Algorithmic Analysis

Tim Gianitsos
CS109, Stanford University

Goal: help you be able to do two pset questions

Expectation of the Sum

$$E[X + Y] = E[X] + E[Y]$$

Generalized: $$E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i]$$

Holds regardless of dependency between X_i's

(e.g. they don't need to be independent)

Differential Privacy

Real World Motivation: Can we learn trends from a dataset without violating privacy of *individuals* in the dataset?

Cynthia Dwork's celebrity look alike is Cynthia Dwork.

Key Idea: Randomize some responses so that individuals have "plausible deniability"
-Warner et al (1965)

Differential Privacy

100 independent values $X_1 \ldots X_{100}$ where $X_i \sim \text{Bern}(p)$

```
# Maximize accuracy, while preserving privacy.
def calculateYi(Xi):
    obfuscate = random()
    if obfuscate:
        return indicator(random())
    else:
        return Xi
```

`random()` returns `True` or `False` with equal likelihood

Differential Privacy

100 independent values $X_1 \ldots X_{100}$ where $X_i \sim \text{Bern}(p)$

```python
# Maximize accuracy, while preserving privacy.
def calculateYi(Xi):
    obfuscate = random()
    if obfuscate:
        return indicator(random())
    else:
        return Xi
```

`random()` returns `True` or `False` with equal likelihood

What is $E[Y_i]$?

$$E[Y_i] = P(Y_i = 1) = \frac{p}{2} + \frac{1}{4}$$

Differential Privacy

100 independent values $X_1 \dots X_{100}$ where $X_i \sim \text{Bern}(p)$

```
# Maximize accuracy, while preserving privacy.
def calculateYi(Xi):
    obfuscate = random()
    if obfuscate:
        return indicator(random())
    else:
        return Xi
```

`random()` returns `True` or `False` with equal likelihood

Let $$Z = \sum_{i=1}^{100} Y_i$$

What is the E[Z]?

$$E[Z] = E\left[\sum_{i=1}^{100} Y_i\right] = \sum_{i=1}^{100} E[Y_i] = \sum_{i=1}^{100} \left(\frac{p}{2} + \frac{1}{4}\right) = 50p + 25$$

Differential Privacy

100 independent values $X_1 \ldots X_{100}$ where $X_i \sim \text{Bern}(p)$

```
# Maximize accuracy, while preserving privacy.
def calculateYi(Xi):
    obfuscate = random()
    if obfuscate:
        return indicator(random())
    else:
        return Xi
```

`random()` returns `True` or `False` with equal likelihood

Let $Z = \sum_{i=1}^{100} Y_i \quad E[Z] = 50p + 25$

How do you estimate *p*?

$$p \approx \frac{Z - 25}{50}$$

Challenge: What is the probability that our estimate is good?

