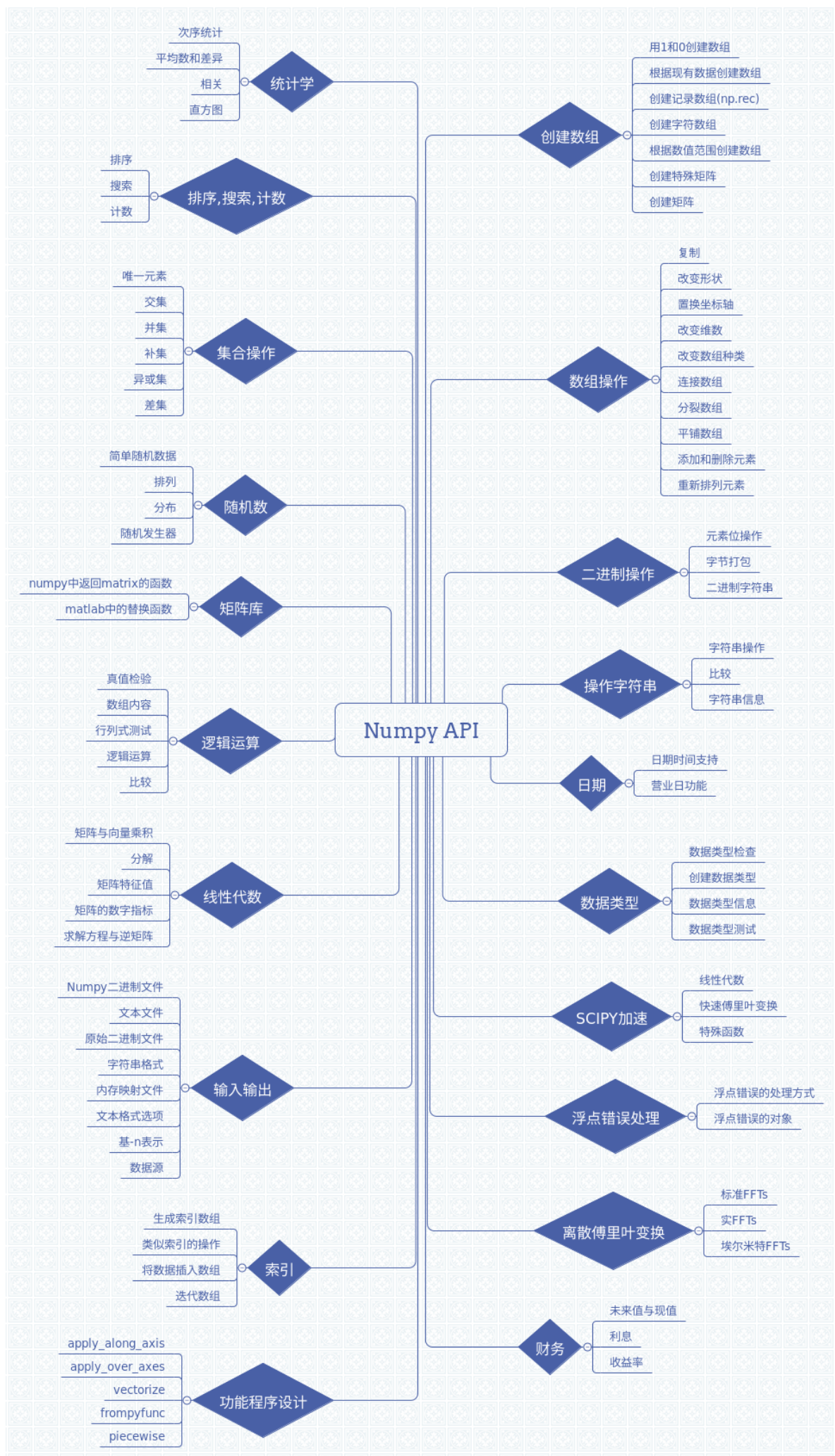




\$1. 创建数组

\$1.1 用1和0创建数组

函数	说明
<code>empty(shape[, dtype, order])</code>	返回给定形状和类型的新数组，而不初始化条目。
<code>empty_like(a[, dtype, order, subok])</code>	返回与给定数组形状和类型相同的新数组。
<code>eye(N[, M, k, dtype, order])</code>	返回对角线上为1、别处为0的二维数组。
<code>identity(n[, dtype])</code>	返回标识数组。
<code>ones(shape[, dtype, order])</code>	返回给定形状和类型的新数组，其中填充了一个。
<code>ones_like(a[, dtype, order, subok])</code>	返回与给定数组形状和类型相同的数组。
<code>zeros(shape[, dtype, order])</code>	返回给定形状和类型的新数组，用零填充。
<code>zeros_like(a[, dtype, order, subok])</code>	返回与给定数组形状和类型相同的零数组。
<code>full(shape, fill_value[, dtype, order])</code>	返回给定形状和类型的新数组，用 <code>fill_value</code> 填充。
<code>full_like(a, fill_value[, dtype, order, subok])</code>	返回与给定数组形状和类型相同的完整数组。

\$1.2 根据现有数据创建数组

函数	说明
<code>array(object[, dtype, copy, order, subok, ndmin])</code>	创建一个数组。
<code>asarray(a[, dtype, order])</code>	将输入转换为数组。
<code>asanyarray(a[, dtype, order])</code>	将输入转换为ndarray，但将ndarray子类传递到。
<code>ascontiguousarray(a[, dtype])</code>	返回内存中的连续数组(C顺序)。
<code>asmatrix(data[, dtype])</code>	将输入解释为矩阵。
<code>copy(a[, order])</code>	返回给定对象的数组副本。
<code>frombuffer(buffer[, dtype, count, offset])</code>	将缓冲区解释为一维数组。
<code>fromfile(file[, dtype, count, sep])</code>	从文本或二进制文件中的数据构造数组。
<code>fromfunction(function, shape, **kwargs)</code>	通过在每个坐标上执行函数来构造数组。
<code>fromiter(iterable, dtype[, count])</code>	从可迭代对象创建新的一维数组。
<code>fromstring(string[, dtype, count, sep])</code>	从字符串中的文本数据初始化的新一维数组。
<code>loadtxt(fname[, dtype, comments, delimiter, ...])</code>	从文本文件加载数据。

\$1.3 创建记录数组(np.rec)

