贝叶斯推断与 Stan 应用

王敏杰

2022-10-15

Stan 是当前主流的概率编程语言，主要用于贝叶斯推断，它使用先进的 Hamiltonian Monte Carlo (HMC) 采样技术，允许复杂的贝叶斯模型快速收敛，在社会学、生物、医学、物理、工程和商业等领域有广泛的应用。本报告介绍贝叶斯统计推断的数学原理以及 Stan 应用，并通过案例演示 Stan 在统计建模中的强大功能。

1 我们今天讲一个数据故事

假定你们校长心血来潮，给你交办了一个任务，让你估算下全校同学的平均身高，你想了想，这好办，来个全校普查，然后统计个均值。但是，很快就发现，这个方法不具备可行性，因为很多同学疫情隔离在家，来不了学校啊，全校普查似乎不现实。这时，统计学院的老师给你出了一个主意，让你随机选取 200 同学，然后根据这 200 名同学的身高，推算全校总体的情况。你觉得这个主意不错，很快就拿到了 200 位同学的身高。数据在这里

id	height
1	173.72
2	170.89
3	182.11
4	176.21
5	167.08

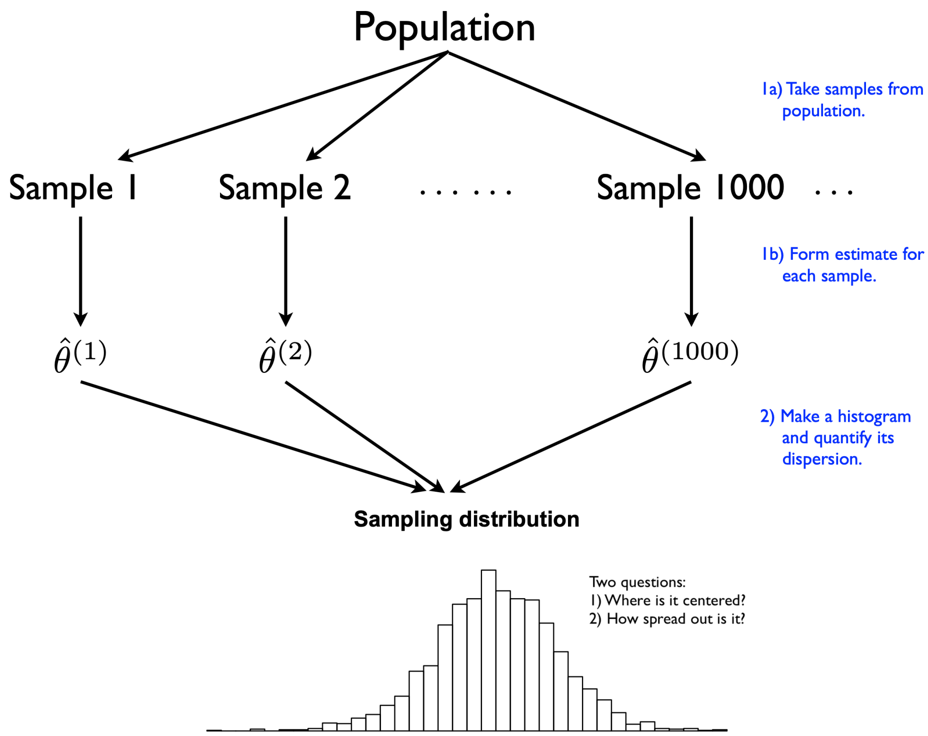
于是，不费吹灰之力很快就计算出 200 个同学的平均身高

mean_height
164.89

马上兴高采烈地报告校长，全校同学的身高均值是 164.89。

如果您是校长，您对结果满意?

应该不满意，因为选择的这 200 个学生，相对全校学生而言，是一个很小的样本，难免以偏概全了。这个 164.89 可靠性有多高，或者不确定性是多少？你看到校长脸色有些不好看，弱弱地说：“要不我重新再找 200 个学生，再做一次，或者再找第三组的 200 个学生，然后第四组，这样重复很多次”，你怕校长不明白你的意思，还把想法画成思维导图



还没等解释完，校长就打断了你的话，“时间来不及了，再说，这又不是做核酸，那有那么多的人力物力，你就用 200 个同学的身高值，给我估算一个吧，给个范围也行”，最后还不忘提醒一句，“今天就要！”

这下有点头疼了，只有一个样本，还要给出范围，那怎么办呢？

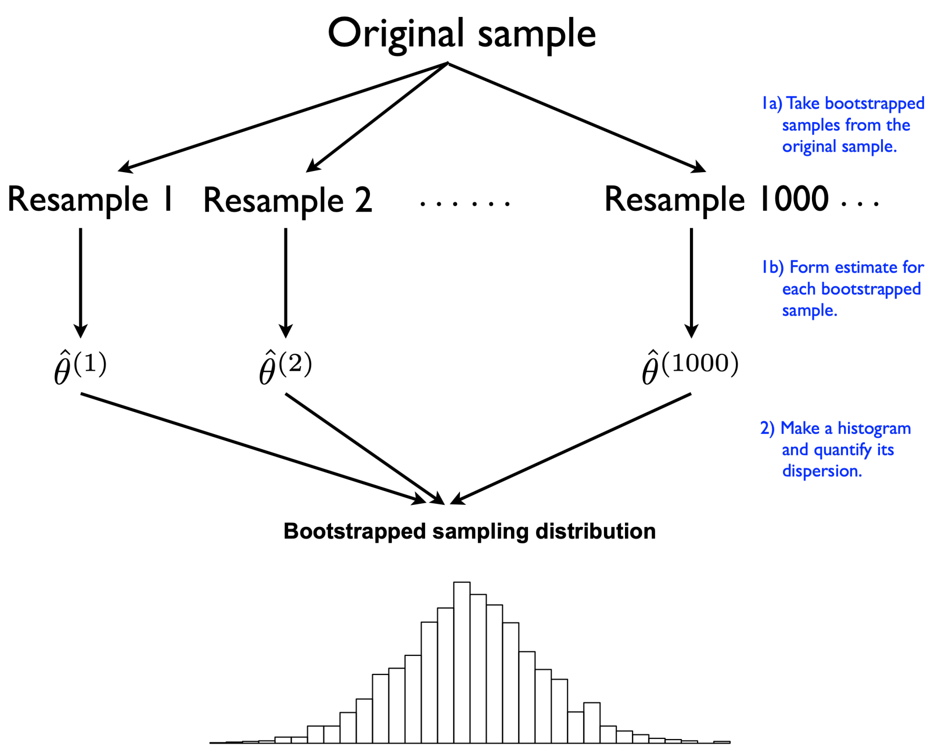
2 Bootstrap resampling

方法总是有的，现实不满足的时候，我们只是需要一点点妥协。校长说了，能拿到 200 个身高记录已经很不错了，那我们就假定我们手头上这唯一样本是随机的，而且能够代表总体分布，这个**假定**就是妥协。接下来那怎么办呢？

搞统计的人发明了一个很不错的方法Bootstrapping，**有放回的重复抽样**，什么意思呢？我举个通俗的例子：

- 假定这里有一个口袋，里面装着 200 个球，你摸一个出来，记录下这个球的重量，然后放回去，搅拌一下，再摸一个出来，称下重量，再放回去，如此往复，记录到 200 个值后，就停下来，这 200 个值称之为第一个重抽样样本，然后计算下**均值**。ok，第一个样本的工作完成。
- 然后第二个样本
- 第三个样本...
- 直到 1000 个重抽样样本，也就得到了 1000 个均值
- 最后看看这 1000 个均值的分布

画出思维导图如下



道理明白了，那马上开干

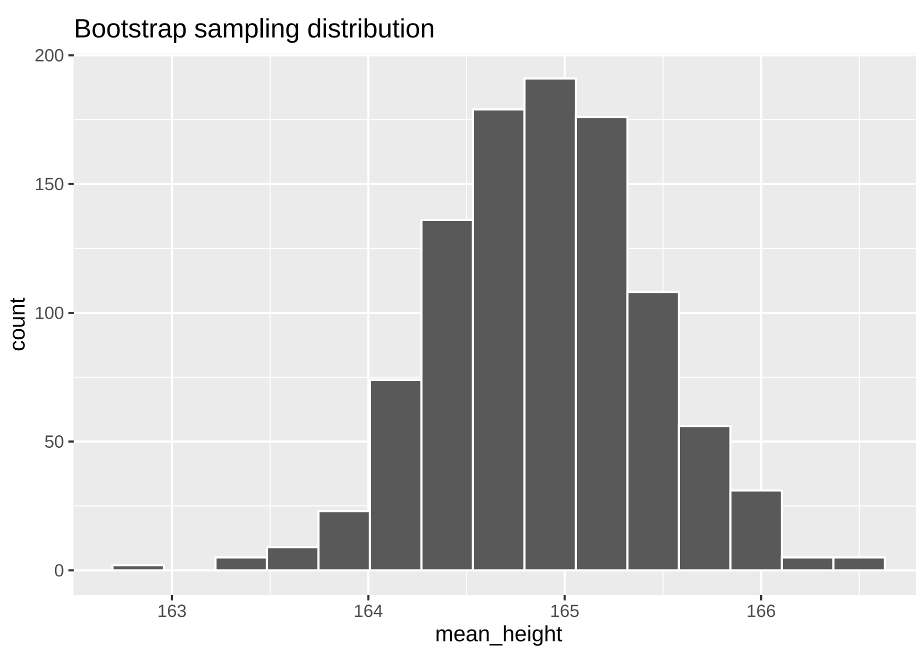
当然也可以使用tidymodels宏包完成

```r
library(tidymodels)

boots <- rsample::bootstraps(data = d, times = 1000)

df_bootstrap <- boots %>%
  mutate(
    mean_height = map_dbl(splits, ~ .x %>% analysis() %>% pull(height) %>% mean())
  )
```