Computer Cluster Utilization

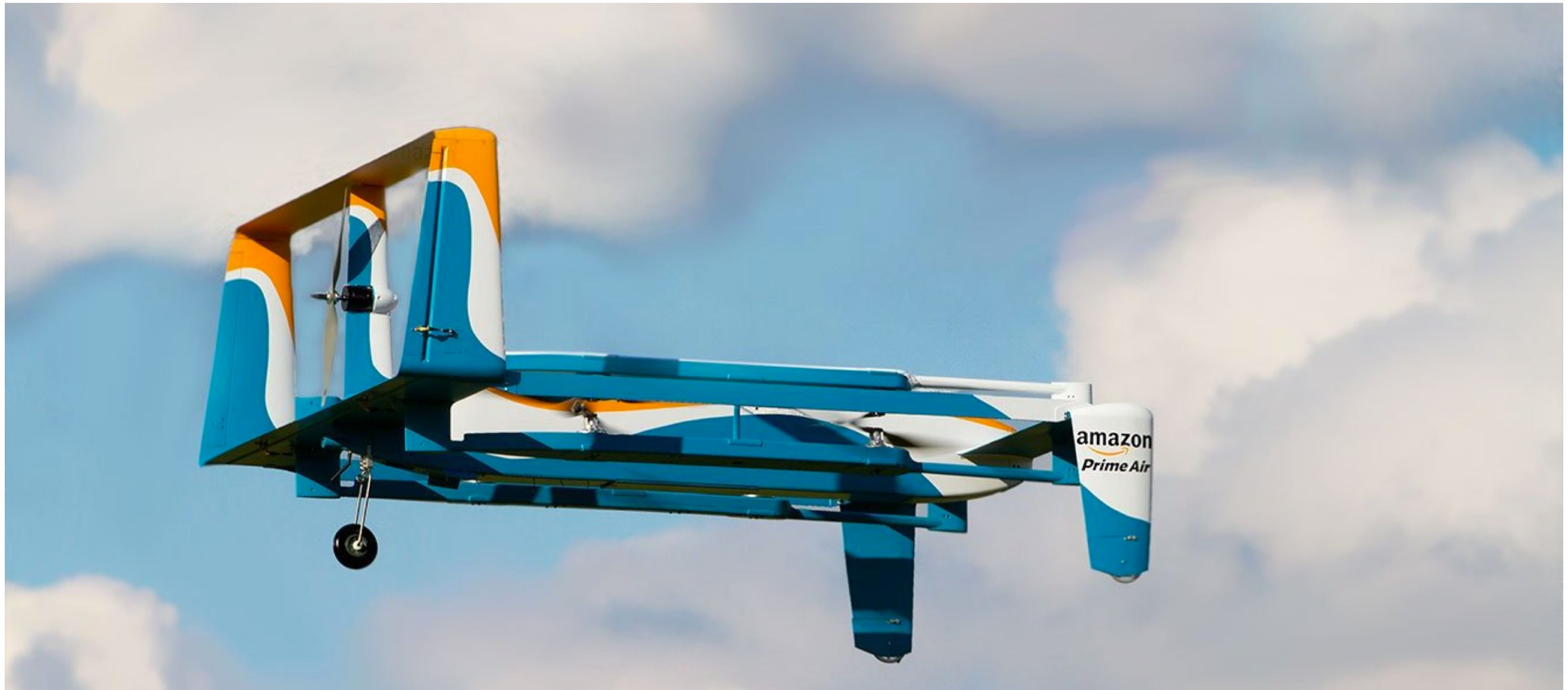
- Computer cluster with k servers
  - Requests independently go to server i with probability p_i
  - Let event A_i = server i receives no requests
  - X = # of events $A_1, A_2, \ldots A_k$ that occur
  - E[X] after first n requests?
  - Since requests independent: $P(A_i) = (1 - p_i)^n$

Let Bernoulli B_i be an indicator for A_i

$$X = \sum_{i=1}^{k} B_i$$

$$E[X] = E\left[\sum_{I=1}^{k} B_i\right] = \sum_{i=1}^{k} E[B_i] = \sum_{i=1}^{k} P(A_i) = \sum_{i=1}^{k} (1 - p_i)^n$$

