第 2 章 线性回归：机器学习基础技术

“有人总想在这项竞技中找到根本不存在的东西，但橄榄球世界里只有两件事——拦截与抢断。”

——文斯·隆巴迪，橄榄球名人堂教练

我们非常有必要从一项简单但又特别有效的技术开始，这项技术已经应用了很长时间，它就是**线性回归**。我记得阿尔伯特·爱因斯坦曾经说过，事情应该尽可能简单，直到不能再简单为止。这真是至理名言，也是我们开发机器学习算法时应该遵循的经验法则。线性回归使用**最小二乘法**预测定量的结果，想想我们随后将讨论的其他技术，真的没有比久经考验的线性回归更简单的模型了。实际上，线性回归是我们后面要讨论的所有方法的基础，很多方法仅是线性回归的扩展。坦白地说，如果你能掌握线性回归方法，那么本书的其余部分就易如反掌。因此，在我们成为机器学习专家的道路上，线性回归是一个非常好的起点。

本章包含一些入门级介绍，如果你是这方面的专家，可以跳过这些内容，直接进入下一主题。否则，请确保你完全理解了线性回归，然后才能开始学习那些更复杂的机器学习方法。你会发现很多项目仅靠随后讨论的技术就可以完成。线性回归可能是最容易向客户解释的模型了，大多数客户都能大致理解**R方**（**R-squared**）的意义，很多人可以理解得更深入，对变量贡献、**共线性**等概念都能接受。

2.1 单变量回归

我们先从一个简单的对定量型响应变量的预测开始。令这个响应变量为Y，还有一个预测变量x，假设Y与x具有线性关系，那么这个预测模型可以表示为$Y=B_0+B_1x+e$。我们规定，Y的预测值是一个函数，等于B_0（截距）加上B_1（斜率）乘以x再加上一个误差项e。最小二乘法选择模型参数，使预测值y和实际值Y的**残差平方和**（**RSS**）最小。举个简单的例子，假设我们有两个实际值Y_1和Y_2，分别等于10和20，两个预测值y_1和y_2分别等于12和18。要计算RSS，只需把它们的差的平方相加：$\text{RSS}=(Y_1-y_1)^2+(Y_2-y_2)^2$，再做一个简单的代入，可以得到：$(10-12)^2+(20-18)^2=8$。

我曾经和一个一起进行精益六西格玛黑带培训的伙伴说过，线性规划中最重要的就是平方和；理解了平方和，其余就水到渠成了。从某种程度来说，的确如此。

开始实际应用之前，我想提醒一下，当你看到有关突破性研究的报道时，先不要轻信，要有质疑的态度，因为媒体发表的结论可能未经验证。我们知道，对于R或其他相关软件，只要有输入，就会给出一个结果。但是，仅凭数学上有意义和很高的相关性以及漂亮的R方统计量，是不能认为结论正确的。

为了说明这个问题，请看R中著名的Anscombe数据集。它由统计学家弗朗西斯·安斯库姆（Francis Anscombe）建立，用来强调数据可视化和异常值在数据分析中的重要性。这个数据集有4对*X*变量和*Y*变量，它们具有相同的统计特性。但如果将其放在统计图中，就会看到一些极大的差异。我用这个数据集进行内部培训，还教育过那些只盯着统计量而不进行数据探索和假设检验的商业伙伴。如果你有同样的需求，那这个例子就是一个非常好的开始。这只是我们正式建模之前的一个小插曲。

```
    > #call up and explore the data

> data(anscombe)

> attach(anscombe)

> anscombe
   x1  x2  x3  x4    y1    y2    y3    y4
1  10  10  10   8  8.04  9.14  7.46  6.58
2   8   8   8   8  6.95  8.14  6.77  5.76
3  13  13  13   8  7.58  8.74 12.74  7.71
4   9   9   9   8  8.81  8.77  7.11  8.84
5  11  11  11   8  8.33  9.26  7.81  8.47
6  14  14  14   8  9.96  8.10  8.84  7.04
7   6   6   6   8  7.24  6.13  6.08  5.25
8   4   4   4  19  4.26  3.10  5.39 12.50
9  12  12  12   8 10.84  9.13  8.15  5.56
10  7   7   7   8  4.82  7.26  6.42  7.91
11  5   5   5   8  5.68  4.74  5.73  6.89
```

可以看到，每对变量都具有相同的相关系数0.816。前两对变量的相关系数如下：

```
    > cor(x1, y1) #correlation of x1 and y1
[1] 0.8164205

> cor(x2, y1) #correlation of x2 and y2

[1] 0.8164205
```

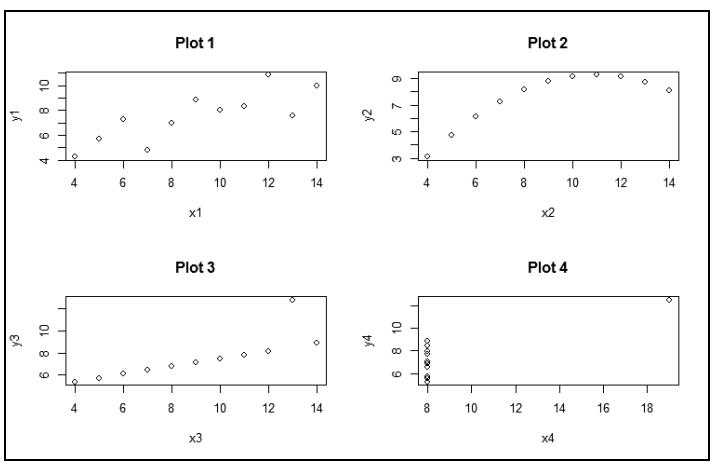
当我们画出这4对变量的统计图时，就能看出问题了，这就是Anscombe这个数据集的设计目的。如下所示：

```
    > par(mfrow = c(2,2)) #create a 2x2 grid for
      plotting

> plot(x1, y1, main = "Plot 1")

> plot(x2, y2, main = "Plot 2")

> plot(x3, y3, main = "Plot 3")

> plot(x4, y4, main = "Plot 4")
```

下载示例代码

可以通过“图灵社区”本书页面（http://www.ituring.com.cn/book/1989）下载书中示例代码。

上述代码输出如下。



可以看到，Plot 1中呈现的是真正的线性关系，Plot 2是一条曲线，Plot 3有一个危险的离群点，Plot 4则完全被离群点“拐跑了”。看到了吧，这就是一则警世恒言，说明了仅看相关性有多么危险。

业务理解

我们的第一个例子重点关注的是预测怀俄明州蛇河流域的水量（以英寸为计量单位），它是年度降雪含水量的一个函数。这项预测有助于管理水流量和蓄水量，因为蛇河是美国西部几个州农牧场灌溉用水的主要来源。数据集snake可以在alr3包中找到（注意，alr表示实用线性回归）：

