3.4.3   Write to Files with Python

Seth thinks this would be a good time for Tom to show you how to write data to a file. Writing data to a file requires similar steps as when we read a file, and is just as important: after you perform your analysis, you will need to write the results to a text file, which will be sent to the election commission.

To write a file to a directory on your computer, perform steps similar to those we followed when we read a file:

1. Create a filename variable to a direct or indirect path where the file is to be located.

2. Use the `open()` function in the `"w"` mode to open a file and write data to the file.

Here's how we would write this in Python.

```python
# Create a filename variable to a direct or indirect path to the file.
file_to_save = os.path.join("analysis", "election_analysis.txt")
# Using the open() function with the "w" mode we will write data to the file
open(file_to_save, "w")
```

When you add this code to the `PyPoll.py` file and run it in VS Code terminal, we'll see that this results in an error.

```
Traceback (most recent call last):
  File "PyPoll.py", line 24, in <module>
    open(file_to_save, "w")
IOError: [Errno 2] No such file or directory: 'analysis/election_analysis.tx
```

The error is an `IOError`, which is an "Input/Output" error, meaning that we used an output directory, `'analysis/election_analysis.txt'`, that doesn't exist with the given file path.

The error states that there is no file or directory `'analysis/election_analysis.txt'`. This error occurs because we don't have a folder named "analysis" where the `election_analysis.txt` file should be saved.

To correct this error, create an empty folder in the Election_Analysis folder and name it "analysis." When we execute the file again, we can open the "analysis" folder and see the `election_analysis.txt` file in the VS Code sidebar.
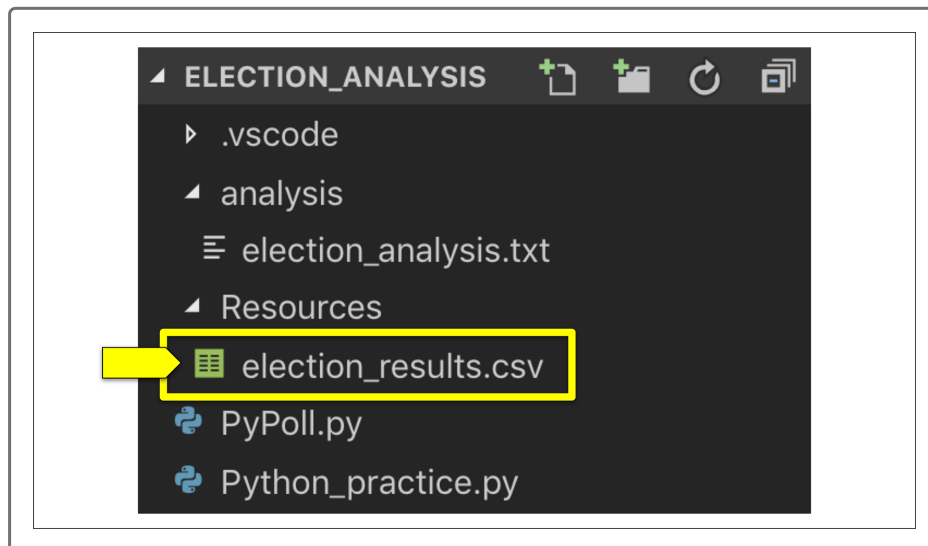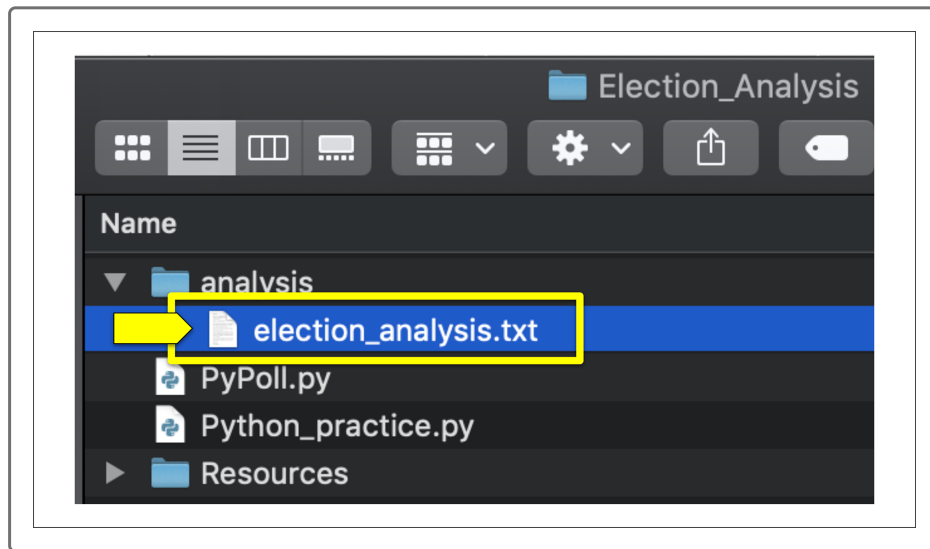
Open `election_analysis.txt` and you'll see that it's empty. As we perform the election analysis, we'll write data to this file. For now, let's practice adding some simple data to this file and saving it in the "analysis" folder.

