

# Team Project

```
$ echo "Data Sciences Institute"
```

# Today

1. Team Project Overview
2. Developing a Project Plan
3. Project Requirements
4. Working as a Team

# Team Project Overview

**Goal:** To enhance your portfolio by showcasing this team project as evidence of your ability to deliver real-world value to an employer.

# What is a Good Portfolio?

A good portfolio showcases your best work and highlights skills that are highly valued by employers in data science and machine learning. It demonstrates your ability to handle real-world tasks, making you an attractive candidate for employment.

## Personalize Your Portfolio:

- **Remove unnecessary content:** As you work on your project, ensure that any unused files or folders are deleted to keep your repository clean and professional.
- **Highlight your unique contributions and skills:** This personalization shows employers that you're not just completing assignments but are engaged and innovating on your projects.

# Types of Projects to Include in Your Portfolio

Including a diverse range of projects in your portfolio can significantly enhance your appeal to potential employers. Consider including a variety of project types to demonstrate the breadth and depth of your data science skills:

- 1. Data Cleaning Project:** Show your ability to prepare data for analysis.
- 2. Data Storytelling and Visualization Project:** Highlight your skills in interpreting and presenting data in compelling ways.
- 3. ML Modeling:** Demonstrate your proficiency in building and tuning models.
- 4. Group Project:** Showcase your teamwork and collaboration skills. (already doing this! 😊)

For more, read "[How to Create a Project Portfolio for Data Science Job Applications](#)"

# Demonstrating the Business Impact of Your Projects

Understanding and articulating the business value of your data science and machine learning projects is crucial for two main reasons:

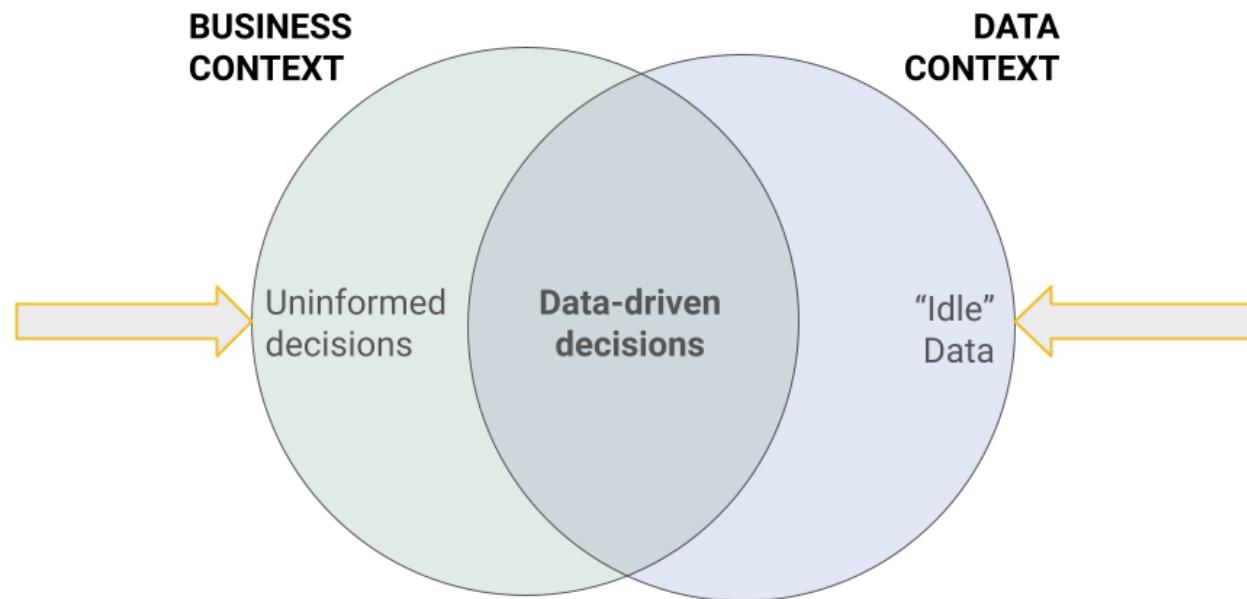
## Benefits:

- **Educational:** Apply new technologies and methodologies from both data science and machine learning.
- **Employability:** Showcase your ability to solve real-world problems, making you attractive to potential employers.

Highlight your skills through projects by using relevant industry examples and metrics, such as operational improvements, technical innovations, or strategic insights.

# Demonstrating the Business Impact of Your Projects

- Creating business value means allocating resources (eg. money, employees, time, hardware) in the way that generates the best return.
  - As a technical team, it is your job to enable the strategic decision makers with the information they need.



