Python in 10 minutes

Part 8

Dr. Mark Williamson, PhD
Biostatistics, Epidemiology, and Research Design Core (BERDC)
Dakota Cancer Collaborative on Translational Activity (DaCCoTA)
University of North Dakota (UND)

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 8: Transforming data

 Last time, we played around with linking different datasets together.

Today, we'll tackle data transformation. We'll examine how to:

1) transform a basic dataset from wide to long

2) transform a more complex dataset from wide to long

3) transform a basic dataset from long to wide

Lesson 8: The Dataset(s) in Question

Loblolly_wide.csv

- Dataset of 14 Loblolly pine trees with 6 columns of growth measurements; transformed from the dataset found in R

	2007	2007	2007	2008	2008	2008	.
	Lung and Bronchus	Melanoma of the Skin	Pancreas	Lung and Bronchus	Melanoma of the Skin	Pancreas	.
San Francisco-Oakland SMSA - 2000+	2297	918	527	2136	1033	547	.
Connecticut - 2000+	2692	916	558	2712	922	554	.
Detroit (Metropolitan) - 2000+	3378	636	603	3411	698	660	.
Hawaii - 2000+	794	253	199	809	302	203	.
.	.	.	.	.	.	.	.

SEER_Cancer_1.csv

- Dataset of cancer incidence counts across 21 SEER Regions, 3 cancer types, and 10 years

Seed	height.3	height.5	height.10	height.15	height.20	height.25
301	4.51	10.89	28.72	41.74	52.7	60.92
303	4.55	10.92	29.07	42.83	53.88	63.39
305	4.79	11.37	30.21	44.4	55.82	64.1
.	.	.	.	.	.	.

Cabbages_long.csv

- Dataset of 60 cabbage plant observations and 4 variables from the MASS package in R

Cult	Date	HeadWt	VitC
c39	d16	2.5	51
c39	d16	2.2	55
c39	d16	3.1	45
c39	d16	4.3	42
c39	d16	2.5	53
c39	d16	4.3	50
c39	d16	3.8	50
c39	d16	4.3	52
c39	d16	1.7	56
c39	d16	3.1	49
.	.	.	.

Lesson 8: Loading in the Data

Goal: Get data into Python

Procedure

- Download the datasets (Loblolly_wide.csv, SEER_Cancer_1.csv, Cabbages_long.csv)
- Open Python and start a new file
- Create a **path** variable
- Create **file1**, **file2**, and **file3** for each of the datasets

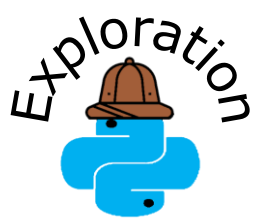
```
#Loading in the data:
path="C:\\Users\\Mark.Williamson.2\\Desktop\\Williamson Data\\Example Datasets\\"
file1="Loblolly_wide.csv"
file2="SEER_Cancer_1.csv"
file3="Cabbages_long.csv"
```

Lesson 8: Basic Wide to Long

Goal: Transform the Loblolly dataset from wide to long

Procedure

- Create a **new_headers1** variable and set it equal to a string with the headers that will be used for the transformed file
- Create **outfile1** and write the new headers to it
- Create a for-loop for each line
- Create an if-else statement that checks if “Seed” is in the line (indicates a header) and passes if true
- Else, split the line into 1 seed variable and 6 height variables (based on time)
- Create a list of the height variables and a list of the ages (strings)
- Create a for-loop for the numbers 0-6
- Write the height, age, and seed to **outfile1**, using the numbers as an index for the lists
- Close **outfile1** when done



Compare outfile1 with the actual Loblolly dataset in R. They should be the same.

```
#Example 1: Basic Wide to Long
new_headers1="height,age,Seed"
outfile1 =open(path+'Loblolly_long.csv', 'w')
outfile1.write(new_headers1 + '\n')

for line in open(path+file1):
    if 'Seed' in line:
        pass
    else:
        Seed,h3,h5,h10,h15,h20,h25=line.split(',')
        h25=h25.strip('\n')
        hgt_list=[h3,h5,h10,h15,h20,h25]
        age_list=['3','5','10','15','20','25']
        for i in [0,1,2,3,4,5]:
            outfile1.write(hgt_list[i] +','+  age_list[i] +','+ Seed + '\n')

outfile1.close()
```

Make sure to strip the new line character

Provides the list items at that index

Lesson 8: More Complex Wide to Long

Goal: Transform the SEER dataset from wide to long

Procedure

- Create a **new_headers2** variable and set it equal to a string with the headers that will be used for the transformed file
- Create **outfile2** and write the new headers to it
- Create a for-loop for each line
- Create a variable called **Region** and use an if else-statement to pass if the region length is less than 2
- Else, split the line into variables for all but the first column, then create a list of the line variables called **line_list**
- Create list of the cancer variables, **c_list**, a **Year** variable that starts at 2007 (the first year in the dataset) and an **ID** variable set to 0
- Create a for loop to go through the **line_list** where each entry is called Rate
- Set the variable **Cancer** equal to the cancer in **c_list** at the index of **ID**
- Write the Region, Cancer, Year, and Rate to **outfile2**
- Add 1 to **ID** and if it is equal to 3, reset it to 0 and add 1 to **Year**
- Close **outfile2** when done

```
#Example 2: More Complex Wide to Long
new_headers2="Region,Cancer,Year,Rate"
outfile2 =open(path+'SEER_Cancer_1_long.csv', 'w')
outfile2.write(new_headers2 + '\n')

for line in open(path+file2):
    Region=line.split(',')[0]
    if len(Region)<2:
        pass
    else:
        line_list=line.split(',')[1:-1]
        line_list[-1]=line_list[-1].strip('\n')
        c_list=['Melanoma of the Skin', 'Pancreas', 'Lung and Bronchus']
        Year=2007
        ID=0
        for Rate in line_list:
            Cancer=c_list[ID]
            outfile2.write(Region +','+ Cancer +','+ str(Year) +','+ Rate + '\n')
            ID+=1
            if ID==3:
                ID=0
                Year+=1

outfile2.close()
```