这里 `numpy.rec` 是 `numpy.core.records` 的首选别名。

函数	说明
<code>core.records.array(obj[, dtype, shape, ...])</code>	从各种各样的对象构造记录数组。
<code>core.records.fromarrays(arrayList[, dtype, ...])</code>	从(平面)数组列表中创建记录数组
<code>core.records.fromrecords(recList[, dtype, ...])</code>	从文本形式的记录列表中创建重新排列
<code>core.records.fromstring(datastring[, dtype, ...])</code>	从字符串中包含的二进制数据创建(只读)记录数组
<code>core.records.fromfile(fd[, dtype, shape, ...])</code>	从二进制文件数据创建数组

\$1.4 创建字符数组(np.char)

这里 `numpy.char` 是 `numpy.core.defchararray` 的首选别名。

函数	说明
<code>core.defchararray.array(obj[, itemsize, ...])</code>	创建一个字符数组。
<code>core.defchararray.asarray(obj[, itemsize, ...])</code>	将输入转换为字符数组，仅在必要时复制数据。

\$1.5 根据数值范围创建数组

函数	说明
<code>arange([start,] stop[, step,][, dtype])</code>	在给定间隔内返回均匀间隔的值。
<code>linspace(start, stop[, num, endpoint, ...])</code>	在指定的时间间隔内返回均匀间隔的数字。
<code>logspace(start, stop[, num, endpoint, base, ...])</code>	返回对数刻度上均匀间隔的数字。
<code>geomspace(start, stop[, num, endpoint, dtype])</code>	返回对数刻度上均匀间隔的数字(几何级数)。
<code>meshgrid(*xi, **kwargs)</code>	从坐标向量返回坐标矩阵。
<code>mgrid</code>	<code>nd_grid</code> 实例，返回密集的多维“网格”。
<code>ogrid</code>	返回开放多维“网格”的 <code>nd_grid</code> 实例。

\$1.6 创建特殊矩阵

函数	说明
<code>diag(v[, k])</code>	提取对角线或构造对角线数组。
<code>diagflat(v[, k])</code>	以展平的输入为对角线创建二维数组。
<code>tri(N[, M, k, dtype])</code>	一个数组，在给定对角线上和下方有一个，在其他地方有零。
<code>tril(m[, k])</code>	数组的下三角形。
<code>triu(m[, k])</code>	数组的上三角形。
<code>vander(x[, N, increasing])</code>	生成范德蒙矩阵。

\$1.7 创建矩阵

函数	说明
<code>mat(data[, dtype])</code>	将输入解释为矩阵。
<code>bmat(obj[, ldict, gdict])</code>	从字符串、嵌套序列或数组构建矩阵对象。

\$2 数组操作

\$2.1 复制

函数	说明
<code>copyto(dst, src[, casting, where])</code>	将值从一个阵列复制到另一个阵列，根据需要进行广播。

\$2.2 改变形状

函数	说明
<code>reshape(a, newshape[, order])</code>	在不更改数组数据的情况下为数组赋予新形状。
<code>ravel(a[, order])</code>	返回连续的扁平数组。
<code>ndarray.flat</code>	数组上的一维迭代器。
<code>ndarray.flatten([order])</code>	返回折叠到一维中的数组副本。

\$2.3 置换坐标轴

函数	说明
<code>moveaxis(a, source, destination)</code>	将数组的轴移动到新位置。
<code>rollaxis(a, axis[, start])</code>	向后滚动指定的轴，直到它位于给定位置。
<code>swapaxes(a, axis1, axis2)</code>	交换数组的两个轴。
<code>ndarray.T</code>	与 <code>self.indim < 2</code> 时返回 <code>self()</code> 相同。
<code>transpose(a[, axes])</code>	置换数组的维度。

\$2.4 改变维数

函数	说明
<code>atleast_1d(*arys)</code>	将输入转换为至少具有一维的数组。
<code>atleast_2d(*arys)</code>	将输入视为至少具有两个维度的数组。
<code>atleast_3d(*arys)</code>	将输入视为至少具有三维的数组。
<code>broadcast</code>	制作模拟广播的对象。
<code>broadcast_to(array, shape[, subok])</code>	将数组广播到新形状。
<code>broadcast_arrays(*args, **kwargs)</code>	相互广播任意数量的阵列。
<code>expand_dims(a, axis)</code>	展开数组的形状。
<code>squeeze(a[, axis])</code>	从数组形状中删除一维条目。

\$2.5 改变数组种类

函数	说明
asarray(a[, dtype, order])	将输入转换为数组。
asanyarray(a[, dtype, order])	将输入转换为ndarray，但将ndarray子类传递到。
asmatrix(data[, dtype])	将输入解释为矩阵。
asfarray(a[, dtype])	返回转换为浮点型的数组。
asfortranarray(a[, dtype])	在内存中返回按Fortran顺序排列的数组。
ascontiguousarray(a[, dtype])	返回内存中的连续数组(C顺序)。
asarray_chkfinite(a[, dtype, order])	将输入转换为数组，检查NaNs或Infs。
asscalar(a)	将大小为1的数组转换为其标量等效数组。
require(a[, dtype, requirements])	返回满足要求的所提供类型的ndarray。

\$2.6 连接数组

函数	说明
concatenate((a1, a2, ...[, axis, out])	沿现有轴连接阵列序列。
stack(arrays[, axis, out])	沿着新轴连接一系列数组。
column_stack(tup)	将一维数组作为列堆叠成二维数组。
dstack(tup)	按顺序深度(沿第三轴)堆叠阵列。
hstack(tup)	水平(按列)按顺序堆叠数组。
vstack(tup)	垂直(按行)顺序堆叠阵列。
block(arrays)	从嵌套的块列表中组装nd数组。

\$2.7 分裂数组

函数	说明
split(ary, indices_or_sections[, axis])	将阵列分割成多个子阵列。
array_split(ary, indices_or_sections[, axis])	将阵列分割成多个子阵列。
dsplit(ary, indices_or_sections)	沿第三轴(深度)将阵列分割成多个子阵列。
hsplit(ary, indices_or_sections)	水平(按列)将阵列拆分为多个子阵列。
vsplit(ary, indices_or_sections)	将阵列垂直(按行)拆分为多个子阵列。

\$2.8 平铺数组

函数	说明
<code>tile(A, reps)</code>	通过重复 代表给定的次数来构造数组
<code>repeat(a, repeats[, axis])</code>	重复数组的元素。

