统计学与 R 语言

第 16 讲　最小二乘估计与最大似然估计

张敬信

2022 年 4 月 25 日

哈尔滨商业大学

一. 最小二乘估计 (OLS)

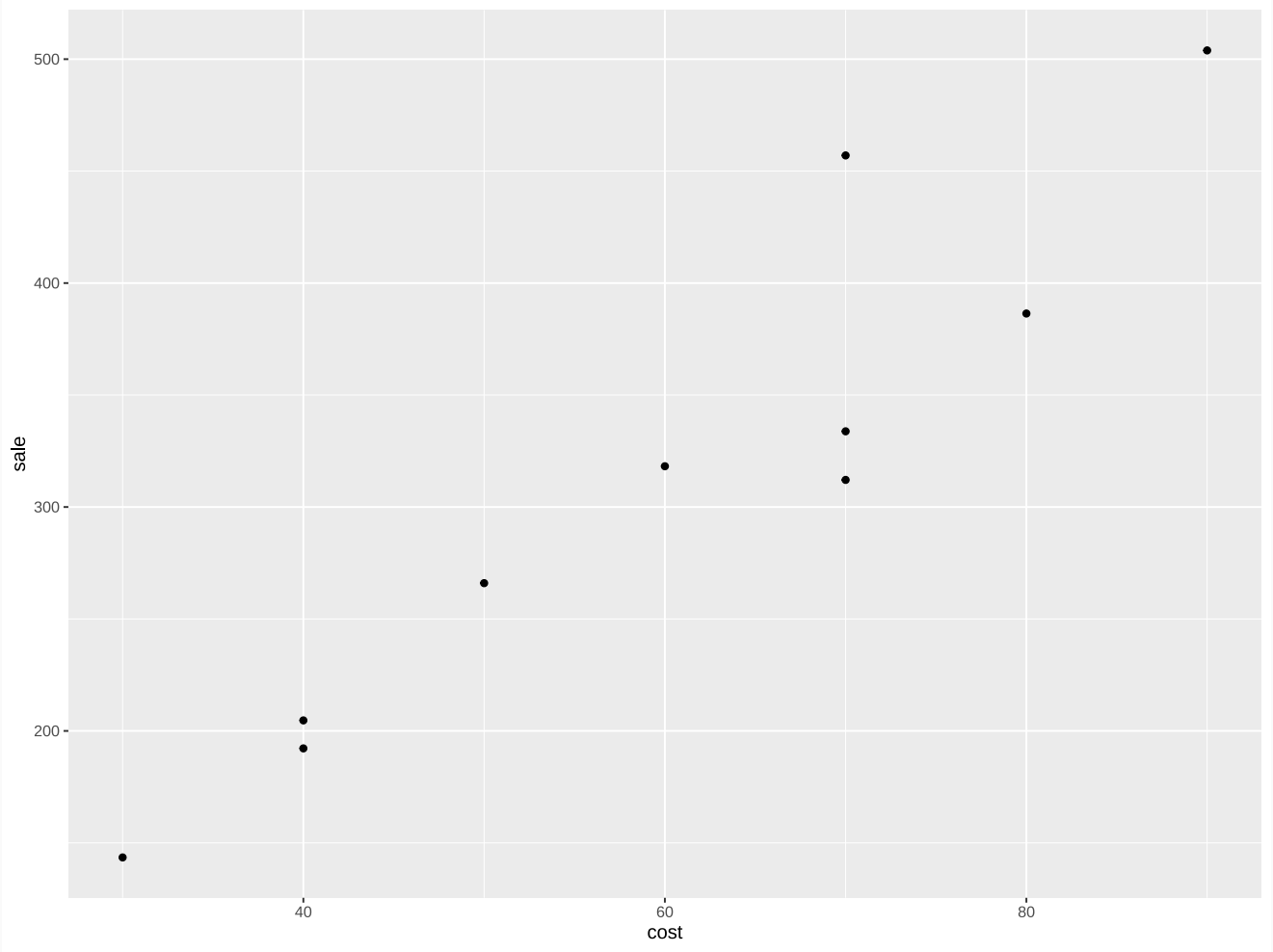
最小二乘估计（Ordinary Least Squares），常用于估计线性回归、曲线拟合的参数，其思想是让实际值与模型预测值的总偏离达到最小，从而得到最优的模型参数估计值。

