CS 237: Probability in Computing

Wayne Snyder
Computer Science Department
Boston University

Lecture 17:

- JRVs: Conditional JRVs and Independence
- Correlation and Covariance

Next: Linear Regression

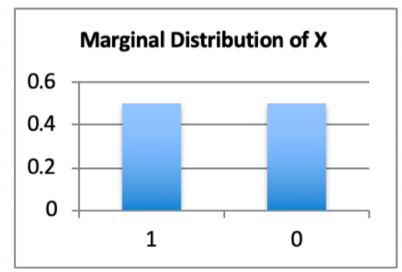
Joint Random Variables: Conditional Probability

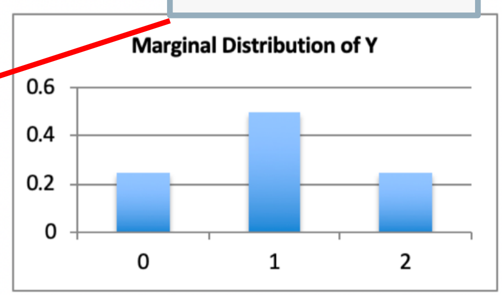
We can easily define the notion of conditional probability, using the expected definition; e.g., the probability that X returns 1, given that we know that Y >= 1, is:

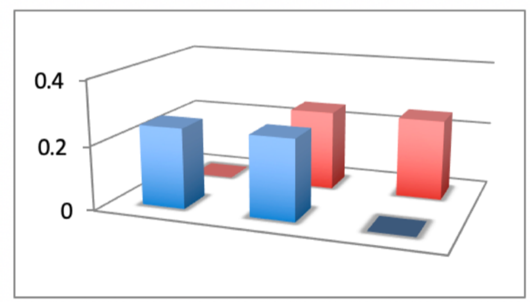
$$P(X = 1 \mid Y \geq 1) = \frac{P(X = 1 \text{ and } Y \geq 1)}{P(Y \geq 1)} = \frac{0.5}{0.75} = \frac{2}{3}$$

Example 3: Toss 2 coins; X = # heads on first coin, Y = total # of heads

X \ Y	0	1	2	
1	0	0.25	0.25	0.5
0	0.25	0.25	0	0.5
	0.25	0.5	0.25	







Conditioned by $P(Y \geq 1)$:

X' \ Y'	1	2	
	1/3	1/3	2/3
	1/3	0	1/3
	2/3	1/3	

$X \sim B(2,0.5)$ $E(X) = 1$ $Var(X) = 2*0.5*0.5 = 0.5$ $\sigma_X = 0.717$

$Y \sim Bern(0.5)$ $E(Y) = 1$ $Var(Y) = 0.5*0.5 = 0.25$ $\sigma_Y = 0.5$

Joint Random Variables: Independence

Again, we can easily define the notion of independence, using the expected definition; e.g., two random variables are independent if and only if

$$f_{X,Y}(j,k) = f_X(j) * f_Y(k)$$

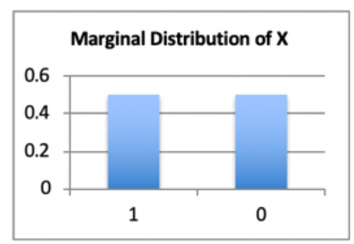
That is, each joint probability is the product of the marginal probabilities.

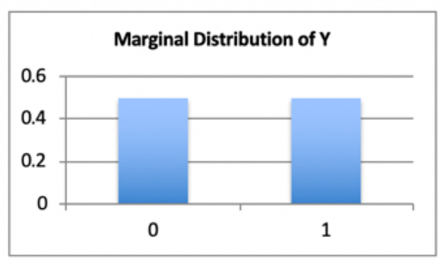
INDEPENDENT:

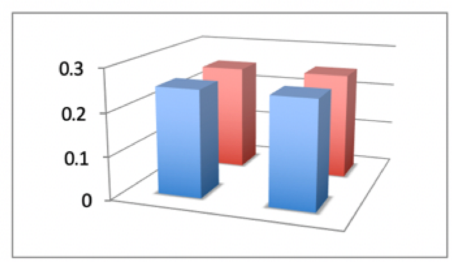
Example 1: Toss 2 coins;
X = # heads on first coin,
Y = # heads on second

<u>Sample Space</u>	<u>X</u>	<u>Y</u>
T T	0	0
T H	0	1
H T	1	0
H H	1	1

X \ Y	0	1	
1	0.25	0.25	0.5
0	0.25	0.25	0.5
	0.5	0.5	







Joint Random Variables: Independence

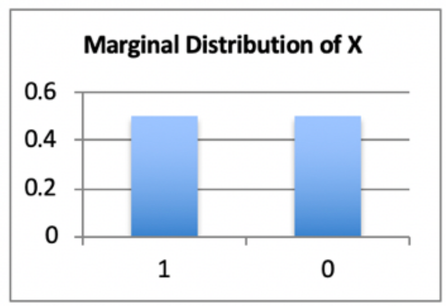
Again, we can easily define the notion of independence, using the expected definition; e.g., two random variables are independent if and only if

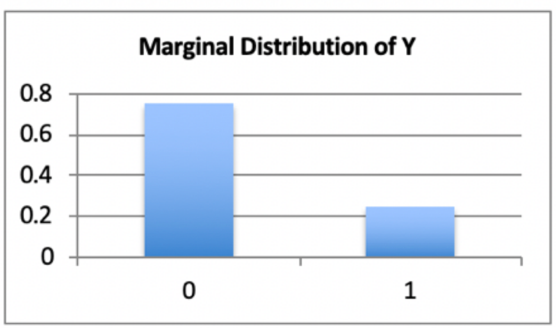
$$f_{X,Y}(j,k) = f_X(j) * f_Y(k)$$

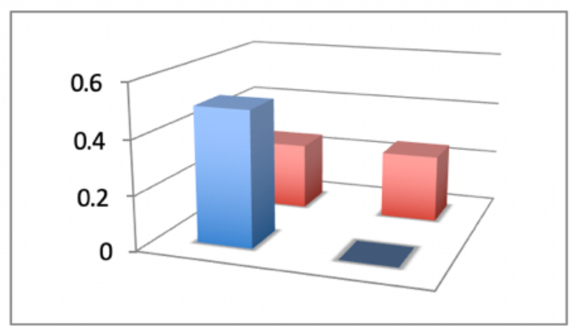
That is, each joint probability is the product of the marginal probabilities.

DEPENDENT:

		Y: 0	Y: 1	
X	1	0.25	0.25	0.5
	0	0.5	0	0.5
		0.75	0.25	







Joint Random Variables

Again, we can easily define the notion of independence, using the expected definition; e.g., two random variables are independent if and only if

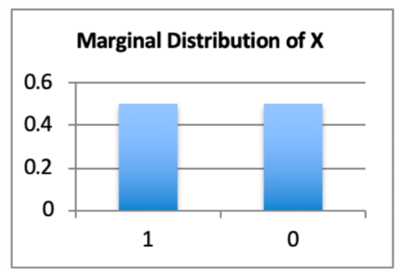
$$f_{X,Y}(j,k) = f_X(j) * f_Y(k)$$

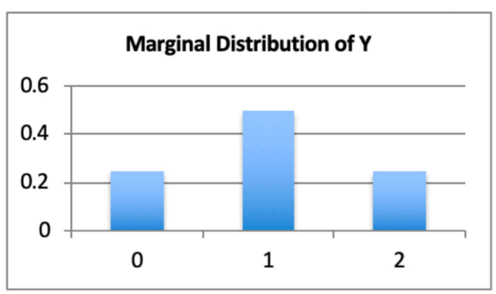
That is, each joint probability is the product of the marginal probabilities.

