Penn Libraries
UNIVERSITY of PENNSYLVANIA

PYTHON FOR LIBRARIANS

July 18, 2023

Stephen Hall, Computer Science and Engineering Librarian
Cynthia Heider, Public Digital Scholarship Librarian

Follow along with links at: **bit.ly/py-librarians**

WHAT IS PYTHON?

Python is a free, popular, open-source programming language that has lots of uses for the humanities.

Our workshop learning goals
Participants can expect to be able to…

- set up a Python coding environment suitable for use with their own computer.
- recognize potential use cases for Python and its libraries as humanistic research tools.
- demonstrate familiarity with fundamental Python syntax and concepts such as variables, data types, lists, and conditionals.
- identify available resources for practicing skills and creating projects with Python.

ADVANTAGES TO USING PYTHON AS A LIBRARIAN

- Open source
- Has stood the test of time
- Lots of support
- Available for everyone
- "General purpose"
- Human-friendly
- Easy to learn
- Automate the boring stuff
- Makes tasks much faster

Back to Overview

THINGS LIBRARIANS CAN DO WITH PYTHON

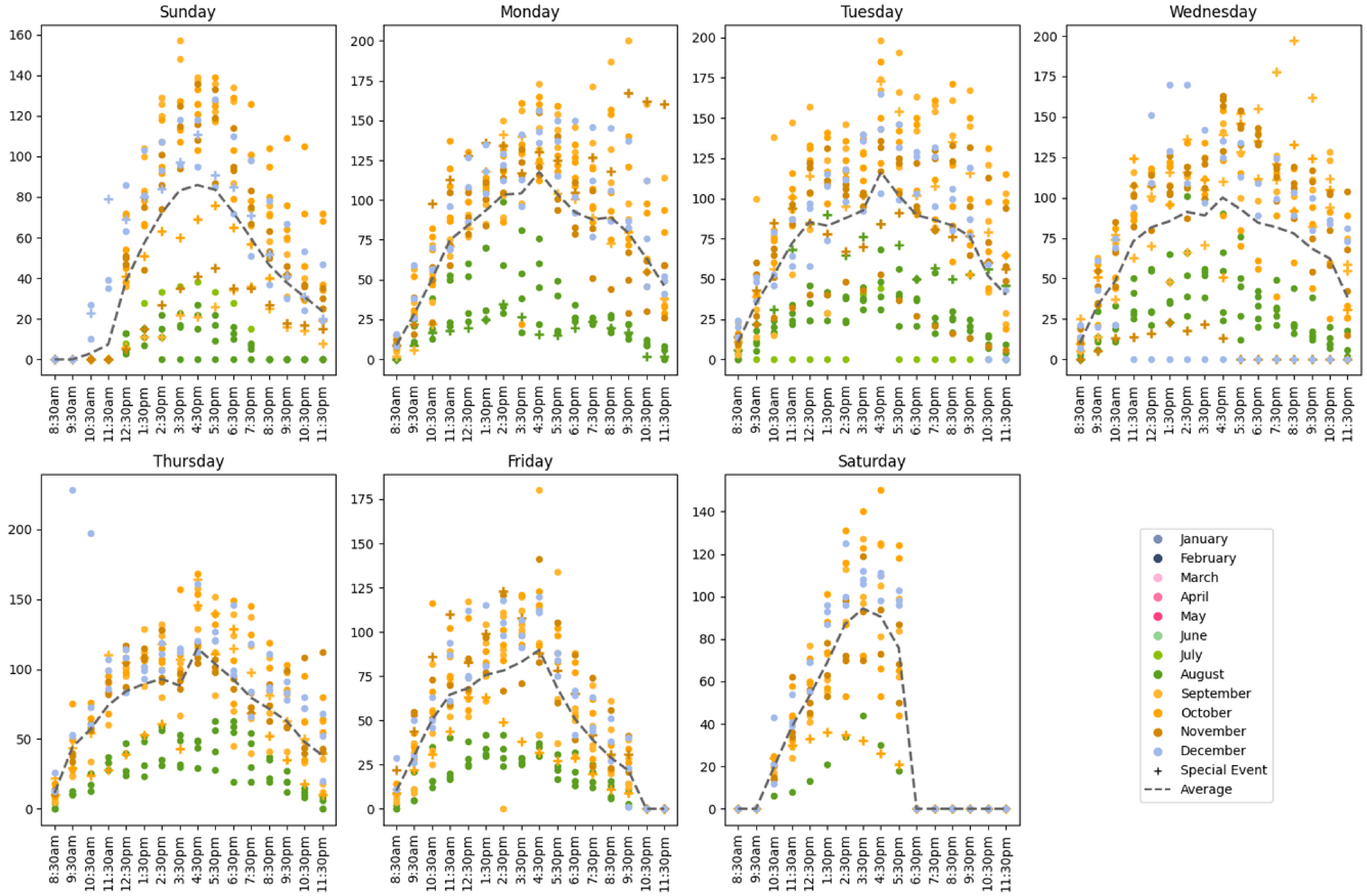
- Gather information
- Parse documents and pulling out relevant information
- Explore patterns
- Format conversion
- Visualizing data in charts, graphs, infographics, maps, and more
- Building interfaces to work with source materials and data
- Math - it does any calculations for you :)
- Your idea here!