用一元线性回归来阐述，比如有 10 组广告费用与销售额的数据：

```
sales = tibble(
  cost = c(30,40,40,50,60,70,70,70,80,90),
  sale = c(143.5,192.2,204.7,266,318.2,457,333.8,312.1,
           386.4,503.9))
```

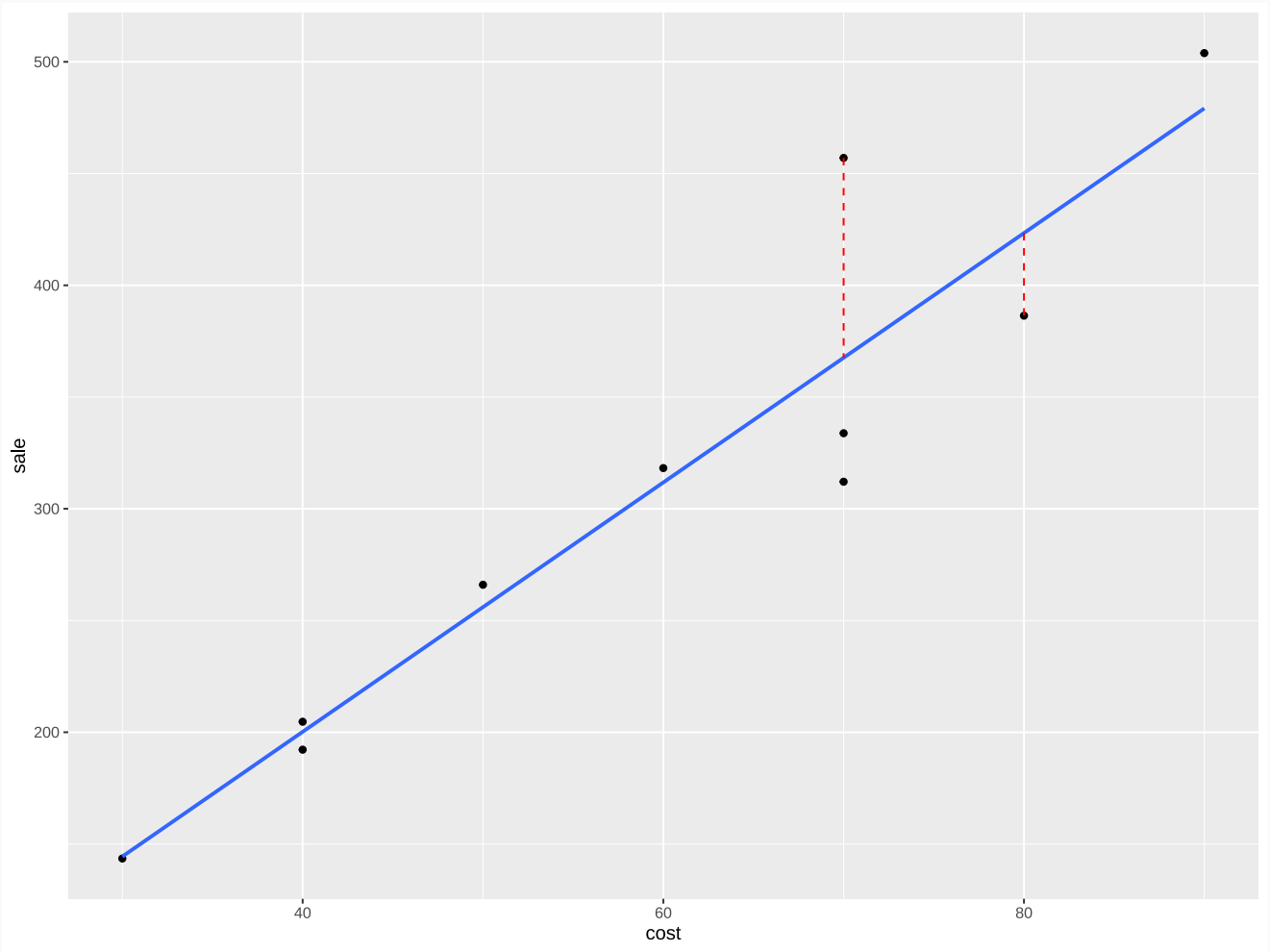
- 绘制散点图：

```
ggplot(sales, aes(cost, sale)) + geom_point()
```



这些散点大致在一条直线上，一元线性回归就是寻找一条直线，使得与这些散点拟合程度最好（越接近直线越好）：

```
m = lm(sale ~ cost, sales)
sales1 = sales[c(6,9),] %>%
  mutate(p = predict(m, .))
ggplot(sales, aes(cost, sale)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  geom_segment(aes(x = cost, y = sale,
                   xend = cost, yend = p),
               data = sales1, linetype = 2, color = "red")
```


500
400
300
200
sale
40
60
80
cost

这样一条直线（一元线性回归模型）方程可写为：

$$y = \beta_0 + \beta_1 x$$

其中，β_0, β_1 为待定参数，目标是找到样本点最接近的直线对应的 β_0, β_1.

