描述性统计方法

一、scipy.stats模块

> 官方文档：Statistics (scipy.stats)

`scipy.stats`模块包含了多种概率分布的随机变量，随机变量分为**连续型**和**离散型**两种。

连续型随机变量对象都有如下方法：

- rvs：产生随机数，可以通过size参数指定输出的数组的大小。
- pdf：随机变量的概率密度函数。【离散型中的调用分辨率函数的方法为pmf】
- cdf：随机变量的分布函数。
- sf：随机变量的生存函数，它的值是1-cdf。
- ppf：分布函数的反函数。【用于计算分布的α分位数】
- stat：计算随机样本的期望和方差。
- fit：对一组随机样本利用极大似然估计法，估计总体中的未知参数。

常用的连续型随机变量【以调用概率密度函数为例】：

分布名称	关键字	概率密度函数调用方式
均匀分布	`uniform`	`uniform.pdf(x,a,b)`：[a,b]区间上的均匀分布
指数分布	`expon`	`expon.pdf(x,scale=theta)`：期望为theta的指数分布
正态分布	`norm`	`norm.pdf(x,mu,sigma)`：均值为mu、标准差为sigma的正态分布
卡方分布	`chi2`	`chi2.pdf(x,n)`：自由度为n的卡方分布
t分布	`t`	`t.pdf(x,n)`：自由度为n的t分布
F分布	`f`	`f.pdf(x,dfn,dfd)`：自由度为dfn,dfd的F分布
Γ分布	`gamma`	`gamma.pdf(x,a=A,scale=B)`：形状参数为A、尺度参数为B的Γ分布

常用的离散型随机变量【以调用分布律函数为例】：

分布名称	关键字	分布律函数调用方式
二项分布	`binom`	`binom.pmf(x,n,p)`：计算x处的概率
几何分布	`geom`	`geom.pmf(x,p)`：计算第x次首次成功的概率
泊松分布	`poisson`	`poisson.pmf(x,lambda)`：计算x处的概率

二、统计量

1. 描述集中程度和位置的统计量

- 样本均值：反映样本观测值的集中趋势或平均水平

$$\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i$$

- 中位数：描述了数据分布的重心位置，不受极端数据或异常值的影响

$$m_e=\begin{cases}x_{\left(\frac{n+1}{2}\right)}, & \text{当 } n \text{ 为奇数时}\\ \frac{x_{\left(\frac{n}{2}\right)}+x_{\left(\frac{n}{2}+1\right)}}{2}, & \text{当 } n \text{ 为偶数时}\end{cases}$$

- 众数：样本观测值中出现频率或次数最大的数值

2. 描述分散或变异程度的统计量

- 标准差：各个数据与均值偏离程度的度量

$$s=\left[\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2\right]^{\frac{1}{2}}$$

- 方差：标准差的平方
- 极差：描述样本数据分散程度，数据越分散，其极差越大

$$R=\max_{1\le i\le n}x_i-\min_{1\le i\le n}x_i$$

- 变异系数：刻画样本数据相对分散性的一种度量

$$CV=\frac{s}{\bar{x}}\times 100\%$$

3. 表示分布形状的统计量

- 偏度：反映分布的对称性（eg. $\nu_1>0$称为右偏态）

$$\nu_1=\frac{1}{s^3}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^3$$

- 峰度：分布形状的另一种度量（标准正态分布的峰度为3）

$$\nu_2=\frac{1}{s^4}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^4$$

4. 表示变量间关系的统计量

- 协方差

$$\operatorname{cov}(x,y)=\frac{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)\left(y_i-\bar{y}\right)}{n-1}$$

- 相关系数

$$\rho_{xy}=\frac{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)\left(y_i-\bar{y}\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2}\sqrt{\sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2}}=\frac{\operatorname{cov}(x,y)}{s_x s_y}$$

5. 原点矩和中心距

- k阶原点矩

$$a_k = \frac{1}{n}\sum_{i=1}^{n} x_i^k, \quad k = 1, 2, \cdots$$

- k阶中心距

$$b_k = \frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^k, \quad k = 2, 3, \cdots$$

Python实现

> ndarray格式的数据习惯以`行为同一个变量、列为一个样本`的形式储存、处理；
> DataFrame格式的数据习惯以`行为一个样本、列为同一个变量`的形式储存、处理