REAL LIFE EXAMPLES

- Comparing lists of ISSNs
- Scraping subject headings for journals
- Planning for parallel book shifting projects
- Calculating usage statistics
- ILS reporting
- Analyzing metadata quality
- Converting between MARC, XML, and BIBFRAME

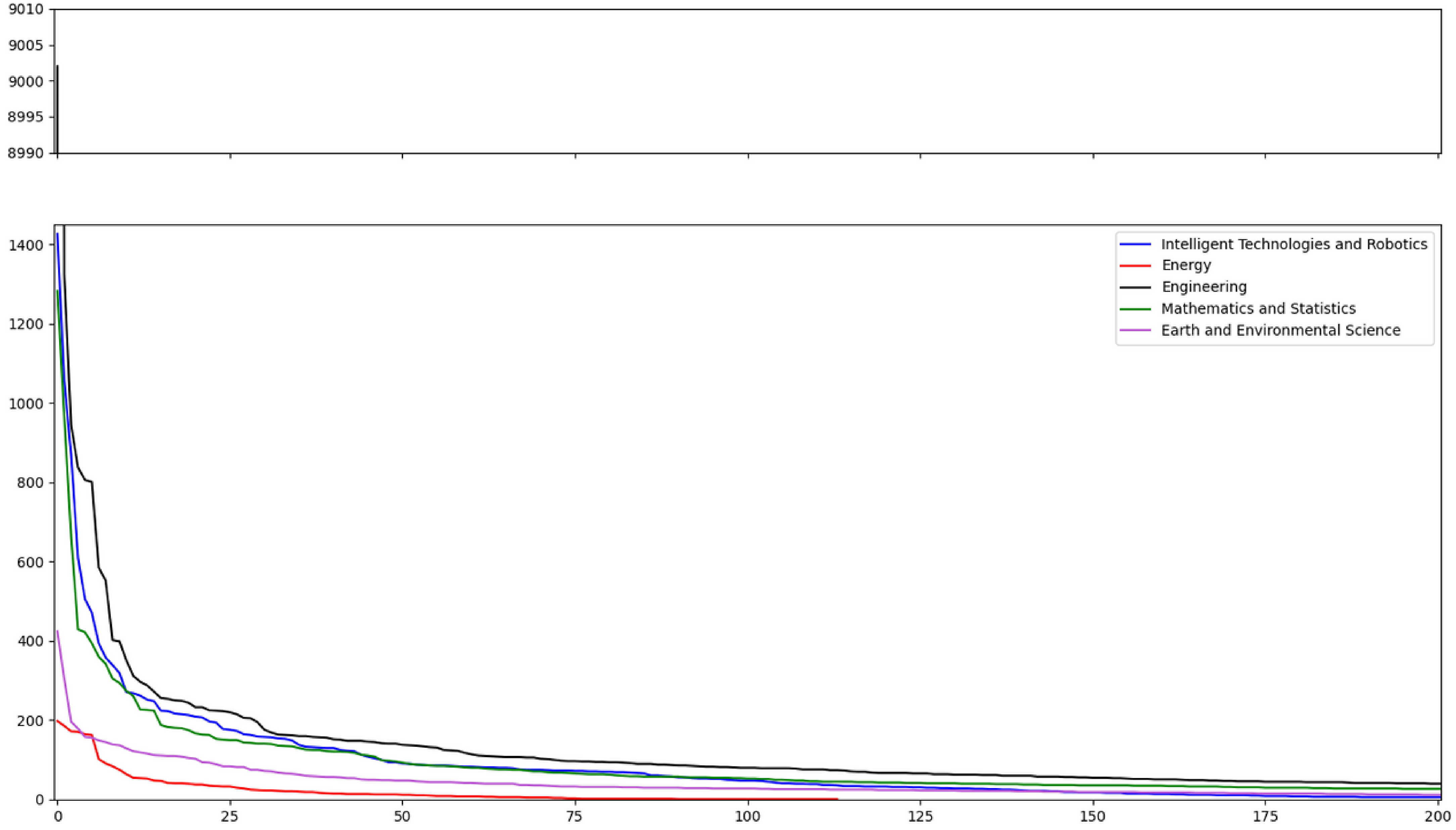
REAL LIFE EXAMPLE

Visualizing patron traffic at the Holman Biotech Commons



REAL LIFE EXAMPLE

Visualizing Springer titles checkout frequency from 2020-2023



GETTING STARTED

Python is an *interpreted* language.

Go to https://colab.google

"Open Colab"

Sign in with your Google account

GitHub tab

Enter https://github.com/smhall817/codingprimer.git

Back to Overview

GETTING STARTED

Python is an *interpreted* language.