- **怎么刻画这种 “最接近”?**

$\hat{y}_i = \beta_0 + \beta_1 x_i$ 是与横轴 x_i 对应的直线上的点的纵坐标（模型预测值），它与样本点 x_i 对应的真实值 y_i 之差，就是预测误差（红虚线长度）：

$$\varepsilon_i = |y_i - \hat{y}_i|, \quad i = 1, \cdots, n$$

适合描述散点到直线的 “接近程度”。

但绝对值不容易计算，改用：

$$\varepsilon_i^2 = (y_i - \hat{y}_i)^2, \quad i = 1, \cdots, n$$

要让所有散点总体上最接近该直线，就是让总的预测误差：

$$J(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} [y_i - (\beta_0 + \beta_1 x_i)]^2$$

最小。

于是，问题转化为优化问题：

$$\arg\min_{\beta_0,\beta_1} J(\beta_0,\beta_1) = \sum_{i=1}^{n}[y_i - (\beta_0 + \beta_1 x_i)]^2$$

其中，$\arg\min$ 意思是求右式达到最小时所对应的参数 β_0, β_1. 这就是 “最小二乘法”，有着很直观的几何解释。

这是求二元函数极小值问题。根据微积分知识，二元函数极值是在一阶偏导等于 0 点处取到：

$$\begin{cases} \dfrac{\partial J}{\partial \beta_0} = -2\sum\limits_{i=1}^{n}[y_i - \beta_0 - \beta_1 x_i] = 0 \\ \dfrac{\partial J}{\partial \beta_1} = -2\sum\limits_{i=1}^{n}[y_i - \beta_0 - \beta_1 x_i]x_i = 0 \end{cases}$$

解关于 β_0, β_1 的二元一次方程组得

$$\begin{cases} \beta_0 = \bar{y} - \beta_1 \bar{x} \\ \beta_1 = \dfrac{\sum\limits_{i=1}^{n} x_i y_i - \bar{y} \sum\limits_{i=1}^{n} x_i}{\sum\limits_{i=1}^{n} x_i^2 - \bar{x} \sum\limits_{i=1}^{n} x_i} = \dfrac{\sum\limits_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum\limits_{i=1}^{n} (x_i - \bar{x})^2} \end{cases}$$

其中，

$$\begin{cases} \bar{x} = \dfrac{1}{n} \sum\limits_{i=1}^{n} x_i \\ \bar{y} = \dfrac{1}{n} \sum\limits_{i=1}^{n} y_i \end{cases}$$

