

上海大学

公共基础课 数据分析与智能计算 上机报告

作业：第二周

姓名：林艺璿

学号：18120189

学院：计算机工程与科学学院

日期：2020 年 4 月 14 日

第二章 多维数据结构与运算

思考与练习 2.1

1. 一维数组访问。

(1) 在 subjects 数组中选择并显示序号 1、2、4 门课的名称，使用负值序号选择并显示 names 数组中‘方绮雯’。

(2) 选择并显示 names 数组从 2 到最后的数组元素；选择并显示 subjects 数组 2 到 4 的数组元素。

(3) 使用布尔条件选择并显示 subjects 数组中的英语和物理科目名称。

```
import numpy as np
names = np.array(['王微', '肖良英', '方绮雯', '刘旭阳', '钱易铭'])
subjects = np.array(['Math', 'English', 'Python', 'Chinese', 'Art', 'Database', 'Physics'])
print("subjects数组中选择并显示序号1、2、4门课的名称")
print(subjects[ [1, 2, 4] ])
print("使用倒序索引选择并显示names数组中'方绮雯'")
print(names[-3])
print("选择并显示names数组从2到最后的数组元素")
print(names[ 2: : ])
print("选择并显示subjects数组正序2~4的数组元素")
print(subjects[ 2:5 ])
print("使用布尔条件选择并显示subjects数组中的英语和物理科目名称")
print(subjects[ (subjects == 'English') | (subjects == 'Physics')])
```



图 1: 思考 1-1 实验结果

2. 二维数组访问。

(1) 选择并显示 scores 数组的 1、4 行。

(2) 选择并显示 scores 数组中行号 2、4 同学的数学和 Python 成绩。

(3) 选择并显示 scores 数组中所有同学的数学和艺术课程成绩。

(4) 选择并显示 scores 数组中“王微”和“刘旭阳”的英语和艺术课程成绩。

```

import numpy as np
names = np.array(['王微', '肖良英', '方绮雯', '刘旭阳', '钱易铭'])
subjects = np.array(['Math', 'English', 'Python', 'Chinese', 'Art', 'Database', 'Physics'])
scores = np.array([[70, 85, 77, 90, 82, 84, 89],
                   [60, 64, 80, 75, 80, 92, 90],
                   [90, 93, 88, 87, 86, 90, 91],
                   [80, 82, 91, 88, 83, 86, 80],
                   [88, 72, 78, 90, 91, 73, 80]])
print("选择并显示scores数组的1、4行")
print(scores[[1,4]])
print("选择并显示scores数组中行序2、4学生的数学和Python成绩")
print(scores[[2,4]][ :, (subjects == 'Math') | (subjects == 'Python')])
print("选择并显示scores数组中所有学生的数学和艺术成绩")
print(scores[ :, (subjects == 'Math') | (subjects == 'Art')])
print("选择并显示scores数组中'王微'和'刘旭阳'的英语和艺术课程成绩")
print(scores[ (names == '王微') | (names == '刘旭阳') ][ :, (subjects == 'English') | (subjects == 'Art')])

```

```

Run: 1+2 x
/Users/reneelin/opt/anaconda3/python.app/Contents/MacOS/p
选择并显示scores数组的1、4行
[[60 64 80 75 80 92 90]
 [88 72 78 90 91 73 80]]
选择并显示scores数组中行序2、4学生的数学和Python成绩
[[90 88]
 [88 78]]
选择并显示scores数组中所有学生的数学和艺术成绩
[[70 82]
 [60 80]
 [90 86]
 [80 83]
 [88 91]]
选择并显示scores数组中'王微'和'刘旭阳'的英语和艺术课程成绩
[[85 82]
 [82 83]]

```

图 2: 思考 1-2 实验结果

3. 生成由整数 10 19 组成的 2×5 的二维数组。

```

import numpy as np
print("生成由正数10~19组成的2×5的二维数组")
print(np.arange(10,20).reshape(2,5))

