

Python 科学计算基础

第十二章 统计计算

2025 年 9 月 5 日

目录

概率分布

随机数和模拟

假设检验

线性回归

概率分布

`scipy.stats` 提供了统计计算使用的类和函数，包括概率分布和假设检验等。

以下两个表分别列出了描述离散概率分布和连续概率分布的一些常用函数和特征，包括 PMF(离散概率分布的概率质量函数)、PDF(连续概率分布的概率密度函数)、CDF(累积分布函数)、Survival Function(生存函数)、PPF(百分比点函数)和 Moments(矩) 等。连续概率分布的参数 L 和 S 分别表示位置和尺度。对于正态分布，位置 (location) 是期望，尺度参数 (scale) 是标准差。

离散概率分布的函数和特征

函数(特征)名称	定义
PMF	$p(x_k) = P[X = x_k]$
CDF	$F(x) = P[X \leq x] = \sum_{x_k \leq x} p(x_k)$ $F(x_k) - F(x_{k-1}) = p(x_k)$
Survival Function	$S(x) = 1 - F(x) = P[X > x]$
PPF	$G(q) = F^{-1}(q)$
Moments	非中心矩 $\mu'_m = E[X^m] = \sum_k x_k^m p(x_k)$ 中心矩 $\mu_m = E[(X - \mu)^m] = \sum_k (x_k - \mu)^m p(x_k)$ 期望 mean $\mu = \mu'_1 = E[X] = \sum_k x_k p(x_k)$ 方差 variance $\mu_2 = E[(X - \mu)^2]$ $= \sum_{x_k} x_k^2 p(x_k) - \mu^2$ 偏度 skewness $\gamma_1 = \mu_3 / \mu_2^{3/2}$ 峰度 kurtosis $\gamma_2 = \mu_4 / \mu_2^2 - 3$

连续概率分布的函数和特征

函数(特征)名称	定义(标准形式)	定义($X = L + SY$)
CDF	$F(x) = \int_{-\infty}^x f(x)dx$	$F(x; L, S) = F(\frac{x-L}{S})$
PDF	$f(x) = F'(x)$	$f(x; L, S) = \frac{1}{S}f(\frac{x-L}{S})$
Survival Function	$S(x) = 1 - F(x)$	$S(x; L, S) = S(\frac{x-L}{S})$
PPF	$G(q) = F^{-1}(q)$	$G(x; L, S) = L + SG(q)$
Moments	非中心矩 $\mu'_n = E[Y^n]$ 中心矩 $\mu_n = E[(X - \mu)^n]$ 期望 mean μ 方差 variance μ_2 偏度 skewness $\gamma_1 = \mu_3/\mu_2^{3/2}$ 峰度 kurtosis $\gamma_2 = \mu_4/\mu_2^2 - 3$	$E[X^n]$ $E[(X - \mu_X)^n]$ $L + S\mu$ $S^2\mu_2$ γ_1 γ_2

概率分布

程序 12.1 获取 `scipy.stats` 提供的离散概率分布和连续概率分布的类的列表。每个类都有文档说明。

程序 12.2 演示了用指定参数创建一个期望为 3 和标准差为 2 的正态分布并输出一些函数值。`In[6]` 行的 `interval` 函数返回一个区间，该区间内的 PDF 曲线和 x 轴之间围成的区域的面积为指定的实参值 0.95，并且该区域在中位数两侧的子区域的面积相同。`In[7]` 行的 `ppf` 函数对于实参列表中的每一个表示分位数的元素返回对应的 CDF 的自变量值。`In[8]` 行的 `rvs` 函数生成指定数量的服从该分布的随机数。

概率分布

程序 12.3 绘制了四个不同参数的正态分布的 PDF。

程序 12.4 绘制了六个概率分布的 PDF(PMF)、CDF、SF 和 PPF。PDF(PMF) 曲线和 x 轴之间围成的的蓝色区域的面积是 0.95，左右边界的 x 值分别是 CDF 的 0.025 分位数和 0.975 分位数。

