Tutorial Directions

In this tutorial, you will learn how to use a Jupyter notebook feature called **gathering**. Gathering helps you clean and review your exploratory code in a notebook.

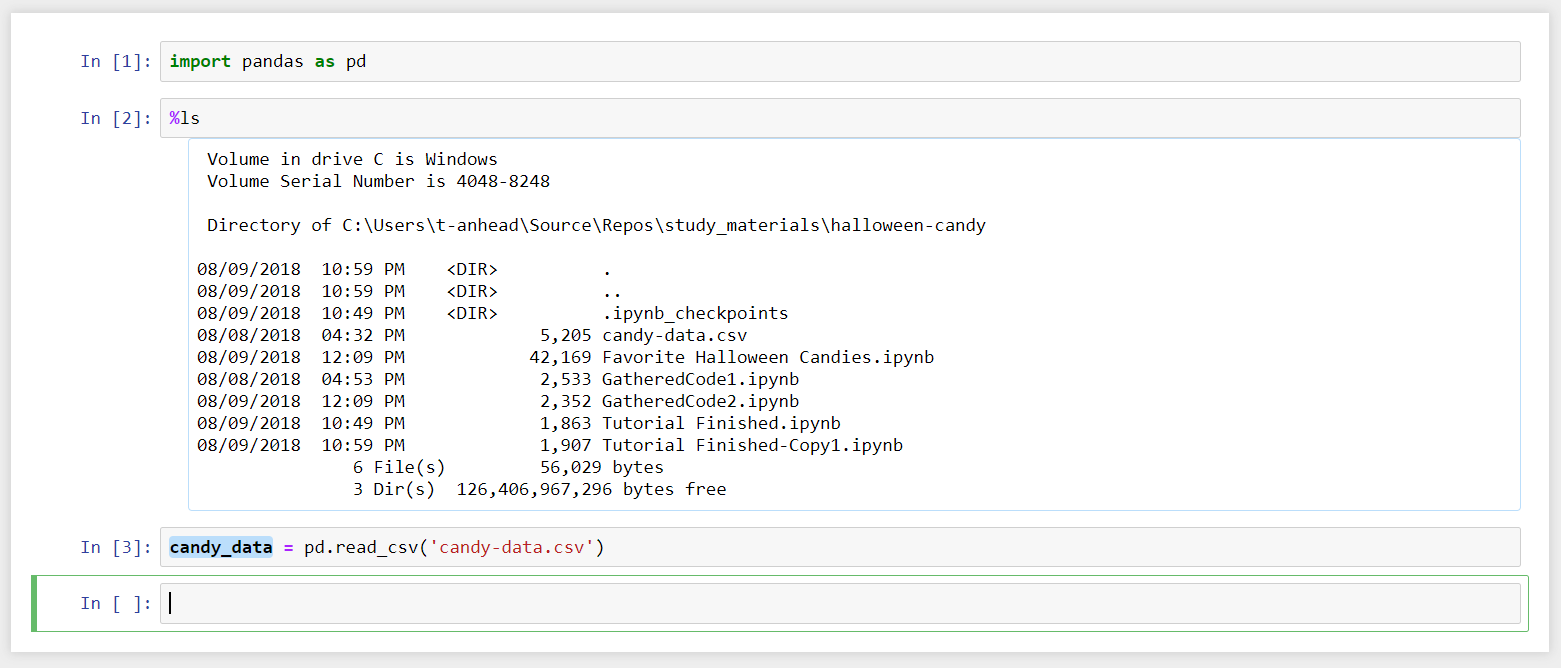
If you have any questions about the instructions, please ask the study facilitator.

# Preparation: Execute the notebook

Before gathering code in a notebook, you first need to run the code. Ask the experimenter to open a test notebook for you. Then run all of the cells.

**Task**: Run all the cells in the notebook.

After running the cells, the notebook should look something like this:



# Step 1: Gathering code that computed a variable

See that candy\_data variable? How was it loaded? You can gather the code that computes this variable by (1) clicking on the variable and (2) clicking one of the gather buttons.

You cangather the code that initializes candy\_data by first clicking on the variable:



... and then clicking on one of the gather buttons (highlighted in blue)



**Task:** Gather to the clipboard, then paste the cells. What code was copied? What code was left out?

**Task:** Gather to a notebook. What happened? Where is your original code?

## Tip: Blue lines show how a variable was computed

When you click a blue variable, all of the lines highlighted in blue are the lines that had to run to compute that variable. This could help you find relevant code scattered across a notebook.

