R 语言编程：基于 tidyverse

第 21 讲　回归分析

张敬信

2022 年 12 月 6 日

哈尔滨商业大学

回归分析（Regression Analysis），是统计学的核心算法，是计量模型和机器学习的最基本算法。

回归分析是确定两个或两个以上变量间相互依赖的定量关系的一种统计分析方法，具体是通过多组自变量和因变量的样本数据，拟合出最佳的函数关系。如果该关系是线性函数关系，就是线性回归。

计量模型和机器学习中的各种回归算法都可以看作是线性回归的扩展，分类算法也可以看作是一种特殊的回归。

回归分析常用于：

- 探索现象/结果的影响因素主要有哪些?
- 影响因素对现象/结果是怎样影响的?
- 预测未来的现象/结果

设 y 为因变量数据，$\mathbf{x}$ 为自变量数据（可以是多维），设二者之间的真实（精确）关系为：

$$y = f(\mathbf{x})$$

这是不可能得到的，所谓回归建模只是试图去找到一种近似的关系来代替它：

$$\hat{f}(\mathbf{x}) \approx f(\mathbf{x})$$

二者之差就是模型的残差：

$$\varepsilon = f(\mathbf{x}) - \hat{f}(\mathbf{x})$$

总是希望把 y 与 $\mathbf{x}$ 的关系都留在模型部分：$\widehat{f}(\mathbf{x})$，让残差部分最好只是白噪声（完全是随机误差，0 均值，微小标准差的正态分布）：

$$\varepsilon \sim N(0,\, \sigma_{\varepsilon}^{2})$$

这说明建模成功；否则，就是模型尚未提取出充分的模型关系（欠拟合）。

构建的模型关系 $\widehat{f}(\mathbf{x})$，可以是简单的线性关系（线性回归）、也可以是复杂的”黑箱”模型（神经网络、支持向量机等），尽管无法得到精确的表达式，但仍可以用于预测。

回归建模的基本原则是：在没有显著差异的情况下，优先选择更简单的模型。简单模型已足够充分建模，非要用更复杂的模型则会适得其反（过拟合），会降低模型的泛化（预测）能力。

一. 一元线性回归

只对一个自变量与因变量之间的线性关系建模，其基本形式为：

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

一元线性回归的全部模型预测值可表示为：

$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}_{n\times 1} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}_{n\times 2} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}_{2\times 1}$$

记

$$\hat{Y} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}_{n\times 1}, \qquad X = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}_{n\times 2}, \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}_{2\times 1}$$

则矩阵形式表示为

$$\hat{Y} = X\beta$$

于是，让总的预测误差最小的”最小二乘法”优化问题就表示为

$$\arg\min_{\beta} J(\beta) = \|Y - \hat{Y}\|^2 = \|Y - X\beta\|^2$$

其中，$\|\cdot\|$ 为向量的范数（长度）。同样地，$J(\beta)$ 的极小值，在其一阶偏导值等于 0 处取到，按矩阵求导法则计算，可得 $2X^TX\beta - 2X^TY = 0.$

若 X 满秩，则 X^TX 可逆，从而

$$\beta = (X^TX)^{-1}X^TY$$

二. 多元线性回归

推广到多元线性回归模型，可对多个自变量与因变量之间的线性关系建模：

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m$$

多元线性回归是找一个超平面，到各个散点的距离总和最小：

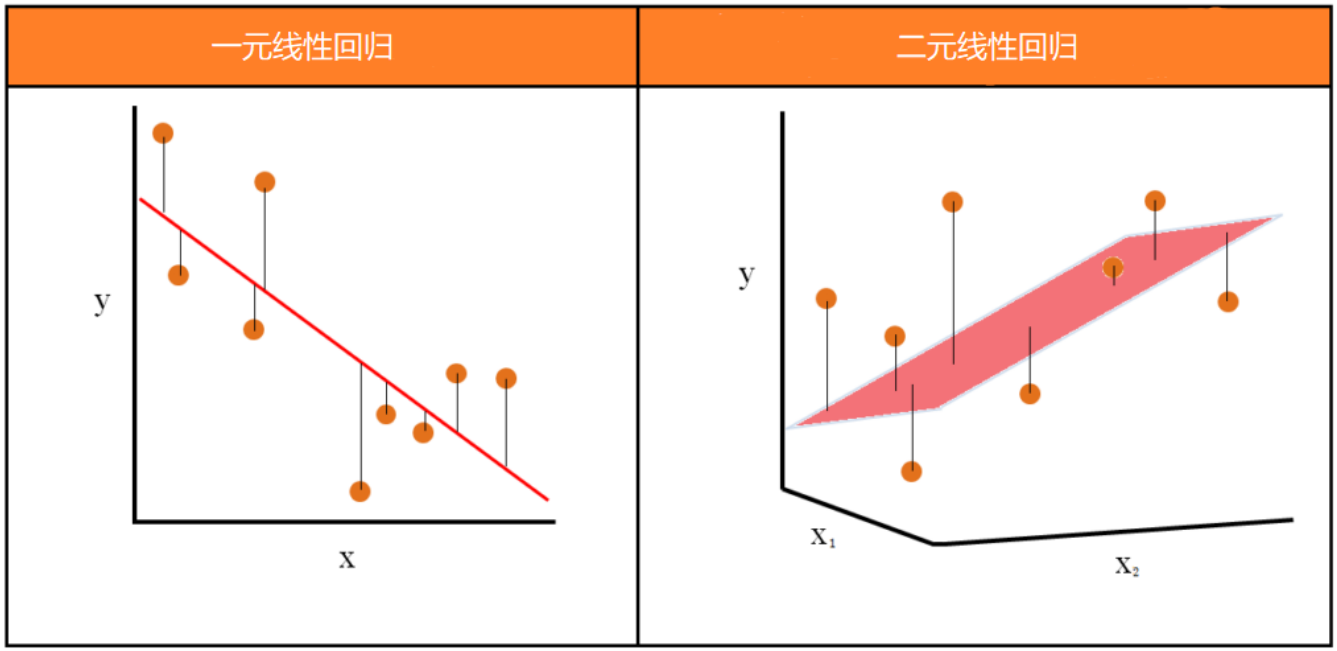


图 1: 线性回归示意图

m 个自变量，n 个样本，构成矩阵 X：

$$X = \begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_m^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & \cdots & x_m^{(n)} \end{bmatrix}$$

第 i 个样本为 $(x_1^{(i)}, \cdots, x_m^{(i)}, y_i)$. 令

$$\beta = (\beta_0, \beta_1, \cdots, \beta_m)^T$$

则 $\hat{Y} = X\beta$. 仍用最小二乘法找到最优的回归系数，结果形式不变：

$$\beta = (X^T X)^{-1} X^T Y$$

称为**正规方程法**。

三. 回归诊断

线性回归模型的成功建模，依赖于如下的假设：

(1) 线性模型假设：$y = X\beta + \varepsilon$
(2) 随机抽样假设：每个样本被抽到的概率相同且同分布；
(3) 无完全共线性假设：X 满秩；
(4) 严格外生性假设：$E(\varepsilon \mid X) = 0$
(5) 球形扰动项假设：$Var(\varepsilon \mid X) = \sigma^2 I_n$
(6) 正态性假设：$\varepsilon \mid X \sim N(0, \sigma^2 I_n)$

