Bayesian Analysis

Sean vdMerwe

University of the Free State

2025

Click here to go fullscreen!

Click here to view in 2 column mode (for large screens)

seanvdm.co.za

Software

- This course will use open source software ONLY.
- All software is free - no excuses for not being able to work outside class times.
- Software can be downloaded off the internet.
- Install R first (R Core Team, 2024)
- Install Rtools second. Rtools version must align (not match) with R version.
- Install RStudio third. Lastly, install Quarto.
- Then install essential R packages using RStudio:
  - rmarkdown, quarto ⟵ this will be used for all assignments
  - openxlsx ⟵ to read Excel files (use this, seriously)
  - rstan ⟵ this will be the focus of half the course
  - tidyverse ⟵ to make code easier

Overview - click to jump

Learning resources

- With Bayes becoming more accessible, the learning resources available to a student of Bayes are growing rapidly.
- You can get help and advice rapidly on social media if you ask in the right way.
  - Always first do a thorough search to make sure someone else hasn't already asked your question (99.9% of the time they have).
  - Provide a reproducible minimal example (a mini version of your problem that people can try at home).
- There are free e-books and similar products you can view online.
- There are video series you can watch.
  - **Warning:** the quality of online resources can vary wildly.
  - Bayesian statistics with R
  - https://www.youtube.com/playlist?list=PLDcUM9US4XdNM4Edgs7weiyIguLSToZRI goes with https://github.com/rmcelreath/stat_rethinking_2026

Read, read, read

- Soon (within a year from now for most of you) you will want to earn a salary and contribute to society. If all you know how to do is copy paste other people's (or AI's) ideas then you will not succeed.
- Don't just find the first shortcut to your immediate problem, **think long term**!
- When you hear something in class you don't understand, **make a note immediately**.
- Everything you don't understand you **must** look up and read up on.
- Don't just trust AI, although it is not the worst place to start.
- Search the internet and read, read, read, **until you understand**.

Bibliography (for interest only - read what interests you) I

Adler, Daniel, Duncan Murdoch, et al. (2014). *rgl: 3D visualization device system (OpenGL)*. R package version 0.93.996. URL: http://CRAN.R-project.org/package=rgl.

Albert, Jim (2009). *Bayesian Computation with R*. Ed. by Robert Gentleman, Kurt Hornik, and Giovanni Parmigiani. Springer. ISBN: 978-0-387-92297-3. DOI: 10.1007/978-0-387-92298-0. eprint: 978-0-387-92298-0.

Ando, Tomohiro (2010). *Bayesian model selection and statistical modeling*. CRC Press, pp. 101–110. URL: http://dx.doi.org/10.1201/EBK1439836149-c5.

Cowles, Mary Kathryn (2013). *Applied Bayesian Statistics*. With R and OpenBUGS Examples. Springer. ISBN: 978-1-4614-5695-7. URL: http://www.springer.com/statistics/statistical+theory+and+methods/book/978-1-4614-5695-7.

Gelman, A. et al. (2013). *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis. ISBN: 9781439840955. URL: http://books.google.co.za/books?id=ZXL6AQAAQBAJ.

Jeffreys, H (1998). *The Theory of Probability*. OUP Oxford. ISBN: 9780191589676. URL: http://books.google.co.za/books?id=vh9Act9rtzQC.

Kruschke, John (2010). *Doing Bayesian Data Analysis - A Tutorial Introduction with R and BUGS*. Academic Press Inc. ISBN: 978-0-12-381485-2.

Lunn, D., David J. Spiegelhalter, A. Thomas, and Nicola G. Best (2009). "The BUGS project: evolution, critique and future directions (with discussion)". In: *Statistics in Medicine* 28, pp. 3049–3082.

Bibliography (for interest only - read what interests you) II

Marin, Jean-Michel and Christian Robert (2014). *Bayesian Essentials with R*. 2nd ed. Springer. ISBN: 978-1-4614-8686-2. URL: http://www.springer.com/statistics/computational+statistics/book/978-1-4614-8686-2.

Minka, T. P. (2012). *Estimating a Dirichlet distribution*. Microsoft Research Paper. URL: http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/minka-dirichlet.pdf.

Ntzoufras, Ioannis (2009). *Bayesian Modeling Using WinBUGS*. 1st ed. Wiley. ISBN: 978-0470141144.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: http://www.R-project.org/.

Robert, Christian and George Casella (2004). *Monte Carlo Statistical Methods*. 2nd ed. Springer. ISBN: 978-0387212395.

Ross, Sheldon M (2006). *Simulation*. Academic Press. ISBN: 9780125980630.

Spiegelhalter, David J., Nicola G. Best, Bradley P. Carlin, and Angelika van der Linde (2002). "Bayesian Measures of Model Complexity and Fit". English. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 64.4, pp. 583–639. ISSN: 13697412. URL: http://www.jstor.org/stable/3088806.

Wichman, B A and I D Hill (1982). "Algorithm AS 183: An Efficient and Portable Pseudo-Random Number Generator". In: *Applied Statistics* 31, pp. 188–190.

Bibliography (for interest only - read what interests you) III


Yang, R and J O Berger (1998). *A Catalog of Noninformative Priors*. Technical Report. stats.org.uk. URL: http://www.stats.org.uk/priors/noninformative/YangBerger1998.pdf.


Zellner, A. (1997). *Bayesian Analysis in Econometrics and Statistics: The Zellner View and Papers*. Economists of the Twentieth Century Series. Edward Elgar Pub. ISBN: 9781858982205. URL: http://books.google.co.za/books?id=ICW7AAAAIAAJ.

Assignment and Vectors

The following are identical but '$<-$' is the standard:

```
x = 5
x <- 5
5 -> x
```

The following are vectors:

```
x <- c(1, 2, 3, 4, 5)
x <- c('Green', 'Red', 'Blue')
x <- 1:10
x <- seq(1, 11, 2)
x <- seq(0, 10, length.out = 20) # You can replace 'length.out'
    with 'length' or 'leng' or 'len' or 'l'.
x <- rep('a', 10)
x <- rep(1:4, 1:4)
x <- rep(1:3, each = 2, times = 3)
```

Matrices

The following are matrices:

```
matrix(0, nrow = 3, ncol = 4) # Or matrix(0, 3, 4)
x <- matrix(1:4, 3, 4, byrow = TRUE)
matrix(1:12, ncol = 4)
diag(rep(1, 4))
```

Data sets are usually not matrices but you can pretend they are and you'll generally be fine.

Matrices can be given row names and column names as follows:

```
rownames(x) <- c('a', 'b', 'c')
colnames(x) <- c('w', 'x', 'y', 'z')
```

Let's break it down

The number in the third column of the second row of matrix x is:

```
x[2, 3]
```

The second column of a matrix x is:

```
x[, 2]
```

Note that this comes out as a vector.
The first 3 columns of matrix x is:

```
x[, 1:3]
# or
x[, -4]
```

Let's build it up

Suppose I want to put two copies of x **next to** each other in a new matrix y.

Method 1:

```
(y <- cbind(x, x))
```

Method 2:

```
(y <- matrix(x, 3, 8))
```

Note that Method 1 preserves the names, while Method 2 does not because it makes a new matrix.

Let's build it up more

Suppose I want to put two copies of x **underneath** each other in a new matrix y.

Method 1:

```
(y <- rbind(x, x))
```

Method 2:

```
(y <- matrix(t(x), 6, 4, byrow = T))
```

Overview - click to jump

Read in survey data

First change directory to where you saved the file, unless you're working in Rmarkdown, in which case it will be done for you automatically. See https://seanvdm.co.za/post/openxlsxtips/ for more information.

```
survey <- read.csv('RWorkshopData.csv', row.names
    =1)
names(survey)
nrow(survey)
class(survey)
class(survey$FavColour)
summary(survey)
attach(survey) # Creates a separate variable for
    each column - bad practice in general but used
     here for ease of understanding the code that
    follows
sd(Age)
```

Summarising data

- All of EXCEL's pivot table functionality is available
- 'table' works well for categorical variables

```
table(ChickenOrFish)
table(ChickenOrFish, FlashDiskSize)
```

- 'tapply' works well for continuous variables

```
tapply(Tweets, ChickenOrFish, sum)
tapply(Tweets, list(FlashDiskSize, ChickenOrFish), sum)
```

- These functions make drawing graphs much easier

Basic Graphs

Let's draw some graphs of our data:

```
hist(Age)

barplot(table(ChickenOrFish))
barplot(table(ChickenOrFish, UGorPGorLec))

plot(Age, Tweets)

qqnorm(Age)
qqline(Age) # adds line to previous qqplot
```

Overview - click to jump

Help

- There are lots of ways to get help in R:
- If you know the name of the command to use but need help with how to use it (or want to know more about what it can do) then use the command '?'
- For example, '?mean', '?cor', '?apply', *etc.*
- '?par' will give you a list of graphics parameters you can use to make your graphs look any way you want.
- If you have some idea of the command you need then use '??'
- '??csv' will lead you to commands that work with '.csv' files.

More Help

- If you know what you want to do but have no idea how to do it then Google/DuckDuckGo
- Just explain what you want to do in the search box and add 'R' somewhere
- This is usually the best way to get answers to complicated problems.
- If you just generally want to know more about R (or forgot the help commands) then click on the Help menu and explore the options there.
- Lastly, there are many useful books.

Overview - click to jump

Pipes

The standard pipe is part of base R and does not need any packages, even though it was originally introduced via the `magrittr` package, which still has many useful advanced pipes. The pipe allows for a different style of coding that appeals to people with non-functional-coding backgrounds. The following lines of code are identical as far as R is concerned (behind the scenes):

```
y = f(x)
y <- f(x)
y <- x |> f()
x |> f() -> y
# think: input |> processing() -> output
```

Pivoting

- The Tidyverse requires data in Tidy form. The basic idea is to have 1 column per variable, and 1 row per observation.
- Typically data is easier to manipulate in long form. Long form has all the measurements of a type underneath each other, even if it means subjects take up multiple rows.
- Long form also makes modelling easier when there are missing or censored values in the dependent variable.
- Should you data be in wide form you can pivot it to long form using `pivot_longer` from the `tidyr` package.

Do **not** use deprecated functions such a `melt` or `reshape`. Avoid deprecated functions in general.

dplyr

Take the time to go to the help now, and read the descriptions of the following data manipulation functions:

```
select # Base 'equivalent' is []
filter # Base equivalent is subset
group_by # Base 'equivalent' is by or split
summarise # Base 'equivalent' is aggregate
mutate # Base 'equivalent' is the dollar symbol
case_when # Base 'equivalent' is switch
arrange # Base 'equivalent' is [order(),]
```

ggplot2

When data is in long form and you want to do plots illustrating multiple variables at once then the grammar of graphics method is ideal.

```
survey |> ggplot(aes(x = Age, y = Tweets,
         colour = ChickenOrFish) +
    geom_point() +
    facet_wrap(~factor(UGorPGorLec))
```

Overview - click to jump

Regression

- Models are specified using a special syntax inside the command 'lm':
  - '$\sim$' separates the dependent variable on the left of it with the model on the right of it.
  - '+' adds a variable to the model
  - '.' indicates all the available variables (requires specifying a dataset)
  - '1' refers to a constant/intercept (column of ones)
  - '-' excludes a variable. For example, the intercept is automatically included in every model, so to exclude it use '-1'.
  - ':' between two variables indicates the interaction
  - '*' includes both variables and their interaction
- Random effects have additional syntax. If you are using 'lmer' then you can include random intercepts using $(1|\textit{VariableName})$ for example.

Regression

- Use 'summary' to get nice output.
- Use 'plot' to get diagnostic plots.
- Use 'anova' to get an ANOVA table.
- Examples:

```
summary(model1 <- lm(Tweets ~ Age))
plot(model1)
summary(lm(Tweets ~ Age + Nexercise))
anova(lm(Age ~ UGorPGorLec))
```

- For GLMS: replace 'lm' with 'glm' and add the link and distribution.

```
summary(model2 <- glm(Age ~ Nexercise, family
   = poisson(link = "log")))
```

Overview - click to jump

Logical indexing

- We saw that [ ] can be used to access pieces of a vector or matrix using indexes.

```
w <- 11:20
w[4:5]
w[5:length(w)]
```

- But subsets can also be obtained using TRUE (or non-zero numbers) and FALSE (or zeros) to indicate which elements should be returned or not returned.

```
w[c(F,FALSE,F,T,T,F,F,F,F,F)]
w[c(F,F,F,F,T,T,TRUE,T,T,T)]
w[c(rep(F, 4), rep(T, 6))]
```

Logical indexing p.2

- The TRUE and FALSE can be generated automatically.
- It helps if you think of the [ ] as meaning 'where'
- Examples:

```
# To look at the exercise of Lecturers use
Nexercise[UGorPGorLec == 'Lecturer']
# To look at undergrad tweeting use
Tweets[UGorPGorLec %in% 'Undergrad']
# The ages of students that chose Chicken:
Age[(((UGorPGorLec == 'Undergrad') | (UGorPGorLec == 'Postgrad
    ')) & (ChickenOrFish == 'Chicken'))]
```

Basic tests

- Are fish lovers older?

```
t.test(Age[ChickenOrFish == 'Chicken'], Age[ChickenOrFish == '
    Fish'])
```

- Do students tweet more than lecturers?

```
t.test(Tweets[((UGorPGorLec == 'Undergrad') | (UGorPGorLec ==
    'Postgrad'))], Tweets[UGorPGorLec == 'Lecturer'])
```

- Is there a relationship between exercise and flash disks?

```
chisq.test(table(Nexercise, FlashDiskSize))
```

- Are ages Normal?

```
shapiro.test(Age)
```

Other tests

- Is there a general preference for Chicken or Fish?

```
prop.test(sum(ChickenOrFish == 'Chicken'), length(
    ChickenOrFish))
# This works because a TRUE is a 1 and a FALSE is a 0, so 'sum
    ' counts the number of people that chose Chicken
```

- What about if we break up the sample according to FlashDiskSize?

```
(FDsizes <- levels(FlashDiskSize))
for (i in FDsizes) {
    print(prop.test(sum((ChickenOrFish == 'Chicken') & (
        FlashDiskSize == i)), sum(FlashDiskSize == i)))
}
```

Overview - click to jump

Installing packages

- Sometimes the functionality you want is not available in standard R.
- There are hundreds of packages available for download that add new functions.
- You can go to http://cran.r-project.org/web/packages/ and download the package you want (or use Google to find exactly what you want).
- If you have downloaded the package you can install it using the menu button 'Install package(s) from local zip files'.

Easy packages

- A much easier way is to install over the internet directly.
- Off-campus, click on 'Install package(s)' on the menu.
- On campus use the following command:

```
install.packages('pkgname', repos='http://
    mirror.ufs.ac.za/cran')
```

- Remember to replace pkgname with the name of your package.
- For today we will only need the 'rgl' package, so install that now.

The RGL package as example

- The RGL package is one of many packages that let you draw 3D graphs. Plotly is a more modern alternative.
- Load it now using the library command or the button on the menu:

```
library(rgl)
```

- Let's draw a graph using 5 of our variables at once.

```
plot3d(Age, Nexercise, Tweets, type='s', size
   = ((ChickenOrFish == 'Chicken')+1), col =
   FavColour)
# If there is a problem with the colours, run
   the following and then try again
for (i in 1:length(FavColour)) { if (!any(
   colours()==FavColour[i])) {FavColour[i] <-
   'black'}}
```

The pipe repeated (it's that important)

- R supports multiple programming styles, including functional programming styles.
- So, typically, an R statement takes the form

```
output <- processing(input)
```

- This aligns with the mathematical style of $y = f(x)$.
- However, in flowcharts you might see something like

```
input |> processing() -> output
```

- And this is valid R syntax.

Pipe example

- So, to calculate and store the average of 100 random standard normal values, we could use any of the following equivalent lines of R code

```
average <- mean(rnorm(100))
average <- rnorm(100) |> mean()
average <- 100 |> rnorm() |> mean()
rnorm(100) |> mean() -> average
```

- Also note that you can put carriage returns anywhere between pieces of a line.

Tidyverse example

- Suppose you have a monthly time series stored as a block, with a column for each month, and the first column called Year:

```
ts_block <- (rnorm(120)*20 + 1:120) |> matrix(10,
    12, byrow = TRUE)
colnames(ts_block) <- month.abb
d_wide <- data.frame(Year = 2011:2020, ts_block)
d_wide |> head()
```

- and you wish to plot it

```
library(tidyverse)
d_long <- d_wide |>
  pivot_longer(-Year, names_to = "Month", values_
      to = "Y") |>
  mutate(Date = ymd(paste(Year, Month, 15)))
d_long |> head()
d_long |> ggplot(aes(x = Date, y = Y)) + geom_line
    ()
```

data.table

- The `data.table` and similar packages were created to make processing large data sets much faster.
- They executing core repetitive operations using compiled C++.
- This can speed up execution massively when data sets are very large (e.g. 1 billion rows).
- Each such package has its own syntax, although there are helper packages (like `dtplyr`) that can bring the syntax gap.
- Another way to gain speed on repetitive operations is to use parallel processing. A few packages may do this automatically, but in generally it requires use of a deliberate effort (by design, to give you maximum power and control).
- A lot of speed can be gained by just using good programming principles, such as not growing large vectors in a loop.
- **However**, in the vast majority of projects the programming time will far exceed the execution time. So, **the best way to gain speed is to practice writing code faster and better**.

Overview - click to jump

Create a data set in EXCEL

- Create a data set in EXCEL with two variables:

1. Random integers from 1 to 9
2. A random sequence of 'a' or 'b'

- It should look something like this:

rownum	randint	randfactor
1	3	a
2	6	b
⋮	⋮	⋮
20	2	a

Do tests

1. Read the data into R.
2. Do a t test to see if the mean of the a's is the same as the mean of the b's.
3. Do a runs test to see if your a's and b's were random.
   - Install package 'lawstat', load it, convert your a's and b's to 'TRUE/FALSE' and use runs.test .

Overview - click to jump

for loops

- For loops are used to repeat boring tasks.
- The basic notation is

```
for (i in 1:n) {
 # Commands to be repeated n times
}
```

- As we saw before, 1:n can be replaced with any sequence or vector,
- BUT `seq_len(n)` (same as 1:n) is actually the best practice!
- Also, remember to create space to store the results in advance - the speed gains are enormous.

for loop examples

- Let's use a loop to get the first 30 numbers in the Fibonacci series

```
fibonacci <- rep(1,30)
for (i in 3:30) {
 fibonacci[i] <- fibonacci[i-2] + fibonacci[i-1]
}
print(fibonacci)
```

- for loops are best for complicated operations or operations that must happen in a specific order,
- Otherwise it's better to use vector operations.
- For example, let's replace all the even numbers in the Fibonacci series with random Normal values:

```
fibonacci[(fibonacci%%2)==0] <- rnorm(sum((fibonacci%%2)==0))
fibonacci
prettyNum(fibonacci)
```

sapply

- Whenever the steps of a loop are independent then replace it with an apply function.
- The apply functions are just like for loops but they take care of creating the empty space to store the results for you!
- They also keep the operations inside the loop contained in a closed environment, so the don't mess with the rest of your code.
- Let's suppose that for each number in our Fibonacci sequence we want a random uniform number between 0 and that number:

```
fib_random <- fibonacci |> sapply(\(number) {
    runif(1, 0, number)
})
```

- The above wasn't necessary here since `runif` is a vectorised function, but a lot of functions need their inputs one at a time.
- The `\()` notation is the same as `function()`, which is explained a few slides on.

sapply's friends

- `sapply` is the coolest function in R once you get used to it, allowing for clean and robust code.
- `sapply` is for doing something to every element of a vector or sequence.
- The `apply` function is for a matrix or array, and lets to tell it which dimension(s) to split on.
- There is also `lapply` that is like `sapply` but always returns a list. This is great when you want to build a data frame one row at a time (just combine with `do.call(rbind, list)` or `bind_rows()` afterwards).
- There is also `tapply` and `vapply`, and the whole `purrr` library to explores!

while loops and ifs

- While loops replace the counter with a condition.
- Let's roll 3 dice until all of them are 5s or at least 2 of them are 6s

```
(dice <- ceiling(runif(3)*6))
while (!(all(dice == 5) || (sum(dice == 6) >= 2))) {
print(dice <- ceiling(runif(3)*6)) }
```

- if statements are used to branch or handle special conditions
- Let's change our loop to only stop on 6s if 6s happened before

```
(dice <- ceiling(runif(3)*6))
SixesHappened <- FALSE
while (!(all(dice == 5) || ((sum(dice == 6) >= 2) &&
    SixesHappened))) {
if (sum(dice == 6) >= 2) { SixesHappened <- TRUE }
print(dice <- ceiling(runif(3)*6)) }
```

Overview - click to jump

Functions

- Functions are closed boxes of code that take in some input, do calculations, and produce output.
  - Actually, any or all of those are optional.
- Functions make your code simpler and easier to understand.
- Functions mean you don't have to rewrite existing code when you have a new problem.
- What happens in the function stays in the function.

```
YourFunctionName <- function(inputs) {
Calculations
return(outputs) }
```

Example function 1

- Let's make a function that checks whether numbers are even.

```
is.even <- function(numbers) {
evens <- (numbers %% 2)==0
names(evens) <- numbers
return(evens) }

is.even(4)
is.even(5)
is.even(5:10)
is.even(matrix(round(runif(16)+2), 4))
```

- Practice: make a function for uneven numbers using this function and the 'not' operation '!'