# Examples Across Sectors (Part 1)

## Healthcare Example

"Improved patient outcomes by developing a predictive model using Python and scikit-learn. Data was preprocessed using pandas to handle missing values and standardize features. Logistic regression was selected through cross-validation due to its interpretability and performance."

## Retail Example

"Optimized inventory management with SQL and Tableau. Data was cleaned and aggregated using SQL queries to identify sales patterns. Visualizations in Tableau helped to forecast demand, leading to cost savings."

 *Continued on next slide...*

# **Examples Across Sectors (Part 2)**

## **Marketing Example**

Increased customer engagement by performing customer segmentation using k-means clustering with Python. Feature engineering was conducted to create meaningful segments based on purchase history and demographics. The optimal number of clusters was determined using the elbow method, which helps identify the point where adding more clusters does not significantly improve the model.

# Selecting Projects That Showcase Your Skills

Choosing the right projects for your portfolio is crucial. Each project should:

- **Solve Real Problems:** Use actual datasets to address genuine issues in your field.  
*(examples in the next slide)*
- **Demonstrate Industry Relevance:** Select projects that are pertinent to your specific area, such as marketing analytics for marketers or predictive maintenance for engineers.
- **Provide Actionable Insights:** Focus on projects that deliver clear, practical outcomes that demonstrate your ability to impact real-world scenarios.

These criteria ensure that your projects not only highlight your technical skills but also your understanding of and adaptability to industry-specific challenges, making you a valuable candidate to potential employers.

# Showcase Projects That Use Diverse Data Types

Enhance your projects by effectively using a mix of structured, unstructured, and time series data:

- **Structured Data:** Employ database data for clear, quantifiable insights.
- **Unstructured Data:** Add depth with text, images, or videos.
- **Time Series Data:** Utilize data in sequential order for trend analysis and forecasting.

Select data types strategically to align with your project's objectives. Ensure each type contributes to a clear and coherent narrative, avoiding unnecessary complexity to maintain focus and utility.

# Demonstrating Technical and Business Skills

- **Demonstrate Coding Proficiency:** All projects should showcase your ability to write clean, efficient, and well-documented code.
- **Showcase Communication Skills:** All projects should effectively communicated complex data-driven insights to non-technical team members.

Your ability to document code clearly and communicate effectively can drastically reduce onboarding times for new team members and help non-technical stakeholders make informed decisions, thereby enhancing overall project success and team efficiency.

# Developing a Project Plan

1. Understand the business context.
2. Identify an opportunity.
3. Scope your analysis.
4. Develop your solution.
5. Present results and recommendations.

## To keep in mind:

- Are your insights *actionable*?
- How robust is your analysis? *Quantify!*
- What are the *risks* and *unknowns*?

# Project Examples

TTC Transit Delay Project

Mortgage Risk Assessment Project  
(Cohort 3)

Mexican Government Report Text Analysis  
(independent)

# Project Requirements

By the end of *Project 2*:

1. Each team member must **submit one Pull Request**, including writing a proper description.
2. Each project must **design, implement, and test regression**.
3. Each team must create a **data visualization or machine learning model**.
4. Each team must **create a README** that explains the project, your team's approach, and any relevant details. Keep in mind that the repo will be on each team member's profile, so be thoughtful about the details you include here.
5. Each team member must **create a video** about their learnings and experience.

# Project Requirements

***There are no requirements for the end of Project 1!***

However, you should think about the milestones you need to reach in Project 1 to set yourself up for success in Project 2.

1. Thoroughly understand your data and the business case for your analysis. What will the impact of your results be?
2. Clean your data. Be confident in the decisions you have made while doing so (eg. default handling of NULL values).
3. Test out regression analyses. It may take several tries before you are satisfied with your results and understand how you can provide the most insight. Projects are usually iterative, so dedicate time to investigation!

***There are no requirements for the end of Project 1!***

However, you should think about the milestones you need to reach in Project 1 to set yourself up for success in Project 2.

4. Make sure your code is well commented and decisions are documented. When Team Project 2 starts, you should be able to jump right back into your project without relying on memory.
5. Keep your README up to date. Not only is that easier than writing it all at the end of your project, it will help keep you on track and ensure your alignment with your business objective.

## Merge Conflicts

It is likely that you will encounter merge conflicts when combining your work as a team. Make sure you've reviewed the Git material regarding how to resolve merge conflicts properly!