程序 12.5 调用 `plot_dist_samples` 函数绘制了三个概率分布的 PDF 以及从中抽样得到的随机数的直方图。

生成随机数

程序 12.6 演示了生成随机数的方法。

random 模块的 uniform(a,b) 函数可生成服从区间 $[a,b)$ 内的均匀分布的随机数，这些随机数构成的序列由随机数种子确定。随机数种子的默认值为当前时间。如果需要程序在每次运行时生成相同序列的随机数，需要设定随机数种子为同一个数值。

生成随机数

numpy 库的 random 模块提供了多个生成随机数的函数。

- ▶ `rand` 函数可生成服从区间 $[0,1)$ 内的均匀分布的随机数：
若不提供参数，生成一个随机数；若提供一个或多个整数作为参数，生成一个由它们指定形状的数组，该数组的每个元素是一个随机数。
- ▶ `randint` 函数生成一个指定形状的数组，由服从某一区间内的均匀分布的随机整数组成，区间的范围 $[low,high)$ 由参数 `low` 和 `high` 指定，数组的形状由参数 `size` 指定。
- ▶ `randn` 函数生成服从标准正态分布的随机数，用法和 `rand` 函数类似。

模拟随机现象

很多现象具有不确定性，例如微观粒子的运动、遗传与变异、股票价格的波动等。生成随机数的一个用途是用计算机程序对随机现象进行模拟。

赌徒破产

赌徒破产 (Gambler's Ruin) 是一个经典的随机过程模型。

这里假定赌徒和资产为无穷大的庄家对赌。赌徒的初始资产为 x 。当赌徒的资产达到目标值 N (N 满足 $0 \leq x \leq N$) 或者变为 0(即破产) 时赌博过程结束。在每一轮赌博中，赌注为 1 并且赌徒获胜的概率为 p 。设 $p_x = P(\text{ruin}|x)$ 表示赌徒以资产 x 开始赌博并最终破产的条件概率。

赌徒破产

考察第一轮赌博的两种结果：获胜则资产变为 $x + 1$ ，失败则资产变为 $x - 1$ 。可得出方程：

$$p_x = P(\text{ruin}|x) = P(\text{ruin}|x+1)p + P(\text{ruin}|x-1)(1-p)。$$

由此可得递推公式： $p_x = p_{x+1}p + p_{x-1}(1-p)$ 。代入边界条件 $p_0 = 1$ 和 $p_N = 0$ 后经演算可得结果：

$$p_x = \begin{cases} 1 - \frac{1 - ((1-p)/p)^x}{1 - ((1-p)/p)^N} & \text{if } p \neq 0.5 \\ 1 - \frac{x}{N} & \text{if } p = 0.5 \end{cases}$$

赌徒破产

程序 12.7 的 `gamble` 函数利用从二项分布生成的随机数模拟了 1000 次赌博过程来估算 $p = 0.49$ 时的破产概率，连续十次的输出结果依次为：0.608, 0.588, 0.602, 0.583, 0.596, 0.594, 0.608, 0.597, 0.595 和 0.605。它们的平均值 0.5976 和理论计算值 0.5987 很接近。

赌徒破产

程序输出的图包含三个子图，这些子图绘制在由大小相同的方块构成的阵列上，每个子图的位置和大小通过其所占据的方块的行号和列号的范围指定。

第一个子图绘制了 p_x 与 p 的关系，可见在 p 接近 0.5 时曲线非常陡峭，即 p 的微小变动导致 p_x 的大幅变动。当 p 从 0.5 逐渐减小时，破产的概率 p_x 急剧增加。

其余两个子图分别绘制了两次赌博过程的资产变化情况，它们分别是资产达到目标值和破产这两种情况的所有赌博过程中选取的轮次最多的例子。

假设检验

假设检验是一种统计推断方法。统计学中的假设是对参数值或总体分布的判断。假设检验根据从总体中抽样得到的样本的计算结果对两个相互对立的假设做出决策。两个假设分别称为原假设 H_0 和备选假设 H_a 。

检验统计量是样本数据的函数，服从特定的概率分布。在检验统计量的取值范围中确定一个非空子集称为拒绝域。如果根据样本数据计算的检验统计量的值落入了拒绝域则拒绝原假设并接受备选假设，否则不能拒绝原假设。不能拒绝原假设并不是肯定原假设成立，只是说明当前样本数据没有提供充分的拒绝原假设的证据。