Downloadable interpreters and Integrated Development Environments (IDEs)

Web-based IDEs

Interactive notebooks

Back to Overview

YOUR FIRST PROGRAM: SHOWING YOUR WORK WITH THE 'PRINT' FUNCTION

Use print to display values

- Python has a built-in function called print that prints things as text.
- Call the function (i.e., tell Python to run it) by using its name.
- Provide values to the function (i.e., the things to print) in parentheses.
- To add a string to the printout, wrap the string in single quotations.
- The values passed to the function are called ‘arguments’

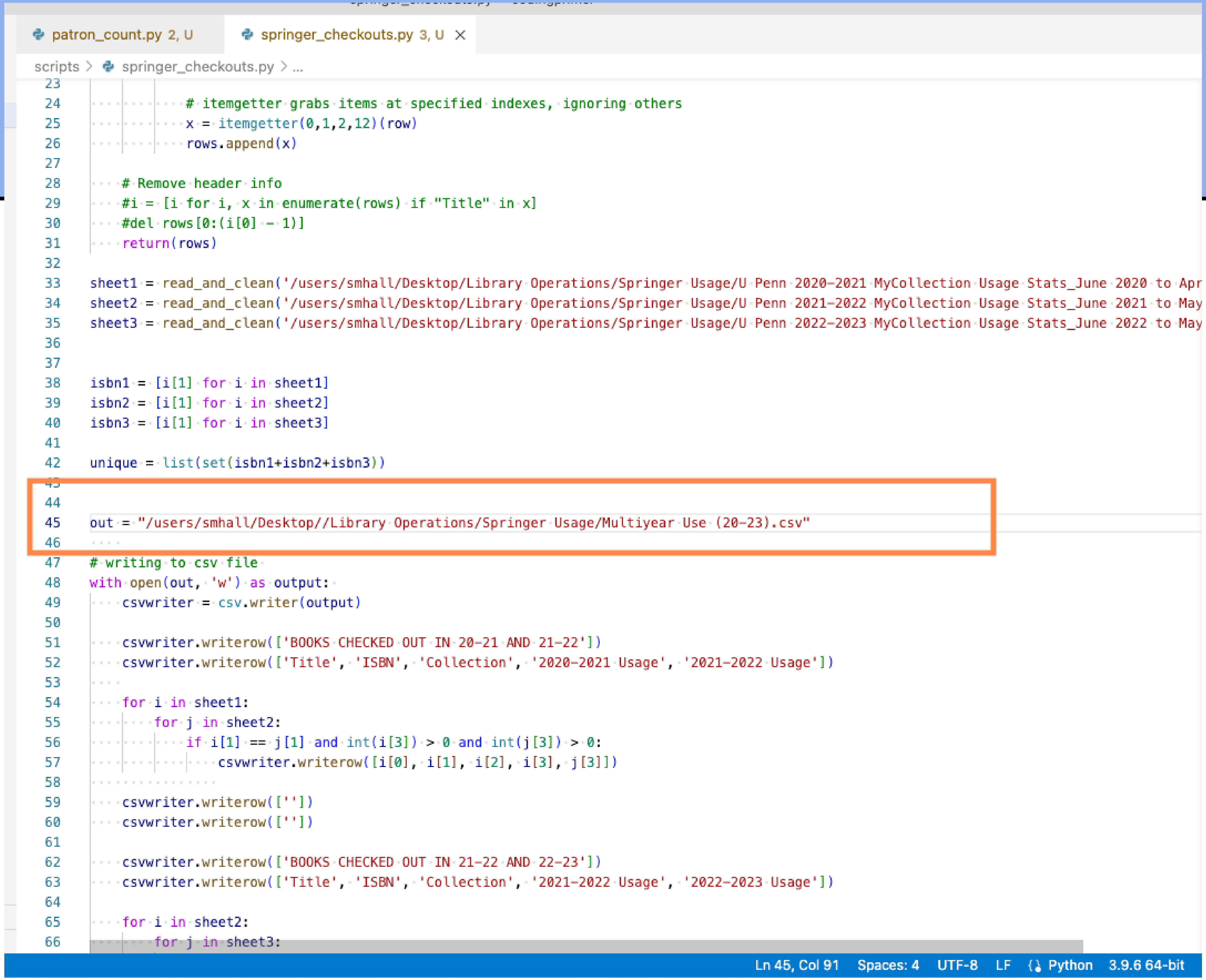
Example

```
>>>print(first_name, ‘is’, age, ‘years old’)
```

DATA TYPES

- booleans - statements that are either "True" or "False"
- integers - e.g. "4", "-17", "19054", etc.
- floats (decimals) - e.g. "3.14", "98.6", etc.
- lists (lists of items saved to a single variable) - "['apples', 'bananas', 'oranges']"
- tuples (similar to lists, but can't be modified) - "(name, SSN, phone_number, address)"
- dictionaries (used for data in key-value pairs) - "my_car = {'make': 'Honda', 'model': 'Accord', 'year': 2020, 'mpg': 32.6}"