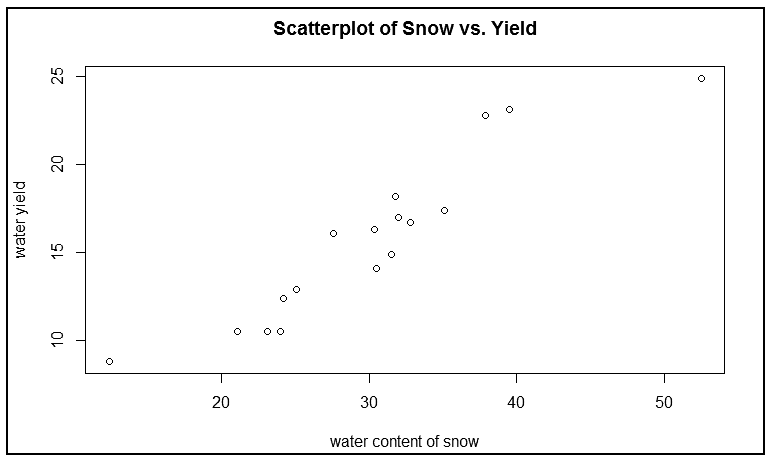
```
    > install.packages("alr3")
> library(alr3)
> data(snake)
> dim(snake)
[1] 17 2
```

```
> head(snake)
     X    Y
1 23.1 10.5
2 32.8 16.7
3 31.8 18.2
4 32.0 17.0
5 30.4 16.3
6 24.0 10.5
```

既然我们有了17行观测值，下面可以进行数据探索了。别急，先将*X*和*Y*换成有意义的变量名，如下所示：

```
    > names(snake) <- c("content", "yield")
> attach(snake) # attach data with new names
> head(snake)

    content yield
1      23.1  10.5
2      32.8  16.7
3      31.8  18.2
4      32.0  17.0
5      30.4  16.3
6      24.0  10.5

> plot(content, yield, xlab = "water content of
       snow", ylab = "water yield")
```

上述代码输出如下。



这张图很有意思，它的数据是线性的，因为被最前端和最后端的两个疑似离群点影响，有一点轻微的曲线形状。所以，有必要进行数据转换并删除无关观测值。

R使用`lm()`函数进行线性回归，`lm()`可以建立一个标准形式的回归模型fit = lm(*Y*~*X*)。建立模型之后，你可以对拟合模型使用各种函数，以检验自己的假设。代码如下：

```
    > yield.fit <- lm(yield ~ content)

> summary(yield.fit)

Call:
lm(formula = yield ~ content)

Residuals:
        Min      1Q Median      3Q     Max
-2.1793 -1.5149 -0.3624 1.6276 3.1973

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.72538 1.54882  0.468 0.646
content     0.49808 0.04952 10.058 4.63e-08
     ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05
      '.' 0.1 ' ' 1

Residual standard error: 1.743 on 15 degrees of
      freedom
Multiple R-squared: 0.8709,     Adjusted R-squared:
       0.8623
F-statistic: 101.2 on 1 and 15 DF, p-value:
       4.632e-08
```

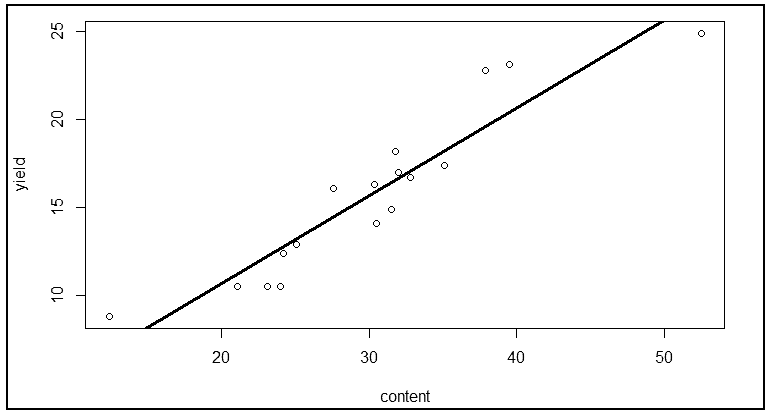
通过summary()函数，我们可以查看模型包含的一些项目，比如模型具体参数、关于残差的描述性统计量、系数、模型显著性代码、模型误差和拟合程度的摘要。现在，让我们重点关注对于相关系数这个参数的估计，看一下我们的预测变量是否具有显著的*p*值，以及整个模型的F检验是否具有显著的*p*值。请看参数估计，模型告诉我们，yield等于0.72538加上0.49808乘以content。可以确定，content每变动1个单位，yield会增加0.49808个单位。F统计量是用来检验原假设的，原假设认为模型的所有系数都是0。

因为*p*值是高度显著的，所以我们可以拒绝原假设。接下来看content变量的t检验值，原假设认为它应该是0，我们又一次拒绝了原假设。此外，还可以看一下Multiple R-squared和Adjusted R-squared的值。我们将在多变量回归部分讨论Adjusted R-squared，所以先关注Multiple R-squared，它的值为0.8709。理论上，Multiple R-squared的取值范围在0和1之间，用来表示*X*和*Y*的相关程度。在本例中，它的意义是**水量**87%的方差可以被**降雪含水量**解释。顺便说一句，R平方项就是[*X*, *Y*]的相关系数的平方。

再回到散点图。可以用下面的代码为散点图加上一条由模型产生的最佳拟合直线。

```
    > plot(content, yield)

> abline(yield.fit, lwd=3, col="red")
```

代码输出如下。



线性回归必须通过假设检验，其中的假设可以总结如下。

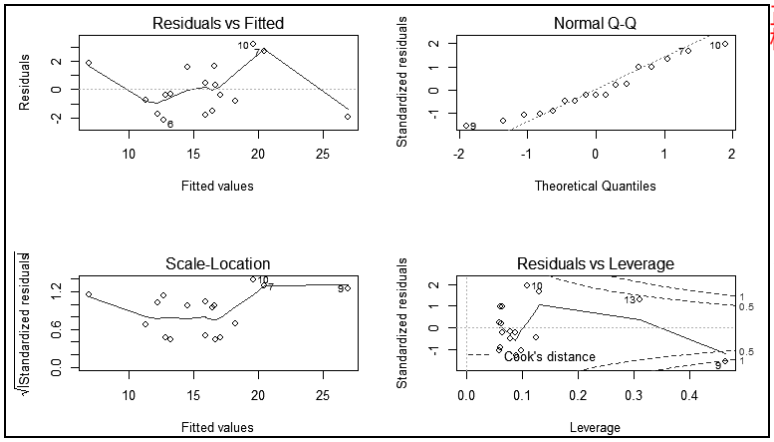
- **线性**：预测变量与响应变量之间的关系是线性的。如果线性关系不能清晰呈现，可以对变量*X*或*Y*进行数据转换（对数转换、多项式转换、指数转换等）以解决问题。
- **误差不相关**：在时间序列和面板数据中，e_n = beta$_{n-1}$是一个常见问题；如果误差是相关的，那么你就有可能建立一个非常不规范的模型。
- **同方差性**：误差是正态分布的，并具有相同的方差。这意味着对于不同的输入值，误差的方差是个固定值。如果违背了这个假设，参数估计就有可能产生偏差，导致对显著性的统计检验结果过高或者过低，从而得到错误的结论。这种情况就称为**异方差性**。
- **非共线性**：两个预测变量之间不存在线性关系，也就是说，特征之间不应该存在相关性。同样地，共线性也会导致估计偏差。 多重共线性
- **存在异常值**：异常值会严重影响参数估计。理想情况下，必须在使用线性回归拟合模型之前就除去异常值。正如我们在Anscombe数据集那个例子中看到的，异常值也会导致具有偏差的估计结果。