```

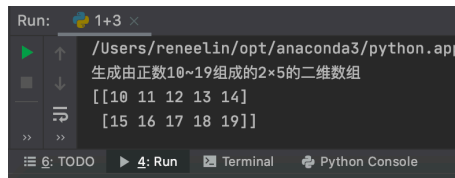


图 3: 思考 1-3 实验结果

思考与练习 2.2

1. 将 `scores` 数组中所有同学的英语成绩减去 3 分并显示。
2. 统计 `scores` 数组中每名同学所有科目的平均分并显示。
3. 使用随机函数生成 $[-1,1]$ 之间服从均匀分布的 3×4 二维数组，并计算所有元素的和。

```
import numpy as np
names = np.array(['王微', '肖良英', '方绮雯', '刘旭阳', '钱易铭'])
subjects = np.array(['Math', 'English', 'Python', 'Chinese', 'Art', 'Database', 'Physics'])
scores = np.array([[70, 85, 77, 90, 82, 84, 89],
                   [60, 64, 80, 75, 80, 92, 90],
                   [90, 93, 88, 87, 86, 90, 91],
                   [80, 82, 91, 88, 83, 86, 80],
                   [88, 72, 78, 90, 91, 73, 80]])

print("将scores数组中所有学生的英语成绩减去3分并显示")
print(scores[:, (subjects == 'English')] - 3)
print("统计scores数组中每名学生所有科目的平均分并显示")
print(scores.mean(axis = 1))
print("使用随机函数生成[-1,1]之间服从均匀分布的3*4二维数组，并计算所有元素的和")
rand = np.random.uniform(-1, 1, size = (3,4))
print(rand)
print("所有元素的和：", rand.sum())
```

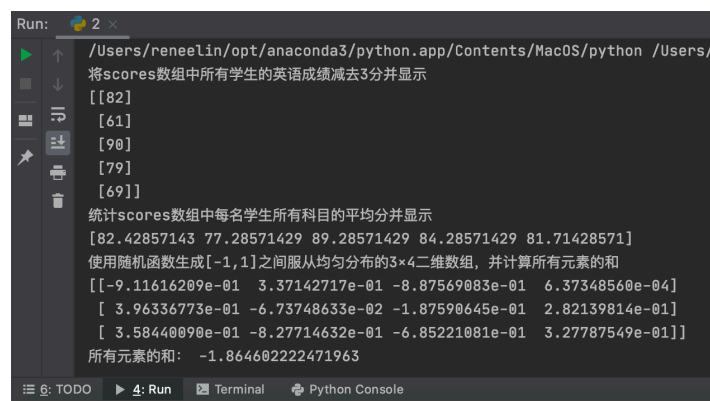


图 4: 思考 2 实验结果

思考与练习 2.3

1. 将随机游走的步数增加到 100 步，计算物体距离原点的距离。

```
import numpy as np
import matplotlib.pyplot as plt
# 模拟每步游走方向
# 将随机游走的步数增加到100步
steps = 100
rndwlk = np.random.randint(0, 2, size = (2, steps))
print(rndwlk)
rndwlk = np.where( rndwlk > 0, 1, -1)
print(rndwlk)
# 计算每部游走后的位置
position = rndwlk.cumsum(axis = 1)
print(position)
# 计算每步游走后至原点的距离
dists = np.sqrt(position[0]**2 + position[1]**2)
print(dists)
np.set_printoptions(precision = 4)
print(dists)
# 计算物体最终到原点的距离
finaldists = np.sqrt(position[0,-1]**2 + position[1,-1]**2)
print("物体最终到原点的距离：", finaldists)
# 绘图展示游走轨迹
x = position[0]
y = position[1]
plt.plot(x, y, c='g', marker='x')
plt.scatter(0, 0, c='r', marker='o')
plt.text(.1, -.1, 'origin')
plt.scatter(x[-1], y[-1], c='r', marker='o')
plt.text(x[-1]+.1, y[-1]-.1, 'stop')
plt.show()
```

