**Project-Based Course Overview**

Welcome to Exploratory Data Analysis with Seaborn! This is a project-based course which should take approximately 1.5 hours to complete. Before diving into the project, please take a look at the course objectives and structure:

**Course Objectives**

In this course, we are going to focus on two learning objectives:

1. Produce data visualizations with Seaborn.
2. Apply graphical techniques used in exploratory data analysis (EDA).

By the end of this course, you will be able to generate publication-quality graphs using Seaborn and Python to analyze data.

**Course Structure**

This course is divided into 3 parts:

1. Course Overview: This introductory reading material.
2. Exploratory Data Analysis with Seaborn Project: This is the hands on project that we will work on in Rhyme.
3. Graded Quiz: This is the final assignment that you need to pass in order to successfully complete the course and earn a Course Certificate.

**Project Structure**

The hands on project on Exploratory Data Analysis with Seaborn is divided into the following tasks:

**Task 1: Introduction and Importing the Data**

* In this task, we are introduced to the project and learning outcomes.
* Once we are familiarized with the Rhyme interface, we begin working in [Jupyter Notebooks](https://jupyter.org/), a web-based interactive computational environment for creating notebook documents.
* Next, we will import essential libraries such as NumPy, pandas, Seaborn, and matplotlib.
* Lastly, we use pandas to load the Breast Cancer Wisconsin (Diagnostic) [Data Set](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)).

**Task 2: Separate Target from Features**

* Now that the data set is in memory, we can explore the characteristics of its attributes and instances.
* We will drop columns that cannot be used for analysis and classification.
* Note that this does not constitute feature selection. We are dropping columns that have no bearing on the analysis we will be conducting, and will instead clutter our analysis. After producing descriptive statistics about the data, we will separate the target from the features.
* The target contains the diagnosis with binary class labels, **M** or **B**, for malignant and benign tumors respectively.

**Task 3: Diagnosis Distribution Visualization**

* A very common question during model evaluation is, "Why isn't the model I've picked predictive?". Most often, it is a result of a class imbalance.
* In this task, we will use Seaborn's [countplot()](https://seaborn.pydata.org/generated/seaborn.countplot.html) method to visualize the target distributions.
* We will also generate descriptive statistics about the features that summarize the central tendency, dispersion and shape of the data set's distribution.

**Task 4: Visualizing Standardized Data with Seaborn**

* As the columns in the data set take on values of varying range, we need to standardize the data before proceeding with further analysis and visualization.
* To begin feature analysis, we use Seaborn's [violinplot()](https://seaborn.pydata.org/generated/seaborn.violinplot.html) method. Violin plots are similar to box plots, except that they also show the probability density of the data at different values, usually smoothed by a kernel density estimator.

**Task 5: Violin Plots and Box Plots**

* We are using violin plots and box plots to identify features that best separate the data for classification.
* Box plots are especially useful in identifying outliers in the data.
* Using violin plots, we are also able to infer whether certain features are correlated.
* To minimize clutter in our visualizations, we divide the features into three batches of ten features and produce separate plots for them.

**Task 6: Using Joint Plots for Feature Comparison**

* Joint plots come in handy to illustrate the relationship between two features.
* We will use seaborn's [jointplot()](https://seaborn.pydata.org/generated/seaborn.jointplot.html) method to draw a scatter plot with marginal histograms and kernel density fits. We can examine the relationship between any two features using the Pearson correlation coefficient of the regression through our scatter plot.

**Task 7: Observing the Distribution of Values and their Variance with Swarm Plots**

* We have learned that violin plots are a great tool for visualizing sparse distributions. As our data set contains close to 600 rows, we might want to simply display each point in the same visualization.
* This need is satisfied by Seaborn's [swarmplot()](https://seaborn.pydata.org/generated/seaborn.swarmplot.html) method. A swarm plot can be drawn on its own, but it is also a good complement to a box or violin plot in cases where you want to show all observations along with some representation of the underlying distribution.

**Task 8: Observing all Pairwise Correlations**

* A good way to identify correlations between features is to visualize the correlation matrix as a heatmap.
* We will make a note of the correlated features so that we can drop them from our data set before building a predictive model in the next project.
* In the next project, we will remove these correlated features and analyze the classification accuracy we get using XGBoost, a boosted decision tree classifier. We will then employ various feature selection and feature extraction methods to get the most predictive features and improve our classification accuracy.

**Exploratory Data Analysis with Seaborn**

Now, you will use [Seaborn](https://seaborn.pydata.org/) to discover and explore the relationships in the Breast Cancer Wisconsin (Diagnostic) [data set](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)). We will accomplish this with the help of following tasks in the project:

* Task 1: Introduction and Importing the Data
* Task 2: Separate Target from Features
* Task 3: Diagnosis Distribution Visualization
* Task 4: Visualizing Standardized Data with Seaborn
* Task 5 : Violin Plots and Box Plots
* Task 6: Using Joint Plots for Feature Comparison
* Task 7: Observing the Distribution of Values and their Variance with Swarm Plots
* Task 8: Observing all Pairwise Correlations