\$2.9 添加和删除元素

函数	说明
<code>delete(arr, obj[, axis])</code>	返回删除了沿轴的子阵列的新阵列。
<code>insert(arr, obj, values[, axis])</code>	在给定索引之前沿给定轴插入值。
<code>append(arr, values[, axis])</code>	将值追加到数组的末尾。
<code>resize(a, new_shape)</code>	返回具有指定形状的新数组。
<code>trim_zeros(filt[, trim])</code>	从一维数组或序列中修剪前导零和/或尾随零。
<code>unique(ar[, return_index, return_inverse, ...])</code>	查找数组的唯一元素。

2.10 重新排列元素

函数	说明
<code>flip(m, axis)</code>	沿给定轴反转数组中元素的顺序。
<code>fliplr(m)</code>	向左/向右翻转阵列。
<code>flipud(m)</code>	沿上下方向翻转阵列。
<code>reshape(a, newshape[, order])</code>	在不更改数组数据的情况下为数组赋予新形状。
<code>roll(a, shift[, axis])</code>	沿给定轴滚动阵列元素。
<code>rot90(m[, k, axes])</code>	在轴指定的平面中将阵列旋转90度。

\$ 3. 二进制操作

\$3.1 元素位操作

函数	说明
<code>bitwise_and(x1, x2, /[, out, where, ...])</code>	逐位计算两个数组的“与”。
<code>bitwise_or(x1, x2, /[, out, where, casting, ...])</code>	逐位计算两个数组的OR。
<code>bitwise_xor(x1, x2, /[, out, where, ...])</code>	逐元素计算两个数组的位异或。
<code>invert(x, /[, out, where, casting, order, ...])</code>	逐位或逐位不逐元素计算反演。
<code>left_shift(x1, x2, /[, out, where, casting, ...])</code>	将整数的位向左移动。
<code>right_shift(x1, x2, /[, out, where, ...])</code>	将整数的位向右移动。

\$3.2 字节打包

函数	说明
<code>packbits(myarray[, axis])</code>	将二进制值数组的元素打包成uint 8数组中的位。
<code>unpackbits(myarray[, axis])</code>	将uint 8数组的元素解包为二进制值输出数组。

\$3.3 输出格式

函数	说明
<code>binary_repr(num[, width])</code>	将输入数字的二进制表示形式返回为字符串。

\$4. 操作字符串

\$4.1 字符串操作

函数	说明
<code>add(x1, x2)</code>	返回字符串或unicode两个数组的逐元素字符串连接。
<code>multiply(a, i)</code>	返回($a * i$), 即字符串多重连接, 按元素排序。
<code>mod(a, values)</code>	返回($a \% i$), 即Python 2.6之前的字符串格式(interpolation), 对于字符串或unicode这样的一对array_likes, 逐元素返回。
<code>capitalize(a)</code>	返回一个副本, 其中每个元素的第一个字符都是大写的。
<code>center(a, width[, fillchar])</code>	传回的复本, 其元素以长度宽度字符串为中心。
<code>decode(a[, encoding, errors])</code>	逐元素调用字符串解码。
<code>encode(a[, encoding, errors])</code>	调用字符串按元素编码。
<code>join(sep, seq)</code>	返回一个字符串, 它是序列seq中字符串的串联。
<code>ljust(a, width[, fillchar])</code>	返回一个数组, 数组中的元素在长度为宽度的字符串中左对齐。
<code>lower(a)</code>	返回元素转换为小写的数组。
<code>lstrip(a[, chars])</code>	对于中的每个元素, 返回删除前导字符的副本。
<code>partition(a, sep)</code>	9月份左右, 将每个元素分区
<code>replace(a, old, new[, count])</code>	对于中的每个元素, 返回字符串的副本, 所有出现的子字符串旧的都替换为新的。
<code>rjust(a, width[, fillchar])</code>	返回一个数组, 数组中的元素在长度为宽度的字符串中右对齐。
<code>rpartition(a, sep)</code>	分隔(拆分)最右边分隔符周围的每个元素。
<code>rsplit(a[, sep, maxsplit])</code>	对于中的每个元素, 使用sep作为分隔符字符串返回字符串中的单词列表。
<code>rstrip(a[, chars])</code>	对于中的每个元素, 返回一个删除了尾随字符的副本。
<code>split(a[, sep, maxsplit])</code>	对于中的每个元素, 使用sep作为分隔符字符串返回字符串中的单词列表。
<code>splitlines(a[, keepends])</code>	对于中的每个元素, 返回元素中的线条列表, 在线条边界处断开。
<code>strip(a[, chars])</code>	对于a中的每个元素, 返回一个删除了前导和尾随字符的副本。
<code>swapcase(a)</code>	以元素方式返回字符串副本, 大写字符转换为小写字符, 反之亦然。

函数	说明
title(a)	返回字符串或unicode的元素标题大小写版本。
translate(a, table[, deletechars])	对于中的每个元素，返回字符串的副本，在该副本中，可选参数deletechars中出现的所有字符都被删除，其余字符已通过给定的转换表映射。
upper(a)	返回元素转换为大写的数组。
zfill(a, width)	返回左填充零的数字字符串

\$4.2 比较

与标准numpy比较运算符不同的是，char模块中的运算符在执行比较之前会剥离尾随空白字符。

函数	说明
equal(x1, x2)	按元素返回($x1 == x2$)。
not_equal(x1, x2)	返回($x1 != x2$)元素方式。
greater_equal(x1, x2)	逐元素返回($x1 \geq x2$)。
less_equal(x1, x2)	逐元素返回($x1 \leq x2$)。
greater(x1, x2)	逐元素返回($x1 > x2$)。
less(x1, x2)	逐元素返回($x1 < x2$)。

\$4.3 字符串信息

函数	说明
count(a, sub[, start, end])	返回一个数组，其中子字符串sub在“[开始，结束]”范围内不重叠的出现次数。
find(a, sub[, start, end])	对于每个元素，返回字符串中找到子字符串sub的最低索引。
index(a, sub[, start, end])	如查找，但在未找到子字符串时引发值错误。
isalpha(a)	如果字符串中的所有字符都是字母，并且至少有一个字符，则返回true，否则返回false。
isdecimal(a)	对于每个元素，如果元素中只有十进制字符，则返回True。
isdigit(a)	如果字符串中的所有字符都是数字，并且至少有一个字符，则返回true，否则返回false。
islower(a)	如果字符串中的所有大小写字符都是小写的，并且至少有一个大小写字符，则返回true，否则返回false。
isnumeric(a)	对于每个元素，如果元素中只有数字字符，则返回True。
isspace(a)	如果字符串中只有空白字符，并且至少有一个字符，则返回true，否则返回false。
istitle(a)	如果元素是标题大小写字符串，并且至少有一个字符，则返回true，否则返回false。
isupper(a)	如果字符串中的所有大小写字符都是大写的，并且至少有一个字符，则返回true，否则返回false。
rfind(a, sub[, start, end])	对于a中的每个元素，返回字符串中找到substring sub的最高索引，以便sub包含在[start, end]中。
rindex(a, sub[, start, end])	类似rfind，但在未找到子字符串子字符串时引发值错误。
startswith(a, prefix[, start, end])	返回一个布尔数组，该数组为True，其中a中的字符串元素以前缀开头，否则为False。

