

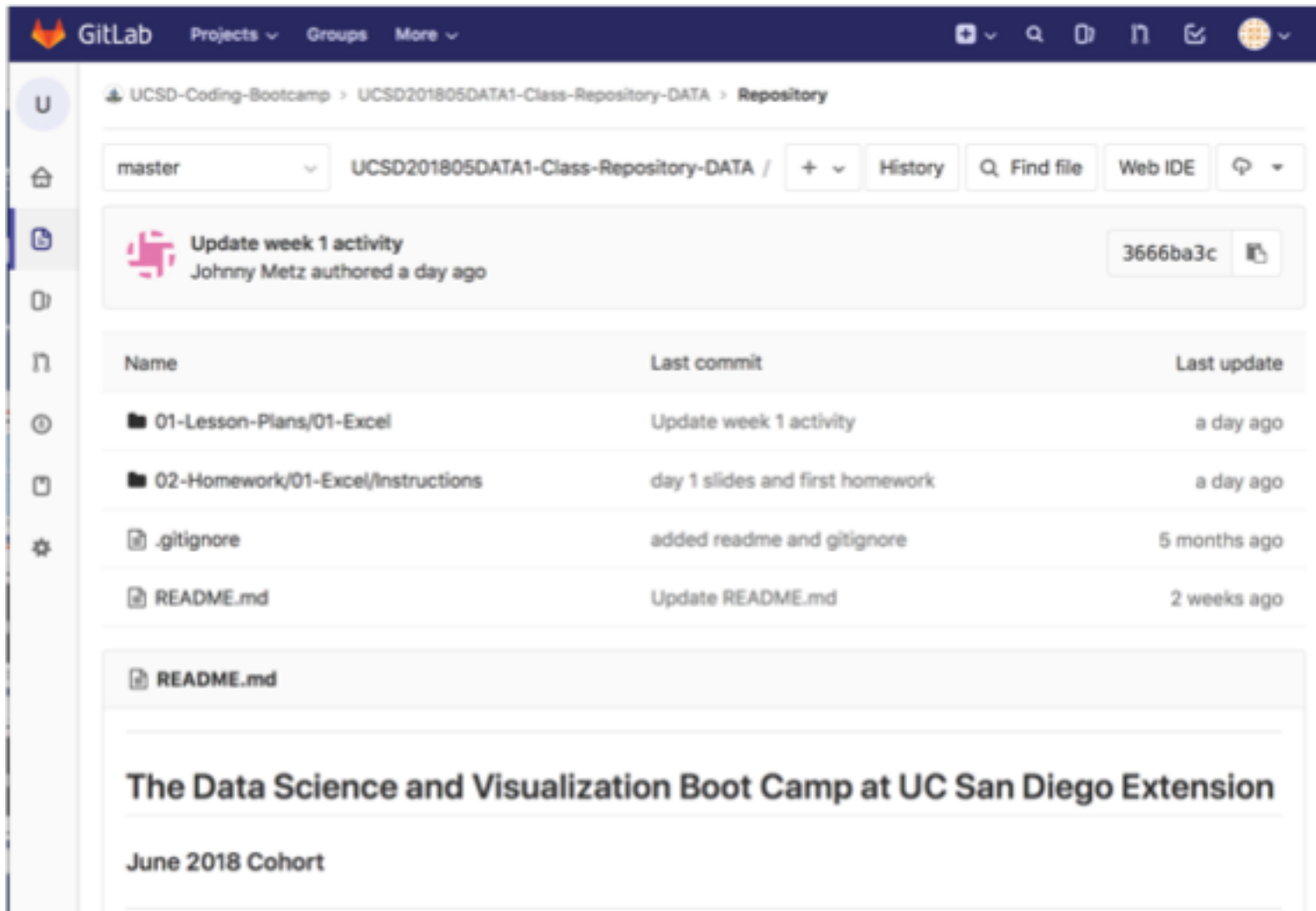
Day 2

Egad! It's Excel

The Data Bootcamp

Admin Stuff

Class Git Repository



The screenshot shows the GitLab interface for a repository named 'UCSD201805DATA1-Class-Repository-DATA'. The breadcrumb path is 'UCSD-Coding-Bootcamp > UCSD201805DATA1-Class-Repository-DATA > Repository'. The current branch is 'master'. A recent commit titled 'Update week 1 activity' by Johnny Metz is shown, with a commit hash of 3666ba3c. Below this is a table of repository contents:

Name	Last commit	Last update
01-Lesson-Plans/01-Excel	Update week 1 activity	a day ago
02-Homework/01-Excel/Instructions	day 1 slides and first homework	a day ago
.gitignore	added readme and gitignore	5 months ago
README.md	Update README.md	2 weeks ago

Below the table, the 'README.md' file is selected, showing the title 'The Data Science and Visualization Boot Camp at UC San Diego Extension' and the subtitle 'June 2018 Cohort'.

All Class Content and Homework will be here:

<https://ucsd.bootcampcontent.com/UCSD-Coding-Bootcamp/UCSD201805DATA1-Class-Repository-DATA>

Homework Assignment #1

The image shows the Kickstarter logo, which consists of the word "KICK" in white and "STARTER" in green, both in a bold, rounded, sans-serif font, set against a dark gray background.

You will be analyzing thousands of Kickstarter projects to look for funding trends across goal targets and topics.

Homework Assignment #1

The image shows the Kickstarter logo, which consists of the word "KICK" in white and "STARTER" in green, both in a bold, rounded, sans-serif font, set against a dark gray background.

Due: Next Saturday (30th)

Recommended Target: Thursday of Next Week

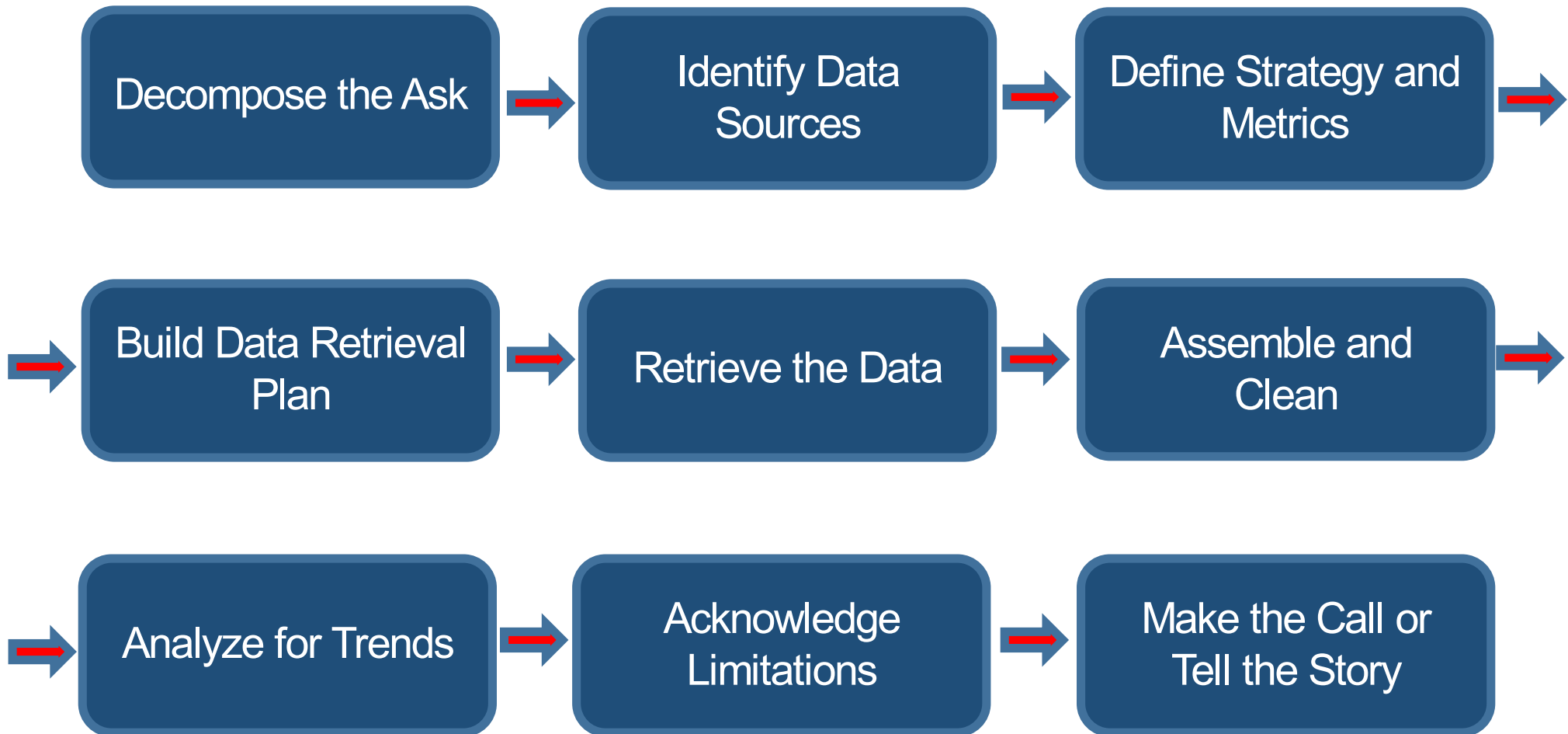
Quick Refresher

Data Science is about what **two** things?

Truth-Telling & Story Telling

What are the steps in the **Analytics Paradigm?**

Analytics Paradigm



Regardless of type or industry, this paradigm provides a repeatable pathway for effective data problem solving.

