Week 10 Tutorial: Data Wrangling in Python

POP77001 Computer Programming for Social Scientists

Module website: tinyurl.com/POP77001

Loading the dataset

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

Loading the dataset

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

In [1]:

```
import pandas as pd
```

Loading the dataset

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

In [1]:

```
import pandas as pd
```

In [2]:

```
# This time let's skip the 2nd row, which contains questions
PATH = '../data/kaggle_survey_2021_responses.csv'

kaggle2021 = pd.read_csv(PATH, skiprows = [1])
kaggle2021.head(n = 1)
```

```
/tmp/ipykernel_272500/798188422.py:4: DtypeWarning: Columns (19
5,201) have mixed types. Specify dtype option on import or set
low_memory=False.
  kaggle2021 = pd.read_csv(PATH, skiprows = [1])
```

Out[2]:

	Time from Start to Finish (seconds)	Q1	Q2	Q3	Q4	Q5	Q6	Q7_Part_1	Q7_Part_2
0	910	50-54	Man	India	Bachelor’s degree	Other	5-10 years	Python	R

1 rows × 369 columns

In [3]:

```
# We will load the questions as a separate dataset
kaggle2021_qs = pd.read_csv(PATH, nrows = 1)
kaggle2021_qs
```

Out[3]:

	Time from Start to Finish (seconds)	Q1	Q2	Q3	Q4	Q5	Q6	Q7_Part_
0	Duration (in seconds)	What is your age (# years)?	What is your gender? - Selected Choice	In which country do you currently reside?	What is the highest level of formal education ...	Select the title most similar to your current ...	For how many years have you been writing code ...	Wha programmin language do you us on a reg.