更一般地，多元线性回归、非线性回归（拟合）的最小二乘法估计待定参数，也是类似的，只需要将线性预测值改成模型预测值：

$$\arg\min_{\beta} J(\beta) = \sum_{i=1}^{n} [y_i - f(x_i, \beta)]^2$$

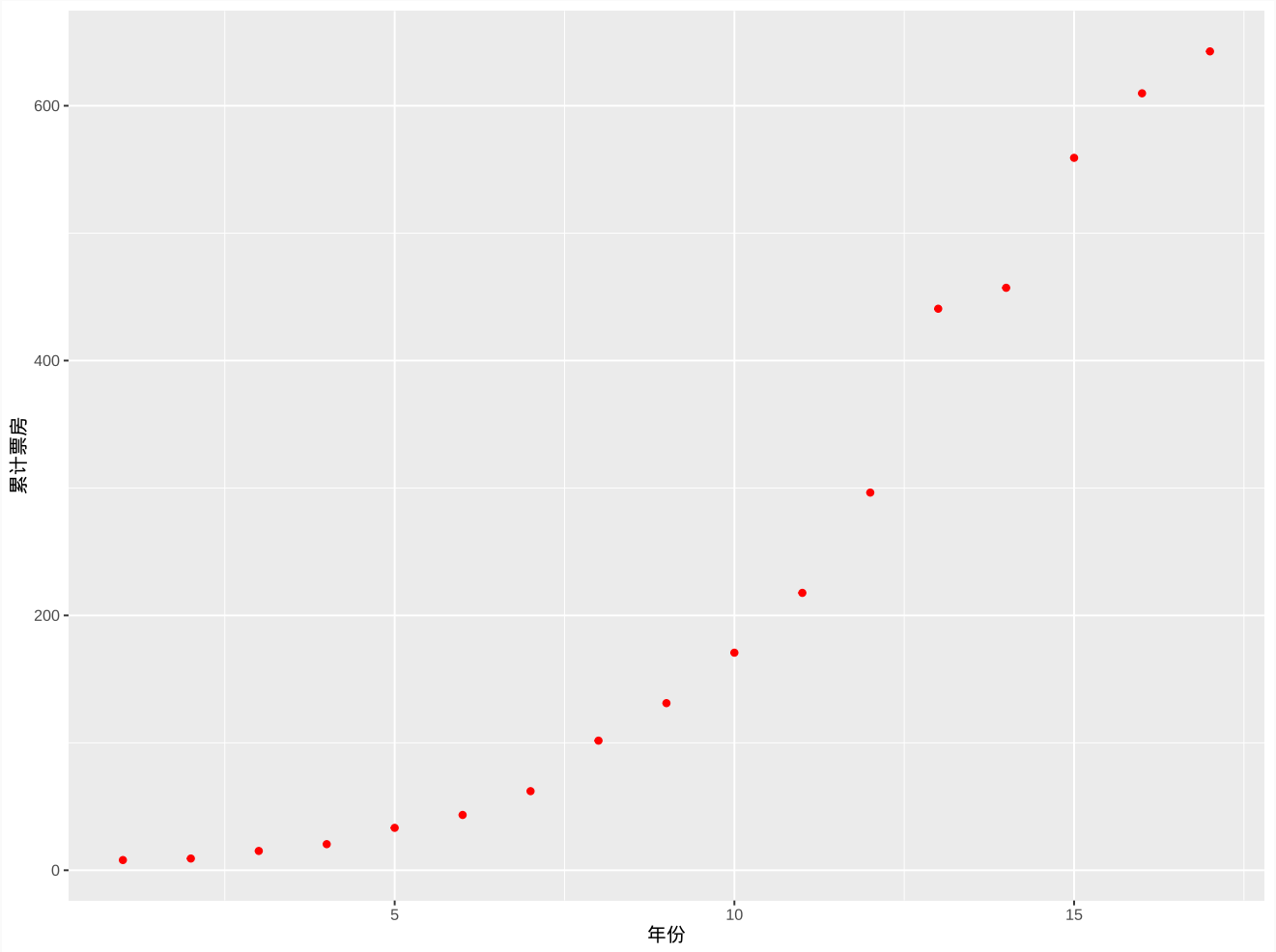
线性回归的最小二乘估计可用 `lm()` 函数实现，非线性回归的最小二乘估计可用 `nls()` 函数实现。

案例：非线性拟合

- 现有我国 2003-2019 年历年累计电影票房数据：

```
df = readxl::read_xlsx("datas/历年累计票房.xlsx") %>%
  mutate(年份 = 年份 - 2002)
df
#> # A tibble: 17 x 2
#>    年份 累计票房
#>   <dbl>    <dbl>
#> 1     1      8
#> 2     2      9.2
#> 3     3     15.1
#> # ... with 14 more rows
```

```
p = ggplot(df, aes(年份, 累计票房)) +
  geom_point(color = "red", size = 1.5)
p
```



非线性回归第一步是找到合适的模型函数。这些散点大致服从 Logistic 分布曲线：

$$N(t) = \frac{\varphi_1}{1 + e^{-(\varphi_2 + \varphi_3 t)}}$$

我们想用 nls() 做非线性拟合，寻找最优的参数值：$\varphi_1, \varphi_2, \varphi_3$.