Let's Start with the Basics

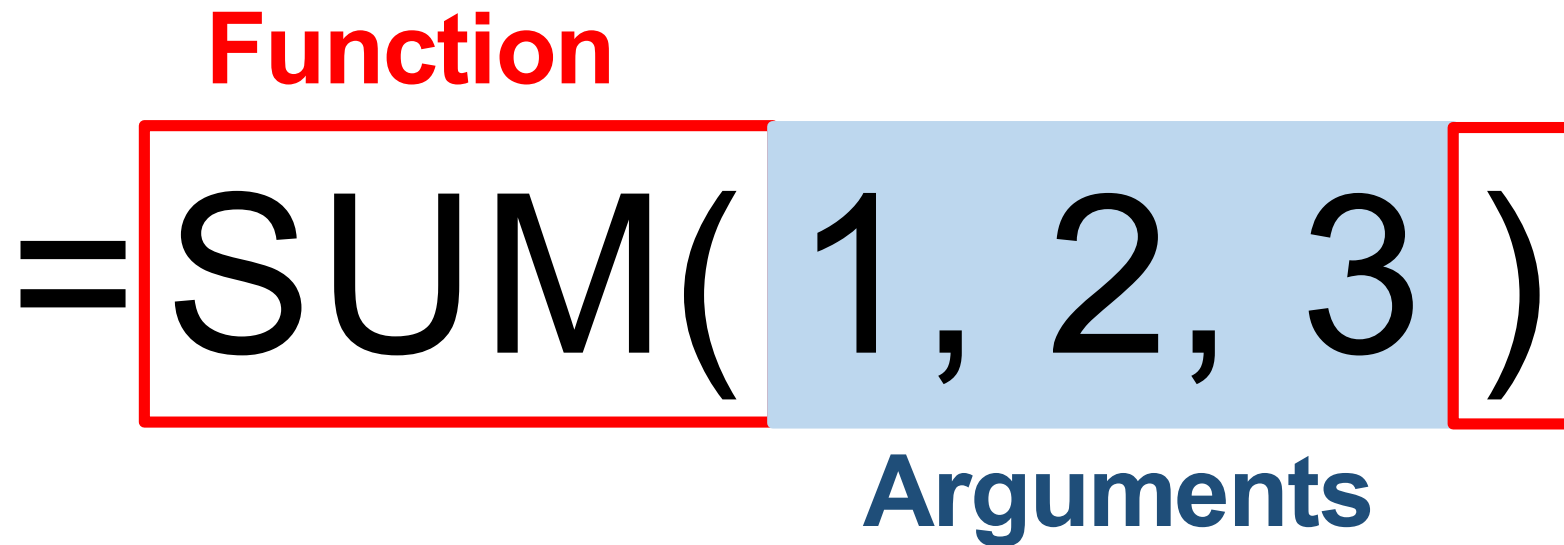
Formulas

Ooh... Coding! (Sort Of)

Function

=SUM(1, 2, 3)

Arguments

A diagram illustrating the components of an Excel formula. The formula "=SUM(1, 2, 3)" is shown. The text "Function" is written in red above the opening parenthesis of the SUM function. The text "Arguments" is written in blue below the numbers 1, 2, and 3. A red rectangular box highlights the entire formula "=SUM(1, 2, 3)". A light blue rectangular box highlights the arguments "1, 2, 3".

In a way, Excel has introduced you to a sort of proto-programming. Throughout your time writing scripts you will rely on **functions** (methods) that do *something* to or with **arguments**.

Ooh... Coding! (Sort Of)

Function

= **AVG**(F4:F6)

Variable Arguments

When we reference a set of range, Excel is being given a set of **variable** inputs. It will determine the actual values of these inputs prior to executing the function.

Ooh... Coding! (Sort Of)

What about this example?

Which is the function? Which are the arguments?

= SUM(AVG(F4:F6), AVG(G4:G6))

Ooh... Coding! (Sort Of)

What about this example?
Which is the function? Which are the arguments?

```
= SUM( AVG(F4:F6), AVG(G4:G6) )
```

It Depends...

Ooh... Coding! (Sort Of)

What about this example?
Which is the function? Which are the arguments?

= SUM(**AVG**(F4:F6), **AVG**(G4:G6))

The **AVG** functions takes as their arguments the ranges provided.

Ooh... Coding! (Sort Of)

What about this example?
Which is the function? Which are the arguments?

= SUM(AVG(F4:F6), AVG(G4:G6))

This is a **nested function**. We'll be doing plenty of complex nests in this class.

You Can Code Too!

Python Snippet from Last Class

```
requests.get(target_url_italian, headers=headers).json()  
requests.get(target_url_mexican, headers=headers).json()
```

You Can Code Too!

Python Snippet from Last Class

```
requests.get(target_url_italian, headers=headers).json()  
requests.get(target_url_mexican, headers=headers).json()
```