很快得到了 1000 个均值

replicate	mean_height
1	164.9945
2	166.1313
3	165.1924
4	165.0120
5	164.2308
6	166.0839

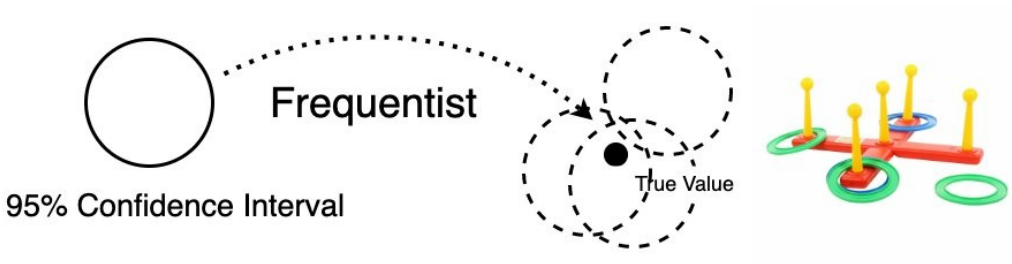
画出了直方图



给出置信区间

mean_height	.lower	.upper	.width	.point	.interval
164.9	163.9	165.93	0.95	mean	qi

置信区间，是频率学的词汇。关于 95% 的置信区间，要多说两句，我们想统计的是全校同学的身高均值，这是我们的目标，对应到这张图中，这个黑色圆点，它代表着全校同学的平均身高，它是客观存在的，并且有一个确定的值（只是我们不知道)。



我们的任务就是，捕获这个黑色圆点，这个过程有点像小朋友玩的**套圈游戏**。我们常说的 95% 的置信区间，置信区间就是圈圈的大小，95% 的意思就是，我们扔套圈 100 次，95 次成功套住黑点，换句话说，扔一次套圈，我有 95% 的概率能捕获到黑色圆点。当然，如果这个套圈做的特别大，都比场地还大，我们自然可以说，我有 100% 的概率能套住这个黑点，但没什么意义。

回到身高问题中来，上面的统计结果告诉我们，我们把套圈设置到 [163.9, 165.93] 这个范围，我就有 95% 的自信说，这个区间能捕获全校同学的平均身高。

需要提醒注意的是，频率学派的思维方式：

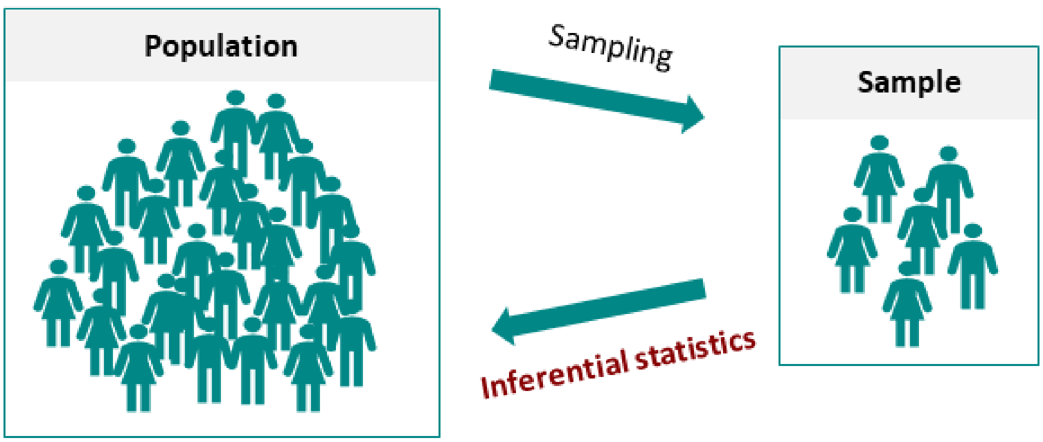
- 我们的目标，全体同学的身高均值，是确定的，没有不确定性。
- 用于捕获固定值的置信区间，具有不确定性，因为它可能捕获到，也有可能捕获不到。

3 来点高级的？

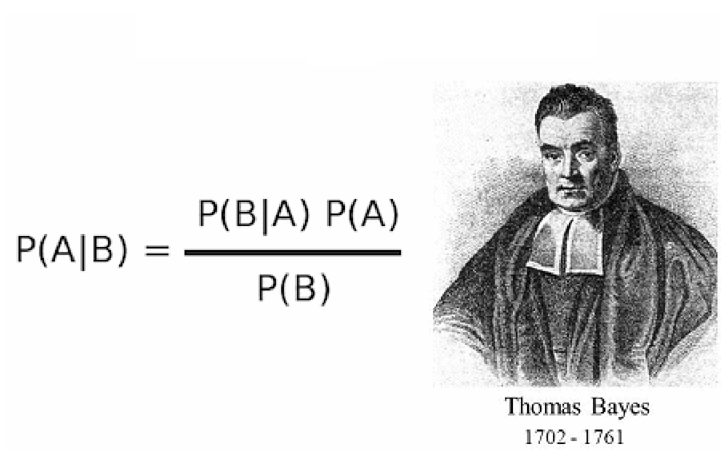
以上是频率学派的方法，没有什么问题。但天天吃这个，吃了几十年了，口味难免越吃越高，想搞点新鲜的。什么是新鲜的呢?

对的，贝叶斯。我们已经久仰贝叶斯大名。

回到故事的开始，观察到样本数据后，如何推断总体分布的参数 θ?



贝爷用贝叶斯公式，告诉我们可以用贝叶斯后验概率 $p(\theta|Y)$ 来回答



三百年前的光辉思想，仍然影响着现在。我们要好好膜拜下这个贝叶斯公式

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)}$$

贝叶斯公式告诉我们，要得到等式的左边，可以用等式的右边来计算。先认识下贝叶斯公式的每个部分。

- 左边 $p(\theta|Y)$ 称之为后验概率，也就是我们的目标
- $p(Y|\theta)$ 是似然函数，在给定参数后，数据出现的概率
- $p(\theta)$ 参数的先验概率，在看到数据前，参数各种可能性的分布
- $p(Y)$ 边际似然，可以忽略

既然分母可以先忽略，就认为它为 1，于是等式可以变成

$$p(\theta|Y) \propto p(Y|\theta)p(\theta).$$

然后，我们把总体的似然函数，写成每个数据点的似然函数**连乘**的形式：

$$p(\theta|Y) \propto p(\theta)\prod_{n=1}^{N} p(y_n|\theta)$$

接着，我们两边取对数，连乘变成了连加。也就说，我们计算的是**对数概率 (log probabilities)**。使用对数概率可以有效提高数值稳定性

$$\log\, p(\theta|Y) \propto \log\, p(\theta) + \sum_{n=1}^{N} \log\, p(y_n|\theta)$$

看看这个等式，感觉技术上有可操作性了，没错，它就是贝叶斯数据分析的**灵魂**，我们做贝叶斯计算都是仰仗这个等式。

我们跃跃欲试了，不过，先忘掉之前的统计方法。

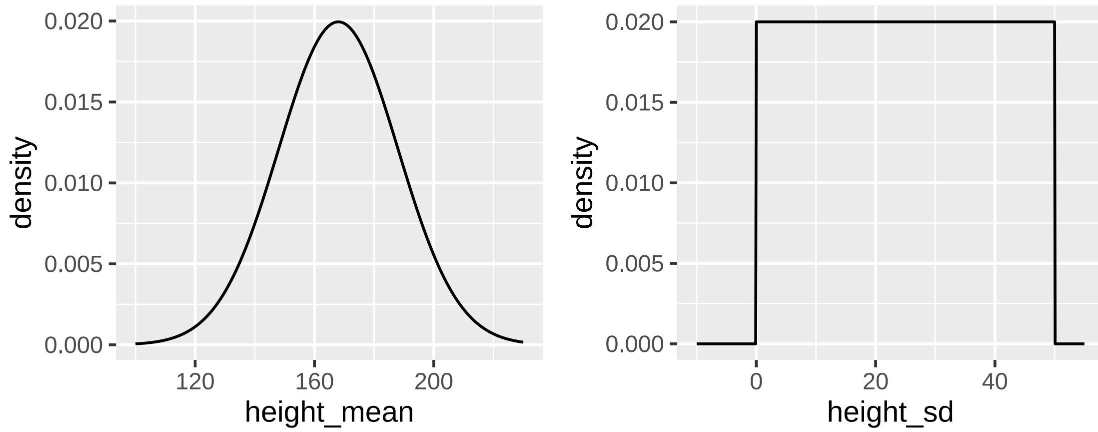
3.1 需要一点前戏

还是校长给出的**身高问题**。通过前面的身高的统计量，我们可以合理的猜测：

- 全校同学的身高均值可能是 160，162，170，172，…，或者说这个均值在一个范围之内，在这个范围内，有些值的可能性大，有些值可能性较低。比如，认为这值游离在 [150,180] 范围，其中 168 左右的可能最大，两端的可能性最低。如果寻求数学语言来描述，它比较符合正态分布的特征，那就这么定了。

