第 9 章

# 主成分分析

9

“有的人追着球跑，我则守候在球的必经之路上。”

——韦恩·格雷茨基

我们在本章依旧致力于讲解无监督学习技术。前一章介绍了聚类分析，它可以将相似的观测归成一类。在这一章，我们将研究**主成分分析**（PCA），它可以对相关变量进行归类，从而降低数据维度，提高对数据的理解。此后，我们会将主成分用于监督式学习。

在很多数据集中，特别是社会科学领域的数据集中，你会发现很多彼此相关的变量。此外，高维度也会带来问题，这就是所谓的**维数灾难**，因为模型估计所需的样本数量是随着输入特征的数量指数增长的。这种数据集中会出现某些变量冗余，因为这些变量最后起的作用与其他变量基本是重复的。比如收入水平与贫穷程度，或者抑郁度与焦虑度。那么我们的目标就是，通过PCA从原始变量集合中找出一个更小的，但是能保留原来大部分信息的变量集合。这样可以简化数据集，并经常能够发现数据背后隐藏的知识。这些新的变量（主成分）彼此高度不相关，除了可以用于监督式学习之外，还经常用于数据可视化。

我使用PCA进行分析已经有十多年了，期间的一个感受就是，PCA虽然被广泛使用，但真正理解它的人却很少，特别是在那些不做分析而只是享受成果的人当中。PCA可以直观地理解为，使用一些相关的变量构造一个新的变量。但是，由于对术语的错误理解，人们并不经常充分使用这项技术，其中的数学概念也使非专业人员无所适从。本章的目的就是说清楚PCA的概念和使用方法，包括以下内容：

- ❑ 为PCA准备好一个数据集
- ❑ 执行PCA
- ❑ 选择主成分
- ❑ 使用主成分建立一个预测模型
- ❑ 使用预测模型进行样本外预测

## 9.1　主成分简介

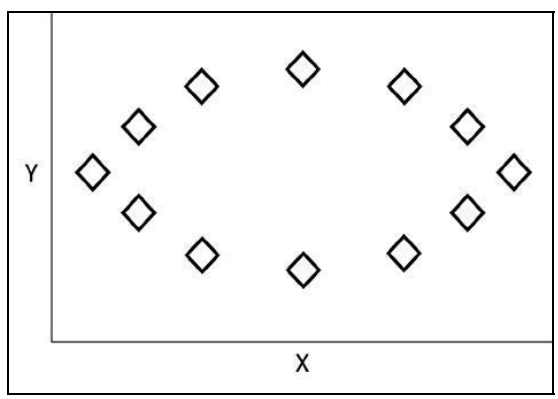
主成分分析就是寻找主成分的过程。那么，主成分到底是什么？

可以认为成分就是特征的规范化线性组合（James，2012）。在一个数据集中，第一主成分就是能够最大程度解释数据中的方差的特征线性组合。第二主成分是另一种特征线性组合，它在方向与第一主成分垂直这个限制条件下，最大程度解释数据中的方差。其后的每一个主成分（可以构造与变量数相等数目的主成分）都遵循同样的规则。

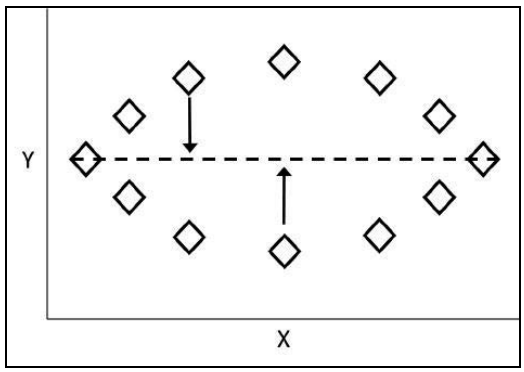
要注意两件事。PCA定义提到了**线性组合**，这是一个关键假设。如果你试图在一个变量之间基本不相关的数据集上使用PCA，很可能会得到一个毫无意义的分析结果。另外一个关键假设是，变量的均值和方差是充分统计量。也就是说，数据应该服从正态分布，这样协方差矩阵即可充分描述数据集。换言之，数据要满足**多元正态分布**。PCA对于非正态分布的数据具有相当强的鲁棒性，甚至可以和二值变量一起使用，所以结果具有很好的解释性。

robustness

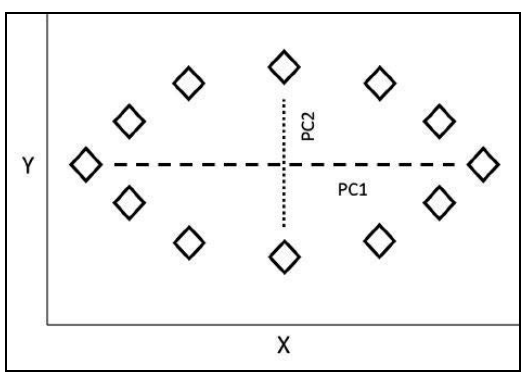
那么，这里说的“方向”是什么？如何确定特征的线性组合呢？理解PCA的最好方式就是可视化。假设有一个小数据集，其中有两个变量，我们可以画出该数据集的分布图。PCA对量度是敏感的，所以数据已经被标准化为均值为0、方差为1。你可以在下面的图中看到，每个菱形代表一个观测，数据正好组成一个椭圆形状。