amazon

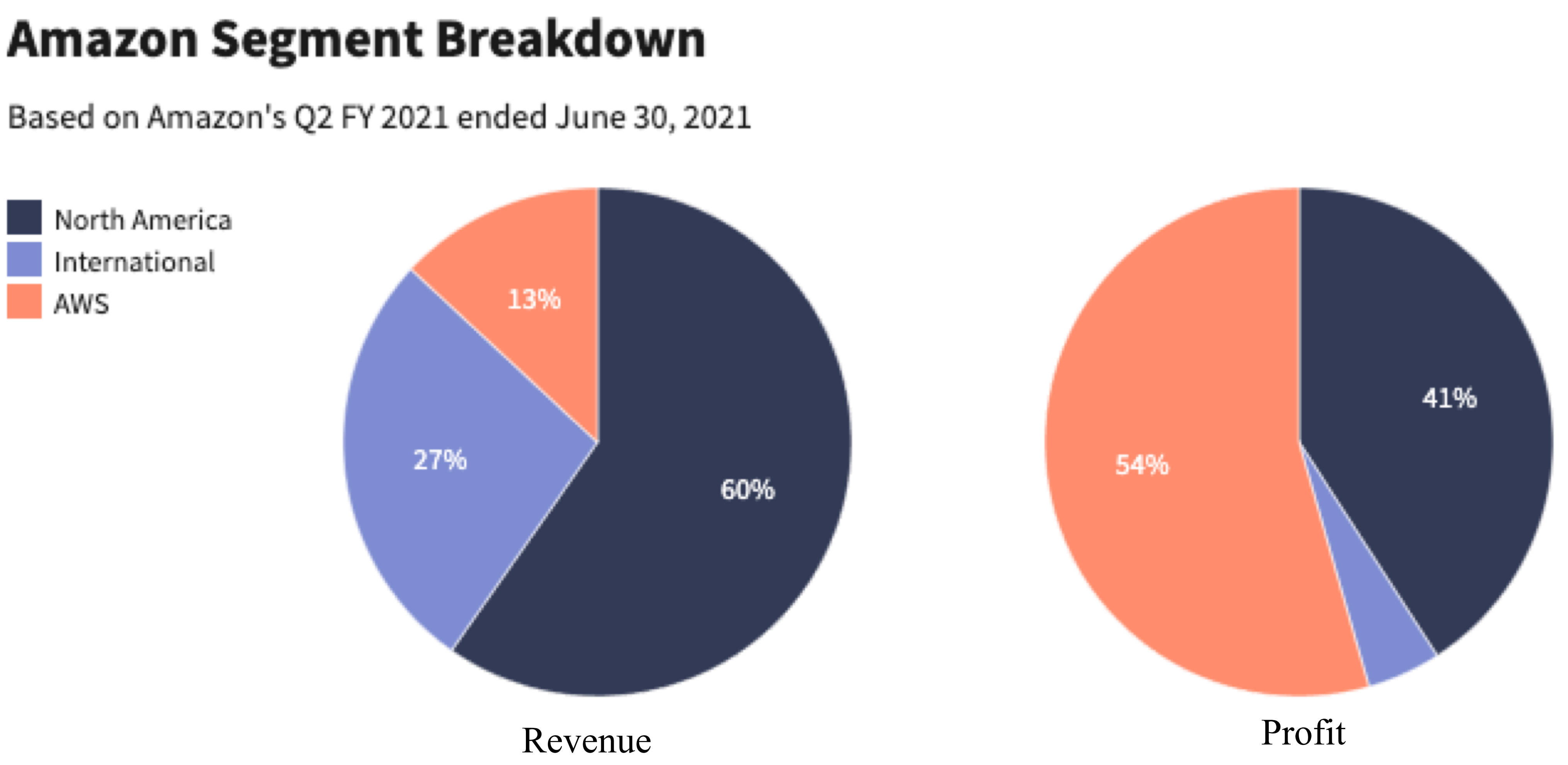


amazon web services™

* 52% of Amazons Profits

**More profitable than Amazon's North America commerce operations

http://www.zdnet.com/article/amazons-finds-its-profit-horse-in-aws-why-its-so-disruptive-to-its-old-guard/


Amazon Segment Breakdown
Based on Amazon's Q2 FY 2021 ended June 30, 2021
North America
International
AWS
13%
27%
60%
41%
54%
Revenue
Profit

One of the reason expectation is so useful is because the expectation of a sum is easy to calculate

Conditional Expectation

Conditional Expectation

Conditional Distributions Expectation

Conditional expectation

Recall the the conditional PMF of X given $Y = y$:

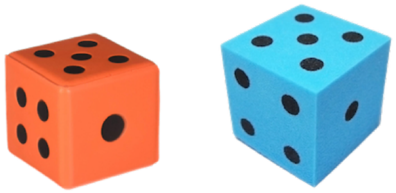
$$P(X = x|Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}$$

The conditional expectation of X given $Y = y$ is

$$E[X|Y = y] = \sum_{x} xP(X = x|Y = y)$$

It’s been so long, our dice friends

$$E[X|Y = y] = \sum_x x p_{X|Y}(x|y)$$



- Roll two 6-sided dice.
- Let roll 1 be D_1, roll 2 be D_2.
- Let S = value of $D_1 + D_2$.

1. What is $E[S|D_2 = 6]$?

$$E[S|D_2 = 6] = \sum_{x=7}^{12} xP(S = x|D_2 = 6)$$

$$= \left(\frac{1}{6}\right)(7 + 8 + 9 + 10 + 11 + 12)$$

$$= \frac{57}{6} = 9.5$$

Intuitively: $6 + E[D_1] = 6 + 3.5 = 9.5$

Let’s prove this!