因为我们建立的单变量模型与时间不相关，所以只须关注线性和异方差性。在下一节中，其他假设会变得重要。对假设进行初步检验的最好方式就是画统计图。`plot()`函数结合线性模型，可以自动生成4张统计图，帮助我们进行假设检验。R会一次性生成这些图，你可以通过回车键进行切换。最好的方式是同时查看这4张图，我们通过以下方式实现：

```
    > par(mfrow = c(2,2))
> plot(yield.fit)
```

上述代码输出如下。

检查误差的同方差性



正态Q-Q图：
检查残差是否服从正态分布

残差杠杆图：
检查异常值

左边的两张图供我们检查误差的同方差性和非线性。我们期望能发现某种模式，或者更重要的是，不存在任何模式。在样本大小只有17个观测值的情况下，不会看到什么明显的模式。误差的异方差性通常会表现为U形曲线或反U形曲线，也可能会紧密聚集在图的左侧，随着拟合值的增加逐渐变宽（漏斗形）。我们完全可以断定，模型明显没有违背同方差性假设。

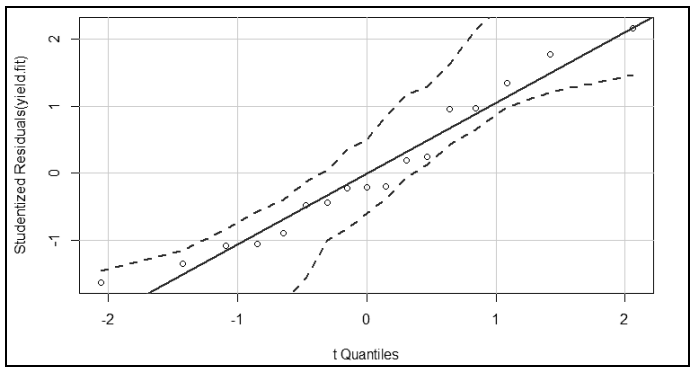
右上角的**正态Q-Q图**可以帮助我们确定残差是否服从正态分布。**分位数–分位数图**表示一个变量的分位数对应于另一个变量的分位数画出的图，从图中可以看出有离群点（第7、9、10个观测），这可能违反假设。**残差杠杆图**可以告诉我们哪个观测值（如果有）会对模型造成过度影响，换句话说，是否存在我们应该关注的异常值。鉴别强影响点的统计量是**库克距离**，一般认为，如果这个统计量的值大于1，就需要进行更深入的检查。

那么到底该如何进行呢？这既是科学，又是艺术。最简单的方法就是直接删除观测值，在本例中我们可以删除第9个观测值，然后重新构造模型。但更合理的选择是，转换预测变量或响应变量的形式。如果仅仅删除第9个观测值，那么第10个与第13个观测值就可能因为库克距离大于1而落到正常区间之外。我相信，这就是领域专家的用武之地。我无数次地发现，对异常值的研究和理解可以得到非常有价值的知识。当我们第一次查看前面的散点图时，我曾经指出几个疑似离群点，恰巧就是第9个和第13个观测值。作为一名分析师，非常重要的一点就是要经常与相关主题专家进行讨论，以弄清为什么会出现这种情况。是测量出现了错误？还是对这些观测值有更合理的解释？我当然给不出答案，但这肯定可以给你的组织带来更大价值。

这个话题先说到此，下面对现有模型做更深一步的研究，看看**正态Q-Q图**的更多细节。R在缺省的Q-Q图上没有提供置信区间，但仔细查看基础Q-Q图后，我们认为有必要检查置信区间。car包中的`qqPlot()`函数可以自动提供置信区间，因为car包与alr3包是一起加载的，所以可以用一行代码生成所需统计图，如下所示：

```
> qqPlot(yield.fit)
```

上述代码输出如下。



如图所示，残差服从正态分布。我认为，这可以使我们更有信心选择使用所有观测值拟合的模型。改进模型时，需要清晰理智的思考和判断。如果我们明确拒绝了误差正态分布假设，那么就应该考虑进行变量转换和删除观测值。

2.2　多变量线性回归

你可能正在扪心自问，在现实世界中，一个预测变量真的够用吗？这确实是一个好问题，而且一个变量在大多数情况下确实不够（时间序列通常是一个例外）。更可能的情况是，模型中不得不引入多个（如果不是许多）预测变量——或特征，机器学习中更喜欢使用这个名词。我们下面将讨论多变量线性回归，并开始一个新的商业案例。

2.2.1　业务理解

我们的主题还是水域保护和预测，使用alr3包中的另一个数据集，名字是water。2014年，美国南加利福尼亚严重的旱灾引起了很大关注，甚至连加州州长杰瑞·布朗都采取了行动，呼吁市民将用水量减少20%。在这个练习中，假设我们被加州政府委托预测水的可用性。提供给我们的数据包含43年的降雪量，收集自欧文斯山谷的6个不同地点。数据集中还有一个表示水的可用性的响应变量——加州毕绍普市附近的河川径流量，这些流量被引入欧文斯山谷的引水渠，最后流入洛杉矶引水渠。对径流量的精确预测可以使工程师、规划者和政策制定者更有效地制定水域保护措施。我们要建立的模型形式是$Y=B_0+B_1x_1+\cdots+B_nx_n+e$，预测变量（特征）可以有$1\sim n$个。

2.2.2　数据理解和数据准备

为开始这一步，我们加载名为water的数据集，并用`str()`函数查看其结构。如下所示：

```
> data(water)

> str(water)
```

```
'data.frame':   43  obs. of  8 variables:
$ Year   : int  1948  1949 1950 1951 1952 1953 1954
      1955 1956 1957 ...
$ APMAM  : num  9.13  5.28 4.2 4.6 7.15 9.7 5.02 6.7
       10.5 9.1 ...
$ APSAB  : num 3.58 4.82 3.77 4.46 4.99 5.65 1.45
       7.44 5.85 6.13 ...
$ APSLAKE: num 3.91 5.2 3.67 3.93 4.88 4.91 1.77
       6.51 3.38 4.08 ...
$ OPBPC  : num 4.1 7.55 9.52 11.14 16.34 ...
$ OPRC   : num 7.43 11.11 12.2 15.15 20.05 ...
$ OPSLAKE: num 6.47 10.26 11.35 11.13 22.81 ...
$ BSAAM  : int 54235 67567 66161 68094 107080
       67594 65356 67909 92715 70024 ...
```

我们有8个特征，其中BSAAM是响应变量。观测值始于1943年，并不间断地进行了43年。因为我们不关心观测发生在哪一年，所以应该建立一个新的数据框，去掉年份向量。这做起来非常简单，用一行代码即可建立新的数据框，然后用`head()`函数检验结果是否正确。如下所示：

```
> socal.water <- water[ ,-1] #new dataframe with
  the deletion of
  column 1

> head(socal.water)
  APMAM APSAB APSLAKE OPBPC  OPRC OPSLAKE  BSAAM
1  9.13  3.58    3.91  4.10  7.43    6.47  54235
2  5.28  4.82    5.20  7.55 11.11   10.26  67567
3  4.20  3.77    3.67  9.52 12.20   11.35  66161
4  4.60  4.46    3.93 11.14 15.15   11.13  68094
5  7.15  4.99    4.88 16.34 20.05   22.81 107080
6  9.70  5.65    4.91  8.88  8.15    7.41  67594
```

