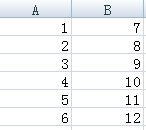
1.描述性统计

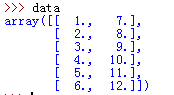


首先导入这两个库文件

数据为







接着使用loadtxt函数对文件进行导入，具体函数做如下讨论









以上可以看出通过控制usecols=（A，）中A的值，可以控制取哪一列，注意即便是只有一列也要加上逗号，因为这个函数不能输入整型数据







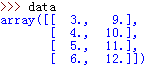


通过控制skiprows=B中的B，来控制数据跳过多少行









可见，unpack为True，则数据发生转置，否则数据不变

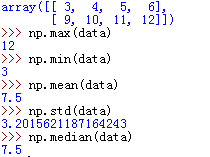




通过dtype来设置数据格式，上者格式为浮点数，下者为INT

通过数组提供的方法与函数均可以进行描述性统计

numpy中的函数：

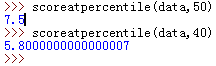


这依次为求最大值、最小值、平均数、标准差和中位数

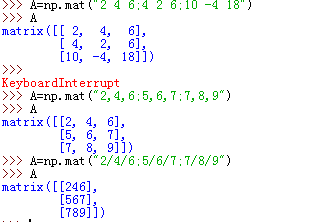
数组提供的方法：







2.线性代数运算



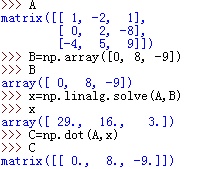
mat函数可以将数据变为矩阵，但是如上图根据分隔符的不同，会变成不同的矩阵





numpy模块中的linalg子模块中的inv函数，可以帮助求矩阵的逆，值得注意的是上图以/分割的矩阵无法求逆

3.解线性方程组



linalg中的solve可以解线性方程组，DOT函数可以求点乘

4.求特征向量与特征值

>>> a

matrix([[ 3, -2],

[ 1, 0]])

>>> b1=np.linalg.eigvals(a)

>>> b1

array([ 2., 1.])

linalg中的eigvals（）函数可以直接求方阵的所有特征值，如上a式就有2和1两个特征值

>>> a=np.mat("3 -2 2;1 0 3;1 5 6")

>>> a

matrix([[ 3, -2, 2],

[ 1, 0, 3],

[ 1, 5, 6]])

>>> c,d=np.linalg.eig(a)

>>> c

array([ 8.20335826, 2.32024623, -1.52360449])

>>> d

matrix([[-0.21175825, -0.95452459, -0.52692158],

[-0.35833383, -0.02756579, -0.67389968],

[-0.90926086, 0.29685506, 0.51789272]])

可以直接使用eig()函数同时将特征值和特征变量求出，注意d每一列为一个特征变量

5.二项分布模拟

import numpy as np

import matplotlib.pyplot as mat

cash=np.zeros(100000)

cash[0]=1000

s=np.random.binomial(9,0.5,size=len(cash))

for i in range(1,len(cash)):

if(s[i]<5):

cash[i]=cash[i-1]-1

i=i+1

elif(s[i]<10):

cash[i]=cash[i-1]+1

i=i+1

else:

print("error")

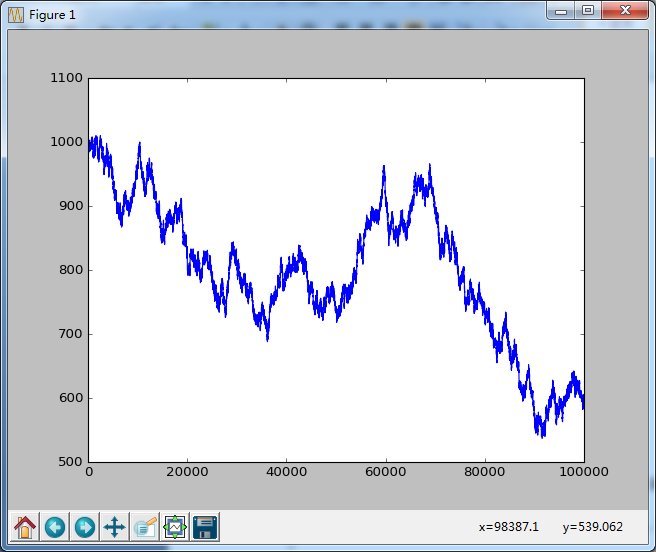
print(s.max(),s.min())

mat.plot(np.arange(len(cash)),cash)

mat.show()

以上代码中binomial(a,b,c)为随机数库中的二项分布模型，以硬币实验为例，共有a个硬币，每个硬币正面向上概率为b，共进行这样的实验c次

最终调用matplotlib.pyplot中的plot（）函数进行图形展示，如下：



6.正态采样与分布

由于这一章知识点较多，所以先介绍函数

np.random.normal(size=N) 表示生成N个基本符合正态分布的随机数

>>> N=100

>>> s=np.random.normal(size=N)

