

# 31

## Dimensionality Reduction in Scikit-Learn

# Scikit-Learn 降维

通过投影、旋转这两个几何视角理解主成分分析



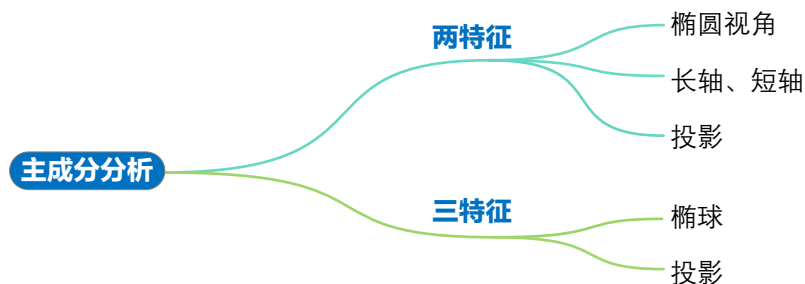
读书好比生火，每一个字都是一个火花。

*To learn to read is to light a fire; every syllable that is spelled out is a spark.*

—— 雨果 (Victor Hugo) | 法国文学家 | 1802 ~ 1885



- ◀ `sklearn.preprocessing.StandardScaler()` 用于对数据进行标准化处理
- ◀ `sklearn.decomposition.PCA()` 执行主成分分析 PCA 以减少数据维度
- ◀ `sklearn.covariance.EmpiricalCovariance()` 计算基于样本的经验协方差矩阵



# 31.1 降维

**降维** (dimensionality reduction) 是机器学习和数据分析领域中的重要概念，指的是将高维数据映射到低维空间中的过程。

在现实世界中，很多数据集都具有很高的维度，每个数据点可能包含大量特征或属性。然而，高维数据在处理和数据分析时可能会面临一些问题，例如计算复杂度增加、维度诅咒、可视化困难等。

**维度诅咒** (curse of dimensionality) 用来描述数据特征（维度）增加时，数据特征空间体积指数增大。

如图 1 所示，一个特征选取 6 个采样点，一维空间就 6 个点，二维空间有 36 ( $6^2$ ) 个点，三维空间有 216 ( $6^3$ ) 个点。

如图 2 所示，四维空间有 1296 ( $6^4$ ) 个点。而 10 个特征则达到让人恐惧的 60466176 ( $6^{10}$ ) 个点。

而降维的目标是通过保留尽可能多的信息，将高维数据投影到一个更低维的子空间，以便更有效地处理和分析数据，减少计算负担，提高模型的性能和可解释性。

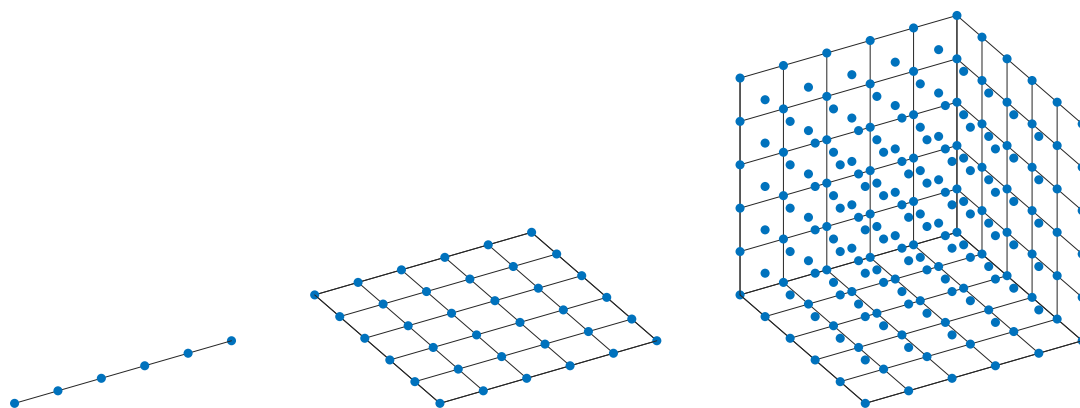


图 1. 一维、二维、三维

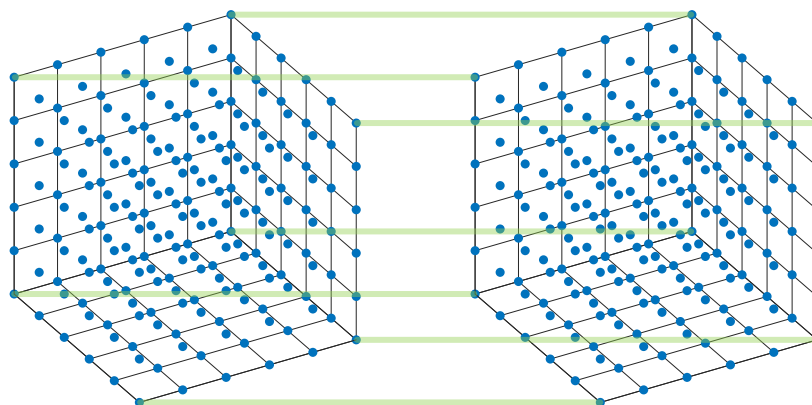


图 2. 四维

本书第 27 章介绍过主成分分析。简单来说，**主成分分析** (Principal Component Analysis, PCA) 将原始特征投影到新的正交特征空间上，以保留最大方差的特征。PCA 能够去除数据中的冗余信息，提取最重要的特征。本章还会采用几何视角继续探讨如何用 PCA 完成降维。

此外，我们也可以利用流形学习完成非线性降维。**流形学习** (manifold learning) 是一种无监督学习方法，用于在高维数据中发现潜在的低维结构。在高维空间中，数据点通常是分散的，而流形学习算法的目标是将这些分散的数据点映射到一个低维流形中，从而更好地理解数据的结构和特征。本书不展开讲解流形学习。

本书前文主要是从数据角度介绍如何使用主成分分析完成数据降维和近似还原；本章则要用几何视角和大家聊聊主成分分析，让大家深度理解主成分分析背后的思想。

➡ 当然想要真正理解主成分分析，离不开线性代数、概率统计工具，这是鸢尾花书《矩阵力量》、《统计至简》要解决的问题。

## 31.2 主成分分析

本书前文介绍过，一般情况，PCA 的基本思路是将数据投影到由主成分构成的新坐标系中，其中主成分是一组方向上方差最大的基向量。

为了方便讨论，如图 3 所示，我们先对数据进行**去均值** (demean)，即**中心化** (centralize)，处理。几何上来看，就是把数据的**质心** (centroid)  $\mu$  移动到原点  $0$ 。

此外，图 3 中椭圆和散点的关系是通过协方差矩阵联系起来的。本书前文介绍高斯分布时，大家已经建立了各种协方差矩阵和椭圆的联系。

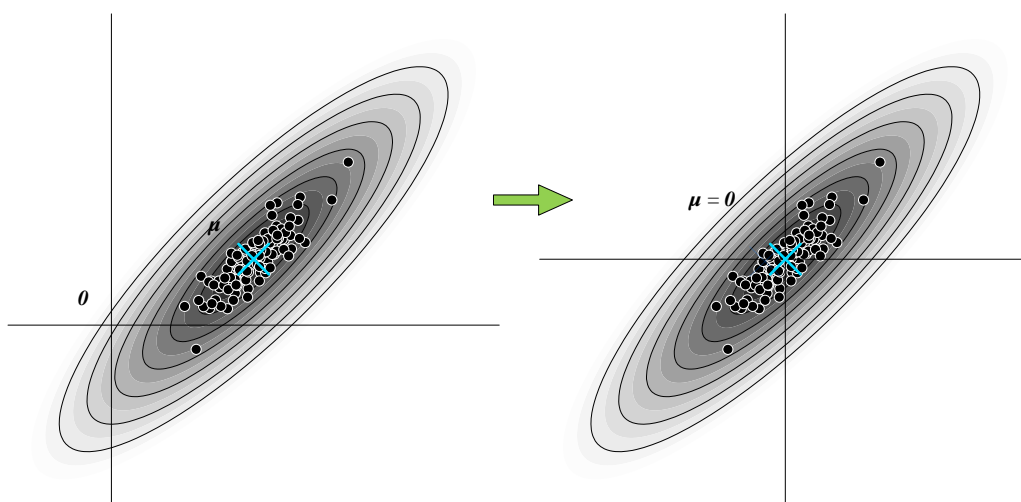


图 3. 将质心移到原点

本书前文介绍过，在进行 PCA 前一般要对数据进行**标准化** (standardization)。标准化可以消除数据不同特征尺度不同的影响，标准化过程还完成了去单位化，每个特征数据都变成了 Z 分数。

PCA 的目标是找到数据中方差最大的方向，即主成分。如果某个特征具有很大的方差，即使它在原始数据中不是最主要的特征，它在 PCA 中仍然可能成为主成分，导致降维后损失了其他重要信息。

标准化可以将所有特征的标准差调整为 1，从而避免特定特征过大方差主导问题。而标准化包含两步——**平移** (translation)、**缩放** (scaling)。其中，平移就是数据去均值，即中心化。