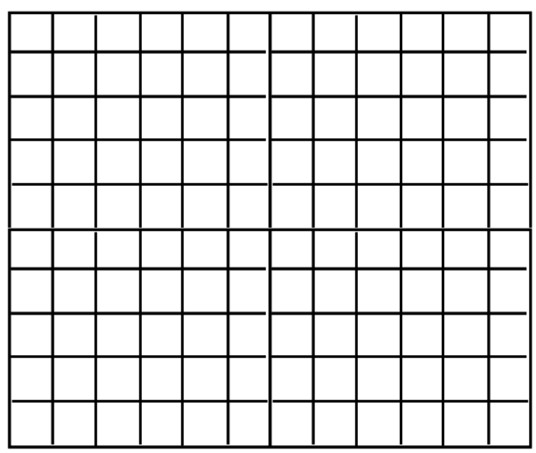
- 方差在 [0, 50] 范围内都有可能，那就假定每个值的可能性都相等吧。

这两点没有问题，因为符合现实情况，合情合理。现在把我们的猜测画出来，就是这样的，



3.2 参数空间

第二步,我们需要构建一个**参数空间**,类似网格一样的东西。具体做法是,先指定参数的范围,大一点没关系,然后把这个范围内的所有**可能参数组合**都罗列出来,类似九九乘法表,比如这里构建 1000x1000 个 (μ, σ) 参数空间

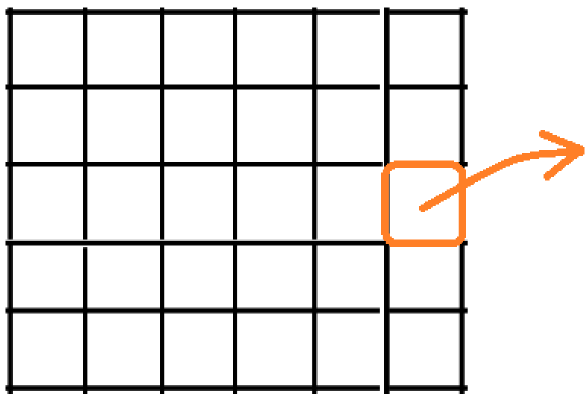


mu	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$	$\sigma = 6$	$\sigma = 7$	$\sigma = 8$
$\mu = 171$	N(171,2)	N(171,3)	N(171,4)	N(171,5)	N(171,6)	N(171,7)	N(171,8)
$\mu = 172$	N(172,2)	N(172,3)	N(172,4)	N(172,5)	N(172,6)	N(172,7)	N(172,8)
$\mu = 173$	N(173,2)	N(173,3)	N(173,4)	N(173,5)	N(173,6)	N(173,7)	N(173,8)
$\mu = 174$	N(174,2)	N(174,3)	N(174,4)	N(174,5)	N(174,6)	N(174,7)	N(174,8)
$\mu = 175$	N(175,2)	N(175,3)	N(175,4)	N(175,5)	N(175,6)	N(175,7)	N(175,8)
$\mu = 176$	N(176,2)	N(176,3)	N(176,4)	N(176,5)	N(176,6)	N(176,7)	N(176,8)
$\mu = 177$	N(177,2)	N(177,3)	N(177,4)	N(177,5)	N(177,6)	N(177,7)	N(177,8)

3.3 先验概率的对数

第三步，在参数空间里，计算每个参数在先验分布下的概率密度对数，即下面等式红色部分 (不是说均值是正态分布的么，那么 160 出现的概率是多少，161 出现的概率是多少，当然最后求对数)

$$\log\, p(\theta|Y) \propto \log\, p(\theta) + \sum_{n=1}^{N} \log\, p(y_n|\theta)$$



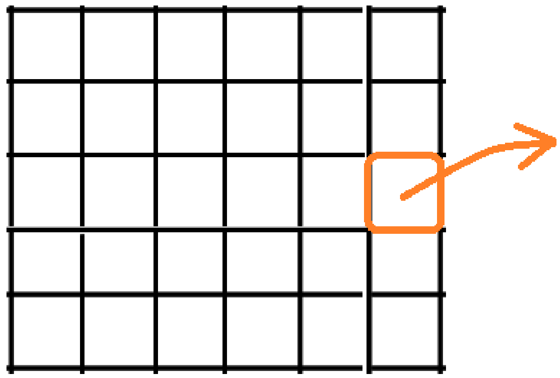
```r
mu = 175
dnorm(mu, mean = 168, sd = 20, log = TRUE)
#> [1] -3.975921


sigma = 10
dunif(sigma, min  = 0,   max = 50, log = T)
#> [1] -3.912023
```

3.3.1 对数似然

第四步，在参数空间里，每个参数组合所对应的分布下，计算观察到的 200 个身高值的**对数似然**之和，即下面等式红色部分

$$\log\, p(\theta|Y) \propto \log\, p(\theta) + \sum_{n=1}^{N} \log\, p(y_n|\theta)$$

这里有 1000x1000 个 (μ, σ) 组合，所以会产生 1000x1000 个值



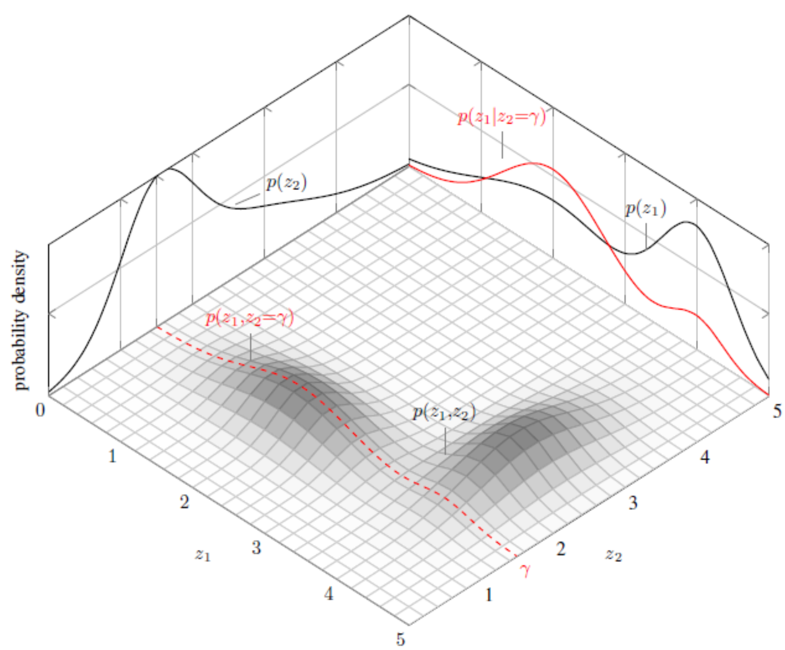
```r
head(d, 6)
#>   id height
#> 1  1 173.72
#> 2  2 170.89
#> 3  3 182.11
#> 4  4 176.21
#> 5  5 167.08
#> 6  6 183.12

sum(dnorm(d$height, mean = 175, sd = 10, log = T))
#> [1] -799.5636
```

3.3.2 后验概率的对数

第五步，把先验概率的对数和似然概率的对数加起来，得到了后验概率对数 (log probabilities)，求指数后，就是**后验概率**

$$\log\, p(\theta|Y) \propto \log\, p(\theta) + \sum_{n=1}^{N} \log\, p(y_n|\theta)$$