Properties of conditional expectation

1. LOTUS:

$$E[g(X)|Y = y] = \sum_x g(x) p_{X|Y}(x|y)$$

2. Sum of conditional expectation:

$$E\left[\sum_{i=1}^{n} X_i \mid Y = y\right] = \sum_{i=1}^{n} E[X_i|Y = y]$$

Conditional Expectation Function

This is a number:

$$E[X]$$

This is a function of y:

$$E[X|Y = y]$$

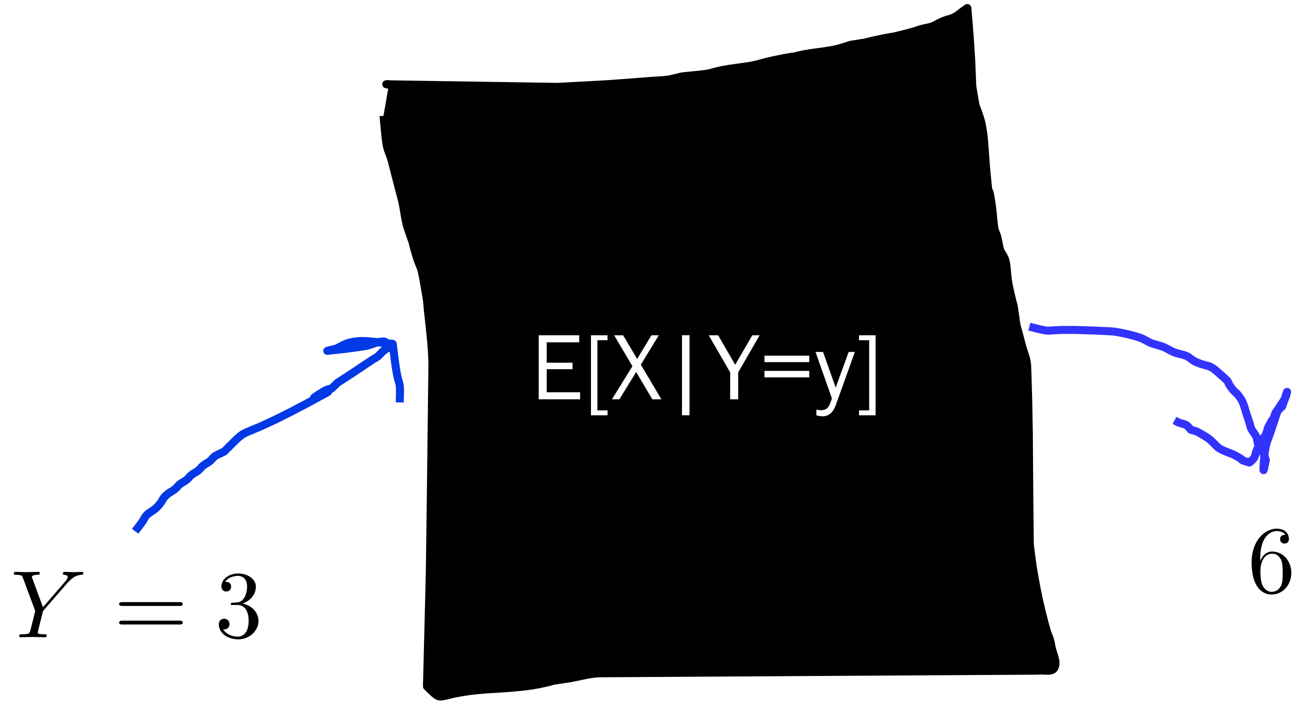
$E[X = 5]$ Doesn't make sense. Take expectation of random variables, not events

