Python in 10 minutes

Part 5

Dr. Mark Williamson

Purpose:

- Quick, bite-size guides to basic usage and tasks in Python
- I’m no expert, I’ve just used it for various tasks, and it has made my life easier and allowed me to do things I couldn’t manually
- I’d like to share that working knowledge with you

Lesson 5: Extracting data

Last time, we learned how to split a large dataset into equal sized chunks and into a subset based on a specific criteria. Today, we'll look at additional ways to pull out specific data. We'll extract 1) a single variable into a list, 2) a pair of variables into a dictionary, and 3) whole lines into a new file.

Lesson 5: The Dataset in Question

County level Brain Cancer Incidence Rates from the NIH state cancer profiles

- All Races, Males, 50+, All Stages, Latest 5-year average
- Age-Adjusted Incidence Rate, cases per 100,000
- Asterisk indicates data that is not available (suppressed due to low counts)
- Cleaned up from raw csv file

- Available at: https://med.und.edu/daccota/files/docs/berdc_docs/county_level_brain_cancer_incidence.csv

First twenty entries

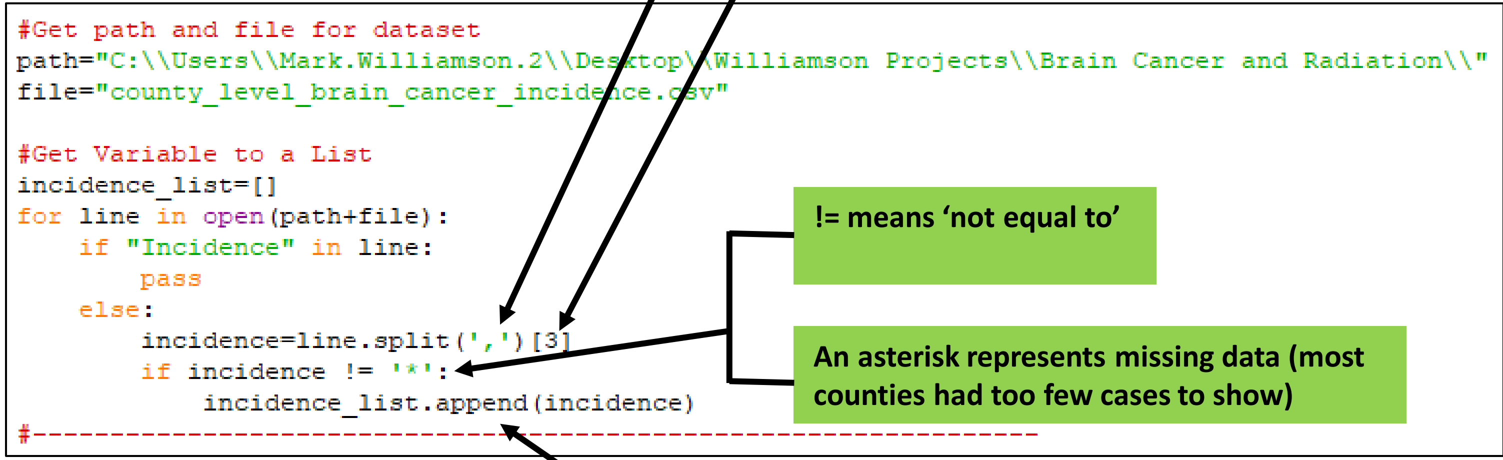
County	State	FIPS	Incidence	LCI	UCI
Autauga County	Alabama	1001	*	*	*
Baldwin County	Alabama	1003	19.1	13.3	26.6
Barbour County	Alabama	1005	*	*	*
Bibb County	Alabama	1007	*	*	*
Blount County	Alabama	1009	*	*	*
Bullock County	Alabama	1011	*	*	*
Butler County	Alabama	1013	*	*	*
Calhoun County	Alabama	1015	*	*	*
Chambers County	Alabama	1017	*	*	*
Cherokee County	Alabama	1019	*	*	*
Chilton County	Alabama	1021	*	*	*
Choctaw County	Alabama	1023	*	*	*
Clarke County	Alabama	1025	*	*	*
Clay County	Alabama	1027	*	*	*
Cleburne County	Alabama	1029	*	*	*
Coffee County	Alabama	1031	*	*	*
Colbert County	Alabama	1033	33.7	19.1	55.1
Conecuh County	Alabama	1035	*	*	*
Coosa County	Alabama	1037	*	*	*
Covington County	Alabama	1039	*	*	*