# Crafting a Comprehensive Main README File

- **Purpose & Overview:** Introduce the project with essential details, concise description and a project objective.
- **Goals & Objectives:** Articulate what the project aims to achieve.
- **Techniques & Technologies:** Highlight the tools and methods used.
- **Key Findings & Instructions:** Summarize outcomes and provide setup instructions.
- **Visuals & Credits:** Enhance with visuals; acknowledge contributors.

## Better Documentation - README

While **you can have multiple README files** in your repository, include only one README file per folder to keep things clean and simple. This structure showcases your ability to manage and present complex information effectively, increasing your appeal to potential employers.

# Better Documentation - Comments

**Effective commenting enhances code readability and maintainability, crucial for collaborative environments.**

If someone with no tech background reads your code, they should get the essence of what you are doing and how the code flows. This practice not only aids in understanding but also facilitates smoother transitions and updates within team projects.

 Best practices for writing code comments

# Example



```
function calculateTotal(price, quantity) {  
    return price * quantity;  
}  
  
let totalPrice = calculateTotal(25, 5);  
console.log(totalPrice); // Output: 125
```



```
// Calculates the total cost by multiplying the price per item with the quantity  
function calculateTotal(price, quantity) {  
    return price * quantity;  
}  
  
// Example usage: Calculates the total price for 5 items at $25 each by multiplying the price  
// per item ($25) with the quantity (5), and stores the result in the totalPrice variable.  
let totalPrice = calculateTotal(25, 5);  
console.log(totalPrice); // Output: 125
```

# Better Documentation - STAR Method

- The STAR method is typically used for answering interview questions, and applying this structure to your project overviews can be highly effective.
- By organizing your projects using the Situation, Task, Action, Result format, your portfolio essentially speaks for you, conducting a virtual interview with potential employers even in your absence.
- This approach ensures that employers can clearly understand the value and impact of your work, even without direct interaction.
- Remember, the results don't have to be ground breaking, anything that showcases your abilities in data science/machine learning, even "**learning**" counts as an important result.

# Effective GitHub Repository Organization

- Ensure your GitHub repository is neatly organized; avoid unused or empty folders and ensure each folder has a clear purpose.
- Use READMEs in key folders (like `src`, `iac`, `backend`, `frontend`, etc.) to detail their contents and purpose, aiding clarity for complex sections.
- Avoid READMEs in simple folders (like `images`, `docs`, etc.), unless there's specific information that needs to be explained.

# Project Structure

- **One** person from each team must host the primary Git repository. Other team members should clone that repository to work on it.
- There is no hard requirement for the folder structure of your project, but best practices should be followed.

## Working as a Team

This week will fly by quickly! How will you ensure that you can accomplish everything you need to as a team in this timeframe?

# Working as a Team

1. **Clear objectives and milestones.** Milestones should be set working backward from the final project goal and deadline.
2. **Accountability.** Every action item should have someone assigned. (Of course, the assigned person can change if workloads become unbalanced.)
3. **Communication of roadblocks.** It is just as important to communicate roadblocks and failures as successes. The faster problems are identified, the faster they can be addressed.

## Working as a Team

4. **Tracking.** Tracking your tasks is necessary to objectively assess your progress as a team. The data-driven mindset applies here too! If you fall behind, update your plan accordingly.
5. **Documentation.** Team members should be able to understand and continue each others' work where necessary.

## This week's schedule:

**Tues, Oct. 15th:** 1 hour of content delivery, 1.5 hours of co-work.

**Wed, Oct. 16th:** 2.5 hours of co-work.

**Thurs, Oct. 17th:** 1 hour of case study, 1.5 hours of co-work.

**Fri, Oct. 18th:** 3 hours of co-work.

**Sat, Oct. 19th:** 3 hours of co-work.

# Handling Uncertainty in Your Project

Throughout your project, there will be moments when you are unsure of how to proceed, or what task to work on next. The *Orders of Ignorance* is a concept that can help guide you in these situations.