既然所有特征都是数值型的，那么就应该检查相关性方面的统计量，并绘出散点图矩阵。相关系数又称**Pearson's r**，可以用来测量两个变量之间线性相关性的强度和方向。这个统计量是一个数值，取值在–1和1之间，–1表示完全负相关，+1表示完全正相关。要计算相关系数，需要使用两个变量的协方差除以它们的标准差的乘积。前面说过，如果取相关系数的平方，就可以得到R方。

有很多方式可以做出相关性矩阵图。有些人喜欢做**热点图**，但是我强烈推荐使用corrplot包做图，它可以用很多方式表示相关系数之间的差异，包括椭圆、圆、正方形、数字、阴影、颜色和饼图。我更喜欢椭圆，当然也不反对你试试其他方式。下面加载corrplot包，使用基础的`cor()`函数建立一个相关性对象，然后看看结果。如下所示：

```
> library(corrplot)

> water.cor <- cor(socal.water)

> water.cor
  APMAM       APSAB    APSLAKE       OPBPC
```

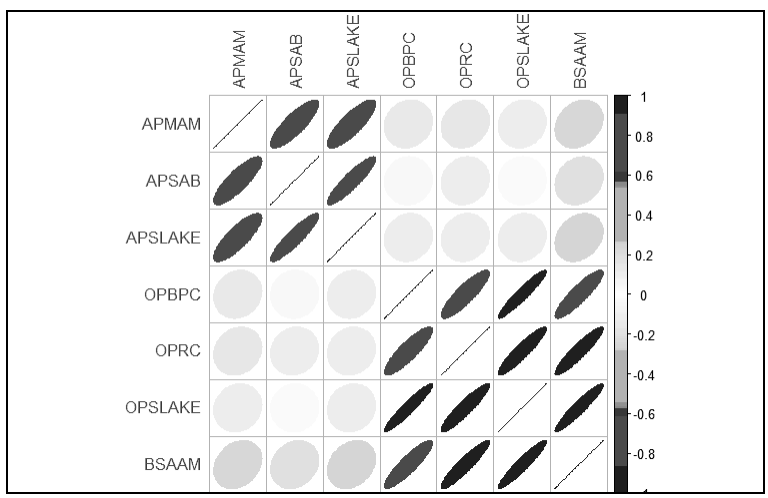
```
APMAM   1.0000000 0.82768637  0.81607595 0.12238567
APSAB   0.8276864 1.00000000  0.90030474 0.03954211
APSLAKE 0.8160760 0.90030474  1.00000000 0.09344773
OPBPC   0.1223857 0.03954211  0.09344773 1.00000000
OPRC    0.1544155 0.10563959  0.10638359 0.86470733
OPSLAKE 0.1075421 0.02961175  0.10058669 0.94334741
BSAAM   0.2385695 0.18329499  0.24934094 0.88574778
           OPRC     OPSLAKE       BSAAM
APMAM   0.1544155 0.10754212  0.2385695
APSAB   0.1056396 0.02961175  0.1832950
APSLAKE 0.1063836 0.10058669  0.2493409
OPBPC   0.8647073 0.94334741  0.8857478
OPRC    1.0000000 0.91914467  0.9196270
OPSLAKE 0.9191447 1.00000000  0.9384360
BSAAM   0.9196270 0.93843604  1.0000000
```

能看出什么？首先，响应变量与那些OP开头的特征高度正相关，与OPBPC的相关系数是0.8857，与OPRC的相关系数是0.9196，与OPSLAKE的相关系数是0.9384。还可以看出，AP开头的特征彼此之间高度相关，OP开头的也是如此。这意味着我们会遇到多重共线性的问题。相关性矩阵图可以用漂亮的可视化方式表示相关性。如下所示：

```
> corrplot(water.cor, method = "ellipse")
```

上述代码片段输出如下。

相关性矩阵图

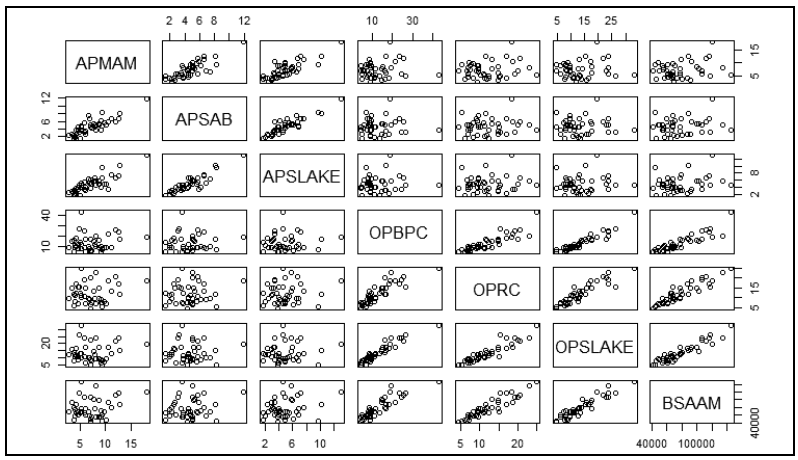


另外一种常用的可视化方式是散点图矩阵，可以通过调用pairs()函数来实现。它可以使我们看到比前面的相关性矩阵图更多的信息：

```
> pairs(~ ., data = socal.water)
```

上述代码片段输出如下。

散点图矩阵



2.2.3 模型构建与模型评价

本节要讨论的一个关键问题就是特征选择，这个任务特别重要。在本章中，我们要讨论最优子集回归和逐步回归方法，使用leaps包。后面的章节会涉及更高级的技术。

前向逐步选择从一个零特征模型开始，然后每次添加一个特征，直到所有特征添加完毕。在这个过程中，被添加的选定特征建立的模型具有最小的RSS。所以理论上，第一个选定的特征应该能最好地解释响应变量，依此类推。

添加一个特征一定会使RSS减少，使R方增加，但不一定能提高模型的拟合度和可解释性。

后向逐步回归从一个包含所有特征的模型开始，每次删除一个起最小作用的特征。现在也有一种混合方法，这种算法先通过前向逐步回归添加特征，然后检查是否有特征不再对提高模型拟合度起作用，如果有则删除。每次建模之后，分析者都可以检查模型输出，并使用各种统计量选择能提供最佳拟合的特征。

这里我要给出一个重要提示，逐步回归技术会遇到非常严重的问题。对于一个数据集，你先用前向逐步回归，然后再用后向逐步回归，可能会得到两个完全矛盾的模型。最重要的一点是，逐步回归会使回归系数发生偏离，换句话说，会使回归系数的值过大，置信区间过窄(Tibrishani，1996)。

对于特征选择，最优子集回归是逐步回归的一个可接受的替代方案。在最优子集回归中，算法使用所有可能的特征组合来拟合模型，所以，如果有3个特征，将生成7个模型。然后和逐步回归一样，分析者需要应用自己的判断和统计分析来选择最优模型，模型选择就是后面工作的关键。正如你猜想的那样，如果数据集有多个特征，工作量就会非常大。当特征数多于观测数时（p大于n），这个方法的效果就不会好。

当然，这些对于最优子集法的限制不会影响我们现在要做的工作，基于现有的限制条件，我们放弃逐步回归。当然，你完全可以试试。首先加载leaps包。为了查看特征选择如何进行，我们先用所有特征建立和检查模型，然后逐渐深入，使用最优子集法选择最优的拟合。