其中，前三个是基础假设，严格外生性和球形扰动项假设分别保证了估计量的无偏性和有效性，正态性假设是为了进行统计推断做的额外假设：

- 前四个假设成立时，估计量无偏；
- 前五个假设成立时，估计量有效，是最优线性无偏估计量；
- 所有假设都成立时，估计量是最优估计量。

1. 拟合优度检验

计算 R^2，也称为可决系数，反映了自变量所能解释的方差占总方差的百分比：

$$\mathrm{SST}=\sum_{i=1}^{n}(y_i-\bar{y})^2,\quad \mathrm{SSR}=\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2,\quad \mathrm{SSE}=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$$

$$R^2=\frac{\mathrm{SSR}}{\mathrm{SST}}=1-\frac{\mathrm{SSE}}{\mathrm{SST}}$$

R^2 值越大说明模型拟合效果越好。

注： R^2 未考虑自由度问题，为避免增加自变量而高估 R^2，选择调整的 R^2 是更合理的：

$$R^2_{adj}=1-\frac{n-1}{n-p-1}\cdot(1-R^2)$$

其中，n 为样本数，p 为自变量个数。

2. 均方误差与均方根误差

均方误差：

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

均方根误差：

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

均方根误差，刻画的是预测值与真实值平均偏离多少，是所有回归模型（包括机器学习中的回归算法）最常用的性能评估指标。

3. 残差检验

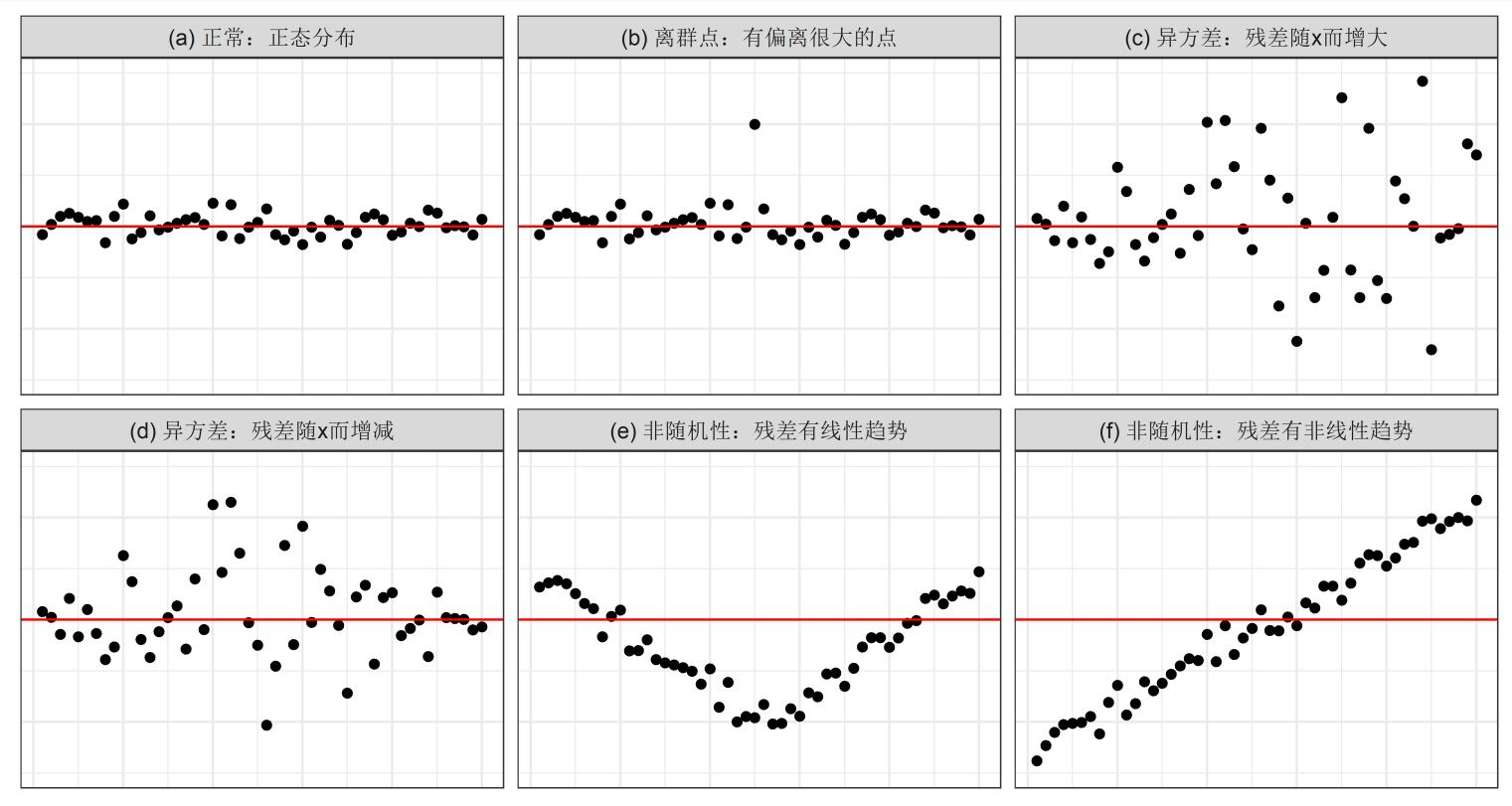


图 2: 残差分类图

- 只有图 (a) 说明模型是成功的，把模型部分都提取出来了；
- **(e)** 和 (f) 属于模型有问题，没有把模型部分提取完全；
- **(b)** 说明数据有异常点，应处理掉它重新建模；
- **(c)** 残差随 x 的增大而增大，(d) 残差随 x 的增大而先增后减，都属于异方差。

(1) 残差正态性检验

用残差检验模型是否成功，就是对残差做正态性检验。也可以考察学生化残差(可回避标准化残差的方差齐性假设）是否服从标准正态分布。

(2) 残差独立性检验

残差是白噪声，也表明不具有自相关性（独立性）。用 `Durbin-Watson` 检验：

$$H_0：\text{残差不存在自相关；} \quad H_1：\text{残差是相关的}$$

$$\mathrm{DW} = \sum_{i=2}^{n} \frac{(\varepsilon_i - \varepsilon_{i-1})^2}{\mathrm{SSE}}$$

用 `lmtest::dwtest()` 实现：

- $\mathrm{DW} \approx 0$，表示残差中存在正自相关；
- $\mathrm{DW} \approx 4$，表示残差中存在负自相关；
- $\mathrm{DW} \approx 2$，表示残差不存在自相关。

若残差存在自相关性，则需要考虑给模型增加自回归项。

(3) 异方差检验

线性回归的模型假设包括 $Var(\varepsilon \mid X) = \sigma^2 I_n$，即要求残差的方差是不随样本而变化的相同值 σ^2，否则就称为残差具有异方差性。

检验残差的异方差性，可用 Breusch-Pagan 检验，原假设是不存在异方差。用 `lmtest::bptest()` 实现。

异方差将导致回归系数的标准误估计错误，一种解决办法是估计异方差—稳健标准误。另一种是在回归之前对数据 y 或 x 进行变换，实现方差稳定后再建模。原则上，当残差方差变化不太快时取开根号变换 $\sqrt{y}$；当残差方差变化较快时取对数变换 $\ln y$；当残差方差变化很快时取逆变换 $1/y$；还有其他变换，如著名的 Box-Cox 变换或 Yeo-Johnson 变换（可应付负值），将非正态分布数据变换为正态分布.

