p, i.e.

$$P\left(y_p \mid x_p, D, M\right) \text{ , a posterior predicative distribution}$$

$$\begin{aligned} P\left(y_p \mid x_p, D, M\right) &= \int P\left(y_p \theta \mid x_p, D, M\right) d\theta \\ &= \int P\left(y_p \mid x_p, \theta D, M\right) P\left(\theta \mid x_p, D, M\right) d\theta \\ &= \int P\left(y_p \mid x_p, \theta M\right) P\left(\theta \mid D, M\right) d\theta \end{aligned}$$

- notice that $P\left(y_p \mid x_p, \theta D, M\right) = P\left(y_p \mid x_p, \theta M\right)$ since it is independent of our data set D, once we have determined the model parameters
- in a similar way, $P\left(\theta \mid x_p, D, M\right) = P\left(\theta \mid DM\right)$ because our knowledge of the model parameters does not depend on where we want to make a prediction

How to compute the posterior distribution

1) The easy way

- select a Prior distribution function which is conjugate to the Likelihood function:

  Examples:

Prior	Likelihood
Beta	Bernoulli / Binomial
Gamma	Poisson
Beta	Geometric
Normal	Normal (with known σ^2)
Inverse Gamma	Normal (with known μ)

2) The brute force approach

- define the Prior on a dense grid of points spacing the range of your parameter θ
- compute the Posterior numerically by summing the product Likelihood $\times$ Prior on the grid

1) works well in a very limited number of cases
2) has severe limitations (memory, computation time) for multiparameter spaces

The Markov Chain Monte Carlo

- is a very powerful, generic method, for *approximately* generating samples from any Posterior distribution
- the Prior distribution, $P(\theta)$, is specified by a function that can be easily evaluated (analytically or numerically)
- the Likelihood function, $P(D \mid \theta)$, can be computed for any values of D and θ
- the method demands that Prior and Likelihood can be computed up to a multiplicative constant ➜ it is not required to compute the Evidence (i.e. the denominator of the Bayes' theorem)
- an approximation of the Posterior distribution, $P(\theta \mid D)$, is produced
- since the Posterior distribution is estimated by randomly generating a large samples from it, it is called a Monte Carlo method (by analogy to 'standard' Monte Carlo methods)

The Island example

- 10 islands of different size form an archipelago
- the number of people living in each island is proportional to its area
- a doctor is continuosly traveling among the islands and she wants to remain in an island for a time proportional to that island's population
- at the beginning of each week, the doctor can

1) stay on the current island
2) move to an adjacent island

- to simplify the problem,
  - we label the island from 1 to 10 and place them on a circle
  - The number of inhabitants is equal to the island label (in some arbitrary unit)

The Island example : the algorithm

- at the beginning of each week, the doctor
  - flips a coin to decide on which island she can go: HEAD ➜ East, TAIL ➜ West
  - if the proposed island has a larger population with respect to her current position, she goes to that island
  - if the proposed island has a smaller population, the probability of moving there is proportional to the ratio of populations

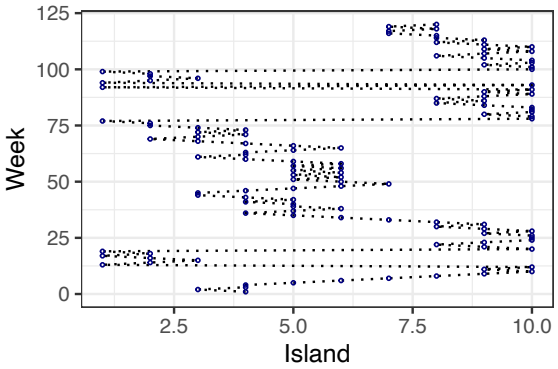
$$P_{\text{proposed}}/P_{\text{current}}$$

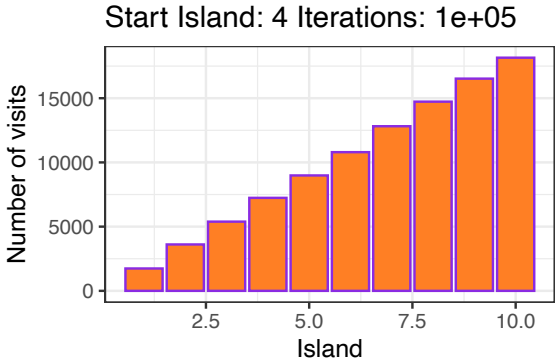
- the moving probability is

$$P_{move} = \text{MIN}\left(\frac{P\left(\theta_{proposed}\right)}{P\left(\theta_{current}\right)}, 1\right)$$

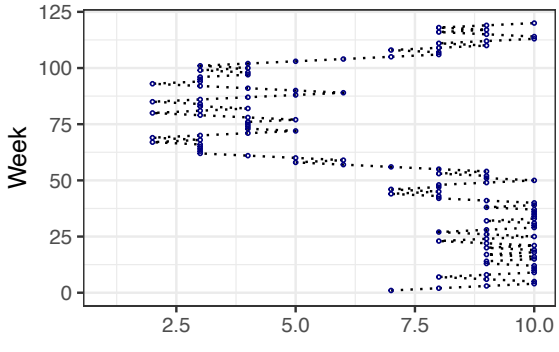
The Island example : test runs

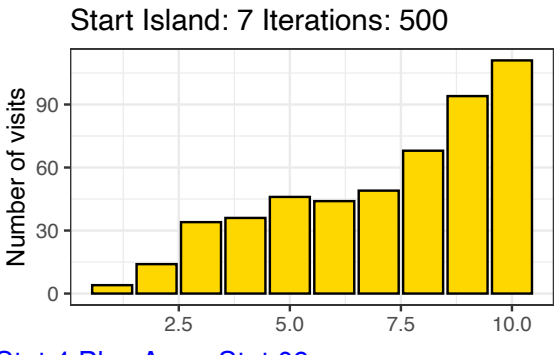
- a long run (10^5 weeks) has been performed:
  - adjacent islands are visited proportionally to their population size (i.e. target distribution)





- a shorter run (500 weeks) has been done:
  - the obtained distributions is a bad approximation of the target distribution





Markov Chain Monte Carlo

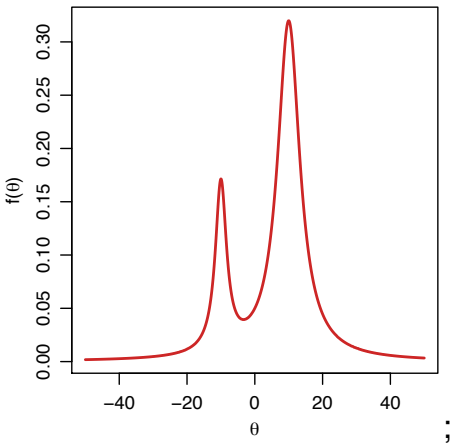
- is a very powerful, generic method, for *approximately* generating samples from any arbitrary distribution
- the MCMC method is due to Metropolis et al [1] and was motivated by computational methods in statistical physics
- it uses the idea of generating a Markov chain whose limiting distribution is equal to to desired target distribution
- many modifications and enhancement were proposed, most notably the one of Hastings [2]
- today, any approach the produces an ergodic Markov chain whose stationary distribution is the target distribution is referred to as MCMC or Markov chain sampling
- the most prominent MCMC algorithms are the Metropolis-Hastings and the Gibbs sampler

[1] N. Metropolis, et al., *Equations of state calculations by fast computing machines*, J Chem Phys, **21**, 1087, 1953

[2] W.K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57, 92, 1970

Markov Chain Monte Carlo

- let's assume we want to sample from a complex distribution $f(\theta)$
- the 'standard' Monte Carlo methods we have discussed would not be very efficient since they would 'waste time' in sampling $f(\theta)$ in regions where the value is small
- we would like to make samples in the region where $f(\theta)$ is high, but keeping the full sample still representative of $f(\theta)$
- this can be done if we relax the constraint of drawing samples independently



;

- the principle behind a Markov Chain Monte Carlo is to setup a random walk over the parameter space which explores the regions of high probability density of $f(\theta)$
- the random walk is done through a Markov Chain:

  a random process in which the probability of evolving from a state $\theta_t \rightarrow \theta_{t+1}$ is defined by a transition probability $Q(\theta_{t+1} \mid \theta_t)$ which does not depend on the previous states
- this is also called a memory-less process

Markov Chain Monte Carlo Algorithm

- as with the rejection-sampling, the MCMC uses a proposal distribution $Q(s|\theta)$
- it is a distribution from which we can easily draw a candidate sample s for the next point in the chain, θ_{t+1}, given the current parameter value θ_t

Algorithm

(0) initialize the chain at some value

(1) draw a random sample from the distribution $Q(s|\theta)$

This is often a multivariate Gaussian where θ_t is the mean and the covariance matrix specifies the the typical size of steps in the chain in each dimension of the parameters θ

(2) decide whether to accept or not the new candidate sample on the basis of the Metropolis ratio

$$\rho = \frac{f(s)}{f(\theta_t)} \frac{Q(\theta_t \mid s)}{Q(s \mid \theta_t)}$$

if $\rho \geq 1$ the new candidate is accepted and $\theta_{t+1} = s$

if $\rho < 1$ we only accept it with probability ρ:

▷ draw $u \sim \mathcal{U}(0,1)$ and set $\theta_{t+1} = s$ only if $u \leq \rho$

if s is not accepted, we set $\theta_{t+1} = \theta_t$, i.e. the existing sample in the chain is repeated

Markov Chain Monte Carlo Algorithm

- the algorithm goes on for a certain number of iterations
- the typical number of steps required to have a good sampling depends on the problem. Typical values are between 10^4 and 10^6
- if a symmetric proposal distribution function is used (like a Gaussian) the term $Q(\theta_t \mid s)/Q(s \mid \theta_t)$ in the definition of ρ is always unity
- this is referred to as the Metropolis algorithm
- depending on the initialization of the chain, the initial samples may not be representative of it and they should be discarded
- the discarded initial samples are called the burn-in
- with a good initialization the burn-in may only be a few percent of the chain

MCMC Exercise 1

MCMC sampling : 1-dim

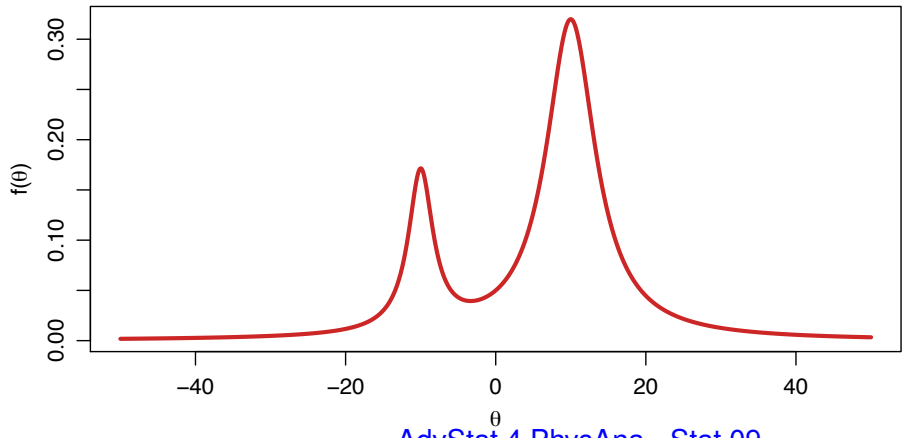
- given a Cauchy distribution function

$$\text{Cauchy}(x \mid x_\circ, \gamma) = \frac{1}{\pi\gamma} \frac{1}{1 + ((x - x_\circ)/\gamma)^2}$$

- where γ and $x_\circ$ are called scale and location parameters, let's consider

$$f(x) = \text{Cauchy}(x_\circ = -10, \gamma = 2) + 4 * \text{Cauchy}(x_\circ = 10, \gamma = 4)$$

- we want to sample from the distribution using a MCMC algorithm



MCMC Metropolis R code (1dim functions)

```
# Parameters:
# func : a function whose first argument is a real vector of parameters
#        func returns a log10 of the likelihood function
# theta.init : the initial value of the Markov Chain (and of func)
# n.sample: number of required samples
# sigma : standar deviation of the gaussian MCMC sampling pdf
metropolis.1dim <- function(func, theta.init, n.sample, sigma) {
  theta.cur <- theta.init
  func.Cur <- func(theta.cur)
  func.Samp <- matrix(data=NA, nrow=n.sample, ncol=2+1)
  n.accept <- 0
  rate.accept <- 0.0

  for (n in 1:n.sample) {

    theta.prop <- rnorm(n=1, mean = theta.cur, sigma)
    func.Prop <- func(theta.prop)
    logMR <- func.Prop - func.Cur  # Log10 of the Metropolis ratio

    if ( logMR>=0 || logMR>log10(runif(1)) ) {
       theta.cur <- theta.prop
       func.Cur  <- func.Prop
       n.accept <- n.accept + 1
    }
    func.Samp[n, 1] <- func.Cur
    func.Samp[n, 2] <- theta.cur
  }
  return(func.Samp)
}
```

MCMC Metropolis R code (1dim functions)

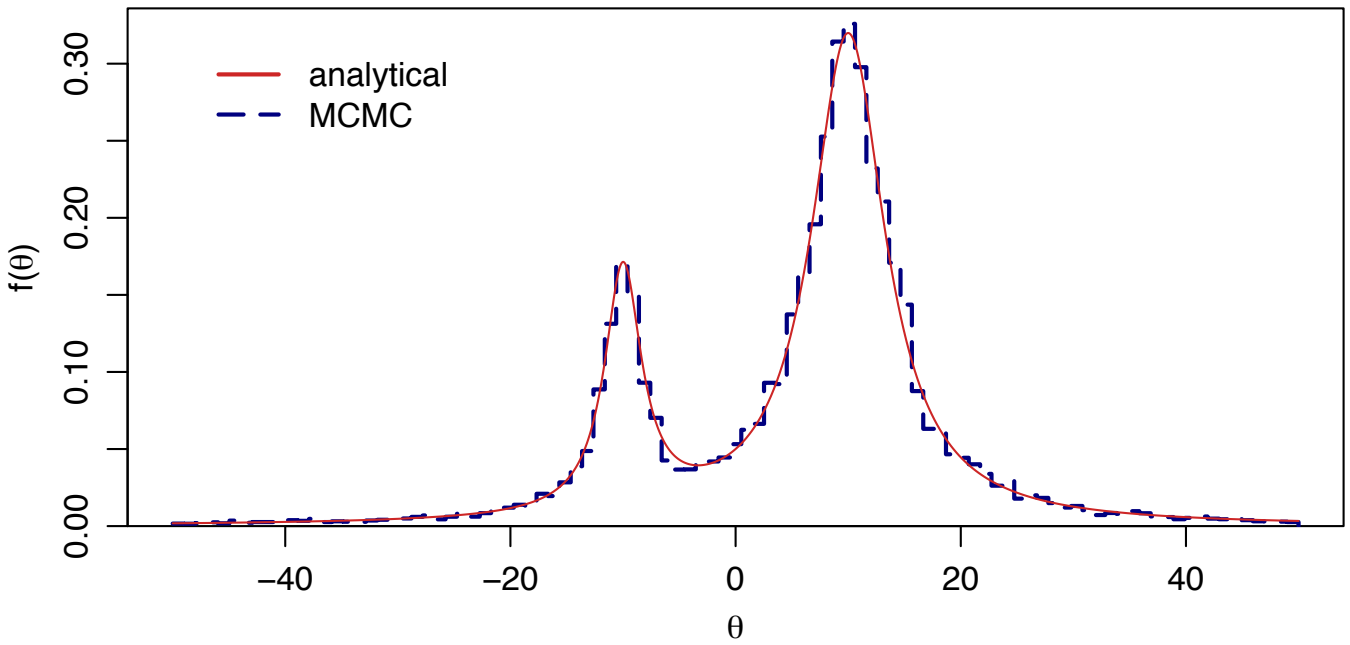
```
#
# Our test function
#
testfunc <- function(theta) {
   return(dcauchy(theta, -10, 2,) + 4*dcauchy(theta, 10, 4))
}

#
# - interface for the metropolis function, gets the log10 of test function
testfunc.metropolis <- function(theta) {
  return(log10(testfunc(theta)))
}

### Running parameters
theta.init <- -5
sample.sig <- 10
n.sample   <- 10^5
demo       <- TRUE

set.seed(20190513)
chain <- metropolis.1dim(func=testfunc.metropolis,
                         theta.init = theta.init,
                         n.sample = n.sample,
                         sigma = sample.sig^2, demo)
```


MCMC Metropolis R code results

```
#
# Here are the plots
#
par(mfrow=c(2,2), mgp=c(2,0.8,0), mar=c(3.5,3.5,1,1), oma=0.1*c(1,1,1,1))

x <- seq(-50, 50, length.out=10^4)
y <- testfunc(x)
ymax <- 1.05 * max(y)
plot(x, y, ylim=c(0,max(y)*1.10),
     type='l', lwd=2, col='firebrick3',
     xlab=expression(theta), ylab=expression(paste('f(',theta,')', sep='')))

plot(x, y, type="n", yaxs="i", ylim=c(0, 1.05*max(y)),
     xlab=expression(theta), ylab=expression(paste('f(',theta,')', sep='')))
sa <- which(chain$func.Samp[,2]>=min(x) & chain$func.Samp[,2]<=max(x))
hist <- hist(chain$func.Samp[sa,2], breaks=seq(from=min(x), to=max(x),
                                    length.out=100), plot=FALSE)
Zhist <- sum(hist$counts)*diff(range(hist$breaks))/(length(hist$counts))
lines(hist$breaks, c(hist$counts*Zfunc/Zhist,0),
      col='navy', type="s", lwd=2, lty=5)
lines(x, y, col='firebrick3', lwd=1, lty=1)

leg.labels = c('analytical', 'MCMC')
leg.ltype = c(1, 5)
leg.colors = c('firebrick3','navy')
legend("topleft", inset=.05, bty='n',
       legend = leg.labels, lty=leg.ltype, col=leg.colors,
       lwd = 2)
```

MCMC plot results

- the histogram reproduces the behavior of our test function
- the acceptance rate of the samples is only 15.84%
- by changing the σ of the proposal distribution function, we get a better rate (40.81% for $\sigma = 5$)



MCMC chain analysis

- the proposed algorithm works in principle, but it may not produce a representative sample ➜ it is important to inspect the chain and check its property
- open points in the recipe are: the covariance matrix of the proposal distribution, how long should the burn-in period be, how many iterations are expected before convergence, etc.
- one of the simplest ways to check whether the chain has reached a steady state is to rerun the sampling several times, with different starting points ➜ all chains should converge to the same region of parameter space
- various metrics exist. One way is to compute an auto-correlation function of the elements of the chain:
- given a chain of length N, at lag h, from the definition of covariance it follows:

$$\mathrm{ACF}(h) = \frac{\frac{1}{N-h}\sum_{t=1}^{N-h}(\theta_t - \overline{\theta})(\theta_{t+h} - \overline{\theta})}{\frac{1}{N-1}\sum_{t=1}^{N}(\theta_t - \overline{\theta})^2}$$

- where θ_{t+h} is the chain offset by h steps

➜ $\mathrm{ACF}(h)$ measures how closely the chain is correlated with itself h steps later

MCMC chain analysis : the coda R package

CODA

- provides functions for summarizing and plotting the output from Markov Chain Monte Carlo (MCMC) simulations, as well as diagnostic tests of convergence to the equilibrium distribution of the Markov chain

➜ https://cran.r-project.org/web/packages/coda/coda.pdf

- the function mcmc and as.mcmc are used to create a Markov Chain Monte Carlo object, that can be digested by the CODA methods and functions. The input data are taken to be a vector, or a matrix with one column per variable

- useful functions are

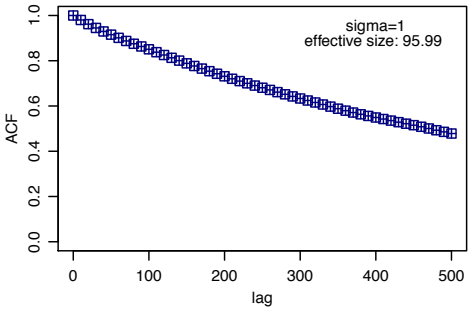
▷ autocorr() : calculates the auto-correlation function for the Markov chain object at the lags given by parameter lags. High auto-correlations within chains indicate slow mixing and, usually, slow convergence

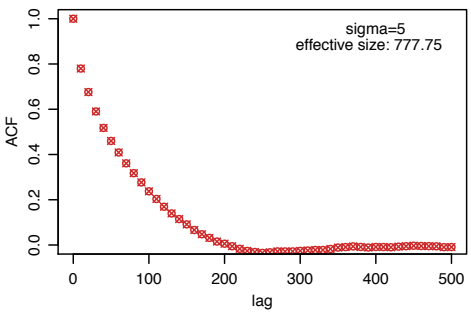
▷ effectiveSize() : computes the sample size adjusted for auto-correlation

MCMC chain analysis example

- we run our previous example changing the σ parameter of the proposal distribution function and analyze the MCMC chain

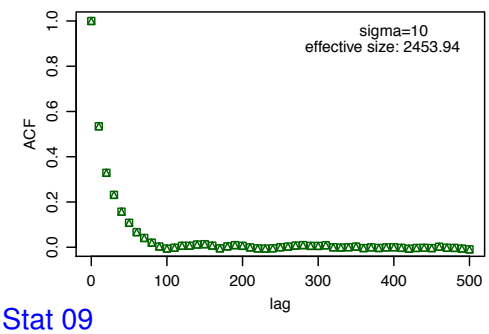
```
library(coda)
c.chain1 <- as.mcmc(chain.r1$func.Samp[,2])

my.lags = seq(0,500,10)
y1 <- autocorr(c.chain1, lags=my.lags)

plot(my.lags, y1, ylim=c(0,1),
     pch=12, col='navy',
     xlab='lag', ylab='ACF', cex=1.3)
text(400,0.9, paste('sigma=1'))
text(400,0.85,
     sprintf("effective size: %.2f",
             effectiveSize(c.chain1)))
```





- the first sample is strongly correlated

σ	$R_{acceptance}$	N_{eff}
1	0.9256	95.99
5	0.4127	777.75
10	0.1585	2453.94



MCMC and parameter transformation

- sometimes it is more efficient to sample over a transformed parameter
- let's consider, as an example, a parameter $\theta > 0$; we could sample $\ln\theta$ since it ensures the parameter θ cannot be negative
- but this means drawing from $P(\ln\theta)$ and not from $P(\theta)$. Since

$$P(\theta)d\theta = P(\ln\theta)d(\ln\theta) \Rightarrow P(\ln\theta) = \theta P(\theta)$$

- when we are using a symmetric proposal distribution

$$Q(\theta_t \mid s) = Q(s \mid \theta_t)$$

- the Metropolis ratio becomes

$$\rho = \frac{sP(s)}{\theta_t P(\theta_t)}$$

- the base of the logarithm in the transformation is irrelevant, since it corresponds to a constant factor that cancels in the ratio
- in general, for a transformation from $(\theta_1, \ldots, \theta_J)$ to $(\phi_1, \ldots, \phi_J)$ we need the Jacobian determinant of the original parameters versus the transformed ones

$$\mathcal{J}_\theta = \left| \frac{\partial(\theta_1, \ldots, \theta_J)}{\partial(\phi_1, \ldots, \phi_J)} \right|$$

- and the Metropolis ratio becomes

$$\rho = \frac{P(s)}{P(\theta_t)} \frac{\mathcal{J}_s}{\mathcal{J}_\theta}$$

Parameter estimation with MCMC

- we will show how to sample posteriors with more than two parameters using MCMC
- as an example we will consider a fit to data, both linear and quadratic, whereby we will infer also the noise on the data

