贝叶斯新统计与 Stan

Bayesian Data Analysis using Stan

王敏杰

38552109@qq.com

四川师范大学

本节课的目的

内容:

- 什么是 Stan
- 为什么学 Stan
    - 案例
- 如何开始

准备:

- 需要一点点的 R 或者 python 知识
- 课件下载 https://github.com/perlatex/why_stan

Stan 是什么？

Stan 是什么？

- Stan 是一门统计编程语言，主要用于贝叶斯推断
- Stan 广泛应用于社会学、生物、物理、工程和商业等领域

Stan 的历史

Stan 名字的由来

- 波兰犹太裔核物理学家 Stanislaw Ulam，在研究核武器时，发明了蒙特卡罗方法
- 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
- 贝叶斯界用这种蒙特卡罗方法开发一套程序，并用它创始人的名字 Stan 命名

Stan 开发团队

- 这套程序是由纽约哥伦比亚大学 Andrew Gelman 发起，在核心开发团队的共同努力下完成

Stan 如何工作

- Stan 首先会把 Stan 代码翻译成 C++，然后在本地编译
- Stan 拥有能支持自动差分的矩阵和数学库包
- 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
    - 在 python 用 PyStan
- **Stan 可以当作你已经掌握的数据分析工具的一种插件、一种扩展和增强**。

为什么学 Stan

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

Stan 的优势

对我们学术研究有什么好处？

- 革新统计方法
- 拓展研究视角

案例

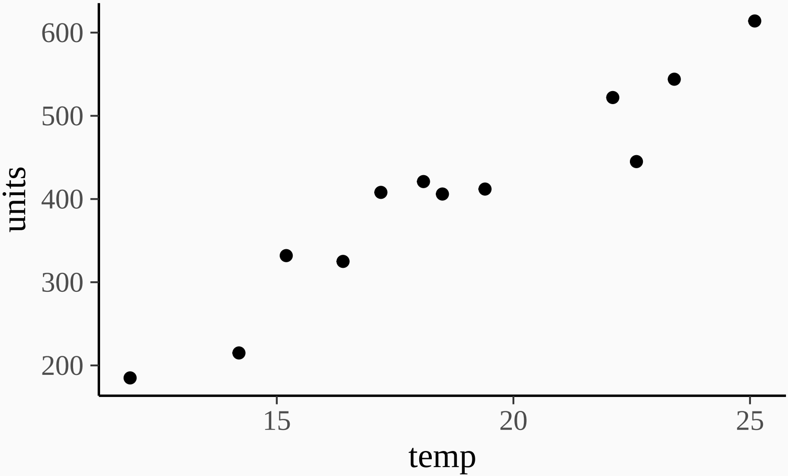
案例

数据是不同天气温度冰淇淋销量

temp	units
11.9	185
14.2	215
15.2	332
16.4	325
17.2	408
18.1	421
18.5	406

我们想估计气温与销量之间的关系

冰淇淋销量与天气温度



传统的方法

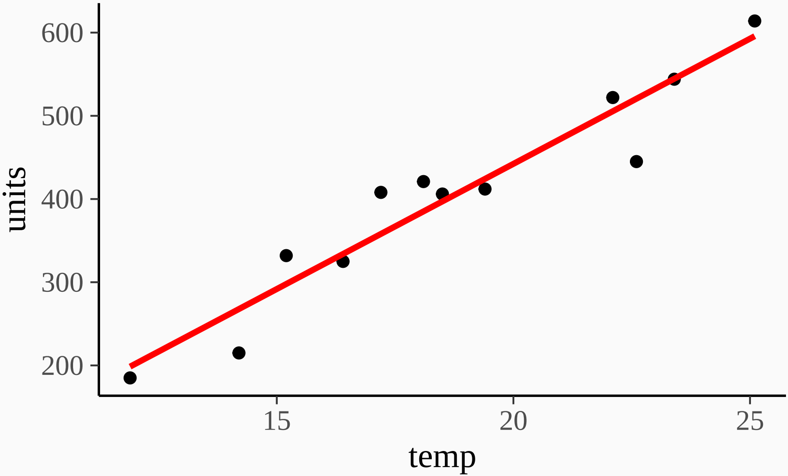
在 R 语言中使用 `lm()`

```
lm(units ~ 1 + temp, data = icecream)
```

传统的方法

```r
fit_lm <- lm(units ~ 1 + temp, data = icecream)
summary(fit_lm)
##
## Call:
## lm(formula = units ~ 1 + temp, data = icecream)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.51 -12.57   4.13  22.24  49.96
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -159.47      54.64   -2.92    0.015 *
## temp           30.09       2.87   10.50    1e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.1 on 10 degrees of freedom
## Multiple R-squared:  0.917,  Adjusted R-squared:  0.909
## F-statistic:  110 on 1 and 10 DF,  p-value: 1.02e-06
```