非线性拟合的算法非常依赖于参数初始值的选取，选取适当（离估计值不远）很快就能收敛到最优估计，否则迭代很可能无法收敛。

参数 φ_1 对应人口容纳量上限，大致为曲线拐点值（目测约为 400）的 2 倍. 一旦确定了 φ_1, 则

$$\text{logit}\left(\frac{N(t)}{\varphi_1}\right) = \varphi_2 + \varphi_3 t$$

其中，$\text{logit}(p) = \ln \frac{p}{1-p}$ 称为 Logit 变换。

于是，用 lm() 做线性回归即可得到 φ_2, φ_3 的估计值。

```
lm.fit = lm(car::logit(累计票房 / 800) ~ 年份, df)
coef(lm.fit)
#> (Intercept)          年份
#>      -5.145         0.391
```

这样就得到了一组较好的参数初始值：

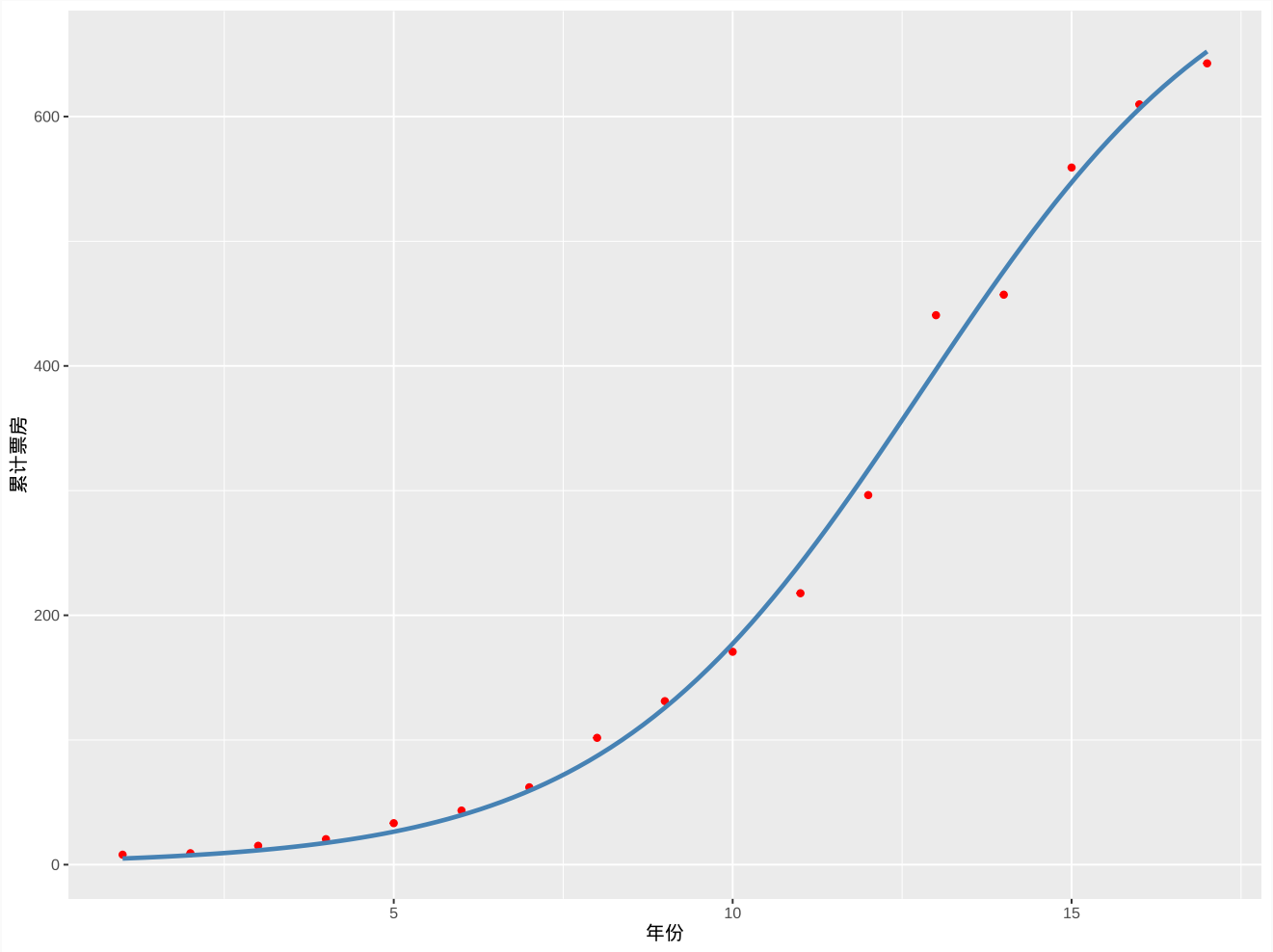
$$\varphi_1 = 800,\ \varphi_2 = -5.14,\ \varphi_3 = 0.39$$