VARIABLES IN ACTION


patron_count.py 2, U
springer_checkouts.py 3, U ×
scripts > springer_checkouts.py > ...
Ln 45, Col 91 Spaces: 4 UTF-8 LF {} Python 3.9.6 64-bit

```

            # itemgetter grabs items at specified indexes, ignoring others
            x = itemgetter(0,1,2,12)(row)
            rows.append(x)

    # Remove header info
    #i = [i for i, x in enumerate(rows) if "Title" in x]
    #del rows[0:(i[0] - 1)]
    return(rows)

sheet1 = read_and_clean('/users/smhall/Desktop/Library Operations/Springer Usage/U Penn 2020-2021 MyCollection Usage Stats_June 2020 to Apr
sheet2 = read_and_clean('/users/smhall/Desktop/Library Operations/Springer Usage/U Penn 2021-2022 MyCollection Usage Stats_June 2021 to May
sheet3 = read_and_clean('/users/smhall/Desktop/Library Operations/Springer Usage/U Penn 2022-2023 MyCollection Usage Stats_June 2022 to May


isbn1 = [i[1] for i in sheet1]
isbn2 = [i[1] for i in sheet2]
isbn3 = [i[1] for i in sheet3]

unique = list(set(isbn1+isbn2+isbn3))



out = "/users/smhall/Desktop//Library Operations/Springer Usage/Multiyear Use (20-23).csv"
    
# writing to csv file
with open(out, 'w') as output:
    csvwriter = csv.writer(output)

    csvwriter.writerow(['BOOKS CHECKED OUT IN 20-21 AND 21-22'])
    csvwriter.writerow(['Title', 'ISBN', 'Collection', '2020-2021 Usage', '2021-2022 Usage'])
    
    for i in sheet1:
        for j in sheet2:
            if i[1] == j[1] and int(i[3]) > 0 and int(j[3]) > 0:
                csvwriter.writerow([i[0], i[1], i[2], i[3], j[3]])
                
    csvwriter.writerow([''])
    csvwriter.writerow([''])

    csvwriter.writerow(['BOOKS CHECKED OUT IN 21-22 AND 22-23'])
    csvwriter.writerow(['Title', 'ISBN', 'Collection', '2021-2022 Usage', '2022-2023 Usage'])

    for i in sheet2:
        for j in sheet3:
```

VARIABLES

Use variables to store values.
A variable is a kind of “sticky note."

You can also use them for calculations!

Example:

```
>>> age = age + 3
>>> print('Age in three years:', age)
```

LISTS

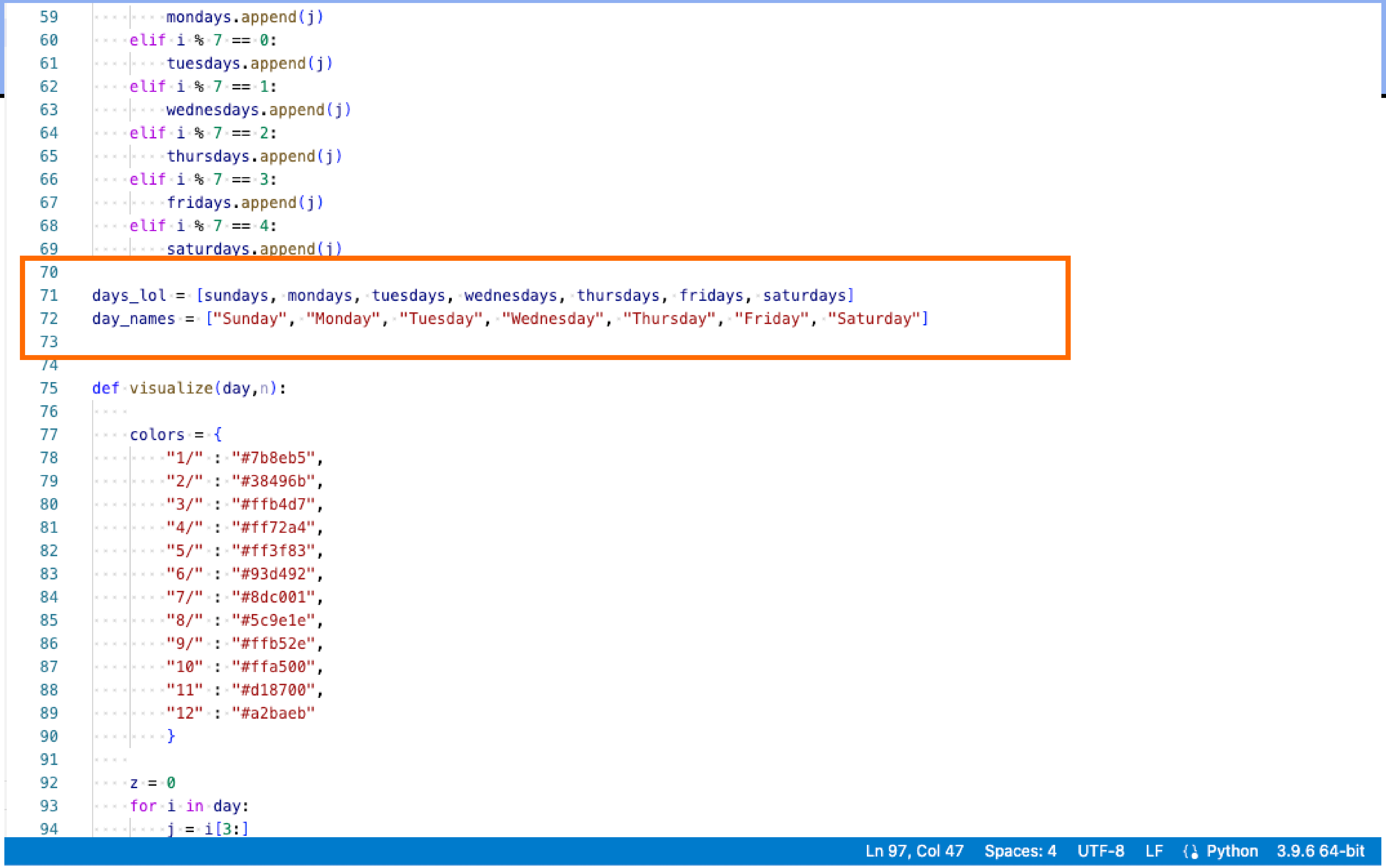
Lists are a common data structure to hold an ordered sequence of elements.