\$4.4 便捷视图

函数	说明
chararray(shape[, itemsize, unicode, ...])	提供字符串和unicode值数组的便捷视图。

\$5 日期支持

\$5.1 日期时间支持函数

函数	说明
<code>datetime_as_string(arr[, unit, timezone, ...])</code>	将日期时间数组转换为字符串数组。
<code>datetime_data(dtype, /)</code>	获取有关日期或时间类型步长的信息。

\$5.2 营业日功能

函数	说明
<code>busdaycalendar([weekmask, holidays])</code>	一个工作日日历对象，有效存储定义工作日系列功能的有效天数的信息。
<code>is_busday(dates[, weekmask, holidays, ...])</code>	计算给定日期中哪些是有效天数，哪些不是。
<code>busday_offset(dates, offsets[, roll, ...])</code>	首先根据滚动规则将日期调整为有效日期，然后将偏移应用于以有效日期计算的给定日期。
<code>busday_count(begindates, enddates[, ...])</code>	计算开始日期和结束日期之间的有效天数，不包括结束日期。

\$6 数据类型

\$6.1 数据类型检查

函数	说明
<code>can_cast(from_, to[, casting])</code>	如果可以根据转换规则在数据类型之间进行转换，则返回 True。
<code>promote_types(type1, type2)</code>	返回具有最小大小和最小标量类型的数据类型，类型1和类型2都可以安全地转换到该数据类型。
<code>min_scalar_type(a)</code>	对于标量a，返回大小最小且标量种类最小的数据类型，该数据类型可以保存其值。
<code>result_type(*arrays_and_dtypes)</code>	传回将NumPy型别升级规则套用至引数所产生的型别。
<code>common_type(*arrays)</code>	返回输入数组共有的标量类型。
<code>obj2sctype(rep[, default])</code>	返回对象Python类型的标量dtype或NumPy等效项。

\$6.2 创建数据类型

函数	说明
<code>dtype(obj[, align, copy])</code>	创建数据类型对象。
<code>format_parser(formats, names, titles[, ...])</code>	类将格式、名称、标题描述转换为dtype。

\$6.3 数据类型信息

函数	说明
<code>finfo(dtype)</code>	浮点类型的机器限制。
<code>iinfo(type)</code>	整数类型的机器限制。
<code>MachAr([float_conv, int_conv, ...])</code>	诊断机器参数。

6.4 数据类型测试

函数	说明
<code>issctype(rep)</code>	确定给定对象是否表示标量数据类型。
<code>issubdtype(arg1, arg2)</code>	如果第一个参数是类型层次结构中较低/相等的类型代码，则返回True。
<code>issubdtype(arg1, arg2)</code>	确定第一个参数是否是第二个参数的子类。
<code>issubclass_(arg1, arg2)</code>	确定一个类是否是第二类的子类。
<code>find_common_type(array_types, scalar_types)</code>	按照标准强制规则确定通用类型。

\$6.5 其它

函数	说明
<code>typename(char)</code>	返回给定数据类型代码的描述。
<code>sctype2char(sctype)</code>	返回标量dtype的字符串表示形式。
<code>mintypecode(typechars[, typeset, default])</code>	返回给定类型可以安全转换到的最小大小类型的字符。

\$7 可选的Scipy加速

```
from numpy.dual
import ...
```

scipy可以被构建为使用加速库或其他改进库来实现FFTs、线性代数和特殊函数。

\$7.1 线性代数

函数	说明
cholesky(a)	乔利斯基分解。
det(a)	计算数组的行列式。
eig(a)	计算方阵的特征值和右特征向量。
eigh(a[, UPLO])	返回埃尔米特矩阵或对称矩阵的特征值和特征向量。
eigvals(a)	计算一般矩阵的特征值。
eigvalsh(a[, UPLO])	计算埃尔米特矩阵或实对称矩阵的特征值。
inv(a)	计算矩阵的(乘法)逆。
lstsq(a, b[, rcond])	将最小二乘解返回线性矩阵方程。
norm(x[, ord, axis, keepdims])	矩阵或向量范数。
pinv(a[, rcond])	计算矩阵的(摩尔-彭罗斯)伪逆。
solve(a, b)	求解线性矩阵方程或线性标量方程组。
svd(a[, full_matrices, compute_uv])	奇异值分解。

\$7.2 快速傅里叶变换

函数	说明
fft(a[, n, axis, norm])	计算一维离散傅立叶变换。
fft2(a[, s, axes, norm])	计算二维离散傅立叶变换
fftn(a[, s, axes, norm])	计算N维离散傅立叶变换。
ifft(a[, n, axis, norm])	计算一维离散傅立叶逆变换。
ifft2(a[, s, axes, norm])	计算二维离散傅立叶逆变换。
ifftn(a[, s, axes, norm])	计算N维离散傅立叶逆变换。

\$7.3 特殊函数

函数	说明
i0(x)	第一类修正贝塞尔函数，0阶。

\$8 浮点错误处理

函数	说明
seterr([all, divide, over, under, invalid])	设置浮点错误的处理方式。
geterr()	获取当前处理浮点错误的方式。
seterrcall(func)	设置浮点错误回调函数或日志对象。
geterrcall()	返回用于浮点错误的当前回调函数。
errstate(**kwargs)	浮点错误处理的上下文管理器。
seterrobj(errobj)	设置定义浮点错误处理的对象。
geterrobj()	返回定义浮点错误处理的当前对象。

\$9 离散傅里叶变换 from numpy.fft import ...