The problem requirements

- we have a 2-dim set of N points $\{x_j; y_j\}$
- the model M predicts:
  - $y = f(x) + \epsilon$
  - where $f(x) = b_0 + b_1 \cdot x$
- $f(x)$ is the generative model, it gives noise-free prediction of the data, given the parameters
- the residuals $\epsilon = y - f(x)$ are modeled as a zero-mean Gaussian function with standard deviation σ. This is the noise model
- assuming $\{x_j\}$ are noise-free, and $\theta = (b_0, b_1, \sigma)$, the likelihood is

$$P\left(y_j \mid x_j, \theta, M\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{\left(y_j - f(x_j; b_0, b_1)\right)^2}{2\sigma^2} \right]$$

Parameter estimation with MCMC

- we try to infere σ from the data
- although the $\{x_j\}$ values are supplied with the data, they are assumed to be fixed and not described by a measurement model. Therefore $D = \{y_j\}$
- assuming data points are independent, the log-likelihood for all the data points is

$$\ln P\left(\{y_j\} \,\middle|\, \{x_j\}, \theta, M\right) = \sum_{j=1}^{N} \ln P\left(y_j \,\middle|\, x_j, \theta, M\right)$$

- in general, none of the parameters is known in advance and we want to infer the posterior from the data

$$P\left(\theta \,\middle|\, D\right) \propto P\left(D \,\middle|\, \theta\right) \times P\left(\theta\right)$$

- given the data, D, the procedure to compute the posterior is as follows:

(1) define the prior pdf for the parameters. Use reasonable and plausible priors and make use, if needed, of variable transformation

(2) define the covariance matrix of the proposal distribution. (a diagonal, multivariate Gaussian distribution)

(3) define the starting point of the MCMC

(4) define the number of burn-in and sampling interactions

Parameter estimation with MCMC

- once the MCMC data have been collected, perform the following analysis

(5) make the chains thinner

(6) plot the chains and the one one-dimensional marginal posterior pdf over the parameters

(7) plot the two-dimensional posterior distributions of all three parameters, simply by plotting the samples, and look for correlations between the parameters

(8) calculate the maximum a posteriori values of the model parameters from the MCMC chains, calculate and plot the resulting model, and compare to the original data

(9) calculate the predictive posterior distribution over y at new data points

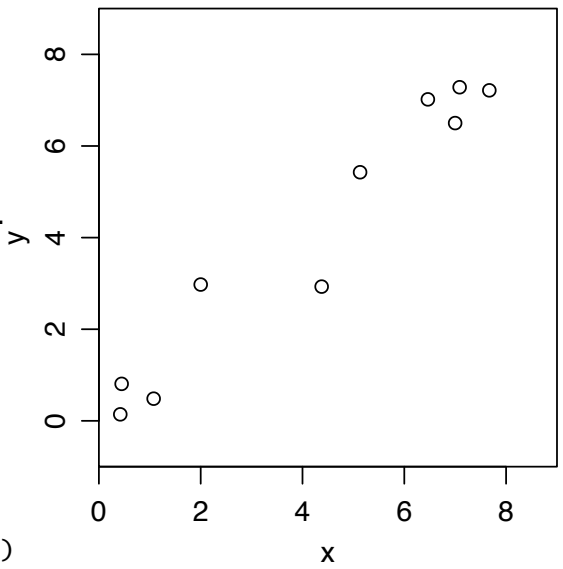
- since we have samples drawn from the posterior, we don't need the actual values of the posterior density in order to plot the posteriors. For the same reason, we don't have to perform any integration to get the one-dimensional marginal distributions

Parameter Priors

- for the intercept, b_0 : $P(b_0) = \mathrm{N}(\mu, \sigma)$, a Gaussian with mean μ and standard deviation σ
- for the gradient, b_1 : we can write it as $b_1 = \tan\alpha$, where α is the angle, in radians, between the horizontal and the model line. Since we have no prior knowledge of the slope, we should use a uniform distribution $P(\alpha) = 1/2\pi$
- standard deviation, σ : in the absence of any other information, a scale parameter such as the standard deviation of a Gaussian should be assigned a Jeffreys prior, $P(\sigma) \propto \log\sigma$. This also prevents σ from becoming negative.
- given these priors, the model parameters are now $(b_0, \alpha, \log\sigma)$. These are the parameters that the Monte Carlo algorithm will sample over. The prior distributions are likewise defined over the parameters, as Gaussian (b_0), uniform (α), and uniform ($\log\sigma$, respectively

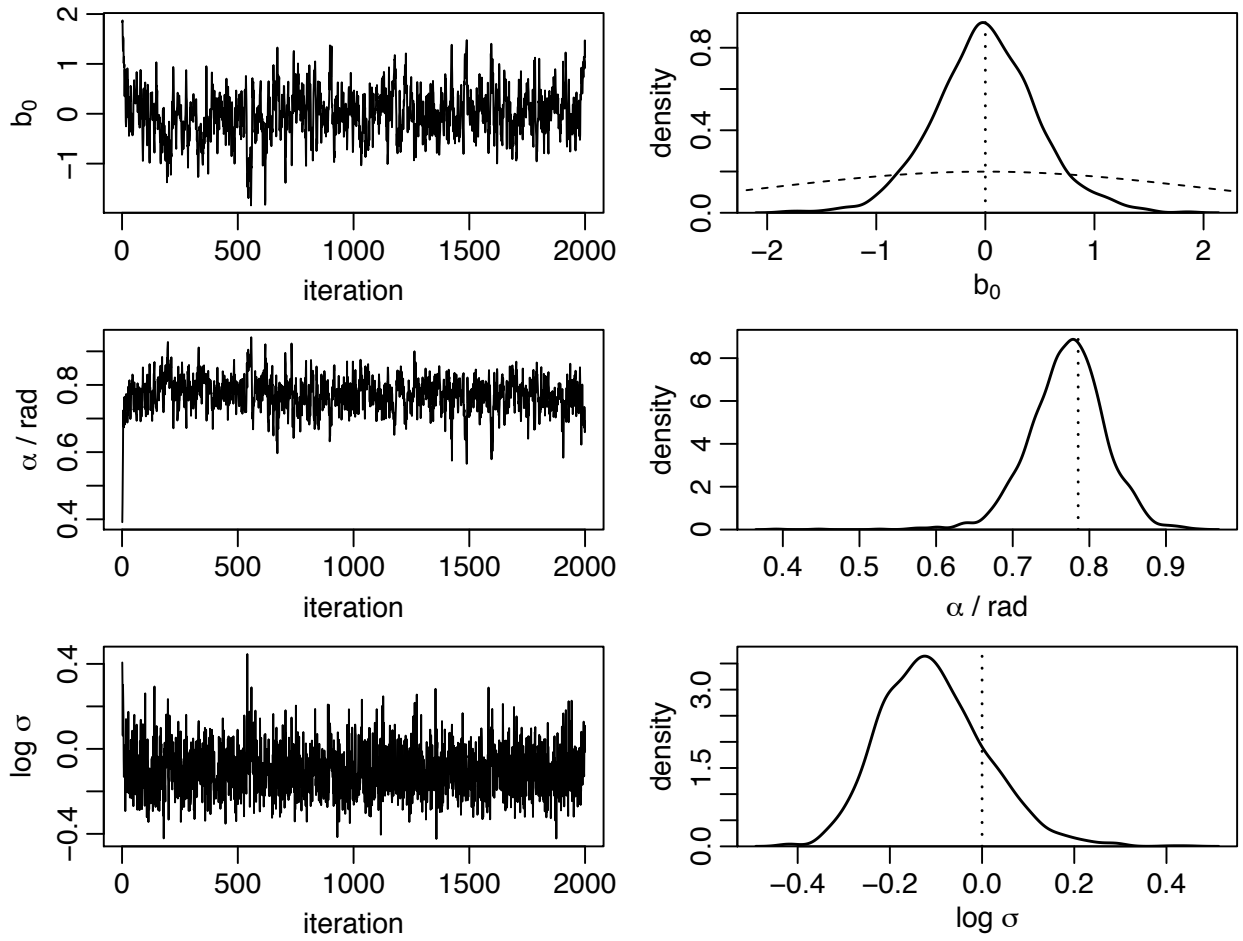
Example: the data

- these are the 10 data points we want to fit to our model
- they have been drawn at fixed x values from a straight line with $b_0 = 0$ and $b_1 = 1$, to which zero mean Gaussian noise with $\sigma = 1$ has been added.




```
Ndat <- 10
x <- sort(runif(Ndat, 0, 10))
sigTrue <- 1

# 1 x P vector: coefficients,
# b_p, of sum_{p=0} b_p*x^p
modMat <- c(0,1)
y <- cbind(1,x) %*% as.matrix(modMat) +
                    rnorm(Ndat, 0, sigTrue)
# Dimensions in the above:
#    [Ndat x 1] = [Ndat x P] %*% [P x 1] + [Ndat]
# cbind does the logical thing when combining
# a scalar and vector, then do vector addition

# finally, convert to a vector
y <- drop(y)
```

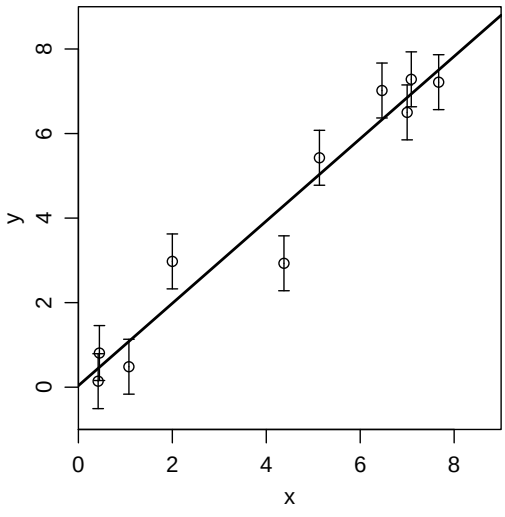
Example: the MCMC

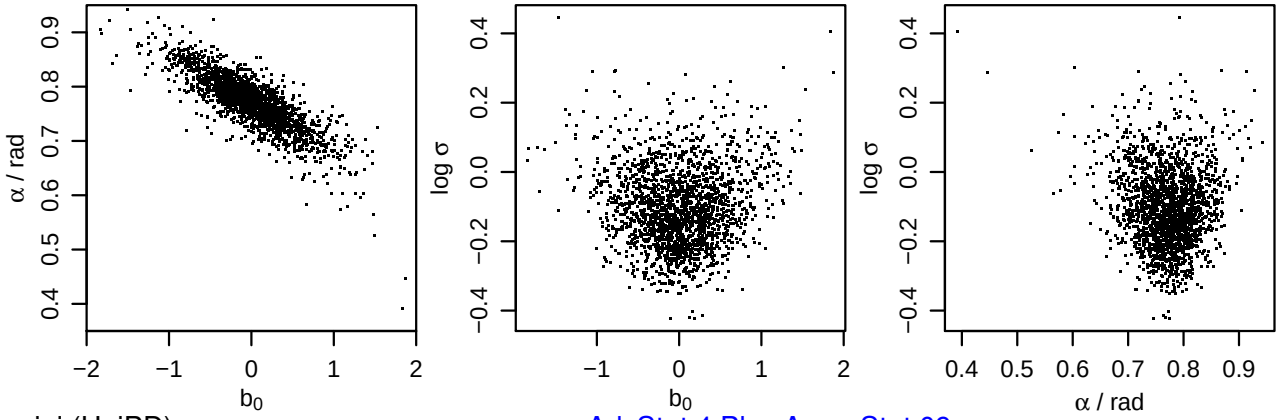


Example: fit and parameters correlations

- the mean of the posterior is $(b_0, \alpha, \log\sigma) = (0.0042, 0.77, -0.11)$
- the covariance of the posterior pdf is

	b_0	α	$\log\sigma$
b_0	0.48		
α	-0.83	0.050	
$\log\sigma$	0.038	-0.073	0.11





R code for the next examples

- the examples discussed in the following are taken from

  Coryn A L. Bailer-Jones, *Practical Bayesian Inference*, Cambridge University Press, 2017, ISBN 978-1-316-64221-4

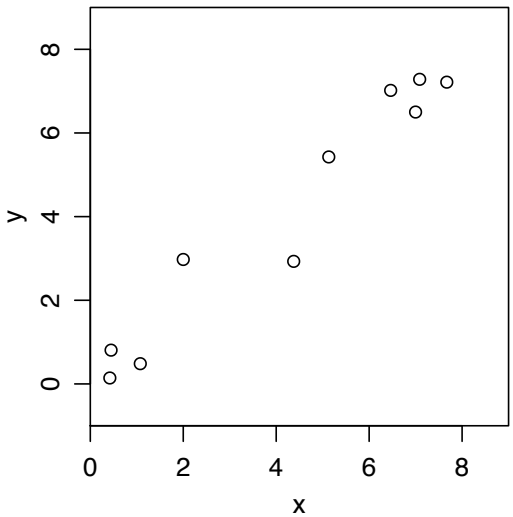
- the R code of the book can be dowloaded from
- https://github.com/ehalley/PBI/tree/master/PBI_scripts:

  - metropolis algorithm:
    https://github.com/ehalley/PBI/blob/master/PBI_scripts/metropolis.R
  - Linear model example main code:
    https://github.com/ehalley/PBI/blob/master/PBI_scripts/linearmodel_posterior.R
  - Linear Model Likelihood, Prior and Posterior probabilities:
    https://github.com/ehalley/PBI/blob/master/PBI_scripts/linearmodel_functions.R
  - quadratic model example main code:
    https://github.com/ehalley/PBI/blob/master/PBI_scripts/quadraticmodel_posterior.R
  - quadratic Model Likelihood, Prior and Posterior probabilities:
    https://github.com/ehalley/PBI/blob/master/PBI_scripts/quadraticmodel_functions.R

Fitting a straight line with noise

The Prior

- the Priors on α and $\log\sigma$ have no parameters
- the Prior on the intercept is driven by the data. A Gaussian distribution is assumed with $\mu = 0$ and a standard deviation $\sigma = 2$



R code

```
#
# parameters:
#   theta[1] -> b_0
#   theta[2] -> alpha
#   theta[3] -> log(sigma)

logprior.linearmodel <- function(theta) {
  b0Prior      <- dnorm(theta[1], mean=0, sd=2)
  alphaPrior   <- 1
  logysigPrior <- 1
  logPrior <- sum( log10(b0Prior),
                   log10(alphaPrior),
                   log10(logysigPrior) )
  return(logPrior)
}
```

Fitting a straight line with noise

The Likelihood

- the logLikelihood is

$$P\left(y_j \mid x_j, \theta, M\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{\left(y_j - f(x_j; b_0, b_1)\right)^2}{2\sigma^2}\right]$$

R code

```
# parameters:
#   theta[1] -> b_0
#   theta[2] -> alpha
#   theta[3] -> log(sigma)

loglike.linearmodel <- function(theta, obsdata) {
  # convert alpha to b_1 and log10(ysig) to ysig
  theta[2] <- tan(theta[2])
  theta[3] <- 10^theta[3]
  modPred <- drop( theta[1:2] %*% t(cbind(1,obsdata$x)) )
  # Dimensions in mixed vector/matrix products:
  #                         [Ndat] = [P] %*% [P x Ndat]
  logLike <- (1/log(10))*sum(dnorm(modPred - obsdata$y, mean=0,
                                   sd=theta[3], log=TRUE) )
  return(logLike)
}
```


Fitting a straight line with noise

The Posterior distribution

- the Posterior is simply given by the product of the Likelihood and Prior

$$P\left(\theta \mid D\right) \propto P\left(D \mid \theta\right) \times P\left(\theta\right)$$

- the function is interfaced to the `metropolis()` function giving a vector with logPrior and logLikelihood values

R code

```
# Return c(log10(prior), log10(likelihood)) (each generally unnormalized)
# of the linear model
logpost.linearmodel <- function(theta, obsdata) {
  logprior <- logprior.linearmodel(theta)
  if(is.finite(logprior)) { # only evaluate model if parameters are sensible
    return( c(logprior, loglike.linearmodel(theta, obsdata)) )
   } else {
    return( c(-Inf, -Inf) )
   }
}
```

Initializing and running the MCMC process

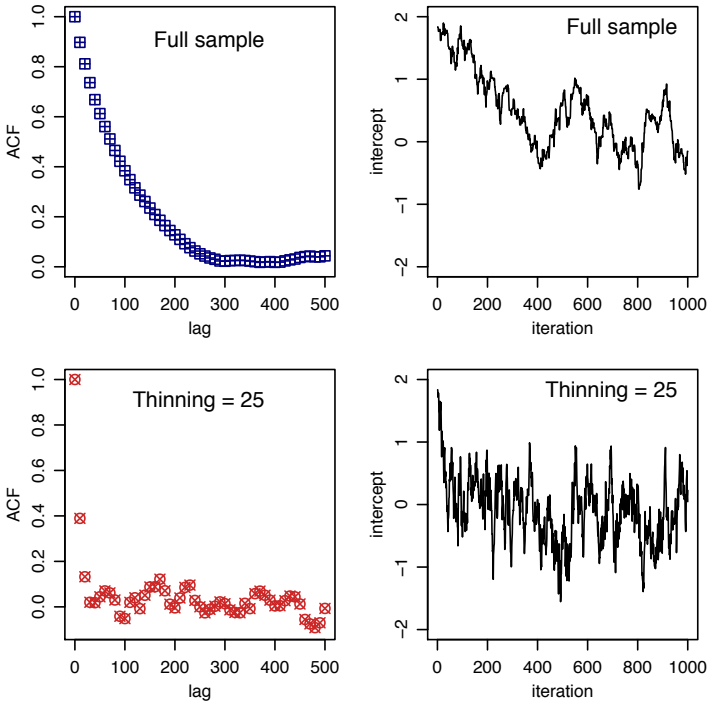
- the starting values for the Markov Chain are $b_0 = 2$, $\alpha = \pi/8$ and $\log_{10}\sigma = \log_{10}(3)$
- the step size for the evolution of the chain are 0.1, 0.02 and 0.1 (respectively for b_0, α and $\log\sigma$)

R code

```
# markov Chain initial values
thetaInit <- c(2, pi/8, log10(3))

# Markov Chain step sizes
sampleCov <- diag(c(0.1, 0.02, 0.1)^2)

set.seed(150)
allSamp <- metrop(func=logpost.linearmodel, thetaInit=thetaInit,
                  Nburnin=0, Nsamp=5e4,
                  sampleCov=sampleCov, verbose=1e3,
                  obsdata=obsdata)

  1000 of       0 +   50000   0.5826
  2000 of       0 +   50000   0.5775
  3000 of       0 +   50000   0.5689
...
 48000 of       0 +   50000   0.5629
 49000 of       0 +   50000   0.5624
 50000 of       0 +   50000   0.5627
```


Analyzing the Markov Chain

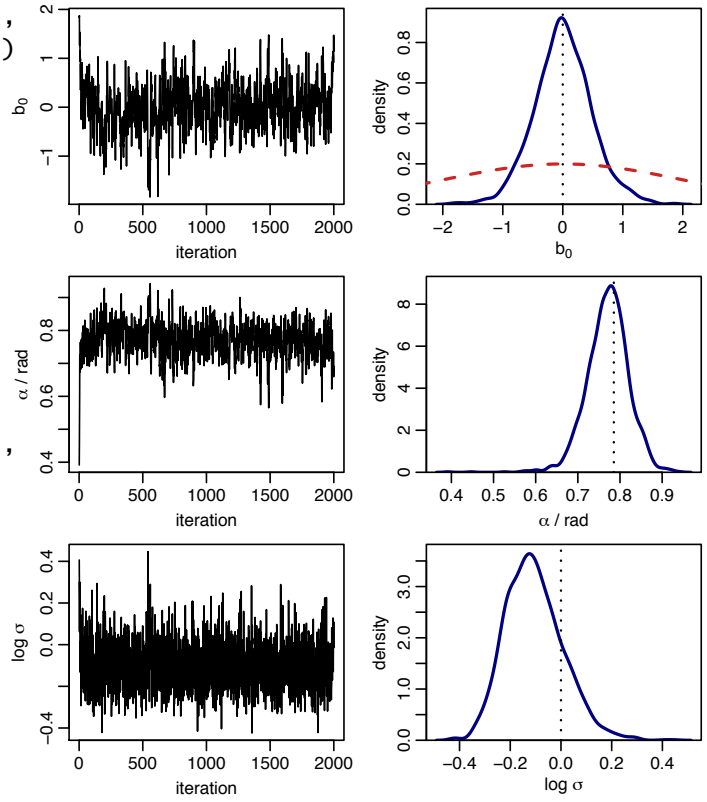
- the unormalized Posterior has been used in the MCMC, the normalization is not needed since samples are drawn with the same relative frequency, independently of the normalization
- in contrast to the Posterior, the Likelihood has to be normalized since it is a pdf over the data and therefore its normalization constant is, in general, a function of the parameters we are sampling
- data are now reduced (thinning = 25) to reduce auto-correlation in the chain
- results and plots are obtained for the last 2k events in the chain




```
allSamp <- metrop(func=logpost.linearmodel, thetaInit=thetaInit, ...)

thinSel  <- seq(from=1, to=nrow(allSamp), by=25) # thin by factor 25

postSamp <- allSamp[thinSel,]
```

Marginal Posterior pdfs

```
parname <- c(expression(b[0]),
    expression(paste(alpha, "␣/␣rad")),
    expression(paste(log, "␣", sigma)))

nr <- nrow(postSamp)
is <- nr-2000

for (j in 3:5) {

  plot(is:nr, postSamp[is:nr,j],
       type="l",
       xlab="iteration",
       ylab=parname[j-2])

  postDen <- density(postSamp[is:nr,j],
                     n=2^10)
  plot(postDen$x, postDen$y,
       col='navy', lwd = 2,
       xlab=parname[j-2],
       ylab="density")

  abline(v=thetaTrue[j-2],
         lwd=1.5, lty=3)
}
```



Posterior parameters estimation

- the joint posterior distribution is the three-dimensional distribution over the MCMC samples, and the one-dimensional marginalized distributions are obtained by making a density estimation of the samples for each parameter
- we evaluate the maximum or mean of the posterior as a single best estimate: the maximum of the posterior is not the peak in each 1-dim pdf, but of the 3-dim pdf

```
# the maximum of the sum of the log(Prior) and log(Likelihood)
posMAP    <- which.max(postSamp[,1] + postSamp[,2])
thetaMAP  <- postSamp[posMAP, 3:5]
thetaMean <- apply(postSamp[,3:5], 2, mean) # Monte Carlo integration
cov(postSamp[, 3:5]) # covariance
cor(postSamp[, 3:5]) # correlation
```

- we get:

$$(b_0, \alpha, \log\sigma) = (0.036, 0.77, -0.19)$$

- if we want to find the mean of the posterior over the original model parameters - (b_0, b_1, σ) - we must transform the individual samples first and then compute the statistic (and not vice versa)

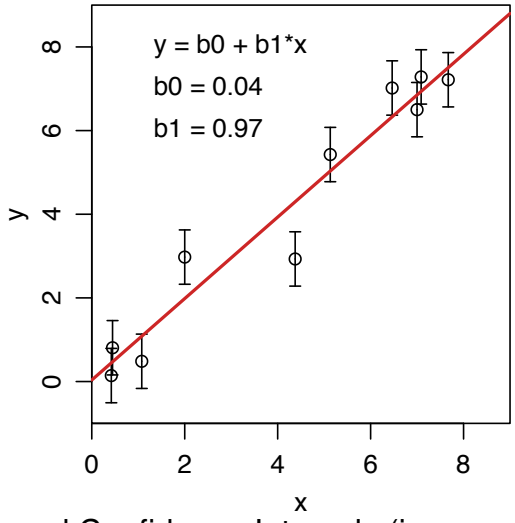
```
mean(tan(postSamp[,4])) # transform alpha to b_1
mean(10^(postSamp[,5])) # transform log10(sigma) to sigma
```

- we get:

$$(b_0, b_1, \sigma) = (0.036, 0.98, 0.81)$$

Linear Fit results

```
plotCI(obsdata$x, obsdata$y,
       xlim=c(0,9), ylim=c(-1,9),
       xaxs="i", yaxs="i",
       xlab="x", ylab="y",
       uiw=10^thetaMAP[3], gap=0)

b0 <- thetaMAP[1]
b1 <- tan(thetaMAP[2])

abline(a=b0, b=b1, lw=2, col='firebrick3')
```



- the R `plotCI()` function is used to plot error bars and Confidence Intervals (in package `gplots`)
- given a set of x and y values and interval width or upper and lower bounds, it plots the points with error bars
  - `uiw` : width of the upper or right error bar. Set to `NA` or to `NULL` to omit upper bars
  - `liw` : width of the lower or left error bar. Defaults to same value as uiw.