接着，用 nls() 做非线性拟合，需要提供模型公式和初始参数值：

```
log.fit = nls(累计票房 ~ phi1 / (1+exp(-(phi2+phi3*年份)))),
              data = df,
              start = list(phi1=800, phi2=-5.14, phi3=0.39))
coefs = coef(log.fit)
coefs
#>    phi1    phi2    phi3
#> 760.577  -5.457   0.427
```

绘图看拟合效果，同时这也是已知函数表达式，绘制 ggplot 图形的方法：

```
LogFit = function(x) coefs[1] / (1+exp(-(coefs[2]+coefs[3]*x)))
p + geom_function(fun=LogFit, color="steelblue", size=1.2)
```



注： `nls()` 拟合依赖于初始值和 `selfstart` 设置，容易拟合失败，若失败可以用 `glsnls` 包。

二. 最大似然估计 (MLE)

最大似然估计 (Maximum Likelihood Estimation) 是频率学派常用的方法，其思想是既然抽取到现在的样本数据，那么最优的模型参数应选择这样的值：让这些样本数据最有可能出现！

比如，你和猎人同时开枪，结果是猎物被击中。用最大似然估计：猎物是猎人打中的，而不是你打中的！因为猎人打中的概率比你大。

假设数据 $x_1, x_2, \cdots, x_n$ 是独立同分布的一组抽样，记 $\mathbf{x} = (x_1, x_2, \cdots, x_n)$, 则最大似然法估计参数 θ, 可推导如下：

$$
\begin{aligned}
\hat{\theta}_{\mathrm{MLE}} &= \arg\max P(\mathbf{x} \mid \theta) \\
&= \arg\max P\left(x_1 \mid \theta\right) P\left(x_2 \mid \theta\right) \cdots P\left(x_n \mid \theta\right) \\
&= \arg\max \ln \prod_{i=1}^{n} P\left(x_i \mid \theta\right) \\
&= \arg\max \sum_{i=1}^{n} \ln P\left(x_i \mid \theta\right)
\end{aligned}
$$

其中，第 1 行到第 2 行是由于独立同分布；第 2 行到第 3 行是由于 $ln(\cdot)$ 单调递增，故做对数变换不影响求最大参数值。

最后要优化的函数记为

$$\mathcal{L}(\theta \mid X) = \sum_{i=1}^{n} \ln P\left(x_i \mid \theta\right)$$

称为对数似然函数，其中，$P(x_i \mid \theta)$ 为给定的 θ 下出现 x_i 的概率（离散情形是概率，连续情形是概率密度)。

于是，最大似然估计的一般步骤：先推导出对数似然函数，再做最大化寻优即可。后一步可用自带的 `optimize()` 函数，或者 `maxLik` 包中的 `maxLik()` 函数来实现。

案例：离散情形，估计伯努利分布参数

例如，已发生事件是：抛 10 次硬币，出现 3 次正面，用最大似然法估计参数 $p = P(\text{正})$.

抛硬币服从伯努利分布，该事件发生的概率（似然函数）可表示为：

$$P(\mathrm{x} \mid p) = C_{10}^{3}\, p^{3}(1-p)^{7}$$

从而，对数似然函数为

$$\mathcal{L}(p \mid \mathrm{x}) = \ln C_{10}^{3} + 3\ln p + 7\ln(1-p)$$

注意第一项是常数，不妨忽略掉它，不影响优化目标。

用 maxLik 包实现，先定义对数似然函数：

```
loglik = function(p) 3 * log(p) + 7 * log(1-p)
```

再调用 maxLik() 函数，需传递对数似然函数，并提供迭代初始值：

```
library(maxLik)
m = maxLik(loglik, start = 0.5)
coef(m)       # 最优参数估计值
#> [1] 0.3
stdEr(m)       # 估计的标准误
#> [1] 0.145
```

不出所料，最优估计 $\hat{p} = 0.3$, 就是正面出现的频率!

