ECT Lesson Plan: Randomness in Stochastic Models

Lesson plan at a glance...

Core subject(s)	Mathematics
Subject area(s)	Statistics and Probability
Suggested age	11 to 18 years old
Prerequisites	Stochastic and Deterministic Modeling (optional)
Time	**Preparation:** 14 to 26 minutes **Instruction:** 105 to 115 minutes
Standards	**Core Subject:** CCSS Math **CS:** CSTA, UK, Australia

In this lesson plan…

Lesson Overview

Most of the experiments students perform in class are deterministic. The expected outcome is the same every time. However, modern scientific research is often stochastic, that is, it depends on random variables and probability. In this lesson, students will be introduced to methods used to create random numbers as well as ways in which they can be used in scientific experiments. This lesson will cover the following **CT** concepts: **pattern recognition** and **algorithms**.

Materials and Equipment

- For the teacher:
  - *Required:* Books or tape to divide the floor
- For the student:
  - *Required:* Internet-connected computers (1 computer per student recommended)
  - *Required:* A coin to flip
  - *Recommended:* Software
    - Python 2.x (https://www.python.org/) and VPython 6.x (http://vpython.org/)

Preparation Tasks

	If students are using computers, confirm that all students' computers are turned on, logged-in, and connected to the Internet	3 to 5 minutes
	This lesson uses VPython. We recommend you use the latest version of VPython (6.x) which requires an older version of Python (2.x). If you have a 64-bit version of Windows you should use a 64-bit version of both Python and VPython. ● Go to https://www.python.org/downloads/ and Install Python 2.x ● Go to http://vpython.org/ and Install VPython 6.x	 5 to 10 minutes 5 to 10 minutes
	Open the ECT Pseudocode Guide	1 minute

The Lesson

Warm-up Activity: What is a random number?	20 minutes
Activity 1: One-dimensional random walk	30 minutes
Activity 2: Two-dimensional random walk and Brownian Motion	45 minutes
Wrap-up Activity: Assessments	10 to 20 minutes

Warm-up Activity: What is a random number? (20 minutes)

Activity Overview: Students will explore what randomness is in a set of numbers and how to determine if a set of numbers is truly randomized. Culture has made "randomness" a common word, and with that students may have misconceptions about what numbers might or might not be **random**. In this activity, students will explore what determines if a number is random and how that relates to the generation of the numbers and not the numbers themselves. This activity explores the relationship between randomness and **patterns** in order to recognise randomness (not patterns).

Notes to the Teacher:
Culture has made "randomness" a common word, and with that students may have misconceptions about what numbers might or might not be random.

Activity:
Ask students for words or situations where they might expect to see randomness present. Write these words on the board. Next, show them sets of numbers and ask them which they think are random and which are not.

- 1, 2, 3, 4, 5
- 6, 5, 6, 3, 3
- 4, 4, 4, 4, 4
- 3, 2, 3, 4, 1

Ask students the following questions.

Q1: If we were to roll five dice, would it be possible to get each of the options above?

Q2: Are any of the options above more likely than the other if all of the dice are rolled at the same time?

Q3: There are stories of people who have won the lottery more than once. How does this affect the randomness of the lottery?

Q4: For each scenario, describe what you would do to ensure your method was random:

1. **Picking a card from a deck**
2. **Rolling a dice**
3. **Driving a car to a destination**
4. **Selecting a word in the dictionary**

Q5: Can a computer generate random numbers? Why or why not?

Teaching Tips:

- The answer is that we don't know which of the above sets of numbers are randomly-generated. The fact that some of them seem to resemble a pattern reflects on what we have learned and experienced, not whether they are random. To an alien race, option **d** might follow a pattern they are aware of. The fact that we don't see randomness in options **a** and **c** is part of a larger philosophical discussion known as the Anthropic Principle (http://wikipedia.org/wiki/Anthropic_principle).
- What makes numbers random are not the results but the method used to determine the numbers.