\$9.1 标准快速傅立叶变换

函数	说明
fft(a[, n, axis, norm])	计算一维离散傅立叶变换。
ifft(a[, n, axis, norm])	计算一维离散傅立叶逆变换。
fft2(a[, s, axes, norm])	计算二维离散傅立叶变换
ifft2(a[, s, axes, norm])	计算二维离散傅立叶逆变换。
fftn(a[, s, axes, norm])	计算N维离散傅立叶变换。
ifftn(a[, s, axes, norm])	计算N维离散傅立叶逆变换。

\$9.2 实FFTs

函数	说明
rfft(a[, n, axis, norm])	计算真实输入的一维离散傅立叶变换。
irfft(a[, n, axis, norm])	计算实际输入的n点DFT的倒数。
rfft2(a[, s, axes, norm])	计算真实阵列的二维FFT。
irfft2(a[, s, axes, norm])	计算真实阵列的二维逆FFT。
rfftn(a[, s, axes, norm])	计算真实输入的N维离散傅立叶变换。
irfftn(a[, s, axes, norm])	计算实际输入的N维FFT的倒数。

\$9.3 埃尔米特快速傅立叶变换

函数	说明
hfft(a[, n, axis, norm])	计算具有厄米对称的信号的正FFT，即实谱。
ihfft(a[, n, axis, norm])	计算具有厄米对称的信号的正FFT。

\$9.4 其它

函数	说明
fftfreq(n[, d])	返回离散傅立叶变换采样频率。
rfftfreq(n[, d])	返回离散傅立叶变换采样频率(用于rfft、irfft)。
fftshift(x[, axes])	将零频率分量移动到频谱中心。
ifftshift(x[, axes])	fftshift反向操作。

\$10 财务

函数	说明
fv(rate, nper, pmt, pv[, when])	计算未来值。
pv(rate, nper, pmt[, fv, when])	计算现值。
npv(rate, values)	返回现金流序列的净现值。
pmt(rate, nper, pv[, fv, when])	根据贷款本金加利息计算付款。
ppmt(rate, per, nper, pv[, fv, when])	根据贷款本金计算付款。
ipmt(rate, per, nper, pv[, fv, when])	计算付款的利息部分。
irr(values)	返回内部收益率(IRR)。
mirr(values, finance_rate, reinvest_rate)	修正内部收益率。
nper(rate, pmt, pv[, fv, when])	计算定期付款的数量。
rate(nper, pmt, pv, fv[, when, guess, tol, ...])	计算每期利率。

\$11 功能程序设计

函数	说明
<code>apply_along_axis(func1d, axis, arr, *args, ...)</code>	沿给定轴将函数应用于二维切片。
<code>apply_over_axes(func, a, axes)</code>	在多个轴上重复应用一个函数。
<code>vectorize(pyfunc[, otypes, doc, excluded, ...])</code>	广义函数类。
<code>frompyfunc(func, nin, nout)</code>	获取任意Python函数并返回NumPy ufunc。
<code>piecewise(x, condlist, funclist, *args, **kw)</code>	评估分段定义的函数。

\$12 索引

\$12.1 生成索引数组

函数	说明
<code>_</code>	将切片对象沿第二轴平移为串联。
<code>r_</code>	将切片对象沿第一轴平移为串联。
<code>s_</code>	建立数组索引元组的更好方法。
<code>nonzero(a)</code>	返回非零元素的索引。
<code>where(condition, [x, y])</code>	根据条件，返回x或y中的元素。
<code>indices(dimensions[, dtype])</code>	返回表示网格索引的数组。
<code>ix_(*args)</code>	从多个序列构建开放网格。
<code>ogrid</code>	返回开放多维“网格”的 <code>nd_grid</code> 实例。
<code>ravel_multi_index(multi_index, dims[, mode, ...])</code>	将索引数组元组转换为平面索引数组，将边界模式应用于多索引。
<code>unravel_index(indices, dims[, order])</code>	将平面索引或平面索引数组转换为坐标数组元组。
<code>diag_indices(n[, ndim])</code>	返回索引以访问数组的主对角线。
<code>diag_indices_from(arr)</code>	返回索引以访问n维数组的主对角线。
<code>mask_indices(n, mask_func[, k])</code>	给定掩蔽函数，返回访问 (n, n) 个数组的索引。
<code>tril_indices(n[, k, m])</code>	返回 (n, m) 数组下三角形的索引。
<code>tril_indices_from(arr[, k])</code>	返回arr的下三角形的索引。
<code>triu_indices(n[, k, m])</code>	返回 (n, m) 数组上三角形的索引。
<code>triu_indices_from(arr[, k])</code>	返回arr上三角形的索引。

\$12.2 类似索引的操作

函数	说明
<code>take(a, indices[, axis, out, mode])</code>	沿轴从数组中提取元素。
<code>choose(a, choices[, out, mode])</code>	从索引数组和一组可供选择的数组构造一个数组。
<code>compress(condition, a[, axis, out])</code>	沿给定轴返回数组的选定切片。
<code>diag(v[, k])</code>	提取对角线或构造对角线数组。
<code>diagonal(a[, offset, axis1, axis2])</code>	返回指定的对角线。
<code>select(condlist, choicelist[, default])</code>	根据条件返回从choicelist中的元素中提取的数组。
<code>lib.stride_tricks.as_strided(x[, shape, ...])</code>	以给定的形状和步幅创建数组视图。

\$12.3 将数据插入数组

函数	说明
<code>place(arr, mask, vals)</code>	根据条件值和输入值更改数组的元素。
<code>put(a, ind, v[, mode])</code>	用给定值替换数组的指定元素。
<code>putmask(a, mask, values)</code>	根据条件值和输入值更改数组的元素。
<code>fill_diagonal(a, val[, wrap])</code>	填充给定维度数组的主对角线。

\$12.4 迭代数组

函数	说明
<code>nditer</code>	高效的多维迭代器对象对数组进行迭代。
<code>ndenumerate(arr)</code>	多维索引迭代器。
<code>ndindex(*shape)</code>	索引数组的N维迭代器对象。
<code>flatiter</code>	要在数组上迭代的平面迭代器对象。
<code>lib.Arrayiterator(var[, buf_size])</code>	大数组缓冲迭代器。

\$13 输入输出

\$13.1 NumPy二进制文件(NPY，NPZ)

函数	说明
load(file[, mmap_mode, allow_pickle, ...])	加载数组
save(file, arr[, allow_pickle, fix_imports])	将数组保存到 NumPy 中的二进制文件中。npz格式。
savez(file, *args, **kwds)	将几个阵列保存到未压缩的单个文件中。npz格式。
savez_compressed(file, *args, **kwds)	将几个数组压缩保存到单个文件中。npz格式。