传统的方法



传统的方法

```
lm(units ~ 1 + temp, data = icecream)
```

但是，我们不满意。不满意在于

- 模型的假设？
- 模型的参数？
- 模型的解释？

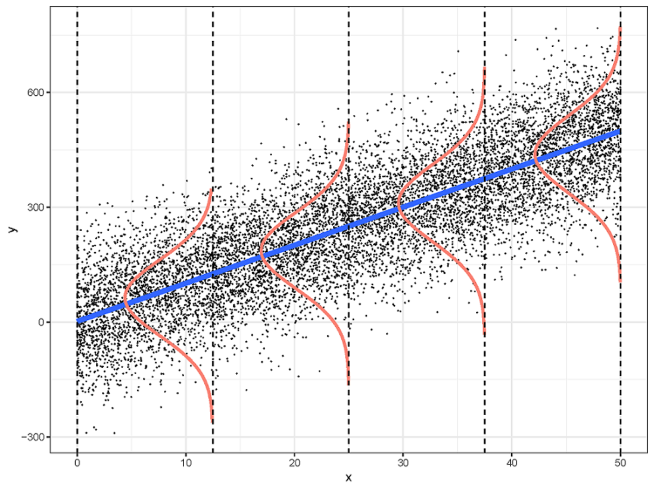
贝叶斯新统计

线性回归需要满足四个前提假设：

1. **Linearity**
   - 因变量和每个自变量都是线性关系
2. **Indpendence**
   - 对于所有的观测值，它们的误差项相互之间是独立的
3. **Normality**
   - 误差项服从正态分布
4. **Equal-variance**
   - 所有的误差项具有同样方差

这四个假设的首字母，合起来就是 **LINE**，这样很好记

把这四个前提画在一张图中



数学表达式

线性模型

$$y_n = \alpha + \beta x_n + \epsilon_n \quad \text{where} \quad \epsilon_n \sim \text{normal}(0, \sigma).$$

等价于

$$y_n - (\alpha + \beta X_n) \sim \text{normal}(0, \sigma),$$

进一步等价

$$y_n \sim \text{normal}(\alpha + \beta X_n, \ \sigma).$$

数学表达式

我**强烈推荐**这样写线性模型的数学表达式

$$
\begin{align}
y_n &\sim \text{normal}(\mu_n,\ \sigma) \tag{1} \\
\mu_n &= \alpha + \beta x_n \tag{2}
\end{align}
$$

因为，这种写法可以很方便地过渡到其它模型。（后面会看到）

Stan 代码框架

```
data{
        // 导入数据
}
parameters{
        // 定义模型要估计的参数
}
model{
        // 后验概率函数
}
```

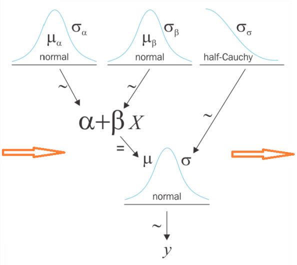
从模型到 Stan 代码

模型

$$y_n \sim \text{normal}(\mu_n,\ \sigma)$$
$$\mu_n = \alpha + \beta x_n$$

$$\alpha \sim \text{normal}(0, 4)$$
$$\beta \sim \text{normal}(0, 4)$$
$$\sigma \sim \text{half-Cauchy}(1)$$

图示



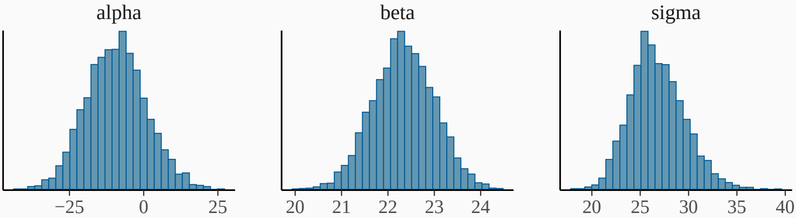
Stan代码

```
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, 4);
  beta   ~ normal(0, 4);
  sigma  ~ cauchy(0, 1);
}
```

