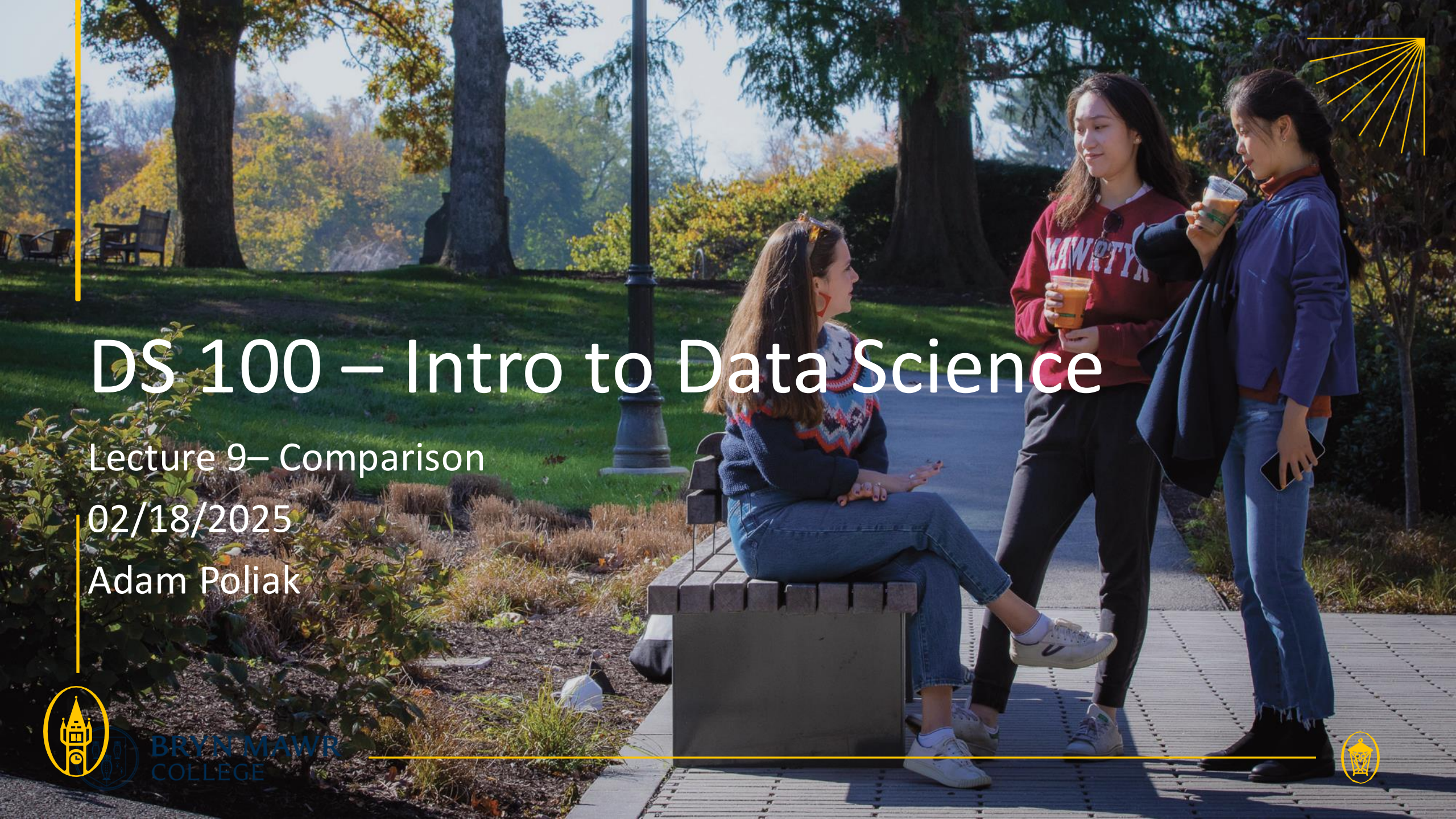


DS 100 – Intro to Data Science

Lecture 9– Comparison

02/18/2025

Adam Poliak

BRYN MAWR COLLEGE

Announcements

Lab04 (Simulations) due Friday

HW03 – Functions, Histograms, and Groups:

- Due Wednesday (02/19)

HW04 – Applying Functions & Iteration

- Due Wednesday (02/26)
- Short assignment

Checkpoint/Project 1:

- Paired assignment that covers the previous section of the course material
- Due Friday 02/28

How to ask for help

Explain what you are trying to do

Give a minimal example

- Someone else should be able to easily replicate the problem
- Should require any information that only you have

Explain what you **think** should happen

Explain what you get instead (copy/paste or screenshot if possible)

Explain what else you've tried

Taken from Jordan Boyd Graber

Table Review

t.select(column, …) or t.drop(column, …)
t.take([row, …]) or t.exclude([row, …])
t.sort(column, descending=False)
t.where(column, are.condition(…))
t.apply(function, column, …)
t.group(column) or t.group(column, function)
t.group([column, …]) or t.group([column, …], function)
t.pivot(cols, rows) or t.pivot(cols, rows, vals, function)
t.join(column, other_table, other_table_column)

https://bmc-ds-100.github.io/python-reference.html

Group vs Pivot

pivot groups together rows that share a combination of values.

It differs from *group* because it organizes the resulting values in a grid

Group vs Pivot

Pivot

One combo of grouping variables

per entry

```
cones.pivot('Flavor', 'Color', values='Price', collect=sum)
```

Color	bubblegum	chocolate	strawberry
dark brown	0	10.5	0
light brown	0	4.75	0
pink	4.75	0	8.8

Group

One combo of grouping variables

per row

```
cones.group(['Flavor','Color'],sum)
```

Flavor	Color	Price sum
bubblegum	pink	4.75
chocolate	dark brown	10.5
chocolate	light brown	4.75
strawberry	pink	8.8

Group vs Pivot

Pivot

- One combo of grouping variables **per entry**
- **Two** grouping variables: columns and rows
- Aggregate values of **values column**
- Missing combos **= 0 (or empty string)**

Group

- One combo of grouping variables **per row**
- **Any number** of grouping variables
- Aggregate values of **all other columns** in table
- Missing combos **absent**

Comparisons

Comparison Operators

Operator	Table predicate
==	are.equal_to
!=	are.not_equal_to
>	are.above
>=	are.above_or_equal_to
<	are.below
<=	are.below_or_equal_to

The result of a comparison expression is a bool value:

True, False

Comparison Operators

The result of a comparison expression is a bool value

$x = 2$ $y = 3$

Comparison Operators

The result of a comparison expression is a bool value

Comparison Operators

The result of a comparison expression is a bool value

```
x = 2                    y = 3
```

Assignment Statements

```
x > 1                    x > y                    y >= 3

x == y                   x != 2                   2 < x < 5
```

Comparison Operators

The result of a comparison expression is a bool value

```
x = 2                    y = 3
```

Assignment Statements

```
x > 1                    x > y                    y >= 3

x == y                   x != 2                   2 < x < 5
```

Comparison Expressions

Combining Comparisons

The result of a comparison expression is a bool value

a = True

b = False

not b

a or b

a and not b

a and b

not (a or b)

b and b

Combining Comparisons

The result of a comparison expression is a bool value

```
a = True        b = False
```