此时，可以想象成一共有 1000 x 1000 个坑，每个坑装着一个后验概率，有高有低，看上去就像若干个小山峰。

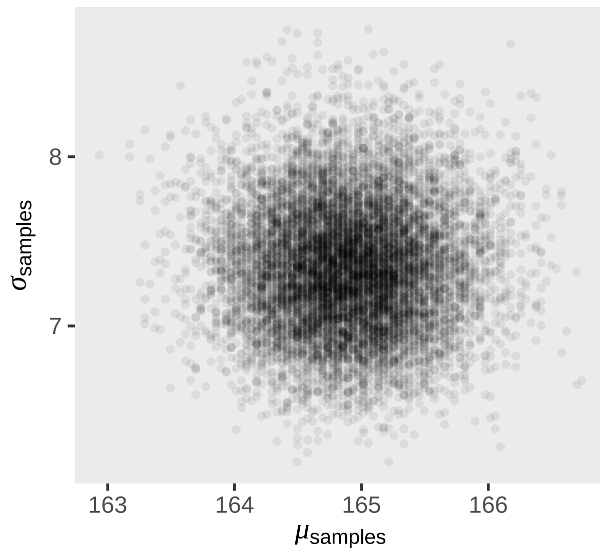


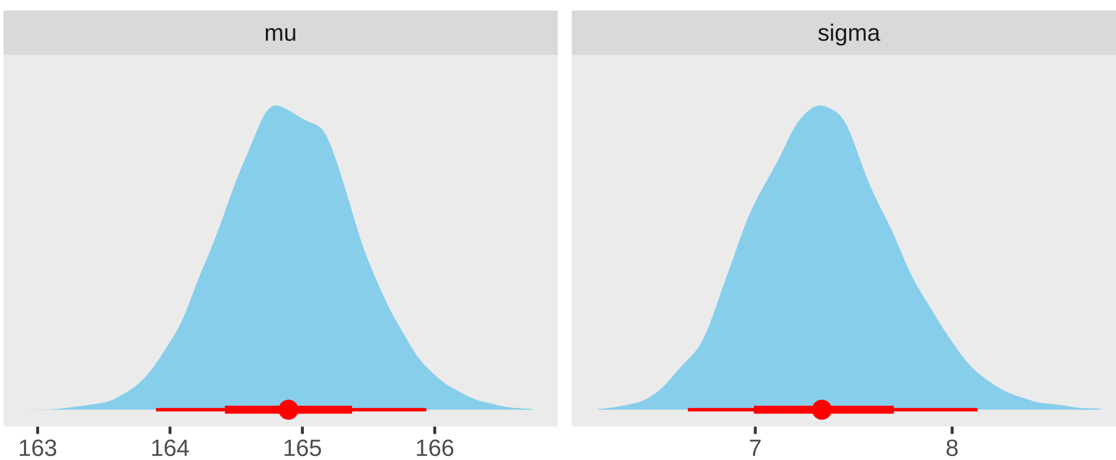
3.3.3 抽样

第六步，按照后验概率值的大小抽取样本，得到后验分布。

为什么要抽样呢，因为目前得到的只是概率对数 (求指数后是概率)，即每个坑出现的概率，而我们要得到是参数的具体值，身高的均值，所以按照概率大小抽取样本。

有了样本，就可以得到均值和标准差的分布



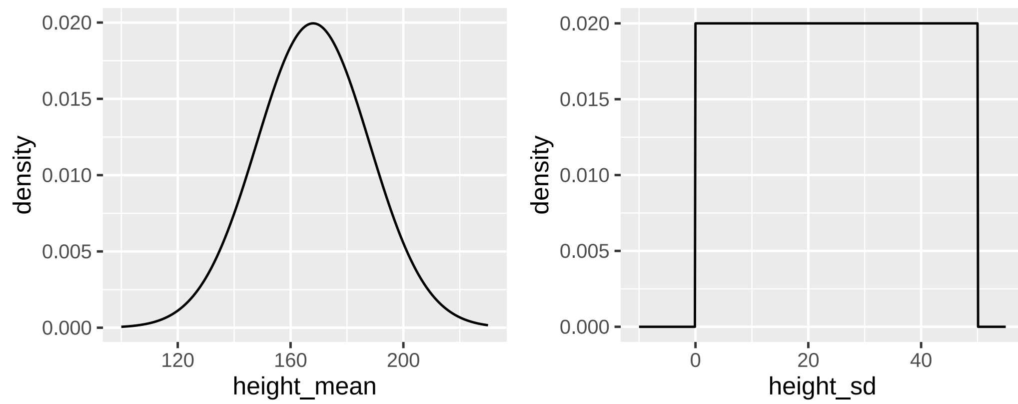


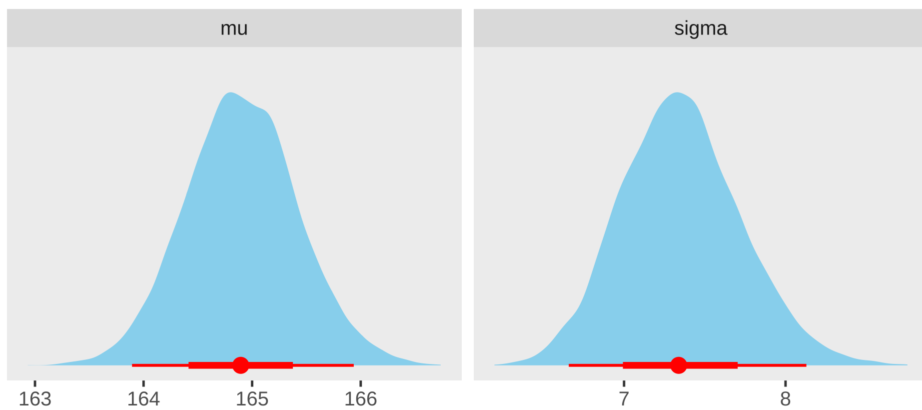
以及后验概率的**最高密度区间**

name	value	.lower	.upper	.width	.point	.interval
mu	164.801337	164.054054	165.695696	0.89	mode	hdi
sigma	7.328328	6.772773	7.943944	0.89	mode	hdi

此时，给出的不在点估计，而是区间估计。给出了各种可能值，以及各可能值的概率，比频率学给出的信息要丰富很多。

3.3.4 后验和先验对比





3.3.5 回望下贝叶斯

回望下贝叶斯的思想

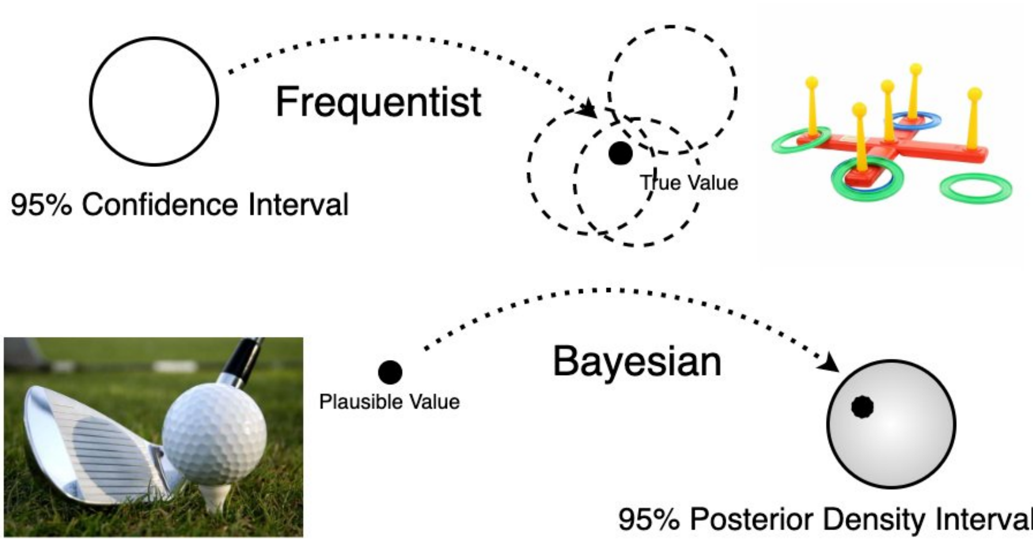
- 我们先赋予参数主观的先验信息，也就意味着参数是变化的值
- 但数据是固定的
- 数据更新了先验信息，得到了后验信息

贝叶斯推断符合人们的认知过程。人们总是根据新获得的信息来修正或更新先前的知识或信念。举个例子，最近有个演员叫李易峰，年轻人长的非常帅，演技又好，又喜欢帮助困难群众，完全是正面的形象，这是我们的先验。或者说是在没有得到进一步信息的之前，我们对这个人物的信念。但他的新闻出来之后，大众就开始修正先前的想法。

我们学习知识的过程，也是贝叶斯的过程。有趣的是，如果你对某个命题有着极强的信念（以 100% 的先验信念度来表示），那么不管关于这个命题的新信息（当前概率）如何，贝叶斯定理表明你的后验信念度与先验信念度一样是 100%，即你的信念不受新信息的影响。这就叫**已经被洗脑**。

而另一方面，如果你对某个命题没有任何偏见（以 50% 的先验信念度来表示），这时如果关于这个命题的新信息给出的当前概率为 P%，贝叶斯定理表明你的后验信念度也是 P%，即你完全接受新信息。实际生活中，人们总会或多或少受到新信息的影响而修正自己先前的想法（信念）。

好，继续回到我们的问题，这里我们把频率学和贝叶斯两种不同的方法，放在一起，对比一下。



下图贝叶斯的黑色圆点，是我们的目标。按照贝叶斯的观点，它不是一个固定的或者确定的值，而是各种可能的值，贝叶斯给出的是，**最有可能的是哪些值，以及这些可能值的概率是多少**。这里用高尔夫球演示，表示的是，很多球都可以打进洞，最有可能落在球洞的是哪些球，以及落入的概率是多少。高尔夫我没玩过，换成足球射门来理解也是一样，位置很偏、角度很刁钻的球，能不能进？也能，但概率很低而已。全校身高 150 可不可能，也有可能，只是这种可能性很低而已。