想要了解主成分分析，就必须理解数据**投影** (projection)。

图 4 所示为二维数据最简单的投影，分别向横轴、纵轴投影。在平面上，二维数据可以用散点图可视化。散点的横轴坐标就是数据的第一特征，散点的纵轴坐标就是数据的第二特征。

因此，图 4 的投影过程实际上就是将数据的第一、第二特征分离，然后分别计算各个特征的均值、标准差。由于数据已经中心化，各个特征的均值为 0。

我们在《矩阵力量》中会详细了解数据投影使用的数学工具。

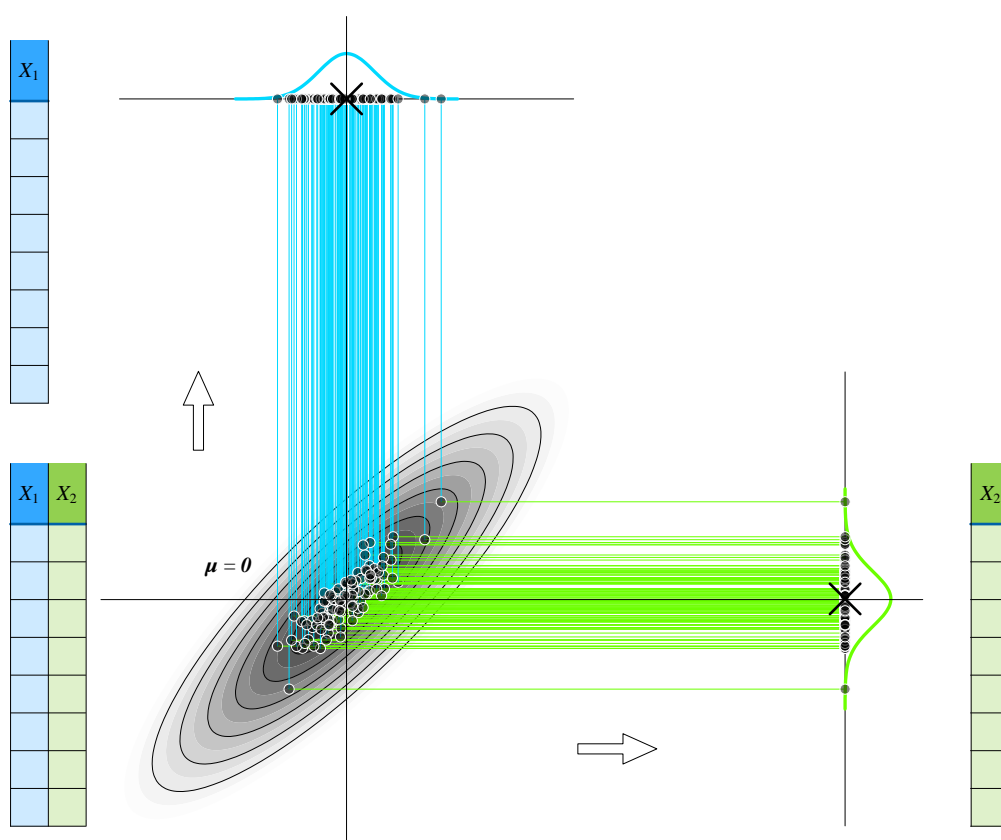


图 4. 分别向横轴、纵轴投影，并绘制一维数据分布

排版时，请尽量不要缩放此图

主成分分析的目标是将原始数据投影到一个新的坐标系中，使得投影后的数据具有最大的方差。通过这种方式，可以捕获数据中的主要变化方向，从而实现数据降维和特征提取。在进行投影时，第一个主成分的方向被选择为使投影后方差最大化的方向。

显然，图 4 所示的两个投影方向并不完美，我们可以尝试找到更好的投影方向。

如图 5 所示，平面散点朝 16 个不同方向投影，并计算投影结果的方差值。

从图 5 中每个投影结果的分佈宽度，用标准差量化，我们就可以得知  $C$ 、 $K$  这两个方向就是我们要找的第一主成分方向。

$G$ 、 $O$  这两个方向也值得我们关注，因为这两个方向上投影结果的方差（标准差的平方）最小。

➔ 鸢尾花书《可视之美》将介绍如何绘制图 5。

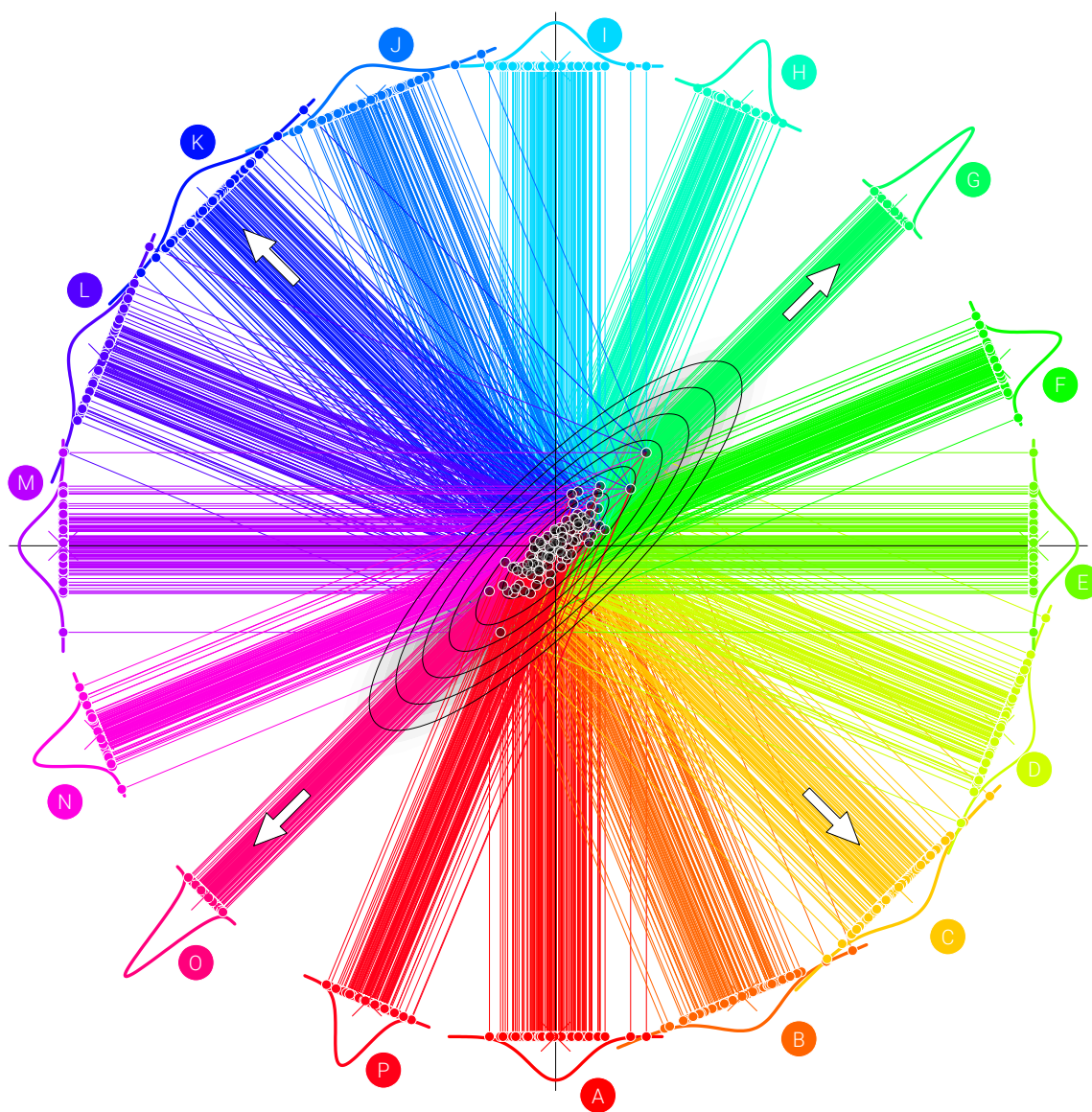


图 5. 二维数据分别朝 16 个不同方向投影

排版时，请尽量不要缩放此图

换个视角来看，如图 6 所示，主成分分析无非就是在不同的坐标系中看同一组数据。

数据朝不同方向投影会得到不同的投影结果，对应不同的分佈；朝椭圆长轴方向投影，得到的数据标准差最大；朝椭圆短轴方向投影得到的数据标准差最小。

$v_1$  对应的便是第一主成分 PC1。这里用到的几何工具就是**旋转**（rotation）。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

从椭圆的视角来看，图 6 中， $v_1$  第一主成分 PC1 方向就是椭圆长轴所在方向， $v_2$  第二主成分 PC2 方向就是椭圆短轴所在方向。显然， $v_1$  和  $v_2$  垂直！

我们管这个新的直角坐标系叫做  $[v_1, v_2]$ 。原来数据的坐标系记做  $[e_1, e_2]$ 。

图 6 的坐标系旋转也完成了旋转椭圆到正椭圆的几何转换过程。

图 7 所示为在  $[v_1, v_2]$  中看数据投影。

大家可能要问，究竟采用怎样的数学工具才能计算得到  $v_1$  和  $v_2$ ？

这就需要我们首先计算**协方差矩阵** (covariance matrix)  $\Sigma$ ，然后对协方差矩阵  $\Sigma$  进行**特征值分解** (Eigen Value Decomposition)。特征向量就是我们要找的主成分方向。