Each element can be accessed by an index. Note that Python indexes start with 0 instead of 1:

EXAMPLE:

```
>>> numbers = [1, 2, 3]
>>> numbers[0]
```

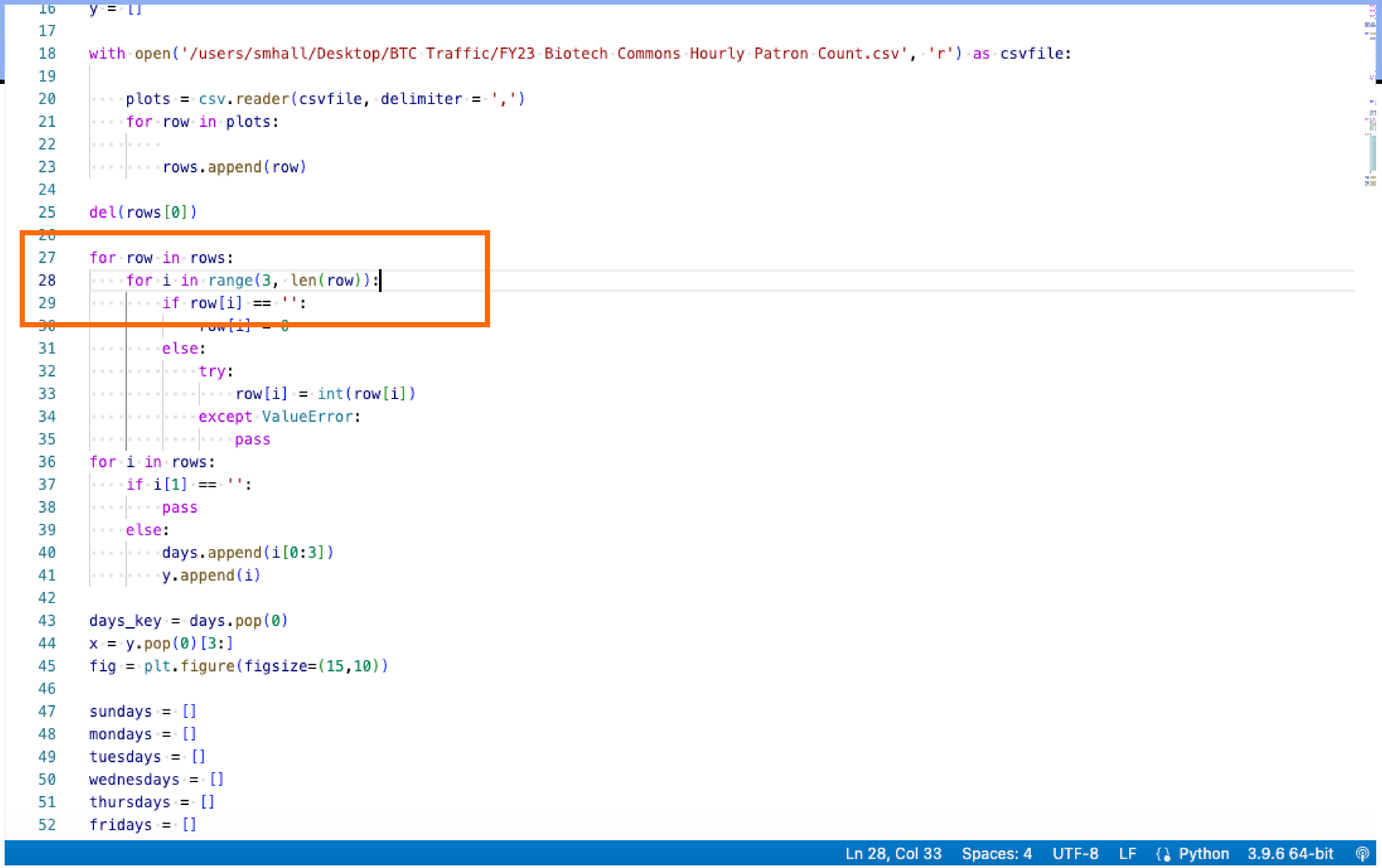
LISTS IN ACTION



```python
        mondays.append(j)
    elif i % 7 == 0:
        tuesdays.append(j)
    elif i % 7 == 1:
        wednesdays.append(j)
    elif i % 7 == 2:
        thursdays.append(j)
    elif i % 7 == 3:
        fridays.append(j)
    elif i % 7 == 4:
        saturdays.append(j)

days_lol = [sundays, mondays, tuesdays, wednesdays, thursdays, fridays, saturdays]
day_names = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]


def visualize(day,n):
    
    colors = {
        "1/" : "#7b8eb5",
        "2/" : "#38496b",
        "3/" : "#ffb4d7",
        "4/" : "#ff72a4",
        "5/" : "#ff3f83",
        "6/" : "#93d492",
        "7/" : "#8dc001",
        "8/" : "#5c9e1e",
        "9/" : "#ffb52e",
        "10" : "#ffa500",
        "11" : "#d18700",
        "12" : "#a2baeb"
        }
    
    z = 0
    for i in day:
        j = i[3:]
```