从图中可知，数据在X轴方向具有最大的方差，所以我们可以画一条水平短划线来表示**第一主成分**，如下图所示。这个主成分是两个变量的线性组合，或表示为$PC_1 = \alpha_{11}X_1 + \alpha_{12}X_2$，这里的系数权重是这个主成分中的变量载荷。该主成分建立了一个基准方向，在这个方向上，数据差异最大。上面的等式中有个限制——系数的平方和为1，这是为了防止随意选择过高的值。主成分的另一种理解方式是，短划线使数据点到它本身的距离最小。随意选出两个点，用箭头表示它们到直线的距离，如下页图所示。




可以用同样的方式得出**第二主成分**，只是它应该与第一主成分不相关。也就是说，它的方向与第一主成分是垂直的。下图显示第二主成分，用点线表示。




通过为每个变量计算出主成分载荷，算法可以为我们提供主成分得分。主成分得分是对于每个主成分和每个观测计算的。对于$PC_1$和第一个观测，主成分得分公式为$Z_{11} = \alpha_{11} \times (X_{11} - X_1$的平均数$) + \alpha_{12} \times (X_{12} - X_2$的平均数$)$。对于$PC_2$和第一个观测，公式为$Z_{12} = \alpha_{21} \times (X_{11} - X_1$的平均数$) + \alpha_{22} \times (X_{12} - X_2$的平均数$)$。这些主成分得分就构成了新的特征空间，你可以使用它们进行各种分析。

我们前面说过，算法可以构造与变量数相同的主成分，这样可以解释100%的方差。那么应该如何精简主成分，来达到降低数据维度这一首要初始目标呢？可以使用一些启发式方法，在下面的建模过程中，我们介绍一种专业但是通用的方法，在**特征值**大于1的情况下选择主成分。估计特征值和**特征向量**所需的线性代数知识超过了本书范围，尽管如此，讨论它们的概念和在PCA中的应用还是很重要的。

最优线性权重是通过线性代数运算得到特征向量而求出的，它们是最优解，因为没有其他可能的权重组合可以比它们更好地解释方差。主成分的特征值是它在整个数据集中能够解释的方差的数量。

回忆一下第一主成分的计算公式是$PC_1 = \alpha_{11}X_1 + \alpha_{12}X_2$。

因为第一特征值可以解释最大数量的方差，它就有最大的特征值；第二主成分有第二大的特征值，依此类推。所以，特征值大于1就表示这个主成分解释的方差比任何一个原始变量都要大。如果通过标准化操作将特征值的总和变为1，就能够得到每个主成分解释的方差的比例。这也有助于确定一个适当的分界点。

特征值原则并不严格、明确，它必须和你的数据分析知识以及实际业务问题结合起来。如果你已经选定了主成分的数量，就可以对主成分进行旋转处理，以简化对它们的解释。

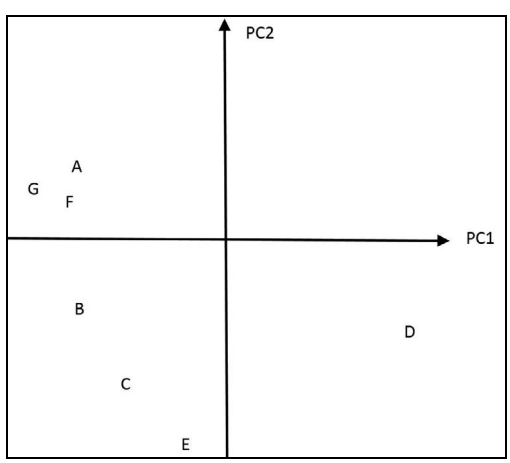
## 主成分旋转

应不应该做旋转？前面提到过，旋转可以修改每个变量的载荷，这样有助于对主成分的解释。旋转后的成分能够解释的方差总量是不变的，但是每个成分对于能够解释的方差总量的贡献会改变。在旋转过程中，你会发现载荷的值或者更远离0，或者更接近0，这在理论上可以帮助我们识别那些对主成分起重要作用的变量。这是一种将变量和唯一一个主成分联系起来的尝试。请记住，PCA是一种无监督学习，所以你是在努力去理解数据，而不是在验证某种假设。总之，旋转有助于你的这种努力。

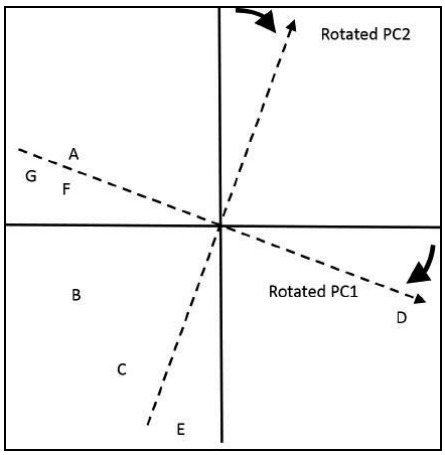
最常用的主成分旋转方法被称为**方差最大法**。虽然还有其他方法，比如**四次方最大法**和**等量最大法**，但我们主要讨论方差最大旋转。根据我的经验，其他方法从来没有提供过比方差最大法更好的解。当然，你可以通过反复实验来决定使用哪种方法。