Example function 2

- For our next function we will take in a numeric variable and a factor.
- We will do the Shapiro-Wilk test for Normality on the variable for each level of the factor.
- We will return the p-values as a named vector.

```
SWpvalues <- function(numericvar, factorvar) {
factorlevels <- levels(factorvar)
numlevels <- nlevels(factorvar) # OR numlevels <- length(
    factorlevels)
pvalues <- rep(0.5, numlevels)
for (i in 1:numlevels) {
 pvalues[i] <- shapiro.test(numericvar[factorvar==factorlevels
     [i]])$p.value
}
names(pvalues) <- factorlevels
return(pvalues) }
```

Example function 2 simplified

- The function on the previous slide can be greatly simplified by using pipes, sapply, and setNames. See if you understand how the two are equivalent:

```
SWpvalues <- \(numericvar, factorvar) {
factorlevels <- levels(factorvar)
factorlevels |> sapply(\(factor_level) {
    shapiro.test(numericvar[factorvar %in% factor_level])$p.
        value
}) |> setNames(factorlevels)
}
```

Running Example function 2

- And now we will apply it to our data.

```
SWpvalues(Age, ChickenOrFish)
SWpvalues(Tweets, FlashDiskSize)
```

- Notice how the same code solves multiple problems, even ones we haven't thought of yet!
- Currently our function requires a factor but what if we want it to work even if the user doesn't supply a factor?
- Just replace the first line with

```
SWpvalues <- function(numericvar, factorvar = factor(rep(1,
    length(numericvar)))) {
```

Abusing Example function 2

- Let's use our function for something we didn't design it for:
- I want to do the Normality test for every column of a matrix.
- No need for a new function or even a loop.

```
Normal.matrix <- matrix(rnorm(200,2,2), 50)
apply(Normal.matrix, 2, SWpvalues)
# 2 means split by dimension 2, this is
   opposite to MATLAB
```

Taking things up a notch

- Let's increase the number and size of each sample.

```
Normal.matrix <- matrix(rnorm(20000000, 2, 2), 5000)
x <- apply(Normal.matrix, 2, SWpvalues)
```

- You should have noticed that it took a while.
- That's because it only used 1 core of the processor.
- If we work with functions we can use multiple cores at the same time using the parallel library.

```
library(parallel)
cl <- makeCluster(4)
x <- parCapply(cl, Normal.matrix, SWpvalues)
stopCluster(cl)
```

Overview - click to jump

An Exponential Regression Model

Consider the following model:

- $y_i \sim Exp(\lambda_i = \frac{1}{\mu_i})$
- $\mu_i = \mathbf{x}_i\boldsymbol{\beta}$
- $f(y_i) = \frac{1}{\mathbf{x}_i\boldsymbol{\beta}} \exp\{-\frac{y_i}{\mathbf{x}_i\boldsymbol{\beta}}\}$
- $Lik(\mathbf{y}) = \left[\prod_{i=1}^{n} \frac{1}{\mathbf{x}_i\boldsymbol{\beta}}\right] \exp\{-\sum_{i=1}^{n} \frac{y_i}{\mathbf{x}_i\boldsymbol{\beta}}\}$
- $\ell(\mathbf{y}) = -\sum_{i=1}^{n} \log(\mathbf{x}_i\boldsymbol{\beta}) - \sum_{i=1}^{n} \frac{y_i}{\mathbf{x}_i\boldsymbol{\beta}}$
- We are interested in the estimates of $\boldsymbol{\beta}$ and are going to try to get them with the ML method.

First program the negative log likelihood

```
nll.expmodel <- function(b,y,X) {
if (any(y<=0) || any(X%*%b<=0)) { return
    (100000000)
} else {
nll <- sum(log(X%*%b)) + sum(y/(X%*%b))
return(nll) } }
```

Then find the point where is it a minimum:

```
X <- cbind(rep(1,length(Age)), Tweets, Nexercise)
(firstguess <- coef(lm(Age ~ Tweets + Nexercise))
    )
(beta.hat <- optim(firstguess, nll.expmodel, y=
    Age, X=X, method="L-BFGS-B")$par)
```

Overview - click to jump

History of Simulation

- Definition of Simulate: To assume the mere appearance of, without the reality; to assume the signs or indications of, falsely; to counterfeit; to feign.
- Simulation was invented long before it was possible to do it properly.
- Most statistical simulation techniques were invented in the 70s and 80s from a theoretical standpoint.
- Doing simulations in those days meant making holes in punch-cards and feeding them to a mainframe computer.
- Basic simulations took days.
- Now it's easy and fast.

Why simulate?

- To calculate probabilities and measures (mean, quantiles, etc.) of distributions when it is impossible or difficult to do so in closed form. **(Happens all the time in Bayes!)**
- To describe and analyse the behaviour of a complex system or process.
- To ask "what if" questions.
- It is a shortcut or check for problems that are difficult to solve from a theoretical basis.
- It can even solve problems that have no theoretical basis.

Pseudo Random Numbers

- A computer doesn't understand 'pick a number from 1 to 10', *i.e.* it can't come up with random numbers at all!
- Instead, it constructs a deterministic (non-random) sequence of numbers that has all the properties of a set of independent $U(0, 1)$ random numbers.
- It basically fakes it well enough to pass every test of randomness that we care to apply to it.
- The most commonly used set of tests is called DIEHARD and were developed by Professor George Marsaglia, Department of Statistics, Florida State University.
- They are available at the following Web site: http://i.cs.hku.hk/~diehard

PRNGs

- Microsoft Excel and most Microsoft products apply the Wichman algorithm (a Linear Congruential Generator) to get random numbers (Wichman and Hill, 1982).
- For 20 years it was the best available and the first to pass the basic DIEHARD tests.
- As people started simulating more it became clear that this algorithm is inadequate, as it can repeat itself as early as every 10^{13} values (depending on the variant) and isn't perfectly fair to all values in the target range.
- It also has serial correlation and the lower order bits start repeating very quickly (the last bit alternates).

Twister

- The Mersenne Twister algorithm by Nishimura and Matsumoto generates double-precision (64-bit) values in the closed interval $[2^{-53}, 1 - 2^{-53}]$.
- It has a period of $(2^{19937} - 1)/2$ (approx. 2×10^{6001})
- For a full description of the Mersenne twister algorithm, see http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html
- This algorithm is standard in newer versions of MATLAB, R, SAS, Mathematica.
- Statistical packages use this to generate other distributions.
- For example, R has these distributions built in (and you can download more): Beta, Binomial, Cauchy, χ^2, Exponential, F, Gamma, Geometric, Hypergeometric, Log-Normal, Multinomial, Negative Binomial, Normal, Poisson, t, Uniform, Weibull, and a few more.

Overview - click to jump

R Code for basic simulation

- 1 random $U(a = 2, b = 3)$ value: runif(1, 2, 3)
- 2 random $N(\mu = 3, \sigma^2 = 2^2 = 4)$ values: rnorm(2, 3, 2)
- 3 random $Gamma(\alpha = 4, \lambda = 5)$ values: rgamma(3, 4, 5)
- 4 random χ^2_6 values: rchisq(4, 6)
- 1 Multivariate Normal vector with mean vector $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ and covariance matrix $\begin{bmatrix} 1 & 0.2 \\ 0.2 & 1 \end{bmatrix}$:

```
library(MASS); mvrnorm(1, c(3, 4), matrix(c(1,0.2,0.2,1), 2,
    2))
```

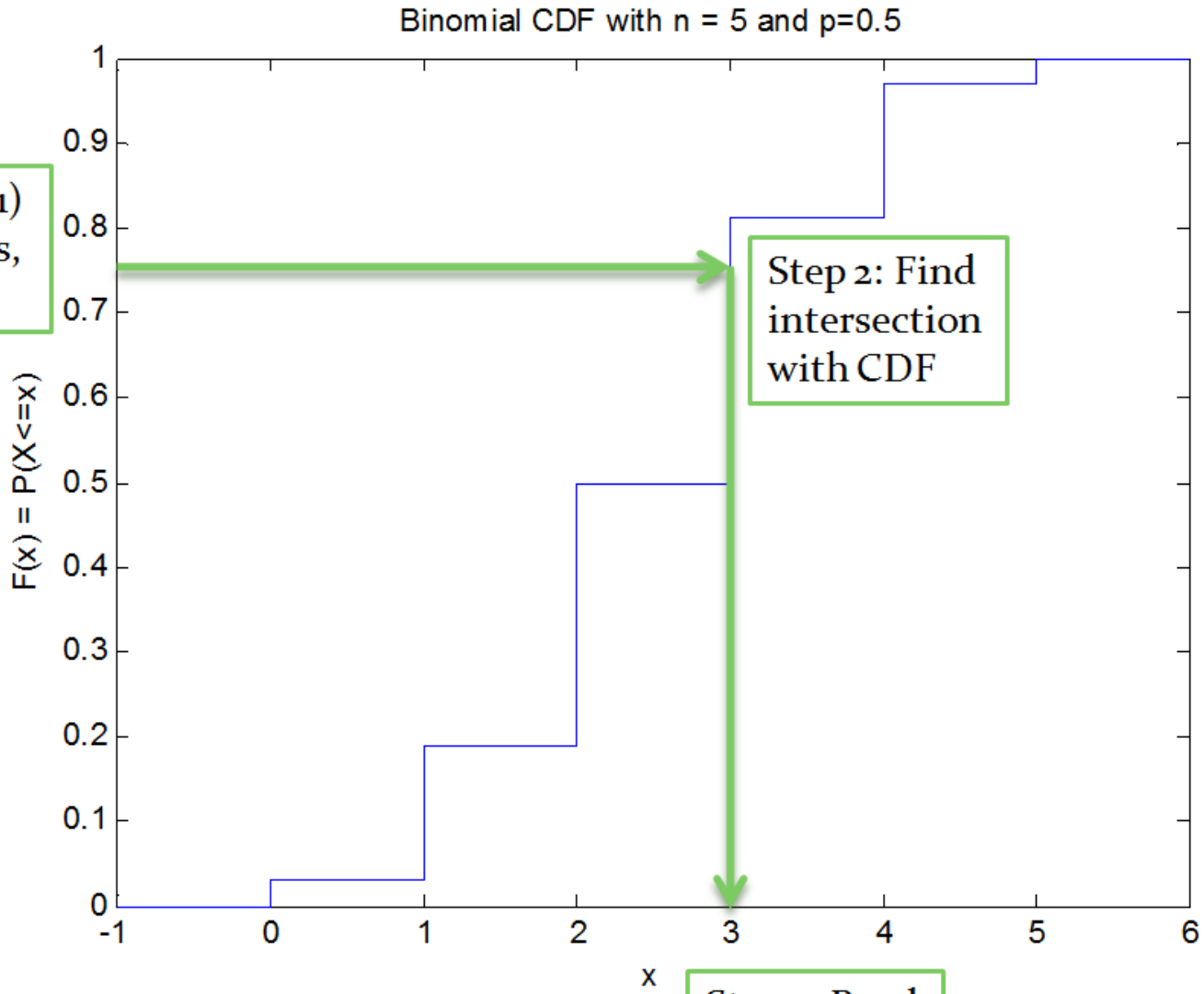
Inverse CDF Method

- To go from a Uniform value to a value from some other distribution we have to transform it in some way.
- The easiest way is to simply apply the inverse cumulative distribution function (quantile function):
- Let $U \sim U(0,1)$ and let $X = F^{-1}(U)$ then the CDF of X is F.

$$Proof : P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x)$$

- Example: If $X \sim U(0,1)$ then $T = -\log(X)/\lambda \sim Exp(\lambda)$.
- Any distribution whose CDF can be inverted will work well with this method, *e.g.* Weibull, Pareto, Gumbel, U(a,b), *etc.*
- It also works well with any finite discrete distribution (see next slide).

Discrete Inverse CDF



Step 1: Pick $U(0,1)$ number on y axis, say 0.7581...

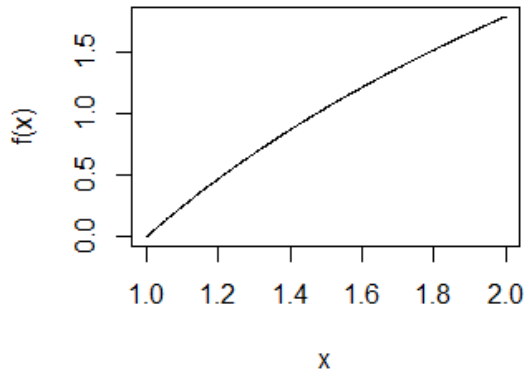
Step 2: Find intersection with CDF

Step 3: Read off x value

Discretization

- What do we do with a continuous distribution where we can't invert the CDF? If the domain is finite and small enough then we can turn it into a discrete distribution!
- We simply calculate the value of the CDF at small intervals of the domain and simulate values from this distribution.
- As long as our intervals are small enough, and we cover almost all the probability, we will get an accurate approximation.
- If you can't calculate the CDF then calculate the PDF at points in the domain and calculate the cumulative sum to get an approximation of the CDF.
- NB: Your CDF must start at zero and end at one!
- We can divide every value of the approximate CDF with its last value (the sum of the entire discretized PDF). This is the easiest option when you have an unknown constant.

Example of Discretisation



- Let $f(x) = c \log x, 1 \leq x \leq 2$.
- In this case we can't invert the CDF.
- R code for simulating from this distribution:

```
n <- 240
accuracy <- 500
x <- seq(1, 2, length.out=accuracy)
fx <- log(x)
sims <- sample(x, n, replace = TRUE, prob = fx)]
print(sims)
```

- The sample command does the standardising and simulating for you

Extra Example: Roulette Table

- If I bet on odds every time with R10, what is my profit distribution after 20 spins of a roulette wheel?
- First let's consider an example spin:

```
(roulettenums <- c('00', as.character(0:36)))
(spins <- sample(roulettenums, 20, replace = T))
```

- Now let's do 20 spins, 1000 times and store the profits:

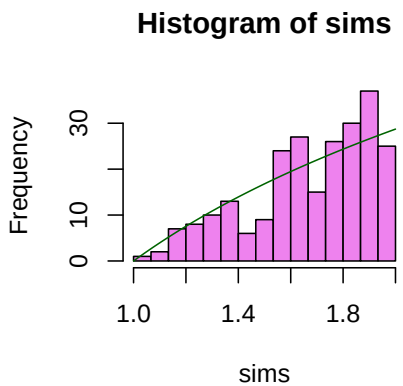
```
profits <- rep(0,1000)
for (i in 1:1000) {
  profits[i] <- sum(((runif(20)<(18/38))*2 - 1)*10)
}
hist(profits, 100, xlab='Profit in Rand', col='navy', density
    =10, angle=30)
```

Overview - click to jump

Overlaying a frequency diagram on a histogram

- Assume you have a histogram of n simulations/observations with h partitions.
- And you have a discretised density over the same domain with d values/partitions.
- Then you calculate the normalising constant η, required to get the density on the same scale as the histogram, by solving the equation $\frac{n}{h} = \frac{\eta}{d}$, *i.e.* $\eta = \frac{n.d}{h}$. Or set the frequency option to false.

```
h <- 15
breaks <- seq(1, 2, l = (h+1))
hist(sims, breaks, col = 'violet')
lines(x, fxs*n*accuracy/h, col = 'dark green')
```



Always work on the log scale

- In R $e^x = \infty \; \forall \; x > 709$.
- Consider the problem $e^{-800} \times e^{800}$. The correct answer is 1 but R will calculate it as $0 \times \infty = NaN$.
- However, if you take the log, do the calculation, and then only take the exponent at the end, then you get the right answer: $\exp\{-800 + 800\} = 1$.
- In practice, always work on the log scale as long as possible.
- If you want to take the log of the gamma function use 'lgamma(x)' instead of 'log(gamma(x))'.
- When working with really small probabilities it may help to work on the log scale. For example, all density functions in R have a 'log' option.
- Lastly, use approximations in extreme cases, *e.g.* $\log(1 + e^{\eta}) \approx \eta \; \forall \; \eta > 709$.

Other Univariate Samplers

- Acceptance-Rejection method:
  - Put the density in a tight box (or more complicated shape).
  - Pick a random point in the box.
  - If the point is below the density curve then accept the x value.
- Sampling Importance Resampling:
  - Pick values from easy density close to target density.
  - Calculate weights of these values using the ratio of densities at these values (target over proposal).
  - Standardize the weights to add up to 1.
  - Resample from previously picked values according to their weights.
- Slice Sampling

Example of Acceptance Rejection

- Let $X \sim Beta(3,4)$. Simulate 1000 numbers from the density and compare to the density.

```
maxval <- dbeta((3-1)/(3+4-2),3,4)
i <- 0; n <- 1000; sims <- rep(0.5,n)
while (i<n) {
  x <- runif(1)
  if (dbeta(x,3,4) > (runif(1)*maxval)) {
    i <- i + 1; sims[i] <- x
  }
}
plot.ecdf(sims)
xvals <- seq(0,1,l=1000); lines(xvals, pbeta(xvals,3,4), col='
    magenta')
```

Other Samplers

The samplers discussed before this point require multivariate problems to be broken down into univariate problems first, say via Gibbs sampling. For example, the inverse CDF does not generally make sense for multivariate problems, so cannot be applied directly.
The samplers below work regardless of the dimension. The algorithms stay the same if you have multivariate problems instead of univariate.

- Metropolis
- Metropolis-Hastings
- Hamiltonian Monte-Carlo
- No-U-Turn-Sampler (NUTS)

The last two are used by Stan, while JAGS & BUGS use Gibbs sampling.

Exercises 1

Next Exercises

For numbers 1 to 5 below, summarise your answers in a nice histogram, as well as an empirical CDF plot with the target CDF overlaid.

1. Simulate 27 random values directly from the $F(2, 7)$ distribution.
2. Construct 34 random $F(3, 4)$ values by simulating values from χ^2 distributions only.
3. Construct 46 random $F(4, 6)$ values by simulating standard Normal random numbers only.
4. Simulate 1000000 random values from the density $f(x) = cx\ , \quad 1 \leq x \leq 2$ (Inverse CDF method).
5. Simulate 100 dice rolls.
6. Simulate 2 packs of cards shuffled together, give the cards in the shuffled order.

Exercises 2

Previous Exercises | Next Exercises

For numbers 1 to 4 below, summarise your answers using histograms/bar charts as well as CDF plots.

1. Simulate 1000 random values from a $N(2, 3)$ distribution that is truncated a zero (negative numbers not allowed).
2. Simulate 100 Likert-scale ('Disagree','Partly Disagree','Neutral','Partly Agree','Agree') values using a $N(3, 1)$ distribution as a basis ('Neutral' should be the most common response).
3. Repeat the above scale simulation using a $t_4 + 3$ distribution instead and then a $Gamma(9, 3)$ distribution.
4. Simulate 10000 random values from the density $f(x) = c \sin(x) \cos(x)\ , \quad 0 \leq x \leq 1.5$ using discretization, another 10000 values using the acceptance-rejection method and then use sampling-importance resampling (using truncated Gamma as easy density) to get another 10000 values. Which set of results was closest to the target for your simulations?

seanvdmerwe.co.za

Overview - click to jump

Things you should know by now

- Frequentist definition of probability (proportion of 'successes' in many repetitions).
- Bayesian definition of probability (personal degree of belief in a future 'success').
- Advantages and disadvantages of each

  Frequentists have one probability for everything, while in Bayesian statistics every person has their own probability for something and this probability can change over time. Only by understanding both definitions can we solve practical problems in the real world.

- What a likelihood is and how to derive one.
- Frequentist methods of parameter estimation (method of moments and method of maximum likelihood).

Bayes' theorem

Discrete Version Given: $P(A|B_i), i \in \{1, \ldots, n\}; \cup_{i=1}^{n} B_i = 1$.
Then:

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

Continuous version

$$f(\theta|x) = \frac{f(\theta, x)}{f(x)}$$
$$\propto f(x|\theta)f(\theta)$$

In general we say that *the posterior distribution of the parameters given the data* **is proportional to** *the likelihood of the data given the parameters* **times** *the prior distribution of the parameters*.

