a = [1.0,2.0,3.0]

class Solution():

def describe(self, a):

n=len(a)

if n==0:

return [None,None,None,None]

sum=0.000000

ave=0.000000

for i in a:

sum+=i

ave=sum/n

if n==1:

return [round(ave,6),None,0,-3]

var=0.000000

si=0.000000

san=0.000000

for i2 in a:

var+=(i2-ave)\*\*2

si+=(i2-ave)\*\*4

san+=(i2-ave)\*\*3

svar=var/n

var=var/(n-1)

if var==0:

return [round(ave,6),round(var,6),None,None]

san=san/n

skew=san/(svar\*\*1.5)

si=si/n

kurt=si/(svar\*\*2)

return [round(ave,6),round(var,6),round(skew,6),round(kurt-3,6)]

p=Solution()

print p.describe(a)

1. 下面为针对蒙提霍尔三门问题进行模拟中奖概率的代码，请仔细阅读，理解代码含义，并试着修改Monty Hall程序来模拟最原始的游戏规则。蒙提霍尔三门问题规则如下：  
（1）参赛者会看见三扇关闭的门，其中一扇的后面有一辆汽车，选中后面有车的那扇门就可以赢得该汽车，另两扇门后则各藏有一只山羊  
（2）当参赛者选定了一扇门，但未去开启它的时候，知道门后情形的节目主持人会开启剩下两扇门的其中一扇，露出其中一只山羊  
（3）主持人其后会问参赛者要不要换另一扇仍然关上的门

import random

from log\_api import log

'''

You can delete the function MontyHall and write

your MontyHall

'''

def MontyHall(Dselect, Dchange):

Dcar = random.randint(1,3)

#hit at the first and not change the mind

if Dcar == Dselect and Dchange == 0:

return 1

#not hit at the first and not change the mind

elif Dcar != Dselect and Dchange == 0:

return 0

#hit at the first and change the mind

elif Dcar == Dselect and Dchange == 1:

return 0

#not hit at the first and change the mind

else:

return 1

if \_\_name\_\_ == '\_\_main\_\_':

#repeat 10000 times

n = 10000

#not sure whether to change the mind

win = 0

for i in range(n):

Dselect = random.randint(1,3)

Dchange = random.randint(0,1)

win = win + MontyHall(Dselect, Dchange)

log(float(win)/float(n))

#be sure not to change the mind

win = 0

for i in range(n):

Dselect = random.randint(1,3)

Dchange = 0

win = win + MontyHall(Dselect, Dchange)

log(float(win)/float(n))

#be sure to change the mind

win = 0

for i in range(n):

Dselect = random.randint(1,3)

Dchange = 1

win = win + MontyHall(Dselect, Dchange)

log(float(win)/float(n))

2. 下面为利用蒙特卡罗方法计算π的python实现代码，请仔细阅读，理解代码含义，并试着写出你自己版本的计算π的蒙特卡罗方法

import random

from log\_api import log

'''

You can delete the MontyCarlo and write your Monty Carlo

'''

#this function may spend a little time to execute

def MontyCarlo():

n = 1000000

k = 0

for i in range(n):

x = random.uniform(-1, 1)

y = random.uniform(-1, 1)

if x\*\*2 + y\*\*2 < 1:

k = k + 1

return 4 \* float(k) / float(n)

if \_\_name\_\_ == '\_\_main\_\_':

pi = MontyCarlo()

#print out the result

log(pi)

3. 利用matplotlib中plot\_api包创建柱状图

#-\*- coding:utf-8 -\*-

import numpy as np

import matplotlib.plot\_api as plt

#you can write your code here

def draw():

#get input data

menMeans = (20, 35, 30, 35, 27)

menStd = (2, 3, 4, 1, 2)

womenMeans = (25, 32, 34, 20, 25)

womenStd = (3, 5, 2, 3, 3)

ind = np.arange(5)

width = 0.35

# the histogram of the data

plt.bar(ind, menMeans, width, color='r')

plt.bar(ind+width, womenMeans, width, color='y')

#show image

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

draw()

4. 利用matplotlib中plot\_api包创建饼图，请完成如下练习：  
（1）请仔细阅读代码，理解其含义  
（2）pie函数中可以通过colors指定饼图颜色，修改一下colors参数，看看饼图怎么变化  
（3）pie函数中可以通过shadow参数指定饼图有无阴影，修改一下shadow参数，看看饼图怎么变化  
（4）试着画出自己的饼图吧！

#-\*- coding:utf-8 -\*-

import numpy as np

import matplotlib.plot\_api as plt

def get\_data():

# example data

mu = 100 # mean of distribution

sigma = 15 # standard deviation of distribution

x = mu + sigma \* np.random.randn(8)

return x

#you can write your code here

def draw():

#get input data

x = get\_data()

c = ('b', 'g', 'r', 'c', 'm', 'y', 'k', 'w')

# the pie chart of the data

plt.pie(x, colors=c, shadow=False)

#show image

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

draw()

5. 利用matplotlib中plot\_api包创建直方图，请完成如下练习：  
（1）hist函数中参数bins指定将输入x分为多少个区间进行统计，默认为10，修改一下bins参数，看看直方图怎么变化  
（2）hist函数中参数color指定直方图颜色，修改一下color参数为'red'，看看直方图怎么变化  
（3）hist函数中参数alpha指定直方图透明程度，为[0,1]之间浮点数，修改一下alpha参数，看看直方图怎么变化  
（4）试着画出自己的直方图吧！

#-\*- coding:utf-8 -\*-

import numpy as np

import matplotlib.plot\_api as plt

def get\_data():

# example data

mu = 100 # mean of distribution

sigma = 15 # standard deviation of distribution

x = mu + sigma \* np.random.randn(10000)

return x

#you can write your code here

def draw():

#get input data

x = get\_data()

# the histogram of the data

plt.hist(x, bins=50, color='g', alpha=0.5)

#show image

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

draw()

6．利用matplotlib中plot\_api包创建折线图，请完成如下练习：  
（1）请仔细阅读代码，理解其含义  
（2）plot函数中可以通过'r','g','y'等参数指定折线图颜色，修改一下'r--'参数，看看折线图怎么变化  
（3）plot函数中可以通过'-','--','-.','+'等指定折线图折线形状，修改一下'r--'参数，看看折线图怎么变化  
（4）试着画出自己的折线图吧！

#-\*- coding:utf-8 -\*-

import numpy as np

import matplotlib.plot\_api as plt

def get\_data():

# example data

mu = 100 # mean of distribution

sigma = 15 # standard deviation of distribution

x = mu + sigma \* np.random.randn(50)

return x

#you can write your code here

def draw():

#get input data

x = get\_data()

# the line chart of the data

plt.plot(x, 'r--')

#show image

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

draw()

7. 利用matplotlib中plot\_api包创建盒状图，请完成如下练习：  
（1）请仔细阅读代码，理解其含义  
（2）试着画出自己的盒状图吧！

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）关于plot\_api的使用教程参见http://121.41.106.89:8081/help#toc\_3

#-\*- coding:utf-8 -\*-

import numpy as np

import matplotlib.plot\_api as plt

def get\_data():

mu = 100

sigma = 15

x = mu + sigma \* np.random.randn(100)

return x

#you can write your code here

def draw():

#get data from mooctest.net

x = get\_data()

#create box plot chart

plt.boxplot(x)

#show image

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

draw()

8. 下面为分赌本问题的python实现代码，规则如下：  
（1）赌徒甲、乙两赌徒赌技相同，每局无平局，他们约定，谁先赢得十局则得到全部赌本。  
（2）当前甲赢了5局，乙赢了2局，因故终止赌博，请问他们按照怎样的比例分赌本？  
下面代码中已经给出此种情况下的实现代码，其中Bookie1函数模拟每次甲、乙赌博的情况，simulate1函数模拟进行10000次赌博后甲乙各自胜出概率  
（1）请仔细阅读，理解代码含义，并运行代码，看看结果如何？思考，如果甲、乙现在都是一局都没赢，Bookie1函数的参数应该怎么设置呢？  
（2）如果模拟5个人赌博的分赌本问题，Bookie2函数应该怎么写呢？  
（3）附加：如果甲乙赌技不相等情形下，每次比赛中甲胜出的概率为2/3，Bookie1函数又应该怎么写呢？

#-\*- coding:utf-8 -\*-

import sys

import random

from log\_api import log

'''

n: the total number of games to win

n1: the number of games player1 won

n2: the number of games player2 won

'''

def Bookie1(n, n1, n2):

for i in range(2\*n-n1-n2-1):#the number of games needed to end

D = random.randint(1,2)

if D==1:

n1+=1

elif D==2:

n2+=1

if n1==n:#player1 wins

return 1

if n2==n:#player2 wins

return 2

def Bookie2(n, n1, n2, n3, n4, n5):

