# Final Assignment

Due Monday, October 26, 2026

This assignment brings together everything from Modules 1 to 9: NumPy simulations, Pandas, Seaborn, SciPy, statsmodels, and scikit-learn. We'll use Jupyter Notebooks with Markdown cells describing what you're doing and why, and code cells showing your work.

It has two parts. As a group, you will analyze a wine dataset (Sections 1 to 8). Individually, you will explore a dataset of your own choosing (Section 9). On October 26th, your group will present its notebook to the class. And then you will each briefly share your dataset and your data visualization for it.

For the group project, we'll look at the physical and chemical properties of wine. The file, `notebooks/data/wine_quality.arff`, has over 6k wines, each with 11 measurements, such as `alcohol`, `pH`, and `residual.sugar`, and a `quality` score. Your group's goal is to form a hypothesis about the data, check it, and then model what makes a wine good. Sections 1 to 8 are done as a group, in one notebook.

The data comes from Cortez et al. (2009), "Modeling wine preferences by data mining from physicochemical properties", *Decision Support Systems* 47(4):547-553. It is also available on OpenML: https://www.openml.org/search?type=data&id=287

## 1. Hypothesis and Simulation

Before we look at any values, let's say what we expect to see.

- Choose any one measurement to focus on, other than `quality`. For example, `alcohol`, `pH`, `residual.sugar`, or `chlorides`, but you can pick any of the others.
- In a Markdown cell, write your hypothesis: what shape do you expect the distribution to have, and roughly what mean and standard deviation? Explain your reasoning. Do not look at the data to answer this.
- Create a seeded random number generator with `np.random.default_rng()`
- Write a function that takes the generator, a mean, a standard deviation, and a number of samples, and returns simulated values from `rng.normal()`. If you expect a skewed distribution, use `rng.lognormal()` instead.
- Use your function to simulate the same number of wines as in the dataset
- Use a `for` loop over sample sizes of 10, 100, 1,000, and 6,497. For each, print the simulated sample's mean, its standard deviation (`ddof=1`), and the standard error, `std / np.sqrt(n)` as a formatted string, f-string, with a description.
- Make a histogram of all your simulated values with a title and labeled axes
- In a Markdown cell, answer: how close is the sample mean to the mean you chose, and how does that change with the sample size? What does the standard error tell you?

## 2. Loading Data

- Load the data with `scipy.io.arff`, which can read the `.arff` format that Pandas can't:

```
from scipy.io import arff

data, meta = arff.loadarff('data/wine_quality.arff')
df = pd.DataFrame(data)
```

Your path depends on where your notebook is saved relative to the `data` folder. - Show the first 5 and the last 5 rows - Print the `shape`, `columns`, and `dtypes` of `df` - The column names contain dots, like `fixed.acidity`, which don't work inside formulas. Replace the dots with underscores, e.g. with `df.columns.str.replace('.', '_', regex=False)`. - Check for missing values with `df.isna().sum()` - In a Markdown cell, answer: how many wines and how many measurements are in the data? What does each row represent? Is anything missing?

## 3. Summarizing the Data

- Run `.describe()` on `df`
- Use the `mean()` method of a `Series` to compute the mean of your chosen measurement, and compute its standard deviation
- Use `.value_counts()` on `quality` and sort it by score. Hint: `.sort_index()` sorts by the index, which is the score.
- Create a Boolean column called `good`, which is `True` for wines with a `quality` of 6 or higher. Convert it to `0` and `1` with `.astype(int)`.
- Use `.query()` to select the wines with `alcohol` above 12, and print how many there are
- Use `groupby` and `agg` to compute the mean and standard deviation of `alcohol` for each `quality` score. Hint: `.agg(['mean', 'std'])`
- Use `groupby` on `good` to compute the mean of `alcohol`, `volatile_acidity`, `sulphates`, and `density`
- In a Markdown cell, answer: what fraction of the wines are `good`? Which of these measurements look different between good and not-good wines?

