Supplemental File: Rename and Combine to Wide-Format

Background

This Jupyter notebook is a resource for researchers interested in bringing LIWC group communication data gathered at intervals into a wide-format for data analysis. This notebook uses the Python programming language to combine these data in a more efficient and reliable approach than manual approaches.

What is jupyter?

Jupyter is an open-source web application designed to help people create and share live programming code, equations, visualizations, and narrative text. You can read more about it at https://jupyter.org/ (https://jupyter.org/).

Why does this notebook use the Python programming language?

Python is a widely-used, open-source programming language. It is general-purpose with English-like syntax, which makes it easy to read, use, and learn. You can read more about it at https://www.python.org/ (https://www.python.org/). Python may already be installed on your computer. Installation tips are provided at https://www.python.org/about/gettingstarted/ (https://www.python.org/about/gettingstarted/).

Command lines and markdown text

The notebook has two elements (1) command lines and (2) Markdown text, which you are reading now. Command lines are executed by selecting the line and either selecting the Run command that appears in the menu or using a shortcut (displayed in the Help dropdown).

The first command line in this notebook is the print command you see right below. To get started, you can run the first line, which will print a welcome message right below it.

```
In [26]: print ("Hello, group researcher!")
```

```
Hello, group researcher!
```

Implementation Example

The implementation example involves time-variant group interaction data (i.e., "Chat 1" and "Chat 2") and time-invariant data (i.e., Attributes). The text data from each interval have been analyzed by the latest version of the Linguistic Inquiry and Word Count automated text analysis tool, LIWC-22. The resulting data files are:

- Attributes.csv
- LIWC-22 Results - Chat 1 - LIWC Analysis.csv
- LIWC-22 Results - Chat 2 - LIWC Analysis.csv

To combine these data into a wide-format you need to change the labels for the LIWC variables, such as ppron, to differentiate the data collection occasion (e.g., ppron_1 and ppron_2 to indicate the measure of personal pronoun during the first and second group chat, respectively). As detailed below, this notebook takes you through the following steps:

1. Read each csv file and make the individual identifiers match, which involves getting rid of the ".docx" in the filename column in the LIWC output
2. Relabel time-variant data with a suffix to denote the time interval, e.g. ppron_1, ppron_2
3. Use the unique identifier to combine the time-invariant data and the time-variant data into one file

Preparation

Run the following command to read pandas (which is an open-source data analysis tool). If you have not yet installed pandas, do that now from https://pandas.pydata.org/ (https://pandas.pydata.org/) and then run the next line.

```
In [27]: import pandas as pd
```

Your source csv files and this notebook need to be in the same directory. To make this happen, run the next command to see which directory you are in and thus where to put the source csv files.

```
In [ ]: !pwd
```

After you have put your csv files into the above directory, confirm that they are there by running the next command that lists all file items in the directory.

```
In [ ]: %ls
```

1. Read the CSV files

Time-invariant data file

Read the group data that do not change over time (Attribute.csv), put them in a data frame (labeled "df0"), and assign the unique identifier (MemberID) as an index variable.

In [28]:

```
filename0 = 'Attributes.csv'
df0 = pd.read_csv(filename0, index_col='MemberID')
```

Execute the below command to show an excerpt of the data frame formed by the first five rows and columns.

In [29]:

```
df0.iloc[:5,:5]
```

Out[29]:

	IndID	GroupID
MemberID		
1027103494blueduck	1	1027103494
1027103494indigopanda	2	1027103494
1027103494siennafly	3	1027103494
1037988226blueshark	4	1037988226
1037988226indigodog	5	1037988226

Time-variant data files

Chat 1 LIWC File

Read the time 1 data and put them in a data frame (df1).

In [30]:

```
filename1 = 'LIWC-22 Results - Chat 1 - LIWC Analysis.csv'
df1 = pd.read_csv(filename1)
```

Execute the below command, and it will show an excerpt of the data frame formed by the first five rows and columns.

```
In [31]: df1.iloc[:5,:5]
```

Out[31]:

	Filename	Segment	WC	Analytic	Clout
0	1027103494blueduck.docx	1	99	30.65	1.00
1	1027103494indigopanda.docx	1	105	62.44	13.30
2	1027103494siennafly.docx	1	128	45.74	58.66
3	1037988226blueshark.docx	1	72	8.44	72.07
4	1037988226indigodog.docx	1	145	35.73	19.09

When we prepared the group interaction data for LIWC analysis, we used the individual identifier (MemberID) as the filename for each individual's transcripts. That identifier appears in the Filename column in the LIWC output, but it has the file extension ".docx" appended at the end.