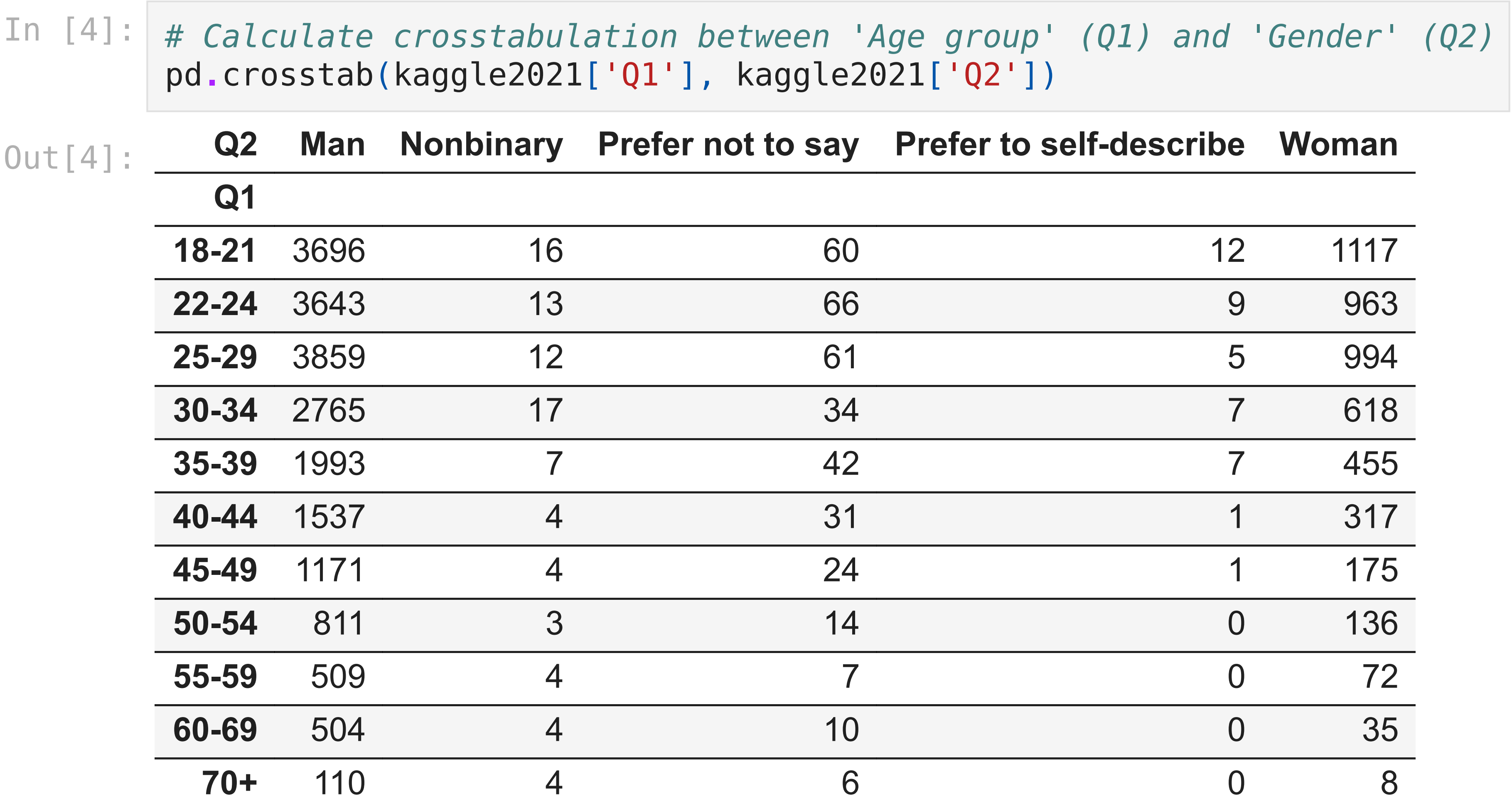
1 rows × 369 columns

Exercise 1: Summarise categorical variable

- Load the dataset (as local file)
- Consider country of residence reported by respondents (question Q3).
- 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

Crosstabulation in pandas



In [4]:

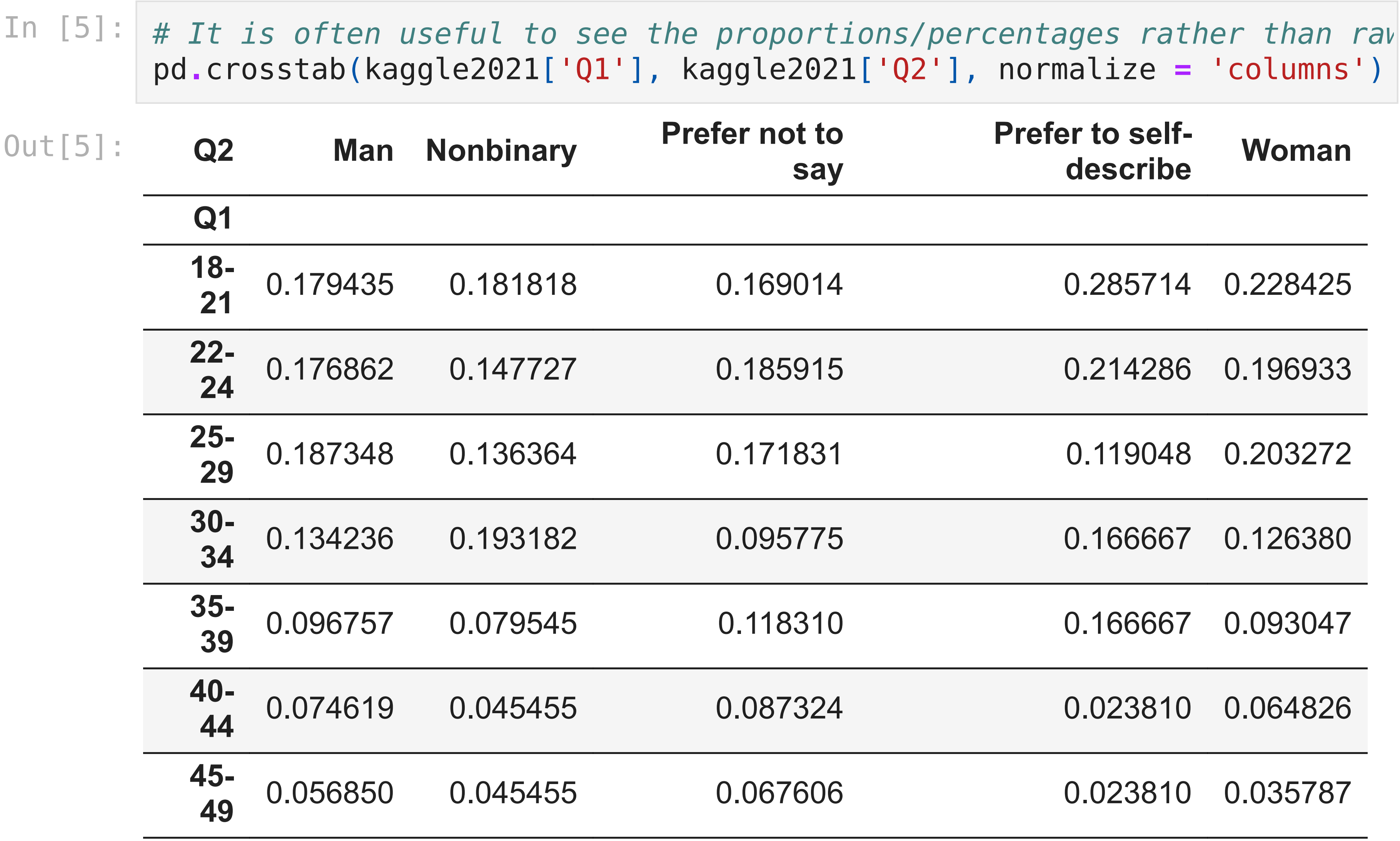
```
# Calculate crosstabulation between 'Age group' (Q1) and 'Gender' (Q2)
pd.crosstab(kaggle2021['Q1'], kaggle2021['Q2'])
```