Assessment:
A1: Yes

A2: No

A3: It doesn't. While the probability of someone winning the lottery once let alone multiple times is very unlikely, randomness has to do with the method of obtaining the results and not the results itself. Unlikely does not mean that it is impossible.

A4: Answers will vary, below are a few possible answers.

1. For picking a card from a deck, close your eyes while choosing
2. For rolling a dice, place the dice in a cup to shake
3. For driving a car to a destination, flip a coin to determine the direction at each intersection
4. For selecting a word in the dictionary, let the book fall open to a page

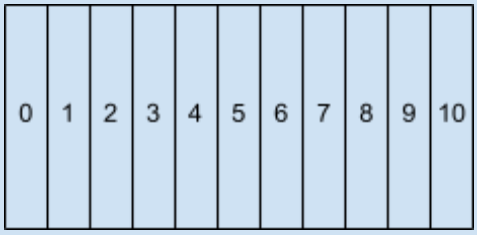
A5: A computer is deterministic, that is, it follows instructions and is incapable of making independent decisions needed for creating a random number. However, some random number generators use information from sensors or human interaction to generate real random numbers. Pseudo-random numbers have faults but are sufficient for most applications.

Activity 1: One-dimensional random walk (30 minutes)

Activity Overview: In this activity, students will practice an activity in which we simulate a one-dimensional random walk using a flipped coin. In this activity, students will will simulate a one-dimensional random walk using a coin to determine if they should move forwards or backwards. Students look for patterns and lack of patterns to give them an overall understanding of the problem and to **generalize and predict behavior** of multiple trials.

Activity:

1. If your classroom has a tiled floor, then you are already set up for this activity. If your floor does not, you might want to place tape or books on the floor to divide the class into rows. A classroom view from above is provided below. Students should be able to easily walk from one section to an adjacent one.



2. Have students line up in box 5 and give each student a coin.
3. Tell the students to flip the coin once when you say, “Go!”
    a. If the coin is heads, step to the box with the next higher number.
    b. If the coin is tails, step into the box with the next lower number.
    c. If there is nowhere left to move, stay in place for that turn.
4. Do a trial run to make sure everyone understands the game. Go back to starting position and play the game for 5 flips
5. After, record the number of students in each box on a whiteboard or spreadsheet.
6. If time permits, do another trial with 10 flips.
7. Have students develop Pseudocode for this simulation:
   Example pseudocode:
    a. Start at spot 0.
    b. Flip a coin.
    c. If it’s heads go forwards.
    d. If it’s tails go backwards.
    e. If moving would cause you to go beyond spot 0 and 10 don’t move.
    f. Repeat steps 2-5 for the number of specified flips.
    g. Keep track of your final position
    h. Repeat steps 1-7 for the number of specified trials.
8. Open Python and type **Ctrl-N** (PC) or **Command-N** (Mac) to have a blank page to enter in code.
9. After entering the code, press **F5** or from the menu **Run** → **Run Module** to run the code.

Example of One-Dimensional Random Walk Code

```
#1D Random Walk Simulation ('#' are comments and are ignored)
from random import *
min_position = 0  #Minimum position of the student
max_position = 10 #Maximum position of the student
start_position = 0 #Position student starts each trial
data_tracker = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] #Records final position
                                               #for each trial

num_trials = input("How many trials?")
num_flips = input("How many flips per trial?")
for trial in range(0, num_trials):
    student_position = start_position
    for flip in range(0, num_flips):
        coin = randint(0, 1) #Returns either 0 or 1 (like heads/tails)
        if coin == 1 and (student_position < max_position):
            student_position += 1 #Add 1 to student_position
        if coin == 0 and (student_position > min_position):
            student_position -= 1 #Subtract 1 from student_position
        else:
            continue #If position is at the edge, don't move.
        data_tracker[student_position] += 1


#Number of times the person finished their walk in that spot.
print data_tracker
```

Sample outputs:

```
How many trials? 100
How many flips per trial? 100
[664, 592, 522, 434, 383, 322, 335, 307, 337, 330]

How many trials? 100
How many flips per trial? 100
[595, 541, 511, 472, 433, 336, 342, 359, 351, 341]
```

Ask students the following questions:

Q1: The "coin flip" in line 14 is pseudo-random. Do you see a pattern in the results that implies that this is not random enough for the program?