To match cases in the Attributes file and these LIWC transcript outputs, we need to get rid of the ".docx" in each entry in the LIWC output, which is done by first defining the column in the dataframe.

```
In [32]: filename = df1['Filename']
```

We can now check that this worked by checking that filename is indeed what we expected.

```
In [33]: filename
```

```
Out[33]: 0           1027103494blueduck.docx
         1        1027103494indigopanda.docx
         2          1027103494siennafly.docx
         3          1037988226blueshark.docx
         4          1037988226indigodog.docx
                           ...
         322        99575806siennamouse.docx
         323         99575806teallizard.docx
         324     997021119chartreusedog.docx
         325         997021119siennafly.docx
         326           997021119tealdog.docx
         Name: Filename, Length: 327, dtype: object
```

Execute the below to get rid of the '.docx' in each Filename entry. You will get a warning that you should ignore because we want to change this 'slice' of the dataframe.

```
In [34]: for i in range(len(filename)):
             filename[i] = filename.values[i].split(".")[0]
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y
```

Execute the below command to see what is displayed in the Filename column and the rest of the dataframe.

```
In [35]: df1
```

Out[35]:

	Filename	Segment	WC	Analytic	Clout	Authentic	Tone	WPS	BigWorc
0	1027103494blueduck	1	99	30.65	1.00	86.68	99.00	14.14	18.1
1	1027103494indigopanda	1	105	62.44	13.30	99.00	99.00	6.56	17.1
2	1027103494siennafly	1	128	45.74	58.66	23.08	99.00	9.14	13.2
3	1037988226blueshark	1	72	8.44	72.07	30.29	40.61	18.00	22.2
4	1037988226indigodog	1	145	35.73	19.09	53.63	63.69	14.50	22.0
...	...	...	...	...	...	...	...	...	
322	99575806siennamouse	1	107	21.20	76.84	47.36	99.00	7.64	16.8
323	99575806teallizard	1	76	29.44	7.50	97.49	99.00	5.85	21.0
324	997021119chartreusedog	1	57	41.78	1.83	87.88	99.00	11.40	15.7
325	997021119siennafly	1	110	15.41	11.33	50.45	98.85	13.75	18.1
326	997021119tealdog	1	54	16.08	2.75	77.17	99.00	13.50	20.3

327 rows × 119 columns

Having confirmed that the 'Filename' column is the unique identifier, make it the index variable.

```
In [36]: df1 = df1.set_index('Filename')
```

Chat 2 LIWC File

Read the time 2 data and put them in a data frame (df2).

```
In [37]: filename2 = 'LIWC-22 Results - Chat 2 - LIWC Analysis.csv'
         df2 = pd.read_csv(filename2)
```

Execute the below command to show an excerpt of the data frame formed by the first five rows and columns.

```
In [38]: df2.iloc[:5,:5]
```

Out[38]:

	Filename	Segment	WC	Analytic	Clout
0	1027103494blueduck.docx	1	117	48.38	10.01
1	1027103494indigopanda.docx	1	205	55.80	18.06
2	1027103494siennafly.docx	1	160	60.29	36.48
3	1037988226blueshark.docx	1	146	30.50	75.09
4	1037988226indigodog.docx	1	138	20.27	79.94

Execute the below to get rid of the '.docx' in each Filename entry. You will get a warning that you should ignore because we want to change this 'slice' of the dataframe.

```
In [39]: filename = df2['Filename']
         for i in range(len(filename)):
             filename[i] = filename.values[i].split(".")[0]
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y
  This is separate from the ipykernel package so we can avoid doing
imports until
```

Execute the below command to see what is displayed in the Filename column and the rest of the dataframe.

```
In [40]: df2
```

```
Out[40]:
```