In `election_analysis.txt` add "Hello World" to the first line by adding the following code to `PyPoll.py` and running the file in VS Code.

```
# Create a filename variable to a direct or indirect path to the file.
file_to_save = os.path.join("analysis", "election_analysis.txt")

# Use the open statement to open the file as a text file.
outfile = open(file_to_save, "w")
# Write some data to the file.
outfile.write("Hello World")

# Close the file
outfile.close()
```

Here's what's happening in this code:

1. After we create the `file_to_save` variable, we set the `open(file_to_save, "w")` to a filename variable, `outfile`.

2. Then, we use the filename variable to write "Hello World" to the file using the `write()` function from the `os` module.

3. Lastly, we use `outfile.close()` to close the file.

When we execute this file and open `election_analysis.txt`, we see the string `Hello World` in the first line.
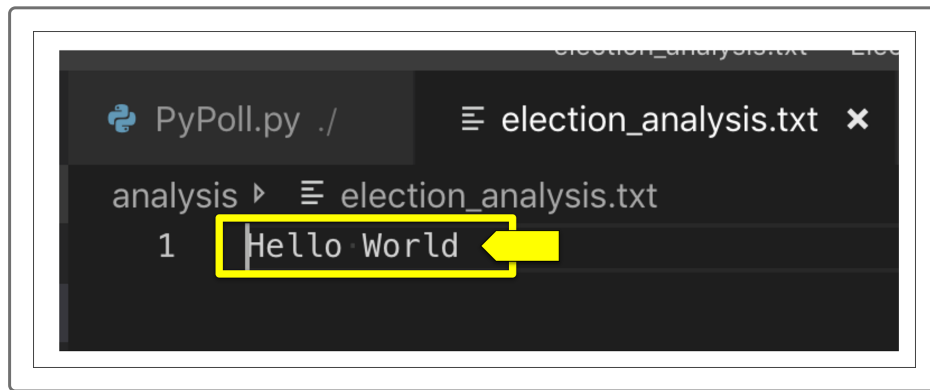
Now that we know how to write data to a file, let's make the code cleaner and more concise. We'll do this by using the `with` statement instead of using the `open()` and `close()` functions.

Your code for writing to a file should look like this:

```python
# Create a filename variable to a direct or indirect path to the file.
file_to_save = os.path.join("analysis", "election_analysis.txt")

# Using the with statement open the file as a text file.
with open(file_to_save, "w") as txt_file:

    # Write some data to the file.
    txt_file.write("Hello World")
```
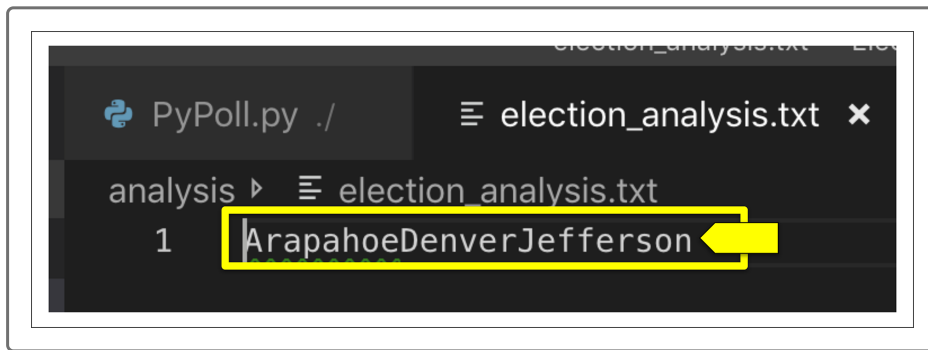
We can continue to add data to this file using the `write()` function.

In place of "Hello World," add the following counties to the file: "Arapahoe," "Denver," and "Jefferson." This can be done in two ways.