Conditional Expectation Function

$E[X|Y=y]$

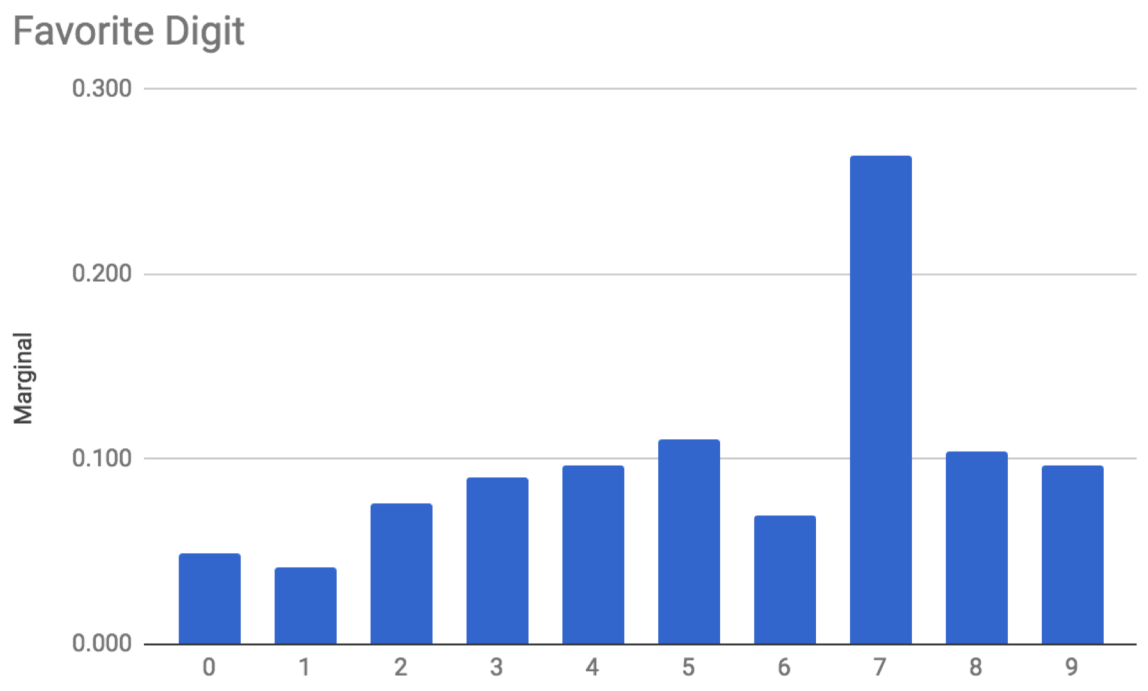
This is a function with y as input

Conditional Expectation Function



Conditional Expectation Example

X = favorite number
Y = year in school



$E[X] = 0 * 0.05 + \ldots + 9 * 0.10 = 5.38$

Conditional Expectation Example

X = favorite number
Y = year in school

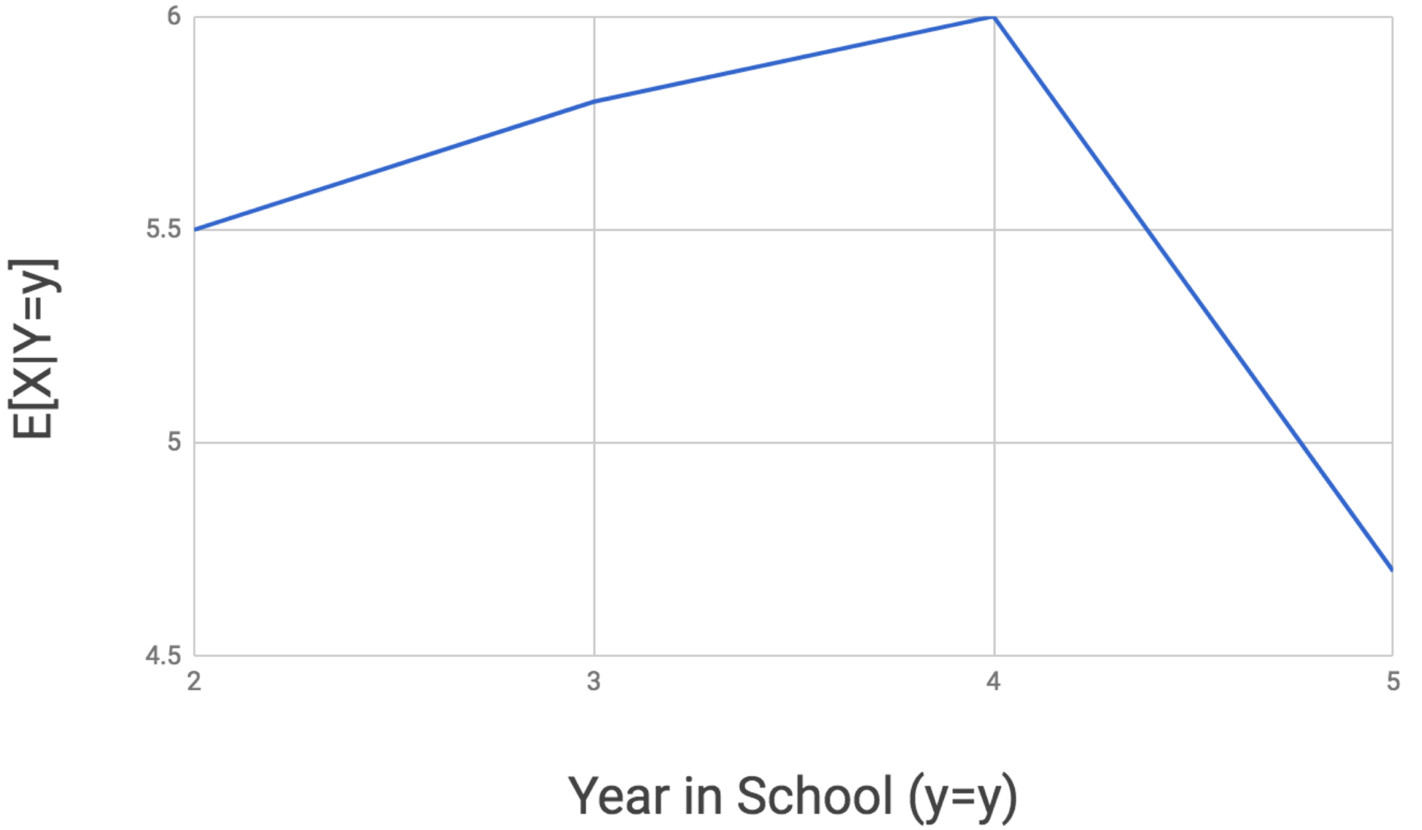
E[X | Y] ?