Q2: How does the result of this experiment compare to the class activity?

Q3: What is the purpose of the `continue` statements in line 20?

Q4: Try a relatively large number of trials and flips (1000 or 10000). Does the data seem to favor one spot over the other? What about a subset (e.g. lower numbers, higher numbers, middle)?

Q5: How would the pattern of the data change if when the `students_position` exceeded the boundaries, it wrapped around to the other side. Example: If the student is in position 0 and flips Tails, they move to position 10.

Q6: With this experiment, is there an expected outcome, can you predict what will happen over the long run?

Q7: After line 20's, `continue` add a new line and type `print "Out of money"`. Why do you think one name for this experiment is Gambler's ruin (http://wikipedia.org/wiki/Gambler%27s_ruin)?

Teaching Tips:

- This simulation did not have enough trials to give meaningful data. Students can simulate this game using a Python program. This allows them to test their predictions and run experiments with huge numbers of trials where the data will approach the expected results. Based on the Law of Large Numbers (https://en.wikipedia.org/wiki/Law_of_large_numbers), with enough simulations we should be able to make better guesses as to the ideal outcomes from this experiment.

Assessment:
A1: There shouldn't be one. It is random enough.

A2: The results should be similar, the exact number may differ for each position, but it will still be randomly distributed.

A3: If changing the student's position would put them outside the boundary of boxes, the position is not changed and the program continues with the next trial.

A4: In this experiment the data may tend to be closer to the boundaries edge because the loop prevents the position from going beyond it.

A5: Since all positions are equally likely now, the data would be more likely to be evenly distributed.

A6: While general conclusions can be made, there is no expected outcome, all scenarios are equally likely.

A7: If the loop gets to a boundary, this is equivalent in a casino to running out of money. A gambler with a finite amount of money playing a fair game against a casino with an infinite amount of money will lose all of their money eventually. In reality, casinos do not have an infinite amount of money but one that far exceeds any individual's funds. This is why casinos are able to stay profitable without cheating, in the end "The House always wins".

Activity 2: Two-dimensional random walk and Brownian Motion (45 minutes)

Activity Overview: What happens if you introduce two degrees of freedom into a random walk? Students will now use computer programming to simulate a two-dimensional random walk which models well the brownian motion of particles diffusing through a medium. In this activity, students will use a Python program to simulate brownian motion using a two-dimensional random walk model. Students modify their **algorithm** to adapt it to new situations. Students also look for **patterns and lack of patterns** to give them an overall understanding of the problem and to **generalize and predict behavior** of multiple trials.

Notes to the Teacher:
The video in the activity shows an experiment where ink is added to two jars of different temperatures. The jar on the right has a temperature of 79.6 degrees Celsius and the one on the left, a temperature of 80.8 degrees celsius. Ink is added to the jars and the ink on the left seems to disappear because it diffuses more quickly into the water. This is because of the higher energy of the water on the left moving the particles much more quickly away from each other. Brownian Motion is the phenomena of particles moving through a medium like a gas or liquid. A random walk can be used to simulate this motion quite effectively.

Activity:

1. Have students watch the following video with students on Brownian Motion (http://www.youtube.com/watch?v=brRLLObw30Y).
2. Explain to the students that calculating the interactions of 3 or more moving objects is extremely complex (example: planets/moon orbits). With molecules in a liquid or in air, the possibilities are so numerous that we can only talk in probabilities.
2. The random walk in 2 or 3 dimensions is an extremely helpful model for mathematicians and scientists. This model has been used to estimate absorption of radioactivity into the body and diffusion of medicines and

nutrients into cells.