Revision

1. If a sample of Log-Normal values has a mean of 10 and a variance of 20, then, according to the method of moments, what are the values of the parameters μ and σ (corresponding Normal mean and variance)?
2. If $X \sim Beta(\alpha, \beta)$ then how can we get moment estimates of the parameters if we have $E(X)$ and $Var(X)$?
3. What is the likelihood if we have a sample of values from the above distribution?
4. How does the data affect the prior distribution?
5. Why do we use the proportional sign in Bayes' theorem?

Overview - click to jump

Example 1 - Bernoulli

Consider a single flip of a possibly biased coin.
Let x be the number of heads (0 or 1).
$\theta = p$ where p is the unknown probability of getting a heads on one flip of this coin.

$$Lik(x|\theta) = p^x(1-p)^{1-x}$$

If we have no idea what the probability is in advance then we can assume a *Uniform* distribution for p, *i.e.* $f(p) = 1$. Then $\pi(p|x) \propto p^x(1-p)^{1-x}$, which means that (because we recognise the Beta distribution)

$$\begin{aligned}
\pi(p|x) &= c * p^x(1-p)^{1-x} \\
&= \frac{\Gamma((x+1)+(2-x))}{\Gamma(x+1)\Gamma(2-x)} p^x(1-p)^{1-x} \\
&= \frac{2p^x(1-p)^{1-x}}{\Gamma(x+1)\Gamma(2-x)}
\end{aligned}$$

Bernoulli with subjective prior

Now suppose the owner of the coin is found and he says that the coin comes up tails 2 out of 3 times, but this varies from day to day like any other coin.
We can use this information by assigning a $Beta(0.5, 1)$ prior for p since this distribution has mean $1/3$ and variance $1/11.25$ which is close to the $1/12$ variance of the *Uniform*.
Explicitly: $f(p) = c_1 p^{-0.5}(1-p)^0$ so that

$$\begin{aligned}
\pi(p|x) &= c_2 * c_1 * p^{x-0.5}(1-p)^{1-x} \\
&= \frac{\Gamma((x+0.5)+(2-x))}{\Gamma(x+0.5)\Gamma(2-x)} p^{x-0.5}(1-p)^{1-x} \\
&= \frac{\frac{3}{4}\sqrt{\pi} p^{x-0.5}(1-p)^{1-x}}{\Gamma(x+1)\Gamma(2-x)}
\end{aligned}$$

Multiple Coin Tosses

Consider n tosses with the number of heads on each toss being x_i.
The "likelihood" of an outcome from this experiment is the probability that $X_1 = x_1$ and $X_2 = x_2$ and so on.
Since these tosses are independent the "and's" become "times".
Formally,

$$\begin{aligned} Lik(\mathbf{x}|p) &= \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} \\ &= p^{\sum_{i=1}^{n} x_i}(1-p)^{n-\sum_{i=1}^{n} x_i} \end{aligned}$$

Applying the same prior as before we get the posterior:

$$\begin{aligned} \pi(p|x) &= c * p^{\sum_{i=1}^{n} x_i - 0.5}(1-p)^{n-\sum_{i=1}^{n} x_i} \\ &= \frac{\Gamma(n+1.5)}{\Gamma(\sum_{i=1}^{n} x_i + 0.5)\Gamma(n+1-\sum_{i=1}^{n} x_i)} p^{\sum_{i=1}^{n} x_i - 0.5}(1-p)^{n-\sum_{i=1}^{n} x_i} \end{aligned}$$

R code

Problem: given the mean and variance of the subjective Beta prior, graph the posterior of p.

Solution:

```
prior.mean <- 1/3
prior.var <- 1/11.25
(prior.mean*(1-prior.mean)/prior.var) -> ab1
(prior.mean*(ab1-1)) -> a
b <- (1-prior.mean)*(ab1-1)
p <- seq(0,1,length.out=400)
n <- 12
(x <- (runif(n) < 0.3))
post <- p^(sum(x)+a-1)*(1-p)^(n-sum(x)+b-1)*gamma(n+a+b)/gamma(sum(
    x)+a)/gamma(n-sum(x)+b)
# OR post <- dbeta(p,(sum(x)+a),(n-sum(x)+b))
plot(p,post,type='l'); grid()
# OR library(LearnBayes); triplot(c(a,b),c(sum(x),(n-sum(x))),'
    right')
```

Overview - click to jump

General Discrete Prior

- Consider now the owner of the coin says he's not sure whether it's the fair coin or the biased coin.
- He says there's a 50% chance that $p = 0.5$ and a 50% chance that $p = 1/3$.
- Then

$$\pi(p_i|x) = \frac{c * Lik(x|p_i)\pi(p_i)}{\sum_{j=1}^{k} c * Lik(x|p_j)\pi(p_j)}$$

$$\therefore \pi(p = 1/3|x) = \frac{Lik(x|p = 1/3)\pi(p = 1/3)}{Lik(x|p = 1/3)\pi(1/3) + Lik(x|p = 0.5)\pi(0.5)}$$

$$= \frac{1/3^{\sum_{i=1}^{n} x_i - 0.5}(2/3)^{n - \sum_{i=1}^{n} x_i}}{1/3^{\sum_{i=1}^{n} x_i - 0.5}(2/3)^{n - \sum_{i=1}^{n} x_i} + 0.5^n}$$

- R Code:

```
library(LearnBayes); pdisc(c(1/3,0.5),c(0.5,0.5),c(sum(x),(n-
    sum(x))))
```

General Discrete Bayes Code

Consider now an arbitrary likelihood function:

```
myLikelihood <- function(thedata, theta) {
#Code to calculate the likelihood should go here:
Lik <- (theta^(sum(thedata)))*((1-theta)^(length(thedata)-sum(
    thedata)));
return(Lik) }
```

Then the code for calculating the posterior is:

```
discrete.posterior <- function(p, prior, thedata, Lik) {
post <- sapply(1:length(p), \(i) { Lik(thedata, p[i])*prior[i] })
post/sum(post) }

discrete.posterior(c(1/3,0.5), c(0.5,0.5), x, myLikelihood)
```

Examples of Conjugate Priors

Density	Parameter	Prior	Posterior
$Poisson(\lambda)$	$\lambda > 0$	$U(0, \infty)$	$Gamma\left(\sum x + 1, n\right)$
$Poisson(\lambda)$	$\lambda > 0$	$Gamma(a, b)$	$Gamma\left(\sum x + a, n + b\right)$
$Gamma(\alpha, \lambda)$	$\lambda > 0$	$U(0, \infty)$	$Gamma\left(n\alpha + 1, \sum x\right)$
$Gamma(\alpha, \lambda)$	$\lambda > 0$	$Gamma(a, b)$	$Gamma\left(n\alpha + a, b + \sum x\right)$
$Weibull(c, \gamma)$	$c > 0$	$U(0, \infty)$	$Gamma\left(n + 1, \sum x^{\gamma}\right)$
$Weibull(c, \gamma)$	$c > 0$	$Gamma(a, b)$	$Gamma\left(n + a, b + \sum x^{\gamma}\right)$
$N(\mu, \sigma^2)$	μ	$U(-\infty, \infty)$	$N\left(\frac{1}{n}\sum x, \frac{\sigma^2}{n}\right)$
$N(\mu, \sigma^2)$	μ	$N(a, b^2)$	$N\left(\frac{b^2 \sum x + a\sigma^2}{nb^2 + \sigma^2}, \frac{b^2\sigma^2}{nb^2 + \sigma^2}\right)$
$LogN(\mu, \sigma^2)$	μ	$U(-\infty, \infty)$	$N\left(\frac{1}{n}\sum \log x, \frac{\sigma^2}{n}\right)$
$Bin(m, p)$	$0 < p < 1$	$U(0, 1)$	$Beta\left(\sum x + 1, nm - \sum x + 1\right)$
$Bin(m, p)$	$0 < p < 1$	$Beta(a, b)$	$Beta\left(\sum x + a, nm - \sum x + b\right)$
$NegBin2(k, p)$	$0 < p < 1$	$U(0, 1)$	$Beta\left(nk + 1, \sum x + 1\right)$
$NegBin2(k, p)$	$0 < p < 1$	$Beta(a, b)$	$Beta\left(nk + a, \sum x + b\right)$

Exercises 3

Previous Exercises | Next Exercises

1. A firm has 2 secretaries: one answers 12 calls an hour; while the other only answers 8, but works twice as many hours. If 4 calls are answered in a particular hour, what is the probability that the better secretary (12/h) was on duty?
2. Now they appoint another 2 secretaries and observe that they manage to answer 4 and 6 calls an hour respectively, even though they both work 20% longer than the one who answers 8. If 9 calls are answered in a particular hour, what is the probability of each secretary being on duty?
3. In a T-shaped maze, a laboratory animal is given a choice of going to the left and getting food or going to the right and receiving a mild electric shock. Assume that before any conditioning (in trial number 1) animals are equally likely to go to the left or to the right. After having received food on a particular trial, the probabilities of going to the left and right become 0.6 and 0.4, respectively, on the following trial. However, after receiving a shock on a particular trial, the probabilities of going to the left and right on the next trial are 0.8 and 0.2, respectively. What is the probability that the animal will turn left on trial number 3? [Hint: this isn't Bayes.]

Exercises 4

Previous Exercises | Next Exercises

1. In a certain town, 35% of the people are DA supporters, 55% are ANC supporters, and 10% are EFF supporters. Records show that in a particular election, 60% of the DA supporters voted, 70% of the ANC supporters voted, and 50% of the EFF supporters voted. If a person in this town is selected at random and it is learned that he did not vote in the last election, what is the probability that he is an EFF supporter?.
2. The number of cars driving down a large highway in a particular 10 min period is Poisson but we don't know the average. A local is asked to guess and suggests a mean of 120 and a standard deviation of 30. The cars are counted during that period on each of 10 days, with a total of only 800 cars. Assuming a Gamma prior, what is the posterior distribution of the average?
3. Suppose that, instead of giving a standard deviation, he says that he's 90% sure the mean is over 50. What is the posterior probability that the mean is over 55?

Exercises 5

Previous Exercises | Next Exercises

1. I have four dice, one with 4 sides (D4), one with six sides (D6), one with 8 sides (D8) and one with 12 sides (D12). I pick one of the dice at random, roll it once and I get a 5. I go to the next room and tell my friend that I rolled a 5 and ask him to guess which die I used. What is their posterior probability for each die?
2. A random variable X can take on the values $0, 1, 2$ or 3. A prior is suggested by an expert such that the probability of getting a 1 is double that of getting a 0, the probability of getting a 2 is double that of getting a 1, *etc*. Give this prior in the form of a proper prior (sums to 1).
3. Suppose this random variable takes on the values $1, 2, \ldots, n$ instead, what is the prior now?

Exercises 6

Previous Exercises | Next Exercises

1. The mean IQ of a random country is assumed to be $N(100, 5^2)$. The IQs of 10000 people are measured in a specific country that has a known standard deviation of 15 IQ points between people and the average is found to be 94. What is the posterior distribution of the mean IQ of that specific country?
2. Derive the log density and log likelihood of the Weibull distribution with density $c\gamma x^{\gamma-1}\exp(-cx^{\gamma})$.
3. Derive the joint log posterior distribution of the parameters of the Weibull distribution, assuming an $Exp(\lambda_1)$ prior for c and an $Exp(\lambda_2)$ prior for γ.
4. What is the log posterior of the χ^2_m distribution with prior $Exp(v)$ on m?

Overview - click to jump

Posterior predictive distribution (critically important!)

In general the distribution of a future sample given the previously observed sample, assuming you DON'T know the parameter values (as is usually the case in real life), is:

$$\pi(\mathbf{x}^*|\mathbf{x}) = \int_{\underline{\theta}} f(\mathbf{x}^*|\underline{\theta})\pi(\underline{\theta}|\mathbf{x})d\underline{\theta}$$

This is called the **posterior predictive distribution**.
This integration is usually impossible. Instead we follow these steps:

- Simulate M sets of parameter values from the posterior.
- For each set of parameter values, simulate new values from the density or likelihood.
- Put these new values together as an approximate posterior predictive distribution and use them to answer questions.

Prediction Example

Let's go back to our Bernoulli example: Given that the last three flips were all tails, what is the probability that the next flip is a head?

$$\begin{aligned}
&\pi(x^* = 1|[0,0,0]) \\
&= \frac{1}{3}^1 * (1 - 1/3)^0 * 0.7032967 + 0.5^1 * (1 - 0.5)^0 * 0.2967033 \\
&= 0.3827839
\end{aligned}$$

By simulation

```
post <- discrete.posterior(c(1/3,0.5),c(0.5,0.5),c(0,0,0),
    myLikelihood)
postsims <- sample(c(1/3,0.5),1000000,T,post)
predsims <- (runif(1000000) < postsims)
mean(predsims)
```

we get 0.3828861, which is close enough.

Calculating probabilities from simulations

- Consider a vector of simulated values $s_1, \ldots, s_m$ and a second vector $t_1, \ldots, t_m$ that is the same length and may be dependent.
- $P(S > 4) =$

```
mean(svec > 4)
```

- $P(2 \leq S < 4) =$

```
mean((2 <= svec) & (svec < 4))
```

- $P(S > T + 2) =$

```
mean(svec > (tvec + 2))
```

- $P(S = 0 \bigcup T \neq 0) =$

```
mean((svec == 0) | (tvec != 0))
```

- Now suppose some values of t are missing, then $P(T > t_0) =$

```
(sum(tvec>t0,na.rm=T))/sum(!is.na(tvec))
```

Overview - click to jump

Simple optimisation

- To find the minimum of a function (say $y = x^2$) we can use numeric optimisation.
- First we code the function as a function, then we find a reasonable starting value.

```
objfunc <- function(x) {x^2}
optim(1, objfunc)
```

- We can restrict the domain if necessary or convenient:

```
optim(1, objfunc, method="Brent", lower=-100 ,upper=100)
```

- We can also use this method to solve equations.
- Say we want a solution to the equation $sin(x) + ln(x) = 1$

```
objfunc <- function(x) {(sin(x)+log(x)-1)^2}
optim(1, objfunc, method="Brent", lower=0, upper=pi/2)
```

Multidimensional optimisation

- Most importantly, 'optim' works for multidimensional problems too.
- Suppose we want the maximum point of the upside down cone like shape $z = -5x^2 - x + 2 - 4y^2 - 2y$.
- We code the negative of the function so that the maximum becomes the minimum, since 'optim' only finds the minimum.

```
objfunc <- function(vars) {z = -5*vars[1]^2-vars[1]+2-4*vars
    [2]^2-2*vars[2]; return(-z)}
optim(c(1,2),objfunc,method="L-BFGS-B",lower=c(-Inf,-Inf),
    upper=rep(Inf,2))$par
```

Always work on the log scale, continued

- In practice, always work with the log posterior as long as possible — it's more accurate and stable.
- When the log posterior is known to a constant only (usually the case) then feel free to subtract or add any constant you wish in order to avoid getting *NaN* when you take the exponent.
- Explicitly: if $\log \pi(\underline{\theta}|\mathbf{x}) = \log Lik(\mathbf{x}|\underline{\theta}) + \log \pi(\underline{\theta}) + c$ then choose $c \approx \max [\log \pi(\underline{\theta}|\mathbf{x}, c = 0)]$.
- For optimisation it is important to remember that the minimum/maximum of the objective function is in exactly the same place as the minimum/maximum of the log objective function!
- Lastly, use approximations in extreme cases, *e.g.* $\log(1 + e^{\eta}) \approx \eta \; \forall \; \eta > 709$.

Exercises 7

Previous Exercises | Next Exercises

1. Consider again the firm with 4 secretaries. If 7 calls are answered in a particular hour, what is the distribution of the possible number of calls to be answered in the next hour?
2. Summarise the posterior predictive distribution for the number of cars question, for each of the *Gamma* priors.
3. Find at least 2 examples in the Bayes literature where the posterior predictive distribution simplifies to a common known distribution.
4. For the IQ problem, summarise the posterior predictive distribution of the next person to be tested in that country.
5. Summarise the posterior predictive distribution of the mean of the next 10 persons to be tested in that country.
6. Summarise the posterior predictive distribution of the standard deviation of the next 100 persons to be tested in that country.

Exercises 8

Previous Exercises | Next Exercises

1. Consider the log posterior of the χ^2_m distribution with prior $Exp(0.04)$ on m. The following 20 observations are recorded: 10 14 19 12 15 11 9 22 12 22 16 12 12 22 15 8 15 8 19 23. Summarise both the posterior distribution of m as well as the posterior predictive distribution of the next potential observation.
2. Illustrate the joint distribution of the next 2 potential observations. Then illustrate the joint distribution of the next 3 potential observations, either 2 at a time or all 3 dimensions at once.
3. Suppose conviction rates in South African courtrooms vary from one judge to another according to a $Beta(2, 5)$ distribution, and that a particular judge dismisses 8 out of 15 cases heard in one day. What is the probability that the same judge dismisses more than 8 out of the 12 cases on their docket the next day?

Overview - click to jump

Posterior Mean

- Let's go back to our $Beta(a, b)$ prior.
- Under Squared Error Loss (an answer twice as far from the truth is 4 times as bad) the optimal estimate of a parameter (in this case p) is its posterior mean.
- There are three ways to calculate it:

1 **Analytically.** Since the posterior is *Beta*, and the mean of a *Beta* distribution is known in general, it's easy to see that the posterior mean should be $\frac{a+\sum x_i}{a+b+n}$.

```
postmean <- (a + sum(x))/(a + b + n)
```

Posterior Mean Continued

2 **Discretization.** If we calculate the posterior at a set of points in the parameter domain, we can estimate the posterior mean $\left(\int_{\underline{\theta}} \underline{\theta}\pi(\underline{\theta}|\mathbf{x})d\underline{\theta}\right)$ as $\sum_{i=1}^{acc.} \underline{\theta}_i\pi(\underline{\theta}_i|\mathbf{x})$.

```
postmean <- sum(p*post)
```

3 **Simulation.** If we simulate a large sample from the posterior we can average the simulated values.

```
postsims <- rbeta(100000, (a+sum(x)), (b + n - sum(x)))
postmean <- mean(postsims)
```

- Simulation is the most common way to calculate the posterior mean in general.
- If there is more than one parameter then the mean must be calculated one parameter at a time.

Posterior Median

- Under Absolute Error Loss (an answer twice as far from the truth is 2 times as bad) the optimal estimate of a parameter (in this case p) is its posterior median.
- There are three ways to calculate it:

1 **Invert the CDF.** Posterior Median $= F^{-1}(0.5)$ where $F = \int \pi(\underline{\theta}|\mathbf{x})\underline{\theta}$.

```
postmedian <- qbeta(0.5,(a+sum(x)), (b + n - sum(x)))
```

Posterior Median Continued

2. **Discretization.** If we calculate the posterior at a set of points in the parameter domain, we can estimate the posterior median by calculating the cumulative sum and finding the first point where this sum exceeds half.

```
F <- cumsum(post)/sum(post)
postmedian <- p[match(1, F>0.5)]
```

3. **Simulation.** The median is approximately the middle sorted simulation.

```
postmedian <- median(postsims)
```

- If there is more than one parameter then the median must be calculated one parameter at a time (remembering to standardise the marginals correctly).

Posterior Mode

- The posterior mode is the analogue of the maximum likelihood (precisely the same in the case of a Uniform prior) and is optimal under all-or-nothing loss.
- **NB:** The mode must be calculated for all parameters simultaneously (not like the mean and median).
- There are three ways to get the posterior mode:

1. **Differentiation.** In very simple cases we can set the derivatives equal to zero and solve for the parameters.
2. **Discretization.** If we calculate the posterior at a set of points in the parameter domain, we can estimate the posterior mode using the 'which.max' function in R:

```
postmode <- p[which.max(post)]
```

Posterior Mode Continued

3 **Optimisation.** We can find the peak of the posterior distribution using optimisation techniques from mathematics. The steps are as follows:

3.1 Code the negative posterior, or better yet, the negative log posterior as a function on its own. Remember that the posterior mode is in the same place as the log posterior mode. The negative is necessary because most optimisation functions try to find the minimum (not maximum).

```
Bernoulli.neglogpost <- function(p,x,n,a,b) {
lpost <- (sum(x)+a-1)*log(p) + (n-sum(x)+b-1)*log(1-p) +
    lgamma(n+a+b) - lgamma(sum(x)+a) - lgamma(n-sum(x)+b)
# OR lpost <- dbeta(p,(sum(x)+a),(n-sum(x)+b),log=TRUE)
return(-lpost)}
```

Optimisation Continued

3.2 *(Optional)* Code the posterior derivative vector as a function on its own (otherwise numeric differentiation will be used). Again remember the negatives.