案例：连续情形，估计正态分布参数

离散情形，用的是单个点的概率；连续情形，单个点的概率为 0, 考虑包含点的任意小区间段的概率才有意义，也就是概率密度：

$$f(x_0) = \lim_{\delta \to 0} \frac{P(X \in [x_0 - \delta, x_0 + \delta])}{2\delta}$$

以 `mtcars$mpg` 数据 (`n=32`) 为例，用最大似然法估计正态分布的参数 μ, σ^2. 该数据出现的概率（似然函数）为：

$$\begin{aligned} f(\mathrm{x} \mid \mu, \sigma) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\Big[-\frac{1}{2\sigma^2}(\mathrm{x} - \mu)^2 \Big] \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\Big[-\frac{1}{2\sigma^2}(x_i - \mu)^2 \Big] \\ &= (2\pi)^{-n/2} (\sigma^2)^{-n/2} \exp\Big[-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2 \Big] \end{aligned}$$

从而，对数似然函数为：

$$\mathcal{L}(\mu, \sigma \mid \mathbf{x}) = -\frac{n}{2}\ln(2\pi) - n\ln\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

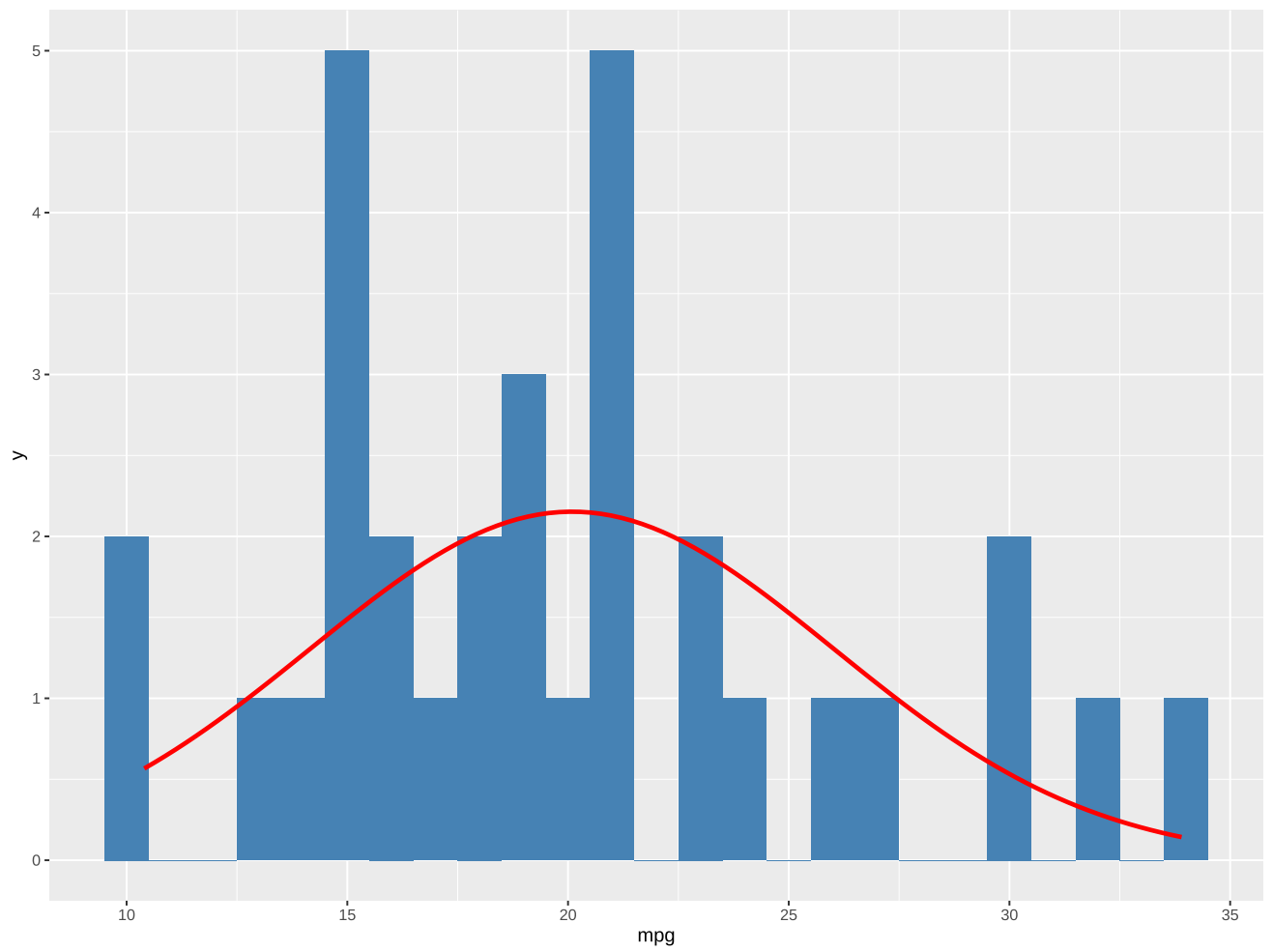
同样忽略掉第一项常数，定义对数似然函数，再调用 `maxLik()` 寻优：

```
loglik = function(theta) {
  mu = theta[1]
  sigma = theta[2]
  n = nrow(mtcars)
  - n*log(sigma) - 1/(2*sigma^2) * sum((mtcars$mpg-mu)^2)
}
```

```r
m = maxLik(loglik, start=c(mu=30, sigma=10))
coef(m)       # 最优参数估计值
#>    mu sigma
#> 20.09  5.93
stdEr(m)      # 估计的标准误
#>    mu sigma
#> 1.049 0.744
```