Ln 97, Col 47 Spaces: 4 UTF-8 LF {} Python 3.9.6 64-bit

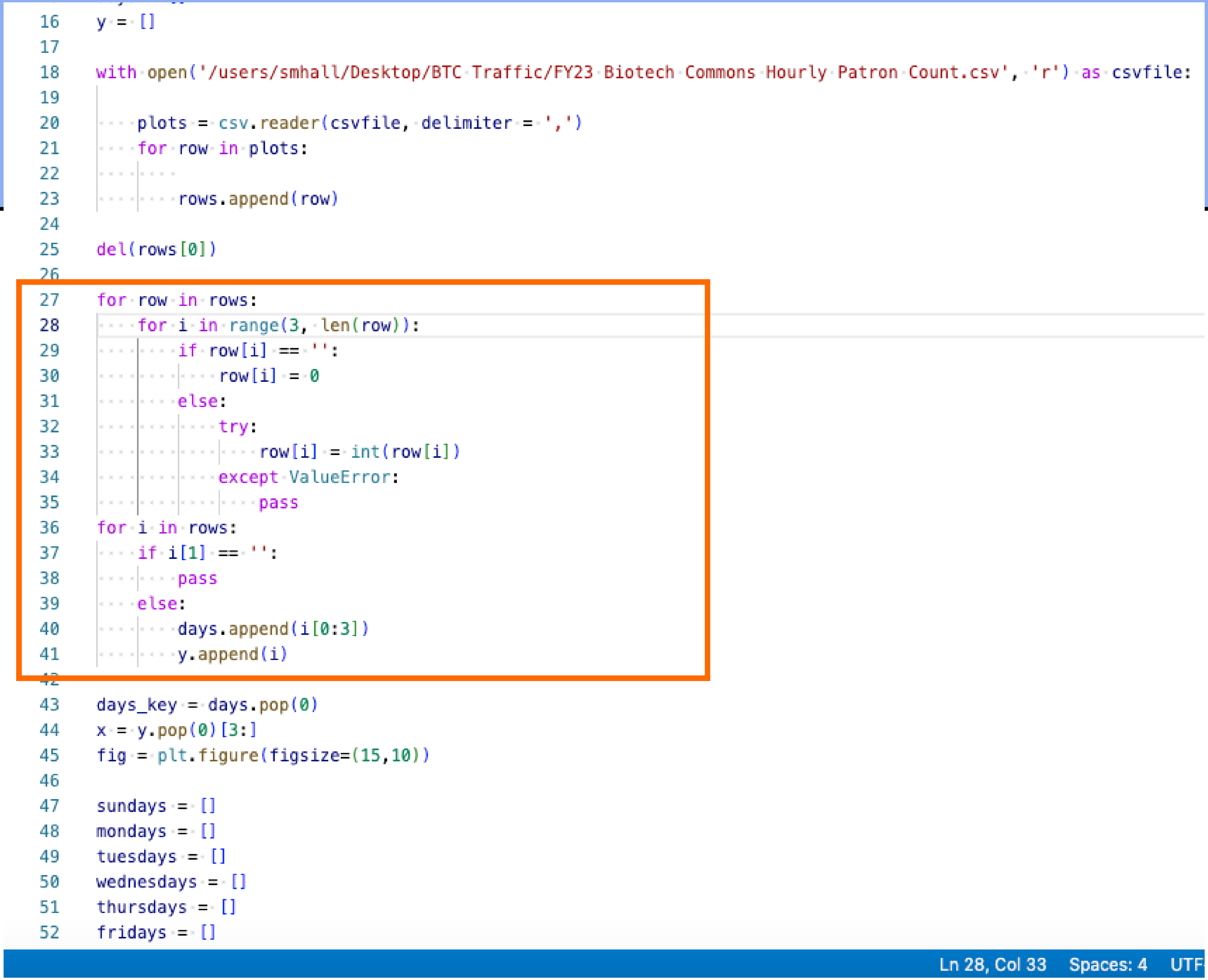
LISTS FUNCTIONS IN ACTION



```python
y = []

with open('/users/smhall/Desktop/BTC Traffic/FY23 Biotech Commons Hourly Patron Count.csv', 'r') as csvfile:

    plots = csv.reader(csvfile, delimiter = ',')
    for row in plots:

        rows.append(row)

del(rows[0])

for row in rows:
    for i in range(3, len(row)):
        if row[i] == '':
            row[i] = 0
        else:
            try:
                row[i] = int(row[i])
            except ValueError:
                pass
for i in rows:
    if i[1] == '':
        pass
    else:
        days.append(i[0:3])
        y.append(i)

days_key = days.pop(0)
x = y.pop(0)[3:]
fig = plt.figure(figsize=(15,10))

sundays = []
mondays = []
tuesdays = []
wednesdays = []
thursdays = []
fridays = []
```