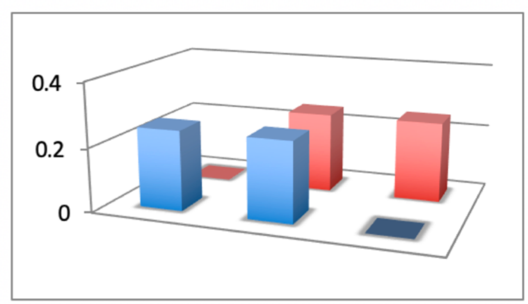
DEPENDENT

Example 3: Toss 2 coins; X = # heads on first coin, Y = total # of heads

X \ Y	0	1	2	
1	0	0.25	0.25	0.5
0	0.25	0.25	0	0.5
	0.25	0.5	0.25	







Joint Random Variables

Again, we can easily define the notion of independence, using the expected definition; e.g., two random variables are independent if and only if

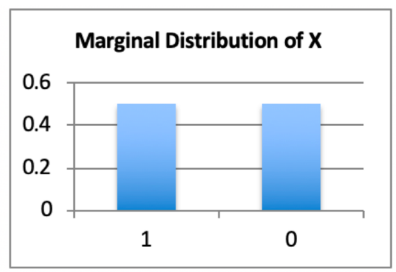
$$f_{X,Y}(j,k) = f_X(j) * f_Y(k)$$

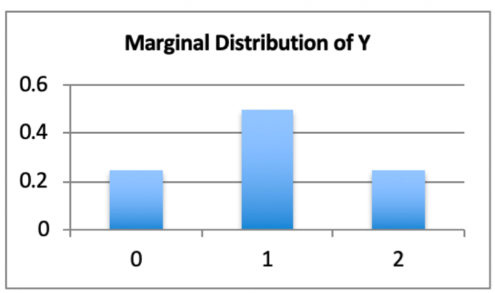
That is, each joint probability is the product of the marginal probabilities.

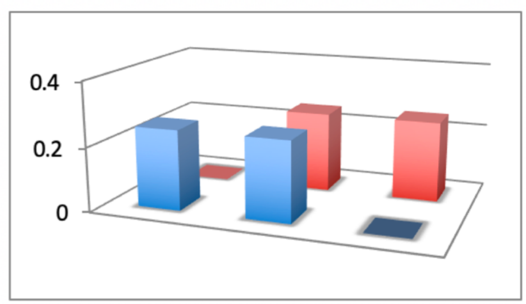
Dependent!

Example 3: Toss 2 coins; X = # heads on first coin, Y = total # of heads

		Y = 0	Y = 1	Y = 2	
X	1	0	0.25	0.25	0.5
	0	0.25	0.25	0	0.5
		0.25	0.5	0.25	







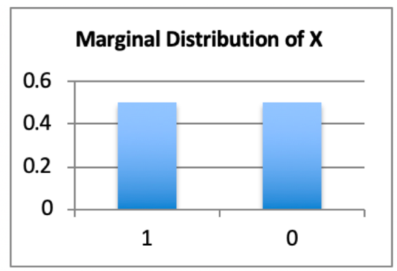
Joint Random Variables as Multivariate Points

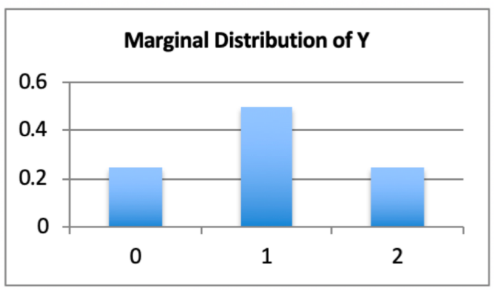
The standard way of presenting a Joint Random Variable is to specify the range of each marginal distribution and the probabilities of each tuple returned by the JRV:

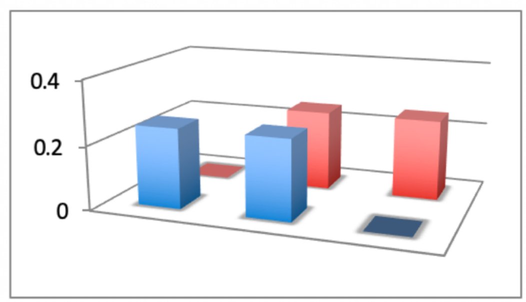
Example 3: Toss 2 coins; X = # heads on first coin, Y = total # of heads

R_Y

R_X	0	1	2	
1	0	0.25	0.25	0.5
0	0.25	0.25	0	0.5
	0.25	0.5	0.25	







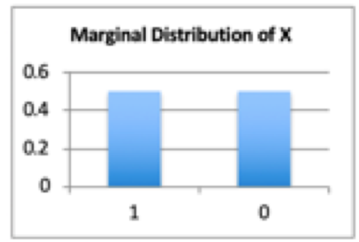
X ~ B(2,0.5) E(X) = 1 Var(X) = 2*0.5*0.5 = 0.5 $\sigma_X = 0.717$

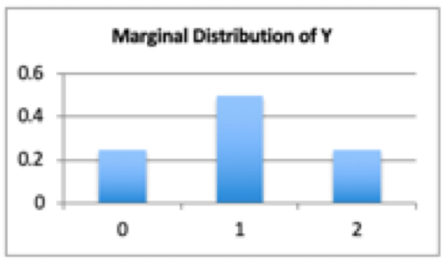
Y ~ Bern(0.5) E(Y) = 1 Var(Y) = 0.5*0.5 = 0.25 $\sigma_Y = 0.5$

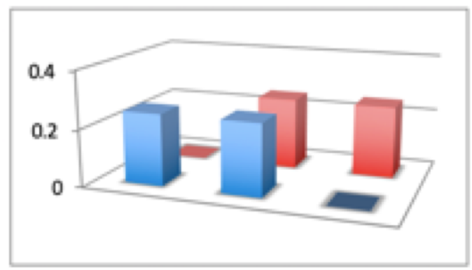
Joint Random Variables as Equiprobable Tuples

However, when not all possible tuples are possible (there are 0's in the matrix), but those that are possible are equiprobable, then it may be simpler to simply list the tuples:

		Y			
		0	1	2	
X	1	0	0.25	0.25	0.5
	0	0.25	0.25	0	0.5
		0.25	0.5	0.25	







$$R_{X,Y} = \{ (0,0), (0,1), (1,1), (1,2) \}$$

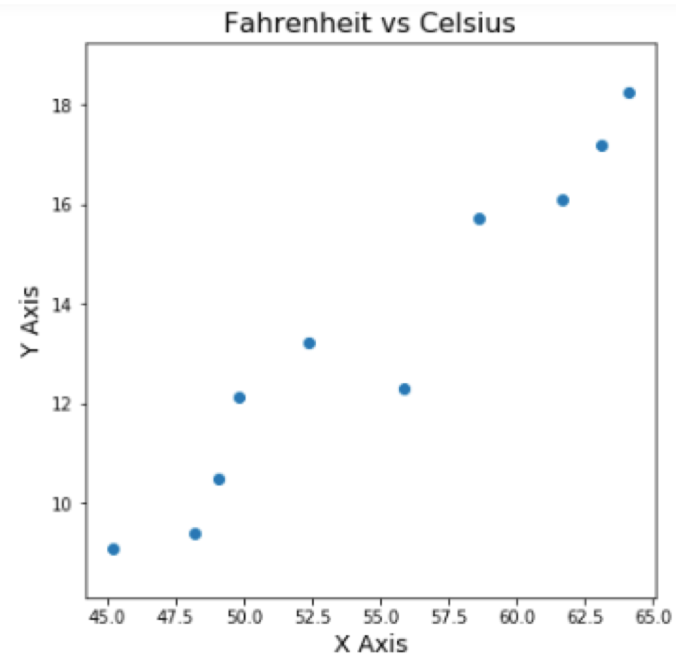
$$f_{X,Y} = \{ \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4} \}$$

Joint Random Variables as Equiprobable Tuples

This is particularly true when thinking about sampling points from a large population, such as tuples of real numbers.