	Filename	Segment	WC	Analytic	Clout	Authentic	Tone	WPS	BigWorc
0	1027103494blueduck	1	117	48.38	10.01	88.67	91.40	9.75	17.0
1	1027103494indigopanda	1	205	55.80	18.06	88.99	58.42	7.59	18.0
2	1027103494siennafly	1	160	60.29	36.48	67.52	59.44	7.62	10.6
3	1037988226blueshark	1	146	30.50	75.09	28.97	40.28	10.43	19.1
4	1037988226indigodog	1	138	20.27	79.94	70.05	76.45	6.57	10.8
...	...	...	...	...	...	...	...	...	
322	99575806siennamouse	1	243	39.22	62.05	35.34	58.91	9.00	12.7
323	99575806teallizard	1	173	21.05	30.40	60.97	74.36	9.11	20.2
324	997021119chartreusedog	1	122	27.14	59.57	37.69	58.75	9.38	20.4
325	997021119siennafly	1	226	61.14	77.27	74.11	80.91	16.14	16.8
326	997021119tealdog	1	85	16.49	3.27	99.00	37.12	9.44	21.1

327 rows × 119 columns

Having confirmed that the 'Filename' column is the unique identifier, make it the index variable.

```
In [41]: df2 = df2.set_index('Filename')
```

2. Relabel time-variant data to denote their interval

Add an underscore 1 to the end of each variable name in the time 1 dataframe.

```
In [42]: df1.columns = df1.columns + '_1'
```

Execute the below command to show an excerpt of the data frame formed by the first five rows and columns.

```
In [43]: df1.iloc[:5,:5]
```

Out[43]:

	Segment_1	WC_1	Analytic_1	Clout_1	Authentic_1
Filename					
1027103494blueduck	1	99	30.65	1.00	86.68
1027103494indigopanda	1	105	62.44	13.30	99.00
1027103494siennafly	1	128	45.74	58.66	23.08
1037988226blueshark	1	72	8.44	72.07	30.29
1037988226indigodog	1	145	35.73	19.09	53.63

Add an underscore 2 to the end of each variable name in the time 2 data frame.

```
In [44]: df2.columns = df2.columns + '_2'
```

Execute the below command to show an excerpt of the data frame formed by the first five rows and columns.

```
In [45]: df2.iloc[:5,:5]
```

Out[45]:

	Segment_2	WC_2	Analytic_2	Clout_2	Authentic_2
Filename					
1027103494blueduck	1	117	48.38	10.01	88.67
1027103494indigopanda	1	205	55.80	18.06	88.99
1027103494siennafly	1	160	60.29	36.48	67.52
1037988226blueshark	1	146	30.50	75.09	28.97
1037988226indigodog	1	138	20.27	79.94	70.05

3. Combine into one file with relabeled time-variant data

Create one file (CombinedData) with the unique identifier, the time-invariant attribute data, the time 1 LIWC results, and the time 2 LIWC results.

```
In [46]: CombinedData = pd.concat([df0,df1,df2],axis=1)
```

Execute the below command to show an excerpt of the combined dataframe formed by the first five rows and a few columns selected from the attribute data (columns 0 and 1), the time 1 data (columns 3, 13) and the time 2 data (columns 121,132).

In [47]:
```
CombinedData.iloc[:5,[0,1,3,14,121,132]]
```

Out[47]:

	IndID	GroupID	WC_1	ppron_1	WC_2	ppron_2
1027103494blueduck	1	1027103494	99	11.11	117	9.40
1027103494indigopanda	2	1027103494	105	14.29	205	11.22
1027103494siennafly	3	1027103494	128	5.47	160	10.63
1037988226blueshark	4	1037988226	72	6.94	146	6.16
1037988226indigodog	5	1037988226	145	6.21	138	10.87

To output the merged data into a csv file, execute the following command.

In [48]:
```
CombinedData.to_csv('WideCombinedData.csv')
```

This csv file will be in your directory. If you need a reminder of which one that is, execute the below command.

In [ ]:
```
!pwd
```

Interested in customizing this notebook?

You can customize the notebook by adding notes or new commands. To do so, select an area with Markdown text, hit the esc key. After that, you can change a cell to Markdown by hitting the m key, or you can change a cell to Code by hitting the y key.

If you need help installing and undertanding Jupyter notebooks, this 30-minute video is a great resource https://www.youtube.com/watch?v=HW29067qVWk (https://www.youtube.com/watch?v=HW29067qVWk).