3. Have students develop pseudocode for this simulation.
   Example Pseudocode for a 2 Dimensional Walk:
   a. Choose a random direction (0, 90, 180, 270 degrees)
   b. Move a certain distance in that direction
   c. Repeat steps 1 and 2.

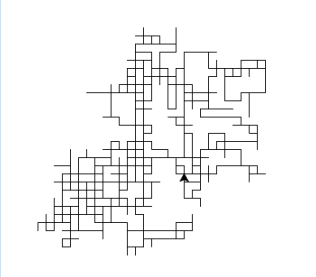
5. Open Python and type **Ctrl-N** (PC) or **Command-N** (Mac) to have a blank page to enter in code.
6. After entering the code, press **F5** or from the menu **Run** → **Run Module** to run the code.

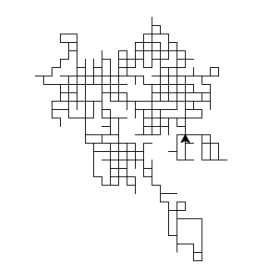
Example of Two-Dimensional Random Walk Code (using only Python)

```
#2D Random Walk Simulation ('#' are comments and are ignored)
from turtle import *
from random import *
d = 10 #The length of each "step"
speed("fast") #Drawing speed. "fastest" "fast" "normal" "slow" "slowest"
steps = input("How many steps?")
for step in range(0, steps): #Pseudocode step c
   right(randint(0, 3) * 90) #Pseudocode step a
   forward(d) #Pseudocode step b. Move forward by d steps.
```

Sample outputs:

```
How many steps? 100
```



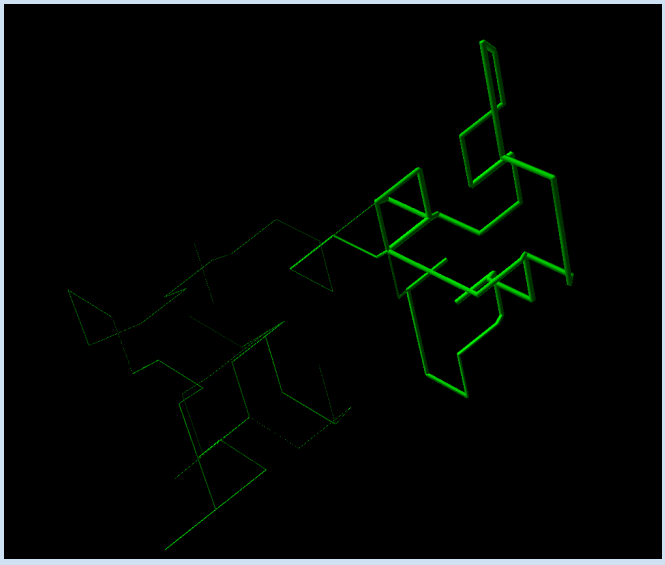


Optional Example of Three-Dimensional Random Walk Code (using VPython)

```
#3D Random Walk Simulation ('#' are comments and are ignored)
#Hold mouse middle button to zoom, hold mouse right button to pan view.
from visual import *
from random import *
scene.autocenter = True #Keeps the screen focused on the center

d = 100 #The length of each "step"
x = 0 #Start the walk at (0,0,0)
y = 0
z = 0

random_walk = curve(pos=[(x, y, z)], radius = 10, color = color.green)
steps = input("How many steps?")

for step in range(0, steps):
    rate(10) #No more than 10 calculations/second, makes it watchable
```

```
    x+= choice((-1, 1)) * d #Either walk forwards, 1*d, or backwards, -1*d
    y+= choice((-1, 1)) * d
    z+= choice((-1, 1)) * d
    random_walk.append(pos=(x, y, z)) #Add this new point to the walk
```

Sample output:
(zoom by holding the middle mouse button, pan with the right mouse button)
`How many steps?100`



(The darker line means the “walk” retraced previous steps)

Ask students the following questions.

Q1: What was the difference between the two jars of water?

Q2: Why is that affecting the way the ink diffuses?

Q3: What would be the difficulty in coming up with a formula or model that models Brownian Motion?