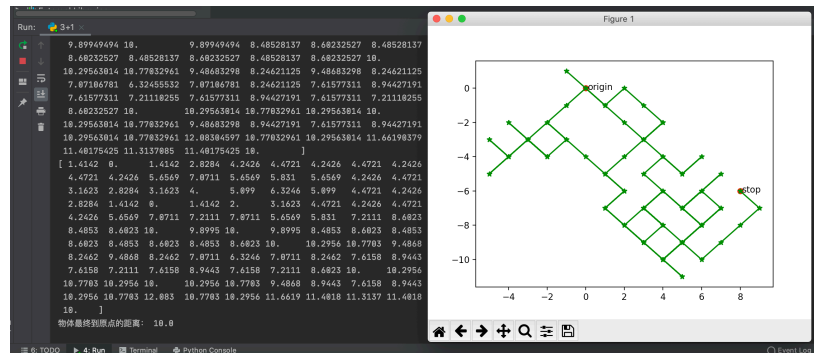


图 5: 思考 3-1 实验结果

2. 重复多次随机游走过程，物体距离原点距离的变化趋势是什么？

随机游走的步数越多，回到原点的概率越小。

```
import numpy as np
import matplotlib.pyplot as plt
# 模拟每步游走方向
# 将随机游走的步数增加到100步
steps = 100
for i in range(0,50):
    rndwlk = np.random.randint(0, 2, size = (2, steps))
    rndwlk = np.where( rndwlk > 0, 1, -1)
    # 计算每部游走后的位置
    position = rndwlk.cumsum(axis = 1)
    # 计算每步游走后至原点的距离
    dists = np.sqrt(position[0]**2 + position[1]**2)
    np.set_printoptions(precision = 4)
    print("物体到原点的最大距离: ", dists.max())
    print("物体到原点的最小距离: ", dists.min())
    # 计算物体最终到原点的距离
    finaldists = np.sqrt(position[0,-1]**2 + position[1,-1]**2)
    print("物体最终到原点的距离: ", finaldists)
    # 物体到原点距离的变化趋势
    x = np.arange(1,steps+1)
    y = dists
    plt.ylim(0, 50)
    plt.plot(x, y, c='g', marker='*')
    plt.scatter(0, 0, c='r', marker='o')
    plt.text(.1, -.1, 'origin')
    plt.show()
```