>>> s

array([ -6.76348393e-01, 1.19321570e-01, 2.54671814e+00,

·······

-1.44429945e-01, 1.57106015e+00, -1.30341809e+00,

1.33869810e+00])

>>> s.max()

2.5467181421182814

>>> s.min()

-2.66217671793139

plt.hist()函数可以根据数据直接生成直方图，如下

s1,s2,s3=plt.hist(s,N,normed=True,lw=1)

其中s为数据，N为将数据分为N组，即分为N箱，normed=True则表示生成的是频率直方图，即直方图中的每个条形的面积就代表着这一组数量占总数据数量的比例，简单说就是s1\*组距=P(这一组），lw表示边界线粗细；

s1返回每一组的顶点值，s2返回组与组之间的分割值

>>> s1,s2,s3=plt.hist(s,10,normed=True,lw=1)

>>> s1

array([ 0.01919793, 0.05759379, 0.15358344, 0.44155239, 0.3839586 ,

0.36476067, 0.21117723, 0.13438551, 0.11518758, 0.03839586])

>>> s2

array([-2.66217672, -2.14128723, -1.62039775, -1.09950826, -0.57861877,

-0.05772929, 0.4631602 , 0.98404968, 1.50493917, 2.02582866,

2.54671814])

>>> V=s2[2]-s2[1]

>>> V

0.52088948600496732

>>> V\*s1

array([ 0.01, 0.03, 0.08, 0.23, 0.2 , 0.19, 0.11, 0.07, 0.06, 0.02])

>>> s1,s2,s3=plt.hist(s,10,lw=1)

>>> s1

array([ 1., 3., 8., 23., 20., 19., 11., 7., 6., 2.])

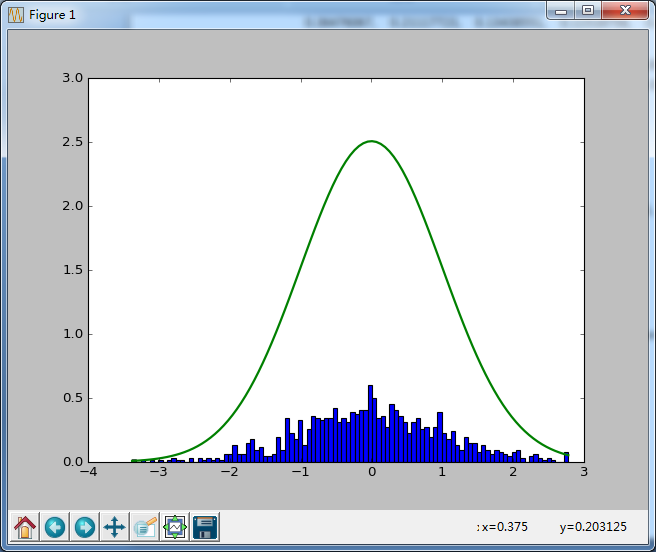
以下为完整将图生成

>>> s1,s2,s1=plt.hist(s,100,normed=True,lw=1)

>>> plt.plot(s2,(1/B\*np.sqrt(2\*np.pi))\*np.exp((-(s2-A)\*\*2)/(2\*B\*\*2)),lw=2)

[<matplotlib.lines.Line2D object at 0x053F5430>]

>>> plt.show()

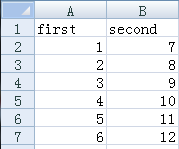
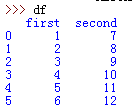


该重点为正态性分布的随机数生成、直方图的画法（plt.hist()的调用）

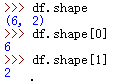
7.pandas初探之DataFrame

import pandas as pd

df=pd.io.parsers.read\_csv("test1.csv")

原始数据为，导入后为

shape属性：



len（）函数可以返回DF的行数



>>> df.columns

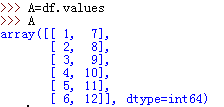
Index(['first', 'second'], dtype='object')

>>> df.index

RangeIndex(start=0, stop=6, step=1)

可以分别返回其列索引和行索引，在这里通常将列索引叫做列标题，行索引叫做索引。

返回各列的属性类型



用values函数可以以np数组的形式便利df，这也可以用作df数组与np数组的转化

8. pandas初探之Series

>>> s=df['first']

>>> s

0 1

1 2

2 3

3 4

4 5

5 6

Name: first, dtype: int64

>>> type(s)

<class 'pandas.core.series.Series'>

>>> s.dtype

dtype('int64')

>>> s.dtypes

dtype('int64')

>>> type(s)

<class 'pandas.core.series.Series'>

>>> s.shape

(6,)

>>> s.index

RangeIndex(start=0, stop=6, step=1)

>>> s.name

'first'

>>> s.values

array([1, 2, 3, 4, 5, 6], dtype=int64)