4. 共线性诊断

多元线性回归建模，若自变量数据之间存在较强的线性相关性，即存在**多重共线性**。

多重共线性，会导致回归模型不稳定，这样得到的回归模型，是伪回归模型，就是并不反映自变量与因变量的真实影响关系。

比如，真实模型关系是 $y = 2x_1 + 3x_2$，若 x_1 与 x_2 存在线性关系：$x_2 = 2x_1$，则建模成 $y = 4x_1 + 2x_2$，$y = 6x_1 + x_2$，$y = 8x_1, \cdots$ 都完全没有问题。

多元线性回归建模，需要做**共线性诊断**，识别出多重共线性，并处理多重共线性再建模。

从线性相关系数、回归模型的方差膨胀因子 VIF（大于 10）来确定：

$$\text{VIF} = \frac{1}{1 - R_j^2}$$

它是 $\text{Var}(\hat{\beta}_j)$ 的决定性因子，其中 R_j 是第 j 个自变量与其余自变量之间的可决系数，R_j^2 越接近 1，说明该变量越能被其余变量所解释。

多重共线性的解决办法（任选其一）：

- 若两个自变量线性相关系数较大，则只用其中一个自变量；
- 用逐步回归，剔除冗余的自变量，得到更稳健的回归模型；
- 用主成分回归，相当于对自变量进行重组（将线性相关性强的变量合成为主成分），再做线性回归；
- 利用正则化回归：岭回归、Lasso 回归、弹性网模型（岭回归与 Lasso 回归的组合）

5. 回归系数的检验

(1) 回归系数的显著性

回归方程反映了因变量 y 随自变量 x 变化而变化的规律，若其系数 $\beta_1 = 0$，则 y 不随 x 变化，此时回归方程无意义。所以，要做 β_1 是否显著非 0 的假设检验：

$$H_0: \beta_1 = 0, \quad H_1: \beta_1 \neq 0$$

F 检验[1]：若 H_0 为真，则回归平方和 RSS 与残差平方和 $\frac{\text{ESS}}{n-2}$ 都是 σ^2 的无偏估计，构造 F 统计量：

$$F = \frac{\text{SSR}/\sigma^2/1}{\text{SSE}/\sigma^2/(n-2)} = \frac{\text{SSR}}{\text{SSE}/(n-2)} \sim F(1, n-2)$$

来检验原假设 $\beta_1 = 0$ 是否为真。

[1]也可以用的 t 检验，与 F 检验是等价的，因为 $t^2 = F$.

(2) 回归标准误与回归系数标准误

回归模型的标准误，衡量的是以样本回归直线为中心分布的观测值同直线上拟合值的平均偏离程度：

$$s = \sqrt{\frac{\mathrm{SSE}}{n-p}} = \sqrt{\frac{\sum_{i=1}^{n}(\varepsilon_i - \bar{\varepsilon}_i)^2}{n-p}} = \sqrt{\frac{\sum_{i=1}^{n}\varepsilon_i^2}{n-p}}$$

其中，SSE 为残差平方和，n 为样本数，$n-p$ 为自由度，p 为包括常数项在内的自变量的个数。

回归系数标准误（抽样误差的标准差），是对回归系数这一估计量标准差的估计值，衡量的是在一定的样本量下，回归系数同其期望的平均偏离程度[2]：

$$SE(\hat{\beta}_k) = \sqrt{\mathrm{Var}(\hat{\beta}_k)} = \sqrt{s^2(X^TX)^{-1}_{kk}}$$

[2]该标准误是来自统计学家得到的理论公式，另一种方法是用 Bootstrap 法.

6. 回归模型预测

通过检验的回归模型，就可以用来做预测：将新的自变量数据代入回归模型计算 y

例如，得到一元线性回归方程 $\hat{y} = \beta_0 + \beta_1 x$ 后，预测 $x = x_0$ 处的 y 值为 $\hat{y}_0 = \beta_0 + \beta_1 x$，其置信区间[3]为：

$$(\hat{y}_0 - t_{\alpha/2}\sqrt{h_0\hat{\sigma}^2},\ \hat{y}_0 + t_{\alpha/2}\sqrt{h_0\hat{\sigma}^2})$$

其中，$t_{a/2}$ 的自由度为 $n-2$，$h_0 = \frac{1}{n} + \frac{(x_0-\bar{x})^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$ 称为杠杆率，$\hat{\sigma}^2 = \frac{\mathrm{SSE}}{n-2}$.

[3]该置信区间是基于理论公式，也可以用 Bootsrap 法.

四. 多元线性回归实例

1. 准备数据与简单探索

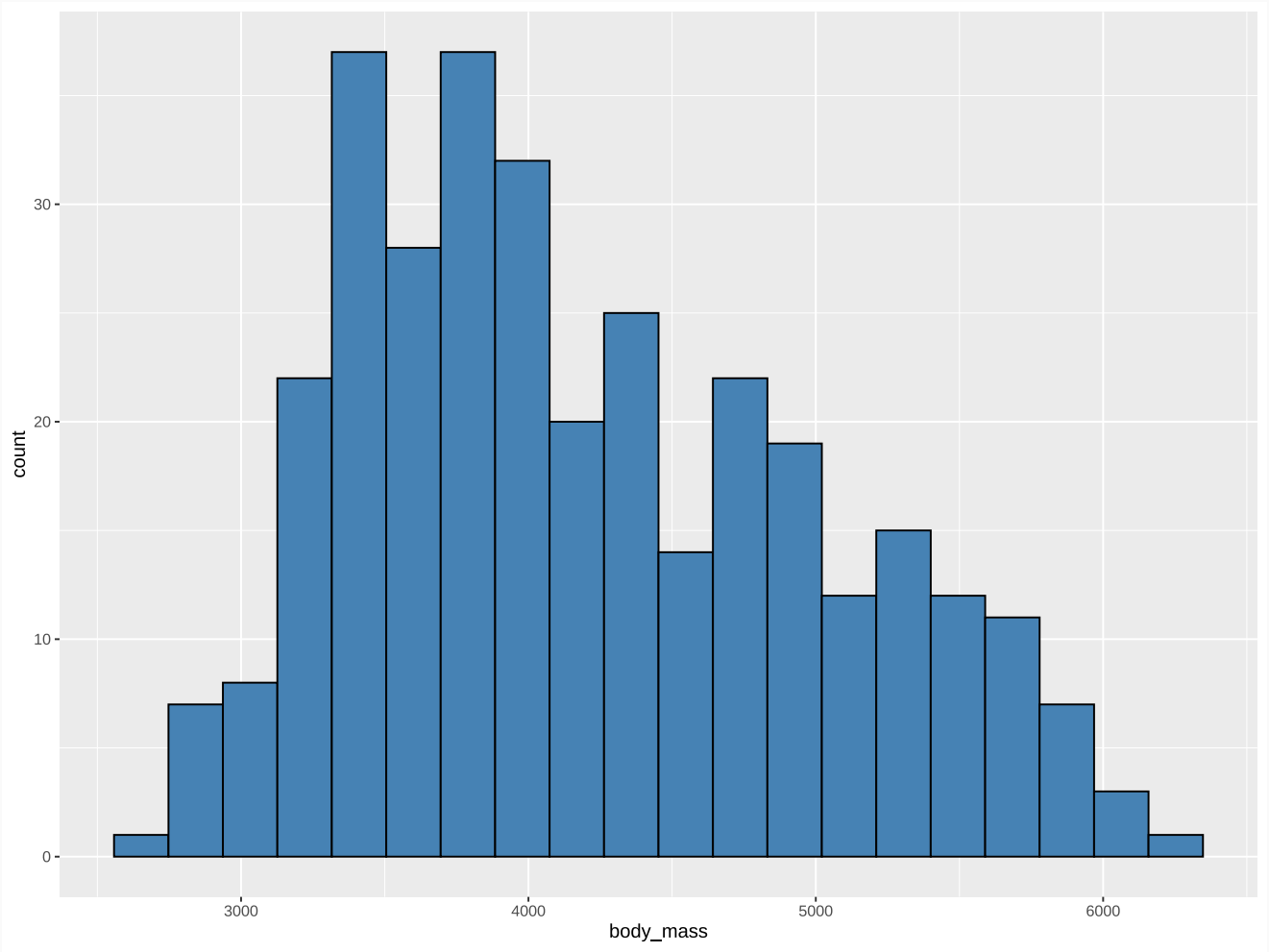
企鹅的数据集 penguins，包含 333 个样本，是有关企鹅的特征信息，包括种类、岛屿、嘴长、嘴宽、鳍长、性别。想确定企鹅体重与这些特征的关系。

```
penguins = read_csv("data/penguins.csv") %>%
  mutate(species = factor(species))
penguins
#> # A tibble: 333 x 7
#>   species island    bill_length bill_depth flipper_leng
#>   <fct>   <chr>           <dbl>      <dbl>          <db
#> 1 Adelie  Torgersen        39.1       18.7            18
#> 2 Adelie  Torgersen        39.5       17.4            18
#> 3 Adelie  Torgersen        40.3       18              19
#> # ... with 330 more rows
```