在方差最大法中，我们要使平方后的载荷的总方差最大。方差最大化过程会旋转特征空间的轴和坐标，但不改变数据点的位置。

演示旋转过程的最好方式是通过一个简单的图示。假设一个数据集中有A~G这7个变量，有两个主成分。画出这些数据，可以得到下图。



为了便于讨论，假设变量A在PC1上的载荷是–0.4，在PC2上的载荷是0.1；变量D在PC1上的载荷是0.4，在PC2上的载荷是–0.3；对于E点，载荷分别是–0.05和–0.7。请注意，载荷的符号与主成分的方向是一致的。运行方差最大化过程，旋转后的主成分如下图所示。




下面是旋转后的PC1和PC2上的新的载荷。

- 变量A：–0.5和0.02
- 变量D：0.5和–0.3
- 变量E：0.15和–0.75

载荷发生了变化，但数据点没有变。通过这个简单的图示，我们还不能说已经简化了对主成分的解释，但可以帮助你理解主成分旋转过程中发生了什么。



## 9.2 业务理解

在这个案例中，我们开始探究体育世界，具体地说，是美国国家冰球大联盟。已经有人在棒球（参见畅销书和电影《点球成金》）和橄榄球上进行了很多研究，二者都是地道的美国式运动，并且被全世界人民所喜爱。但在我看来，没有比冰球更激动人心的运动了，可能这就是生长于北达科他州冰冻荒原才能获得的特有恩赐吧。不管怎么说，我都可以从这个分析开始我的冰球淘金之旅。

在这个分析中，我们要研究30支大联盟球队的统计数据，这个数据集中的数据是我从www.nhl.com和www.puckalytics.com这两个网站上整理的。我们的目标是建立一个模型来预测一只队伍的总积分，通过PCA建立一个输入特征空间，目的是揭示哪些因素能够造就一支顶级职业球队。首先从2015~2016赛季的数据中学习出一个模型，在这个赛季，匹兹堡企鹅队最终加冕冠军，然后，使用当前赛季至2017年2月15日为止的结果来检验这个模型的性能。数据文件的名称是nhlTrain.csv和nhltest.csv，地址为https://github.com/datameister66/data/。

NHL基于一个计分系统对球队进行排名，所以我们的预测结果就是球队每场比赛的得分。清楚NHL如何给球队奖励积分是非常重要的。与橄榄球及棒球不同，它们只计算胜场数和负场数，职业冰球对每场比赛使用下面的积分规则：

- 胜者得2分，不论是在常规时间、加时还是加时后的点球大战中获胜；
- 常规时间的负者得0分；
- 加时赛或点球大战的负者得1分，这就是所谓**负者分**。

NHL从2005年开始使用这种积分系统，这种系统并非没有争议，但它确实没有减少这项运动中优雅而又得体的力量与激情。

## 数据理解与数据准备

为了下载数据和进行后面的分析，先加载必需的程序包。在加载之前，请确保你已经安装完毕：

```
> library(ggplot2) #support scatterplot

> library(psych) #PCA package
```

假设你已经将两个.csv文件保存到工作目录，那么可以使用`read.csv()`函数读取训练数据：

```
> train <- read.csv("NHLtrain.csv")
```

使用结构函数`str()`检查数据。为了节省篇幅，我只列出函数输出的开始几行：

```
> str(train)
'data.frame': 30 obs. of 15 variables:
$ Team : Factor w/ 30 levels "Anaheim","Arizona",..: 1 2 3 4 5 6 7
  8 9 10 ...
$ ppg : num 1.26 0.95 1.13 0.99 0.94 1.05 1.26 1 0.93 1.33 ...
$ Goals_For : num 2.62 2.54 2.88 2.43 2.79 2.39 2.85 2.59 2.6 3.23
  ...
$ Goals_Against: num 2.29 2.98 2.78 2.62 3.13 2.7 2.52 2.93 3.02
  2.78 ...
```

下面需要查看变量名：

```
    > names(train)
[1] "Team" "ppg" "Goals_For" "Goals_Against" "Shots_For"
    [6] "Shots_Against" "PP_perc" "PK_perc" "CF60_pp" "CA60_sh"
    [11] "OZFOperc_pp" "Give" "Take" "hits" "blks"
```