## The Five Orders of Ignorance

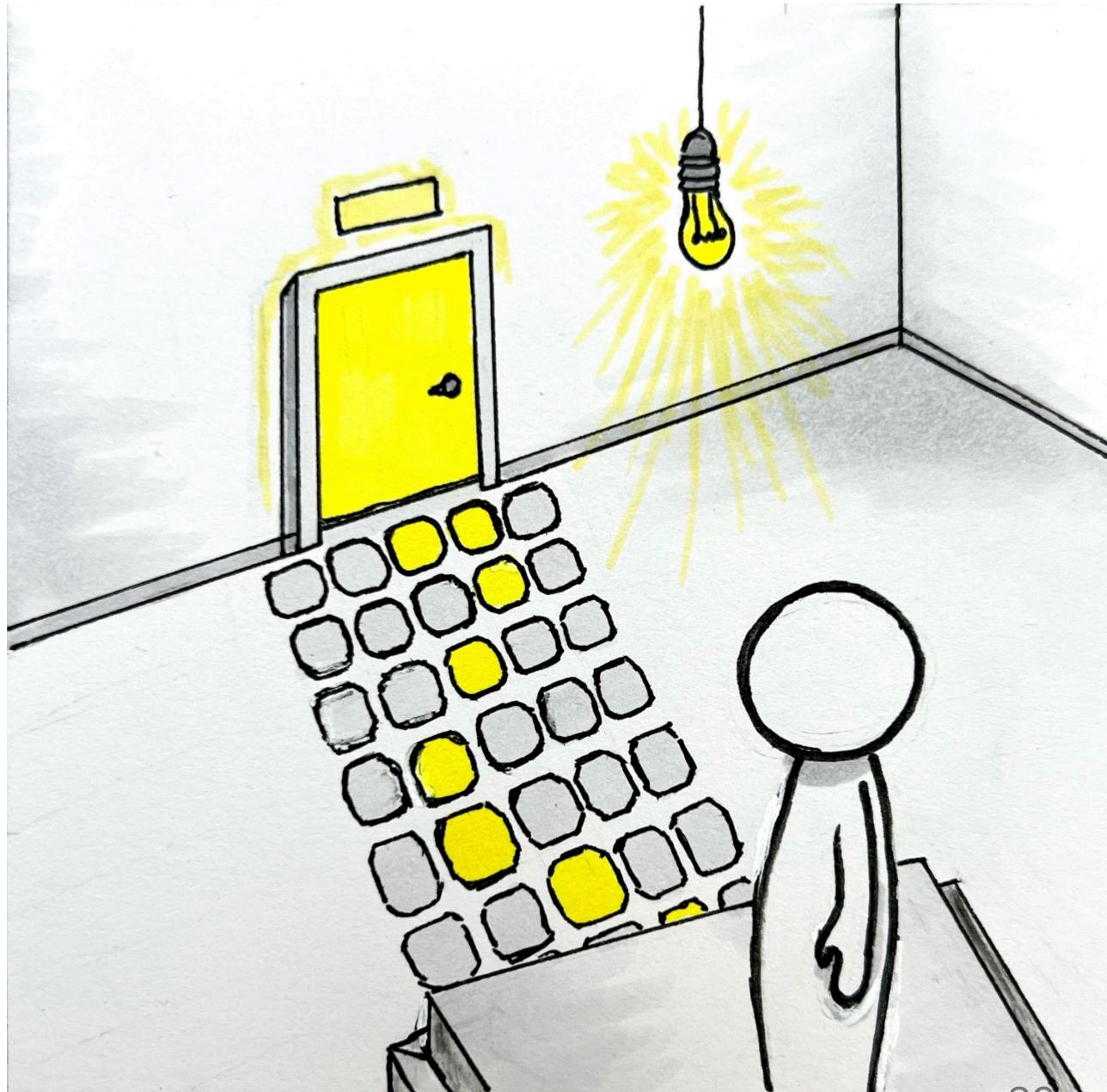
A "meta-model of ignorance" that can help us put the unknowns of projects into context.

0. Zeroth Order Ignorance (0OI): Lack of Ignorance
1. First Order Ignorance (1OI): Lack of Knowledge
2. Second Order Ignorance (2OI): Lack of Awareness
3. Third Order Ignorance (3OI): Lack of Process
4. Fourth Order Ignorance (4OI): Meta Ignorance