The first way is to add each county with its own `write()` function on a separate line, like this:

```python
    # Write three counties to the file.
    txt_file.write("Arapahoe")
    txt_file.write("Denver")
    txt_file.write("Jefferson")
```

When we execute the file and open `election_analysis.txt` we'll see the names of the three counties, but there is no space between each county name.
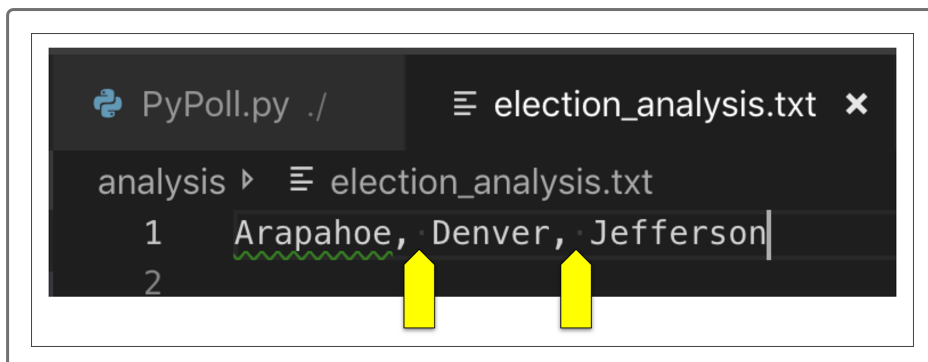
To separate "Arapahoe" and "Denver" by a comma and space, we need to add them to the end of the county name in the `write()` function as follows:

```
# Write three counties to the file.
txt_file.write("Arapahoe, ")
txt_file.write("Denver, ")
txt_file.write("Jefferson")
```

The second method is adding all three counties to one line, like this:

```
# Write three counties to the file.
txt_file.write("Arapahoe, Denver, Jefferson")
```

After editing the code and executing one of these two options, open `election_analysis.txt`. You'll see that the counties are separated by a comma and a space.

If we want to write each county on a separate line, we need to add the newline escape sequence to the end of each county name. The **newline**

escape sequence is the letter "n" preceded by the backward slash like this:
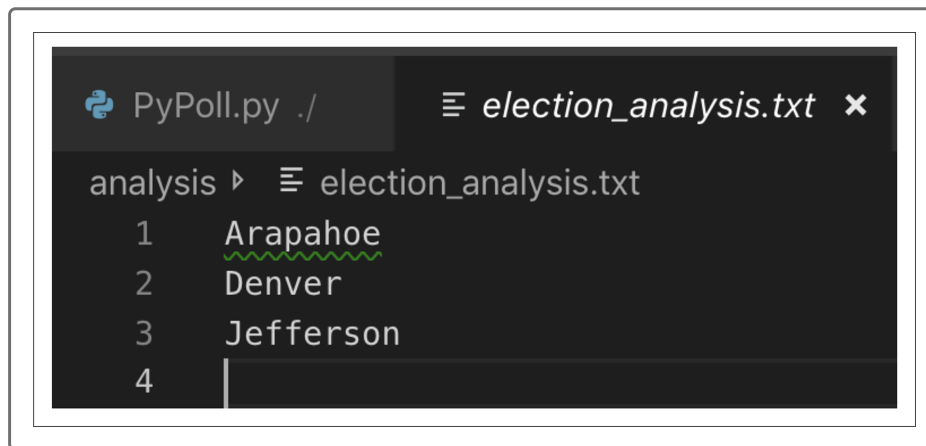`\n`.

NOTE

> The newline escape sequence will create a newline, like pressing
> "return" when it is read. Everything after the `\n` will be on the next line.

Add the newline escape sequence to the end of the first two county names
so that the code looks like this:

```python
# Write three counties to the file.
txt_file.write("Arapahoe\nDenver\nJefferson")
```

Edit the code and execute the `PyPoll.py` file. Then open
`election_analysis.txt` to see that the counties on separate lines.

```
PyPoll.py ./          ≡ election_analysis.txt ✕

analysis ▷  ≡ election_analysis.txt
  1      Arapahoe
  2      Denver
  3      Jefferson
  4      |
```

SKILL DRILL

Modify your code so that the output file looks like
this:

```
analysis  ▷  ≡ election_analysis.txt
    1     Counties·in·the·Election
    2     ─────────────────────────
    3     Arapahoe
    4     Denver
    5     Jefferson
```

Congratulations—you now know how to read a CSV file and write to a text file using Python! This is a huge step because accessing data on a file and writing data to a file are common tasks that programmers and analysts perform.