要想使用所有特征构建线性模型，仍然可以使用`lm()`函数。模型形式为fit = lm (y~$x_1 + x_2 + x_3 + \cdots + x_n$)。一个简单的技巧是，如果你想包括所有特征，只要在波形符号后面加一个句号，而不用把它们全打出来。对于初学者来说，我们要加载leaps包，建立一个包含所有特征的模型进行研究，如下所示：

```
    > library(leaps)

> fit <- lm(BSAAM ~ ., data = socal.water)

> summary(fit)

Call:
lm(formula = BSAAM ~ ., data = socal.water)

Residuals:
   Min     1Q Median    3Q    Max
-12690  -4936  -1424  4173  18542

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
(Intercept) 15944.67    4099.80   3.889 0.000416
      ***
APMAM         -12.77     708.89  -0.018 0.985725
APSAB        -664.41    1522.89  -0.436 0.665237
APSLAKE      2270.68    1341.29   1.693 0.099112 .
OPBPC          69.70     461.69   0.151 0.880839
OPRC         1916.45     641.36   2.988 0.005031 **
OPSLAKE      2211.58     752.69   2.938 0.005729 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
      '.' 0.1 ' ' 1
Residual standard error: 7557 on 36 degrees of
      freedom
Multiple R-squared: 0.9248,     Adjusted R-squared:
      0.9123
F-statistic: 73.82 on 6 and 36 DF, p-value: <
      2.2e-16
```

与单变量回归一样，我们要检查F统计量的p值，以检验是否至少有一个非零系数。确实，p值是高度显著的。还可以看到，OPRC和OPSLAKE这两个参数具有显著的p值。有趣的是，OPBPC并不显著，尽管它与响应变量高度相关。简言之，当我们控制其他OP开头的特征时，OPBPC无法对预测方差提供任何有意义的解释。这就是说，模型中存在OPRC和OPSLAKE时，特征OPBPC从统计学角度来看没有任何作用。

建立初始模型之后，使用最优子集法。我们使用leaps包中的`regsubsets()`函数建立一个

sub.fit对象。如下所示：

```
> sub.fit <- regsubsets(BSAAM ~ ., data =
   socal.water)
```

这样就生成了best.summary对象，帮助我们更深入地研究模型。对于R中的所有对象，都可以使用names()函数列出输出结果。如下所示：

```
> best.summary <- summary(sub.fit)

> names(best.summary)
[1] "which"  "rsq"    "rss"    "adjr2"   "cp"
       "bic"    "outmat" "obj"
```

其他对于模型选择有价值的函数还有which.min()和which.max()，它们分别给出具有某个输出的最小值和最大值的模型，如下代码片段所示：

```
> which.min(best.summary$rss)
[1] 6
```

以上代码告诉我们，有6个特征的模型具有最小的RSS。本应如此，因为它有最多的输入，输入越多，RSS越小。请注意，增加特征必然会减少RSS！而且必然会增加R方。我们可以添加一个完全不相关的特征，比如洛杉矶湖人队的胜场数，RSS也会减少，R方也会增加。变动值可能微不足道，但聊胜于无。看来，我们需要一种切实有效的方法来恰当地选择相关特征。

本章将讨论4种用于特征选择的统计方法：**赤池信息量准则**、**马洛斯的Cp**、**贝叶斯信息量准则**和修正R方。前三种方法的目标是追求统计量的值最小化，修正R方的目标是追求统计量的值最大化。这些统计方法的目的是建立一个尽可能简约的模型，换句话说，要对模型复杂性进行“惩罚”。

上述4种统计量的公式如下。

Akaike information criterion

- AIC $= n * \log\left(\frac{\mathrm{RSS}_p}{n}\right) + 2 * p$，$p$为被检验模型中的特征数量。
- CP $= \frac{\mathrm{RSS}_p}{\mathrm{MSE}_f} - n + 2 * p$，$p$为被检验模型中的特征数量，$\mathrm{MSE}_f$是包含所有特征的模型误差的平方均值（均方误差），$n$为样本大小。

Bayesian Information Criterion

- BIC $= n * \log\left(\frac{\mathrm{RSS}_p}{n}\right) + p * \log(n)$，$p$为被检验模型中的特征数量，$n$为样本大小。

Adjusted R square

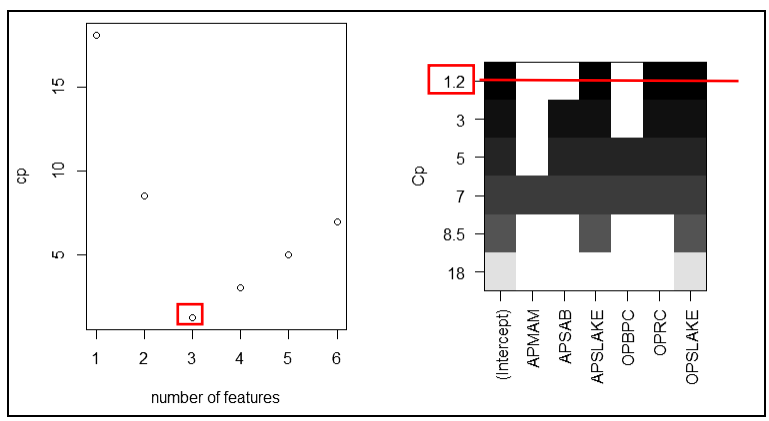
修正R方 $= 1 - \left(\frac{\mathrm{RSS}}{n-p-1}\right) / \left(\frac{\text{R方}}{n-1}\right)$，$p$为被检验模型中的特征数量，$n$为样本大小。

在线性模型中，AIC和Cp成正比，所以我们只需关注Cp，在leaps包的输出中可以找到它。

BIC与Cp相比，更倾向于选择变量较少的模型，所以我们要对二者进行比较。为此，生成两张并列的统计图进行分析。使用下面的代码片段比较Cp和BIC：

```
    > par(mfrow = c(1,2))

> plot(best.summary$cp, xlab = "number of
       features", ylab = "cp")

> plot(sub.fit, scale = "Cp")
```

上述代码片段输出如下。



在左侧的图中，可以看出有3个特征的模型具有最小的Cp值。在右侧的图中，显示了能给出最小Cp的特征组合。这张图应该这么看：先在Y轴的最高点找到最小的Cp值，此处是1.2；然后向右在X轴上找到与之对应的色块。通过这张图，我们可以看到这个具有最小Cp值的模型中的3个特征是**APSLAKE**、**OPRC**和**OPSLAKE**。通过`which.min()`和`which.max()`函数，我们可以进行Cp与BIC和修正R方的比较。

```
    > which.min(best.summary$bic)
[1] 3

> which.max(best.summary$adjr2)
[1] 3
```

可以看出，在本例中，BIC、修正R方和Cp选择的最优模型是一致的。现在，与单变量线性回归一样，我们需要检查模型并进行假设检验。正像之前做的那样，我们需要生成一个线性模型对象并检查统计图。如下所示：