Year in school, Y = y	E[X | Y = y]
2	5.5
3	5.8
4	6.0
5	4.7

Conditional Expectation Example

X = favorite number
Y = year in school

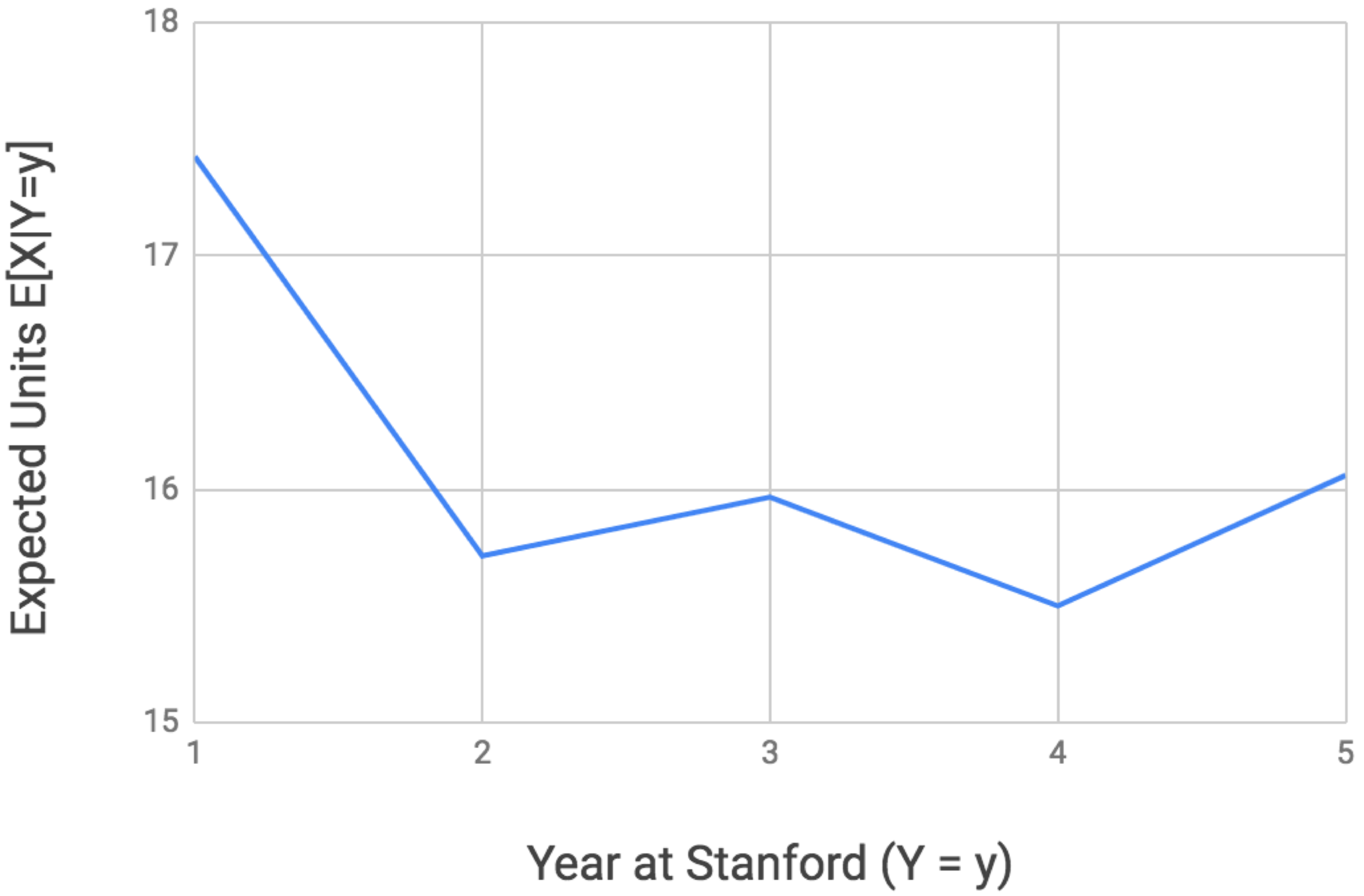
E[X | Y] ?