## 4. Checking the Hypothesis

- Make a figure with two subplots side by side:
  - On the left, a `sns.histplot()` of your simulated values from Section 1
  - On the right, a `sns.histplot()` of your chosen measurement from the real data
- Use the same x-axis limits on both, so that they can be compared. Add a title and label your axes for each plot.
- Compute the real mean and standard deviation of your measurement, and compare them with your hypothesis
- In a Markdown cell, answer: was your hypothesis right about the shape, the mean, and the standard deviation? If not, what did you get wrong, and why do you think so?

## 5. Plotting with Seaborn

- Make a figure with two subplots using `fig, axes = plt.subplots()`:
  - On the left, a `sns.boxplot()` of `alcohol` by `quality`
  - On the right, a `sns.histplot()` of `alcohol` with `hue='good'`
- Make a second figure with `sns.regplot()` of `alcohol` against `density`
- Add a title and axis labels to every plot, and save your figures with `fig.savefig()`
- In a Markdown cell, answer: how does `alcohol` change as `quality` goes up? What do you notice about `density` and `alcohol`?

## 6. Correlation and t-tests with SciPy

- Use `stats.pearsonr()` to get the correlation between `alcohol` and `quality`. Then use `stats.spearmanr()`. Compare the two and describe the difference.
- Use `stats.ttest_ind()` to compare the `alcohol` of good wines and not-good wines. Pull each group out with `.query()`.
- In a Markdown cell, describe the results as you would in a journal article.

## 7. Regression with statsmodels

- Use `model_matrix()` from `formulaic` to build `y` and `X` from the formula `'quality ~ alcohol + volatile_acidity + sulphates'`
- Fit it with `sm.OLS(y, X).fit()` and print the `summary()`
- Fit a second model that adds two more measurements of your choice
- In a Markdown cell, answer: explain what coefficient fo `alcohol` represent and mean? Which predictors are significant? How much did the `R-squared` change when you added predictors?

## 8. Prediction with scikit-learn

- Use `train_test_split()` to hold out 20% of the wines, with `random_state=42`
- Train a `LinearRegression` to predict `quality` from `alcohol`, `volatile_acidity`, and `sulphates`. Report the test $R^2$ and the RMSE.
- Train a `LogisticRegression` to predict `good` from the same features. Use `stratify` in your split. Report the accuracy and the confusion matrix on the test set.

## 9. Your Own Dataset (Individual)

This section is done on your own, in a separate notebook from your group's. Choose a dataset about something that interests you: sports, music, movies, health, travel, the environment, or anything else. Good places to look are Kaggle, OpenML, and data.gov. Pick a table with at least 100 rows and a few numeric columns, so that there is something to summarize.

- In a Markdown cell, say where the data comes from, with a link, what each row represents, and why it interests you
- Load the data with Pandas, and show the first 5 rows and the `shape`, `columns`, and `dtypes`. Check for missing values.
- Summarize it: run `.describe()`, and compute the mean and median of at least two columns. If the data has a categorical column, use `groupby` and `agg` to compare a numeric column across its categories.
- Explore it: ask at least two questions about your data and answer each with code, e.g. with `.query()`, `.value_counts()`, or a correlation from SciPy
- Create a data visualization that fits one of your questions, using Seaborn or Matplotlib. Add a title and axis labels, and save your figure with `fig.savefig()`.
- In a Markdown cell, answer: what did you find? What does your figure show, and what is one limitation of this data?

## 10. Present Your Notebook

On October 26th, your group will get about five minutes to walk us through your wine notebook. Be ready to cover:

- Your hypothesis from Section 1, and whether the data agreed with it
- One figure that you find most informative
- What you learned doing the assignment

Then each of you will present your personal notebook for two minutes. Make sure to cover:

- Your dataset
- Your data visualization
- Any highlights you learned

## 11. Submit Your Work

Download your group's notebook and your own notebook from Jupyter Lab and upload them to https://mskeducation.mskcc.org/ by Monday, October 26th, 1:30PM.