>> NumPy库

函数	np.mean(x)	np.median(x)	np.ptp(x)	np.var(x)	np.std(x)	np.cov(x)	np.corrcoef(x)
计算功能	均值	中位数	极差	方差	标准差	协方差矩阵	相关系数矩阵

```
import numpy as np

x = np.array([[1, 2, 3], [1, 4, 3]])  # 一列表示一个样本
mean = np.mean(x)
print(mean)  # 输出：2.3333

mean_0 = np.mean(x, axis=0)
print(mean_0)  # 输出：[1. 3. 3.]

mean_1 = np.mean(x, axis=1)  # ndarray格式的数据通常用`axis=1`
print(mean_1)  # 输出：[2.         2.66666667]

cov = np.cov(x)
print(cov)  # 输出：[[1.         1.        ]
            #       [1.         2.33333333]]

cov_0 = np.cov(x, ddof=0)
print(cov_0)  # 输出：[[0.66666667 0.66666667]
              #       [0.66666667 1.55555556]]

cov_1 = np.cov(x, ddof=1)
print(cov_1)  # 输出：[[1.         1.        ]
              #       [1.         2.33333333]]
```

>> Pandas库（Series和DataFrame）

方法	mean()	median()	ptp()	var()	std()	cov()	corrcoef()	skew()	kurt()
计算功能	均值	中位数	极差	方差	标准差	协方差矩阵	相关系数矩阵	偏度	峰度

```
import pandas as pd

df = pd.DataFrame([[1, 1], [2, 4], [3, 3]])  # 一行表示一个样本
mean = df.mean()
print(mean.values)  # 输出：[2.         2.66666667]

mean_0 = df.mean(axis=0)  # DataFrame格式的数据通常用`axis=0`
print(mean_0.values)  # 输出：[2.         2.66666667]

mean_1 = df.mean(axis=1)
print(mean_1.values)  # 输出：[1. 3. 3.]

cov = df.cov()
print(cov.values)  # 输出：[[1.         1.        ]
                   #       [1.         2.33333333]]

cov_0 = df.cov(ddof=0)
print(cov_0.values)  # 输出：[[0.66666667 0.66666667]
                     #       [0.66666667 1.55555556]]

cov_1 = df.cov(ddof=1)
print(cov_1.values)  # 输出：[[1.         1.        ]
                     #       [1.         2.33333333]]
```

三、统计图

1. 频数表和直方图

>> Python实现

```
import pandas as pd
from matplotlib import pyplot as plt

df = pd.DataFrame([[60, 79, 48, 76, 67, 58, 65, 78, 64, 75, 76, 78, 84, 48, 25,
                    90, 98, 70, 77, 78, 68, 74, 95, 80, 90, 78, 73, 98, 85, 56],
                   [91, 74, 62, 72, 90, 94, 76, 83, 92, 85, 94, 83, 77, 82, 84,
                    60, 80, 78, 88, 90, 65, 77, 89, 86, 56, 87, 66, 56, 83,
67]]).T

plt.subplot(121)
h1 = plt.hist(df[0], 5, rwidth=0.75)  # h1为频数表
plt.xticks(h1[1])
plt.yticks(h1[0])
plt.subplot(122)
plt.hist(df[1], 5, rwidth=0.75)
```

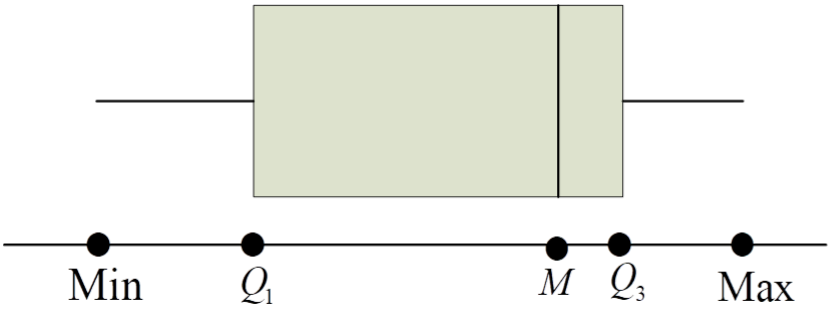
```
df.hist(grid=False, bins=5, rwidth=0.75)  # 另一种方法画直方图
plt.show()
```

2. 箱线图

>> 表示方式

数据集的箱线图是由箱子和直线组成的图形，它是基于以下5个数的图形概括：

- 最小值Min
- 第一四分位数Q_1
- 中位数M
- 第三四分位数Q_3
- 最大值Max



>> 异常值

对于第一四分位数Q_1与第三四分位数Q_3之间的距离$IQR = Q_3 - Q_1$，称为四分位数间距。

数据小于$Q_1 - 0.5IQR$或大于$Q_3 + 1.5IQR$，就认为它是疑似异常值。

>> Python实现

```
import pandas as pd
from matplotlib import pyplot as plt

df = pd.DataFrame([[60, 79, 48, 76, 67, 58, 65, 78, 64, 75, 76, 78, 84, 48, 25,
                    90, 98, 70, 77, 78, 68, 74, 95, 80, 90, 78, 73, 98, 85, 56],
                   [91, 74, 62, 72, 90, 94, 76, 83, 92, 85, 94, 83, 77, 82, 84,
                    60, 80, 78, 88, 90, 65, 77, 89, 86, 56, 87, 66, 56, 83,
67]]).T
df.boxplot(sym="rx")  # `sym`参数设置异常点的样式(包括颜色、形状等)
plt.show()
```