Lesson 5: Variable to a List

Goal: Pull out brain cancer incidence rates into a list

Procedure

- Download the dataset
- Open Python and start a new file
- Create a **path** and **file** variable
- Create an empty list called **incidence_list** (set it equal to empty square brackets)
- Create a for-loop for each line
- Create an if-else statement that checks if "Incidence" is in the line and passes if true (skips the first line, which is the column headers)
- Else create an **incidence** variable by splitting the 4th variable of the line by a comma
- Create an if statement that checks if incidence is **NOT** an asterisk (*) and then appends **incidence** to the **incidence_list** if that is the case



```
#Get path and file for dataset
path="C:\\Users\\Mark.Williamson.2\\Desktop\\Williamson Projects\\Brain Cancer and Radiation\\"
file="county_level_brain_cancer_incidence.csv"

#Get Variable to a List
incidence_list=[]
for line in open(path+file):
    if "Incidence" in line:
        pass
    else:
        incidence=line.split(',')[3]
        if incidence != '*':
            incidence_list.append(incidence)
#-----------------------------------------------------------------
```

Since it is a comma separated values (CSV) file, each entry in a row is separated by a comma

Need to use [3] rather than [4] because in Python, iterations start at 0 rather than 1

!= means 'not equal to'

An asterisk represents missing data (most counties had too few cases to show)

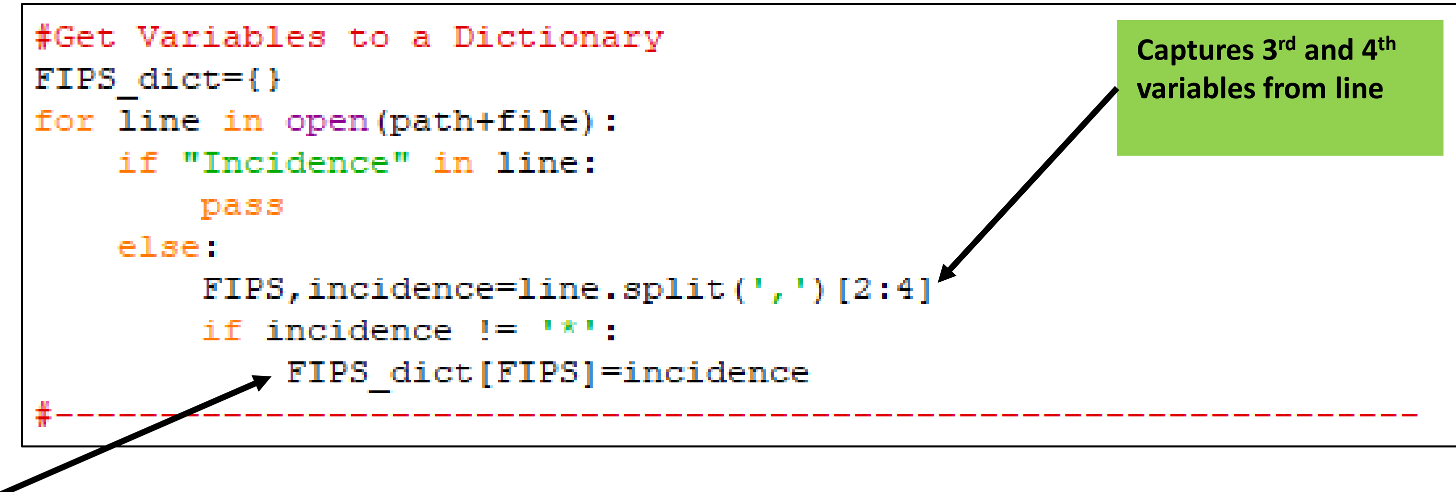
Lists can be added to using LIST.append(VARIABLE)