假设检验

由于抽样的随机性，做出以上决策时难免发生错误。第Ⅰ类错误是在原假设为真的情况下拒绝原假设，第Ⅱ类错误则是在原假设为假的情况下接受原假设。

由于实践中犯第Ⅰ类错误导致的后果更严重，需要确保犯第Ⅰ类错误的概率不超过一个主观选定的值 α ，称为显著性水平 (significance level)。拒绝域的选取方式是使得检验统计量落入其中的概率为 α 。 α 通常取值为 0.05, 0.01 或更小的值。如果检验统计量的值落入了拒绝域，则认为发生了小概率事件，因此有充分理由拒绝原假设。置信区间是拒绝域的补集，即检验统计量落入置信区间的概率为 $1 - \alpha$ 。

假设检验

显著性水平是一个主观选定的值。为了更客观地报告假设检验的结果，可以使用 P 值。P 值表示在假定 H_0 成立的前提下，检验统计量的取值比当前样本计算值更加否定 H_0 的概率。

P 值越小，说明当前样本已经提供了越充分的拒绝 H_0 的证据。得出 P 值以后可以无需再计算拒绝域。如果 P 值小于或等于显著性水平 α 则拒绝 H_0 ，否则不能拒绝 H_0 。

假设检验

这里列举几种常用的假设检验类型和使用方法。

- ▶ z 检验：当总体服从标准差为 σ 的正态分布时， z 检验从总体中抽取一个样本对总体的未知期望进行检验。
- ▶ 单样本 t 检验：当总体服从标准差未知的正态分布时，单样本 t 检验从总体中抽取一个样本对总体的未知期望进行检验，公式中的 s 是样本标准差。
- ▶ 双样本 z 检验：当两个总体服从标准差分别为 σ_1 和 σ_2 的正态分布时，双样本 z 检验从它们中各抽取一个样本，对两个总体的未知期望的差进行检验。两个样本的大小分别为 m 和 n 。

假设检验

- ▶ 双样本 t 检验：当两个总体服从标准差未知但相同的正态分布时，双样本 t 检验从它们中各抽取一个样本，对两个总体的未知期望的差进行检验。两个样本的大小分别为 m 和 n 。公式中的 s_p 满足 $s_p^2 = \frac{m-1}{m+n-2} s_1^2 + \frac{n-1}{m+n-2} s_2^2$ ，其中 s_1 和 s_2 分别是两个样本的样本标准差。
- ▶ 配对 t 检验：配对 t 检验对同一组大小为 n 的个体在不同情况下的观测值所属分布的未知期望的差进行检验。设两组观测值分别为 X_i 和 Y_i ($1 \leq i \leq n$)，假设 $D_i = X_i - Y_i$ 服从正态分布，则可对样本 D_i 进行单样本 t 检验。