Function



Arguments



**Another Function
(Chained)**



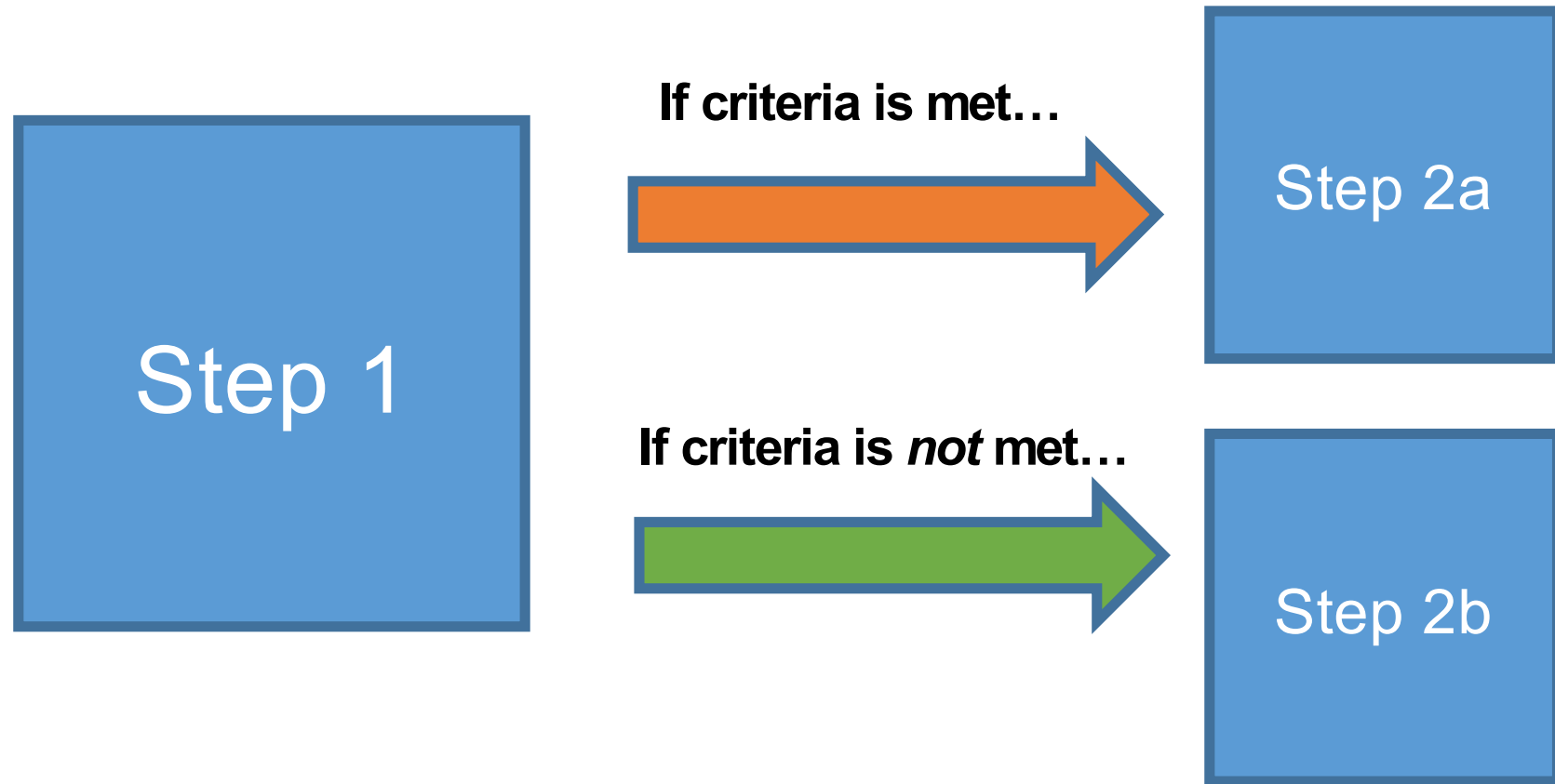
Syntax and capabilities may differ across technologies and platforms, but fundamental concepts remain the same.

Demo Time!

(01-ExcelPlayground, 02-NamedRanges)