Posterior Predictive distribution

- once we have inferred the "best" values for the model parameters, we can use them to predict the value of the Model y_p at any specific value x_p
- the rules of probability lead us to incorporate uncertainties in parameters by marginalizing over them
- we define a posterior predictive distribution

$$P\left(y_p \mid x_p, D\right) = \int P\left(y_p \mid x_p, \theta\right) P\left(\theta \mid D\right) d\theta$$

- the distribution can be evaluated in two ways

Direct method (accurate, but slow)

- is based on evaluating $P(y_p \mid x_p, D)$ over a grid $\{y_p\}$
- at a fixed value of y_p we take our set of N_s posterior samples $\{\theta_l\}$ (obtained by MCMC), calculate the likelihood at each of these, and then average these likelihoods, i.e.

$$P\left(y_p \mid x_p D\right) \sim \frac{1}{N_s} \sum_{j=1}^{N_s} P\left(y_p \mid x_p \theta_j\right)$$

- the posterior predictive distribution is a posterior-weighted average of the predictions (the likelihood) made at each θ

Posterior Predictive distribution

Indirect method

- is based on sampling the joint distribution $P(y_p, \theta \mid x_p, D)$ directly, and marginalizing it over θ
- we can factorize the joint distribution

$$P\left(y_p \theta \mid x_p D\right) = P\left(y_p \mid x_p \theta\right) P\left(\theta \mid D\right)$$

- each of the two pdfs on the right side can be represented by samples drawn from them. The second term is the posterior pdf; we already obtained the set of samples $\{\theta_j\}$ from this with the MCMC. The first term is the likelihood
- As the likelihood is a uni-variate Gaussian, it may be sampled using a standard function. Its mean is the evaluation of the straight line at (b_0, b_1), and its standard deviation is σ
- the R code is:

```
likeSamp <- rnorm(n=length(modPred), mean=modPred, sd=10^postSamp[,5])
```

- where `modPred` (of length N_s) is the evaluations of the straight line at the posterior samples. We now have samples of θ and y_p
- we marginalize their joint distribution simply by ignoring the θ, to give the required distribution $P(y_p \mid x_p D)$

Posterior Predictive distribution - example

```
xnew <- 6

# Evaluate generative model at posterior samples (from MCMC).
# Dimensions in matrix multiplication: [Nsamp x 1] = [Nsamp x P] %*% [P x 1]
modPred <- cbind(postSamp[,3], tan(postSamp[,4])) %*% t(cbind(1,xnew))

# ---- Direct method ----
# ycand must span full range of likelihood and posterior
dy    <- 0.01
ymid  <- thetaMAP[1] + xnew*tan(thetaMAP[2]) # to center choice of ycand

ycand <- seq(ymid-10, ymid+10, dy) # uniform grid of y with step size dy
ycandPDF <- vector(mode="numeric", length=length(ycand))

for(k in 1:length(ycand)) {
  like <- dnorm(ycand[k], mean=modPred, sd=10^postSamp[,5]) # [Nsamp x 1]
  ycandPDF[k] <- mean(like) # integration by rectangle rule. Gives a scalar
}

# Note that ycandPDF[k] is normalized, i.e. sum(dy*ycandPDF)=1.
# Find peak and approximate confidence intervals at 1sigma on either side
peak.ind  <- which.max(ycandPDF)

lower.ind <- max( which(cumsum(dy*ycandPDF) < pnorm(-1)) )
upper.ind <- min( which(cumsum(dy*ycandPDF) > pnorm(+1)) )
yPredDirect <- ycand[c(peak.ind, lower.ind, upper.ind)]
```

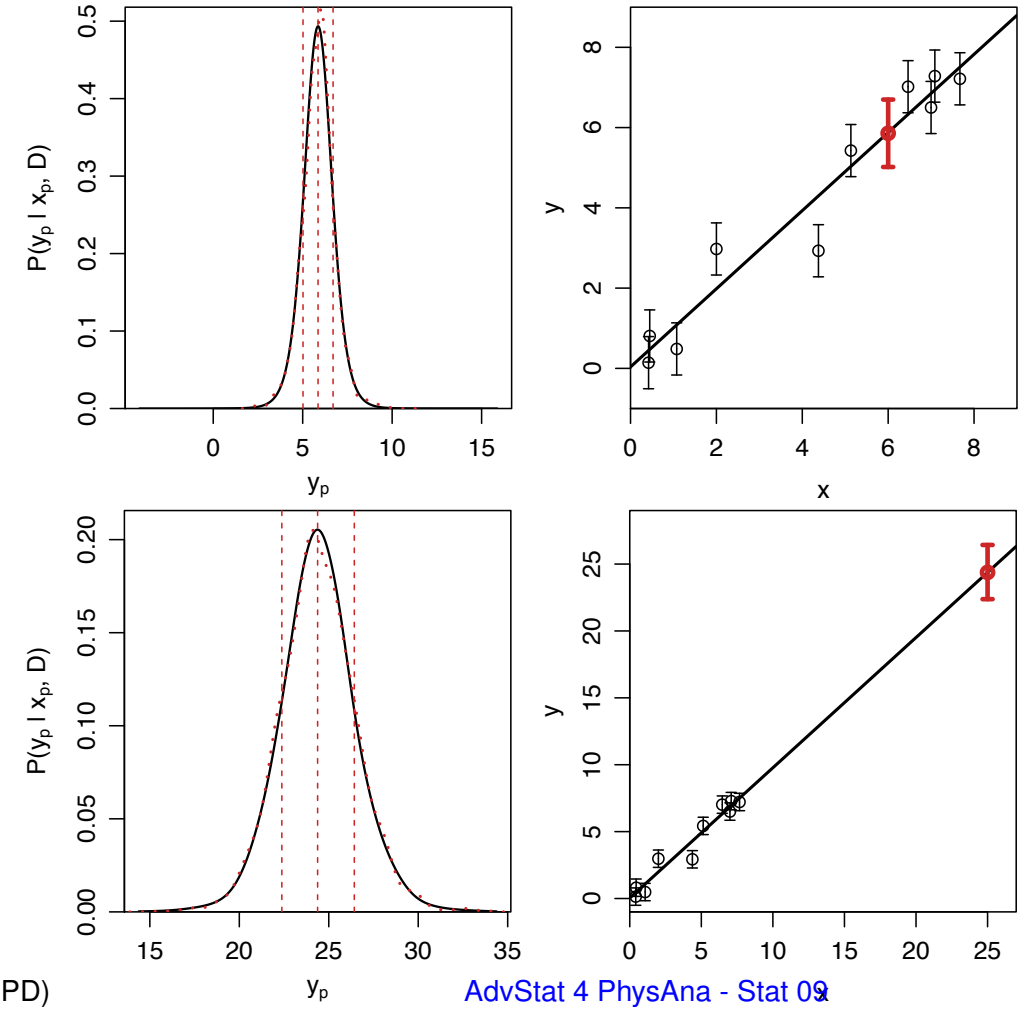
Posterior Predictive distribution - example

```
xnew <- 6

# Evaluate generative model at posterior samples (from MCMC).
# Dimensions in matrix multiplication: [Nsamp x 1] = [Nsamp x P] %*% [P x 1]
modPred <- cbind(postSamp[,3], tan(postSamp[,4])) %*% t(cbind(1,xnew))

# ---- Indirect method ----
likeSamp <- rnorm(n=length(modPred), mean=modPred, sd=10^postSamp[,5])
likeDen  <- density(likeSamp, n=2^10)

# Find peak and confidence intervals
yPredIndirect <- c(likeDen$x[which.max(likeDen$y)], quantile(likeSamp,
                          probs=c(pnorm(-1), pnorm(+1)), names=FALSE))
```


Posterior Predictive distribution - example

```
plot(ycand, ycandPDF, type="l", lwd=1.5,
     ylim=1.05*c(0,max(ycandPDF)), xlab=expression(y[p]),
     ylab=expression(paste("P(", y[p], "␣|␣", x[p], ",␣D)")))

abline(v=yPredDirect, col='firebrick3', lty=2)

# overplot result from the indirect method
lines(likeDen$x, likeDen$y, col='firebrick3', type="l", lty=3, lwd=2)


> rbind(yPredDirect, yPredIndirect)
                  [,1]     [,2]     [,3]
yPredDirect   5.858070 5.018070 6.698070
yPredIndirect 5.876795 5.037817 6.665148


# Overplot direct prediction with original data and the MAP model
plotCI(obsdata$x, obsdata$y, xlim=xlim, ylim=ylim,
       uiw=10^thetaMAP[3], gap=0, xlab="x", ylab="y")
abline(a=thetaMAP[1], b=tan(thetaMAP[2]), lwd=2) # MAP  model

plotCI(xnew, ycand[peak.ind], li=ycand[lower.ind], ui=ycand[upper.ind],
       gap=0, add=TRUE, lwd=3, col='firebrick3')
```

Linear Fit Prediction results

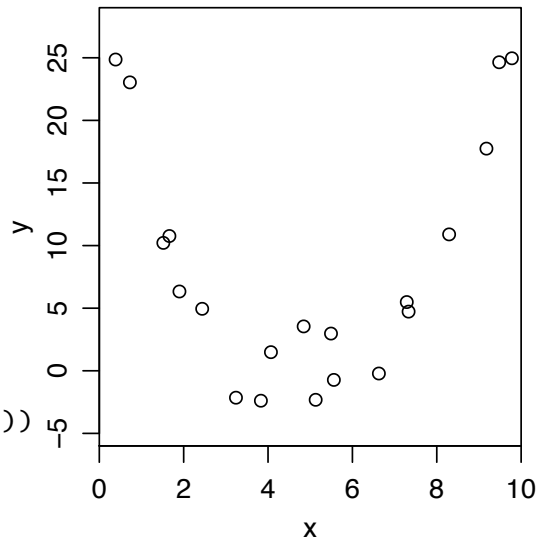


Fitting a quadratic curve with noise

- we have a new set of data we want to fit to a generative model

$$y = f(x) + \epsilon \text{ with } f(x) = b_0 + b_1 x + b_2 x^2$$

- they have been drawn at fixed x values from a straight line with $(b_0, b_1, b_2) = (25, -10, 1)$, to which zero mean Gaussian noise with $\sigma = 2$ has been added.




```
Ndat <- 20
xra <- c(0,10)
x <- sort(runif(Ndat, min=xra[1], max=xra[2]))
sigTrue <- 2

# 1 x P vector: coefficients,
#       b_p, of sum_{p=0} b_p*x^p
modMat <- c(25, -10, 1)
y <- cbind(1,x,x^2) %*% as.matrix(modMat) + rnorm(Ndat, 0, sigTrue)
# Dimensions in matrix multiplication:
#    [Ndat x 1] = [Ndat x P] %*% [P x 1] + [Ndat]
# cbind does the logical thing combining a scalar
# and vector; then do vector addition

# finally, convert to a vector
y <- drop(y)
```

Fitting a quadratic curve with noise

The Prior

- a Gaussian prior is used on b_0, $b_0 \sim \mathcal{N}(0, 10)$
- b_1 is transformed to $\alpha = \arctan b_1$, and a uniform prior is used $\alpha \sim \mathcal{U}(0, 2\pi)$
- a Gaussian prior is used on b_2, $b_2 \sim \mathcal{N}(0, 5)$
- σ is transformed to $\log \sigma$ and an improper uniform prior is used

R code

```
#
# parameters:
#   theta[1] -> b_0
#   theta[2] -> alpha
#   theta[3] -> b_2
#   theta[4] -> log(sigma)

logprior.quadraticmodel <- function(theta) {
  b0Prior      <- dnorm(theta[1], mean=0, sd=10)
  alphaPrior   <- 1
  b2Prior      <- dnorm(theta[3], mean=0, sd=5)
  logysigPrior <- 1
  logPrior <- sum(log10(b0Prior), log10(alphaPrior),
                  log10(b2Prior), log10(logysigPrior) )
  return(logPrior)
}
```


Fitting a quadratic curve with noise

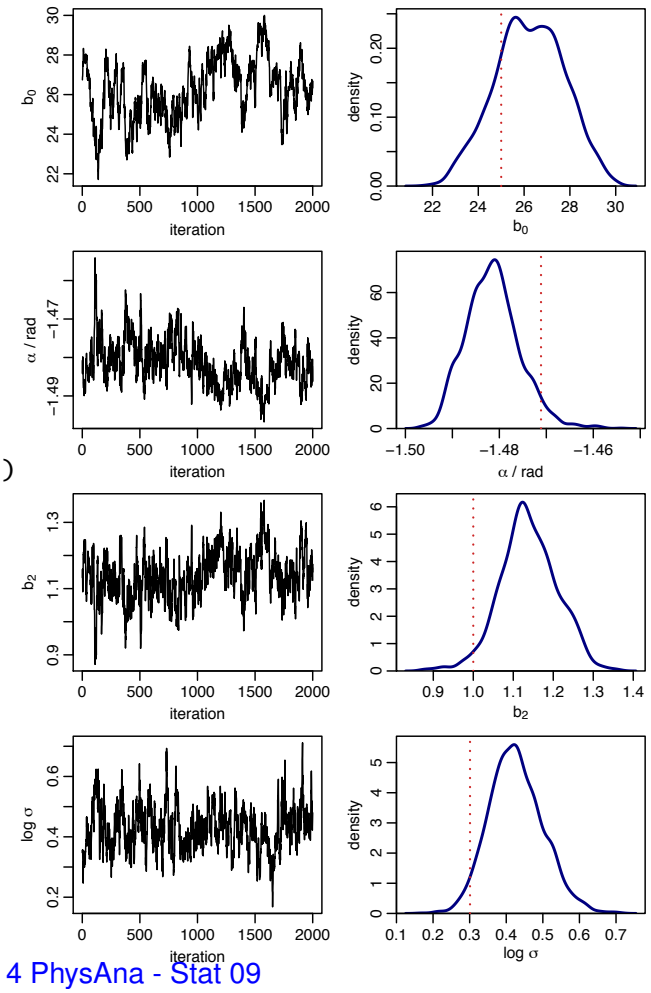
- the parameters step sizes (Gaussian standard deviations) are chosen as $(b_0, \alpha, b_2, \log \sigma) = (0.1, 0.01, 0.01,$ and $0.01)$
- as starting point any value can be in principle chosen, but it could take a large number of steps to locate the high density region of the posterior. Therefore a classical approach (`lm()` function) has been used
- to achieve good chains more iterations than in the straight line problem are needed (higher complexity of the model)
- after a burn-in of 20 k iterations, further 200 k iterations are sampled
- to reduce the auto-correlation, a thinning factor of 100, is used

R code

```
sampleCov <- diag(c(0.1, 0.01, 0.01, 0.01)^2)
thetaInit <- c(27.4, atan(-11.7), 1.18, log10(2.4))
set.seed(250)
allSamp <- metrop(func=logpost.quadraticmodel, thetaInit=thetaInit,
                  Nburnin=2e4, Nsamp=2e5,
                  sampleCov=sampleCov, verbose=2e3, obsdata=obsdata)
  2000 of  20000 +  2e+05  0.0729
  4000 of  20000 +  2e+05  0.0940
...
216000 of  20000 +  2e+05  0.1039
218000 of  20000 +  2e+05  0.1039
220000 of  20000 +  2e+05  0.1041
```

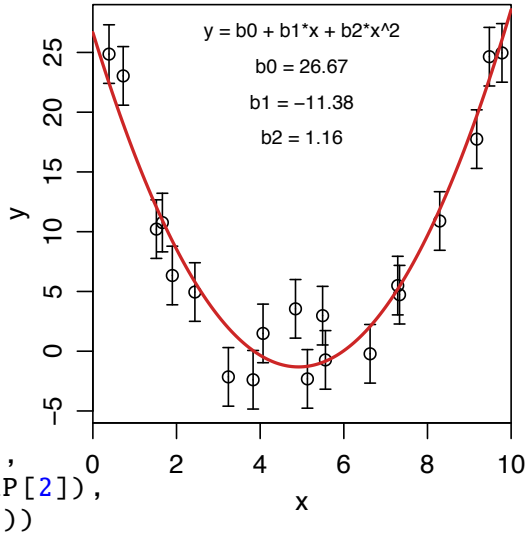
Quadratic curve : posterior pdfs

```
parnames <- c(expression(b[0]),
     expression(paste(alpha, "␣/␣rad")),
     expression(b[2]),
     expression(paste(log, "␣", sigma)))

for (j in 3:6) {

  plot(1:nrow(postSamp), postSamp[,j],
     type="l", xlab="iteration",
     ylab=parnames[j-2])

  postDen <- density(postSamp[,j], n=2^10)

  plot(postDen$x, postDen$y, type="l",
     lwd=2, yaxs="i", col='navy',
     ylim=1.05*c(0,max(postDen$y)),
     xlab=parnames[j-2], ylab="density")

  abline(v=thetaTrue[j-2],
       lwd=1.5, lty=3, col='firebrick3')
}
```



Quadratic Fit results

```
plotCI(obsdata$x, obsdata$y,
       xlim=xrange, ylim=c(-6,29),
       xaxs="i", yaxs="i",
       xlab="x", ylab="y",
       uiw=10^thetaMAP[4], gap=0)

b0 <- thetaMAP[1]
b1 <- tan(thetaMAP[2])
b2 <- thetaMAP[3]

ysamp <- cbind(1,
         xsamp,
         xsamp^2) %*% as.matrix(c(thetaMAP[1],
                                  tan(thetaMAP[2]),
                                  thetaMAP[3]))
lines(xsamp, drop(ysamp), lwd=2, col='firebrick3')
```



- the R `plotCI()` function is used to plot error bars and Credibility Intervals
- given a set of *x* and *y* values and interval width or upper and lower bounds, it plots the points with error bars
- `uiw` : width of the upper or right error bar. Set to `NA` or to `NULL` to omit upper bars
- `liw` : width of the lower or left error bar. Defaults to same value as uiw.

Gibbs sampling and JAGS

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Stat Lect 10

Computational Bayesian Statistics

- the posterior distribution itself is the essence of bayesian inference

$$P(\theta \mid y) = \frac{f(y \mid \theta)\, g(\theta)}{\int f(y \mid \theta)\, g(\theta) d\, \theta}$$

- but most of the time it is not known analytically, and it must be computed numerically with Monte Carlo methods
- Markov Chain Monte Carlo (MCMC) methods are commonly used for sampling from a posterior distribution: we let the Markov chain *run* long enough ➜ a random draw from the chain can be considered a random draw from the posterior
- it's a radically different approach: instead of computing numerically the posterior distribution, we draw a sample from the posterior distribution.
- two main MCMC methods are commonly used:
  - the Metropolis-Hastings algorithm
  - the Gibbs sampling algorithm

Metropolis-Hastings : 1-dim example

- it samples from a target density by choosing values from a candidate density
- the acceptance of the new value (*proposal*) depends only on the previously accepted value (*current value*)
- using a symmetric transition probability, we generate a Markov Chain
- the acceptance probability, also called Metropolis ratio, is

$$\rho = \texttt{MIN}\left(1,\ \frac{f(s)}{f(\theta_t)}\,\frac{Q(\theta_t \mid s)}{Q(s \mid \theta_t)}\right)$$

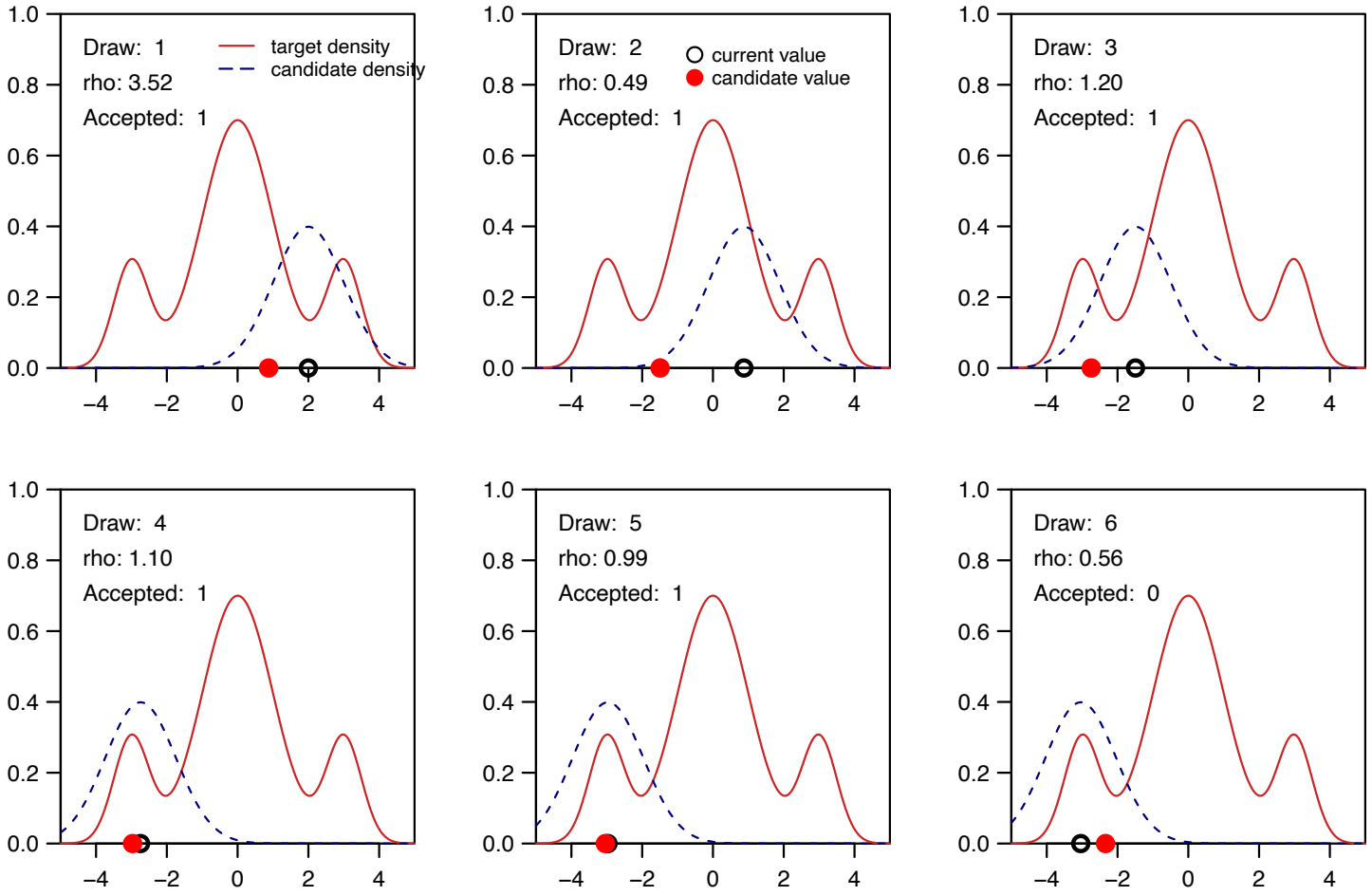
Problem

- let's assume ve have a target density that is the sum of three Normal distributions

$$f(x) = a_1\ \texttt{Norm}(0,\ 1) + a_2\ \texttt{Norm}(3,\ 0.5^2) + a_3\ \texttt{Norm}(-3,\ 0.5^2)$$