先探索因变量 body_mass（体重）的分布[4]：

```
ggplot(penguins, aes(body_mass)) +
  geom_histogram(bins = 20, fill = "steelblue",
                 color = "black")
```

[4]若因变量是右偏分布，可以尝试做对数变换变成近似正态分布，这里不做变换.


count
30
20
10
0
3000
4000
5000
6000
body_mass

2. 构建多元线性回归模型

`lm(formula, data, ...)`: 拟合多元线性回归模型

- `formula` 为要拟合的回归模型的形式，例如：`y ~ x1 + x2`, 对应模型 $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$, 默认包含截距项，若不想包含截距项用 `y ~ x1 + x2 -1`
- 返回值列表包含回归系数、统计量、拟合值、残差等，用 `summary()` 查看汇总模型结果，或者用 `broom` 包提供的 `tidy()`, `glance()`, `augment()` 将模型结果变成整洁数据框。

- `formula` 设定模型公式，遵从 `Wilkinson` 表示规则，更多常用写法：
  - `y ~ .`：包含所有自变量的主效应
  - `x1:x2`：交互效应，即 $x_1 x_2$ 项
  - `x1*x2`：包含全部主效应和交互效应，`x1 + x2 + x1:x2` 的简写
  - `I()`：打包式子作为整体
  - `y ~ poly(x, 2, raw = TRUE)`：一元二次多项式回归，同 `y ~ x + I(x^2)`
  - `y ~ polym(x1, x2, degree = 2, raw = TRUE)`：二元二次多项式回归
  - `log(y) ~ x`：对 `y` 做对数变换
- 先把自变量都用上，构建初始多元线性回归模型（往往不是成功模型）：

```
mdl0 = lm(body_mass ~ ., penguins)    # 结果略
```

3. 共线性诊断与逐步回归

用 `car::vif()`[5]诊断回归模型的多重共线性：

```
car::vif(mdl0)
#>                    GVIF Df GVIF^(1/(2*Df))
#> species           63.52  2             2.82
#> island             3.73  2             1.39
#> bill_length        6.10  1             2.47
#> bill_depth         6.10  1             2.47
#> flipper_length     6.80  1             2.61
#> sex                2.33  1             1.53
```

只有分类变量 `species` 的 VIF 值较大，其余均小于 10，说明不存在共线性。

[5] `mctest::imcdiag()` 诊断回归模型的多重共线性更全面，除了计算 VIF 值外，还计算其他诊断指标值.

处理该共线性，可以剔除相对不那么重要的变量，或者用 `step()` 做逐步回归，它可以剔除不显著的自变量，顺便剔除共线性的自变量。

逐步回归是以 AIC 值（越小越好）作为加入和剔除变量的判别条件，参数 `direction` 设置逐步选择的方法："both"，"backward"（逐步剔除），"forward"（逐步加入）。

Akaike 信息准则（AIC）常用来比较不同回归模型的拟合效果，优点是既考虑模型的拟合效果又对模型参数过多施加一定惩罚，其定义为：

$$\text{AIC} = 2(p+1) - 2\ln(L)$$

其中，p 为回归模型中自变量的个数，L 为回归模板的对数似然。

```r
mdl1 = step(mdl0, direction = "backward", trace = 0)
summary(mdl1)
#>
#> Call:
#> lm(formula = body_mass ~ species + bill_length + bill_d
#>     flipper_length + sex, data = penguins)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -779.7 -173.2   -9.1  186.6  914.1
#>
#> Coefficients:
#>                   Estimate Std. Error t value Pr(>|t|)
#> (Intercept)      -1460.99     571.31   -2.56  0.01100 *
#> speciesChinstrap  -251.48      81.08   -3.10  0.00209 **
#> speciesGentoo     1014.63     129.56    7.83  6.9e-14 **
#> bill_length         18.20       7.11    2.56  0.01086 *
```

结果给出了回归系数的标准误、显著性、回归模型的标准误等，基于理论的回归系数的置信区间，可用 confint() 来提取：

```
confint(mdl1)
#>                        2.5 % 97.5 %
#> (Intercept)        -2584.91 -337.1
#> speciesChinstrap    -410.98  -92.0
#> speciesGentoo        759.75 1269.5
#> bill_length            4.22   32.2
#> bill_depth            28.38  106.1
#> flipper_length        10.23   21.7
#> sexmale              295.76  484.0
```

该模型基本上是成功的模型，回归系数都是显著的，模型的调整 R^2 为 0.873。

要计算模型的均方根误差：

```
library(modelr)
rmse(mdl1, penguins)
#> [1] 284
```

4. 关于回归模型中的分类变量

分类变量，取值是有限的类别值，如性别：男、女。分类变量是不能直接用到回归模型中的，即使用 1 表示男，用 0 表示女，这个 1 和 0 仍然只能是起类别区分的作用，如果不加处理让它们当数值 1 和 0 使用了，那么整个模型的逻辑和结果都是不正确的！

分类变量要想正确地用到回归模型，必须处理成虚拟变量[6]。

[6] R 中分类变量只要是因子型或字符型，当加入回归模型时，不需要做任何额外操作将自动处理成虚拟变量用进模型.