以下是其各自的含义。

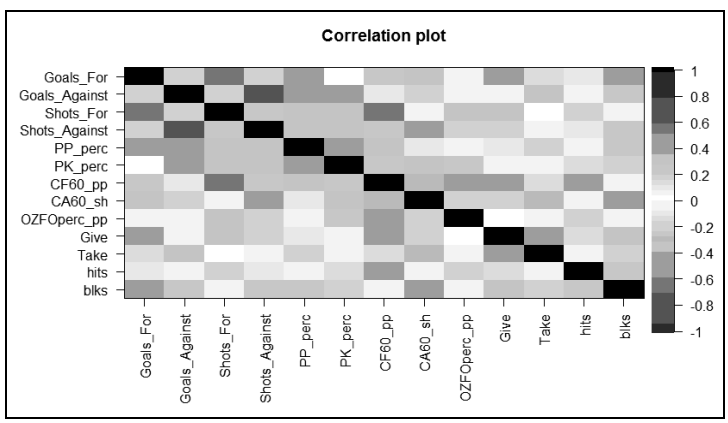
- `Team`：球队所在城市。
- `ppg`：平均每场得分，得分规则如前所述。
- `Goals_For`：平均每场进球数。

- Goals_Against：平均每场失球数。
- Shots_For：平均每场射中球门次数。
- Shots_Against：平均每场被射中球门次数。
- PP_perc：球队获得以多打少机会时的进球百分比。
- PK_perc：对方获得以多打少机会时，球队力保球门不失的时间百分比。
- CF60_pp：球队在每60分钟以多打少时间内获得的Corsi分值；Corsi分值是射门次数总和，包括射中球门次数（Shots_For）、射偏次数和被对方封堵的次数。
- CA60_sh：对方以多打少时，即本方人数劣势时，对方每60分钟获得的Corsi分值。
- OZFOperc_pp：球队以多打少时，在进攻区域发生的争球次数百分比。
- Give：平均每场丢球次数。
- Take：平均每场抢断次数。
- hits：平均每场身体冲撞次数。
- blks：平均每场封堵对方射门次数。

我们需要对数据进行标准化，使数据的均值为0，标准差为1。完成标准化后，使用psych包提供的cor.plot()函数，创建一个输入特征的相关性统计图：因为数据的量纲统一之后才有可比性

```
> train.scale <- scale(train[, -1:-2])

> nhl.cor <- cor(train.scale)

> cor.plot(nhl.cor)
```

上述命令输出如下。




可以看出一些有意思的事情。我们发现Shots_For与Goals_For相关，反之，Shots_Against与Goals_Against也相关。PP_perc及PK_perc与Goals_Against之间存在某种负相关。

由此可知，这个数据集非常适合提取主成分。

请注意，这些特征（或变量）是我根据自己的兴趣选择的，你还可以选择各种不同的统计量，看看能否提高预测能力。

## 9.3　模型构建与模型评价

对于模型构建过程，我们按照以下几个步骤进行：

(1) 抽取主成分并决定保留的数量；
(2) 对留下的主成分进行旋转；
(3) 对旋转后的解决方案进行解释；
(4) 生成各个因子的得分；
(5) 使用得分作为输入变量进行回归分析，并使用测试数据评价模型效果。

R中有许多方法和程序包可以进行主成分分析，其中最常用的是R基础包中的`prcomp()`和`princomp()`函数。但是在我看来，psych包最灵活，它有最好的选项。

### 9.3.1　主成分抽取

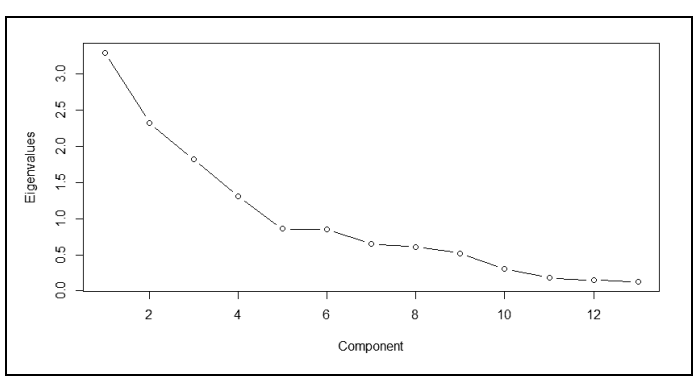
通过psych包抽取主成分要使用`principal()`函数，这个函数的语法中要包括数据和是否要进行主成分旋转：

```
> pca <- principal(train.scale, rotate="none")
```

可以调用函数生成的pca对象检查各个成分，但我们的主要目的是确定要保留的成分的数量。为此，使用碎石图即可。碎石图可以帮助你评估能解释大部分数据方差的主成分，它用X轴表示主成分的数量，用Y轴表示相应的特征值：

```
> plot(pca$values, type="b", ylab="Eigenvalues", xlab="Component")
```

上述命令输出如下。



需要在碎石图中找出使变化率降低的那个点，也就是我们常说的统计图中的“肘点”或弯曲点。在统计图中，肘点表示在这个点上新增加一个主成分时，对方差的解释增加得并不太多。换句话说，这个点就是曲线由陡变平的转折点。从这个图中可以看出，5个主成分是很令人信服的。