- 可视化估计的效果：

```r
ggplot(mtcars, aes(mpg)) +
  geom_histogram(binwidth = 1, fill = "steelblue") +
  stat_function(fun = ~ dnorm(.x, mean=20.09, sd=5.93) * 32,
                color = "red", size = 1.2)
```



实际上，正态分布两个参数的最大似然估计分别为：

$$\hat{\mu} = \bar{\mathbf{x}}, \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{\mathbf{x}})^2.$$

案例：线性回归系数的最大似然估计

真实数据中，一组 x 值对应的 y 观测值可以看作是来自真实 y 值的一次抽样，因为 y 值可能受多种因素的影响，故可以假设任意一组 x 值对应的真实 y 值是服从正态分布的随机变量。

想要找到最优的回归系数，根据最大似然估计的思想，最优的回归系数就是让 y 观测值出现的概率最大时所对应的回归系数。

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \qquad i = 1, \cdots, n$$

其中，ε_i 为预测误差，即不能被线性模型刻画的部分。根据线性回归模型假设：ε_i 独立同分布于 $N(0, \sigma^2)$, 否则说明数据不适合用线性回归模型建模。

由正态分布的性质，$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$, 从而在 x_i, β 已知的条件下，y_i 的概率密度为：

$$f(y_i \,|\, x_i, \beta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{[y_i - (\beta_0 + \beta_1 x_i)]^2}{2\sigma^2}\right), \qquad i = 1, \cdots, n$$

从而，所有 y 的观测数据出现的概率（似然函数）为：

$$\begin{aligned} f(\mathbf{y} \,|\, \mathbf{x}, \beta) &= \prod_{i=1}^{n} f(y_i | x_i, \beta) \\ &= \prod_{i=1}^{n} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{[y_i - (\beta_0 + \beta_1 x_i)]^2}{2\sigma^2}\right)\right] \end{aligned}$$

于是，对数似然函数为：

$$\begin{aligned}\mathcal{L}(\beta_0, \beta_1 \mid \mathbf{x}) &= \ln\left(f(\mathbf{y} \mid \mathbf{x}, \beta)\right) \\ &= n \ln\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \sum_{i=1}^{n} \frac{[y_i - (\beta_0 + \beta_1 x_i)]^2}{2\sigma^2}\end{aligned}$$

注意，要做的是选取 β_0, β_1 让上式达到最大，第一项以及第二项中的 $2\sigma^2$ 不起作用。故最大化该对数似然函数，就等价于：

$$\arg\min_{\beta_0, \beta_1} \sum_{i=1}^{n} [y_i - (\beta_0 + \beta_1 x_i)]^2$$

这与最小二乘估计是等价的！

最后注： `maxLik()` 函数在优化时，默认是根据数据计算数值梯度（只适合简单问题），若推导出梯度（甚至是 Hessian 矩阵）的解析式，并提供给相应参数，则估计速度更快、结果更稳定。

本篇主要参阅 (张敬信, 2022), (冯国双, 2018), 以及包文档，模板感谢 (黄湘云, 2021), (谢益辉, 2021).

参考文献

冯国双 (2018). 白话统计. 电子工业出版社, 北京, 1 edition.

张敬信 (2022). *R* 语言编程：基于 *tidyverse*. 人民邮电出版社, 北京.

谢益辉 (2021). *rmarkdown: Dynamic Documents for R*.

黄湘云 (2021). *Github: R-Markdown-Template*.