normal models

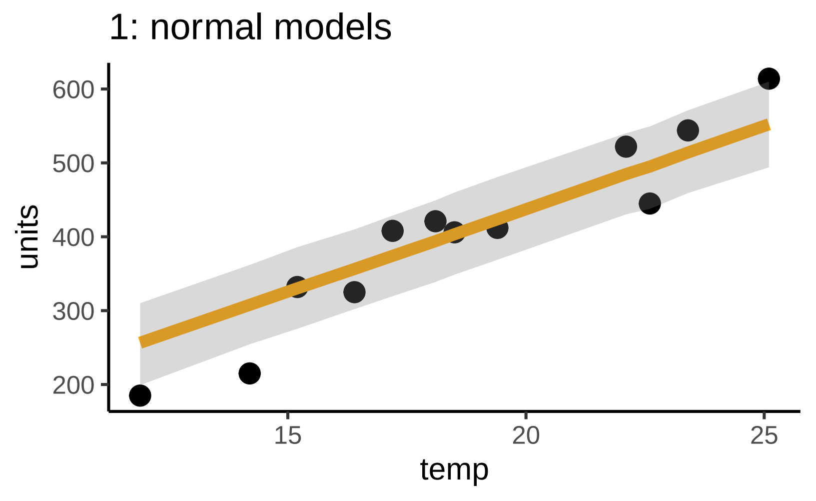
```
data {
  int<lower=0> N;
  vector[N] y;
  vector[N] x;
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}
model {
  y ~ normal(alpha + beta * x, sigma);

  alpha  ~ normal(0, 4);
  beta   ~ normal(0, 4);
  sigma  ~ cauchy(0, 1);
}
```

normal models



	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%
alpha	-9.23	0.238	10.072	-28.3	-16.2	-9.3	-2.53	11.1
beta	22.34	0.016	0.659	21.0	21.9	22.3	22.79	23.6
sigma	26.77	0.060	2.782	21.9	24.9	26.5	28.56	32.7

normal models



log normal models

有时候，我们对响应变量做 log 转化，

$$
\begin{aligned}
\log(y_n) &\sim \text{normal}(\mu_n,\ \sigma) \qquad (3)\\
\mu_n &= \alpha + \beta x_n \qquad (4)
\end{aligned}
$$

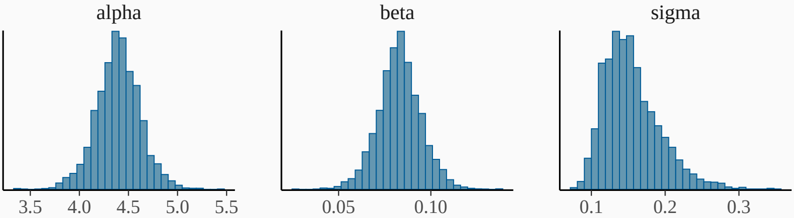
等价于

$$
\begin{aligned}
y_n &\sim \text{Lognormal}(\mu_n,\ \sigma) \qquad (5)\\
\mu_n &= \alpha + \beta x_n \qquad (6)
\end{aligned}
$$