For example, suppose you have two thermometers, one showing Fahrenheit, and one showing Celsius, and you test them by taking 10 samples, yielding 10 pairs of floating-point numbers, which can be shown by a scatterplot:

```
[(45.2, 9.1),
 (48.2, 9.4),
 (49.1, 10.5),
 (49.8, 12.1),
 (52.4, 13.2),
 (55.9, 12.3),
 (58.6, 15.7),
 (61.7, 16.1),
 (63.1, 17.1),
 (64.1, 18.2)  ]
```

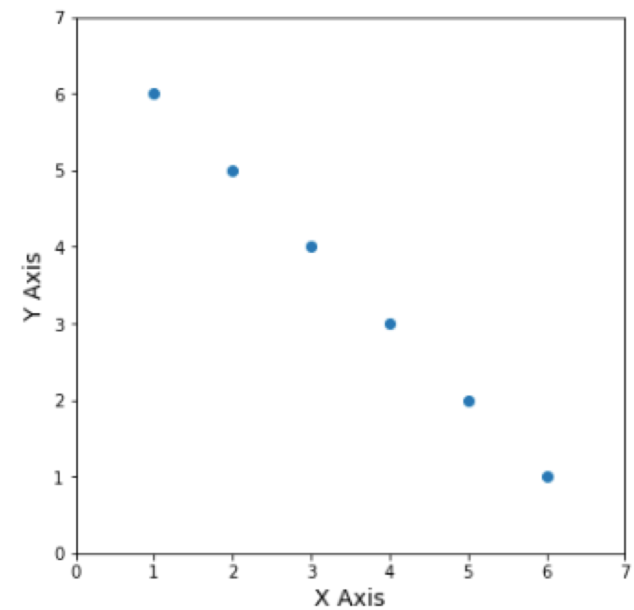


X = [45.2, 48.2, 49.1, 49.8, 52.4, 55.9, 58.6, 61.7, 63.1, 64.1]
Y = [9.1, 9.4, 10.5, 12.1, 13.2, 12.3, 15.7, 16.1, 17.1, 18.2]

Joint Random Variables as Equiprobable Tuples

When the number of points is not too large, then we can represent the sampled points as a matrix, giving all the equiprobable tuples equal probability:

Example: Roll a single die.
X = number showing on the die
Y = 7 – X



Sampling Version:

$R_{X,Y}$ = { (1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1) }

X = [1,2,3,4,5,6] Y = [6,5,4,3,2,1]

$f_{X,Y}$ ={ 1/6, 1/6, 1/6, 1/6, 1/6, 1/6 }

Matrix Version:

$R_{X,Y}$ = { (1,1), (1,2), …, (1,6),
.
.
(6,1), (6,2), …, (6,6) }

		Y					
		1	2	3	4	5	6
	1	0.1667	0	0	0	0	0
	2	0	0.1667	0	0	0	0
X	3	0	0	0.1667	0	0	0
	4	0	0	0	0.1667	0	0
	5	0	0	0	0	0.1667	0
	6	0	0	0	0	0	0.1667

Joint Random Variables as Equiprobable Tuples

The standard libraries for statistics will give you the option of specifying the PDF (as "weights") or leaving it out, in which case the assumption is that it is equiprobable:

numpy.average

numpy.average(*a*, *axis=None*, *weights=None*, *returned=False*) [source]

Compute the weighted average along the specified axis.

Parameters: a : *array_like*

Array containing data to b
attempted.

axis : *None or int or tuple of in*

Axis or axes along which t
over all of the elements o
last to the first axis.
New in version 1.7.0.
If axis is a tuple of ints, av
the tuple instead of a sing

weights : *array_like, optional*

An array of weights associated with the values in *a*. Each value in *a* contributes to the average according to its associated weight. The weights array can either be 1-D (in which case its length must be the size of *a* along the given axis) or of the same shape as *a*. If *weights=None*, then all data in *a* are assumed to have a weight equal to one.

```
import numpy as np

X = [1,2,3,4,5,6]

print( np.average(X)  )        # Default is equiprobable

print( np.average(X,weights=[0.1,0.2,0.1,0.2,0.1,0.3])  )

print( np.average(X,weights=[1/6,1/6,1/6,1/6,1/6,1/6])  )
```

```
3.5
3.9
3.5000000000000004
```

Joint Random Variables as Equiprobable Tuples

The standard libraries for statistics will give you the option of specifying the PDF (as "weights") or leaving it out, in which case the assumption is that it is equiprobable:

numpy.cov

numpy.cov(*m, y=None, rowvar=True, bias=False, ddof=None, fweights=None, aweights=None*) [source]

Estimate a covariance matrix, given data and weights.

Covariance indicates the level to which two variables vary together. If we examine N-dimensional samples, X = [x_1, x_2, ... x_N]^T, then the covariance matrix element C_{ij} is the covariance of x_i and x_j. The element C_{ii} is the variance of x_i.

See the notes for an outline of the algorithm.

fweights : *array_like, int, optional*

1-D array of integer frequency weights; the number of times each observation vector should be repeated.
New in version 1.10.

aweights : *array_like, optional*

1-D array of observation vector weights. These relative weights are typically large for observations considered "important" and smaller for observations considered less "important". If `ddof=0` the array of weights can be used to assign probabilities to observation vectors.
New in version 1.10.

`fweights` = frequency counts, as in a histogram

`aweights` = probabilities, as in a PDF

Joint Random Variables as Equiprobable Tuples

The standard libraries for statistics will give you the option of specifying the PDF (as "weights") or leaving it out, in which case the assumption is that it is equiprobable:

In [21]:

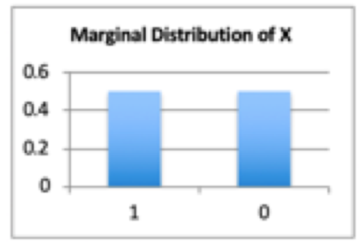
```
X = [1,2,3,4,5,6]
Y = [6,5,4,3,2,1]

XX = [ X, X ]
XY = [ X, Y ]

print(XX)
print(XY)
print()


print( np.cov(XX,bias=True)  )      # Default is equiprobable
print()

print( np.cov(XX,bias=True)[0][1]  )      # Default is equiprobable
print()

print( np.cov(XY,bias=True)[0][1]  )      # Default is equiprobable
print()
# Weights same as frequency counts for (x1,y1), ....

print( np.cov(XY,bias=True,fweights=[10, 20, 10, 20, 10, 30])[0][1]  )
print()

# Weights same as PDF = probabilities for each member of X
print( np.cov(XY,bias=True,aweights=[0.1,0.2,0.1,0.2,0.1,0.3])[0][1]  )
```

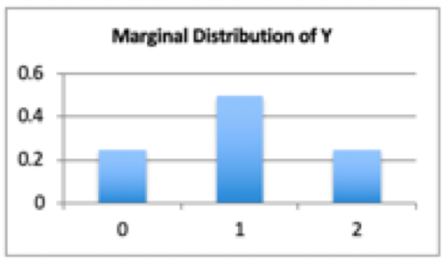
```
[[1, 2, 3, 4, 5, 6], [1, 2, 3, 4, 5, 6]]
[[1, 2, 3, 4, 5, 6], [6, 5, 4, 3, 2, 1]]

[[2.91666667 2.91666667]
 [2.91666667 2.91666667]]

2.9166666666666665

-2.9166666666666665

-3.09

-3.09
```

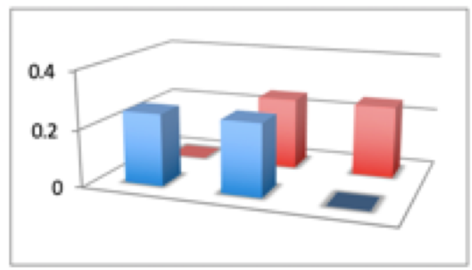
Joint Random Variables as Equiprobable Tuples

However, when not all possible tuples are possible (there are 0's in the matrix), but those that are possible are equiprobable, then it may be simpler to simply list the tuples:

	Y: 0	Y: 1	Y: 2	
X: 1	0	0.25	0.25	0.5
X: 0	0.25	0.25	0	0.5
	0.25	0.5	0.25	







$$R_{X,Y} = \{\ (0,0),\ (0,1),\ (1,1),\ (1,2)\ \}$$

$$f_{X,Y} = \{\ \tfrac{1}{4},\ \tfrac{1}{4},\ \tfrac{1}{4},\ \tfrac{1}{4}\ \}$$

Joint Random Variables: Mean and Variance?