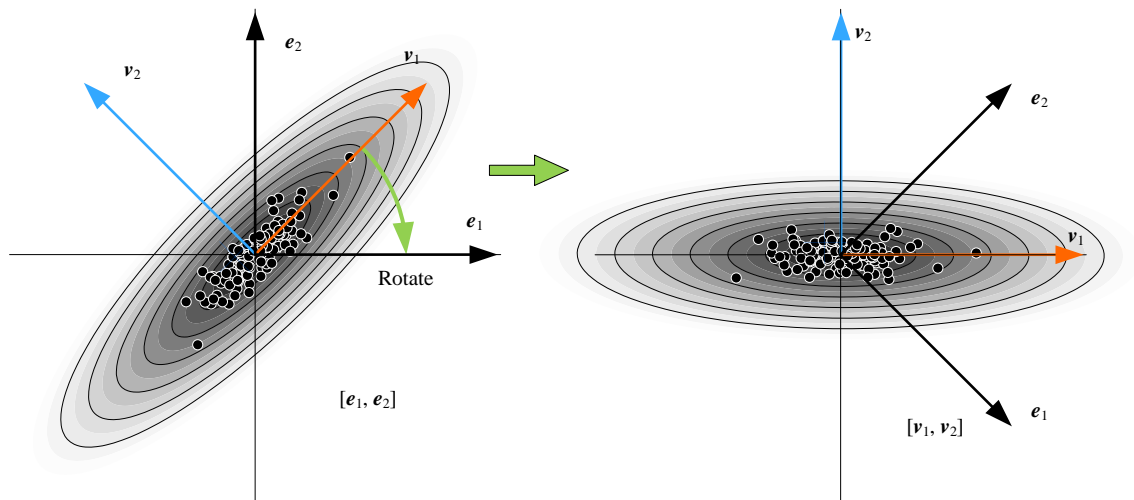


图 6. 坐标系旋转

排版时，请尽量不要缩放此图

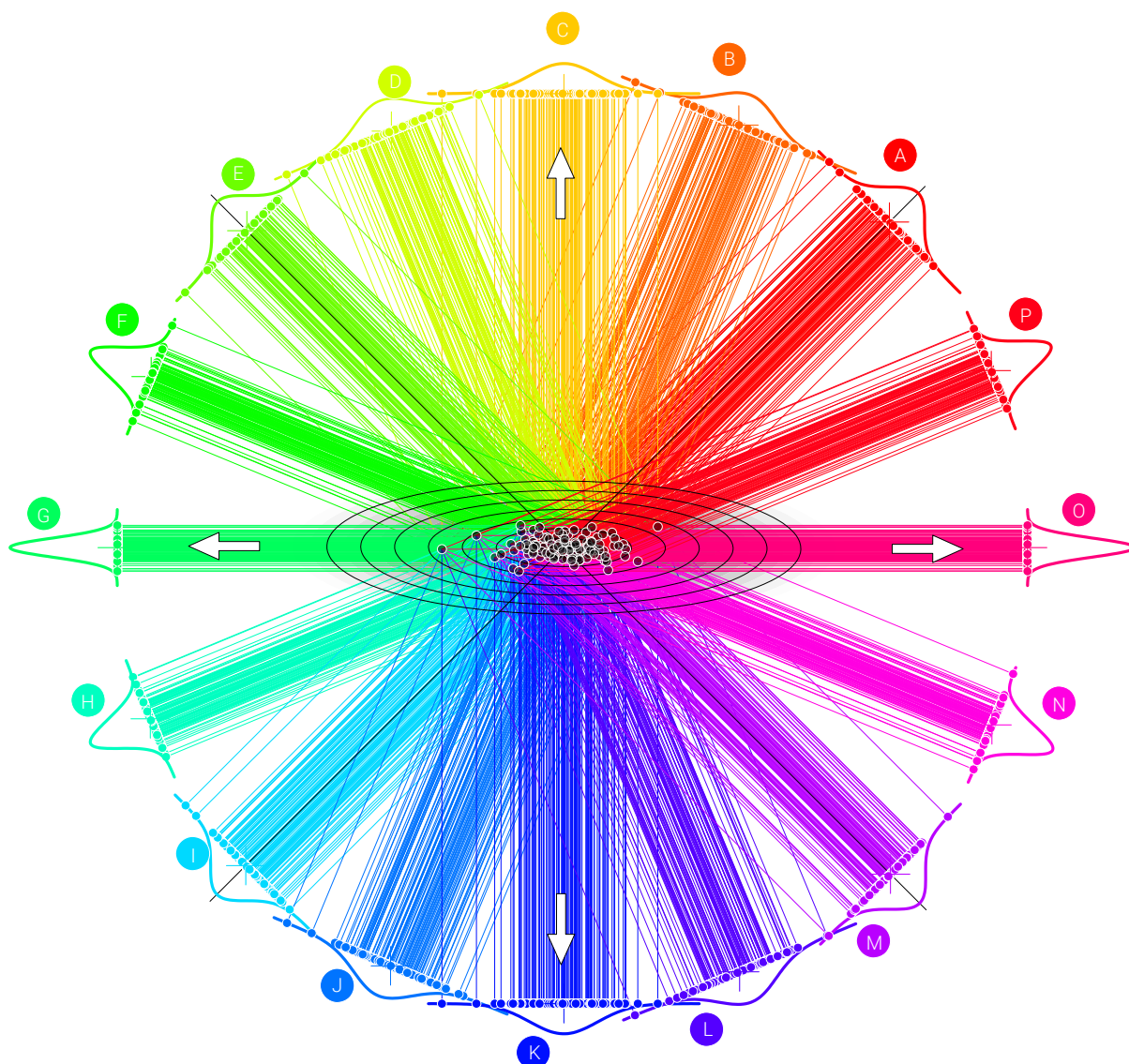


图 7. 换个坐标系看投影  
排版时，请尽量不要缩放此图

此外，除了特征值分解协方差矩阵，还有其他不同的主成分分析技术路线。鸢尾花书《数据有道》会专门比较不同技术路线的异同。

虽然，我们不会具体介绍计算协方差、特征值分解背后的数学工具，以及这两个工具和椭圆的联系；但是大家可能已经发现，想要深入理解主成分分析，离不开概率统计、线性代数、几何这些视角。这都是鸢尾花“数学三剑客”要介绍的内容。

在主成分分析中，主成分通常是原始特征的线性组合。也就是说，PCA 是一种线性降维方法，它只能捕捉数据中的线性相关性。如果数据具有复杂的非线性关系，PCA 可能无法很好地捕捉这些模式，从而导致信息丢失。

**核主成分分析** (Kernel Principal Component Analysis), 也叫核 PCA, 在高维特征空间中使用**核技巧** (kernel trick) 来进行 PCA, 从而能够处理非线性关系。

核 PCA 可以解决传统 PCA 无法处理的非线性问题。在处理非线性数据时, 传统 PCA 可能会损失数据的重要信息, 因为它只能发现线性关系。

核 PCA 通过将数据映射到高维特征空间, 将数据从原始空间中的非线性关系转化为高维空间中的线性关系, 因此可以有效地保留数据的非线性结构信息。

与传统的主成分分析不同, 核 PCA 不直接使用原始数据来计算主成分, 而是通过将数据映射到高维特征空间来获取主成分。核技巧的基本思想是通过**核函数** (kernel function) 将数据映射到高维特征空间中, 从而使得线性模型能够处理非线性数据。

常用的核函数包括, **径向基核函数** (radial basis function kernel, RBF kernel), 也叫高斯核函数, **多项式核** (polynomial kernel), **Sigmoid 核** (Sigmoid kernel)。我们在本书第 32 章讲解**支持向量机** (Support Vector Machine, SVM) 还会用到核技巧。本书不展开讲解核主成分分析。

下面, 我们还是利用本书前文用过的利率数据, 用几何视角 (投影、旋转) 和 Scikit-Learn 函数, 和大家分别聊聊两特征、三特征主成分分析。

## 31.2 两特征 PCA

代码 1 首先还是导入利率数据。这部分内容大家已经在本书前文用过, 下面简单介绍。

**a** 中, `pandas_datareader` 是一个用于从各种数据源中获取金融和经济数据的 Python 库。大家在使用前, 需要用 `pip install pandas_datareader` 安装库, 大家可以回顾本书第 1 章如何安装库。

通常, `pandas_datareader` 用于从互联网上的各种金融数据提供商获取数据, 例如股票市场数据、货币汇率、股票指数、债券价格等。类似前文, 如 **b**, 我们下文利用 `pandas_datareader` 从 FRED 下载半年期、一年期利率历史数据。

- c** 修改数据帧列标题。
- d** 计算利率日收益率。
- e** 删除数据帧的缺失值。
- f** 对数据进行标准化。

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
a import pandas_datareader as pdr
# 需要先安装库 pip install pandas_datareader
import seaborn as sns