Q4: How does a random walk simulate brownian motion of particles in water?

Q5: Random walks are used to predict the diffusion rate of medicines into cells. How can a random simulation tell us specific information?

Q6: What change could you make to the VPython code to make it 2D?

Q7: What is the purpose of `rate(10)` in line 18 of the VPython code? Hint: Try commenting it out or increasing/decreasing the number.

Q8: In the VPython code, why was `choice` used rather than `randint`?

Q9: In the video shown at the beginning, the water with the higher temperature diffused the ink molecules more quickly. How could you modify either code to simulate a higher temperature/velocity?

Assessment:
A1: The one on the left has a higher temperature than the one on the right.

A2: The higher temperature is causing the ink particles to move away from each other faster until they are no longer visible.

A3: Too many interactions between particles, a seemingly endless number of possibilities and combinations.

A4: Particles are colliding at such a high rate in the water that their movements are essentially random.

A5: The simulation is random but if performed numerous times, it can give a general timeframe for how long it would take to diffuse into a cell. In a real situation all probabilities are possible.

A6: Comment out line 20 (add a # at the beginning of the line).

A7: Slows down the code so you can see it being traced out. Removing it would make the walk appear instantly.

A8: So the particle would always be moving, a zero would give it zero velocity which would be impossible in a ideal fluid.

A9: Increased, the size of the step.

Wrap-up Activity: Assessment (10 to 20 minutes)

Activity Overview: Now, it is time to assess student understanding of randomness and random walk simulations. In this activity, students will assess their understanding of random walks. Students will be assessed on **algorithms**, **pattern recognition**, **pattern generalization**, and the distinction between patterns and randomness.

Notes to the Teacher:
Lead students through an assessment activity to assess understanding and skills.

Activity:
Have students who have answered the questions throughout assess the other students' understanding of methods for obtaining a random number.
Discuss the below question with students.

Q1: Some models of stock market fluctuations involve a random walk. Why or why not is this a good idea?

Assessment:
A1: While it is effective to a degree, a random walk requires that each step be independent (not influence) the next step. Some would argue that unlike natural phenomena, such as water molecules, stock traders do have prior knowledge that influences the movement of the market.

Learning Objectives and Standards

Learning Objectives	Standards
LO1: Students will be able to define randomness and label the output of a situation, scenario, or task as	*Core Subject* CCSS HS Math S-MD 6**:** Use probabilities to make unbiased decisions (e.g., drawing by lots, using a random number generator). CCSS MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

random or not random or pseudorandom.	CCSS MATH.PRACTICE.MP7: Look for and make use of structure. *Computer Science* AUSTRALIA 8.4 (Collecting, managing and analyzing data): Analyse and visualise data using a range of software to create information; and use structured data to model objects or events. CSTA L3A.CT.8: Use modeling and simulation to represent and understand natural phenomena. UK 4.2: Develop and apply their analytic, problem-solving, design, and computational thinking skills.
LO2: Students will be able to determine the relative probability of any outcome in a set of randomly-generated numbers.	*Common Core* CCSS.MATH.CONTENT.7.SP.C.5: Understand that the probability of a chance event is a number between 0 and 1 that expresses the likelihood of the event occurring. Larger numbers indicate greater likelihood. A probability near 0 indicates an unlikely event, a probability around 1/2 indicates an event that is neither unlikely nor likely, and a probability near 1 indicates a likely event. CCSS.MATH.CONTENT.7.SP.C.6: Approximate the probability of a chance event by collecting data on the chance process that produces it and observing its long-run relative frequency, and predict the approximate relative frequency given the probability. For example, when rolling a number cube 600 times, predict that a 3 or 6 would be rolled roughly 200 times, but probably not exactly 200 times. CCSS.MATH.CONTENT.HSS.CP.A.1: Describe events as subsets of a sample space (the set of outcomes) using characteristics (or categories) of the outcomes, or as unions, intersections, or complements of other events ("or," "and," "not"). *Computer Science* AUSTRALIA 6.4 (Creating digital solutions by: defining): Define problems in terms of data and functional requirements, and identify features similar to previously solved problems. CSTA L2.CT.1: Use the basic steps in algorithmic problem-solving to design solutions (e.g., problem statement and exploration, examination of sample instances, design, implementing a solution, testing and evaluation). UK 3.2: Understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem.
LO3: Students will be able to write pseudocode to correctly model a 1D and 2D or 3D random walk.	*Core Subject* CCSS.MATH.PRACTICE.MP4: Model with mathematics. MS-PS1-4: Develop a model that predicts and describes changes in particle motion, temperature, and state of a pure substance when thermal energy is added or removed. HS-PS1-5: Apply scientific principles and evidence to provide an explanation about the effects of changing the temperature or concentration of the reacting particles on the rate at which a reaction occurs. HS-PS3-2: Develop and use models to illustrate that energy at the macroscopic scale can be accounted for as either motions of particles or energy stored in fields. *Computer Science*