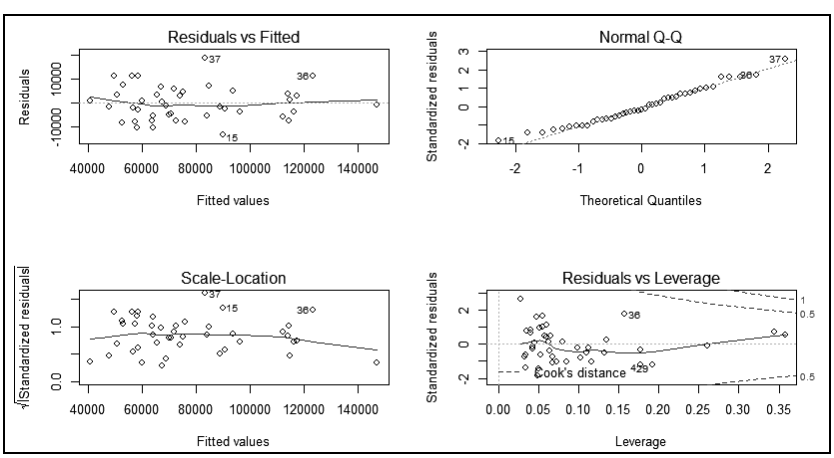
```
    > best.fit <- lm(BSAAM ~ APSLAKE + OPRC + OPSLAKE,
      data =
      socal.water)

> summary(best.fit)
```

```
Call:
lm(formula = BSAAM ~ APSLAKE + OPRC + OPSLAKE)

Residuals:
   Min     1Q Median  3Q    Max
-12964 -5140 -1252 4446 18649

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  15424.6     3638.4   4.239 0.000133
    ***
APSLAKE       1712.5       500.5  3.421 0.001475 **
OPRC          1797.5       567.8  3.166 0.002998 **
OPSLAKE       2389.8       447.1  5.346 4.19e-06
    ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05
      '.' 0.1 ' ' 1

Residual standard error: 7284 on 39 degrees of
      freedom
Multiple R-squared:  0.9244, Adjusted R-squared:
      0.9185
F-statistic: 158.9 on 3 and 39 DF, p-value: <
      2.2e-16
```

在这个三特征模型中，F统计量和所有*t*检验都具有显著的*p*值。通过了第一个检验，我们即可生成诊断图，如下所示：

```
    > par(mfrow = c(2,2))

> plot(best.fit)
```

上述代码片段输出如下。



通过这4张图，我们完全可以认为，残差具有固定的方差并且服从正态分布。杠杆图中也没有什么需要我们进一步处理的异常。

Variance inflation factor, VIF

为了研究共线性的问题，我们要引入**方差膨胀因子**这个统计量。VIF是一个比率，分子为使用全部特征拟合模型时该特征的系数的方差，分母为仅使用该特征拟合模型时这个特征的系数的方差。计算公式为$1/(1-R_i^2)$，其中R_i^2为以第i个特征作为响应变量，其余所有特征作为解释变量进行线性回归得到的R方值。VIF能取得的最小值是1，表示根本不存在共线性。现在还没有一个确定的准则决定共线性的严重程度，一般认为，VIF值超过5（有些人认为是10）就说明存在严重的共线性（James，p.101，2013）。我们难以选定一个精确的阈值，因为没有一个确定的统计学标准来决定多重共线性什么时候会使模型变得不可接受。

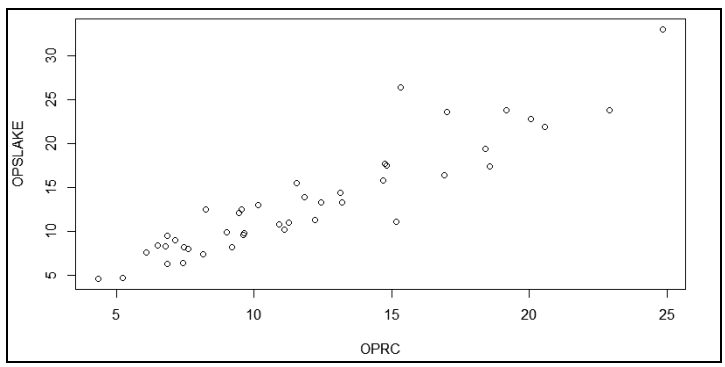
car包中的`vif()`函数完全可以计算出VIF值，参见下面的代码片段：

```
    > vif(best.fit)

APSLAKE      OPRC OPSLAKE
1.011499 6.452569 6.444748
```

基于相关性分析，我们发现OPRC和OPSLAKE存在潜在的共线性问题（VIF值大于5），这也没什么可大惊小怪的。这两个变量的关系图揭示了问题的根源，参见下面的屏幕截图。

```
> plot(socal.water$OPRC, socal.water$OPSLAKE, xlab
  = "OPRC", ylab = "OPSLAKE")
```

上述命令输出如下。



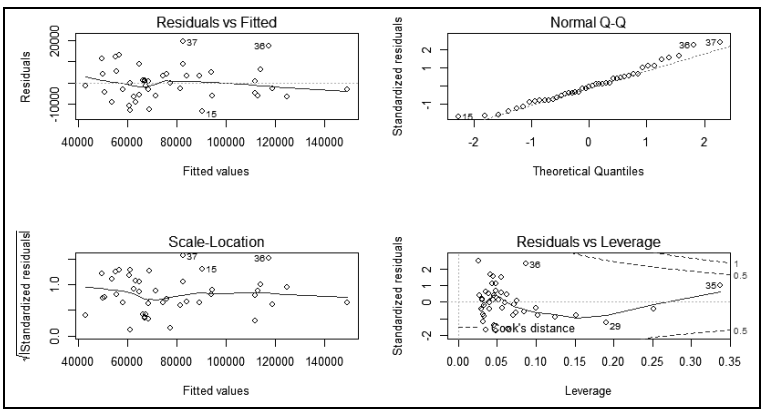
解决共线性的简单方式就是，在不影响预测能力的前提下去掉这个变量。看一下最优子集法中生成的修正R方的值就会发现，APSLAKE和OPSLAKE组成的两变量模型的值为0.90，加入OPRC之后仅有一个微不足道的提升，到了0.92。如下所示：

```
    > best.summary$adjr2 #adjusted r-squared values
[1] 0.8777515 0.9001619 0.9185369 0.9168706
      0.9146772 0.9123079
```

看一下这个两变量模型及其假设检验结果，如下所示：

```
    > fit.2 <- lm(BSAAM ~ APSLAKE+OPSLAKE, data =
      socal.water)

> summary(fit.2)

Call:
lm(formula = BSAAM ~ APSLAKE + OPSLAKE)

Residuals:
          Min        1Q  Median        3Q     Max
-13335.8  -5893.2   -171.8  4219.5  19500.2

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
(Intercept)  19144.9     3812.0   5.022  1.1e-05
    ***
APSLAKE       1768.8      553.7   3.194  0.00273 **
OPSLAKE       3689.5      196.0  18.829  < 2e-16
    ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05
      '.' 0.1 ' ' 1

Residual standard error: 8063 on 40 degrees of
      freedom
Multiple R-squared:  0.9049,    Adjusted R-squared:
      0.9002
F-statistic: 190.3 on 2 and 40 DF,  p-value: <
      2.2e-16

> par(mfrow=c(2,2))

> plot(fit.2)
```

上述代码片段输出如下。



模型是显著的，诊断图中也没发现什么问题，共线性问题应该得到了解决。再使用vif()函数检查一下，如下所示：

```
> vif(fit.2)

 APSLAKE  OPSLAKE
1.010221 1.010221
```