#you can write your code here

pass

def simulate1():

n = 10000

win1 = 0

win2 = 0

for i in range(n):#simulate 10000 games

#simulate game whose total number is 10, and player1 won 5, player2 won 2

result = Bookie1(10,5,2)

if result==1:

win1+=1

elif result==2:

win2+=1

log('player1 wins: ' + str(float(win1)/float(n)))

log('player2 wins: ' + str(float(win2)/float(n)))

def simulate2():

n = 10000

win1 = 0

win2 = 0

win3 = 0

win4 = 0

win5 = 0

for i in range(n):#simulate 10000 games

#simulate game whose total number is 10, and player1 and player3 won 1

result = Bookie2(10,1,0,1,0,0)

if result==1:

win1+=1

elif result==2:

win2+=1

elif result==3:

win3+=1

elif result==4:

win4+=1

elif result==5:

win5+=1

log('player1 wins: ' + str(float(win1)/float(n)))

log('player2 wins: ' + str(float(win2)/float(n)))

log('player3 wins: ' + str(float(win3)/float(n)))

log('player4 wins: ' + str(float(win4)/float(n)))

log('player5 wins: ' + str(float(win5)/float(n)))

if \_\_name\_\_ == '\_\_main\_\_':

simulate1()

#simulate2()

9. 下面为庞加莱买面包问题的python实现代码，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）比较first\_year与second\_year结果，特别是平均值和偏度，看看有无出入？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.statas.norm(loc, scale)函数返回符合正态分布的随机变量X，接下来可以通过调用X.rvs(size)得到size次的随机取样值的序列，loc参数代表期望值，scale参数代表方差  
（3）numpy.mean(sequence)函数求取sequence序列的平均值  
（4）scipy.statas.skew(sequence)函数求取sequence序列的偏度值

#-\*- coding:utf-8 -\*-

import sys

import numpy

import scipy.stats as sta

import matplotlib.plot\_api as plt

from log\_api import log

def first\_year():

X=sta.norm(loc=950, scale=20)#generate random data in normal distribution whose expectation is 950 and standard deviation is 20

wbread=[]

for i in range(365):

x=X.rvs(size=100)

wbread.append(x[0])#get the random data for one day

log(numpy.mean(wbread))#print mean value

log(sta.skew(wbread))#print skew value

plt.hist(wbread,color='grey')

plt.savefig('first\_year.png')

def second\_year():

X=sta.norm(loc=950, scale=20)#generate random data in normal distribution whose expectation is 950 and standard deviation is 20

wbread=[]

for i in range(365):

x=X.rvs(size=100)

wbread.append(max(x))#get the random data for one day

log(numpy.mean(wbread))#print mean value

log(sta.skew(wbread))#print skew value

plt.hist(wbread,color='grey')

plt.savefig('second\_year.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

first\_year()

second\_year()

10. 二项分布是最重要的离散概率分布之一。假设有一种只有两个结果的试验，其成功概率为p,那么二项分布描述了进行n次伯努利试验，成功k次的概率。下面模拟了进行10次伯努利试验，每次成功概率为0.5，分别求最后成功次数为0、1、2...、10次的概率。在上述基础上，利用成功次数作为x轴，相应概率值作为y轴，画出其直方图，可大概体现二项分布的概率质量图的特征。请完成如下练习：  
（1）请仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）假如进行100次伯努利试验，每次成功概率为1/6，怎么修改代码中binom函数、arange函数的参数呢？  
（3）设想一种可使用二项分布计算概率的情景，并用代码实现吧！

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.binom(n,p)函数返回符合二项分布的离散随机变量rv,其独立试验次数为n，每次成功的概率为p，经常配合使用rv.pmf(x)计算其概率质量值，参数x为序列  
（3）numpy.arange(start,stop,step)返回给定step间隔的[start,stop)之间的均匀分布值序列

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

import matplotlib.plot\_api as plt

from scipy.stats import binom as B

def binom\_pmf():

rv=B(10, 0.5)#10 independent trials, probability of success is 0.5

x=np.arange(0, 11, 1)#return evenly spaced values within 1 interval between [0,11)

y=rv.pmf(x)#probability mass function

plt.bar(x, y, width=0.6, color='grey')#make bar chart

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

binom\_pmf()

11. 几何分布的一种定义为：每次伯努利试验成功的概率为p，试验k次才得到第一次成功的概率。下面模拟了每次伯努利试验成功的概率为0.2，分别求经过1、2...、10次才得到第一次成功的概率。在上述基础上，利用成功次数作为x轴，相应概率值作为y轴，画出其直方图，可大概体现几何分布的概率质量图的特征。请完成如下练习：  
（1）请仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）假如每次成功概率为3/4，最多经过100次伯努利试验才得到第一次成功，怎么修改代码中genom函数、arange函数的参数呢？  
（3）设想一种可使用几何分布计算概率的情景，并用代码实现吧！

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.geom(p)函数返回符合几何分布的离散随机变量rv,每次成功的概率为p，经常配合使用rv.pmf(x)计算其概率质量值，参数x为序列  
（3）numpy.arange(start,stop,step)返回给定step间隔的[start,stop)之间的均匀分布值序列

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

import matplotlib.plot\_api as plt

from scipy.stats import geom as G

def geom\_pmf():

rv=G(0.2)#probability of success is 0.2

x = np.arange(1, 11, 1)#return evenly spaced values within 1 interval between [1,11)

y = rv.pmf(x)#probability mass function

plt.bar(x, y, width=0.6, color='grey')#make bar chart

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

geom\_pmf()

12. 在二项分布中，如果试验次数n很大，每次试验成功概率p很小，其乘积np比较适中时，那么试验成功的次数的概率可以用泊松分布近似描述。在泊松分布中使用λ描述单位时间中随机事件的平均发生率，其中λ=np。下面模拟了事件发生率（每秒内事件平均发生次数）λ为4.5,观察时间为0-10秒的情况。在上述基础上，利用观察时间作为x轴，相应概率值作为y轴，画出其直方图，可大概体现泊松分布的概率质量图的特征。请完成如下练习：  
（1）请仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）假如单位时间内随机事件的平均发生率为1/3，怎么修改代码中poisson函数的参数呢？  
（3）设想一种可使用泊松分布计算概率的情景，并用代码实现吧！

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.poisson(λ)函数返回符合泊松分布的离散随机变量rv,单位时间内随机事件的平均发生率为λ，经常配合使用rv.pmf(x)计算其概率质量值，参数x为序列  
（3）numpy.arange(start,stop,step)返回给定step间隔的[start,stop)之间的均匀分布值序列

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

import matplotlib.plot\_api as plt

from scipy.stats import poisson as Pie

def poisson\_pmf():

rv=Pie(4.5)#the average incident is 4.5

x = np.arange(0, 11, 1)#return evenly spaced values within 1 interval between [1,11)

y = rv.pmf(x)#probability mass function

plt.bar(x, y, width=0.6, color='grey')#make bar chart

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

poisson\_pmf()

13. 下面模拟了期望值分别为0、-5、0，标准差分别为1、1、3的正态分布随机变量rv1、rv2、rv3。并在[-10,10]平均间隔地取100个值作为随机值。在上述基础上，利用随机值作为x轴，相应概率值作为y轴，画出其直方图，可大概体现正态分布的概率密度图的特征。请完成如下练习：  
（1）请仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）假如描述期望值为3，标准差为5的正态分布，怎么修改代码中norm函数的参数呢？  
（3）设想一种可使用正态分布计算概率的情景，并用代码实现吧！

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.norm(loc,scale)函数返回符合正态分布的期望值为loc、标准差为scale的随机变量rv，经常配合使用rv.pdf(x)计算其概率密度值，参数x为序列  
（3）numpy.linspace(start,stop,num)返回[start,stop)之间的均匀间隔的num个数字组成的序列

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

import matplotlib.plot\_api as plt

from scipy.stats import norm as N

def norm\_pdf():

x = np.linspace(-10, 10, 100)#return evenly spaced samples, calculated over the interval [-10,10]

rv1=N(loc=0, scale = 1)#the mean is 0, the standard deviation is 1

rv2=N(loc=-5, scale = 1)#the mean is -5, the standard deviation is 1

rv3=N(loc=0, scale = 3)#the mean is 0, the standard deviation is 3

plt.plot(x, rv1.pdf(x), color='green')#make chart

plt.plot(x, rv2.pdf(x), color='blue')#make chart

plt.plot(x, rv3.pdf(x), color='red')#make chart

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

norm\_pdf()

14. 下面模拟了分别服从参数1.5、1.0、0.5的指数分布随机变量rv1、rv2、rv3。并在[0,20]平均间隔地取100个值作为随机值。在上述基础上，利用随机值作为x轴，相应概率值作为y轴，画出其直方图，可大概体现指数分布的概率密度图的特征。请完成如下练习：  
（1）请仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）假如参数为5.0，怎么修改代码中expon函数的参数呢？  
（3）设想一种可使用指数分布计算概率的情景，并用代码实现吧！