企鹅数据 `species` 列是分类变量，包含 3 个类别："Adelie"，"Gentoo"，"Chinstrap".

```
table(penguins$species)
#>
#>    Adelie Chinstrap    Gentoo
#>       146        68       119
```

虚拟变量是一种二值变量（0-1），只表示是否。二分类或多分类变量，可以转化为多个二值变量：

species 是否为 Adelie, species 是否为 Gentoo, species 是否为 Chinstrap

比如第 1 个样本，其 `species = Adelie`，要用上述 3 个二值变量表示的话，就是分别为 `1，0，0`.

每个样本都做这样的处理，这就是分类变量转化为虚拟变量，可用 modelr::model_matrix() 函数实现，其参数 data 为数据，formula 为模型公式。

若给 formula 参数提供用于 lm() 的模型公式，则返回真正用于回归模型的分类变量处理成虚拟变量的自变量数据：

```
# species 变成虚拟变量的效果
model_matrix(penguins, ~ species - 1)
#> # A tibble: 333 x 3
#>   speciesAdelie speciesChinstrap speciesGentoo
#>           <dbl>            <dbl>         <dbl>
#> 1             1                0             0
#> 2             1                0             0
#> 3             1                0             0
#> # ... with 330 more rows
```

不是将原 `species` 列，而是换成新的虚拟变量列用到回归模型，注意：这 3 个虚拟变量列是线性相关的：每一列都能用其余 2 列线性表示（1 减去其余 2 列），即有一列是冗余的，这是线性回归所不允许的。

故需要任意去掉一列，再线性回归建模。去掉哪一列都可以，去掉哪一列，做回归建模就相当于以谁为参照列。

比如去掉 `species` 是否为 `Adelie` 列，就相当于"Adelie" 组是参照组，另外 2 组"Gentoo"、"Chinstrap" 与参照组做比较。

去掉冗余列，再增加截距列（一列 1），才是将 species 列真正用于回归模型的转化为虚拟变量后的数据：

```
model_matrix(penguins, ~ species)
#> # A tibble: 333 x 3
#>   `(Intercept)` speciesChinstrap speciesGentoo
#>           <dbl>            <dbl>         <dbl>
#> 1             1                0             0
#> 2             1                0             0
#> 3             1                0             0
#> # ... with 330 more rows
```

冗余列默认是去掉第一水平，若想去掉另一水平（该组作为参照组），可以借助 relevel() 修改第一水平，再处理成虚拟变量：

```
penguins$species = relevel(penguins$species, ref = "Gentoo"
```

根据逐步回归得到的 `md11` 的回归系数估计，可以写出拟合的回归方程：

$$\begin{aligned}\text{body\_mass} = &-1460.995 - 251.477 * \text{speciesChinstrap} \\ &+ 1014.627 * \text{speciesGentoo} + 18.204 * \text{bill\_length} \\ &+ 67.218 * \text{bill\_depth} + 15.950 * \text{flipper\_length} \\ &+ 389.892 * \text{sexmale}\end{aligned}$$

连续变量的回归系数好解释，比如 `bill_length` 的系数 `18.204`，表示嘴长每增加 1 个单位（毫米），体重将增加 `18.204` 个单位（克）。

分类变量回归系数的解释

原二分类变量 `sex`，变成虚拟变量去掉冗余列后只剩一列 `sexmale`（是否为雄性，1 是 0 否），代入模型来看：

- 若性别不是雄性，`SexMale = 0`

$$\text{body\_mass} = \cdots + 389.892 * 0 + \cdots$$

- 若性别是雄性，`SexMale = 1`

$$\text{body\_mass} = \cdots + 389.892 * 1 + \cdots$$

即雌性则 `+ 0`，雄性则 `+ 389.892`，这就相当于是以雌性为参照组，雄性的体重平均比雌性重 `389.982` 克，这就是该回归系数的解释。

原多分类变量 `species`（3 分类），变成虚拟变量去掉冗余列 `speciesAdelie` 后剩下 2 列，若种类是 `Adelie`，这 2 列均为 0，即回归模型不包含这 2 项，此时是参照组；若种类是任一非参照的种类，比如 `Gentoo`，则 `speciesGentoo = 1`，此时回归模型多了一项：

$$\text{body\_mass} = \cdots + 1014.627 * 1 + \cdots$$

这就相当于是以 `Adelie` 为参照组，`Gentoo` 组相对于参照组 `Adelie` 平均体重要重 `1014.627` 克。

分类变量用于回归模型，所起的作用就是分组之间做比较，也只能是起分组比较的作用。这实际上也等效于分别对各分组建立线性回归模型，再做比较。

切记：分类变量用于建模时，始终是起分类的作用，绝对不能因为表示为数值形式，就直接当数值使用。

5. 模型改进

自变量又称为特征，利用原有自变量构造新的自变量，就是特征工程。特征工程是改进模型的重要手段，也是数据挖掘/机器学习中的关键步骤。

多元线性回归相当于是用 1 次多项式去逼近真实的函数关系，如果提高的 2 次，即把所有二次项包括交互项[7]：$x_1^2, x_2^2, x_1x_2, \cdots$ 都加入模型，拟合效果大概率会有提升。但新加入的项，可能会有不显著或产生共线性。解决办法，就是用逐步回归进行变量筛选。

常用的构建特征方法还有：对特征做各种变换，连续特征离散化，比如年龄相差 1 岁影响不一定显著，但年龄段的差异，比如从青年到中年到老年，很可能会显著。

---

[7]关于交互项 `x1:x2` 的解释：`x1` 对 `y` 的影响受 `x2` 的调节，反之亦同，其回归系数相当于 `y` 对 `x1` 和 `x2` 的二阶偏导.

将三个数值变量的二次项，以及交互项 sex:island 加入模型，再接逐步回归剔除不显著项：

```
mdl2 = lm(body_mass ~ species + sex * island + bill_length
          + I(bill_length^2) + bill_depth + I(bill_depth^2)
          + flipper_length + I(flipper_length^2),
          penguins) %>%
  step(direction = "backward", trace = 0)
```

```
summary(md12)
#>
#> Call:
#> lm(formula = body_mass ~ species + sex + island + bill_
#>     bill_depth + I(flipper_length^2) + sex:island, data
#>
#> Residuals:
#>    Min      1Q  Median      3Q     Max
#> -720.5 -186.5  -12.4   170.4   866.2
#>
#> Coefficients:
#>                              Estimate Std. Error t value Pr
#> (Intercept)                  1.09e+03   4.30e+02    2.54  0
#> speciesAdelie               -9.98e+02   1.36e+02   -7.31  2
#> speciesChinstrap            -1.25e+03   1.29e+02   -9.68  <
#> sexmale                      4.81e+02   5.72e+01    8.41  1
#> islandDream                  9.42e+01   6.74e+01    1.40  0
```

二次项 `flipper_length^2` 项和交互项 `sexmale:islandDream` 都非常显著的，模型的修正 R^2 比 `md11` 稍有提高 (0.0028)。

这说明 `md12` 相比 `md11` 有所改进，但同时也增加了模型的复杂度（多了 4 项)。那么，接受哪个模型更好呢?