3.3 Find suitable starting values, say using the method of moments.

3.4 Figure out whether there are constraints on the parameters.

3.5 Use an optimisation technique built into your statistical package, or code your own. In R we usually use the function 'optim'.

```
postmode <- optim((sum(x)/n),Bernoulli.neglogpost,x=x,n=n,a=a,
    b=b,method='L-BFGS-B',lower=1e-15,upper=(1-1e-15))$par
```

Accuracy Statistics

In general, if model A gives point estimates that are closer to the observed values than the point estimates of model B then model A is more accurate than model B.
Consider errors $e_i = y_i - \hat{y}_i$ and percentage errors $pe_i = \frac{e_i}{|y_i|}$ then useful statistics include

- $ME = \frac{1}{n}\sum_{i=1}^{n} e_i$ and MPE for measuring bias,
- $MAE = \frac{1}{n}\sum_{i=1}^{n} |e_i|$ and $MAPE$ for measuring the distance between observed and predicted values,
- $MSE = \frac{1}{n}\sum_{i=1}^{n} (e_i)^2$ and $MSPE$ for measuring the deviation between observed and predicted values and
- $RMSE = \sqrt{MSE}$ and $RMSPE$ for viewing the MSE on the same scale as the data and other statistics.

In the case of parameter estimation, we can simulate data so that we know the true parameter values and then see which method gets closer to those values. In that case $y_i = \theta$ and $\hat{y}_i = \hat{\theta}_i$.

Reparameterisation or equating parameters

- Textbooks, papers, websites and programming languages often give or require the parameters of standard distributions in a specific form.
- One must always be careful when implementing a problem from a text in a computer program to convert the given parameter values to the form required by the program.
- For example, the Actuarial Education Company gives the Weibull density as $f(x|c,\gamma) = c\gamma x^{\gamma-1}\exp(-cx^{\gamma})$;
- while R requires the Weibull density in the form $f(x|a,b) = \frac{a}{b}\left(\frac{x}{b}\right)^{a-1}\exp\left[-\left(\frac{x}{b}\right)^{a}\right]$.
- Since these need to be equivalent, we can equate any components of these formulae.
- $cx^{\gamma} \equiv \left(\frac{x}{b}\right)^{a} = b^{-a}x^{a}$
- Thus, $a = \gamma$ and $b = c^{-a^{-1}}$.
- This technique is very useful in calculating specific integrals.

Exercises 9

Previous Exercises | Next Exercises

1. Simulate 1200 samples of size 40 from a $Poisson(8)$ distribution. Assume a Uniform prior. For each sample calculate the posterior mean, median and mode in as many ways as you can. Calculate the ME, MAE and RMSE for λ.
2. Simulate 1200 samples of size 40 from a $Weibull(0.12, 0.5)$ distribution. Assume a Uniform prior for c and that γ is fixed at 0.5. For each sample calculate the posterior mean, median and mode in as many ways as you can. Calculate the ME, MAE and RMSE for c.
3. Simulate 1200 samples of size 40 from a $NegBin2(8, 0.8)$ distribution. Assume a Uniform prior for p and that k is fixed at 8. For each sample calculate the posterior mean, median and mode in as many ways as you can. Calculate the ME, MAE and RMSE for p.

Overview - click to jump

Equal tailed intervals

- A point estimate is always wrong!
- Intervals are better.
- Bayesian Credibility intervals are even better still! This is because they give probabilities.
- Equal tailed intervals are the easiest because you know which quantiles you need, *e.g.* the 95% interval goes from 2.5% to 97.5%.

```
(postinterval <- qbeta(c(0.025,0.975),(a+sum(x)), (b + n - sum
    (x))))
# OR
(postinterval <- c(p[match(1, F>0.025)],p[match(1, F>0.975)]))
# OR
(postinterval <- quantile(postsims,c(0.025,0.975)))
```

Highest posterior density intervals

- Equal tailed intervals are not the best intervals.
- You can get shorted intervals for a given probability and shorter intervals are better because they convey more information.
- For example, for an Exponential distribution the shortest 95% interval is from 0% to 95%, not 2.5% to 97.5%.
- In general, the shortest interval is tough to find analytically, and may require nested optimisation, or approximation via simulation (see next slide).

HPD intervals by simulation

Consider every interval

$$\left(x_{\left(\frac{i}{n+1}\right)}, x_{\left(\frac{i+(1-\alpha)n}{n+1}\right)}\right), i = 1, \ldots, \lfloor \alpha n \rfloor$$

and see which one is the shortest $\left(x_{\left(\frac{i+(1-\alpha)n}{n+1}\right)} - x_{\left(\frac{i}{n+1}\right)} \text{ is a minimum}\right)$.

```
nsims <- 10000
postsims <- rbeta(nsims, 3.5, 5)
sorted.postsims <- sort(postsims)
numints <- floor(nsims*0.05)
gap <- round(nsims*0.95)
widths <- sorted.postsims[(1+gap):(numints+gap)] - sorted.postsims
    [1:numints]
HPD<- sorted.postsims[c(which.min(widths),(which.min(widths)+gap))]
```

Exercises 10

Previous Exercises | Next Exercises

1. Simulate 1200 samples of size 40 from a *Poisson*(8) distribution. Assume a Uniform prior. For each sample obtain a symmetric and an HPD interval for λ. On average, what proportion of the intervals cover the true parameter value?
2. Simulate 1200 samples of size 40 from a *Weibull*$(0.12, 0.5)$ distribution. Assume a Uniform prior for c and that γ is fixed at 0.5. For each sample obtain a symmetric and an HPD interval for c. On average, what proportion of the intervals cover the true parameter value?
3. Simulate 1200 samples of size 40 from a *NegBin2*$(8, 0.8)$ distribution. Assume a Uniform prior for p and that k is fixed at 8. For each sample obtain a symmetric and an HPD interval for p. On average, what proportion of the intervals cover the true parameter value?

seanvdm.co.za
UNIVERSITY OF THE FREE STATE
UNIVERSITEIT VAN DIE VRYSTAAT
YUNIVESITHI YA FREISTATA
UFS

Overview - click to jump

Introduction to Objective Priors

- The idea with objective priors is to let the data speak for itself.
- They are useful when we have no prior knowledge of the parameters but still want to apply Bayesian principles.
- They sometimes allow Bayes results to match frequentist properties.
- They are often the limit of a subjective prior as the variance of that prior goes to infinity (or closest substitute).
  - For example, in the Bernoulli case the *Uniform* prior is the same as the *Beta*(α, β) prior with $\alpha = \beta = 1$.

MDI Prior (Zellner, 1997)

Definition

$$\pi(\theta) = \exp\left\{E_X\left[\log f(x|\theta)\right]\right\}$$
$$\therefore \log \pi(\theta) = E_X\left[\log f(x|\theta)\right]$$
$$\therefore \log \pi(\theta|\mathbf{x}) = E_X\left[\log f(x|\theta)\right] + \sum_{i=1}^{n} \log f(x_i|\theta)$$

Example — Gamma Distribution

$$f(x|\theta) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}$$
$$\log f(x|\theta) = \alpha \log \lambda - \log \Gamma(\alpha) + (\alpha - 1)\log x - \lambda x$$
$$\therefore \log \pi(\theta) = \alpha \log \lambda - \log \Gamma(\alpha) + (\alpha - 1)(\psi(\alpha) - \log \lambda) - \alpha$$

Jeffreys' Prior (Jeffreys, 1998)

Square root of determinant of Fischer Information matrix.

Definition

Let $g = -\log f(x|\theta)$

$$\pi(\theta) \propto \left| \begin{matrix} E\frac{\partial g}{\partial \theta_1 \theta_1} & E\frac{\partial g}{\partial \theta_1 \theta_2} & \cdots & E\frac{\partial g}{\partial \theta_1 \theta_k} \\ E\frac{\partial g}{\partial \theta_2 \theta_1} & E\frac{\partial g}{\partial \theta_2 \theta_2} & & \vdots \\ \vdots & & \ddots & \vdots \\ E\frac{\partial g}{\partial \theta_k \theta_1} & \cdots & \cdots & E\frac{\partial g}{\partial \theta_k \theta_k} \end{matrix} \right|^{\frac{1}{2}}$$

Jeffreys' Prior Example - Poisson

$$f(x|\lambda) = \frac{e^{-\lambda}\lambda^x}{x!}$$

Using the fact that $x! = \Gamma(x+1)$,

$$g = -\log f(x|\lambda) = \lambda + \log\Gamma(x+1) - x\log\lambda$$

and so,

$$\frac{\partial g}{\partial \lambda} = 1 - \frac{x}{\lambda},$$

$$\frac{\partial^2 g}{\partial \lambda^2} = \frac{x}{\lambda^2},$$

$$E\left[\frac{\partial^2 g}{\partial \lambda^2}\right] = \frac{1}{\lambda},$$

and Jeffreys Prior $\propto \lambda^{-0.5}$.

Jeffreys' Independence Prior

Same as Jeffreys' prior but using only diagonal of Fischer matrix.

Definition

$$\pi(\theta) \propto \left\{ \prod_{i=1}^{k} -E \frac{\partial^2 \log f(x|\theta)}{\partial \theta_i^2} \right\}^{\frac{1}{2}}$$

$$\log \pi(\theta) = \frac{1}{2} \left\{ \sum_{i=1}^{k} \log -E \frac{\partial^2 \log f(x|\theta)}{\partial \theta_i^2} \right\} + c$$

MDI and Jeffreys' Prior Example - GPD

The Generalised Pareto Distribution is used to model the tail (extreme values) of a distribution beyond a given threshold. If the threshold is known we can subtract it from all the observations and consider it to be zero. This makes the GPD a 2-parameter distribution.

$$f(x|\gamma,\sigma) = \frac{1}{\sigma}\left[1+\frac{\gamma x}{\sigma}\right]^{-\frac{1}{\gamma}-1}, 0 < x < \begin{cases} -\frac{\sigma}{\gamma} \text{ if } \gamma < 0 \\ \infty \text{ if } \gamma > 0 \end{cases}$$

For $\gamma < 1$ we know that $E(X) = \frac{\sigma}{1-\gamma}$ and for $\gamma < 0.5$ we know that $Var(X) = \frac{\sigma^2}{(1-\gamma)^2(1-2\gamma)}$.

Putting these together we see that $E(X^2) = \frac{2\sigma^2}{(1-\gamma)(1-2\gamma)}$.

GPD Example p.2

In order to derive the objective priors we need some initial results.
Let $A = E\left[\log\left(1+\frac{\gamma x}{\sigma}\right)\right]$, then $A = \int_0^\infty \log\left(1+\frac{\gamma x}{\sigma}\right)\frac{1}{\sigma}\left(1+\frac{\gamma x}{\sigma}\right)^{-\frac{1}{\gamma}-1} dx$.
Applying integration by parts, $A =$

$$\left[\log\left(1+\frac{\gamma x}{\sigma}\right)(-1)\left(1+\frac{\gamma x}{\sigma}\right)^{-\frac{1}{\gamma}}\right]_0^\infty - \int_0^\infty \frac{\gamma}{\sigma}\left(1+\frac{\gamma x}{\sigma}\right)^{-1}\left[-\left(1+\frac{\gamma x}{\sigma}\right)^{-\frac{1}{\gamma}}\right] dx$$

Applying L'Hôpital to the first term we find it to be zero:

$$\lim_{x\to\infty} \log\left(1+\frac{\gamma x}{\sigma}\right)\left(1+\frac{\gamma x}{\sigma}\right)^{-\frac{1}{\gamma}} \doteq \lim_{x\to\infty}\left(1+\frac{\gamma x}{\sigma}\right)^{-1}\left(1+\frac{\gamma x}{\sigma}\right)^{-\frac{1}{\gamma}+1}\sigma$$

$$\doteq \lim_{x\to\infty}\left(1+\frac{\gamma x}{\sigma}\right)^{-\frac{1}{\gamma}}\sigma = 0$$

$$\therefore A = \gamma\int_0^\infty \frac{1}{\sigma}\left(1+\frac{\gamma x}{\sigma}\right)^{-\frac{1}{\gamma}-1} dx = \gamma$$

If $\gamma < 0$ then replace ∞ with $-\sigma/\gamma$ everywhere. The L'Hôpital step becomes unnecessary and we arrive at the same answer.

GPD Example p.3

Let $B = E\left[x\left(1+\frac{\gamma x}{\sigma}\right)^{-1}\right] = \int_0^\infty x\left(1+\frac{\gamma x}{\sigma}\right)^{-1}\frac{1}{\sigma}\left(1+\frac{\gamma x}{\sigma}\right)^{-\frac{1}{\gamma}-1}dx$, then

$B = \int_0^\infty x\frac{1}{\sigma}\left(1+\frac{\gamma x}{\sigma}\right)^{w}dx$, where

$w = -\frac{1}{\gamma}-1-1 = -\frac{1+\gamma}{\gamma}-1 = -\frac{1}{\frac{\gamma}{1+\gamma}}-1.$

Let $\gamma^* = \frac{\gamma}{1+\gamma}$ and $\sigma^* = \frac{\sigma}{1+\gamma} \Rightarrow \frac{1}{\sigma} = \frac{1}{1+\gamma}\frac{1}{\sigma^*}$, then

$B = \frac{1}{1+\gamma}\int_0^\infty x\frac{1}{\sigma^*}\left(1+\frac{\gamma^* x}{\sigma^*}\right)^{-\frac{1}{\gamma^*}-1}dx = \frac{1}{1+\gamma}\frac{\sigma^*}{1-\gamma^*} = \frac{\sigma}{1+\gamma}$

Now let $C = E\left[x^2\left(1+\frac{\gamma x}{\sigma}\right)^{-2}\right]$, then

$C = \int_0^\infty x^2\frac{1}{\sigma}\left(1+\frac{\gamma x}{\sigma}\right)^{v}dx$, where

$v = -\frac{1}{\gamma}-1-2 = -\frac{1+2\gamma}{\gamma}-1 = -\frac{1}{\frac{\gamma}{1+2\gamma}}-1.$

Let $\tilde{\gamma} = \frac{\gamma}{1+2\gamma}$ and $\tilde{\sigma} = \frac{\sigma}{1+2\gamma} \rightarrow \frac{1}{\sigma} = \frac{1}{1+2\gamma}\frac{1}{\tilde{\sigma}}$, then

$C = \frac{1}{1+2\gamma}\int_0^\infty x^2\frac{1}{\tilde{\sigma}}\left(1+\frac{\tilde{\gamma}x}{\tilde{\sigma}}\right)^{-\frac{1}{\tilde{\gamma}}-1}dx$

$= \frac{1}{1+2\gamma}\frac{2\tilde{\sigma}^2}{(1-\tilde{\gamma})(1-2\tilde{\gamma})} = \frac{2\sigma^2}{(1+\gamma)(1+2\gamma)}$

GPD Example p.4

$$\log f(x) = -\log\sigma - \left(\frac{1}{\gamma}+1\right)\log\left(1+\frac{\gamma x}{\sigma}\right)$$

$$\log MDIP = -\log\sigma - \left(\frac{1}{\gamma}+1\right)E\left[\log\left(1+\frac{\gamma x}{\sigma}\right)\right] + c$$

$$= -\log\sigma - \left(\frac{1}{\gamma}+1\right)A + c$$

$$= -\log\sigma - \left(\frac{1}{\gamma}+1\right)\gamma + c$$

$$= -\log\sigma - 1 - \gamma + c$$

$$\therefore MDIP \propto \frac{e^{-\gamma}}{\sigma}$$

GPD Example p.5

Left $g = -\log f(x) = \log\sigma\left(\frac{1}{\gamma}+1\right)\log\left(1+\frac{\gamma x}{\sigma}\right)$, then

$$\frac{\partial g}{\partial \gamma} = -\gamma^{-2}\log\left(1+\frac{\gamma x}{\sigma}\right) + \left(\frac{1}{\gamma}+1\right)\left(1+\frac{\gamma x}{\sigma}\right)^{-1}\frac{x}{\sigma}$$

$$\frac{\partial g}{\partial \sigma} = \sigma^{-1} + \left(\frac{1}{\gamma}+1\right)\left(1+\frac{\gamma x}{\sigma}\right)^{-1}(-\gamma x\sigma^{-2})$$

$$= \sigma^{-1} - \sigma^{-2}(1+\gamma)(x)\left(1+\frac{\gamma x}{\sigma}\right)^{-1}$$

$$\frac{\partial^2 g}{\partial \gamma^2} = 2\gamma^{-3}\log\left(1+\frac{\gamma x}{\sigma}\right) + 2\left[(-\gamma^{-2})\left(1+\frac{\gamma x}{\sigma}\right)^{-1}\frac{x}{\sigma}\right]$$

$$+\left(\frac{1}{\gamma}+1\right)\left(1+\frac{\gamma x}{\sigma}\right)^{-2}\left(\frac{x}{\sigma}\right)^2(-1)$$

$$= 2\gamma^{-3}\log\left(1+\frac{\gamma x}{\sigma}\right) - 2(\gamma^{-2}\sigma^{-1})(x)\left(1+\frac{\gamma x}{\sigma}\right)^{-1}$$

$$-(\gamma^{-1}\sigma^{-2})(1+\gamma)(x^2)\left(1+\frac{\gamma x}{\sigma}\right)^{-2}$$

GPD Example p.6

$$\frac{\partial^2 g}{\partial \sigma^2} = -\sigma^{-2} + 2\sigma^{-3}(1+\gamma)(x)\left(1+\frac{\gamma x}{\sigma}\right)^{-1} - \sigma^{-4}\gamma(1+\gamma)(x^2)\left(1+\frac{\gamma x}{\sigma}\right)^{-2}$$

$$\frac{\partial^2 g}{\partial \gamma \partial \sigma} = \sigma^{-3}(1+\gamma)(x^2)\left(1+\frac{\gamma x}{\sigma}\right)^{-2} - \sigma^{-2}(x)\left(1+\frac{\gamma x}{\sigma}\right)^{-1}$$

$$\begin{aligned} E\left[\frac{\partial^2 g}{\partial \gamma^2}\right] &= 2\gamma^{-3}A - 2(\gamma^{-2}\sigma^{-1})B - (\gamma^{-1}\sigma^{-2})(1+\gamma)C \\ &= 2\gamma^{-2} - 2\gamma^{-2}(1+\gamma)^{-1} - 2\gamma^{-1}(1+2\gamma)^{-1} \\ &= \frac{2}{(1+\gamma)(1+2\gamma)} \end{aligned}$$

$$E\left[\frac{\partial^2 g}{\partial \sigma^2}\right] = -\sigma^{-2} + 2\sigma^{-3}(1+\gamma)B - \sigma^{-4}\gamma(1+\gamma)C = \frac{1}{\sigma^2(1+2\gamma)}$$

$$E\left[\frac{\partial^2 g}{\partial \gamma \partial \sigma}\right] = \sigma^{-3}(1+\gamma)C - \sigma^{-2}B = \frac{1}{\sigma(1+\gamma)(1+2\gamma)}$$

GPD Example p.7

Independence Jeffreys Prior $\propto$

$$\sqrt{2(1+\gamma)^{-1}\sigma^{-2}(1+2\gamma)^{-2}}$$

$$= \sigma^{-1}(1+2\gamma)^{-1}\sqrt{2(1+\gamma)^{-1}}$$

Jeffreys' Prior $\propto$

$$\sqrt{2(1+\gamma)^{-1}\sigma^{-2}(1+2\gamma)^{-2} - \sigma^{-2}(1+\gamma)^{-2}(1+2\gamma)^{-2}}$$

$$= \sqrt{(1+\gamma)^{-2}\sigma^{-2}(1+2\gamma)^{-1}} = (1+\gamma)^{-1}\sigma^{-1}(1+2\gamma)^{-0.5}$$

Reference Priors

- Attempts to reduce the dependence between parameters induced by the standard Jeffreys prior, which can cause problems.
- Not as aggressive as the independence Jeffreys prior.
- Tries to maximise the expected divergence between the posterior and prior.
- In one dimensional case this turns out to be the same as the Jeffreys.
- With multiple parameters first do nuisance parameters conditional on target parameter.
- Reference Priors depend on the experimental design.

Berkeley Lectures on Objective Priors

- http://www.cs.berkeley.edu/~jordan/courses/260-spring10/lectures/lecture7.pdf
- http://www.cs.berkeley.edu/~jordan/courses/260-spring10/lectures/lecture8.pdf
- http://www.cs.berkeley.edu/~jordan/courses/260-spring10/lectures/lecture9.pdf
- http://www.cs.berkeley.edu/~jordan/courses/260-spring10/lectures/lecture10.pdf

Probability Matching Prior

- Posterior probabilities of certain regions should coincide with their coverage probabilities.
- The idea is that we would like the 95% interval to cover the 'true' value 95% of the time.
- The Probability Matching Prior (PMP) does this better than other priors.
- http://www.ucl.ac.uk/statistics/research/pdfs/rr252.pdf
- http://www.utstat.utoronto.ca/reid/research/cjs.staicu_reid.pdf

Exercises 11

Previous Exercises | Next Exercises

Note that not all priors can be derived for all distributions.