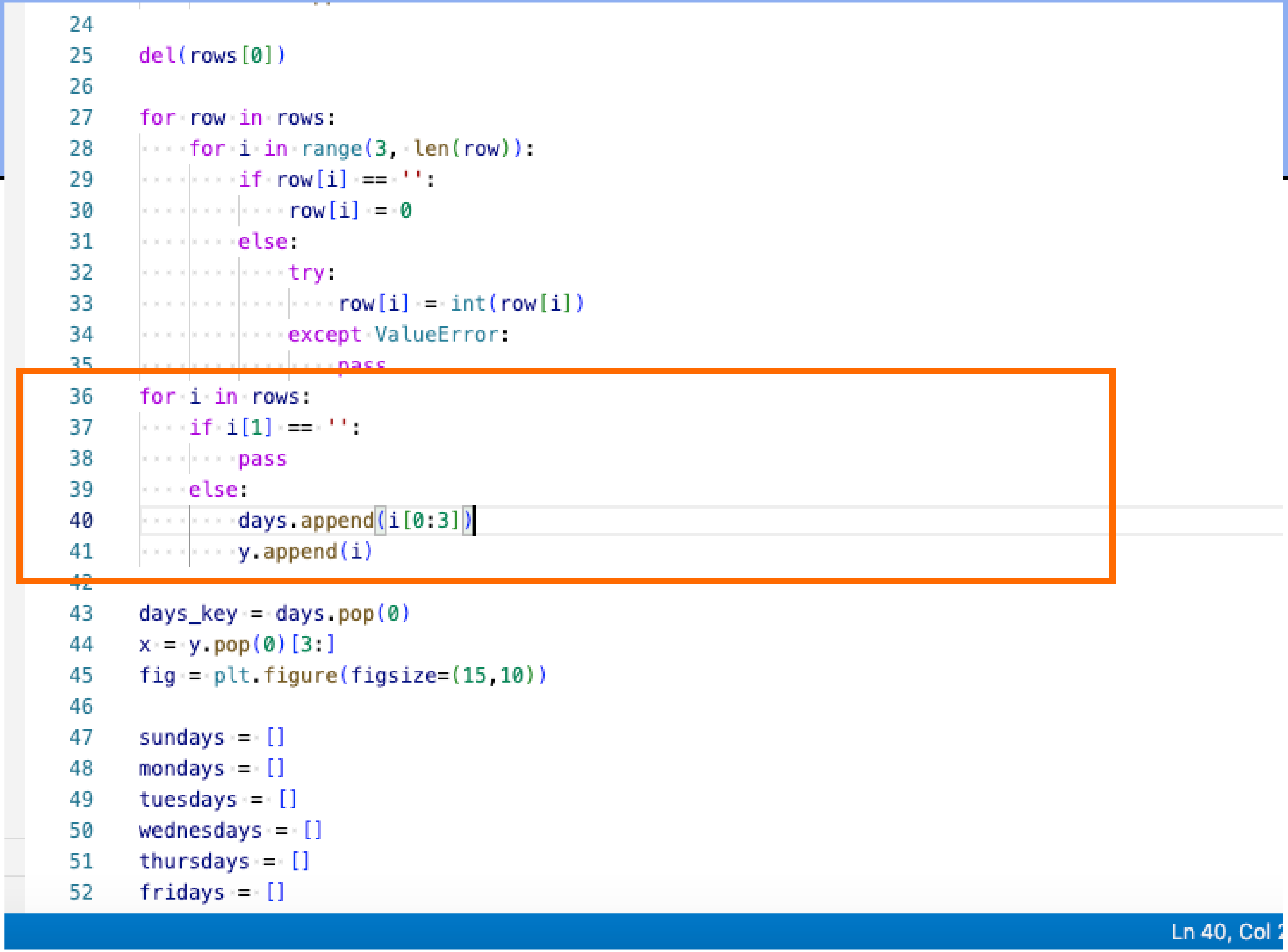
Ln 28, Col 33   Spaces: 4   UTF-8   LF   Python   3.9.6 64-bit