Conditional Expectation Example

X = units in fall quarter
Y = year in school

E[X | Y] ?



Law of Total Expectation

Properties of conditional expectation

1. LOTUS:

$$E[g(X)|Y=y] = \sum_x g(x) p_{X|Y}(x|y)$$

2. Linearity of conditional expectation:

$$E\left[\sum_{i=1}^{n} X_i \mid Y=y\right] = \sum_{i=1}^{n} E[X_i|Y=y]$$

3. Law of total expectation:

$$E[X] = E\big[E[X|Y]\big]$$

what?!

Law of Total Expectation

If $E[X|Y=y]$ is a function of y, then $E[X|Y]$ is a random variable!

If $E[X|Y]$ is a random variable, we can get its expected value, $\mathrm{E}[E[X|Y]]$!

Remarkably, $\mathrm{E}\big[E[X|Y]\big] = E[X]$

$$E[X] = \sum_y E[X|Y=y]P(Y=y)$$

Proof of Law of Total Expectation

$E[X] = E[E[X|Y]]$

$$E[E[X|Y]] = \sum_y E[X|Y=y]P(Y=y)$$

g(Y) = E[X|Y]

$$= \sum_y \sum_x xP(X=x|Y=y)P(Y=y)$$

Def of E[X|Y]

$$= \sum_y \sum_x xP(X=x, Y=y)$$

Chain rule!

$$= \sum_x \sum_y xP(X=x, Y=y)$$

I switch the order of the sums

$$= \sum_x x \sum_y P(X=x, Y=y)$$

Move that x outside the y sum

$$= \sum_x xP(X=x)$$

Marginalization

$$= E[X]$$

Def of E[X]

Another way to compute $E[X]$

$E[X] = E[E[X|Y]]$

$$E[E[X|Y]] = \sum_y P(Y = y)E[X|Y = y] = E[X]$$

If we only have a conditional PMF of X on some discrete variable Y, we can compute $E[X]$ as follows:

1. Compute expectation of X given some value of $Y = y$
2. Repeat step 1 for all values of Y
3. Compute a weighted sum (where weights are $P(Y = y)$)

```
def recurse():
  if (random.random() < 0.5):
      return 3
  else: return (2 + recurse())
```

Useful for analyzing recursive code!!

Quick check

A. value
B. one RV
C. two RVs
D. doesn't make sense

1. $E[X]$
2. $E[X, Y]$
3. $E[X + Y]$
4. $E[X|Y]$
5. $E[X|Y = 6]$
6. $E[X = 1]$
7.* $E[Y|X = x]$

Quick check

A. value
B. one RV
C. two RVs
D. doesn't make sense

1. $E[X]$ A
2. $E[X, Y]$ D (maybe interpret as E[vector of R.V.s]? But that's beyond scope)
3. $E[X + Y]$ A
4. $E[X|Y]$ B
5. $E[X|Y = 6]$ A
6. $E[X = 1]$ D

7.* $E[Y|X = x]$ A (function of x)

Analyzing recursive code

$$E[X] = E\big[E[X|Y]\big] = \sum_y E[X|Y=y]P(Y=y)$$

```
def recurse():
  # equally likely values 1,2,3
  x = np.random.choice([1,2,3])
  if (x == 1): return 3
  elif (x == 2): return (5 + recurse())
  else: return (7 + recurse())
```

Let Y = return value of `recurse()`.