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.expon(scale)函数返回符合指数分布的参数为scale的随机变量rv，经常配合使用rv.pdf(x)计算其概率密度值，参数x为序列  
（3）numpy.linspace(start,stop,num)返回[start,stop)之间的均匀间隔的num个数字组成的序列

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

import matplotlib.plot\_api as plt

from scipy.stats import expon as E

def expon\_pdf():

x = np.linspace(0, 20, 100)#return evenly spaced samples, calculated over the interval [0,20]

rv1=E(scale = 1.5)#the scale is 1.5

rv2=E(scale = 1.0)#the scale is 1.0

rv3=E(scale = 0.5)#the scale is 0.5

plt.plot(x, rv1.pdf(x), color='green')#make chart

plt.plot(x, rv2.pdf(x), color='blue')#make chart

plt.plot(x, rv3.pdf(x), color='red')#make chart

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

expon\_pdf()

15. 累计分布CDF是指随机变量小于或者等于某个数值的概率P（X<=x），即F(x)=P(X<=x)，为概率密度函数PDF的积分，下面模拟了服从参数1.0的指数分布随机变量rv，并分别求取其PDF、CDF值作为y轴，画出其直方图，比较PDF、CDF的特征。请完成如下练习：  
\n\t（1）请仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）怎么画出期望值为1.0，标准差为2.0满足正态分布的随机值的PDF和CDF呢？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.expon(scale)函数返回符合指数分布的参数为scale的随机变量rv，经常配合使用rv.pdf(x)计算其概率密度值，使用rv.cdf(x)计算其累计概率密度值，参数x为序列  
（3）numpy.linspace(start,stop,num)返回[start,stop)之间的均匀间隔的num个数字组成的序列

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

import matplotlib.plot\_api as plt

from scipy.stats import expon as E

def expon\_cdf\_pdf():

x = np.linspace(0, 5, 100)##return evenly spaced samples, calculated over the interval [0,5]

rv=E(scale = 1)#the scale is 1

plt.plot(x, rv.pdf(x), color='blue')#make pdf chart

plt.plot(x, rv.cdf(x), color='red') #make cdf chart

plt.savefig('fig.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

expon\_cdf\_pdf()

16. 下面利用exp(lam)函数指数分布的计算公式返回服从参数lam的指数分布的随机变量概率值，并分别在[0,1]和[0,5]之间的均匀间隔地取100个值分别作为x轴，利用自定义exp(lam)函数和内置expon()函数求出的相应概率值作为y轴，画出其直方图，对比分析。请完成如下练习：  
（1）请仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）怎么求取期望值为1.0，标准差为2.0满足正态分布的随机值呢？tips：自定义norm函数  
（3）试着自定义binom、geom、poisson函数吧！

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.expon(scale)函数返回符合指数分布的参数为scale的随机变量rv，经常配合使用rv.cdf(x)计算其累计概率密度值，参数x为序列  
（3）numpy.linspace(start,stop,num)返回[start,stop)之间的均匀间隔的num个数字组成的序列  
（4）random.random()函数返回[0,1)之间的浮点数

#-\*- coding:utf-8 -\*-

import sys

import math

import random

import numpy as np

import matplotlib.plot\_api as plt

from scipy.stats import expon as E

def exp(lam):

p=random.random()#return float in [0,1)

return -math.log(1-p)/lam#calculate expon value

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

#generate 100 random variables calculated by self-defined exp function

x1=[]

for i in range(100):

x1.append(exp(1))

x1=sorted(x1)

y1=np.linspace(0, 1, 100)

plt.plot(x1,y1,color='blue')

#generate 100 random varaibles calculated by inner-defined expon function

rv=E(scale=1)

x2=np.linspace(0, 5, 100)

plt.plot(x2, rv.cdf(x2), color='red')

plt.savefig('fig.png')#make chart

17. 下面为Matplotlib包进行3D作图的直方图示例，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）修改Z轴的值，运行代码，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）add\_subplot(\*args,\*\*kwargs)当设定参数projection='3d'时返回Axes3D对象  
（3）Axes3D.bar(left,height,zs=0,zdir=u'z',\*args,\*\*kwargs)绘制直方图，还可以通过color参数、alpha参数分别控制颜色与透明度

#-\*- coding:utf-8 -\*-

from mpl\_toolkits.mplot3d import Axes3D

import matplotlib.plot\_api as plt

import numpy as np

from log\_api import log

def bars\_3d():

fig = plt.figure()

#add a 3d subplot

ax = fig.add\_subplot(111, projection='3d')

for c, z in zip(['r', 'g', 'b', 'y'], [30, 20, 10, 0]):

xs = np.arange(20)

ys = np.random.rand(20)

#cs is an array to set color

cs = [c] \* len(xs)

# the first bar of each set will be colored cyan.

cs[0] = 'c'

ax.bar(xs, ys, zs=z, zdir='y', color=cs, alpha=0.8)

#set label

ax.set\_xlabel('X')

ax.set\_ylabel('Y')

ax.set\_zlabel('Z')

plt.savefig('figure.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

bars\_3d()

18. 下面为吸烟与健康关系问题的直方图python实现代码，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）修改Z轴的值，运行代码，看看结果如何？

输入说明:

无

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）add\_subplot(\*args,\*\*kwargs)当设定参数projection='3d'时返回Axes3D对象  
（3）Axes3D.bar3d(x,y,z,dx,dy,dz,color=u'b',\*args,\*\*kwargs)绘制多直方图，其中x,y,z为数组

#-\*- coding:utf-8 -\*-

from mpl\_toolkits.mplot3d import Axes3D

import matplotlib.plot\_api as plt

import numpy as np

from log\_api import log

def random\_variables():

fig = plt.figure()

#add a 3d subplot

ax = fig.add\_subplot(111, projection='3d')

#set X,Y,Z

dx=0.3

dy=0.3

dz=[0.02, 0.025, 0.35, 0.1, 0.15, 0.04, 0.25, 0.04, 0.025]

zpos=0

i=0

for xpos in [1, 2, 3]:

for c, ypos in zip(['r','y','g'], [-1, 0, 1]):

ax.bar3d(xpos, ypos, zpos, dx, dy, dz[i], color=c)

i=i+1

plt.savefig('random\_variables.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

random\_variables()

19．下面为二维均匀分布绘图的python代码实现，其X，Y处于长、宽均为[0,1]的正方形G内，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）修改X、Y，使得X、Y位于长为2，宽为1的长方形G内  
（3）对f(x,y)=1/π,x²+y²<=1代表的二维均匀分布进行图形展现，注意X、Y处于什么区域范围内

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）gca(\*\*kwargs)返回当前Axes对象，如果没有则创建一个返回，当设定参数projection='3d'时返回Axes3D对象  
（3）numpy.meshgrid(\*xi,\*\*kwargs)利用坐标向量生成并返回坐标矩阵  
（4）Axes3D.plot\_surface(X,Y,Z,\*args,\*\*kwargs)生成曲面图

#-\*- coding:utf-8 -\*-

from mpl\_toolkits.mplot3d import Axes3D

import matplotlib.plot\_api as plt

import numpy as np

from log\_api import log

def uniform\_distribution():

fig = plt.figure()

#add a 3d subplot

ax = fig.gca(projection='3d')

#set X,Y,Z

X = np.arange(0, 1, 0.01)

Y = np.arange(0, 1, 0.01)

#create coordinate matrices

X, Y = np.meshgrid(X, Y)

Z1 = 1

Z2 = 0

#create surface 1

surf = ax.plot\_surface(X, Y, Z1, color='b')

#create surface 2

surf = ax.plot\_surface(X, Y, Z2, color='r')

plt.savefig('uniform\_distribution.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

uniform\_distribution()

20. 下面为二维正态分布绘图的python代码实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）修改X轴方向正态分布的sigma，mu值，看看结果如何？  
（3）修改Y轴方向正态分布的sigma，mu值，看看结果如何？  
（4）修改sigmaxy值为1，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）add\_subplot(\*args,\*\*kwargs)当设定参数projection='3d'时返回Axes3D对象  
（3）numpy.meshgrid(\*xi,\*\*kwargs)利用坐标向量生成并返回坐标矩阵  
（4）matplotlib.mlab.bivariate\_normal(X,Y,sigmax=1.0,sigmay=1.0,mux=0.0,muy=0.0,sigmaxy=0.0)返回符合二维高斯分布的变量值  
（5）Axes3D.plot\_surface(X,Y,Z,\*args,\*\*kwargs)生成曲面图

#-\*- coding:utf-8 -\*-

from mpl\_toolkits.mplot3d import Axes3D

import matplotlib.plot\_api as plt

import matplotlib.mlab as mlab

import numpy as np

from log\_api import log

def bivariate\_normal():

fig = plt.figure()

#add a 3d subplot

ax = fig.add\_subplot(111, projection='3d')

#set X,Y,Z

x = np.linspace(-10, 10, 200)

y = x

X,Y = np.meshgrid(x, y)