\$13.2 文本文件

函数	说明
loadtxt(fname[, dtype, comments, delimiter, ...])	从文本文件加载数据。
savetxt(fname, X[, fmt, delimiter, newline, ...])	将数组保存到文本文件中。
genfromtxt(fname[, dtype, comments, ...])	从文本文件加载数据，缺失值按指定处理。
fromregex(file, regexp, dtype[, encoding])	使用正则表达式解析从文本文件构造数组。
fromstring(string[, dtype, count, sep])	从字符串中的文本数据初始化的新一维数组。
ndarray.tofile(fid[, sep, format])	将数组以文本或二进制形式写入文件(默认)。
ndarray.tolist()	将数组作为(可能嵌套的)列表返回。

\$13.3 原始二进制文件

函数	说明
fromfile(file[, dtype, count, sep])	从文本或二进制文件中的数据构造数组。
ndarray.tofile(fid[, sep, format])	将数组以文本或二进制形式写入文件(默认)。

\$13.4 字符串格式

函数	说明
array2string(a[, max_line_width, precision, ...])	返回数组的字符串表示形式。
array_repr(arr[, max_line_width, precision, ...])	返回数组的字符串表示形式。
array_str(a[, max_line_width, precision, ...])	返回数组中数据的字符串表示形式。
format_float_positional(x[, precision, ...])	在位置符号中将浮点标量格式化为十进制字符串。
format_float_scientific(x[, precision, ...])	在科学记数法中将浮点标量格式化为十进制字符串。

\$13.5 内存映射文件

函数	说明
mmap	创建存储到磁盘上二进制文件中存储的阵列的内存映射。

\$13.6 文本格式选项

函数	说明
set_printoptions([precision, threshold, ...])	设置打印选项。
get_printoptions()	返回当前打印选项。
set_string_function(f[, repr])	设置一个Python函数，用于漂亮的打印数组。

\$13.7 基-n表示

函数	说明
binary_repr(num[, width])	将输入数字的二进制表示形式返回为字符串。
base_repr(number[, base, padding])	返回给定基本系统中数字的字符串表示形式。

\$13.8 数据源

函数	说明
DataSource([destpath])	一般资料来源档案(档案、http、FTP、...)。

\$14. 线性代数from numpy.linalg

```
import...
```

\$14.1 矩阵和向量乘积

函数	说明
<code>dot(a, b[, out])</code>	两个数组的点积。
<code>linalg.multi_dot(arrays)</code>	在单个函数调用中计算两个或多个数组的点积，同时自动选择最快的求值顺序。
<code>vdot(a, b)</code>	返回两个向量的点积。
<code>inner(a, b)</code>	两个数组的内积。
<code>outer(a, b[, out])</code>	计算两个向量的外积。
<code>matmul(a, b[, out])</code>	两个数组的矩阵乘积。
<code>tensordot(a, b[, axes])</code>	计算阵列 $\geq 1 - d$ 的沿指定轴的张量点积
<code>einsum(subscripts, *operands[, out, dtype, ...])</code>	评估操作数上的爱因斯坦求和约定。
<code>einsum_path(subscripts, *operands[, optimize])</code>	通过考虑中间数组的创建来评估einsum表达式的最低成本收缩顺序。
<code>linalg.matrix_power(M, n)</code>	将方阵提高到(整数)幂n
<code>kron(a, b)</code>	两个数组的kronecker乘积。

\$14.2 分解

函数	说明
<code>linalg.cholesky(a)</code>	乔利斯基分解。
<code>linalg.qr(a[, mode])</code>	计算矩阵的QR分解。
<code>linalg.svd(a[, full_matrices, compute_uv])</code>	奇异值分解。

\$14.3 矩阵特征值

函数	说明
<code>linalg.eig(a)</code>	计算方阵的特征值和右特征向量。
<code>linalg.eigh(a[, UPLO])</code>	返回埃尔米特矩阵或对称矩阵的特征值和特征向量。
<code>linalg.eigvals(a)</code>	计算一般矩阵的特征值。
<code>linalg.eigvalsh(a[, UPLO])</code>	计算埃尔米特矩阵或实对称矩阵的特征值。

\$14.4 其他数字

函数	说明
<code>linalg.norm(x[, ord, axis, keepdims])</code>	矩阵或向量范数。
<code>linalg.cond(x[, p])</code>	计算矩阵的条件数。
<code>linalg.det(a)</code>	计算数组的行列式。
<code>linalg.matrix_rank(M[, tol, hermitian])</code>	用奇异值分解法求数组的矩阵秩
<code>linalg.slogdet(a)</code>	计算数组行列式的符号和(自然)对数。
<code>trace(a[, offset, axis1, axis2, dtype, out])</code>	沿数组对角线返回总和。

\$14.5 求解方程和逆矩阵

函数	说明
<code>linalg.solve(a, b)</code>	求解线性矩阵方程或线性标量方程组。
<code>linalg.tensorsolve(a, b[, axes])</code>	求解张量方程
<code>linalg.lstsq(a, b[, rcond])</code>	将最小二乘解返回线性矩阵方程。
<code>linalg.inv(a)</code>	计算矩阵的(乘法)逆。
<code>linalg.pinv(a[, rcond])</code>	计算矩阵的(摩尔-彭罗斯)伪逆。
<code>linalg.tensorinv(a[, ind])</code>	计算N维数组的“逆”。

\$14.6 其它

函数	说明
<code>linalg.LinAlgError</code>	<code>linalg</code> 函数引发的异常派生对象。

\$15 逻辑运算

\$15.1 真值检验

函数	说明
<code>all(a[, axis, out, keepdims])</code>	测试沿给定轴的所有数组元素是否评估为True。
<code>any(a[, axis, out, keepdims])</code>	测试沿给定轴的任何数组元素是否评估为True。

\$15.2 数组内容

函数	说明
isfinite(x, /[, out, where, casting, order, ...])	测试元素的有限性(不是无穷大或不是数字)。
isinf(x, /[, out, where, casting, order, ...])	以元素方式测试正无穷大或负无穷大。
isnan(x, /[, out, where, casting, order, ...])	对NaN进行逐个元素的测试，并将结果返回为布尔数组。
isnat(x, /[, out, where, casting, order, ...])	逐个元素测试NaT (不是时间)，并将结果作为布尔数组返回。
isneginf(x[, out])	逐个元素测试负无穷大，结果返回为bool数组。
isposinf(x[, out])	测试元素为正无穷大，返回结果为bool数组。