1. Derive objective prior(s) for the Poisson distribution.
2. Derive objective prior(s) for the Binomial distribution.
3. Derive objective prior(s) for the Multinomial distribution.
4. Derive objective prior(s) for the Beta distribution.
5. Derive objective prior(s) for the Dirichlet distribution.
6. Derive objective prior(s) for the Gumbel distribution.

Overview - click to jump

Why Subjective Priors

- Let's you incorporate expert knowledge into your model.
- Let's you put in fuzzy boundaries to stop nonsense results.
- Proper subjective priors guarantee that the posterior will be proper.
- Helps ensure your parameters are identifiable in some problems.
- Allows for model comparison using Bayes Factors and *DIC*.
- **BUT** regardless of what priors you use,
- remember to repeat your analysis for different priors and see whether the results change (sensitivity testing). See http://onlinelibrary.wiley.com/doi/10.1002/sim.2112/pdf for an example.

Easy Univariate Priors

- For parameters without restriction: *Normal* prior with low precision.
  - Choose mean to be some 'middle' value *WITHOUT* considering the data.
  - Some packages (like OpenBUGS) ask for the precision $\tau = \frac{1}{\sigma^2} = \sigma^{-2}$ (sometimes defined as τ^2). R and STAN use the standard deviation.
- For strictly positive parameters: *Exponential* prior with high mean works well (small scale parameter).
- For probability parameters ($0 < p < 1$): *Beta* prior with low parameter values.
  - Usually equal parameters so that the mean is 0.5.
  - This is a rare case where objective priors are proper and can be used in subjective Bayes.
  - The *Uniform* prior works just fine.
  - In this case the Jeffreys prior is $Beta(0.5, 0.5)$ and comes highly recommended.

Gamma Prior

- For strictly positive parameters, consider the $Gamma(0.001, 0.001)$ prior.
- If $\theta \sim Gamma(0.001, 0.001)$ then $\pi(\theta)$ is approximately proportional to θ^{-1}.
  - $\pi(\theta) = \frac{0.001^{0.001}}{\Gamma(0.001)}\theta^{-0.999}\exp(-0.001\theta) \approx \frac{1}{1000}\theta^{-1}.1$
- θ^{-1} is the Jeffreys prior for scale or precision parameters in many cases.
  - Consider the Normal density with mean 0 and precision τ
  - $g = \log f(x) = 0.5\log\tau - 0.5\tau x^2 + c$
  - $\frac{dg}{d\tau} = 0.5\tau^{-1} - 0.5x^2$ and $\frac{d^2g}{d\tau^2} = -0.5\tau^{-2}$,
  - so $-E\left(\frac{d^2g}{d\tau^2}\right) = 0.5\tau^{-2}$ and $\sqrt{-E\left(\frac{d^2g}{d\tau^2}\right)} \propto \tau^{-1}$.
  - As an exercise, show that the same result holds for standard deviation σ instead.
- This prior can cause issues in more complex models. The half Cauchy prior is usually better.

Easy Multivariate Priors

- For parameters without restriction: Multivariate *Normal* prior with low precision parameters.
- For strictly positive parameters: Independent *Exponential* priors with high means.
- For probability parameters ($0 < \mathbf{p} < 1$): *Dirichlet* prior with low parameter values if they sum to one, or independent *Beta* priors if they don't.
- And don't forget that conjugate priors are always nice:
  - see http://en.wikipedia.org/wiki/Conjugate_prior

Overview - click to jump

The classical p-value

- The classical p-value is the probability of observing a value of a statistic as or more extreme than what we observed from the data, assuming the null hypothesis holds.
- The null hypothesis is usually the neutral stance, often chosen as a matter of convenience.
- The idea is that a low p-value is unlikely under the null hypothesis and thus provides evidence against it.
- The null hypothesis is usually simple, facilitating clear mathematical results and easy calculations.
- The fallacy (logical error) is that evidence against the null hypothesis is necessarily evidence for the alternative.
- This led to the ban of the p-value from the journal Basic and Applied Social Psychology.

Bayes' Theorem applied to hypotheses

- In most studies we are trying to support the alternative.
  - If your case is different then simply swap H_0 and H_1 below.
- $P(H_1|S \geq s) = \frac{P(S \geq s|H_1)P(H_1)}{P(S \geq s|H_0)P(H_0)+P(S \geq s|H_1)P(H_1)}$
- We can try to be objective and set $P(H_0) = P(H_1) = 0.5$, then $P(H_1|S \geq s) = \frac{P(S \geq s|H_1)}{p+P(S \geq s|H_1)}$.
  - This might not always be appropriate, if 4 out of 5 previous studies supported H_0 then this should affect the prior probabilities.
- The important thing to note here is that the absolute value of p is irrelevant, only its size relative to the corresponding probability under the alternative hypothesis.
- No matter how unlikely a statistic is, if it's just as unlikely under the alternative then the posterior probability of H_1 is still 0.5.

The popularity of the p-value

- In spite of the issues on the previous slides, the p-value is the most popular way of reporting empirical results.
- In general it can be very difficult or impossible to calculate $P(S \geq s|H_1)$ and so people just stop at the p-value.
- The p-value, on the other hand, is often easy to calculate, and Bayes can make this calculation even easier or more accurate.

Final warning though, calculating multiple p-values on the same data usually invalidates any conclusions that are drawn, more so as the number of tests increases, unless some appropriate adjustment is made to account for this.

Region of practical equivalence (ROPE)

In frequentist hypothesis testing we first test statistical significance, and then when we have statistical significance we try to judge practical significance as a second step (usually without using statistics).

In a real world situation you can often come up with a practical notion of what is means for two things to be the same or different. **Bayes allows us to connect probabilities to practical, real-world, effects.**

- Instead of having a point null hypothesis (*e.g.* $\mu = 0$) we have a region of practical equivalence.
- We can calculate probabilities for the region directly.
- Or compare the region to intervals from our models.
- This lets us actually *accept* a null hypothesis instead of merely *failing to reject it*.

For a good explanation see The bayestestR vinette

Overview - click to jump

Missing data

- In EXCEL missing data should always be coded as blank (an empty cell). In R this translates to a 'NA' in place of each missing value.
- When only a few values are missing it is common (although not ideal) to drop problematic observations from the data set (can cause bias).
- In cases where a large proportion of the data is missing it may be preferable to separate the modelling step from the imputation step.
- The fully Bayesian version of the separation concept works as follows:
  1. Impute or otherwise obtain a large number of filled in data sets that adequately capture the uncertainty in both the missingness mechanism and the imputation mechanism.
  2. Fit the Bayesian model normally using these complete data sets one at a time.
  3. For each fit obtain random draws from the joint posterior distribution of the parameters of interest.
  4. Combine the draws from all data sets to obtain a picture of the joint posterior that takes all the missing data uncertainties into account.

Long form data

- In general data is in long form when the observed values of interest are all listed below each other; while wide form suggests that some values of interest are recorded next to each in the data table.
- Long form is almost always better as it accommodates irregular data structures in an efficient way.
- More importantly, if there is only one value of interest in a row, and that value is missing, then dropping rows with missing values causes no additional loss of information.
- Thus, when the goal is to fit a regression style model on a specific variable, then that variable should ideally be recorded in long form in the data table.
- In this scenario multiple imputation of the missing values is not required in order to retain the available information.

Overview - click to jump

Hierarchical Sampler

- Suppose you have the joint density of 2 parameters that are dependent.
- Can't just use marginals because then you lose dependence, which breaks the posterior predictive distribution.
- Sometimes you can decompose joint density into 1 marginal and at least 1 conditional distribution, *e.g.*
  - $f(\theta_1, \theta_2|\mathbf{x}) = f(\theta_1|\theta_2, \mathbf{x})f(\theta_2|\mathbf{x})$
  - $f(\theta_1, \theta_2, \theta_3|\mathbf{x}) = f(\theta_1, \theta_2|\theta_3, \mathbf{x})f(\theta_3|\mathbf{x})$
  - $f(\theta_1, \theta_2, \theta_3|\mathbf{x}) = f(\theta_1|\theta_2, \theta_3, \mathbf{x})f(\theta_2, \theta_3|\mathbf{x})$
  - $f(\theta_1, \theta_2, \theta_3|\mathbf{x}) = f(\theta_1|\theta_2, \theta_3, \mathbf{x})f(\theta_2|\theta_3, \mathbf{x})f(\theta_3|\mathbf{x})$
  - *etc.*
- Order is not important.

[Note that this subsection is not for examination, it is to help you understand how multivariate simulation works.]

Hierarchical Sampler Example - Gamma Distribution

- Gamma density: $\frac{\lambda^\alpha}{\Gamma(\alpha)} x^{(\alpha-1)} e^{-\lambda x}$
- MDI prior: $\frac{\lambda}{\Gamma(\alpha)} e^{(\alpha-1)\psi(\alpha)-\alpha}$ where $\psi()$ refers to the digamma function.
- Posterior: $c\lambda^{(n\alpha+2)-1} e^{-\lambda \sum x_i} \Gamma(\alpha)^{-(n+1)} e^{\alpha[\sum \log x_i + \psi(\alpha) - 1] - \psi(\alpha)}$
- $\therefore \lambda|\alpha, \mathbf{x} \sim Gamma(n\alpha + 2, \sum x_i)$
  - with density $\frac{(\sum x_i)^{n\alpha+2}}{\Gamma(n\alpha+2)} \lambda^{(n\alpha+2)-1} e^{-\lambda \sum x_i}$
- And, since $\pi(\alpha, \lambda|\mathbf{x}) = \pi(\lambda|\alpha, \mathbf{x})\pi(\alpha|\mathbf{x})$, (by dividing) we see that

$$\pi(\alpha|\mathbf{x}) \propto \Gamma(\alpha)^{-(n+1)} e^{\alpha[\sum \log x_i + \psi(\alpha) - 1] - \psi(\alpha)} \Gamma(n\alpha+2) \left(\sum x_i\right)^{-(n\alpha+2)}$$

Hierarchical Sampler - Gamma Posterior Simulation

The steps for simulating the Gamma posterior are then:

- Simulate a vector of α's by discretization or a better method.
  - Pick a set of candidate values around the moments estimate,
  - calculate density at these values, and
  - simulate from this discrete density.
- For each α, simulate a value from a $Gamma(n\alpha + 2, \sum x_i)$ density.
- Put everything together in a matrix.

```
mean(x)^2/var(x) -> a
asv <- 10^(seq(-0.7,0.9,0.005))*a
lfas <- asv*(sum(log(x)) + digamma(asv) - 1) - digamma(asv) - (n+1)
    *lgamma(asv) - (n*asv+2)*log(sum(x)) + lgamma(n*asv+2)
fas <- exp(lfas - max(lfas))
fas <- fas/sum(fas)
sims <- matrix(1,nsim,2)
sims[,1] <- asv[sample.int(length(asv),nsim,prob=fas)]
sims[,2] <- rgamma(nsim,n*sims[,1]+2,sum(x))
```

Hierarchical Sampler - Gamma Posterior SIR

The steps for simulating the Gamma posterior - alternative:

- Simulate a vector of α's by sampling-importance resampling.

```
lfas <- function(asv,x,n) { asv*(sum(log(x)) + digamma(asv) -
    1) - digamma(asv) - (n+1)*lgamma(asv) - (n*asv+2)*log(sum(
    x)) + lgamma(n*asv+2) }
mean(x)^2/var(x) -> a
asv <- 10^(seq(-0.7,0.9,0.005))*a
fas <- exp(lfas(asv,x,n) - max(lfas(asv,x,n)))
fas <- fas/sum(fas)
postmean <- sum(asv*fas)
postvar <- sum(fas*((asv-postmean)^2))
alpha <- postmean^2/postvar
lambda <- postmean/postvar
candidates <- rgamma(10000,alpha,lambda)
densratio <- post(candidates)/dgamma(candidates,alpha,lambda)
densratio <- densratio/sum(densratio)
sims <- matrix(1,nsim,2)
sims[,1] <- sample(candidates,10000,T,densratio)
sims[,2] <- rgamma(nsim,n*sims[,1]+2,sum(x))
```

Hierarchical Sampler Example - Bayesian OLS Regression

- The model is built in stages as follows:
  1. $Y \sim N(X\beta, \sigma^2)$
  2. $\beta \sim N(\hat{\beta} = (X'X)^{-1}X'y, \Sigma_\beta = (X'X)^{-1}\sigma^2$
  3. $\frac{RSS}{\sigma^2} \sim \chi^2_{n-k}$ where $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- Hence, to implement this model, we will proceed as follows:
  1. Start by simulating a χ^2_{n-k} value.
  2. Divide *RSS* by the simulated value to get a random σ^2 value.
  3. Simulate a random set of βs (random *Normal*s times the symmetric matrix square root of Σ_β, plus $\hat{\beta}$).
  4. Simulate random predictions (X times simulated βs, plus random *Normal*s times σ.
- We repeat the above procedure above a large number of times in order to obtain a set of vector samples from both the posterior and posterior predictive distributions.

Hierarchical Sampler - Bayesian OLS Regression Code

```
sqrtm <- function(x) {
a <- eigen(x,T)
s <- a$vectors %*% diag(sqrt(a$values)) %*% t(a$vectors)
return(s) }

bayesreg <- function(y,x,newx=x,acc=9999) {
n <- nrow(x); k <- ncol(x); k1 <- (k + 1); if (!is.matrix(newx)) { newx <- matrix(newx,
     ncol=k) }; m <- nrow(newx);
basemodel <- lm(y ~ x)
res <- resid(basemodel)
b <- as.matrix(coefficients(basemodel))
if (any(x[,1]!=1)) { x <- matrix(c(rep.int(1,n),x),n); k1 <- (k + 1) }
if (any(newx[,1]!=1)) { newx <- matrix(c(rep.int(1,m),newx),m)}
sigs <- vector(length=acc)
betas <- matrix(nrow=k1, ncol=acc)
newy <- matrix(nrow=m, ncol=acc)
xx <- solve(t(x) %*% x)
rr <- sum(res^2)
for (i in 1:acc) {
  u <- rchisq(1, n - k)
  sig2 <- rr / u
  covb <- xx * sig2
  B <- b + (sqrtm(covb) %*% matrix(rnorm(k + 1),(k + 1)))
  betas[,i] <- B
  sig <- sqrt(sig2)
  sigs[i] <- sig
  newy[,i] <- newx %*% B + matrix(rnorm(m),m)*sig }
return(list(b=b,betas=betas,r=res,sigs=sigs,newy=newy,newyhat=apply(m2$newy,1,mean))) }
```

Gibbs Sampler

- The Gibbs Sampler is simply a theorem that says we can break down a complicated multidimensional simulation problem into simpler problems by going back and forth between the conditional distributions.
- Downside: simulations are not independent!
- Instead we get a Markov Chain (each simulation depends on the previous one but not the ones before that).
- This is referred to as a Markov Chain Monte Carlo simulation technique (MCMC).
- We can counter the dependence by only considering every m^{th} vector of simulated values and dropping the first b vectors of simulated values (called the burn-in period).

Gibbs - Example 1

- Consider simulating from this joint density:

$$f(x_1, x_2) \propto \binom{n}{x_1} x_2^{x_1+\alpha-1}(1-x_2)^{n-x_1+\beta-1}$$

- In this case the conditional distributions are much simpler:

$$f(x_1|x_2) = Binomial(n, x_2)$$
$$f(x_2|x_1) = Beta(x_1 + \alpha, n - x_1 + \beta)$$

- Start with a random x_2, simulate $x_1|x_2$, simulate $x_2|x_1$, ...
- after some time ... simulate $x_1|x_2$, simulate $x_2|x_1$ and store latest (x_1, x_2) vector,
- simulate $x_1|x_2$, simulate $x_2|x_1$, simulate $x_1|x_2$, simulate $x_2|x_1$ and store latest (x_1, x_2) vector,
- simulate $x_1|x_2$, simulate $x_2|x_1$, simulate $x_1|x_2$, simulate $x_2|x_1$ and store latest (x_1, x_2) vector,
- repeat until the joint distribution converges *in distribution*.

Gibbs - Example 1

- Consider simulating from this joint density:

$$f(x_1, x_2) \propto \binom{n}{x_1} x_2^{x_1+\alpha-1}(1-x_2)^{n-x_1+\beta-1}$$

- In this case the conditional distributions are much simpler:

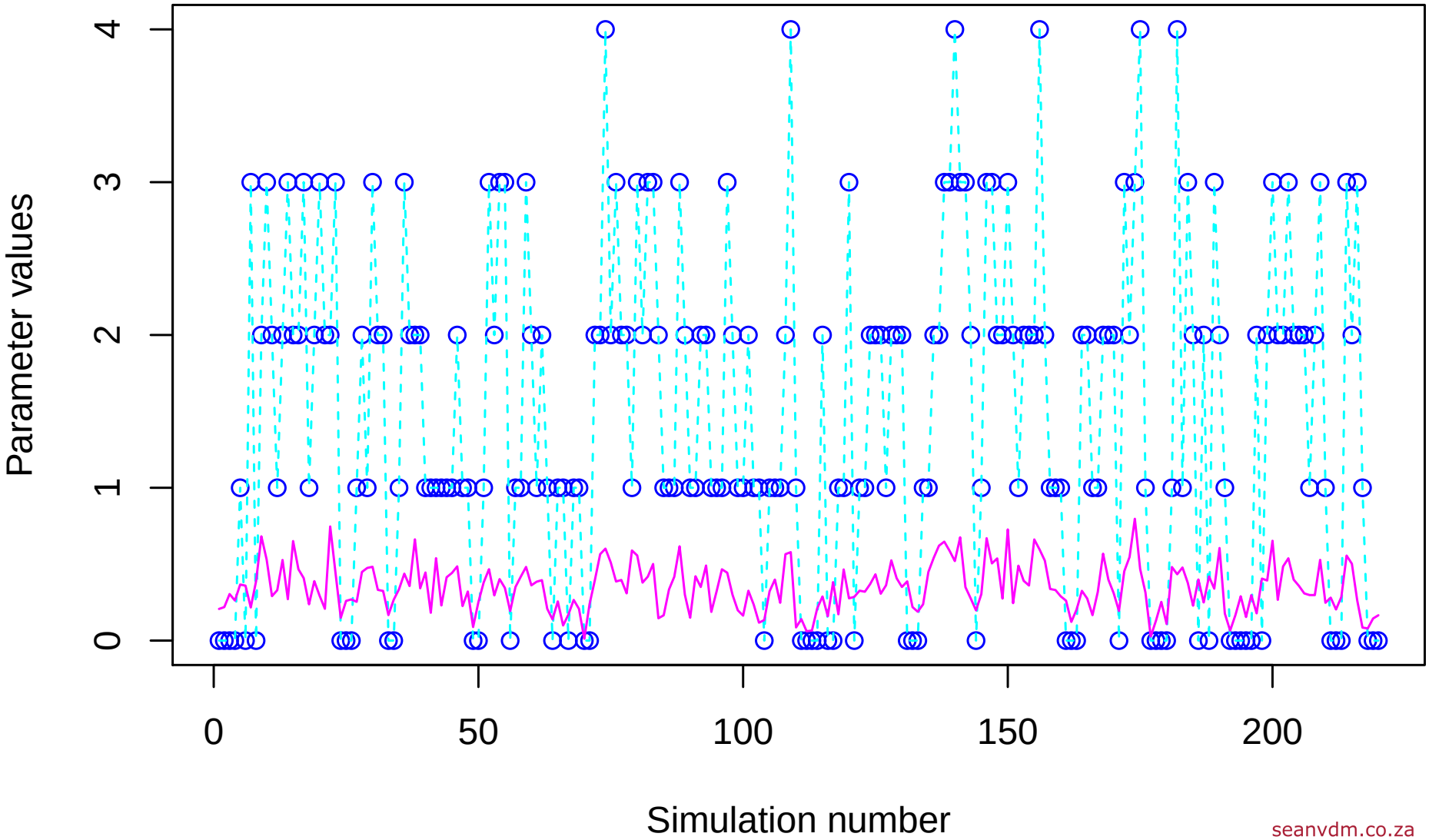
$$f(x_1|x_2) = Binomial(n, x_2)$$
$$f(x_2|x_1) = Beta(x_1+\alpha, n-x_1+\beta)$$

- Start with a random x_2, simulate $x_1|x_2$, simulate $x_2|x_1$, ...
- after some time ... simulate $x_1|x_2$, simulate $x_2|x_1$ and store latest (x_1, x_2) vector,
- simulate $x_1|x_2$, simulate $x_2|x_1$, simulate $x_1|x_2$, simulate $x_2|x_1$ and store latest (x_1, x_2) vector,
- simulate $x_1|x_2$, simulate $x_2|x_1$, simulate $x_1|x_2$, simulate $x_2|x_1$ and store latest (x_1, x_2) vector,
- repeat until the joint distribution converges *in distribution*.

