Assignment 3

Due Friday, October 9, 2026

This assignment covers material from Modules 5 and 6: objects and classes, Pandas, and Seaborn. We'll use Jupyter Notebooks with Markdown cells describing what you're doing and why, and code cells showing your work.

Let's work on our `personal` branch. First, bring it up to date with the course repo:

```
git switch main
git pull upstream main
git switch personal
git rebase main
```

Save your work on `personal` as you go with `git add` and `git commit`.

Throughout this assignment, we'll look at RNA-seq data. RNA-seq measures how active each gene is in a sample by counting the sequencing reads that map to it. Our file, `notebooks/data/RNAcountTable_chr1_geneNames.tsv`, has read counts for the genes on chromosome 1, from three cell types (`FB`, `HSC`, and `TREG`) with two replicates of each (`_1` and `_2`). Our goal is to explore the data, summarize it by cell type, and make some plots.

1. Objects and Classes

- Create a seeded random number generator with `np.random.default_rng()`
- Define a class `Experiment` that stores a `name` and a NumPy array of `values`, with a method `summary()` that returns the mean and standard deviation of `values`
- Create two `Experiment` objects with simulated data from `rng.normal()`, each with a different `loc` and `scale`, and call `summary()` on each
- In a Markdown cell, answer: which parts of `Experiment` are attributes and which are methods? How close is each `summary()` to the `loc` and `scale` you used?

2. Loading Data

- Load `notebooks/data/RNAcountTable_chr1_geneNames.tsv` into a DataFrame called `df` with `pd.read_csv()`. The columns are separated by tabs, and 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`
- In a Markdown cell, answer: how many genes and how many samples are in the data? What does each row represent, and what does each number in the table represent?

3. Renaming and Selecting Data

- Rename the `Gene` column to `gene`
- Use `.iloc` to select the first 5 genes and the first 3 sample columns
- Use `.loc` to select the `gene`, `FB_1`, and `FB_2` columns for the genes `Fcmr`, `Sell`, and `Ptprc`. Hint: `df['gene'].isin(['Fcmr', 'Sell', 'Ptprc'])` gives a Boolean Series.
- Use `.query()` to select the genes with a count above 100,000 in `FB_1`
- Count how many genes have a count of 0 in all six samples. Hint: `df[sample_columns].sum(axis=1)` adds up each row, where `sample_columns` is a list of your six sample column names.
- In a Markdown cell, answer: how do `Fcmr`, `Sell`, and `Ptprc` compare across the two replicates of `FB`? What fraction of the genes were not detected in any sample?

4. Summary Statistics

- Compute the total counts for each sample: `df[sample_columns].sum()`

- Compute the mean and median count of each sample
- Run `.describe()` on `df`
- In a Markdown cell, answer: are the six samples similar in their total counts? For each sample, is the mean or the median larger, and what does that tell you about how the counts are distributed across genes?

5. Extracting Metadata

Right now, the cell type and the replicate are stuck in the column names. Let's put them in their own columns.

- Use `.melt()` to turn `df` into a long DataFrame called `df_long` with the columns `gene`, `sample`, and `count`
- Use `.str.split()` on `sample` to add `group` (the cell type) and `replicate` columns
- Print the `shape` of `df_long`
- In a Markdown cell, answer: why does `df_long` have the number of rows that it has? What does each row represent now?

6. Group-Apply-Combine

- Use `groupby` to compute the mean and median `count` for each `group`
- Use `groupby` on `group` and `gene` to compute the mean `count` of each gene in each cell type, and use `.reset_index()`
- Sort that result to find the top 5 genes in each cell type. Hint: `.sort_values('count', ascending=False)` and `.groupby('group').head(5)`.
- In a Markdown cell, answer: which gene has the highest mean count in each cell type? Are any genes in the top 5 of more than one cell type?

7. Plotting with Seaborn

Read counts span several orders of magnitude, so we'll plot them on a log scale.

- Add a column `log_count` to `df_long` with `np.log2(df_long['count'] + 1)`. The `+ 1` is there because `log2(0)` is undefined.
- Using `fig, axes = plt.subplots()` with two subplots side by side:
  - On the left, a `sns.histplot()` of `log_count` with `hue='group'`
  - On the right, a `sns.boxplot()` of `log_count` by `group`, with `hue='replicate'`
- Make a second figure: a scatterplot of the two `FB` replicates against each other, using `np.log2(df['FB_1'] + 1)` and `np.log2(df['FB_2'] + 1)`
- Add a title and axis labels to every plot, and save your figures with `fig.savefig()`
- In a Markdown cell, answer: what shape do the distributions of `log_count` have, and why? How well do the two `FB` replicates agree, and why would we want them to?

8. Submit Your Work

Download your notebook from Jupyter Lab and upload it to https://mskeducation.mskcc.org/ by Friday, October 9th 11:59PM.