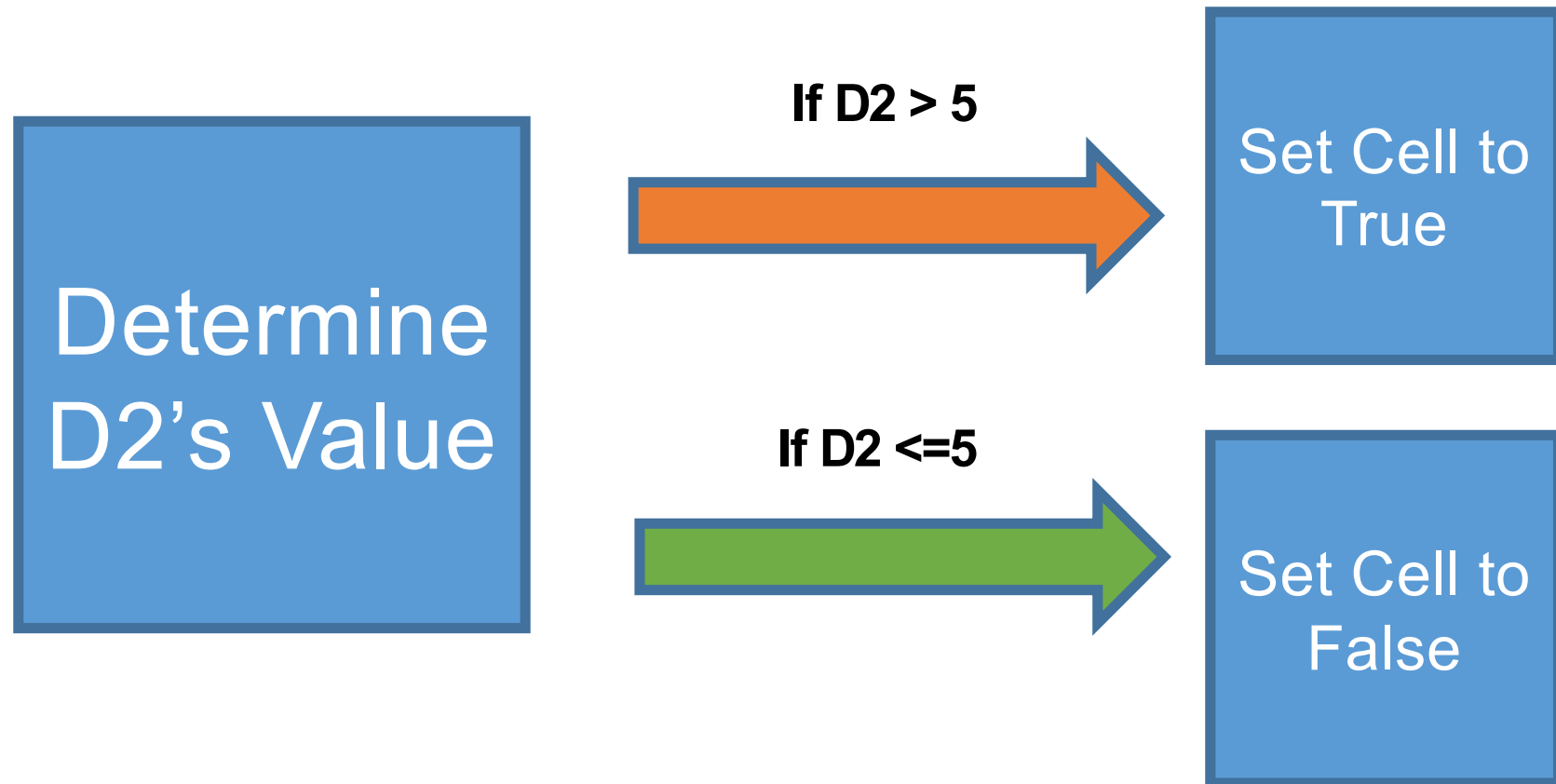
Conditionals

Conditionals: If This... Then That



Conditionals present a way to **control the flow** of logic based on certain criteria being met. This is a *core building block* in all languages.

