

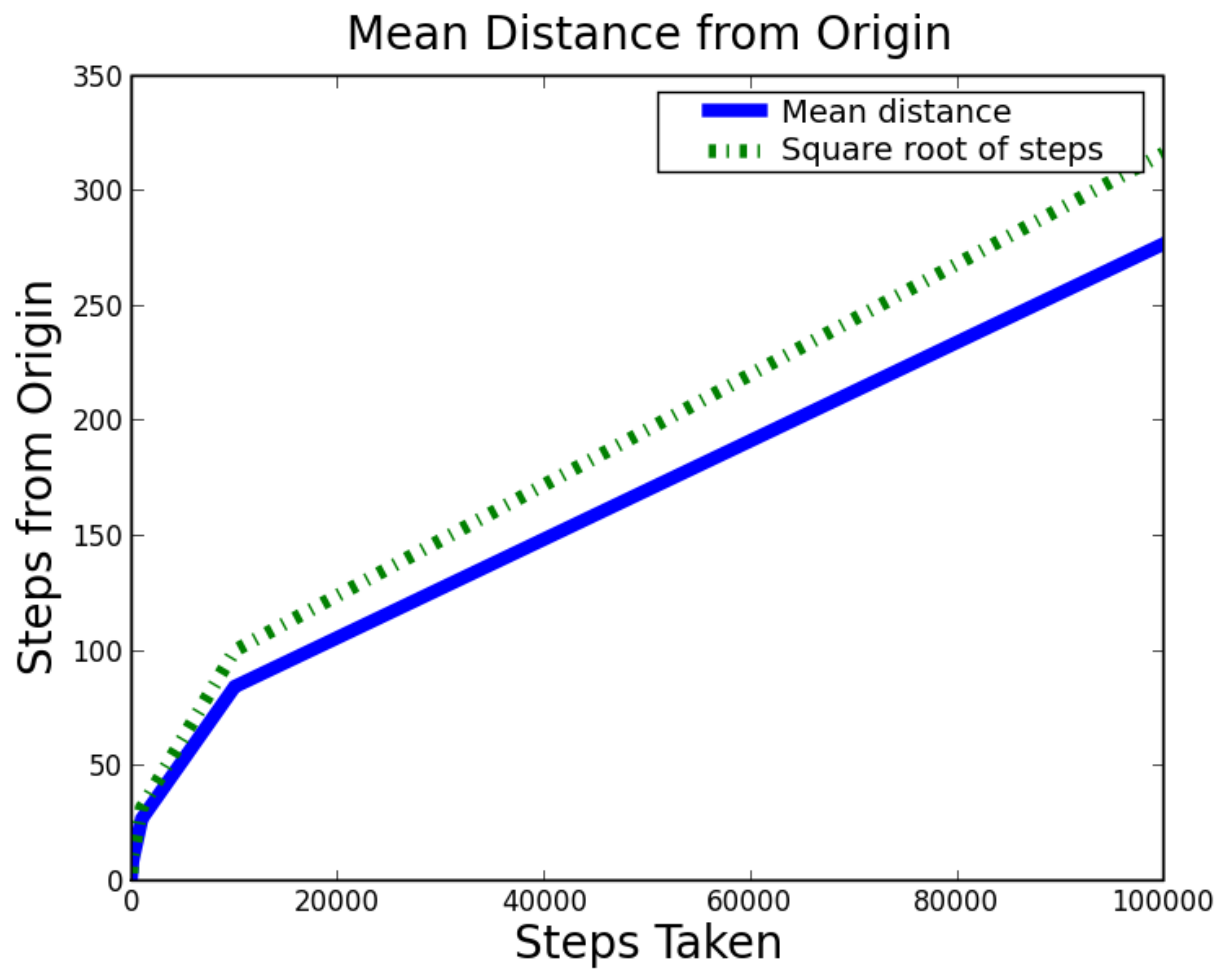
Random Walks and Simulation Models

Lecturer: John Guttag



6.00x

Random Walks



6.00x

Random Walks

```
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 = \
            [(0.0,0.95),(0.0,-1.0),(1.0,0.0),(-1.0,0.0)]
        return random.choice(stepChoices)
```

```
class EDrunk(Drunk):  
    def takeStep(self):  
        deltaX = random.random()  
        if random.random() < 0.5:  
            deltaX = -deltaX  
        deltaY = random.random()  
        if random.random() < 0.5:  
            deltaY = -deltaY  
        return (deltaX, deltaY)
```

```
def simWalks(numSteps, numTrials):  
    homer = UsualDrunk('Homer')  
    origin = Location(0, 0)  
    distances = []  
    for t in range(numTrials):  
        f = Field()  
        f.addDrunk(homer, origin)  
        distances.append(walk(f, homer, numSteps))  
    return distances
```

```
def drunkTestP(numTrials = 100):  
    stepsTaken = [10, 100, 1000, 10000]  
    for dClass in (UsualDrunk, ColdDrunk, EDrunk):  
        meanDistances = []  
        for numSteps in stepsTaken:  
            distances = simWalks(numSteps, numTrials, dClass)  
            meanDistances.append(sum(distances)/len(distances))  
    pylab.plot(stepsTaken, meanDistances,  
               label = dClass.__name__)
```

■ ■ ■