基本原则是在模型没有显著差异的情况下，优先选择更简单的模型。

可用似然比检验 lmtest::lrtest() 或方差分析 anova() 比较两个模型有无显著差异:

```
anova(mdl1, mdl2)
#> Analysis of Variance Table
#>
#> Model 1: body_mass ~ species + bill_length + bill_depth
#>     sex
#> Model 2: body_mass ~ species + sex + island + bill_leng
#>     I(flipper_length^2) + sex:island
#>   Res.Df      RSS Df Sum of Sq    F Pr(>F)
#> 1    326 26915647
#> 2    322 26000518  4    915128 2.83  0.025 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.
```

检验 P 值 = 0.025 小于 0.05, 说明两个模型有显著差异，应该选择 mdl2.

6. 回归诊断

- **残差检验**

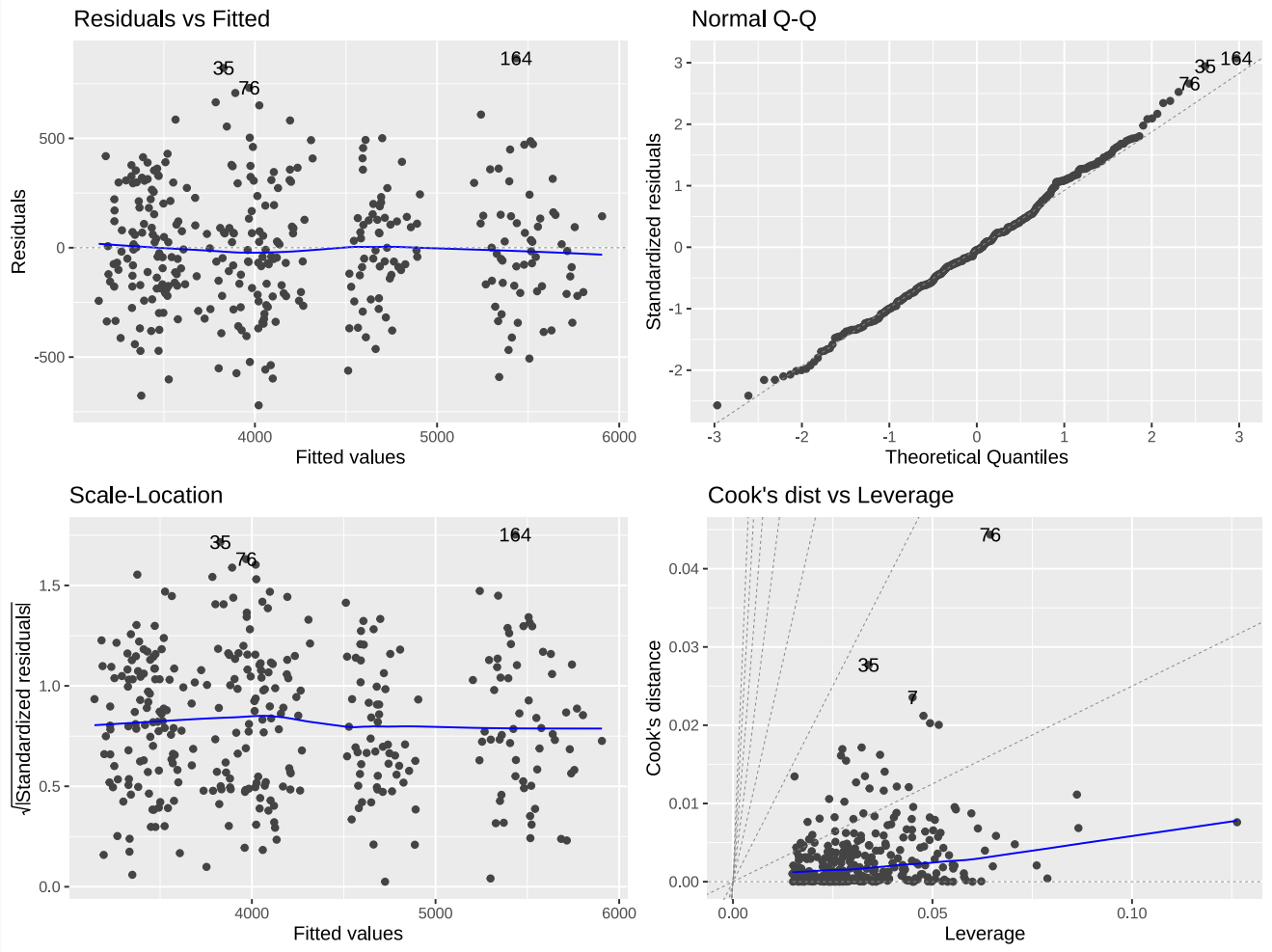
理想的模型（标准化）残差应服从"0 均值小方差"（标准）正态分布，对于残差，通常是绘制（标准化）残差图、残差 QQ 图、残差直方图，或者对（标准化）残差的正态性、独立性、异方差性做统计检验。

- **强影响分析**

对参数估计或预测值有异常影响的数据，称为强影响数据。回归模型应当具有一定的稳定性，若个别样本数据对估计有异常大的影响，剔除后将得到与原来差异很大的回归方程，从而有理由怀疑原回归方程是否真正描述了变量间的客观存在的关系。这些强影响样本是异常值，应当识别出来剔除之后，再重新拟合回归模型。

用 `ggfortify::autoplot()` 绘制回归诊断图，包括：残差图、残差 QQ 图、标准化残差图、强影响图等，还能同时标记强影响样本。

```
library(ggfortify)
autoplot(mdl2, which = c(1:3,6))    # 6 个图形可选
```



```
shapiro.test(mdl2$residuals)          # 残差正态性检验
#>
#>  Shapiro-Wilk normality test
#>
#> data:  mdl2$residuals
#> W = 1, p-value = 0.4
```

```
library(lmtest)
dwtest(mdl2)                                    # 残差独立性检验
#>
#>  Durbin-Watson test
#>
#> data:  mdl2
#> DW = 2, p-value = 0.9
#> alternative hypothesis: true autocorrelation is greater
bptest(mdl2)                                    # 残差异方差检验
#>
#>  studentized Breusch-Pagan test
#>
#> data:  mdl2
#> BP = 15, df = 10, p-value = 0.1
```

可见，mdl2 能通过残差正态性、独立性检验。

回归模型预测

通过检验的回归模型，提供新的自变量数据框，用 predict() 就可以预测因变量值。

```
newdat = slice_sample(penguins[,-6], n = 5)
predict(mdl2, newdat, interval = "confidence")
#>     fit  lwr  upr
#> 1 3323 3226 3419
#> 2 5576 5504 5648
#> 3 3672 3523 3821
#> 4 4686 4604 4768
#> 5 4754 4682 4826
```

五. 梯度下降法

正规方程法求解多元线性回归，简单、容易实现，但有其缺点：

- 若 $X^T X$ 不可逆，则正规方程法失效
- 若样本量非常大（$n > 10000$），矩阵求逆会非常慢

梯度下降法是广泛用于机器学习，其核心思想是迭代地调整参数，使得损失函数达到最小值。

梯度下降法，就好比在浓雾笼罩的山上下山，每次只能看到前方一步远，那么就 360° 每个方向迈一步的话下降的最多，那就往哪个方向迈一步，重复该过程，逐步到达较低点（不一定是最低点）。