我从多年经验中总结出的另外一条原则是，你应该解释总数70%左右的方差。这意味着你选择的主成分解释的方差累加起来，应该能够解释70%所有成分解释的方差。

炼数成金说能解释80%以上的方差，这个模型就算可信的。每个人和书的说法不尽相同，关键看自己的取舍。

### 9.3.2 正交旋转与解释

我们在前面提到过，旋转背后的意义是使变量在某个主成分上的载荷最大化，这样可以减少（或消灭）主成分之间的相关性，有助于对主成分的解释。进行正交旋转的方法称为“方差最大法”。还有其他非正交旋转方法，这种方法允许主成分（因子）之间存在相关性。如何在实际工作中选择旋转方法需要参考相关文献，这已经超出了本书范围。你可以使用这个数据集做一些实验，但我认为，无法做出明确选择时，应该选择正交旋转作为主成分分析的起点。

要想进行正交旋转，依然要使用`principal()`函数，语法要稍作修改。我们设定使用5个主成分，并需要进行正交旋转。如下所示：

```
> pca.rotate <- principal(train.scale, nfactors = 5, rotate =
  "varimax")
```

varimax: 方差最大法

```
> pca.rotate
Principal Components Analysis
Call: principal(r = train.scale, nfactors = 5, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation
  matrix
```

RC: 每个主成分的变量载荷

```
                 RC1   RC2   RC5   RC3   RC4   h2   u2 com
Goals_For      -0.21  0.82  0.21  0.05 -0.11 0.78 0.22 1.3
Goals_Against   0.88 -0.02 -0.05  0.21  0.00 0.82 0.18 1.1
Shots_For      -0.22  0.43  0.76 -0.02 -0.10 0.81 0.19 1.8
Shots_Against   0.73 -0.02 -0.20 -0.29  0.20 0.70 0.30 1.7
PP_perc        -0.73  0.46 -0.04 -0.15  0.04 0.77 0.23 1.8
PK_perc        -0.73 -0.21  0.22 -0.03  0.10 0.64 0.36 1.4
CF60_pp        -0.20  0.12  0.71  0.24  0.29 0.69 0.31 1.9
CA60_sh         0.35  0.66 -0.25 -0.48 -0.03 0.85 0.15 2.8
OZFOperc_pp    -0.02 -0.18  0.70 -0.01  0.11 0.53 0.47 1.2
Give           -0.02  0.58  0.17  0.52  0.10 0.65 0.35 2.2
Take            0.16  0.02  0.01  0.90 -0.05 0.83 0.17 1.1
hits           -0.02 -0.01  0.27 -0.06  0.87 0.83 0.17 1.2
blks            0.19  0.63 -0.18  0.14  0.47 0.70 0.30 2.4

                       RC1  RC2   RC5  RC3  RC4
SS loadings           2.69 2.33  1.89 1.55 1.16
Proportion Var        0.21 0.18  0.15 0.12 0.09
Cumulative Var        0.21 0.39  0.53 0.65 0.74
Proportion Explained  0.28 0.24  0.20 0.16 0.12
Cumulative Proportion 0.28 0.52  0.72 0.88 1.00
```

每个主成分的特征值

对特征值进行标准化，表示每个主成分解释的方差的比例

输出中有两个部分比较重要，第一部分就是5个主成分中每个主成分的变量载荷，分别标注为RC1至RC5。我们看到，对于第一个主成分，变量Goals_Against和Shots_Against具有非常高的正载荷，而PP_perc和PK_perc具有高的负载荷。对于第二个主成分，具有高载荷的是Goals_For。第五个主成分在Shots_For、ff和OZFOperc_pp上具有高载荷。第三个主成分看上去只与变量take有关系，第四个主成分则只与hits有关。下面看一下第二个重要部分，就是以平方和SS loading开始的表格。SS loading中的值是每个主成分的特征值。如果对特征值进行标准化，就可以得到Proportion Explained行。你应该已经猜到，这一行表示的是每个主成分解释的方差的比例。可以看到，对于旋转后的5个主成分能够解释的所有方差，第一个主成分可以解释其中的28%。回忆一下前面提到的经验原则，你选择的主成分应该至少解释大约70%的全部方差。查看Cumulative Var行可以知道，这5个旋转后的主成分可以解释74%的全部方差。所以我们可以充满信心地认为，已经找到了合适数量的主成分，可以进行下一步的建模工作了。

### 9.3.3　根据主成分建立因子得分

现在检查旋转后的主成分载荷，并将其作为每个球队的因子得分。这些得分说明了每个观测（在我们的案例中是NHL球队）与旋转后的主成分的相关程度。查看得分并将其保存到数据框，因为要使用它们进行回归分析：

```
> pca.scores <- data.frame(pca.rotate$scores)

> head(pca.scores)

          RC1          RC2         RC5         RC3         RC4
1 -2.21526408  0.002821488  0.3161588 -0.1572320  1.5278033
2  0.88147630 -0.569239044 -1.2361419 -0.2703150 -0.0113224
3  0.10321189  0.481754024  1.8135052 -0.1606672  0.7346531
4 -0.06630166 -0.630676083 -0.2121434 -1.3086231  0.1541255
5  1.49662977  1.156905747 -0.3222194  0.9647145 -0.6564827
6 -0.48902169 -2.119952370  1.0456190  2.7375097 -1.3735777
```

