

# Introduction to Jupiter Notebooks: Problem Solving

Let's practice the skills we learned during the last lesson on Jupyter Notebooks.

## Create a Spreadsheet of Data

Create an Excel spreadsheet as follows.

- Sign into <https://www.office.com/launch/excel> using your `memphis.edu` email.
- Name the spreadsheet file *scores*.
- The first row must be headings only.
- The first three columns must have the following headings: *Name*, *Exam1*, and *Exam2*.
- Below the headings row, there must be 15 rows of data. Fill the rows in with a variety of made-up names and exam scores (assuming that each exam score is between 0 and 100 points).

Upload the spreadsheet into your JupyterLab workspace.

## Create a New Notebook

Create a new xpython notebook named `Jupyter-notebooks-intro-PS.ipynb` in your JupyterLab workspace.

- Be sure that the notebook is using the *xpython* kernel (NOT the *Python 3* kernel).

## Add a Markdown Cell

Make the first cell in the notebook a Markdown cell that looks like the following figure.

### Introduction to Jupyter Notebooks: Problem Solving

In this notebook, we practice some common data analysis tasks.

#### Import the Pandas Library

First, we import the Pandas library.

- Don't forget to set the cell type to *Markdown*.
- Hint: The cell includes a level-1 heading and a level-2 heading.
- Hint: See the GitHub Markdown Cheatsheet for Markdown examples.

Run the cell to make it render the Markdown as pretty headings and text.

## Import the Pandas Library

Add a code cell that imports the Pandas library.

- Import the Pandas library as `pd`.

Run the code cell. (No output should be displayed.)

## Read the Spreadsheet into a DataFrame

Add a Markdown cell that looks like this:

### Read the Spreadsheet into a DataFrame

Next, we read the spreadsheet into a DataFrame named `df`.

- Hint: The heading in this cell (and all future Markdown cells) is a level-2 heading.
- Hint: To format the `df` variable, we surround the `df` in backticks (i.e., “s”).

Add a code cell that reads the spreadsheet you uploaded into a DataFrame.

- Name the DataFrame variable `df`.

Run the code cell. (No output should be displayed.)

## Display the Contents of the DataFrame

Add a Markdown cell that looks like this:

### Display the Contents of the DataFrame

Next, we display a summary of the contents of the DataFrame.

Add a code cell that displays the contents of the DataFrame.

- Hint: This involves only adding a variable block.

Run the code cell and note the output, which should resemble the following:

```
[ 8]:
```

|    | Name   | Exam1 | Exam2 |
|----|--------|-------|-------|
| 0  | Bob    | 45    | 56    |
| 1  | Alice  | 89    | 87    |
| 2  | Jane   | 62    | 98    |
| 3  | Rick   | 99    | 100   |
| 4  | Perry  | 100   | 100   |
| 5  | Louis  | 76    | 79    |
| 6  | Mike   | 88    | 85    |
| 7  | Romeo  | 88    | 80    |
| 8  | Juliet | 84    | 96    |
| 9  | Wendy  | 96    | 93    |
| 10 | Peter  | 98    | 88    |
| 11 | Homer  | 91    | 100   |
| 12 | Bart   | 55    | 34    |
| 13 | Lisa   | 87    | 86    |
| 14 | Marge  | 79    | 97    |

## Display an Individual Column in the DataFrame

Add a Markdown cell that looks like this:

### Display an Individual Column in the DataFrame

Next, we display the Exam1 column of the DataFrame.

Add a code cell that displays a summary of only the data in the Exam1 column.

- Hint: This involves one of the *Lists* blocks.

Run the code cell and note the output, which should resemble the following:

```
[9]:
```

|    | Exam1 |
|----|-------|
| 0  | 45    |
| 1  | 89    |
| 2  | 62    |
| 3  | 99    |
| 4  | 100   |
| 5  | 76    |
| 6  | 88    |
| 7  | 88    |
| 8  | 84    |
| 9  | 96    |
| 10 | 98    |
| 11 | 91    |
| 12 | 55    |
| 13 | 87    |
| 14 | 79    |

## Calculate the Average (Mean) Value of a Column in the DataFrame

Add a Markdown cell that looks like this:

### Calculate the Average (Mean) Value of a Column in the DataFrame

Next, we calculate the average value in the Exam1 column.

Add a code cell that calculate the average value in only the data in the Exam1 column.

- Hint: First, store the Exam1 column as a list in a new variable (call it `exam1`). Then, run the `mean` operation on the `exam1` variable.

Run the code cell and note the output, which should resemble the following:

```
[10]: Exam1    82.466667
      dtype: float64
```

## Calculate the Average (Mean) Values of All Columns in the DataFrame

Add a Markdown cell that looks like this:

### Calculate the Average (Mean) Values of All Columns in the DataFrame ¶

Next, we calculate the average value of each and every column in the DataFrame.

Add a code cell that calculates the average value for each column in the DataFrame.

- Hint: This can be done by calling the `mean` operation on the entire DataFrame.

Run the code cell and note the output, which should resemble the following:

```
[11]: Exam1    82.466667
      Exam2    85.266667
      dtype: float64
```

## Display a Summary of Descriptive Statistics for Each Column in the DataFrame

Add a Markdown cell that looks like this:

### Display a Summary of Descriptive Statistics for Each Column in the DataFrame

Lastly, we calculate a variety of descriptive statistics for each column of `df`.

Add a code cell that displays a summary of descriptive statistics for each column in the DataFrame.

- Hint: This can be done by calling the `describe` operation on the entire DataFrame.

Run the code cell and note the output, which should resemble the following:

```
[12]:
```

|       | Exam1      | Exam2      |
|-------|------------|------------|
| count | 15.000000  | 15.000000  |
| mean  | 82.466667  | 85.266667  |
| std   | 16.552370  | 18.300924  |
| min   | 45.000000  | 34.000000  |
| 25%   | 77.500000  | 82.500000  |
| 50%   | 88.000000  | 88.000000  |
| 75%   | 93.500000  | 97.500000  |
| max   | 100.000000 | 100.000000 |

*Congratulations! You've completed the first problem-solving notebook!*