Out[4]:

Q2	Man	Nonbinary	Prefer not to say	Prefer to self-describe	Woman
Q1					
18-21	3696	16	60	12	1117
22-24	3643	13	66	9	963
25-29	3859	12	61	5	994
30-34	2765	17	34	7	618
35-39	1993	7	42	7	455
40-44	1537	4	31	1	317
45-49	1171	4	24	1	175
50-54	811	3	14	0	136
55-59	509	4	7	0	72
60-69	504	4	10	0	35
70+	110	4	6	0	8

Margins in crosstab

Margins in crosstab



In [5]:

```
# It is often useful to see the proportions/percentages rather than rav
pd.crosstab(kaggle2021['Q1'], kaggle2021['Q2'], normalize = 'columns')
```

Out[5]:

Q2	Man	Nonbinary	Prefer not to say	Prefer to self-describe	Woman
Q1					
18-21	0.179435	0.181818	0.169014	0.285714	0.228425
22-24	0.176862	0.147727	0.185915	0.214286	0.196933
25-29	0.187348	0.136364	0.171831	0.119048	0.203272
30-34	0.134236	0.193182	0.095775	0.166667	0.126380
35-39	0.096757	0.079545	0.118310	0.166667	0.093047
40-44	0.074619	0.045455	0.087324	0.023810	0.064826
45-49	0.056850	0.045455	0.067606	0.023810	0.035787

Crosstabulation with `pivot_table`

Crosstabulation with `pivot_table`

```
In [6]: # For `values` variable we use `Q3`, but any other would work equally w
        pd.pivot_table(kaggle2021, index = 'Q1', columns = 'Q2', values = 'Q3',
```

Out[6]:

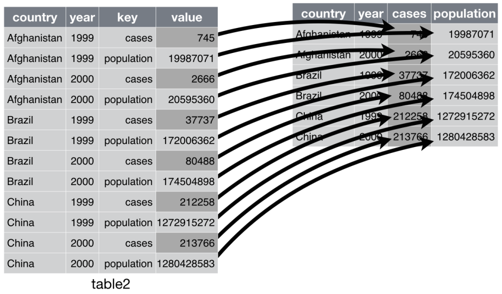
Q2	Man	Nonbinary	Prefer not to say	Prefer to self-describe	Woman
Q1					
18-21	3696	16	60	12	1117
22-24	3643	13	66	9	963
25-29	3859	12	61	5	994
30-34	2765	17	34	7	618
35-39	1993	7	42	7	455
40-44	1537	4	31	1	317
45-49	1171	4	24	1	175
50-54	811	3	14	0	136
55-59	509	4	7	0	72
60-69	504	4	10	0	35
70+	110	4	6	0	8

Exercise 2: 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

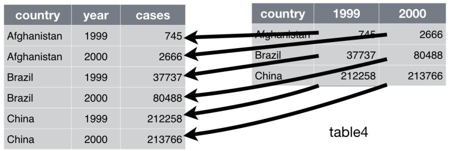
- Recall pivoting from R.
- The two main operations are:
  - Spreading some variable across columns (`pd.DataFrame.pivot()`)
  - Gathering some columns in a variable pair (`pd.DataFrame.melt()`)



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

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



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

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

table4

pd.DataFrame.pivot()	pd.DataFrame.melt()