Conditionals: If This... Then That



=IF(D2>5,TRUE,FALSE)

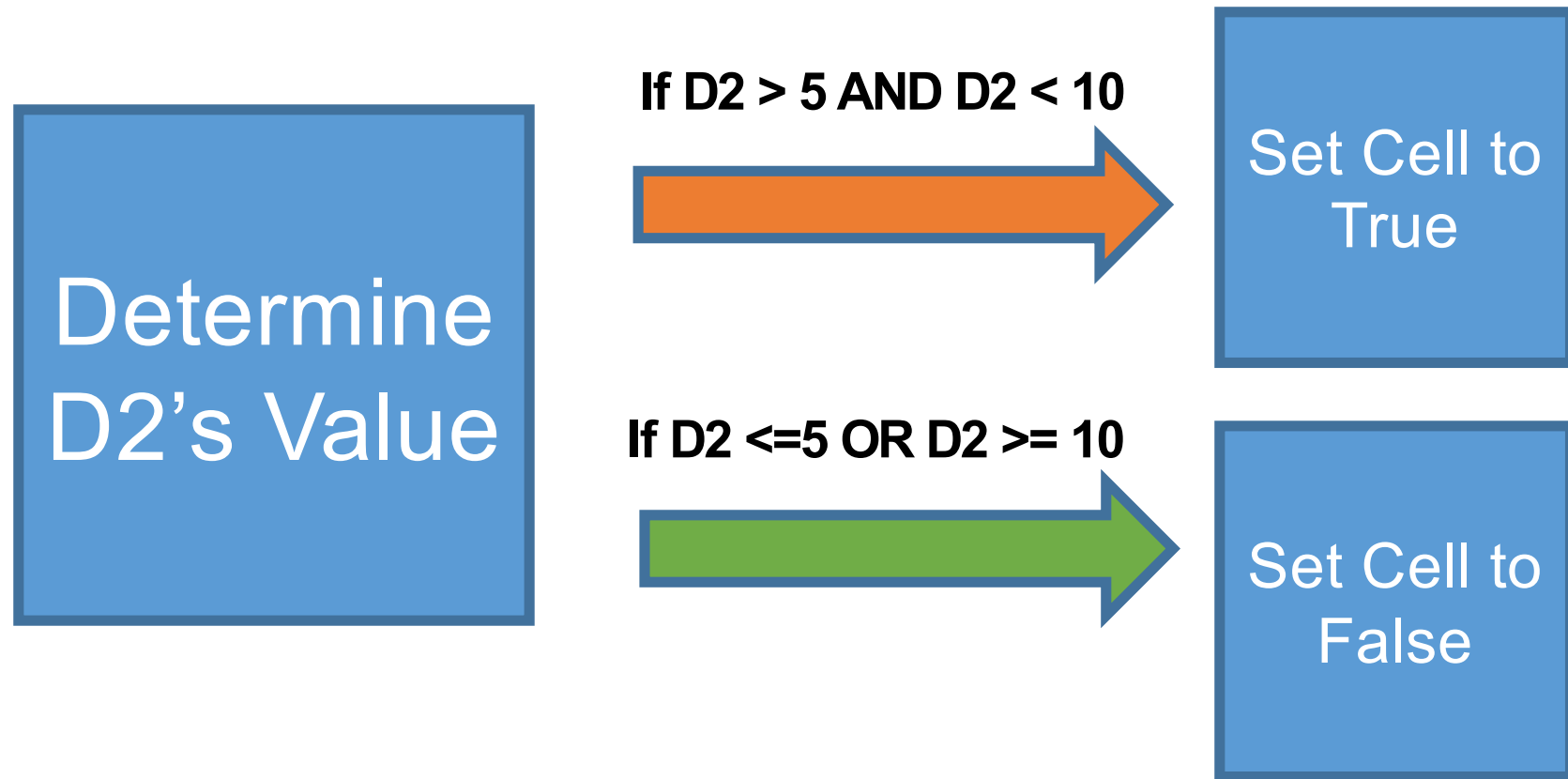
But what if... we wanted to combine conditions?

AND, NOT, OR

Conditionals: If This... Then That

=IF(AND(D2>5, D2<10),TRUE,FALSE)

Conditionals: If This... Then That



Nesting conditionals can quickly become a very convoluted (albeit necessary) part of your data prep.

Demo Time!