#create bivariate gaussian distribution for equal shape X,Y

Z = mlab.bivariate\_normal(X, Y, 1, 5, 0, -5, 0)

#create surface

ax.plot\_surface(X, Y, Z, cmap='OrRd')

plt.savefig('bivariate\_normal.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

bivariate\_normal()

21. 下面为Matplotlib包进行3D作图的曲面图示例，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）删掉负责缩放的代码，即23-26行，看看结果如何？  
（3）对于概率密度函数f(x,y)=x²+y²，怎么样绘制其曲面图？  
（4）对于概率密度函数f(x,y)=6e^(-2x-3y),x>0,y>0，怎么样绘制其曲面图？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）gca(\*\*kwargs)返回当前Axes对象，如果没有则创建一个返回，当设定参数projection='3d'时返回Axes3D对象  
（3）numpy.meshgrid(\*xi,\*\*kwargs)利用坐标向量生成并返回坐标矩阵  
（4）Axes3D.plot\_surface(X,Y,Z,\*args,\*\*kwargs)生成曲面图  
（5）Axes3D.set\_zlim(bottom=None,top=None)设定Z轴下限、上限  
（6）mpl\_toolkits.mplot3d.axis3d.Zxis.set\_major\_locator(Locator)设定Z轴locator  
（7）matplotlib.ticker.LinearLocator(numticks=None,presets=None)通过numticks进行数据切割  
（8）mpl\_toolkits.mplot3d.axis3d.Zxis.set\_major\_formatter(Formatter)设定Z轴Formatter  
（9）matplotlib.ticker.FormatStrFormatter(fmt)利用'%操作符'格式对数据进行格式化  
（10）figure.colorbar(mappable,shrink=1.0,aspect=20)对mappable实例创建colorbar并返回，其中shrink参数调节缩放比例，aspect参数调节长短比例

#-\*- coding:utf-8 -\*-

from mpl\_toolkits.mplot3d import Axes3D

from matplotlib import cm

from matplotlib.ticker import LinearLocator, FormatStrFormatter

import matplotlib.plot\_api as plt

import numpy as np

from log\_api import log

def surface\_3d():

fig = plt.figure()

#add a 3d subplot

ax = fig.gca(projection='3d')

#set X,Y,Z

X = np.arange(-5, 5, 0.25)

Y = np.arange(-5, 5, 0.25)

X, Y = np.meshgrid(X, Y)

R = np.sqrt(X\*\*2 + Y\*\*2)

Z = np.sin(R)

#create surface

surf = ax.plot\_surface(X, Y, Z, rstride=1, cstride=1, cmap=cm.coolwarm,

linewidth=0, antialiased=False)

ax.set\_zlim(-1.01, 1.01)#set z limits

ax.zaxis.set\_major\_locator(LinearLocator(10))#set tick locator to linear locator

ax.zaxis.set\_major\_formatter(FormatStrFormatter('%.02f'))#set tick formatter to str formatter

fig.colorbar(surf, shrink=0.5, aspect=5)#add a colorbar to plot

plt.savefig('surface\_3d.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

surface\_3d()

22. 下面为二项分布的数字特征示例，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.binom(n,p)函数返回符合二项分布的离散随机变量rv,其独立试验次数为n，每次成功的概率为p  
（3）rv.mean()返回随机变量rv的均值  
（4）rv.var()返回随机变量rv的方差值  
（5）rv.moment(n,\*args,\*kwds)返回随机变量的n阶距  
（6）rv.stats(moments='mvsk')返回随机变量rv的状态，moments的参数可为m(均值),v(方差值),s(偏度),k(峰度),默认为mv

#-\*- coding:utf-8 -\*-

from scipy.stats import binom

from log\_api import log

def nc\_of\_binom():

rv = binom(10,0.2)

log(rv.mean())

log(rv.var())

log(rv.moment(1))

log(rv.moment(2))

log(rv.moment(3))

log(rv.moment(4))

log(rv.stats(moments='mvsk'))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

nc\_of\_binom()

23. 下面为泊松分布的数字特征示例，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.poisson(mu)函数返回符合泊松分布的离散随机变量rv,单位时间内随机事件的平均发生率为mu  
（3）rv.mean()返回随机变量rv的均值  
（4）rv.var()返回随机变量rv的方差值  
（5）rv.moment(n,\*args,\*kwds)返回随机变量的n阶距  
（6）rv.stats(moments='mvsk')返回随机变量rv的状态，moments的参数可为m(均值),v(方差值),s(偏度),k(峰度),默认为mv

#-\*- coding:utf-8 -\*-

from scipy.stats import poisson

from log\_api import log

def nc\_of\_poisson():

rv = poisson(mu=5)

log(rv.mean())

log(rv.var())

log(rv.moment(1))

log(rv.moment(2))

log(rv.moment(3))

log(rv.moment(4))

log(rv.stats(moments='mvsk'))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

nc\_of\_poisson()

24. 下面为均匀分布的数字特征示例，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.Uniform(loc=0,scale=1)函数返回符合均匀分布的离散随机变量rv,rv处于区间[loc,loc+scale]  
（3）rv.mean()返回随机变量rv的均值  
（4）rv.var()返回随机变量rv的方差值  
（5）rv.moment(n,\*args,\*kwds)返回随机变量的n阶距  
（6）rv.stats(moments='mvsk')返回随机变量rv的状态，moments的参数可为m(均值),v(方差值),s(偏度),k(峰度),默认为mv

#-\*- coding:utf-8 -\*-

from scipy.stats import uniform

from log\_api import log

def nc\_of\_uniform():

rv = uniform(loc=2, scale=6)

log(rv.mean())

log(rv.var())

log(rv.moment(1))

log(rv.moment(2))

log(rv.moment(3))

log(rv.moment(4))

log(rv.stats(moments='mvsk'))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

nc\_of\_uniform()

25. 下面为指数分布的数字特征示例，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.expon(scale)函数返回符合指数分布的参数为scale的随机变量rv  
（3）rv.mean()返回随机变量rv的均值  
（4）rv.var()返回随机变量rv的方差值  
（5）rv.moment(n,\*args,\*kwds)返回随机变量的n阶距  
（6）rv.stats(moments='mvsk')返回随机变量rv的状态，moments的参数可为m(均值),v(方差值),s(偏度),k(峰度),默认为mv

#-\*- coding:utf-8 -\*-

from scipy.stats import expon

from log\_api import log

def nc\_of\_expon():

rv = expon(scale=2)

log(rv.mean())

log(rv.var())

log(rv.moment(1))

log(rv.moment(2))

log(rv.moment(3))

log(rv.moment(4))

log(rv.stats(moments='mvsk'))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

nc\_of\_expon()

26. 为正态分布的数字特征示例，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）lambda argument\_list:expression用来表示匿名函数  
（3）scipy.stats.norm.expect(func,loc=0,scale=1)函数返回func函数相对于正态分布的期望值，其中函数f(x)相对于分布dist的期望值定义为E[x]=Integral(f(x) \* dist.pdf(x))

#-\*- coding:utf-8 -\*-

from scipy.stats import norm

from log\_api import log

def nc\_of\_norm():

f1 = lambda x: x\*\*4

f2 = lambda x: x\*\*2-x+2

log(norm.expect(f1, loc=1, scale=2))

log(norm.expect(f2, loc=2, scale=5))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

nc\_of\_norm()

27. 下面为连续指数分布的数字特征示例，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）lambda argument\_list:expression用来表示匿名函数  
（3）numpy.exp(x)计算x的指数  
（4）numpy.inf表示正无穷大  
（5）scipy.integrate.quad(func,a,b)计算func函数从a至b的积分  
（6）scipy.stats.expon(scale)函数返回符合指数分布的参数为scale的随机变量rv  
（7）rv.moment(n,\*args,\*kwds)返回随机变量的n阶距

#-\*- coding:utf-8 -\*-

import numpy as np

from scipy import integrate

from scipy.stats import expon

from log\_api import log

def nc\_of\_expon():

#1st non-center moment of expon distribution whose lambda is 0.5

E1 = lambda x: x\*0.5\*np.exp(-x/2)

#2nd non-center moment of expon distribution whose lambda is 0.5

E2 = lambda x: x\*\*2\*0.5\*np.exp(-x/2)

log(integrate.quad(E1, 0, np.inf))

log(integrate.quad(E2, 0, np.inf))

log(expon(scale=2).moment(1))

log(expon(scale=2).moment(2))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

nc\_of\_expon()

28. 下面为大数定律一，伯努利大数定律的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）调整泊松分布的参数，看看结果会如何变化？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.bernoulli.rvs(p,loc=0,size=1)返回size个符合泊松分布的随机变量

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

import matplotlib.plot\_api as plt

from scipy.stats import bernoulli

def law\_of\_large\_numbers():

x = np.arange(1, 1001, 1)

r = bernoulli.rvs(0.3, size=1000)

y = []

rsum =0.0

for i in range(1000):

if r[i]==1:

rsum=rsum+1

y.append(rsum/(i+1))

plt.plot(x, y, color='red')

plt.savefig('law\_of\_large\_numbers.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

law\_of\_large\_numbers()