scipy.stats 的假设检验的函数

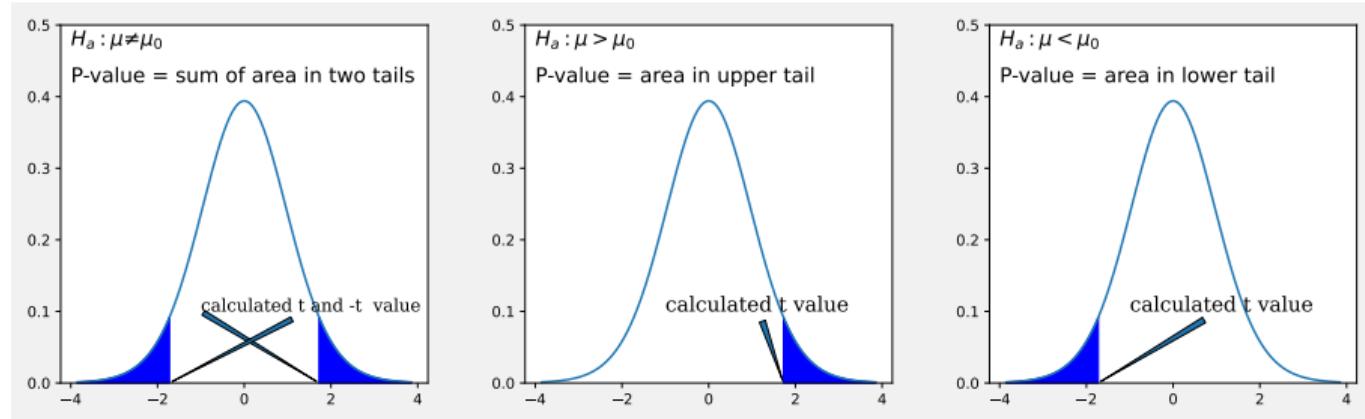
scipy.stats 提供了进行多种假设检验的函数。

`ttest_1samp` 函数进行单样本 t 检验。关键字参数 `alternative` 的默认值为 `two-sided`，即 H_0 为 $\mu = \mu_0$ 且 H_a 为 $\mu \neq \mu_0$ 。如果将其设为 `greater`，则 H_a 为 $\mu > \mu_0$ 。如果将其设为 `less`，则 H_a 为 $\mu < \mu_0$ 。

scipy.stats 的假设检验的函数

下图绘制了以上三种情形下根据从样本计算得到的 t 统计量的值计算 P 值的方法。

- ▶ 第一种情形下，P 值等于两个尾部区域的面积，这两个区域都位于 t 分布的 PDF 和 x 轴之间， $x \in (-\infty, -|t|]$ 和 $x \in [|t|, +\infty)$ 。
- ▶ 第二种情形下，P 值等于 $x \in [t, +\infty)$ 的尾部区域的面积。
- ▶ 第三种情形下，P 值等于 $x \in (-\infty, t]$ 的尾部区域的面积。



scipy.stats 的假设检验的函数

程序 12.8 演示了基于两组大小均为 40 的样本进行的单样本 t 检验。当检验值 4.0 恰好等于正态总体的均值时 P 值较大，而当检验值 1.0 与正态总体的均值相差较大时 P 值很小。

程序 12.9 演示了使用 `ttest_ind` 函数进行双样本 t 检验。关键字参数 `equal_var` 的默认值为 `True`，即假定两个总体的方差相同。如果这一条件不成立，将其设定为 `False`，则执行 Welch 检验。关键字参数 `alternative` 的默认值为 `two-sided`，即 H_0 为 $\mu_1 = \mu_2$ 且 H_a 为 $\mu_1 \neq \mu_2$ 。如果将其设为 `greater`，则 H_a 为 $\mu_1 > \mu_2$ 。如果将其设为 `less`，则 H_a 为 $\mu_1 < \mu_2$ 。

程序 12.10 演示了使用 `ttest_rel` 函数进行配对 t 检验，用来比较同一个班级在两次考试中的平均成绩是否存在显著不同。

线性回归

线性回归是一种统计模型，描述了因变量和一个或多个自变量之间的带有不确定性的线性依赖关系。线性回归的用途是可以根据构建的模型从给定的自变量值预测因变量值。

只有一个自变量的简单线性回归模型定义为

$Y = \beta_0 + \beta_1 x + \epsilon$ 。因变量 Y 是一个随机变量，它的每次观测值是自变量 x 的线性函数和一个随机偏差 ϵ 的叠加，后者服从正态分布 $N(0, \sigma^2)$ ，其中方差 σ^2 和 x 无关。给定观测到的 n 个数据对 $(x_1, y_1), \dots, (x_n, y_n)$ ，最小二乘原理给出的估计为

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}, \quad \hat{\sigma}^2 = \frac{SSE}{n - 2}$$

线性回归

决定系数 (coefficient of determination) 是一个衡量线性回归模型拟合数据的效果的指标，定义为 $r^2 = 1 - SSE/SST$ ，其中 $SST = \sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2/n$ 称为总平方和 (total sum of squares)。

决定系数表示在解释因变量随自变量的变化时，回归模型能够说明的部分所占的比例。决定系数的取值范围是区间 $[0,1]$ ，它的值越高说明模型拟合数据的效果越好。

线性回归

有 k 个自变量的多元回归模型定义为

$Y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + \epsilon$ 。给定观测到的 n 个数据对 $([x_{11}, \dots, x_{1k}], y_1), \dots, ([x_{n1}, \dots, x_{nk}], y_n)$ ，使用矩阵定义的线性回归方程为

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1k} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \dots & x_{nk} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