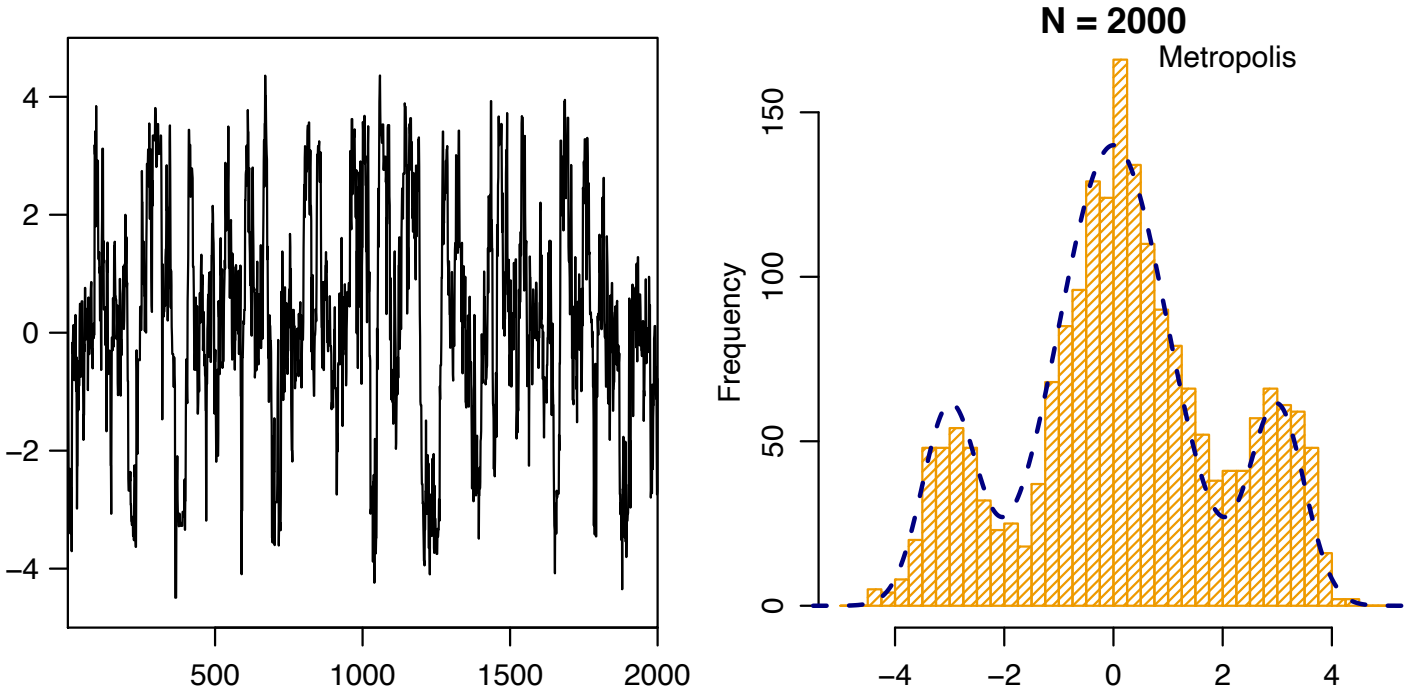
- with $a_1 = 0.7, a_2 = 0.15$ and $a_3 = 0.15$

Metropolis-Hastings : 1-dim example



Metropolis-Hastings : 1-dim example

- the sample is moving through the space quite satisfactory
- extreme values are selected, from time to time, but the chain tends to jump back to the central region (with higher probability) very quickly



Metropolis-Hastings : indep. cand. chain

- a variant to the Metropolis-hastings algorithm uses an independent candidate density
- Hastings (1970) introduced Markov Chains with candidate densities that did not depend on the current value in the chain

$$Q(s \mid \theta) = Q_2(s)$$

- Q_2 is some function that dominates the target density in the tails
- therefore the acceptance probability, the Metropolis ratio, simplifies to

$$\rho = \texttt{MIN}\left(1,\ \frac{f(s)}{f(\theta_t)}\frac{Q(\theta_t \mid s)}{Q(s \mid \theta_t)}\right) = \texttt{MIN}\left(1,\ \frac{f(s)}{f(\theta_t)}\frac{Q_2(\theta_t)}{Q_2(s)}\right)$$

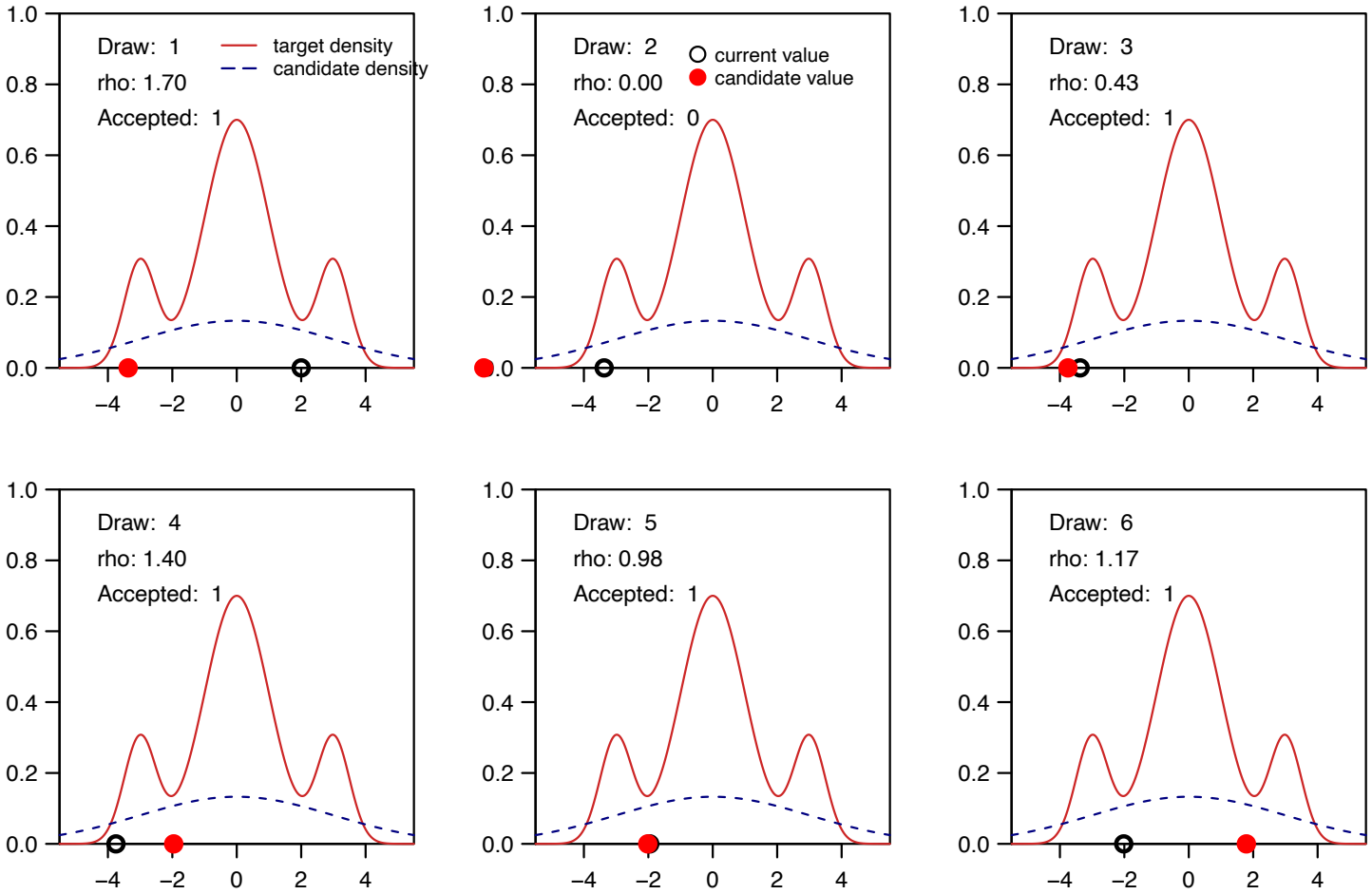
Problem

- let's study the same problem

$$f(x) = a_1\ \texttt{Norm}(0,\ 1) + a_2\ \texttt{Norm}(3,\ 0.5^2) + a_3\ \texttt{Norm}(-3,\ 0.5^2)$$

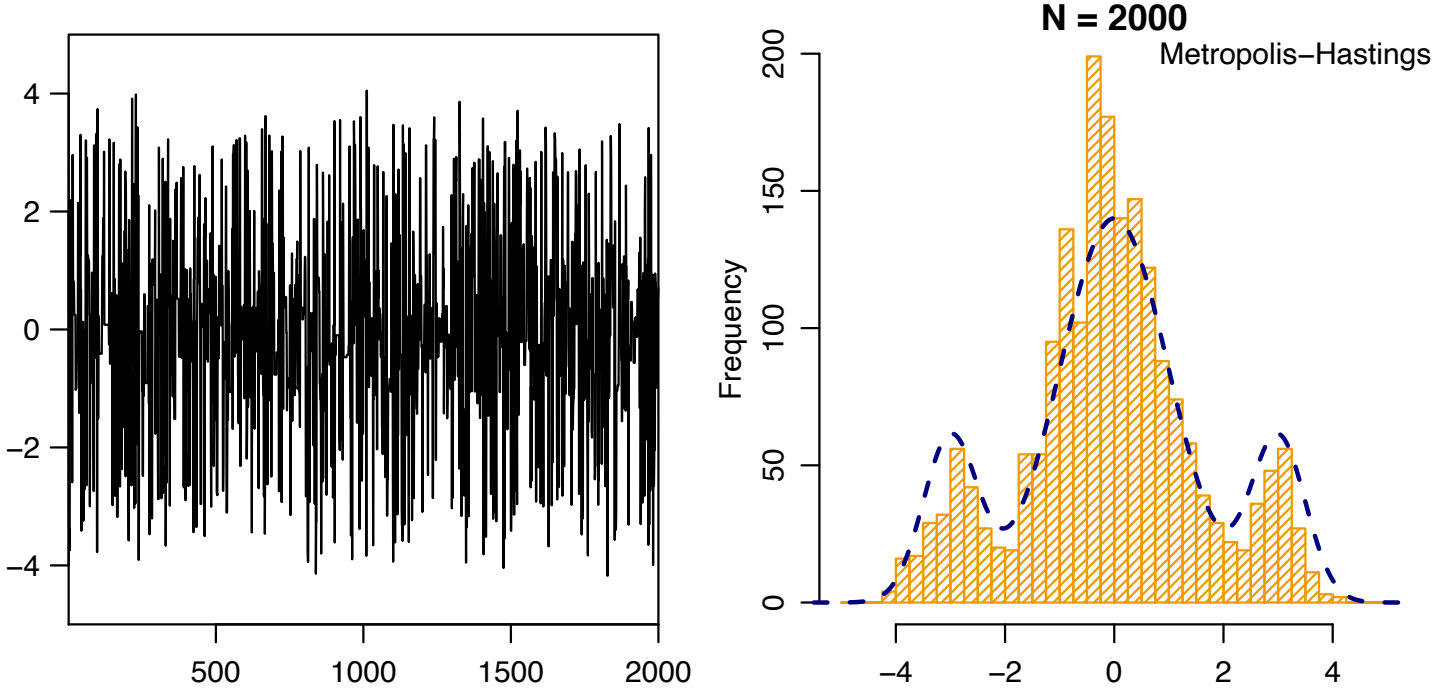
- with $a_1 = 0.7, a_2 = 0.15$ and $a_3 = 0.15$
- assuming that the candidate density is a $\texttt{Norm}(0, 3^2)$ distribution function

Metropolis-Hastings : indep. cand. chain



Metropolis-Hastings : indep. cand. chain

- the sample is moving through the space quite satisfactory
- the independent candidate density allows for larger jumps, but it may accept fewer proposals than the random-walk chain
- nevertheless the acceptance is larger and the chain will potentially explore the parameter space faster



Gibbs sampling

- it is one of the most widely used algorithms for simulating Markov chains
- it is a special case of the Metropolis-Hastings algorithm and it is most relevant with multi-parameters problems
- in general, Metropolis-hasting can be improved by only updating a block of parameters at each iteration ➜ blockwise Metropolis-Hastings algorithm
- the Gibbs sampling algorithm is a special case of the blockwise Metropolis-Hastings
- it generates a multi-dimensional Markov chain by splitting the vector of random variables θ into subvectors and sampling each subvector in turn, conditional on the most recent values of all other elements of θ
- the beauty of Gibbs sampling is that simulation from a complex, high-dimensional joint posterior distribution is reduced to a sequence of algorithms for sampling from one or low-dimensional distributions
- Gibbs sampling is most suited for hierarchical models, where the dependencies between model parameters is well-defined

Gibbs sampling algorithm

(1) choose arbitrary starting values $\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_k^{(0)}$
(subscript = component, superscript = iteration step)

(2) sample new values for each element with the following steps:

- sample $\theta_1^{(1)}$ from the full-conditional distribution,

$$\theta_1^{(1)} \sim P(\theta_1 \mid \theta_2^{(0)}, \theta_3^{(0)}, \ldots \theta_k^{(0)}, y)$$

where y indicates the data

- sample a new $\theta_2^{(1)}$, for the second component, from its full conditional distribution

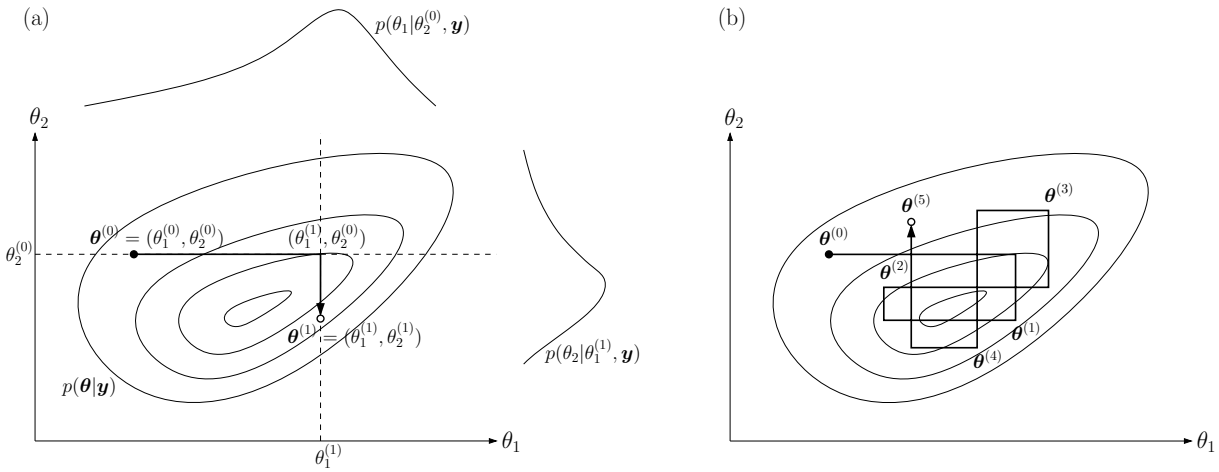
$$\theta_2^{(1)} \sim P(\theta_2 \mid \theta_1^{(1)}, \theta_3^{(0)}, \ldots \theta_k^{(0)}, y)$$

- complete the step for all the other components, obtaining a sequence of dependent realization of $\theta_1^{(1)}, \theta_2^{(1)}, \ldots, \theta_k^{(1)}$

(3) repeat `step 2` many times conditioning on the most recent values of other parameters

Gibbs sampling algorithm : 2-dim example

- picture (b) shows the first five iterations of the Gibbs sampler
- the sampler always moves parallel to the axes
- the starting point, $\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)})$ is shown
- $p(\theta_1 | \theta_2^{(0)}, y)$, shown on top, is the univariate density and is obtained by taking a horizontal "slice" through the 2 joint posterior distribution at the value $\theta_2 = \theta_2^{(0)}$ (horizontal dashed line)
- a new value for $\theta_1^{(0)}$ is generated from this full conditional, and then a "slice" parallel to the θ_2 axis is taken through the joint posterior (vertical dashed line)



JAGS: Just Another Gibbs Sampler

- JAGS is a program mainly written by M. Plummer with the aim of providing a BUGS (Bayesian Inference Using Gibbs Sampling) engine for UNIX
  - more infos are available at `http://sourceforge.net/projects/mcmc-jags/`
  - the latest version is 4.3.1 (April 12, 2022)

- `rjags` is another `R` package that allows to run JAGS from within `R`
  `https://cran.r-project.org/web/packages/rjags/`
  `https://cran.r-project.org/web/packages/rjags/rjags.pdf`
  - available for Linux-64 (`v4.6`) and osx-64 (`v4.6`)
    `conda install -c conda-forge r-rjags`

- `R2jags` is an `R` package that allow to fit JAGS models from within `R`
  `https://cran.r-project.org/web/packages/R2jags/`
  `https://cran.r-project.org/web/packages/R2jags/R2jags.pdf`
  - available only for Linux-64 (`v0.5_7`)
    `conda install -c glaxosmithkline r-r2jags`

rjags

- an analysis with `rjags` proceeds through the following steps:

(1) define the model using the BUGS language in a separate file

(2) read in the model file using the `jags.model` function. This creates an object of class `jags`

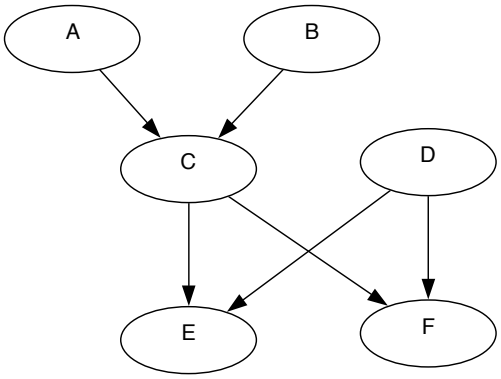
(3) update the model using the update method for `jags` objects. This constitutes the *burn-in* part

(4) extract samples from the model object using the `coda.samples` function. This creates an object of class `mcmc.list` which can be used to summarize the posterior distribution. The `coda` package also provides convergence diagnostics to check that the output is valid for analysis

The BUGS language

- BUGS (Bayesian inference Using Gibbs Sampling) is also a language that allows to specify Bayesian models for Bayesian computation
- it is based on graphical representation which is used to express the joint relationship between all known and unknown quantities in a model through a series of simple local relationships

- let's consider the graph in the figure:
- A, B and D have no parents and are therefore marginally independent
- A and B are parents of C which, in turn, is a parent (with D) of E and F
- if we observe E, this will induce a dependency between C and D and between A and B, since two nodes without common parents are only independent given no descendants have been observed
- from the graph we can see that the joint distribution of the set of quantities may be written



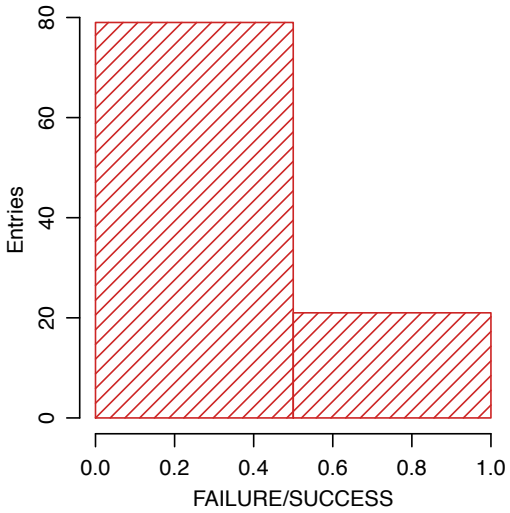
$$P(A,B,C,D,E,F) = P(A)P(B)P(C|A,B)P(D)P(E|C,D)P(F|C,D)$$

Ex 1: Bernoulli process

The Problem

- given a set of observation, coming from a Bernoulli process, we want to infere the probability p of the process from the sequence of success/failure, and predict the number of successes in the future
- the observed sequence is the following:

```
X <- c(0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 1)
```



- we describe the model with BUGS and let `jags` solve our inference problem

Ex 1: BUGS model and parameters

- file: `s11_inf_p_pred.bug`

```
model {
   # data likelihood
   for (i in 1:length(X)) {
     X[i] ~ dbern(p);
   }
   # a uniform prior for p
   p ~ dbeta(1, 1);

   # Predicted data, given p
   y ~ dbin(p, n_next);
}
```

- a list with the data for the model :

```
data <- NULL
data$X <- data_obs     # Set of observations
data$n <- length(X)    # those to be considered

data$n_next <- 10      # Predictions
```

- the model is created passing the BUGS data file and a list with all the data and model parameters

```
jm <- jags.model(model, data)
```

Ex 1: running jags

```
# Update the Markov chain (Burn-in)
update(jm, 1000)
chain <- coda.samples(jm, c("p", "y"), n.iter=10000)
print(summary(chain))
```

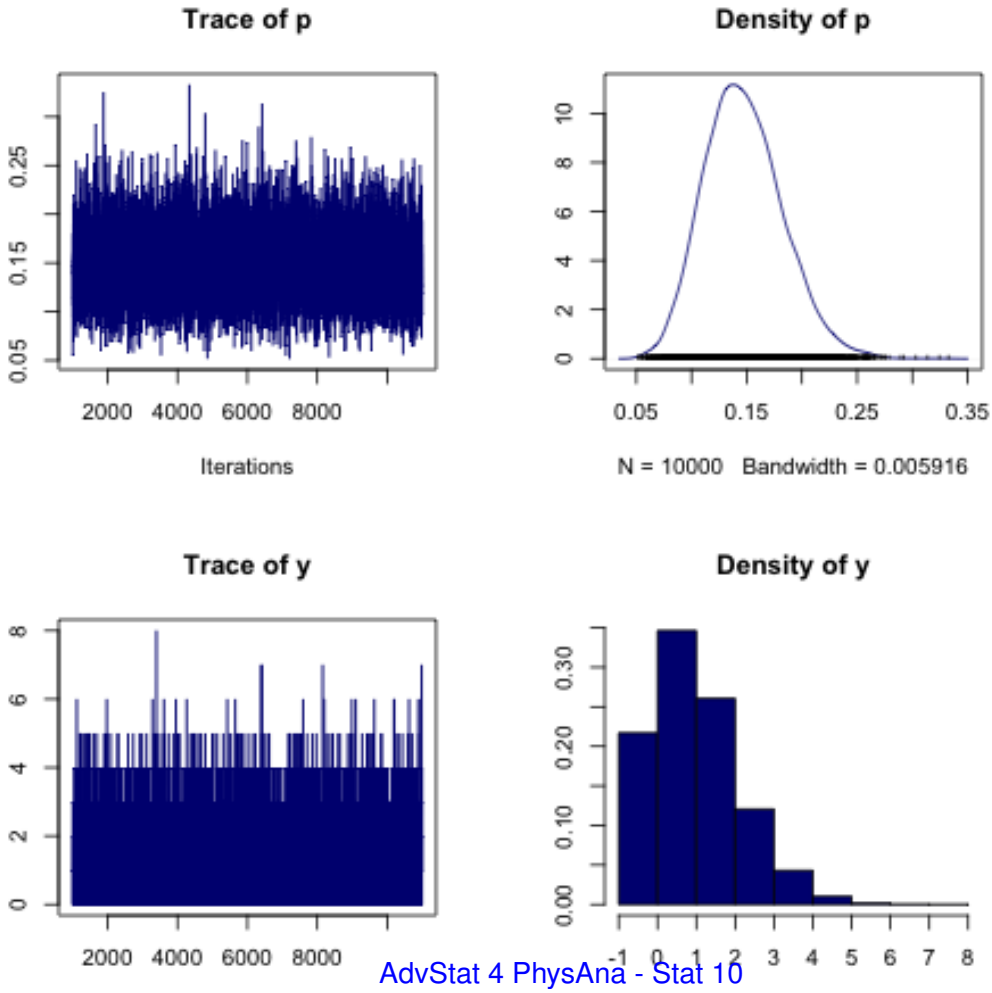
- Output from R:

```
Compiling model graph
   Resolving undeclared variables /  Allocating nodes
Graph information:
   Observed stochastic nodes: 100 / Unobserved stochastic nodes: 2
   Total graph size: 105

Initializing model
  |**************************************************| 100%
  |**************************************************| 100%

Iterations = 1001:11000
Thinning interval = 1 / Number of chains = 1 | SampSize/chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:
    Mean      SD  Naive SE Time-series SE
p 0.1474 0.03495 0.0003495      0.0003547
y 1.4872 1.17489 0.0117489      0.0117489

2. Quantiles for each variable:
     2.5%    25%    50%    75%  97.5%
p 0.08681 0.1225 0.1448 0.1695 0.2224
y 0.00000 1.0000 1.0000 2.0000 4.0000
```