Assignment Statements

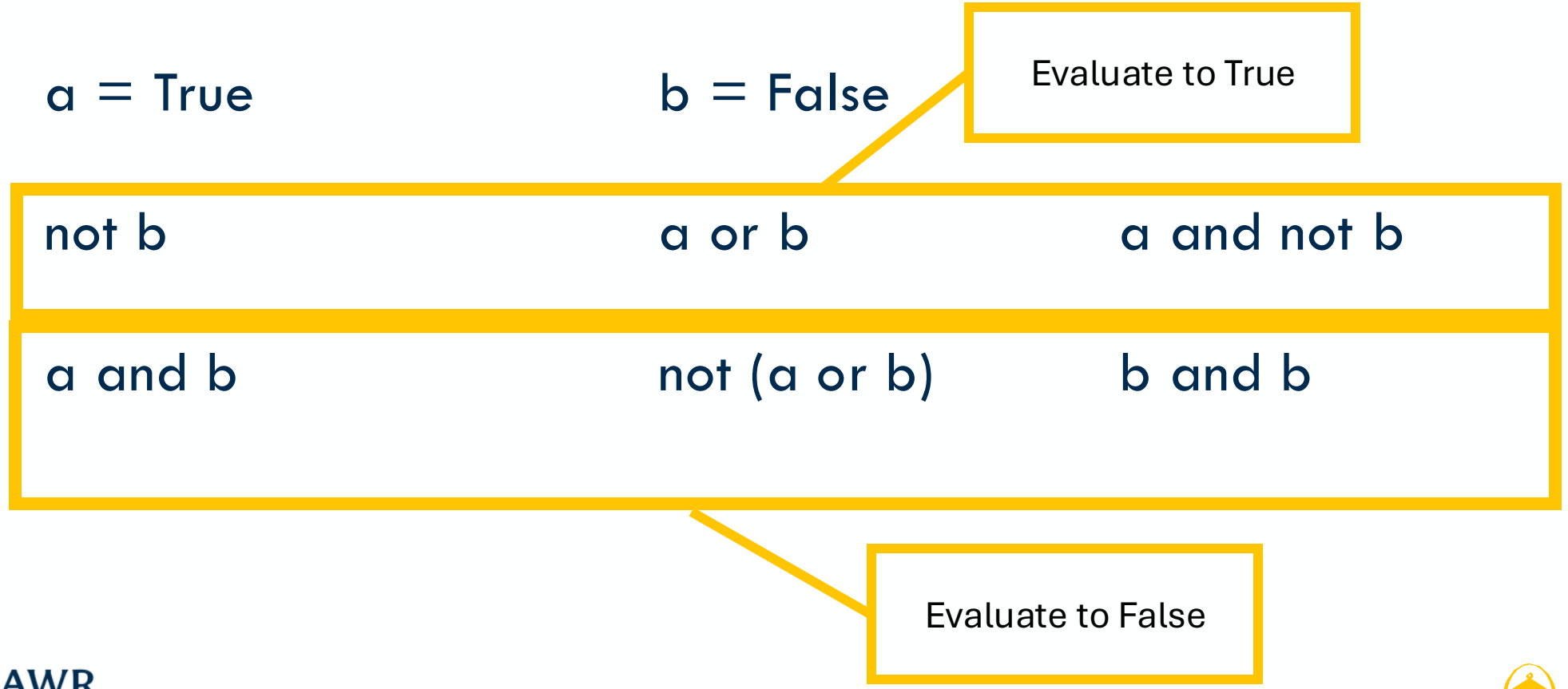
```
not b           a or b          a and not b

a and b         not (a or b)    b and b
```

Comparison Expressions

Combining Comparisons

The result of a comparison expression is a bool value



Aggregating Comparisons

Summing an array or list of bool values count the number of True values

```
1 + 0 + 1
True + False + True
sum([1     , 0     , 1])
sum([True, False, True)]
```