The notion of a mean (a single number representing the ”center” of the distribution) can be extended to a

Midpoint = Mean Vector = means of the marginal distributions

This defines the “centroid” or “center of gravity” of the distribution:

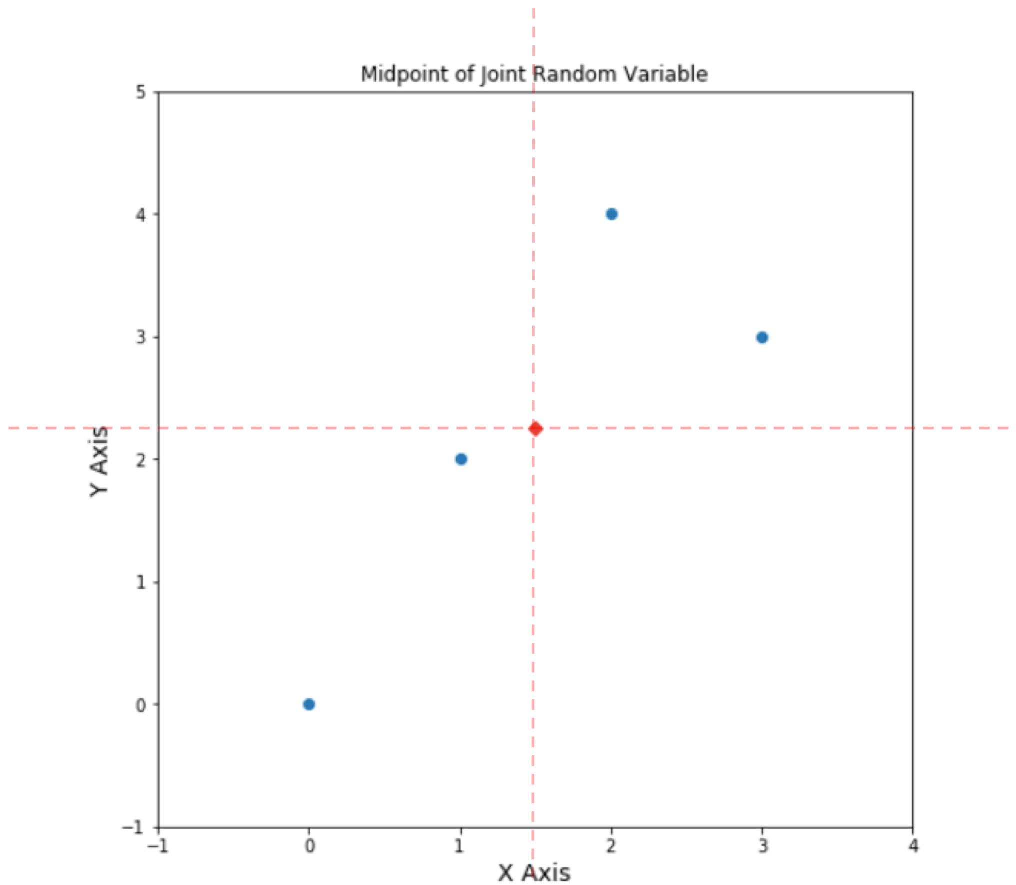
Example:

```
X = [0, 1, 2, 3]
Y = [0, 2, 4, 3]
```

```
XY = [ (0,0),
       (1,2),
       (2,4),
       (3,3)   ]
```

Midpoint = (μ_X, μ_Y)

= (3/2, 9/4)

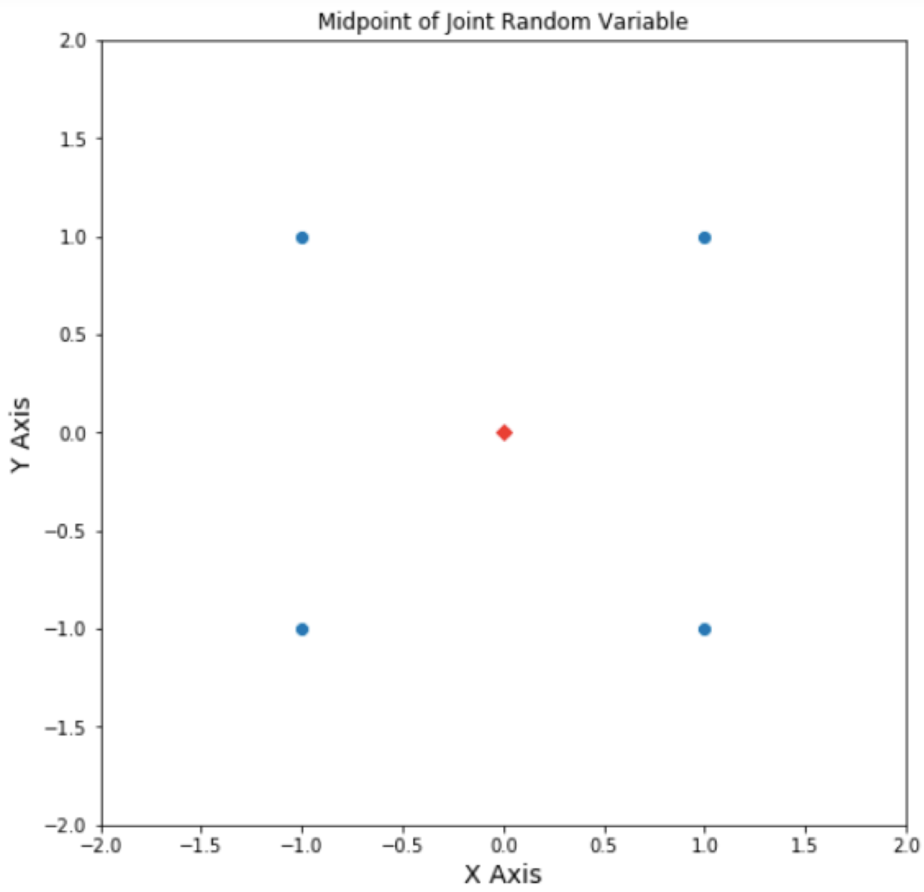


Joint Random Variables: Mean and Variance?

The notion of a mean (a single number representing the ”center” of the distribution) can be extended to a

Midpoint = Mean Vector = means of the marginal distributions

This defines the “centroid” or “center of gravity” of the distribution:



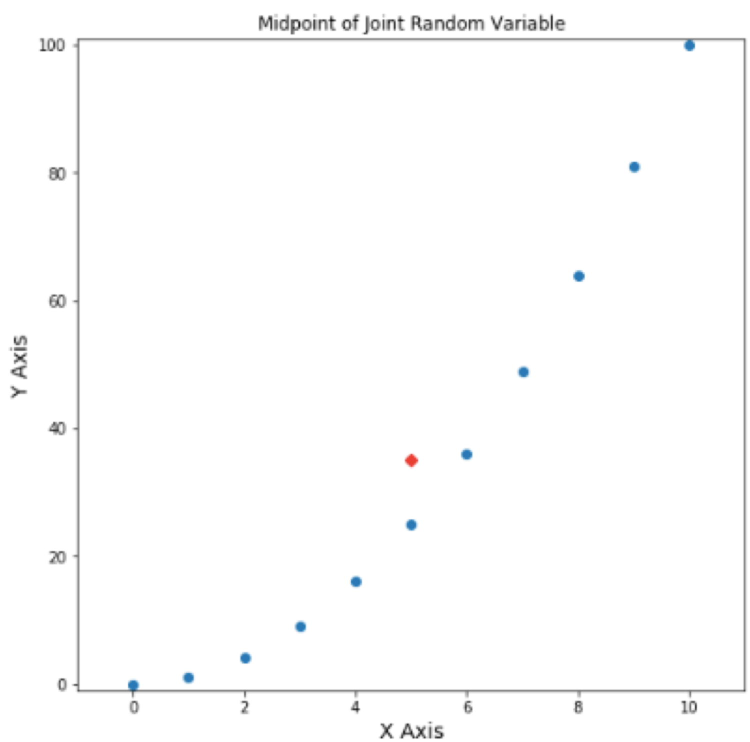
Joint Random Variables: Mean and Variance?