Ex 1: producting control plots

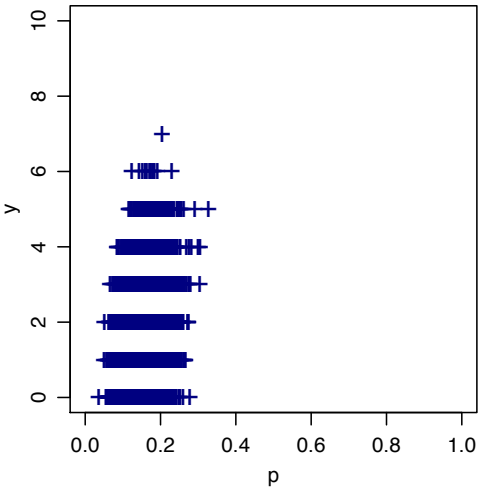
```
plot(chain, col="navy")

# Let's format our chain
chain.df <- as.data.frame( as.mcmc(chain) )
cat(sprintf("\n Correlation matrix: \n"))
print(cor(chain.df))

#
# p inference result
#
hist(chain.df$p, nc=50, prob=TRUE, col='darkolivegreen2',
     xlab='p', ylab='f(p)', main='Inference on p')

#
# next data prediction probability
#
ty <- table(chain.df$y)
barplot(ty/sum(ty), col='firebrick2', xlab='y', ylab='f(y)',
        ylim=c(0,0.40),
        main=sprintf('Number of successes in %d future trials', data$n_next))

#
# Correlation between p and predicted variable
#
plot(chain.df$p, chain.df$y, xlab='p', ylab='y', main="",
     pch='+', col='navy', cex=1.5,
     xlim=c(0,1), ylim=c(0,10))
```

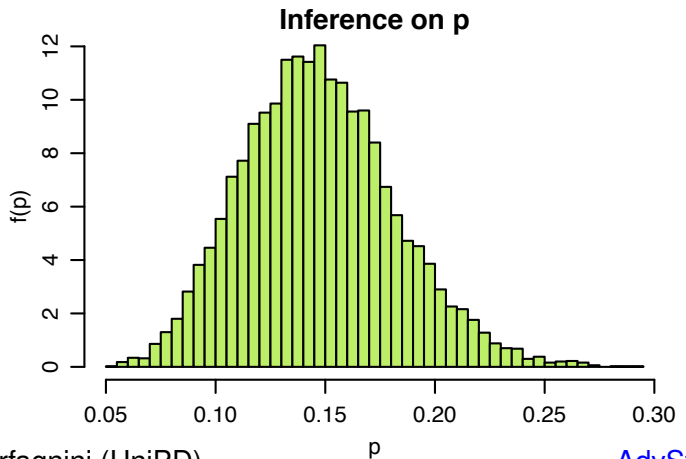
Ex 1: jags chains

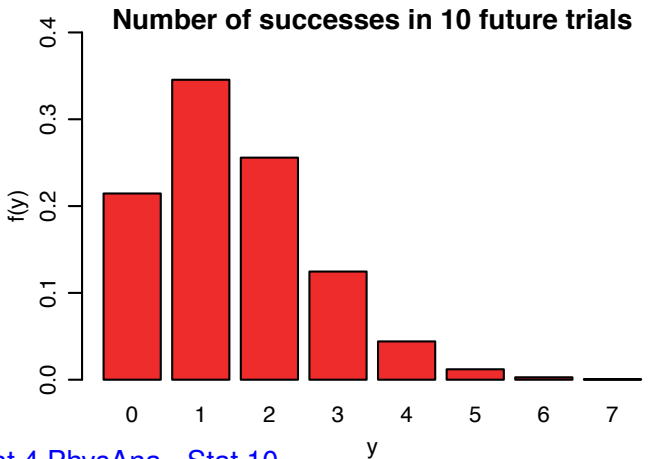


Ex 1: jags results

```
     Mean       SD  Naive SE Time-series SE
p 0.1474 0.03495 0.0003495      0.0003547
y 1.4872 1.17489 0.0117489      0.0117489

     2.5%    25%    50%    75%  97.5%
p 0.08681 0.1225 0.1448 0.1695 0.2224
y 0.00000 1.0000 1.0000 2.0000 4.0000

 Correlation matrix:
          p         y
p 1.0000000 0.3031662
y 0.3031662 1.0000000
```







Ex 2: Poisson inference

The Problem

- given the number of counts from a ionizing radiation detector, we want to infer the parameter λ of the underlying Poisson process
- the BUGS model (file: `s11_inf_lambda_pred.bug`) is the following:

```
model {
   # data likelihood
   X ~ dpois(lambda);

   # a uniform prior for lambda
   lambda ~ dexp(0.00001)

   # Predicted data, given lambda
   Y ~ dpois(lambda);
}
```

- and our data:

```
data <- NULL
data$X <- 100  # number of counts
```

- we create the jags model:

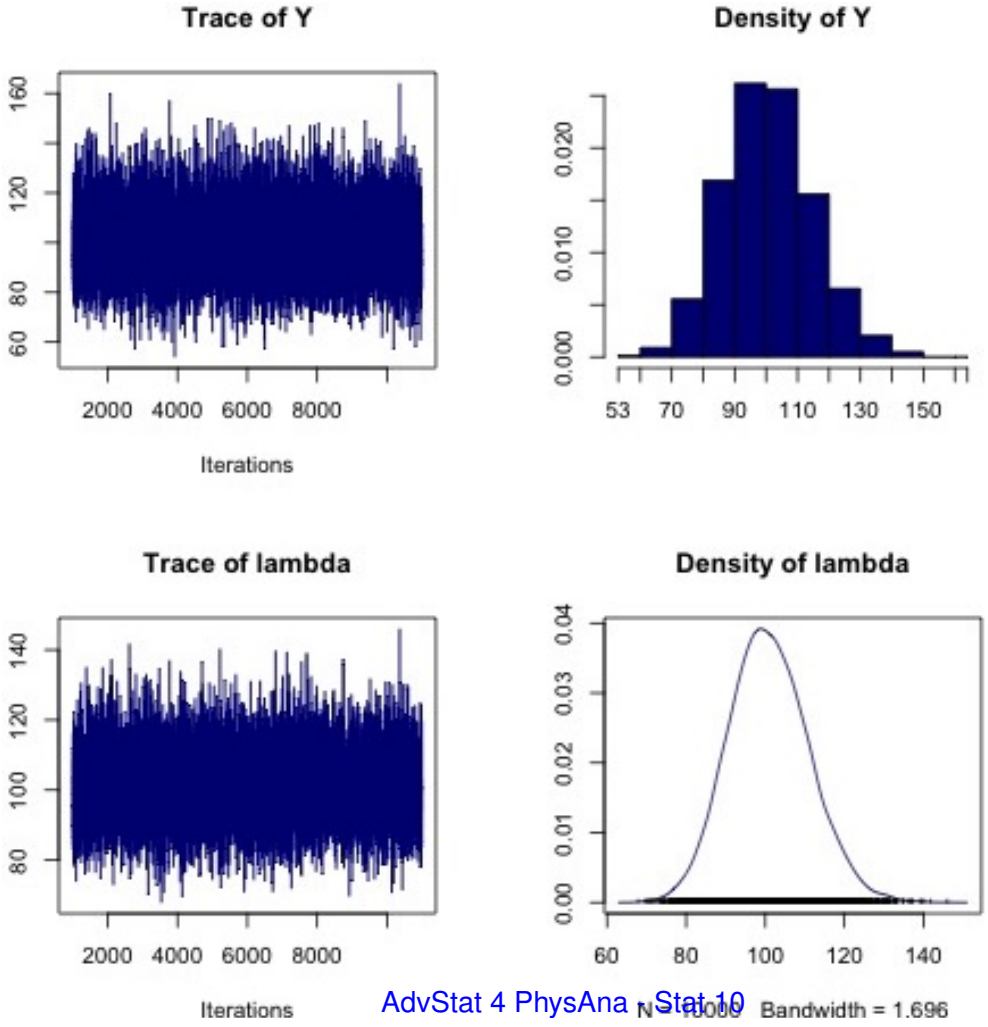
```
library(rjags)
model <- "s11_inf_lambda_pred.bug"
jm <- jags.model(model, data)
```


Ex 2: Poisson inference

- the rest of the code is:

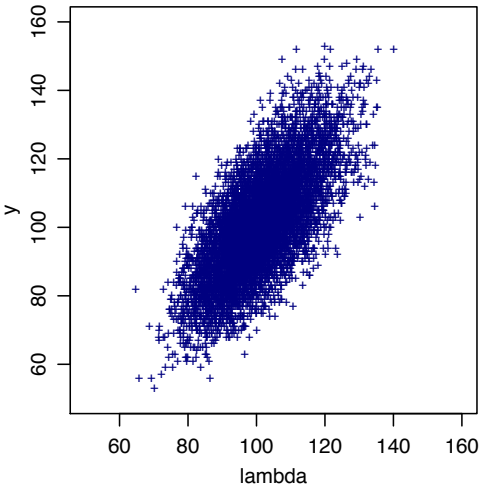
```
# Update the Markov chain (Burn-in)
update(jm, 1000)

chain <- coda.samples(jm, c("lambda", "Y"), n.iter=10000)

plot(chain, col="navy")

# Let's format our chain
chain.df <- as.data.frame( as.mcmc(chain) )

#
# Probability plots
par(mfrow=c(3,2), mgp=c(2.0,0.8,0), mar=c(3.5,3.5,1,1), oma=0.1*c(1,1,1,1))
hist(chain.df$lambda, nc=100, prob=TRUE, col='darkolivegreen2',
     xlim=c(40, 170),
     xlab='lambda', ylab='f(lambda)', main='Inference on lambda')

ty <- table(chain.df$Y)
barplot(ty/sum(ty), col='firebrick2', xlab='Y', ylab='f(Y)',
        # ylim=c(0,0.40),
        main=sprintf('Predicted counts'))

#
# And present/ future prediction correlations
plot(chain.df$lambda, chain.df$Y, xlab='lambda', ylab='y', main="",
     pch='+', col='navy', cex=0.75, asp=1,
     xlim=c(50,160), ylim=c(50,160))
```

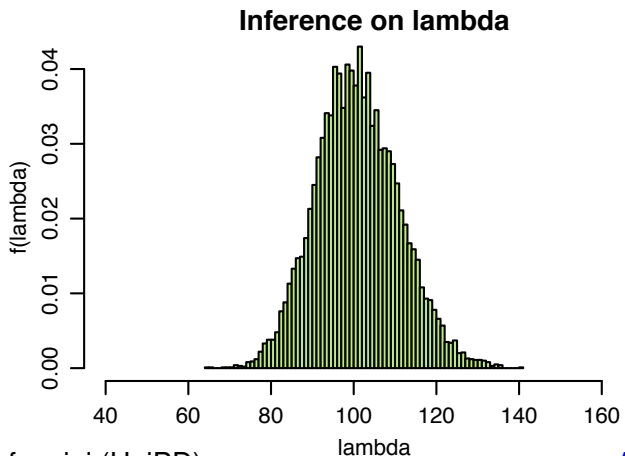
Ex 2: jags chains

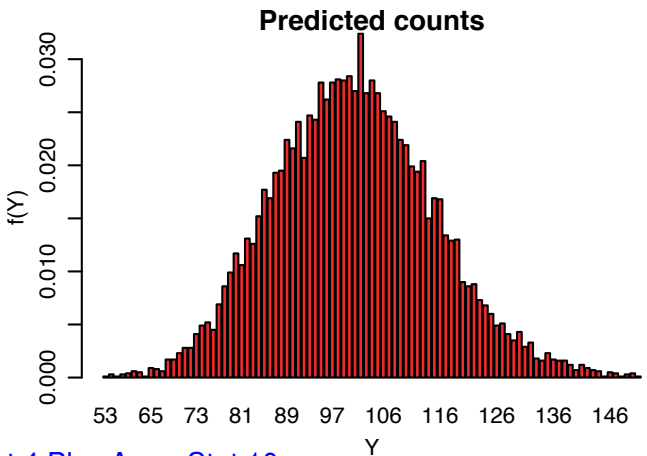


Ex 2: jags Poisson results

```
1. Empirical mean and standard deviation for
   each variable,
   plus standard error of the mean:

         Mean     SD Naive SE Time-series SE
Y       101.1 14.26   0.1426          0.1426
lambda 100.9 10.10   0.1010          0.1010

2. Quantiles for each variable:

         2.5%   25%   50%   75% 97.5%
Y       75.00 91.00 101.0 110.0 131.0
lambda 82.24 93.99 100.5 107.6 121.7
```





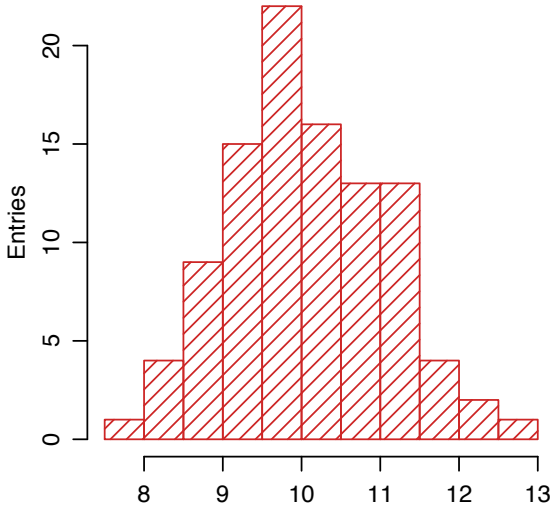


Ex 3: Normal inference

Problem

- given a set of 100 measurements, we want to infer the mean and sigma, assuming they are coming from a gaussian distribution with unknown mean and sigma
- the BUGS model (file: `s11_norm_pred.bug`) is the following:

```
#
# Gaussian model with unknown mean and sigma
#
model {
   for (i in 1:length(X)) {
      X[i] ~ dnorm(mu, tau);
   }
   mu ~ dnorm(0.0, 1.0E-6);

   tau ~ dgamma(1.0, 1.0E-4);
   sigma <- 1.0/sqrt(tau);

   # future observation
   Y ~  dnorm(mu, tau);
}
```



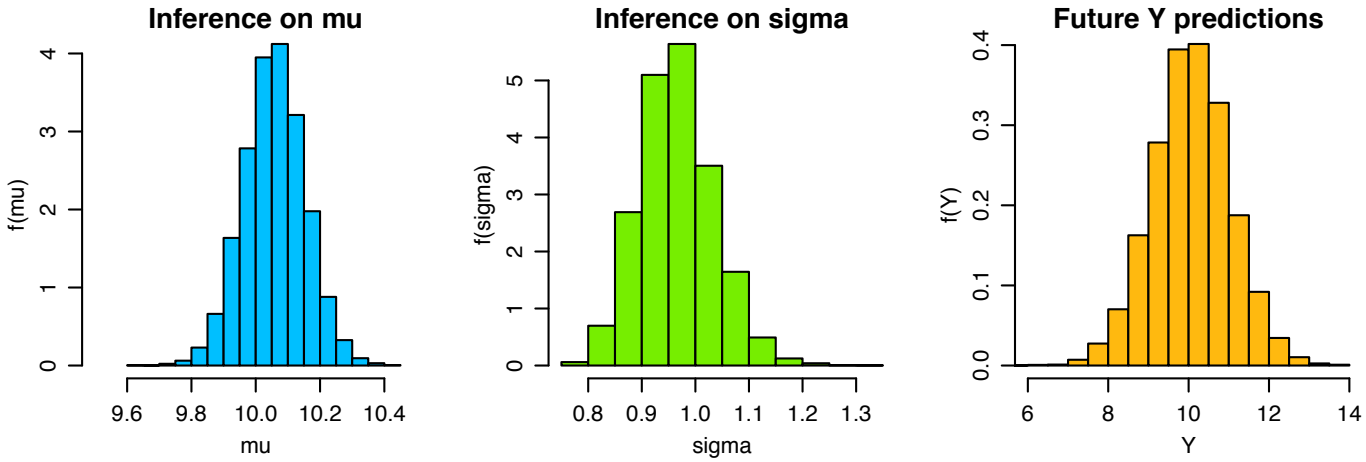
Ex 3: Normal inference

```
library(rjags)

set.seed(20190522)

#
# Generate the observed data
data_size <- 100
data_mu <- 10
data_sigma <- 1
data_obs <- rnorm(data_size, data_mu, data_sigma)

# - Specify the Generative Model with BUGS
model <- "s11_norm_pred.bug"

# Our data for the model
data <- NULL
data$X <- data_obs    # Set of observations

# Create the model and pass the parameters
jm <- jags.model(model, data)

# Update the Markov chain (Burn-in)
update(jm, 1000)

chain <- coda.samples(jm, c("mu", "sigma", "Y"), n.iter=10000)
print(summary(chain))
```

Ex 3: jags Normal results

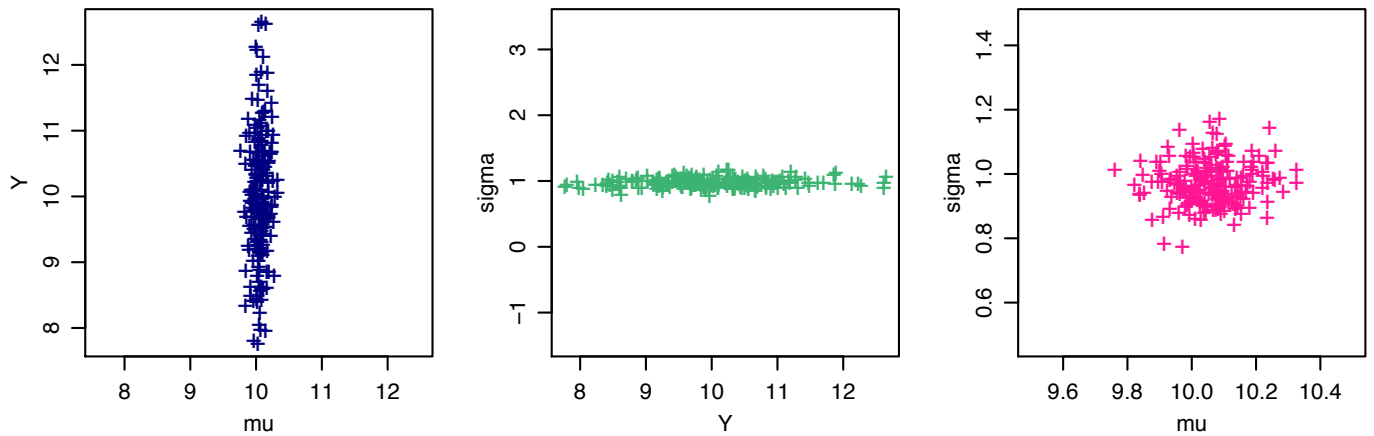
```
          Mean       SD  Naive SE Time-series SE
Y       10.0655  0.97382 0.0097382      0.0097382
mu      10.0577  0.09610 0.0009610      0.0009610
sigma    0.9656  0.06926 0.0006926      0.0006926

          2.5%     25%     50%     75%   97.5%
Y       8.1399  9.4278  10.067  10.711  11.983
mu      9.8668  9.9945  10.058  10.122  10.247
sigma   0.8398  0.9176   0.962   1.009   1.111
```

Note the different x-axis limits for μ and future Y predictions



Ex 3: jags Normal variables correlations

```
 Correlation matrix:
                Y           mu        sigma
Y       1.000000000 0.101044364 0.008388187
mu      0.101044364 1.000000000 0.002831711
sigma   0.008388187 0.002831711 1.000000000
```



Ex 4: Hook's law inference

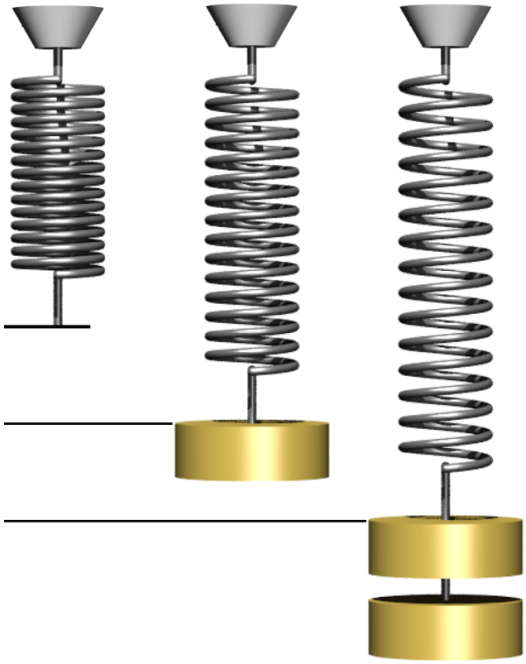
The Problem

- a spring with elastic constant k and mass m_{spring} is held vertically, at rest, under the influence of the Earth's gravitational field
- the lower end of the spring is loaded with equal mass discs and both spring elongation and oscillation periods are measured
- we want to infer, from the data, the spring elastic constant, k and, eventualy, the Earth gravity constant, g
- calling $l_\circ$, the unloaded spring length, we get at equilibrum

$$l = l_\circ + \frac{g}{k}(m_{spring} + n \cdot m_{disc})$$

  where m_{disc} is a disc mass and n the number of discs connected to the spring
- if one end of the spring is perturbated from the equilibrium position, it oscillates with period

$$T = 2\pi\sqrt{M/k} \text{ where } M = m_{spring} + n \cdot m_{disc}$$



measuring the oscillation period. as a function of the applied mass, it is possible to measure k, and from l, infer g

Ex 4: the collected data

- the following data come from https://www.roma1.infn.it/~dagos/BMS/node22.html

		I series		II series		III series	
n	M	l	$T \times 10$	l	$T \times 10$	l	$T \times 10$
	(g)	(mm)	(s)	(mm)	(s)	(mm)	(s)
0	63	0	-	0	-	0	-
1	142	0	-	0	-	0	-
2	221	0	-	0	-	0	-
3	300	14	5.01	16	5.09	16	5.19
4	379	32	5.57	33	5.66	33	5.68
5	458	49	6.24	51	6.27	51	6.34
6	536	66	6.78	68	6.82	69	6.94
7	615	85	7.28	86	7.33	87	7.28
8	694	103	7.79	103	7.81	103	7.86
9	773	119	8.13	121	8.31	121	8.24
10	852	137	8.63	139	8.77	139	8.70

- Notes: $l_\circ$, the unloaded spring rest length has been subtracted from data. Since the oscillation priod is below 1 s, the measurements have been taken for 10 periods

Ex 4: the BUGS model

```
model {
   # l Vs m
   for (i in 1:length(l)) {
      mu.l[i] <- c.l + m.l * (m_spring + (Nmin-1 + i) * m_disc);
      l[i] ~dnorm(mu.l[i], tau.l);
   }
   c.l ~ dnorm(0.0, 1.0E-4);
   m.l ~ dnorm(0.0, 1.0E-4);

   tau.l ~ dgamma(1.0E-3, 1.0E-6);
   sigma.l <- 1/sqrt(tau.l);

   # t vs sqrt(m)
   for (i in 1:length(t)) {
      mu.t[i] <- c.t + m.t * sqrt(m_spring + (Nmin-1 + i) * m_disc);
      t[i] ~dnorm(mu.t[i], tau.t);
   }
   c.t ~ dnorm(0.0, 1.0E-4);
   m.t ~ dnorm(0.0, 1.0E-4);

   tau.t ~ dgamma(1.0E-3, 1.0E-5);
   sigma.t <- 1/sqrt(tau.t);

   # k e g
   k <- 4*pi2 / (m.t*m.t)
   g <- m.l * k
}
```