# 下载数据，两个 tenors
b df = pdr.data.DataReader(['DGS6M0', 'DGS1'],
                           data_source='fred',
                           start='01-01-2022',
                           end='12-31-2022')

df = df.dropna()

# 修改数据帧的 column names
c df = df.rename(columns={'DGS6M0': 'X1',
                          'DGS1': 'X2'})

# 计算日收益率
d X_df = df.pct_change()
# 删除缺失值
e X_df = X_df.dropna()
# 数据标准化
scaler = StandardScaler()
f X_scaled = scaler.fit_transform(X_df)

```

代码 1. 导入利率历史数据 | Bk1\_Ch31\_01.ipynb

图 8 所示为标准化数据的散点图。在这幅图上，我们还用椭圆代表数据的分布；更准确地说，这些椭圆代表了数据的协方差矩阵。这些椭圆等高线实际上是**马氏距离** (Mahalanobis distance)。

与**欧氏距离** (Euclidean distance) 不同，马氏距离考虑了数据之间的协方差结构，因此可以更准确地捕捉数据的相关性和分布情况。图 8 这些同心椭圆就是马氏距离的等距线。



鸢尾花书《矩阵力量》《统计至简》会从不同角度介绍马氏距离背后的数学工具。

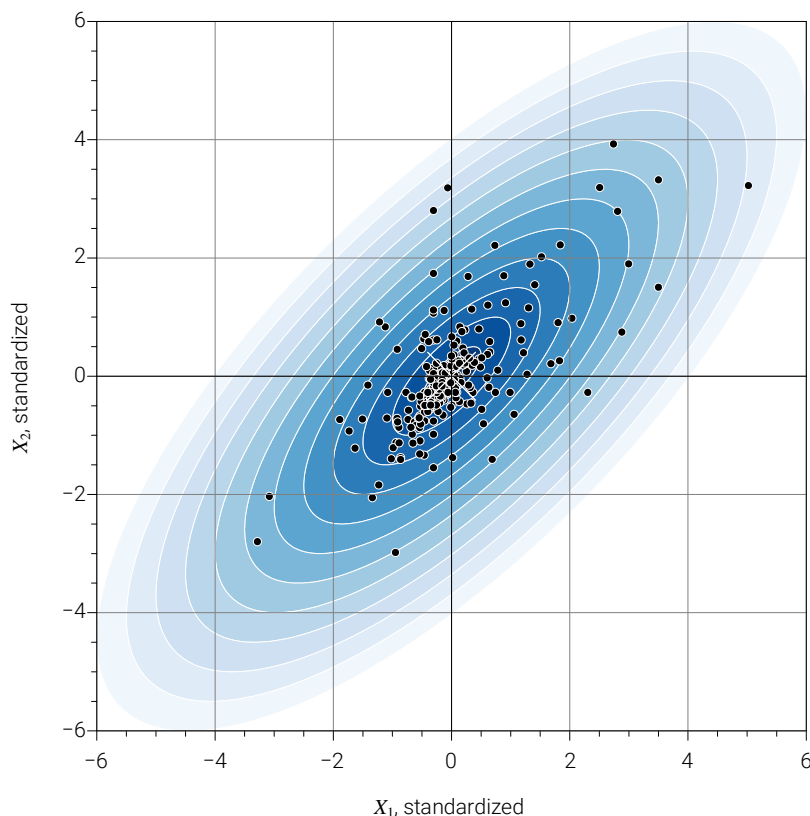


图 8. 标准化数据的散点图 |  Bk1\_Ch31\_01.ipynb

代码 2 中 <sup>a</sup> 从 Scikit-Learn 机器学习库中导入 `EmpiricalCovariance` 类。这个类是 Scikit-Learn 中用于计算数据集的经验协方差矩阵。

<sup>b</sup> 生成网格化数据，用来可视化马氏距离等高线。

<sup>c</sup> 中用 `EmpiricalCovariance` 的 `fit` 方法接受标准化数据集 `X_scaled` 作为参数，并使用这个数据集来拟合估计器，从而计算出协方差矩阵。然后，大家可以用 `COV.covariance_` 获得协方差矩阵的具体值。大家会发现，协方差矩阵对角线元素均为 1，请大家思考为什么？

<sup>d</sup> 根据样本协方差矩阵计算网格化数据的马氏距离平方值。这里需要大家格外注意，网格数据点应该与原始数据集 `X_scaled` 具有相同的特征维度（两列）。这就是为什么我们需要用 <sup>e</sup> 调整马数组形状，以便后续可视化。

此外，大家需要注意，输出的结果为马氏距离的平方。<sup>f</sup> 开平方后获得马氏距离。

<sup>g</sup> 绘制马氏距离填充等高线。大家会发现这些等高线都是椭圆，而且椭圆的半长轴和横轴夹角为 45 度。大家需要《矩阵力量》《统计至简》的数学工具才能理解为什么夹角为 45 度。

<sup>h</sup> 用散点可视化标准化样本数据。这些样本数据的质心位于原点  $(0, 0)$ 。

```

a from sklearn.covariance import EmpiricalCovariance
x1_array = np.linspace(-6,6,601)
x2_array = np.linspace(-6,6,601)
b xx1, xx2 = np.meshgrid(x1_array, x2_array)
xx12 = np.c_[xx1.ravel(), xx2.ravel()]
# 加载学习样本数据
c COV = EmpiricalCovariance().fit(X_scaled)
# 计算网格化数据的马氏距离
d mahal_sq_Xc = COV.mahalanobis(xx12)
e mahal_sq_dd = mahal_sq_Xc.reshape(xx1.shape)
f mahal_dd = np.sqrt(mahal_sq_dd)

fig, ax = plt.subplots()
# 绘制马氏距离填充等高线
g plt.contourf(xx1, xx2, mahal_dd,
               cmap='Blues_r', levels=np.linspace(0,6,13))
# 绘制样本数据（标准化）散点图
h plt.scatter(X_scaled[:,0], X_scaled[:,1],
              s = 38, edgecolor = 'w', alpha = 0.5,
              marker = '.', color = 'k')
# 绘制样本数据质心
plt.plot(X_scaled[:,0].mean(), X_scaled[:,1].mean(),
         marker = 'x', color = 'k', markersize = 18)