Gibbs - Example 1 Code

```
x2 <- runif(1); n <- 4; a <- 3; b <- 6
nsim <- 60; burnin <- 100; keepevery <- 2; totsim <- nsim*keepevery
    +burnin
allsims <- matrix(0,totsim,2)
for (i in 1:totsim) {
   x1 <- rbinom(1,n,x2)
   x2 <- rbeta(1,(x1+a),(n-x1+b))
   allsims[i,] <- c(x1,x2)
}
sims <- allsims[-burnin:-1,]
sims <- sims[(1:nsim)*keepevery,]

# Draw graphs to investigate convergence
plot(rep(1:totsim,2),c(allsims),type='n',xlab='Simulation number',
    ylab='Parameter values')
points(1:totsim,allsims[,1],col='blue')
lines(1:totsim,allsims[,1],col='cyan',lty=2)
lines(1:totsim,allsims[,2],col='magenta')
```

Gibbs - Example 1 Graph



Gibbs - Example 2 p.1

Consider the regression model $Y_i \sim Gamma\left(\mu_i = \frac{\alpha_i}{\lambda_i} = \mathbf{x_i'}\boldsymbol{\beta},\ \lambda_i\right)$.

```
gammalpost <- function(X,y,B,l,prior='mdi') {
p <- length(B);   n <- length(y);   mu <- X%*%B;   al <- mu*l
if (any((c(mu,l)<=0))) {
   return(NaN)
} else {
   if (prior == 'mdi') {
      logPosterior <- sum((al + 3)*log(l) - (2*lgamma(al)) + ((al-1)*(log(y) + digamma(al
           ) - 1)) - (l*y))
   } else {
      lik <- sum((al+1)*log(l) - lgamma(al) + (al - 1)*log(y) - l*y)
      if (prior == 'unif') {
         logPosterior <- lik
      } else {
         A <- diag(c(1/(l^2) - mu/l - (mu^2)*trigamma(al)))
         E <- matrix(0,p,p)
         for (j in 1:p) {
            for (k in 1:p) {
               E[j,k] <- -sum(trigamma(al)*X[,j]*l*X[,k]*l) }   }
         D <- matrix(0,n,p)
         for (j in 1:p) {
            D[,j] <- X[,j]*(1 - log(l) - al*trigamma(al)) }
         Fischer.Matrix <- rbind(cbind(A,D),cbind(t(D),E))
         if (prior == 'Jeffreys') {
            Jeff <- sqrt(det(Fischer.Matrix))
            logPosterior <- log(Jeff) + lik
```

Gibbs - Example 2 p.2

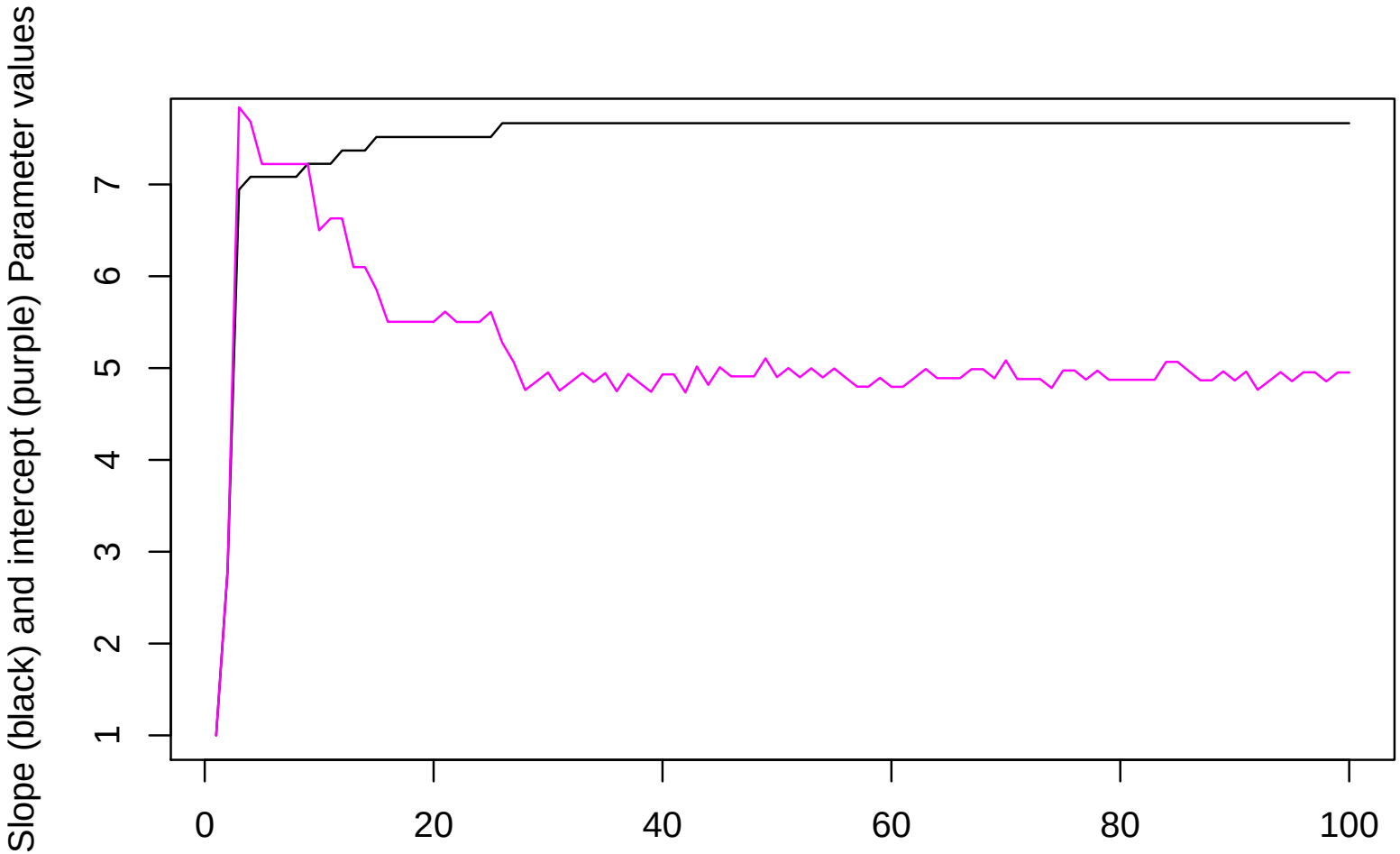
```
        } else {
           lijp <- 0.5*sum(log(diag(Fischer.Matrix)))
           logPosterior <- lijp + lik } }   }
   return(logPosterior) } }

lpostoptimwrapper <- function(P,X,y,prior='mdi') {
p <- ncol(X);   B <- P[1:p];   l <- P[-(1:p)]
f <- gammalpost(X,y,B,l,prior)
if (!is.finite(f)) { return(1000) } else { return(-f) } }

gammaregpostsim <- function(X,y,nsim=200,prior='mdi',onelambda=TRUE,burnin=100,dropfactor
     =1) {
p <- ncol(X);   n <- nrow(X)
if (onelambda) { ll <- 1 } else { ll <- n }
np <- p + ll
totsim <- burnin + (nsim*dropfactor)
sims <- matrix(0,totsim,np)
sims[1,] <- optim(rep(1,np),lpostoptimwrapper,X=X,y=y,prior=prior,method="L-BFGS-B",lower
     =c(rep(-Inf,p),rep(0,ll)),upper=rep(Inf,np))$par
bseq <- seq(-1.2,2.8,0.02);   lbseq <- length(bseq)
lseq <- 4^seq(-1,1,0.02);   llseq <- length(lseq)
```

Gibbs - Example 2 p.3

```
for (i in 2:totsim) {
   l <- sims[i-1,(p+1):np]
   for (j in 1:ll) {
      loptions <- l[j]*lseq
      f <- rep(0,llseq)
      for (k in 1:llseq) {
         l[j] <- loptions[k]
         f[k] <- gammalpost(X,y,B,l,prior) }
      f[!is.finite(f)] <- -Inf
      f <- exp(f-max(f))
      f[!is.finite(f)] <- 0
      f <- f/sum(f)
      l[j] <- loptions[sample.int(llseq,1,prob=f)] }
   boptions <- t(sims[i-1,1:p,drop=F])%*%t(bseq)
   B <- sims[i-1,1:p]
   for (j in 1:p) {
      f <- rep(0,lbseq)
      for (k in 1:lbseq) {
         B[j] <- boptions[j,k]
         f[k] <- gammalpost(X,y,B,l,prior) }
      f[!is.finite(f)] <- -Inf
      f <- exp(f-max(f))
      f[!is.finite(f)] <- 0
      f <- f/sum(f)
      B[j] <- boptions[j,sample.int(lbseq,1,prob=f)] }
   sims[i,] <- c(B,l) }
if (burnin > 0) {sims <- sims[-burnin:-1,] }
sims <- sims[(1:nsim)*dropfactor,]
return(sims) }
```

Gibbs - Example 2 Graph

What happens if you force the simulation to start at a bad place?
(black line should be around 8, purple line around 4)



Metropolis-Hastings (MH) Sampler

MH provides an easy method to simulate from a multivariate distribution (f) and works as follows:

- Start by picking a proposal/jump distribution q and then repeat the following steps:
  - Simulate a jump candidate x_c from $q(x|x_j)$
  - Set $x_{j+1} = x_c$ if $\frac{q(x_j|x_c)f(x_c)}{q(x_c|x_j)f(x_j)} > u$, where $u \sim U(0,1)$
- If $q(x|x_j)$ is symmetric then this simplifies to $\frac{f(x_c)}{f(x_j)} > u$
  - Remember that you can pick q to suit your needs.
  - A q that is similar to f makes for faster convergence, but
  - a symmetric q makes calculations easier.
  - A good compromise is usually $q(x|x_j) = N_p(x_j, \Sigma)$, where Σ is a diagonal matrix with entries chosen to be close to the variances of f.
- Another option is the independence sampler $q(x|x_j) = q(x)$, in which case the rule is $\frac{q(x_j)f(x_c)}{q(x_c)f(x_j)} > u$

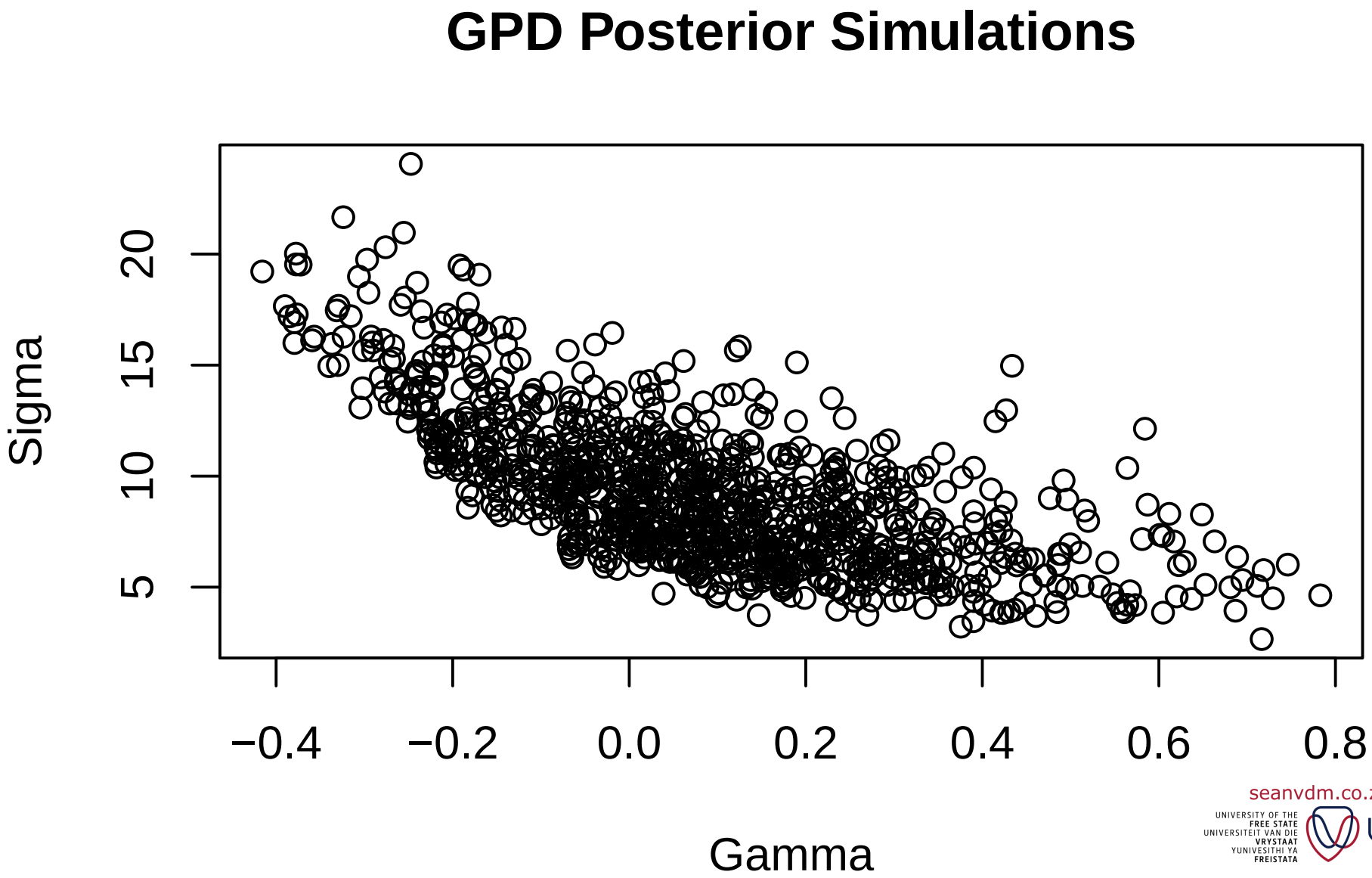
Metropolis-Hastings Example 1

- Consider simulating the GPD posterior (with MDI prior).
  $\log \pi(\gamma, \sigma) = -(n+1)\log\sigma - \gamma - \left(\frac{1}{\gamma}+1\right)\sum_{i=1}^{n}\log\left(1+\frac{\gamma x_i}{\sigma}\right)$
- The restriction that $\sigma > 0$ is a problem that needs addressing.
- Let $\tau = \log\sigma$ then the Jacobian is $e^{\tau} = \sigma$ and the log posterior is
  $\log \pi(\gamma, \tau) = -\gamma - \left(\frac{1}{\gamma}+1\right)\sum_{i=1}^{n}\log\left(1+\gamma x_i e^{-\tau}\right).$
- Consider proposal $\gamma_c \sim N(\gamma_j, 0.05^2)$ and $\tau_c \sim N(\tau_j, 0.5^2)$.
- Since this proposal is symmetric we can use the basic Metropolis sampler:
  Accept $\boldsymbol{\theta}_c = (\gamma_c, \tau_c)$ if $\log\pi(\boldsymbol{\theta}_c) - \log\pi(\boldsymbol{\theta}_j) > \log u_j$.
- Reasonable starting values: $\boldsymbol{\theta}_1 = (0.2, 0)$
- One remaining restriction: $\gamma > \frac{-e^{\tau}}{\max_i x_i}$

Metropolis-Hastings Example 1 Code

```
logposterior <- function(x,n,evi,taw) {
lpost <- -(sum(log(x*evi*exp(-taw)+1))*(evi+1)/evi + evi + (n+2)*taw)
return(lpost) }

simpost <- function(sampl,nsim=1000,burn=100,threshold=0,eviguess=0.25,sigguess=1) {
sampl <- sampl[sampl>=threshold]-threshold
n <- length(sampl)
evi0 <- eviguess; lsig0 <- log(sigguess)
lpost0 <- logposterior(sampl,n,evi0,lsig0)
i <- 0; stp <- nsim*2+burn
sims <- matrix(0,stp,2)
mx <- max(sampl)
while (i < stp) {
   evi1 <- evi0 + rnorm(1,0,0.05)
   while (evi1 <= (-exp(lsig0)/mx)) { evi1 <- evi0 + rnorm(1,0,0.05) }
   lsig1 <- lsig0 + rnorm(1,0,0.5)
   while ((evi1 <= 0) && (exp(lsig1) <= (-evi1*mx))) { lsig1 <- lsig0 + rnorm(1,0,0.5) }
   lpost1 <- logposterior(sampl,n,evi1,lsig1)
   if (lpost1 - lpost0 - log(runif(1)) > 0) {
      evi0 <- evi1; lsig0 <- lsig1; lpost0 <- lpost1
      i <- i + 1
      sims[i,1] <- evi1
      sims[i,2] <- exp(lsig1) } }
sims <- sims[-burn:-1,] # Throw away burn in simulations
sims <- sims[(1:nsim)*2,] # Discard every second simulation to reduce dependence
return(sims) }
```

Metropolis-Hastings Example 1 Results



Overview - click to jump

About Stan

- https://mc-Stan.org/
- "Stan is a state-of-the-art platform for statistical modeling and high-performance statistical computation."
- You type out your model in Stan language
- Then Stan does all the work, including:
  - Simulating the whole posterior distribution,
  - Simulating the approximate posterior, and/or
  - Maximum likelihood

Advantages of Stan

- You specify the model once and then fit it as much as you want, however you want
- It compiles your model into C++ code, so it's *very* fast
- Latest simulation methods: NUTS and HMC, better than old Gibbs and MH
- If your model is super complicated or simulation is too slow, then you can do variational Bayes or optimisation (including Maximum Likelihood)
  - You don't change your model, just one line of code
- Most importantly, if you want to change your model then you change only the part of the model that needs changing
  - No changing software, no installing new packages, no changing notation for a different 'PROC', *etc.*

Main Advantage of Stan: Bayes

- Stan automatically picks practical priors **and initial values**, so you don't have to
- But you can easily add your own priors if you like
- Gives all the advantages of Bayes without effort:
  - Prediction is easy on any scale
  - Intervals are better because they incorporate more sources of variation and don't have to be symmetric
  - Probabilities can be estimated directly

Stan also has the current advantage of popularity. It is well supported via both good documentation and a supportive online user base.

Stan documentation

Stan has excellent documentation:
https://mc-stan.org/users/documentation/

- There are basically three parts:
  1. The user guide that explains key concepts and approaches in detail. This is about statistics and how to translate ideas into Stan models.
  2. The reference guides that cover the finer details of how to program Stan models. This includes distributions and parameterisations, as well as notation details.
  3. Other sources of information. This part merely mentions that the internet has other sources of help that are worth investigating, such as forums, blogs, and statistics social media.
- The user guide is the place to start, while the reference guides are useful when you have your model written down mathematically (in extreme detail) but need help with the programming.

Stan user guide

https://mc-stan.org/docs/2_28/stan-users-guide/index.html

The following parts of the user guide should be considered compulsory reading for this course:

- Regression models, particularly linear, robust, and logistic models.
- Truncated and censored data (and missing data in general).
- Matrices, vectors, arrays, and indexing of them. Logical indexing, multiple indexing, *etc.* are also important.
- Custom probability functions (custom likelihoods, custom priors, and so on).
- Sections discussing the multivariate normal.
- Sections discussing mixture models (leave this for later - towards the end of the course).

Why not use Stan for everything?

- Basic tools are better for basic problems
- If you really only want to do a standard regression, or fit a standard distribution, then Stan is overkill
  - You don't need to build a model if all you need is descriptive statistics
  - Stan is fast, but a standard regression is 'instant'
  - Standard models have standard interpretations
- Stan is new so it needs explanation
- Stan is for modelling, so don't use it for exploratory data analysis
- Don't use Stan for data mining tasks
- However, do look into packages like `rstanarm` which has easy Stan equivalents for most common model types.