The notion of a mean (a single number representing the ”center” of the distribution) can be extended to a

Midpoint = Mean Vector = means of the marginal distributions

This defines the “centroid” or “center of gravity” of the distribution:

```
X = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Y = [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

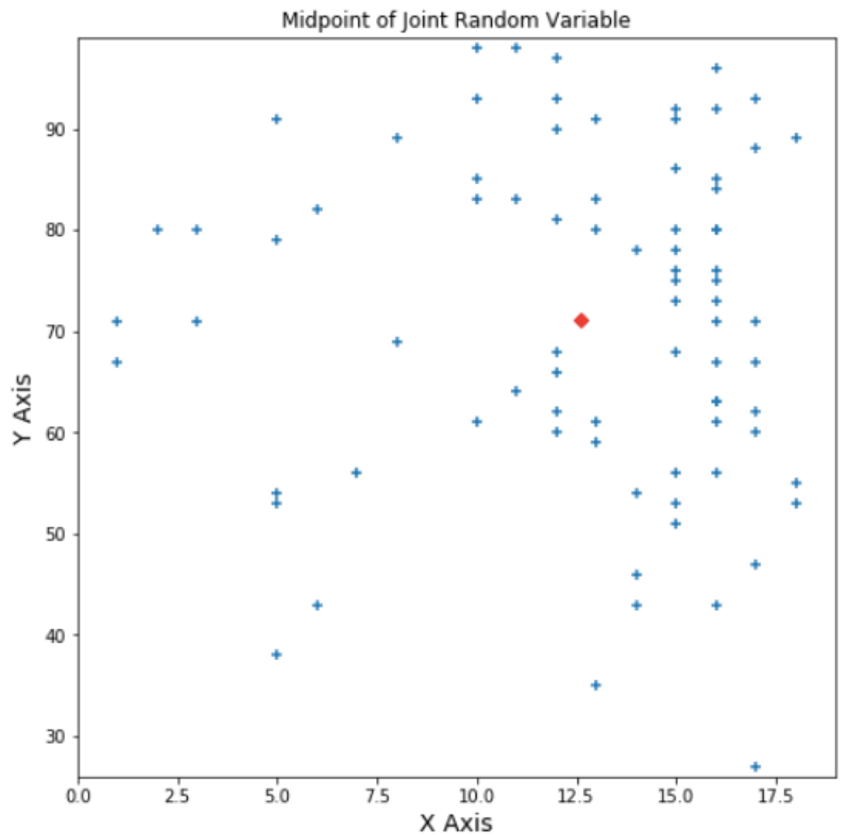


Joint Random Variables: Mean and Variance?

The notion of a mean (a single number representing the ”center” of the distribution) can be extended to a

Midpoint = Mean Vector = means of the marginal distributions

This defines the “centroid” or “center of gravity” of the distribution:

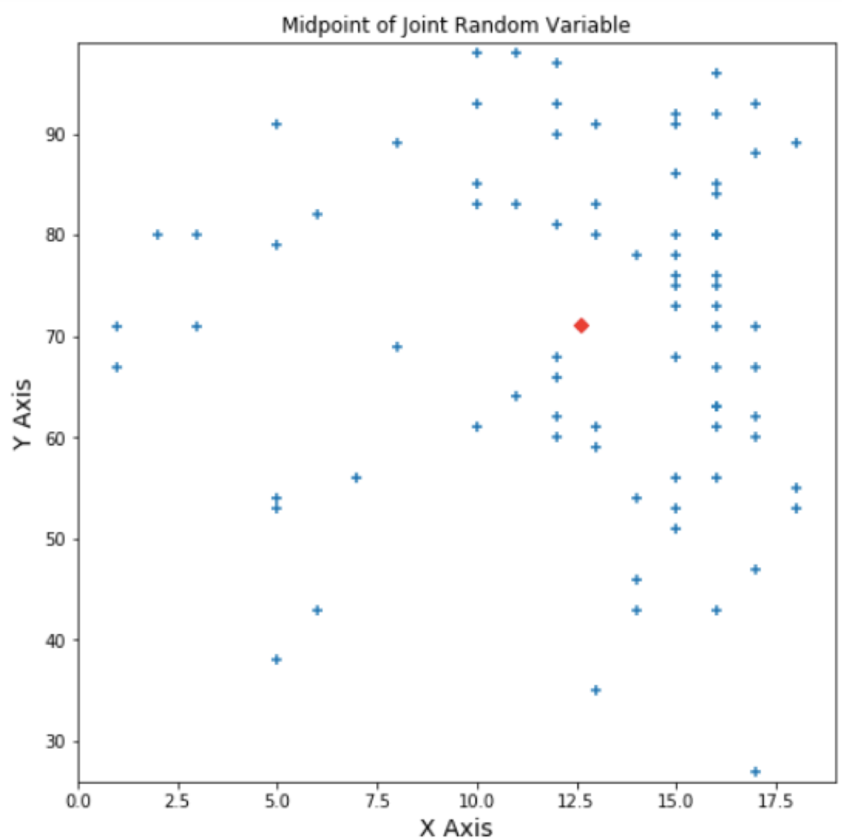


Joint Random Variables: Mean and Variance?

The notion of a mean (a single number representing the ”center” of the distribution) can be extended to a

Midpoint = Mean Vector = means of the marginal distributions

This defines the “centroid” or “center of gravity” of the distribution:

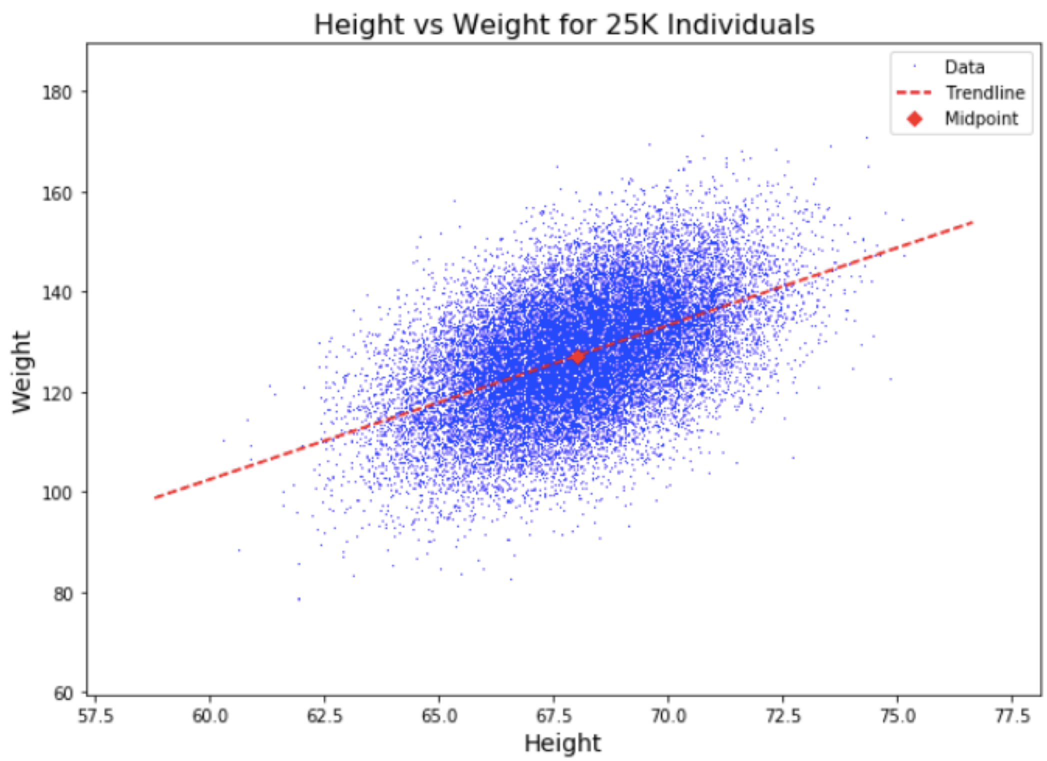


Joint Random Variables: Mean and Variance?

The notion of a mean (a single number representing the ”center” of the distribution) for a can be extended to a

Midpoint = Mean Vector = means of the marginal distributions

This defines the “centroid” or “center of gravity” of the distribution:



Joint Random Variables: Covariance

Recall: The Variance of X is defined b y:

$$Var(X) = E[\,(X - \mu_X\,)^2\,] \quad = E[\,(X - \mu_X) * (X - \mu_X)\,]$$

$$= E(X^2) - \mu_X^2 \qquad = E(X * X) - \mu_X * \mu_X$$

The Covariance of two JRVs X and Y is defined as follows:

$$Cov(X, Y) = E[\,(X - \mu_X\,) * (Y - \mu_Y\,)\,]$$

$$= E(X * Y) - \mu_X * \mu_Y$$

The Covariance of two JRVs X and Y has the same defects as the variance of a single RV:

- The units are the product of the units of X and Y: if X = height and Y = weight, then the units might be foot-pounds!
- The scale is hard to work with: What does a covariance of 123.445 foot-pounds mean?

JRVs: Covariance and Correlation Coefficient

Therefore we **standardize** the covariance so it is unit-less and in the interval [-1 .. 1].

The Correlation Coefficient of X and Y is defined as:

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X * \sigma_Y} = \frac{E[\,(X - \mu_X\,) * (Y - \mu_Y\,)}{\sigma_X * \sigma_Y} = E\left[\frac{X - \mu_X}{\sigma_X} * \frac{Y - \mu_Y}{\sigma_X}\right]$$

$$= E[\,Z_X * Z_Y\,]$$

where Z_X and Z_Y are the standardized forms of X and Y.

To compute, it is best to use:

$$\rho_{X,Y} = \frac{E(X * Y) - \mu_X * \mu_Y}{\sigma_X * \sigma_Y}$$

Joint Random Variables: Correlation Coefficient

Motivation for the Correlation Coefficient

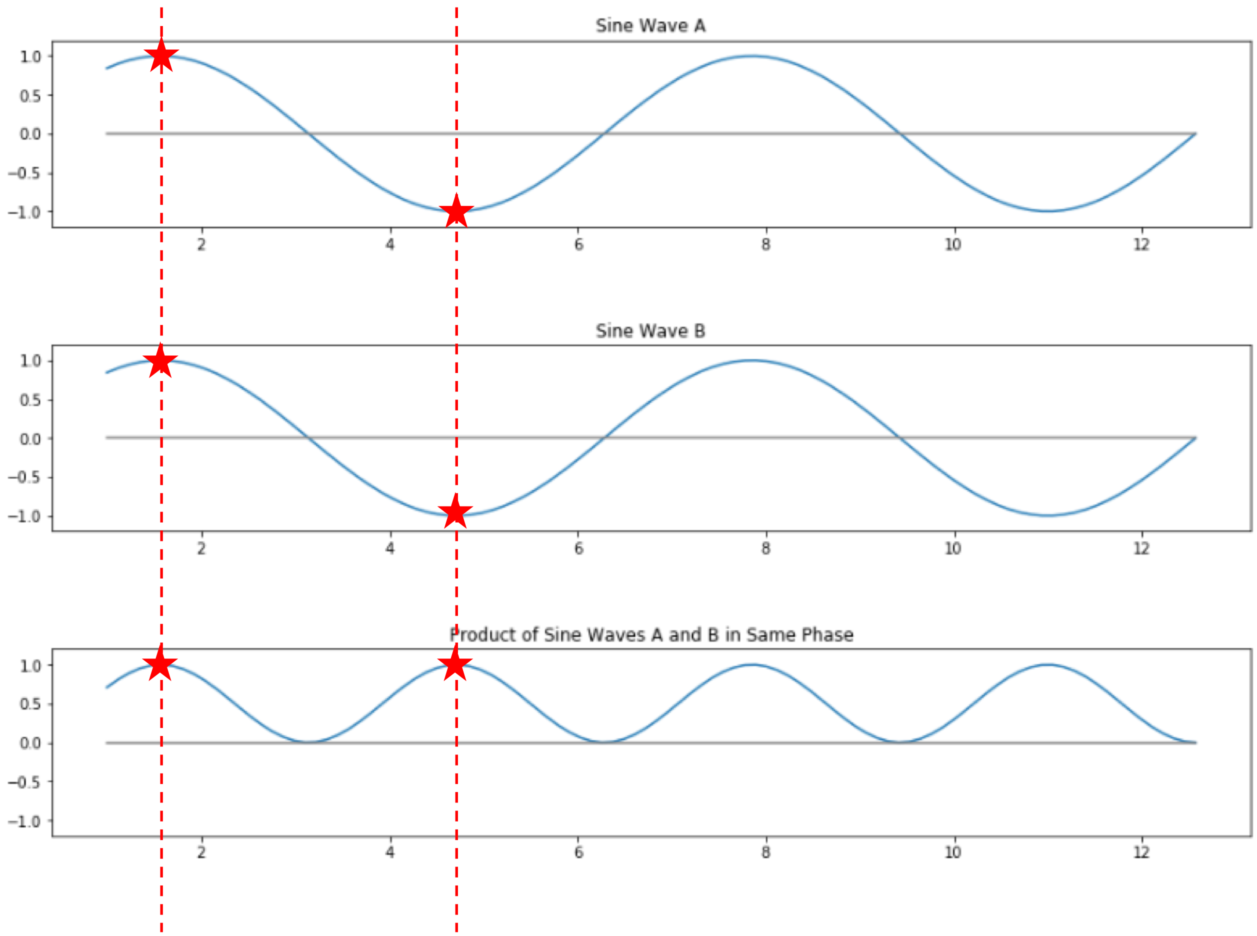
$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X * \sigma_Y} = E[\, Z_X * Z_Y \,]$$

Example: Two Sine Waves in Phase (Perfectly Correlated)

Z_X

Z_Y

$Z_X * Z_Y$



$$E[Z_X * Z_Y] > 0$$

$1 * 1 = 1$ $\quad$ $-1 * -1 = 1$

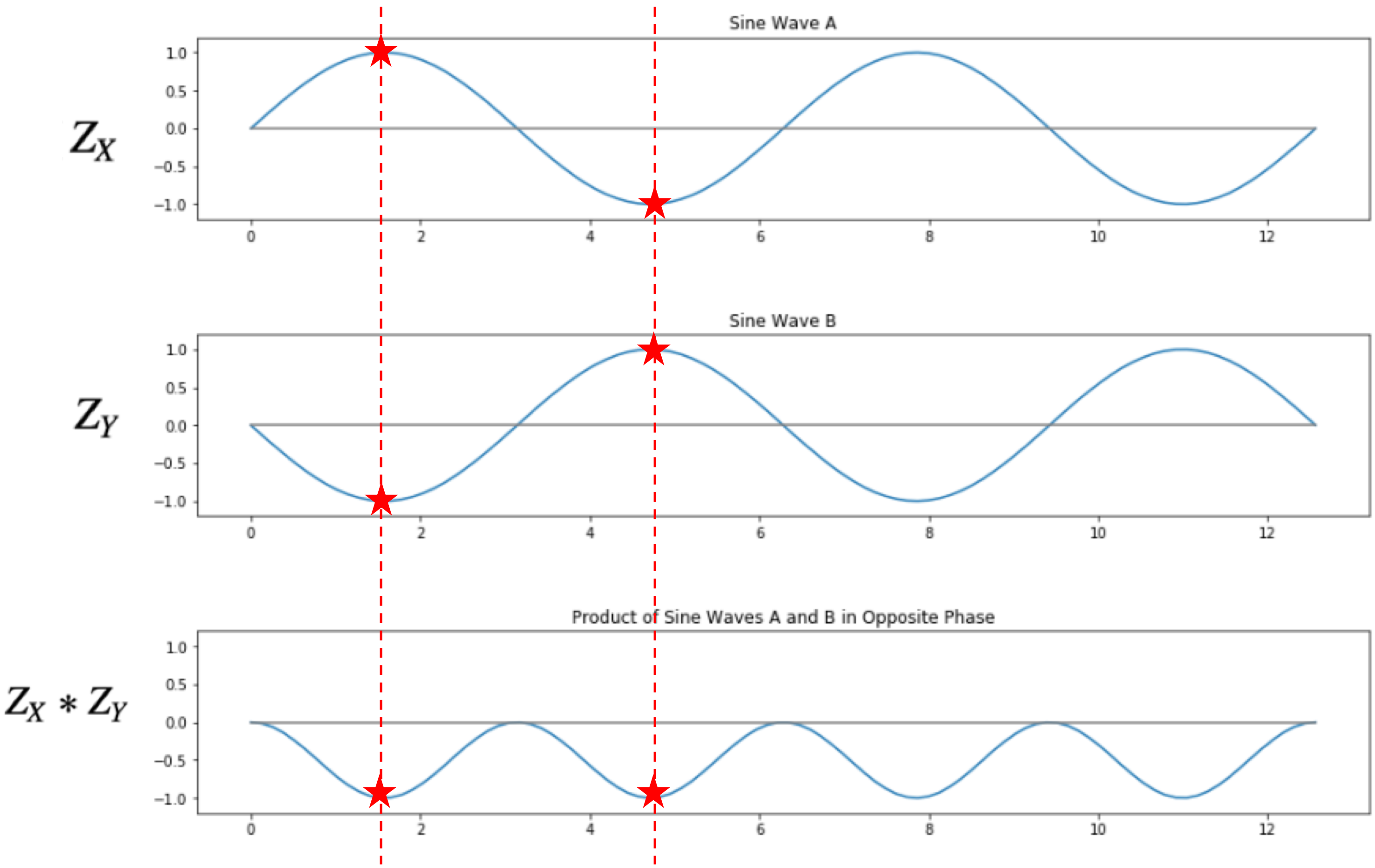
$$cos(x)^2 = \frac{cos(2x)}{2} + \frac{1}{2}$$

Joint Random Variables: Correlation Coefficient

Motivation for the Correlation Coefficient

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X * \sigma_Y} = E[\, Z_X * Z_Y \,]$$

Example: Two Sine Waves 180^{o} out of Phase (Perfectly Anti-Correlated)



$$E[Z_X * Z_Y] < 0$$

$1 * -1 \;=\; -1$

$-1 * 1 \;=\; -1$

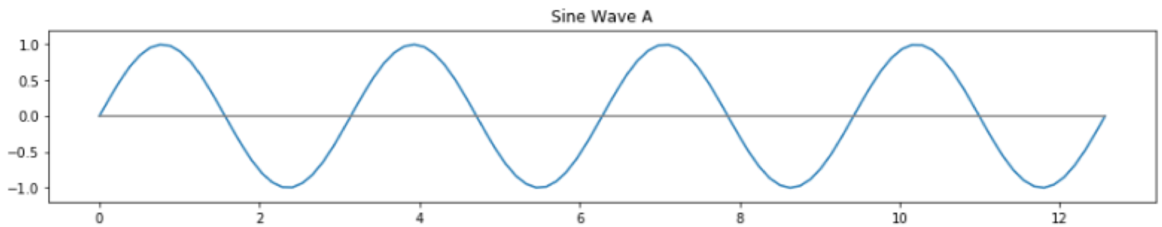
Joint Random Variables: Correlation Coefficient

Motivation for the Correlation Coefficient

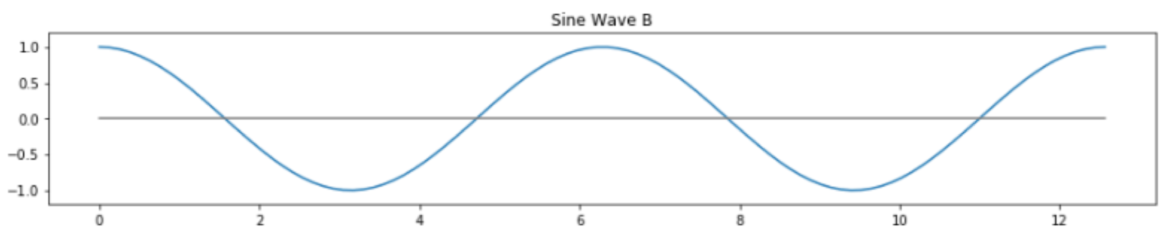
$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X * \sigma_Y} = E[\, Z_X * Z_Y \,]$$

Example: Two Random Sine Waves (No Correlation)

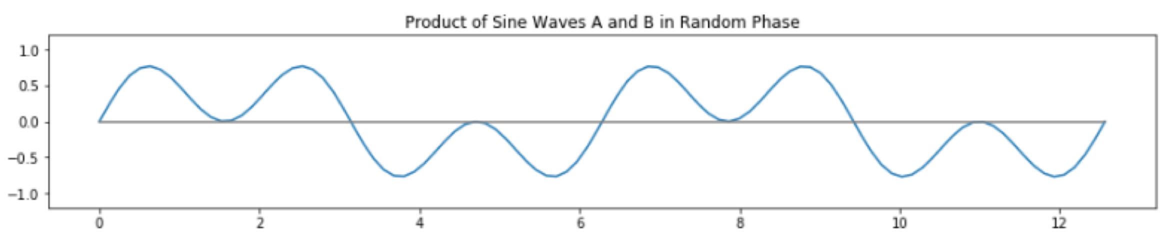
Z_X



Z_Y



$Z_X * Z_Y$

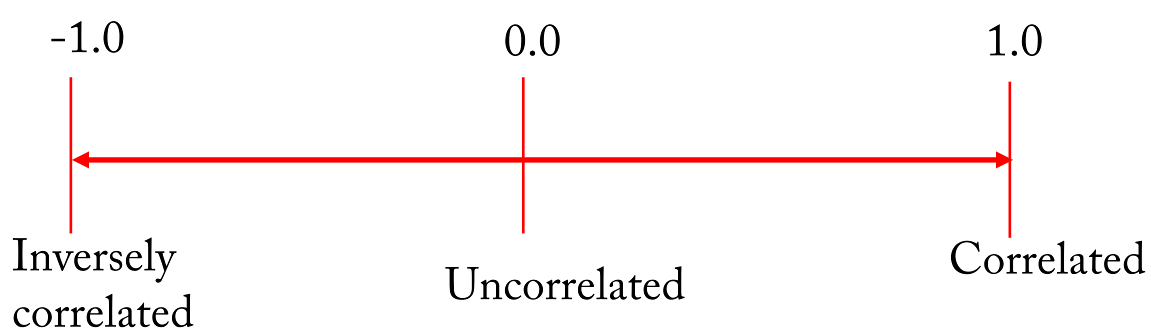


$$E[Z_X * Z_Y] = 0$$

JRVs: Covariance and Correlation Coefficient

The range of the Correlation Coefficient of X and Y is from -1 to 1:

$$\rho_{X,Y} = \frac{E(X * Y) - \mu_X * \mu_Y}{\sigma_X * \sigma_Y}$$



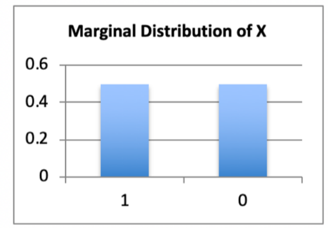
Joint Random Variables

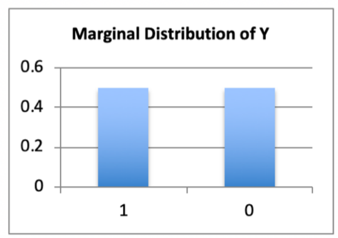
Example 1: Toss 2 coins; X = # heads on first, Y = # heads on second

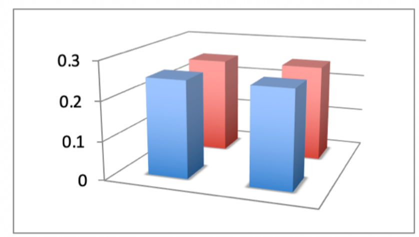
Joint Distribution for X and Y

Ex 1: toss 2 coins: X = # heads first, Y = # heads second

					Y			
							0.50	$:\sigma_Y$
				$(y_j\text{-}mu_Y)$^2 * $p(y_j)$:	0.125	0.125	0.25	:var(X)