Lesson 5: Variables to a Dictionary

Goal: Create a dictionary that links county FIPS codes to brain cancer incidence

Procedure

- Create an empty dictionary called **FIPS_dict** (set it equal to empty curly brackets)
- Create a for-loop for each line
- Create an if-else statement that checks if “Incidence” is in the line and passes if true
- Else create a **FIPS** and **incidence** variable by splitting the 3rd and 4th variables of the line by a comma
- Create an if-statement that checks if incidence is **NOT** an asterisk (*) and then sets **FIPS** as the ***key*** and **incidence** as the ***value*** in the **FIPS_dict**



```
#Get Variables to a Dictionary
FIPS_dict={}
for line in open(path+file):
    if "Incidence" in line:
        pass
    else:
        FIPS,incidence=line.split(',')[2:4]
        if incidence != '*':
            FIPS_dict[FIPS]=incidence
#-------------------------------------------------------------------
```

Captures 3rd and 4th variables from line

Dictionary[key] = pair

Exploration

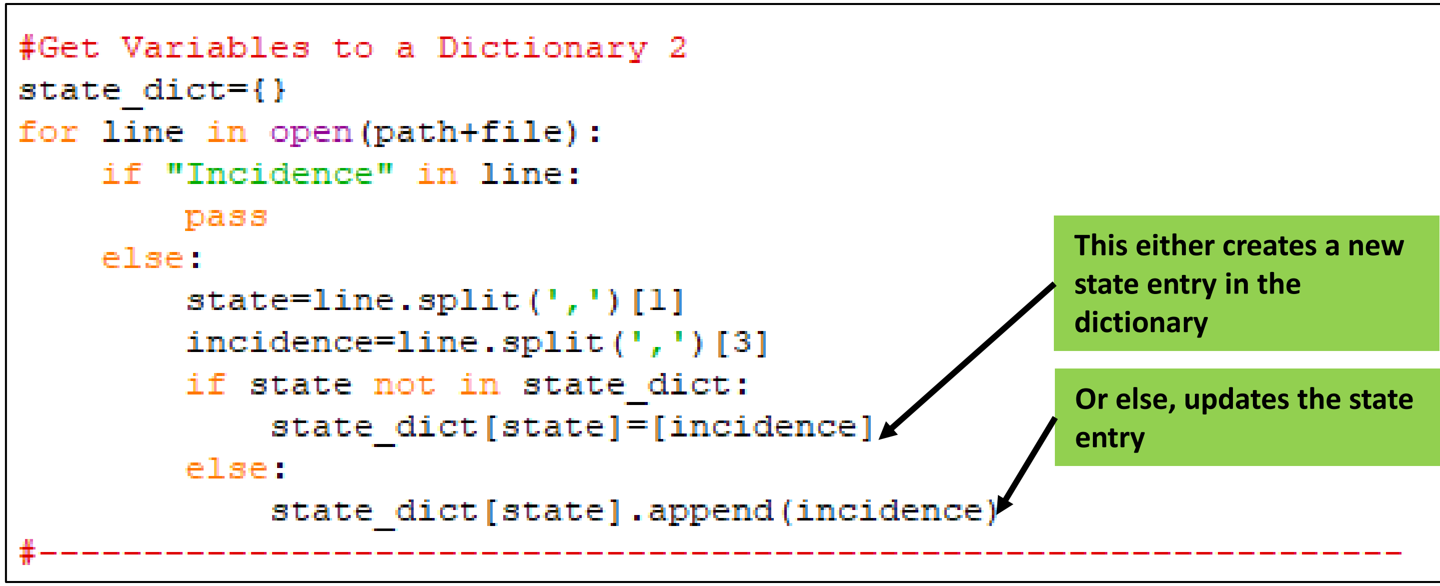
see if your county is in the dictionary by typing FIPS_dict[FIPS] using your county’s FIPS number

Lesson 5: Variables to a Dictionary 2

Goal: Create a dictionary that links state name with a list of all county brain cancer incidences (missing or not)