## The Five Orders of Ignorance

### 0. Zeroth Order Ignorance (0OI): Lack of Ignorance

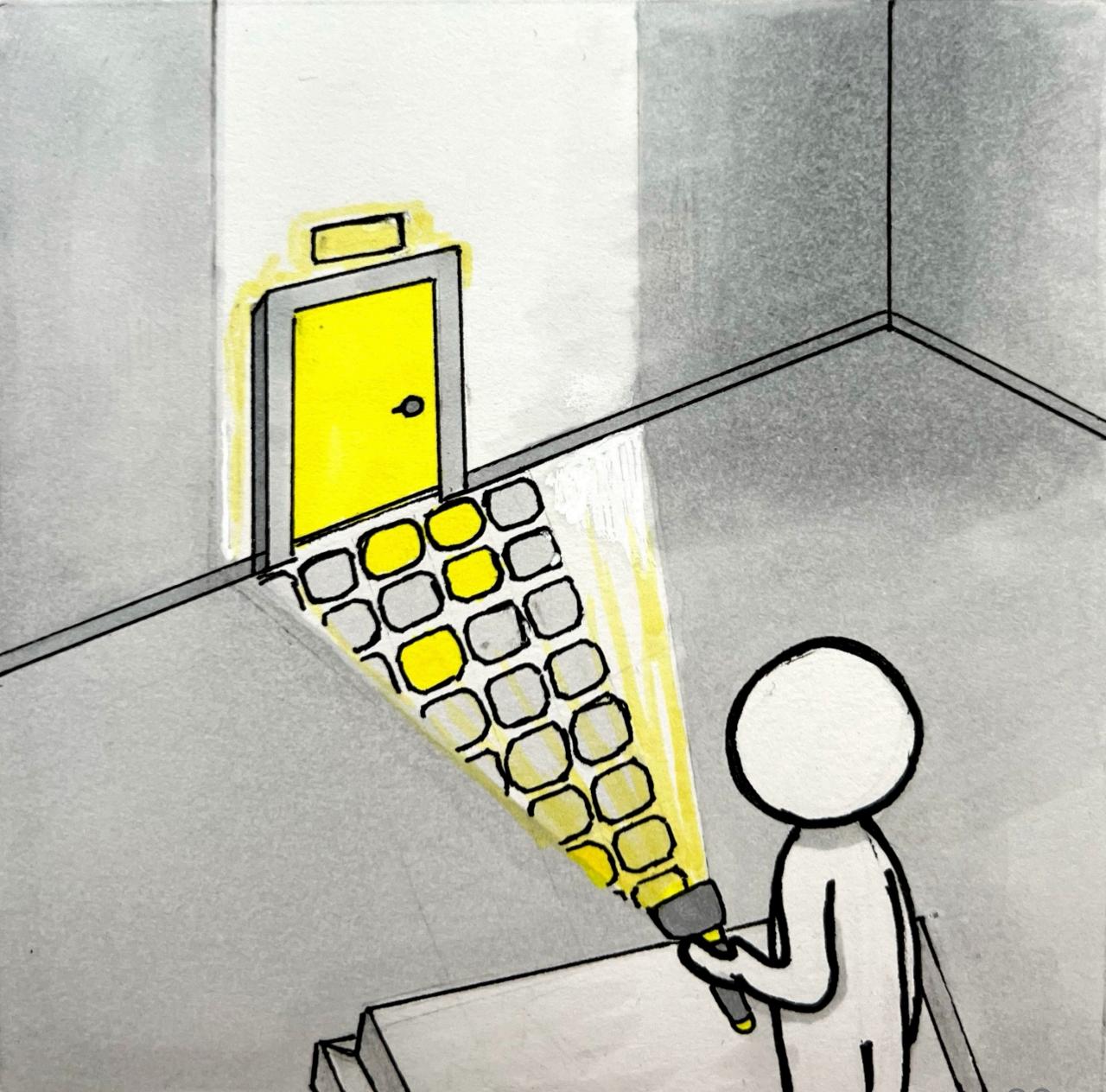
- You know there is a door.
- You know where the door is.
- You can see the path to the door clearly.



# The Five Orders of Ignorance

## 1. First Order Ignorance (1OI): Lack of Knowledge

- You know there is a door.
- You do not know the path to the door, but you know you can find the path with a tool that you have (eg. a flashlight).



## The Five Orders of Ignorance

### Second Order Ignorance (2OI): Lack of Awareness

- You know there is a door, but you do not know how to get to it or where to start looking.



## The Five Orders of Ignorance

### Third Order Ignorance (3OI): Lack of Process

- You know you are in a room, but don't know if there is any way out.



## The Five Orders of Ignorance

### Fourth Order Ignorance (4OI): Meta Ignorance

- Complete and utter darkness.



# Leveraging Project Management Methodologies

Explore methodologies like Scrum, Kanban, Waterfall, and Agile on your own. For now, let's focus on an essential concept: **spikes**.

**Spikes** are short, time-boxed research tasks designed to answer specific questions or solve particular problems before starting the main project work.

# Benefits of Spikes

- **Clarify Requirements:** Quickly gather necessary information.
- **Solve Problems Early:** Address technical uncertainties before main development.
- **Prevent Delays:** Avoid excessive brainstorming that can stall progress.
- **Reduce Unknowns:** Lower the project's uncertainties.
- **Efficient Estimates:** Provide well-constrained estimates on specific work items.
- **Maximize Value:** A one-day spike giving 80% confidence can be more valuable than a three-day spike giving 90%.

# Goals for Today

- Understand your objective and your data.
- Break down project into milestones and assign tasks.
- Identify spikes.
- Get started!

# Questions?