Overview - click to jump

Meta analysis

- What if you know the variance of each observation, but not the true mean?
- What if that mean depends on explanatory factors?
- What if different levels of an explanatory factor have different variances in their means?
- Standard software does not allow us to answer these questions properly
- With Stan it doesn't matter how weird the model is, as long as you can write it down in standard notation
- This model is rather complicated so we won't go into it
  - I recommend you learn Stan with easy models

Fitting a distribution

- What if you want to fit a distribution to data, but it's not in any statistics package?
- Or maybe you want to play with the priors?
- Let's look at two examples:
  1. https://seanvdm.co.za/post/tfit1/
  2. https://seanvdm.co.za/post/simulationsversterv1/
- More complex example where priors are compared via simulation study done in Stan: https://seanvdm.co.za/files/workshops/skewt2021/SvdMSkewt2021.html#1

Regression

- What if you want to fit a logistic regression model and get prediction intervals?
- Let's look at the famous Challenger O-ring failure question:
- https://bookdown.org/egarpor/PM-UC3M/glm-challenger.html
- https://seanvdm.co.za/post/challenger1/
- Or how about a bi-phasic regression?
- https://seanvdm.co.za/post/potatoes/

Overview - click to jump

Specifying the core model

- Always start with the likelihood — the part of the model closest to the data.
- Second, specify priors for all the parameters in the likelihood.
  - Remember that constants and hyperparameters are specified separately later.
- Third, specify priors for any parameters that do not have them yet.
- Fourth, specify functions of parameters you wish to monitor
- Fifth, specify variables corresponding to the posterior predictive distribution for any new data.
  - New data must be loaded with the old data.
- Then add all the definitions and constraints needed for your model to make sense.
  - Only the likelihood, definitions, and constraints are compulsory. Priors are recommended. Other items can be done afterwards.

rstan

- Let's you work with the data in R, send everything to Stan and get the results back in R where you can perform further calculations on the posterior and posterior predictive distributions.
- First, store your model
  - Can be done in RStudio or in a text file, *e.g.* 'mymodel.stan'
  - Easiest is to use a Stan code block. Optionally store the resulting model in a Rds file.
- Second, create your data list by pointing to existing R variables.

```
STANdata <- list(STANvariable1=Rvariable1, STANvariable2=
    Rvariable2, STANvariable3=Rvariable3)
```

- Third, create a function called 'inits' that takes no inputs but gives as output an R list consisting of named components with random (or deterministic) starting values. *You may specify none, some, or all initial values.*

Loading rstan

- Note that you must load *rstan* every time you open R with, 'library(rstan)'.
- If it is not installed:
  - Install Rtools first
  - Then in R:

```
install.packages('rstan')
```

- You should set the *auto_write* option to TRUE for a bit of extra speed in development
- More importantly, **set the number of cores you want to use**
  - Align the number of chains you specify during sampling to this core number for optimal efficiency
  - Generally it is a good idea to use most, but not all of your CPU's power

```
library(rstan)
library(parallel)
mycores <- max(1,floor(detectCores(logical = FALSE)*0.75))
options(mc.cores = mycores)
rstan_options(auto_write = TRUE)
```

Working with rstan

- Use the 'sampling' function and specify, in any order (by name):
  - model
  - data list
  - parameters to monitor, as text vector
  - number of iterations desired after thinning but before removing burn-in
  - number of chains (default of 3 is good)
  - burn-in period, say 20% to 50% (default) of iterations
  - thinning value, set to 2 or 3 if autocorrelation is bad, not higher
  - inits function
  - algorithm

```
simoutput <- sampling(mymodel, stan_data, pars=c('alpha','beta
    '), iter = 12000, chains = mycores)
```

- Only the model and data is compulsory, the rest have defaults.

Working with results from STAN

Useful commands for working with the STAN object include:

- traceplot — to see all the simulation patterns
- extract — to get the simulations themselves
- plot — to view plots
- summary — to get a summary of the model

```
summary(simoutput)
traceplot(simoutput)
list_of_draws <- extract(simoutput)
hist(list_of_draws$beta)
```

Assessing convergence

- Autocorrelation graphs (correllograms) show how much each round of the simulation depends on the previous round.
- Use this to figure out which simulations to keep and which to throw out *IF* you need an independent sample.
- The idea is to start multiple chains at very different initial values and see when they 'join up' in distribution.
- Useful in complicated models (simple models converge almost immediately).
- The Monte Carlo error (MC error) is approximately the extra uncertainty introduced by simulating instead of integrating analytically.
- A rule of thumb is to simulate until this is 5% or less of the parameter's standard deviation.
- The Brooks, Gelman and Rubin Diagnostic (called R or GRdiag) can be used to determine the required burn-in.

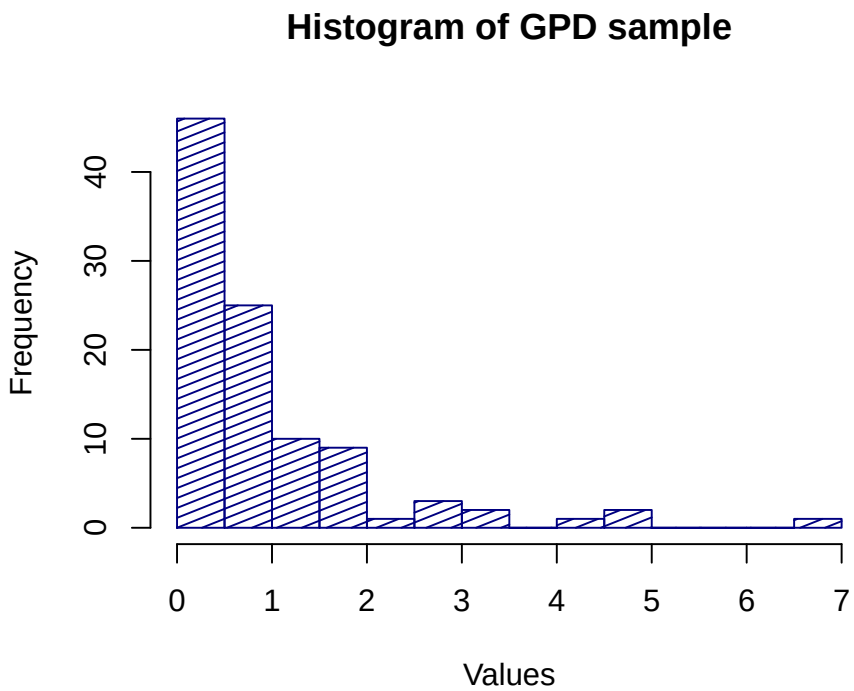
Minimal example code (can be used to test installation)

- In this example we simulate a sample from a gamma density and then fit a gamma distribution to that density using Stan, via RMarkdown:

````
```{r}
n <- 50
simdata <- rgamma(n, 3.14, 0.789)
library(rstan)
options(mc.cores = 4)
```
```{stan output.var='gammamodel1'}
data {
  int n;
  real x[n];
}
parameters {
  real<lower=0> alpha;
  real<lower=0> lambda;
}
model {
  x ~ gamma(alpha, lambda);
}
```
```{r}
results <- sampling(gammamodel1, data = list(n=n, x=simdata))
```
````

Example: Fitting a GPD

Consider the following GPD sample with threshold 0, scale (σ) 0.5 and EVI (η) 0.3:



- Let's try to simulate the posterior of σ and η.

Fitting a GPD: Details

- In general the GPD has complex parameter restrictions which make simulation difficult.
- For this example we will consider only the case where the EVI is positive.
- When the EVI is positive we can rewrite the GPD as a mixture. See https://en.wikipedia.org/wiki/Generalized_Pareto_distribution
  - $Y \sim Exp(\lambda)$
  - $\lambda \sim Gamma(\alpha, \beta)$
  - where α is the tail index (1/EVI) and $\sigma = \beta/\alpha$
- We will use the MDI prior here, which is written on the log scale as $c_2 + \log \alpha - \log \beta - \alpha$

GPD model code

The first step is to define the model:

```
data {
  int<lower=0> n;                    // number of observations
  real<lower=0> y[n];                      // observations
}
// The parameters of the model
parameters {
  real<lower=0> a;               // tail index
  real<lower=0> b;                  // scale parameter
  real<lower=0> lambda[n];     // mixing variable
}
// The model to be estimated.
model {
  y ~ exponential(lambda);    // GPD as mixture
  lambda ~ gamma(a,b);
  target += log(a)-log(b)-a;  // MDI prior
}
```

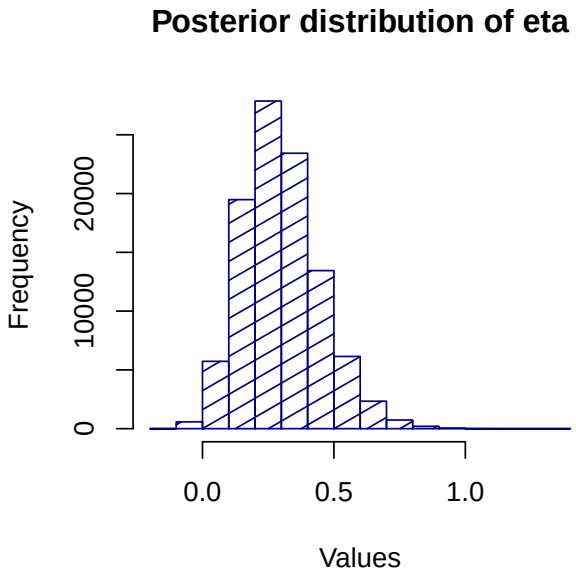
GPD model run

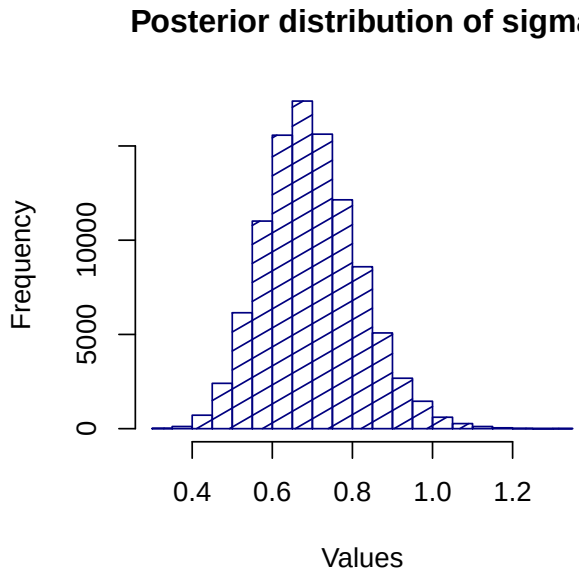
- We then ask Stan to compile the model into C++ code. The easiest approach is by pressing the play button in RStudio that corresponds with the code chunk. This model was saved in the R variable 'GPD1'.
- And we're ready to run the model:

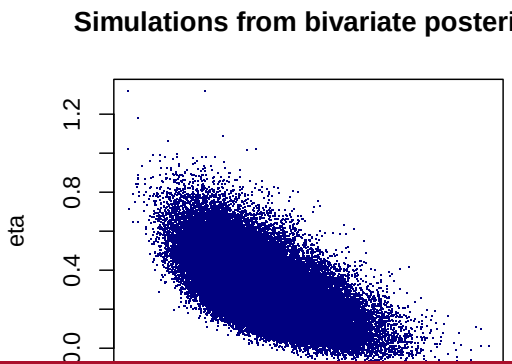
```
ModelFit1 <- sampling(GPD1, list(n = samplesize, y =
    datavector))
traceplot(ModelFit1)
summary(ModelFit1)
```

GPD results in R

Instead of using the raw results we can manipulate the results in R to get custom histograms, scatter plots or intervals to our liking:







seanvdm.co.za

BUGS/JAGS Example: Skew-t Regression

- This is a regression model where we begin by assuming the random errors have a t distribution with unknown degrees of freedom.
- Subtract a random value (z_i) from each observation to indicate to what extent that observation is affected by the skewness.
- These random values are distributed truncated $N(0, 1)$, such that they are all strictly non-negative.
- We also multiply these values by a general skewness parameter (δ), which can take on any value in order to accommodate skewness both ways (but not at the same time).
- This gives us the following log likelihood (without unknown constant):

$$n \log \Gamma\left(\tfrac{\nu+1}{2}\right) + \tfrac{\nu}{2}\log(\nu) - \log\Gamma\left(\tfrac{\nu}{2}\right) - \log(\sigma) - \left(\tfrac{\nu+1}{2}\right)\sum_{i=1}^{n} \log\left[1 + \left(\frac{y_i - \mathbf{x}_i\boldsymbol{\beta} - \delta z_i}{\sigma}\right)^2\right]$$

- We implement this model in BUGS as follows:

```
model {
 for (i in 1:samplesize) {
 mu[i] <- beta0 + beta1*x[i] + delta*z[i]
 y[i] ~ dt(mu[i], taou, v)
 z[i] ~ dnorm(0,1)T(0,)}
 v <- k + 2.2
 beta0 ~ dflat()
 beta1 ~ dflat()
 taou ~ dexp(0.0004)
 k ~ dexp(0.02)
 delta ~ dnorm(0,0.001)
 sigma <- 1/sqrt(taou) }
```

- JAGS (Just Another Gibbs Sampler) and BUGS (Bayes Using Gibbs Sampling) are alternatives to Stan.
- They pre-date Stan but are still popular because they are quite different:
  - They use Gibbs sampling, which enables some models that Stan has trouble with: certain missing value models and certain mixture models.
  - Gibbs sampling is also faster in a few cases, such as Dirichlet distributions.
  - However, they require the specification of proper priors for all parameters, as well as initial values for most.

Skew-t Regression p.3

- The data is very easy:

```
BUGSdata <- list(x = x, y = y, samplesize = samplesize)
```

- The initial values are a little more involved, although you can make them easy if you want to.

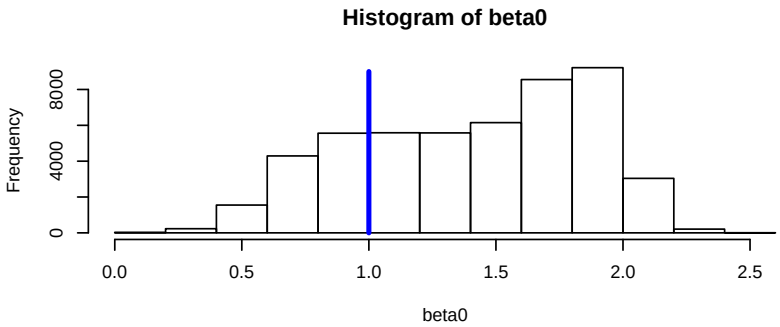
```
inits <- function() {return(list(beta0 = rnorm(1,0,0.001),
    beta1 = rnorm(1,0,0.001), taou = 1, k = rchisq(1,80), z =
    abs(rnorm(samplesize)), delta = rnorm(1,0,0.001)))}
```

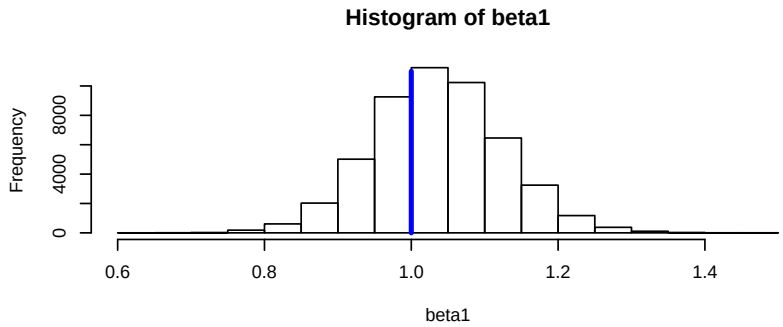
- Finally, we run the model:

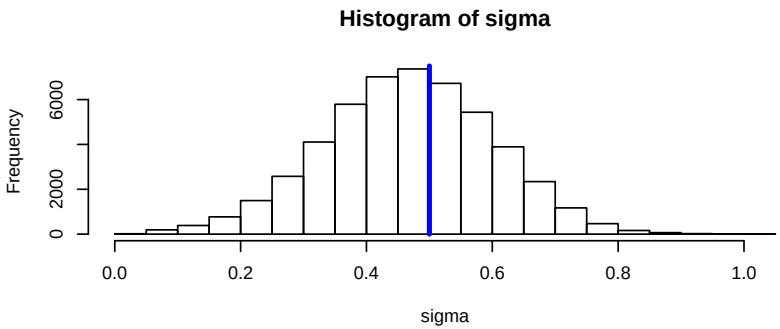
```
mymodel <- bugs(BUGSdata, inits, 'skewtreg.txt', parameters =
    c('beta0','beta1','sigma','v','delta','z'), n.chains=5,n.
    iter=20000, debug=TRUE)
```

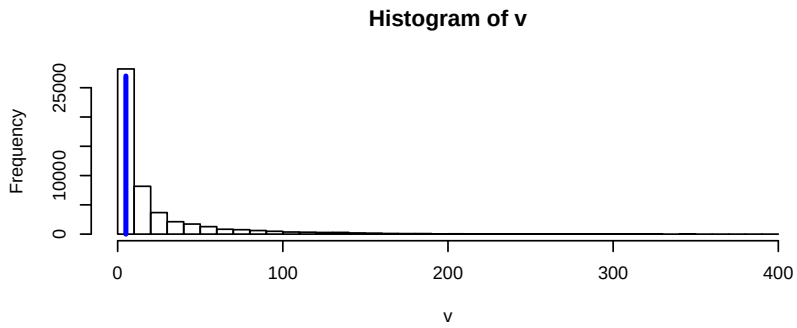
Skew-t Regression results

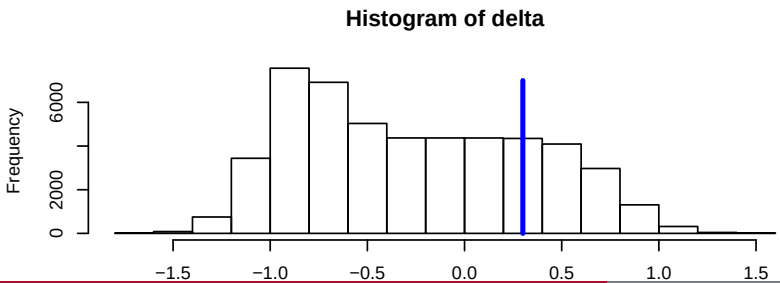
This particular fit seems like a success, other than the inherent correlation between β_0 and δ, as the blue lines are the true parameter values.











seanvdm.co.za

Skew-t in Stan

- The above model can also be implemented in Stan, but that is left as an exercise.
- Instead we illustrate the fitting of a variant of the Skew-t distribution in Stan.
- Presentation on why and how to use Skew-t distributions

Binomial models with subject effect

Approximately, the following counts of surviving critters were observed out of 50:

Control	Treatment 1	Treatment 2
33	0	0
31	0	1
28	0	2
28	0	3
31	0	0
23	0	0
11	0	0
38	0	0

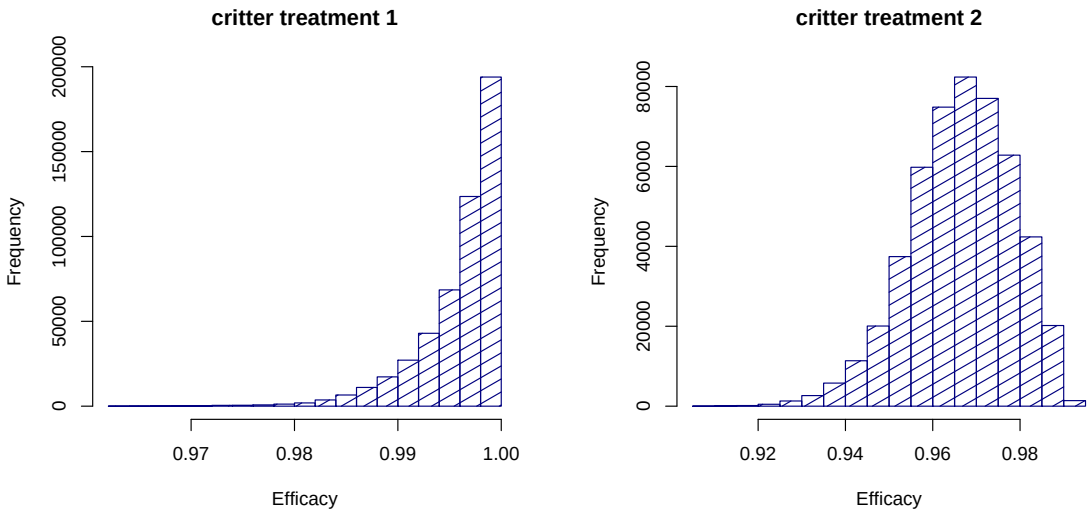
We assume each of the 24 subjects had their own Binomial probability parameter that is Beta distributed but that each treatment group has the same Beta parameters.

Binomial model options

- We are interested in the treatment efficacy $1 - \frac{\pi_T}{\pi_C}$.
- Since the efficacy of the first treatment is so high, maximum likelihood approaches do not yield sensible and useful results. Maximum likelihood would suggest that Treatment 1 has perfect efficacy, which is impossible.
- For a Bayesian approach, we could assume that each observation is binomial with 50 trials, and that the probability of success on each trial varies from subject to subject.
- One popular approach is to assume beta distributions for the subject variation, with parameters dependent on treatment.
- Another popular approach is to assume normal distributions for the subject variation, via a logit link to the probability of success.

Beta-Binomial model results

Normally Bayes models converge quickly, but I found that this model required at least 100,000 Gibbs loops to converge.



We can see that the posterior distribution of efficacy is very skew due to being up against the limit of 1. This means we can construct shorter intervals by using the Highest Posterior Density (HPD) method than by using symmetric intervals.
The 95% HPD intervals are (0.9878; 1) and (0.9451; 0.9891).

Dependent Binomial regression

- Students were asked to watch a subtitled film and then to identify which words from a list of 79 they had read in the film.
- Of these 79, 15 were actually in the film and the other 64 were there as a form of control.
- Prior to the experiment, some students (at random) had received an 'intervention'.
- The question is whether the 'intervention' made a difference.
- Turns out the answer is yes, but only if the statistician isn't lazy:
  - If you only consider the accuracy of the students with respect to the 15 words in the film then the difference is far from significant (p-value is 89%),
  - but if you take all the data into account (using a more complex model) then the difference is highly significant (p-value is 0.86%)!