Ex 4: data, init values, and model running

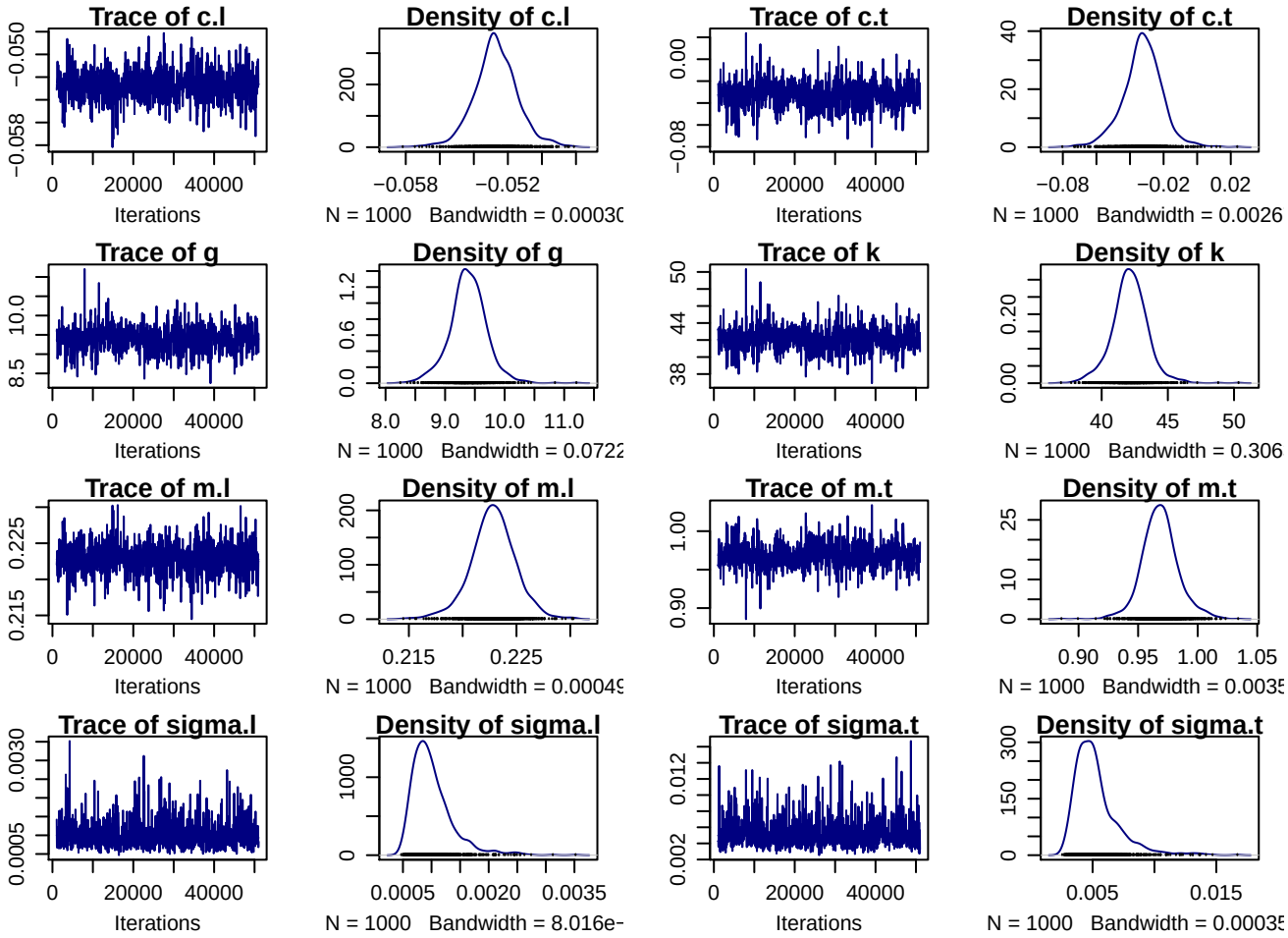
```
# Experimental data - Series I
data <- NULL
data$m_spring <- 0.063
data$m_disc   <- 0.0789
data$l        <- c(0.014, 0.032, 0.049, 0.066, 0.085, 0.103, 0.119, 0.137)
data$t        <- c(0.501, 0.557, 0.624, 0.678, 0.728, 0.779, 0.813, 0.863)
data$Nmin     <- 3
data$pi2      <- pi^2

# Generative model initial values
inits <- NULL
inits$c.l   <- 0
inits$m.l   <- 0
inits$tau.l <- 1000
inits$tau.t <- 1000
inits$m.t   <- 1

# Create the model and pass the parameters
jm <- jags.model("s13_spring.bug", data, inits)

# Update the Markov chain (Burn-in)
update(jm, 1000)

chain <- coda.samples(jm, c("c.l","m.l","sigma.l","c.t",
                            "m.t","sigma.t","k", "g"),
                      n.iter = 50000, thin = 50)
```

Ex 4: `jags` run results

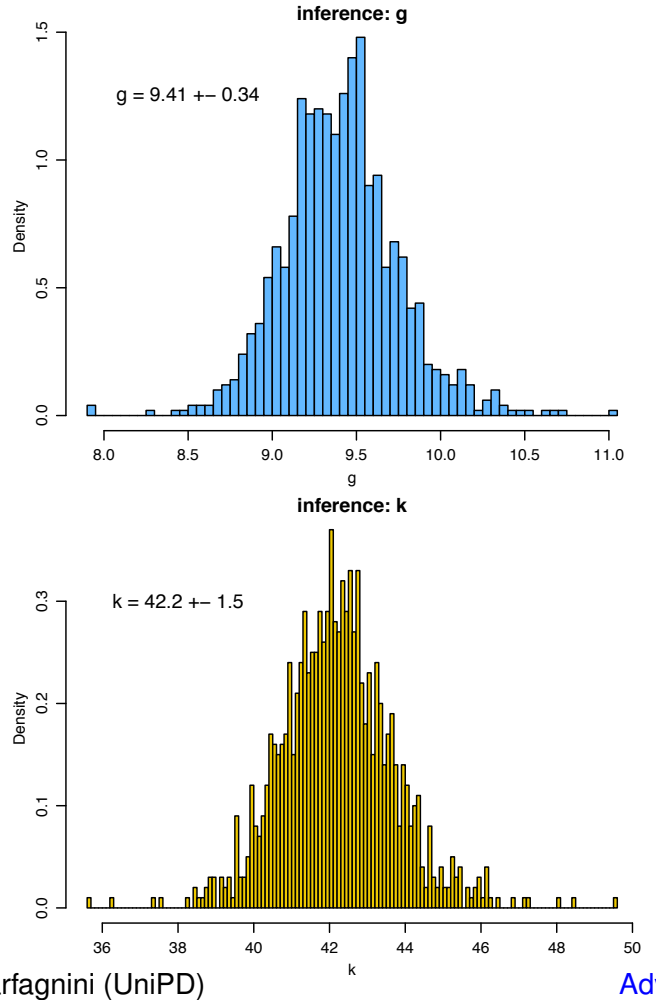
- the model produces the following output:

```
Iterations = 1050:51000 / Thinning interval = 50 / Number of chains = 1
Sample size per chain = 1000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:
            Mean         SD  Naive SE Time-series SE
c.l     -0.0527146 0.0012529 3.962e-05      3.962e-05
c.t     -0.0313581 0.0141587 4.477e-04      5.596e-04
g        9.4168338 0.3718726 1.176e-02      1.502e-02
k       42.2535756 1.6111324 5.095e-02      6.667e-02
m.l      0.2228622 0.0020382 6.445e-05      6.445e-05
m.t      0.9671275 0.0184045 5.820e-04      7.612e-04
sigma.l  0.0009943 0.0003414 1.080e-05      1.080e-05
sigma.t  0.0056743 0.0022657 7.165e-05      9.858e-05

2. Quantiles for each variable:
              2.5%         25%         50%         75%      97.5%
c.l     -0.0551276 -0.0534807 -0.0527385 -0.051996 -0.050116
c.t     -0.0622894 -0.0386805 -0.0312846 -0.023451 -0.003571
g        8.6816215  9.2058887  9.4212113  9.621637 10.127262
k       38.9954677 41.3512168 42.2117668 43.125521 45.426046
m.l      0.2188010  0.2216423  0.2229252  0.224129  0.226905
m.t      0.9322393  0.9567813  0.9670816  0.977093  1.006173
sigma.l  0.0005608  0.0007633  0.0009172  0.001125  0.001863
sigma.t  0.0030946  0.0042109  0.0050638  0.006621  0.011691
```


Ex 4: Markov chains plots



Ex 4: Inference, *g* and *k* results




```
sch <- summary(chain)
names(sch)
[1] "statistics" "quantiles"
"start"      "end"        "thin"
[6] "nchain"

> sch$statistics[,1:2]
                Mean
SD
c.l      -0.052742040 0.0013533757
c.t      -0.031768656 0.0131463627
g
9.407672249 0.3556819257
k        42.203673989 1.5186476007
m.l
0.222907961 0.0022597906
m.t
0.967640043 0.0173119299
sigma.l
0.001029015 0.0003875593
sigma.t
0.005411466 0.0018208406

sprintf("g␣=␣%.2f␣+-␣%.2f",
        sch$statistics["g","Mean"],
        sch$statistics["g","SD"])
[1] "g␣=␣9.41␣+-␣0.36"
```


References

Exercises

- http://www.roma1.infn.it/~dagos/prob+stat.html

Reference Books

- C.P. Robert and G.Casella, *Introducting Monte Carlo Methods with R*, Springer, 2010
- C.P. Robert and G.Casella, *Monte Carlo Statistical methods*, Springer, 1999
- D. Lunn, et. al., *The BUGS Book, a prctical introduction to Bayesian Analysis*, CRC Press, 2012

Additional Material

- `JAGS` user manual: https://sourceforge.net/projects/mcmc-jags/files/Manuals/
- `rjags` user manual https://cran.r-project.org/web/packages/rjags/rjags.pdf

MCMC with Stan

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Stat Lect 11

Stan

- Stan is another program that allows to specify statistical models, and to sample from a posterior distribution
- it is named after Stanislaw Ulam (1909-1984), a pioneer of Monte Carlo methods
- Stan means also Sampling Through Adaptive Neighborhoods
- it uses a different method than JAGS for generating Monte Carlo steps, called Hamiltonian Monte Carlo (HMC)
- HMC uses a sampling scheme creating proposal distributions pulled toward the modes of the posterior distribution instead of being symmetrical around the current position

References

- Stan web site: `https://mc-stan.org/`
- R interfaces to Stan:

  rstan: the R interface to Stan, `https://github.com/stan-dev/rstan`

  brms: an interface to fit Bayesian generalized non-linear multivariate multilevel models using Stan, `https://github.com/paul-buerkner/brms`

Using Stan

- to use Stan, the user writes a Stan program representing the statistical model
- the user specifies:
  - the parameters in the model
  - the target posterior density
- Stan code is compiled, run along with the data, and it provides a set of posterior simulations of the parameters
- Stan is written in C++ and interfaces with several popular data analysis languages are available (R, Python, MATLAB, Julia, ...)

Stan program structure

- a stan program is organized in blocks
- all blocks are optional ➜ an empty string is a valid Stan program, even if it will trigger a warning message from the Stan compiler
- the gory details on the Stan language can be found here: https://mc-stan.org/docs/2_29/reference-manual/index.html
- the three basics blocks are the following:

  for the declaration of variables that are read in as data

```
data {
  //  ... data ...
}
```

  these are the variables being sampled by Stan's samplers

```
parameters {
   // ... declarations ...
}
```

  it define the model. Here probability statements are given

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

Stan program structure

- a stan program is organized in blocks

```
functions {
  // ... function declarations and definitions ...
}


transformed data {
   // ... declarations ... statements ...
}


transformed parameters {
   // ... declarations ... statements ...
}
```


Chains initialization

- JAGS and Stan can automatically start the MCMC chains at default values
- but the efficiency of the MCMC process can be improved using appropriate starting values
- guideline: figure out values for the parameters in the model that are a reasonable description of the data, and of the posterior distribution
- a good choice: maximum likelihood estimate (MLE) of the parameters : i.e. maximize the probability of the data
- another approach is to start the chains at random points near the MLE

Example: Bernoulli trial

- the MLE of the parameter is $p = y/N$
- it maximizes $p^y(1-p)^{N-y}$

Three ways to initialize a chain in JAGS/Stan

- a single named list with a single initial point for the parameters ➜ all chains start there
- a list of lists, with as many sub-lists as chains ➜ with specific initial values in each sub-list
- define a function that returns initial values when called

rstan

Running `stan` in R

```
library(rstan)

#> Loading required package: StanHeaders
#> Loading required package: ggplot2
#> rstan (Version 2.21.3, GitRev: 2e1f913d3ca3)
#> For execution on a local, multicore CPU with excess RAM we recommend call
#> options(mc.cores = parallel::detectCores()).
#> To avoid recompilation of unchanged Stan programs, we recommend calling
#> rstan_options(auto_write = TRUE)
```

- as the startup message says, if you are using rstan locally on a multicore machine and have plenty of RAM to estimate your model in parallel, at this point execute

```
options(mc.cores = parallel::detectCores())

# on my machine:
parallel::detectCores()
#> [1] 12
```

- to avoid recompiling of C++ code every time (unless there are changes)

```
rstan_options(auto_write = TRUE)
```


Bernoulli example: JAGS

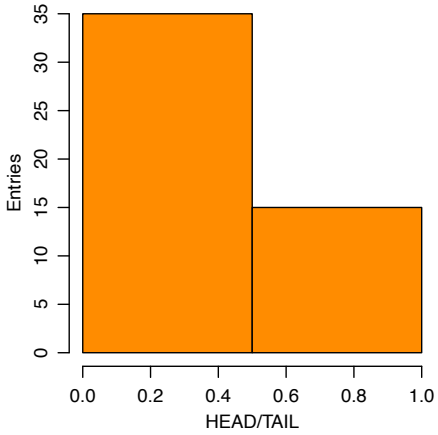
1) define the model and write it into a file

```
modelString = "
  model {
    for ( i in 1:Ntotal ) {
      y[i] ~ dbern( theta )
    }
    theta ~ dbeta( 1 , 1 )
  }
"
writeLines(modelString , con="jags_bern01_model.txt")
```

```
cat bern_jags-stan.csv
"y"
0
1
...
1
0
```

2) import data from a file and create a named-list

```
myData <- read.csv("bern_jags-stan.csv")
y <- myData$y
Ntotal <- length(y)
dataList <- list(y = y, Ntotal = Ntotal)
```



3) gets all the information into JAGS and lets it figure out appropriate samplers for the model

```
jagsModel <- jags.model(file="jags_model.txt",
                        data=dataList,
                        # inits=initsList,
                        n.chains=3,
                        n.adapt=500)
```

Bernoulli example: JAGS

4) run the chains for a burn-in period

```
update(jagsModel, n.iter=500)
```

the `update` function returns no values.
It does not record the sampled parameter values during the updating

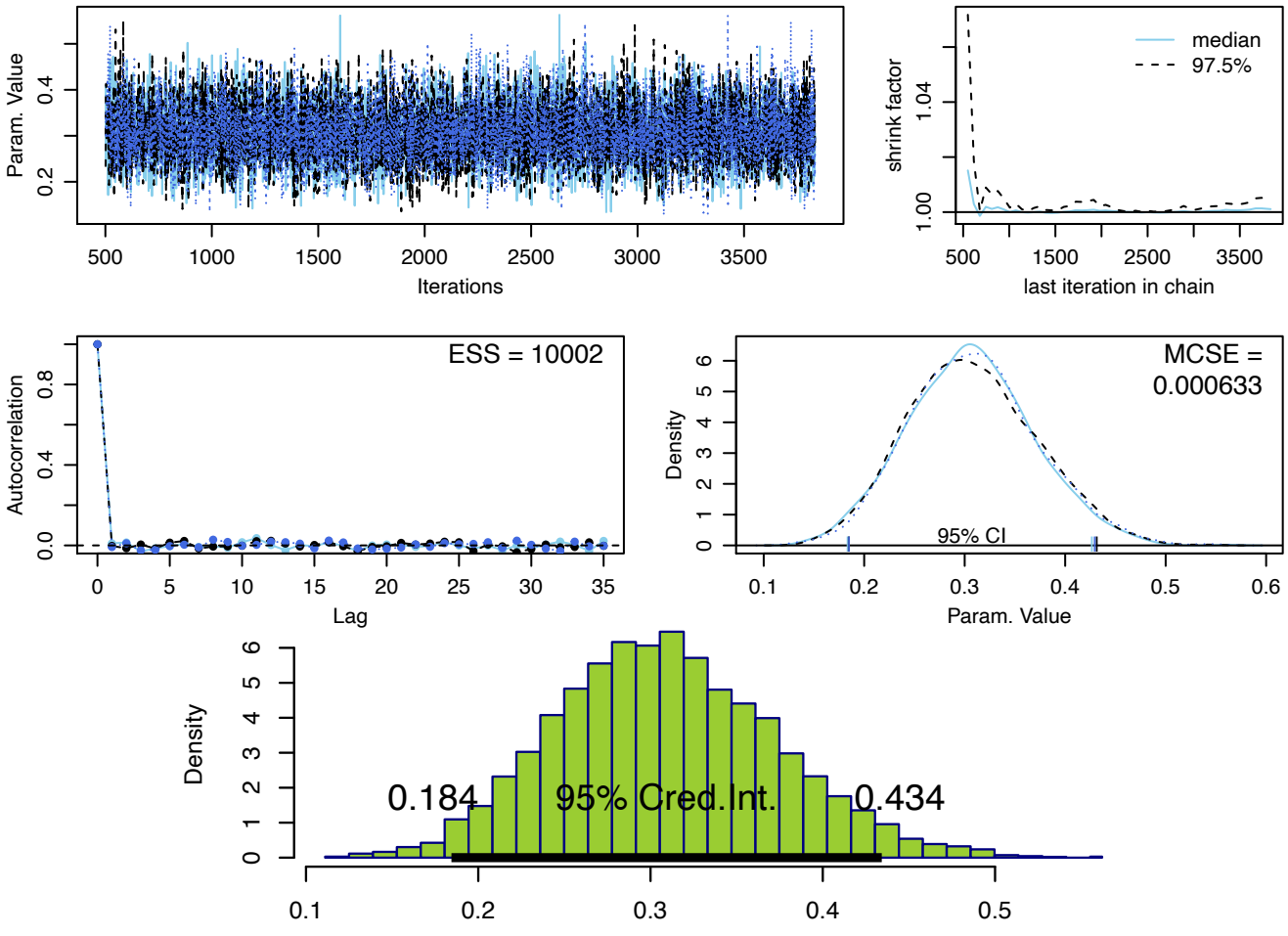
5) let JAGS generate MCMC samples that will be used to represent the posterior distribution

```
codaSamples <- coda.samples(jagsModel, variable.names=c("theta") ,
                            n.iter=3334)
```

the chains are arranged in a specialized format so that various functions from the coda package can be used to examine the chains

`variable.names` argument must be a vector of character strings

Bernoulli example: JAGS



Bernoulli example: JAGS

- the effective sample size is computed by coda by summing across chains

```
postSummary[,"ESS"] <- coda::effectiveSize(sampleVec)
```

- we can compute mean and median and the mode through the `density` function (which computes kernel density estimates)

```
postSummary[,"mean"]   <- mean(sampleVec)
postSummary[,"median"] <- median(sampleVec)

mcmcDensity <- density(sampleVec)
postSummary[,"mode"] <- mcmcDensity$x[which.max(mcmcDensity$y)]
```

- Credibility Interval:

```
sPts <- sort(sampleVec)
cIdx_step <- ceiling(credInt * length(sPts))
nCIs <- length(sPts) - ciIdx_step
ciWidth <- rep(0, nCIs)
for (j in 1:nCIs) {
  ciWidth[i] <- sPts[ j + ciIdx_step ] - sPts[j]
}
HDImin = sPts[ which.min( ciWidth ) ]
HDImax = sPts[ which.min( ciWidth ) + ciIdx_step ]
HDIlim = c( HDImin , HDImax )
```


Bernoulli example: Stan

1) define the model and write it into a file

   note that every line ends with ';' ➜ this is the C++ syntax, used in Stan

```
modelString = "
  data {
    int<lower=0> N;
    int y[N];
  }
  parameters {
    real<lower=0,upper=1> theta;
  }
  model {
    theta ~ beta(1,1);
    y ~ bernoulli(theta);
  }"
writeLines(modelString , con="stan_bern01_model.txt")
```

   Note: Stan allows and encourages vectorization of operations.

   a single line can indicate that every y_i follows the Bernoulli distribution:

```
y ~ bernoulli(theta);
```

Bernoulli example: Stan

2) translate the model to Stan C++ *Dynamic Shared Object (DSO) code*

```
stanDso <- stan_model(model_code = modelString)
```

once created, the DSO can be used for generating a Monte Carlo sample from the posterior distribution

3) *specify the data exactly as done for JAGS*

```
# Read the data and put it in a list
myData <- read.csv("bern_jags-stan.csv")
y <- myData$y   # The y values are in the column named y.
N <- length(y)  # Total number of coin flips
dataList <- list(y = y , N = N)
```

4) *generate the MC sample with the sampling command*

```
stanFit <- sampling(object=stanDso ,
                    data = dataList ,
                    chains = 3 ,
                    iter = 1000 ,
                    warmup = 200 ,
                    thin = 1)
```

Note: `warmup` *is used instead of* `burnin`

`iter` *is the total number of steps per chain*

Bernoulli example: Stan

3) RStan has methods for the standard R plot and summary commands and also its own version of the `traceplot` command

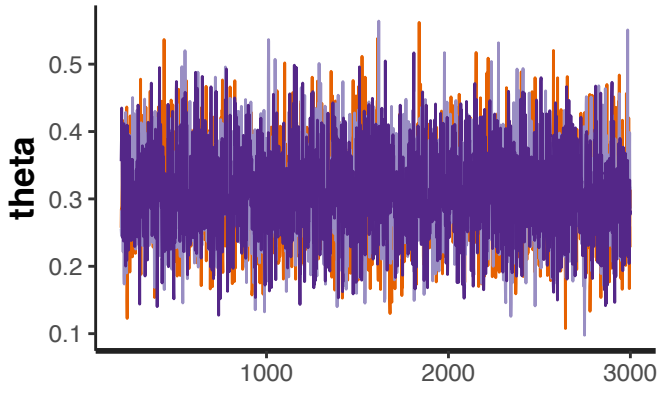
```
rstan::traceplot(stanFit,pars=c("theta"))
```

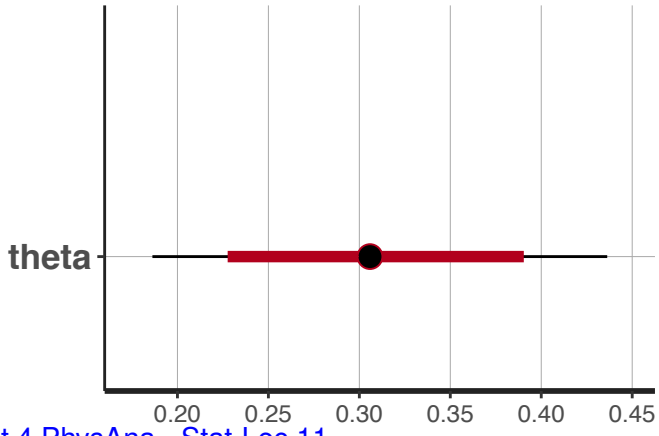
4) and a specialized plot version

```
plot(stanFit,pars=c("theta"))
```