得到每个球队在每个因子上的得分，这些得分的计算非常简单，每个观测的变量值乘以载荷然后相加即可。现在可以将响应变量（ppg）作为一列加入数据：

```
> pca.scores$ppg <- train$ppg
```

做完这项工作之后，即可开始建模预测。

### 9.3.4　回归分析 主成份回归分析

要完成这部分内容，只需重复第2章中的步骤和代码。如果你没有学习第2章，那么请回过头去学习一下，这样才能看懂下面的输出结果。

通过lm()函数建立线性模型，使用所有因子作为输入，然后查看结果摘要：
PCA就是通过对变量进行线性变换得到的

```
> nhl.lm <- lm(ppg ~ ., data = pca.scores)

> summary(nhl.lm)

Call:
lm(formula = ppg ~ ., data = pca.scores)

Residuals:
      Min        1Q   Median       3Q      Max
-0.163274 -0.048189 0.003718 0.038723 0.165905

Coefficients:
      Estimate            Std. Error t value Pr(>|t|)
(Intercept)     1.111333   0.015752  70.551  < 2e-16  ***
RC1            -0.112201   0.016022  -7.003  3.06e-07 ***
RC2             0.070991   0.016022   4.431  0.000177 ***
RC5             0.022945   0.016022   1.432  0.164996
RC3            -0.017782   0.016022  -1.110  0.278044
RC4            -0.005314   0.016022  -0.332  0.743003
---
Signif. codes: 0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1

Residual standard error: 0.08628 on 24 degrees of freedom
Multiple R-squared: 0.7502, Adjusted R-squared: 0.6981
F-statistic: 14.41 on 5 and 24 DF, p-value: 1.446e-06
```

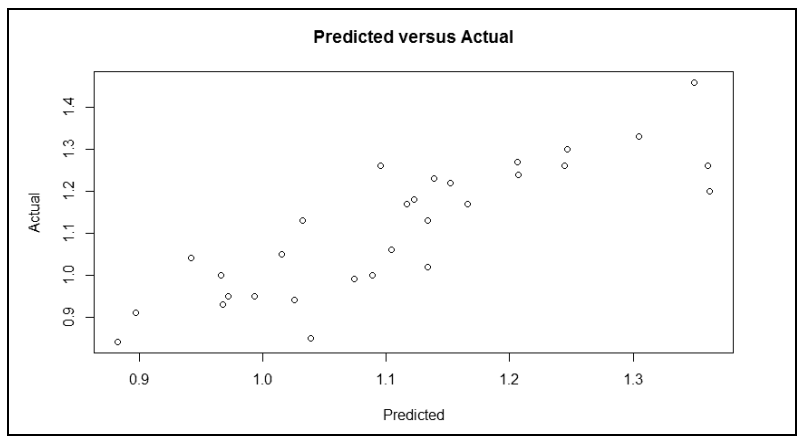
好消息是，我们的整体模型在统计上是高度显著的，*p*值为1.446e－06，修正R方几乎是70%。坏消息是，有3个主成分是不显著的。可以简单处理，选择将其保留在模型中。但是我们先看看如果把它们排除出模型，只保留RC1和RC2，会发生什么：

```
> nhl.lm2 <- lm(ppg ~ RC1 + RC2, data = pca.scores)

> summary(nhl.lm2)

Call:
lm(formula = ppg ~ RC1 + RC2, data = pca.scores)

Residuals:
     Min       1Q  Median      3Q     Max
-0.18914 -0.04430 0.01438 0.05645 0.16469

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.11133    0.01587  70.043  < 2e-16 ***
RC1        -0.11220    0.01614  -6.953  1.8e-07 ***
RC2         0.07099    0.01614   4.399  0.000153 ***
---
Signif. codes: 0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1

Residual standard error: 0.0869 on 27 degrees of freedom
Multiple R-squared: 0.7149, Adjusted R-squared: 0.6937
F-statistic: 33.85 on 2 and 27 DF, p-value: 4.397e-08
```

模型还是能得到一个几乎一样的修正R方（69.37%）值，因子的系数在统计上也是显著的。诊断测试的细节就不介绍了，代之以统计图来更加深入地进行分析。可以通过R基础包中的图形功能生成一张散点图，查看预测值和实际值（所有球队的积分）之间的关系。如下所示：

```
> plot(nhl.lm2$fitted.values, train$ppg,
  main="Predicted versus Actual",
  xlab="Predicted",ylab="Actual")
```