29. 下面为大数定律二，辛钦大数定律的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）试利用其他独立同分布的随机变量验证辛钦大数定律。

输入说明:

无

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.binom.rvs(n,p,size=1)函数返回size个符合二项分布的随机变量,其独立试验次数为n，每次成功的概率为  
（3）scipy.stats.poisson.rvs(mu,size=1)函数返回size个符合泊松分布的随机变量,其单位时间内随机事件的平均发生率为mu  
（4）scipy.stats.norm.rvs(loc=0,scale=1,size=1)函数返回size个符合正态分布的随机变量，其数学期望为loc，标准差为scale

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

import matplotlib.plot\_api as plt

from scipy.stats import binom

from scipy.stats import poisson

from scipy.stats import norm

def law\_of\_large\_numbers():

x = np.arange(1, 1001, 1)

r1 = binom.rvs(10, 0.6, size=1000)

r2 = poisson.rvs(mu=6, size=1000)

r3 = norm.rvs(loc=6, size=1000)

y = []

rsum=0.0

for i in range(1000):

rsum=rsum+(r1[i]+r2[i]+r3[i])

y.append(rsum/((i+1)\*3)-6)

plt.plot(x, y, color='red')

plt.savefig('law\_of\_large\_numbers.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

law\_of\_large\_numbers()

30. 下面为独立同分布的中心极限定理的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）试利用其他分布函数验证此中心极限定理。

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.expon.rvs(scale=1,size=1)函数返回size个符合指数分布的参数为scale的随机变量

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

from scipy.stats import expon

import matplotlib.plot\_api as plt

def central\_limit\_theorem():

y = []

n=100

for i in range(1000):

r = expon.rvs(scale=1, size=n)

rsum=np.sum(r)

z=(rsum-n)/np.sqrt(n)

y.append(z)

plt.hist(y,color='grey')

plt.savefig('central\_limit\_theorem.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

central\_limit\_theorem()

31. 下面为独立同分布的中心极限定理，拉普拉斯定理的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）试调整二项分布的参数，看看结果如何变化？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.binom.rvs(n,p,size=1)函数返回size个符合二项分布的随机变量,其独立试验次数为n，每次成功的概率为p

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

from scipy.stats import binom

import matplotlib.plot\_api as plt

def central\_limit\_theorem():

y = []

n=100

for i in range(1000):

r = binom.rvs(n, 0.3)

rsum=np.sum(r)

z=(rsum-n\*0.3)/np.sqrt(n\*0.3\*0.7)

y.append(z)

plt.hist(y,color='grey')

plt.savefig('central\_limit\_theorem.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

central\_limit\_theorem()

32. 下面为卡方分布的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）替换df为不同的自由度，看看结果如何变化？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）matplotlib.pyplot.subplot(nrows,ncols,plot\_number)生成nrows \* ncols 个subplot并返回plot\_number所指定plot  
（3）numpy.linspace(start,end,num=50)返回start到end之间num个等间距数字  
（4）scipy.stats.chi2.ppf(q,df)是ccf函数的反函数，计算自由度为df的卡方分布累积概率值为q时的随机变量值  
（5）scipy.stats.chi2.pdf(x,df)概率密度函数  
（6）scipy.stats.norm.rvs(size=1)返回size个符合标准正态分布的随机变量

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

from scipy.stats import norm

from scipy.stats import chi2

import matplotlib.plot\_api as plt

def chi2\_distribution():

fig, ax = plt.subplots(1, 1)

#display the probability density function

df = 10

x=np.linspace(chi2.ppf(0.01, df), chi2.ppf(0.99, df), 100)

ax.plot(x, chi2.pdf(x,df))

#simulate the chi2 distribution

y = []

n=10

for i in range(1000):

chi2r=0.0

r = norm.rvs(size=n)

for j in range(n):

chi2r=chi2r+r[j]\*\*2

y.append(chi2r)

ax.hist(y, normed=True, alpha=0.2)

plt.savefig('chi2\_distribution.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

chi2\_distribution()

33. 下面为t分布的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）替换df为不同的自由度，看看结果如何变化？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）matplotlib.pyplot.subplot(nrows,ncols,plot\_number)生成nrows \* ncols 个subplot并返回plot\_number所指定plot  
（3）numpy.linspace(start,end,num=50)返回start到end之间num个等间距数字  
（4）scipy.stats.t.pdf(x,df)概率密度函数  
（5）scipy.stats.norm.rvs()返回符合标准正态分布的随机变量  
（6）scipy.stats.chi2.rvs(df)返回自由度为df的卡方分布的随机变量

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

from scipy.stats import norm

from scipy.stats import chi2

from scipy.stats import t

import matplotlib.plot\_api as plt

def t\_distribution():

fig, ax = plt.subplots(1, 1)

#display the probability density function

df = 10

x=np.linspace(-4, 4, 100)

ax.plot(x, t.pdf(x,df))

#simulate the t-distribution

y = []

for i in range(1000):

rx = norm.rvs()

ry = chi2.rvs(df)

rt = rx/np.sqrt(ry/df)

y.append(rt)

ax.hist(y, normed=True, alpha=0.2)

plt.savefig('t\_distribution.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

t\_distribution()

34. 下面为F分布的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）替换dfn,dfd为不同的自由度，看看结果如何变化？

输入说明:

无

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）matplotlib.pyplot.subplot(nrows,ncols,plot\_number)生成nrows \* ncols 个subplot并返回plot\_number所指定plot  
（3）numpy.linspace(start,end,num=50)返回start到end之间num个等间距数字  
（4）scipy.stats.f.ppf(q,dfn,dfd)是ccf函数的反函数，计算自由度为dfn和dfd的F分布累积概率值为q时的随机变量值  
（5）scipy.stats.f.pdf(x,dfn,dfd)概率密度函数  
（6）scipy.stats.chi2.rvs(df)返回自由度为df的卡方分布的随机变量

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

from scipy.stats import norm

from scipy.stats import chi2

from scipy.stats import f

import matplotlib.plot\_api as plt

def F\_distribution():

fig, ax = plt.subplots(1, 1)

#display the probability density function

dfn, dfm = 10, 5

x = np.linspace(f.ppf(0.01, dfn, dfm), f.ppf(0.99, dfn, dfm), 100)

ax.plot(x, f.pdf(x, dfn, dfm))

#simulate the F-distribution

y = []

for i in range(1000):

rx = chi2.rvs(dfn)

ry = chi2.rvs(dfm)

rf = np.sqrt(rx/dfn)/np.sqrt(ry/dfm)

y.append(rf)

ax.hist(y, normed=True, alpha=0.2)

plt.savefig('F\_distribution.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

F\_distribution()

35. 下面为样本均值抽样分布-expon分布的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）使用其他分布替代expon分布，看看结果如何变化？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）matplotlib.pyplot.subplot(nrows,ncols,plot\_number)生成nrows \* ncols 个subplot并返回plot\_number所指定plot  
（3）numpy.linspace(start,end,num=50)返回start到end之间num个等间距数字  
（4）scipy.stats.norm.pdf(x)概率密度函数  
（5）scipy.stats.expon.rvs(scale=1,size=1)函数返回size个符合指数分布的参数为scale的随机变量

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

from scipy.stats import norm

from scipy.stats import expon

import matplotlib.plot\_api as plt

def sampling\_distribution():

fig, ax = plt.subplots(1, 1)

#display the probability density function

x = np.linspace(-4, 4, 100)

ax.plot(x, norm.pdf(x))

#simulate the sampling distribution

y = []

n=100

for i in range(1000):

r = expon.rvs(scale=1, size=n)

rsum=np.sum(r)

z=(rsum-n)/np.sqrt(n)

y.append(z)

ax.hist(y, normed=True, alpha=0.2)

plt.savefig('sampling\_distribution.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

sampling\_distribution()

36. 下面为样本均值抽样分布-norm分布的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）使用其他分布替代norm分布，看看结果如何变化？

输入说明:

无

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）matplotlib.pyplot.subplot(nrows,ncols,plot\_number)生成nrows \* ncols 个subplot并返回plot\_number所指定plot  
（3）numpy.linspace(start,end,num=50)返回start到end之间num个等间距数字  
（4）scipy.stats.norm.pdf(x)概率密度函数  
（5）scipy.stats.norm.rvs(loc=0,scale=1,size=1)函数返回size个符合正态分布的随机变量，其数学期望为loc，标准差为scale

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

from scipy.stats import norm

import matplotlib.plot\_api as plt

def sampling\_distribution():

fig, ax = plt.subplots(1, 1)

#display the probability density function

x = np.linspace(-4, 4, 100)

ax.plot(x, norm.pdf(x))