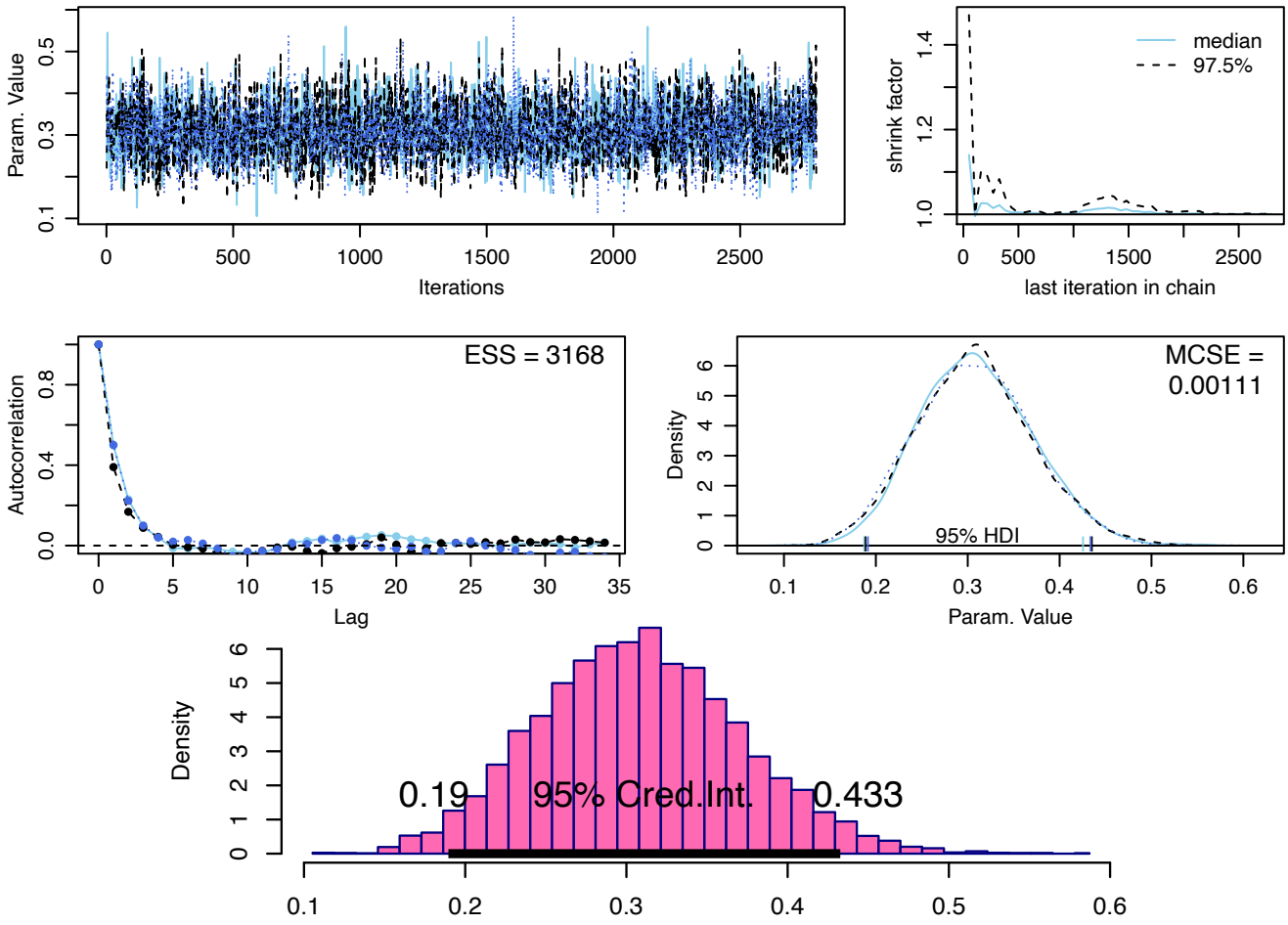
5) but it is always possible to transform the chain to a coda object list

```
mcmcCoda <- mcmc.list(lapply(1:ncol(stanFit),
                      function(x) { mcmc(as.array(stanFit)[,x,]) }))
class(mcmcCoda)
#> [1] "mcmc.list"
```





Bernoulli example: Stan



Example: Vaccine effectiveness

Randomized controlled trials (RCTs)

- volunteers are assigned randomly to receive an influenza vaccine or a placebo
- vaccine efficacy is measured by comparing the frequency of influenza illness in the vaccinated and the unvaccinated (placebo) groups
- the RCT study design minimizes bias that could lead to invalid study results
- vaccine allocation is usually double-blinded : neither the volunteers nor the researchers know if a given person has received vaccine or placebo

Observational Studies

- compare the occurrence of influenza among people who have been vaccinated compared to people not vaccinated
- vaccine effectiveness is the percent reduction in the frequency of illness among vaccinated people compared to people not vaccinated
- adjustment for factors (like presence of chronic medical conditions) are considered

References

1) Center for Disease and Control Prevention:
https://www.cdc.gov/flu/vaccines-work/effectivenessqa.htm

2) European Center for Disease and Control Prevention:
https://www.ecdc.europa.eu/en/covid-19/prevention-and-control/vaccines

Pfizer example: runjags

1) Pfizer announced that their Vaccine against COVID-19 is more than 90% effective: https://www.npr.org/sections/health-shots/2020/11/09/933006651/pfizer-says-experimental-covid-19-vaccine-is-more-than-90-effective?t=1622093442237
   they studied 43538 volunteers and found 94 evaluable cases of COVID-19
   the American Food and Drug adimistration set a minimum effectiveness level at 50%: https://www.fda.gov/media/139638/download
2) collect and organize the data from RCT

```
tot_vaccine <- 21999
tot_placebo <- 21539
patient <- c(rep("Vaccine", tot_vaccine),
             rep("Placebo", tot_placebo))

# Number of patients tested postive after RCT:
pos_vaccine <- 8
pos_placebo <- 86
tested  <- c(rep("Pos", pos_vaccine),
             rep("Neg", tot_vaccine - pos_vaccine),
             rep("Pos", pos_placebo),
             rep("Neg", tot_placebo - pos_placebo))

pfizer.tb <- tibble(tested = tested, patient=patient)
table(pfizer.tb[[2]], pfizer.tb[[1]])
            Neg   Pos
  Placebo 21453    86
  Vaccine 21991     8
```


Pfizer example: runjags

3) define the JAGS model : we do not use a flat prior, since we have pretty good information on how likely is to get COVID. We use a `beta(3,100)` prior:

```
modelString <- "
  model {
    for ( i in 1:Ntot ) {
      tested[i] ~ dbern( theta[patient[i]] )
    }
    for ( k in 1:Nclass ) {
      theta[k] ~ dbeta(3 , 100)
    }
  }"
```

4) organize our data in a list for usage in JAGS

```
dataList = list(

   tested = ifelse(pfizer.tb$tested == "Neg", 0, 1),
   patient = as.integer(factor(pfizer.tb$patient)),

   Ntot = nrow(pfizer.tb) ,
   Nclass = nlevels(factor(pfizer.tb$patient))

)
```

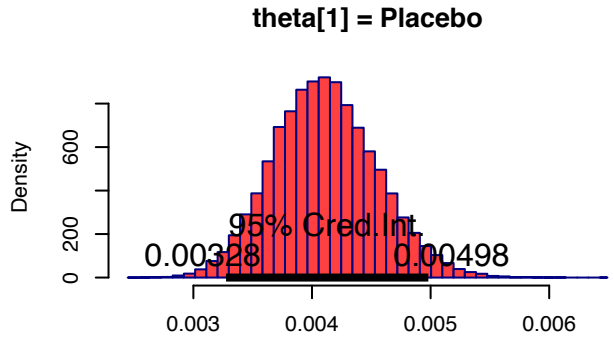
Pfizer example: runjags

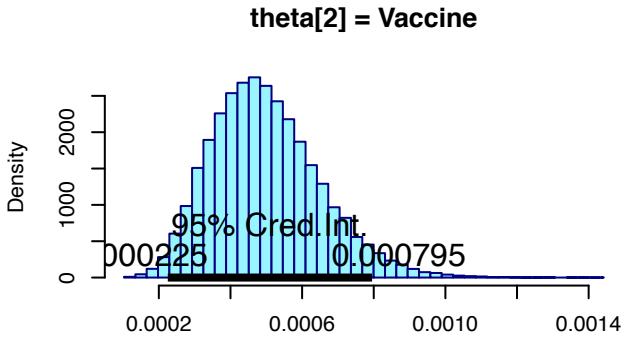
4) run JAGS

```
pfizer_chains <- run.jags(modelString,
                          sample = 15000,
                          n.chains = 4,
                          method = "parallel",
                          monitor = "theta",
                          data = dataList)
```

5) quick check JAGS run results:

```
summary(pfizer_chains)
             Lower95       Median     Upper95         Mean           SD Mode
theta[1] 0.003294610 0.0041017350 0.004989380 0.0041159597 0.0004345924 NA
theta[2] 0.000223376 0.0004822135 0.000792961 0.0004975752 0.0001496030 NA
                MCerr MC%ofSD SSeff        AC.10      psrf
theta[1] 2.172962e-06     0.5 40000 -0.002488248 0.9999861
theta[2] 7.389484e-07     0.5 40988 -0.004059210 1.0000766
```





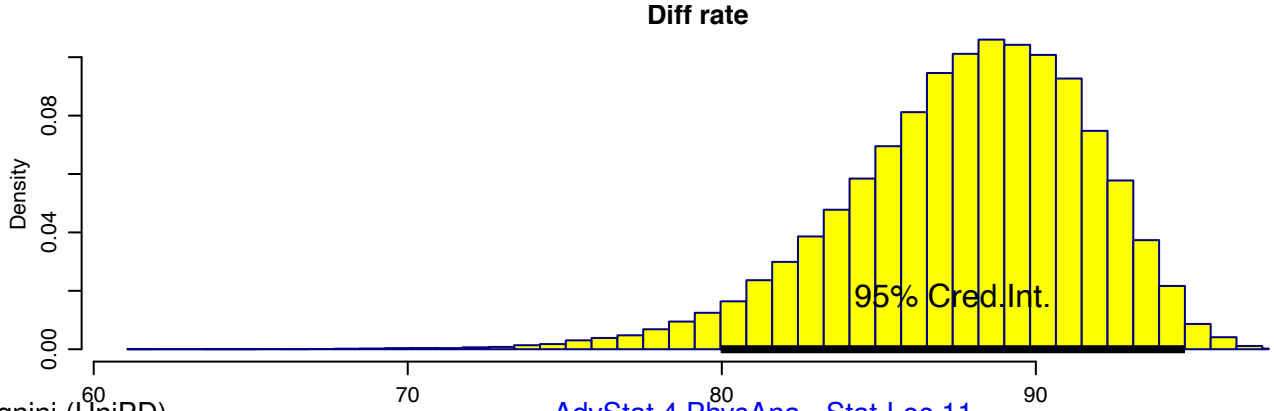
Pfizer example: runjags

1) have computed estimates for the rate of infection for both those who received the placebo and those who received the actual vaccine.
now want to investigate which is the percentage difference in infection rates

```
library(tidybayes)
pfizer_res   <- tidybayes::tidy_draws(pfizer_chains) %>%
                select(`theta[1]`:`theta[2]`) %>%
                rename(Placebo = `theta[1]`, Vaccine = `theta[2]`) %>%
                mutate(diff_rate = (Placebo - Vaccine) / Placebo * 100,
                Placebo_perc = Placebo * 100,
                Vaccine_perc = Vaccine * 100)
```

2) encapsulate the data in Coda so we can reuse our plotting function

```
allmcmc2 <- as.mcmc(pfizer_res, vars="diff_rate")
pt3 <- plotPosterior(allmcmc2[,"diff_rate"], 0.95, "yellow", "Diff␣rate")
              ESS     mean   median     mode CrIntLevel CrIntLow CrIntHigh
Param. Val. 60000 87.74049 88.17833 88.96141       0.95 79.96449  94.74905
```



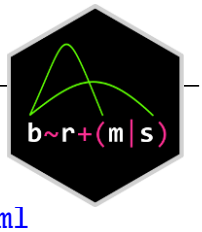
Pfizer example: testing Bayes factors

- finally, we want to check the bayes facotor to determine what are the chances that Pfizer vaccine is more than 50% effective
- we define a conservative prior assuming that Pfizer vaccine is 50% effective with a standard deviation of 15% ➜ we assume a Normal distribution with `mean = 50%` and `sd = 15%`
- we then compute the odds that with our prior and posterior the Vaccine is more that 50% effective
- we use the `bayestestR` package:

  https://github.com/easystats/bayestestR
- the odds given our data are more than 400,000:1 that the vaccine is more than 50% effective, a very strong evidence

```
prior <- bayestestR::distribution_normal(60000, mean = 50, sd = 15)

bayestestR::bayesfactor_parameters(pfizer_res$diff_rate,
                                   prior,
                   direction = "two-sided", null = 50)

#> Bayes Factor (Savage-Dickey density ratio)
#> BF
#> ---------
#> 6.425e+05

* Evidence Against The Null: [50]
```


The `brms` package



References:

- CRANL: https://cran.r-project.org/web/packages/brms/index.html
- GitHub: https://github.com/paul-buerkner/brms

From the `brms` GitHub pages

The brms package provides an interface to fit Bayesian generalized (non-)linear multivariate multilevel models using Stan, which is a C++ package for performing full Bayesian inference (see https://mc-stan.org/). Formula syntax is very similar to that of the package lme4 to provide a familiar and simple interface for performing regression analyses. A wide range of response distributions are supported, allowing users to fit – among others – linear, robust linear, count data, survival, response times, ordinal, zero-inflated, and even self-defined mixture models all in a multilevel context. Further modeling options include non-linear and smooth terms, auto-correlation structures, censored data, missing value imputation, and quite a few more. In addition, all parameters of the response distribution can be predicted in order to perform distributional regression. Multivariate models (i.e., models with multiple response variables) can be fit, as well. Prior specifications are flexible and explicitly encourage users to apply prior distributions that actually reflect their beliefs. Model fit can easily be assessed and compared with posterior predictive checks, cross-validation, and Bayes factors.

brms example: Moderna age data

- Moderna released their RCT data grouped by age, considering older (age > 65 yr) and younger (≤ 65 yr) patients

Treatment	Age	Positive	Negative	Total
Placebo	≤ 65	79	8271	8350
Placebo	> 65	11	4494	4505
Vaccine	≤ 65	1	8293	8294
vaccine	> 65	4	4497	4501

- we treat COVID-19 outcomes as simple bernoulli events (again)
- brm allows us to specify the outcomes aggregated: i.e. the number of those that become infected according to their conditions (treatment and age)

```
age <- c(rep("lt65", 8350), rep("Older", 4505),
         rep("lt65", 8294), rep("Older", 4501))
treatment <- c(rep("Placebo", 8350), rep("Placebo", 4505),
         rep("Vaccine", 8294), rep("Vaccine", 4501))
tested <- c(rep("Pos", 79), rep("Neg", 8271),
            rep("Pos", 11), rep("Neg", 4494),
            rep("Pos", 1),  rep("Neg", 8293),
            rep("Pos", 4),  rep("Neg", 4497))

moderna_tb <- tibble(age = age, tested = tested,
                     treatment = treatment)
```


brms example: Moderna age data

- running brms

```
moderna_bf <- brm(data = moderna_tb,
                  family = bernoulli(link = logit),
                  tested ~ age + treatment + age:treatment,
                  iter = 12500, warmup = 500, chains = 4, cores = 12,
                  control = list(adapt_delta = .99, max_treedepth = 12),
                  seed = 9,
                  file = "moderna_long")

summary(moderna_bf)
 Family: bernoulli
  Links: mu = logit
Formula: tested ~ age + treatment + age:treatment
   Data: moderna_tb (Number of observations: 25650)
Samples: 4 chains, each with iter = 12500; warmup = 500; thin = 1;
         total post-warmup samples = 48000

Population-Level Effects:
                        Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
Intercept                  -4.66      0.11    -4.88    -4.44 1.00    39093
ageOlder                   -1.39      0.33    -2.07    -0.79 1.00    16404
treatmentVaccine           -4.74      1.17    -7.51    -2.95 1.00    10026
ageOlder:treatmentVaccine   3.65      1.32     1.42     6.66 1.00    10513
                        Tail_ESS
Intercept                  30740
ageOlder                   18439
treatmentVaccine            9939
ageOlder:treatmentVaccine  10761
```

a note on the logistic model

- a logistic model (logit) is used to model the probability of binary dependent variables
- let's have a model with one predictor x and one binary response variable y. y can be 0 or 1 and follows a Bernoulli probability
- we assume a linear relationship between the probability of success $p = P(y = 1)$ and the log-odds l:

$$l = \log \frac{p}{1-p} = a + b \cdot x$$

- the inverse transform

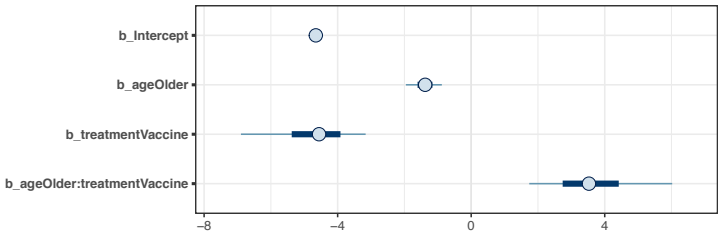
$$\frac{p}{1-p} = \exp a + b \cdot x$$

- gives p through of a Sigmoid function

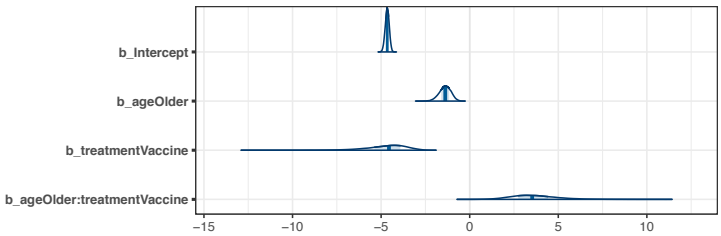
$$p = \Sigma(a + b \cdot x) = \frac{1}{1 - \exp a + b \cdot x}$$

brms example: Moderna age data

```
mcmc_plot(moderna_bf)
```




```
mcmc_plot(moderna_bf, type='areas')
```




```
bayestestR::describe_posterior(
              moderna_bf,
              ci=0.95,
              test=c("p_direction"),
              centrality="MAP")

Summary of Posterior Distribution

Parameter             |   MAP |         95% CI |     pd |  Rhat |
ESS
-------------------------------------------------------------------------
(Intercept)           | -4.66 | [-4.88, -4.43] |   100% | 1.000 | 38859.00
ageOlder              | -1.38 | [-2.04, -0.76] |   100% | 1.000 | 16170.00
treatmentVaccine      | -4.25 | [-7.07, -2.71] |   100% | 1.001 |  8919.00
ageOlder:treatVaccine |  3.21 | [ 1.22,  6.35] | 99.97% | 1.001 |  9677.00
```

Bayesian Networks

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Stat Lect. 12

The truck driver example

- a truck driver is due to make a 800 km trip
- we analyze the risk of his falling asleep while driving

- there may be causal relationships between:

1) the driver's sleep (did he sleep well and for more than 7 hours the night before ?)
2) his perceived fatigue (does he feel tired at the beginning of the trip ?)
3) the risk of falling asleep while driving

Current situation

- the truck driver feels tired at the beginning of the trip

➜ weather or not this is due to a bad sleep the night before, or any other reason, is of no use to evaluate the risk

- the driver feels perfectly fit before starting to drive

➜ the quality of sleep the night before has no influence on his current condition

➜➜ the risk of falling asleep is conditionally independent of the quality of his sleep, given the driver's current fatigue

The truck driver example (2)

Our formal model:

- we model it with binary variables which tell us if
  - X_1: the truck driver slept well the night before
  - X_2: he feels tired at the beginning of the trip
  - X_3: he will fall asleep while driving
- but, the quality of sleep the night before has no influence on his current condition

$$P\left(X_3 \mid X_1, X_2\right) = P\left(X_3 \mid X_2\right)$$

therefore:

$$\begin{aligned} P\left(X_1, X_2, X_3\right) &= P\left(X_1 \mid X_2, X_3\right) P\left(X_2, X_3\right) \\ &= P\left(X_3 \mid X_2\right) P\left(X_2 \mid X_1\right) P\left(X_1\right) \end{aligned}$$

The doped athlete example

- during a sports competition, each athlete undergoes two doping tests
  - test A is a blood test
  - test B a urine test
- the two tests are carried out in two different laboratories, no contact between the two labs is possible

Current situation

- the results of the two tests are not independent variables:
  - if test A is positive ➜ the participant is likely to have used a banned product ➜ test B will probably be also positive
  - the athlete has taken a detectable substance

➜ tests A and B can be considered independent, since the two laboratories use different detection methods

  - the athlete has NOT taken any prohibited substance

➜ tests A and B can be again considered independent ➜ they may give a negative response according to the test efficacy

➜➜ the results of both tests are conditionally independent, given the status of the tested athlete

The doped athlete example (2)

Our formal model:

- we model it with binary variables which tell us if
  - X_1: the athlete is *clean* or not
  - X_2: the result of test A
  - X_3: the result of test B
- let's write the conditional probability

$$P\left(X_3 \mid X_2, X_1\right) = P\left(X_3 \mid X_1\right)$$

➜ knowing whether the athlete has taken the substance is enough information to estimate the chances of test B being positive

- the symmetric equation holds

$$P\left(X_2 \mid X_3, X_1\right) = P\left(X_2 \mid X_1\right)$$

combining the results:

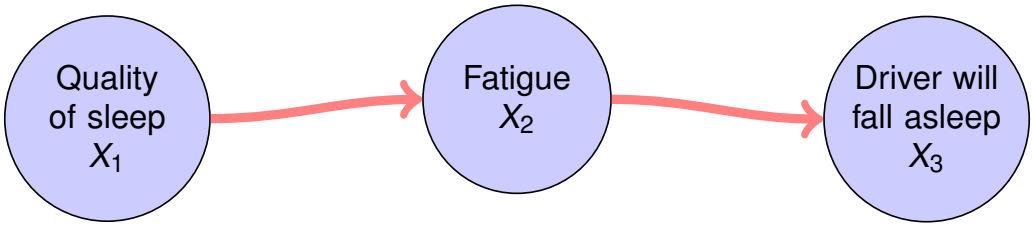
$$\begin{aligned} P\left(X_1, X_2, X_3\right) &= P\left(X_3 \mid X_2, X_1\right) P\left(X_2 \mid X_1\right) P\left(X_1\right) \\ &= P\left(X_3 \mid X_1\right) P\left(X_2 \mid X_1\right) P\left(X_1\right) \end{aligned}$$

Discrete Bayesian Networks

- both examples can be expressed in a Bayesian Network

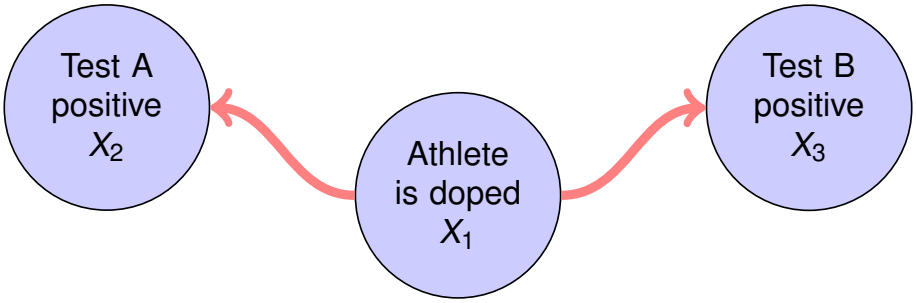
The Truck Driver Example

- there is an influence of variable X_1 on variable X_2, and of variable X_2 on variable X_3



The Doped Athlete Example

- X_2 and X_3 are conditionally independent given X_1



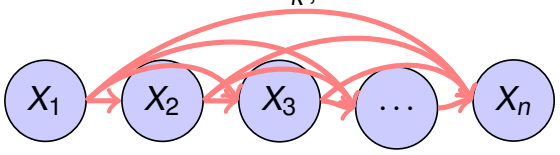
Bayesian Network

Definition

- given
  - n random variables $X_1, X_2, \ldots, X_n$
  - a directed acyclic graph with n numbered nodes
  - suppose node j of the graph associated to the X_j variable
- the graph is a Bayesian network, representing the variables $X_1, X_2, \ldots, X_n$, if

$$P(X_1, X_2, \ldots, X_n) = \prod_{j=1}^{n} P\left(X_j \mid \text{parents}(X_J)\right)$$

where $\text{parents}(X_j)$ denotes the set of all variables X_k, such that there is an arc from node k to node j