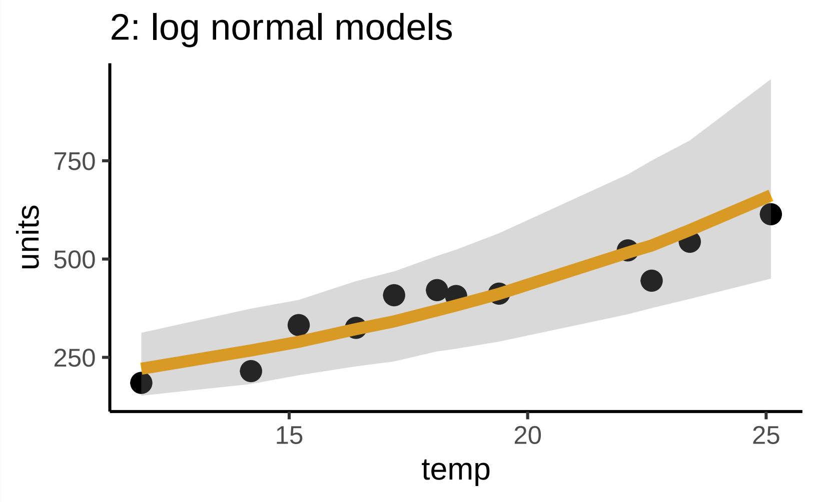
log normal models

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

  alpha  ~ normal(0, 10);
  beta   ~ normal(0, 10);
  sigma  ~ exponential(1);
}
```

log normal models



	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%
alpha	4.395	0.006	0.224	3.940	4.255	4.396	4.540	4.843
beta	0.083	0.000	0.012	0.060	0.076	0.083	0.090	0.107
sigma	0.155	0.001	0.038	0.098	0.128	0.149	0.176	0.248


2: log normal models
750
500
250
units
15
20
25
temp

Poisson Models

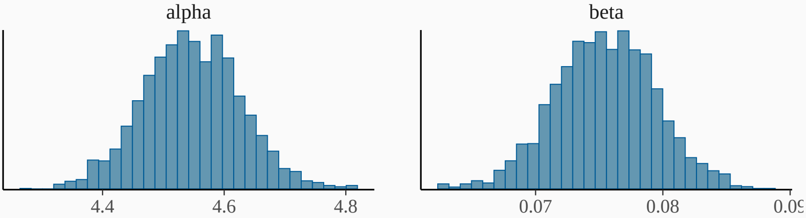
冰激凌销量，是一种**计数类型**的变量，因此可以用泊松回归分析

$$
\begin{aligned}
y_n &\sim \mathrm{Poisson}(\lambda_n) &(7)\\
\log(\lambda_n) &= \alpha + \beta x_n &(8)
\end{aligned}
$$

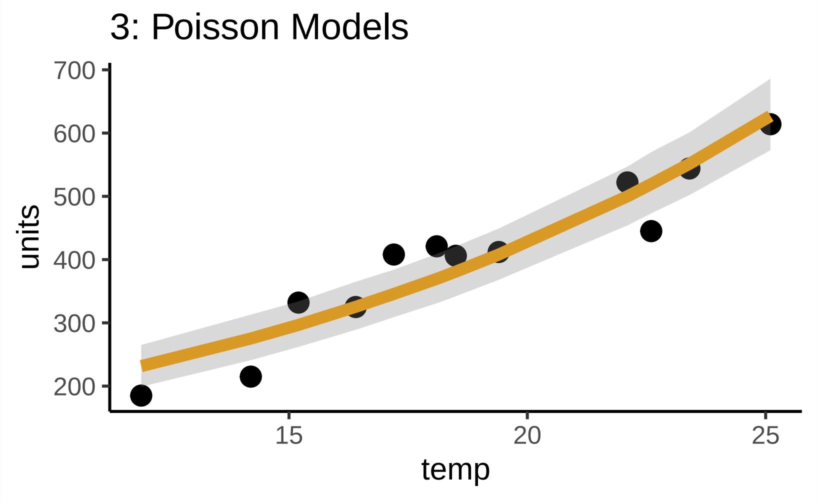
Poisson Models

```
data {
  int N;
  int<lower=0> y[N];
  vector[N] x;
}
parameters {
  real alpha;
  real beta;
}
model {
  y ~ poisson(alpha + beta * x);

  alpha  ~ normal(0, 10);
  beta   ~ normal(0, 10);
}
```

Poisson Models



	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%
alpha	4.547	0.003	0.080	4.389	4.491	4.545	4.601	4.706
beta	0.075	0.000	0.004	0.068	0.073	0.075	0.078	0.083


3: Poisson Models
700
600
500
400
300
200
units
15
20
25
temp

泊松分布可看成是二项分布的极限，因此可以使用更灵活的二项式回归模型

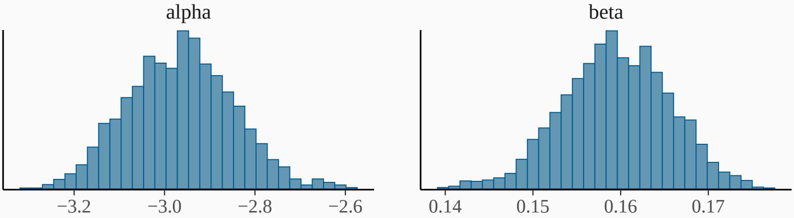
$$y_n \sim \text{binomial}(N, \theta_n) \tag{9}$$

$$\text{logit}(\theta_n) = \log\left(\frac{\theta_n}{1-\theta_n}\right) = \alpha + \beta x_n \tag{10}$$

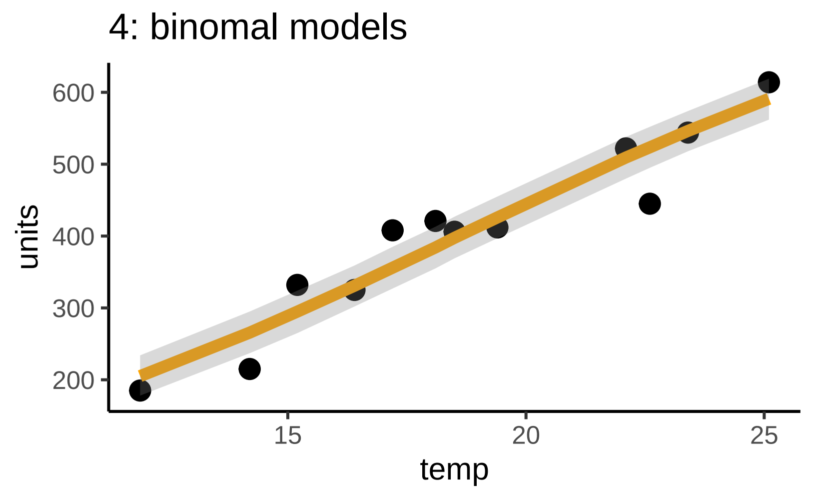
binomal models