#simulate the sampling distribution

y = []

n=100

for i in range(1000):

r = norm.rvs(loc=5, scale=2, size=n)

rsum=np.sum(r)

z=(rsum-n\*5)/np.sqrt(n\*4)

y.append(z)

ax.hist(y, normed=True, alpha=0.2)

plt.savefig('sampling\_distribution.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

sampling\_distribution()

37. 下面为样本方差抽样分布的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）修改正态分布函数norm的参数，看看结果如何变化？

输入说明:

无

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）matplotlib.pyplot.subplot(nrows,ncols,plot\_number)生成nrows \* ncols 个subplot并返回plot\_number所指定plot  
（3）numpy.linspace(start,end,num=50)返回start到end之间num个等间距数字  
（4）scipy.stats.chi2.ppf(q,df)是ccf函数的反函数，计算自由度为df的卡方分布累积概率值为q时的随机变量值  
（5）scipy.stats.chi2.pdf(x,df)概率密度函数  
（6）scipy.stats.norm.rvs(loc=0,scale=1,size=1)函数返回size个符合正态分布的随机变量，其数学期望为loc，标准差为scale

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

from scipy.stats import chi2

from scipy.stats import norm

import matplotlib.plot\_api as plt

def sampling\_distribution():

fig, ax = plt.subplots(1, 1)

#display the probability density function

df = 10

x=np.linspace(chi2.ppf(0.01, df), chi2.ppf(0.99, df), 100)

ax.plot(x, chi2.pdf(x, df))

#simulate the sampling distribution

y = []

for i in range(1000):

r = norm.rvs(loc=5, scale=2, size=df+1)

rchi2 =(df)\*np.var(r)/4

y.append(rchi2)

ax.hist(y, normed=True, alpha=0.2)

plt.savefig('sampling\_distribution.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

sampling\_distribution()

38. 下面为抽样分布-t分布的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）修改正态分布函数norm的参数，看看结果如何变化？  
（3）修改便利次数<1000，看看结果如何变化？>1000呢？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）matplotlib.pyplot.subplot(nrows,ncols,plot\_number)生成nrows \* ncols 个subplot并返回plot\_number所指定plot  
（3）numpy.linspace(start,end,num=50)返回start到end之间num个等间距数字  
（4）scipy.stats.t.ppf(q,df)是ccf函数的反函数，计算自由度为df的t分布累积概率值为q时的随机变量值  
（5）scipy.stats.t.pdf(x,df)概率密度函数  
（6）scipy.stats.norm.rvs(loc=0,scale=1,size=1)函数返回size个符合正态分布的随机变量，其数学期望为loc，标准差为scale

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

from scipy.stats import t

from scipy.stats import norm

import matplotlib.plot\_api as plt

def sampling\_distribution():

fig, ax = plt.subplots(1, 1)

#display the probability density function

df = 10

x=np.linspace(t.ppf(0.01, df), t.ppf(0.99, df), 100)

ax.plot(x, t.pdf(x, df))

#simulate the sampling distribution

y = []

for i in range(1000):

r = norm.rvs(loc=5, scale=2, size=df+1)

rt =(np.mean(r)-5)/np.sqrt(np.var(r)/df)

y.append(rt)

ax.hist(y, normed=True, alpha=0.2)

plt.savefig('sampling\_distribution.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

sampling\_distribution()

39. 下面为抽样分布-F分布的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）修改正态分布函数norm的参数，看看结果如何变化？  
（3）修改便利次数<1000，看看结果如何变化？>1000呢？

输入说明:

无

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）matplotlib.pyplot.subplot(nrows,ncols,plot\_number)生成nrows \* ncols 个subplot并返回plot\_number所指定plot  
（3）numpy.linspace(start,end,num=50)返回start到end之间num个等间距数字  
（4）scipy.stats.f.ppf(q,dfn,dfd)是ccf函数的反函数，计算自由度为dfn和dfd的F分布累积概率值为q时的随机变量值  
（5）scipy.stats.f.pdf(x,dfn,dfd)概率密度函数  
（6）scipy.stats.norm.rvs(loc=0,scale=1,size=1)函数返回size个符合正态分布的随机变量，其数学期望为loc，标准差为scale

#-\*- coding:utf-8 -\*-

import sys

import numpy as np

from scipy.stats import f

from scipy.stats import norm

import matplotlib.plot\_api as plt

def sampling\_distribution():

fig, ax = plt.subplots(1, 1)

#display the probability density function

dfn, dfm = 10, 5

x=np.linspace(f.ppf(0.01, dfn, dfm), f.ppf(0.99, dfn, dfm), 100)

ax.plot(x, f.pdf(x, dfn, dfm))

#simulate the sampling distribution

y = []

for i in range(1000):

r1 = norm.rvs(loc=5, scale=2, size=dfn+1)

r2 = norm.rvs(loc=3, scale=2, size=dfm+1)

rf =np.var(r1)/np.var(r2)

y.append(rf)

ax.hist(y, normed=True, alpha=0.2)

plt.savefig('sampling\_distribution.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

sampling\_distribution()

其中40-48无答案

## 40. Working backwards[我的提交](http://121.41.106.89:8081/problem/41/submissions/)

A 95% confidence interval for a population mean, u, is given as (18.985, 21.015). This confidence interval is based on a simple random samples of 36 observations. Calculate the sample mean and standard deviation. Assume that all conditions necessary for inference are satisfied. Use the t-distribution in any calculations.

#### 输入说明:

none 

#### 输出说明:

[mean, standard deviation], accurate to the second decimal place

#### 样例输入:

none

#### 样例输出:

none

#### 提示:

## 41. Sleep habits of New Yorkers[我的提交](http://121.41.106.89:8081/problem/42/submissions/)

New York is known as "the city that never sleeps". A random sample of 25 New Yorkers were asked how much sleep they get per night. Statistical summaries of these data are shown below. Do these data provide strong evidence that New Yorkers sleep less than 8 hours per night on average? Null-hypothesis is H0: u=8, and alpha is 0.05

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | mean | stand-variance | min | max |
| 25 | 7.73 | 0.77 | 6.17 | 9.78 |

Extra: (1) If you were to construct a 90% confidence interval that corresponded to this hypothesis test, would you expect 8 hours to be in the interval? Explain your reasoning.

#### 输入说明:

none 

#### 输出说明:

[degree-of-freedom-of-distribution, statistical values, conclusion],'degree-of-freedom-of-distribution' and 'statistical values' are accurate to the second decimal place, 'conclusion' is True, which means the H0 is accepted, or False

#### 样例输入:

none

#### 样例输出:

[2, 1.63, False]

#### 提示:

1) t-distribution

## 42. Play the piano[我的提交](http://121.41.106.89:8081/problem/43/submissions/)

Georgianna claims that in a small city renowned for its music school, the average child takes at least 5 years of piano lessons. We have a random sample of 20 children from the city with a mean of 4.6 years of piano lessons and a standard deviation of 2.2 years. Evaluate Georgianna's claims using a hypothesis test. alpha is 0.05.   
Extra:  
(1) Construct a 95% confidence interval for the number of years students in this city takes piano lessons and interpret it in context of the data.  
(2) Do your results from the hypothesis test and the confidence interval agree? Explain your reasoning.

#### 输入说明:

none 

#### 输出说明:

[degree-of-freedom-of-distribution, statistical values, conclusion],'degree-of-freedom-of-distribution' and 'statistical values' are accurate to the second decimal place, 'conclusion' is True, which means the H0 is accepted, or False

#### 样例输入:

none

#### 样例输出:

[2, 1.63, False]

#### 提示:

## 43. Global Warming[我的提交](http://121.41.106.89:8081/problem/44/submissions/)

Is there strong evidence of global warming? Let's consider a small scale example, comparing how temperatures have changed is the US from 1968 to 2008. The daily high temperature reading on January 1 was collected in 1968 and 2008 for 51 randomly selected locations in the continental US. Then the difference between the two readings (temperature in 2008 - temperature in 1968) was calculated for each of the 51 values was 1.1 degree with a standard deviation of 4.9 degrees. We are interested in determining whether these data provide strong evidence of temperature warming in the continental US.  
(1) Is there a relationship between the observations collected in 1968 and 2008? Or are the observations in the two groups independent? Explain  
(2) What's the hypothesis for this research?  
(3) Check the conditions required to complete this test.  
(4) Calculate the freedom, test statistical value and give the conclusion. alpha is 0.05, **coding this part**  
(5) What type of error might we have made?  
(6) Based on the results of this hypothesis test, would you expect a confidence interval for the average difference between the temperature measurements from 1968 and 2008 to include 0? Explain

#### 输入说明:

none 

#### 输出说明:

[degree-of-freedom-of-distribution, statistical values, conclusion],'degree-of-freedom-of-distribution' and 'statistical values' are accurate to the second decimal place, 'conclusion' is True, which means the H0 is accepted, or False

#### 样例输入:

none

#### 样例输出:

[2, 1.63, False]