ax.axvline(x = 0, c = 'k'); ax.axhline(y = 0, c = 'k')
ax.grid('off'); ax.set_aspect('equal', adjustable='box')
ax.set_xbound(lower = -6, upper = 6)
ax.set_ybound(lower = -6, upper = 6)

```

代码 2. 马氏距离等高线，使用时配合前文代码 | Bk1\_Ch31\_01.ipynb

下面利用 Scikit-Learn 中的主成分分析工具完成样本数据的 PCA 分析。

代码 3 中 **a** 从 Scikit-learn 库中导入 PCA (Principal Component Analysis) 类。

**b** 创建了一个 PCA 对象的实例，并且指定了降维后的维度为 2。本例中，样本数据的特征数（维度）为 2，PCA 分析前后维度不变。

**c** 在 PCA 对象上拟合（训练）样本数据。这个过程会计算数据的协方差矩阵，然后找到主成分方向。

**d** 用属性 `components_` 获得 PCA 主成分的**载荷** (loadings)，这个矩阵的每一行代表一个主成分方向。矩阵经过**转置** (transpose) 后，每一列代表一个主成分。本书前文提过，这些主成分向量是本质上是原始特征数据的线性组合。我们把这个转置后的矩阵记做  $V$ 。

**e** 计算  $V^T @ V$ ，大家可以发现结果近似为  $2 \times 2$  **单位矩阵** (identity matrix)  $I$ 。

**f** 计算  $V @ V^T$ ，可以发现结果同样近似为  $2 \times 2$  单位矩阵  $I$ 。满足以上两个条件的矩阵  $V$  叫做**正交矩阵** (orthogonal matrix)，这是《矩阵力量》要讲解的重要概念之一。

**g** 取出矩阵  $V$  的第 1 列  $v_1$ ，即第一主成分方向。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

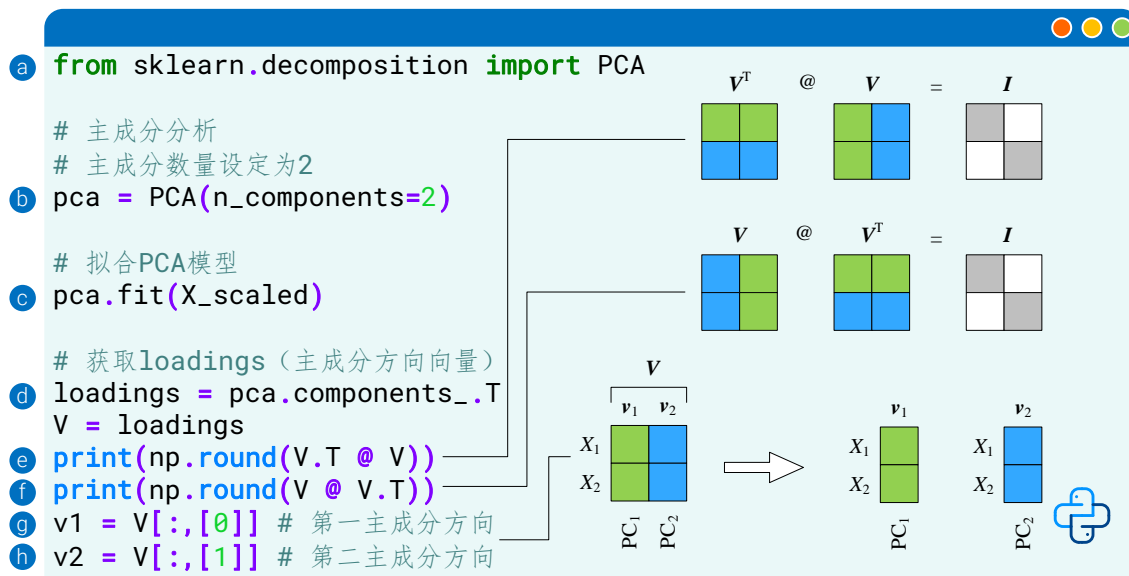
版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

h 取出矩阵  $V$  的第 2 列  $v_2$ ，即第二主成分方向。



代码 3. 主成分分析，使用时配合前文代码 | Bk1\_Ch31\_01.ipynb

图 9 展示了数据的主成分方向。容易发现， $v_1$  对应椭圆的长轴方向， $v_2$  对应椭圆的短轴方向。代码 4 在前文可视化基础上又可可视化了两个主成分方向。

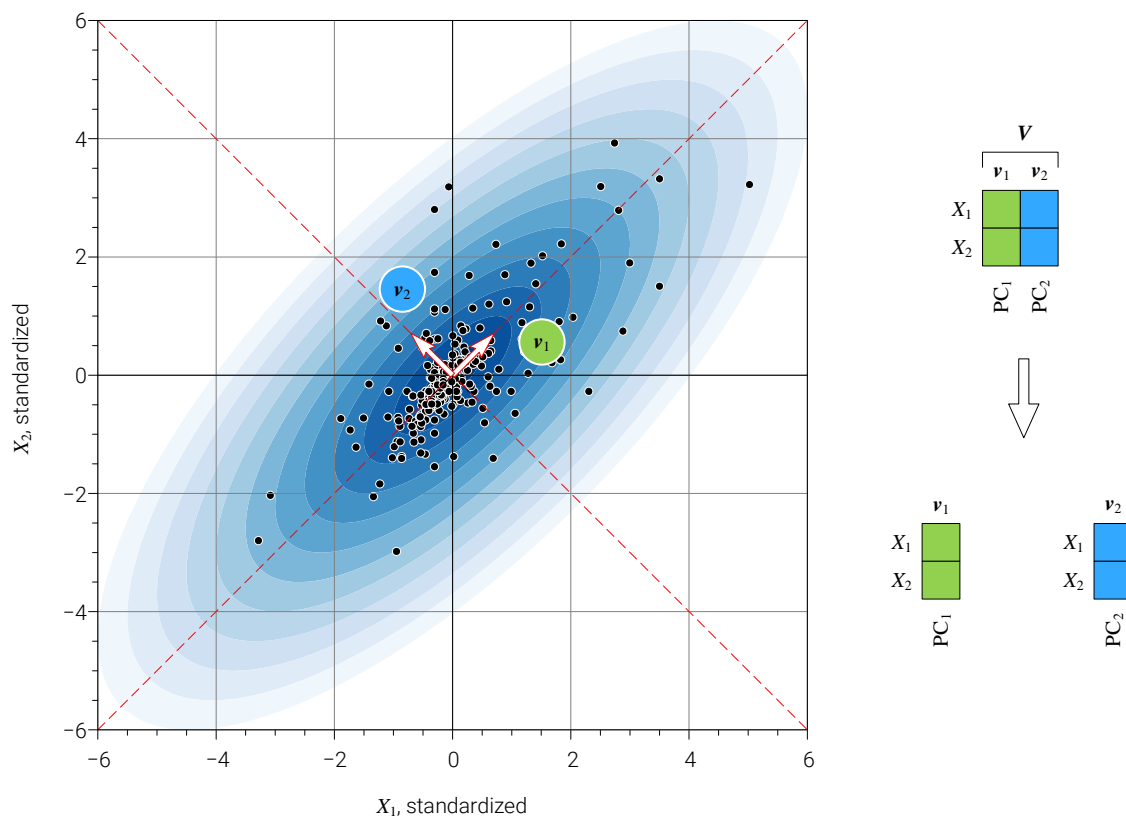


图 9. 主成分方向 |  Bk1\_Ch31\_01.ipynb

```

# 自定义绘制向量函数
a def draw_vector(vector, RGB):
    array = np.array([[0, 0, vector[0], vector[1]]], dtype=object)
    b X, Y, U, V = zip(*array)
    c plt.quiver(X, Y, U, V, angles='xy',
               scale_units='xy', scale=1, color = RGB,
               zorder = 1e5)

fig, ax = plt.subplots()
# 绘制马氏距离等高线
plt.contourf(xx1, xx2, mahal_dd,
             cmap='Blues_r', levels=np.linspace(0,6,13))
# 绘制标准化数据散点图
plt.scatter(X_scaled[:,0], X_scaled[:,1],
           s = 38, edgecolor = 'w', alpha = 0.5,
           marker = '.', color = 'k')
# 绘制质心
plt.plot(X_scaled[:,0].mean(), X_scaled[:,1].mean(),
        marker = 'x', color = 'k', markersize = 18)
# 可视化两个主成分方向
d draw_vector(v1, 'r')
e draw_vector(v2, 'r')

# 绘制两条参考线
f ax.plot(x1_array, x1_array*v1[1]/v1[0], 'r',
        lw = 0.25, ls = 'dashed')
g ax.plot(x1_array, x1_array*v2[1]/v2[0], 'r',
        lw = 0.25, ls = 'dashed')