\$15.3 阵列式测试

函数	说明
iscomplex(x)	返回bool数组，如果输入元素复杂，则返回True。
iscomplexobj(x)	检查复数类型或复数数组。
isfortran(a)	如果数组是Fortran连续的但不是C连续的，则返回True。
isreal(x)	返回bool数组，如果输入元素为实，则返回True。
isrealobj(x)	如果x不是复杂类型或复数数组，则返回True。
isscalar(num)	如果num的类型是标量类型，则返回True。

\$15.4 逻辑运算

函数	说明
logical_and(x1, x2, /[, out, where, ...])	逐元素计算x1和x2的真值。
logical_or(x1, x2, /[, out, where, casting, ...])	逐元素计算x1或x2的真值。
logical_not(x, /[, out, where, casting, ...])	不按x元素计算真值。
logical_xor(x1, x2, /[, out, where, ...])	逐元素计算x1 XOR x2的真值。

\$15.5 比较

函数	说明
<code>allclose(a, b[, rtol, atol, equal_nan])</code>	如果两个数组在容差内元素相等，则返回True。
<code>isclose(a, b[, rtol, atol, equal_nan])</code>	返回一个布尔数组，其中两个数组在容差内按元素相等。
<code>array_equal(a1, a2)</code>	如果两个数组具有相同的形状和元素，则为true，否则为False。
<code>array_equiv(a1, a2)</code>	如果输入数组形状一致且所有元素相等，则返回True。
<code>greater(x1, x2, /[, out, where, casting, ...])</code>	逐元素返回($x1 > x2$)的真值。
<code>greater_equal(x1, x2, /[, out, where, ...])</code>	逐元素返回($x1 \geq x2$)的真值。
<code>less(x1, x2, /[, out, where, casting, ...])</code>	逐元素返回($x1 < x2$)的真值。
<code>less_equal(x1, x2, /[, out, where, casting, ...])</code>	逐元素返回($x1 \leq x2$)的真值。
<code>equal(x1, x2, /[, out, where, casting, ...])</code>	按元素返回($x1 == x2$)。
<code>not_equal(x1, x2, /[, out, where, casting, ...])</code>	返回($x1 \neq x2$)元素方式。

\$16 矩阵库 `from numpy.matlib` `import ...`

此模块包含numpy命名空间中的所有函数，只是返回的是matrix对象而不是ndarrays。

\$16.1 在numpy命名空间并返回matrix函数

函数	说明
<code>mat(data[, dtype])</code>	将输入解释为矩阵。
<code>matrix(data[, dtype, copy])</code>	从类似阵列的物件或资料字符串传回矩阵。
<code>asmatrix(data[, dtype])</code>	将输入解释为矩阵。
<code>bmat(obj[, ldict, gdict])</code>	从字符串、嵌套序列或数组构建矩阵对象。

\$16.2 MATLAB中的替换函数

函数	说明
<code>empty(shape[, dtype, order])</code>	返回给定形状和类型的新矩阵，而不初始化条目。
<code>zeros(shape[, dtype, order])</code>	返回给定形状和类型的矩阵，用零填充。
<code>ones(shape[, dtype, order])</code>	一矩阵。
<code>eye(n[, M, k, dtype, order])</code>	返回对角线上为1、别处为0的矩阵。
<code>identity(n[, dtype])</code>	返回给定大小的平方单位矩阵。
<code>repmat(a, m, n)</code>	重复 阵列或矩阵MxN次。
<code>rand(*args)</code>	返回给定形状的随机值矩阵。
<code>randn(*args)</code>	返回带有“标准正态”分布数据的随机矩阵。

\$17 随机数

\$17.1 简单随机数据

函数	说明
<code>rand(d0, d1, ..., dn)</code>	给定形状中的随机值。
<code>randn(d0, d1, ..., dn)</code>	从“标准正态”分布中返回一个或多个样本。
<code>randint(low[, high, size, dtype])</code>	返回从低(含)到高(含)的随机整数。
<code>random_integers(low[, high, size])</code>	NP . int类型的随机整数，介于低和高之间，包括在内。
<code>random_sample([size])</code>	返回半开区间 [0.0, 1.0) 中的随机浮动。
<code>random([size])</code>	返回半开区间 [0.0, 1.0) 中的随机浮动。
<code>ranf([size])</code>	返回半开区间 [0.0, 1.0) 中的随机浮动。
<code>sample([size])</code>	返回半开区间 [0.0, 1.0) 中的随机浮动。
<code>choice(a[, size, replace, p])</code>	从给定的一维数组生成随机样本
<code>bytes(length)</code>	返回随机字节。

\$17.2 排列

函数	说明
<code>shuffle(x)</code>	通过移动序列的内容就地修改序列。
<code>permutation(x)</code>	随机置换序列，或返回置换范围。

\$17.3 分布

函数	说明
beta(a, b[, size])	从Beta分布中抽取样本。
binomial(n, p[, size])	从二项分布中抽取样本。
chisquare(df[, size])	从卡方分布中抽取样本。
dirichlet(alpha[, size])	从Dirichlet分布中抽取样本。
exponential([scale, size])	从指数分布中抽取样本。
f(dfnum, dfden[, size])	从F分布中抽取样本。
gamma(shape[, scale, size])	从伽玛分布中抽取样本。
geometric(p[, size])	从几何分布中抽取样本。
gumbel([loc, scale, size])	从秋葵分布中抽取样本。
hypergeometric(ngood, nbad, nsample[, size])	从超几何分布中提取样本。
laplace([loc, scale, size])	从拉普拉斯分布或双指数分布中抽取具有指定位置(或平均值)和标度(衰减)的样本。
logistic([loc, scale, size])	从物流配送中抽取样本。
lognormal([mean, sigma, size])	从对数正态分布中抽取样本。
logseries(p[, size])	从对数级数分布中抽取样本。
multinomial(n, pvals[, size])	从多项式分布中抽取样本。
multivariate_normal(mean, cov[, size, ...])	从多元正态分布中抽取随机样本。
negative_binomial(n, p[, size])	从负二项分布中抽取样本。
noncentral_chisquare(df, nonc[, size])	从非中心卡方分布中抽取样本。
noncentral_f(dfnum, dfden, nonc[, size])	从非中心F分布中抽取样本。
normal([loc, scale, size])	从正态(高斯)分布中抽取随机样本。
pareto(a[, size])	从Pareto II或Lomax分布中抽取具有特定形状的本。
poisson([lam, size])	从泊松分布中抽取样本。
power(a[, size])	在[0, 1]从正指数为a - 1的幂分布中抽取样本。
rayleigh([scale, size])	从瑞利分布中抽取样本。
standard_cauchy([size])	从模式= 0的标准Cauchy分布中抽取样本。
standard_exponential([size])	从标准指数分布中抽取样本。
standard_gamma(shape[, size])	从标准伽马分布中抽取样本。