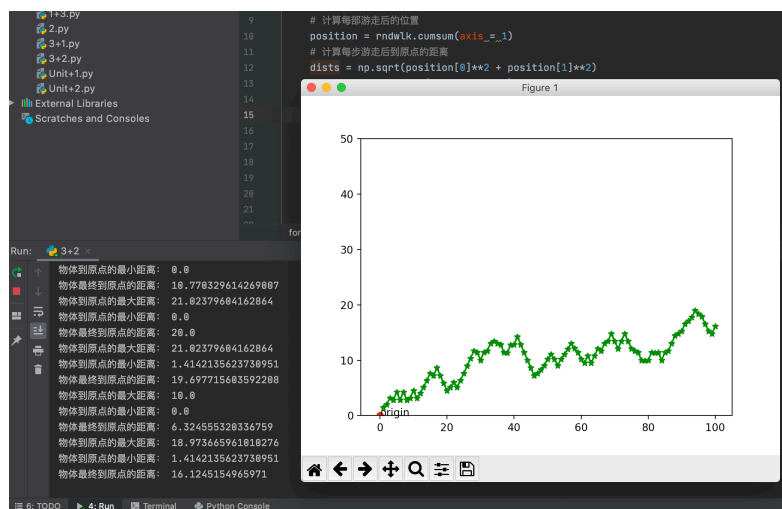


图 6: 思考 3-2 实验结果

综合练习题

1. “大润发”、“沃尔玛”、“好德”和“农工商”四个超市都卖苹果、香蕉、橘子和芒果四种水果。使用 NumPy 的 ndarray 实现以下功能。

- (1) 创建 2 个一维数组分别存储超市名称和水果名称；
- (2) 创建 1 个 4×4 的二维数组存储不同超市的水果价格，其中价格由 4 到 10 范围内的随机数生成；
- (3) 选择“大润发”的苹果和“好德”的香蕉，并将价格增加 1 元；
- (4) “农工商”水果大减价，所有水果价格减少 2 元；
- (5) 统计四个超市苹果和芒果的销售均价；
- (6) 找出橘子价格最贵的超市名称（不是编号）。

```
import numpy as np
supermarkets = np.array(['大润发', '沃尔玛', '好德', '农工商'])
fruits = np.array(['苹果', '香蕉', '橘子', '芒果'])
prices = np.random.randint(4, 10, size = (4,4))
print("初始水果价格")
print(prices)
prices[supermarkets == '大润发', fruits == '苹果'] += 1
prices[supermarkets == '好德', fruits == '香蕉'] += 1
print("\n选择'大润发'的苹果和'好德'的香蕉，并将价格增加1元")
print(prices)
prices[supermarkets == '农工商'] -= 2
print("\n'农工商'水果大减价，所有水果价格减2元")
print(prices)
mapple = prices[fruits == '苹果'].mean()
print("\n苹果均价", mapple)
mmango = prices[fruits == '芒果'].mean()
print("芒果均价", mmango)
print("\n橘子价格最贵的超市名称：", supermarkets[prices[:, fruits == '橘子'
].argmax()])
```

```
Run: Unit+1 x
/Users/reneelin/opt/anaconda3/python.app/Contents/MacOS/Python
初始水果价格
[[6 7 8 5]
 [6 5 7 6]
 [9 6 5 7]
 [8 8 7 7]]

选择'大润发'的苹果和'好德'的香蕉, 并将价格增加1元
[[7 7 8 5]
 [6 5 7 6]
 [9 7 5 7]
 [8 8 7 7]]

'农工商'水果大减价, 所有水果价格减2元
[[7 7 8 5]
 [6 5 7 6]
 [9 7 5 7]
 [6 6 5 5]]

苹果均价 6.75
芒果均价 5.5

橘子价格最贵的超市名称: 大润发
```

图 7: 综合 1 实验结果

2. 基于 2.3 节中随机游走的例子, 使用 `ndarray` 和随机数生成函数模拟一个物体在三维空间随机游走的过程。

- (1) 创建 3×10 的二维数组, 记录物体每一步在三个轴向上的移动距离。在每个轴向的移动距离服从标准正态分布 (期望为 0, 方差为 1)。行序 0、1、2 分别对应 x、y 和 z 轴;
- (2) 计算每一步走完后物体在三维空间的位置;
- (3) 计算每一步走完后物体距离原点的距离;
- (4) 统计物体在 z 轴上到达的最远距离; (提示: 使用 `abs()` 绝对值函数对 z 轴每一步运动后的位置求绝对值, 然后求最大距离)
- (5) 统计物体在三维空间距离原点的最近距离值。

```
import numpy as np
steps = 10
rndwlk = np.random.normal(0, 1, size = (3, steps))
position = rndwlk.cumsum(axis = 1)
print("每步走完后物体在三维空间的位置: \n", position)
dists = np.sqrt(position[0]**2 + position[1]**2 + position[2]**2)
np.set_printoptions(precision = 2)
print("每步走完后物体到原点的距离 (只显示两位小数): ", dists)
print("物体在z轴上到达的最远距离: ", abs(position[2]).max())
print("物体在三维空间距离远点的最近值: ", dists.min())
```

```
Run: Unit+2
/Users/reneelin/opt/anaconda3/python.app/Contents/MacOS/Python /Users/reneelin/PycharmProjects/DATA+W2/Unit+2.py
每步走完后物体在三维空间的位置:
[[ 0.59399432  0.88435552  0.69839387  0.82076881  1.0224471  0.79278343
  1.13918385 -0.92132897 -0.85881887  0.23728545]
 [ 1.68261893  1.4468887  0.58457387  1.6847482  2.59831191  2.13232981
  2.4983631  0.77124765 -0.23458898 -0.79738136]
 [ 1.81176244  0.86953879  0.81521481 -0.19968646  1.53438342  2.16274987
  1.83815938  1.99545664  1.58889484  0.23289926]]
每步走完后物体到原点的距离 (只显示两位小数): [1.99 1.91 0.91 1.62 3.18 3.14 3.29 2.33 1.52 0.86]
物体在z轴上到达的最远距离: 2.1627498781957
物体在三维空间距离远点的最近值: 0.8637879885439622
```

图 8: 综合 2 实验结果