根据数学知识，一步下降最快的方向就是梯度方向！

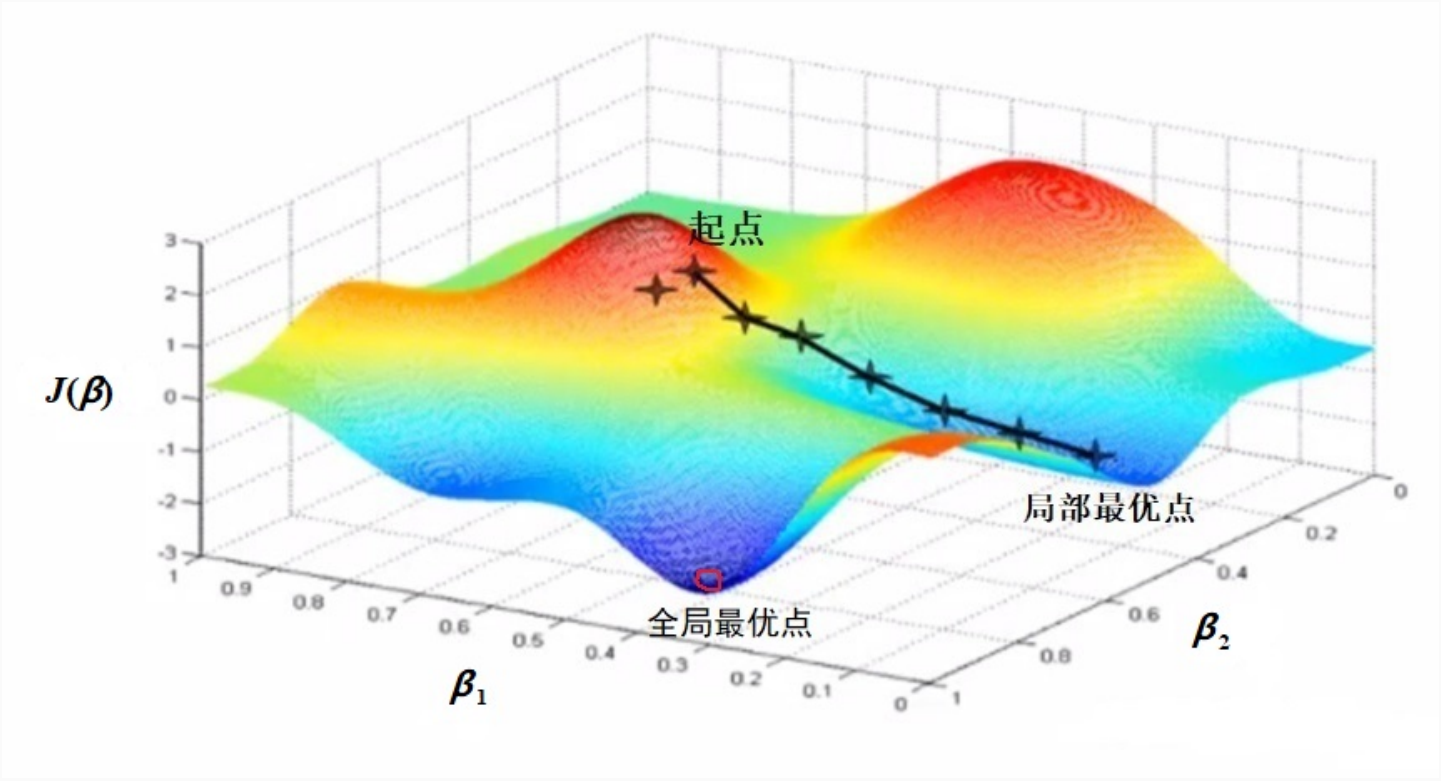


图 3: 梯度下降法示意图

线性回归问题，就是计算损失函数 $J(\beta)$ 关于参数向量 β 的局部梯度，同时它沿着梯度下降的方向进行下一次迭代。当梯度值为零的时候，就达到了损失函数最小值。

开始需要选定一个随机的 β（初始值），然后逐渐去改进它，每一次变化一小步，每一步都试着降低损失函数 $J(\beta)$，直到算法收敛到一个极小值。

该极小值不一定是全局最小值，若损失函数是凸函数（线性回归损失函数是凸函数），则极小值就是唯一的全局最小值。

梯度下降法的重要参数是每一步的步长，叫作**学习率**。一个好的策略是，开始的学习率大一些以更快速趋于收敛，让学习率慢慢减小，最后阶段要足够小以稳定地到达收敛点。

注：梯度下降法对自变量取值的量级是敏感的，若所有自变量的数量级基本相当，则能更快地收敛到最小值。所以，在用梯度下降法训练模型时，有必要对数据做归一化（放缩），以加速训练。

线性回归模型的损失函数为（除以 2 可抵消求偏导的系数）：

$$J(\beta) = \frac{1}{2n}\sum_{i=1}^{n}(x^{(i)}\beta - y_i)^2$$

在梯度下降法过程中，需要计算每一个 β_j（维度）下损失函数的梯度。即当 β_j 变化一点点时，损失函数改变了多少，这就是偏导数：

$$\frac{\partial}{\partial\beta_j}J(\beta) = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)}\beta - y_i)x_{ij}, \quad j = 1, \cdots, m$$

改为向量化表示，得到损失函数的梯度向量：

$$\nabla J(\beta) = \left[\frac{\partial}{\partial \beta_1} J(\beta), \cdots, \frac{\partial}{\partial \beta_m} J(\beta)\right] = \frac{1}{n} X^T (X\beta - y)$$

注意，梯度下降法每一步梯度向量的计算，都是基于整个训练集，故称为批量梯度下降：每一次训练过程都使用所有的训练数据。因此，在大数据集上，训练速度也会变得很慢[8]，但其复杂度是 $O(n)$，比正规方程法 $O(n^3)$ 快的多。

梯度向量有了，只需要每步以学习率 η 调整参数即可：

$$\beta^{\text{next}} = \beta - \eta \nabla J(\beta)$$

[8]进一步提速，还有随机梯度下降算法，每次只用一个随机样本计算梯度向量，但是收敛过程不够稳定，折衷的做法是小批量梯度下降算法.

- 定义函数实现梯度下降法求解线性回归

```
gd = function(X, y, init, eta = 1e-3, err = 1e-3,
              maxit = 1000, adapt = FALSE) {
  ## X 为自变量数据矩阵，y 为因变量向量，init 为参数初始值
  ## eta 为学习率，err 为误差限，maxit 为最大迭代次数，
  ## adapt 是否自适应修改学习率
  ## 返回回归系数估计，损失向量，迭代次数，拟合值，RMSE
  # 初始化
  X = cbind(Intercept = 1, X)
  beta = init
  names(beta) = colnames(X)
  loss = crossprod(X %*% beta - y)
  tol = 1
  iter = 1
```

