

Week 10 Tutorial: Data Wrangling in Python

POP77001 Computer Programming for Social Scientists

Loading the Dataset

- Replace filepath with the location of the file on your computer

```
1 import pandas as pd

1 # This time let's skip the 2nd row, which contains questions
2 PATH = '../data/kaggle_survey_2022_responses.csv'
3
4 kaggle2022 = pd.read_csv(PATH, skiprows = [1])
5 kaggle2022.head(n = 1)
```

	Duration (in seconds)	Q2	Q3	Q4	Q5	Q6_1	Q6_2	Q6_3	Q6_4	Q6_5	...	Q44_
0	121	30-34	Man	India	No	NaN	NaN	NaN	NaN	NaN	...	NaN

1 rows × 296 columns

```
1 # We will load the questions as a separate dataset
2 kaggle2022_qs = pd.read_csv(PATH, nrows = 1)
3 kaggle2022_qs
```

	Duration (in seconds)	Q2	Q3	Q4	Q5	Q6_1	Q6_2
0	Duration (in seconds)	What is your age (# years)?	What is your gender? - Selected Choice	In which country do you currently reside?	Are you currently a student? (high school, uni...	On which platforms have you begun or completed...	On which platforms have you begun or completed...

1 rows × 296 columns

Exercise: Categorical Variables

- Load the dataset (as local file)
- Consider country of residence reported by respondents (question Q4).
- Make sure you can select the column both using label and index.
- Calculate the percentages of top 3 countries of residence in the sample.

Crosstabulation in **pandas**

```
1 # Calculate crosstabulation between 'Age group' (Q2) and 'Gender' (Q3)
2 pd.crosstab(kaggle2022['Q2'], kaggle2022['Q3'])
```

Q3	Man	Nonbinary	Prefer not to say	Prefer to self-describe	Woman
Q2					
18-21	3310	13	69	7	1160
22-24	3168	15	50	6	1044
25-29	3425	14	56	6	971
30-34	2248	12	43	6	663
35-39	1791	6	36	1	519
40-44	1480	6	29	2	410
45-49	997	6	10	1	239
50-54	759	0	14	1	140
55-59	506	3	13	0	89
60-69	470	3	10	1	42

Q3	Man	Nonbinary	Prefer not to say	Prefer to self-describe	Woman
Q2					
70+	112	0	4	2	9

Margins in Crosstab

```
1 # It is often useful to see the proportions/percentages rather than raw counts
2 pd.crosstab(kaggle2022['Q2'], kaggle2022['Q3'], normalize = 'columns')
```

Q3	Man	Nonbinary	Prefer not to say	Prefer to self-describe	Woman
Q2					
18-21	0.181211	0.166667	0.206587	0.212121	0.219448
22-24	0.173437	0.192308	0.149701	0.181818	0.197503
25-29	0.187507	0.179487	0.167665	0.181818	0.183693
30-34	0.123070	0.153846	0.128743	0.181818	0.125426
35-39	0.098051	0.076923	0.107784	0.030303	0.098184
40-44	0.081025	0.076923	0.086826	0.060606	0.077563
45-49	0.054582	0.076923	0.029940	0.030303	0.045214
50-54	0.041553	0.000000	0.041916	0.030303	0.026485
55-59	0.027702	0.038462	0.038922	0.000000	0.016837
60-69	0.025731	0.038462	0.029940	0.030303	0.007946

Q3	Man	Nonbinary	Prefer not to say	Prefer to self-describe	Woman
Q2					
70+	0.006132	0.000000	0.011976	0.060606	0.001703

Crosstab with `pivot_table`

```
1 # For `values` variable we use `Q4`, but any other would work equally well
2 pd.pivot_table(
3     kaggle2022,
4     index = 'Q2',
5     columns = 'Q3',
6     values = 'Q4',
7     aggfunc = 'count',
8     fill_value = 0
9 )
```

