In this tutorial, we'll leverage Python's Pandas and NumPy libraries to clean data. We'll cover the following:

- Dropping unnecessary columns in a `DataFrame`
- Changing the index of a `DataFrame`
- Using `.str()` methods to clean columns
- Using the `DataFrame.applymap()` function to clean the entire dataset, element-wise
- Renaming columns to a more recognizable set of labels
- Skipping unnecessary rows in a CSV file

```
(vpanda) C:\virtualenv\vpanda>pip install pandas
(vpanda) C:\virtualenv\vpanda>pip install matplotlib

(vpanda) C:\virtualenv\vpanda>python
```

```
>>> import pandas as pd
>>> import numpy as np

>>> df = pd.read_csv('2016-pydata-carolinas-pandas/data/gapminder.tsv', sep='\t')
>>> df.head()
>>> type(df)
>>> df.shape
>>> df.columns
>>> df.dtypes

>>> country_df = df['country']
>>> subset = df[['country','continent','year']]

>>> row_100 = df.loc[99]
>>> type(row_100)
<class 'pandas.core.series.Series'>

>>> df.iloc[0]
>>> df.ix[[0, 99, 999]]
>>> df.ix[[0, 99, 999]]

### df.ix[rows, 'columns']
>>> df.ix[0, 'continent']
>>> df.ix[[0, 99, 999], ['continent', 'year']]

>>> df.groupby('year')
<pandas.core.groupby.groupby.DataFrameGroupBy object at 0x000001F492D233C8>

>>> df.groupby('year')['lifeExp'].mean()
>>> df.groupby(['year', 'continent'])['lifeExp'].mean()
>>> # step 1 - create boolean Series
>>> criteria = df['year'] == 1952
>>> continen = df['continent'] == ‘Asia’
>>> # step 2 - do boolean selection
>>> df[criteria].head()
>>> df[continen].head()
```

```
>>> x = [1,2,3]
>>> y = [1,4,9]
>>> z = [10,5,0]
>>>
>>> plt.plot(x,y)
[<matplotlib.lines.Line2D object at 0x000001BB7B46A7B8>]
>>> plt.plot(x,z)
[<matplotlib.lines.Line2D object at 0x000001BB6DFBC160>]
>>> plt.title("test plot")
Text(0.5, 1.0, 'test plot')
>>> plt.xlabel("x")
Text(0.5, 0, 'x')
>>> plt.ylabel("y and z")
Text(0, 0.5, 'y and z')
>>> plt.legend(["this is y", "this is z"])
>>> plt.show()
```

```
>>> data = pd.read_csv("panda-visualization/countries.csv")

>>> plt.plot(us.year, us.population / 10**6)
>>> plt.plot(china.year, china.population / 10**6)
>>> plt.xlabel('year')
>>> plt.ylabel('population')

>>> plt.plot(us.year, us.population / us.population.iloc[0] * 100)
>>> plt.plot(china.year, china.population / china.population.iloc[0] * 100)
>>> plt.legend(['United States', 'China'])
>>> plt.show()
```