```
data {
  int<lower=1> N;
  int<lower=1> trials;
  vector[N] x;
  int y[N];
  real new_x;
}
parameters {
  real alpha;
  real beta;
}
model {
  y ~ binomial_logit(trials, alpha + beta * x);

  alpha  ~ cauchy(0, 5);
  beta   ~ normal(0, 5);
}
```

binomal models



	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%
alpha	-2.964	0.005	0.116	-3.182	-3.045	-2.962	-2.887	-2.734
beta	0.159	0.000	0.006	0.147	0.155	0.159	0.164	0.171


4: binomal models
units
600
500
400
300
200
15
20
25
temp

模型比较

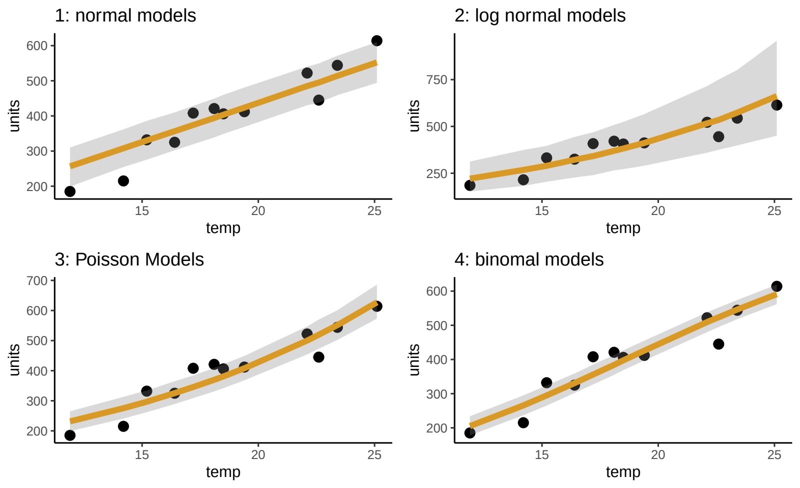
衡量预测准确性，可以用 loo 宏包**比较模型**

```
loo_compare(loo_normal,
            loo_lognormal,
            loo_poisson,
            loo_binomial)
##        elpd_diff se_diff
## model2   0.0       0.0
## model1  -5.8       7.4
## model3 -17.8      10.2
## model4 -28.1      19.4
```

结果显示：

- 第二个模型 `lognormal` 相对最优
- 这个负数，代表该模型比最优秀的模型，预测能力差了多少

模型可视化



1: normal models
2: log normal models
3: Poisson Models
4: binomal models
units
temp
600
500
400
300
200
750
250
700
15
20
25

更多

Stan 可以做更多:

- 假设检验
- 线性模型
- 广义线性模型
- 多层模型
- 混合模型
- 高斯过程
- 时间序列
- 机器学习
- 常微分方程

如何开始

配置环境

- 第 1 步，安装R
- 第 2 步，安装Rstudio

配置环境

- 第 3 步，安装Rtools4.0到 C 盘
- 第 4 步，添加系统路径 (电脑 - 属性 - 高级系统设置 - 环境变量 - 系统变量 - Path)
  - `C:\rtools40`
  - `C:\rtools40\mingw64\bin`
  - `C:\rtools40\usr\bin`
- 第 5 步，配置

```
writeLines('PATH="${RTOOLS40_HOME}\\usr\\bin;${PATH}"', con = "~/.Renviron")
```

配置环境

- 第 6 步，安装 rstan 宏包

```
install.packages(c("StanHeaders", "rstan"), type = "source")
install.packages(c("tidyverse", "tidybayes", "bayesplot"))
```

- 第 7 步，遇到问题，请参考

https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started

欢迎来到新世界

https://mc-stan.org/

欢迎关注选课信息

课程	内容	时间
《数据科学中的 R 语言 (上)》	R, tidyverse	上学期
《数据科学中的 R 语言 (下)》	Stan 概率编程	下学期

王敏杰

38552109@qq.com