#### 提示:

## 44. Gaming and distracted eating[我的提交](http://121.41.106.89:8081/problem/45/submissions/)

A group of researchers are interested in the possible effects of distracting stimuli during eating, such as an increase or decrease in the amount of food consumption. To test this hypothesis, they monitored food intake for a group of 44 patients who were randomised into two equal groups. The treatment group ate lunch while playing solitaire, and the control group ate lunch without any added distractions. Patients in the treatment group ate 52.1 grams of biscuits, with a standard deviation of 45.1 grams, and patients in the control group ate 27.1 grams of biscuits with a standard deviation of 26.4 grams. Do these data provide convincing evidence that the average food intake is different for the patients in the treatment group? Assume the conditions for inference are satisfied.  
Null hypothesis is H0: u\_t - u\_c = 0, alpha is 0.05

#### 输入说明:

none 

#### 输出说明:

[degree-of-freedom-of-distribution, statistical values, conclusion],'degree-of-freedom-of-distribution' and 'statistical values' are accurate to the second decimal place, 'conclusion' is True, which means the H0 is accepted, or False

#### 样例输入:

none

#### 样例输出:

[2, 1.63, False]

#### 提示:

## 45. Increasing corn yield[我的提交](http://121.41.106.89:8081/problem/46/submissions/)

A large farm wants to try out a new type of fertilizer to evaluate whether it will improve the farm's corn production. The land is broken into plots that produce an average of 1.215 pounds of corn with a standard deviation of 94 pounds per plot. The owner is interested in detecting any average difference of at least 40 pounds per plot. How many plots of land would be needed for the experiment if the desired power level is 90%? Assume each plot of land gets treated with either the current fertilizer or the new fertilizer.

#### 输入说明:

none

#### 输出说明:

plot\_num, int type

#### 样例输入:

none

#### 样例输出:

20

#### 提示:

## 46. Rock-paper-scissors[我的提交](http://121.41.106.89:8081/problem/47/submissions/)

Rock-paper-scissors is a hand game played by two or more people where players choose to sign either rock, paper or scissors with their hands. For your statistics class project, you want to evaluate whether players choose between these three options randomly, or if certain options are favoured above others. You ask two friends to play rock-paper-scissors and count the time each option is played. The following table summaries the data:

|  |  |  |
| --- | --- | --- |
| Rock | Paper | Scissors |
| 43 | 21 | 35 |

Use these data to evaluate whether players choose between these three options randomly, or if certain options are favored above others. alpha is 0.05.

#### 输入说明:

none 

#### 输出说明:

[degree-of-freedom-of-distribution, statistical values, conclusion],'degree-of-freedom-of-distribution' and 'statistical values' are accurate to the second decimal place, 'conclusion' is True, which means the H0 is accepted, or False

#### 样例输入:

none

#### 样例输出:

[2, 1.63, True]

#### 提示:

## 47. Quitters[我的提交](http://121.41.106.89:8081/problem/48/submissions/)

Does being part of a support group affect the ability of people to quit smoking? A country health department enrolled 300 smokers in a randomized experiment. 150 participants were assigned to a group that used a nicotine patch and met weekly with a support group; the other 150 received the patch and did not meet with a support group. At the end of the study, 40 of the participants in the patch plus support group had quit smoking while only 30 smokers had quit in the other group.  
(1) Create a two-way table presenting the results of this study.  
(2) Answer each of the following questions under the null hypothesis that being part of a support group does not affect the ability of people to quit smoking, and indicate whether the expected values are higher or lower than the observed values.  
     \* How many subjects in the "patch+support" group would you expect to quit? **coding this part**  
     \* How many subjects in the "patch only" group would you expect to not quit?

#### 输入说明:

none 

#### 输出说明:

subjects\_num, int type

#### 样例输入:

none

#### 样例输出:

20

#### 提示:

## 48. Offshore drilling[我的提交](http://121.41.106.89:8081/problem/49/submissions/)

The table below summaries a data set that examines the response of a random sample of college graduates and non-graduate on the topic of oil drilling. Complete a chi-square test for test data to check whether there is a statistically significant difference in responses from college graduates and non-graduates.

|  |  |  |  |
| --- | --- | --- | --- |
| College Grad? | Yes | No | Total |
| Support | 154 | 132 | 286 |
| Oppose | 180 | 126 | 306 |
| Do not know | 104 | 131 | 235 |
| Total | 438 | 389 | 827 |

#### 输入说明:

none 

#### 输出说明:

[degree-of-freedom-of-distribution, statistical values, conclusion],'degree-of-freedom-of-distribution' and 'statistical values' are accurate to the second decimal place, 'conclusion' is True, which means the H0 is accepted, or False

#### 样例输入:

none

#### 样例输出:

[2, 1.63, True]

#### 提示:

49. 铁路公司将它所有的火车头都进行了编号，从1到N。有一天你看见一个编号为60的火车头，那该铁路公司共有多少火车头呢？试用极大似然估计、最小偏方差估计、无偏估计分别估算火车头数目x，并重复10000次试验，计算命中率、均偏方差(x-N)^2、均偏差(x-N)并进行比较，体会各种估计方法的优缺点。  
下面为火车头问题的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）假如你看到了两辆火车，编号分别为46、24，那么如何进行参数估计呢？（提示：多次估算计算平均值or修改估算公式？）  
（3）假如你看到了多辆（多于两辆）火车呢？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）round(number[,ngigits])返回浮点数number的四舍五入值

#-\*- coding:utf-8 -\*-

import random

from log\_api import log

def number\_trans(upper\_bound):

return random.randint(1,upper\_bound)

def train\_seen(N):

return random.randint(1,N)

def MLE\_estimation(evidence):

return evidence

def MSE\_estimation(evidence):

return round(1.5\*evidence)

def ME\_estimation(evidence):

return 2\*evidence

def estimate():

number\_experiments = 10000

upper\_bound = 100

H1 = H2 = H3 = MSE1 = MSE2 = MSE3 = ME1 = ME2 = ME3 = 0.0

for j in range(number\_experiments):

N = number\_trans(upper\_bound)

evidence = train\_seen(N)

hypo1 = MLE\_estimation(evidence)

hypo2 = MSE\_estimation(evidence)

hypo3 = ME\_estimation(evidence)

#calculating hits

H1 = H1 + 1 if hypo1==N else H1

H2 = H2 + 1 if hypo2==N else H2

H3 = H3 + 1 if hypo3==N else H3

#calculating mean squared error

MSE1 = MSE1 + (hypo1-N)\*\*2

MSE2 = MSE2 + (hypo2-N)\*\*2

MSE3 = MSE3 + (hypo3-N)\*\*2

#calculating mean error

ME1 = ME1 + (hypo1-N)

ME2 = ME2 + (hypo2-N)

ME3 = ME3 + (hypo3-N)

log(H1/number\_experiments)

log(H2/number\_experiments)

log(H3/number\_experiments)

log(MSE1/number\_experiments)

log(MSE2/number\_experiments)

log(MSE3/number\_experiments)

log(ME1/number\_experiments)

log(ME2/number\_experiments)

log(ME3/number\_experiments)

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

estimate()

50. 单总体t检验通常用于检验一个样本平均数与一个已知的总体平均数的差异是否显著。下面为t检验问题的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.ttest\_1sample(a,popmean)对给定样本a与总体平均值进行t检验

#-\*- coding:utf-8 -\*-

from scipy import stats

from log\_api import log

def ttest():

x = stats.norm.rvs(loc=5, scale=10, size=50)

log(stats.ttest\_1samp(x,5.0))

log(stats.ttest\_1samp(x,1.0))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

ttest()

51. 独立双总体t检验通常用于检验两个相互独立的样本平均数与其各自所代表的总体平均数的差异是否显著。下面为t检验问题的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.ttest\_ind(a,b,equal\_var=True)对给定样本a与样本b进行t检验，其中equal\_var为True时表明两样本所代表总体需要相同方差

#-\*- coding:utf-8 -\*-

from scipy import stats

from log\_api import log

def ttest():

x = stats.norm.rvs(loc=5, scale=1, size=50)

y = stats.norm.rvs(loc=2, scale=10, size=50)

log(stats.ttest\_ind(x, y))

log(stats.ttest\_ind(x, y, equal\_var = False))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

ttest()

52. 卡方拟合优度检验通常用于根据样本的频数分布来推断总体的分布。下面为卡方拟合优度检验问题的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.chisquare(f\_obs,f\_exp=None)对给定样本f-obs及其期望分布f\_exp进行检验，其中f\_exp为None时认为期望分布为均匀分布，其值为样本平均值

#-\*- coding:utf-8 -\*-

from scipy import stats

from log\_api import log

def chisquare():

A=[16, 18, 16, 14, 12, 12]

B=[16, 16, 16, 16, 16, 8]

log(stats.chisquare(A))

log(stats.chisquare(A, f\_exp=B))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

chisquare()

53. Wilcoxon秩和检验通常用于检验成对观测数据之差是否来自均值为0的总体。下面为Wilcoxon秩和检验问题的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.chisquare(x,y)对给定样本x及y进行秩和检验