Control Statements

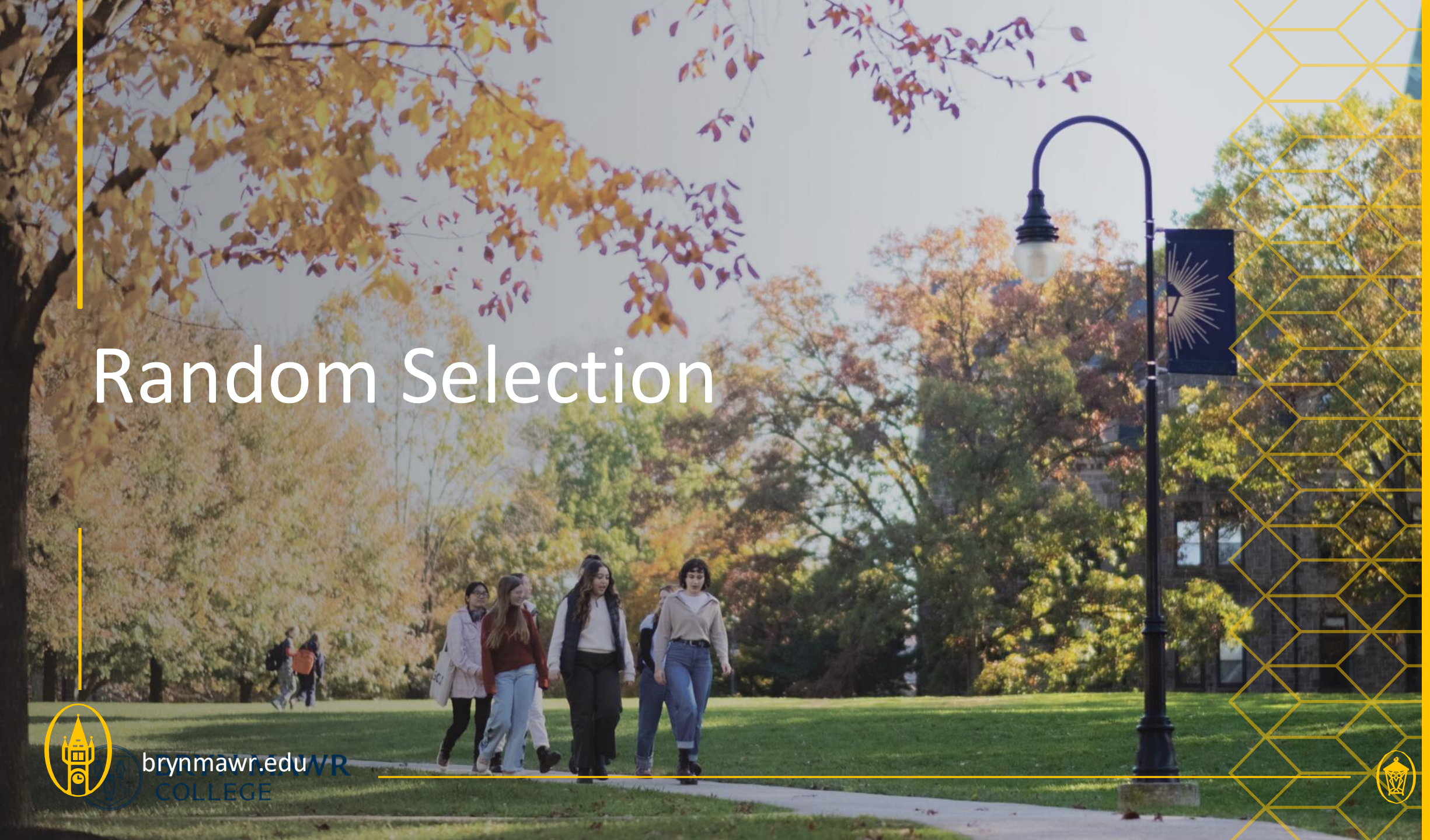
Control Statements

These statements *control* the sequence of computations that are performed

The keywords if and for begin control statements

The purpose of if is to define functions that choose different behavior based on their arguments

Random Selection



brynmawr.edu

BRYN MAWR COLLEGE

Random Selection

np.random.choice

- Selects at random
- With replacement
- From an array
- A specific number of times

np.random.choice(some_array, sample_size)