Proposition

- any joint probability distribution may be represented by a Bayesian network

$$\begin{aligned} P\left(X_1 \mid X_2, \ldots, X_n\right) &= P(X_1)\, P\left(X_2 \ldots X_n \mid X_1\right) \\ &= P(X_1)\, P(X_2 \ldots X_1)\, P\left(X_3 \ldots X_n \mid X_1 X_2\right) \\ &= \ldots \\ &= P(X_1)\, P\left(X_2 \mid X_1\right) P\left(X_3 \mid X_2, X_1\right) \ldots P\left(X_n \mid X_1, X_2, \ldots, X_{n-1}\right) \end{aligned}$$

Bayesian Networks in R: case study

The transportation means survey

- let's consider an hypothetical survey whose aim is to investigate the usage patterns of different means of transport, with a focus on private cars and public trains or buses
- each regular commuting individual fills a questionnaire on the following six discrete variables

  Age (`A`): below 30 (`young`), between 30 and 60 (`adult`) greater than 60 (`senior`)

  Sex (`S`): male (M) or female (F)

  Education (`E`): highest individual degree between high school (`high`) and university or higher (`uni`)

  Occupation (`O`): weather the individual is an employee (`empl`) or a self-employed worker (`self`)

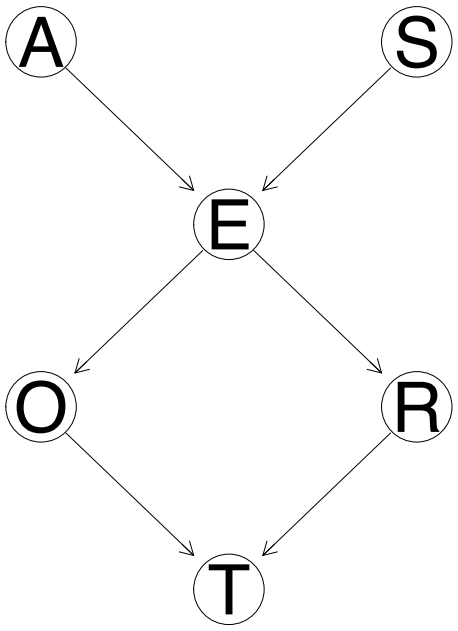
  Residence (`R`): the size of the city the individual lives in, a (`small`) or (`big`) town

  Travel (`T`): the means of transport flavored by the individual, `car`, `train` or `bus`
- the variables can be grouped into demographic indicators (`Age` and `Sex`), socioeconomic indicators (`Education`, `Occupation` and `Residence`) and the target of the survey (`Travel`)

Travel Survey in bnlearn

- we use the bnlearn R package to build a directed acyclic graphs (DAG) that describes the network

```
library(bnlearn)

tus_dag <- empty.graph(nodes = c("A", "S",
                                 "E", "O", "R", "T"))
tus_dag <- set.arc(tus_dag, from = "A", to = "E")
tus_dag <- set.arc(tus_dag, from = "S", to = "E")
tus_dag <- set.arc(tus_dag, from = "E", to = "O")
tus_dag <- set.arc(tus_dag, from = "E", to = "R")
tus_dag <- set.arc(tus_dag, from = "O", to = "T")
tus_dag <- set.arc(tus_dag, from = "R", to = "T")
```

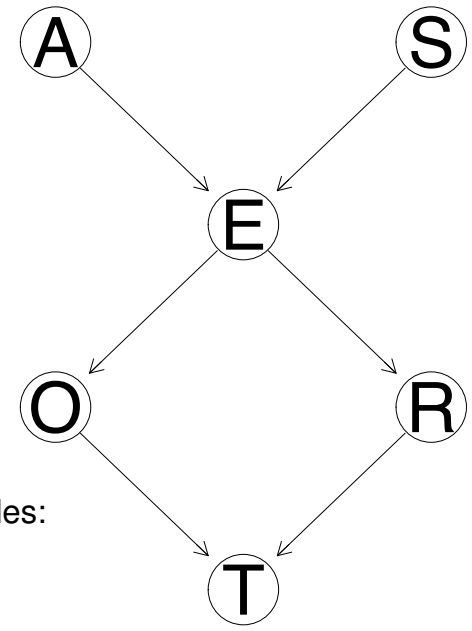



```
tus_dag
#> Random/Generated Bayesian network
#> model:
#>  [A][S][E|A:S][O|E][R|E][T|O:R]
#> nodes:                                 6
#> arcs:                                  6
#>   undirected arcs:                     0
#>   directed arcs:                       6
#> average markov blanket size:           2.67
#> average neighbourhood size:            2.00
#> average branching factor:              1.00
#> generation algorithm:                  Empty
```


Exploring the Travel Survey DAG bnlearn

- show the nodes of a graph:

```
nodes(tus_dag)
[1] "A" "S" "E" "O" "R" "T"
```



- examine the nodes close to a target

```
# The neighbourhood of 'E'
nbr(tus_dag, "E")
#> [1] "A" "S" "O" "R"
parents(tus_dag, "E")
#> [1] "A" "S"
children(tus_dag, "E")
#> [1] "O" "R"
```

- look for roots (no parents) and leaves (no children) nodes:

```
root.nodes(tus_dag)
#> [1] "A" "S"
leaf.nodes(tus_dag)
#> [1] "T"
```

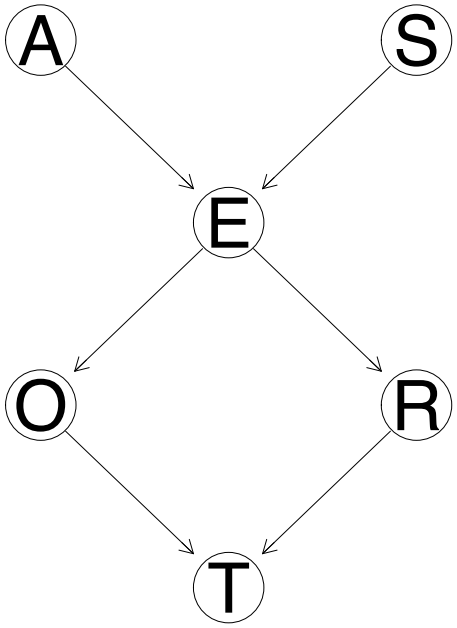
- and plot the graph

```
library(Rgraphviz)
graphviz.plot(plant)
```

Travel Survey in bnlearn

- another way to create the network is using the model formula interface provided by `modelstring`

```
dag <- empty.graph(nodes = c("A", "S", "E", "O", "R", "T"))

dag2 <- model2network("[A][S][E|A:S][O|E][R|E][T|O:R]")

all.equal(tus_dag, dag2)
[1] TRUE
```



- we now define the levels of the nodes, i.e. the discrete values defined on a non-ordered set

```
A_lvl <- c("young", "adult", "old")
S_lvl <- c("M", "F")
E_lvl <- c("high", "uni")
O_lvl <- c("emp", "self")
R_lvl <- c("small", "big")
T_lvl <- c("car", "train", "other")
```


Travel Survey in bnlearn

- To complete the BN modelling the survey➜ specify the joint probabilities

```
A_prob <- array(c(0.30, 0.50, 0.20), dim = 3,
                dimnames = list(A = A_lvl))
S_prob <- array(c(0.60, 0.40), dim = 2, dimnames = list(S = S_lvl))

O_prob <- array(c(0.96, 0.04, 0.92, 0.08), dim = c(2, 2),
                dimnames = list(O = O_lvl, E = E_lvl))
R_prob <- matrix(c(0.25, 0.75, 0.20, 0.80), ncol = 2,
                 dimnames = list(R = R_lvl, E = E_lvl))

E_prob <- array(c(0.75, 0.25, 0.72, 0.28, 0.88, 0.12,
                  0.64, 0.36, 0.70, 0.30, 0.90, 0.10), dim = c(2, 3, 2),
                dimnames = list(E = E_lvl, A = A_lvl, S = S_lvl))
T_prob <- array(c(0.48, 0.42, 0.10, 0.56, 0.36, 0.08,
                  0.58, 0.24, 0.18, 0.70, 0.21, 0.09), dim = c(3, 2, 2),
                  dimnames = list(T = T_lvl, O = O_lvl, R = R_lvl))
```

- and finally associate the probabilities to the BN model

```
cpt <- list(A = A_prob, S = S_prob, E = E_prob, O = O_prob,
            R = R_prob, T = T_prob)
bn <- custom.fit(tus_dag, cpt)
```

- variables that are not linked by an arc are conditionally independent

➜ we can factorise the global distribution

$$P(A,S,E,O,R,T) = P(A)\cdot P(S)\cdot P(E \mid A,S)\cdot P(O \mid E)\cdot P(R \mid E)\cdot P(T \mid O,R)$$

Travel Survey : estimate the probability table

- we knew the DAG and the probabilities defining the BN ➜ BNs are used as expert systems
- but in most cases, the parameters of the local distributions will be inferred (i.e. learned) from the observed sample

```
survey <- read.table("survey.txt", header = TRUE,
                     stringsAsFactors = TRUE)

head(transp_survey, n=3)
#>       A     R    E   O S     T
#> 1 adult   big high emp F   car
#> 2 adult small  uni emp M   car
#> 3 adult   big  uni emp F train

tail(transp_survey, n=3)
#>          A   R    E   O S     T
#> 498   old big high emp M train
#> 499 adult big high emp F other
#> 500 adult big high emp M other
```

- the conditional probabilities can be estimated looking at the corresponding empirical frequencies in the data set

$$P\left(O = \text{emp} \,\middle|\, E = \text{high}\right) = \frac{P(O = \text{emp}, E = \text{high})}{P(E = \text{high})}$$
$$= \frac{\text{number of observations for which O = emp and E = high}}{\text{number of observations for which E = high}}$$

Travel Survey : estimate the probability table

- the `bn.fit()` function computes the classic frequentist and maximum likelihood estimates from the data
- it complements the `custom.fit()` function which constructs a BN using a set of custom parameters specified by the user

```
travel_dag <- model2network("[A][S][E|A:S][O|E][R|E][T|O:R]")

bn.mle <- bn.fit(travel_dag, data = transp_survey,
                 method = "mle")
```

- as an alternative, the same conditional probabilities can be estimated in the Bayesian framework, using their posterior distributions

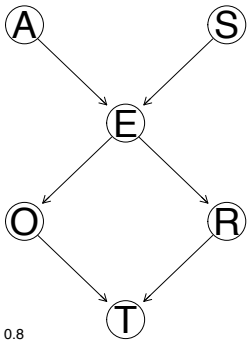
```
bn.bayes <- bn.fit(travel_dag, data = transp_survey,
                   method = "bayes", iss = 10)
```

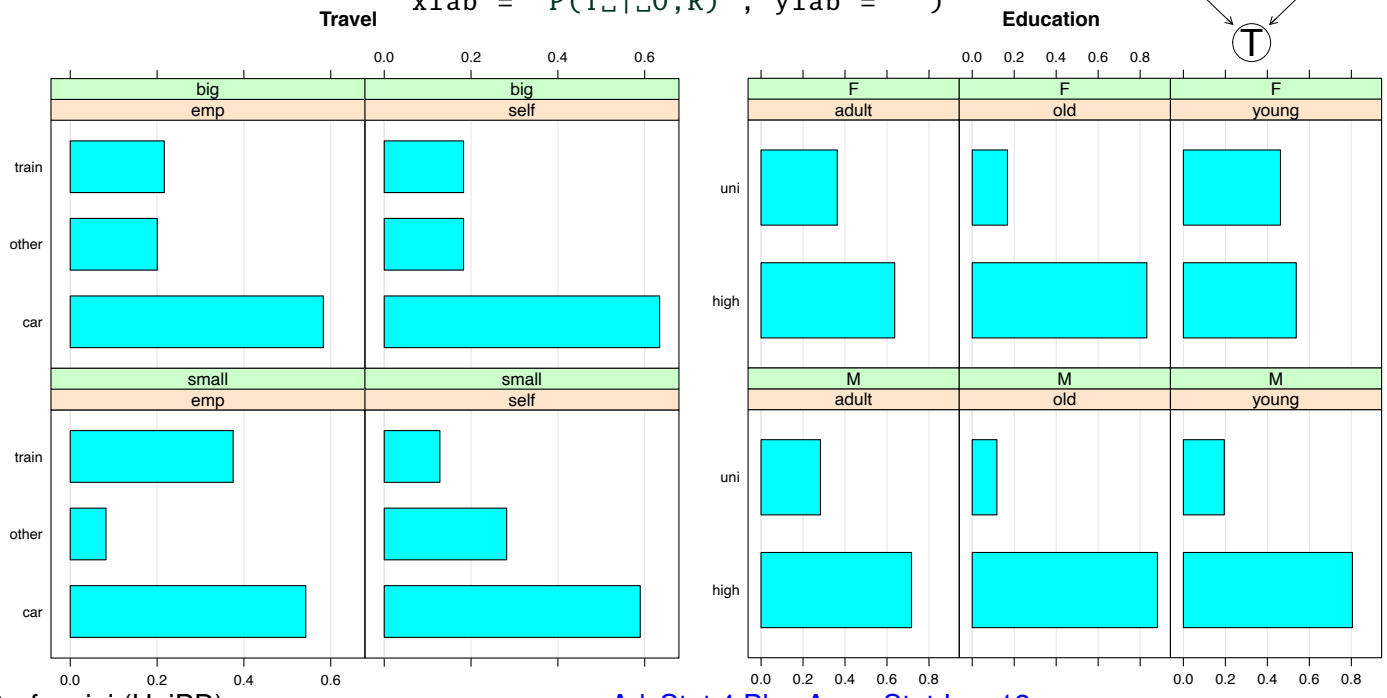
- the `iss` optional argument (imaginary sample size) determines how much weight is assigned to the prior distribution compared to the data when computing the posterior
- `iss` is typically chosen between 1 and 15, to allow the prior distribution to be easily dominated by the data (see bnlearn documentation)

Plotting the probability between links

- we can plot the probabilities associated to each link of the network

```
bn.fit.barchart(bn.bayes$E, main = "Education",
                xlab = "P(E␣|␣A,S)", ylab = "")

bn.fit.barchart(bn.bayes$T, main = "Travel",
                xlab = "P(T␣|␣O,R)", ylab = "")
```





Investigating a DAG structure from data

- so far we have assumed that the DAG underlying the BN is known

➜ we rely on prior knowledge on the phenomenon to decide which arcs are present in the graph and which are not ➜ expert system

- the structure of the DAG itself may be the object of our investigation

➜ in genetics and systems biology it is common to reconstruct the molecular pathways and networks underlying complex diseases and metabolic processes

- learning the DAG of a BN is a complex task

1) the space of the possible DAGs is very big ➜ it grows exponentially with the number of nodes
2) this space is very different from real spaces : it is not continuous and has a finite number of elements ➜ ad-hoc algorithms are required to explore it

- two classes of statistical criteria used to evaluate DAGs

1) conditional independence tests
2) network scores

Conditional Independence Test

- arcs encode a probabilistic dependence ➜ conditional independence tests can be used to verify if that probabilistic dependence is supported by the data

E ➜ T

- H$_\circ$: Travel is independent of Education
- use the log-likelihood ratio G^2

$$P\left(T, E \,\middle|\, O, R\right) = \sum_{t \in T} \sum_{e \in E} \sum_{k \in O \times R} n_{tek} \log \frac{n_{tek}\, n_{++k}}{n_{t+k}\, n_{+ek}},$$

  - the use of a "+" subscript denotes the sum over that index :
  - n_{t+k} = sum over the second index, $e \in E$
  - n_{++k} = sum over the first ($t \in T$) and second ($e \in E$) indexes, respectively
- or Pearson's X^2:

$$P\left(T, E \,\middle|\, O, R\right) = \sum_{t \in T} \sum_{e \in E} \sum_{k \in O \times R} \frac{(n_{tek} - m_{tek})^2}{m_{tek}} \quad \text{with } m_{tek} = \frac{n_{t+k}\, n_{+ek}}{n_{++k}}$$

- both tests have an asymptotic χ^2 distribution under H$_\circ$

Travel Survey: evaluate DAG arcs

Mutual Information, log-likelihood ratio G^2

```
transp_survey <- read.table("survey.txt", header = TRUE,
                            stringsAsFactors = TRUE)

ci.test("T", "E", c("O", "R"), test = "mi", data = transp_survey)

#>       Mutual Information (disc.)
#>
#> data:  T ~ E | O + R
#> mi = 9.8836, df = 8, p-value = 0.2733
#> alternative hypothesis: true value is greater than 0
```

Pearson's X^2

```
ci.test("T", "E", c("O", "R"), test = "x2", data = transp_survey)

#>       Pearson's X^2
#>
#> data:  T ~ E | O + R
#> x2 = 8.2375, df = 8, p-value = 0.4106
#> alternative hypothesis: true value is greater than 0
```

- both tests return very large p-values ➜ the dependence relationship encoded by $E \times T$ is not significant given the current DAG structure

Network Scores

- network scores focus on the DAG as a whole
- they provide a statistical measurement of how well the DAG mirrors the dependence structure of the data

Bayesian Information criterion (BIC)

$$
\begin{aligned}
\text{BIC} &= \log P\left(A, S, E, O, R, T\right) - \frac{d}{2} \log n \\
&= \log P(A) - \frac{d_A}{2} \log n + \log P(S) - \frac{d_S}{2} \log n \\
&+ \log P(E \mid A, S) - \frac{d_E}{2} \log n + \log P(O \mid E) - \frac{d_O}{2} \log n \\
&+ \log P(R \mid E) - \frac{d_R}{2} \log n + \log P(T \mid O, R) - \frac{d_T}{2} \log n
\end{aligned}
$$

- with n the sample size
- d the number of parameters of the network
- d_A, d_S, d_E, d_O, d_R and d_T the number of parameters associated with each node

Travel Survey: evaluate scores

several scores are available in `bnlearn`:

- Bayesian Information criterion (BIC)

```
score(travel_dag, data = transp_survey, type = "bic")
#> [1] -2012.687
```

- Bayesian Dirichlet equivalent uniform (BDe)

```
score(travel_dag, data = transp_survey, type = "bde")
#> [1] -2015.647
```

- using such scores it is possible to compare different DAGs and investigate which of them fits the data better

```
nparams(travel_dag, transp_survey)
[1] 21

dag_new <- set.arc(travel_dag, from = "E", to = "T")
nparams(dag_new, transp_survey)
[1] 29

score(dag_new, data = transp_survey, type = "bic")
[1] -2032.603
```

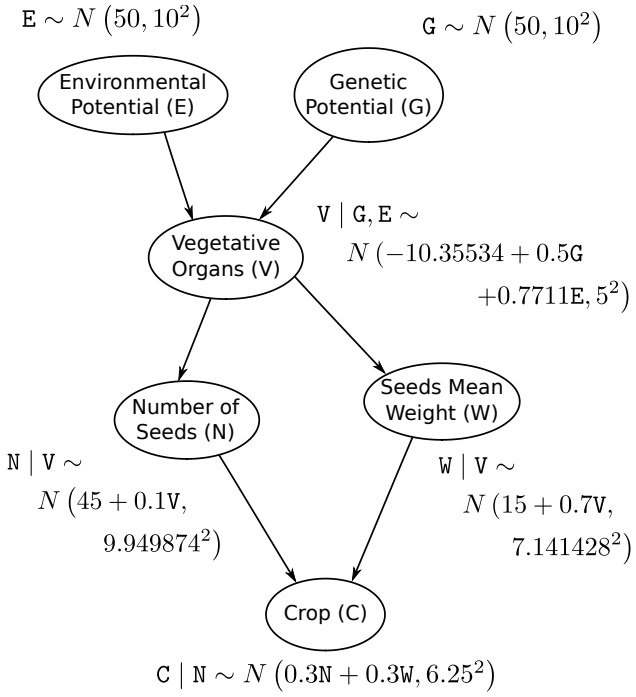
➜ adding E➜ T is not beneficial
the increase in $\log P(A, S, E, O, R, T)$ is not sufficient to offset the heavier penalty from the additional parameters

Continuous Bayesian Networks

- focus on modelling continuous data under a multivariate Normal distribution hypothesis
- we are interested in the analysis of a particular plant and study
  - the potential of the plant and the environment
  - the production of vegetative mass
  - the harvested grain mass, i.e. the *crop*

The plant network

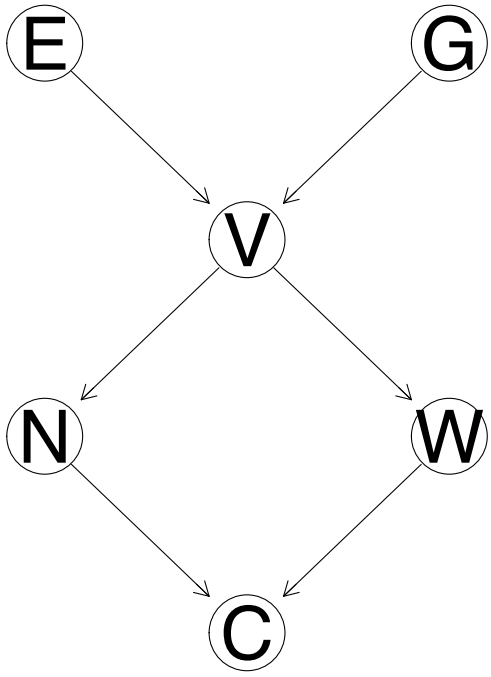
- environmental potential `(E)`
- genetic potential `(G)`
- vegetative organs `(V)`
- number of seeds `(N)`
- seeds mean weight `(W)`
- crop `(C)`



The Plant example: network structure

- let's build the DAG bayesian network:

```
plant_dag <- model2network("[G][E][V|G:E][N|V][W|V][C|N:W]")
plant_dag

#>   Random/Generated Bayesian network

#>   model:
#>    [E][G][V|E:G][N|V][W|V][C|N:W]
#>   nodes:                                6
#>   arcs:                                 6
#>     undirected arcs:                    0
#>     directed arcs:                      6
#>   average markov blanket size:          2.67
#>   average neighbourhood size:           2.00
#>   average branching factor:             1.00
#>
#>   generation algorithm:                 Empty

nparams(plant)
#> [1] 18
```



The Plant example: connection probabilities

- to make quantitative statements about the behavior of the variables in the BN, we need to completely specify their joint probability distribution
- if we are modelling n variables we must specify n means, n variances and $n(n-1)/2$ correlation coefficients

```
disE <- list(coef = c("(Intercept)" = 50), sd = 10)
disG <- list(coef = c("(Intercept)" = 50), sd = 10)
disV <- list(coef = c("(Intercept)" = -10.35534,
             E = 0.70711, G = 0.5), sd = 5)
disN <- list(coef = c("(Intercept)" = 45,
             V = 0.1), sd = 9.949874)
disW <- list(coef = c("(Intercept)" = 15,
             V = 0.7), sd = 7.141428)
disC <- list(coef = c("(Intercept)" = 0,
             N = 0.3, W = 0.7), sd = 6.25)
dis.list = list(E = disE, G = disG, V = disV,
                N = disN, W = disW, C = disC)

plant <- custom.fit(plant_dag, dist = dis.list)
```


References

R packages

- `bnlearn` https://cran.r-project.org/web/packages/bnlearn/, https://www.bnlearn.com/
- `BayesNetBP` https://cran.r-project.org/web/packages/BayesNetBP/index.html
- `bnstruct` https://github.com/sambofra/bnstruct

Books

- O. Pourret et al, *Bayesian Networks, A Practical Guide to Applications*, J.Wiley and Sons, 2008, ISBN 978-0-470-06030-8
- M. Scutari, J.B. Denis, *Bayesian Networks with Examples in R*, CRC Press, 2015, ISBN 978-1-4822-2559-4