ax.axvline(x = 0, c = 'k'); ax.axhline(y = 0, c = 'k')
ax.grid('off')
ax.set_aspect('equal', adjustable='box')
ax.set_xbound(lower = -6, upper = 6)
ax.set_ybound(lower = -6, upper = 6)

```

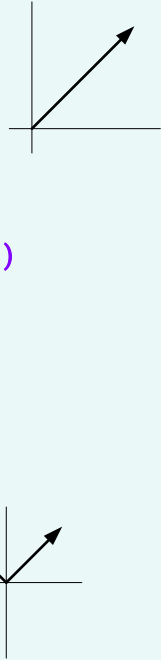


代码 4. 绘制主成分方向，使用时配合前文代码 |  Bk1\_Ch31\_01.ipynb

图 10 所示为数据朝第一主成分方向  $v_1$  投影的结果。根据前文介绍的内容，大家应该清楚朝  $v_1$  投影的得到结果的方差最大。图 11 所示为数据朝第一主成分方向  $v_2$  投影的结果，对应方差最小。

$[v_1, v_2]$  本身也是一个直角坐标系，在  $[v_1, v_2]$  中看到的数据如图 12 所示。绘制这三幅图的代码，请大家参考本章配套文件。

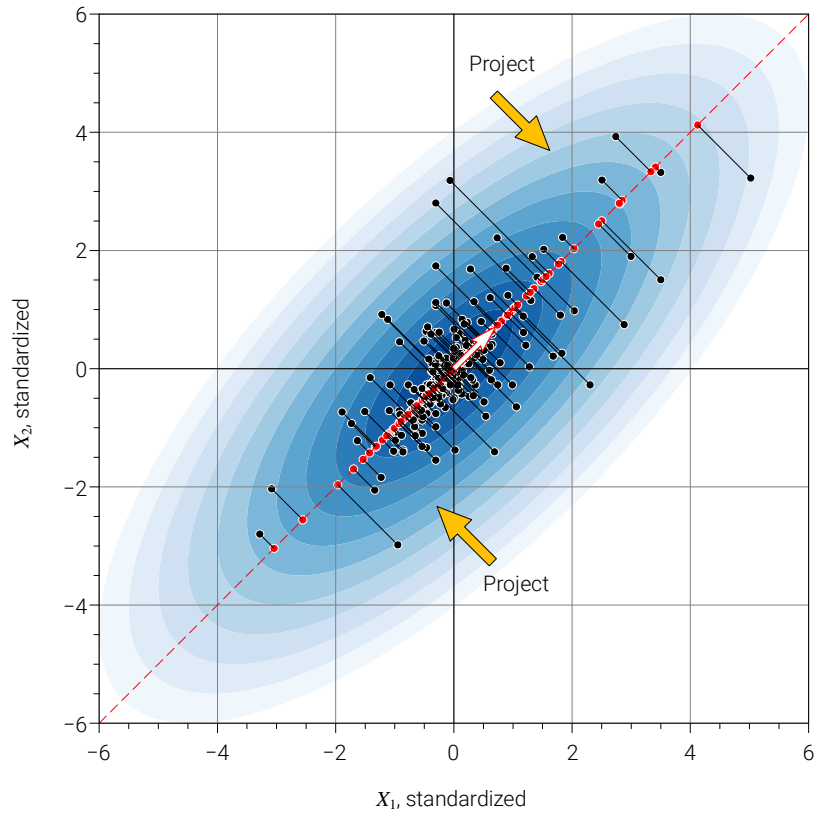


图 10. 朝第一主成分方向投影 | [Bk1\\_Ch31\\_01.ipynb](#)

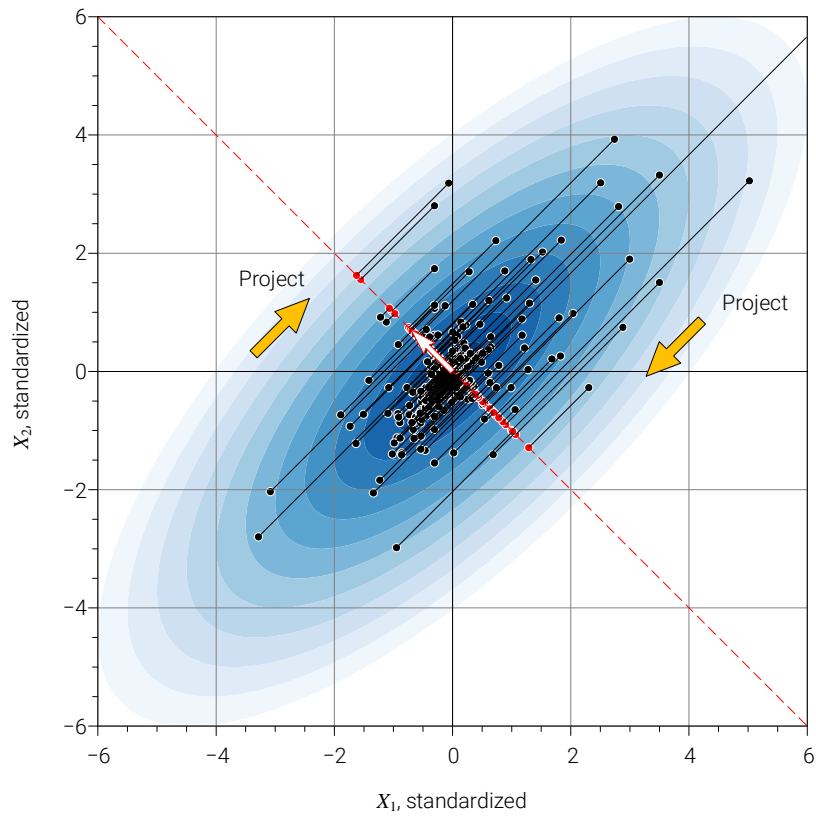
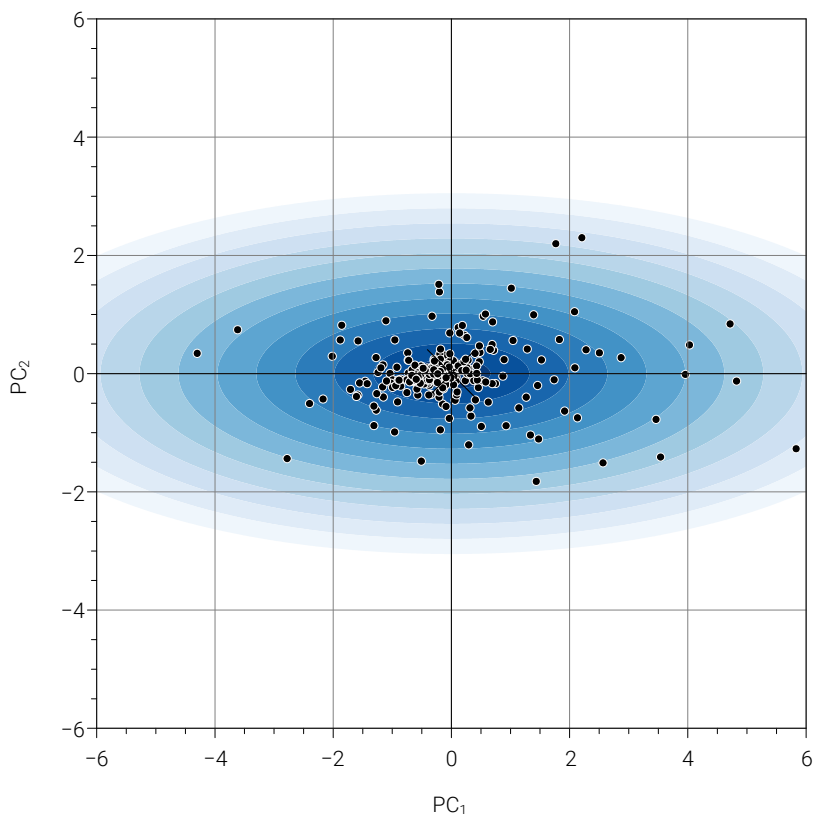


图 11. 朝第二主成分方向投影 |  Bk1\_Ch31\_01.ipynb图 12.  $[v_1, v_2]$  中看数据散点 |  Bk1\_Ch31\_01.ipynb

## 31.4 三特征 PCA

既然，我们可以用一个旋转椭圆代替二维散点图；这一节，我们则把三维散点抽象成一个椭球。

图 13 所示为在直角坐标系  $[e_1, e_2, e_3]$  中看椭球。显然这是一个旋转椭球。红色箭头  $v_1$ 、绿色箭头  $v_2$ 、蓝色箭头  $v_3$  分别指向了椭球的三个主轴方向。这三个方向也就是主成分分析中三个主成分方向。