以上是通过**网格近似**的方法得到身高分布的后验概率，这个理解起来并不难，但这种方法做起来比较麻烦，需要构建参数网格，对于较复杂的模型，计算量会陡增，内存占用大、比较费时，因此在实际的数据中，一般不采用这种方法，但网格近似的方法可以帮助我们很好地理解贝叶斯数据分析。

4 轮到今天的主角们

4.1 概率编程工具

- BUGS (Bayesian inference Using Gibbs Sampling)，2007 年后没有维护
- JAGS (Just Another Gibbs Sampler)
- PyMC (Python)
- Turing.Jl (Julia)
- Stan

前面四个我都没用过，接触到 Stan，也纯属偶然，我也只懂一点皮毛。总体感觉 Stan 比较新，更新比较快（不一定是坏事，可以和它同步成长），Stan 的数据结构和 R 比较接近，还有一点就是它的学习资料相对丰富点，Stan 生态要成熟点，有专门的团队维护和答疑。

4.2 什么是 Stan

Stan 是一门统计编程语言，主要用于贝叶斯推断。贝叶斯统计已经有近 300 年的历史，直到最近几十年，得益于理论进步和计算机计算能力的提升，贝叶斯统计才越来越多得到了统计学、其它学科以

及工业领域的重视。Stan 广泛应用于社会学、生物、医学、物理、工程和商业等领域。也就说，贝叶斯不是新东西，但 Stan 是新东西。

可以点开主页 https://mc-stan.org/ 了解

4.3 Stan 的历史

John von Neumann， Stanislav Ulam， Nicholas Metropolis

波兰犹太裔核物理学家 Stanislaw Ulam (1909-1984)，于美国在第二次世界大战期间，在研究原子弹时，发明了蒙特卡罗 (MonteCarlo) 方法。蒙特卡罗方法是什么呢？以概率统计理论为指导的数值计算方法。为什么叫蒙特卡罗呢？据说，因为他的叔叔喜欢去摩纳哥的蒙特卡洛赌场，而且经常输钱。他就研究了赌博的问题，他认为赌城赌钱是概率统计问题。后来他就把他研究的这种方法，叫做蒙特卡罗方法。也就说，用一个赌城的名字，命名指代一种统计方法。总之，蒙特卡罗方法很优秀，所以贝叶斯界用这种方法开发一套程序，就是今天我们讲的这个，并用它创始人的名字 Stan 命名。

这套程序是由纽约哥伦比亚大学 Andrew Gelman 于 2012 年发起，由核心开发团队共同开发和维护。

4.4 Stan 如何工作

Stan 先编译再抽样

- Stan 首先会把 Stan 代码翻译成 C++，然后在本地编译。
- Stan 使用先进的采样技术，允许复杂的贝叶斯模型快速收敛，Stan 用的是 Hamiltonian Monte Carlo 技术的 No-U-turn 采样器。

把你模型的后验概率想象成一个有峰有谷的小行星。HMC/NUTS 的宇航员是在这个星球上着陆的，宇航员们的目标是在这片土地上找到最高的山峰。在每一个点上测量海拔，这是可行的，但如果小行星真的很大的话，那就要花很长时间。这就需要高级的算法了。

这里面太多数学和计算机的内容了，是核心科技，我真不太懂，求放过。不懂原理，也不用太纠结，Stan 只是一个工具，好比汽车一样，安全驾驶就好，至于发动机原理可以先放一放。

4.5 如何使用 Stan

我是把 Stan 当作 R/Python 的一个宏包。一般来说，我们不会单独使用 Stan，而是在 R 语言里完成数据规整变型后，喂给 Stan，Stan 返回样本后，可以接着在 R 里统计分析。也就说，**Stan 可以当作你已经掌握的数据分析工具的一种插件，当作 R 语言的一种扩展和增强**。

- Stan 提供了与（R，Python，shell，MATLAB，Julia，Stata）流行语言的接口
  - 在 R 语言里用 rstan，CmdStanR 包，CmdStanR 能随时跟进最新 Stan 的更新
  - 在 Python 用 PyStan 包
- 在 R 语言里，还可以用 bayesplot，tidybayes, loo 等宏包帮助我们完成 Stan 模型可视化、规整和分析。也就说，R 与 Stan 配合非常默契，简直就是绝配。

4.6 Stan 的优势

相比于传统的方法来说，Stan 模型

- 更好的可操作性

  - 从模型表达式到代码，更符合人的直觉
  - 模型灵活性。修改几行代码，就转化成一个新的模型
- 更好的透明性
  - 模型的假设
  - 模型的参数
- 更好的可解释性
  - 从贝叶斯公式出发，解释起来更符合常识

什么叫可操作性，我们写出了数学表达，应用到具体案例中，就需要转换成代码，这个转换是最痛苦的。很多同学和我一样，知道了数学模型或者数学表达式，但不知道怎么写代码去实现它，然后去百度 R 语言代码，依葫芦画瓢地把数据套进去，一运行，有结果了，但是不知道结果什么意思。从这个角度讲，R 语言对新手并不友好，因为 R 它认为我们什么都懂，你应该能看懂结果。但事实上，我们还是看不懂。造成这个现状的原因是，**往往用统计的都不是学统计**。Stan 语言让**从模型到代码的转化**更加自然，有更好的直觉（这点和 tidyverse 一样，机器语言接近人类语言，看代码和与人说话一样）。同时，可操作性还表现在模型的**灵活性**，一个模型只需要修改几行代码，就可以转化成一个新的模型。

什么叫透明性？指的是模型结构，比如模型的假设，模型的参数。我们刚才说过，在 R 语言里面，经常会遇到，代码跑出来了，但是不知道怎么解释模型结果，主要原因还是对模型结构不清楚。**因为**，任何模型都有假设的。不知道模型基于的哪些假设，只拿到一些参数结果，解释起来也自然会很痛苦。比如：

- 线性回归 lm() 有哪些假设？
- 多层模型 lmer() 有哪些参数？
- t.test() 检验的啥？

这些函数都很强大，但使用也有一些代价，就是需要对它足够了解，否则容易弄错，导致很古怪的解释。**对于 Stan 用户而言，就不会有这样的抱怨和困扰，因为模型的结构很清晰**

可解释性？可解释性是上面透明性的延续。一方面，模型结构弄清楚了，模型的参数解释起来就很自然了，另一方面，模型是从贝叶斯公式出发的，融入了先验知识后，从贝叶斯的角度解释参数，比基于频率学的角度，解释起来更符合常识。

4.7 Stan 代码框架

Stan 语法非常严谨，数据结构接近 R 语言，声明语句类似 C++ 语言，但我们并不需要搞懂 C++ 后才来学 Stan，能看懂 R 语言，也能看懂 Stan，具体可以参考官方手册。

```
data{
        // 导入数据
}
parameters{
        // 定义模型要估计的参数
}
```

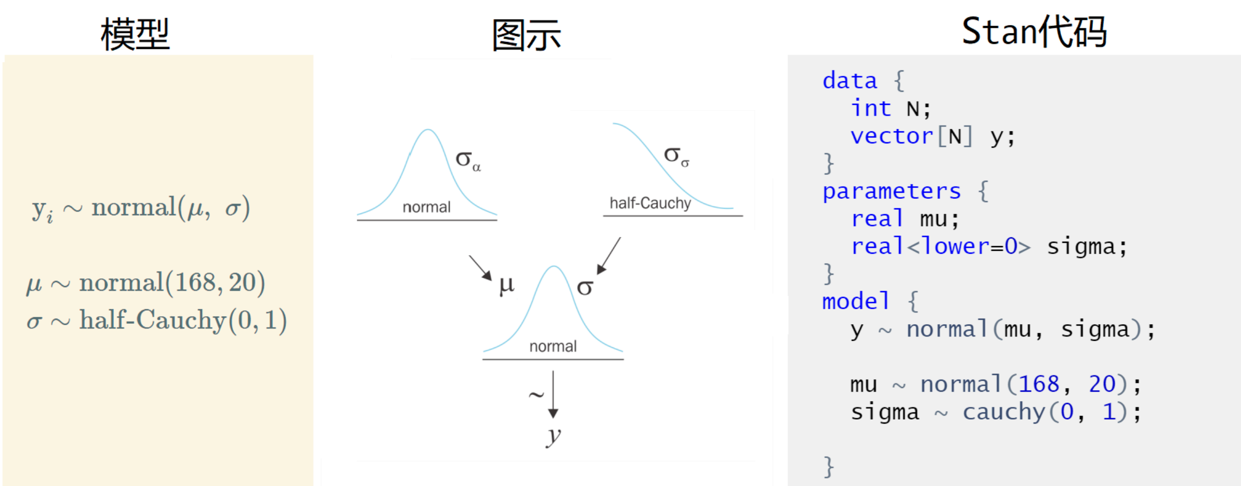
```
model{
        // 后验概率函数
}
```

下面通过一些案例，先让大家了解 Stan 的强大和价值。

4.8 从模型到 Stan 代码

还是全校同学的身高问题，我们可以写成如下表达式

$$
\begin{aligned}
\text{height}_i &\sim \text{normal}(\mu,\ \sigma) \\
\mu &\sim \text{normal}(168, 20) \\
\sigma &\sim \text{half-Cauchy}(0, 1)
\end{aligned}
$$