如前所述，我不认为残差与拟合图会有什么问题。如果你不相信，可以用R对误差的同方差性进行正式的假设检验。这个检验称为**Breusch-Pagan（BP）**检验。要想做这个检验，需要加载lmtest包，然后运行一行代码。BP检验的原假设是误差方差为0，对应的备择假设是误差方差不为0。

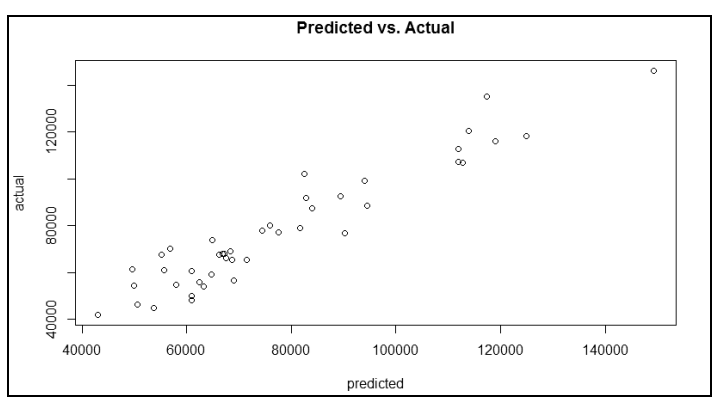
```
> library(lmtest)

> bptest(fit.2)

  studentized Breusch-Pagan test
data:  fit.2
BP = 0.0046, df = 2, p-value = 0.9977
```

我们没有证据拒绝认为“误差方差为0”的原假设，因为p值＝0.9977。检验结果中，BP＝0.0046是一个卡方值。

所有的事情都搞定了，看上去，最好的预测模型应该由APSLAKE和OPSLAKE两个特征组成。这个模型可以解释河川径流量的90%的方差。为预测流量，应该用19 145（截距）加上1769乘以ASLAKE的值，再加上3690乘以OPSLAKE的值。在R基础包中，使用模型拟合值与响应变量的值可以生成**预测值**相对于**实际值**的散点图。如下所示：model$fitted.values: 模型拟合值，也就是预测值

```
> plot(fit.2$fitted.values, socal.water$BSAAM, xlab
 = "predicted", ylab = "actual", main = "Predicted
   vs.Actual")
```

上述代码片段输出如下。



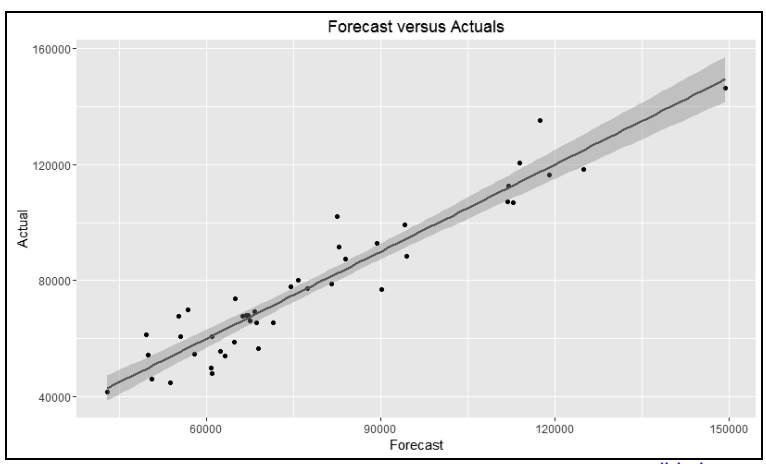
尽管很有信息量，但R的基础图形功能依然不适用于商业性展示。但在R中，我们可以轻松美化图形。现在有很多R包加强了图形功能，我们此处将使用ggplot2。生成统计图之前，必须将预测值放入数据框`socal.water`。还要把BSAAM重命名为`Actual`，并在数据框中加入一个新的向量，如下所示：

```
> socal.water["Actual"] = water$BSAAM #create the
   vector Actual

> socal.water$Forecast = predict(fit.2) #populate
   Forecast with the predicted values
```

然后加载ggplot2包，使用一行代码生成一个更漂亮的统计图：

```
> library(ggplot2)

> ggplot(socal.water, aes(x = Forecast, y =
   Actual)) + geom_point() + geom_smooth(method =
      lm) + labs(title = "Forecast versus Actuals")
```

上述代码片段输出如下。



cross-validation

继续本章内容之前，我们先讨论一项终极的模型选择技术——交叉验证。交叉验证用于模型选择和检验，这种有效方法得到了广泛应用。交叉验证为什么是必须的？这还要归因于偏差/方差的权衡问题。美国莱特州立大学的塔皮教授对此有精彩的论述：

> “我们经常使用回归模型预测未来观测值。我们用数据去拟合模型是完全可以的，但如果用来预测模型响应的数据和用来估计模型的数据属于同一批，那么说预测效果有多么好就有欺骗的嫌疑了。在评价模型对未来观测值的预测效果方面，这样做往往会得到过于乐观的结果。如果我们留下一个观测值，用其余观测值拟合模型，然后再预测这个留下的观测值，那么评价模型预测效果时，得到的结论会具有更少偏差。”

leave-one-out cross validation

塔皮教授在上述论述中提出的交叉验证技术被称为**留一法交叉验证**。在线性模型中，你可以很容易地进行LOOCV，检测**预测误差平方和**这个统计量，然后选择具有最小值的模型即可。R中的MPV库可以计算这个统计量，代码如下：

```
    > library(MPV)

> PRESS(best.fit)
 [1] 2426757258

> PRESS(fit.2)
 [1] 2992801411
```

如果仅参考这个统计量，我们应该选择模型best.fit。但如前所述，我还是认为更简约的模型更好。你可以利用简洁优雅的矩阵代数，自己编写一个简单的函数来计算这个统计量，代码如下所示：

```
    > PRESS.best = sum((resid(best.fit)/(1 -
       hatvalues(best.fit)))^2)

> PRESS.fit.2 = sum((resid(fit.2)/(1 -
       hatvalues(fit.2)))^2)

> PRESS.best
[1] 2426757258

> PRESS.fit.2
[1] 2992801411
```

你可能会问，什么是hatvalues（帽子值）？如果你有一个线性模型$Y = B_0 + B_1x + e$，我们可以将其转换为矩阵表示形式：$Y = XB + E$。在这种表示形式下，Y保持不变，X是输入值矩阵，B是系数，E代表误差。从这个线性模型可以解出B的值。不用进行繁冗的矩阵乘法运算，回归过程可以得到一个所谓**帽子矩阵**。这个矩阵将模型中计算的值映射（或称“投影”）到实际值。因此，它反映了一个特定的观测值在模型中有多大的影响力。所以，先用残差除以1减去帽子值的差，再对结果求平方和，最后就可以得到与LOOCV相同的结果。

2.3　线性模型中的其他问题

进行下一章之前，关于线性模型还有两个额外的问题需要讨论。第一个问题是如何使模型包含定性特征，第二个问题是如何处理交互项。

2.3.1　定性特征 categorical variable

定性特征经常被称为“因子”，它可以有两个或更多个水平，比如“男/女”或“差/中/好”。如果我们有一个具有两个水平的特征，比如性别，那么可以建立一个指标，或称“虚拟特征”。

dummy variable

任意地将一个水平设为0，另一个水平设为1。如果只用这个指标建立模型，那么线性模型的形式还是和前面一样，即$Y=B_0+B_1x+e$。如果对特征进行编码，使男＝0，女＝1，那么当特征为“男”时，模型的期望结果就是截距B_0；当特征为“女”时，模型的期望结果就是$B_0 + B_1x$。如果特征的水平多于两个，你就可以建立$n-1$个指标；所以3个水平需要两个指标。如果你建立的指标数和水平数相同，就会掉入虚拟变量陷阱，导致完全的多重共线性。