Procedure

- Create an empty dictionary called **state_dict** (set it equal to empty curly brackets)
- Create a for-loop for each line
- Create an if-else statement that checks if “Incidence” is in the line and passes if true
- Else create a **state** and **incidence** variable by splitting the 2nd and 4th variables of the line by a comma
- Create an if-else statement to check if the state is **NOT** in the **state_dict**
  - If true (state not in dictionary), add **state** to dictionary as the ***key*** with the ***value*** being a list with one entry, the **incidence**
  - If false, append the list stored in that state’s dictionary entry with the next **incidence** value



```
#Get Variables to a Dictionary 2
state_dict={}
for line in open(path+file):
    if "Incidence" in line:
        pass
    else:
        state=line.split(',')[1]
        incidence=line.split(',')[3]
        if state not in state_dict:
            state_dict[state]=[incidence]
        else:
            state_dict[state].append(incidence)
#---------------------------------------------------------------------
```

This either creates a new state entry in the dictionary

Or else, updates the state entry

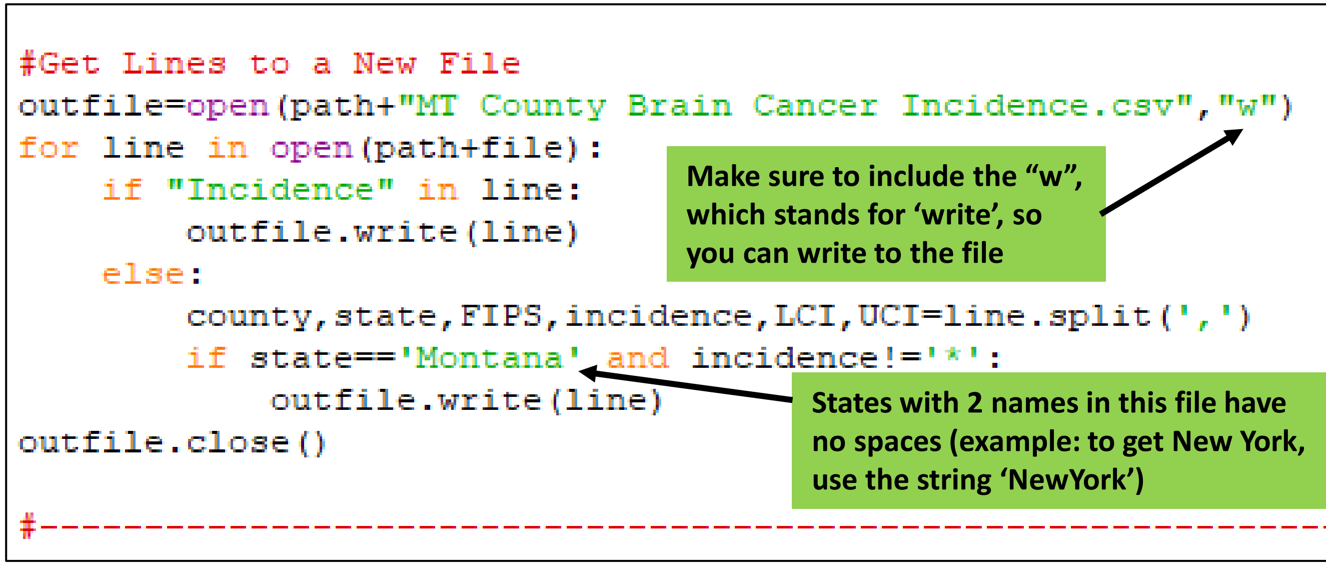
Print out the incidence list for your state

Lesson 5: Lines to a New File

Goal: Create a new file for a single state with only non-missing incidence data

Procedure

- Create a variable called **outfile** that open to a new file to your path
    - Use a state of your choice and include the initials in the file name
    - This example uses Montana (MT)
- Create a for-loop for each line
- Create an if-else statement that checks if "Incidence" is in the line and writes that line to **outfile** if true
- Else split the line by comma to the six variables of **county**, **state**, **FIPS**, **incidence**, **LCI**, and **UCI**
- Inside the first if-else statement, create an if-statement that checks to see if the state is the state you've chosen, and the incidence is **NO**T missing (*) and writes the line to the file if both are true
- Close **outfile**



```
#Get Lines to a New File
outfile=open(path+"MT County Brain Cancer Incidence.csv","w")
for line in open(path+file):
    if "Incidence" in line:
        outfile.write(line)
    else:
        county,state,FIPS,incidence,LCI,UCI=line.split(',')
        if state=='Montana' and incidence!='*':
            outfile.write(line)
outfile.close()

#----------------------------------------------------------------------
```