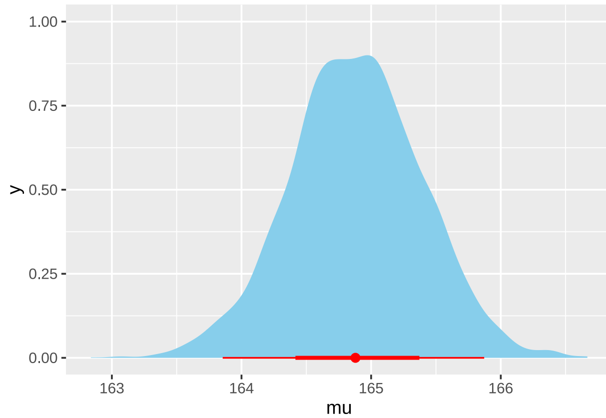

```
##
## data {
##   int<lower=0> N;
##   vector[N] y;
## }
## parameters {
##   real mu;
##   real<lower=0> sigma;
## }
## model {
##   mu ~ normal(168, 20);
##   sigma ~ cauchy(0, 1);
##
##   y ~ normal(mu, sigma);
## }
```

默认情况下，派出的 4 个宇航员从不同的起点出发去采样，每个宇航员带回 1000 个样本，因此会带回 4000 个样本。

lp__	mu	sigma	.chain	.iteration	.draw
-500.157	165.343	6.92914	1	1	1
-500.707	165.127	6.71700	1	2	2
-499.970	165.005	6.86549	1	3	3
-500.474	164.472	6.82684	1	4	4
-500.804	165.022	7.95160	1	5	5
-501.697	164.910	6.53771	1	6	6
-502.611	164.125	8.15492	1	7	7
-502.303	165.401	6.53663	1	8	8
-501.953	164.805	8.19843	1	9	9
-501.790	164.482	6.58063	1	10	10
-500.228	164.699	6.81306	1	11	11
-499.476	164.721	7.48232	1	12	12
-500.486	164.089	7.34872	1	13	13
-500.629	165.698	7.11352	1	14	14
-499.854	164.929	7.68416	1	15	15
-500.210	164.771	6.80551	1	16	16

啊哈，得到了样本。是最高兴的事情

还是看图过瘾，赶紧画个图吧



.variable	.value	.lower	.upper	.width	.point	.interval
mu	164.974106	164.10100	165.71000	0.89	mode	hdi
sigma	7.347706	6.71451	7.88741	0.89	mode	hdi

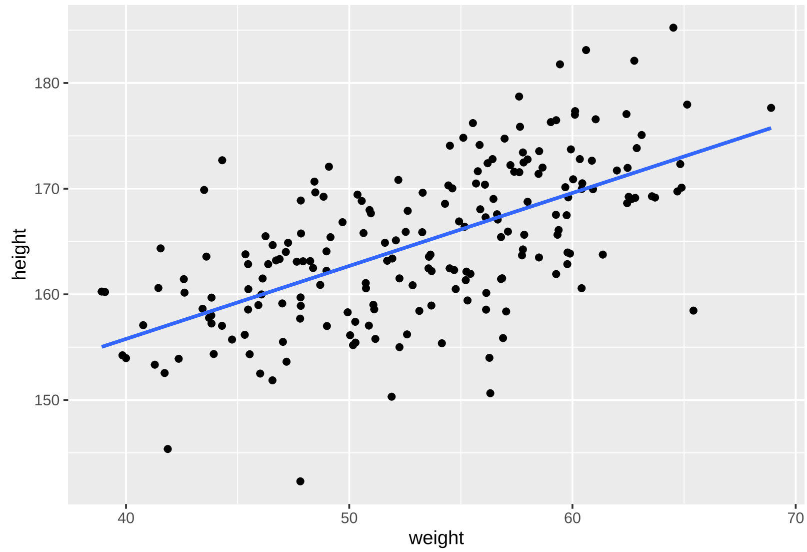
5 线性模型

这么好的东西，只用来估计一个均值，太可惜了啊，用它可以搞很多模型啊

还是身高数据，只是我在测量身高的时候，偷偷也测量了体重

id	height	weight
1	173.72	59.93
2	170.89	60.03
3	182.11	62.77
4	176.21	55.54
5	167.08	56.65
6	183.12	60.61

那我们可以探索身高和体重的关联了

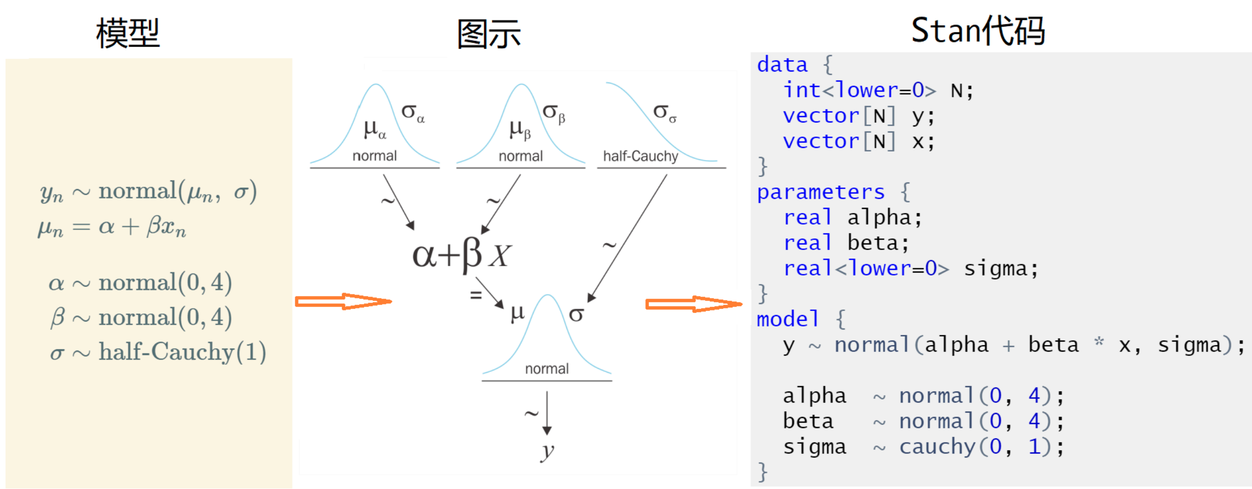


假定身高和体重满足线性关系，数学表达式如下

$$
\begin{aligned}
\text{height}_i &\sim \text{normal}(\mu_i,\ \sigma) \\
\mu_i &= \alpha + \beta\ \text{weight}_i \\
\alpha &\sim \text{normal}(0, 4) \\
\beta &\sim \text{normal}(0, 4) \\
\sigma &\sim \text{half-Cauchy}(1)
\end{aligned}
$$

我**强烈推荐**这样写，因为，这种写法可以很方便地过渡到其它模型。

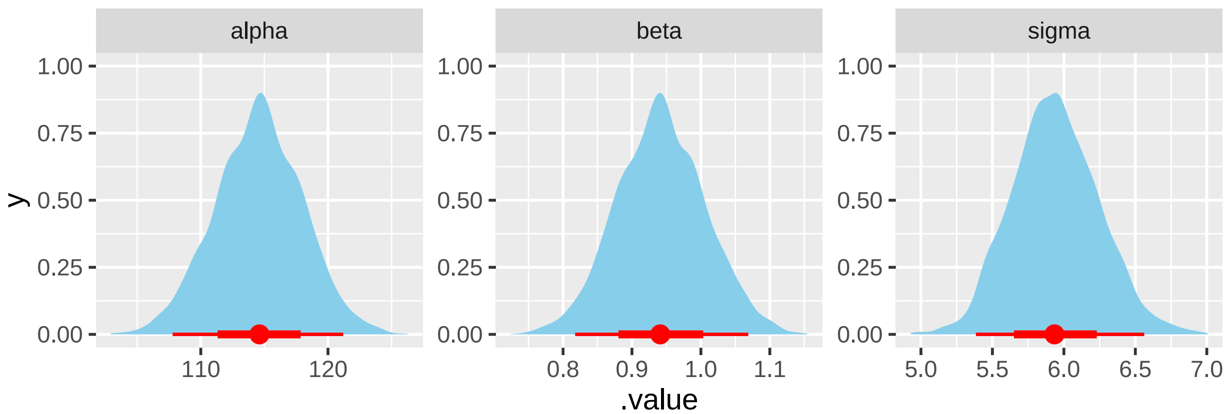
5.1 从模型到 Stan 代码




```
##
## data {
```

```
##   int<lower=0> N;
##   vector[N] y;
##   vector[N] x;
## }
## parameters {
##   real alpha;
##   real beta;
##   real<lower=0> sigma;
## }
## model {
##   y ~ normal(alpha + beta * x, sigma);
##
##   alpha  ~ normal(0, 10);
##   beta   ~ normal(0, 10);
##   sigma  ~ exponential(1);
## }
```