\mathbf{X} 称为设计矩阵 (design matrix)。最小二乘原理给出的估计为 $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ 。决定系数 $r^2 = 1 - SSE/SST$ ，其中 $SSE = (\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}})$ ， $SST = (\mathbf{y} - \bar{\mathbf{y}})^T (\mathbf{y} - \bar{\mathbf{y}})$ 。

线性回归

程序 12.11 演示了使用 `numpy.linalg` 模块的 `linalg` 函数求解回归模型 $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$ 。

根据观测数据设计一个线性回归模型时，如果模型的决定系数较低就需要修改模型，即重新计算设计矩阵 \mathbf{X} 。使用 Patsy 公式语言定义 `statsmodels` 库的线性回归模型可以自动计算设计矩阵，因此在修改模型时更加便捷。

程序 12.11 的 In[4] 行创建了一个 `pandas` 模块的 `DataFrame` 对象，它存储了所有的输入数据。In[5] 行计算设计矩阵，并指定返回类型为 `pandas` 模块的 `DataFrame`。In[7] 行使用 `OLS` 函数创建了线性回归模型，然后使用 `fit` 函数对其进行拟合。

线性回归

程序 12.14 的第 5 行定义了一个基于自变量 x_1 和 x_2 的二次多项式函数作为真实模型。第 7 行根据随机生成的自变量值计算对应的因变量值，然后在第 9 行叠加服从正态分布的随机偏差以模拟观测值。接着用复杂度从低到高的四个模型依次进行拟合并输出决定系数。前三个模型的决定系数都较低。第四个模型的决定系数接近 1，说明拟合效果较好。

在模型中增加自变量的数量会导致决定系数不断增加，但是过多的变量可能导致过度拟合 (overfitting)，即模型不仅反映了因变量和自变量之间的依赖关系，还反映了数据的随机噪声。过度拟合的模型在预测时的可靠性会降低。

线性回归

程序 12.14 的输出结果包含了模型中需要求解的参数值及其显著性水平为 0.05 的置信区间和多种假设检验的结果。如果拟合的效果较好，则决定系数 (R-squared) 应接近 1 且残差应服从正态分布。

残差服从正态分布的依据有：

- ▶ 偏度 (Skew) 接近 0;
- ▶ 峰度 (kurtosis) 接近 3;
- ▶ Omnibus 检验的 P 值不是太小;
- ▶ Jarque-Bera 检验的 P 值也不是太小。

logistic 回归

在某些问题中因变量的取值不连续，例如因变量 y 取值为 1 和 0 分别表示某一事件发生和不发生。这种情形下线性回归模型不适用。设 x 为唯一自变量，需要预测给定 x 值时事件发生的概率： $p(x) = P(y = 1|x)$ 。由于 $p(x)$ 的取值范围是 $[0,1]$ ，所以使用 logit 函数 $f(t) = \log \frac{t}{1-t}$ 对其变换以后作为线性回归模型的因变量，这样得到的回归模型称为单个自变量的 logistic 回归模型。

$$\log \frac{p(x)}{1 - p(x)} = \beta_0 + \beta_1 x + \epsilon$$

logistic 回归

若问题有 k 个自变量 x_1, \dots, x_k , 则 k 个自变量的 logistic 回归模型为

$$\log \frac{p(x_1, \dots, x_k)}{1 - p(x_1, \dots, x_k)} = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + \epsilon$$

1986 年 1 月 28 日, 美国挑战者号航天飞机在进行第 10 次太空任务时, 因为右侧固态火箭推进器上面的一个 O 形环失效而导致的连锁反应, 在升空后 73 秒时爆炸解体。**程序 12.15** 根据历次发射时的 O 形环失效事件和温度数据构建了一个 logistic 回归模型, 它可以对于给定的温度(自变量)预测 O 形环失效的概率。

logistic 回归

程序 12.15 的输出结果显示：对于温度的系数 β_1 的假设检验 $H_0 : \beta_1 = 0$ 和 $H_a : \beta_1 \neq 0$ 的 P 值为 0.032。在 0.05 显著性水平上可以拒绝该假设，表明温度对于 O 形环失效发挥重要作用。

事故当天的温度为华氏 31 度，代入公式

$P(y = 1|x) \approx \frac{e^{15.043 - 0.232x}}{1 + e^{15.043 - 0.232x}}$ 得到的计算结果为 0.99961，这说明当天 O 形环的失效几乎是必然的。