#-\*- coding:utf-8 -\*-

from scipy import stats

from log\_api import log

def chisquare():

A=[16, 18, 16, 14, 12, 12]

B=[16, 16, 16, 16, 16, 8]

log(stats.chisquare(A))

log(stats.chisquare(A, f\_exp=B))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

chisquare()

54. KS检验通常用于检验单一样本是否服从某特定分布，或者两样本是否来自相同分布。下面为KS检验问题的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.ks\_2samp(data1,data2)对给定样本data1及data2进行KS双边检验

#-\*- coding:utf-8 -\*-

from scipy import stats

from log\_api import log

def kstest():

n1=200

n2=300

a = stats.norm.rvs(size=n1, loc=0, scale=1)

b = stats.norm.rvs(size=n2, loc=0.5, scale=1.5)

c = stats.norm.rvs(size=n2, loc=0.01, scale=1)

log(stats.ks\_2samp(a, b))

log(stats.ks\_2samp(a, c))

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

kstest()

55. 单因素方差分析用于推断各样本所代表的各总体均数是否相等。下面为单因素方差分析问题的python模拟实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）仿照oneway\_anova函数设计自己的双因素方差分析函数twoway\_anova

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题

#-\*- coding:utf-8 -\*-

from scipy import stats

import numpy as np

from log\_api import log

def oneway\_anova():

A1=[27.0, 26.2, 28.8, 33.5, 28.8]

A2=[22.8, 23.1, 27.7, 27.6, 24.0]

A3=[21.9, 23.4, 20.1, 27.8, 19.3]

A4=[23.5, 19.6, 23.7, 20.8, 23.9]

A=[A1, A2, A3, A4]

n=20

As=np.sum(A, axis=1)

QA=0.0

for i in range(4):

QA=QA+As[i]\*As[i]

QA=QA/5

QT=0.0

for i in range(4):

for j in range(5):

QT=QT+A[i][j]\*A[i][j]

C=np.sum(A)\*np.sum(A)/n

ST=QT-C

SA=QA-C

Se=ST-SA

F=(SA/3)/(Se/16)

log('QA is ' + str(QA))

log('QT is ' + str(QT))

log('C is ' + str(C))

log('ST is ' + str(ST))

log('SA is ' + str(SA))

log('Se is ' + str(Se))

log('F is '+ str(F))

def twoway\_anova():

pass

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

oneway\_anova()

twoway\_anova()

56. 皮尔森相关系数用于度量两个变量之间相关程度，介于-1到1之间，其中-1表示完全负相关，0表示无关，1表示完全正相关。下面为皮尔森相关系数的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）试试自己构造其他变量组X、Y，并计算它们之间的皮尔森相关系数？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.pearsonr(x,y)计算x,y之间的相关系数，此外，并返回p-value值

#-\*- coding:utf-8 -\*-

import matplotlib.plot\_api as plt

import numpy as np

from scipy import stats

from log\_api import log

def pearsonr():

x = np.linspace(-5, 5, num=150)

y = x + np.random.normal(size=x.size)

y[12:14] += 10

log(stats.pearsonr(x, y))

plt.scatter(x,y)

plt.savefig('pearsonr.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

pearsonr()

57. 回归分析是确定两种或两种以上变量之间相互依赖的定量关系的方法，分为一元回归分析和多元回归分析、线性回归分析和非线性回归分析等。下面为回归分析的python实现，请完成以下练习：  
（1）仔细阅读，理解代码含义，并运行代码，看看结果如何？  
（2）假如X、Y之间满足y=a\*x\*\*2+b的关系，怎么随机生成X、Y变量并画出其回归曲线呢？  
（3）假如X、Y之间满足y=a\*e\*\*x+b的关系，怎么随机生成X、Y变量并画出其回归曲线呢？

输入说明:

无 

输出说明:

无

样例输入:

无

样例输出:

无

提示:

（1）此题为普通编程练习题  
（2）scipy.stats.Linregress(x,y=None)计算两组测量值X，Y之间的回归曲线，并返回曲线斜率slope、曲线截距intercept、相关系数r-value，检验值p-value，估计标准误差stderr

#-\*- coding:utf-8 -\*-

import matplotlib.plot\_api as plt

import numpy as np

from scipy import stats

from log\_api import log

def linregress1():

x = np.linspace(-5, 5, num=150)

y = x + np.random.normal(size=x.size)

y[12:14] += 10

slope, intercept, r\_value, p\_value, std\_err = stats.linregress(x,y)

log(slope)

log(intercept)

log(r\_value)

log(p\_value)

log(std\_err)

plt.plot(x, y, 'b.')

plt.plot(x, slope \* x + intercept, 'r-')

plt.savefig('linregress1.png')

def linregress2():

x = np.linspace(-5, 5, num=150)

y = x\*\*2 + np.random.normal(size=x.size)

y[12:14] += 10

y[137:141] -= 6

x1=x\*\*2

slope, intercept, r\_value, p\_value, std\_err = stats.linregress(x1,y)

log(slope)

log(intercept)

log(r\_value)

log(p\_value)

log(std\_err)

plt.plot(x, y, 'b.')

plt.plot(x, slope \* x1 + intercept, 'r-')

plt.savefig('linregress2.png')

#the code should not be changed

if \_\_name\_\_ == '\_\_main\_\_':

linregress1()

linregress2()

58. 创建一个ndarray数组对象，输入一组100以内的整数，数组shape为4\*4，将数组中所有元素的总和、每行的平均值以及每列的平均值共9个数存入一个列表中，并返回该列表。

输入说明:

用np.array()函数生成的一个4\*4数组对象，确定一组值 

输出说明:

计算后的9个数的列表

样例输入:

1～16，共16个数

样例输出:

[136,2.5,6.5,10.5,14.5,7,8,9,10]

提示:

无

#-\*- coding:utf-8 -\*-

from log\_api import log

class Solution():

def solve(self, A):

pass

## 59. 寻找回文串[我的提交](http://121.41.106.89:8081/problem/67/submissions/)

对于一个包含一系列数字字符串的列表，寻找其中的回文串存入一个列表中并返回

#### 输入说明:

一个包含一系列数字字符串的列表 

#### 输出说明:

仅包含回文串的列表

#### 样例输入:

[‘123’, ‘232’, ‘4556554’, ‘12123’, ‘3443’,'1314131']

#### 样例输出:

['232', '4556554', '3443','1314131']

#### 提示:

无

## 60. 素数判断[我的提交](http://121.41.106.89:8081/problem/68/submissions/)

完成函数solve，判断传入的整数列表A中的数字是否是素数，并将所有的素数保存到另一个列表中并返回。

#### 输入说明:

A: 一个含素数和非素数的多个整数列表 

#### 输出说明:

素数组成的列表

#### 样例输入:

[23, 45, 76, 67, 17]

#### 样例输出:

[23,67,17]

#### 提示:

特殊整数的素数判断

61. 在Numpy中，多项式函数的系数可以用一维数组表示，例如对于f(x)=2x^3-x+1可表示为f=np.array([2.0,0.0,-1.0,1.0])，而np.poly1d()方法可以将多项式转换为poly1d(一元多项式)对象，返回多项式函数的值，请利用poly1d()方法计算多项式g(x)(例如g(x)=x^2+2x+1)和f(x)的乘积并将结果返回。

输入说明:

多项式g(x)的一维数组表示 

输出说明:

f(x)\*g(x)的poly1d对象

样例输入:

np.array([1, 2, 1])

样例输出:

np.poly1d([2.0,4.0,1.0,-1.0,1.0,1.0])

提示:

样式输出结果表示两个多项式f(x)\*g(x)=2x^5+4x^4+x^3-x^2+x+1

## 62. 数字字符统计[我的提交](http://121.41.106.89:8081/problem/70/submissions/)

已知有一个由数字字符串构成的列表，统计列表中数字字符'0'-'9'各自出现的次数并返回统计结果

#### 输入说明:

一个由数字字符串构成的列表 

#### 输出说明:

各数字字符的出现次数

#### 样例输入:

['12','34','567', '36','809','120']

#### 样例输出:

{0:2, 1:2, 2:2, 3:2, 4:1, 5:1, 6:2, 7:1, 8:1, 9:1}

#### 提示:

返回结果为一个字典

63. 已知列表fruits中顺序保存了某商店每日出售的水果品名，例如fruits=['apple','banana','cherry','pineapple','banana','peach','pear','peach','cherry' ]，完成函数solve()计算每一种水果的出售次数，存入字典result中并将结果返回

输入说明:

一个纪录了水果出售纪录的列表 

输出说明:

水果出售次数统计结果

样例输入:

['apple', 'banana', 'cherry', 'pineapple', 'banana', 'peach', 'pear','peach', 'cherry' ]

样例输出:

{'pear':1, 'banana':2, 'cherry':2, 'peach':2, 'pineapple':1, 'apple':1}

提示:

返回结果为一个字典