This notebook is an exercise in the [Data Cleaning (https://www.kaggle.com/learn/data-cleaning)](https://www.kaggle.com/learn/data-cleaning) course. You can reference the tutorial at [this link (https://www.kaggle.com/alexisbcook/inconsistent-data-entry)](https://www.kaggle.com/alexisbcook/inconsistent-data-entry).

In this exercise, you'll apply what you learned in the **Inconsistent data entry** tutorial.

Setup

The questions below will give you feedback on your work. Run the following cell to set up the feedback system.

In [3]:

```
from learntools.core import binder
binder.bind(globals())
from learntools.data_cleaning.ex5 import *
print("Setup Complete")
```

```
Setup Complete
```

Get our environment set up

The first thing we'll need to do is load in the libraries and dataset we'll be using. We use the same dataset from the tutorial.

In [4]:

```
# modules we'll use
import pandas as pd
import numpy as np

# helpful modules
import fuzzywuzzy
from fuzzywuzzy import process
import charset_normalizer

# read in all our data
professors = pd.read_csv("../input/pakistan-intellectual-capital/pakistan_inte

# set seed for reproducibility
np.random.seed(0)
```

Next, we'll redo all of the work that we did in the tutorial.

In [5]:
```
# convert to lower case
professors['Country'] = professors['Country'].str.lower()
# remove trailing white spaces
professors['Country'] = professors['Country'].str.strip()

# get the top 10 closest matches to "south korea"
countries = professors['Country'].unique()
matches = fuzzywuzzy.process.extract("south korea", countries, limit=10, score

def replace_matches_in_column(df, column, string_to_match, min_ratio = 47):
    # get a list of unique strings
    strings = df[column].unique()

    # get the top 10 closest matches to our input string
    matches = fuzzywuzzy.process.extract(string_to_match, strings,
                                         limit=10, scorer=fuzzywuzzy.fuzz.toke

    # only get matches with a ratio > 90
    close_matches = [matches[0] for matches in matches if matches[1] >= min_ra

    # get the rows of all the close matches in our dataframe
    rows_with_matches = df[column].isin(close_matches)

    # replace all rows with close matches with the input matches
    df.loc[rows_with_matches, column] = string_to_match

    # let us know the function's done
    print("All done!")

replace_matches_in_column(df=professors, column='Country', string_to_match="so
countries = professors['Country'].unique()
```

```
All done!
```

1) Examine another column

Write code below to take a look at all the unique values in the "Graduated from" column.

In [6]:
```
# TODO: Your code here
```

Do you notice any inconsistencies in the data? Can any of the inconsistencies in the data be fixed by removing white spaces at the beginning and end of cells?

Once you have answered these questions, run the code cell below to get credit for your work.

In [7]:

```
# Check your answer (Run this code cell to receive credit!)
q1.check()
```

Correct:

There are inconsistencies that can be fixed by removing white spaces at the beginning and end of cells. For instance, "University of Central Florida" and " University of Central Florida" both appear in the column.

In [8]:

```
# Line below will give you a hint
#q1.hint()
```

2) Do some text pre-processing

Convert every entry in the "Graduated from" column in the `professors` DataFrame to remove white spaces at the beginning and end of cells.

In [9]:

```
# TODO: Your code here
professors["Graduated from"] = professors["Graduated from"].str.strip()


# Check your answer
q2.check()
```

Correct

In [10]:

```
# Lines below will give you a hint or solution code
#q2.hint()
#q2.solution()
```

3) Continue working with countries

In the tutorial, we focused on cleaning up inconsistencies in the "Country" column. Run the code cell below to view the list of unique values that we ended with.

In [11]:

```
# get all the unique values in the 'City' column
countries = professors['Country'].unique()

# sort them alphabetically and then take a closer look
countries.sort()
countries
```

Out[11]:

```
array(['australia', 'austria', 'canada', 'china', 'finland', 'france',
       'germany', 'greece', 'hongkong', 'ireland', 'italy', 'japan',
       'macau', 'malaysia', 'mauritius', 'netherland', 'new zealand',
       'norway', 'pakistan', 'portugal', 'russian federation',
       'saudi arabia', 'scotland', 'singapore', 'south korea', 'spain',
       'sweden', 'thailand', 'turkey', 'uk', 'urbana', 'usa', 'usofa'],
      dtype=object)
```

Take another look at the "Country" column and see if there's any more data cleaning we need to do.

It looks like 'usa' and 'usofa' should be the same country. Correct the "Country" column in the dataframe to replace 'usofa' with 'usa'.

Use the most recent version of the DataFrame (with the whitespaces at the beginning and end of cells removed) from question 2.

In [12]:

```
# TODO: Your code here!
professors.loc[professors["Country"] == "usofa","Country"] = "usa"

#matches = fuzzywuzzy.process.extract("usa", countries, limit=10, scorer=fuzzyw
#replace_matches_in_column(df=professors, column='Country', string_to_match="us

print(professors["Country"].unique())

# Check your answer
q3.check()
```

```
['thailand' 'pakistan' 'germany' 'austria' 'australia' 'uk' 'china'
 'france' 'usa' 'south korea' 'malaysia' 'sweden' 'italy' 'canada'
 'norway' 'ireland' 'new zealand' 'urbana' 'portugal' 'russian federation'
 'finland' 'netherland' 'greece' 'turkey' 'macau' 'singapore' 'spain'
 'japan' 'hongkong' 'saudi arabia' 'mauritius' 'scotland']
```

Correct

In [13]:

```
# Lines below will give you a hint or solution code
#q3.hint()
#q3.solution()
```

Congratulations!

Congratulations for completing the **Data Cleaning** course on Kaggle Learn!

To practice your new skills, you're encouraged to download and investigate some of Kaggle's Datasets (https://www.kaggle.com/datasets).

Have questions or comments? Visit the course discussion forum (https://www.kaggle.com/learn/data-cleaning/discussion) to chat with other learners.