获得样本后，很容易得到参数的后验分布



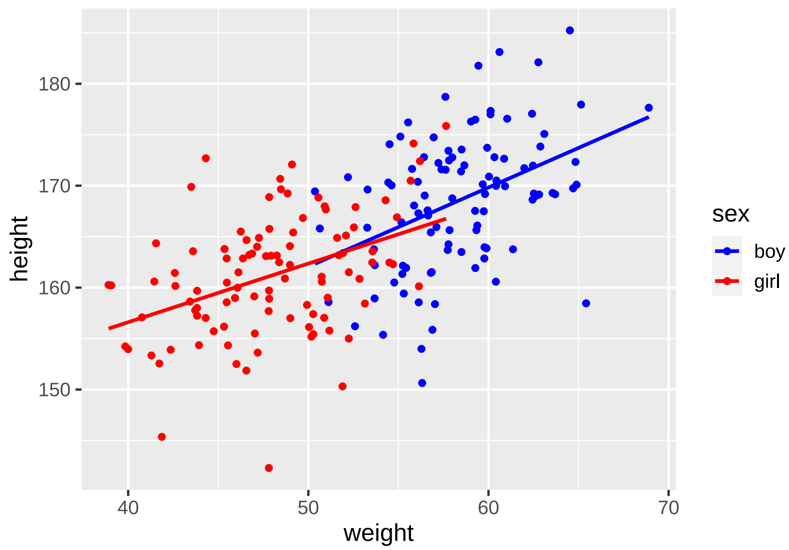
以及最高密度区间

.variable	.value	.lower	.upper	.width	.point	.interval
alpha	114.7106536	109.186000	120.14600	0.89	mode	hdi
beta	0.9407827	0.844006	1.04860	0.89	mode	hdi
sigma	5.9403992	5.445640	6.40979	0.89	mode	hdi

6 多层模型

我们再进一步，不同性别身高和体重的关系，应该是不一样的，我们也探索下吧

id	sex	height	weight
1	boy	173.72	59.93
2	boy	170.89	60.03
3	boy	182.11	62.77
4	boy	176.21	55.54
5	boy	167.08	56.65
6	boy	183.12	60.61



这里不是单纯的两个独立的回归分析，而是分成男孩和女孩两组，模型中我们既要考虑组内的变化，又要考虑组与组的之间的变化。因此，多层模型写成如下形式

$$
\begin{aligned}
\text{height}_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha_{j[i]} + \beta_{j[i]}\text{weight}_i \\
\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} &\sim N\left(\begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}, \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} \right)
\end{aligned}
$$

然后加上先验

$$
\begin{aligned}
\mu_\alpha &\sim \text{Normal}(0, 2) \\
\mu_\beta &\sim \text{Normal}(0, 2) \\
\sigma &\sim \text{Exponential}(1) \\
\sigma_\alpha &\sim \text{Exponential}(1) \\
\sigma_\beta &\sim \text{Exponential}(1) \\
\rho &\sim \text{LKJcorr}(2)
\end{aligned}
$$

```
##
## data {
##   int N;                        // number of obs
##   int K;                        // number of predictors
##   matrix[N, K] X;               // model_matrix
##   vector[N] y;                  // y
##   int J;                        // number of grouping
##   int<lower=1, upper=J> g[N]; // index for grouping
## }
## parameters {
##   array[J] vector[K] beta;
##   vector[K] MU;
##   real<lower=0> sigma;
##
##   vector<lower=0>[K] tau;
##   corr_matrix[K] Rho;
## }
## model {
##   vector[N] mu;
##   sigma ~ exponential(1);
##   tau ~ exponential(1);
##   Rho ~ lkj_corr(2);
##
##   for(i in 1:N) {
##     mu[i] = X[i] * beta[g[i]];
##   }
##   y ~ normal(mu, sigma);
##
##   beta ~ multi_normal(MU, quad_form_diag(Rho, tau));
## }
```

可以得到男孩和女孩不同的截距和斜率

variable	mean	median	sd	mad	q5	q95	rhat	ess_bulk	ess_tail
beta[1,1]	130.680	130.435	5.215	5.237	121.638	138.420	1.009	228.339	551.652
beta[2,1]	130.773	130.688	5.081	5.123	122.065	138.553	1.009	227.615	500.693
beta[1,2]	0.649	0.652	0.090	0.090	0.514	0.807	1.010	223.772	598.904
beta[2,2]	0.634	0.636	0.105	0.105	0.472	0.815	1.008	229.950	529.351

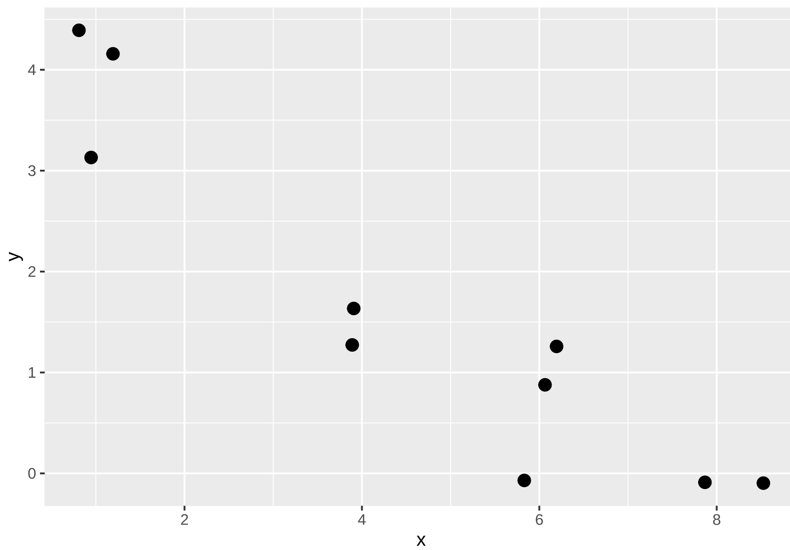
给出相关系数

variable	mean	median	sd	mad	q5	q95	rhat	ess_bulk	ess_tail
Rho[1,1]	1.000	1.000	0.000	0.000	1.000	1.000	NA	NA	NA
Rho[2,1]	-0.002	-0.014	0.425	0.513	-0.676	0.673	1.006	469.908	1113.303
Rho[1,2]	-0.002	-0.014	0.425	0.513	-0.676	0.673	1.006	469.908	1113.303
Rho[2,2]	1.000	1.000	0.000	0.000	1.000	1.000	NA	NA	NA

7 非线性的案例

我们再来看看，非线性的例子

图中的数据点很少，只有 10 个



假定 x 和 y 满足下面等式的关系

$$y_i = ae^{-bx_i} + \epsilon_i$$
$$\epsilon_i \sim \text{normal}(0,\ \sigma)$$

写成如下等价这种形式，更好理解

$$y_i \sim \text{normal}(\mu_i, \sigma)$$
$$\mu_i = ae^{-bx_i}$$

这里的问题是，如何估计 a 和 b？

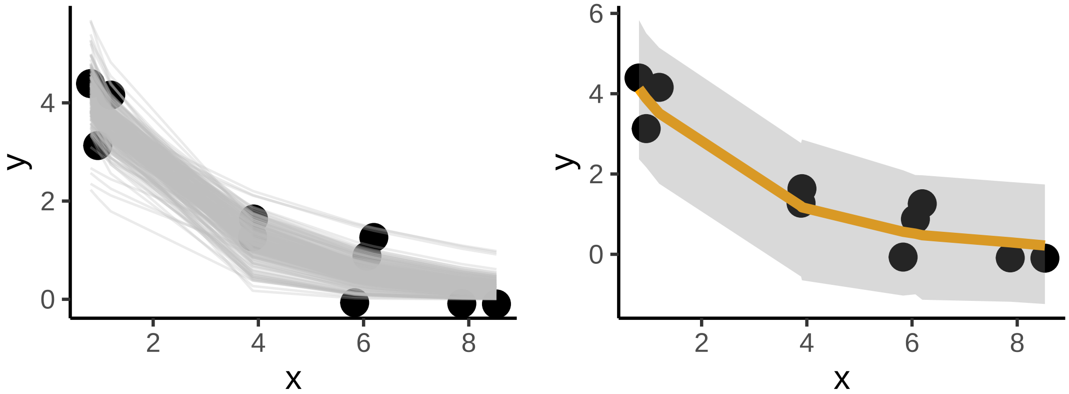
李教授给我布置的作业，是多层和非线性的组合。我们这里的问题要简单很多，但结构上是类似的。

```
##
## data {
```

```
##   int N;
##   vector[N] x;
##   vector[N] y;
## }
## parameters {
##   real a;
##   real b;
##   real sigma;
## }
## model {
##
##   y ~ normal(a * exp(-b * x), sigma);
##
##   a ~ normal(0, 10);
##   b ~ normal(0, 10);
##   sigma ~ normal(0, 3);
## }
##
## generated quantities {
##   vector[N] y_rep;
##   vector[N] y_fit;
##   for (n in 1:N) {
##     y_fit[n] = a * exp(-b * x[n]);
##     y_rep[n] = normal_rng(a * exp(-b * x[n]), sigma);
##   }
## }
```