Source: R for Data Science

Pivoting data example

Pivoting data example

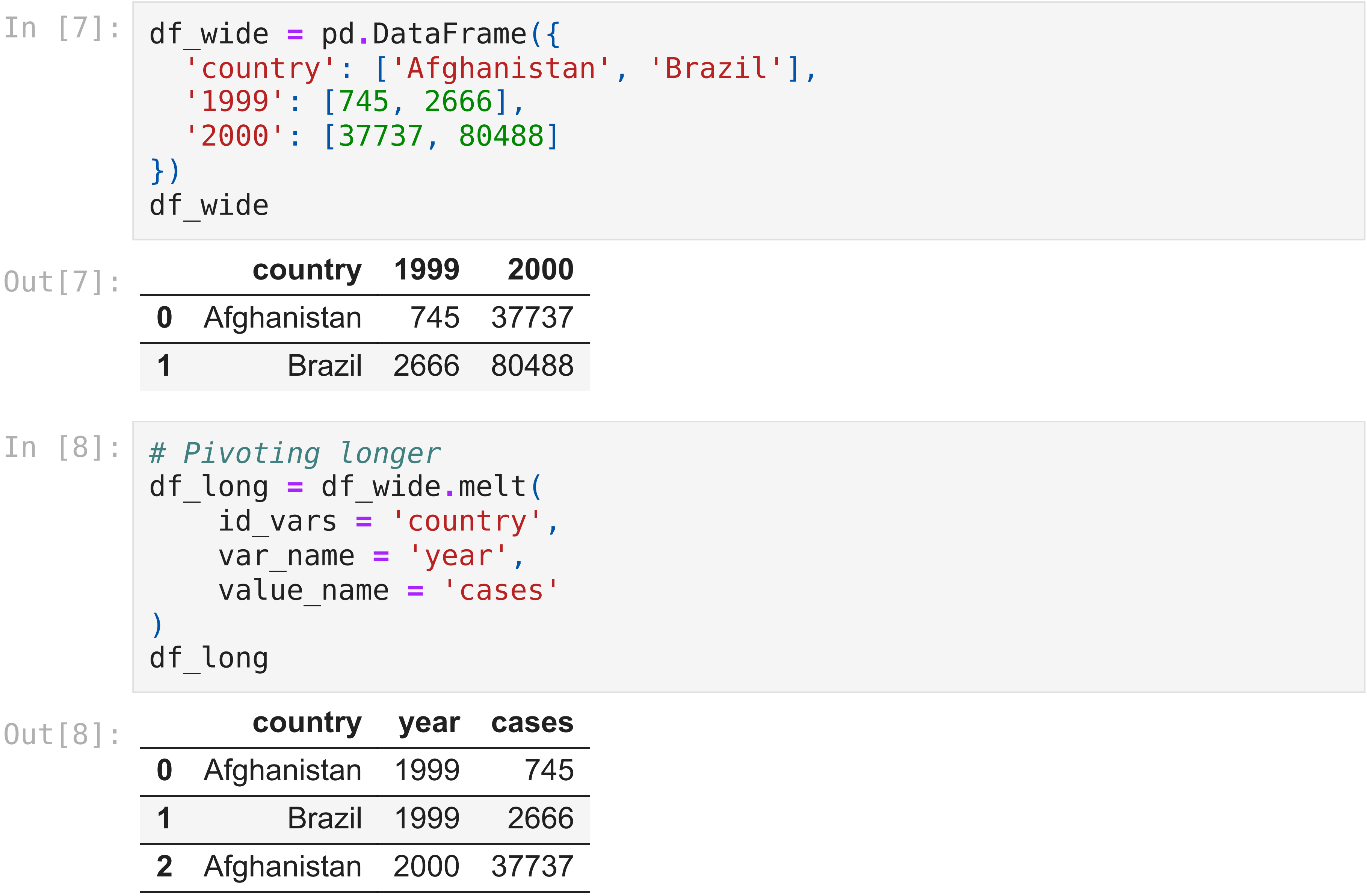
In [7]:

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

Out[7]:

	country	1999	2000
0	Afghanistan	745	37737
1	Brazil	2666	80488

Pivoting data example



In [7]:

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

Out[7]:

	country	1999	2000
0	Afghanistan	745	37737
1	Brazil	2666	80488

In [8]:

```
# Pivoting longer
df_long = df_wide.melt(
    id_vars = 'country',
    var_name = 'year',
    value_name = 'cases'
)
df_long
```

Out[8]:

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

	country	year	cases
3	Brazil	2000	80488

Pivoting data example continued

Pivoting data example continued

```
In [9]:  # Pivoting wider
         df_wide = df_long.pivot(
             index = 'country',
             columns = 'year',
             values = 'cases'
         )
         df_wide
```

```
Out[9]:
               year  1999   2000
            country
        Afghanistan   745  37737
             Brazil  2666  80488
```

Pivoting data example continued

In [9]:

```
# Pivoting wider
df_wide = df_long.pivot(
    index = 'country',
    columns = 'year',
    values = 'cases'
)
df_wide
```

Out[9]:

```
       year   1999   2000
    country
Afghanistan    745  37737
     Brazil   2666  80488
```

In [10]:

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

Out[10]:

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

Exercise 3: Pivoting

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

Week 10 Exercise (unassessed)

- Exercise 3: Pivoting