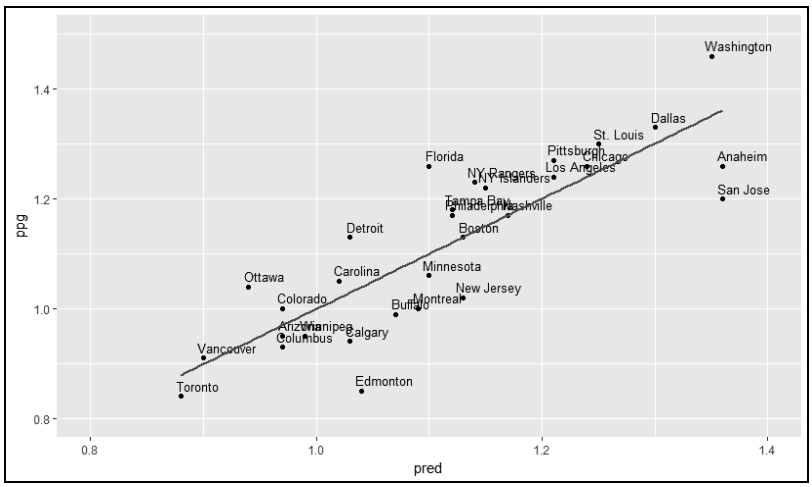
上述命令输出如下。




这张图证实，我们的模型在使用两个因子预测球队的比赛结果方面表现得非常好，它也凸显了主成分和球队积分之间存在强烈的线性相关性。再进一步，使用ggplot2包生成一张带有球队名字的散点图。唯一的问题是，这个函数功能非常强大，里面的设置非常多。有很多在线资源可以为你指点迷津，但是下面的代码可以帮助你快速实现。首先生成基准图，并将它赋给一个名为p的对象，然后添加各种绘图功能。

```
> train$pred <- round(nhl.lm2$fitted.values, digits = 2)

> p <- ggplot(train, aes(x = pred,
  y = ppg,
  label = Team))

> p + geom_point() +
  geom_text(size = 3.5, hjust = 0.1, vjust = -0.5, angle = 0) +
  xlim(0.8, 1.4) + ylim(0.8, 1.5) +
  stat_smooth(method = "lm", se = FALSE)
```

上述命令输出如下页图。




建立对象p的代码非常简单，只需指定数据框，在aes()中设定X轴和Y轴的变量，并指定使用哪个变量作为标签即可。下面将基准图美化一下，先加上数据点。在代码中使用+操作符，可以在图中加上任何需要的内容。如下所示：

```
> p + geom_point() +
```

然后设置球队标签的显示方式。需要多试几次，以确定合适的字体大小和位置：

```
geom_text() +
```

在这之后，可以设定X轴和Y轴的界限，否则统计图中就不会出现落到界限之外的那些观测点。如下所示：

```
xlim() + ylim() +
```

最后，添加一条不带标准差的最佳拟合线：

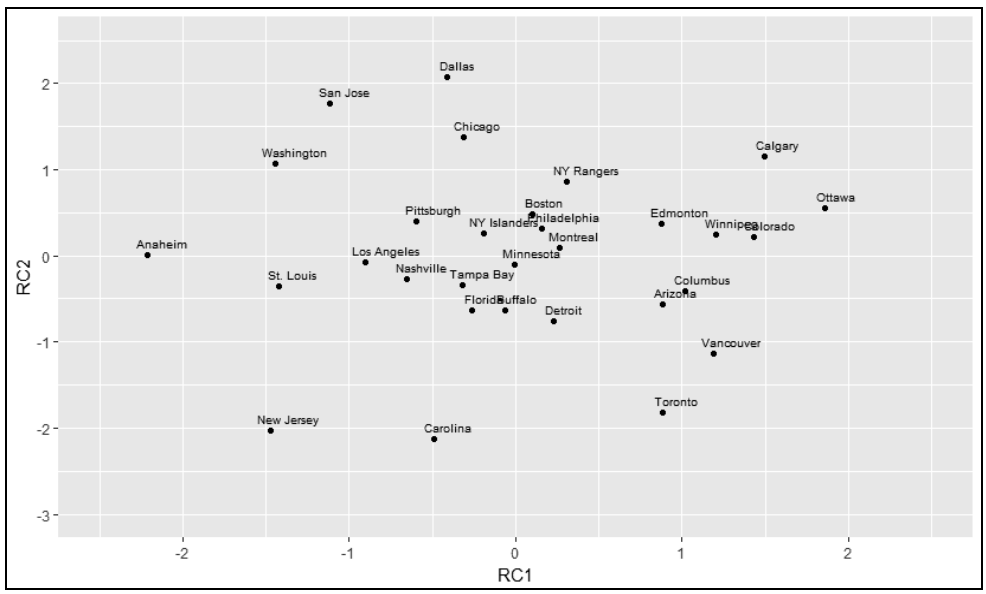
```
stat_smooth(method = "lm", se = FALSE)
```

我认为可以这样解释这张图：位于斜线下方的球队发挥欠佳，位于斜线上方的球队则超过预期。

另一项分析内容是绘制出球队及其因子得分之间的关系，这样的图称为**双标图**。依然使用ggplot()来帮助分析。以前面的示例代码为基础，修改设置并查看结果：

```
> pca.scores$Team <- train$Team

> p2 <- ggplot(pca.scores, aes(x = RC1, y = RC2, label = Team))

> p2 + geom_point() +
  geom_text(size = 2.75, hjust = .2, vjust = -0.75, angle = 0) +
  xlim(-2.5, 2.5) + ylim(-3.0, 2.5)
```