主成分分解得到的载荷矩阵  $V$  的每一个列依次对应红色箭头  $v_1$ 、绿色箭头  $v_2$ 、蓝色箭头  $v_3$ 。 $[v_1, v_2, v_3]$  也是一个三维直角坐标系。数据在  $v_1$  上投影结果的方差最大，在  $v_2$  上投影结果的方差次之，在  $v_3$  上投影结果的方差最小。

图 14 所示为在平面直角坐标系  $[e_1, e_2]$  中看椭球。也就是说，椭球在  $[e_1, e_2]$  投影为旋转椭圆。图 14 这个椭圆就是图 8 中马氏距离为 1 的椭圆。

图 14 还展示了红色箭头  $v_1$ 、绿色箭头  $v_2$ 、蓝色箭头  $v_3$  在  $[e_1, e_2]$  中的投影。

图 15 所示为在平面直角坐标系  $[e_1, e_3]$  中看椭球。

图 16 所示为在平面直角坐标系  $[e_2, e_3]$  中看椭球。

➡ 鸢尾花书《可视之美》将专门介绍这种可视化方案。

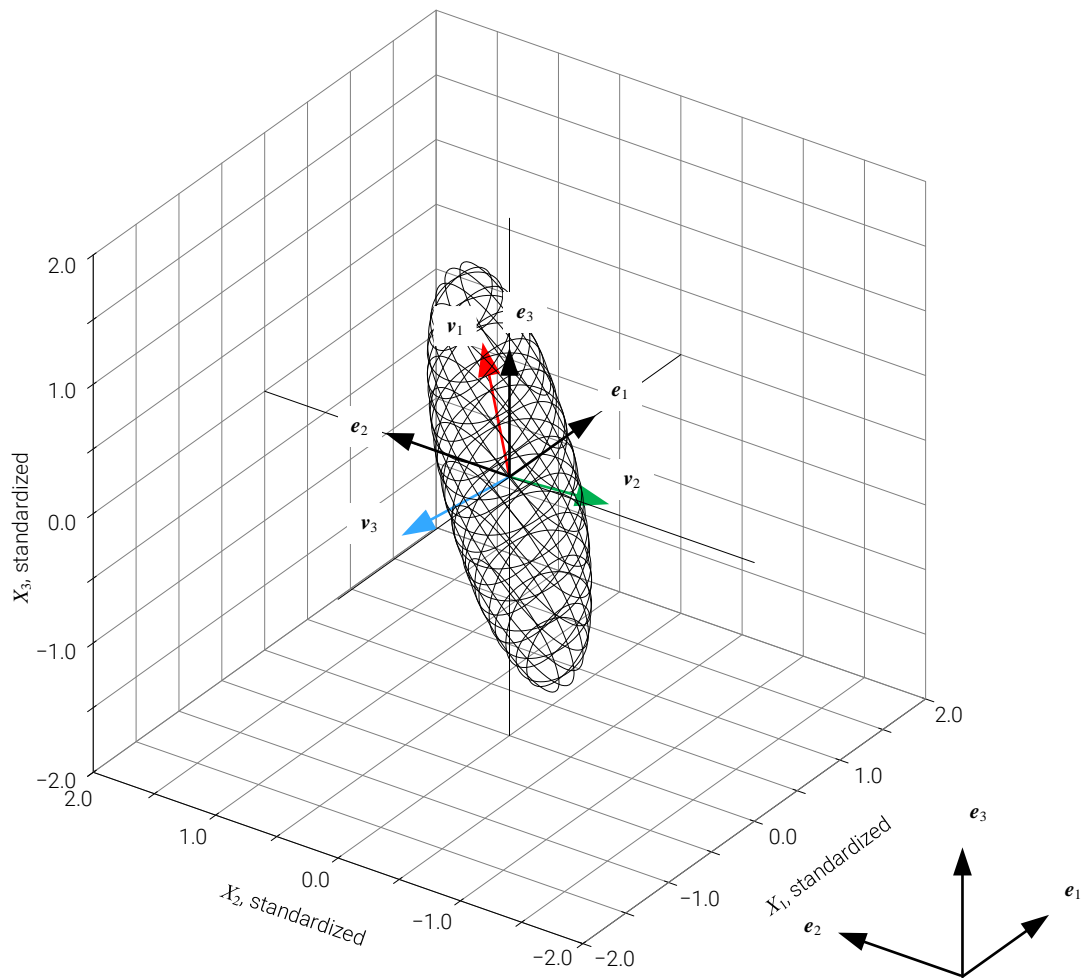
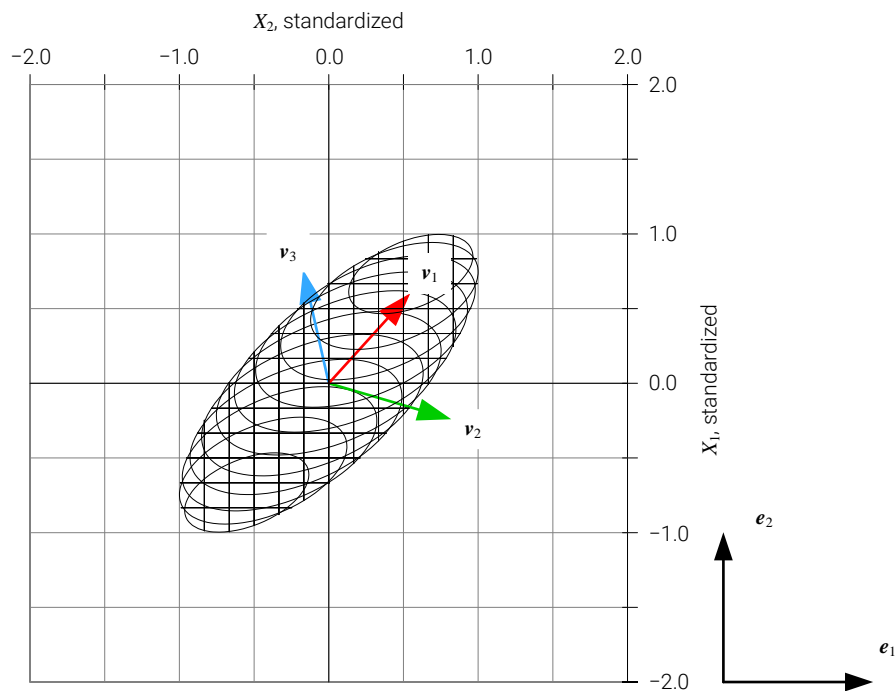
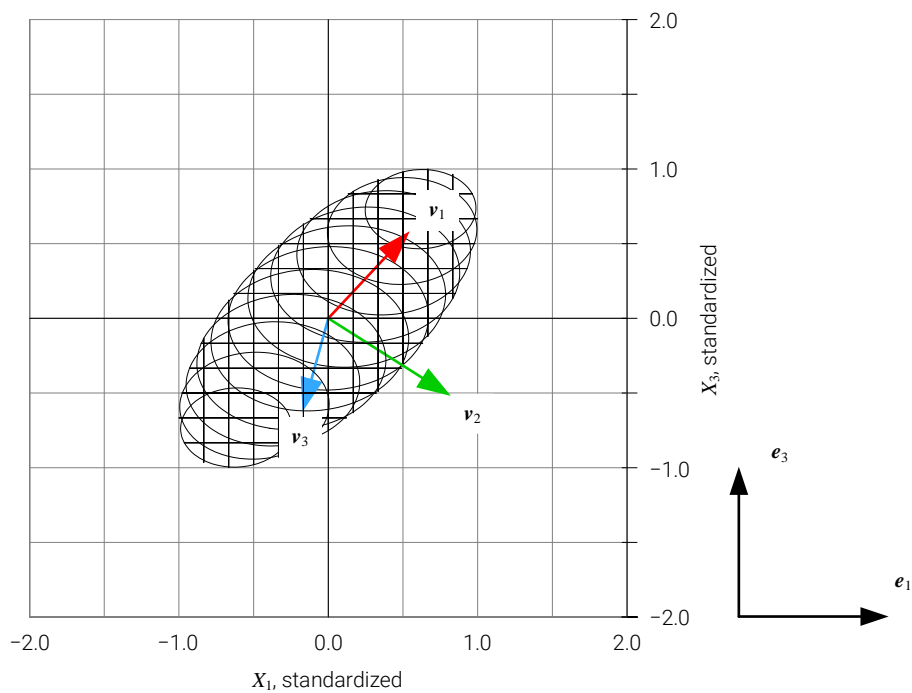
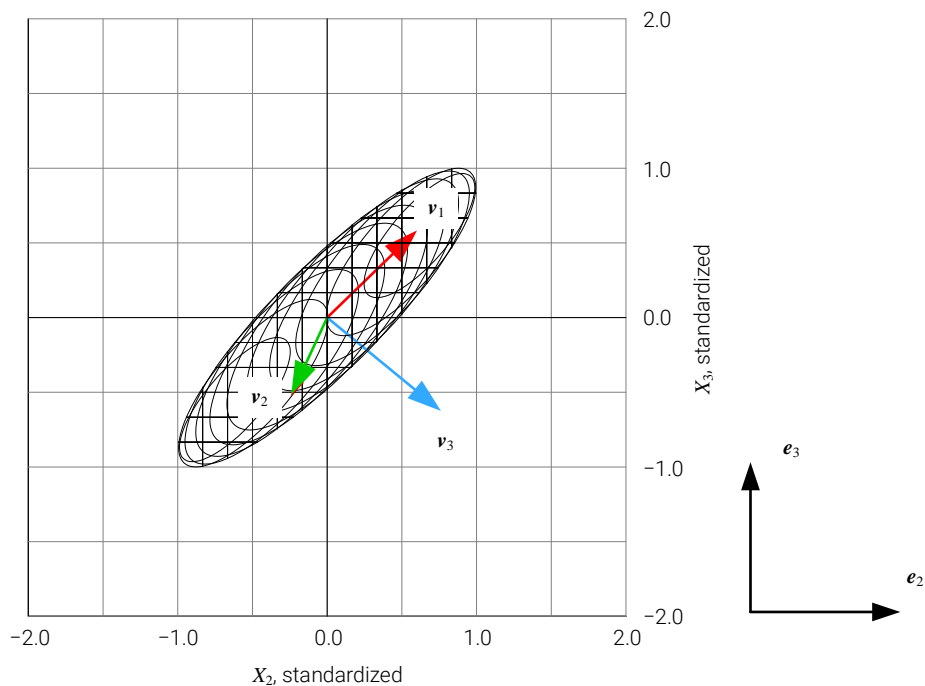


图 13.  $[e_1, e_2, e_3]$  中看椭球  
排版时，请尽量不要缩放此图

图 14.  $[e_1, e_2]$  中看椭圆图 15.  $[e_1, e_3]$  中看椭圆

图 16.  $[e_2, e_3]$  中看椭球

由于  $[v_1, v_2, v_3]$  也是一个三维直角坐标系，我们当然也可以在  $[v_1, v_2, v_3]$  中观察椭球。如图 17 所示，在  $[v_1, v_2, v_3]$  中，我们看的是正椭球。