Q3	Man	Nonbinary	Prefer not to say	Prefer to self-describe	Woman
Q2					
18-21	3310	13	69	7	1160
22-24	3168	15	50	6	1044
25-29	3425	14	56	6	971
30-34	2248	12	43	6	663
35-39	1791	6	36	1	519
40-44	1480	6	29	2	410
45-49	997	6	10	1	239

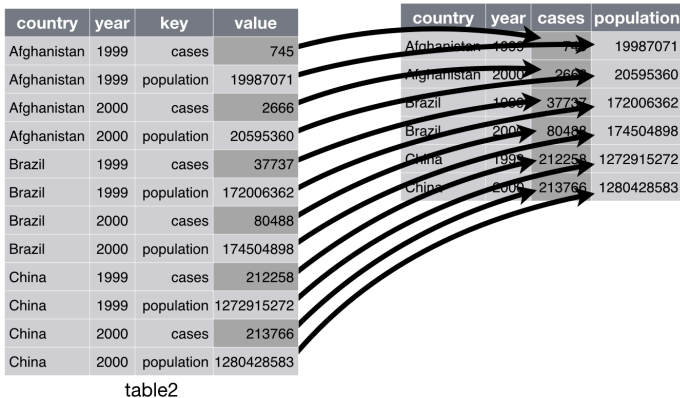
Q3	Man	Nonbinary	Prefer not to say	Prefer to self-describe	Woman
Q2					
50-54	759	0	14	1	140
55-59	506	3	13	0	89
60-69	470	3	10	1	42
70+	112	0	4	2	9

Exercise: Manipulating Columns

- Let's take a look at the first column of the dataset.
- It lists the time it took respondents to complete the survey (in seconds).
- First, change column's long name to `duration_min`.
- Now modify the column such that it shows time in minutes.
- Filter dataset leaving only respondents who took more than 3 mins to respond.
- How many are dropped?

Pivoting Data in pandas

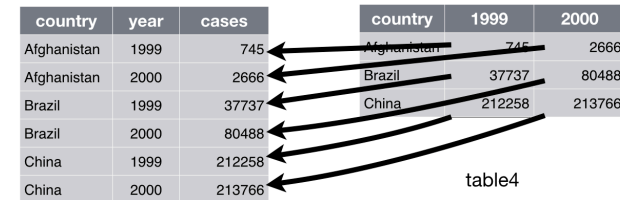
- Recall pivoting from R.
- The two main operations are:
 - Spreading some variable across columns (`pd.DataFrame.pivot()`)
 - Gathering some columns in a variable pair (`pd.DataFrame.melt()`)



country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2

`pd.DataFrame.pivot()`



country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

table4

`pd.DataFrame.melt()`

Pivoting Data to Long

```
1 df_wide = pd.DataFrame({
2     'country': ['Afghanistan', 'Brazil'],
3     '1999': [745, 2666],
4     '2000': [37737, 80488]
5 })
6 df_wide
```

	country	1999	2000
0	Afghanistan	745	37737
1	Brazil	2666	80488

```
1 # Pivoting longer
2 df_long = df_wide.melt(
3     id_vars = 'country',
4     var_name = 'year',
5     value_name = 'cases'
6 )
7 df_long
```

	country	year	cases
0	Afghanistan	1999	745
1	Brazil	1999	2666
2	Afghanistan	2000	37737

Pivoting Data to Wide

```
1 # Pivoting wider
2 df_wide = df_long.pivot(
3     index = 'country',
4     columns = 'year',
5     values = 'cases'
6 )
7 df_wide
```

	year	1999	2000
	country		
	Afghanistan	745	37737
	Brazil	2666	80488

```
1 # As using pivot creates an index from
2 # the column used as the row labels, we
3 # may want to use reset_index to move
4 # the data back into a column
5 df_wide.reset_index()
```

	year	country	1999	2000
0		Afghanistan	745	37737
1		Brazil	2666	80488

Week 10 Exercise (unassessed)

- Try replicating Exercise 5 from Assignment 2 using `pandas`.
- You can use `pd.DataFrame.isna()` or `pd.DataFrame.notna()` for filtering.