	CSTA L2.CT.14: Examine connections between elements of mathematics and computer science including binary numbers, logic, sets and functions.
LO4: Students will be able to simulate multiple trials or events of a 1D or 2D random walk with Python code, recognize patterns and lack of patterns in the collected data, generalise and predict probabilities based on the randomness of the model or for a large number of trials, alter the code in order to produce a different result, and predict the output result of altered code of 1D and 2D or 3D random walk scenarios.	*Core Subject* CCSS.MATH.PRACTICE.MP4: Model with mathematics. MS-PS1-4: Develop a model that predicts and describes changes in particle motion, temperature, and state of a pure substance when thermal energy is added or removed. HS-PS1-5: Apply scientific principles and evidence to provide an explanation about the effects of changing the temperature or concentration of the reacting particles on the rate at which a reaction occurs. HS-PS3-2: Develop and use models to illustrate that energy at the macroscopic scale can be accounted for as either motions of particles or energy stored in fields. CCSS MATH.PRACTICE.MP2: Reason abstractly and quantitatively. CCSS MATH.PRACTICE.MP7: Look for and make use of structure. CCSS.MATH.PRACTICE.MP8: Look for and express regularity in repeated reasoning. CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically. *Computer Science* CSTA L2.CT.9: Interact with content-specific models and simulations (e.g., ecosystems, epidemics, molecular dynamics) to support learning and research.
LO5: Students will be able to analyze a code and determine the purpose of a given function with the code.	*Common Core* CCSS MATH.PRACTICE.MP2: Reason abstractly and quantitatively. CCSS MATH.PRACTICE.MP7: Look for and make use of structure. CCSS.MATH.PRACTICE.MP3: Construct viable arguments and critique the reasoning of others. *Computer Science* CSTA L2.CT.14

Additional Information and Resources

Lesson Vocabulary

Term	Definition	For Additional Information
Pseudocode	A list of easily understood instructions used for developing algorithms.	http://en.wikipedia.org/wiki/Pseudocode
Random	Numbers or objects selected where there is equal probability to the others in the set (e.g. each side of the dice is equally likely to be rolled) and zero ability to predict for each individual roll.	http://en.wikipedia.org/wiki/Randomness
Stochastic Models	Equations and algorithms involving randomness and probability, which therefore may return a different	http://en.wikipedia.org/wiki/Stochastic_process

	output for any given input. It is usually not possible to work backwards to determine the inputs from the results and the equations/algorithms.	

Computational Thinking Concepts

Concept	Definition
Pattern Recognition	Identifying trends and commonalities between data points, groups, or sets
Pattern Generalization	Making predictions and testing models
Algorithm Design	Creating an ordered series of instructions for solving similar problems

Administrative Details

Contact info For more info about Exploring Computational Thinking (ECT), visit the ECT website (g.co/exploringCT)

Credits Developed by the Exploring Computational Thinking team at Google and reviewed by K-12 educators from around the world.

Last updated on 07/02/2015