我们通过一个简单的例子学习如何解释输出结果。加载ISLR包，使用Carseats数据集建立一个模型，代码片段如下：

```
    > library(ISLR)

> data(Carseats)

> str(Carseats)

'data.frame':   400 obs. of 11 variables:
$ Sales       : num  9.5 11.22 10.06 7.4 4.15 ...
$ CompPrice   : num  138 111 113 117 141 124 115 136
      132 132 ...
$ Income      : num  73 48 35 100 64 113 105 81 110
      113 ...
$ Advertising: num  11 16 10 4 3 13 0 15 0 0 ...
$ Population : num  276 260 269 466 340 501 45 425
      108 131 ...
$ Price       : num  120 83 80 97 128 72 108 120 124
      124 ...
$ ShelveLoc : Factor w/ 3 levels
       "Bad","Good","Medium": 1 2 3 3 1
      1 3 2 3 3 ...
$ Age         : num 42 65 59 55 38 78 71 67 76 76
      ...
$ Education : num 17 10 12 14 13 16 15 10 10 17
      ...
$ Urban      : Factor w/ 2 levels "No","Yes": 2 2 2
      2 2 1 2 2 1 1
      ...
$ US         : Factor w/ 2 levels "No","Yes": 2 2 2
      2 1 2 1 2 1 2
      ..
```

在这个例子中，我们将预测Carseats中的sales变量，仅用一个定量特征Advertising和一个定性特征ShelveLoc。定性特征是一个因子，具有3个水平：Bad、Good和Medium。对于因子，R会在分析时自动对指标进行编码。模型的建立和分析如下所示：

```
    > sales.fit <- lm(Sales ~ Advertising + ShelveLoc,
       data = Carseats)

> summary(sales.fit)

Call:
```

```
lm(formula = Sales ~ Advertising + ShelveLoc, data =
Carseats)

Residuals:
    Min      1Q  Median      3Q     Max
-6.6480 -1.6198 -0.0476 1.5308 6.4098

Coefficients:
     Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.89662    0.25207  19.426  < 2e-
      16 ***
Advertising      0.10071    0.01692   5.951 5.88e-
      09 ***
ShelveLocGood    4.57686    0.33479  13.671  < 2e-
      16 ***
ShelveLocMedium  1.75142    0.27475   6.375 5.11e-
      10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
      '.' 0.1 ' ' 1

Residual standard error: 2.244 on 396 degrees of
      freedom
Multiple R-squared:  0.3733,    Adjusted R-squared:
      0.3685
F-statistic: 78.62 on 3 and 396 DF, p-value: <
      2.2e-16
```

截距的值是4.89662，如果货架位置好，销售量的估计值几乎是货架位置差的两倍。如果想知道R如何对指标特征进行编码，可以使用`contrasts()`函数，如下所示：

```
> contrasts(Carseats$ShelveLoc)

       Good Medium
Bad       0      0
Good      1      0
Medium    0      1
```

sales=advertising*0.10071 + ShalveLocGood*4.57686 + ShalveLocMedium*1.75142 + 4.89662

Good: sales=advertising*0.10071+1*4.57686+0+4.89662
Medium: sales=advertising*0.10071+0+1*1.75142+4.89662
Bad: sales=advertising*0.10071+0+0+4.89662

2.3.2　交互项

R中对交互项的处理也很容易。当一个特征的预测效果依赖于另一个特征的值时，这两个特征就是交互作用的。有交互项的模型可以表示为$Y = B_0 + B_1x_1 + B_2x_2 + B_1B_2x_1x_2 + e$。MASS包中提供了一个现成的例子，使用了Boston数据集。响应变量为房屋价值的中位数，用medv表示。我们用两个特征作为预测变量，一个是低社会经济地位家庭百分比，用lstat表示；一个是房龄年数，用age表示。代码及输出如下：

```
> library(MASS)

> data(Boston)
```

```
> str(Boston)

'data.frame':    506 obs. of  14 variables:
$ crim   : num   0.00632 0.02731 0.02729 0.03237
       0.06905 ...
$ zn     : num   18 0 0 0 0 0 12.5 12.5 12.5 12.5
       ...
$ Indus  : num   2.31 7.07 7.07 2.18 2.18 2.18 7.87
       7.87 7.87 7.87
      ...
$ chas   : int   0 0 0 0 0 0 0 0 0 0 ...
$ nox    : num   0.538 0.469 0.469 0.458 0.458 0.458
       0.524 0.524
       0.524 0.524 ...
$ rm     : num   6.58 6.42 7.18 7 7.15 ...
$ age    : num   65.2 78.9 61.1 45.8 54.2 58.7 66.6
       96.1 100 85.9
       ...
$ dis    : num   4.09 4.97 4.97 6.06 6.06 ...
$ rad    : int   1 2 2 3 3 3 5 5 5 5 ...
$ tax    : num   296 242 242 222 222 222 311 311 311
       311 ...
$ ptratio: num   15.3 17.8 17.8 18.7 18.7 18.7 15.2
      15.2 15.2 15.2
      ...
$ black  : num   397 397 393 395 397 ...
$ lstat  : num   4.98 9.14 4.03 2.94 5.33 ...
$ medv   : num   24 21.6 34.7 33.4 36.2 28.7 22.9
      27.1 16.5 18.9 ...
```

在lm()函数中使用feature1*feature2，将两个特征及其交互项加入模型，如下所示：

```
    > value.fit <- lm(medv ~ lstat * age, data =
      Boston)

> summary(value.fit)

Call:
lm(formula = medv ~ lstat * age, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-15.806 -4.045  -1.333   2.085  27.552

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept) 36.0885359  1.4698355  24.553   < 2e-16
      ***
lstat       -1.3921168  0.1674555  -8.313  8.78e-16
      ***
age -0.0007209 0.0198792 -0.036 0.9711
lstat:age 0.0041560 0.0018518 2.244 0.0252
      *
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
      '.' 0.1 ' ' 1

Residual standard error: 6.149 on 502 degrees of
      freedom
Multiple R-squared: 0.5557, Adjusted R-squared:
      0.5531
F-statistic: 209.3 on 3 and 502 DF, p-value: <
      2.2e-16
```

检查输出可以知道，社会经济地位是个具有高预测性的特征，而房龄则不是。但这两个变量具有显著的交互作用，可以对房屋价值进行正向解释。

2.4　小结

在机器学习的语境下，我们训练和检测模型，用来预测或预报结果。在本章中，我们深入而彻底地研究了线性回归方法，它预测定量结果，虽然简单，但极其有效。后续章节会涉及更高级的技术，但其中很多技术只是本章所学内容的扩展。我们还讨论了不对数据集进行可视化检查，而只依赖统计量进行模型选择时可能出现的问题。

通过寥寥几行代码，你就可以做出非常强大的有洞察力的预测，以支持决策。这不仅简单有效，还可以在特征中包含定量变量和交互项。在机器学习领域内深耕的每个人都绝对应该精通线性回归。