Appending Arrays

A longer array

`np.append(array_1, value)`:

- new array with value appended to array_1
- value has to be of the same type as elements of array_1

`np.append(array_1, array_2)`:

- new array with array_2 appended to array_1
- Elements of array_2 have to be of the same type as elements of array_1

Iteration

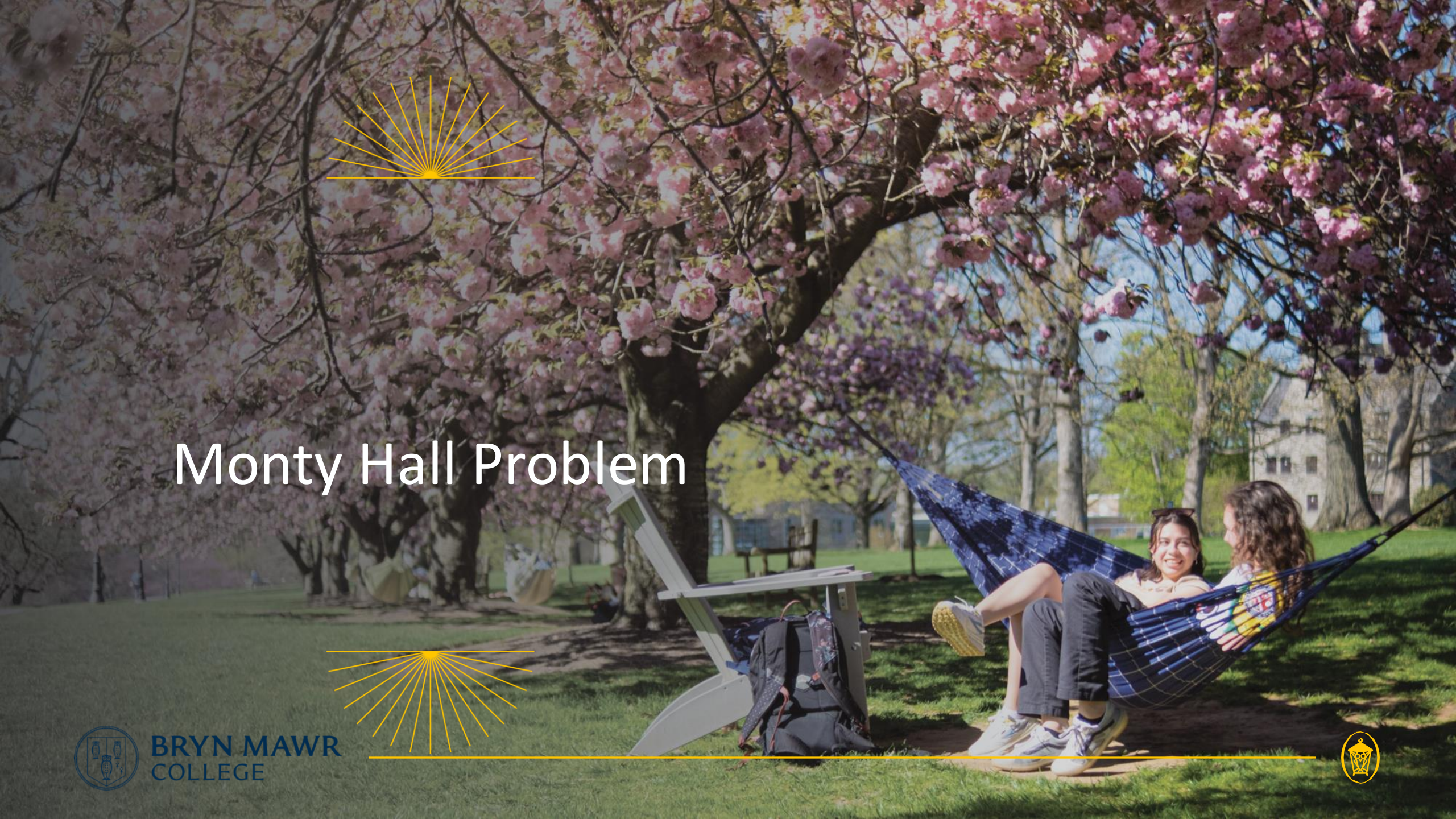
for statements

for is a keyword that begins a control statement

The purpose of for is to perform a computation for every element in a list or array