# Step 2: Gathering code that produced an output

You can also gather code that produces any output in the notebook. See the light blue border surrounding one of the table outputs? This mean that output can be gathered.

**Task**: Gather the code that produces one of the output tables by clicking on it, and then clicking one of the gather buttons. What code does it produce? Does it preserve the table when you gathered code?

## Tip: Gather code that produces multiple outputs or variables

Want to gather the code that generates multiple outputs or variables all at once? Hold down **Shift** while you click on those variables or outputs before clicking one of the gather buttons.

**Task**: Gather the code that computes both output tables.

# Step 3: Gather to clean up a messy notebook

When programming in a notebook, that notebook can become a mess. Gathering can help you reduce the mess by helping you straighten out your code and recover code you deleted.

**Task**: Ask the study facilitator to mess up your Jupyter notebook. Then gather code that produced one of the output tables. What code does it collect? Can you run it to reproduce that table?

## Tip: Gather code for a variable that was deleted from the notebook

Did you delete a variable definition that was relevant? Have you restarted the kernel yet? (1) Make a new cell (2) Type the name of the variable (3) Run the cell and (4)

**Task**: Ask the study facilitator to delete a variable definition in the notebook and try out this trick to recover the code that computes a variable.

# Step 4: Review an output’s history

You can also use gathering to recover and compare versions of an output.

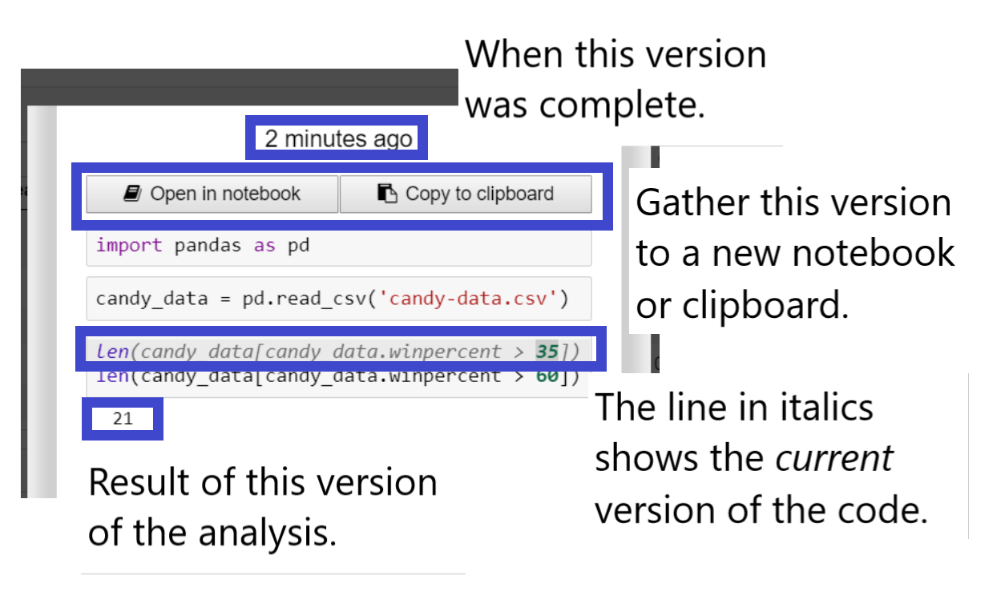
**Task**: The study facilitator will make a few variations to an existing cell. Click on the output from that cell, and then click on the **Revisions** button. What do you see? Where is the current version of the code? Where are the past versions of the code? How many revisions are there?

**Task**: What happens when you click one of the buttons to open a revision in a notebook?

**Task**: What happens when you click one of the buttons to copy a revision to the clipboard?

## Tip: Comparing past and present versions of the code

In past revisions, lines that are greyed out and in italics represent the *current* version of the code, and are shown side-by-side with the code in the past revision to let you compare the revisions.



# More tips for effective gathering

## Gathering Gets More Useful Over Time

Gathering is the most useful when you’re trying to clean up or make sense of large notebooks with a lot of testing an investigations. So if you don’t find chances to use the tool right away, that’s okay. Just keep it in mind as you get deeper into writing code, want to clean up that code and compare revisions.