CONDITIONALS IN ACTION



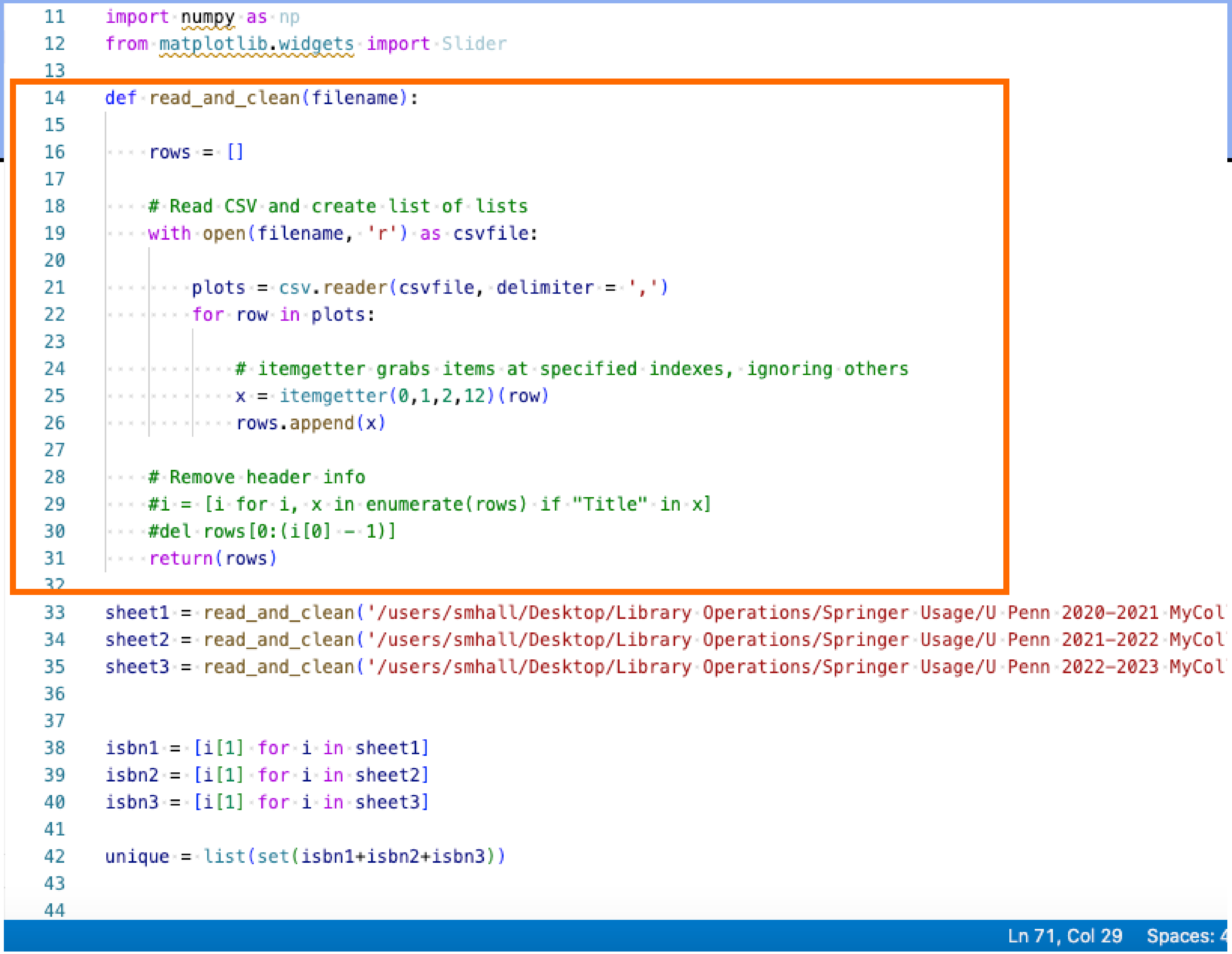
```python
y = []

with open('/users/smhall/Desktop/BTC Traffic/FY23 Biotech Commons Hourly Patron Count.csv', 'r') as csvfile:

    plots = csv.reader(csvfile, delimiter = ',')
    for row in plots:
        
        rows.append(row)

del(rows[0])

for row in rows:
    for i in range(3, len(row)):
        if row[i] == '':
            row[i] = 0
        else:
            try:
                row[i] = int(row[i])
            except ValueError:
                pass
for i in rows:
    if i[1] == '':
        pass
    else:
        days.append(i[0:3])
        y.append(i)

days_key = days.pop(0)
x = y.pop(0)[3:]
fig = plt.figure(figsize=(15,10))

sundays = []
mondays = []
tuesdays = []
wednesdays = []
thursdays = []
fridays = []
```

LOOPS IN ACTION



```python

del(rows[0])

for row in rows:
    for i in range(3, len(row)):
        if row[i] == '':
            row[i] = 0
        else:
            try:
                row[i] = int(row[i])
            except ValueError:
                pass
for i in rows:
    if i[1] == '':
        pass
    else:
        days.append(i[0:3])
        y.append(i)

days_key = days.pop(0)
x = y.pop(0)[3:]
fig = plt.figure(figsize=(15,10))

sundays = []
mondays = []
tuesdays = []
wednesdays = []
thursdays = []
fridays = []
```

FUNCTIONS IN ACTION



```python
import numpy as np
from matplotlib.widgets import Slider

def read_and_clean(filename):

    rows = []

    # Read CSV and create list of lists
    with open(filename, 'r') as csvfile:

        plots = csv.reader(csvfile, delimiter = ',')
        for row in plots:

            # itemgetter grabs items at specified indexes, ignoring others
            x = itemgetter(0,1,2,12)(row)
            rows.append(x)

    # Remove header info
    #i = [i for i, x in enumerate(rows) if "Title" in x]
    #del rows[0:(i[0] - 1)]
    return(rows)

sheet1 = read_and_clean('/users/smhall/Desktop/Library Operations/Springer Usage/U Penn 2020-2021 MyCol
sheet2 = read_and_clean('/users/smhall/Desktop/Library Operations/Springer Usage/U Penn 2021-2022 MyCol
sheet3 = read_and_clean('/users/smhall/Desktop/Library Operations/Springer Usage/U Penn 2022-2023 MyCol


isbn1 = [i[1] for i in sheet1]
isbn2 = [i[1] for i in sheet2]
isbn3 = [i[1] for i in sheet3]

unique = list(set(isbn1+isbn2+isbn3))
```