(03-ColorCounter – 08-McDonalds)

BREAK

Pivot Tables

Get Pivot With It

The screenshot shows an Excel spreadsheet with a PivotTable. The PivotTable has 'Row Labels' with months from January to December for the years 2014 and 2015. The columns are 'Cambridge', 'Piccadilly', and 'Grand Total'. The values are summed revenue. A dialog box 'Insert Calculated Field' is open, showing the name 'AverageRevenue' and the formula '= Revenue/ Reservations'. The 'Fields' list includes Year, Quarter, Month, RoomType, Revenue, and Reservations. The 'PivotTable Builder' pane on the right shows the same PivotTable structure.

Row Labels	Cambridge	Piccadilly	Grand Total
2014	\$ 1,111,886	\$ 1,214,733	\$ 2,326,619
January	\$ 90,005	\$ 94,910	\$ 184,915
February	\$ 104,397	\$ 133,914	\$ 238,311
March	\$ 53,546	\$ 80,115	\$ 133,661
April	\$ 103,543	\$ 98,960	\$ 202,503
May	\$ 111,353	\$ 93,664	\$ 205,017
June	\$ 94,292	\$ 98,108	\$ 192,400
July	\$ 112,334	\$ 73,953	\$ 186,287
August	\$ 68,446	\$ 76,590	\$ 145,036
September	\$ 82,581	\$ 152,078	\$ 234,659
October	\$ 103,366	\$ 78,984	\$ 182,350
November	\$ 82,564	\$ 134,740	\$ 217,304
December	\$ 105,459	\$ 98,717	\$ 204,176
2015	\$ 1,286,966	\$ 1,523,054	\$ 2,810,020
January	\$ 134,521	\$ 96,206	\$ 230,727
February	\$ 85,955	\$ 140,144	\$ 226,099
March	\$ 129,781	\$ 151,357	\$ 281,138

Pivot Tables are one of the most important data visualization concepts to master in this class.

(Don't worry. They are a cinch to deal with)

Get Pivot With It

Seller	Qty. Sold	Date
Joseph	\$42.50	1/1/17
Jacob	\$65.00	1/3/17
Jacob	\$5.25	1/6/17
Joseph	\$125.00	1/6/17
Jacob	\$3.50	1/7/17
Matt	\$32.00	1/9/17

Seller	Total Sold
Joseph	\$167.50
Jacob	\$73.75
Matt	\$32.0

In essence, Pivot tables are a **summative** analytic tool that allows us to perform aggregate functions that along any combination of fields.

(The name comes from the fact that we are pivoting along a data axis)

Demo Time!

(09-PivotTables, 10-TopSongs)

Lookups

Look It Up with Lookups

Planet	Population
Zeelo	5020
Merinoa	380
Cardboard Box	2
...	...
Asteroid 9	95

Assume this table is gigantic...

How would we **retrieve** the population of a specific planet for use in another formula?

Look It Up with Lookups

Planet	Population
Zeelo	5020
Merinoa	380
Cardboard Box	2
...	...
Asteroid 9	95

Assume this table is gigantic...

How would we **retrieve** the population of a specific planet for use in another formula?

=vlookup(<value>, <full table>, <column to retrieve>,<match parameter>)

What Will This Yield?

Planets

Planet	Population	Species
Zeelo	5020	Zoltans
Merinoa	380	Murphies
Cardboard Box	2	Hambones
...	...	
Asteroid 9	95	The Asterisks

=vlookup(“Asteroid 9”, Planets, 3, FALSE)

What Will This Yield?

Planets

Planet	Population	Species
Zeelo	5020	Zoltans
Merinoa	380	Murphies
Cardboard Box	2	Hambones
...	...	
Asteroid 9	95	The Asterisks

=vlookup(“Astroid 9”, Planets, 3, FALSE)

The Asterisks

Demo Time!

(11-Lookups, 12-ProductPivot)

Questions / Discussion