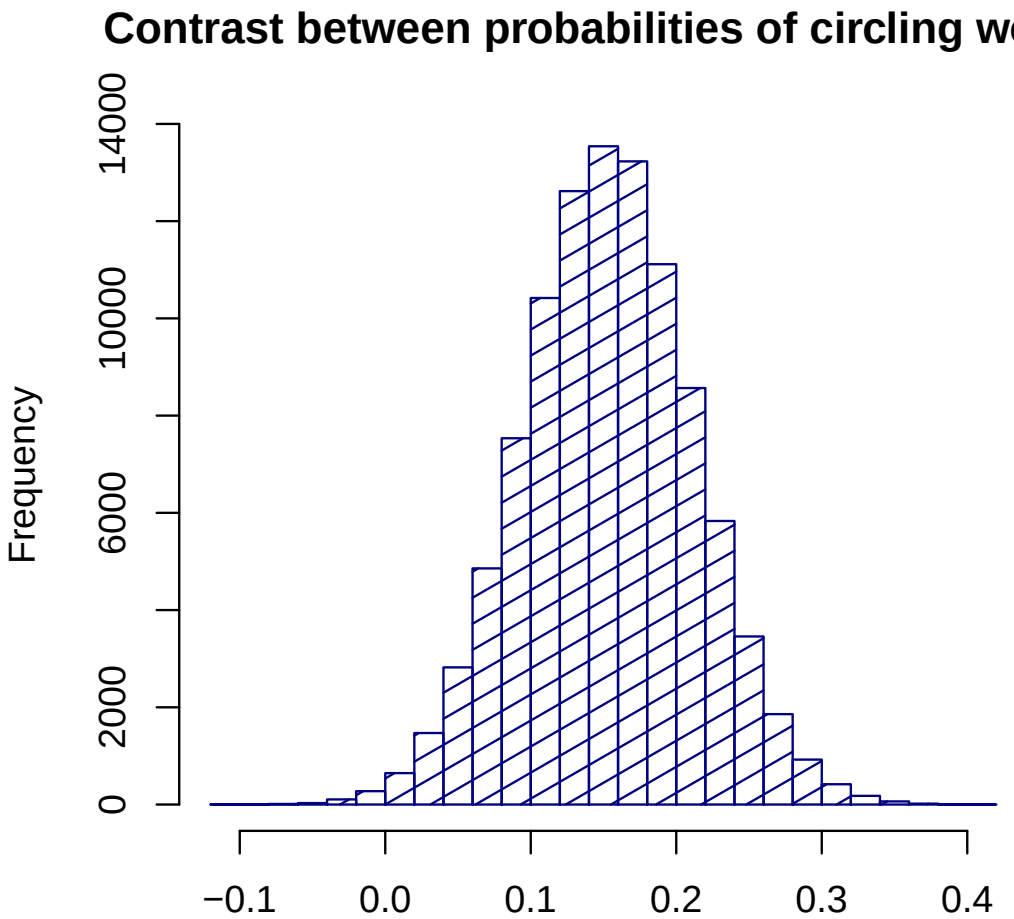
Dependent Binomial regression model

- We attempt to express the model in hierarchical form below.
- We use bold to denote vectors and square brackets to denote group membership, *i.e.* $treat[i]$ is the treatment received by person i (intervention or control)
- j is the section of the test (words present or absent).

$$
\begin{aligned}
y_{ij} &\sim binomial(\pi_{ij},\ n_j) \\
\mathbf{n} &= [15,\ 64] \\
\alpha_{ij} &= logit(\pi_{ij}) \\
\boldsymbol{\alpha}_i &\sim normal_2(\boldsymbol{\mu}_i,\ \Sigma) \\
\boldsymbol{\mu}_i &\sim normal_2(\boldsymbol{\mu}_{treat[i]},\ \Sigma_{treat[i]})
\end{aligned}
$$

Dependent Binomial regression model results

After some additional calculations we get a posterior distribution for the difference in the probability of correctly circling and not circling words as a result of the 'intervention'.



Censored data example

- The number of live ticks (out of 50) on a set of animals were counted at times 3, 6, 12 and 24 hours.
- We are interested in the mean time of 'death' of the ticks, given the specific treatment.
- The easiest model is to work on the log time scale and consider each observation as random Normal value in the interval between when the tick was observed on the animal and then not on the animal.
- The mean of these random Normal variables depends on a fixed treatment effect and a random animal effect.
- We are interested in the time of death ratio — the exponent of the difference between mean times of death on the log scale.
- Interval censoring is implemented by integration of the density between the limits where the observations occurred.

Student Plan Comparison — Introduction

This example is from the 'Random Effects Models' notes of Prof. Robert Schall, UFS.

- Students at a specific law faculty normally undergo a 4-year degree program.
- However, applicants who do not have the required minimum Admission Point are enrolled in a 5-year program; that is, they join the regular 4-year program after an initial year of preparatory classes.
- In order to evaluate the effectiveness of the preparatory year, academic results (module marks) of students in the two programs were compared.
- Marks for 10 first year modules were available for the 2006 cohort of students.
- The objective of the analysis was to compare the mean module marks of the two programs statistically.

Student Plan Comparison — Missing data issue

A naive analysis approach would be as follows: For a given module, calculate the average mark for students from the 5-year and 4-year programs, respectively, and compare those marks using a two-sample t-test (say). However, the data set is characterised by the fact that for many students some of the module marks are missing, mostly because the student in question discontinued his or her studies, often for lack of academic success. Simply ignoring the missing data and performing an analysis only of the available data for a given module would cause the estimates of average module marks to be biased upwards (because the missing data are likely to come from the weaker students). While this upward bias applies to both the 4-year and the 5-year program, the drop-out rate is higher among students in the 5-year program than among students in the 4-year program. Therefore, the upward bias is stronger for the 5-year program than for the 4-year program, and this differential bias causes a bias in the estimate of the "5-year — 4-year" difference in average module marks.
It is plausible, and in fact clearly observed in practise, that the module marks for a given student are highly correlated. Thus available marks for a given student can provide information on module marks that are missing for that student. In order to exploit the correlation of module marks and thus "recover" information on missing marks, we must fit a model to all the module marks jointly. Missing data problems of this type can often be handled successfully by fitting a mixed model.

Student Plan Comparison — Fitted Effects

We are going to fit the following explanatory effects for the marks:

- Module as fixed effect (different modules have different average marks)
- Program as fixed effect (the average marks in the 4-year and 5-year programs differ)
- Module.Program as fixed effect (the differences between 4-year and 5-year programs differ for different modules)
- Student as random effect (students differ in their average module mark)
- Students may also differ greatly in the variance of their marks and thus we could incorporate a student effect in the variance model.
- It may also make sense to compare different versions of the model to be more confident about our conclusions.

Random effects versus fixed effects

The differences between random and fixed effects are subtle and thus generally poorly understood, even by experienced scientists. *Even the Wikipedia page does a terrible job of trying to explain it.*

- Random effects are usually created when observations are grouped according to the levels of a factor, but the specific factor levels are not of particular interest.
- With random effects we are interested in a random future factor level from the population of possibilities; whereas with fixed effects we are interested in the closed set of observed factor levels.

Random effects examples

- In clinical trials we sometimes have multiple measurements taken from patients over time. In this case we have what is called a **subject effect** because the observations from a subject (in this case a person) are related to each other.
  - These subjects/people can be systematically different from each other.
  - Not all subjects have the same number of observations, and this imbalance often creates bias in the model estimates if we not do adjust for it.
  - We are interested in the effect of the medicine on a random future subject, not the ones we already treated.
- If we are studying children then we tend to have a school effect because the teachers, facilities, and teaching approaches can differ a lot between schools to the point of dominating the individual expression of the children.
  - Here the subject is a school or classroom.
  - We are usually interested in a general school situation, not just the specific schools that were sampled in the study.

Fixed effects examples

Random effects may be easier to understand in contrast to fixed effects.

- In clinical trials the medicines or treatments are fixed effects because we are trying to compare the existing limited selection of specific treatments to each other.
  - We are not interested in some random future treatment, we want to know what is the best treatment that is currently available.
- In farming trials we will often compare fixed levels and types of fertiliser.
  - These are fixed effects because there are a fixed selection of fertilisers available and of interest.
  - **In future we will know the contents of the fertiliser being used exactly, they will not be random.**

However, not all cases are clear as to whether fixed or random effects are more appropriate.

Implementing random effects

In Bayesian hierarchical models we can implement random effects by:

1. Associating a parameter with each factor level (just like with fixed effects)
2. and then considering these parameters as a sample from a general population
   - Usually a normal population, for convenience
   - Suppose you have an explanatory factor X_3 with K levels, then we might say $\beta_{3k} \sim N\left(\beta_3, \sigma^2_{\beta_3}\right) \quad , \quad k = 1, \ldots, K$

Student Plan Comparison — Implementation and results

- For a detailed explanation of how the models are implemented, and the results evaluated, see https://seanvdm.co.za/post/academicplanexample/.
- The data is available at https://seanvdm.co.za/post/academicplanexample.csv.
- In summary, the students who enrolled for the longer programme did systematically better in all modules, in spite of having generally weaker school performance prior to entering the programme.

Exercises 12

Previous Exercises | Next Exercises

1. Simulate a sample from a Skew-t density with sensible parameters of your choosing. Estimate the moments and check that they make sense, given your chosen parameters.
2. Draw plots of the log likelihood and log posterior over a sensible range for each parameter individually, fixing the other parameters at rough initial estimates (say from the method of moments or some approximation). What insights might you gain from this process if you were applying it to real data?
3. Obtain posterior estimates of the parameters. Are they reasonable?
4. Generate a posterior predictive sample and compare it visually to your original sample.
5. Look online for at least two advantages of using the Bayes approach that apply to your desired career path and discuss.

Exercises 13

Previous Exercises | Next Exercises

1. Adapt and implement any item response theory model (using Bayes) on the marks obtained by the class for Test 1 of this course. Which student's total mark changes the most when you properly take question difficulty into account using this model?
2. Design a survey to pose to the rest of the class. It should have at least 1 Likert scale question and then some demographic variables. Do a Bayesian regression of the ordinal response on the demographic variables.
3. Record, along with the rest of the class, the number of meals and snacks you have each day for 1 specific week. Model the results as Poisson given the subject and day effects.

Overview - click to jump

Coverage

- We've already covered accuracy statistics.
- Main principal: if model A gives shorter 95% intervals than model B then model A is more accurate than model B.
- However, it is very important to first check that the intervals cover the target values as much as they say they do, otherwise such comparisons are meaningless.
- Basic check: do approximately 95% of the target values lie within their respective 95% intervals?
- OR: does the 95% interval cover the true value at least 95% of the time?
- Advanced check: do approximately w% of the target values lie within their respective w% intervals, for say $w = 1\%, 3\%, 5\%, 7\%, \ldots, 49\%$?

Coverage in practice

- Consider an observation column vector y of dimension $n \times 1$,
- and an $n \times k$ matrix of simulated predictions (let's call it K).
- First use the *quantile* function to get all desired quantiles simultaneously (much more efficient),
- then (vector) compare y to the quantiles and add up the results.

```
mu <- seq(0,1,0.01)
n <- length(mu)
y <- rnorm(n,mu,1)
k <- 999
K <- matrix(rnorm(n*k,rep(mu,k),1),n,k)
qs <- t(apply(K,1,quantile,c(0.025,0.975)))
(coverage <- mean(((qs[,1]<y) & (y<qs[,2]))))
```

Distribution matching

- In general, if model A produces a posterior predictive distribution that more closely matches the data than that of model B then model A is more accurate than model B.
- How do we measure 'more closely matches'?
  - Visually, using an empirical CDF (ECDF) step plot or an empirical QQ plot, or
  - Analytically, using a two sample ECDF test like the Kolmogorov-Smirnov (KS) test.
- Useful R commands include *ks.test*, *ecdf*, *plot*, *qqplot*.
- For more information see the Goodness-of-fit section of these slides.

Information Criteria

- None of the previous statistics take model complexity into account.
- More complex models tend to be more accurate on existing data because they are more complex, not because they are better models.
- However, this accuracy does not always carry over to new data, even if the new data is from the same source and nothing has changed in the Data Generating Process (DGP).
- A model that works just as well on unseen data as it did on the data used to build it is said to generalise well.
- Models that do not generalise well are said to be guilty of overfitting.
- More complex models are often more expensive to build, implement and run and often more difficult to understand or interpret.
- When comparing models we should take the complexity of the model into account.

Information Criteria Basics

- Models are simplifications of reality.
- Information criteria are based on information entropy — the relative information lost when a given model is used to approximate the DGP.
- All information criteria are relative! They are only useful if you have more than one of the same type to compare to each other.
  - The number you calculate has no meaning.
  - It has no units and no scale.
  - There is no such thing as a big or small IC.
- Basic terminology: Let
  - $\hat{L}$ be the maximum value of the likelihood.
    - Not the point at which it is a maximum, but the value of the likelihood at that point.
  - k be the number of parameters in the model.
  - n be the number of observation vectors.

Standard Information Criteria

- Akaike information criterion (AIC)
  - $AIC = 2k - 2\log(\hat{L})$
  - Small penalty for extra variables.
  - Penalty is not related to sample size and becomes meaningless in large samples.
  - In the case of linear regression $-2\log(\hat{L}) = n\log\left(\frac{RSS}{n}\right)$.
- Bayesian information criterion (BIC)
  - $BIC = k\log(n) - 2\log(\hat{L})$
  - Strong penalty for extra variables.
- Deviance information criterion (DIC)
  - Define the Deviance $D(\theta) = -2\log(p(\theta|\mathbf{y})) + c$ where c is a meaningless constant.
  - Define $\bar{D} = E_\theta[D(\theta)]$ and $p_D = \bar{D} - D(E(\theta))$, then $DIC = p_D + \bar{D}$ (Spiegelhalter et al., 2002).
  - p_D denotes the effective number of parameters.
  - For hierarchical models and complex models with lots of correlated parameters (if 2 parameters do the same thing then counting them both doesn't make sense).

Calculating Information Criteria

- All information criteria can be calculated manually if you can maximise the likelihood (or simulate from the posterior in the case of the *DIC*).
- For built-in models in R you can use the *AIC* command.
- In BUGS you can get *DIC* as part of the output (from a second run after burn-in).
- In all cases, **the lower value of the IC indicates the more parsimonious (better) model**.
- Alternatively, you can get the *Bayes Factor* of two models and see if it is much bigger than 1 (bigger than 10 is considered strong).

$$BF_{1,0} = \frac{p(\mathbf{y}|M_1)}{p(\mathbf{y}|M_0)} = \frac{\int p(\boldsymbol{\theta}_1|M_1)p(\mathbf{y}|\boldsymbol{\theta}_1, M_1)d\boldsymbol{\theta}_1}{\int p(\boldsymbol{\theta}_0|M_0)p(\mathbf{y}|\boldsymbol{\theta}_0, M_0)d\boldsymbol{\theta}_0}$$

Bayes factors

- Can also be interpreted (and sometimes calculated) as the posterior odds divided by the prior odds.

$$BF_{1,0} = \frac{\frac{P(M_1|\mathbf{y})}{P(M_0|\mathbf{y})}}{\frac{P(M_1)}{P(M_0)}}$$

- It tries to put all the focus on the data and which model the data supports (taking away the prior information).
- Can be very difficult to calculate outside of simple problems.
- In difficult problems we focus on posterior probabilities for comparing parameter options and information criterion for comparing models.
- Nevertheless, it is possible to calculate Bayes factors for complex problems using pseudo-priors or fractional likelihoods (where some data is incorporated into the prior). There are also intrinsic Bayes factors and cross validation Bayes factors.

Bayes factors example

- This example is from Ando (2010)
- Suppose we have 11 independent drivers and we count their accidents per year.
- We assume the accidents are Poisson but we have 2 priors in mind:
  1. $Gamma(2, 2)$, implying a mean of 1 and s.d. of 0.71.
  2. $Gamma(10, 10)$, implying a mean of 1 and s.d. of 0.32.
- We observe that 6 had zero accidents, 4 had one accident and 1 had three accidents.
- We thus arrive at the following posteriors:
  1. $Gamma(9, 13)$, implying a mean of 0.69 and s.d. of 0.23.
  2. $Gamma(17, 21)$, implying a mean of 0.81 and s.d. of 0.2.
- So which one is better, and how much better?

Bayes factors example p.2

- First consider model 0 on its own and try to obtain $p(\mathbf{y}|M_0)$.
- $p(\boldsymbol{\theta}_0|\mathbf{y}, M_0) = p(\mathbf{y}|M_0)^{-1} p(\boldsymbol{\theta}_0|M_0) p(\mathbf{y}|\boldsymbol{\theta}_0, M_0)$
- $\therefore \frac{13^9}{\Gamma(9)} \theta_0^8 e^{-13\theta_0} = p(\mathbf{y}|M_0)^{-1} \frac{2^2}{\Gamma(2)} \theta_0^1 e^{-2\theta_0} e^{-11\theta_0} \frac{\theta_0^7}{0!0!0!0!0!1!1!1!1!3!}$
- $\therefore p(\mathbf{y}|M_0) = \frac{2^2 \Gamma(9)}{13^9 \Gamma(2) 3!} \approx 0.000002534773$
- Similarly, $\frac{21^{17}}{\Gamma(17)} = p(\mathbf{y}|M_1)^{-1} \frac{10^{10}}{\Gamma(10)} \frac{1}{3!}$
- $\therefore p(\mathbf{y}|M_1) \approx 0.000003198728$
- And $BF_{1,0} \approx 1.26$
- Meaning that model 1 is slightly more supported by the data than model 0 but the difference is too small to trust.

Bayesian Goodness-of-fit

- People like a good *p-value* and it is understood outside statistics.
- So the question arises, 'Can we calculate a *p-value* for our model?'
- The answer is 'possibly', but only if we rephrase the question to:
- Assuming H_0 : the data did originate from the proposed model, what is the probability of observing a value of a test statistic at least as large as the one observed?
- In order to ensure that this probability behaves like an actual *p-value* we must obtain the probability by integrating over all possible replicate samples obtained from the posterior predictive distribution.
- Can only be done by simulation and very slow.
- Always remember that frequentist models assume a single true value for each parameter, but Bayes models don't!

Pragmatic Bayesian Goodness-of-fit: DHARMa

- A more pragmatic approach is to compare each individual observation to its own posterior predictive distribution.
- Practically, we look at the proportion of predictive simulations that are less than the observation, giving a value between 0 and 1 for each observation.
- If the model assumptions hold then these values as a group should be roughly uniform in distribution.
- We can then summarise them in various ways and test them against the uniform distribution.
- The above approach is easy to implement yourself for models fit using Bayes simulation tools; but applies to frequentist models also.
- The use of the DHARMa package in R is highly recommended, and the source of this idea.
- https://cran.r-project.org/web/packages/DHARMa/vignettes/DHARMa.html

Exercises 14

Previous Exercises

1. Find a regression example from any undergraduate course you did that had at least 30 observations. Implement the model in using Stan twice, first using *Normal* errors and then assuming t errors. Compare the two models using RMSE, coverage and DIC.
2. Simulate two sets of numbers, the first should be 12 numbers from a $N(0, 1)$ distribution and the second should be 18 numbers from a $N(3, 4)$ distribution. Do a Welch t test and then a Bayesian t test in assuming different variances for the two samples. Which one produces the smallest p-value?
3. What does the ROPE approach suggest for the above hypothesis test? What if the first sample was from a $N(2, 3)$ distribution instead?

Portfolio I

1. This is an individual assignment in which you explain everything you learned through the course.
2. This may be submitted on paper or electronically, but not both (either scan your paper notes or print the electronic notes).
3. It should be a mixture of rough notes and papers (like your class scribbles and test & assignment submissions) on the one hand; and neat summaries of what you learned from each activity on the other hand.
4. It must explain which parts of the course you found easy and which parts you struggled with, and why.
    - Some parts of the work are just difficult, but most often students struggle because they had other priorities and didn't invest the appropriate effort.
    - You could mention how your background prepared you for some parts better than others.

Portfolio II

5. You should discuss how you overcame any difficulties you encountered in the course.
6. You may also make constructive suggestions for improving the course. For example, can you think of ways to increase how much everybody learns without making the course more difficult?
7. NB: Keep this assignment in mind throughout the course so that it's easy at the end. Keep notes throughout the course.
8. The portfolio will be assessed on how well you captured what your learnt, **and** how much you actually learned. If you are organised and methodical then you will get a better result. **Being organised and keeping an audit trail of your work is incredibly important in industry.**

The end, for now...

- Thank you for doing this course.
- I hope you learned something useful!

Click here to quit (only works in Acrobat)