7.1 模型的预测能力

模型推断的好不好呢？是否捕获到数据的特征了呢？我们可以看下模型的预测能力。



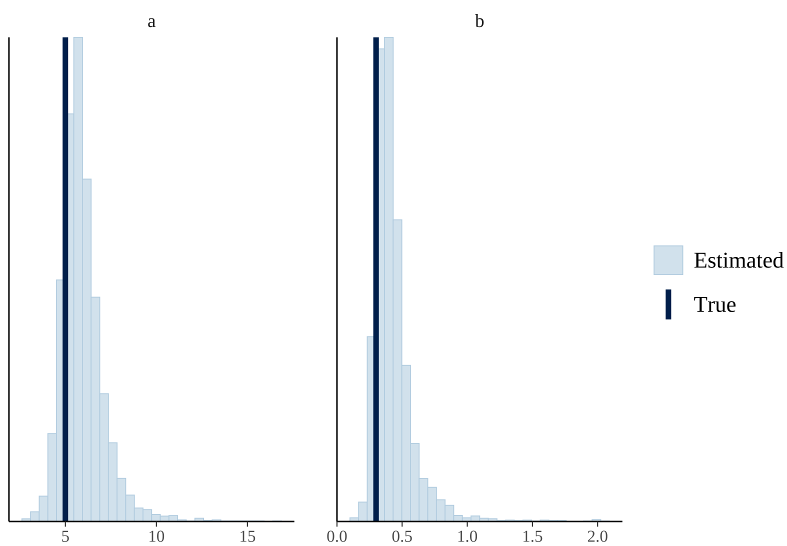
一般来说，模型是一种探测手段，用于探测数据的产生机制。我们推断出参数的分布，就可以从后验分布中随机抽取重复样本集。如果一个贝叶斯模型是“好”的，那么从它模拟产生的数据应该与实际观察到的数据很类似。

7.2 模型对参数的恢复

事实上，数据是我模拟的，真实值 $a = 5, b = 0.3$，模型给出的参数估计是

variable	mean	median	sd	mad	q5	q95	rhat	ess_bulk	ess_tail
a	5.939	5.746	1.207	0.870	4.445	7.950	1.002	1093.555	776.777
b	0.432	0.397	0.165	0.096	0.270	0.726	1.002	1013.420	740.488

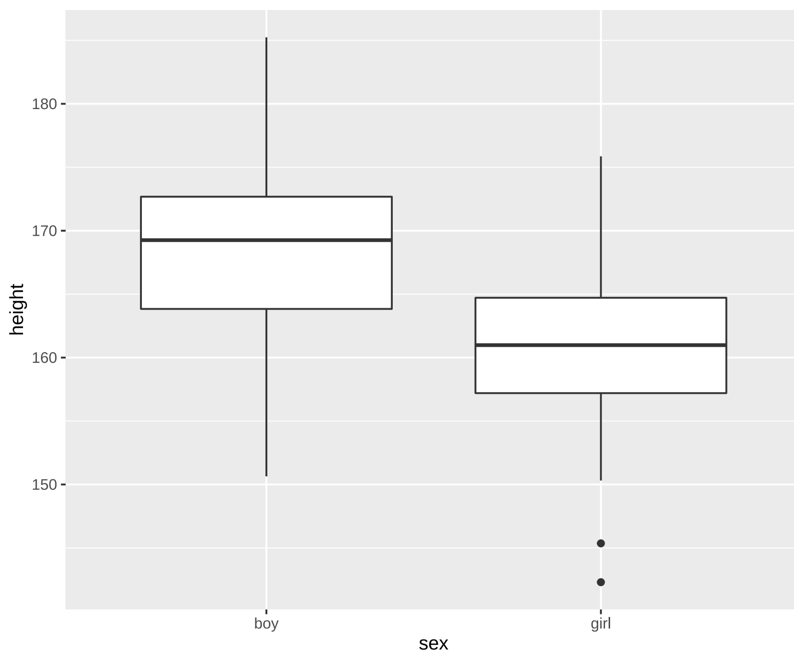
模型捕获和还原了参数



8 假设检验

直播的时候最后一个女生提问，我没听清楚，所以答非所问了，抱歉抱歉。事实上，假设检验的本质是线性回归，所以贝叶斯也能做假设检验。这里，我们仍然以身高为例，检验男生的身高是否明显高于女生身高？数据中用 1 代表女孩，2 代表男孩

id	sex	height	gender
1	boy	173.72	2
2	boy	170.89	2
3	boy	182.11	2
4	boy	176.21	2
5	boy	167.08	2
6	boy	183.12	2



8.1 Stan 代码

假定男孩和女孩的身高服从**各自的**正态分布

$$
\begin{aligned}
\text{height}_i &\sim \text{normal}(\mu_{j[i]}, \sigma_{j[i]}) \\
\mu_j &\sim \text{normal}(\text{mean\_height}, 2) \\
\sigma_j &\sim \text{cauchy}(0, 1)
\end{aligned}
$$

```
##
## data {
##   int<lower=1> N;
```

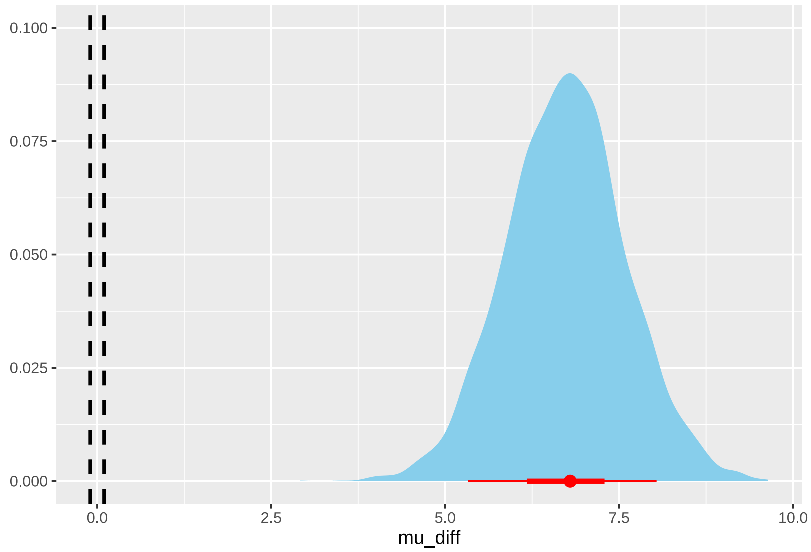
```
##   int<lower=2> n_groups;
##   vector[N] y;
##   int<lower=1, upper=n_groups> group_id[N];
## }
## transformed data {
##   real mean_y;
##   mean_y = mean(y);
## }
## parameters {
##   vector[2] mu;                    // Estimated group means for each
##   vector<lower=0>[2] sigma;        // Estimated group sd for each
## }
## model {
##   mu ~ normal(mean_y, 2);
##   sigma ~ cauchy(0, 1);
##   for (n in 1:N){
##     y[n] ~ normal(mu[group_id[n]], sigma[group_id[n]]);
##   }
## }
## generated quantities {
##   real mu_diff;
##   real cohen_d;
##   real cles;
##   mu_diff = mu[2] - mu[1];
##   cohen_d = mu_diff / sqrt(sum(sigma)/2);
##   cles = normal_cdf(mu_diff / sqrt(sum(sigma)), 0, 1);
## }
```

8.2 等效检验

我们一般会采用实用等效区间 region of practical equivalence ROPE。实用等效区间，就是我们感兴趣值附近的一个区间，比如这里的均值差。频率学中的零假设是看均值差是否为 0，贝叶斯则是看均值差有多少落入 0 附近的区间。具体方法就是，先算出后验分布的高密度区间，然后看这个高密度区间落在 [-0.1, 0.1] 的比例。

在做假设检验的时候，我们内心是期望，后验概率的**高密度区间**落在**实际等效区间**的比例越小越小，如果小于 2.5%，我们就可以拒绝零假设了；如果大于 97.5%，我们就接受零假设。

CI	ROPE_low	ROPE_high	ROPE_Percentage
0.89	-0.7306188	0.7306188	0



9 如何开始

9.1 配置环境

- 第 1 步，安装 R
- 第 2 步，安装 Rstudio
- 第 3 步，安装 Rtools42 到 `C:\rtools42`，(苹果系统不需要这一步)
- 第 4 步，安装 CmdStanR

9.2 参考书籍

- https://mc-stan.org/
- https://discourse.mc-stan.org/
- Gelman, Andrew, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2013. *Bayesian Data Analysis*, Third Edition. Boca Raton: Chapman; Hall/CRC.
- Kruschke, John K. 2014. *Doing Bayesian Data Analysis: A Tutorial Introduction with R*. 2nd Edition. Burlington, MA: Academic Press.
- McElreath, Richard. 2020. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. 2nd ed. CRC Texts in Statistical Science. Boca Raton: Taylor; Francis, CRC Press.

10 感谢

感谢 Stan 语言之美!