



You are taking "Quiz" as a timed exam. The timer on the right shows the time remaining in the exam. To receive credit for problems, you must select "Submit" for each problem before you select "End My Exam".

结束我的考试**1:09:22**

[课程](#) > [Quiz](#) > [Quiz](#) > Proble...

Problem 5

In lecture, we explored the concept of a random walk, using a set of different models of drunks. Below is the code we used for locations and fields and the base class of drunks – you should not have to study this code in detail, since you have seen it in lecture.

Code from Lecture

[Click to see the Location, Field, and Drunk classes from lecture](#)

New code

The following function is new, and returns the actual x and y distance from the start point to the end point of a random walk.

```
def walkVector(f, d, numSteps):
    start = f.getLoc(d)
    for s in range(numSteps):
        f.moveDrunk(d)
    return(f.getLoc(d).getX() - start.getX(),
           f.getLoc(d).getY() - start.getY())
```

drunk variations

Here are several different variations on a drunk.



```

class UsualDrunk(Drunk):
    def takeStep(self):
        stepChoices =\
            [(0.0, 1.0), (0.0, -1.0), (1.0, 0.0), (-1.0, 0.0)]
        return random.choice(stepChoices)

class ColdDrunk(Drunk):
    def takeStep(self):
        stepChoices =\
            [(0.0, 0.9), (0.0, -1.03), (1.03, 0.0), (-1.03, 0.0)]
        return random.choice(stepChoices)

class EDrunk(Drunk):
    def takeStep(self):
        ang = 2 * math.pi * random.random()
        length = 0.5 + 0.5 * random.random()
        return (length * math.sin(ang), length * math.cos(ang))

class PhotoDrunk(Drunk):
    def takeStep(self):
        stepChoices =\
            [(0.0, 0.5), (0.0, -0.5),
             (1.5, 0.0), (-1.5, 0.0)]
        return random.choice(stepChoices)

class DDrunk(Drunk):
    def takeStep(self):
        stepChoices =\
            [(0.85, 0.85), (-0.85, -0.85),
             (-0.56, 0.56), (0.56, -0.56)]
        return random.choice(stepChoices)

```

problem

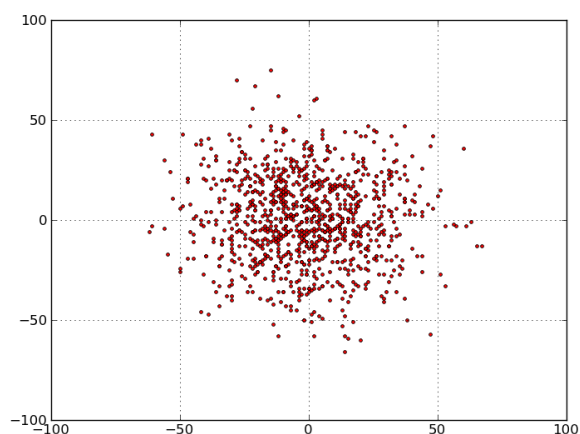
Suppose we use a simulation to simulate a random walk of a class of drunk, returning a collection of actual distances from the origin for a set of trials.

Each graph below was generated by using one of the above five classes of a drunk (UsualDrunk, ColdDrunk, EDrunk, PhotoDrunk, or DDrunk). For each graph, indicate which Drunk class is mostly likely to have resulted in that distribution of distances. Click on each image to see a larger view.



Problem 5-1

2.0/2.0 points (graded)



Select from drop down box.

UsualDrunk ▼

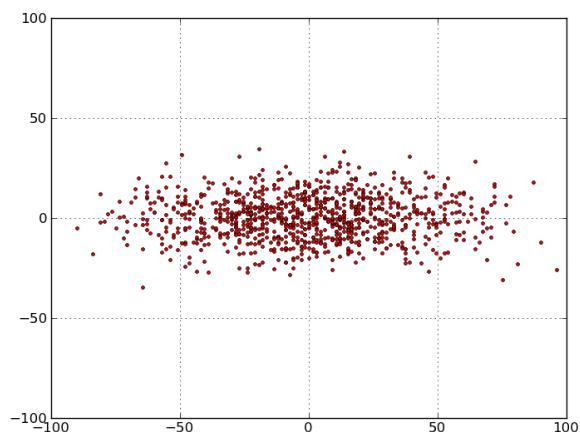


提交

You have used 1 of 1 attempt

Problem 5-2

2.0/2.0 points (graded)



Select from drop down box.

PhotoDrunk ▼



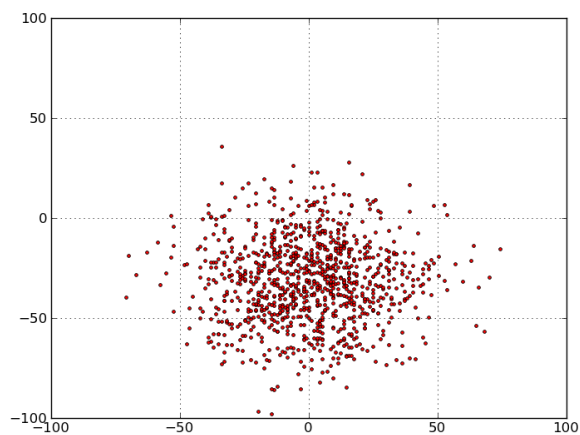
提交

You have used 1 of 1 attempt



Problem 5-3

2.0/2.0 points (graded)



Select from drop down box.

ColdDrunk ▼

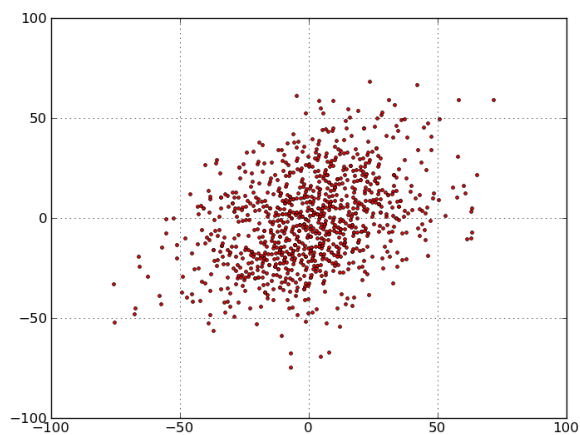


提交

You have used 1 of 1 attempt

Problem 5-4

2.0/2.0 points (graded)



Select from drop down box.

DDrunk ▼



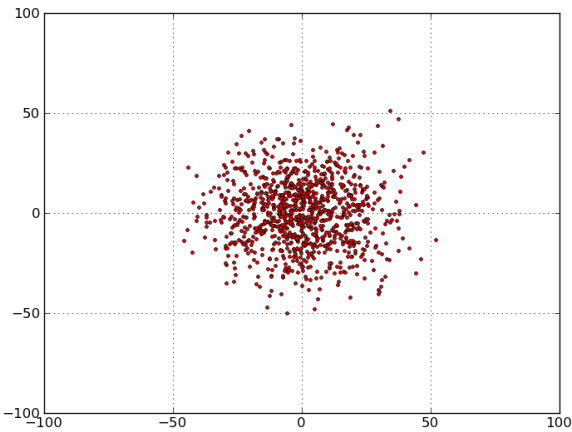
提交

You have used 1 of 1 attempt



Problem 5-5

2.0/2.0 points (graded)



Select from drop down box.

EDrunk ▼



提交

You have used 1 of 1 attempt

[Learn About Verified Certificates](#)

© All Rights Reserved