函数	说明
standard_normal([size])	从标准正态分布中抽取样本(平均值= 0， stdev = 1)。
standard_t(df[, size])	从具有df自由度的标准学生t分布中抽取样本。
triangular(left, mode, right[, size])	从[左、右区间的三角形分布中抽取样本]。
uniform([low, high, size])	从均匀分布中抽取样本。
vonmises(mu, kappa[, size])	从冯·米塞斯分布中抽取样本。
wald(mean, scale[, size])	从瓦尔德分布或逆高斯分布中提取样本。
weibull(a[, size])	从威布尔分布中抽取样本。
zipf(a[, size])	从Zipf分发中抽取样本。

\$17.4 随机发生器

函数	说明
RandomState([seed])	梅森捻线机伪随机数发生器的容器。
seed([seed])	给发电机播种。
get_state()	返回表示生成器内部状态的元组。
set_state(state)	从元组中设置生成器的内部状态。

\$18 集合操作

函数	说明
unique(ar[, return_index, return_inverse, ...])	查找数组的唯一元素。
in1d(ar1, ar2[, assume_unique, invert])	测试一维数组的每个元素是否也存在于第二个数组中。
intersect1d(ar1, ar2[, assume_unique])	找到两个数组的交集。
isin(element, test_elements[, ...])	计算 test_elements 中的元素，仅在元素上广播。
setdiff1d(ar1, ar2[, assume_unique])	找出两个数组的集合差。
setxor1d(ar1, ar2[, assume_unique])	查找两个数组的异或集。
union1d(ar1, ar2)	找到两个数组的并集。

\$19 排序、搜索、计数

\$19.1 整理

函数	说明
<code>sort(a[, axis, kind, order])</code>	返回数组的排序副本。
<code>lexsort(keys[, axis])</code>	使用键序列执行间接排序。
<code>argsort(a[, axis, kind, order])</code>	返回对数组进行排序的索引。
<code>ndarray.sort([axis, kind, order])</code>	就地排序数组。
<code>msort(a)</code>	返回沿第一轴排序的数组副本。
<code>sort_complex(a)</code>	首先使用实部，然后使用虚部对复杂数组进行排序。
<code>partition(a, kth[, axis, kind, order])</code>	返回数组的分区副本。
<code>argpartition(a, kth[, axis, kind, order])</code>	使用kind关键字指定的算法沿给定轴执行间接分区。

\$19.2 搜索

函数	说明
<code>argmax(a[, axis, out])</code>	返回沿轴的最大值的索引。
<code>nanargmax(a[, axis])</code>	忽略NaNs返回指定轴上最大值的索引。
<code>argmin(a[, axis, out])</code>	返回沿轴的最小值的索引。
<code>nanargmin(a[, axis])</code>	忽略NaNs返回指定轴上最小值的索引。
<code>argwhere(a)</code>	查找非零数组元素的索引，按元素分组。
<code>nonzero(a)</code>	返回非零元素的索引。
<code>flatnonzero(a)</code>	返回在a的扁平版本中非零的索引
<code>where(condition, [x, y])</code>	根据条件，返回x或y中的元素。
<code>searchsorted(a, v[, side, sorter])</code>	查找应该插入元素以维持秩序的索引。
<code>extract(condition, arr)</code>	返回满足某些条件的数组元素。

\$19.3 计数

函数	说明
<code>count_nonzero(a[, axis])</code>	统计数组a中非零值的数量

\$20 统计学

\$20.1 次序统计

函数	说明
<code>amin(a[, axis, out, keepdims])</code>	沿轴返回数组的最小值或最小值。
<code>amax(a[, axis, out, keepdims])</code>	返回数组的最大值或沿轴的最大值。
<code>nanmin(a[, axis, out, keepdims])</code>	返回数组的最小值或沿轴的最小值，忽略任何南线。
<code>nanmax(a[, axis, out, keepdims])</code>	沿轴返回数组的最大值或最大值，忽略任何南线。
<code>ptp(a[, axis, out])</code>	沿轴的值范围(最大-最小)。
<code>percentile(a, q[, axis, out, ...])</code>	沿指定轴计算数据的qth百分位。
<code>nanpercentile(a, q[, axis, out, ...])</code>	计算沿指定轴的数据的qth百分位，而忽略nan值。

\$20.2 平均数和差异

函数	说明
<code>median(a[, axis, out, overwrite_input, keepdims])</code>	计算沿指定轴的中间值。
<code>average(a[, axis, weights, returned])</code>	沿指定轴计算加权平均值。
<code>mean(a[, axis, dtype, out, keepdims])</code>	沿指定轴计算算术平均值。
<code>std(a[, axis, dtype, out, ddof, keepdims])</code>	计算沿指定轴的标准偏差。
<code>var(a[, axis, dtype, out, ddof, keepdims])</code>	计算沿指定轴的方差。
<code>nanmedian(a[, axis, out, overwrite_input, ...])</code>	计算沿指定轴的中间值，而忽略南社。
<code>nanmean(a[, axis, dtype, out, keepdims])</code>	沿指定轴计算算术平均值，忽略南社。
<code>nanstd(a[, axis, dtype, out, ddof, keepdims])</code>	计算沿指定轴的标准偏差，而忽略NaNs。
<code>nanvar(a[, axis, dtype, out, ddof, keepdims])</code>	计算沿指定轴的方差，而忽略NaNs。

\$20.3 相关

函数	说明
<code>corrcoef(x[, y, rowvar, bias, ddof])</code>	返回皮尔逊积-矩相关系数。
<code>correlate(a, v[, mode])</code>	两个一维序列的互相关。
<code>cov(m[, y, rowvar, bias, ddof, fweights, ...])</code>	给定数据和权重，估计协方差矩阵。

\$20.4 直方图

函数	说明
histogram(a[, bins, range, normed, weights, ...])	计算一组数据的直方图。
histogram2d(x, y[, bins, range, normed, weights])	计算两个数据样本的二维直方图。
histogramdd(sample[, bins, range, normed, ...])	计算一些数据的多维直方图。
bincount(x[, weights, minlength])	统计非负整数数组中每个值的出现次数。
digitize(x, bins[, right])	返回输入数组中每个值所属的箱的索引。

公众号：潮汐朝夕



记录一个算法工程师的职业生涯
文章内容：金融科技; 风控&合规&黑产; 读书笔记; CS&AI技能学习