这幅图中，我们还看到了  $[e_1, e_2, e_3]$ 。图 18 所示为在  $[v_1, v_2]$  中看椭球；而  $e_1, e_2, e_3$  在  $[v_1, v_2]$ ，即第一、第二主成分方向，中的投影也叫**双标图** (biplot)。

双标图可以用于可视化原始多维数据在主成分分析下的投影降维结果。

图 19 所示为在  $[v_1, v_3]$  中看椭球。图 20 所示为在  $[v_2, v_3]$  中看椭球。

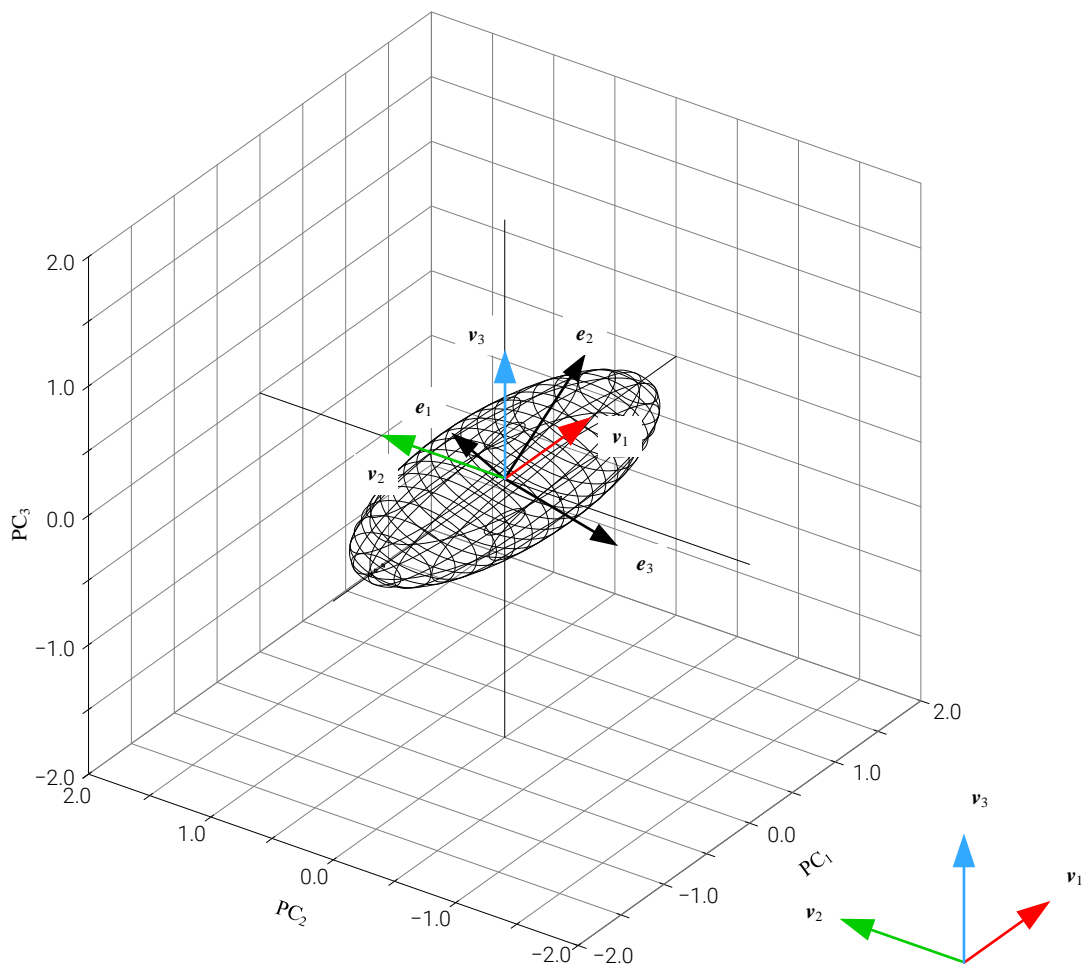
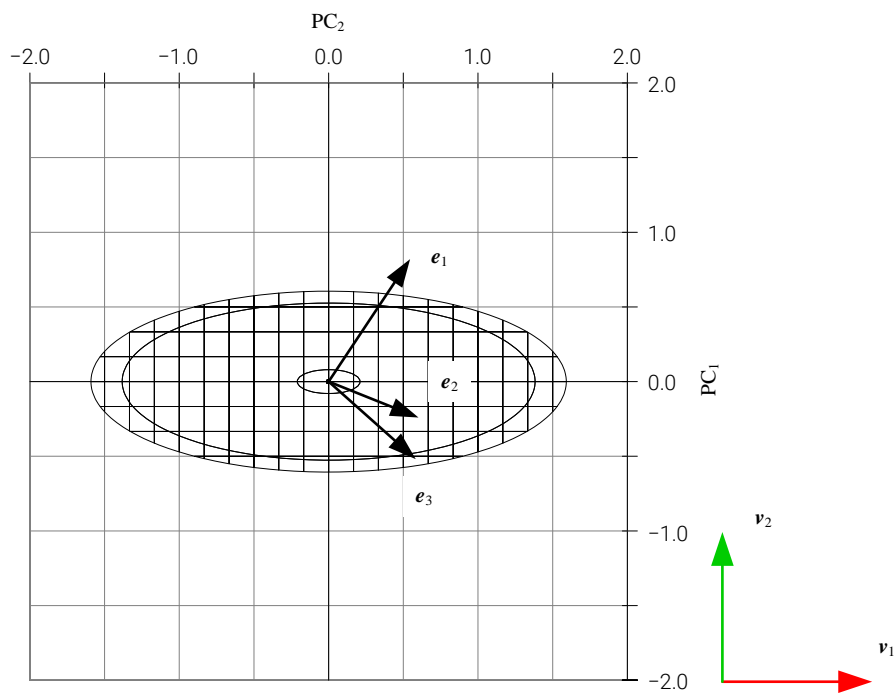


图 17.  $[v_1, v_2, v_3]$  中看椭球  
排版时，请尽量不要缩放此图



本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

图 18.  $[v_1, v_2]$  中看椭球

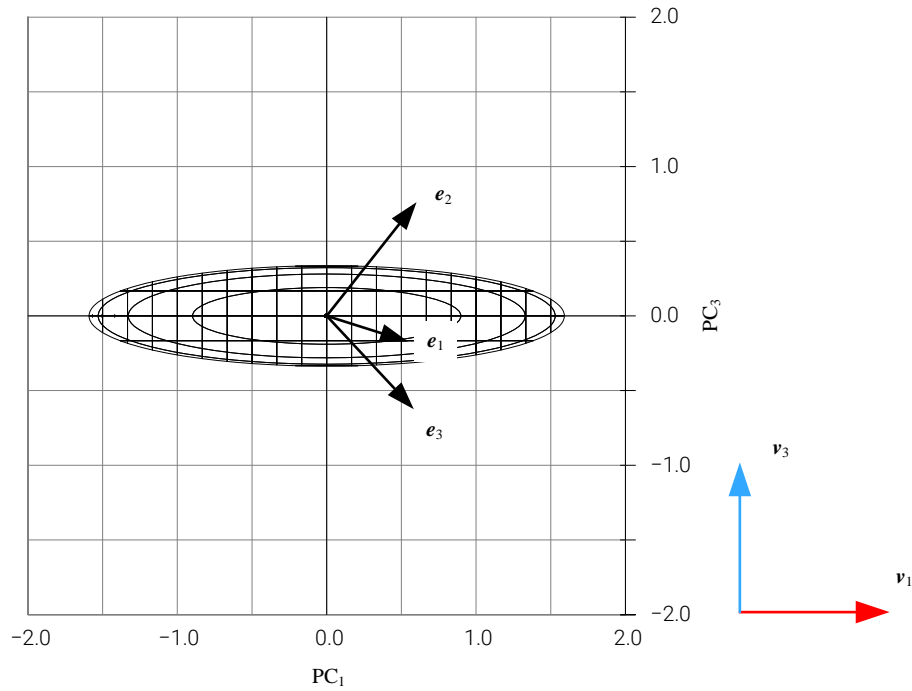


图 19.  $[v_1, v_3]$  中看椭球

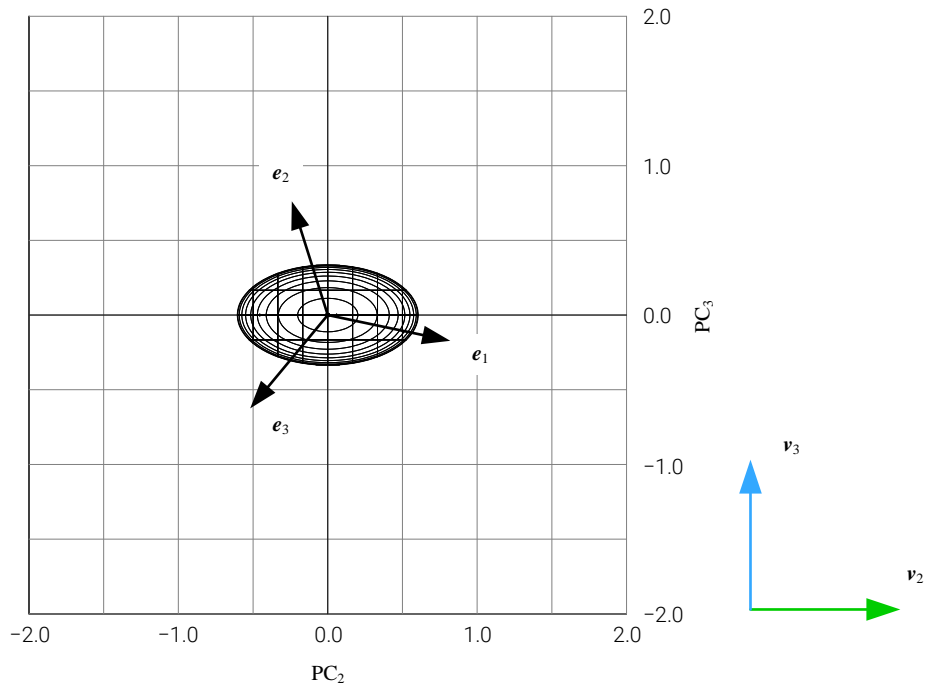


图 20.  $[v_2, v_3]$  中看椭球



请大家完成如下题目。

Q1. 修改 `Bk1_Ch31_01.ipynb`，将样本数据换成鸢尾花特征数据，设定主成分数量为 2，重新完成本章代码中所有分析。

\* 题目很基础，本书不给答案。



表面上，本章介绍了 `Scikit-learn` 中完成 PCA 的工具；但是，更要的是引入了几何视角帮大家更好地理解 PCA 原理。这也是本书反复提到的，“调包”并不是我们的终极目的；搞清楚这些函数背后的数学工具、算法逻辑才是我们想要达成的目标。

当然，本书仅仅要求大家知其然，不要求大家知其所以然；即便如此，在合适的时机，让大家一窥数学之美还是有必要的。