What is $E[Y]$?

Analyzing recursive code

$$E[X] = E[E[X|Y]] = \sum_y E[X|Y = y]P(Y = y)$$

```python
def recurse():
  # equally likely values 1,2,3
  x = np.random.choice([1,2,3])
  if (x == 1): return 3
  elif (x == 2): return (5 + recurse())
  else: return (7 + recurse())
```

Let Y = return value of `recurse()`.
What is $E[Y]$?

$$E[Y] = E[Y|X = 1]P(X = 1) + E[Y|X = 2]P(X = 2) + E[Y|X = 3]P(X = 3)$$

$$E[Y|X = 1] = 3$$

When $X = 1$, return 3.

Analyzing recursive code

If Y discrete
$$E[X] = E\big[E[X|Y]\big] = \sum_y E[X|Y=y]P(Y=y)$$

```
def recurse():
  # equally likely values 1,2,3
  x = np.random.choice([1,2,3])
  if (x == 1): return 3
  elif (x == 2): return (5 + recurse())
  else: return (7 + recurse())
```

Let Y = return value of `recurse()`.
What is $E[Y]$?

$$E[Y] = E[Y|X=1]P(X=1) + E[Y|X=2]P(X=2) + E[Y|X=3]P(X=3)$$

$$E[Y|X=1] = 3$$

What is $E[Y|X=2]$?

A. $E[5] + Y$

B. $E[Y+5] = 5 + E[Y]$

C. $5 + E[Y|X=2]$

(by yourself)

Analyzing recursive code

If Y discrete
$$E[X] = E\big[E[X|Y]\big] = \sum_y E[X|Y=y]P(Y=y)$$

```python
def recurse():
  # equally likely values 1,2,3
  x = np.random.choice([1,2,3])
  if (x == 1): return 3
  elif (x == 2): return (5 + recurse())
  else: return (7 + recurse())
```

Let Y = return value of `recurse()`.
What is $E[Y]$?

$$E[Y] = E[Y|X=1]P(X=1) + E[Y|X=2]P(X=2) + E[Y|X=3]P(X=3)$$

$E[Y|X=1] = 3$

When $X = 2$, return 5 + a future return value of `recurse()`.

What is $E[Y|X=2]$?

A. $E[5] + Y$
B. $E[Y+5] = 5 + E[Y]$
C. $5 + E[Y|X=2]$

Analyzing recursive code

If Y discrete
$$E[X] = E\big[E[X|Y]\big] = \sum_y E[X|Y=y]P(Y=y)$$

```
def recurse():
  # equally likely values 1,2,3
  x = np.random.choice([1,2,3])
  if (x == 1): return 3
  elif (x == 2): return (5 + recurse())
  else: return (7 + recurse())
```

Let Y = return value of `recurse()`.
What is $E[Y]$?

$$E[Y] = E[Y|X=1]P(X=1) + E[Y|X=2]P(X=2) + E[Y|X=3]P(X=3)$$

$E[Y|X=1] = 3$

$E[Y|X=2] = E[5+Y]$

When $X = 3$, return $7 +$ a future return value of `recurse()`.

$$E[Y|X=3] = E[7+Y]$$

Analyzing recursive code

If Y discrete
$$E[X] = E\big[E[X|Y]\big] = \sum_y E[X|Y=y]P(Y=y)$$

```python
def recurse():
  # equally likely values 1,2,3
  x = np.random.choice([1,2,3])
  if (x == 1): return 3
  elif (x == 2): return (5 + recurse())
  else: return (7 + recurse())
```

Let Y = return value of `recurse()`. What is $E[Y]$?

$$E[Y] = E[Y|X=1]P(X=1) + E[Y|X=2]P(X=2) + E[Y|X=3]P(X=3)$$

$$E[Y|X=1] = 3 \qquad E[Y|X=2] = E[5+Y] \qquad E[Y|X=3] = E[7+Y]$$

$$E[Y] = 3(1/3) + (5+E[Y])(1/3) + (7+E[Y])(1/3)$$

$$E[Y] = (1/3)(15 + 2E[Y]) = 5 + (2/3)E[Y]$$

$$E[Y] = 15$$