Make sure to include the "w", which stands for 'write', so you can write to the file

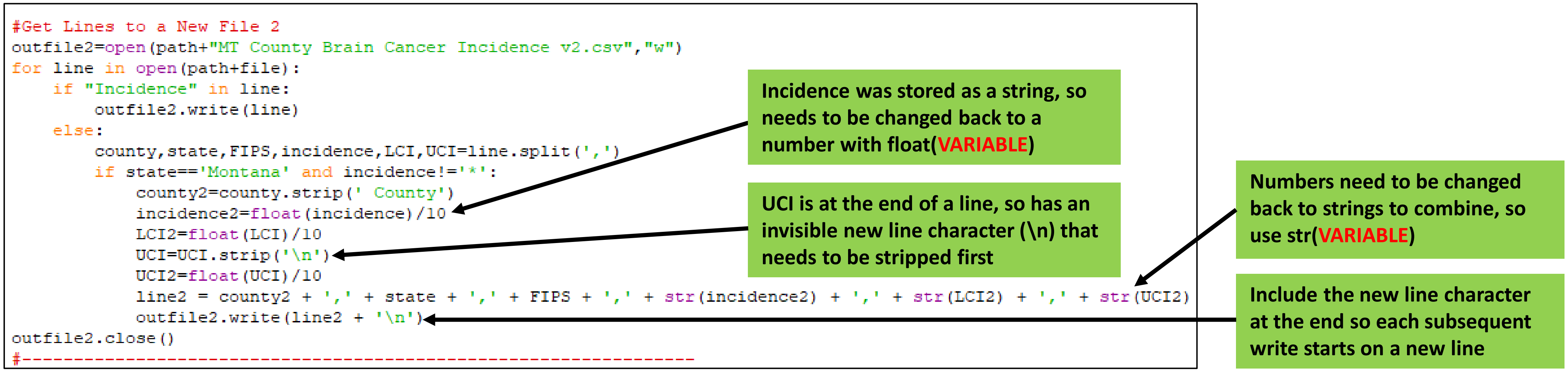
States with 2 names in this file have no spaces (example: to get New York, use the string 'NewYork')

Lesson 5: Lines to a New File 2

Goal: Create a new file for a single state with only non-missing incidence data and modified data

Procedure

- Create another outfile (**outfile2**) and write the first line (contains “Incidence”) to it
- For all other lines, split the line by comma
- Create an if-statement to check for state and non-missing data
- Create a **county2** variable that strips the unneeded ‘ County’ from **county**
- Create an **incidence2** variable that changes divides **incidence** by 10 to get incidence per million (original incidence is per 100,000, so dividing by ten turns it into per 1,000,000)
- Divide LCI and UCI (confidence intervals) by ten as well
- Create a **line2** variable and put all the updated variables together in a string separated by commas and then write line2 to the **outfile2**



```
#Get Lines to a New File 2
outfile2=open(path+"MT County Brain Cancer Incidence v2.csv","w")
for line in open(path+file):
    if "Incidence" in line:
        outfile2.write(line)
    else:
        county,state,FIPS,incidence,LCI,UCI=line.split(',')
        if state=='Montana' and incidence!='*':
            county2=county.strip(' County')
            incidence2=float(incidence)/10
            LCI2=float(LCI)/10
            UCI=UCI.strip('\n')
            UCI2=float(UCI)/10
            line2 = county2 + ',' + state + ',' + FIPS + ',' + str(incidence2) + ',' + str(LCI2) + ',' + str(UCI2)
            outfile2.write(line2 + '\n')
outfile2.close()
#------------------------------------------------------------------
```

Incidence was stored as a string, so needs to be changed back to a number with float(VARIABLE)

UCI is at the end of a line, so has an invisible new line character (\n) that needs to be stripped first

Numbers need to be changed back to strings to combine, so use str(VARIABLE)

Include the new line character at the end so each subsequent write starts on a new line

Lesson 5: Summary

- Python can quickly extract data from files
- Data can be modified and stored in a variety of useful ways, such as lists, dictionaries, and new files
- Data can be converted from number/strings to strings/numbers or edited to removed things like whitespace characters