上述命令输出如下。




可以看出，X轴是球队在RC1上的得分，Y轴则是RC2上的得分。看一下“阿纳海姆小鸭队”，它在RC1上的分数最低，在RC2上的分数位于中游。考虑一下，这意味着什么。在RC1上，以多打少进球（PP_perc）和以少打多失球（PK_perc）具有负载荷，平均每场失球数（Goals_Against）具有正载荷，这说明这支球队的防守组织得非常好，并在处于人数劣势时表现得很好。顺便说一句，“匹兹堡企鹅队”最终获得了这赛季的斯坦利杯。他们的分数很实在，但也没什么出奇之处。请注意，这支球队在赛季初有个噩梦般的开始，并解雇了原来的教练。如果对他们上半赛季和下半赛季的表现做一番分析和比较，那会非常有意思。

像之前做过的那样，你还可以评价模型误差。下面看看**均方根误差**。

```
> sqrt(mean(nhl.lm2$residuals^2))
[1] 0.08244449
```

这些工作完成之后，看看这个模型在样本外数据上的效果。需要加载测试数据，通过主成分预测球队得分，然后基于线性模型做出预测。psych包中的`predict()`函数会自动对测试数据进行标准化：

```
> test <- read.csv("NHLtest.csv")

> test.scores <- data.frame(predict(pca.rotate, test[, c(-1:-2)]))

> test.scores$pred <- predict(nhl.lm2, test.scores)
```

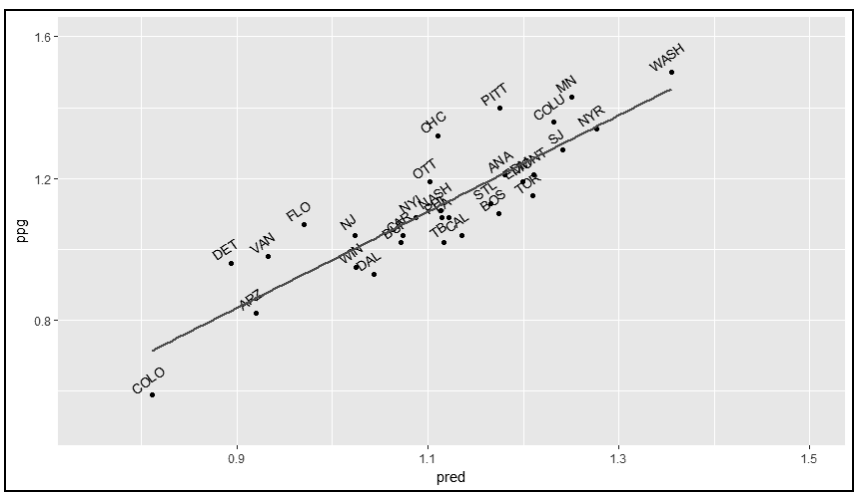
我觉得应该和前面一样，将结果绘制出来并标上球队名称。先将所有信息放在一个数据框中：

```
> test.scores$ppg <- test$ppg

> test.scores$Team <- test$Team
```

然后让ggplot()大显身手：

```
> p <- ggplot(test.scores, aes(x = pred,
                               y = ppg,
                               label = Team))

> p + geom_point() +
  geom_text(size=3.5, hjust=0.4, vjust = -0.9, angle = 35) +
  xlim(0.75, 1.5) + ylim(0.5, 1.6) +
  stat_smooth(method="lm", se=FALSE)
```

上述命令输出如下。




我对球队名称进行了缩写，使这幅图更加清晰易读。在平均每场得分上，“华盛顿首都”队一马当先，“科罗拉多雪崩”队则叨陪末座。实际上，在我采集这份数据的时候，“科罗拉多雪崩”队已经连续输掉了5场比赛，直到他们通过加时赛击败“卡罗莱纳飓风”队，才止住了连败势头。

最后，再检查一下RMSE。

```
> resid <- test.scores$ppg - test.scores$pred

> sqrt(mean(resid^2))
[1] 0.1011561
```

与样本内误差0.08比起来，0.1的样本外误差并不坏。我认为，可以宣布这个模型是有效的。但还有很多球队统计数据可以添加到模型中，以提高预测能力和减小误差。我会一直致力于完善这个模型，希望你也同样如此。

## 9.4　小结

本章再次讨论了无监督学习技术，研究了主成分分析，介绍其概念并对这种技术进行了应用。我们研究了如何使用PCA降低数据的维度和提高对数据的理解，特别当数据具有很多高度相关的变量时。然后，我们在一个来自美国国家冰球联盟的真实数据集上应用了这种技术，并使用从数据集中抽取的主成分进行了回归分析，以预测球队的得分。此外，我们还介绍了对数据和主成分进行可视化的方法。

作为一种无监督学习技术，主成分分析需要一定的判断能力以及反复实验，才能得到被商业伙伴所接受的最优解。尽管如此，主成分分析依然是一种可以挖掘潜在知识并支持监督式学习的强大技术。

下一章将介绍如何使用无监督学习技术进行购物篮分析和实现推荐引擎，PCA会在其中发挥重要作用。