The first two lines in the first column are blank, so this statement skips over them

The bracket covers all columns except the first, which is already captured in the Region variable

Because the initial dataset is set up where the columns are the three cancer rates for each year, then on to the next year, etc., this setup with the ID and Year will capture the correct Cancer and Year categories for each of the transformed lines

Lesson 8: Basic Long to Wide

The wide dataset created here has twenty observations (10 per cultivar) with each observation having three weight variables (one for each time period). Each of the initial observations (n=60) was actually a different plant, so it is not actually 20 plants each weighed at 3 time periods, so the wide dataset here is not appropriate for actual analysis, just for the purpose of example.

Goal: Transform the cabbage dataset from long to wide

Procedure

- Create a **new_headers3** variable and set it equal to a string with the headers that will be used for the transformed file
- Create **outfile3** and write the new headers to it
- Create an empty dictionary called **cabb_dict** and a variable **n** set to 1
- Create a for-loop for each line
- Create an if-else statement that checks if "HeadWt" is in the line (indicates a header) and passes if true
- Else, split the line into **Cult**, **Date**, **HeadWt**, and **ViC** and set the variable **ID** to **n**
- Create an if-elif-else statement to check the identity of the **Date** variable and then creating a **key** variable to reflect that date
- Create a variable called **obs** and set it as a string that combines **ID** and **Cult** together
- Create an if-else statement to check if **obs** is in the **cabb_dict**
- If **obs** is not, use **obs** as the key in **cabb_dict** and the variables **key** and **HeadWt** as a key:value pair as a new nested dictionary as the value in **cabb_dict**
- If **obs** is in the dictionary, use the variables **key** and **HeadWt** as a key:value pair to add to the nested dictionary
- Add 1 to **n** and if it is equal to 11, reset it to 1

```
#Example 3: Basic Long to Wide
new_headers3="ID,Cult,Weight_16,Weight_20,Weight_21"
outfile3 =open(path+'Cabbages_wide.csv', 'w')
outfile3.write(new_headers3 + '\n')

cabb_dict={}
n=1
for line in open(path+file3):
    if "HeadWt" in line:
        pass
    else:
        Cult, Date, HeadWt, VitC =line.split(',') #won't use VitC
        ID=n
        if Date=='d16':
            key='Wt_d16'
        elif Date=='d20':
            key='Wt_d20'
        else:
            key='Wt_d21'
        obs=str(ID) + '-' + Cult
        if obs not in cabb_dict:
            cabb_dict[obs]={key:HeadWt}
        else:
            cabb_dict[obs][key]=HeadWt
        n+=1
        if n==11:
            n=1
```

Even though this variable isn't used, it can be useful to code it for potential future use

Creation of a new dictionary nested within the first dictionary

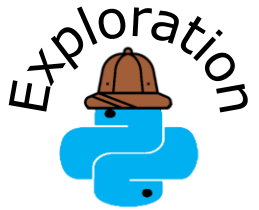
Addition of entries to the nested dictionary

Lesson 8: Basic Long to Wide cont.

Goal: Transform the cabbage dataset from long to wide

Procedure

- Create a for-loop for each item in **cabb_dict**
- Split each item (was the **obs** variable) back into its separate **ID** and **Cult**
- Set **ID** as an integer
- Create a for loop and add 10 to **ID** if the cultivar is equal to ‘c52’
- Set three different weight-day variables as the values from the nested dictionaries for each of the weight-day keys
- Set a variable called **line** to the ID, Cult, and three weight-day variables
- Write **line** to **outfile3**
- Close **outfile3** when done



Try creating another wide dataset but use VitC instead of HeadWt (header names would be something like “VitC_d16”, “VitC_d20”, and “VitC_ 21”)

```
for item in cabb_dict:
    ID,Cult=item.split('-')
    ID=int(ID)
    if Cult=='c52':
        ID+=10
    Wt_d16=cabb_dict[item]['Wt_d16']
    Wt_d20=cabb_dict[item]['Wt_d20']
    Wt_d21=cabb_dict[item]['Wt_d21']
    line=str(ID) +','+ Cult +','+ Wt_d16 +','+ Wt_d20 +','+ Wt_d21
    outfile3.write(line + '\n')

outfile3.close()
```

This sets up the ID to run from 1-20, where 1-10 are for ‘c39’ and 11-20 are for ‘c52’

Lesson 8: Summary

- Datasets can be transformed from both long to wide and from wide to long
- Wide to long is easier than long to wide
- The steps shown here are just one way to transform your data in Python; other ways exists, and many other software (R, SAS) can do it much more quickly and efficiently
- It still is edifying and useful to understand how to run transformations with Python

- Please complete a brief, 5-question assessment:

  https://und.qualtrics.com/jfe/form/SV_7ZFtFjukgWNfz0y