for *name* **in** *sequence:*

    BODY where we use the value in the name

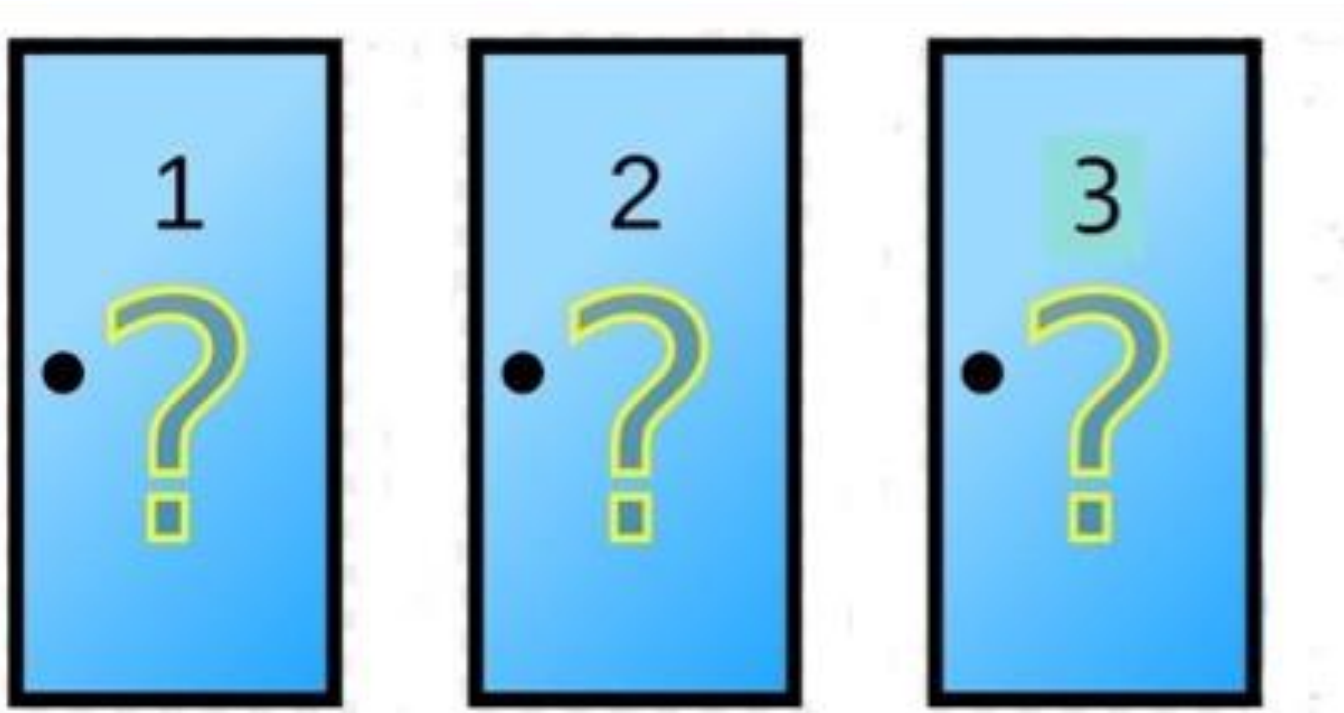


Monty Hall Problem

BRYN MAWR
COLLEGE


LET'S
MAKE
A
DEAL

Monty Hall Problem



Monty Hall Problem