3. 经验分布函数

>> 定义

设$X_1, X_2, \cdots, X_n$是总体F的一个样本，用$S(n)$表示$X_1, X_2, \cdots, X_n$中不大于x的随机变量的个数。定义经验分布函数$F_n(x)$为

$$F_n(x) = \frac{1}{n}S(x), \quad -\infty < x < \infty$$

>> 观测值

设$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$，则经验分布函数$F_n(x)$的观测值为

$$F_n(x)=\begin{cases}0, & \text{若 } x < x_{(1)} \\ \frac{k}{n}, & \text{若 } x_{(k)} \leq x < x_{(k+1)}, \quad k=1,2,\cdots,n-1 \\ 1, & \text{若 } x \geq x_{(n)}\end{cases}$$

可以证明，当$n \to \infty$时，$F_n(x)$以概率1一致收敛于分布函数$F(x)$。

当n充分大时，观测值$F_n(x)$可以当作总体分布函数$F(n)$使用

>> Python实现

```
import numpy as np
from matplotlib import pyplot as plt

d = np.array([60, 79, 48, 76, 67, 58, 65, 78, 64, 75, 76, 78, 84, 48, 25,
              90, 98, 70, 77, 78, 68, 74, 95, 80, 90, 78, 73, 98, 85, 56])
h1 = plt.hist(d, cumulative=True, density=True, histtype="step")
plt.xticks(h1[1])
plt.yticks(h1[0])
plt.show()
```

4. Q-Q图

>> 原理

Q-Q图将经验分布函数的分位数点和分布模型的理论分位数点作为一对数组画在直角坐标图上，n个观测数据对应n个点。如果这n个点看起来像一条直线，说明观测数据与分布模型的拟合效果很好。

>> Python实现

```
import numpy as np
from matplotlib import pyplot as plt
from scipy.stats import norm

plt.rc('font', size=16)
plt.rc('font', family='SimHei')

d = np.array([60, 79, 48, 76, 67, 58, 65, 78, 64, 75, 76, 78, 84, 48, 25,
              90, 98, 70, 77, 78, 68, 74, 95, 80, 90, 78, 73, 98, 85, 56])
sd = sorted(d)
x = (np.arange(len(d))+1/2)/len(d)
yi = norm.ppf(x, d.mean(), d.std())  # 获取正态分布的分位点

plt.scatter(yi, sd, label='Q-Q图')
plt.plot(sd, sd, label='参照直线', color="red")
plt.legend()
plt.show()
```

四、参数估计和假设检验

> 注意区分观测值和实际值，如总体均值为μ、其观测值为$\bar{X}$，总体标准差为σ、其观测值为s

1. 单个正态总体参数的置信区间

总体$N(\mu,\sigma^2)$的参数区间估计	分布	区间估计
σ^2已知，求μ	$\frac{\bar{X}-\mu}{\sigma/\sqrt{n}}\sim N(0,1)$	$\left(\bar{X}-u_{\alpha/2}\frac{\sigma}{\sqrt{n}},\bar{X}+u_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right)$
σ^2未知，求μ	$\frac{\bar{X}-\mu}{s/\sqrt{n}}\sim t(n-1)$	$\left(\bar{X}-t_{\alpha/2}(n-1)\frac{S}{\sqrt{n}},\bar{X}+t_{\alpha/2}(n-1)\frac{S}{\sqrt{n}}\right)$
μ已知，求σ^2	$\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{\sigma^2}\sim\chi^2(n)$	$\left(\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{\chi^2_{\alpha/2}(n)},\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{\chi^2_{1-\alpha/2}(n)}\right)$
μ未知，求σ^2	$\frac{(n-1)S^2}{\sigma^2}\sim\chi^2(n-1)$	$\left(\frac{(n-1)S^2}{\chi^2_{\alpha/2}(n-1)},\frac{(n-1)S^2}{\chi^2_{1-\alpha/2}(n-1)}\right)$

2. 参数假设检验

3. 非参数假设检验

>> 分布拟合检验

>> 柯尔莫哥洛夫检验（Kolmogorov-Smirnov检验，KS检验）