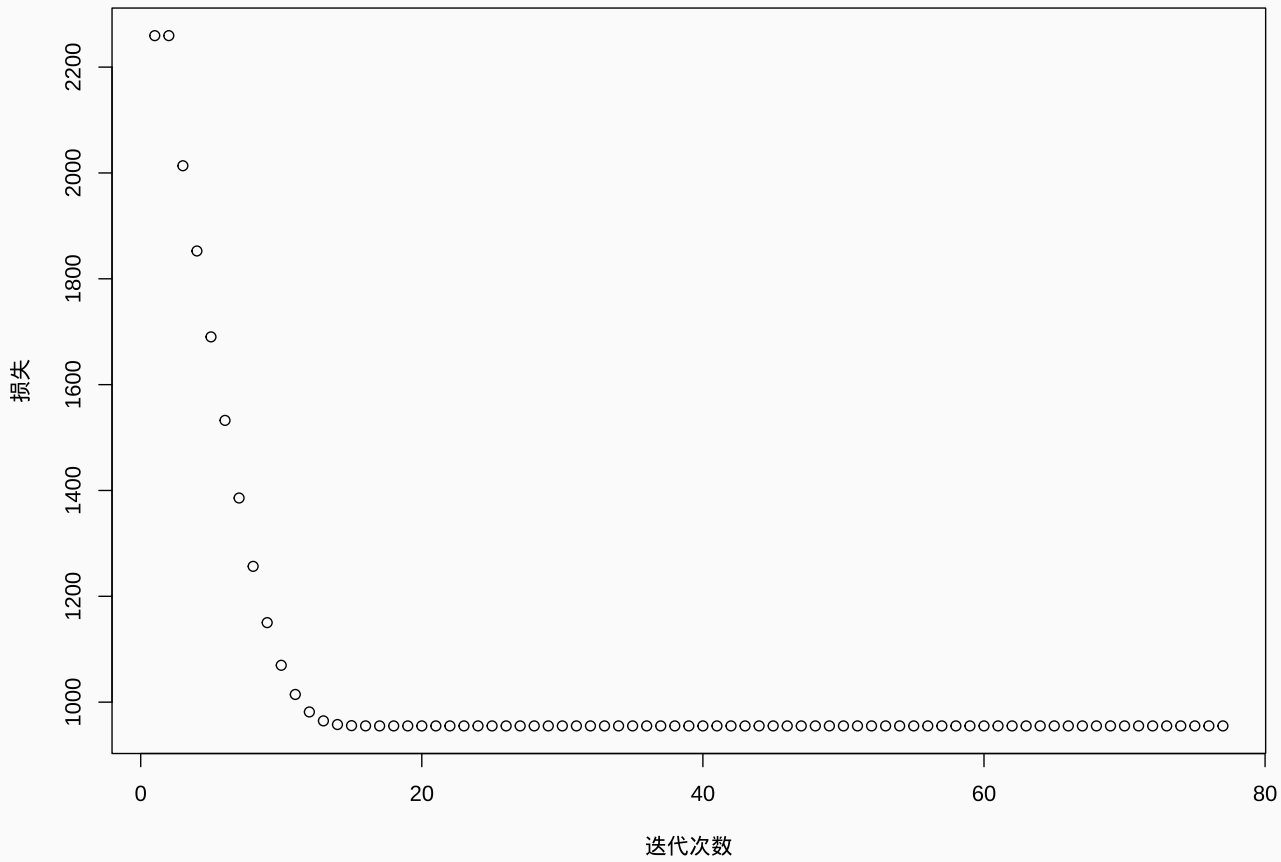
```
  while(tol > err && iter < maxit) {          # 迭代
    LP = X %*% beta
    grad = t(X) %*% (LP - y)
    betaC = beta - eta * grad
    tol  = max(abs(betaC - beta))
    beta = betaC
    loss = append(loss, crossprod(LP - y))
    iter = iter + 1
    if(adapt)
      eta = ifelse(loss[iter] < loss[iter-1], eta * 1.2,
                   eta * 0.8)
  }
  list(beta = beta, loss = loss, iter = iter, fitted = LP,
    RMSE = sqrt(crossprod(LP - y) / (nrow(X) - ncol(X))))
}
```

- 用随机生成数据的二元线性回归来测试函数

```
n = 1000
set.seed(123)
x1 = rnorm(n)
x2 = rnorm(n)
y = 1 + 0.6*x1 - 0.2*x2 + rnorm(n)
X = cbind(x1, x2)
```

```r
gd_rlt = gd(X, y, rep(0,3), err = 1e-8, eta = 1e-4,
            adapt = TRUE)
rbind(gd = round(gd_rlt$beta[, 1], 5),
      lm = coef(lm(y ~ x1 + x2)))       # 与 lm 结果对比
#>    Intercept    x1     x2
#> gd     0.979 0.579 -0.172
#> lm     0.979 0.579 -0.172
gd_rlt$iter                             # 迭代次数
#> [1] 77
```

```
plot(gd_rlt$loss, xlab = " 迭代次数", ylab = " 损失")
```



可见，算法收敛速度非常快，迭代 14 步损失函数基本就不再减小。

六. 广义线性模型

线性回归是回归家族的基本模型，从不同角度进行扩展可以衍生出几十种回归模型。

线性回归要求残差满足正态性：$\varepsilon = y - X\beta \sim N(0, \sigma^2)$, 则 $y \sim N(X\beta, \sigma^2)$. 这说明线性回归通常要求因变量 y 是近似服从正态分布的连续数据。

但实际中，因变量数据可能会是类别型、计数型等，可以考虑对 y 做变换，或直接考虑广义线性模型。

要让线性回归也适用于因变量非正态连续情形，就需要推广到广义线性模型。Logistic 回归、softmax 回归、泊松回归、Probit 回归、二项回归、负二项回归、最大熵模型等都是广义线性模型的特例。

广义线性模型，相当于是复合函数。先做线性回归，再接一个变换：

$$\mathbf{w}^T X + \mathbf{b} = \ u \sim \text{正态分布}$$

$$\downarrow$$

$$g(u) = y$$

经过变换后到达非正态分布的因变量数据。

一般更习惯反过来写：即对因变量 y 做一个变换，就是正态分布，从而就可以做线性回归：

$$\sigma(y) = \mathbf{w}^T X + \mathbf{b}$$

其中，$\sigma(\cdot)$ 称为连接函数。

回归模型	变换	连接函数	逆连接函数	误差
线性回归	恒等	$\mu_Y = X^T\beta$	$\mu_Y = X^T\beta$	正态分布
Logistic 回归	Logit	$\text{Logit}\,\mu_Y = \mathrm{X^T}\beta$	$\mu_Y = \frac{\exp(X^T\beta)}{1+\exp(X^T\beta)}$	二项分布
泊松回归	对数	$\ln\mu_Y = X^T\beta$	$\mu_Y = \exp(X^T\beta)$	泊松分布
负二项回归	对数	$\ln\mu_Y = X^T\beta$	$\mu_Y = \exp(X^T\beta)$	负二项分布
Gamma 回归	逆	$\frac{1}{\mu_Y} = X^T\beta$	$\mu_Y = \frac{1}{X^T\beta}$	Gamma 分布

图 4: 常见连接函数与误差函数

因变量数据只要服从指数族分布：正态分布、伯努利分布、泊松分布、指数分布、Gamma 分布、卡方分布、Beta 分布、狄里克雷分布、Categorical 分布、Wishart 分布、逆 Wishart 分布等，就可以使用对应的广义线性模型。广义线性模型用 `glm()` 函数实现，通过 `family` 参数设置分布名，以决定选用的模型。

注：泊松回归和负二项回归都是针对因变量是计数数据，区别是泊松回归一般用于个体之间独立的情形；负二项回归则可用于个体之间不独立的情形。

本篇主要参阅 (张敬信, 2022), Michael Clark: gradient_descent，模板感谢 (黄湘云, 2021), (谢益辉, 2021).

参考文献

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*.