				$(y_j\text{-}mu_Y)$:	-0.50	0.50		
			$p(x_i,y_j)$	y_j*$p(y_j)$:	0.00	0.50	0.50	$:\mu_Y$
	$(x_i\text{-}mu_X)^2 * p(x_i)$	$(x_i\text{-}mu_X)$	x_i*$p(x_i)$		**0**	**1**	$p(x_i)$	
X	0.125	0.50	0.50	**1**	0.25	0.25	0.5	
	0.125	-0.50	0.00	**0**	0.25	0.25	0.5	
var(X):	0.25	μ_X:	0.50	$p(y_j)$:	0.5	0.5	1	
σ_X:	0.50						0.25	$:\sigma_X * \sigma_Y$







$$X * Y$$

	0	1
1	0 / 0.25	1 / 0.25
0	0 / 0.25	0 / 0.25

cov(X,Y):	0.00
ρ(X,Y):	0.00

$(x_i\text{-}mu_X)$*$(y_j\text{-}mu_Y)$*$p(x_i,y_j)$:

-0.0625	0.0625
0.0625	-0.0625

$$E(X * Y) = 0 * 0.25 + 1 * 0.25 + 0 * 0.25 + 0 * 0.25 = 0.25$$

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X * \sigma_Y} = \frac{E(X * Y) - \mu_X * \mu_Y}{\sigma_X * \sigma_Y} = \frac{0.25 - 0.5 * 0.5}{0.5 * 0.5} = 0.0$$

Joint Random Variables

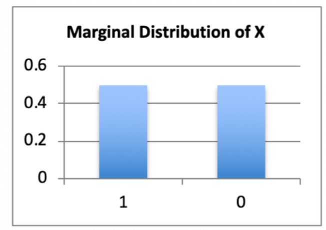
Calculating the Covariance and Correlation Coefficient is best done with either a spreadsheet or Python:

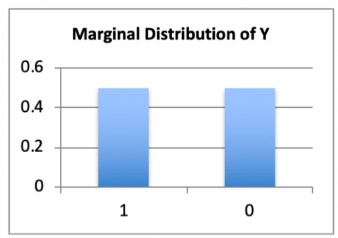
Example: Toss 1 coin; X = # heads, Y = # heads

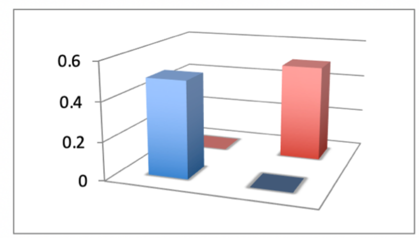
Joint Distribution for X and Y

Ex 2: toss 1 coins: X = # heads, Y = # heads

					Y			
							0.50	$:\sigma_Y$
				$(y_j\text{-mu}_Y)$^2 * $p(y_j)$:	0.125	0.125	0.25	:var(X)
				$(y_j\text{-mu}_Y)$:	-0.50	0.50		
			p(x_i,y_j)	y_j*p(y_j):	0.00	0.50	0.50	$:\mu_Y$
	$(x_i\text{-mu}_X)^2$ * p(x_i)	$(x_i\text{-mu}_X)$	x_i*p(x_i)		**0**	**1**	p(x_i)	
X	0.125	0.50	0.50	**1**	0	0.5	0.5	
	0.125	-0.50	0.00	**0**	0.5	0	0.5	
var(X):	0.25	μ_X:	0.50	p(y_j):	0.5	0.5	1	
σ_X:	0.50						0.25	$:\sigma_X * \sigma_Y$







$X * Y$

	0	1
1	0 / 0.0	1 / 0.5
0	0 / 0.5	0 / 0.0

cov(X,Y):	0.25
ρ(X,Y):	1.00

$(x_i\text{-mu}_X)*(y_j\text{-mu}_Y)*p(x_i,y_j)$:

0.0000	0.1250
0.1250	0.0000

$$E(X * Y) = 0 * 0.0 + 1 * 0.5 + 0 * 0.5 + 0 * 0.0 = 0.5$$

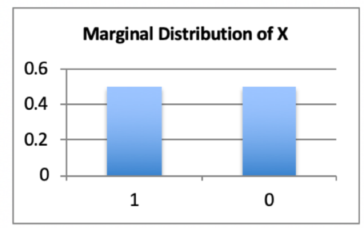
$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X * \sigma_Y} = \frac{E(X * Y) - \mu_X * \mu_Y}{\sigma_X * \sigma_Y} = \frac{0.5 - 0.5 * 0.5}{0.5 * 0.5} = \frac{0.25}{0.25} = 1.0$$

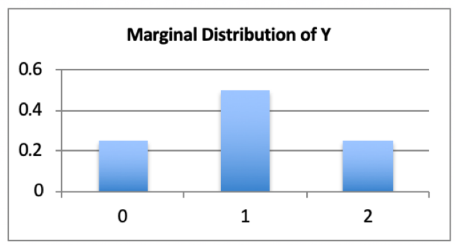
Joint Random Variables

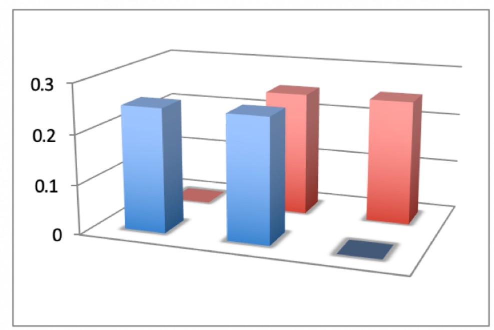
Calculating the Covariance and Correlation Coefficient is best done with either a spreadsheet or Python:

Example: Toss 2 coins; X = # heads on first coin, Y = total # of heads

					Y				
								0.71	:σ_Y
				$(y_j\text{-mu}_Y)$^2 * $p(y_j)$:	0.250	0.000	0.250	0.50	:var(X)
				$(y_j\text{-mu}_Y)$:	-1.00	0.00	1.00		
			$p(x_i,y_j)$	y_j*$p(y_j)$:	0.00	0.50	0.50	1.00	:μ_Y
	$(x_i\text{-mu}_X)^2$ * $p(x_i)$	$(x_i\text{-mu}_X)$	x_i*$p(x_i)$		**0**	**1**	**2**	$p(x_i)$	
X	0.125	0.50	0.50	**1**	0	0.25	0.25	0.5	
	0.125	-0.50	0.00	**0**	0.25	0.25	0	0.5	
var(X):	0.25	μ_X:	0.50	$p(y_j)$:	0.25	0.5	0.25	1	
σ_X:	0.50							0.35	:σ_X * σ_Y







$X * Y$

cov(X,Y):	0.25
ρ(X,Y):	0.707

$(x_i\text{-mu}_X)$*$(y_j\text{-mu}_Y)$*$p(x_i,y_j)$:

0.0000	0.0000	0.1250
0.1250	0.0000	0.0000

	0	1	2
1	0 / 0.0	1 / 0.25	2 / 0.25
0	0 / 0.25	0 / 0.25	0 / 0.0

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X * \sigma_Y} = \frac{E(X*Y) - \mu_X * \mu_Y}{\sigma_X * \sigma_Y} = \frac{0.75 - 0.5 * 1.0}{0.5 * 0.707} = \frac{0.25}{0.354} = 0.707$$

Joint Random Variables

Calculating the Covariance and Correlation Coefficient is best done with either a spreadsheet or Python:

Example: Toss 2 coins; X = # heads on first coin, Y = total # of heads

						Y		0.71	$:\sigma_Y$
				$(y_j\text{-}mu_Y)$^2 * $p(y_j)$:	0.250	0.000	0.250	0.50	:var(X)
				$(y_j\text{-}mu_Y)$:	-1.00	0.00	1.00		
			$p(x_i,y_j)$	y_j*$p(y_j)$:	0.00	0.50	0.50	1.00	$:\mu_Y$
	$(x_i\text{-}mu_X)^2$ * $p(x_i)$	$(x_i\text{-}mu_X)$	x_i*$p(x_i)$		**0**	**1**	**2**	$p(x_i)$	
X	0.125	0.50	0.50	**1**	0	0.25	0.25	0.5	
	0.125	-0.50	0.00	**0**	0.25	0.25	0	0.5	
var(X):	0.25	μ_X:	0.50	$p(y_j)$:	0.25	0.5	0.25	1	
σ_X:	0.50							0.35	$:\sigma_X * \sigma_Y$

H H => (1,2)
H T => (1,1)
T H => (0,1)
T T => (0,0)

X = [1,1,0,0]
Y = [2,1,1,0]

In [3]:

```python
from scipy.stats import pearsonr

# Calculation the correlation coefficient rho(X,Y)

def rho(X,Y):
    return pearsonr(X,Y)[0]

X = [1,1,0,0]
Y = [2,1,1,0]

rho(X,Y)
```

Out[3]: 0.7071067811865475

Scatterplots of Joint Random Variables

We often can see the relationship between a bivariate RV if we create a scatterplot of the data viewed as points:

Example: Toss 2 coins; X = # heads on first coin, Y = total # of heads

H H => (1,2)
H T => (1,1)
T H => (0,1)
T T => (0,0)

X = [1,1,0,0]
Y = [2,1,1,0]